Florian Brinkmann
Simon Kraft

Inkl. Website-Editor



WordPress 6

Das umfassende Handbuch

- Installation, Einrichtung und Einsatz für Websites und Blogs
- Wartung, Sicherheit, Performance- und Suchmaschinenoptimierung
- Inkl. WooCommerce für erfolgreiche Webshops

Alle Beispielprojekte zum Download

Rheinwerk Computing

Florian Brinkmann, Simon Kraft

WordPress 6

Das umfassende Handbuch

3., aktualisierte und erweiterte Auflage 2025

Rheinwerk

Impressum

Dieses E-Book ist ein Verlagsprodukt, an dem viele mitgewirkt haben, insbesondere:

Lektorat Stephan Matteschеck
Korrektorat Isolde Kommer, Großerlach
Covergestaltung Bastian Illerhaus
Herstellung E-Book Stefanie Häb
Satz E-Book SatzPro, Krefeld

Bibliografische Information der Deutschen Nationalbibliothek: Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über http://dnb.dnb.de abrufbar.

ISBN 978-3-367-10165-8

3., aktualisierte und erweiterte Auflage 2025
© Rheinwerk Verlag GmbH, Bonn 2025

Rheinwerk Verlag GmbH • Rheinwerkallee 4 • 53227 Bonn
service@rheinwerk-verlag.de

Liebe Leserin, lieber Leser,

ungefähr ein Drittel aller Websites weltweit nutzen WordPress. Das ist eine beeindruckende Zahl und ein verdienter erster Platz unter den Content-Management-Systemen. Es ist mächtig, anpassungsfähig und dabei trotzdem leicht zu bedienen. So können Sie schnell ein einfaches Blog mit fertigen Themes erstellen oder einen ausgewachsenen, komplexen Webauftritt inklusive Shop und selbst entwickelten Plugins aufbauen. Den Grad der individuellen Anpassung bestimmen Sie selbst, und dieses Buch wird Sie in allen Phasen der Planung und des Betriebs Ihrer Website unterstützen. Neben den neuen Funktionalitäten, die WordPress 6.7 zu bieten hat, werden Sie auch weiterführende Themen wie SEO, Performance-Optimierung und Sicherheitsaspekte kennenlernen.

Der leider viel zu früh verstorbene ursprüngliche Autor, Richard Eisenmenger, hat mit seinem Werk zu vorherigen WordPress-Versionen den Grundstein für dieses Buch gelegt. In dieser neuen Auflage haben es Florian Brinkmann und Simon Kraft vollständig überarbeitet und auf den neusten Stand gebracht. Beide sind tief in der WordPress-Community verwurzelt und bringen ihre langjährige Erfahrung als Entwickler und Autoren in dieses Buch ein. Florian Brinkmann ist als Webentwickler tätig und schreibt unter anderem für sein eigenes WordPress-Entwicklerblog sowie für Fachmagazine wie t3n und KrautPress.de. Simon Kraft ist ebenfalls ein erfahrener Entwickler und engagiertes Mitglied der Community. Gemeinsam haben sie das Standardwerk sorgfältig aktualisiert, überarbeitet und

auf den neuesten Stand gebracht. Richard wäre stolz auf die neue Auflage.

Ein Wort noch in eigener Sache: Dieses Buch wurde mit großer Sorgfalt lektoriert und produziert. Sollten Sie dennoch Fehler finden oder inhaltliche Anregungen haben, scheuen Sie sich nicht, mit uns Kontakt aufzunehmen. Ihre Fragen und Änderungsvorschläge sind jederzeit willkommen.

Ich wünsche Ihnen viel Freude mit WordPress 6.7 und viel Erfolg bei der Umsetzung Ihrer Projekte!

Ihr Stephan Matteschеck
Lektorat Rheinwerk Computing

stephan.matteschеck@rheinwerk-verlag.de
www.rheinwerk-verlag.de
Rheinwerk Verlag • Rheinwerkallee 4 • 53227 Bonn

Inhaltsverzeichnis

Aus dem Lektorat
Inhaltsverzeichnis

Vorwort

1 Was Sie mit WordPress vorhaben und wo Sie jetzt weiterlesen

1.1 Das richtige Handbuch für Sie
1.1.1 Das Blog
1.1.2 Die persönliche Website
1.1.3 Die Unternehmens-Website
1.1.4 Der Online-Shop
1.1.5 Die Vereinswebsite
1.1.6 Sie wollen für WordPress programmieren
1.2 Eine kurze Geschichte von WordPress
1.3 Warum WordPress?
1.3.1 Schnell zum ersten veröffentlichten Inhalt

1.3.2 Fast unendliche Möglichkeiten
1.3.3 Hoheit über die eigenen Daten
1.3.4 Open Source
1.3.5 Server-Voraussetzungen für WordPress
1.3.6 Wann WordPress nicht die richtige Wahl ist
1.4 Anleitung zum Handbuch
1.5 Nach der letzten Seite
1.5.1 Dokumentation auf WordPress.org
1.5.2 Suchmaschine des Vertrauens
1.5.3 Dokumentation von Plugin- oder Theme-Herstellern
1.5.4 WordPress-Support-Foren
1.5.5 Buchtipps

Teil I WordPress installieren und bedienen

2 WordPress-Installation in wenigen Minuten mit Playground

2.1 WordPress Playground – eine ganze WordPress-Installation lokal im Browser
2.1.1 Plugins vorinstallieren
2.1.2 Theme vorinstallieren
2.1.3 Theme- und Plugin-Parameter kombinieren

2.1.4 Eine Playground-Website speichern und importieren
2.2 wp-now: Playground für Entwickler*innen
2.2.1 Installation von wp-now
2.2.2 WordPress-Instanz mit wp-now erstellen
2.2.3 Verfügbare Parameter zur Anpassung einer WordPress-Instanz
2.2.4 Speicherorte der Dateien
2.3 Playground mit Blueprints vorkonfigurieren
2.3.1 Blueprint erstellen
2.3.2 Blueprint anwenden
2.3.3 Verfügbare Blueprint-Optionen
2.4 Komfortablere Playground-Nutzung mit »Studio« von WordPress.com

3 WordPress-Installation – flexibel

3.1 Installation von Docker
3.2 Installation von Lando
3.3 Lando-Installation testen

4 Hoster-Auswahl und Installation

4.1 Welches Hosting darf's denn sein?

4.1.1 Welches Hosting passt zu Ihnen?

4.1.2 Was für ein Website-Projekt haben Sie vor sich?

4.1.3 Wo befindet sich das Publikum Ihrer Website?

4.1.4 Minimale Voraussetzungen für WordPress

4.1.5 Produktkategorien und Auswahl

4.2 Installation mit einem One-Click-Installer

5 Administration und Konfiguration von WordPress

5.1 Das Backend kennenlernen

5.1.1 Einloggen ins Backend

5.1.2 Die Admin-Oberfläche und das Dashboard

5.1.3 Das linke Admin-Menü

5.2 Einstellungen

5.2.1 Allgemein

5.2.2 Schreiben

5.2.3 Lesen

5.2.4 Diskussion

5.2.5 Medien

5.2.6 Permalinks

5.2.7 Datenschutz
5.3 Einstellungstipps für Fortgeschrittene
5.3.1 Die wp-config.php
5.3.2 Die options.php

6 Inhalte veröffentlichen

6.1 Beiträge schreiben/Seiten anlegen
6.1.1 Erster Blick in den Editor; Titel und Permalink festlegen
6.1.2 Hinweise zur Block-Editor-Bedienung
6.1.3 Alle Blöcke von WordPress 6.7
6.1.4 Verkleinern-Funktion nutzen
6.1.5 Zwischenstände speichern
6.1.6 Vorschau anzeigen lassen
6.1.7 Bilder/Dateien einfügen
6.1.8 Beitragsbild festlegen
6.1.9 Einen Inhalt veröffentlichen
6.2 Beiträge/Seiten verwalten
6.2.1 Liste der Beiträge/Seiten anschauen
6.2.2 Inhalte bearbeiten (QuickEdit und normal) oder löschen
6.2.3 Kategorien und Schlagwörter verwalten
6.3 Mit Revisionen arbeiten
6.4 Individuelle Felder nutzen
6.4.1 Individuelles Feld anlegen
6.5 Medien organisieren

6.5.1 Medien verwalten, bearbeiten und löschen
6.5.2 Bilder optimieren
6.5.3 Videos bei YouTube oder Vimeo hosten
6.6 Die mobilen Apps
6.7 Inhalte planen
6.7.1 Konzept
6.7.2 Ohne gute Inhalte läuft nichts
6.7.3 Barrierefreiheit von Inhalten
6.7.4 Struktur der Inhalte mit einer Sitemap planen
6.8 Classic Editor statt Block-Editor nutzen
6.9 Impressum und Datenschutzerklärung nicht vergessen

7 Benutzer*innen verwalten

7.1 Rollen verstehen
7.2 Benutzer*innen hinzufügen
7.3 Benutzer*innen verwalten: sortieren, filtern, suchen
7.4 Benutzerprofil bearbeiten
7.5 Benutzer*in deaktivieren oder löschen

8 Kommentare managen

8.1 Kommentar-Einstellungen
8.2 Kommentare freigeben
8.3 Datenschutz-Bedenken und -Lösungen
8.3.1 IP-Adresse bei Kommentaren nicht speichern
8.3.2 Profilbilder vom eigenen Server laden
8.3.3 DSGVO-Anfragen bearbeiten
8.4 Spamschutz mit Antispam Bee
8.5 Webmentions

9 WordPress Multisite

9.1 Was ist WordPress Multisite?
9.1.1 Vor- und Nachteile
9.1.2 Mögliche Anwendungsfälle
9.2 Die Multisite-Einrichtung
9.2.1 Vorbereitung
9.2.2 Aktivieren des Multisite-Modus
9.2.3 Konfiguration der Multisite
9.3 Bedienung des Netzwerks und Unterschiede zum Standard
9.3.1 Eine neue User-Rolle
9.3.2 Das Netzwerk-Verwaltungs-Interface
9.3.3 Upload-Verzeichnisse
9.3.4 Datenbank-Tabellen

Teil II WordPress-Website anpassen und erweitern

10 Design anpassen

10.1 Classic Themes und Block-Themes: Was ist das?
- 10.1.1 Classic Themes
- 10.1.2 Block-Themes

10.2 Installierte Themes vorab ansehen und aktivieren
- 10.2.1 Trennung von Design und Inhalt
- 10.2.2 Standard-Themes

10.3 Block-Theme anpassen: den Website-Editor kennenlernen
- 10.3.1 Die Sidebar links
- 10.3.2 Die Vorschau und Bearbeitungsansicht rechts

10.4 Classic Theme anpassen
- 10.4.1 Classic Theme über den Customizer konfigurieren
- 10.4.2 Navigationsmenü erstellen
- 10.4.3 Widget-Bereiche füllen

10.5 Neues Theme suchen und installieren
- 10.5.1 Ein Theme finden
- 10.5.2 Theme-Installation

10.6 Barrierefreiheit von Themes testen
- 10.6.1 Tastaturbedienbarkeit

10.6.2 Farbkontraste
10.6.3 Lesbarkeit von Texten
10.6.4 Labels für Formularfelder

11 Neue Funktionen mit Plugins hinzufügen

11.1 Plugins testen – sicher
11.2 Von der Anforderung zum Plugin
11.3 Gute Plugins erkennen
11.3.1 Quellen für Plugins
11.3.2 Checkliste zur Plugin-Auswahl
11.4 Plugins installieren
11.4.1 Installation über das WordPress-Interface
11.4.2 Manuelle Installation per ZIP-Datei
11.4.3 Manuelle Installation per (S)FTP
11.5 Plugin aktualisieren
11.5.1 Aktualisierungen mit dem WordPress-Updater
11.5.2 Automatische Updates einstellen
11.5.3 Manuelle Aktualisierungen
11.6 Plugins deaktivieren und löschen

Teil III Beispiel-Projekte

12 Einen Online-Shop mit WooCommerce betreiben

12.1 Die Vor- und Nachteile von WooCommerce

12.1.1 Selbst gehostet vs. SaaS

12.1.2 Das WordPress-Ökosystem

12.1.3 Skalierbarkeit

12.1.4 Anbindung an andere Geschäftssysteme

12.2 WooCommerce installieren und einrichten

12.2.1 Der WooCommerce-Einrichtungsassistent

12.2.2 WooCommerce anpassen

12.2.3 Rechtssicherheit mit »Germanized für WooCommerce«

12.2.4 Produkte hinzufügen

12.2.5 Erste Zahlungsmethoden einrichten

12.2.6 Versand leicht gemacht mit Versandklassen

12.3 Letzter Schliff und Testkauf

12.4 Zusätzliche Erweiterungen für Ihren Shop

12.4.1 Geld entgegennehmen mit »PayPal Payments for WooCommerce«

12.4.2 Produkt-Abos mit »WooCommerce Subscriptions«

12.4.3 Komplex konfigurierbare Produkte mit »Composite Products for WooCommerce«

12.4.4 Vermieten und Planen mit »WooCommerce Bookings«

12.4.5 Mehrere Produkte verkaufen mit »Product Bundles for WooCommerce«
12.4.6 Eröffnen Sie einen Marktplatz mit »Product Vendors for WooCommerce«

13 Das Blog

13.1 Zielgruppe und Plan
13.1.1 Welche Grundmotivation?
13.1.2 Welche Inhalte?
13.1.3 Besonderheiten Ihres Blogs?
13.1.4 Haben Sie das nötige Durchhaltevermögen?
13.1.5 Was ist Ihre Zielgruppe?
13.1.6 Wie steht es um Monetarisierung?
13.2 Design und Layout
13.3 Struktur und Navigation
13.4 Funktionen und Plugins
13.4.1 Statify
13.4.2 Public Post Review
13.4.3 Real Cookie Banner
13.4.4 Avatar Privacy
13.4.5 Embed Privacy
13.4.6 Worthy (VG-Wort-Integration)
13.4.7 Aged Content Message
13.5 Blogging-Tipps
13.5.1 Ideen sammeln
13.5.2 Ablenkungen vermeiden

13.5.3 Regelmäßige Beiträge – ein Redaktionsplan hilft
13.5.4 Aufgaben gruppieren
13.5.5 Artikel attraktiv gestalten
13.5.6 Social Media
13.5.7 SEO im Blick behalten

14 Eine Community-Website

14.1 Aufgaben einer Vereins-Website
14.1.1 Externe Darstellung
14.1.2 Neuigkeiten aus dem Vereinsleben
14.1.3 Bessere innere Vernetzung
14.2 Design, Struktur und Navigation
14.3 Funktionalitäten und Plugins
14.3.1 BuddyPress
14.3.2 Newsletter versenden – Mailjet
14.3.3 Kontaktformulare – Form Block
14.3.4 Zugriffsrechte und Rollen justieren – User Role Editor

15 IndieWeb und Fediverse

15.1 Das IndieWeb
15.1.1 Konferenzen und Meetups
15.1.2 Technik und Protokolle
15.1.3 Plugins
15.2 Das Fediverse

15.2.1 ActivityPub
15.2.2 WebFinger
15.2.3 Plugins
15.3 IndieWeb oder Fediverse?

Teil IV WordPress optimieren und warten

16 WordPress-Wartung und -Pflege

16.1 WordPress, Plugins und Themes aktualisieren
16.1.1 Kleine WordPress-Updates
16.1.2 Größere WordPress-Updates
16.1.3 WordPress-6-Update für alte Systeme
16.1.4 Update-Verhalten konfigurieren
16.1.5 WordPress-Downgrade
16.1.6 Plugin- und Theme-Updates
16.1.7 Update-Benachrichtigungen steuern
16.2 Wartungsmodus aktivieren – Plugin »Slim Maintenance Mode«
16.3 Backups planen und durchführen
16.3.1 Wann Backups erstellt werden sollten
16.3.2 Wie lange Backups aufbewahrt werden sollten

16.3.3 Backups anlegen und verwalten – Plugin »BackWPup«

17 Migrationen, Synchronisationen und Deployments

17.1 Testen und entwickeln mit Entwicklungs-, Staging- und Live-Umgebungen

17.1.1 Einrichten einer Staging-Umgebung

17.1.2 Nutzung von Git für eigenen Code und Deployment über CI

17.2 Eine Website migrieren

17.2.1 Manuell

17.2.2 Mit dem Plugin »Duplicator«

17.2.3 Schnell eine Kopie einer Instanz in einem Unterverzeichnis mit »WP Staging«

17.2.4 Von einem anderen System zu WordPress

17.2.5 Von WordPress zu einem anderen System

18 WordPress absichern

18.1 Login-Sicherheit

18.1.1 Passwort-Sicherheit

18.1.2 Sichere Passwörter durchsetzen und Login schützen

18.1.3 Zwei-Faktor-Authentifizierung
18.2 Die Wichtigkeit von SSL/TLS
18.2.1 Kostenloses SSL aktivieren
18.2.2 SSL erzwingen
18.2.3 Unsaubere HTTP-Elemente aufräumen
18.3 WordPress abhärten
18.3.1 XML-RPC-Schnittstelle deaktivieren
18.3.2 REST-API prüfen
18.3.3 Verzeichnisschutz per Basic Authentication
18.3.4 Datei-Editor deaktivieren
18.3.5 Plugin- und Theme-Installation verhindern
18.3.6 Zugriff auf IP-Adresse begrenzen?
18.4 No-Gos der WordPress-Sicherheit
18.4.1 Login verstecken
18.4.2 WordPress-Version verstecken
18.4.3 Datenbank-Präfix ändern

19 Performance-Optimierung

19.1 Grundlagen zur Optimierung
19.2 PageSpeed Insights – Missstände verstehen und bei Bedarf abarbeiten
19.2.1 Core Web Vitals und weitere Messwerte
19.2.2 PageSpeed-Tests im Bereich »Leistung«
19.2.3 PageSpeed-Tests im Bereich »Barrierefreiheit«
19.2.4 PageSpeed-Tests im Bereich »Best Practices«
19.2.5 PageSpeed-Tests im Bereich »SEO«

19.3 Optimierungs-Plugin »Autoptimize«
19.3.1 JS, CSS & HTML
19.3.2 Bilder
19.3.3 Kritisches CSS
19.3.4 Extras
19.3.5 »Mehr optimieren« und »Seiten-Cache«
19.4 Einfaches Caching-Plugin ohne Schnickschnack – »Cachify«
19.5 Profi-Cache mit WP Rocket
19.5.1 Dashboard
19.5.2 Datei-Optimierung
19.5.3 Medien
19.5.4 Cache füllen
19.5.5 Erweiterte Regeln
19.5.6 Datenbank
19.5.7 CDN
19.5.8 Heartbeat
19.5.9 Add-ons
19.5.10 Werkzeuge
19.6 Bilder optimieren
19.7 Optimieren für den Klimaschutz
19.7.1 CO_2-Ausstoß ermitteln
19.7.2 CO_2-Ausstoß reduzieren
19.7.3 Weitreichendere Verbesserungen

20 Suchmaschinenoptimierung

20.1 Schlank mit allen wichtigen Funktionen – Plugin »The SEO Framework«
20.1.1 Allgemeine Einstellungen
20.1.2 Titel-Einstellungen
20.1.3 Meta-Description-Einstellungen
20.1.4 Startseiten-Einstellungen
20.1.5 Soziale Netzwerk-Meta-Einstellungen
20.1.6 Schema.org-Einstellungen
20.1.7 Robots-Meta-Einstellungen
20.1.8 Webmaster-Meta-Einstellungen
20.1.9 Sitemap-Einstellungen
20.1.10 Feed-Einstellungen
20.1.11 Einstellungsmöglichkeiten im Editor
20.1.12 Erweiterungen und Bezahlfunktionen
20.2 Viele SEO-Maßnahmen in einem Plugin mit »Yoast SEO«
20.2.1 Der Konfigurationsassistent
20.2.2 Einstellungen
20.2.3 Integration
20.2.4 Werkzeuge
20.2.5 Verschiedene Bezahlfunktionen
20.2.6 Optionen im Inhaltseditor
20.3 Sitemap bei Google und Bing einreichen
20.3.1 Einreichung bei Google über die Search Console
20.3.2 Einreichung bei Bing

21 Notfallmaßnahmen

21.1 Gehackt, was nun?
21.1.1 Sofortmaßnahmen-Checkliste
21.1.2 Schadcode suchen
21.1.3 Website reparieren oder neu aufbauen?
21.2 Die häufigsten Standardproblemlösungen
21.2.1 Standardproblemlösung
21.2.2 Website langsam mit vielen Zugriffen auf xmlrpc.php
21.2.3 404-Fehler – Seite nicht gefunden
21.2.4 Änderung nicht sichtbar
21.2.5 Keine Suchmaschinen-Indexierung
21.2.6 Ausgesperrt – Passwort wiederherstellen
21.2.7 Ausgesperrt – kein Zugriff auf die Website
21.2.8 Serverfehler
21.2.9 Datenbankproblem
21.2.10 Fatal Error: Allowed Memory exhausted
21.2.11 Verbindung ist nicht sicher
21.2.12 White Screen of Death

Teil V WordPress für Entwickler*innen

22 Grundwissen für WordPress-Entwickler*innen

22.1 Coding-Standards
22.1.1 Accessibility
22.1.2 CSS
22.1.3 HTML
22.1.4 PHP
22.1.5 Versionierung
22.2 WordPress-Konzepte
22.2.1 Hooks
22.2.2 WordPress-APIs, Template-Tags (Classic Themes)
22.2.3 Übersetzungen (Internationalisierung)
22.2.4 Übersetzungen anlegen
22.3 Entwicklungs- und Testumgebung einrichten
22.3.1 Code-Editor auswählen
22.3.2 Programmierhilfen finden
22.3.3 Debuggen

23 Theme entwickeln

23.1 Block-Theme erstellen
23.1.1 Grundlagen zur Theme-Entwicklung
23.1.2 Block-Theme umsetzen
23.2 Classic Theme anpassen und erstellen
23.2.1 Bestehendes Theme mit einem Child-Theme anpassen
23.2.2 Ein Theme von Grund auf neu entwickeln
23.2.3 Theme-Aufbau

23.3 Theme veröffentlichen
23.3.1 Ein kostenloses Theme veröffentlichen
23.3.2 Ein kostenpflichtiges Theme veröffentlichen
23.4 Weitere nützliche Funktionen für Themes
23.4.1 Zusätzliche Bildgrößen registrieren
23.4.2 Block-Stile registrieren
23.4.3 Core-Block-Stile entfernen
23.4.4 Sperren (Locking) von Blöcken und Templates

24 Plugin und Block entwickeln

24.1 Installation von Node.js
24.2 Das Grundgerüst des Plugins
24.3 Inhaltstyp erstellen
24.3.1 Permalink-Struktur automatisch aktualisieren
24.3.2 Metadaten für den Inhaltstyp registrieren
24.4 Panel in der Editor-Seitenleiste zur Bearbeitung der Metadaten
24.4.1 Das @wordpress/scripts-Paket installieren
24.4.2 Das Panel erstellen
24.5 Editor-Blöcke erstellen
24.5.1 Array mit Infos zu den Metadaten im Editor verfügbar machen
24.5.2 Block mit Liste der Metawerte erstellen
24.5.3 Block zur Anzeige eines einzelnen Metawerts erstellen
24.5.4 Die Blöcke nur für den talk-Inhaltstyp erlauben

24.6 Eine Checkbox zur Hervorhebung eines Vortrags ins Veröffentlichungs-Panel einfügen
24.6.1 Script erstellen
24.6.2 Meta-Key registrieren
24.6.3 Highlight-Klassen zum Styling im Frontend ausgeben
24.7 Block-Template für den Inhaltstyp erstellen
24.8 Übersetzung
24.9 Inhalte beim Deinstallieren löschen
24.10 Plugin veröffentlichen
24.10.1 Kostenlos
24.10.2 Kostenpflichtig
24.11 Weitere nützliche Funktionen für Plugins
24.11.1 Die Möglichkeiten der block.json im Detail
24.11.2 Block Bindings API: Metadaten und andere Quellen mit Core-Blöcken verknüpfen
24.11.3 Interactivity API: Interaktive Frontend-Erfahrung umsetzen
24.11.4 Eigene Block-Editor-Sidebar erstellen
24.11.5 Optionen in der Dokument-Sidebar deaktivieren
24.11.6 Bestimmte Blöcke deaktivieren
24.11.7 Konfiguration bestehender Blöcke anpassen
24.11.8 Shortcode erstellen
24.11.9 Eigene Taxonomie erstellen
24.11.10 Einstellungen im Backend einfügen

A Glossar

Stichwortverzeichnis

Rechtliche Hinweise
Über das Autorenteam

Vorwort

WordPress ist ein Content-Management-System, mit dem Sie sehr flexibel eine eigene Website erstellen und verwalten können. Dieses Buch zeigt Ihnen in fünf Teilen alle wichtigen Themen rund um WordPress, von den Grundlagen bis zur Entwicklung eigener Themes und Plugins. Es ist also ganz egal, ob Sie als WordPress-Neuling ganz vorne einsteigen oder ob Sie sich bereits auskennen und ein Nachschlagewerk zu WordPress 6 suchen: Dieses Handbuch wird Ihnen bestimmt weiterhelfen.

Die Teile des Buchs bauen aufeinander auf, und wir empfehlen Ihnen, die Kapitel chronologisch durchzuarbeiten – zumindest wenn dies Ihre erste Begegnung mit WordPress ist. Sollten Sie später einmal vergessen haben, welches tolle Plugin Sie in diesem Buch zum Suchen und Ersetzen in der Datenbank kennengelernt haben – oder wie Sie den Block-Editor anpassen –, dann hilft Ihnen der Index bei der Suche nach dem richtigen Kapitel.

Das Buch unterstützt Sie im ersten Teil dabei, WordPress kennenzulernen, zu installieren und zu bedienen. Im zweiten Teil erfahren Sie, wie Sie das Aussehen Ihrer WordPress-Website mit Themes verändern und wie Sie zusätzliche Funktionalitäten mit Plugins hinzufügen können. In Teil vier werden Ihnen verschiedene Beispielprojekte vorgestellt, zum Beispiel wie Sie einen Online-Shop mit WooCommerce umsetzen können oder Ihre Website mit dem IndieWeb verbinden können, um etwa Reaktionen aus den sozialen Medien auf Ihrer Website anzuzeigen. Teil vier beschäftigt sich mit

Aspekten, die im laufenden Betrieb einer Website wichtig sind, wie etwa Performance-Optimierung und Backups. Im fünften Teil geht es darum, selbst ein Theme und ein Plugin für WordPress zu erstellen, also selbst in die Entwicklung einzusteigen. Dabei kann dieses Buch allerdings wirklich nur als Einstieg in die Programmierung für WordPress dienen, und auch Grundwissen in der Webentwicklung wird vorausgesetzt. Sollten Sie tiefer in die Entwicklung einsteigen wollen, dann finden Sie auf der Seite des Rheinwerk Verlags weitere Bücher zum Thema WordPress-Entwicklung.

Nachdem die Vorauflage eine Überarbeitung des Buchs des im Frühjahr 2021 verstorbenen Richard Eisenmenger war, haben wir uns für die Neuauflage entschieden, weite Teile neu zu schreiben, sofern sie nicht für die Vorauflage bereits neu geschrieben wurden. Wir sind dabei der Struktur, die Richard erarbeitet hat, treu geblieben und sehen diese neue Auflage daher als Fortsetzung seiner Arbeit.

Diese Idee haben wir später das eine oder andere Mal bereut und den Aufwand »etwas« unterschätzt. Aber wir sind der Meinung, dass er sich letztlich gelohnt hat, und deshalb froh über die Entscheidung. Thematisch ganz neu hinzugekommen ist ein Kapitel zum IndieWeb, für das wir zum Glück unseren Freund Matthias Pfefferle als Autor gewinnen konnten. Matthias beschäftigt sich schon viele Jahre mit dem IndieWeb und hat viele der Plugins entwickelt, an denen kein Weg vorbeiführt, wenn eine WordPress-Website ans IndieWeb andocken möchte. Genau der Richtige für dieses Kapitel also – vielen Dank, dass du das Buch bereichert hast!

Ebenfalls ein eigenes Kapitel bekommen hat das Thema Multisite-Installation, bei der mehrere WordPress-Websites in einer WordPress-Instanz betrieben werden können. In der Vorauflage

wurde das Thema zwar schon angerissen, aber eine so mächtige Funktion hat es verdient, ausführlicher vorgestellt zu werden. Im Bereich des Block- und Website-Editors hat sich seit Erscheinen der Vorauflage vieles getan, angefangen von neuen Blöcken bis hin zu neuen APIs für Entwickler*innen.

Aufseiten des Rheinwerk Verlags hat uns Stephan Mattescheck durch das Projekt begleitet, was vermutlich auch nicht immer ganz einfach war. Vielen Dank für die gute Zusammenarbeit.

Jetzt aber: Ihnen viel Freude beim Lesen und beim Arbeiten mit WordPress.

Florian Brinkmann und **Simon Kraft**
Flensburg und Konstanz

1 Was Sie mit WordPress vorhaben und wo Sie jetzt weiterlesen

WordPress ist zwar aus einem Blog-System heraus entstanden, aber seit Langem viel mehr als das: nämlich ein vollwertiges Content-Management-System, mit dem Sie nahezu alle Arten von Websites umsetzen können, die Sie sich vorstellen können – egal ob einfaches Blog, Unternehmens-Website, Online-Shop oder -Community. Dieses Buch wird Ihnen einen umfassenden Einblick in WordPress geben und als Nachschlagewerk dienen. Wenn Sie bisher hauptsächlich von WordPress gehört haben, ohne sich bereits viel damit beschäftigt zu haben, und das ändern möchten, sind Sie bei diesem Buch genau an der richtigen Adresse. Egal, ob Sie WordPress hobbymäßig nutzen möchten oder Geld mit WordPress verdienen wollen: Das hier erworbene Wissen wird Ihnen helfen.

1.1 Das richtige Handbuch für Sie

Wegen seiner Beliebtheit gibt es eine große Menge Plugins, mit denen der Funktionsumfang von WordPress erweitert werden kann, sodass sich viele Arten von Websites realisieren lassen – oft ohne dass Sie selbst Programmierkenntnisse mitbringen müssen. In späteren Kapiteln wird beispielhaft gezeigt, wie sich unterschiedliche Website-Typen mit WordPress umsetzen lassen,

welche Plugins sich eignen und so weiter. Aber auch wenn Sie in die Programmierung für WordPress einsteigen wollen, hilft Ihnen dieses Handbuch weiter.

Wenn Sie sich in einem der folgenden Punkte wiederfinden, sollte Ihnen das Buch einen guten Einstieg geben.

1.1.1 Das Blog

WordPress ist als Blog-System gestartet, wenn Sie also Artikel schreiben und veröffentlichen möchten, vielleicht mit einer Kommentarfunktion, um direkt bei den Texten Feedback anzuzeigen, sind Sie bei WordPress genau richtig und können an sich schon ohne ein einziges Plugin direkt loslegen. Sinnvoll wären aber wohl zumindest ein paar, etwa eins, das gegen Spam-Kommentare hilft (später im Buch wird eins näher vorgestellt), und ein leichtgewichtiges SEO-Plugin. Und vielleicht möchten Sie auch Reaktionen aus sozialen Netzwerken anzeigen, etwa wenn ein Post mit einem Link zu Ihrem Beitrag geteilt wird. Hier ist das Stichwort IndieWeb, dem in diesem Buch ein eigenes Kapitel gewidmet ist.

1.1.2 Die persönliche Website

Diese Website kann natürlich auch ein Blog sein, aber hier ist damit eine Website gemeint, mit der Sie auf mehreren »statischen« Seiten (»statisch« im Sinne von »kein Blogartikel oder Ähnliches«) über sich informieren, beispielsweise über Ihre Hobbys. Vielleicht gibt es auch einen geschützten Bereich, in dem Sie Fotos für Familie und Freundeskreis hochladen. Auch dafür hat WordPress grundsätzlich alles an Bord.

1.1.3 Die Unternehmens-Website

Sie haben ein Unternehmen oder sollen für ein Unternehmen eine Website erstellen? Abgesehen von den Inhalten unterscheidet sich diese Art der Website im einfachsten Fall nicht groß von einer persönlichen Website, und wenn es nicht der einfachste Fall ist, lässt sich WordPress entsprechend erweitern, zum Beispiel um einen Bereich für Kund*innen zu erstellen.

1.1.4 Der Online-Shop

Sie möchten Dinge über das Internet verkaufen und dafür einen eigenen Online-Shop starten? Auch das kann WordPress (nicht von Haus aus, aber mit Unterstützung von Plugins). Weiter hinten im Buch widmet sich ein ganzes Kapitel dem Aufsetzen und Betreiben eines Online-Shops mit WordPress und erläutert, was es dabei zu beachten gilt.

1.1.5 Die Vereinswebsite

Auch bei Vereinswebsites gibt es eine große Spanne an Websites, von ganz einfachen, die rein informativ sind, bis zu Websites mit Accounts und internem Bereich für die Mitglieder, in dem beispielsweise Termine angezeigt werden und miteinander kommuniziert werden kann, vielleicht noch mit angebundenem Online-Shop, über den Tickets für Veranstaltungen und anderes verkauft werden.

1.1.6 Sie wollen für WordPress programmieren

Wenn Sie lernen möchten, wie Sie WordPress mit Plugins anpassen und erweitern sowie eigene Themes entwickeln können, werden Sie ebenfalls in diesem Buch fündig. Die Kapitel des letzten Teils widmen sich dem Einstieg in die Programmierung von WordPress-Plugins und -Themes.

1.2 Eine kurze Geschichte von WordPress

In einer Zeit vor WordPress, Anfang der Neunzigerjahre des letzten Jahrhunderts, startete am CERN das World Wide Web. Während dieser Zeit sind verschiedene Programmiersprachen durch diverse Versionen gegangen, haben neue Webbrowser das Licht der Welt erblickt und sind wieder vergangen, sind Trends in Gestaltung und Aufbau von Websites entstanden und wieder aus der Mode gekommen. Was über die gesamte Zeit geblieben ist, sind HTML und CSS, ohne die das Web auch heute nicht denkbar wäre.

Das Web hat uns so lange begleitet, dass sein täglicher Gebrauch für die meisten von uns kaum noch wegzudenken ist. Sei es als Konsument*innen auf der Suche nach den Öffnungszeiten des netten veganen Cafés am See oder als Produzent*innen bei der Pflege unserer eigenen Website.

Und während unser heutiges Web von großen Unternehmen, sozialen Netzwerken und gigantischen Online-Shops geprägt ist, war es früher ein verrückter, etwas chaotischer und manchmal etwas schwierig zu verstehender Ort, an dem Menschen aus den verschiedensten Teilen der Welt ihre Gedanken und Ideen veröffentlichten, und die Websites waren dabei recht individuell.

Wenn Sie sich heute entschließen sollten, ein eigenes Blog zu starten, treten Sie in die Fußstapfen dieser frühen Pionier*innen des Internets und können dazu beitragen, dass es auch in der Zukunft ein wenig wild bleibt und nicht alles hinter Bezahlschranken, Newsletter-Anmeldungen und Werbung verschwindet. WordPress ist dafür genau das richtige Werkzeug.

Doch schauen wir noch ein wenig in die Vergangenheit von WordPress. Wenn Sie im Moment dieses Buch in Händen halte, haben Sie vermutlich schon einmal von WordPress gehört. Mit Sicherheit haben Sie schon Websites besucht, die mit WordPress betrieben werden. Eventuell haben Sie sogar schon einmal an einer solchen WordPress-Website gearbeitet? Um zu verstehen, warum das CMS zu Unrecht immer noch von einigen als Blog-System verschrien ist, müssen wir einen Blick in die Vergangenheit werfen.

Zeitraum	Version	Bemerkung
2003–2005	0.71–1.x	Mike Little und Matt Mullenweg bauen auf Basis des eingestellten Projekts b2/cafelog eine frühe Version von WordPress. Dieser sogenannte Fork war nur dank der Open-Source-Lizenz möglich, unter der b2 stand und die noch heute ein wichtiger Grundpfeiler des WordPress-Ökosystems ist. Das System im Hintergrund, das WordPress überhaupt erst für die Nutzung von Plugins bereit macht, wurde vorgestellt. Gegen Ende dieses Zyklus wurden mit WordPress 1.5 Themes eingeführt, und das ikonische Kubrick-Theme wurde zum Standard jeder neuen Website gemacht.
2005–2009	2.x	Gleich zweimal wird in dieser Zeit die Admin-Oberfläche überarbeitet. Die zweite Überarbeitung hat sich sogar knappe 15 Jahre gehalten. Widgets, die als kompakte Blöcke Informationen etwa in der Seitenleiste oder im Fußbereich einer Website anzeigen können, wurden in dieser Zeit ebenso eingeführt wie die Versionierung von Beiträgen und Seiten.

Zeitraum	Version	Bemerkung
2010–2014	3.x	Erst 2010 löst das neue Standard-Theme Twenty Ten das 2005 vorgestellte Kubrick ab. Der Multisite-Modus wird ein offizielles WordPress-Core-Feature. Mit besseren Kontrollen, einem Admin-Bar und dem Customizer lässt WordPress endgültig seine reinen Blog-System-Wurzeln hinter sich und wird ein vollwertiges CMS. In Version 3.7 wird mit Auto-Updates ein Feature eingeführt, das in seiner Wichtigkeit schnell übersehen werden kann: Einfaches Aktualisieren des *WordPress-Core* und später auch von Themes und Plugins direkt aus dem Admin-Interface heraus ist eine Funktion, die viele andere Content-Management-Systeme erst Jahre später nachrüsten werden.

Zeitraum	Version	Bemerkung
2014–2017	4.x	Im Rückblick fühlt sich der komplette Release-Zyklus der 4er-Versionen wie eine konstante Verbesserung bestehender Features an. Viele etwas unrunde Ecken im Interface und im Funktionsumfang des CMS werden glatt geschliffen. Für Entwickler*innen gibt es aber zwei große Neuerungen: Die Einführung der REST-API macht WordPress zu einem Kandidaten für exotischere Einsätze als bloße Websites. Apps oder sogar Druck-Erzeugnisse können WordPress über die REST-API einbinden und als CMS nutzen. Und zur selben Zeit wird das WordPress-Kommandozeilen-Tool WP-CLI veröffentlicht, das komplexe Operationen in WordPress ohne grafische Oberfläche ermöglicht. Im Sommer 2017 wird mit Gutenberg die experimentelle Version des späteren Block-Editors vorgestellt und ist zunächst nur als Plugin verfügbar.

Zeitraum	Version	Bemerkung
2018–2022	5.x	Der Block-Editor wird im Winter 2018 zum neuen Standard in WordPress. Alle Inhalte von Beiträgen und Seiten können in diesem Editor als Inhaltsblöcke zusammengestellt und manipuliert werden. Das Konzept wird in den folgenden Jahren weiter ausgeweitet und umfasst schließlich auch den Widget-Bereich und später sogar Themes. In WordPress 5.9 wird die erste Version des Website-Editors veröffentlicht, der in den folgenden Versionen weiter verfeinert werden wird, um mit einer Block-basierten Umgebung auch Teile einer Website anpassen zu können, die bisher ausschließlich mit HTML- und PHP-Kenntnissen anpassbar waren.
seit 2022	6.x	Der Block-Editor wird weiter verbessert, und die neuen Block-Themes erhalten eine Vielzahl neuer Funktionen. Stil-Varianten machen sie flexibler, und eine Schriftbibliothek soll das Anpassen des Designs weiter erleichtern. Diverse Programmierschnittstellen ermöglichen immer tiefer gehende Anpassung von Blöcken für Entwickler*innen. Die dritte Phase des Gutenberg-Projekts startet. Sie wird in der Zukunft allerlei Verbesserungen für die kollaborative Arbeit an Inhalten in WordPress bringen.

1.3 Warum WordPress?

Content-Management-Systeme gibt es viele, warum sollten Sie sich für WordPress entscheiden – und in welchen Fällen vielleicht auch nicht?

1.3.1 Schnell zum ersten veröffentlichten Inhalt

Anders als bei einigen anderen Content-Management-Systemen ist es bei WordPress möglich, direkt nach der Installation eine fertige Website zu haben. Wenn Sie mit einem der vielen fertigen Designs (»Themes« genannt) zufrieden sind, können Sie direkt mit dem Einpflegen der Inhalte starten.

Bei einigen anderen Systemen stehen nach der Installation erst die Konfiguration des Admin-Bereichs und die Entwicklung der Ausgabe an, bevor Inhalte eingegeben und angezeigt werden können (verbunden damit sind aber auch einige Vorteile, die WordPress von Haus aus nicht mitbringt).

1.3.2 Fast unendliche Möglichkeiten

Da WordPress sehr weit verbreitet ist, gibt es Tausende Plugins, um den Funktionsumfang zu erweitern. Wenn Ihnen eine Funktion in WordPress fehlt, ist die Chance groß, dass es dafür ein Plugin gibt.

Sie können mit WordPress deshalb viele verschiedene Arten von Websites umsetzen, von einer einfachen Web-Visitenkarte über ein persönliches Blog oder eine Unternehmens-Website hin zu einem Onlineshop.

1.3.3 Hoheit über die eigenen Daten

Das ist kein Punkt nur für WordPress, sondern für das Selberhosten Ihrer Website. Neben der Möglichkeit, WordPress selbst zu hosten, gibt es noch *WordPress.com*, einen Dienst, bei dem Sie sich sehr einfach ein WordPress einrichten können. Von der Art ist das vergleichbar mit Diensten wie Wix, Squarespace oder Jimdo.

Auch wenn es zunächst verlockend ist, so einen Dienst zu nutzen, weil die Erstellung einer Website dort sehr schnell gehen kann und Sie sich nicht um Themen wie Webhosting oder Registrierung einer Domain kümmern müssen, binden Sie sich damit doch an einen Drittanbieter, der theoretisch irgendwann nicht mehr existieren könnte oder Funktionen entfernt/anpasst, Preise ändert, ohne dass Sie etwas dagegen tun können. Und würden Sie dann gerne mit Ihrer Website woanders hin umziehen, müssten Sie sie wahrscheinlich neu erstellen. *WordPress.com* ist hier ein Sonderfall, weil Sie die Inhalte exportieren und in ein selbst gehostetes WordPress importieren können.

Kurz: Egal, ob Sie sich für WordPress oder ein anderes System entscheiden, hosten Sie Ihre Website selbst.

1.3.4 Open Source

WordPress ist, wie viele andere Content-Management-Systeme auch, Open Source. Das bedeutet, der Code von WordPress ist öffentlich, Sie dürfen den Code verändern und, wenn Sie möchten, auch die veränderte Version neu veröffentlichen. Vor ein paar Jahren wurde beispielsweise mit »ClassicPress« ein sogenannter Fork von WordPress erstellt, der nicht den Block-Editor enthält, und das ist dank der Veröffentlichung unter der GPL-v2-Lizenz kein Problem.

1.3.5 Server-Voraussetzungen für WordPress

Die Server-Voraussetzungen zur Nutzung von WordPress sind recht übersichtlich, sodass die Software bei den meisten Hosting-Anbietern problemlos laufen sollte, sofern PHP und MySQL/MariaDB unterstützt werden. Die empfohlenen Mindestanforderungen finden Sie auf *wordpress.org/about/requirements/*.

1.3.6 Wann WordPress nicht die richtige Wahl ist

Auch wenn fast alles mit WordPress möglich ist, bedeutet das nicht, dass WordPress für alles die richtige Wahl ist. Wenn beispielsweise ein Forum eingerichtet werden soll, lässt sich das zwar in WordPress mit Plugins realisieren, aber gegebenenfalls ist eine Foren-Software besser geeignet und auf längere Sicht einfacher zu handhaben, weil sie von Haus aus die ganzen Funktionen schon mitbringt.

Hier gibt es aber natürlich keine feste Regel, nach der gesagt werden kann: Anwendungsfall X ist vielleicht doch zu speziell für WordPress. Behalten Sie nur im Hinterkopf, dass Sie mit WordPress nicht alle Anwendungsfälle erfüllen können müssen, nur weil es theoretisch vielleicht irgendwie geht.

1.4 Anleitung zum Handbuch

Das Handbuch ist in fünf Teile unterteilt, die zunehmend anspruchsvoller werden:

1. Teil I beschäftigt sich mit der Installation von WordPress sowie der Bedienung der Administrationsoberfläche (des Backends).
2. Teil II dreht sich darum, das Design einer WordPress-Website mit einem Theme anzupassen sowie den Funktionsumfang mit Plugins zu erweitern.
3. Teil III zeigt die Umsetzung von vier unterschiedlichen Beispiel-Websites, mit Empfehlungen für passende Plugins.
4. Teil IV behandelt die Wartung und Optimierung einer WordPress-Instanz. Dazu gehören beispielsweise Dinge wie die Einrichtung regelmäßiger Backups und Performance-Optimierung.
5. Teil V bietet einen Einstieg in die Programmierung von WordPress-Themes und -Plugins.

In den unterschiedlichen Teilen begegnen Ihnen die folgenden Elemente:

Theorie

Auch wenn wir versuchen, rein theoretische Abschnitte kurz zu halten, sind diese doch an mancher Stelle wichtig, um beispielsweise Hintergründe zu erläutern oder eine Technologie etwas detaillierter zu erklären.

Schritt für Schritt

Sie werden im Buch immer wieder auf Anleitungen stoßen, denen Sie Schritt für Schritt folgen können.

Links

Sie werden im Buch natürlich abgedruckte Links antreffen, die Sie zum Beispiel zu einer Download-Seite für ein Plugin oder zu einer Seite mit weiteren Informationen über ein Thema führen. Um Ihnen das Abtippen der Links zu erleichtern, wenn Sie die Printausgabe des Buchs in der Hand halten, wird ein URL-Shortener verwendet. Das bedeutet, alle URLs in diesem Buch fangen mit *r-wrk.de* an, gefolgt von einem möglichst kurzen Teil dahinter, anhand dessen auf die korrekte Seite weitergeleitet wird (fiktives Beispiel: *r-wrk.de/playground* leitet auf *playground.wordpress.net* weiter).

Code-Listings

Besonders im letzten Teil des Buchs werden viele Listings mit Quellcode abgedruckt sein, die Sie natürlich gerne abtippen können, wenn Sie möchten, aber auch einfach von der Rheinwerk-Website kopieren können: *rheinwerk-verlag.de/5893*.

Kästen

Wenn Sie im Buch einem grau hinterlegten Kasten begegnen, finden Sie darin zusätzliche Infos zum aktuellen Thema, die nicht direkt in den Text passen. Es gibt ein paar unterschiedliche Arten an Kästen:

Achtung: Das sollten Sie unbedingt lesen

In »Achtung«-Kästen finden Sie sehr wichtige Informationen, die beispielsweise auf Sicherheitsrisiken oder Einschränkungen hinweisen können.

Tipp: Ein kleiner, hilfreicher Tipp

In den »Tipp«-Kästen finden Sie, wie der Name sagt, hilfreiche Tipps, die Ihnen zum Beispiel nützliche Hinweise zu Konfigurationen von Tools zeigen.

Info: Eine interessante Zusatzinformation

Nicht entscheidend, aber im »Info«-Kasten finden Sie beispielsweise interessante Hintergrundinformationen.

Textformate und typografische Konventionen

Im Fließtext des Buchs finden Sie immer wieder Verweise auf Elemente, die in Screenshots gezeigt werden, und Dinge wie Programm-Code oder URLs. Damit sie nicht im normalen Text untergehen, sind sie entsprechend formatiert:

- Buttons, Beschriftungen, Links: alles, was auf einer Website sichtbar ist
- Wenn ein Klickpfad beschrieben wird, also eine Reihe Elemente, die Sie nacheinander anklicken sollen, werden sie durch den Bullet-Punkt (•) getrennt. Zum Beispiel: Menü Einstellungen • Allgemein • Datumsformat.

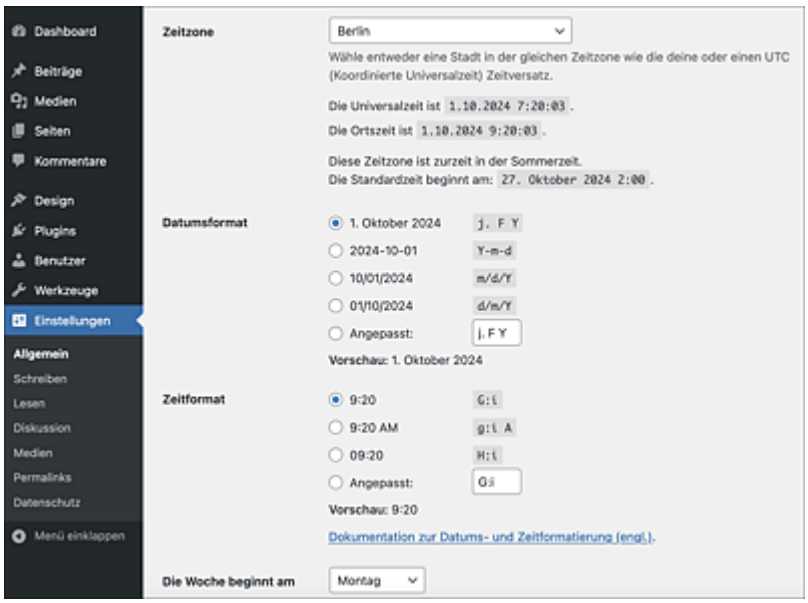


Abbildung 1.1 In einer Bildunterschrift werden die Interaktionselemente in Anführungszeichen gesetzt: »Einstellungen« • »Allgemein« • »Datumsformat«.

- *URLs* und Bestandteile davon, wie nur eine Domain oder ein Pfad
- »Texteingaben finden Sie in Anführungszeichen«.
- *Verzeichnisse* oder *Dateien* auf Ihrem Rechner
- `jede Art Code, wobei` **`einzelne Teile`** *`hervorgehoben`* `sein können`
- Bei Macs gibt es eine Besonderheit bei Tastenkombinationen: Hier verwenden Sie statt `Strg` meist `cmd`.
- Bei den WordPress-Datenbanktabellen wird am Anfang ein Sternchen `*` genutzt, zum Beispiel bei `*_posts` oder `*_options`, da der Standardwert `wp` des Präfixes geändert werden kann.
- Gibt es verschiedene mögliche Werte, wie bei vielen CSS-Eigenschaften, werden sie mit dem Pipe-Symbol `|` getrennt. Wenn Sie `auto|initial|inherit` sehen, bedeutet das also entweder `auto`, `initial` oder `inherit`.
- Wenn Sie ein kursiv geschriebenes Wort sehen, das nichts mit einer URL zu tun hat, handelt es sich dabei um ein wichtiges

Wort, das neu eingefügt wird und im weiteren Teil wichtig ist. Manchmal dient die Kursivschreibung auch nur der Betonung.

Im Fehlerfall: Errata

Wenn sich doch vereinzelt Fehler eingeschlichen haben, zum Beispiel in einem Code-Listing, oder sich etwas grundlegend verändert hat, sodass eine Beschreibung hier im Buch nicht mehr funktioniert, hilft die Seite zum Buch des Verlags: *https://www.rheinwerk-verlag.de/5893*. Dort finden Sie alle eventuellen Korrekturen und die Listings zum Download.

1.5 Nach der letzten Seite

Obwohl dieses Handbuch, wie der Titel sagt, »umfassend« ist, behandelt es natürlich nicht alle erdenklichen Themen, die mit WordPress zu tun haben. Sie werden also, auch wenn Sie das Buch komplett durcharbeiten, früher oder später auf Probleme stoßen, deren Lösung nicht im Buch steht. Was dann? Hier ein paar Tipps.

1.5.1 Dokumentation auf WordPress.org

WordPress bietet online Anleitungen und Dokumentation zu vielen Themen an. Wenn englische Texte für Sie kein Problem darstellen, ist das ein guter Ansatzpunkt.

- Unter *learn.wordpress.org* finden Sie Videos zu unterschiedlichen Themen, zu Funktionen von neuen Versionen, wie ein Beta-Release getestet oder wie ein Menü mit dem Navigations-Block erstellt werden kann.

  Außerdem gibt es hier Online-Workshops, zu denen auch Aufzeichnungen gemacht werden, die später angeschaut werden können.

- *wordpress.org/documentation* bietet Hilfeartikel zu verschiedenen Themen, die bei der Nutzung von WordPress helfen – angefangen bei Dingen wie der Installation von WordPress über eine Liste der standardmäßig verfügbaren Blöcke inklusive Beschreibung jedes Blocks bis zur Anpassung der eigenen Site mit einem Block-Theme.

- *developer.wordpress.org* ist die Anlaufstelle für Ressourcen rund um die Entwicklung für WordPress, egal, ob Sie ein Theme

entwickeln, ein Plugin programmieren oder einen WP-CLI-Command erstellen möchten.

1.5.2 Suchmaschine des Vertrauens

Da WordPress so viele Nutzer*innen hat, ist es wahrscheinlich, dass jemand schon ein ähnliches Problem hatte wie Sie und eine Lösung dafür gefunden und veröffentlicht hat. Suchen Sie also in einer Suchmaschine nach dem Problem, am besten in Zusammenhang mit dem Begriff »WordPress«. Wenn Sie auf Deutsch nicht fündig werden, versuchen Sie es auf Englisch.

Achten Sie bei Online-Artikeln darauf, wann sie geschrieben wurden. Besonders im Bereich des Block-Editors, Website-Editors und der Block-Themes gibt es eine zügige Weiterentwicklung, sodass Lösungen relativ schnell veralten können. Wenn es um Entwicklungsthemen geht, können Sie die beschriebene Lösung aber in jedem Fall als Ausgangspunkt für weitere Recherchen nutzen, beispielsweise in der offiziellen Doku von WordPress nach den verwendeten Funktionen suchen und schauen, wie sie dort in Beispielen verwendet wird.

1.5.3 Dokumentation von Plugin- oder Theme-Herstellern

Wenn Sie ein Problem mit einem Plugin oder Theme haben, schauen Sie, ob auf der Website des Herstellers vielleicht eine Dokumentation vorhanden ist, die weiterhelfen kann, oder eine Support-Möglichkeit, die Sie kontaktieren können.

Bei der Kontaktaufnahme höflich zu sein, gebietet sich natürlich in jedem Fall, aber wenn es sich um ein kostenloses Plugin oder Theme

handelt, seien Sie zusätzlich geduldig, wenn eine Antwort auf sich warten lässt oder vielleicht auch gar keine kommt. Entwickler*innen von kostenlosen Plugins oder Themes machen das meistens in ihrer Freizeit.

1.5.4 WordPress-Support-Foren

Unter *de.wordpress.org/support/* finden Sie die Einstiegsseite zu den deutschsprachigen Support-Foren von WordPress. Auch wenn das sehr offiziell klingt, sind die meisten Nutzer*innen, die im Forum helfen, unbezahlte Freiwillige, die das in ihrer Freizeit tun. Auch hier ist also keine Erwartungshaltung an eine schnelle, besonders umfangreiche oder überhaupt eine Antwort angebracht. Die englischsprachigen Foren finden Sie unter *wordpress.org/support/forums/*.

1.5.5 Buchtipps

Wenn Sie tiefer in die Entwicklung einsteigen möchten, als hier im Buch Platz dafür ist, werden Sie unter anderem – natürlich – beim Rheinwerk-Verlag fündig, was Literatur angeht. Zum Beispiel mit dem Titel zur Entwicklung von Plugins und Themes: *https://www.rheinwerk-verlag.de/5524*.

TEIL I

WordPress installieren und bedienen

2 WordPress-Installation in wenigen Minuten mit Playground

> *Wenn Sie WordPress ausprobieren möchten, tun Sie das am besten lokal, nutzen also nicht direkt einen Webhoster. Es gibt unterschiedlich komplexe Varianten für lokale Installationen, die sich auch in den Möglichkeiten unterscheiden, die Sie als Anwender*in haben. In diesem Kapitel werden Sie Playground kennenlernen, das Ihnen sehr einfach ermöglicht, WordPress auszuprobieren.*

In der Vorauflage dieses Buchs wurde als einfachste Möglichkeit zur lokalen Installation ein Bitnami-Paket empfohlen. Das ist allerdings erstens inzwischen nicht mehr so einfach wie bei Entstehung der Vorauflage, und zweitens gibt es jetzt direkt vom WordPress-Projekt eine noch einfachere Möglichkeit.

2.1 WordPress Playground – eine ganze WordPress-Installation lokal im Browser

Unter dem Titel *WordPress Playground* wurde 2022 ein Projekt gestartet, das eine komplette WordPress-Installation im Browser laufen lässt. Damit entfällt die Notwendigkeit, sich um einen Webserver kümmern zu müssen, bevor WordPress ausprobiert werden kann. Technische Details zu Playground können Sie in

einem web.dev-Artikel nachlesen (*web.dev/articles/wordpress-playground*).

Unter *playground.wordpress.net* können Sie Playground aufrufen. Daraufhin öffnet sich eine WordPress-Installation, die nur lokal in Ihrem Browser existiert und zurückgesetzt wird, sobald Sie den Browser-Tab neu laden (Abbildung 2.1). Oben finden Sie die Adminbar, über die Sie zum Backend gelangen. Falls Sie sich einmal ausloggen, sind die Zugangsdaten »admin« und »password«.

Ganz oben in der Ansicht, noch über der Adminbar, sehen Sie einen Bereich mit einer Adressleiste und links und rechts davon je einem Icon. Das linke Icon, das ein Raster von vier Kästen anzeigt, öffnet eine zwei- oder dreispaltige Ansicht, abhängig von der Größe Ihres Viewports (siehe Abbildung 2.2).

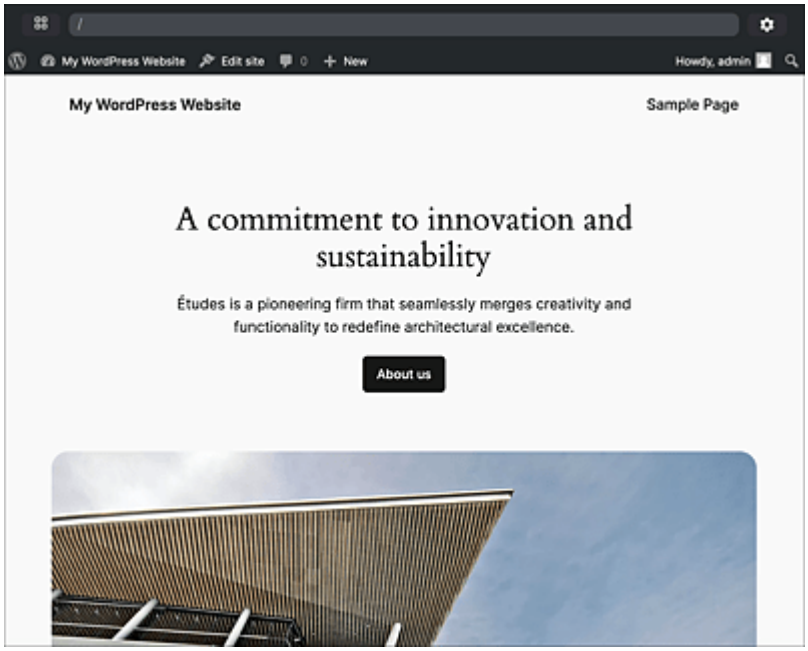


Abbildung 2.1 Wenn Sie »playground.wordpress.net« aufrufen, wird in Ihrem Browser eine frische WordPress-Installation erstellt.

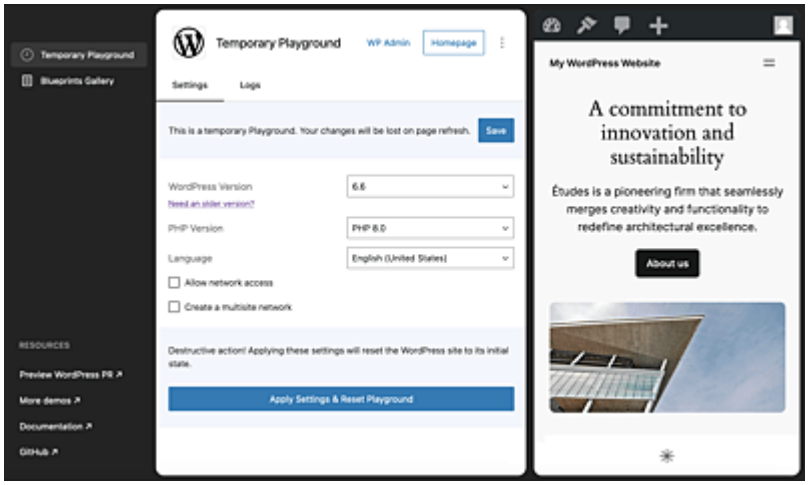


Abbildung 2.2 In dieser Ansicht des Playground haben Sie die Möglichkeit, links aus fertigen Playground-Blueprints zu wählen und in der Mitte die Einstellungen der Playground-Instanz zu ändern.

Links sehen Sie eine Seitenleiste, in der der Punkt Temporary Playground aktiv ist. Über den Button Blueprints Gallery gelangen Sie in eine Liste von vorkonfigurierten Playgrounds, die Sie über einen Klick auswählen können, sodass die entsprechende Konfiguration angewendet wird (Sie lernen Blueprints später in diesem Abschnitt kennen). In der zweiten Spalte von links können Sie die Einstellungen des Playgrounds anpassen. So ist es hier möglich, die WordPress- und PHP-Version anzupassen, die Sprache (Language) der Installation einzustellen sowie Netzwerkzugriff zu erlauben (Allow network access) und WordPress als Multisite-Installation zu erstellen (dem Thema Multisite widmet sich das Buch in Kapitel 7). Die Netzwerkzugriff-Option ist notwendig, wenn Sie direkt über das Backend von WordPress Plugins und/oder Themes aus dem offiziellen WordPress.org-Verzeichnis installieren möchten.

Nachdem Sie Änderungen bei den Einstellungen vorgenommen haben, müssen Sie sie mit einem Klick auf den Button Apply Settings & Reset Playground anwenden. Wie der Button schon sagt, verlieren Sie dabei Änderungen, die Sie am Playground vorgenommen haben.

Auf diese Einstellungen haben Sie auch über das Zahnrad-Icon rechts neben der Adressleiste Zugriff (siehe Abbildung 2.3).

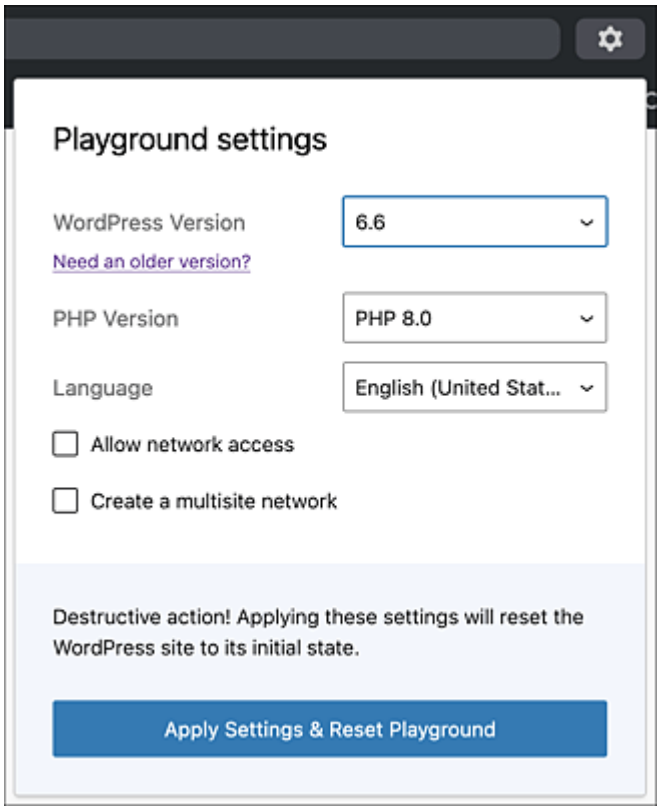


Abbildung 2.3 Einstellungen wie PHP- und WordPress-Version können auch über das Zahnrad-Icon rechts neben der Adressleiste erreicht werden.

Sie können den Stand des Playgrounds optional im Browser oder einem lokalen Verzeichnis speichern, sodass er auch beim Neuladen eines Tabs erhalten bleibt. Klicken Sie dafür in den Einstellungen des Playgrounds auf den Save-Button (Abbildung 2.4). Je nach Browser sind nicht alle Optionen verfügbar.

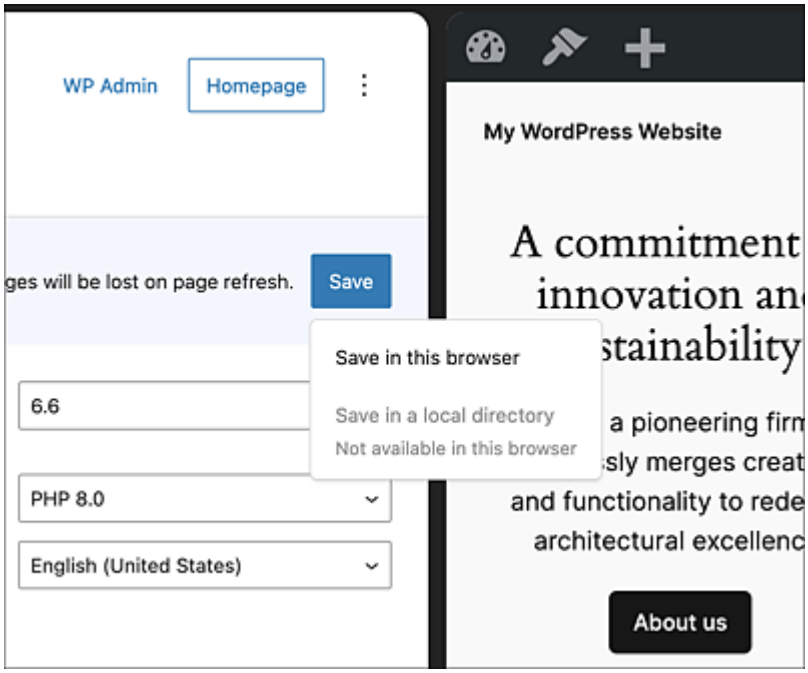


Abbildung 2.4 Sie können den Playground im Browser oder – in Browsern, die dies unterstützen – lokal speichern.

2.1.1 Plugins vorinstallieren

Wenn Sie ein oder mehrere Plugins aus dem WordPress.org-Verzeichnis in der Playground-Instanz installieren und aktivieren möchten, können Sie das direkt über die URL von Playground tun. Dafür nutzen Sie den Parameter *plugin* und übergeben als Wert den Slug des Plugins aus dem Plugin-Verzeichnis (den können Sie aus der URL entnehmen, wenn Sie das Plugin auf *WordPress.org/plugins* aufrufen – der Teil nach */plugins/* ist der Slug).

Mit der folgenden URL würden Sie beispielsweise eine Playground-Installation mit dem Antispam-Plugin *Antispam Bee* erstellen.

```
https://playground.wordpress.net/?plugin=antispam-bee
```

Sie können auch mehrere Plugins installieren, indem Sie den `plugin`-Parameter häufiger übergeben:

```
https://playground.wordpress.net/?plugin=antispam-bee&plugin=webmention
```

2.1.2 Theme vorinstallieren

Um direkt beim Aufruf ein bestimmtes Theme aus dem WordPress.org-Theme-Verzeichnis zu aktivieren, können Sie den `theme`-Parameter an die URL hängen und den Slug des Themes übergeben. Mit der folgenden URL erstellen Sie eine Instanz mit dem Vermeer-Theme:

```
https://playground.wordpress.net/?theme=vermeer
```



Abbildung 2.5 Das Theme ist nicht nur vorinstalliert, sondern auch direkt aktiv.

2.1.3 Theme- und Plugin-Parameter kombinieren

Natürlich können Sie auch beide Fälle kombinieren. Für eine Website mit Vermeer und den Plugins Antispam Bee und Webmention nutzen Sie zum Beispiel diese URL:

```
https://playground.wordpress.net/?theme=vermeer&plugin=antispam-
bee&plugin=webmention
```

2.1.4 Eine Playground-Website speichern und importieren

Wenn Sie eine Instanz über *playground.wordpress.net* erstellt haben, finden Sie in den Einstellungen des Playgrounds (diese erreichen Sie über einen Klick ganz oben links auf das Icon mit den vier Kästen) oben rechts über das Drei-Punkte-Menü die Möglichkeit, die Website als ZIP zu speichern (Download as .zip) und eine Playground-ZIP zu importieren (Restore from .zip).

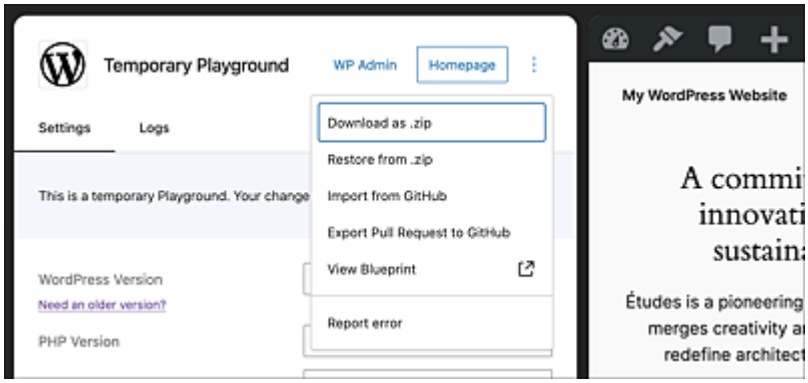


Abbildung 2.6 Obwohl die Instanz temporär in Ihrem Browser entsteht, können Sie sie als ZIP exportieren und später wieder importieren, um weiterzuarbeiten.

2.2 wp-now: Playground für Entwickler*innen

Wenn Sie ein Plugin oder Theme programmieren, können Sie das CLI-Tool (CLI steht für Command Line Interface, ein Kommandozeilen-Interface) *wp-now* verwenden, um lokal eine Playground-Instanz zu erstellen, mit der Sie den aktuellen Entwicklungsstand testen können.

2.2.1 Installation von wp-now

Alles, was Sie zur Installation von wp-now auf Ihrem lokalen System benötigen, ist eine aktuelle Version von Node.js (*nodejs.org/en/download*). Um alle Funktionen von wp-now verwenden zu können, ist aktuell (November 2024) die Node.js-Version 20 nötig.

Wenn Sie Node.js installiert haben, können Sie wp-now auf unterschiedlichen Wegen einsetzen, die in der Doku beschrieben sind (*github.com/WordPress/playground-tools/tree/trunk/packages/wp-now#usage*). Der einfachste Weg ist die Nutzung über `npx`. Das ist ein Node.js-Befehl, mit dem ein Paket ausgeführt werden kann, unabhängig davon, ob es lokal installiert ist oder nicht (*docs.npmjs.com/cli/v10/commands/npx*). Wenn lokal keine Installation gefunden wird, wird der Befehl im Hintergrund im Cache installiert und dann ausgeführt.

Alternativ können Sie wp-now mit dem folgenden Befehl global installieren, um von jedem Verzeichnis aus mit `wp-now` darauf zugreifen zu können.

```
npm install -g @wp-now/wp-now
```

2.2.2 WordPress-Instanz mit wp-now erstellen

Abhängig davon, ob Sie `npx` nutzen oder wp-now lokal installiert haben, müssen Sie im Folgenden die Befehle anpassen. Mit `npx` führen Sie wp-now wie folgt aus:

```
npx @wp-now/wp-now
```

Wenn Sie es lokal installiert haben, nutzen Sie:

```
wp-now
```

Der Befehl zum Erstellen einer WordPress-Installation ist `start`. Abhängig davon, in welcher Art Ordner Sie sich befinden, unterscheidet sich die Installation:

- Wenn Sie sich in dem Verzeichnis mit den Dateien eines WordPress-Themes oder -Plugins befinden, wird bei der erstellten WordPress-Installation dieses Plugin oder Theme direkt aktiviert sein, und Änderungen an den Dateien spiegeln sich sofort in der Installation wider.
- Wenn Sie sich im `wp-content`-Verzeichnis einer WordPress-Installation befinden, wird der gesamte Ordner auch für die Playground-Instanz verwendet.
- Wenn Sie den Befehl im obersten Verzeichnis einer WordPress-Installation ausführen, werden diese Dateien für das wp-now-WordPress genutzt.
- Wird der Befehl in dem obersten Verzeichnis von `wordpress-develop`, also der Entwicklungs-Ordnerstruktur des WordPress-Core, ausgeführt, gilt dasselbe Verhalten wie bei den Dateien einer normalen WordPress-Installation, nur dass das *build*-

Verzeichnis für die Ausgabe der Website genutzt wird. Nützlich, wenn Sie an WordPress selbst entwickeln wollen.

- Wenn Sie den Befehl in einem WordPress-unabhängigen Verzeichnis mit `index.php` ausführen, wird ein PHP-Webserver gestartet, der die `index.php` ausliefert.
- Wenn Sie den `start`-Befehl in einem anderen Verzeichnis ausführen, wird eine Standard-WordPress-Installation mit dem Default-Theme erstellt (das ist der »playground«-Modus).

Falls Sie an einem Theme oder Plugin arbeiten möchten, wechseln Sie nun in das entsprechende Verzeichnis und führen den `start`-Befehl aus. Wenn Sie gerade kein konkretes Beispiel haben, an dem Sie wp-now ausprobieren möchten, können Sie es einfach in einem beliebigen Verzeichnis ausführen:

```
wp-now start
```

Oder, wenn Sie `npx` nutzen:

```
npx @wp-now/wp-now start
```

Info: Ab hier ohne npx

Ab hier wird nur noch die Variante mit `wp-now` gezeigt. Alle Beispiele lassen sich aber genauso auch mit `npx @wp-now/wp-now` ausführen.

Wenn Sie den Befehl ausgeführt haben, sollten Sie in der Konsole Statusinfos bekommen, etwa dass WordPress heruntergeladen und SQLite installiert wird (wenn Sie den Befehl weitere Male ausführen und die Komponenten lokal schon vorliegen, werden sie nicht erneut heruntergeladen, sondern wird die lokale Version genutzt).

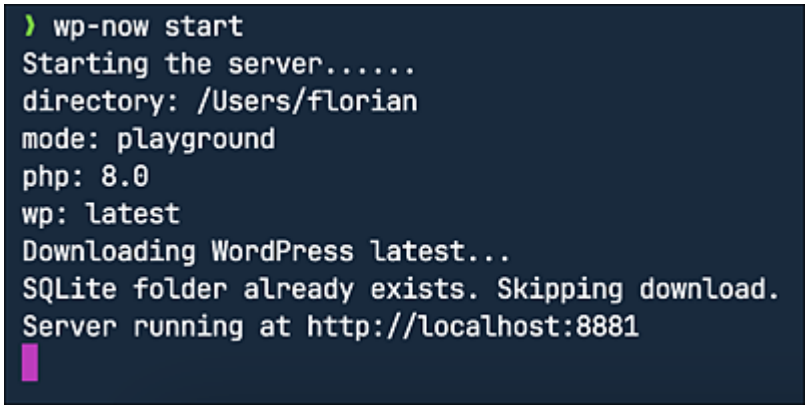


Abbildung 2.7 So sieht beispielhaft die Ausgabe von wp-now im Terminal aus.

Am Ende bekommen Sie eine `http://localhost`-URL mit einem Port, unter der Sie die Website erreichen können, und die Site wird gegebenenfalls automatisch im Browser geöffnet.

2.2.3 Verfügbare Parameter zur Anpassung einer WordPress-Instanz

Eine der Stärken von Playground und damit wp-now ist, dass Sie einfach eine andere PHP- und/oder WordPress-Version nutzen können. So können Sie mit `--php` die PHP-Version und mit `--wp` die WordPress-Version festlegen.

Um zum Beispiel eine Playground-Instanz mit PHP 8.2 und WordPress 6.4 zu erstellen, führen Sie den folgenden Befehl aus.

```
wp-now start --php=8.2 --wp=6.4
```

2.2.4 Speicherorte der Dateien

Anders als bei der Playground-Nutzung über *playground.wordpress.net* (wenn Sie nicht die Speicherung der Daten aktiviert haben) speichert wp-now die Daten im Verzeichnis *.wp-now* im Benutzerordner.

Da der Ordnername mit einem Punkt beginnt, müssen Sie vielleicht erst die Anzeige von solchen »versteckten« Ordnern und Dateien aktivieren. Wie das geht, ist abhängig von Ihrem genutzten Betriebssystem, eine Suche nach »versteckte Dateien anzeigen {Betriebssystem-Name}« sollte aber schnell ans Ziel führen.

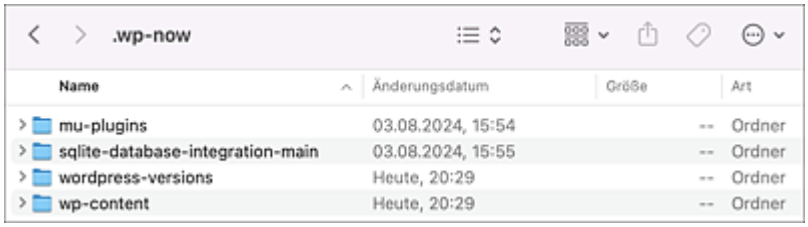


Abbildung 2.8 So in etwa sieht der Inhalt des ».wp-now«-Verzeichnisses nach Erstellen einer Instanz aus.

In dem Verzeichnis finden Sie die folgenden Ordner, in denen wiederum jeweils pro Playground-Site ein Verzeichnis liegt:

- *sqlite-database-integration-main* – hier sind die SQLite-Datenbanken gespeichert.
- *wordpress-versions* – beinhaltet die WordPress-Core-Dateien.
- *wp-content* – hier finden Sie die *wp-content*-Dateien für wp-now-Instanzen, die aus einem Plugin-, Theme- oder *wp-content*-Verzeichnis ausgeführt wurden, oder im »playground«-Modus.
- *mu-plugins* – hier liegen die Inhalte vom *mu-plugins*-Verzeichnis für dieselben Instanzen wie bei *wp-content*.

2.3 Playground mit Blueprints vorkonfigurieren

Blueprints sind eine Playground-Funktion, mit der Sie Aktionen definieren können, die bei der Erstellung der Playground-Instanz ausgeführt werden. Beispielsweise können Sie bestimmte Plugins installieren, Konstanten in der *wp-config.php* setzen oder auch beliebigen PHP-Code ausführen, sodass Sie nach Erstellung einer Playground-Instanz eine bereits vorkonfigurierte Installation haben.

Blueprints können sowohl mit *playground.wordpress.net* als auch mit wp-now genutzt werden.

2.3.1 Blueprint erstellen

Blueprints sind in JSON geschrieben, und ein ganz einfaches Blueprint sieht so aus:

```
{
    "$schema": "https://playground.wordpress.net/blueprint-schema.json",
    "preferredVersions": {
        "php": "8.2",
        "wp": "latest"
    }
}
```

Listing 2.1 Ein Playground-Blueprint, das die PHP- und WordPress-Version festlegt

Die Angaben bedeuten Folgendes:

- `$schema` ist optional, aber hilfreich, da Code-Editoren damit Autovervollständigung für Blueprints bieten können – wenn Sie die URL aufrufen, sehen Sie eine JSON-Datei mit allen möglichen Optionen.

- Mit `preferredVersions` geben Sie an, dass die Website mit PHP 8.2 laufen und die neueste WordPress-Version installiert werden soll.

2.3.2 Blueprint anwenden

Die einfachste Methode ist wohl die Nutzung des Blueprint-Builders unter *playground.wordpress.net/builder/builder.html*. Hier haben Sie auf der linken Seite ein Feld für das Blueprint und sehen rechts das Ergebnis in Form der WordPress-Instanz.

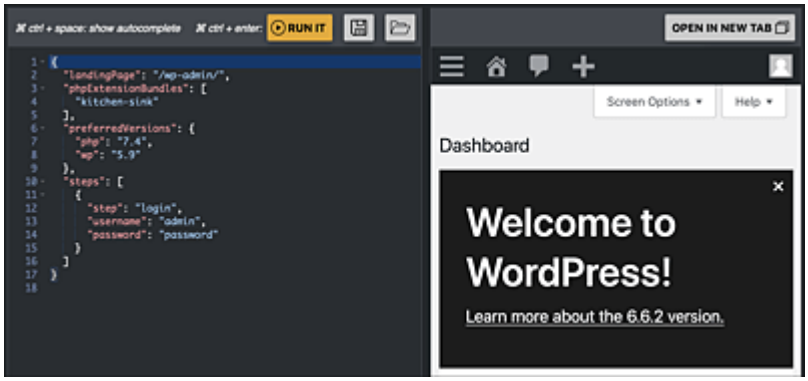


Abbildung 2.9 Der Blueprint-Builder ist ein nützliches Tool beim Erstellen von Blueprints.

Eine weitere, sehr einfache Art des Testens ist, das Blueprint als URL-Fragment an *playground.wordpress.net* zu übergeben:

```
https://playground.wordpress.net/#{"$schema":
"https://playground.wordpress.net/blueprint-schema.json","preferredVersions":
{"php": "8.2","wp": "latest"}}
```

Alternativ können Sie mit dem *blueprint-url*-Query-Parameter auch eine URL zu einer öffentlich zugänglichen Blueprint-Datei angeben.

Auch bei wp-now können Blueprints verwendet werden. Dafür geben Sie beim Aufruf des `start`-Befehls mit dem Parameter `--blueprint` den Pfad zu einer Blueprint-JSON-Datei an, wie im folgenden Beispiel:

```
wp-now start --blueprint=./blueprint.json
```

2.3.3 Verfügbare Blueprint-Optionen

Vermutlich haben Sie es sich schon gedacht: Die Möglichkeiten von Blueprints sind nicht nach der Angabe von PHP- und WordPress-Version erschöpft. Hier werden Sie die aktuell (November 2024) verfügbaren Optionen kennenlernen, die Sie in Ihrer Blueprint-JSON-Datei verwenden können. Immer aktuell ist die Dokumentation unter *wordpress.github.io/wordpress-playground/blueprints-api/index*.

Die Pfade zu den JSON-Keys werden mit Punkten getrennt. Wenn Sie also beispielsweise von `preferredVersions.php` lesen, meint das den folgenden Eintrag im JSON:

```
{
    […],
    "preferredVersions": {
        "php": "8.2",
        […]
    }
}
```

Listing 2.2 Mit der Angabe »preferredVersions.php« ist das »php«-Attribut des »preferredVersions«-Objekts gemeint.

> **Info: Die Werte lassen sich beliebig kombinieren**
>
> Die Beispiel-Blueprints enthalten hier nur die jeweils besprochenen Werte, Sie können diese aber beliebig kombinieren.

Zu Beginn eine bestimmte Seite öffnen

Über den Schlüssel `landingPage` können Sie eine URL angeben, die nach der Abarbeitung des Blueprints aufgerufen wird. Um zum Beispiel direkt die »About«-Seite aufzurufen, können Sie folgendes Blueprint definieren.

```
{
    "landingPage": "/?page_id=2"
}
```

Metadaten angeben

Wenn Sie ein Blueprint nicht nur für sich allein erstellen, ist es eine gute Idee, Metainformationen hinzuzufügen, etwa was das Blueprint macht und wer es erstellt hat. Auch wenn Sie nur für sich Blueprints erstellen, kann das übrigens nicht schaden – Ihr Zukunfts-Ich wird Ihnen dankbar sein, wenn ein komplexes Blueprint eine Beschreibung enthält, was passiert.

Die Metadaten werden unter dem Schlüssel `meta` angegeben, und es können vier Werte übergeben werden:

- `meta.title` beinhaltet den Titel für das Blueprint.
- `meta.description` bietet Platz für eine kurze Beschreibung davon, was das Blueprint tut.
- `meta.author` – hier geben Sie Ihren Namen ein.
- `meta.categories` kann ein Array von Kategorien enthalten.

Diese Metadaten können in einem zukünftig existierenden Blueprints-Bereich auf WordPress.org genutzt werden, `title` und `author` sind in dem Blueprint-Schema als erforderlich markiert.

Ein Meta-Blueprint könnte so aussehen:

```
{
    "meta": {
        "title": "Metadaten-Blueprint",
        "description": "Ein Blueprint nur aus Metadaten",
        "author": "Florian Brinkmann",
        "categories": ["meta", "example"]
    }
}
```

Listing 2.3 In einem Blueprint können verschiedene Metadaten angegeben werden.

WordPress- und PHP-Version angeben

Als erstes Beispiel eines Blueprints haben Sie bereits die beiden Werte `preferredVersions.php` und `preferredVersions.wp` kennengelernt.

- Für PHP können Sie zum Zeitpunkt der Entstehung dieser Seiten (November 2024) alle Minor-Versionen von 7.0 bis 8.3 angeben (bis auf den Patch-Level genau, also etwa 8.1.2, funktioniert nicht), oder `latest`, damit immer die neueste unterstützte Version genutzt wird.
- Auch bei WordPress können Sie die Version nicht bis auf die dritte Versionsnummer genau angeben, sondern nur auf die zweite Nummer einschränken, zum Beispiel `6.4` für die neueste Version des 6.4-Zweigs.
  Standardmäßig können Sie aber nicht beliebig alte Versionen installieren, aktuell (November 2024) ist WordPress 6.2 die ältestmögliche Version. Sie können über einen Umweg eine ältere Version installieren, wie hier beschrieben: *wordpress.github.io/wordpress-playground/blueprints-api/examples/#load-an-older-wordpress-version* (wegen technischer Beschränkungen ist 5.9.9 aber die älteste Version, die möglich ist).

Zusätzliche Funktionen aktivieren oder deaktivieren

Unter dem `features`-Schlüssel ist momentan nur der Eintrag `networking` möglich. Darüber kann gesteuert werden, ob die Playground-Instanz Requests nach außen machen kann, was beispielsweise für die Installation von Plugins oder Themes aus den WordPress.org-Verzeichnissen direkt aus dem Backend heraus

notwendig ist. `features.networking` ist standardmäßig auf `false` gesetzt.

Ein Blueprint, um die Networking-Funktion zu aktivieren, sieht wie folgt aus:

```
{
    "features": {
        "networking": true
    }
}
```

Listing 2.4 Dieses Blueprint aktiviert die Networking-Funktion von Playground, damit beispielsweise Plugins direkt aus dem Backend heraus installiert werden können.

WordPress-Benutzer*in einloggen

Mit dem Key `login` können Sie angeben, ob direkt ein*e Benutzer*in angemeldet werden soll. Hier gibt es die Wahl zwischen

- `true` oder `false` (bei `true` wird der User `admin` mit dem Passwort `password` angemeldet),
- einem Objekt aus `username` und `password`, um einen beliebigen User anzumelden.

Nachfolgend finden Sie ein Beispiel für einen Login des Standard-Users unter konkreter Angabe von User und Passwort. `"login": true` würde dasselbe Ergebnis liefern.

```
{
    "login": {
        "username": "admin",
        "password": "password"
    }
}
```

Listing 2.5 Mit diesem Blueprint wird angegeben, dass der »admin«-User mit dem Passwort »password« eingeloggt werden soll.

Info: Blueprint-Kurzformen

Es gibt ein paar Kurzformen bei den Blueprints. Die Nutzung von `login` ist letztlich eine Kurzform des `login`-Steps, der weiter unten erwähnt wird.

Bundle von PHP-Erweiterungen

Standardmäßig werden bei Playground die folgenden PHP-Erweiterungen geladen: `gd`, `mbstring`, `iconv`, `openssl`, `libxml`, `xml`, `dom`, `simplexml`, `xmlreader` und `xmlwriter`, was einer Dateigröße von ungefähr 6 MB entspricht.

Über die Angabe von `"phpExtensionBundles": "light"` kann das deaktiviert werden. Hier müssen Sie aufpassen und genau testen, ob nach der Deaktivierung noch alles wie gewünscht funktioniert, da es sich dabei um recht grundlegende Erweiterungen handelt.

Um explizit anzugeben, dass die Erweiterungen geladen werden, nutzen Sie den Wert `kitchen-sink`.

Zusätzliche Tools installieren

Mit `extraLibraries` können Sie ein Array von Tools angeben, die zusätzlich installiert werden können. Im November 2024 ist nur `wp-cli` möglich, um das WP-CLI zu installieren.

Konstanten setzen

Über den `constants`-Key können Sie PHP-Konstanten setzen, die für alle Aufrufe der Playground-Website gesetzt werden. `constants` ist

eine Kurzform des `defineWpConfigConsts`-Schritts, den Sie später kennenlernen.

```
{
    "constants": {
        "WP_DEBUG": true,
        "WP_DEBUG_LOG": true,
        "WP_DEBUG_DISPLAY": false
    }
}
```

Plugins vorinstallieren

Über `plugins` können Sie ein Array von Plugin-Slugs oder URLs zu Plugin-ZIP-Dateien angeben, damit diese Plugins automatisch bei der Erstellung der Playground-Website installiert und aktiviert werden. Kurzform des `installPlugin`-Schritts, der gleich behandelt wird:

```
{
    "plugins": {
        "antispam-bee",
        "webmention"
    }
}
```

Site-Options setzen

Mit `siteOptions`, einer Kurzform für den Schritt `setSiteOptions`, können Sie Werte in der `wp_options`-Tabelle von WordPress setzen, um beispielsweise den Titel der Website zu ändern:

```
{
    "siteOptions": {
        "blogname": "Eine WordPress-Installation"
    }
}
```

Schritte definieren

Die Option, die Sie bei einem Blueprint wahrscheinlich am intensivsten nutzen werden, ist die Definition von Schritten unter dem Schlüssel `steps`. `steps` beinhaltet ein Array von Step-Objekten, bei denen es immer die Eigenschaft `step` gibt, und abhängig vom definierten Schritt weitere Eigenschaften.

In den Beispielen wird immer nur ein Schritt einzeln behandelt, aber wie bei den bisherigen Beispielen können Sie die Schritte einfach kombinieren, indem Sie die Objekte untereinander in das `steps`-Array schreiben.

- **Plugin installieren**

  Der Schritt `installPlugin` ermöglicht Ihnen die Installation eines Plugins. Innerhalb von `pluginZipFile` können Sie unterschiedliche Werte für `resource` übergeben. Hier werden drei davon vorgestellt:

  - `wordpress.org/plugins` erlaubt die Installation eines Plugins aus dem WordPress.org-Plugin-Verzeichnis. Wenn Sie diesen Wert für `pluginZipFile.resource` setzen, muss der Slug des zu installierenden Plugins über `pluginZipFile.slug` angegeben werden, wie im folgenden Beispiel am Beispiel von Antispam Bee gezeigt.

```
{
    "steps": [
        {
            "step": "installPlugin",
            "pluginZipFile": {
                "resource": "wordpress.org/plugins",
                "slug": "antispam-bee"
            },
            "options": {
                "activate": true
            }
        }
    ]
}
```

Listing 2.6 Ein Blueprint zur Installation und Aktivierung des Antispam-Bee-Plugins aus dem WordPress.org-Verzeichnis

- Im Beispiel sehen Sie noch die Möglichkeit, das Plugin über `options.activate` direkt nach der Installation zu aktivieren.
- Wenn Sie `pluginZipFile.resource` auf den Wert `url` setzen, können Sie mit der zusätzlichen Eigenschaft `url` auf eine ZIP verweisen, die öffentlich über eine URL zugänglich ist.

```
{
    "steps": [
        {
            "step": "installPlugin",
            "pluginZipFile": {
                    "resource": "url",
                    "url": "https://example.com/plugin.zip"
            }
        }
    ]
}
```

Listing 2.7 So können Sie ein Plugin von einer URL installieren

- Der Wert `vfs` erlaubt Ihnen, mit `pluginZipFile.path` den Pfad zu einem Plugin-ZIP anzugeben, um dieses zu installieren.

```
{
    "steps": [
        {
            "step": "installPlugin",
            "pluginZipFile": {
                "resource": "vfs",
                "path": "/wordpress/plugin.zip"
            }
        }
    ]
}
```

Listing 2.8 Auch ein lokales ZIP kann installiert werden.

- **Installiertes Plugin aktivieren**

  Mit dem Schritt `activatePlugin` können Sie ein bereits installiertes Plugin aktivieren. Dafür muss der Pfad zum Plugin

oder der *Basename* angegeben werden. Der Basename besteht aus dem Ordner- und dem Dateinamen. Um beispielhaft das Plugin Antispam Bee zu aktivieren, können Sie folgenden Schritt verwenden.

```
{
    "steps": [
        {
            "step": "activatePlugin",
            "pluginName": "Antispam Bee",
            "pluginPath": "antispam-bee/antispam_bee.php"
        }
    ]
}
```

Listing 2.9 Wenn ein Plugin nicht direkt bei der Installation aktiviert wird, kann das in einem eigenen Schritt erledigt werden.

- **Theme installieren**

  Die Installation eines Themes ist mit dem Schritt `installTheme` möglich und bietet sehr ähnliche Optionen wie `installPlugin`. Statt `pluginZipFile` gibt es `themeZipFile`, aber für die Werte `vfs` und `url` in `themeZipFile.resource` sind die Optionen identisch wie für Plugins beschrieben.

  Wenn Sie ein Theme aus dem WordPress.org-Verzeichnis installieren möchten, nutzen Sie für `themeZipFile.resource` den Wert `wordpress.org/themes` und geben über `themeZipFile.slug` den Slug des Themes an. Im folgenden Listing wird das Vermeer-Theme installiert.

```
{
    "steps": [
        {
            "step": "installTheme",
            "themeZipFile": {
                "resource": "wordpress.org/themes",
                "slug": "vermeer"
            }
        }
    ]
}
```

Listing 2.10 Beispiel eines Blueprints zur Installation des Vermeer-Themes von WordPress.org

- **Theme aktivieren**

  Um ein Theme zu aktivieren, können Sie den `activateTheme`-Schritt nutzen, dem Sie den Namen des Theme-Ordners mitgeben:

```
{
    "steps": [
        {
            "step": "activateTheme",
            "themeFolderName": "vermeer"
        }
    ]
}
```

Listing 2.11 Der Schritt zur Aktivierung eines Themes

- **Multisite aktivieren**

  Um die Multisite-Funktion von WordPress zu aktivieren, gibt es den Schritt `enableMultisite`:

```
{
    "steps": [
        {
            "step": "enableMultisite"
        }
    ]
}
```

Listing 2.12 Ganz unspektakulär kurz, aber sehr nützlich: die Aktivierung von Multisite in einem Blueprint

  Intern geht dieser Schritt die Aktivierung von Multisite durch, wie Anwender*innen es auch machen würden:

  - Die `WP_ALLOW_MULTISITE`-Konstante wird gesetzt.
  - Es wird geprüft, ob die WordPress-URL eine Portangabe hat oder der Host `localhost` ist. In diesen Fällen funktioniert die

Aktivierung nicht.

- Alle aktiven Plugins werden deaktiviert.
- Die Multisite-Installation wird durchgeführt.
- Die weiteren Multisite-Konstanten werden gesetzt.
- Die vor der Installation aktiven Plugins werden wieder aktiviert.

- **Website-URL ändern**

  Um die URL einer Website festzulegen, können Sie den `defineSiteUrl`-Schritt nutzen. Der Schritt setzt die `WP_HOME`- und `WP_SITEURL`-Konstanten auf den Wert, den Sie mit `siteUrl` angeben.

```
{
    "steps": [
        {
            "step": "defineSiteUrl",
            "siteUrl": "https://example.com"
        }
    ]
}
```

Listing 2.13 Über »defineSiteUrl« können Sie die Website-URL anpassen.

Konstanten in der wp-config.php setzen

Sie haben eben schon einige Schritte gesehen, die Konstanten setzen, um die WordPress-Installation zu beeinflussen. Über den Schritt `defineWpConfigConsts` können Sie beliebige Konstanten in der `wp-config.php` setzen, etwa um den Debug-Mode zu aktivieren und die Fehler in einer Log-Datei zu speichern, ohne sie direkt ausgeben zu lassen.

```
{
    "steps": [
        {
```

```
            "step": "defineWpConfigConsts",
            "consts": {
                "WP_DEBUG": true,
                "WP_DEBUG_LOG": true,
                "WP_DEBUG_DISPLAY": false
            }
        }
    ]
}
```

Listing 2.14 Bereit zum Debuggen mit den über das Blueprint gesetzten Debugging-Konstanten in der »wp-config.php«

Neben dem `step` geben Sie in `consts` ein Objekt von Konstantennamen und Werten an, die gesetzt werden sollen.

Darüber hinaus gibt es noch eine weitere optionale Angabe: `method` ermöglicht Ihnen zu steuern, wie die Konstanten gesetzt werden sollen.

Der Standardwert ist `rewrite-wp-config`, was bedeutet, dass die Konstanten per `define` in die `wp-config.php`-Datei geschrieben werden. Der Vorteil dieser Methode ist, dass der Wert einer bereits definierten Konstante aktualisiert wird und es so keine Konflikte gibt, wenn dieselbe Konstante mehrfach definiert wird.

Der zweite mögliche Wert ist `define-before-run`, was bedeutet, dass die Konstante(n) vor der Ausführung des Skripts definiert werden, ohne dass die `wp-config.php` verändert wird. Dadurch kann es zu Problemen mit mehrfach definierten Konstanten kommen.

- **Site-Optionen setzen**

  Wenn Sie während der WordPress-Installation schon bestimmte Werte für Optionen setzen möchten, wie es über `update_option` möglich ist, können Sie den `setSiteOptions`-Schritt verwenden. So können Sie beispielsweise den Namen der WordPress-Site setzen:

  ```
  {
      "steps": [
          {
  ```

```
            "step": "setSiteOptions",
            "options": {
                "blogname": "Eine WordPress-Installation"
            }
        }
    ]
}
```

Listing 2.15 So können Sie eine Option in der Datenbank setzen oder anpassen

- **WordPress-Export-Datei importieren**

  Wenn Sie die normale Exportfunktion von WordPress nutzen, werden die Inhalte als `.wxr`-Datei gespeichert (WXR steht für »WordPress eXtended RSS«). Eine solche Datei können Sie mit dem Schritt `importWxr` importieren, indem Sie über `file` eine Datei angeben, wobei Sie die Ressourcen-Typen nutzen können, die Sie weiter oben beim Installieren eines Plugins kennengelernt haben.

  Ein Beispiel für den Import einer öffentlich zugänglichen Datei über eine URL sieht wie folgt aus:

```
{
    "steps": [
        {
            "step": "importWxr",
            "file": {
                "resource": "url",
                "url": "https://example.com/export.wxr"
            }
        }
    ]
}
```

Listing 2.16 Wenn Sie eine Installation mit Starter-Inhalt beginnen möchten, bietet sich der Import eines WordPress-Daten-Exports an.

- **WordPress-Dateien importieren**

  Wenn Sie über ein Blueprint den Stand einer bestehenden Site abbilden möchten, ist es das Einfachste, wenn der Inhalt des *wp-content*-Ordners importiert werden würde. Mit dem

`importWordPressFiles`-Schritt ist das möglich. Sie geben bei dem Schritt über `wordPressFilesZip` einen Verweis auf ein ZIP an und optional mit `pathInZip` den Pfad zu den Ordnern innerhalb der ZIP:

```
{
    "steps": [
        {
            "step": "importWordPressFiles",
            "wordPressFilesZip": {
                "resource": "url",
                "url": "https://example.com/wp-files.zip"
            }
        }
    ]
}
```

Listing 2.17 Eine andere Variante, nicht bei null zu starten, ist der Import eines gesamten »wp-content«-Ordners.

- **Benutzer*in anmelden**

  Mit dem `login`-Schritt können Sie unter Angabe von `username` und `password` eine*n Benutzer*in einloggen:

```
{
    "steps": [
        {
            "step": "login",
            "username": "admin",
            "password": "password"
        }
    ]
}
```

Listing 2.18 Mit dem »login«-Schritt melden Sie eine*n Benutzer*in an.

- **Weitere Schritte**

  In der Playground-Doku finden Sie im Abschnitt zur Blueprints-API alle vorhandenen Schritte (*wordpress.github.io/wordpress-playground/blueprints-api/steps*). Neben den beschriebenen WordPress-spezifischen Schritten gibt es noch viele weitere,

beispielsweise um Dateien zu kopieren, PHP-Code oder SQL-Statements auszuführen und mehr.

So können Sie sehr komplexe Blueprints bauen, mit denen Sie eine Installation erstellen können, die genau Ihren Start-Anforderungen entspricht.

2.4 Komfortablere Playground-Nutzung mit »Studio« von WordPress.com

Ein neues Produkt von WordPress.com ist *Studio* (*https://developer.wordpress.com/studio/*), eine App, mit der lokale WordPress-Installationen erstellt und verwaltet werden können. Im Hintergrund setzt es auf Playground, ist also quasi ein Wrapper für eine einfachere Benutzbarkeit und Verwaltung mehrerer Sites.

Nach dem Download und der Installation von Studio (es gibt Studio im November 2024 für macOS und Windows) erscheint nach dem Start direkt eine Oberfläche, in der der Name der zu erstellenden WordPress-Installation sowie der Pfad für die Dateien angegeben werden müssen. Nach einem Klick auf Weiter wird die Website erstellt.

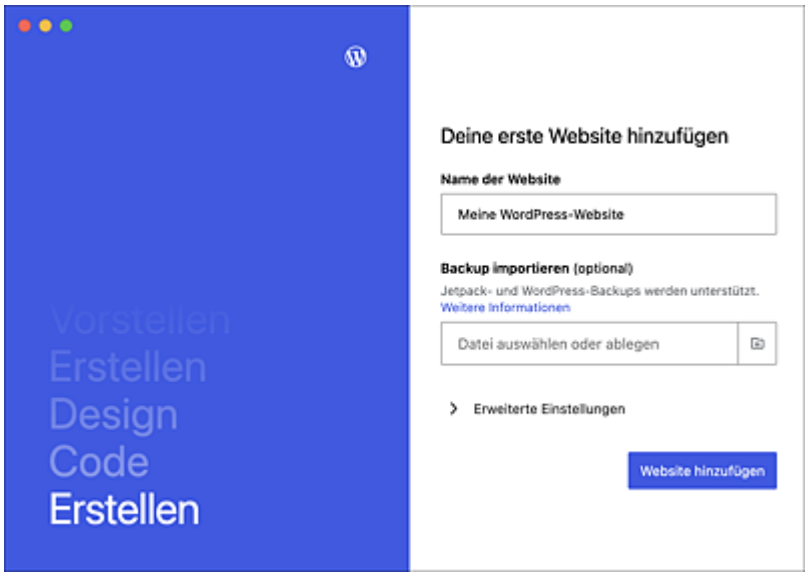


Abbildung 2.10 Mit »Studio« haben Sie ein hübsches Interface für Playground-Instanzen.

Ist die Website fertig eingerichtet, öffnet sich eine Ansicht, in der links die Websites (oder momentan die einzelne Website) aufgelistet

und rechts Informationen zur aktuell gewählten Website angezeigt werden. Die rechte Ansicht ist in den Tabs Überblick, Teilen, Import/Export und Einstellungen organisiert.

- Überblick zeigt Ihnen, welches Theme aktiv ist, und bietet Links zu unterschiedlichen Bereichen der WordPress-Installation, etwa dem Site-Editor, den Vorlagen oder den Seiten.
- Teilen erlaubt Ihnen, den aktuellen Stand der Website öffentlich zu teilen, wofür Sie sich mit einem WordPress.com-Account anmelden müssen.
- Import/Export erlaubt Ihnen, ein mit dem Plugin Jetpack gemachtes Backup oder eine SQL-Datei zu importieren, sowie die gesamte Website oder nur die Datenbank zu exportieren.
- Einstellungen listet Informationen wie den Pfad zu den WordPress-Dateien, die WordPress-Version, die lokale Domain und das Admin-User/-Passwort auf und erlaubt das Löschen der Website.

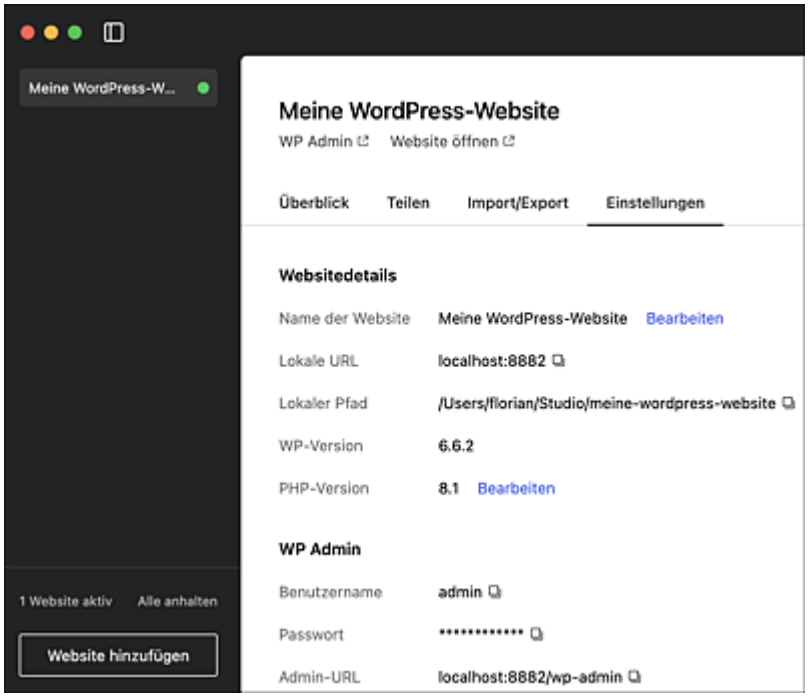


Abbildung 2.11 In den Einstellungen zu einer Site sehen Sie Informationen wie den Pfad zu den WordPress-Dateien.

3 WordPress-Installation – flexibel

> *Es kann vorkommen, dass Ihnen die in Kapitel 2 vorgestellte Lösung einer lokalen Installation über Playground nicht genug Möglichkeiten bietet, um das System anzupassen, auf dem WordPress läuft. Abhilfe schafft hier die separate Installation des Webservers und aller weiteren WordPress-Voraussetzungen.*

Hier lernen Sie die flexible Installation von WordPress am Beispiel von Lando kennen, einer Software, die auf Docker aufsetzt. Docker ist eine Lösung, die Anwendungen in sogenannten Containern ausführt, wodurch sie vom System, auf dem der Container läuft, getrennt sind. Was ein Container zur Ausführung enthalten muss, wird in sogenannten Container-Images definiert. Es gibt beispielsweise das *wordpress*-Docker-Image, das bei der Ausführung alles installiert, was für WordPress notwendig ist, angefangen beim Webserver über Datenbank und PHP bis hin zu WordPress selbst. Über eine sogenannte Compose-Datei kann die Konfiguration eines Projekts, die auch aus mehreren Docker-Images bestehen kann, mit anderen geteilt werden, sodass sie sich dieselbe Umgebung erstellen können.

Wenn Sie bisher nichts mit Docker zu tun hatten, klingt das jetzt wahrscheinlich recht kompliziert, und genau diesen Eindruck bekommt man von Docker auch, wenn man sich in die Dokumentation einliest.

Es gibt zum Glück ein paar Lösungen, die zwar auf Docker aufsetzen, aber einiges von der Komplexität wegnehmen und speziell für die Webentwicklung gemacht sind. Eine davon ist Lando.

3.1 Installation von Docker

Bevor Sie aber Lando installieren, benötigen Sie Docker auf Ihrem System. Wenn Sie macOS verwenden, können Sie alternativ zum offiziellen Docker Desktop mal einen Blick auf OrbStack werfen (*orbstack.dev/*), das bei größeren Projekten gegebenenfalls auftretende Performance-Probleme verhindert oder zumindest mindert.

Die Installationsschritte für Docker Desktop für Ihr System finden Sie auf *docs. docker.com/engine/install/*.

3.2 Installation von Lando

Die Installation für die unterschiedlichen Betriebssysteme ist unter *docs.lando.dev/basics/installation.html* beschrieben. Folgen Sie den Schritten für Ihr System, und wenn Sie Windows nutzen, aktivieren Sie das Windows Subsystem for Linux (WSL) und arbeiten damit, um Probleme zu vermeiden.

Info: Windows Subsystem for Linux (WSL)

Mit Windows Subsystem for Linux kann parallel unter Windows eine Linux-Umgebung laufen, ohne dass eine klassische virtuelle Maschine installiert werden muss. Zugriff auf dieses Untersystem wird über ein Terminal ermöglicht, es gibt also keine visuelle Linux-Benutzeroberfläche. Da viele entwicklungsspezifische Tools in erster Linie für Linux und/oder macOS erstellt werden, funktionieren sie nicht so leicht oder gar nicht unter Windows. Das WSL bietet dafür eine Lösung und vereinfacht das Arbeiten mit einer Vielzahl von Werkzeugen und Technologien der Webentwicklung unter Windows.

3.3 Lando-Installation testen

Wenn Sie die Installation erfolgreich durchlaufen haben, können Sie mit den folgenden Befehlen eine WordPress-Installation mit Lando erstellen:

```
lando init \
    --source remote \
    --remote-url https://wordpress.org/latest.tar.gz \
    --recipe wordpress \
    --webroot wordpress \
    --name my-first-wordpress-app

lando start
```

Listing 3.1 Mit den beiden Befehlen können Sie eine WordPress-Installation mit Lando erstellen und starten.

Der Befehl `lando init` erstellt auf Basis der übergebenen Parameter eine `.lando.yml`-Datei, die die Konfiguration für die Anwendung enthält. In dem Beispiel für WordPress wird als Quelle des Codes der Link auf die neueste Version von WordPress übergeben und als `recipe` das von Lando unterstützte `wordpress`-Rezept. Das sorgt unter anderem dafür, dass ein Webserver, PHP und MySQL installiert werden und direkt eine Datenbank angelegt wird. Mit `lando start` wird Lando gestartet.

Problemlösung: »lando start« hängt beim Composer-Download

Eventuell stoßen Sie auf das Problem, dass unter Windows mit WSL 2 der Befehl `lando start` nicht durchläuft, sondern beim Herunterladen von Composer hängen bleibt. In diesem Fall kann es helfen, Docker über den Befehl `sudo service docker restart` neu zu starten.

Hat alles geklappt, sehen Sie etwas wie in Abbildung 3.1. Wenn die Domain nicht erreichbar ist, beachten Sie den nachfolgenden Kasten zur Problemlösung im Zusammenhang mit FRITZ!Boxen.

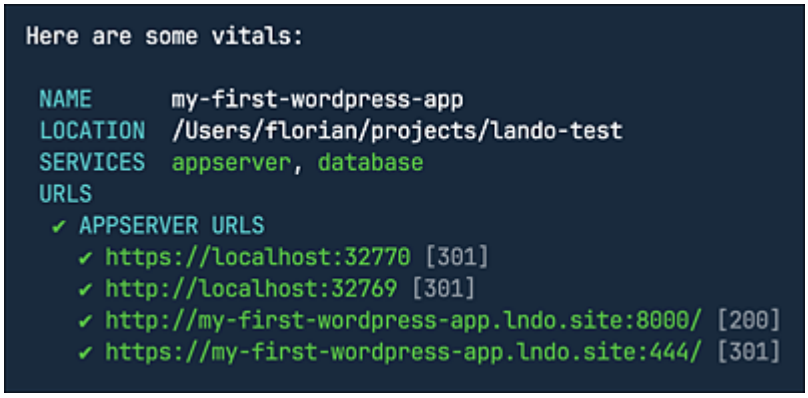


Abbildung 3.1 Ein paar Daten zur Erreichbarkeit der Lando-WordPress-Installation am Ende des »lando start«-Befehls

Wenn Sie die Domain oder *localhost* mit Port aufrufen, sollten Sie den bekannten Start der WordPress-Installation sehen. Der Wert für Datenbankname, Benutzername und Passwort ist `wordpress`, für den Datenbankhost geben Sie `database` an.

Und damit haben Sie eine fertige WordPress-Installation mit Lando eingerichtet. Unter *docs.lando.dev* finden Sie die Dokumentation zu Lando, die Sie – wenn Sie das Tool nutzen – ruhig durchgehen sollten, um alle Möglichkeiten kennenzulernen, die Lando bietet.

Problemlösung: Problem mit *.lndo.site-Domain mit einer FRITZ!Box

Sollte beim Einrichten die Domain *hello-lando.lndo.site* nicht erreichbar sein bzw. sollten Sie nur über *localhost* mit den entsprechenden Ports auf das Lando-Beispiel zugreifen können, hängt das Problem mit dem DNS-Rebind-Schutz Ihrer FRITZ!Box zusammen. Dieser sorgt dafür, dass DNS-Antworten unterdrückt

werden, die auf lokale IPs verweisen. Und genau das tut Lando: Es verweist die **.lndo.site*-Domains auf die IP `127.0.0.1`, sodass die *hosts*-Datei nicht wie bei XAMPP angepasst werden muss. Um das Problem zu lösen, ist eine Änderung in den FRITZ!Box-Einstellungen (erreichbar über die Domain *fritz.box*) unter **Heimnetz** • **Netzwerk** • **Netzwerkeinstellungen** • **DNS-Rebind-Schutz** notwendig (im Bereich **weitere Einstellungen** im **Netzwerkeinstellungen**-Tab). In das Feld mit der Überschrift **Hostnamen-Ausnahmen** schreiben Sie die Domain *hello-lando.lndo.site*, wie in Abbildung 3.2 gezeigt, und damit sollte die lokale URL erreichbar sein.

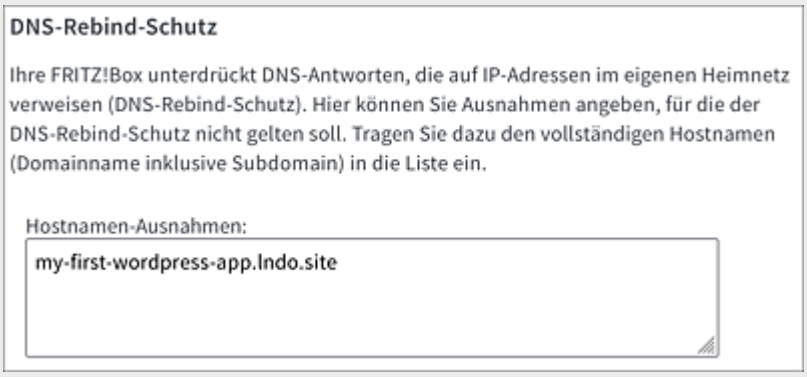


Abbildung 3.2 Die Angabe der Lando-Domain als Hostnamen-Ausnahme im »DNS-Rebind-Schutz« einer FRITZ!Box kann das Problem der Erreichbarkeit der lokalen Domain beheben.

Falls das Problem damit weiter besteht, nutzen Sie einfach die *localhost*-Domain. Das ist zwar nicht so schön wie eine »richtige« Domain, aber potenziell lange nach einer Lösung für dieses Problem zu suchen, nur um dann nicht mehr *localhost*, sondern eine Domain zu haben, ist es nicht so richtig wert.

4 Hoster-Auswahl und Installation

Wenn Sie eine öffentlich zugängliche Website betreiben, werden Sie dafür in der Regel auf die Dienste eines sogenannten Webhosters zurückgreifen. Diese Dienstleister stellen Ihnen Speicherplatz, eine Datenbank, Bandbreite und Rechenleistung auf spezialisierten Computern zur Verfügung. Auf diesen auch Server genannten Computern werden in den meisten Fällen viele Hundert oder Tausend Websites betrieben.

Ein einzelnes Hosting-Paket kann im Preis stark variieren, zumindest die Einstiegspakete kosten aber nicht die Welt. Neben einfachem Webhosting, in dem in der Regel nicht nur Grundlegendes wie Webspace, Datenbank und oft auch eine Domain enthalten ist, bieten immer mehr Webhosting-Anbieter auch sogenanntes *Managed WordPress Hosting* an. Dabei übernehmen die Anbieter zum Beispiel auch Aktualisierungen und das Beheben von Sicherheitsproblemen für Sie, lassen sich dafür aber auch entsprechend bezahlen.

In diesem Kapitel führen wir Sie durch den Webhosting-Dschungel und helfen Ihnen bei der Auswahl eines zuverlässigen Webhosters, der Ihren individuellen Ansprüchen genügt. Neben den absoluten Basics wie ausreichendem Speicherplatz, Inklusiv-Domains und Postfächern gibt es bei der Auswahl einige Punkte zu beachten. Der nachträgliche Wechsel von einem Hosting-Anbieter zum anderen

ist zwar grundsätzlich möglich, bringt aber einigen Mehraufwand mit sich, den Sie mit einer guten Auswahl des Anbieters vorerst vermeiden können.

4.1 Welches Hosting darf's denn sein?

Bevor wir an die Auswahl eines *Hosting*-Anbieters gehen, sollten Sie sich über Ihre konkreten Anforderungen klar werden. Anders als vor 10 Jahren haben Sie heute eine deutlich breitere Auswahl an verfügbaren Hosting-Produkten.

Nehmen wir uns also einen Moment Zeit, und überlegen wir uns, wie Sie und Ihr Projekt einzuordnen sind.

4.1.1 Welches Hosting passt zu Ihnen?

Die Frage, welches Hosting zu Ihnen passt, mag im ersten Moment etwas merkwürdig erscheinen. Am Ende ist aber vor allem ein Punkt interessant: Sehen Sie sich eher als Einsteiger*in oder als Profi, was den Betrieb eigener Websites angeht? Oder anders ausgedrückt: Wie tief möchten Sie in die Abgründe des Hostings hinabsteigen?

Das eine Extrem wäre, dass Sie sich gar nicht mit dem Betrieb Ihrer Website befassen wollen. Daran ist nichts Verwerfliches. In den meisten Fällen betreiben Menschen Websites nicht zum Selbstzweck, sondern verfolgen damit einen bestimmten Zweck – seien es Berichte von der Fahrradreise durch Neuseeland, das Präsentieren eigener Strickmuster oder der Verkauf von Produkten in einem Onlineshop.

Am anderen Ende des Spektrums befinden Sie sich, wenn Sie alle Aspekte Ihres Hostings selbst kontrollieren wollen. Aber bevor Sie

sich mit der Idee anfreunden, einen kleinen Serverschrank in den Keller zu stellen und Ihre Website zwischen Sicherungskasten und Heizungstherme zu betreiben, bedenken Sie die Verantwortung, die Sie damit übernehmen. Sie machen damit die Erreichbarkeit Ihrer Website zu Ihrem Job. Nicht unbedingt in Vollzeit, doch rund um die Uhr in Bereitschaft. Nun ist der Heimbetrieb eines Servers wirklich das absolute Extrem und allein schon wegen der unzureichenden Anbindung der meisten Privathaushalte ans Internet nicht empfehlenswert. Aber auch wenn Sie sich mit der Hardware nicht beschäftigen wollen, gibt es diverse Hosting-Produkte für Sie, bei denen Sie weitgehende Freiheiten haben.

Egal, ob Sie sich an einem dieser beiden Extreme oder irgendwo auf dem Spektrum dazwischen wiederfinden, es gibt am Markt mehr als genug Angebote, die Ihren Anforderungen entsprechen werden.

4.1.2 Was für ein Website-Projekt haben Sie vor sich?

Neben den Ansprüchen, die Sie als Kund*in an Ihr Hosting-Paket stellen, sollten wir auch betrachten, welche Anforderungen das Projekt, das Sie vor sich haben, stellt. Diese Anforderungen können sich dramatisch unterscheiden und hängen in vielen Fällen einfach an der Frage, ob die Website, die Sie betreiben wollen, in irgendeiner Art *missionskritisch* ist oder nicht.

Sind Menschen direkt oder indirekt auf die Funktionsfähigkeit der Website angewiesen – sei es, weil sie dort Informationen zum örtlichen Weihnachtsmarkt finden, weil sie dort Termine für ihre Hausärztin buchen können oder weil ihr Einkommen am Funktionieren eines Webshops hängt?

Wenn Ihr Projekt in diese Kategorie fällt, sollten Sie ein besonderes Augenmerk auf *Verfügbarkeit*, *gute Backups* und einen

kompetenten und vor allem *gut erreichbaren Support* legen. Vor allem an Letzterem scheidet sich auch heute noch die Spreu vom Weizen.

Wenn Sie noch dazu mit besonders hohen zeitgleichen Zugriffszahlen rechnen, lohnt sich unter Umständen die Investition in ein Hosting-Angebot mit besserer Last-Verteilung (*Load Balancing*) oder einem automatisch skalierenden Hosting, das sich dynamisch den Anforderungen anpasst (und außerdem auch dynamisch bezahlt wird).

Wenn Sie allein von der Vorstellung schon leicht erhöhten Blutdruck und schwitzige Hände bekommen, können wir Sie an dieser Stelle beruhigen. Sollte Ihr Projekt nicht in die Kategorie der *missionskritischen* Websites fallen, werden Sie sich mit den meisten dieser Themen nicht mit derselben Intensität beschäftigen müssen.

Lesen Sie das bitte nicht als Empfehlung, einfach ein beliebiges Angebot eines beliebigen Hosting-Anbieters anzunehmen, aber zumindest muss Ihr Fokus bei der Auswahl nicht so sehr auf diesen Themen liegen.

4.1.3 Wo befindet sich das Publikum Ihrer Website?

Nachdem wir uns nun schon über Sie und Ihr Projekt Gedanken gemacht haben, sollten wir auch noch kurz über Ihre voraussichtliche Zielgruppe nachdenken.

Wir werden hier nicht allzu tief in Details einsteigen, über die sich sonst eher die Kolleg*innen aus dem Marketing Gedanken machen. Stattdessen stellt sich uns eine einfache Frage: Wo befinden sich die Menschen, die Ihre Website aufrufen? Diese banale Frage, hat

Einfluss auf die Auswahl Ihres Hosting-Anbieters. Genauer gesagt: auf den Standort Ihres Hostings.

Machen wir dafür einen kurzen Exkurs in die Hosting-Theorie: Auf die Geschwindigkeit eines Hosting-Pakets haben die unterschiedlichsten Parameter Einfluss. Viele davon können Sie mit der Auswahl des richtigen Pakets verändern. Sie wählen mehr Rechenleistung, mehr Arbeitsspeicher, schnelleren Speicher oder eine bessere Netzwerk-Anbindung, und Ihre Website kann schneller übertragen werden.

Allen technischen Lösungen zum Trotz gibt es eine Einschränkung: Ihre Website ist auf einem Server gespeichert. Dieser Server befindet sich in einem Rechenzentrum. Die Entfernung von besagtem Rechenzentrum zu den Besucher*innen Ihrer Website müssen die Daten, die beim Besuch abgerufen werden, erst einmal überwinden. Und egal, ob diese Daten über alte Kupferkabel oder moderne Glasfaser übertragen werden, die Physik diktiert, dass sie nicht schneller als mit Lichtgeschwindigkeit übertragen werden können.

So lange sich sowohl Ihre Besucher*innen als auch Ihr Server irgendwo in Europa befinden, ist das kein nennenswertes Problem. Wollen Sie aber zum Beispiel eine Unternehmens-Website betreiben, die vor allem Kund*innen in Australien ansprechen soll, bietet es sich an, einen *etwas* regionaleren Anbieter zu wählen. Gleiches gilt mit einem rein amerikanischen Hosting für europäische Besucher*innen.

In diesem Szenario kommt außerdem noch die Datenschutz-Grundverordnung (DSGVO) hinzu. Diese macht die Auswahl nicht europäischer Hosting-Anbieter seit 2018 zwar nicht zwingend unmöglich, verkompliziert sie aber so weit, dass wir auf jeden Fall auf der sichereren Seite sind, wenn wir zuerst nach einem

europäischen Anbieter schauen, bevor wir den Blick über den Großen Teich werfen.

4.1.4 Minimale Voraussetzungen für WordPress

Neben diesen eher subjektiven Kriterien gibt es einige essenzielle Punkte, die ein Hosting-Paket erfüllen muss, damit Sie eine WordPress-Website darauf betreiben können.

- **PHP** ist die Programmiersprache, auf der ein Großteil von WordPress basiert. Ist PHP nicht vorhanden, läuft WordPress nicht. Bei den meisten modernen Hosting-Paketen und vor allem bei expliziten WordPress-Produkten sollte das kein Problem darstellen. Während WordPress 6.7 mindestens PHP in Version 7.2.24 voraussetzt, sollten Sie darauf achten, dass der Hosting-Anbieter Ihrer Wahl aktuelle PHP-Versionen unterstützt und dieses Angebot auch regelmäßig pflegt. Mindestens PHP 8.2 sollten Sie für eine neue WordPress-Website verwenden.

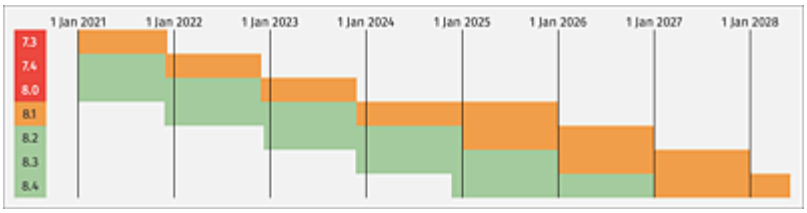


Abbildung 4.1 PHP erhält wie andere Programmiersprachen Software-Updates. Zwei Jahre nach Veröffentlichung wechselt eine PHP-Version in eine Phase, in der sie nur noch Sicherheitsupdates bekommt, nach weiteren zwei Jahren enden auch diese. Ein im November 2024 veröffentlichtes PHP 8.4 erreicht damit also Ende 2028 sein Ende und sollte spätestens dann nicht mehr eingesetzt werden.

- **MySQL/MariaDB** sind SQL-basierte Datenbanken und werden von WordPress zum Speichern dynamischer Daten genutzt. Ähnlich wie bei PHP wird ohne eine MySQL 8- oder MariaDB 10.4-Datenbank der Betrieb einer WordPress-Website nahezu unmöglich.

- **SSL/TLS**-Zertifikate werden von WordPress seit 2016 als Standard empfohlen. Mithilfe eines solchen Zertifikats können Daten verschlüsselt zwischen Ihrem Webserver und dem Browser Ihrer Besucher*innen übertragen werden. In der Vergangenheit waren solche Zertifikate kostenpflichtig und wurden daher eher für besonders sicherheitskritische Vorgänge oder Websites genutzt. In den 2020er-Jahren sollten Sie aber kein Hosting-Paket ohne kostenfreie SSL-Zertifikate mehr nutzen.
- **SFTP- und SSH-Zugang** mit **WP-CLI** stellen sicher, dass Sie abgesichert auf das Dateisystem Ihres Webhosting-Pakets zugreifen können. Das ältere unverschlüsselte FTP sollte nicht mehr zum Einsatz kommen. Es ist gut möglich, dass Sie SSH und WP-CLI nie selbst nutzen werden, zu einem zeitgemäßen WordPress-Hosting gehört es aber dazu und könnte in der Zukunft nützlich werden, wenn Sie zum Beispiel eine Dienstleister*in mit Arbeiten an Ihrer Website beauftragen.

4.1.5 Produktkategorien und Auswahl

Zu Beginn dieses Kapitels sind wir schon kurz auf die verschiedenen Produktkategorien im Hosting eingegangen. Setzen wir uns kurz noch mal etwas genauer mit ihnen auseinander.

Format	Beschreibung
»Einfaches« Shared Webhosting	keine WordPress-Spezialisierung, geteilte Ressourcen auf großen Servern
WordPress-Hosting	oft einfach Webhosting-Pakete mit vorinstalliertem oder einfach nachinstallierbarem WordPress

Format	Beschreibung
»Managed« WordPress Hosting	Pakete mit in der Regel vorinstalliertem WordPress und automatisiertem Management wie Updates, Backups etc.
Root-Server	privater Server, der nicht vom Hosting-Anbieter verwaltet wird
Managed Server	privater Server, der vom Hoster verwaltet wird und oft mit leichten Einschränkungen im Vergleich zum Root-Server kommt
VPS	ein *virtueller* privater Server, oft eine kostengünstige Alternative zu Root- oder Managed Servern
»Cloud«-Hosting	skalierende Hosting-Produkte auf *Hyperscaler*-Infrastruktur

In welcher dieser Kategorien finden Sie sich selbst und Ihr Website-Projekt am ehesten wieder? Sollten Sie nach diesen Abwägungen immer noch unsicher sein oder noch nicht genau wissen, wohin Sie ihr Projekt führen wird, sind Sie vermutlich mit einem günstigeren (Managed) WordPress-Tarif gut bedient. Achten Sie für den Anfang auf Angebote mit monatlicher Kündigungsfrist. Preislich dürften die meisten dieser Pakte zwischen 5 und 10 € pro Monat liegen. Außerhalb von Sonderangeboten sollten Sie bei Preisen deutlich unter der 5-€-Schwelle skeptisch sein. Das könnte ein Indiz für überbuchte Server, unterbesetzten Support oder überalterte Systeme sein.

Ob Sie Domains auch bei Ihrem Hosting-Anbieter registrieren wollen, ist in vielen Fällen eine Frage der persönlichen Vorliebe und der Menge an Domains und Hosting, mit denen Sie umgehen. Wenn Sie nur eine oder zwei Domains für eine einzige Website verwalten,

fahren Sie vermutlich auch gut damit, diese bei dem Hosting-Anbieter Ihrer Wahl abzulegen (und vielleicht sogar Inklusiv-Domains Ihres Hosting-Pakets zu nutzen).

4.2 Installation mit einem One-Click-Installer

Haben Sie sich für ein Hosting-Paket mit vorinstalliertem WordPress entschieden, so entfällt dieser Schritt vollständig.

Doch auch wenn Ihr Paket diesen kleinen Luxus nicht enthält, sind Ihnen die meisten Hosting-Anbieter bei der Installation von WordPress behilflich. In den meisten Fällen finden Sie in der Verwaltungsoberfläche Ihres Anbieters einen »Installer« oder »1-Klick-Installer«, mit dem Sie oft neben WordPress noch eine Vielzahl weiterer Open-Source-Applikationen installieren können. Bevor es losgeht, müssen Sie in einigen Fällen zunächst noch Benutzername, Passwort und E-Mail-Adresse festlegen und eventuell die zu verwendende Domain auswählen. Im Anschluss kümmert sich der Installer in der Regel selbst um das Anlegen einer Datenbank und die korrekte Ablage der WordPress-Core-Dateien auf Ihrem Webspace.

Nach erfolgter Installation können Sie sich zum ersten Mal einloggen. Prüfen Sie unbedingt, ob die aktuellste Version von WordPress installiert wurde. Gelegentlich sind 1-Klick-Installer mit veralteten Versionen ausgestattet, und nach dem ersten Login ist ein Update erforderlich. Wenn nicht, finden Sie in Abschnitt 16.1 eine kleine Anleitung zum Aktualisieren von WordPress.

Außerdem sollten Sie überprüfen, ob für die neue Installation ein SSL-Zertifikat eingerichtet wurde. Rufen Sie Ihre Domain nach der Installation einfach mit einem vorangestellten *https://* in Ihrem Browser auf. Wenn keine Fehlermeldung erscheint oder sogar ein kleines Schloss in der Adresszeile Ihres Browsers angezeigt wird

(abhängig vom verwendeten Browser), hat alles geklappt. Sie können jetzt loslegen.

5 Administration und Konfiguration von WordPress

Nach der Installation von WordPress können Sie mit Ihrer Website loslegen. Bevor Sie aber mit Inhalten starten, lernen Sie das Backend von WordPress kennen, in dem neben der Inhaltsverwaltung noch deutlich mehr passiert.

In diesem Kapitel lernen Sie das *Backend* einer WordPress-Installation kennen. Dieser Teil ist nur für angemeldete Benutzer*innen sichtbar. Der zweite Teil der Website ist das *Frontend*, das (abgesehen von gegebenenfalls passwortgeschützten Bereichen) für alle sichtbar ist.

5.1 Das Backend kennenlernen

5.1.1 Einloggen ins Backend

Ins Backend einer WordPress-Website gelangen Sie, indem Sie an Ihre Domain den Pfad */wp-admin/* anhängen, also beispielsweise *example.com/wp-admin/*. Wenn Sie nicht eingeloggt sind, erscheint jetzt der Login-Screen, auf dem Sie sich mit den Daten anmelden müssen, die Sie während der Installation vergeben haben.

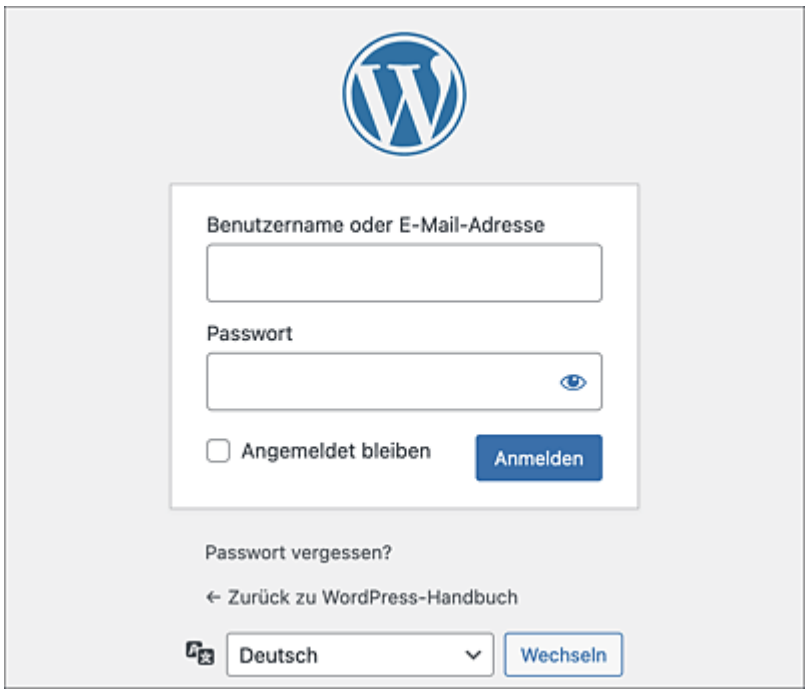


Abbildung 5.1 Loggen Sie sich mit Ihrer während der Installation vergebenen Benutzername-Passwort-Kombination im Backend ein.

Info: Login-Daten im Browser merken

Wenn Sie sich im Backend anmelden, wird höchstwahrscheinlich Ihr Browser fragen, ob die Login-Daten gespeichert werden sollen. Eine bessere Variante ist die Nutzung eines Passwort-Managers, auf die in Kapitel 18 näher eingegangen wird.

5.1.2 Die Admin-Oberfläche und das Dashboard

Nach dem erfolgreichen Login landen Sie im Dashboard Ihrer WordPress-Installation. Am oberen Rand des Viewports sehen Sie die *Adminbar*, die auf allen Seiten des Backends angezeigt wird, außer auf denen, wo der Block-Editor übernimmt, also aktuell dem Website- und dem Inhalts-Editor. Standardmäßig wird sie auch im

Frontend angezeigt, wenn Sie eingeloggt sind, das können Sie in Ihrem Profil aber deaktivieren.

Schauen wir uns einmal die Funktionen in der Adminbar von links nach rechts an.



Abbildung 5.2 Die Adminbar am oberen Rand des Browserfensters beinhaltet hilfreiche Links und sieht je nach installierten Plugins unterschiedlich aus.

❶ WordPress-Menü
Beim Hover über das WordPress-Logo erscheint eine Liste mit Links zu WordPress-Seiten wie der Dokumentation und den Foren.

❷ Home (Name Ihrer Website)
Verlinkt aus dem Backend ins Frontend und, wenn Sie sich im Frontend befinden, ins Backend. Aus dem Frontend haben Sie außerdem über ein Dropdown noch direkt Verlinkungen zu den Plugins und Themes.

❸ Aktualisierungen
Zeigt die Anzahl der Updates, die für Ihre Website verfügbar sind. Darunter fallen Updates für WordPress, Plugins, Themes und Übersetzungen. Durch einen Klick gelangen Sie auf die AKTUALISIERUNGEN-Seite, von der Sie die Updates vornehmen können. *Tipp:* Wenn Ihre Website komplexer ist oder es sehr wichtig ist, dass sie nicht zwischendurch offline ist, sollten Sie Updates auf einer Kopie testen (siehe Kapitel 16, »WordPress-Wartung und -Pflege«).

❹ Kommentare
Zeigt die Anzahl der ausstehenden Kommentare an. Ein Klick bringt Sie auf die Liste der Kommentare.

❺ Neu
Bringt Sie zum Anlegen eines neuen Beitrags in den Editor. Wenn Sie den Cursor über den Punkt bewegen, öffnet sich ein Dropdown mit Links zum Anlegen einer Seite, Hochladen einer Datei und Hinzufügen einer Benutzer*in. Wenn durch Plugins weitere Inhaltstypen hinzugefügt werden, werden diese ebenfalls angezeigt.

❻ Buttons und Links von Plugins
Zwischen Neu und Willkommen können Plugins eigene Einträge in der Admin-Bar registrieren.

❼ Willkommen
Link zur Bearbeitungsansicht Ihres Profils und ein Link zum Abmelden

Welche Elemente unter der Admin-Bar im Inhaltsbereich des Dashboards angezeigt werden, können Sie über den Button Ansicht anpassen rechts oben komplett frei entscheiden. Standardmäßig sind bei einer neuen Installation die folgenden Elemente sichtbar (siehe Abbildung 5.3):

❽ Willkommen bei WordPress
Als erstes Element sehen Sie im Dashboard einen Banner, der ein paar Sprunglinks zu gängigen Aktionen enthält, die am Anfang eines Website-Projekts gefragt sind, etwa zum Website-Editor, um die Website an die eigenen Bedürfnisse anzupassen.

❾ **Zustand der Website**
WordPress bietet mit seinem »Website-Zustand«-Werkzeug ein nützliches Hilfsmittel an, das Ihre Installation auf unterschiedliche Aspekte untersucht und bei Bedarf in diesem Dashboard-Widget einen Hinweis anzeigt, wenn etwas verbessert werden sollte (mehr zu diesem Werkzeug erfahren Sie später in diesem Kapitel).

❿ **Schneller Entwurf**
Ermöglicht Ihnen das Anlegen eines neuen Beitrags als Entwurf direkt aus dem Dashboard heraus. Das ist natürlich nicht für ganze Beiträge zu gebrauchen, aber für das schnelle Festhalten eines Gedankens kann es ausreichen.

⓫ **Auf einen Blick**
Zeigt Ihnen die Anzahl der veröffentlichten Beiträge, Seiten und Kommentare sowie das verwendete Theme an. Kann durch Plugins erweitert werden.

⓬ **WordPress-Veranstaltungen und Neuigkeiten**
Im oberen Teil des Widgets sehen Sie Termine von WordCamps (ein- oder mehrtägige Veranstaltungen rund ums Thema WordPress) und Meetups (regelmäßig stattfindende lokale Zusammentreffen von WordPress-Interessierten, meist abends, manchmal einfach als offene Runde ohne festes Thema, manchmal zu festen Themen mit einem Vortrag; die Termine der deutschsprachigen Meetups finden Sie unter *http://r-wrk.de/wp6_wpmeetups*). Darunter sehen Sie Neuigkeiten aus der WordPress-Welt. Die Hinweise zu einer neu erschienenen Version sind interessant, um zu erfahren, was sich alles ändert. Oft werden hier auch praktische Tipps verlinkt, manchmal Hinweise für Programmierer*innen – ein bunter Informationsmix also.

⓭ Aktivität

Eine Ansicht darüber, was aktuell auf Ihrer Website los ist. Sie finden hier eine Liste der neuesten veröffentlichten Inhalte und Kommentare.

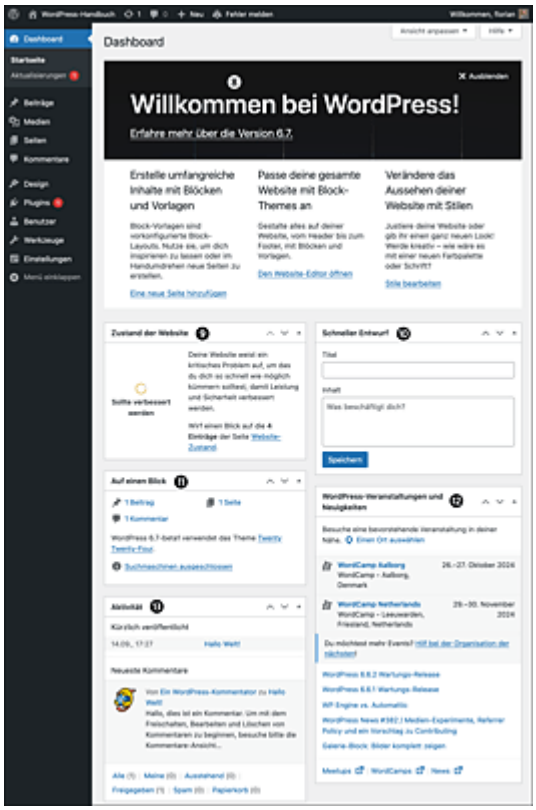


Abbildung 5.3 Das Dashboard ist die erste Seite, die Sie nach dem Einloggen ins Backend sehen.

Vermutlich werden Sie einige dieser Informationen nicht benötigen, deshalb können Sie, wie am Anfang angedeutet, rechts oben über den Button Ansicht anpassen auswählen, was zu sehen sein soll (Abbildung 5.4).

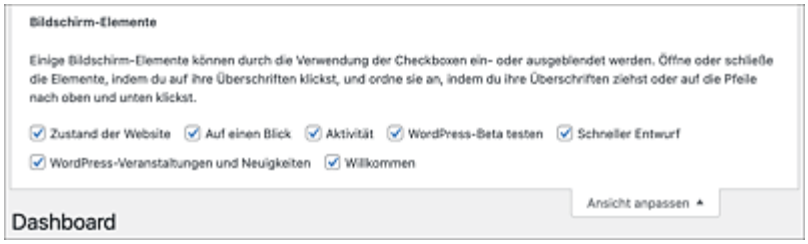


Abbildung 5.4 Die angezeigten Widgets im Dashboard lassen sich über den Button »Ansicht anpassen« ändern.

5.1.3 Das linke Admin-Menü

Nachdem Sie die Adminbar und den Inhaltsbereich des Dashboards kennengelernt haben, fehlt jetzt noch das linke Admin-Menü, das wie die Adminbar auf allen Seiten im Backend angezeigt wird, abgesehen von den Block-Editor-Seiten.

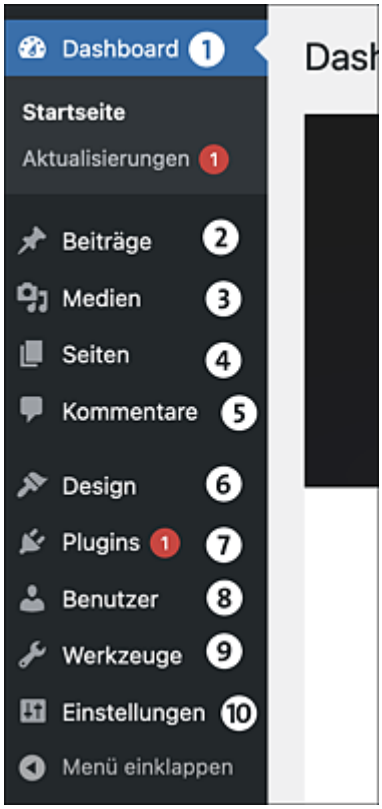


Abbildung 5.5 Über das Admin-Menü auf der linken Seite gelangen Sie zu allen Bereichen des Backends.

❶ Dashboard

- Startseite
  das Dashboard, das im vorherigen Abschnitt behandelt wurde
- Aktualisierungen
  Hier sehen Sie auf einer Seite alle ausstehenden Aktualisierungen für Ihre WordPress-Website, egal ob für den WordPress-Core, Plugins, Themes oder Übersetzungen. Sie können die Updates hier natürlich auch direkt durchführen, nicht nur anschauen (Abbildung 5.6).

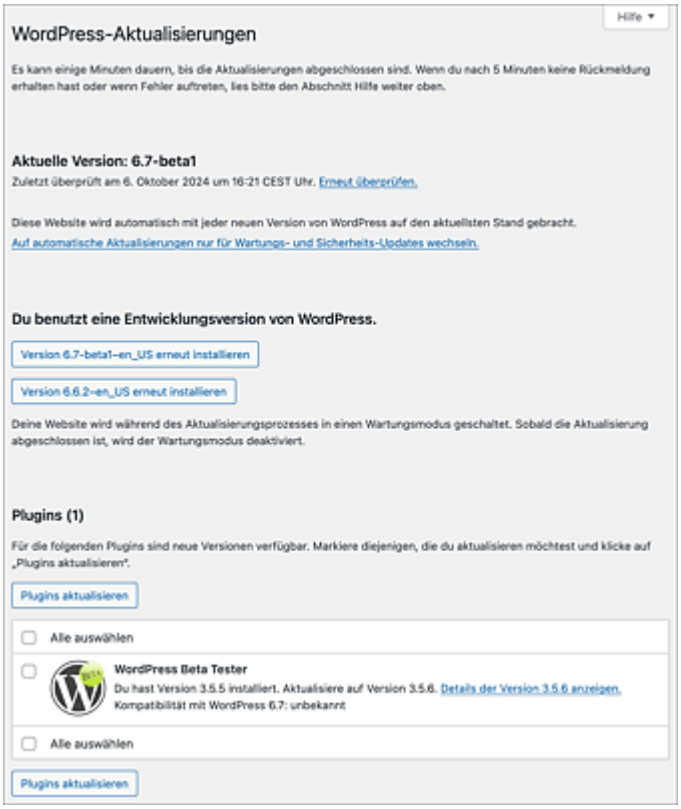


Abbildung 5.6 Über »Dashboard« • »Aktualisierungen« können Sie alle WordPress-Komponenten aktualisieren.

❷ Beiträge

- Alle Beiträge
  Zeigt die Liste aller Blogbeiträge der Website (Abbildung 5.7).

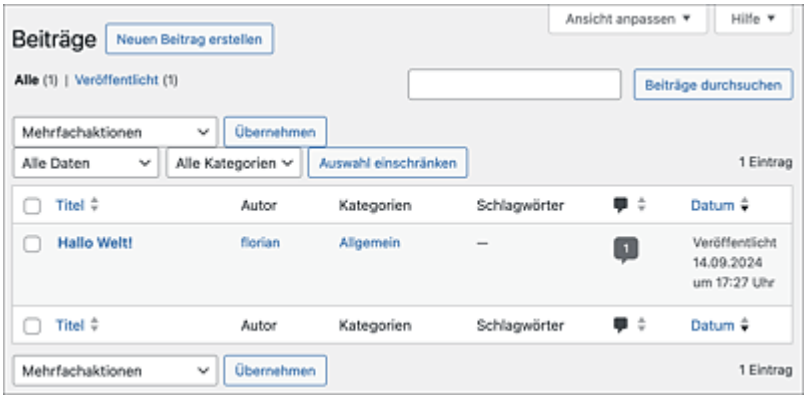


Abbildung 5.7 Die Liste mit allen Beiträgen Ihrer Website

- Neuen Beitrag erstellen
  bringt Sie in die Editor-Ansicht zum Anlegen eines neuen Beitrags (Abbildung 5.8)

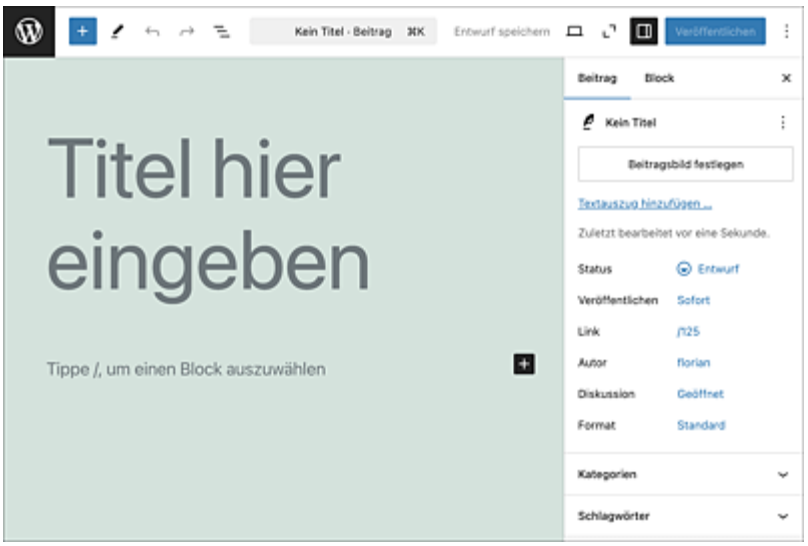


Abbildung 5.8 Der Block-Editor zum Anlegen eines neuen Beitrags.

- Kategorien
  Übersichtsseite der angelegten Beitragskategorien und Möglichkeit, neue Kategorien zu erstellen (Abbildung 5.9). Wie der Name sagt, können Sie mit Kategorien Ihre

Beiträge kategorisieren. Im Frontend ist es dann in der Regel so, dass neben der Liste aller Beiträge auch für jede Kategorie eine Seite existiert, die die Beiträge aus dieser Kategorie zeigt.

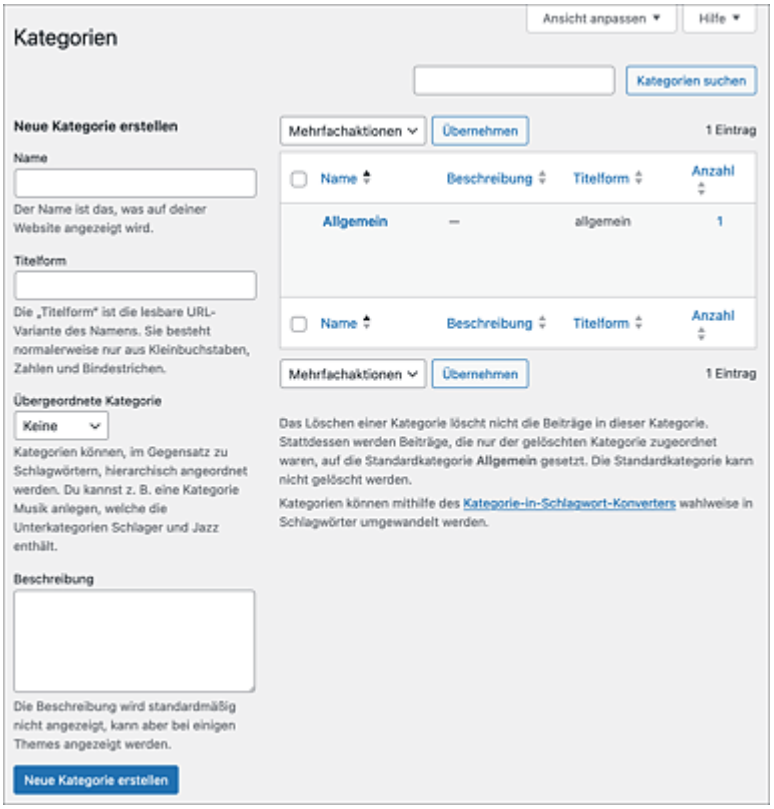


Abbildung 5.9 Über Kategorien können Sie Beiträge nach thematischer Gemeinsamkeit zusammenfassen.

- SCHLAGWÖRTER

  Übersichtsseite der Schlagwörter. Ein weiteres System, um Ordnung in Ihre Beiträge zu bringen, sind die Schlagwörter. Es kann ein bisschen verwirrend sein, dass es zwei Mechanismen gibt, die quasi dasselbe machen. Denken Sie bei Schlagwörtern aber vielleicht eher an Hashtags, also kleinteiligere Beschreibungen von Inhalten, die auch nur ein- oder zweimal genutzt werden, während Kategorien gröber sind und sich meist auf eine Handvoll beschränken.

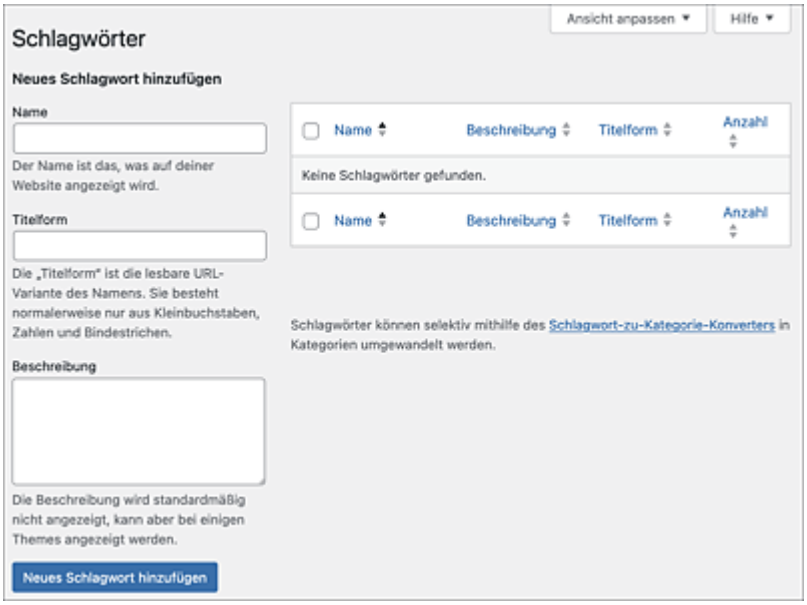


Abbildung 5.10 Mit Schlagwörtern können Sie Ihre Beiträge bei Bedarf feiner unterteilen.

❸ Medien

- **Mediathek**
  Zeigt alle Dateien, die zur Website hochgeladen wurden – sei es über den Button oben in der Mediathek oder direkt zu einem Beitrag, einer Seite oder Ähnlichem (siehe Abbildung 5.11). In der Regel werden das wahrscheinlich meist Bilder sein, aber dazu zählen auch Dokumente, Audio- und Video-Dateien usw. Die Ansicht der Mediathek kann über zwei Icons ganz links in der Werkzeugleiste über den Dateien zwischen einer Kachel- und einer Listenansicht hin- und hergeschaltet werden.

  Durch einen Klick auf eine der Dateien öffnet sich eine Detailansicht, in der Sie für Bilder beispielsweise den Alternativtext angeben können.

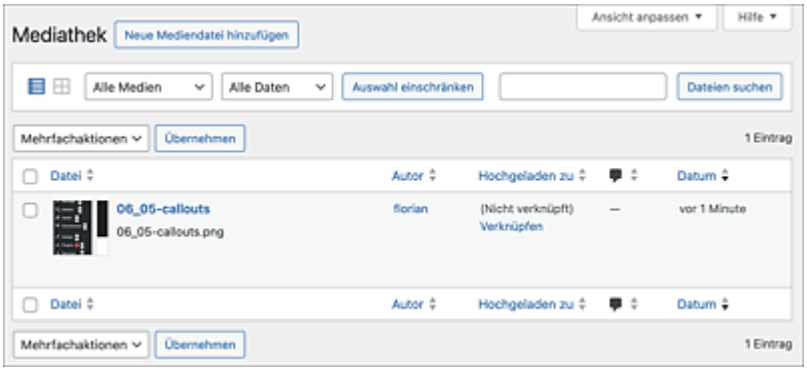


Abbildung 5.11 Die Mediathek zeigt Ihnen alle Medien an, die über WordPress zur Website hochgeladen wurden.

Neben der Upload-Möglichkeit über den Button oben können Sie Dateien auch per Drag-an-drop hochladen, indem Sie sie auf die Liste der Dateien ziehen (die Ansicht verändert sich, wenn WordPress registriert, dass Sie Dateien hochladen möchten).

- Neue Mediendatei hinzufügen
  Seite zum Hochladen neuer Dateien (siehe Abbildung 5.12). Sie können Dateien über den Button Dateien auswählen oder per Drag-and-drop in den gestrichelten Bereich hochladen.

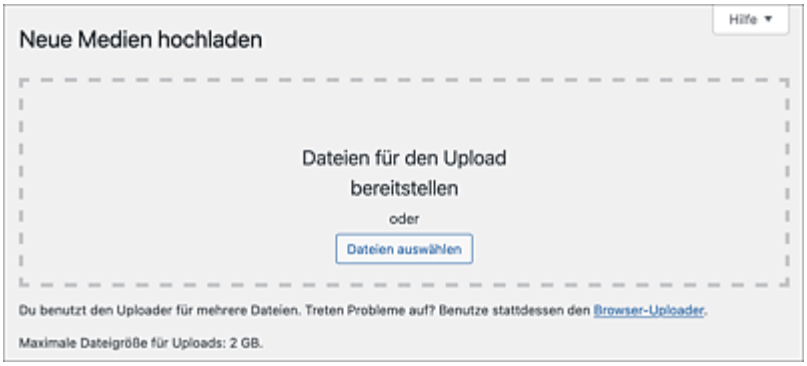


Abbildung 5.12 Neben den Möglichkeiten des Uploads per Drag-and-drop über die Mediathek an sich gibt es noch eine separate Seite zum Hochladen neuer Dateien.

❹ Seiten

- Alle Seiten
  Neben Beiträgen der zweite Inhaltstyp (genau genommen der dritte, weil Medien technisch gesehen auch ein Inhaltstyp sind und einige Elemente des Website-Editors intern ebenfalls als Inhaltstypen verwaltet werden), den Sie in einer Standard-Installation über WordPress verwalten. Wie bereits bei den Beiträgen gesehen, landen Sie hier auf einer Liste aller Seiten (Abbildung 5.13).

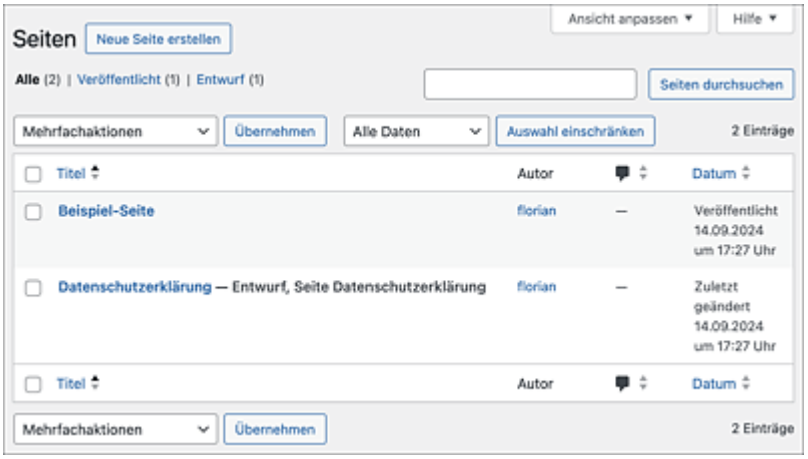


Abbildung 5.13 Auf einer frischen Installation finden Sie in der Liste aller Seiten eine Beispielseite und eine Vorlage für die Datenschutzerklärung, die Sie vor Live-Schaltung der Website unbedingt bearbeiten müssen.

- Erstellen
  öffnet wie bei den Beiträgen den Block-Editor, nur diesmal zum Anlegen einer Seite

❺ Kommentare

Die Liste aller Kommentare, die auf der Website gemacht wurden (Abbildung 5.14). Hier finden Sie auch die Kommentare, die noch freigeschaltet werden müssen, falls neue Kommentare nicht automatisch freigeschaltet werden.



Abbildung 5.14 Hier finden Sie alle Kommentare, die auf Ihrer Website zu Inhalten geschrieben wurden. Wenn Sie ein Plugin wie das Webmention-Plugin installiert haben, finden Sie hier auch Reaktionen von anderen Sites und Netzwerken.

❻ Design

Hinweis: Die Unterpunkte in diesem Menü unterscheiden sich je nach Art des Themes, das Sie nutzen. Wenn Sie eins der relativ neuen Block-Themes einsetzen, zu denen auch die neuen Standard-Themes von WordPress gehören, sehen Sie unter Design die Punkte Themes und Website-Editor.
Nutzen Sie ein Classic Theme wie Twenty Twenty-One, finden Sie die Einträge Themes, Vorlagen, Customizer, Widgets, Menüs, Hintergrund und Theme-Datei-Editor (je nach Theme kann es sein, dass mehr oder weniger Punkte angezeigt werden).

Es wird im nachfolgenden auf alle Punkte eingegangen.

- Themes
  Hier sehen Sie eine Liste der installierten Themes und können das Theme wechseln, sich eine Live-Vorschau eines

anderen Themes anschauen, neue Themes installieren und bestehende löschen (Abbildung 5.15).

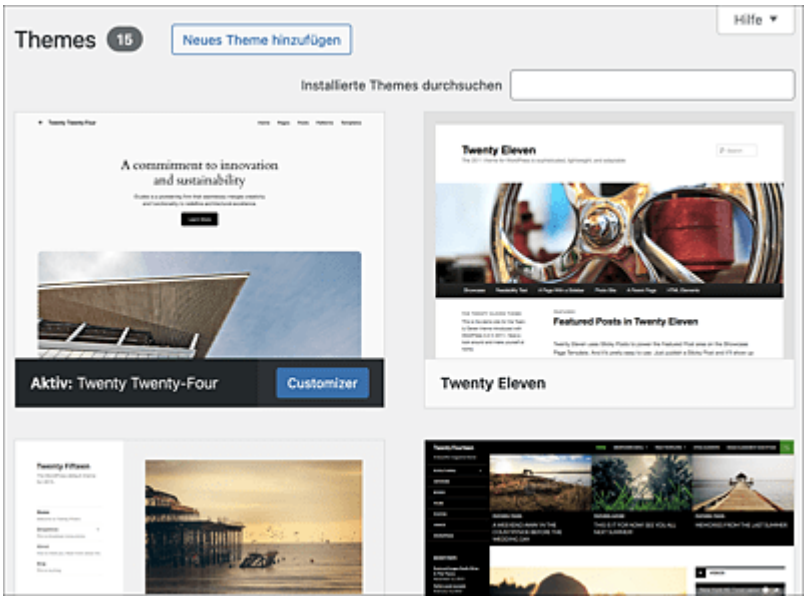


Abbildung 5.15 Mit Themes können Sie das Layout Ihrer Website wechseln.

- Vorlagen (Classic Themes)
  Öffnet den Vorlagenbereich im Website-Editor, in dem Sie Block-Vorlagen bearbeiten und erstellen können. Dieser Bereich wird in Abschnitt 10.3 besprochen, in dem es um die Anpassung eines Block-Themes geht.
- Customizer (Classic Themes)
  Der Link öffnet eine Ansicht, in der Sie auf der rechten Seite das Frontend Ihrer Website sehen und auf der linken Seite eine Seitenleiste mit unterschiedlichen Einstellungsmöglichkeiten (Abbildung 5.16). Sie können etwa den Website-Titel verändern, Menüs anlegen und Menü-Positionen zuweisen, Widget-Bereiche bearbeiten und, je nach Theme, Farbwerte anpassen. Das Tolle: Sie sehen die Änderungen auf der rechten Seite direkt als Vorschau, ohne dass sie bereits live sind. Wenn Sie mit den Änderungen zufrieden sind, können Sie sie aus dem Customizer heraus veröffentlichen.

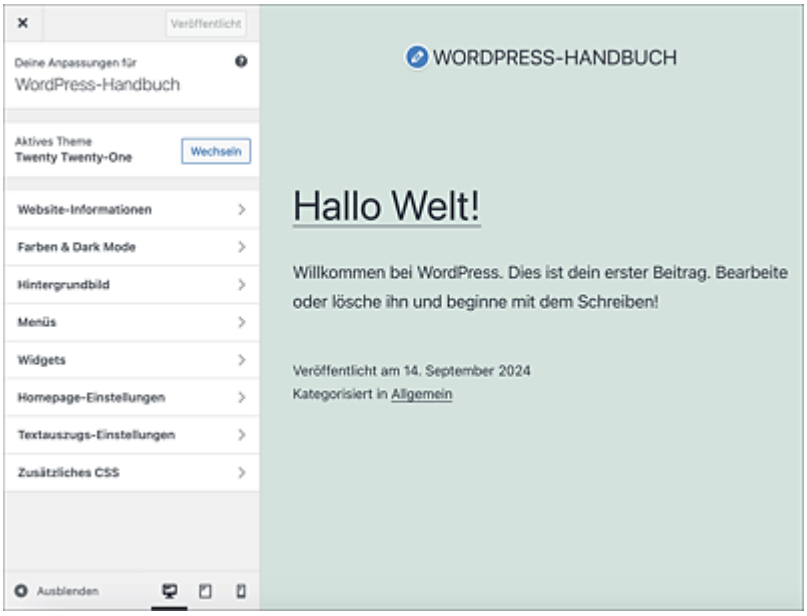


Abbildung 5.16 Im Customizer können Sie Classic Themes mit einer Live-Vorschau anpassen.

- Widgets (Classic Themes)
  Classic Themes können Widget-Bereiche definieren, oft beispielsweise im Footer oder – inzwischen nicht mehr ganz so oft – als klassische Sidebar. Diese Widget-Bereiche können Sie hier mit Inhalt füllen, und seit WordPress 5.8 besteht der Inhalt aus Blöcken, genau wie Ihre Beiträge und Seiten (Abbildung 5.17).
- Menüs (Classic Themes)
  Je nach Theme können Sie unterschiedliche Menüs anzeigen, typischerweise im Header-Bereich einer Website und optional an weiteren Positionen, etwa im Footer.

  Über den Bereich Menüs bearbeiten (Abbildung 5.18) können Sie Menüs erstellen und dann bearbeiten.

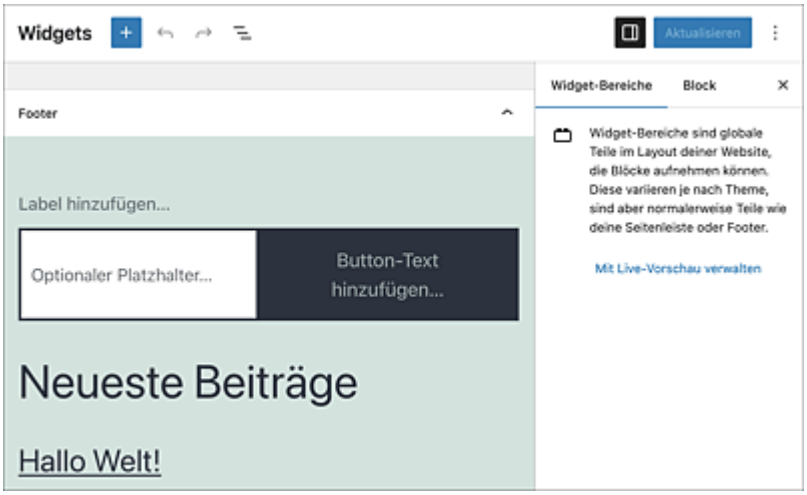


Abbildung 5.17 Seit WordPress 5.8 werden Widget-Bereiche auch mit dem Block-Editor bearbeitet, mit dem Sie seit WordPress 5 Inhalte für Seiten und Beiträge anlegen. Abhängig vom Theme gibt es hier mehrere Bereiche, in die Sie Blöcke einfügen können.

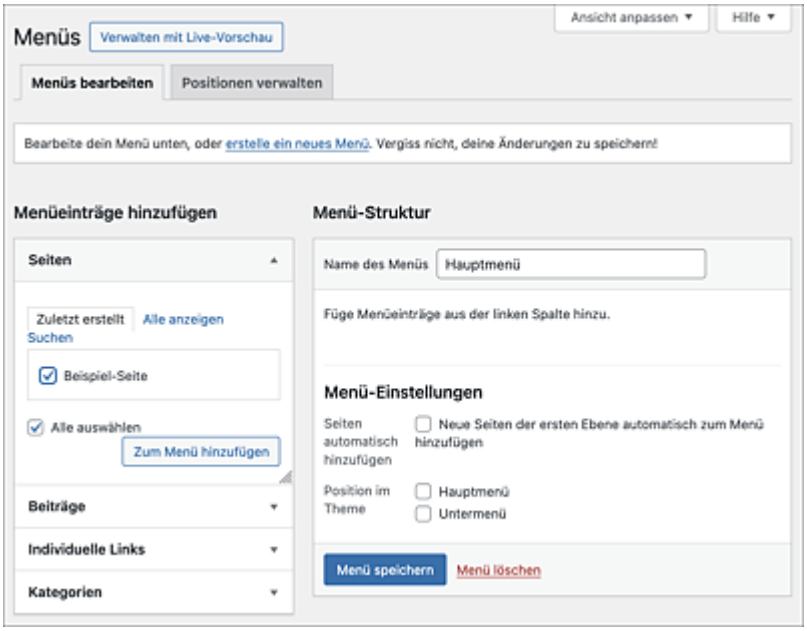


Abbildung 5.18 Bei Classic Themes werden Menüs über eine eigene Einstellungsseite angelegt und verwaltet.

Unter der Liste der Links eines Menüs können Sie die Position im Theme bestimmen – ob und welche hier erscheinen, hängt vom Theme ab. Wie die Pflege von Menüs genauer funktioniert, erfahren Sie in Kapitel 10, »Design anpassen«. Dort wird auch auf die Menü-

Verwaltung in Block-Themes eingegangen, die direkt über den Website-Editor stattfindet.

- Hintergrund (Classic Themes)
  Dieser Link wird angezeigt, wenn das Theme Einstellungsmöglichkeiten für den Hintergrund (Farbe und/oder Bild) anbietet, und öffnet den entsprechenden Bereich im Customizer.

- Theme-Datei-Editor (Classic Themes)
  Achtung: nur für Menschen mit Programmierkenntnissen! Hier öffnet sich ein Editor, mit dem Sie direkt die Dateien des installierten Themes bearbeiten können (Abbildung 5.19).

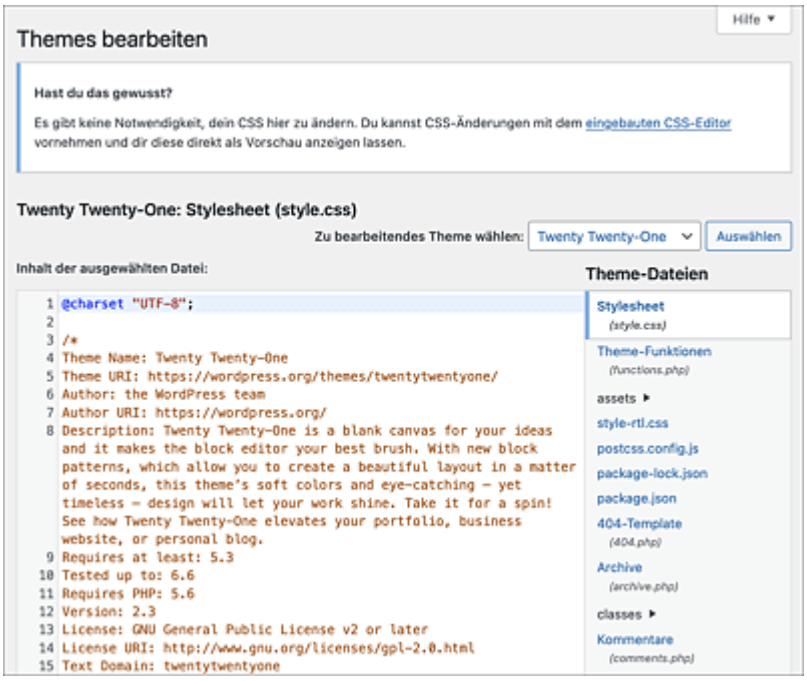


Abbildung 5.19 Für Notfälle kann der Theme-Datei-Editor mal ganz nützlich sein, aber er ist mit Vorsicht zu genießen (sonst führt er selbst zu einem Notfall).

Dringende Fixes können hier vielleicht in Ausnahmefällen vorgenommen werden, aber wenn es blöd läuft, können Sie sich durch einen Syntaxfehler die Website zerschießen und

sich aus dem Backend aussperren. Es gibt zwar inzwischen einen Mechanismus, der verhindern soll, dass hier gemachte Änderungen zu einem so fatalen Fehler auf der Website führen, dass das Backend nicht mehr aufgerufen werden kann (und Sie die Änderung nicht mehr über diesen Datei-Editor rückgängig machen können), aber diese Möglichkeit sollte trotzdem mit Vorsicht und nur in Ausnahmefällen genutzt werden – und auch nur, wenn es sich um Ihr eigenes Theme handelt, da sonst nach einem Theme-Update die Änderungen weg sind. Wenn Sie Anpassungen am CSS machen möchten, gibt es dafür unter Customizer • Zusätzliches CSS ein Feld.

- Website-Editor (Block-Themes)
  Dieser Menüpunkt führt Sie zum in WordPress 5.9 eingeführten Website-Editor (Abbildung 5.20).

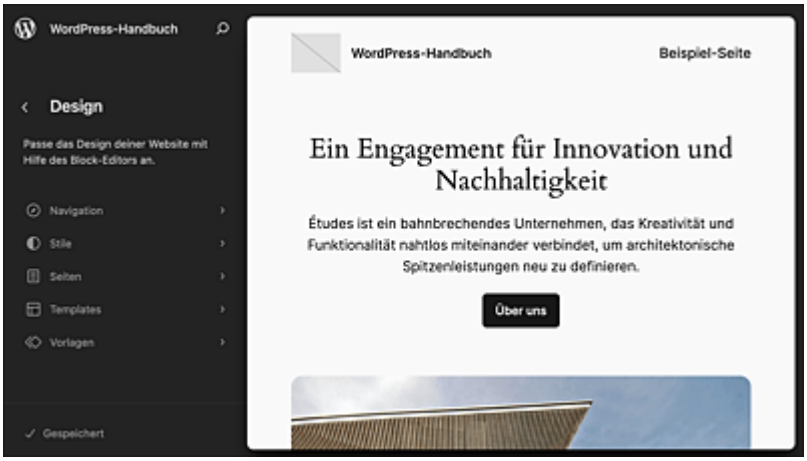


Abbildung 5.20 Im »Website-Editor« können Sie das gesamte Layout eines Block-Themes anpassen.

Hier können Sie, sofern ein Block-Theme aktiv ist, alle Bereiche Ihrer Website bearbeiten, inklusive Header und Footer. Eine detailliertere Einführung in den Website-Editor gibt es in Abschnitt 10.3, »Block-Theme anpassen: den Website-Editor kennenlernen«.

❼ Plugins

- Installierte Plugins
  Zeigt eine Liste der installierten Plugins an und gibt Ihnen die Möglichkeit, die Plugins zu aktivieren und zu deaktivieren, zu aktualisieren und zu löschen (siehe Abbildung 5.21).
- Neues Plugin hinzufügen
  Hier können Sie nach neuen Plugins suchen (Abbildung 5.22).

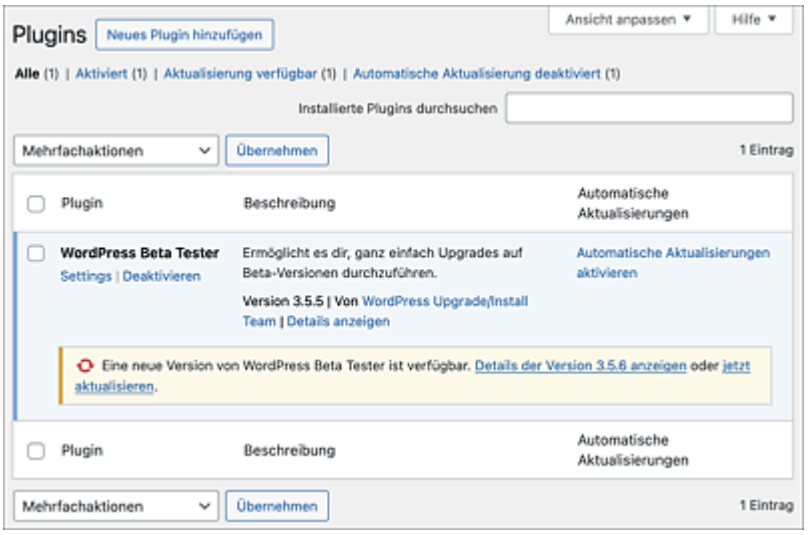


Abbildung 5.21 Über Plugins können Sie WordPress um weitere Funktionen erweitern. In dieser Liste sehen Sie alle installierten Plugins.

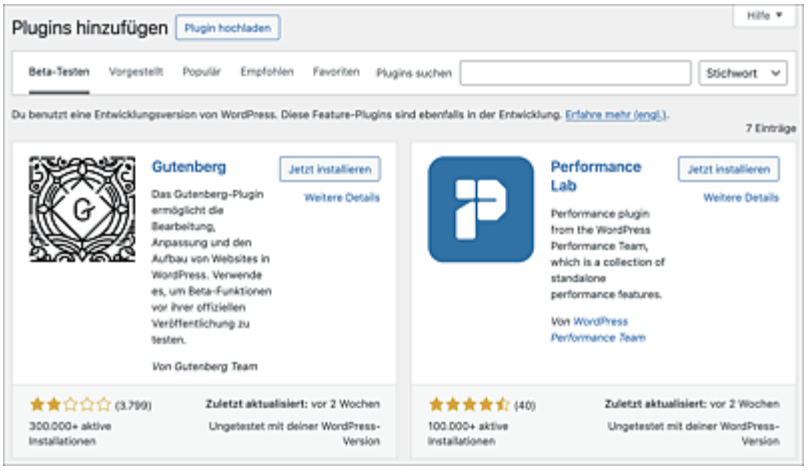


Abbildung 5.22 Eine der Stärken von WordPress ist die große Anzahl an Plugins, um den Funktionsumfang zu erweitern. Installieren Sie am besten aber nicht einfach blind irgendwelche Plugins, sondern stellen Sie sicher, dass ein Plugin wirklich das macht, was Sie brauchen, und dass es noch gepflegt wird.

Bevor Sie ein neues Plugin installieren, sollten Sie prüfen, ob es halbwegs aktuell ist (Datum der letzten Aktualisierung), und mal im Support-Forum auf WordPress.org vorbei schauen, das in der Detail-Ansicht der Plugins verlinkt ist, ob Anfragen dort noch beantwortet werden.

- Plugin-Datei-Editor
  Achtung: Wie beim Theme-Datei-Editor sollten Sie diesen Editor nur nutzen, wenn es gar nicht anders geht. Falls Sie nach der Nutzung statt Ihrer Website oder des Backends mal eine weiße Seite sehen, schauen Sie in Kapitel 21, »Notfallmaßnahmen«.

❽ Benutzer

- Alle Benutzer
  Zeigt alle Accounts an, die in der WordPress-Installation registriert sind (Abbildung 5.23). Mehr dazu in Kapitel 7, »Benutzer*innen verwalten«.



Abbildung 5.23 Auf einer frischen Installation wird in der Liste der registrierten Accounts nur Ihr eigener auftauchen, den Sie bei der Installation erstellt haben. Falls Sie den Dienst »Gravatar« nutzen und Ihre E-Mail-Adresse hinterlegt ist, die bei der Installation angegeben wurde, wird Ihr Profilbild anstelle eines generischen Avatars angezeigt.

- Neuen Benutzer hinzufügen
  Seite zum Erstellen eines neuen WordPress-Benutzerkontos, bei dem Sie Ihre Daten eingeben können (siehe Abbildung 5.24).

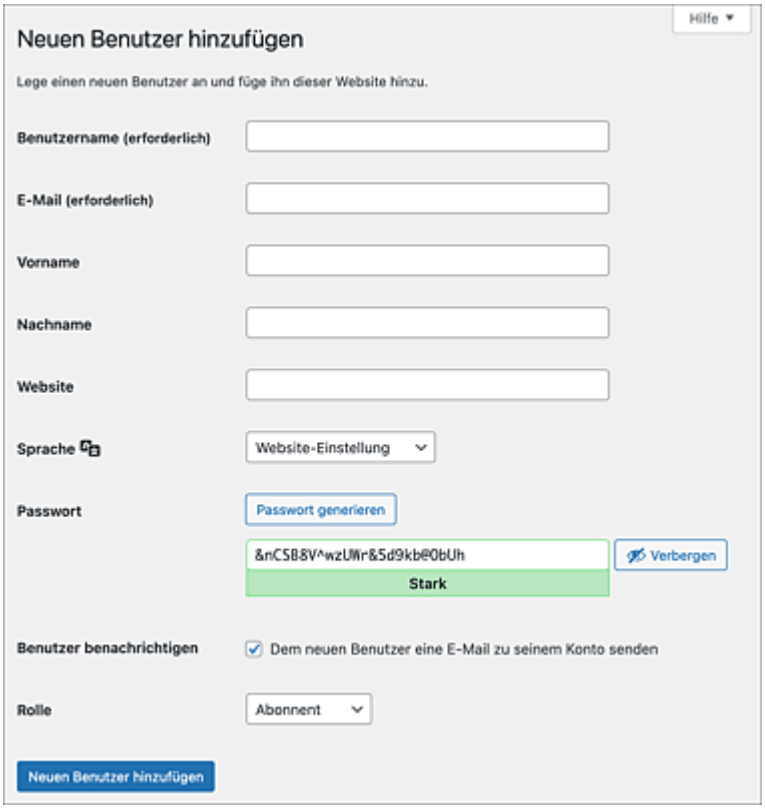


Abbildung 5.24 Wenn Sie mit anderen Personen an der Website arbeiten, können Sie unter »Benutzer« • »Neuen Benutzer hinzufügen« Accounts für diese Personen anlegen.

- Profil
  Link zu Ihrem persönlichen Profil, wo Sie beispielsweise Ihr Passwort und die E-Mail-Adresse ändern können (siehe Abbildung 5.25).



Abbildung 5.25 Unter »Benutzer« • »Profil« können Sie Ihr eigenes Profil bearbeiten.

❾ Werkzeuge

- ### Verfügbare Werkzeuge
  Ein nicht besonders hilfreicher Punkt, der lediglich darauf verweist, dass über den Punkt Werkzeuge • Daten importieren Kategorien in Schlagwörter umgewandelt werden können.

- ### Daten importieren
  Wenn Sie schon eine Website hatten, bevor Sie WordPress selbst aufgesetzt haben, finden Sie hier eine Liste an Plattformen, von denen Sie mit einem offiziellen Plugin exportierte Daten importieren können. Neben Diensten wie Blogger und Tumblr gibt es natürlich auch einen Importer für Daten, die aus einer anderen WordPress-

Instanz exportiert wurden. Praktisch unter anderem auch, wenn Sie vom Anbieter WordPress.com auf eine selbst gehostete Instanz wechseln möchten. Neben diesen hier aufgelisteten Diensten gibt es auch Lösungen für weitere. Suchen Sie einfach einmal in der Suchmaschine Ihres Vertrauens, vielleicht haben Sie Glück.

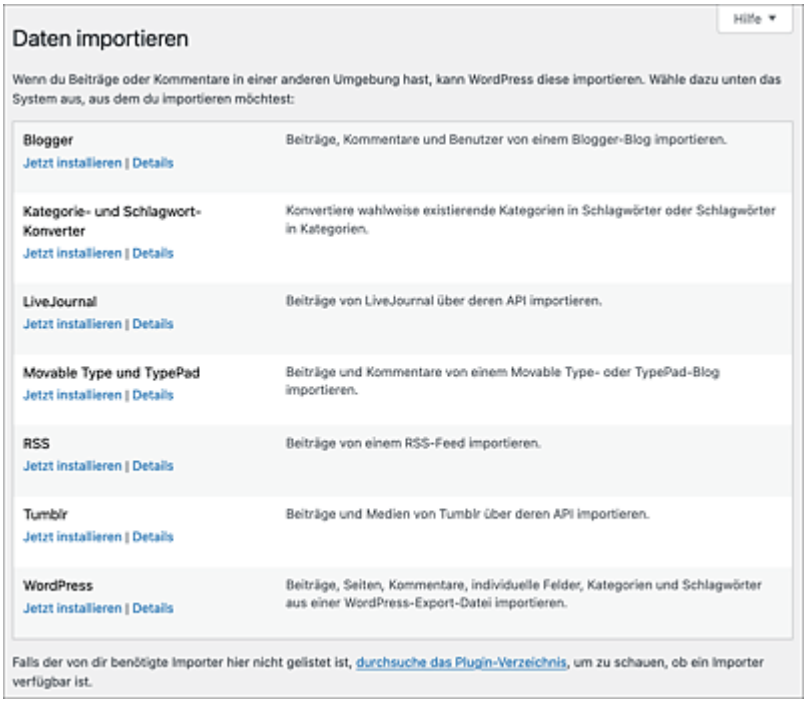


Abbildung 5.26 Zum Import von Inhalten anderer Plattformen gibt es ein paar offizielle Plugins.

- Daten exportieren
  Hier können Sie genau den anderen Weg beschreiten, nämlich Ihre Website-Daten exportieren, etwa um auf ein anderes System zu wechseln (Abbildung 5.27). Sie müssen hier nicht alles für den Export berücksichtigen, sondern können beispielsweise nur die Seiten in den Export einschließen. Als Ergebnis bekommen Sie beim Export eine XML-Datei.

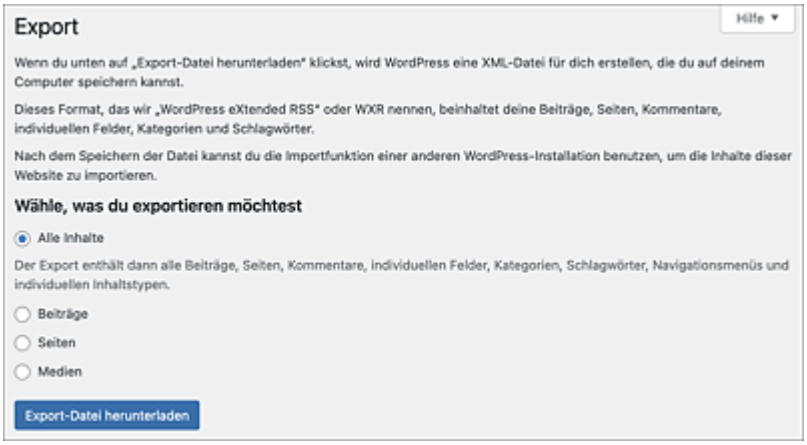


Abbildung 5.27 Hier können Sie Ihre Daten exportieren.

- Website-Zustand
  Hier finden Sie allerlei Hintergrundinformationen und Handlungsempfehlungen zu Ihrer Installation und dem Server, auf dem sie installiert ist (siehe Abbildung 5.28). Die Seite teilt sich in die Tabs Status und Bericht ein. Unter »Status« sehen Sie beispielsweise, ob Ihre Website HTTPS verwendet, ob inaktive Themes und Plugins installiert sind oder ob empfohlene PHP-Module fehlen. Unter Bericht sehen Sie beispielsweise Informationen wie die PHP-Version des Servers und den Server-Pfad, unter dem die Installation liegt. Außerdem finden Sie eine Liste der Plugins und Themes, unterteilt in aktiv und inaktiv, sowie Must-Use-Plugins. Alle Infos dieses Tabs können Sie sich einfach über einen Button in die Zwischenablage kopieren, um sie beispielsweise bei einer Support-Anfrage mit angeben zu können.
- Personenbezogene Daten exportieren/löschen
  Im Rahmen der Europäischen Datenschutzgrundverordnung wurden Funktionen eingefügt, um personenbezogene Daten exportieren und löschen zu können. In Abschnitt 8.3, »Datenschutz-

Bedenken und -Lösungen«, wird näher darauf eingegangen.

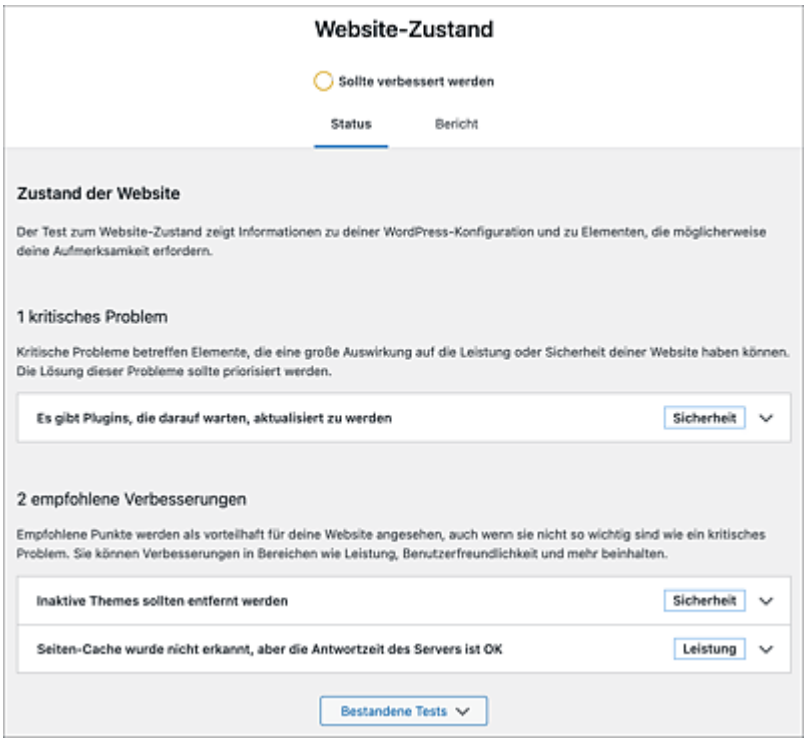


Abbildung 5.28 Das »Website-Zustand«-Werkzeug ist ein nützliches Helferlein, um Ihnen Hinweise zu geben, ob an Ihrer WordPress-Installation und/oder dem Webserver etwas verändert werden könnte.

⑩ Einstellungen

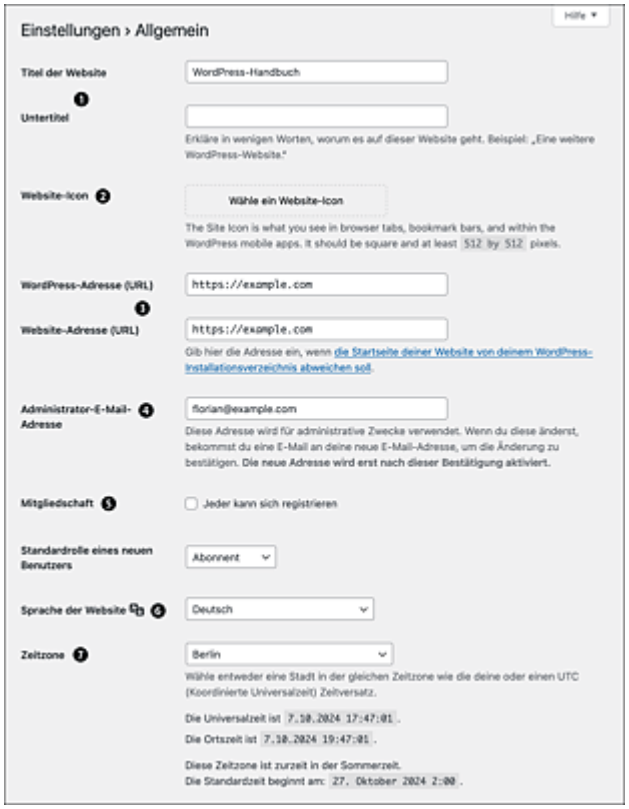
Zu diesem Punkt und seinen Unterpunkten kommen wir im nächsten Abschnitt.

5.2 Einstellungen

Die oberen Punkte des Admin-Menüs, die Sie eben im Schnelldurchlauf kennengelernt haben, werden im nächsten Kapitel zur Inhaltserstellung näher thematisiert, die Punkte unter Einstellungen werden Sie jetzt schon näher kennenlernen, weil Sie an der ein oder anderen Stelle vermutlich direkt am Anfang Änderungen vornehmen möchten.

5.2.1 Allgemein

Hier nehmen Sie grundlegende Einstellungen für die Website vor, wie etwa den Namen und die Administrations-E-Mail-Adresse (siehe Abbildung 5.29).



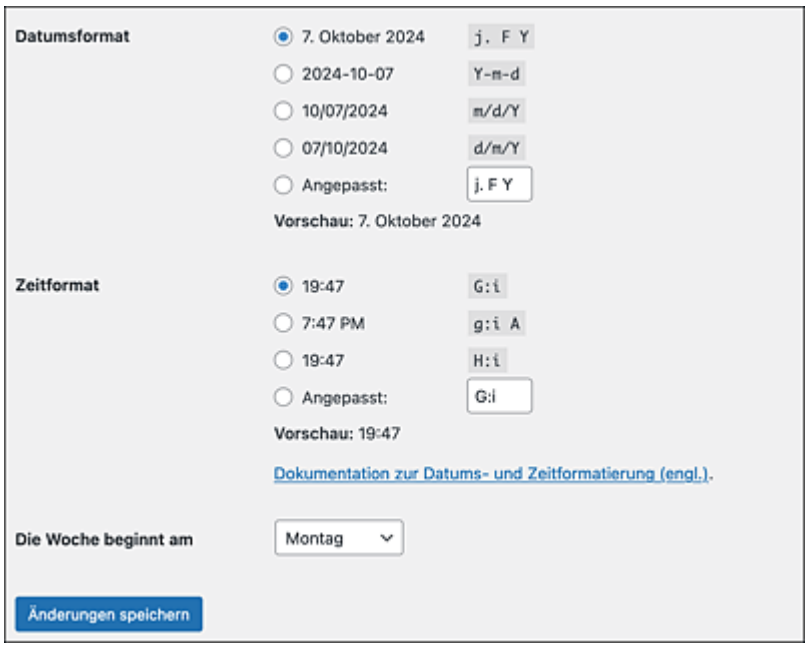


Abbildung 5.29 Auf der »Allgemein«-Einstellungsseite werden wichtige Einstellungen vorgenommen, etwa der Website-Titel.

❶ Titel der Website und Untertitel

Titel der Website ist genau, was es sagt: der Name Ihrer Website. Er taucht beispielsweise oben links in der Admin-Bar auf und wird Besucher*innen auf Ihrer Website meist recht prominent angezeigt, wenn nicht gerade ein Logo genutzt wird. Außerdem wird der Titel standardmäßig als Zusatz in dem `title`-Tag der Website verwendet. Der Wert dieses Tags wird unter anderem für den Inhalt des Browser-Tabs genutzt. Der Untertitel wird auf der Startseite als Zusatz zum Website-Titel angezeigt und je nach Theme auch noch an prominenterer Stelle direkt auf der Website, meist beim Titel.

Die meisten Websites werden ein SEO-Plugin nutzen, mit dem diese Informationen auf Ebene einer Seite oder eines Beitrags überschrieben werden können – mehr zu dem Thema erfahren Sie in Kapitel 20, »Suchmaschinenoptimierung«.

❷ **Website-Icon**
Hier können Sie ein Bild hochladen, das unter anderem als Favicon genutzt wird. Ein Favicon ist das Bild, das im Browser-Tab angezeigt wird. Da es dort recht klein ist, sollten Sie kein Bild nehmen, das zu viele Details aufweist, da diese vermutlich verloren gehen werden.

❸ **WordPress-Adresse (URL)** und **Website-Adresse (URL)**
Meist ist in beiden Feldern dieselbe URL eingetragen, Sie können hier aber, wenn es Ihr Setup erfordert, die URL für das Backend (WordPress-Adresse) auf eine andere URL ändern als die Adresse der öffentlichen Website. Wichtig: In der Regel sollten Sie nach der Installation hier keine Änderungen vornehmen müssen – und wenn doch eine Änderung nötig ist, sollten Sie vorsichtig sein, keinen Fehler zu machen. Sonst könnte es dazu führen, dass das Backend nicht mehr erreichbar ist und Sie die Anpassung nicht einfach über das Backend rückgängig machen können.

❹ **Administrator-E-Mail-Adresse**
Das ist die E-Mail-Adresse, die Sie während der Installation vergeben haben. An diese E-Mail-Adresse werden beispielsweise Benachrichtigungen über automatische Updates geschickt.

❺ Mitgliedschaft und Standardrolle eines neuen Benutzers
Mit der Einstellung Jeder kann sich registrieren können Sie die Registrierung auf Ihrer Website für alle öffnen. Meist werden Sie das nicht wollen, sondern manuell neue Accounts anlegen für die paar Personen, die einen Account brauchen.
Wenn die Option allerdings aktiv ist, können Sie mit Standardrolle eines neuen Benutzers festlegen, welche Rolle der neue Account haben soll (mehr zum Thema Rollen lesen Sie in Kapitel 7, »Benutzer*innen verwalten«). Wenn Sie die Registrierung für alle öffnen, sollten Sie hier keine Rolle auswählen, die Dinge im Backend verändern kann. Es ist also eine gute Idee, die Standard-Auswahl Abonnent beizubehalten.

❻ Sprache der Website
Hier legen Sie die Sprache Ihrer Website fest. Standardmäßig wirkt sich eine Änderung auch auf die Sprache des Backends aus, Sie können in Ihrem Profil aber unabhängig von der Website-Sprache eine andere Sprache für das Backend einstellen.

❼ Zeitzone, Datumsformat, Zeitformat und Die Woche beginnt am
Bei den weiteren Feldern können Sie die Zeitzone Ihrer Website einstellen (wenn Sie in Deutschland sind, ist Berlin eine gute Wahl) und das Datums- und Zeitformat festlegen, das sich beispielsweise auf die Anzeige des Beitragsdatums im Frontend auswirkt. Hier haben Sie neben ein paar vordefinierten Optionen jeweils auch die Möglichkeit, über ein Textfeld ein eigenes Format festzulegen, wobei Sie Platzhalter von PHP verwenden müssen, beispielsweise F für den ausgeschriebenen Namen des Monats. Abschließend können Sie festlegen, mit welchem Tag die Woche beginnen soll.

5.2.2 Schreiben

Auf dieser Einstellungsseite werden Sie vermutlich nicht oft vorbeischauen, aber sie gesehen zu haben, kann ja nicht schaden (Abbildung 5.30).

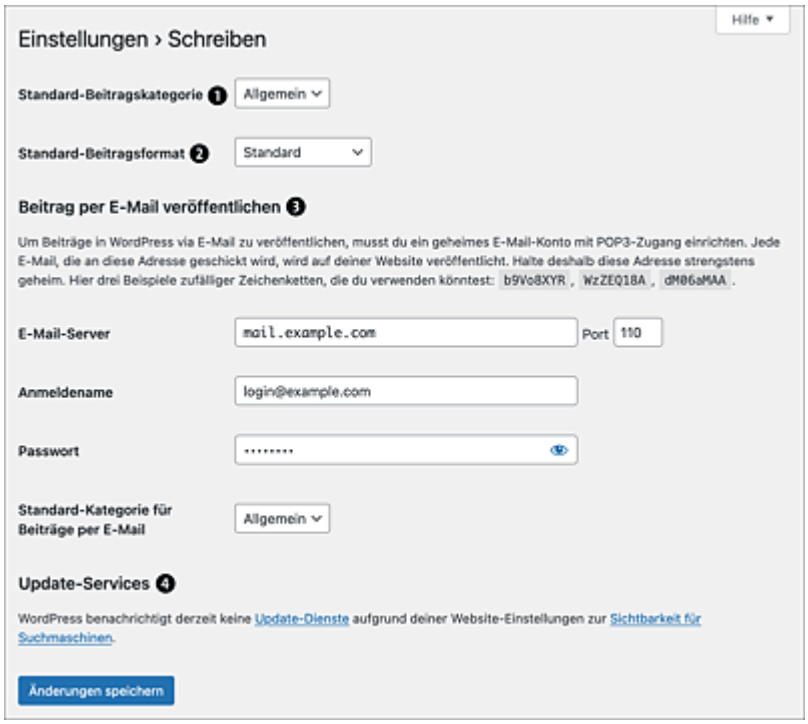


Abbildung 5.30 Die Einstellungsseite »Schreiben« wird meist vermutlich nur zur Kenntnis genommen, ohne Änderungen zu machen.

❶ STANDARD-BEITRAGSKATEGORIE
Hier können Sie festlegen, welche Kategorie beim Erstellen eines neuen Beitrags automatisch vorausgewählt ist.

❷ STANDARD-BEITRAGSFORMAT
Sie können optional für einen Beitrag das Beitragsformat ändern und hier festlegen, welches Format standardmäßig genutzt werden soll. WordPress bringt unter anderem noch die Formate GALERIE oder KURZMITTEILUNG mit. Wenn Sie nicht gerade ein Theme nutzen, das diese Formate unterstützt, werden Sie diese Möglichkeit vermutlich kaum benötigen.

❸ Beitrag per E-Mail veröffentlichen
Hier haben Sie die Möglichkeit, ein E-Mail-Konto anzugeben, um jede bei diesem Konto eingehende Mail als Beitrag zu veröffentlichen. Oft werden Sie das wahrscheinlich nicht nutzen, aber Sie haben zumindest die Option. Denkbar wäre dieser Weg beispielsweise, wenn jemand Inhalte veröffentlichen soll, die Person aber nicht mit dem WordPress-Editor zurechtkommt oder aus irgendwelchen Gründen keinen Zugriff auf das Backend bekommen soll.

❹ Update-Services
Hier können Sie URLs von Services eintragen, die über die Veröffentlichung eines Beitrags auf Ihrer Website informiert werden sollen. Es schadet nichts, den Standard-Wert *http://rpc.pingomatic.com/* stehen zu lassen, Ping-o-Matic ist ein Dienst von der WordPress Foundation.

5.2.3 Lesen

Auf der Lesen-Einstellungsseite (siehe Abbildung 5.31) nehmen Sie Einstellungen vor, die sich direkt auf Ihre Website auswirken, etwa ob die Startseite eine Seite oder die neuesten Beiträge zeigt.

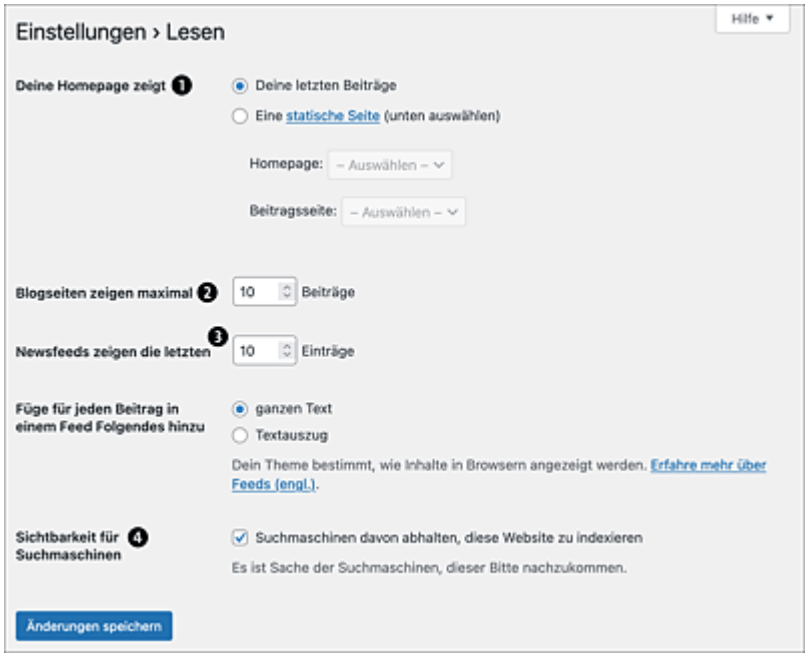


Abbildung 5.31 Bei den »Lesen«-Einstellungen können Sie unter anderem festlegen, ob die Startseite die letzten Beiträge oder eine statische Seite anzeigt.

❶ **Deine Homepage zeigt**

Hier stellen Sie ein, welche Seite für die Homepage und die Beitragsseite genutzt wird. Wenn Sie beim vorausgewählten Wert **Deine letzten Beiträge** bleiben, haben Sie eine Blogseite als Startseite, auf der die letzten X Beiträge angezeigt werden. Wählen Sie **Eine statische Seite (unten auswählen)** aus, werden die beiden Auswahlfelder **Homepage** und **Beitragsseite** aktiv, da Sie jetzt nicht nur die Seite für die Homepage auswählen müssen, sondern auch die, auf der die Beiträge angezeigt werden sollen.

Seit der Block-Editor Einzug gehalten hat und es den Block zur Darstellung der neuesten Beiträge gibt, ist die Trennung zwischen Blogseite und statischer Seite nicht mehr so klar wie vorher. Sie können eine statische Seite mit Blöcken problemlos zu einer Seite machen, auf der Ihre neuesten Beiträge angezeigt werden.

❷ Blogseiten zeigen maximal

Hier stellen Sie ein, wie viele Beiträge maximal in einer Liste angezeigt werden, etwa der Blog-Übersicht oder einer Kategorieseite. Sie sollten hier ein bisschen das Layout beachten, in dem die Beiträge angezeigt werden: Wenn Ihr Theme eine mehrspaltige Darstellung nutzt, sollte die hier eingetragene Anzahl ein Vielfaches davon sein, damit die Spalten schön aufgefüllt werden. Zu hoch sollten Sie mit der Anzahl aber nicht gehen, da darunter die Performance der Seiten mit den Beitragslisten leiden kann.

❸ Newsfeeds zeigen die letzten

Hier stellen Sie ein, wie viele Beiträge in den sogenannten RSS-Feeds Ihrer Website ausgegeben werden. WordPress erstellt mehrere dieser RSS-Feeds, beispielsweise einen für alle Beiträge Ihrer Website (erreichbar unter *example.com/feed*), Feeds für Beiträge einer Kategorie usw. Diese Feeds können mit Feed-Readern abonniert werden. Dadurch werden diese Personen zwar keine direkten Besucher*innen Ihrer Website, Sie sollten aber dennoch als gute Bürger*in des Webs nicht der Versuchung erliegen, den Wert der Option Füge für jeden Beitrag in einem Feed Folgendes hinzu von ganzen Text auf Textauszug zu stellen. RSS-Feeds waren ein wichtiger Baustein des offenen Webs und werden von vielen auch heute noch gern genutzt, um an einer zentralen Stelle die für sie interessanten Inhalte lesen zu können. Machen Sie es Ihren potenziellen Leser*innen daher nicht unnötig schwer.

❹ Sichtbarkeit für Suchmaschinen

Normalerweise möchten Sie, dass Suchmaschinen Ihre Website indexieren und in den Suchergebnissen anzeigen. Damit das passiert, gibt es sogenannte Bots der Suchmaschinen, die das Web durchforsten und Inhalte speichern. Über diese Option können Sie einen Eintrag in der robots.txt ergänzen, der Suchmaschinen davon abhalten soll, Ihre Inhalte zu indexieren.

Wenn Sie unter allen Umständen vermeiden möchten, dass Ihre Inhalte in den Indizes von Suchmaschinen oder beispielsweise im Datenpool von Chatbot-Anbietern wie OpenAI landen, sollten Sie die Website mit einem Passwort schützen.

Hinweis: Hinter den Kulissen des RSS-Feeds

Ein RSS-Feed besteht aus XML-Code und unterscheidet sich streng genommen gar nicht so stark von dem HTML-Quellcode einer Webseite mit sich öffnenden und schließenden Tags. Hier sehen Sie das XML des Feeds der letzten Beiträge einer frischen WordPress-Installation:

```
<?xml version="1.0" encoding="UTF-8"?><rss version="2.0"
    xmlns:content="http://purl.org/rss/1.0/modules/content/"
    xmlns:wfw="http://wellformedweb.org/CommentAPI/"
    xmlns:dc="http://purl.org/dc/elements/1.1/"
    xmlns:atom="http://www.w3.org/2005/Atom"
    xmlns:sy="http://purl.org/rss/1.0/modules/syndication/"
    xmlns:slash="http://purl.org/rss/1.0/modules/slash/"
    >

<channel>
    <title>Meine WordPress-Website</title>
    <atom:link href="http://localhost:8882/feed/" rel="self"
               type="application/rss+xml" />
    <link>http://localhost:8882</link>
    <description>Tadaaa</description>
    <lastBuildDate>Sat, 03 Aug 2024 15:27:01 +0000</lastBuildDate>
```

```
    <language>de-DE</language>
    <sy:updatePeriod>hourly</sy:updatePeriod>
    <sy:updateFrequency>1</sy:updateFrequency>
    <generator>https://wordpress.org/?v=6.6.1</generator>
    <item>
        <title>Hallo Welt!</title>
        <link>http://localhost:8882/hallo-welt/</link>
        <comments>http://localhost:8882/hallo-welt/#comments</comments>

        <dc:creator><![CDATA[admin]]></dc:creator>
        <pubDate>Sat, 03 Aug 2024 15:27:01 +0000</pubDate>
                <category><![CDATA[Allgemein]]></category>
        <guid isPermaLink="false">http://localhost:8882/?p=1</guid>

                    <description><![CDATA[Willkommen bei WordPress. Dies ist dein
erster Beitrag. Bearbeite oder lösche ihn und beginne mit dem Schreiben!]]>
</description>
                                            <content:encoded><![CDATA[
<p>Willkommen bei WordPress. Dies ist dein erster Beitrag. Bearbeite oder lösche
ihn und beginne mit dem Schreiben!</p>
]]></content:encoded>

                    <wfw:commentRss>http://localhost:8882/hallo-
welt/feed/</wfw:commentRss>
            <slash:comments>1</slash:comments>
            </item>
    </channel>
</rss>
```

1. Der Feed-Header gibt Auskunft über die Kodierung der Daten.
2. Innerhalb des `channel`-Elements gibt es zunächst allgemeine Informationen zum Feed, wie den Titel der Website, die Sprache und die Uhrzeit der letzten Aktualisierung.
3. Die Blogbeiträge sind am umschließenden `item`-Element zu erkennen.
4. Der Titel des Blogbeitrags wird mit `title` angegeben.
5. `link` beinhaltet den Link zum Blogbeitrag und `comments` den Link zu den Kommentaren.
6. In `dc:creator` ist der Autorenname abgelegt, in `pubDate` der genaue Zeitpunkt der Veröffentlichung, und für jede Beitragskategorie gibt es ein `category`-Element.

7. In `guid` ist der Link zum Beitrag ohne schön lesbares URL-Format abgespeichert, dieser Link funktioniert immer (solange nicht aus irgendeinem Grund die ID des Beitrags geändert wird), auch wenn die URL-Struktur geändert wird.
8. In `description` ist der manuell oder automatisch erstellte Beitragsauszug angegeben und in `content:encoded` der Beitragsinhalt.

5.2.4 Diskussion

Auf dieser Einstellungsseite geht es hauptsächlich um das Kommentar-System von WordPress (siehe Abbildung 5.32): Sollen Kommentare erlaubt sein, wenn ja, wie lange und so weiter. Wenn Sie nicht gerade eine sehr aktive Community haben oder über Themen schreiben, die viele Rückfragen hervorrufen (auf Rezeptblogs scheint das beispielsweise der Fall zu sein), werden sich die Kommentare wahrscheinlich ziemlich in Grenzen halten. Reaktionen auf Beiträge gibt es heutzutage meist eher auf den unterschiedlichen Social-Media-Kanälen (Sie können sich diese Reaktionen aber auf Ihr Blog holen. Mehr über die Technische Umsetzung dieser Integration erfahren Sie in Kapitel 15 zum IndieWeb).

Die weit überwiegende Anzahl Kommentare wird wahrscheinlich Spam sein, weshalb Sie ein Anti-Spam-Plugin installieren sollten, wenn Sie die Kommentarfunktion aktiv lassen (in Kapitel 8 wird Ihnen Antispam Bee vorgestellt, das diesen Zweck erfüllt).

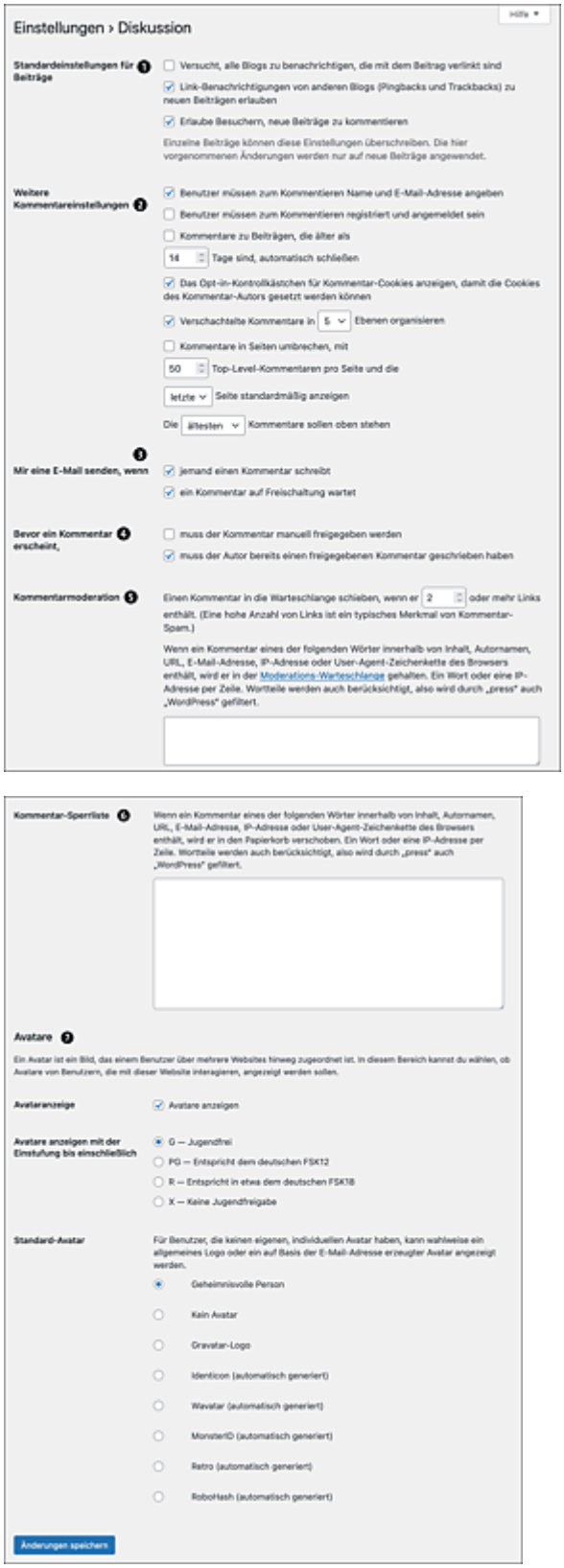


Abbildung 5.32 Auf der »Diskussion«-Einstellungsseite können Sie die Kommentarfunktion Ihrer Website konfigurieren.

❶ Standardeinstellungen für Beiträge

- Mit Versucht, alle Blogs zu benachrichtigen, die mit dem Beitrag verlinkt sind können Sie einstellen, ob Sie beim Veröffentlichen andere Blogs benachrichtigen möchten, auf die Sie verlinkt haben, sodass diese – sofern sie es unterstützen – einen Verweis auf diesen Artikel bei sich anzeigen können (Pingbacks).
- Link-Benachrichtigungen von anderen Blogs (Pingbacks und Trackbacks) zu neuen Beiträgen erlauben behandelt den umgekehrten Fall: Hier legen Sie fest, ob Sie Link-Benachrichtigungen erlauben.
- Über Erlaube Besuchern, neue Beiträge zu kommentieren stellen Sie ein, ob bei neuen Beiträgen standardmäßig die Kommentarfunktion aktiv oder inaktiv sein soll. Auf Beitragsebene kann diese Einstellung angepasst werden.

❷ Weitere Kommentareinstellungen

- Wenn Sie komplett anonyme Kommentare erlauben möchten, entfernen Sie bei Benutzer müssen zum Kommentieren Name und E-Mail-Adresse angeben den Haken. Auch wenn der Haken gesetzt ist, ist das natürlich kein Schutz dagegen, dass falsche Angaben bei Namen und E-Mail-Adresse gemacht werden.
- Benutzer müssen zum Kommentieren registriert und angemeldet sein ist eine höhere Hürde, denn damit wird zum Kommentieren ein Konto bei Ihrer Website benötigt. Das wird wahrscheinlich auch viele abschrecken, die eigentlich einen hilfreichen Kommentar schreiben wollen, aber unter Umständen kann dies eine nützliche Funktion sein, wenn

Sie etwa die Kommentarfunktion auf einen gewissen Kreis beschränken möchten.

- Wenn Sie aus irgendeinem Grund die Kommentarfunktion nur für eine gewisse Zeit offen haben möchten, etwa weil Sie nur sehr aktuelle, kurze Zeit relevante Themen behandeln, können Sie die Option Kommentare zu Beiträgen, die älter als X Tage sind, automatisch schliessen nutzen.
- Mit Das Opt-in-Kontrollkästchen für Kommentar-Cookies anzeigen, damit die Cookies des Kommentar-Autors gesetzt werden können können Sie beim Kommentarformular eine Checkbox anzeigen, mit der die Daten (Name, E-Mail-Adresse und Website) der kommentierenden Person in einem Cookie gespeichert werden, sodass sie beim nächsten Kommentar vorausgefüllt werden können (siehe Abbildung 5.33).
- Verschachtelte Kommentare in X Ebenen organisieren gibt an, wie tief Kommentare mit Antworten verschachtelt werden können.
- Kommentar in Seiten umbrechen ist gegebenenfalls sinnvoll, wenn Sie sehr viele Kommentare haben. Die Funktion unterteilt die Liste der Kommentare dann auf mehrere Seiten und stellt eine Paginierung dar, mit der die anderen Seiten angezeigt werden können. Mit den Einstellungen unter der Checkbox können Sie festlegen, wie viele Top-Level-Kommentare pro Seite angezeigt werden sollen, ob die letzte oder erste Seite standardmäßig angezeigt werden soll und welche Kommentare oben stehen sollen.

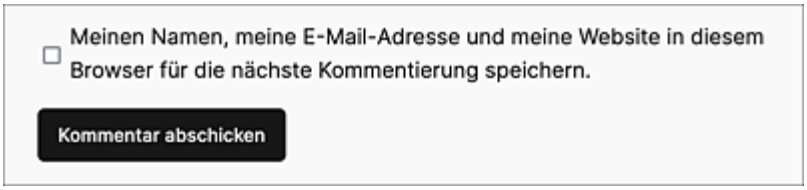


Abbildung 5.33 Wenn Personen oft Kommentare schreiben, kann es hilfreich sein, wenn Name, Mail-Adresse und Website automatisch vorausgefüllt werden.

❸ Mir eine E-Mail senden, wenn

Hier können Sie einstellen, ob Sie benachrichtigt werden möchten, wenn jemand einen Kommentar schreibt und/oder ein Kommentar auf Freischaltung wartet. So behalten Sie den Überblick, auch wenn Kommentare automatisch freigeschaltet werden (siehe Abbildung 5.34).

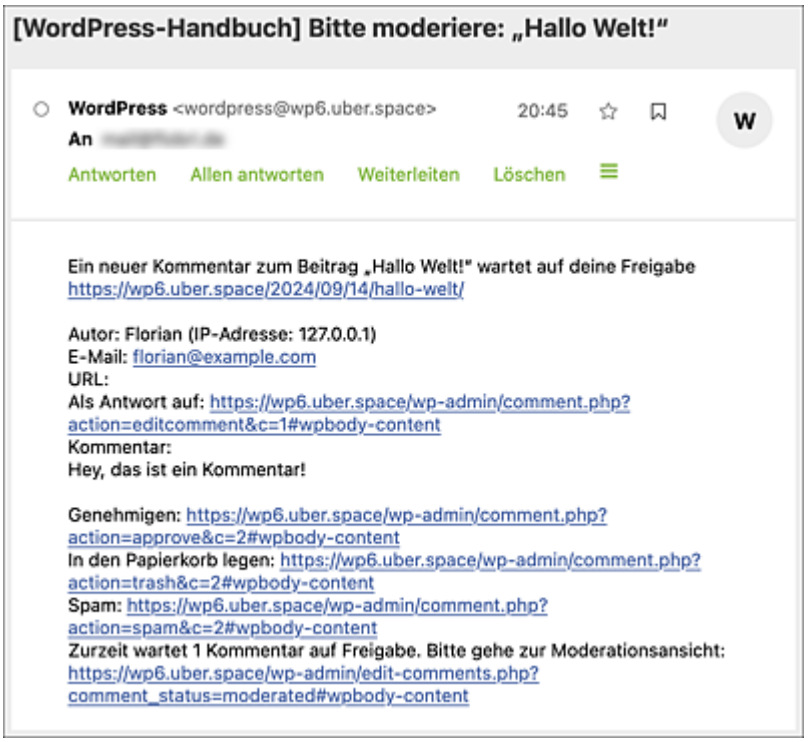


Abbildung 5.34 Wenn Sie E-Mail-Benachrichtigungen eingestellt haben, bekommen Sie bei einem neuen Kommentar eine solche E-Mail mit hilfreichen Links zum Markieren als Spam oder zum Löschen.

❹ **Bevor ein Kommentar erscheint**
Sie können Kommentare automatisch freischalten, in der Regel wollen Sie aber wahrscheinlich vor der Freischaltung schauen, was kommentiert wurde, um keine problematischen Äußerungen auf der Website zu haben. Sie haben die Wahl, alle Kommentare manuell freizuschalten (**muss der Kommentar manuell freigegeben werden**) oder Personen zu vertrauen, die bereits einen freigeschalteten Kommentar haben (**muss der Autor bereits einen freigegebenen Kommentar geschrieben haben**). Die zweite Option ist natürlich keine absolute Sicherheit, denn niemand wird daran gehindert, eine beliebige E-Mail-Adresse im Kommentarformular einzugeben, und wenn dann zufällig eine dabei ist, die schon für einen freigegebenen Kommentar angegeben wurde, wird der Kommentar einer anderen Person direkt freigeschaltet.

❺ **Kommentarmoderation**
Hier geht es um die Verhinderung von Spam, wenn Kommentare automatisch freigeschaltet werden. So werden Kommentare mit zwei oder mehr Links nicht automatisch freigeschaltet, da mehrere Links ein Indiz für Spam sein können. Die Anzahl können Sie mit der Option **Einen Kommentar in die Warteschlange schieben, wenn er X oder mehr Links enthält** anpassen. Darüber hinaus gibt es ein Textfeld, in das Sie pro Zeile ganze Begriffe oder auch nur Teile schreiben können, nach denen WordPress den Kommentar selbst, den Namen der kommentierenden Person, die URL, E-Mail-Adresse, IP-Adresse und den User-Agent-String durchsucht. Bei einem Treffer wird der Kommentar zurückgehalten und nicht automatisch freigeschaltet.

❻ Kommentar-Sperrliste

Hier haben Sie ein ähnliches Feld wie bei Kommentarmoderation, mit dem wichtigen Unterschied, dass Kommentare hier bei einem Treffer direkt in den Papierkorb gelegt werden. Solch eine Ausschlussliste wird auch Blocklist genannt.

Tipp: Nutzen Sie eine fertige Sperrliste

Von null an eine Sperrliste zu erstellen, ist quasi ein unmögliches Unterfangen, es gibt einfach zu viele Zeichenketten, die auf Spam hinweisen. Glücklicherweise gibt es ein Projekt, das eine Blocklist pflegt, die Sie einfach verwenden können. Infos zu dem Projekt finden Sie unter *http://r-wrk.de/wp6_sperrliste_info* und die zum Kopieren bereite Liste unter *http://r-wrk.de/wp6_sperrliste*. Fügen Sie die Liste einfach in das Feld der **Kommentar-Sperrliste** ein (siehe Abbildung 5.35).

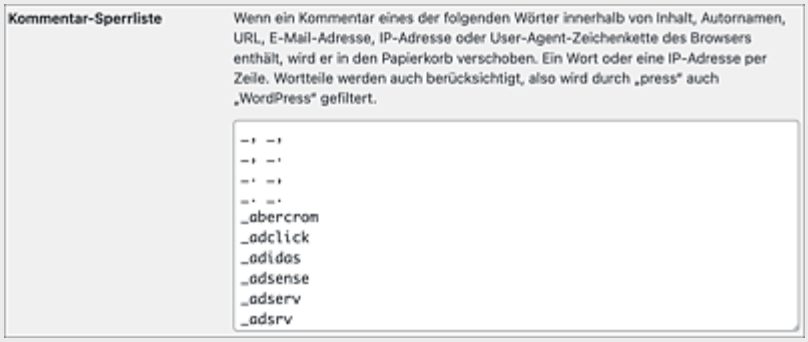


Abbildung 5.35 Die empfohlene Blocklist ist über 700 KB groß und beinhaltet entsprechend viele Begriffe und andere Zeichenketten aus aller Welt.

❼ Avatare

Ein Avatar ist eine andere Bezeichnung für ein Profilbild. Dabei muss es natürlich kein Foto, sondern kann auch eine Grafik sein. Diese Bilder werden in vielen Themes neben den Kommentaren angezeigt, wobei die Verknüpfung anhand der

E-Mail-Adresse über den Dienst *Gravatar* läuft, bei dem Sie Ihre E-Mail-Adresse(n) hinterlegen und diesen Bilder zuweisen können (siehe Abbildung 5.36).

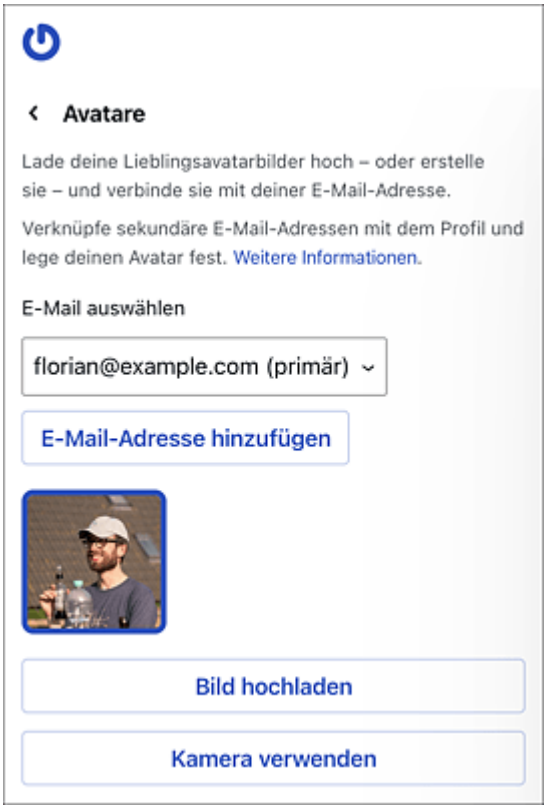


Abbildung 5.36 Gravatar ist ein Dienst von Automattic, der Firma hinter WordPress.com. Daher nutzt der Dienst auch denselben Login wie WordPress.com, Sie müssen sich also für die Nutzung von Gravatar dort einen Account erstellen, falls noch nicht geschehen.

- Avataranzeige lässt Sie steuern, ob Avatare angezeigt werden.
- Avatare anzeigen mit der Einstufung bis einschliesslich bietet keine absolute Sicherheit, dass die angezeigten Avatare auch wirklich dieser Einstufung entsprechen, denn jede*r nimmt für den eigenen Avatar diese Einstufung selbst vor. Indem Sie die Option auf G — Jugendfrei setzen, werden aber immerhin keine Avatare von Menschen angezeigt, die es korrekt einer anderen Einstufung zugeordnet haben.

- Standard-Avatar ermöglicht Ihnen festzulegen, was angezeigt werden soll, wenn jemand ohne Gravatar einen Kommentar schreibt.

Info: Datenschutzprobleme beim Einsatz von Avataren

Standardmäßig werden die Avatar-Bilder, die bei Gravatar hinterlegt sind, in Ihrem Kommentarbereich direkt von den Gravatar-Servern eingebunden. Dadurch werden beim Aufruf einer Seite mit Kommentaren und Gravatar-Avataren persönliche Daten (die IP derjenigen, die die Seite aufgerufen haben) an Gravatar übertragen. Abhilfe schafft hier das Plugin Avatar Privacy (*http://r-wrk.de/wp6_avatarprivacy*), das die Bilder auf Ihrem Server zwischenspeichert, sodass die bei den Kommentaren angezeigten Bilddateien vom Server Ihrer Website geladen werden statt von Gravatar.

5.2.5 Medien

Auf der Medien-Einstellungsseite (siehe Abbildung 5.37) geht es hauptsächlich um Bilder, auch wenn Sie natürlich über die Mediathek oder Beiträge/Seiten alle Arten von Medien zu WordPress hochladen können (bei Videos sollten Sie sich aber überlegen, lieber auf einen Dienst wie Vimeo oder YouTube zurückzugreifen, da dort die Videos optimiert und kleiner an die Nutzer*innen ausgeliefert werden als nicht bearbeitete Videos).

❶ Bildgrösse

Beim Upload werden für Bilder unterschiedliche kleinere Bildgrößen erstellt, weil selten die volle Bildgröße auch in einen Beitrag oder eine Seite eingefügt werden muss. Meist reicht eine kleinere Größe, die dann vielleicht auf die große Version verlinken kann. WordPress selbst registriert einige Bildgrößen, und Plugins und Themes können zusätzliche Bildgrößen registrieren (wie das geht, erfahren Sie im Plugin-Entwicklungskapitel). Die Maße der WordPress-Größen können Sie hier anpassen. Sie geben die maximale Breite und Höhe an, und die Bilder werden darin eingepasst. Eine Ausnahme gilt für Vorschaubilder, hier haben Sie die Möglichkeit, das Bild auf die exakten Maße zu beschneiden.

Bei der anfänglichen Konfiguration müssen Sie hier nichts anpassen, aber eventuell später, wenn klar ist, ob das eingesetzte Theme eigene passende Bildgrößen mitbringt oder nicht.

❷ Dateien hochladen

Standardmäßig ist die Option Meine Uploads in monats- und jahresbasierten Ordnern organisieren angehakt, was bedeutet, dass im *wp-content/uploads*-Verzeichnis für jedes Jahr, in dem mindestens eine Datei hochgeladen wurde, ein Jahresordner existiert. In diesem Ordner wiederum gibt es für jeden Monat einen Ordner, in dem eine Datei hochgeladen wurde. In diesen Ordnern liegen dann die Dateien. Ein möglicher Pfad wäre also wp-content/uploads/2024/08/handbuch.pdf.

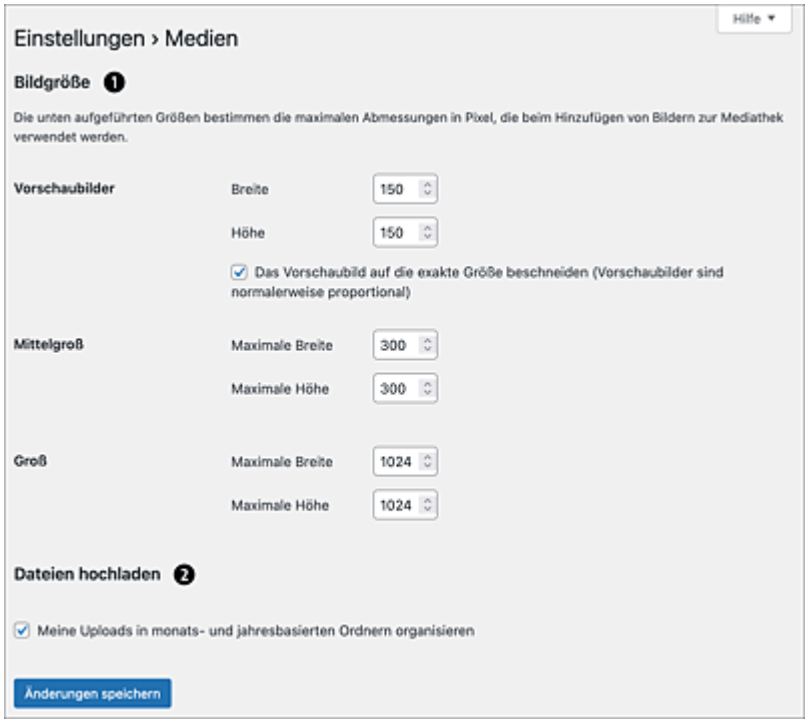


Abbildung 5.37 Auf der »Medien«-Einstellungsseite gibt es Optionen für die von WordPress generierten Vorschaubilder.

5.2.6 Permalinks

Die Optionen dieser Seite sind dafür da, das Format der URLs zu ändern, die WordPress für Seiten, Beiträge, Übersichten und alle anderen Ansichten generiert (siehe Abbildung 5.38).

❶ Permalink-Struktur

Permalink meint die URL zu einem Inhalt, also beispielsweise einer Seite. Mit dieser Option können Sie das Format dieser URLs ändern und haben mehrere Möglichkeiten:

- Einfach ist die Standard-Einstellung und sollte nur genutzt werden, wenn auf dem Webserver nicht die notwendigen Module installiert sind, um eine der anderen Optionen zu verwenden.
- Tag und Name bildet für Beiträge das Veröffentlichungsdatum in der URL mit ab. Für Seiten ist die Struktur dann */beispielseite/*.
- Monat und Name – wenn Sie zwar einen Hinweis auf das Datum in der URL haben möchten, aber nicht bis auf den Tag genau, können Sie diese Option nutzen, die nur Jahr und Monat anzeigt.
- Numerisch ist ähnlich wie Einfach ziemlich wenig hilfreich, wenn es darum geht, von der URL auf den Inhalt schließen zu können (was bei einer URL bestenfalls möglich sein sollte).
- Beitragsname ist vermutlich die am häufigsten gewählte Option. Hier haben Sie eine kurze URL-Struktur nicht nur für Seiten, sondern auch für Beiträge.
- Individuelle Struktur ermöglicht Ihnen, ganz individuell eine eigene Struktur festlegen, mit den über Buttons unter dem Feld angebotenen Platzhaltern.

❷ **Kategorie-Basis**
Standardmäßig wird die URL für eine Übersichtsseite für Beiträge einer Kategorie in WordPress nach dem Format */category/name-der-kategorie* gebildet. Den *category*-Teil können Sie im Feld **Kategorie-Basis** ändern.

❸ **Schlagwort-Basis**
Macht dasselbe wie das Kategorie-Basis-Feld für die Übersichtsseiten von Beiträgen mit einem bestimmten Schlagwort. Die Standard-Basis ist *tag*.

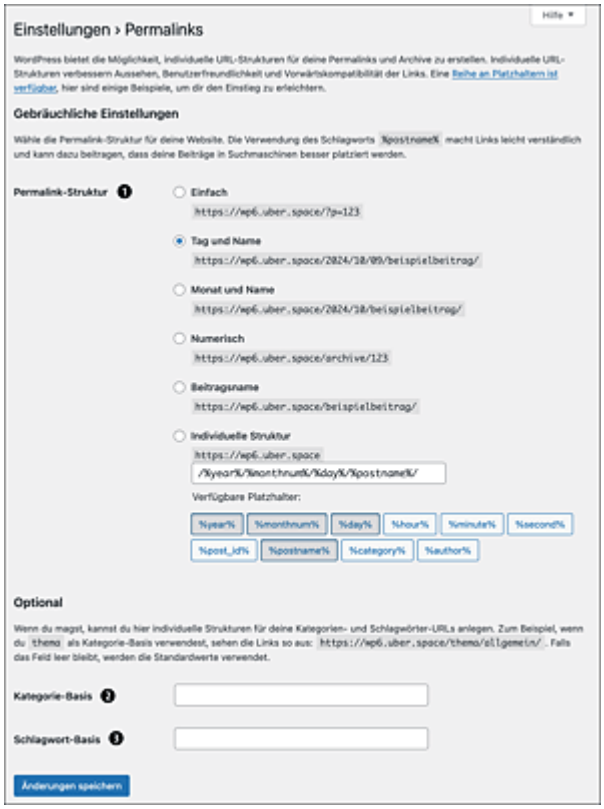


Abbildung 5.38 Auf der Einstellungsseite »Permalinks« können Sie die URL-Struktur anpassen, die WordPress für Ihre Inhalte erzeugt.

5.2.7 Datenschutz

Von hier aus können Sie EINE NEUE SEITE FÜR DIE DATENSCHUTZERKLÄRUNG ERSTELLEN (siehe Abbildung 5.39), die Textvorschläge enthält (bei der Installation von WordPress wird automatisch ein Entwurf für eine Datenschutzerklärung angelegt) – diese Vorschläge können Sie im Tab RICHTLINIEN-LEITFADEN einsehen. Darüber hinaus können Sie im EINSTELLUNGEN-Tab noch die SEITE FÜR DIE DATENSCHUTZERKLÄRUNG ÄNDERN (siehe Abbildung 5.40). Die hier verknüpfte Seite wird beispielsweise unter dem Login-Formular verlinkt. Der Leitfaden kann von Plugins und Themes erweitert werden, um Ihnen einen Startpunkt für Abschnitte für die Datenschutzerklärung zum eigenen Produkt zu geben.

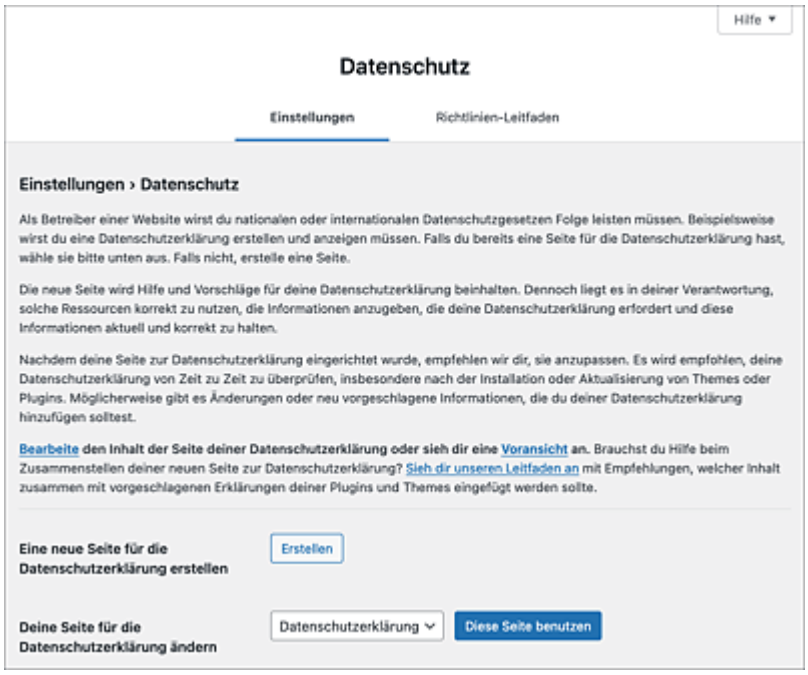


Abbildung 5.39 Im »Einstellungen«-Tab können Sie eine Seite für die Datenschutzerklärung erstellen und festlegen, welche Seite die Datenschutzerklärung enthält.

Denken Sie daran, sich vor dem Livegang Ihrer Website mit der Datenschutzerklärung zu befassen. Es gibt verschiedene Online-Generatoren, bei denen Sie beispielsweise angeben können, ob YouTube-Videos eingebunden werden, und dann einen angepassten

Text bekommen. Wenn Sie sensible Daten von Ihren Besucher*innen verarbeiten, einen Shop haben oder Ähnliches, sollten Sie darüber nachdenken, einen auf Datenschutz spezialisierten Rechtsanwalt zu kontaktieren.

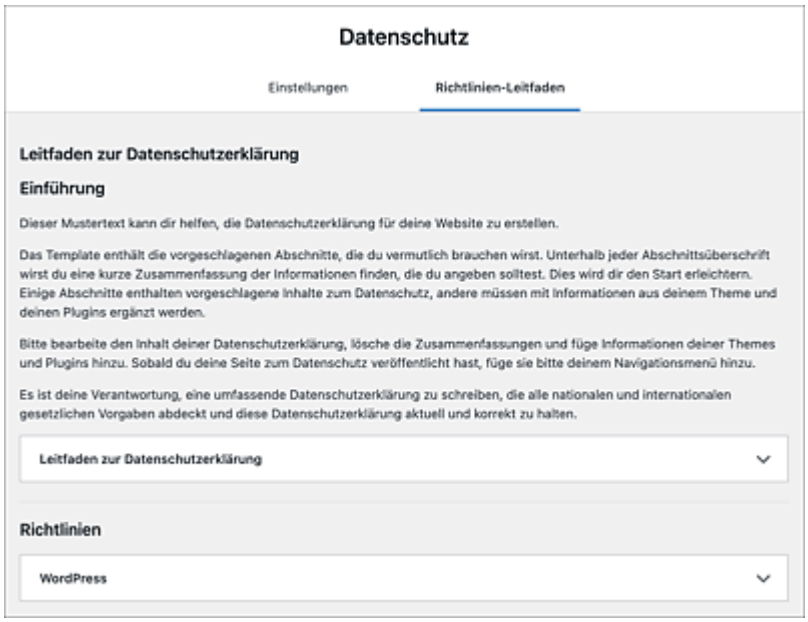


Abbildung 5.40 Im Tab »Richtlinien-Leitfaden« sehen Sie nützliche Tipps zum Anlegen einer Datenschutzerklärung. Die Richtlinien können von Plugins und Themes erweitert werden.

5.3 Einstellungstipps für Fortgeschrittene

5.3.1 Die wp-config.php

So liegt im Hauptverzeichnis Ihrer WordPress-Installation die Datei *wp-config.php*, in der Sie unter anderem die Zugangsdaten zur Datenbank finden, die WordPress während der Installation in die Datei schreibt. Wenn Sie die Datenbank-Daten mal ändern müssen, zum Beispiel bei einem Umzug, ist die *wp-config.php* die richtige Anlaufstelle. In einer frischen *wp-config.php* finden Sie neben den Datenbank-Daten noch Sicherheitsschlüssel, das Tabellenpräfix und den Status des Debug-Modes (`WP_DEBUG`). Daneben gibt es noch einige weitere Möglichkeiten, von denen ein paar im Folgenden beschrieben werden – einige dieser und weitere Optionen werden Sie auch an anderen Stellen des Buchs wiederfinden. Alle möglichen Optionen finden Sie auf der entsprechenden Seite in der WordPress-Dokumentation (*https://developer.wordpress. org/advanced-administration/wordpress/wp-config/*).

Setzen der WordPress- und Website-Adresse

Mit den beiden Konstanten `WP_SITEURL` und `WP_HOME` können Sie die WordPress- und Website-Adresse ändern, die Sie auch in den Einstellungen unter EINSTELLUNGEN • ALLGEMEIN finden. Das ist ein guter Weg, um bei der Änderung einer der beiden URLs die Anpassung erst mal vorzunehmen, ohne dass der Wert in der Datenbank angepasst werden muss.

```
define( 'WP_SITEURL', 'https://example.com/wordpress' );
define( 'WP_HOME', 'https://example.com' );
```

Laut Dokumentation können auch ein paar Datenbankabfragen gespart werden, wenn die Werte in der *wp-config.php* gesetzt werden.

Den wp-content-Ordner verschieben

Normalerweise liegt der Ordner *wp-content*, in dem sich unter anderem Ihre installierten Themes, Plugins und hochgeladenen Dateien befinden, im Hauptverzeichnis der WordPress-Installation. Wenn Sie das ändern möchten, können Sie die Konstanten `WP_CONTENT_DIR` und `WP_CONTENT_URL` verwenden. Der Wert von `WP_CONTENT_DIR` muss ein absoluter Pfad zu dem neuen Verzeichnis sein, `WP_CONTENT_URL` die URL zu dem Verzeichnis – beide Angaben ohne Slash am Ende.

```
define( 'WP_CONTENT_DIR', dirname(__FILE__) . '/content' );
define( 'WP_CONTENT_URL', 'https://example.com/content' );
```

plugins und uploads-Ordner verschieben

Statt den gesamten *wp-content*-Ordner zu verschieben, können Sie zwei der darin enthaltenen Verzeichnisse auch einzeln verschieben, nämlich *plugins* und *uploads*.

Zum Verschieben von *plugins* gibt es die Konstanten `WP_PLUGIN_DIR` und `WP_PLUGIN_URL`, die sich verhalten wie die Konstanten für das *wp-content*-Verzeichnis. Falls Sie Kompatibilitätsprobleme mit Plugins haben, können Sie versuchen, die `PLUGINDIR`-Konstante auf denselben Pfad zu setzen wie `WP_PLUGIN_DIR`.

Für *uploads* gibt es nur eine Konstante: `UPLOADS` können Sie auf einen relativen Pfad setzen, absolute Pfade sind hier nicht möglich.

```
define( 'UPLOADS', 'content/uploads' );
```

Das *themes*-Verzeichnis kann nicht unabhängig von *wp-content* verschoben werden, da es relativ zum *wp-content*-Verzeichnis hart gecoded ist.

Revisionen

WordPress legt standardmäßig sogenannte Revisionen für Beiträge und Seiten an, das heißt, Sie können zu früheren Zuständen eines Inhalts zurückspringen und ihn wiederherstellen (mehr zu Revisionen finden Sie in Kapitel 6).

Ohne Anpassung werden unendlich viele Revisionen gespeichert. Wenn Sie diese Anzahl auf beispielsweise `20` reduzieren möchten, können Sie `WP_POST_REVISIONS` setzen:

```
define( 'WP_POST_REVISIONS', 20 );
```

Auch wenn Sie die Funktion komplett deaktivieren möchten (was nicht zu empfehlen ist, ein paar ältere Versionen für den Notfall zur Hand zu haben, ist nie verkehrt), ist `WP_POST_REVISIONS` zur Stelle: Setzen Sie den Wert dafür auf `false`.

```
define( 'WP_POST_REVISIONS', false );
```

Zusammenfassung von JavaScript im Backend deaktivieren

Alle Skripte für das Backend werden in einer URL zusammengefasst. Falls Sie Probleme im Backend haben, etwa dass der Block-Editor nicht funktioniert, können Sie versuchen, die Konstante `CONCATENATE_SCRIPTS` auf `false` zu setzen.

Zeitraum zum automatischen Löschen des Papierkorbs anpassen

Standardmäßig werden Inhalte beim Löschen erst mal in den Papierkorb geschoben und dort 30 Tage aufbewahrt. Mit `EMPTY_TRASH_DAYS` können Sie die Anzahl der Tage anpassen:

```
define( 'EMPTY_TRASH_DAYS', 25 );
```

Wenn Sie die Konstante auf `0` setzen, wird der Papierkorb deaktiviert, sodass Inhalte sofort dauerhaft gelöscht werden (*Achtung:* WordPress fragt nicht nach einer Bestätigung, bevor ein Inhalt dauerhaft gelöscht wird).

Editor für Plugin- und Theme-Dateien deaktivieren

Wenn Sie nicht möchten, dass über das Backend die Dateien der installierten Plugins und Themes direkt bearbeitet werden können (Sie haben die beiden Editoren weiter oben in diesem Kapitel kurz gesehen), können Sie `DISALLOW_FILE_EDIT` auf `true` setzen. *Hinweis:* Wenn Plugins auf die User-Berechtigung `edit_plugins` prüfen, funktioniert diese Prüfung nicht mehr, wenn die Konstante gesetzt ist. Falls Sie Probleme mit einem Plugin haben, könnte es daher kommen.

SSL für das Backend erzwingen

Mit der Konstante `FORCE_SSL_ADMIN` können Sie festlegen, dass das Backend und die Login-Seite nur über HTTPS aufgerufen werden können.

5.3.2 Die options.php

Unter dem Pfad *wp-admin/options.php* öffnet sich eine Ansicht, die eine lange Liste an Formularfeldern anzeigt (siehe Abbildung 5.41),

mit der Sie die Werte aus der WordPress-Tabelle für Einstellungen, `wp_options`, direkt anpassen können (abgesehen von Feldern, die sogenannte serialisierte Daten enthalten. Das sind besonders kodierte Daten, deren manuelle Änderung sehr fehleranfällig ist). Nutzen Sie diese Seite aber mit Vorsicht und nur, wenn Sie genau wissen, was Sie tun.

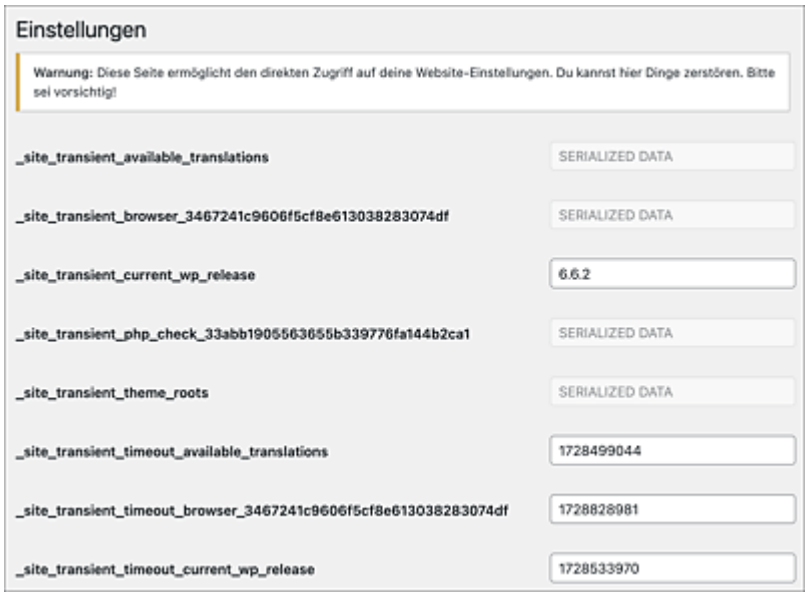


Abbildung 5.41 Unter »wp-admin/options.php« gibt es eine Seite, auf der alle Einstellungen aus der »wp_options«-Tabelle aufgelistet sind und viele direkt verändert werden können.

Wenn Sie beispielsweise nach »blogname« suchen, finden Sie ein Feld, in dem der Website-Titel steht, den Sie unter Einstellungen • Allgemein eingetragen haben und den Sie jetzt auch an dieser Stelle ändern könnten.

6 Inhalte veröffentlichen

Nachdem Sie die Einstellungen und Konfigurationsmöglichkeiten von WordPress kennengelernt haben, geht es jetzt an das Wichtigste: den Inhalt. In diesem Kapitel lernen Sie alles, um Ihre Inhalte veröffentlichen zu können.

Ohne gute Inhalte können Sie das Projekt »Website« eigentlich direkt vergessen, denn natürlich kommen Besucher*innen nicht wegen eines schönen Designs vorbei, sondern wegen der Inhalte. In einer Standard-Installation sind die Inhalte in Beiträge und Seite aufgeteilt; je nach installierten Plugins können weitere Inhaltstypen dazukommen, die in der Regel aber denselben Editor nutzen, wie Sie ihn für Beiträge und Seiten kennenlernen werden.

In diesem Kapitel werden Sie neben dem Anlegen neuer Beiträge und Seiten unter anderem auch (hoffentlich) nützliche Tipps zum Umgang mit Bildern und anderen Medien erhalten sowie die mobilen Apps kennenlernen, mit denen Sie Ihre WordPress-Installation verwalten können. Am Ende des Kapitels sollten alle Grundlagen gelegt sein, damit Sie Ihre Website mit Inhalten füllen können.

Tipp: Inhalte von anderen CMS importieren

Wenn Sie bereits Inhalte von einer alten Website haben und diese mit einem Content-Management-System gebaut wurde, schauen

Sie online, ob es vielleicht eine Export-Möglichkeit der Inhalte gibt und einen entsprechenden Importer für WordPress. WordPress pflegt einige Importer-Plugins für Systeme wie Blogger oder LiveJournal; mehr Informationen zu den Import-Möglichkeiten gibt es in Abschnitt 17.2.4.

6.1 Beiträge schreiben/Seiten anlegen

Nun zu Ihrem ersten Beitrag oder Ihrer ersten Seite. Der Editor für beide Inhaltstypen unterscheidet sich in ein paar wenigen Optionen, ist davon abgesehen aber gleich. Wir werden im Folgenden einen Beitrag erstellen und Besonderheiten gesondert kennzeichnen, die nur für den Beitrags- oder nur den Seiten-Editor gelten.

6.1.1 Erster Blick in den Editor; Titel und Permalink festlegen

Öffnen Sie über Beiträge • Neuen Beitrag erstellen oder Seiten • Neue Seite erstellen den *Block-Editor* (siehe Abbildung 6.1).

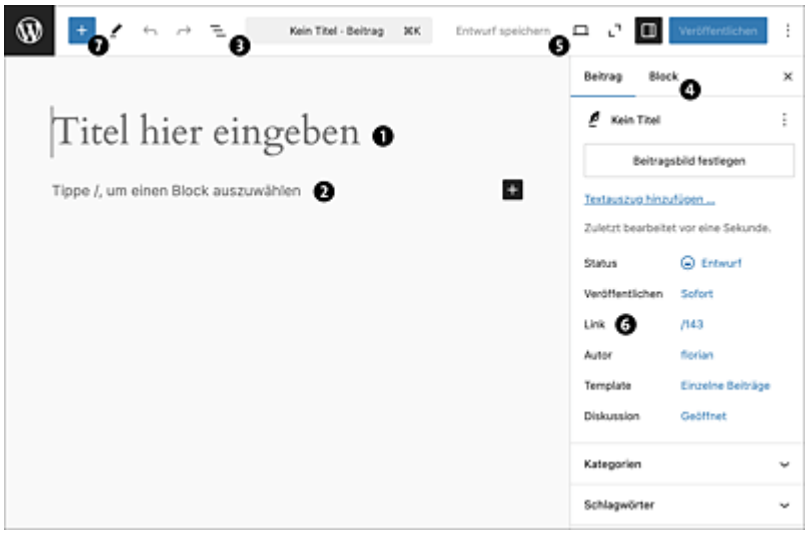


Abbildung 6.1 Im Editor haben Sie links einen großen Bereich für Ihre Inhalte, auf der rechten Seite eine Seitenleiste mit Einstellungsmöglichkeiten für den Beitrag/die Seite und Optionen für ausgewählte Blöcke.

❶ TITEL HIER EINGEBEN

Dieser Titel wird sowohl im Backend von WordPress in der Beitrags-/Seitenliste angezeigt als auch im Frontend als Titel für Ihre Besucher*innen. Dementsprechend sollte der Titel vor der Veröffentlichung gut gewählt werden. Da Sie den Titel jederzeit anpassen können, müssen Sie aber direkt am Anfang noch nicht allzu viel Zeit darauf verwenden – manchmal ist es auch einfacher, einen guten Titel zu finden, wenn der Inhalt fertig ist.

Wenn Sie den Titel eingeben, können Sie rechts in der Seitenleiste sehen, dass der Titel in den »Link« (auch *Permalink* genannt, Sie erinnern sich vielleicht, in Abschnitt 5.2.6 waren Permalinks bereits bei der Anpassung der Permalink-Struktur Thema) des Beitrags oder der Seite umgewandelt wird ❻. Mit einem Klick auf den Link öffnet sich ein Overlay mit der Möglichkeit, den Link anzupassen (siehe Abbildung 6.2). Vergessen Sie nicht, vor der Veröffentlichung einmal den Permalink zu überprüfen und gegebenenfalls anzupassen. *Wichtig:* Wenn Sie den Permalink nachträglich verändern, sollten Sie sich darum kümmern, dass die alte URL auf die neue URL weiterleitet, damit Links auf die alte URL nicht auf eine 404-Seite führen. Dafür können Sie beispielsweise das Plugin »Safe Redirect Manager« nutzen (*https://de.wordpress.org/plugins/safe-redirect-manager/*).

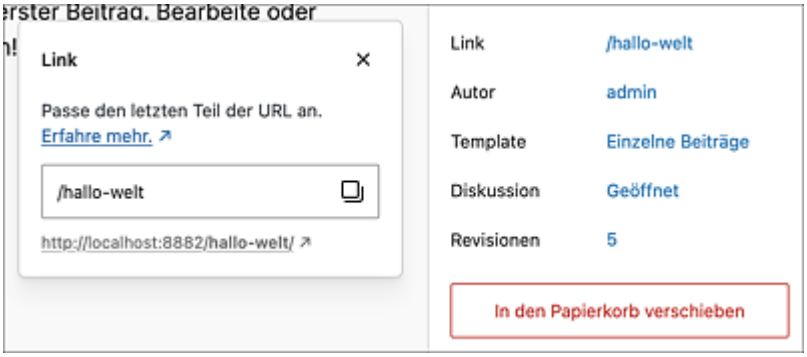


Abbildung 6.2 In der Seitenleiste finden Sie die Option, den »Link« eines Beitrags oder einer Seite anzupassen. Wenn Sie aber den Link eines schon veröffentlichten Inhalts anpassen, sollten Sie daran denken, eine Weiterleitung von der alten auf die neue URL einzurichten.

❷ ### Tippe /, um einen Block auszuwählen

Nachdem Sie einen Titel vergeben haben, wechseln Sie mit Drücken der ↵-Taste oder durch Auswahl mit dem Cursor in den Bereich darunter. Wenn Ihr Beitrag oder Ihre Seite mit einem Absatz beginnen soll, können Sie jetzt direkt lostippen und haben Ihren ersten Absatzblock erzeugt.

Möchten Sie lieber mit einem anderen Block starten, können Sie auf unterschiedliche Art vorgehen:

- Wenn Sie schon genau wissen, welchen Block Sie einfügen möchten und wie er heißt, führt der schnellste Weg über den /-Kurzbefehl. Geben Sie einfach in einen leeren Block, wie es der Platzhalter-Text vorschlägt, einen Schrägstrich ein, und Sie erhalten eine Auswahlliste an Blöcken, die Sie entweder mit der Maus oder mit den Hoch/Runter-Pfeiltasten auswählen können (siehe Abbildung 6.3). Wenn Ihr Block in der Liste nicht dabei ist, geben Sie direkt nach dem Schrägstrich den Namen des Blocks ein. Während Sie tippen, aktualisiert sich die Liste und zeigt zu Ihrer Eingabe passende Blöcke an, sodass Sie eigentlich nie den ganzen Namen eingeben müssen, sondern ein paar Anfangsbuchstaben ausreichen.

- Wenn Sie den passenden Block gefunden und in der Auswahlliste angewählt haben, bestätigen Sie mit ↵ (wenn Sie den Eintrag mit der Maus ausgewählt haben, erübrigt sich der Schritt), und der Block wird im Editor eingefügt.
- Die zweite Möglichkeit ist über das Plus-Symbol auf der rechten Seite eines »leeren« Blocks, das auftaucht, wenn Sie den Block ausgewählt haben. Durch Klick auf das Icon öffnet sich ein kleiner Block-Inserter, der zunächst eine ähnliche Funktion bietet wie die Auswahlliste beim `/`-Befehl: Sie bekommen ein paar Blöcke angezeigt und können die Liste über das Suchfeld auf den gewünschten Block einschränken (siehe Abbildung 6.4).

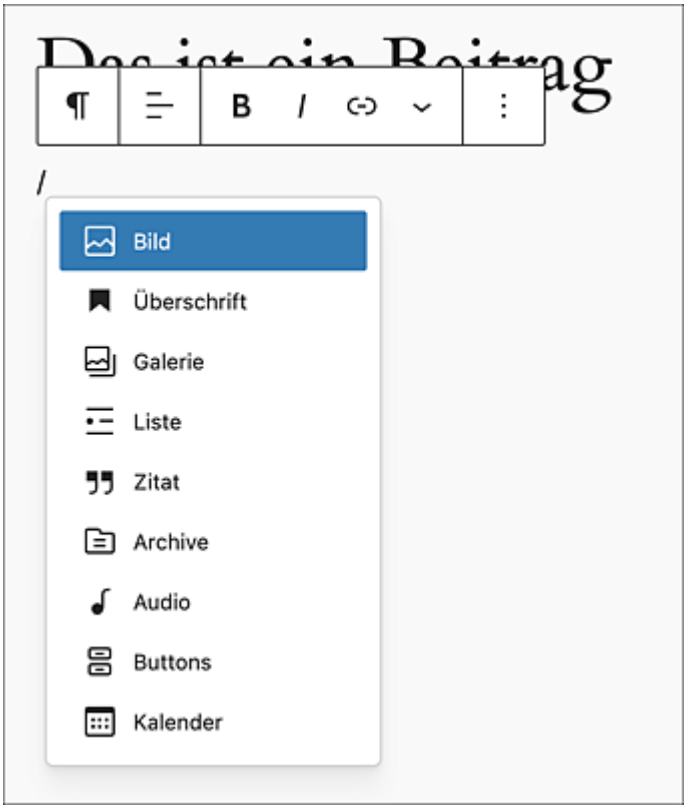


Abbildung 6.3 Der schnellste Weg zum neuen Block ist der »/«-Befehl, mit dem sich eine Auswahlliste an Blöcken öffnet, die Sie über Eingabe des Blocknamens einschränken können.

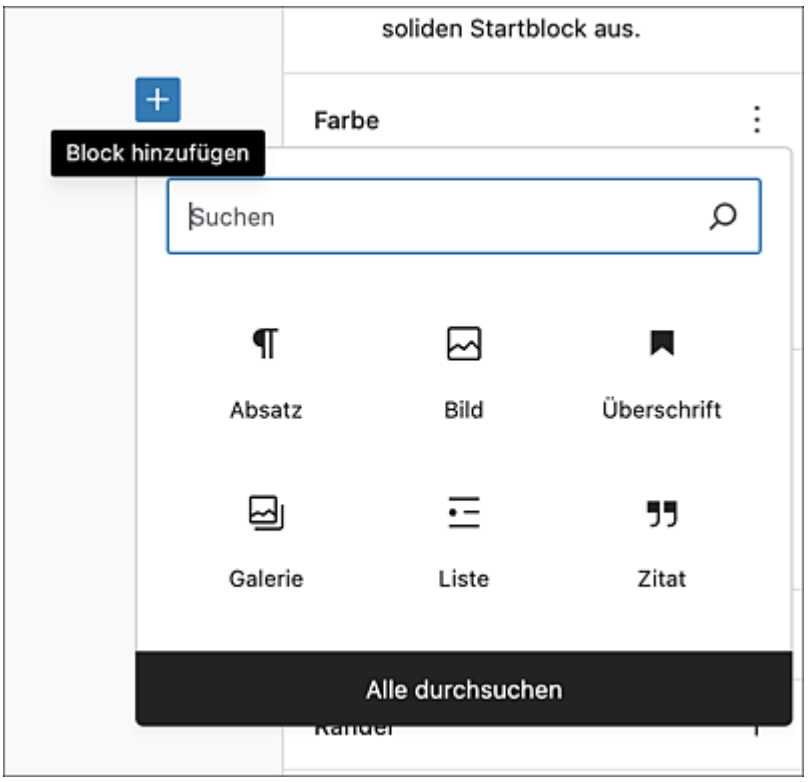


Abbildung 6.4 Über das Plus-Symbol auf der rechten Seite bei einem leeren Absatzblock gelangen Sie in eine kleine Variante des Block-Inserters, der im nächsten Punkt vorgestellt wird. Auch hier können Sie über die Suche die Liste der angezeigten Blöcke einschränken.

- Ein Unterschied zum /-Befehl ist, dass Sie über die Suche auch die Block-Vorlagen durchsuchen können. Das sind von Themes oder Plugins bereitgestellte vorkonfigurierte Blöcke oder Block-Kombinationen, die die Erstellung komplexerer Layout-Elemente vereinfachen. Wenn Sie im kleinen Inserter unten auf ALLE DURCHSUCHEN klicken, öffnet sich die große Version des Block-Inserters, den Sie auch über den dritten Weg direkt erreichen können.
- Über das Plus oben links in der Werkzeugleiste ❼ oder den Button ALLE DURCHSUCHEN im kleinen Inserter öffnet sich der umfangreiche Block-Inserter (siehe Abbildung 6.5). Der Inserter besteht aus drei Tabs: BLÖCKE, VORLAGEN und MEDIEN. Nach dem Öffnen des Inserters ist der Tab BLÖCKE aktiv. Hier

werden Ihnen, gruppiert nach Kategorien, alle Blöcke angezeigt, die Sie im Beitrag oder der Seite nutzen können.

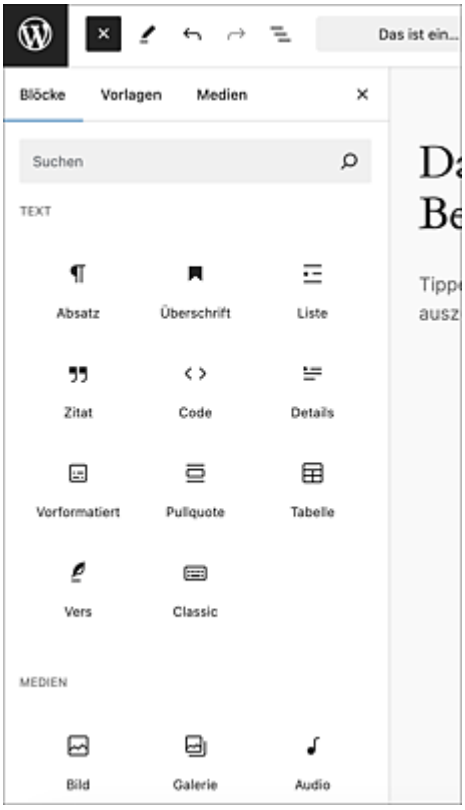


Abbildung 6.5 Die große Version des Block-Inserters erreichen Sie oben links über den Plus-Button oder den Button »Alle durchsuchen« in der kleinen Inserter-Version, die Sie über eins der Plus-Symbole in der Bearbeitungsansicht öffnen können.

- Im Vorlagen-Tab können Sie die eben schon kurz angesprochenen Block-Vorlagen des Themes sowie die kuratierten Vorlagen aus dem WordPress-Patterns-Verzeichnis (*https://wordpress.org/patterns*) durchsuchen und einfügen. Nach dem Einfügen sind die Vorlagen ganz normale Blöcke, die Sie auch einzeln in den Editor einfügen können (eine Ausnahme sind »gesperrte« Vorlagen, die nur bestimmte Aktionen erlauben. Mehr zu Vorlagen und dem Sperren von Vorlagen lesen Sie in Kapitel 10).
- Im Medien-Tab können Sie die Bilder der Mediathek durchsuchen und auf Bilder von Openverse zugreifen, eine

Sammlung frei nutzbarer Bilder von WordPress.org. Durch einen Klick auf ein Bild wird es in den Editor eingefügt.

❸ In der eben schon erwähnten Werkzeugleiste finden Sie neben dem Plus für den Block-Inserter noch die Möglichkeit, über das Stift-Symbol vom Bearbeiten- in den Auswählen-Modus zu wechseln, Aktionen rückgängig zu machen und wiederherzustellen, sowie die Dokumentenansicht der Blöcke, die besonders hilfreich ist, wenn Sie Blöcke mit tieferer Verschachtelung einsetzen, da es dann teilweise etwas unübersichtlich werden kann (siehe Abbildung 6.6).

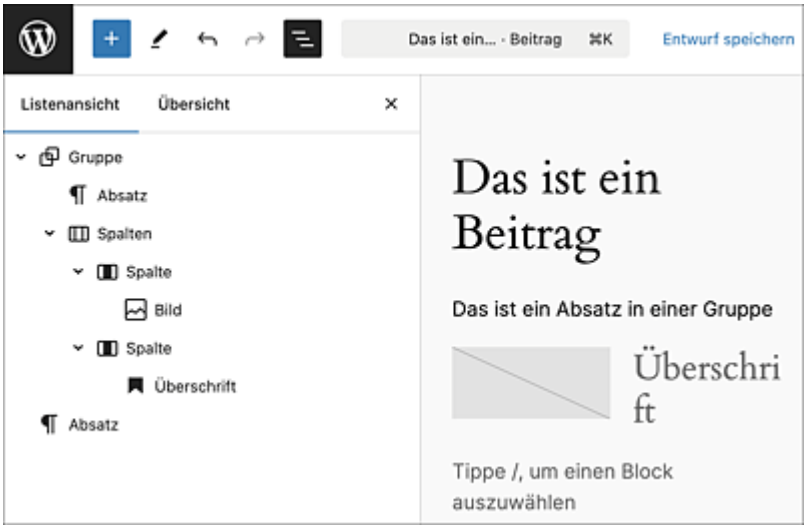


Abbildung 6.6 Mit der Dokumentenansicht haben Sie Zugriff auf eine Listenansicht der im Beitrag oder der Seite eingefügten Blöcke und können so gegebenenfalls einfacher den Überblick bei verschachtelten Block-Strukturen behalten.

❹ Die Seitenleiste ist in zwei Tabs aufgeteilt: Beitrag (oder Seite) und Block. Im Tab Beitrag/Seite finden Sie Einstellungsmöglichkeiten, die die gesamte Seite beziehungsweise den gesamten Beitrag betreffen, wie das Veröffentlichungsdatum, die URL und bei Beiträgen die Kategorien sowie das Beitragsbild. Im Block-Tab sehen Sie die Einstellungsmöglichkeiten zum aktuell ausgewählten Block. Beim Absatz-Block können das zum Beispiel je nach aktivem Theme Optionen für die Textfarbe und die Schriftgröße sein.

❺ Oben rechts finden Sie die Möglichkeit, Ihren Beitrag oder Ihre Seite zu speichern (falls noch nicht veröffentlicht), sich eine Vorschau anzusehen oder den Inhalt zu veröffentlichen (zu all diesen Punkten später in diesem Kapitel mehr).

6.1.2 Hinweise zur Block-Editor-Bedienung

Jeder Block verfügt über eine eigene Werkzeugleiste, die auftaucht, wenn Sie den Block auswählen (siehe Abbildung 6.7). Die in der Werkzeugleiste verfügbaren Optionen unterscheiden sich von Block zu Block, aber einige Buttons finden Sie überall.

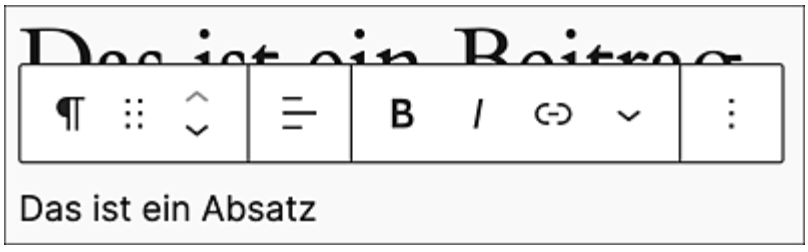


Abbildung 6.7 Die Block-Werkzeugleiste bietet je nach ausgewähltem Block unterschiedliche Optionen – beim Absatz-Block zum Beispiel zur Textausrichtung und zum Setzen einer Verlinkung.

- Das Icon ganz links ist das Icon des aktuellen Blocks. Mit einem Klick auf das Icon bekommen Sie je nach Block-Typ

gegebenenfalls die Möglichkeit, den Block in einen anderen Block umzuwandeln. So können Sie beispielsweise einen Absatz-Block in eine Überschrift oder ein Zitat umwandeln.

- Rechts neben dem Block-Icon ist ein Symbol mit sechs Punkten zu sehen. Das können Sie mit der Maus anklicken und gedrückt halten, um den Block per Drag-and-drop an eine andere Position zu ziehen. Bei verschachtelten Blöcken kann es einfacher sein, die Position über die Listenansicht in der weiter oben beschriebenen Dokumentenansicht zu ändern, auch dort können Sie per Drag-and-drop Blöcke umsortieren (siehe Abbildung 6.6).
- Noch eine Position weiter rechts sehen Sie zwei Pfeil-Icons, eins nach oben und eins nach unten. Mit diesen Buttons können Sie die Position des Blocks je um eine Stelle im Dokument nach oben oder unten verschieben.
- Jetzt folgen die pro Block individuellen Buttons.
- Ganz rechts sehen Sie einen Button mit drei untereinander angeordneten Punkten. Dieses Drei-Punkte-Menü finden Sie an vielen anderen Stellen im Editor wieder, und dahinter verbergen sich immer weitere Einstellungsmöglichkeiten. Im Fall des Drei-Punkte-Menüs in der Block-Toolbar bieten sich Ihnen folgende Optionen (siehe Abbildung 6.8, es kann sein, dass nicht alle Optionen zur Verfügung stehen):
  - Kopieren packt den Block in die Zwischenablage, sodass Sie ihn mit cmd/Strg + V an anderer Stelle (auch in einem anderen Beitrag oder einer anderen Seite) einfügen können.
  - Duplizieren fügt eine Kopie des Blocks nach dem aktuellen Block ein.
  - Davor hinzufügen fügt einen leeren Absatz-Block vor dem Block ein.

- Danach einfügen macht dasselbe, nur hinter dem aktuellen Block.
- Stile kopieren ermöglicht Ihnen, die Design-Einstellungen zu kopieren, die Sie für einen Block in der Block-Seitenleiste vorgenommen haben (zum Beispiel Hintergrundfarbe oder Schriftgröße).
- Mit Stile einfügen können Sie die vorher von einem anderen Block kopierten Stile für einen anderen Block übernehmen.
- Gruppieren umschließt den Block mit einem Gruppe-Block.
- Sperren gibt Ihnen unterschiedliche Optionen, die Bearbeitbarkeit des Blocks und optional seiner Kind-Blöcke einzuschränken.



Abbildung 6.8 Über das Drei-Punkte-Menü in der Block-Toolbar können Sie unter anderem eine Vorlage aus dem oder den Blöcken erstellen, die gerade ausgewählt sind, und den Block verschieben.

- Umbenennen erlaubt Ihnen, dem spezifischen Block einen eigenen Namen zu geben, der in der Listenansicht der Dokumentenansicht auftaucht.
- Mit Vorlage erstellen können Sie den oder die Blöcke in eine Block-Vorlage umwandeln (mehr dazu im Kapitel »Design anpassen«).
- Verschieben nach blendet eine blaue Linie über dem Block ein, die Sie mit den Pfeiltasten nach oben und unten verschieben können. An der Position, an die Sie den Block verschieben möchten, drücken Sie die ↵-Taste und verschieben damit den Block.
- Als HTML bearbeiten zeigt den Code des Blocks an und erlaubt auch die Bearbeitung. Hier müssen Sie bei Änderungen aufpassen, dass Sie den Block nicht kaputtmachen, etwa wenn Markup verändert wird.

Weitere Block-Einstellungen finden Sie in der Block-Seitenleiste. Wenn Sie einen Block auswählen, wechselt die Seitenleiste auf der rechten Seite automatisch in den Tab Block, und Sie finden je nach ausgewähltem Block unterschiedliche Einstellungsmöglichkeiten. Beim Bild-Block können Sie beispielsweise das Seitenverhältnis ändern (siehe Abbildung 6.9).

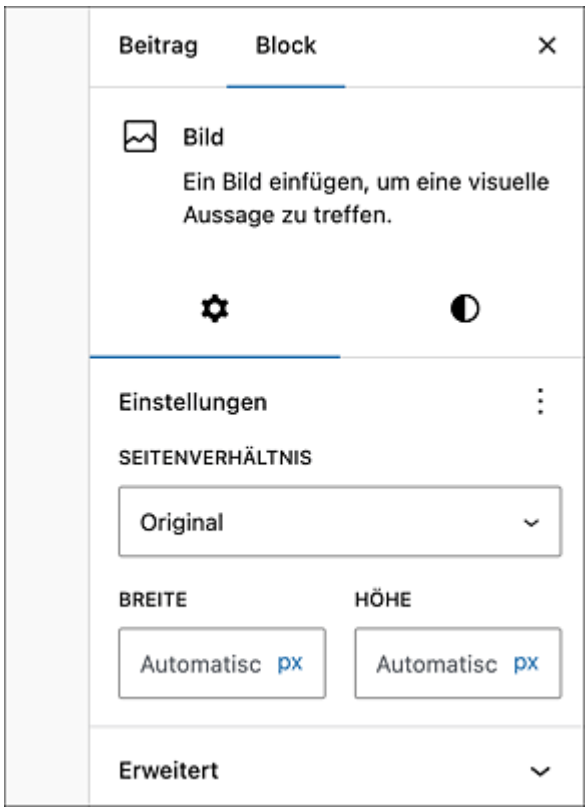


Abbildung 6.9 Für den ausgewählten Block können Sie in der Seitenleiste mehr Einstellungen vornehmen als direkt über die Block-Werkzeugleiste.

Verschieben von Blöcken und mehrere Blöcke auswählen kann im Editor direkt manchmal etwas hakelig sein. Eben haben Sie schon gelernt, dass das Verschieben in der Dokumentenansicht möglich ist, und auch das Markieren von mehreren Blöcken funktioniert in der Listenansicht der Blöcke. Wählen Sie dazu einen Block aus und dann mit gedrückter ⇧-Taste einen anderen, und Sie haben diese beiden und alle dazwischen liegenden Blöcke ausgewählt (siehe Abbildung 6.10). Im Bearbeitungsbereich können Sie mehrere Blöcke auswählen, indem Sie quasi mit gedrückter Maustaste die Blöcke markieren, die Sie auswählen möchten, so als würden Sie Text auswählen wollen.

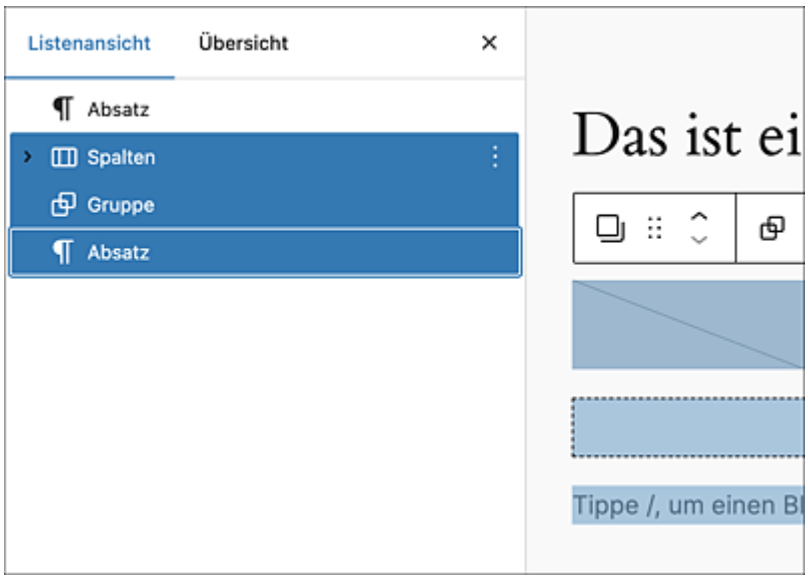


Abbildung 6.10 Die Auswahl mehrere Blöcke ist in der Dokumentenansicht meist einfacher als im Editor.

Die Block-Breadcrumb ganz unten am Rand des Editors ist ein nützlicher kleiner Helfer, wenn es darum geht, in einer verschachtelten Struktur von Blöcken einen Block weiter oben in der Struktur auszuwählen, ohne in die Listenansicht zu wechseln (siehe Abbildung 6.11).

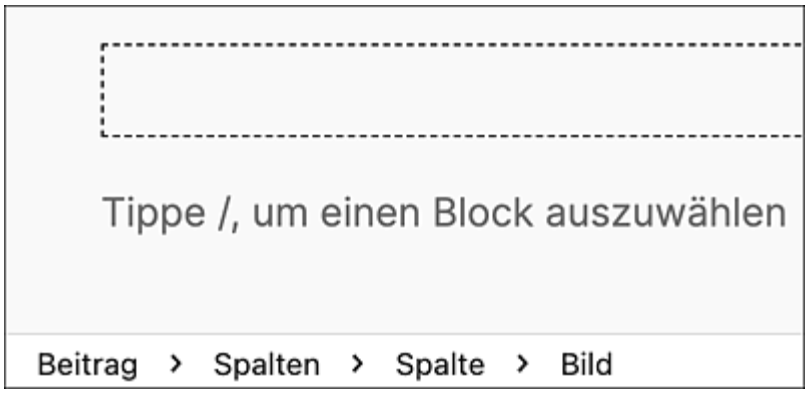


Abbildung 6.11 Diese kleine unscheinbare Breadcrumb ist besonders nützlich bei tief verschachtelten Block-Strukturen.

Editor-Einstellungen und andere Links finden Sie ganz rechts oben über das Drei-Punkte-Menü (siehe Abbildung 6.12). So können Sie hier beispielsweise über Obere Werkzeugleiste aktivieren, dass die

Block-Werkzeugleiste nicht direkt über dem ausgewählten Block angezeigt wird, sondern immer ganz oben mit in der Editor-Toolbar (siehe Abbildung 6.13).

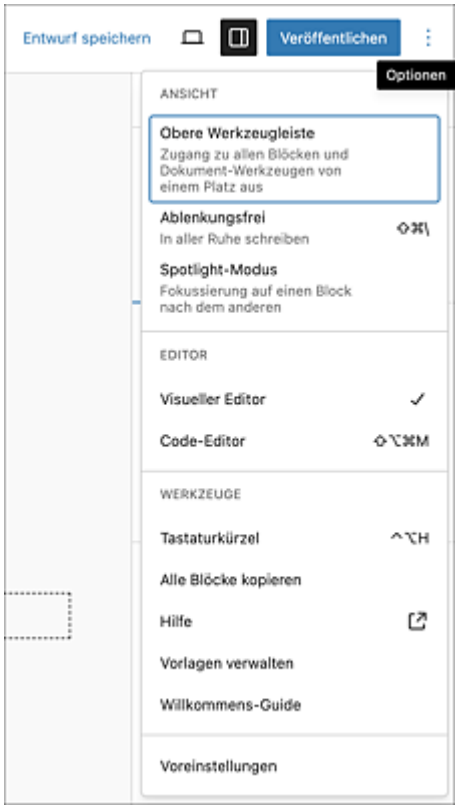


Abbildung 6.12 Im Editor-Drei-Punkte-Menü können Sie unter anderem den Editor-Modus ändern und die Voreinstellungen anpassen.



Abbildung 6.13 Wenn Sie die Option »Obere Werkzeugleiste« aktivieren, werden die Optionen aus der Block-Toolbar mit in der Editor-Toolbar angezeigt; die Blöcke haben dann selbst keine eigene Toolbar mehr.

Darüber hinaus haben Sie die Möglichkeit, den Modus des Editors zu ändern, etwa in einen Ablenkungsfrei-Modus, in dem alle Elemente bis auf den Bereich zur Inhaltsbearbeitung ausgeblendet werden. Wenn Sie in dem Modus den Cursor an den oberen Bildschirmrand bewegen, wird die Editor-Werkzeugleiste eingeblendet, von der aus Sie den Modus wieder beenden können (siehe Abbildung 6.14). Sie können hier auch den Vollbildmodus des Editors deaktivieren, der standardmäßig aktiv ist, sodass Sie auch in der Block-Editor-Ansicht die Admin-Bar und das linke Admin-Menü sehen.



Abbildung 6.14 Im »Ablenkungsfrei«-Modus werden alle gegebenenfalls störenden Elemente des Editors ausgeblendet.

6.1.3 Alle Blöcke von WordPress 6.7

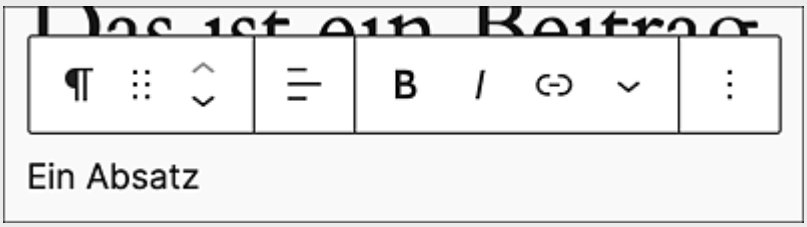
In der folgenden Tabelle sind alle Blöcke mit ihren Einstellungsmöglichkeiten und Optionen aufgelistet, die WordPress 6.7 mitbringt, so wie sie mit dem Theme Twenty Twenty-Four unterstützt werden. Legen Sie sich vielleicht einfach mal eine

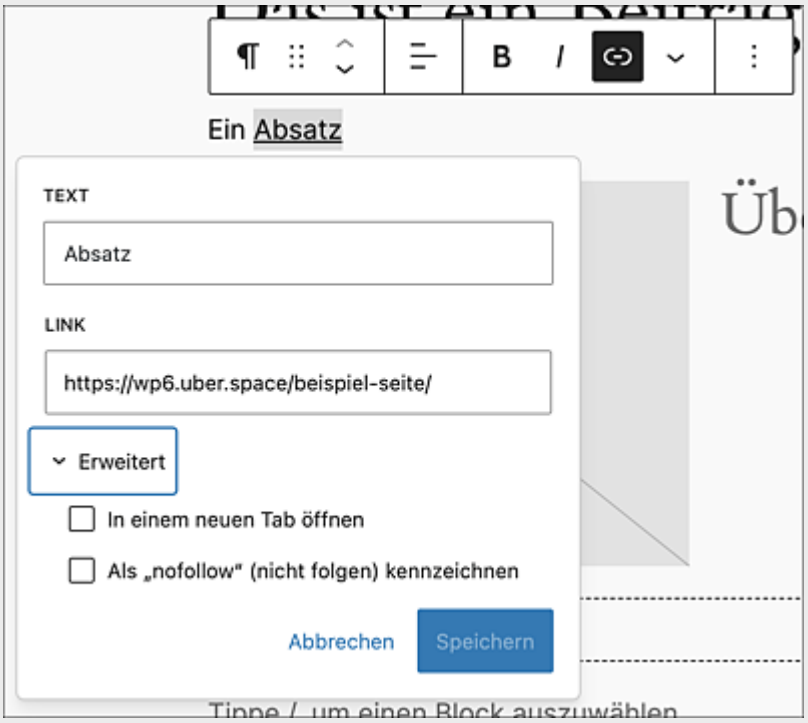
Seite oder einen Beitrag an, und experimentieren Sie ein wenig mit den unterschiedlichen Blöcken und ihren Einstellungen. Und keine Angst: Es sind zwar sehr viele Blöcke, im Alltag wird sich der Pool an Blöcken, die Sie immer wieder benutzen, aber wahrscheinlich auf eine überschaubare Menge beschränken.

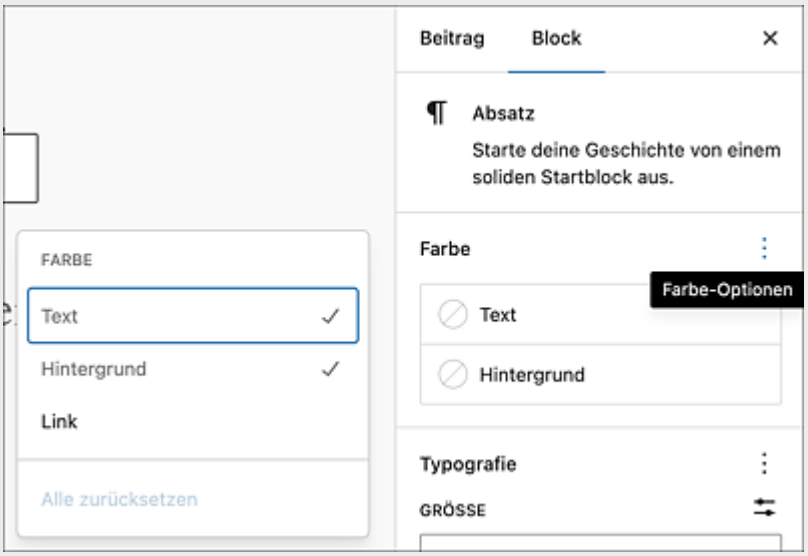
Beachten Sie, dass mit neuen WordPress-Versionen mehr Blöcke dazukommen und bestehende Blöcke verändert werden können. Das heißt, dass die hier beschriebenen Optionen in späteren Versionen eventuell nicht mehr genau so vorhanden sind, anders aussehen et cetera. Außerdem werden die Blöcke, wie kurz angerissen, mit dem Twenty-Twenty-Four-Theme gezeigt, und da Themes über die *theme.json*-Datei einen Einfluss auf den Editor (mehr dazu in Kapitel 23), die Blöcke und die verfügbaren Optionen haben, ist es auch Theme-abhängig, welche Optionen vorhanden sind.

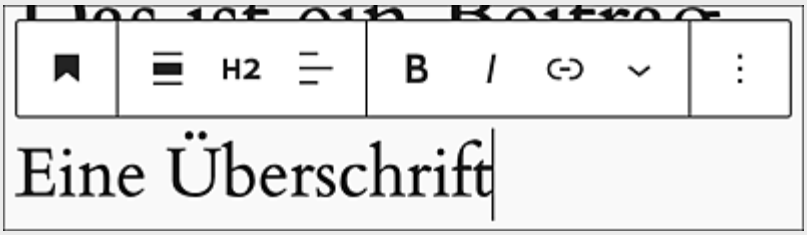
Die Blöcke werden nach den im Block-Inserter angezeigten Kategorien gruppiert.



Text	
Absatz	Der Block für einfachen Fließtext, der Ihnen im leeren Zustand die Möglichkeit für den /-Befehl gibt, um beliebige andere Blöcke einzufügen.  Buttons (die bereits weiter oben beschriebenen Buttons der Werkzeugleiste, die bei jedem Block gleich sind, werden nicht beschrieben): • **Blockname** ermöglicht Ihnen das **Umwandeln in** einen anderen Block, beispielsweise eine Überschrift oder ein Zitat. Sie können auch mehrere ausgewählte Blöcke in beispielsweise eine Liste umwandeln. Welche Optionen hier zur Verfügung stehen, hängt vom Block ab. • **Text ausrichten** bietet Ihnen die Option zur linken, zentrierten oder rechten Ausrichtung des Absatz-Textes. Die Option bezieht sich auf den gesamten Absatz. • **Fett** und **Kursiv** sind Zeichenformate, das bedeutet, sie beziehen sich auf den von Ihnen markierten Text, nicht auf den gesamten Absatz.
Absatz (Forts.)	• **Link** öffnet ein Overlay, in dem Sie eine URL eingeben oder nach einem Inhalt Ihrer WordPress-Installation suchen können, um den aktuell ausgewählten Textabschnitt zu verlinken. Wenn Sie den Link eingefügt

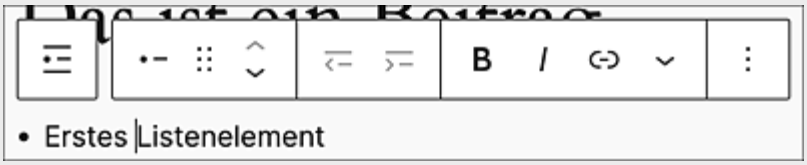
Text	
	haben und den Cursor im verlinkten Text platzieren, wird Ihnen der Link angezeigt sowie die Möglichkeit, den Link zu bearbeiten (siehe Abbildung). Hier können Sie neben dem Linktext und der URL unter **ERWEITERT** auch optional den Link in einem neuen Tab öffnen lassen.  • Bevor Sie das machen, sollten Sie sich aber genau überlegen, ob es in dem Fall notwendig ist, denn zumindest ein Teil Ihrer Besucher*innen hat sicher gerne selbst die Kontrolle darüber, ob ein Link in einem neuen Tab geöffnet wird. Sinnvoll wäre es beispielsweise, wenn durch das Anklicken des Links versehentlich Formularinhalte verloren gingen, die eingetragen wurden, also beispielsweise bei einem Link zur Datenschutzerklärung in einem Formular.

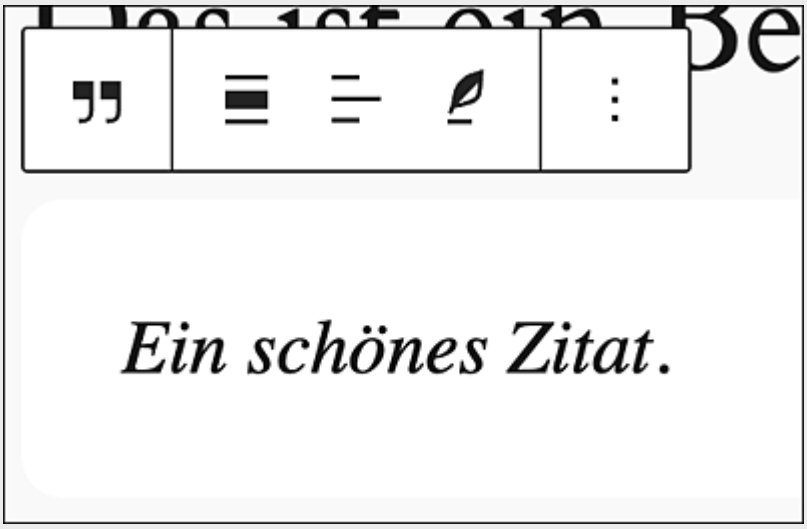
Text	
	• Über den Pfeil rechts neben dem Linkbutton erreichen Sie weitere Funktionen: **Durchgestrichen**, **Fußnote**, **Hervorheben**, **Hochgestellt**, **Inline-Bild**, **Inline-Code**, **Sprache**, **Tastatur-Eingabe** und **Tiefgestellt**.
Absatz (Forts.)	Seitenleiste (Optionen abhängig vom Theme): • **Farbe**: Hier können Sie die Farbe für den Text, den Hintergrund und für Links ändern. Die Möglichkeit für die Linkfarbe können Sie über das Drei-Punkte-Menü in der Panel-Titelzeile einblenden (siehe Abbildung), falls es nicht vorhanden ist.  • **Typografie**: Hier können Sie Dinge wie die Schriftgröße und die Schriftart ändern oder einen Initialbuchstaben aktivieren. Auch dabei lassen sich über das Icon mit den drei Punkten gegebenenfalls weitere Optionen einblenden. • **Größe**: Statt des Drei-Punkte-Menüs haben Sie hier ein Plus-Symbol, über das Sie

Text	
	Optionen für den Innen- und Außenabstand ein- oder ausblenden können. • **Ränder** ermöglicht wie bei **Größe** über das Plus-Symbol das Einblenden von Optionen. Sie können einen Rand (oder Rahmen) definieren und die Ecken abrunden. • **Erweitert**: In das Textfeld **HTML-Anker** können Sie eine Zeichenkette ohne Leerzeichen eingeben, die als Wert für das `id`-Attribut des Absatzes genutzt wird. So könnten Sie einen Sprunglink erstellen, der direkt auf den Absatz verweist (das sieht dann so aus: *example.com/#htmlanker*). Mit dem Feld **Zusätzliche CSS-Klasse(n)** können Sie eigene Klassen angeben, die im `class`-Attribut ausgegeben werden.
Überschrift	Überschrift von `h1` bis `h6`:  • **Ausrichtung**: Hier können Sie die Überschrift über die **Weite** oder **Gesamte Breite** gehen lassen. Die **Weite Breite** ist in der Regel etwas breiter als der normale Inhalt, bei der Option **Gesamte Breite** geht die Überschrift über die gesamte Viewport-Breite. • **Ebene wechseln**: Hier wählen Sie die richtige Überschriftenebene für Ihre Überschrift aus, von **H1** bis **H6**. Sie sollten

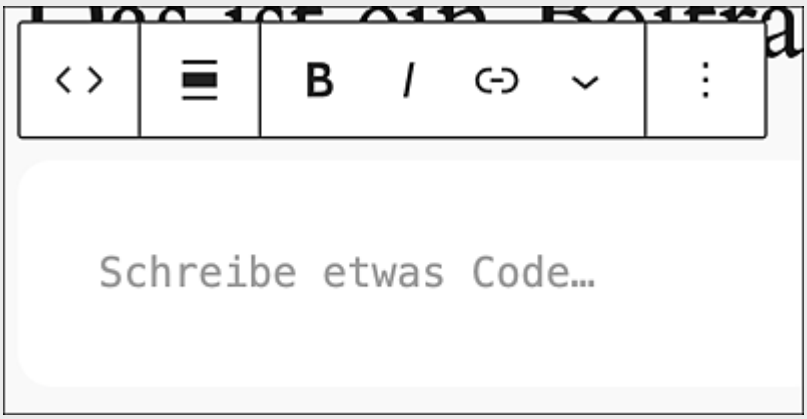
Text	
	nur eine `h1` pro Seite haben, in der Regel wird der Titel der Seite oder des Beitrags als `h1` ausgegeben, weshalb die Standard-Ebene auch `h2` ist. Wichtig außerdem: Wählen Sie die Ebene nicht nach der Schriftgröße aus, sondern logisch – es dürfen keine Ebenen übersprungen werden: So darf auf eine `h2` nur eine andere `h2` oder eine `h3` folgen, aber keine `h4`. • **Text ausrichten**: Ändern Sie die Textausrichtung von links auf zentriert oder rechts. • **Fett**, **Kursiv** und **Link** sowie weitere Optionen über das Kontextmenü hinter dem kleinen Pfeil wurden beim **Absatz** genauer beschrieben. Seitenleiste (Theme-abhängig): • **Stile**: Hier können Sie aus unterschiedlichen Block-Stilen wählen, die teilweise vom Core und teilweise von Themes oder Plugins mitgebracht werden. • **Farbe**, **Typografie**, **Größe**, **Ränder** und **Erweitert** wurden beim **Absatz**-Block beschrieben.
Liste	Eine Liste mit ungeordneten oder geordneten (nummerierten) Listen-Elementen. Wenn Sie einen Listen-Block einfügen, landen Sie direkt im ersten Kind-Block des Listen-Blocks, einem **Listen-Element**-Block (das können Sie gut unten an der Block-Breadcrumb sehen und darüber auch auf die **Liste** wechseln. Ebenfalls einfach vom Listen-Element zur Liste kommen

Text	
	Sie mit einem Klick auf den einzelnen Button ganz links von der Block-Toolbar, der mit etwas Abstand angeordnet ist). Um den Block **Listen-Element** geht es gleich nach der **Liste**.  Buttons: • **Unsortiert** und **Sortiert** ermöglichen das Umschalten zwischen einer ungeordneten Liste (`ul`-HTML-Element) und einer geordneten Liste (`ol`-HTML-Element). • Der Button **Ausrücken** ist nur aktiv, wenn Sie sich in einem verschachtelten **Liste**-Block befinden, also einer eingerückten Liste innerhalb einer anderen Liste. Die Option zur Einrückung gibt es im **Listen-Element**-Block. Seitenleiste (Theme-abhängig): • Mit **Stile** können Sie den Block-Stil wechseln. • Alle mit Twenty Twenty-Four vorhandenen Optionen (**Farbe**, **Typografie**, **Größe**,

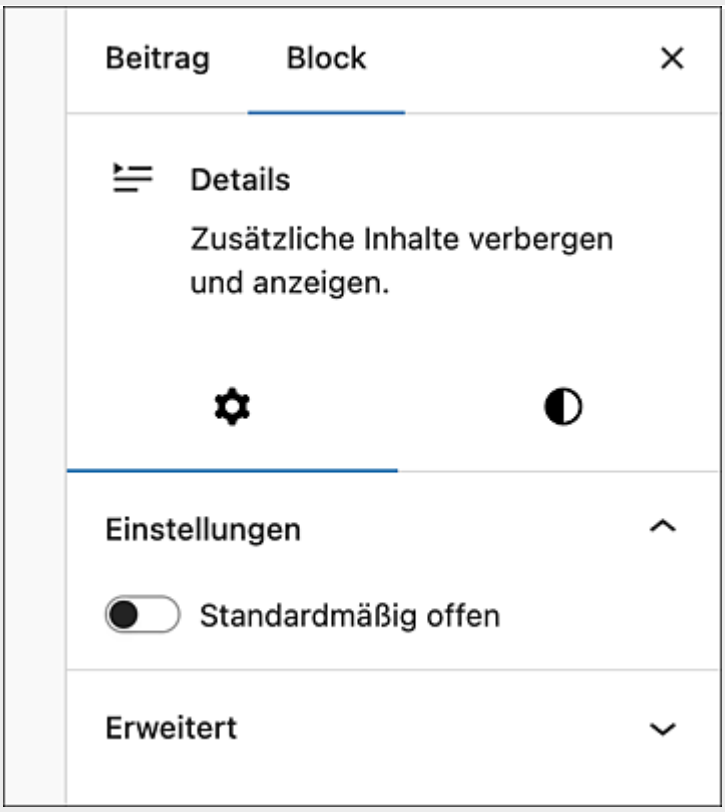
Text	
	Ränder und **Erweitert** wurden beim **Absatz**-Block beschrieben.)
Listen-Element (nur im Listen-Block)	Dieser Block ist nicht im Block-Inserter auswählbar, da er nur als Kind-Block eines Listen-Blocks funktioniert. Wenn Sie eine Liste einfügen, landen Sie direkt im Listen-Element-Block.  Buttons: • **Übergeordneten Block auswählen**: Dieser Button sitzt links neben der Block-Werkzeugleiste etwas abgesetzt und ermöglicht das einfache Auswählen des übergeordneten Blocks. Im Fall eines Listen-Element-Blocks ist das die umgebende Liste. • **Ausrücken** und **Einrücken** zum Ändern der Ebene des Listen-Elements (je nach Position in der Liste unterscheidet sich, welche der beiden Buttons aktiv sind, falls überhaupt – das erste Listen-Element einer Liste kann weder eingerückt noch ausgerückt werden). • **Fett**, **Kursiv**, **Link** und die Optionen des Pfeil-Menüs wurden beim **Absatz**-Block erklärt. Standard: **Typografie**, **Größe**, **Ränder** und **Erweitert** sind ebenfalls bereits vom **Absatz**-Block bekannt.

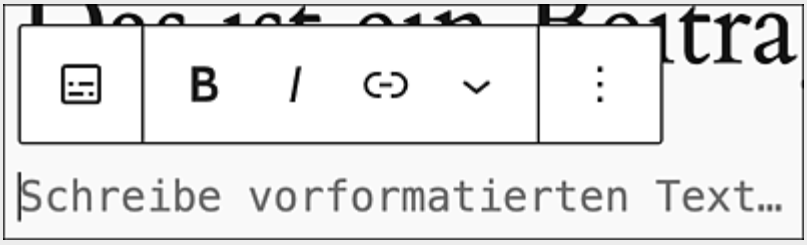
Text	
Zitat	Ein Block, der als Wrapper für alle möglichen anderen Blöcke fungiert, nicht wie eben der Listen-Block nur für Listen-Elemente. Die Inhalte werden in ein `blockquote`-Element eingefügt, wodurch nicht nur visuell, sondern auch für Maschinen (beispielsweise Screen-Reader für Personen mit Sehbehinderung) verständlich ist, dass es sich um ein Zitat handelt.
Zitat (Forts.)	Auch hier landen Sie wieder direkt in einem Kind-Block, wie bei der Liste (beim Zitat ist es ein Absatz) – wechseln Sie über den Button neben der Block-Werkzeugleiste oder die Block-Breadcrumb zum **Zitat**-Block.  Buttons: • **Ausrichtung**: Die Optionen zur Änderung der Breite des Blocks sind bereits bekannt, zusätzlich können Sie den gesamten Block links oder rechts ausrichten. • **Text ausrichten**: bereits bekannt

Text	
	• **Quellenangabe hinzufügen**: Fügt ein Textfeld ganz unten im Zitat-Block ein, in dem Sie angeben können, von wem das Zitat ist. Bei der Quellenangabe haben Sie die schon vorgestellten Möglichkeiten zur Textauszeichnung wie Fettung, Kursivschrift, Verlinkung und Durchstreichung. Seitenleiste (Theme-abhängig): • Sie können je nach Theme verschiedene **Stile** wählen. • **Farbe**, **Typografie**, **Ränder** und **Erweitert** sind vom **Absatz**-Block bekannt, bei **Größe** gibt es beim Zitat die Möglichkeit, die **Block-Abstände** anzupassen. Das ist der Abstand zwischen den Blöcken, die im Zitat-Block eingefügt werden. Außerdem können Sie ein **Hintergrundbild** definieren.

Text	
Code	Zur Darstellung von Code in einer Monospace-Schrift, so wie Sie es hier im Buch an vielen Stellen sehen. Der Inhalt steht in einem `code`-Element, das von `pre` umschlossen ist.  Buttons: • Sie können die **Ausrichtung** des Blocks zwischen normaler und weiter Breite ändern. • Darüber hinaus gibt es wieder die schon vorgestellten Zeichenformat-Möglichkeiten, um Text etwa fett oder kursiv darzustellen. Seitenleiste: • **Farbe**, **Typografie**, **Größe**, **Ränder** und **Erweitert** wurden beim **Absatz**-Block näher vorgestellt.

Text	
Details	Dieser Block ermöglicht Ihnen das Anlegen von Inhalt, der erst bei Klick auf einen Titel/eine Zusammenfassung angezeigt wird und initial ausgeblendet ist. Genutzt wird dafür das HTML-`details`-Element im Zusammenspiel mit dem `summary`-Element. Der Block selbst hat wenig Optionen, im Inhaltsbereich des Blocks können Sie wie beispielsweise beim Zitat-Block beliebige andere Blöcke einfügen und natürlich deren Optionen nutzen.  Button: **Ausrichtung**: Legen Sie fest, wie breit der Block sein soll.
Details (Forts.)	Seitenleiste: Bei diesem Block sehen Sie zum ersten Mal das Pattern, dass die Block-Seitenleiste noch mal in zwei Bereiche unterteilt ist, nämlich in einen **Einstellungen**-Tab mit dem Zahnrad und einen **Stile**-Tab mit dem Schwarz-weiß-Icon (siehe Abbildung).

Text	
	 • **Einstellungen** • **Standardmäßig öffnen** erlaubt Ihnen, den Inhalt direkt anzuzeigen statt initial zu verstecken. • Im **Erweitert**-Panel können Sie zusätzliche CSS-Klassen für das Block-Element übergeben. • Im **Stile**-Tab gibt es nichts, was nicht bereits bei anderen Blöcken beschrieben wurde.

Text	
Vorformatiert	Stellt Ihren Text genau so dar, wie Sie ihn eingeben, inklusive Leerzeichen am Anfang oder mehreren hintereinander, die normalerweise zu einem Leerzeichen schrumpfen würden. Als HTML-Element kommt `pre` zum Einsatz, wie beim Code-Block (nur ohne das `code`-Element darin), deshalb wird zur Darstellung auch wieder ein Monospace-Font genutzt.  Buttons: Die bekannten Blöcke zur Formatierung von markiertem Text sind auch bei diesem Block am Start: **Fett**, **Kursiv**, **Link**, **Durchgestrichen** und so weiter.
Vorformatiert (Forts.)	Seitenleiste: Ebenfalls bereits bekannt sind die Optionen in der Seitenleiste für **Farbe**, **Typografie**, **Größe**, **Ränder** und der **Erweitert**-Bereich.

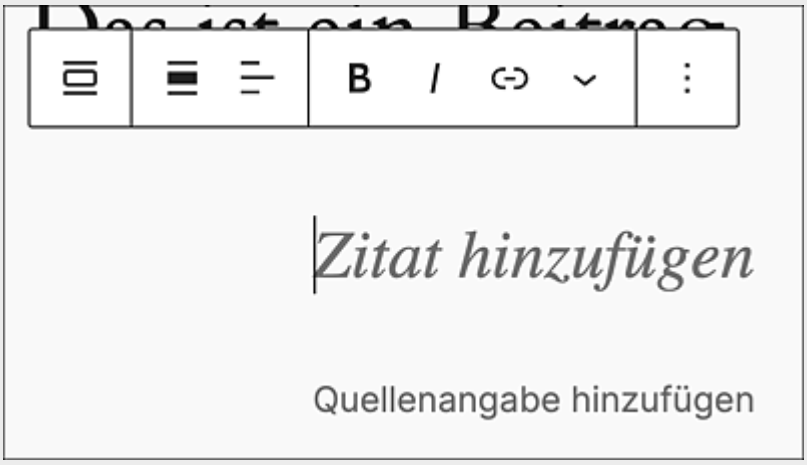
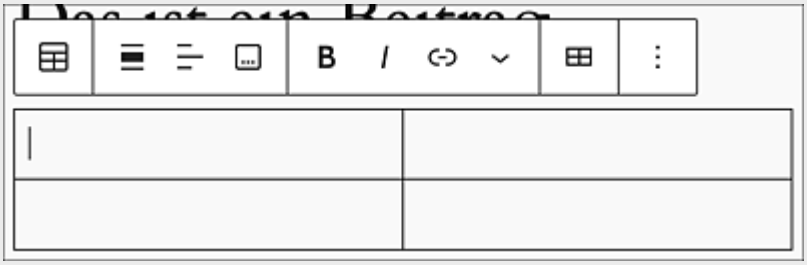
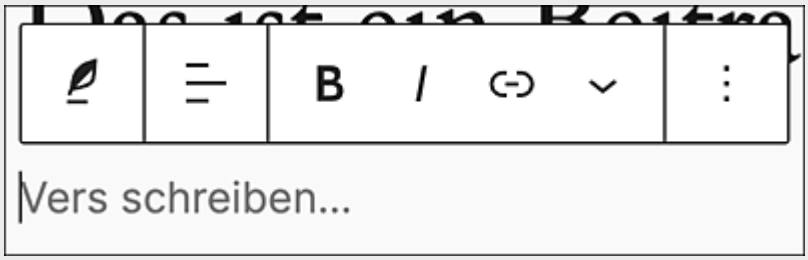
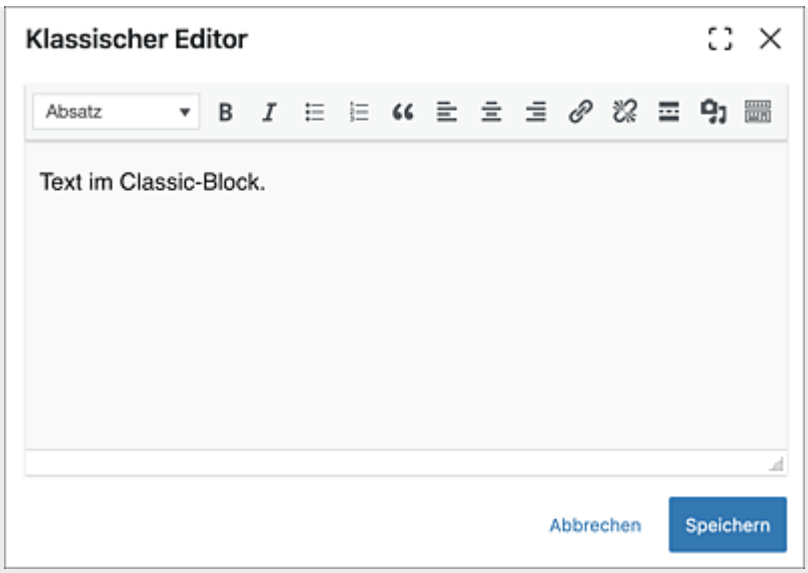
Text	
Pullquote	Dieser Block wird wie der Zitat-Block für die Auszeichnung von Zitaten benutzt, ist aber auffälliger gestaltet. Anders als der **Zitat**-Block ist dieser Block kein Wrapper für andere Blöcke.  Buttons: • **Ausrichtung**: die bekannten Optionen zur Änderung der Block-Breite und Ausrichtung des Blocks links oder rechts • **Text ausrichten** und die weiteren Text-Formatierungsoptionen sind bereits aus den vorherigen Blöcken bekannt und beim **Absatz**-Block näher beschrieben. Seitenleiste: Hier gibt es nichts Neues zu sehen: **Farbe**, **Hintergrundbild**, **Typografie**, **Größe**, **Ränder** und **Erweitert** wurden bereits behandelt.

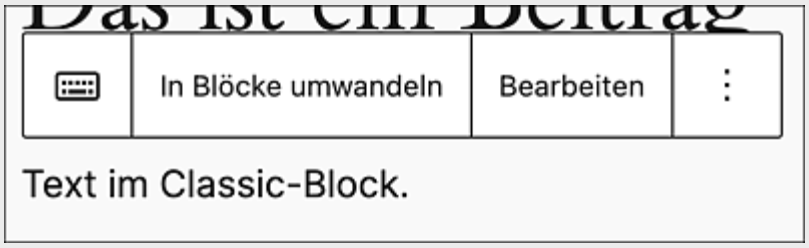
Tabelle	Ihr Block, wenn es um die Darstellung tabellarischer Daten geht. Beim Erstellen werden Sie zunächst gefragt, wie viele Spalten und Zeilen die Tabelle haben soll. 
Tabelle (Forts.)	Buttons: Auch hier treffen Sie hauptsächlich auf Buttons, die Ihnen schon bekannt vorkommen. Zwei Ausnahmen: Über den Button rechts neben der Textausrichtung können Sie eine **Beschriftung hinzufügen**, und über den Button links neben dem Drei-Punkte-Menü können Sie die **Tabelle bearbeiten**. Hier können Sie neue Spalten und Zeilen einfügen und Spalten oder Zeilen löschen. Seitenleiste: • Im **Einstellungen**-Tab können Sie einstellen, ob die **Tabellenzellen mit fester Breite** ausgegeben werden sollen und ob die Tabelle eine **Kopfzeile** und/oder eine **Fußzeile** haben soll. Im **Erweitert**-Panel finden Sie die bekannten Optionen zum Setzen eines HTML-Ankers und zusätzlicher CSS-Klassen. • Im **Stile**-Tab finden Sie wieder Altbekanntes: alternative Block-**Stile** und Optionen für **Farbe**, **Typografie**, **Größe** und **Ränder**.

Vers	Der Vers-Block ist quasi der Vorformatiert-Block ohne Monospace-Font. Sie können also wieder mit Leerzeichen arbeiten, um Textzeilen einzurücken, und die Einrückungen werden genau so im Frontend ausgegeben.  Buttons: Bei Buttons nichts Neues, Sie können die Textausrichtung ändern sowie die Zeichenformate anwenden, die Sie ganz am Anfang beim **Absatz** kennengelernt haben. Seitenleiste: Auch hier gibt es keine noch nicht gesehenen Optionen. Sie können Einstellungen für **Farbe**, **Typografie**, **Größe** und **Ränder** vornehmen sowie ein **Hintergrundbild** festlegen und haben unter **Erweitert** die Möglichkeit für **HTML-Anker** und **Zusätzliche CSS-Klasse(n)**.
Classic	Wenn Sie schon mit WordPress gearbeitet haben, bevor der Block-Editor eingefügt wurde, dann kennen Sie den TinyMCE-Editor, der vorher für die Erstellung von Inhalten genutzt wurde. Und genau diesen Editor können Sie mit dem Classic-Block wieder zum Leben erwecken.



Dieser Block kommt auch zum Einsatz, wenn von einer TinyMCE-Editor-Version auf den Block-Editor gewechselt wird, damit die alten Inhalte nicht einfach so in Block-Markup umgewandelt werden, sondern das alte Markup erhalten bleibt.

Wenn Sie den Block einfügen, öffnet sich ein Overlay mit dem TinyMCE-Feld. Wenn Sie auf **Speichern** klicken, sehen Sie den Inhalt ganz normal im Block-Editor.



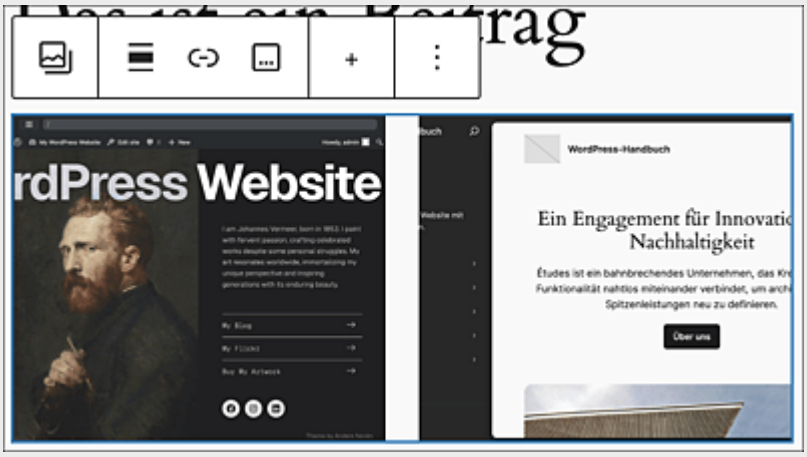
Buttons:

- **In Blöcke umwandeln**: Wandelt den Inhalt des Classic-Blocks in die passenden anderen Blöcke um, also beispielsweise Absätze in Absatz-Blöcke und Listen in Listen + Listen-Element-Blöcke.
- **Bearbeiten** öffnet wieder das Overlay mit dem TinyMCE-Editor.

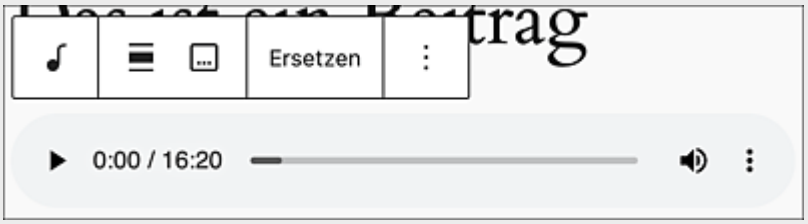
	Seitenleiste: Hier gibt es keine Optionen.
Medien	
Bild	Ein Bild, nicht mehr und nicht weniger. Nach dem Einfügen des Buttons werden Sie mit einem Dialog begrüßt, der Ihnen über die Buttons **Hochladen**, **Mediathek** und **Von URL einfügen** drei Wege anbietet, um ein Bild auszuwählen. Sie können auch per Drag-and-drop ein Bild auf den Block ziehen, um es hochzuladen.  Buttons: • **Ausrichtung**: bereits bekannt, für die Ausrichtung des Blocks • **Duoton-Filter anwenden** ermöglicht Ihnen, einen Farbeffekt auf das Bild anzuwenden, bei dem das

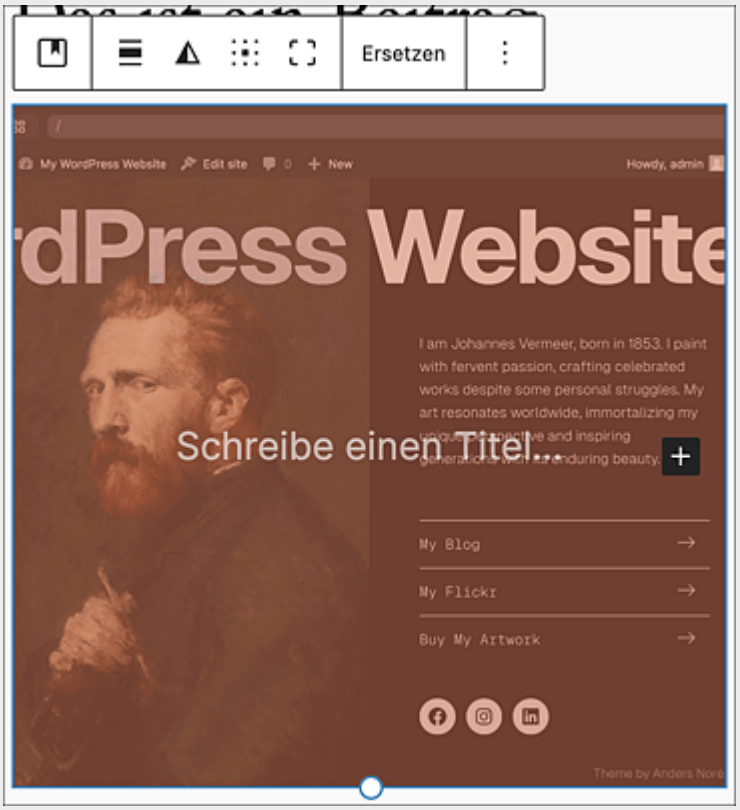
	Bild auf zwei Farbtöne reduziert wird, die Sie frei auswählen können. • **LINK** erlaubt Ihnen, das Bild entweder auf eine beliebige URL, auf die Bilddatei oder die Anhangseite zu verlinken. Außerdem gibt es die Option, eine größere Version des Bilds in einer Lightbox zu öffnen. • **ZUSCHNEIDEN**: Blendet drei weitere Buttons ein, über die Sie den gewünschten Ausschnitt des Bilds festlegen können. Mit **ÜBERNEHMEN** wird die Änderung am Bild gespeichert, mit **ABBRECHEN** lassen Sie das Bild, wie es ist.

Bild (Forts.)	• **Text über dem Bild einfügen**: Damit wird der Bild-Block in einen Cover-Block umgewandelt. • **Beschriftung hinzufügen**: Mit diesem Button wird ein Textfeld aktiviert, in dem Sie eine Bildbeschriftung eingeben können. • **Ersetzen**: Bietet Ihnen die Optionen **Mediathek öffnen**, um ein Bild aus der Mediathek auszuwählen, oder **Hochladen**, um ein neues Bild hochzuladen, mit dem das aktuelle Bild ersetzt wird. Über den Button **Zurücksetzen** können Sie den Block wieder in den Zustand ohne Bild versetzen. Seitenleiste: • **Einstellungen** • **Alternativtext**: Hier können (und sollten) Sie eine Textalternative für das Bild eingeben, das unter anderem auch Nutzer*innen mit Sehbehinderung ermöglicht, den Inhalt des Bilds zu erfassen. Ein netter Nebeneffekt ist, dass der Alternativtext auch für die Suchmaschinenoptimierung vorteilhaft ist. • **Einstellungen** • **Seitenverhältnis**, **Breite**, **Höhe** und **Größe**: Erlaubt Ihnen, die Maße des Bilds anzupassen. • Unter **Erweitert** finden Sie die Optionen für den HTML-Anker und die zusätzlichen CSS-Klassen sowie ein Feld, mit dem Sie das `title`-Attribut setzen können. • Im **Stile**-Tab gibt es keine Optionen, die Sie noch nicht gesehen haben, abgesehen von der Möglichkeit, auch hier den **Duoton**-Effekt zu definieren.

Galerie	Ein Wrapper-Block für mehrere **Bild**-Blöcke. Ähnlich wie beim Bild-Block gibt es beim Einfügen des Blocks erst mal einen Platzhalter, der Ihnen die Möglichkeit gibt, Bilder hochzuladen oder aus der Mediathek einzufügen. 
Galerie (Forts.)	Buttons: • **Ausrichtung**: Diesen Button kennen Sie schon, er ermöglicht die Änderung der Block-Ausrichtung. • **Link**: Ermöglicht die Anpassung des Linkziels für alle Bilder innerhalb des Galerie-Blocks. Sie können den Wert für einzelne Bilder überschreiben, indem Sie die Option beim entsprechenden Bild-Block nutzen. • **Beschriftung einfügen** haben Sie eben beim Bild-Block bereits gesehen. Hier können Sie für eine ganze Galerie eine Beschriftung anlegen. • Plus-Icon: Öffnet ein Dropdown mit den Optionen **Mediathek öffnen** und **Hochladen**, um neue Bilder zur Mediathek hinzuzufügen. Seitenleiste: • **Einstellungen** • **Spalten**: Hier können Sie festlegen, in wie viele Spalten die Bilder aufgeteilt werden sollen.

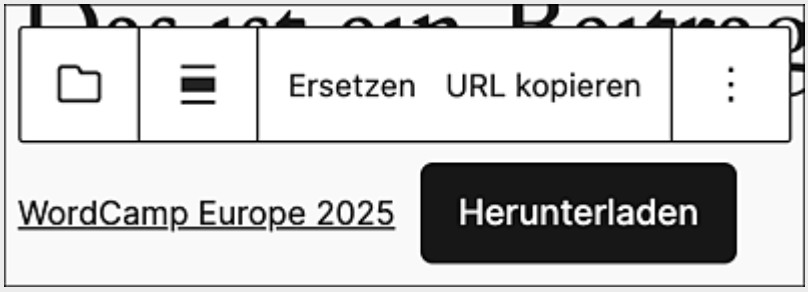
- **Einstellungen** • **Auflösung**: Erlaubt Ihnen, die Auflösung für alle Bilder der Galerie zu ändern.
- **Einstellungen** • **Bilder passend zuschneiden**: Sorgt dafür, dass Bilder in einer Zeile auch bei unterschiedlichen Formaten mit derselben Höhe angezeigt werden.
- **Einstellungen** • **Reihenfolge zufällig festlegen**: Hiermit können Sie einstellen, dass beim Anzeigen der Galerie im Frontend jedes Mal die Reihenfolge der Bilder zufällig bestimmt wird.
- **Erweitert** mit den zwei bekannten Optionen **HTML-Anker** und **Zusätzliche CSS-Klasse(n)**
- Im **Stile**-Tab gibt es Optionen für **Farbe**, **Größe** und **Ränder**, die Sie bereits kennengelernt haben.

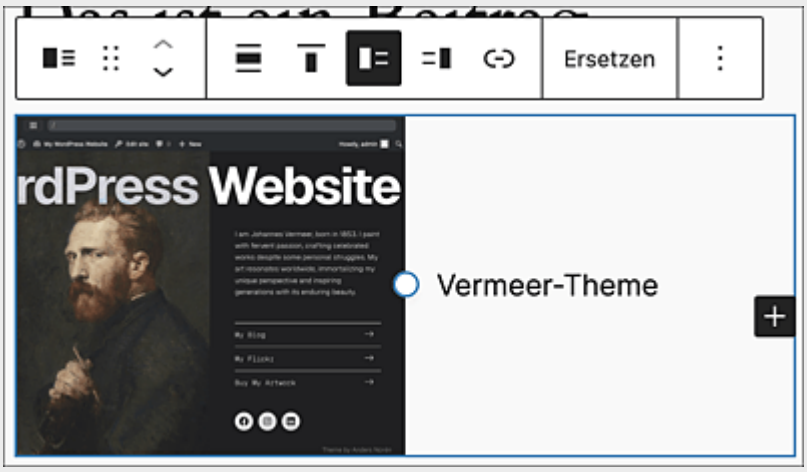
Audio	Ein Block zum Einbinden einer Audio-Datei, sodass sie im Browser abgespielt werden kann. Wenn Sie den Block einfügen, bekommen Sie die Möglichkeit, eine Audiodatei hochzuladen, aus der Mediathek auszuwählen oder von einer URL aus einzufügen.  Buttons: • **Ausrichtung** zur Änderung der Block-Ausrichtung • Wie beim Bild- und Galerie-Block können Sie auch für Audio-Blöcke eine **Beschriftung hinzufügen**. • Über **Ersetzen** tauschen Sie die Audio-Datei aus. Seitenleiste: • **Einstellungen** • **Automatische Wiedergabe** und **Schleife**: Steuern Sie, ob die Audio-Datei automatisch abgespielt werden sollte (bitte nicht) und ob sie immer wieder von vorne starten soll. • **Einstellungen** • **Vorladen**: Legen Sie fest, ob und wie genau die Datei bereits vorgeladen werden soll, bevor sie abgespielt wird. • **Erweitert** bietet die bekannten Optionen. • Im **Stile**-Tab können Sie Innen- und Außenabstand steuern.
Cover	Dieser Block ist dafür da, um Text auf einem Bild darzustellen. Wenn Sie den Block einfügen, werden Sie wieder von einem Dialog begrüßt, um ein Bild festzulegen. Sie können auch dynamisch das als Beitragsbild festgelegte Bild anzeigen.

Cover (Forts.)	Der Cover-Block ist ein Wrapper für andere Blöcke, sodass Sie auf dem Bild alle möglichen anderen Blöcke verwenden können.  Buttons: • **Ausrichtung**: Sie wissen Bescheid: weite und volle Breite, links- und rechtsbündig sowie zentriert. • **Duoton** kennen Sie bereits vom Bild-Block. • **Inhaltsposition ändern**: Standardmäßig wird der Inhalt zentriert über dem Bild angezeigt. Hier haben Sie ein Steuerungselement, mit dem Sie die Position anpassen können. • **Auf volle Höhe umschalten** ermöglicht Ihnen, den Block so hoch wie den Browser-Viewport anzuzeigen. • **Dateien hinzufügen/Ersetzen** öffnet ein Dropdown zum Austauschen des Bilds.

	Seitenleiste: Bei diesem Block besteht der **Einstellungen**-Tab aus den Panels **Layout**, **Einstellungen** und **Erweitert**. • **Layout** • **Die inneren Blöcke verwenden die Breite des Inhalts**: Die im Cover-Block eingefügten Blöcke orientieren sich an der normalen Breitenangabe für Inhalte, die vom Theme vorgesehen ist. Ist die Option aktiviert, können Sie optional noch angepasste Breiten für **Inhalt** (die normale Breite) und **Weite Breite** angeben.

Cover (Forts.)	• **Einstellungen** • **Hintergrund fixiert**: Ist diese Option aktiv, bleibt das Bild beim Scrollen an einer Position, der Inhalt scrollt also quasi über das Bild hinüber. • **Einstellungen** • **Hintergrund wiederholen**: Sie können einstellen, dass das Hintergrundbild wiederholt wird, wenn Sie beispielsweise ein Bild haben, das bei Wiederholung ein Muster ergibt. • **Einstellungen** • **Fokuspunkt**: Wählen Sie den wichtigsten Punkt des Bilds, sodass dieser immer zu sehen ist. • **Einstellungen** • **Alternativtext**: Das haben Sie schon beim Bild-Block gesehen. Unscheinbarer Name, wichtige Funktion, damit alle etwas von den Bildern haben, auch wenn sie sie nicht sehen können. • Unter **Erweitert** haben Sie neben den bekannten Optionen **HTML-Anker** und **Zusätzliche CSS-Klasse(n)** die Möglichkeit, das **HTML-Element** des Blocks zu ändern. Standardmäßig ist es ein `div`, aber wenn Sie beispielsweise entscheiden, dass der gesamte Inhalt Ihrer Website in einem Cover-Block stehen soll, wäre `main` das richtige Element (solche grundlegenderen Strukturentscheidungen des Markups werden meist vom Theme getroffen). • **Stile** erlaubt Ihnen, neben bereits bekannten Optionen die Farbe und Deckkraft des Overlays anzupassen und eine **Minimale Höhe** einzustellen sowie einen Schatten zu aktivieren.

Datei	Dieser Block stellt den Link zu einer beliebigen von Ihnen gewählten Datei dar, optional mit einem Download-Button. 
Datei (Forts.)	Buttons: • **Ausrichtung** und **Ersetzen** sind alte Bekannte. • **URL kopieren** kopiert die URL zur gewählten Datei in die Zwischenablage. Seitenleiste: • **Einstellungen** • **Link zu** ist vom Bild-Block bekannt. • **Einstellungen** • **In einem neuen Tab öffnen**: Diese Option sollten Sie nur nutzen, wenn die Datei zum Beispiel auf einer Seite mit einem Formular vorkommt, es also sein kann, dass durch das Wechseln auf eine neue Seite eingegebene Formulardaten verloren gehen. • **Einstellungen** • **Download-Button anzeigen**: Tut genau, was die Beschriftung vermuten lässt. • Unter **Erweitert** haben Sie die schon bekannten Optionen zum Setzen einer `id` und zusätzlicher HTML-Klassen. • Im **Stile**-Tab gibt es Einstellungen zu **Farbe**, **Größe** und **Rändern**, die Sie ebenfalls bereits kennen.

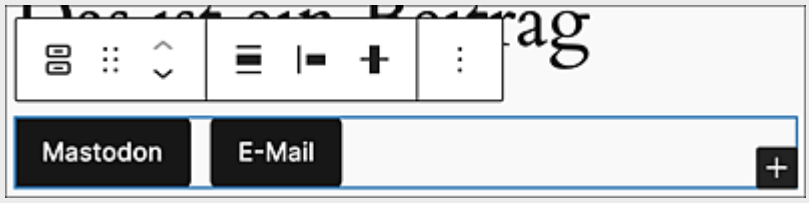
Medien und Text	Dieser Block stellt ein Bild oder Video neben einem Textbereich dar. Der Textbereich ist wieder ein Wrapper für beliebige Blöcke, theoretisch könnten Sie hier auch Bilder oder Videos einfügen.  Buttons: • **Ausrichtung** steuert die horizontale Ausrichtung und ist bekannt.

Medien und Text (Forts.)	• **Vertikale Ausrichtung ändern** erlaubt Ihnen, das Medium auf der einen Seite in bestimmter Weise zum Textbereich auf der anderen Seite zu positionieren, beispielsweise zentriert, sodass der **niedrigere** Teil immer in der Mitte des höheren Teils ausgerichtet ist. • **Medien links anzeigen**/**Medien rechts anzeigen** erlaubt Ihnen, die Reihenfolge der beiden Spalten zu wechseln. • **Link** ermöglicht die Verlinkung des Bilds. • **Ersetzen** ist zum Austausch des Mediums da. Seitenleiste: • **Einstellungen** • **Medienbreite**: Hier können Sie festlegen, welche prozentuale Breite das Medium einnehmen soll. • **Einstellungen** • **Auf Mobilgeräten stapeln**: Gibt an, ob die beiden Spalten auf kleineren Viewports untereinander angezeigt werden sollen, was wegen des verringerten Platzangebots meist sinnvoll ist. • **Erweitert** und **Stile**: nur bereits Bekanntes

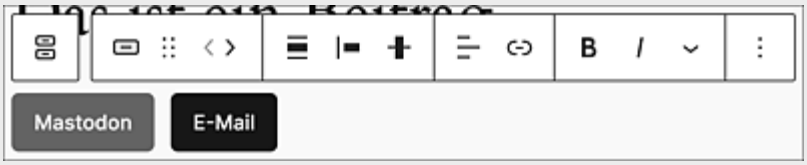
Video	Mit diesem Block können Sie ein Video aus der Mediathek oder von einer URL in eine Seite oder einen Beitrag einfügen. Nach dem Einfügen des Blocks bekommen Sie die Optionen, ein Video hochzuladen, aus der Mediathek auszuwählen oder eine Video-URL anzugeben.  Buttons: • **Ausrichtung** (bekannt von anderen Blöcken)

Video (Forts.)	• **Beschriftung hinzufügen** ermöglicht Ihnen, unter dem Video eine Beschriftung anzuzeigen. • Über **Texthinweise** können Sie Tracks angeben, die beispielsweise Untertitel enthalten können. • **Ersetzen** gibt Ihnen die Möglichkeit, ein anderes Video auszuwählen. Seitenleiste: • Im **Einstellungen**-Tab finden Sie unterschiedliche Optionen zur Anpassung des Video-Verhaltens: Sie können die **Automatische Wiedergabe** aktivieren (das funktioniert in einigen Browsern nur, wenn das Video **Stummgeschaltet** ist – generell sollten Sie vermeiden, diese Option zu nutzen), das Video in einer **Schleife** abspielen, den Besucher*innen die Buttons zur **Wiedergabe-Steuerung** anzeigen oder das Bild **Inline abspielen**. • Darüber hinaus können Sie wie beim **Audio**-Block einstellen, wie das Video **Vorladen** soll, und ein **Vorschaubild** festlegen, das vor dem Abspielen des Videos angezeigt wird. • Der **Erweitert**-Bereich ist von den anderen Blöcken bekannt. • Im **Stile**-Tab gibt es die bereits bekannten Optionen für den **Innen**- und **Außenabstand**.

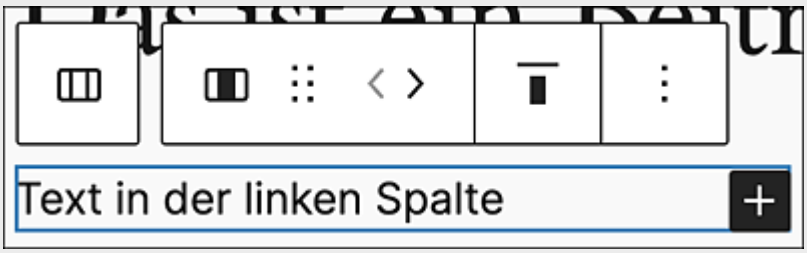
Design

Buttons	Ein Wrapper-Block für einen oder mehrere »Button«-Blöcke. Wie bei anderen Wrapper-Blöcken wie dem Listen- oder Zitat-Block landen Sie beim Einfügen direkt im ersten Kind-Block. Wechseln Sie für die Optionen des **Buttons**-Blocks eine Ebene nach oben. 

Buttons (Forts.)	Buttons: • Sie haben die Möglichkeit, die **Ausrichtung** des Blocks zu verändern, wie Sie es bereits von vielen der anderen Blöcke kennen. • Über **Ausrichtung der Elemente ändern** können Sie einstellen, wie die einzelnen Button-Blöcke innerhalb des Buttons-Blocks ausgerichtet sein sollen. • Über **Vertikale Ausrichtung ändern** können Sie die vertikale Ausrichtung der Buttons zueinander einstellen. Seitenleiste: • Im Tab zu den Einstellungen gibt es **Layout**-Optionen, die Sie teilweise auch in der Werkzeugleiste finden (**Anordnung**). Daneben gibt es die Option, die **Richtung** der Blöcke von horizontal nach vertikal zu ändern, sodass sie untereinander angezeigt werden. • Mit **Zeilenumbruch auf mehrere Zeilen erlauben** stellen Sie ein, ob die Buttons in mehrere Zeilen umbrechen dürfen, wenn der Platz nicht ausreicht, alle nebeneinander anzuzeigen. • **Erweitert** hat nur bereits Bekanntes im Angebot. • Auch **Stile** bietet mit den Optionen für **Farbe**, **Typografie**, **Größe** und **Ränder** nur bereits bekannte Optionen.

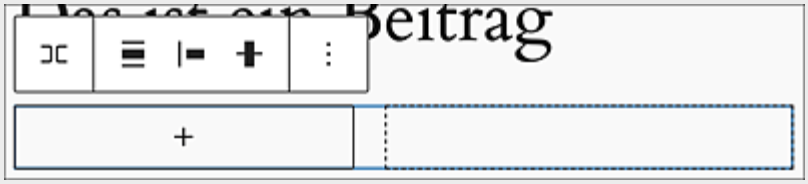
Button (nur im Buttons-Block)	Dieser Block kann nicht über den Block-Inserter eingefügt werden, sondern ist Teil des **Buttons**-Blocks und repräsentiert einen einzelnen Button.  Buttons: • Auch beim Button können Sie auf die Ausrichtungsoptionen des **Buttons**-Blocks zugreifen.
Button (nur im Buttons-Block) (Forts.)	• **Text ausrichten**, **Link** und die anderen Textformate wie **Fett** und **Kursiv** sind bereits beim **Absatz**-Block behandelt worden. Seitenleiste: • In den **Einstellungen** können Sie für den Button eine von vier prozentualen Breiten-Angaben auswählen, wenn er nicht automatisch die zum Inhalt passende Breite haben soll. • Die Optionen unter **Erweitert** sind bereits bekannt, genauso wie die **Stile**-Optionen für die Auswahl unterschiedlicher Stile sowie der Anpassung von Optionen für **Farbe**, **Typografie**, **Größe** sowie **Rand und Schatten**.

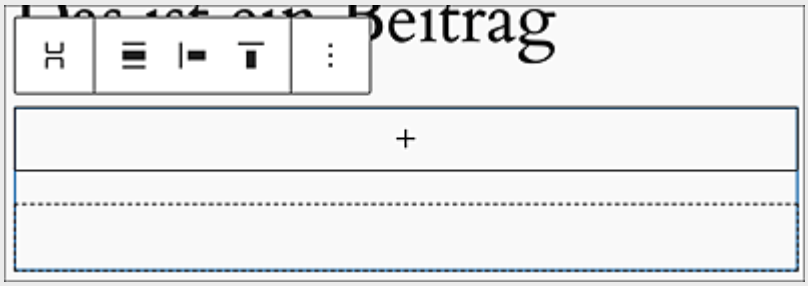
Spalten	Erlaubt Ihnen, Inhalte in mehreren Spalten nebeneinander darzustellen. Beim Einfügen können Sie optional aus vordefinierten Layouts wählen, wie etwa drei gleich breite Spalten oder zwei Spalten, von denen eine ein Drittel und die andere zwei Drittel der Breite einnimmt. Wie etwa der **Buttons**-Block ist der **Spalten**-Block ein Wrapper für Blöcke vom Typ **Spalte**.  Buttons: • **Ausrichtung** zur Steuerung der Block-Ausrichtung • **Vertikale Ausrichtung ändern** zur Anpassung der vertikalen Ausrichtung der Spalten zueinander Seitenleiste: • **Einstellungen** • **Spalten** ermöglicht Ihnen, nachträglich die Anzahl der Spalten zu ändern. • **Einstellungen** • **Auf Mobilgeräten stapeln** gibt an, ob die Spalten bei kleinen Viewports untereinander angezeigt werden sollen. • **Erweitert** hält mal wieder die Optionen für **HTML-Anker** und **Zusätzliche CSS-Klasse(n)** bereit.
Spalten (Forts.)	• **Stile** beinhaltet bereits bekannte Optionen für **Farbe**, **Typografie**, **Größe** sowie **Rand und Schatten**.

Spalte (nur im Spalten-Block)	Ein Block, der sich nicht direkt über den Inserter einfügen lässt, sondern ein direktes Kind-Element eines Spalten-Blocks ist. Der Block ist ein Wrapper für beliebige weitere Blöcke, um die Spalte zu füllen.  Buttons: Sie können die **Vertikale Ausrichtung ändern**, um die Spalte im Verhältnis zu den anderen Spalten anders auszurichten. Seitenleiste: • Die **Layout**-Option **Die inneren Blöcke verwenden die Breite des Inhalts** haben Sie bereits beim **Cover**-Block gesehen. Sie bedeutet, dass die Blöcke innerhalb der Spalte die Maximal-Werte für die normale und gegebenenfalls weite Breite nutzen, die vom Theme definiert ist. Optional können Sie eigene Werte vergeben, wenn die Option aktiv ist. • **Einstellungen** • **Breite** ermöglicht Ihnen, die Breite der Spalte festzulegen. • **Erweitert** hält auch hier keine Überraschungen bereit. • Im **Stile**-Tab können Sie schon bekannte Optionen für **Farbe**, **Typografie**, **Größe** und **Ränder** verwenden. Bei **Größe** haben Sie unter anderem die Option, über **Block-Abstände** den Abstand zwischen den Kind-Blöcken einzustellen.

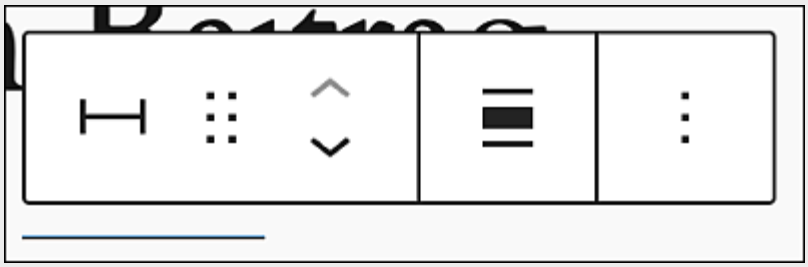
Gruppe	Ein Wrapper-Block ohne besondere Darstellungsweisen wie Spalten- oder Cover-Blöcke. Einfach ein Block zur Gruppierung mehrerer Blöcke, um sie beispielsweise in einem Bereich mit anderer Breite oder Hintergrundfarbe darzustellen.  Buttons: Hier finden Sie die Optionen zur Änderung der Block-**Ausrichtung** sowie der Ausrichtung der Elemente innerhalb der Gruppe. Seitenleiste: • Direkt unter der Block-Beschreibung haben Sie die Möglichkeit, die Gruppe in die Blöcke (oder genauer: Variationen des Gruppen-Blocks) **Zeile**, **Stapel** oder **Raster** umzuwandeln, um die inneren Blöcke zum Beispiel alle nebeneinander anzuzeigen. Falls Sie sich ein bisschen mit CSS auskennen: Die Variationen **Zeile** und **Stapel** sind Block-gewordene Flex-Container, und die Raster-Variation ist ein CSS-Grid. • Im **Layout**-Panel gibt es wieder die Option **Die inneren Blöcke verwenden die Breite des Inhalts**, mit der Sie angeben können, ob die Blöcke innerhalb der Gruppe sich auch ganz normal an die vom Theme angegebenen Breiten halten sollen. Andernfalls füllen sie die Breite der Gruppe

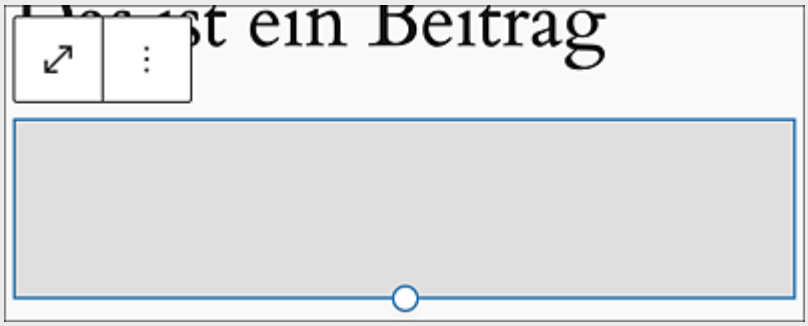
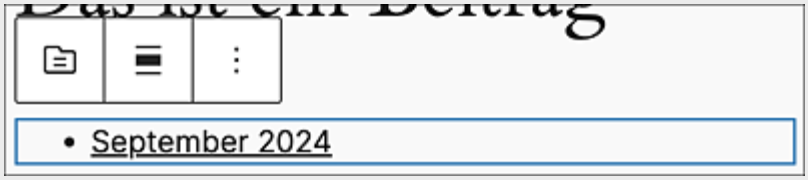
	aus, auch wenn diese beispielsweise über die gesamte Breite geht. • Daneben können Sie mit **Anordnung** steuern, ob die Elemente in der Gruppe links, zentriert oder rechts ausgerichtet sein sollen. • Über **Position** können Sie einstellen, dass der Block **Oben gehalten** wird, das heißt, er bleibt am oberen Viewport-Rand *kleben* und weiter sichtbar, auch wenn Sie weiterscrollen.
Gruppe (Forts.)	• Im **Erweitert**-Panel gibt es die schon bekannten Optionen für HTML-Anker und zusätzliche CSS-Klassen sowie die bereits einmal vorgekommene Option, das **HTML-Element** zu ändern. Besonders nützlich im Zusammenhang mit Block-Themes, mit denen die gesamte Website aus Blöcken gebaut wird. • Im **Stile**-Tab können Sie die bekannten Optionen **Farbe**, **Hintergrundbild**, **Typografie**, **Größe** sowie **Ränder und Schatten** nutzen, um die Gruppe anzupassen.

Zeile	Bei der Gruppe wurde es schon kurz angerissen: Der Zeilen-Block ist streng genommen kein eigener Block, sondern eine Block-Variation der Gruppe. Statt untereinander werden die Elemente hier nebeneinander angeordnet. Dafür wird die CSS-Eigenschaft `display: flex` genutzt, deren weitere Optionen Sie in den Block-Einstellungsmöglichkeiten wiederfinden.  Buttons: Hier finden Sie die schon vom Buttons-Block bekannten unterschiedlichen Ausrichtungsoptionen, sowohl für den Block selbst als auch die Blöcke innerhalb der Zeile. Seitenleiste: • Wie bei der Gruppe gibt es ganz oben die Option zum Umschalten zwischen den unterschiedlichen Variationen. Aus der Zeile in eine **Gruppe**, einen **Stapel** oder ein **Raster**. • Die **Layout**-Optionen sind bereits aus dem **Buttons**-Block bekannt, so können Sie die **Anordnung** und **Richtung** der Blöcke ändern (hiermit schalten Sie zwischen den Variationen **Zeile** und **Stapel** um) und steuern, ob Elemente in eine neue Zeile umbrechen dürfen. • Die anderen Einstellungsmöglichkeiten sind wie bei der Gruppe.

STAPEL	Wie die **ZEILE**, nur untereinander.  Buttons: die bekannten Optionen für Block-Ausrichtung und Ausrichtung der Kind-Blöcke Seitenleiste: die gleichen Optionen wie bei der **ZEILE**

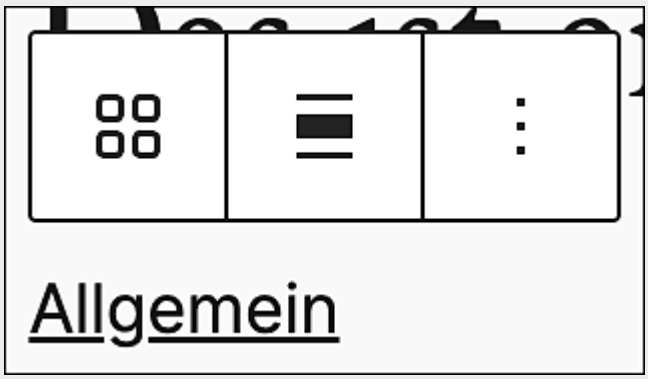
Raster	Diese Variation des Gruppen-Blocks nutzt das CSS-Grid-Layout, um die inneren Blöcke als Grid beziehungsweise **Raster** anzuzeigen.  Buttons: **Ausrichtung** erlaubt die Änderung der Block-Ausrichtung. Seitenleiste: • Mit der **Layout**-Option **Typ** können Sie einstellen, ob das Raster **Automatisch** oder **Manuell** sein soll. Bei der automatischen Variante werden die Elemente im Raster nebeneinander angeordnet, bis kein Platz mehr ist und in die nächste Zeile umgebrochen wird. Sie können eine **Minimale Spaltenbreite** angeben. Beim manuellen Typ können Sie die Anzahl der **Spalten** festlegen. • Die übrigen Optionen sind vom Gruppen-Block bekannt.
Mehr	Dieser Block ist für Blogbeiträge nützlich und erzeugt den **Weiterlesen**-Link auf der Übersichtsseite. Alle Inhalte davor werden in der Übersicht angezeigt, die Inhalte danach erst auf der Einzelansicht.

Mehr (Forts.)	Buttons: – Seitenleiste: Sie haben die Option, **Den Textauszug auf der kompletten Inhaltsseite auszublenden**, also auf der Einzelansicht wirklich nur den Teil des Inhalts nach dem Weiterlesen-Block anzuzeigen.
Seitenumbruch	Ermöglicht Ihnen die Unterteilung einer Seite oder eines Beitrags in mehrere Seiten. Der Block hat keine Optionen.
Trenner	Fügen Sie mit diesem Block einen visuellen Trenner in Ihre Inhalte ein.  Buttons: nur die Option zum Ändern der Block-**Ausrichtung** Seitenleiste: Sie können aus unterschiedlichen Stilen wählen, die **Farbe** anpassen und den **Außenabstand** verändern. Auch die Optionen unter **Erweitert** sind bereits bekannt.

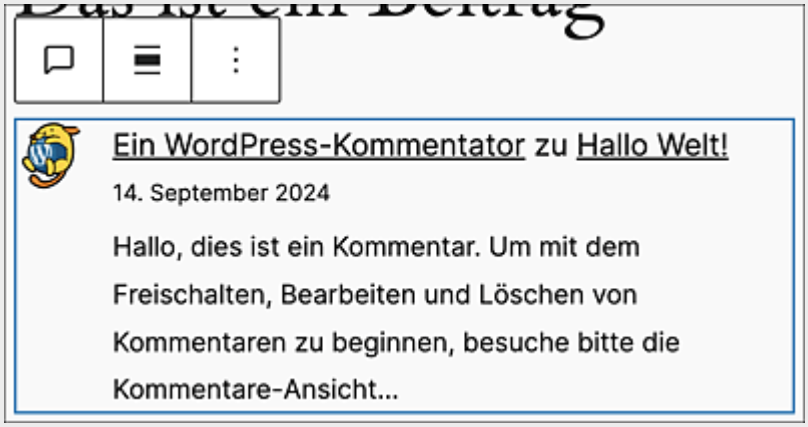
Abstandshalter	Wenn Ihnen zwischen zwei Blöcken der Abstand mal zu gering ist, greifen Sie zum »Abstandshalter«. Damit fügen Sie einen unsichtbaren Block ein, der einen von Ihnen definierten Platz freihält.  Buttons: – Seitenleiste: Sie können die **Höhe** anpassen (das ist auch per Drag-and-drop direkt in der Bearbeitungsansicht möglich) und im **Stile**-Tab einen Außenabstand einstellen. Die Optionen für den **HTML-Anker** und die **Zusätzlichen HTML-Klasse(n)** sind unter **Erweitert** natürlich auch wieder am Start.
Widgets	
Archive	Erzeugt eine Liste, über die Besucher*innen die Datum-Archiv-Übersichten Ihres Blogs erreichen können.  Buttons: nur für die **Ausrichtung** des Blocks Seitenleiste:

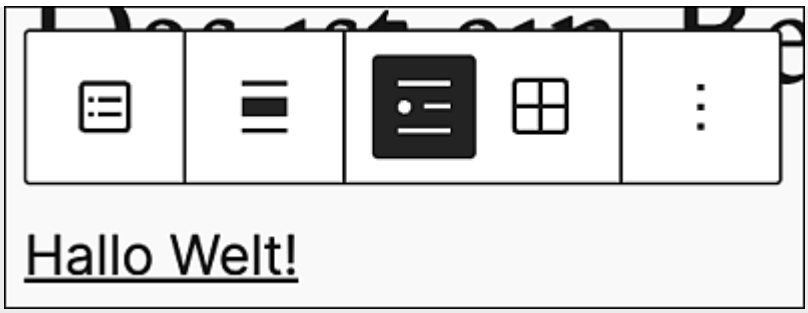
	• **Einstellungen** • **Als Drop-down-Liste anzeigen** wandelt die Liste in ein Drop-down-Feld um, der Sprung zur Archivseite geschieht automatisch nach dem Auswählen eines Eintrags. Wenn die Option aktiv ist, erhalten Sie die Möglichkeit, eine **Beschriftung** anzuzeigen. Das sollten Sie auch nutzen, da ein Formular-Feld immer eine Beschriftung haben sollte. • **Einstellungen** • **Beitragsanzahl anzeigen**: Hier können Sie entscheiden, ob hinter der Zeitangabe die Anzahl der in dieser Zeitspanne veröffentlichten Beiträge angezeigt werden soll. • **Einstellungen** • **Gruppieren nach**: Wählen Sie aus, ob die Liste nach Jahr, Monat, Woche oder Tag gruppiert sein soll. Wenn Sie nicht gerade mehrere Beiträge in der Woche schreiben, bietet sich wahrscheinlich am ehesten Monat oder Jahr an. • **Erweitert** bietet hier ausnahmsweise nur die Option **Zusätzliche CSS-Klasse(n)**. • Im **Stile**-Tab können Sie **Typografie**- und **Größen**-Einstellungen ändern.

Kalender	Zeigt einen Kalender an, in dem die Tage verlinkt sind, an dem Beiträge veröffentlicht wurden. Die Links führen zum Tagesarchiv.  Buttons: **Ausrichtung** des Blocks Seitenleiste: sehr übersichtlich mit Optionen für **Farbe** und **Typografie** sowie **Zusätzliche CSS-Klasse(n)** im **Erweitert**-Panel

Kategorien-Liste	Zeigt eine verlinkte Liste mit allen Kategorien an, die auf Ihrer WordPress-Site angelegt sind. Ein Klick auf einen der Links führt zum Kategorie-Archiv.  Buttons: Wie bei den bisherigen anderen Widgets auch gibt es nur die **Ausrichtung**-Option. Seitenleiste: • Das **Einstellungen**-Panel bietet Ihnen folgende Optionen: **Als Drop-down-Liste anzeigen** (mit der Option für eine Beschriftung), die **Beitragszahl anzeigen**, **Nur Kategorien der obersten Ebene anzeigen**, **Leere Kategorien anzeigen** (sie werden standardmäßig ausgeblendet) und die **Hierarchie anzeigen**, also Kind-Kategorien unter der Eltern-Kategorie auflisten.
Kategorien-Liste (Forts.)	• In **Erweitert** können Sie wieder optional CSS-Klassen angeben, und im **Stile**-Tab haben Sie Optionen für **Typografie**, **Größe** und **Ränder**.

Individuelles HTML	Hier können Sie Ihren HTML-Kenntnissen freien Lauf lassen und beliebigen HTML-Code einfügen, der im Frontend dann interpretiert wird.  Buttons: Sie können zwischen **HTML** und **Vorschau** wechseln. Seitenleiste: –

Neueste Kommentare	Zeigt die neuesten Kommentare an, die auf Ihrem Blog abgegeben wurden.  Buttons: wieder nur für die **Ausrichtung** des Blocks Seitenleiste: • **Einstellungen** bietet Ihnen die Möglichkeit, unterschiedliche Elemente der Kommentare ein- und auszublenden: **Avatar**, **Datum** und **Textauszug**. • Darüber hinaus können Sie die **Anzahl der Kommentare** steuern, die angezeigt werden sollen. • Unter **Erweitert** setzen Sie bei Bedarf wieder **Zusätzliche CSS-Klasse(n)**. • Der **Stile**-Tab bietet bereits bekannte Optionen für **Farbe**, **Typografie** und **Größe**.
Neueste Beiträge	Mit diesem Block können Sie eine Liste der neuesten Beiträge anzeigen.



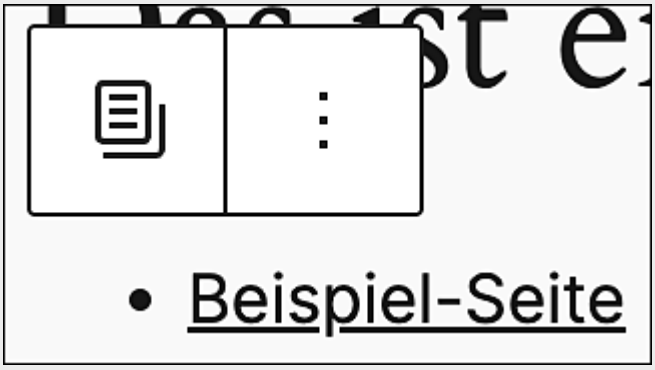
Buttons:

- **Ausrichtung** des Blocks
- Mit den Buttons **Listenansicht** und **Rasteransicht** können Sie das Layout des Blocks ändern.

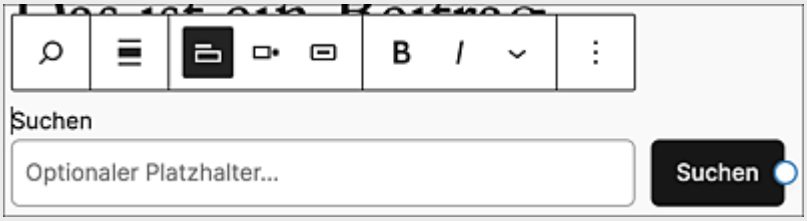
Seitenleiste:

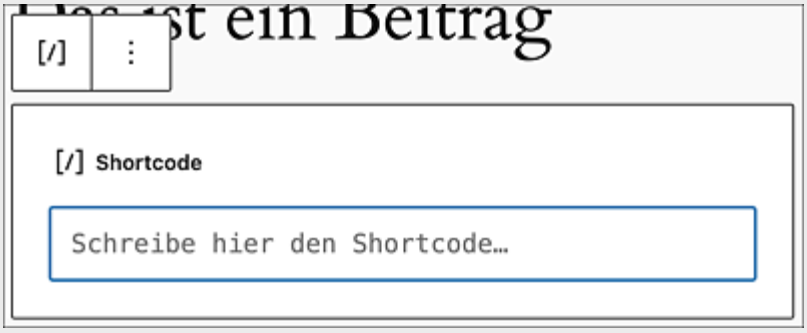
- Bei aktivierter **Beitragsinhalt**-Option können Sie wählen, ob Sie den **Textauszug** anzeigen möchten oder mit der Option **Vollständiger Beitrag** den gesamten Inhalt. Wenn Sie die **Textauszug**-Option gewählt haben, können Sie die **Max. Anzahl an Wörtern** einstellen, die der automatisch generierte Auszug lang ist.
- Das **Beitrags-Meta**-Panel bietet die Optionen, den **Autorennamen** und/oder das **Beitragsdatum** anzuzeigen.
- Außerdem können Sie wählen, ob Sie das **Beitragsbild anzeigen** möchten.
- Das Panel **Sortieren und filtern** lässt Sie Einfluss darauf nehmen, welche Beiträge in welcher Reihenfolge angezeigt werden. So können Sie über **Reihenfolge von** die Reihenfolge der Beiträge einstellen und zwischen Sortierung nach Datum und Alphabet wählen.

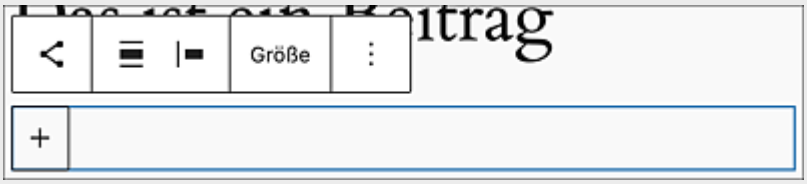
	• Das Eingabefeld **Kategorien** ermöglicht Ihnen, nur Beiträge einer oder mehrerer Kategorien anzuzeigen, mit **Autor** können Sie auf Beiträge einer bestimmten Person einschränken. • Mit **Anzahl der Elemente** geben Sie die Beitragsanzahl an, die maximal angezeigt werden soll.
Neueste Beiträge (Forts.)	• Im **Erweitert**-Panel finden Sie die CSS-Klassen-Option. • Im **Stile**-Tab der Block-Seitenleiste können Sie bereits bekannte Optionen für **Farbe**, **Typografie** und **Größe** nutzen.

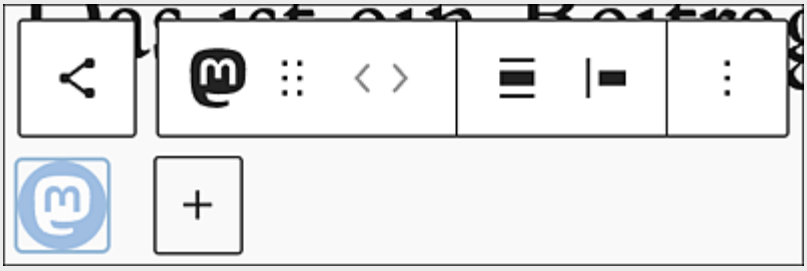
Seitenliste	Wenn Sie nicht die Beiträge, sondern Seiten anzeigen möchten, etwa für eine Sitemap, nutzen Sie den **Seitenliste**-Block.  Buttons: – Seitenleiste: • **Übergeordnet** lässt Sie eine Seite auswählen, zu der die Kind-Seiten angezeigt werden. • **Erweitert** bietet, Sie erraten es, **Zusätzliche CSS-Klasse(n)**. • Im **Stile**-Tab können Sie Einstellungen für **Typografie** vornehmen.

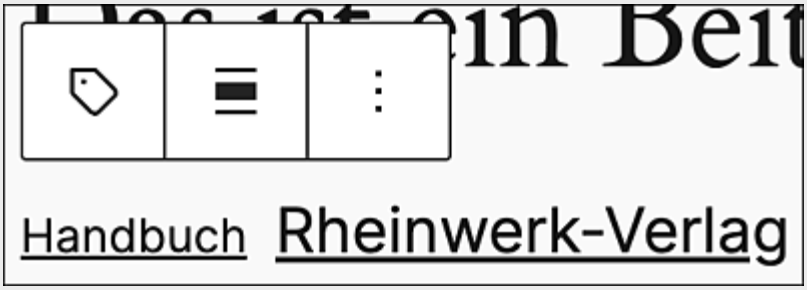
RSS	Dieser Block ermöglicht es Ihnen, Einträge eines beliebigen RSS-Feeds anzuzeigen. Dafür geben Sie nach dem Einfügen des Blocks einfach die entsprechende URL an (der Haupt-Feed für Ihre Website ist unter *example.com/feed* zu finden, wenn Sie nicht die **Einfach**-Permalinkstruktur nutzen. Ist die einfache Struktur aktiv, finden Sie den Feed unter *example.com/?feed=rss2*).  Buttons: • **Ausrichtung** zur Änderung der Block-Ausrichtung • **RSS-URL bearbeiten**, um die Quelle der angezeigten Beiträge zu ändern • **Listenansicht** und **Rasteransicht**, um das Layout der Beiträge zu ändern
RSS (Forts.)	Seitenleiste: • Unter **Einstellungen** können Sie die **Anzahl der Elemente** ändern, die angezeigt werden sollen. Außerdem können Sie steuern, ob Sie **Autor**, **Datum** und/oder **Textauszug** anzeigen möchten oder nicht. • **Erweitert** hält wie bei den anderen Widget-Blöcken nur die Option für CSS-Klassen bereit.

Suchen	Bietet Ihnen die Möglichkeit, ein Suchformular einzubinden. Innerhalb der Bearbeitungsansicht können Sie die Beschriftung des Felds sowie des »Suchen«-Buttons ändern und einen optionalen Platzhaltertext eingeben.  Buttons: • **Ausrichtung** zur Anpassung der Block-Ausrichtung • **Suchbeschriftung ein-/ausblenden** zum Anzeigen oder Verstecken der Feldbeschriftung • **Buttonposition ändern** gibt Ihnen unterschiedliche Optionen zur Position des Suchen-Buttons. So kann er beispielsweise optisch innerhalb des Such-Felds angezeigt oder komplett ausgeblendet werden. • **Button mit Icon benutzen** tauscht den Button-Text gegen ein Lupen-Icon aus. • Wenn Sie eine der Beschriftungen bearbeiten, stehen Ihnen noch die gängigen Textformatierungsoptionen zur Verfügung. Seitenleiste: • **Einstellungen** bietet die Option, die **Breite** zu ändern. Hier können Sie entweder einen freien Wert eingeben oder aus den Optionen 25, 50, 75 und 100 Prozent Breite wählen.

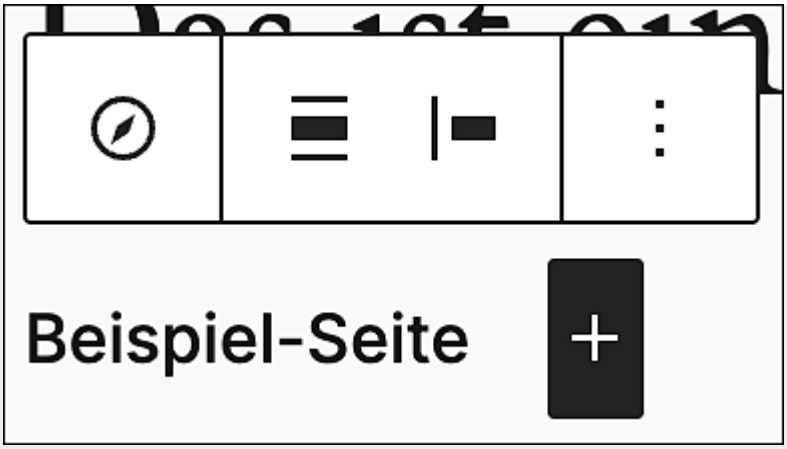
Suchen (Forts.)	• **Erweitert** bringt wieder das Feld für **Zusätzliche CSS-Klasse(n)** mit. • Im **Stile**-Tab haben Sie Einstellungsmöglichkeiten für **Farbe**, **Typografie**, **Größe** und **Ränder**.
Shortcode	Ein Block zum Einfügen eines Shortcodes. Bevor es den Block-Editor gab, waren Shortcodes die einzige Möglichkeit, komplexere Inhaltselemente im Editor einzufügen, ohne selbst HTML schreiben zu müssen. Der Block bietet keine Optionen, weder in der Block-Toolbar noch in der Block-Seitenleiste. 

Social Icons	Ein Wrapper-Block für einen oder mehrere »Social Icon«-Blöcke, mit denen Sie Links zu unterschiedlichen externen Sites als Icons anzeigen können, beispielsweise zu Ihrem Mastodon-Account.  Buttons: • **Ausrichtung** zur Steuerung der Block-Ausrichtung • **Ausrichtung der Elemente ändern** zur Steuerung der Icons innerhalb des Blocks • **Größe** zum Ändern der Icon-Größe Seitenleiste: • Unter **Layout** haben Sie die bereits von **Zeile** und **Stapel** bekannten Optionen, die **Anordnung** der Elemente sowie die **Richtung** zu ändern und den **Zeilenumbruch auf mehrere Zeilen zu erlauben**.

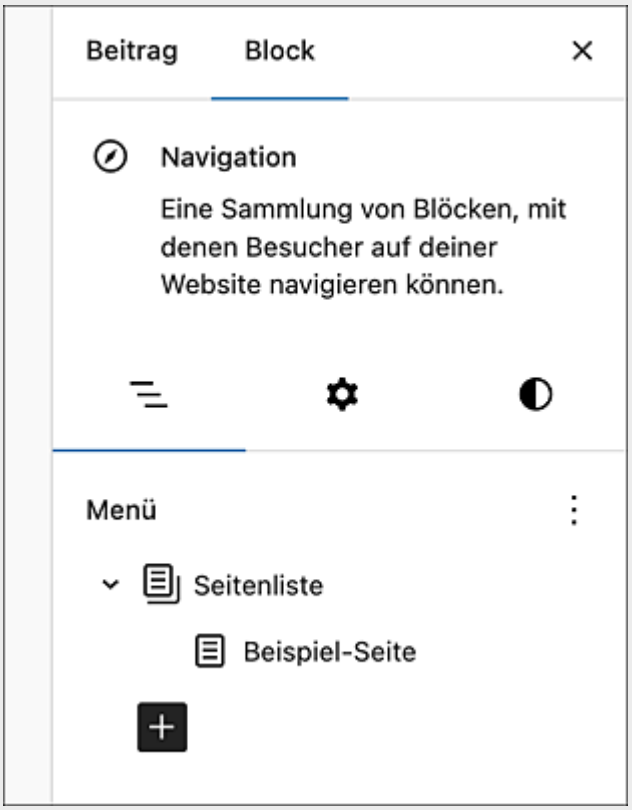
Social Icons (Forts.)	• Im **Einstellungen**-Panel können Sie einstellen, ob die **Links in neuem Tab öffnen** sollen und ob Sie mit **Text anzeigen** den Namen der Plattform/des Netzwerks neben dem Icon anzeigen möchten. • **Erweitert** bietet hier neben den CSS-Klassen auch wieder die **HTML-Anker**-Option an. • Im **Stile**-Tab können Sie unterschiedliche **Stile** auswählen sowie Einstellungen für **Farbe**, **Größe** und **Ränder** anpassen.
Social Icon (nur im Social-Icons-Block)	Dieser Block ist nur als Kind-Block von »Social Icons« vorhanden, kann also nicht direkt im Block-Inserter ausgewählt werden. Der Block besteht aus Variationen für die unterschiedlichen Plattformen und Netzwerke und wird deshalb im Editor beispielsweise auch als **Mastodon**-Block angezeigt.  Buttons: Wie beim **Button**-Block können Sie auch hier bei den Kind-Blöcken die Ausrichtungseinstellungen des Wrapper-Blocks **Social Icons** ändern. Seitenleiste:

	• Mit **Text** können Sie die Beschriftung ändern, die neben dem Button angezeigt wird, falls die Option beim Block **Social Icons** aktiviert wurde. • Unter **Erweitert** können Sie neben CSS-Klassen das `rel`-Attribut des Links anpassen. So können Sie `rel="me"` für einen Link setzen, um beispielsweise auf Ihr persönliches Mastodon-Profil zu verlinken, und wenn Sie dann von Ihrem Mastodon-Profil auf Ihre Website zurückverweisen, wird der Link als verifiziert angezeigt. Dadurch ist auf Mastodon erkennbar, dass dieses Profil zu der Person gehört, die hinter der Website steht.
Schlagwörter-Wolke	Zeigt alle Schlagwörter an, die Beiträgen zugeordnet sind. Je mehr Beiträgen ein Schlagwort zugeordnet ist, desto größer wird es dargestellt. Jedes Schlagwort verlinkt auf die Archiv-Seite mit der Liste aller Beiträge zu diesem Schlagwort.  Buttons: nur der **Ausrichtung**-Button Seitenleiste: • Auch wenn der Block **Schlagwörter-Wolke** heißt, können Sie unter **Einstellungen** • **Taxonomie** eine andere

	Taxonomie auswählen, zu der die Terme angezeigt werden sollen. Bei einer Standard-WordPress-Installation können Sie hier noch **Kategorien** auswählen. • Die weiteren Optionen im **Einstellungen**-Panel sind **Beitragsanzahl anzeigen**, **Anzahl der Schlagwörter** (oder Kategorien) sowie die Angabe der kleinsten und größten Schriftgröße, die WordPress nutzen soll (**Kleinste Abmessung** und **Größte Abmessung**). • Unter **Erweitert** gibt es bei diesem Block wieder nur die Option für **Zusätzliche CSS-Klasse(n)**. • Im **Stile**-Tab können Sie andere **Stile** für den Block wählen sowie Optionen in den Bereichen **Typografie**, **Größe** und **Ränder** ändern.
Theme	
Navigation	Dieser Block ermöglicht die Erstellung und Anzeige eines Menüs. Der Block ist ein Wrapper-Block, in den theoretisch beliebige Blöcke eingebunden werden können. Standardmäßig wird der **Seitenliste**-Block als Kind-Block eingefügt, sodass automatisch die Links zu allen veröffentlichten Seiten angezeigt werden.
Navigation (Forts.)	



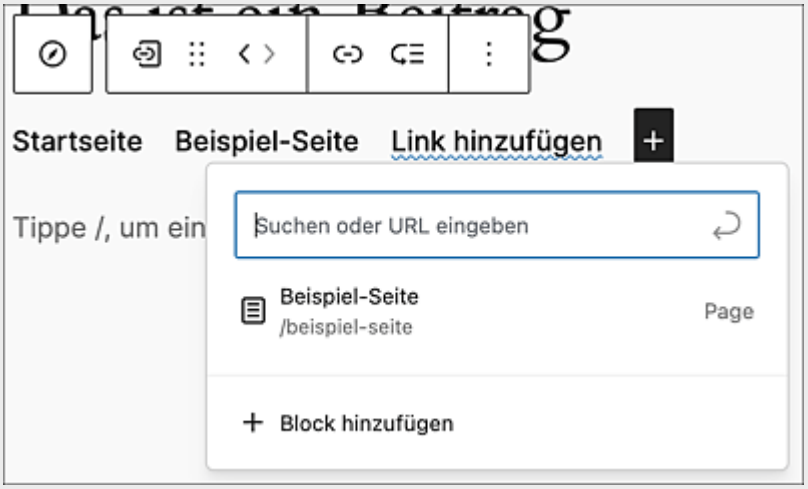
Sie können das Menü direkt über die Bearbeitungsansicht im Editor bearbeiten oder rechts in der Block-Seitenleiste, wo es einen eigenen Tab **LISTENANSICHT** gibt, in dem die Blöcke des Menüs dargestellt sind (siehe nachfolgende Abbildung).



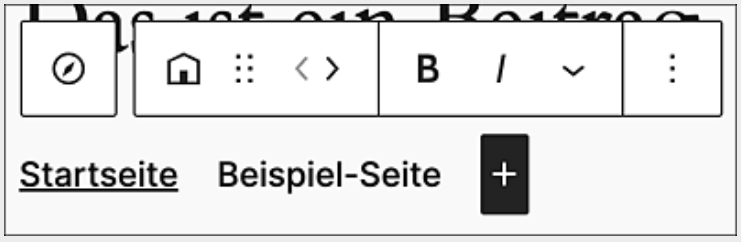
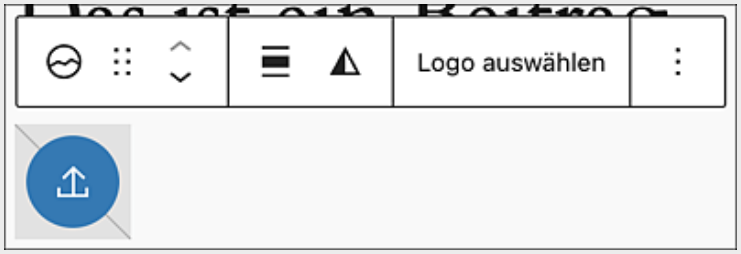
Buttons:
Optionen zur Ausrichtung des Blocks selbst

	sowie der Kind-Blöcke Seitenleiste: • Wie schon erwähnt, gibt es bei diesem Block einen Tab **Listenansicht**, der Ihnen die Blöcke anzeigt, die im Navigation-Block eingefügt sind. Außerdem können Sie hier die Navigation bearbeiten. Über das Drei-Punkte-Menü oben rechts in dem Panel können Sie ein bereits vorhandenes Menü auswählen oder neu beginnen. • Im **Einstellungen**-Tab können Sie das **Layout** anpassen, wie Sie es beispielsweise beim Buttons-Block schon gesehen haben.

Navigation (Forts.)	• Außerdem gibt es im **Darstellung**-Panel die Möglichkeit, das Icon für das mobile Menü zu ändern oder gegen Text-Versionen auszutauschen und einzustellen, ob das **Overlay-Menü**, also die Version, die über einen Button geöffnet wird, **Aus** ist, nur **Mobil** oder **Immer** angezeigt werden soll. • Unter **Erweitert** haben Sie neben der Angabe von CSS-Klassen die Möglichkeit, den Namen des Menüs zu ändern. Wenn Sie die Seite oder den Beitrag speichern, wird Ihnen dann angezeigt, dass neben dem Inhalt auch das Menü gespeichert wird. Außerdem wird im **Erweitert**-Panel ein Link zur Liste aller Menüs angezeigt, um sie bei Bedarf alle auf einen Blick zu haben und schnell bearbeiten zu können. • Im **Stile**-Tab können Sie neben bekannten Optionen für **Farbe**, **Typografie** und **Größe** die Farbe für Untermenü- und Overlay-Texte sowie die Hintergrundfarbe für Untermenüs und das Overlay anpassen.

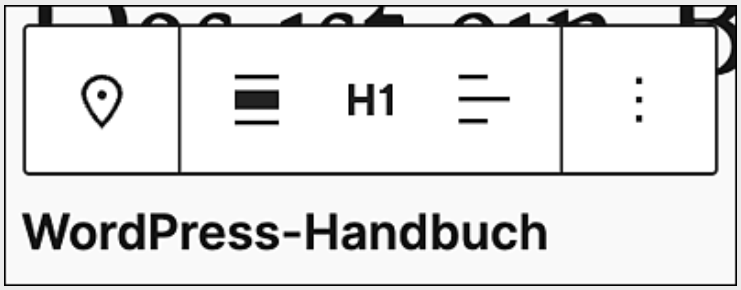
Individueller Link **Seitenlink** **Beitragslink** (nur im Navigations-Block)	Wenn Sie den Navigations-Block auswählen, wird ein Plus-Symbol nach dem letzten Eintrag angezeigt, über den Sie einen neuen Menüeintrag einfügen können. Wenn Sie das Plus anklicken, wird Ihnen ein Dialog angezeigt, in dem Sie einen individuellen Link oder einen bestehenden Inhalt auswählen können.  Wenn Sie stattdessen **Block hinzufügen** auswählen, können Sie einen beliebigen anderen Block auswählen, unter anderem die Blöcke **Seitenlink** und **Beitragslink**, die Variationen des Link-Blocks sind.

Individueller Link **Seitenlink** **Beitragslink** (nur im Navigations-Block) (Forts.)	Die Ansicht zum Setzen eines Links ist der von der Standard-Variante **Individueller Link** sehr ähnlich, Sie können über die Suche aber nur Inhalte des jeweiligen Inhaltstyps finden. Buttons: • **Link** ermöglicht die Bearbeitung der URL. • **Untermenü hinzufügen**: Über diesen Button können Sie zu einem Menüpunkt ein Untermenü erstellen, um untergeordnete Menüpunkte festzulegen. • die bekannten Text-Formatierungsoptionen Seitenleiste: Im **Einstellungen**-Panel können Sie den Linktext sowie die URL bearbeiten und eine Beschreibung angeben, deren Anzeige aber abhängig vom Theme ist. Darüber hinaus können Sie das `title`- und `rel`-Attribut setzen. Unter **Erweitert** finden Sie die Möglichkeit, **Zusätzliche CSS-Klasse(n)** anzugeben. Im **Stile**-Tab können Sie Einstellungen für **Typografie** und **Größe** vornehmen und unterschiedliche **Stile** wählen.

Home-Link (nur im Navigations-Block)	Verlinkt immer auf die Startseite.  Buttons: Möglichkeiten zur Text-Formatierung Seitenleiste: Hier können Sie **Typografie**- und **Größe**-Optionen anpassen sowie unter **Erweitert** CSS-Klassen vergeben.
Untermenü (nur im Navigations-Block)	Wenn Sie einem Menüeintrag ein Untermenü hinzufügen, wird dieser Punkt in der Block-Sidebar zum **Untermenü**-Block, die Einstellungsmöglichkeiten bleiben aber gleich.
Website-Logo	Zeigt das Website-Logo an und ermöglicht auch, ein Logo festzulegen. Dieses Logo wird dann für die gesamte Website genutzt und wird bei anderen Website-Logo-Blöcken oder anderen Funktionen, die das Website-Logo zeigen, ausgegeben.  Buttons: Möglichkeiten zur **Ausrichtung** und

	Anwendung von **Duoton**-Effekten sowie, wenn ein Logo ausgewählt ist, zum **Zuschneiden** und **Ersetzen** des Bilds. Andernfalls haben Sie einen Button zum Auswählen eines Logos. Die Optionen sind bereits vom **Bild**-Block bekannt. Seitenleiste: • Unter **Medien** können Sie das Bild auswählen, das als Logo genutzt werden soll. Wenn Sie ein Logo gesetzt haben, bekommen Sie weitere **Einstellungen** und können die **Bildbreite** angeben, das **Bild auf die Homepage verlinken**, den Link **In einem neuen Tab öffnen** und das Logo **Als Website-Icon verwenden**. Das Website-Icon ist das Bild, das Sie auch links im Admin-Menü unter **Einstellungen** • **Allgemein** festlegen können, und das unter anderem als Favicon im Browser-Tab angezeigt wird, wenn Ihre Website geöffnet ist. • **Erweitert** wartet wieder mit der CSS-Klassen-Option auf. • Im **Stile**-Tab können Sie den Block-Stil ändern sowie einen **Duoton**-Filter einstellen und Innen- und Außenabstand anpassen.
Website-Titel	Zeigt den Titel der Website an, der im Admin-Menü unter **Einstellungen** • **Allgemein** festgelegt werden kann. Der

Titel kann wie das Logo auch direkt über den Block geändert werden.

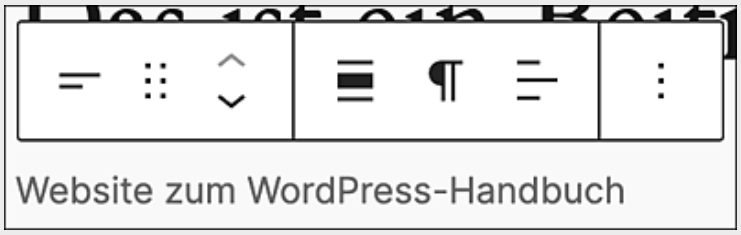


Buttons:

- **Ausrichtung** zur Änderung der Block-Ausrichtung
- **Ebene wechseln** lässt Sie festlegen, ob der Titel eine Überschrift (**H1** bis **H6**) oder ein Absatz sein soll.
- **Text ausrichten** zur Änderung der Text-Ausrichtung auf links, zentriert oder rechts

Seitenleiste:

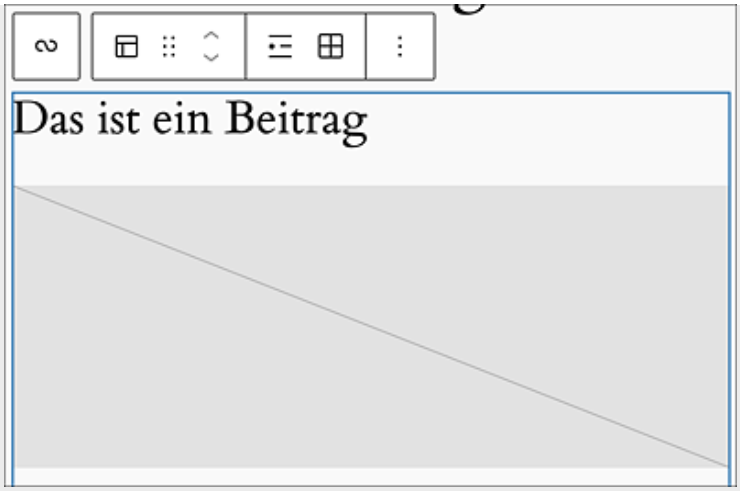
- Unter **Einstellungen** können Sie **Den Titel auf die Startseite verlinken**, und wenn diese Option aktiv ist, können Sie den Link **in einem neuen Tab öffnen**.
- Im **Erweitert**-Panel können Sie **Zusätzliche CSS-Klasse(n)** angeben.
- Im **Stile**-Tab können Sie bereits bekannte Optionen für **Farbe**, **Typografie**, **Größe** und **Ränder** nutzen.

Website-Untertitel	Zeigt den Untertitel der Website an, der im Admin-Menü unter **Einstellungen** • **Allgemein** festgelegt werden kann, und wie der Titel und das Logo auch direkt über den Block bearbeitet werden können.  Buttons: • **Ausrichtung** zum Festlegen der Block-Ausrichtung • **Ebene wechseln** lässt Sie wie bei dem Titel-Block die Ebene wechseln. Der Standard ist ein Absatz, Sie können aber auch eine Überschriftenebene wählen. • **Text ausrichten** kennen Sie auch schon.
Website-Untertitel (Forts.)	Seitenleiste: Sie können Optionen für **Farbe**, **Typografie**, **Größe** und **Ränder** nutzen sowie unter **Erweitert** eigene CSS-Klassen angeben.
Abfrage-Loop	Ein mächtiger Block, der den Loop aus den Classic-Themes in den Block-Editor bringt. Mit diesem Block können Sie eine Liste von Einträgen eines Inhaltstyps (standardmäßig Beiträge oder Seiten) anzeigen und flexibel entscheiden, aus welchen Blöcken ein Eintrag dieser Liste,

	also etwa der Teaser für einen Beitrag, zusammengesetzt sein soll. Mit diesem Block werden beispielsweise bei Block-Themes die Archiv-Templates umgesetzt oder die Blog-Übersicht. Wenn Sie einen Abfrage-Loop-Block einbauen, können Sie aus verschiedenen Layouts wählen, um nicht bei null starten zu müssen.  Buttons: • **Ausrichtung** zur Anpassung der Block-Ausrichtung • Mit dem **Anzeige-Einstellungen**-Button können Sie die Anzahl der **Elemente pro Seite**, einen optionalen **Offset** und die **Maximale Seitenanzahl** einstellen (der Button wird nur angezeigt, wenn die Option **Einstellungen • Abfrage vom Template übernehmen** in der Seitenleiste nicht aktiv ist).

Abfrage-Loop (Forts.)	• **Ersetzen** ermöglicht Ihnen, das Block-Layout mit einer Vorlage zu verändern. Seitenleiste: • **Layout** bietet dieselben Optionen wie im **Gruppe**-Block. • Im **Einstellungen**-Panel können Sie, falls Sie im Website-Editor ein Template bearbeiten, das eine Liste von Beiträgen anzeigt, die Art der Abfrage einstellen und zwischen **Standard** und **Individuell** wählen. Die **Standard**-Option bedeutet, dass die passenden Beiträge zum aufgerufenen Archiv angezeigt werden. Über die Option **Individuell** können Sie die Abfrage mit den folgenden Optionen selbst beeinflussen. Diese Optionen werden auch angezeigt, wenn Sie den Block beispielsweise in einem Beitrag nutzen. Bei der individuellen Einstellung können Sie den **Inhaltstyp** auswählen, von dem die Inhalte angezeigt werden sollen und mit **Reihenfolge von** die Reihenfolge ändern. Zusätzlich können Sie angeben, ob **Oben gehaltene Beiträge** enthalten oder nicht enthalten sein sollen. • Im **Darstellung**-Panel können Sie einstellen, wie viele **Beiträge pro Seite** angezeigt werden sollen, ob ein **Offset** genutzt werden soll, also

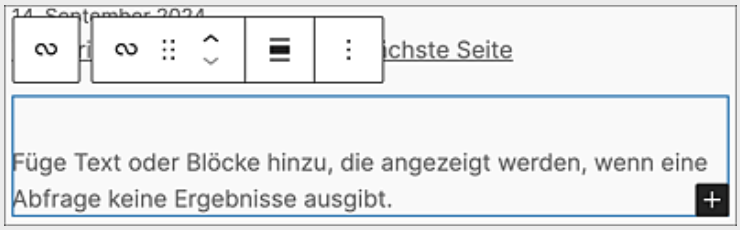
	beispielsweise beim dritten Beitrag angefangen werden soll, und ob eine Maximalanzahl von Seiten angezeigt werden soll. • Unter **Filter** haben Sie die Optionen, die Liste der Beiträge nach Autor*innen, Taxonomie-Termen (Kategorien/Schlagwörter) und nach einem beliebigen Stichwort zu filtern. • Im **Erweitert**-Panel gibts wieder die CSS-Klassen-Option und diesmal auch die zur Änderung des **HTML-Element**s. Außerdem können Sie einstellen, ob die gesamte Seite neu laden soll. Andernfalls wird, falls alle Blöcke kompatibel sind, die Seite nicht neu geladen, wenn zwischen Seiten navigiert wird.
Beitrags-Template (nur innerhalb eines Abfrage-Loop-Blocks)	Dieser Block enthält die Einträge des Abfrage-Loops, er kann nur als Block innerhalb des Abfrage-Loop-Blocks genutzt werden und wird standardmäßig eingefügt. Blöcke innerhalb dieses Blocks werden für jeden Beitrag oder jede Seite wiederholt, die der Abfrage-Loop ausgibt.

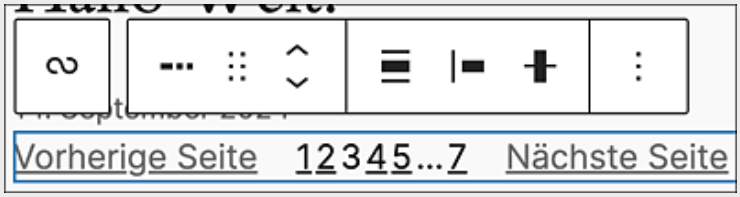


Buttons:
Sie können zwischen einer **Listenansicht** und einer **Rasteransicht** wechseln.

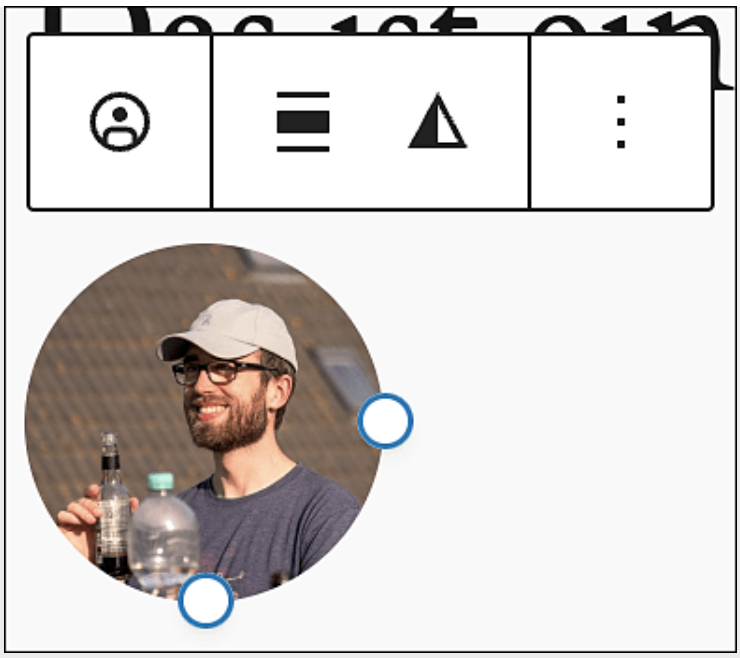
Seitenleiste:

- Im **Layout**-Panel haben Sie bei der Listenansicht die Option zur Steuerung der Breite der Blöcke innerhalb des Beitragstemplates-Blocks und bei der Rasteransicht dieselben Optionen wie beim **Raster**-Block: Sie können zwischen den Typen **Automatisch** und **Manuell** wählen und bei **Automatisch** die **Minimale Spaltenbreite** festlegen sowie bei **Manuell** die Anzahl der **Spalten**.
- Im **Erweitert**-Panel gibt es nur die CSS-Klassen-Option.
- Im **Stile**-Tab haben Sie Einstellungsmöglichkeiten für **Farbe**, **Typografie** und **Größe**.

Keine Ergebnisse (nur im Abfrage-Loop-Block)	Auch dieser Block kann nur im **Abfrage-Loop**-Block genutzt werden. Der Inhalt dieses Blocks wird angezeigt, wenn es keine Ergebnisse im Abfrage-Loop gibt.
Keine Ergebnisse (nur im Abfrage-Loop-Block) (Forts.)	Sie könnten also beispielsweise den Suche-Block einfügen oder die Liste der Seiten, um den Besucher*innen eine Möglichkeit zu geben, sich weiter auf Ihrer Website umzuschauen.  Buttons: **Ausrichtung** zur Änderung der Block-Ausrichtung Seitenleiste: Sie können Änderungen in den Bereichen **Farbe** und **Typografie** vornehmen und natürlich **Zusätzliche CSS-Klasse(n)** angeben.

Seitennummerierung (nur im Abfrage-Loop-Block)	Dieser Block zeigt, falls notwendig, die Paginierung für Abfrage-Loop-Blöcke an.  Buttons: alle Ausrichtungsoptionen, sowohl für den Block selbst als auch für die Kind-Blöcke Seitenleiste: • Unter **Layout** haben Sie die Optionen, die beispielsweise schon von den Blöcken **Zeile** und **Stapel** bekannt sind. • Im **Einstellungen**-Panel können Sie angeben, ob bei den Links für die vorherige und nächste Seite ein Pfeil-Symbol angezeigt werden soll. Wenn Sie die Option aktivieren, können Sie den Beschriftungstext für diese Links optional ausblenden. • Unter **Erweitert** gibt es wieder die CSS-Klassen-Option. • Im **Stile**-Tab finden Sie Einstellungen zu **Farbe** und **Typografie**.

Nächste/Vorherige Seite (nur im Seitennummerierungs-Block)	Zwei Blöcke zur Anzeige des Links zur nächsten beziehungsweise vorherigen Seite im Abfrage-Loop. Buttons: – Seitenleiste: Optionen für **Farbe** und **Typografie** sowie das obligatorische CSS-Klassen-Feld
Seitenzahlen (nur im Seitennummerierungs-Block)	Ohne Seitenzahlen ist eine Paginierung nicht vollständig, und dafür gibt es den Seitenzahlen-Block. Buttons: – Seitenleiste • In den **Einstellungen** können Sie die Anzahl der Links einstellen, die vor und nach dem Link mit der aktuellen Seitenzahl erscheinen sollen, bevor ein Auslassungszeichen kommt. • Unter **Erweitert** haben Sie, natürlich, die CSS-Klassen-Option. • Im **Stile**-Tab können Sie Anpassungen für **Farbe** und **Typografie** vornehmen.
Avatar	Zeigt den Avatar eines Accounts Ihrer Website an.

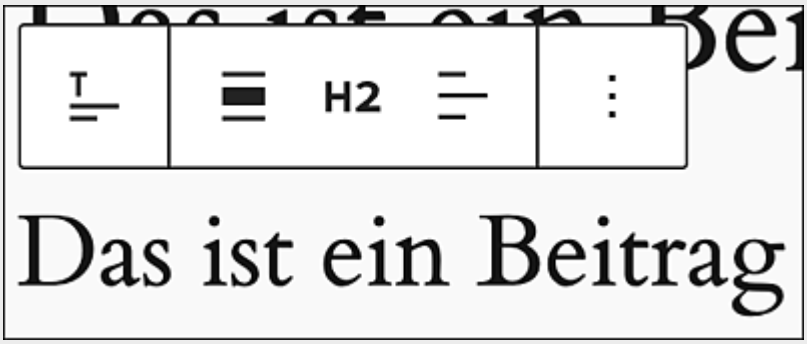


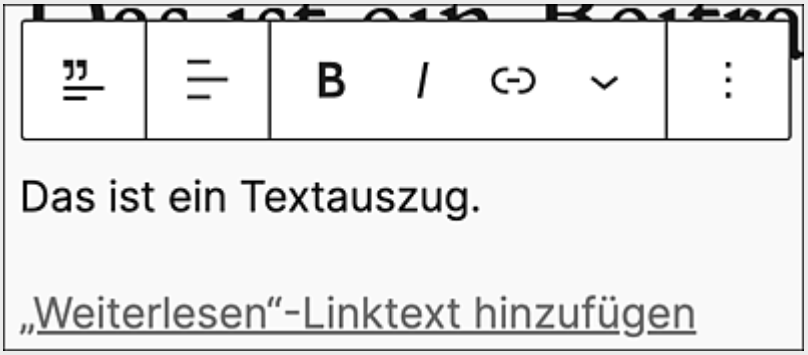
Buttons:

- **Ausrichtung** des Blocks
- **Duoton-Filter anwenden** für einen interessanten visuellen Effekt

Seitenleiste:

- Im **Einstellungen**-Panel können Sie die **Bildgröße** anpassen, das Bild mit einem **Link zum Benutzerprofil** ausstatten und den **Benutzer** auswählen, von dem der Avatar angezeigt werden soll.
- Unter **Erweitert** haben Sie wieder die Option für **Zusätzliche CSS-Klasse(n)**.

Avatar (Forts.)	• Im Tab **Stile** können Sie den **Duoton**-Filter setzen oder anpassen sowie Einstellungen für **Größe** und **Ränder** vornehmen.
Titel	Ein Block zur Anzeige des Titels der aktuellen Seite, des aktuellen Beitrags oder der Einzelansicht einer anderen Inhaltsart. Zusätzlich zur Anzeige kann der Titel auch im Block bearbeitet werden.  Buttons: • Block-**Ausrichtung** • Ändern der Überschriftenebene • Anpassen der Textausrichtung Seitenleiste: • Sie können den **Titel als Link darstellen** und natürlich CSS-Klassen definieren, die zusätzlich eingefügt werden sollen. • Unter **Stile** können Sie Optionen für **Farbe**, **Typografie**, **Größe** und **Ränder** anpassen.

Textauszug	Zeigt den Textauszug eines Inhalts an und ermöglicht die Bearbeitung.  Buttons: Textausrichtung und die bekannten Textformatierungsoptionen wie Verlinkung, Kursivschrift et cetera. Seitenleiste: • **Einstellungen** • **Link in neuer Zeile anzeigen** steuert, ob der Weiterlesen-Link direkt am Ende des Textauszugs angezeigt wird oder in einer neuen Zeile.
Textauszug (Forts.)	• **Einstellungen** • **Max. Anzahl an Wörtern** ermöglicht Ihnen, die Wortanzahl zu ändern, die maximal angezeigt wird. • **Erweitert** • **Zusätzliche CSS-Klasse(n)**, allseits bekannt • Im **Stile**-Tab können Sie **Farbe**, **Typografie**, **Größe** und **Ränder** anpassen.
Beitragsbild	Zeigt das Beitragsbild an und erlaubt, das Bild zu ändern.



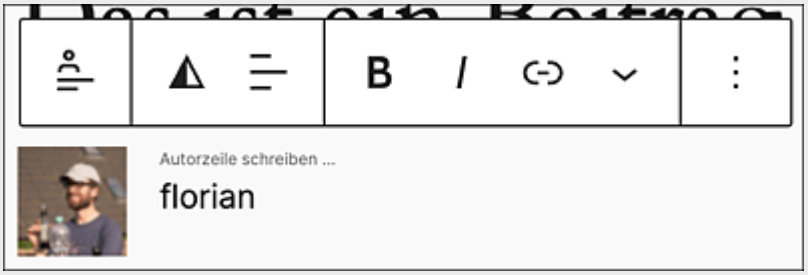
Buttons:

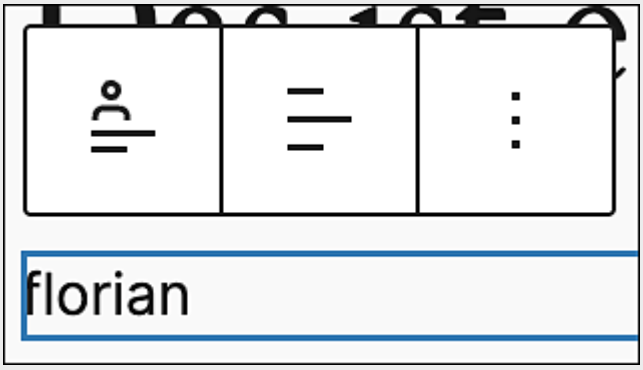
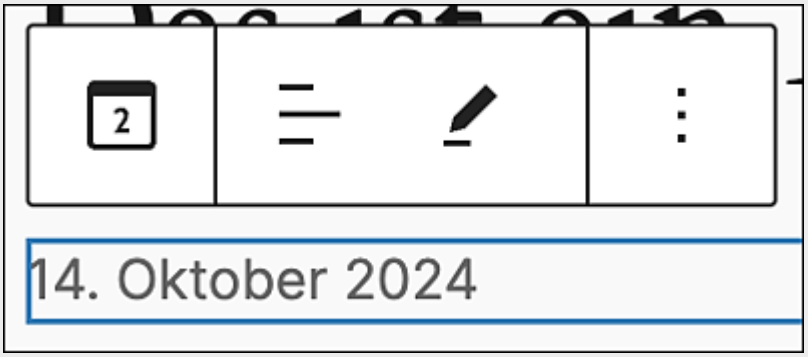
- Block-**Ausrichtung**
- **Duoton-Filter anwenden**
- **Ersetzen**, um ein gewähltes Bild zu ersetzen

Seitenleiste:

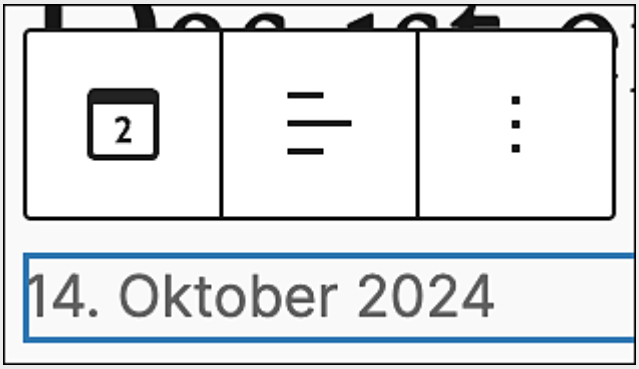
- Unter **Einstellungen** können Sie das Beitragsbild als **Link** zum Beitrag auszeichnen und unter **Erweitert** zusätzliche CSS-Klassen setzen.
- Im **Stile**-Tab können Sie Overlay-Farbe und -Deckkraft anpassen, den Duoton-Filter anpassen und **Größe**-Optionen nutzen, wobei Sie auch das Seitenverhältnis sowie Höhe und Breite anpassen können. Außerdem sind

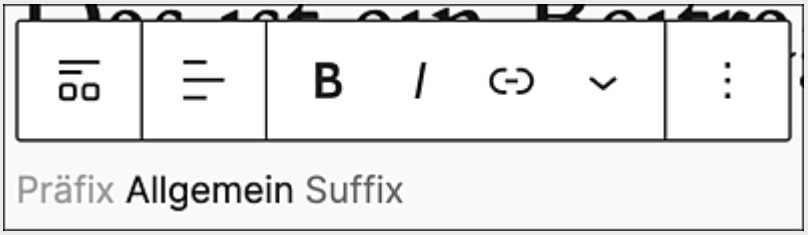
	Optionen für **Rand und Schatten** verfügbar.

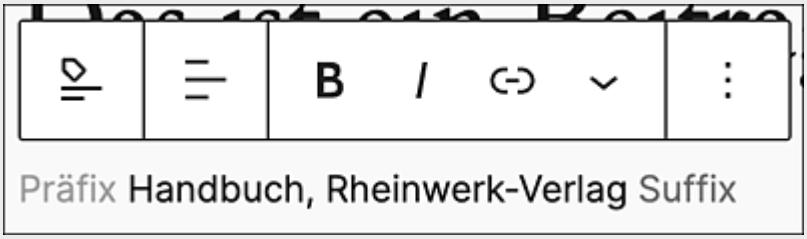
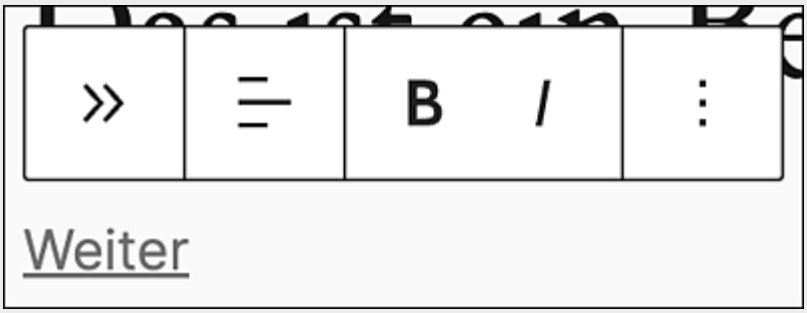
Autor	Zeigt Name und gegebenenfalls weitere Infos zur Person an, die den aktuellen Inhalt verfasst hat.  Buttons: • **Duoton-Filter anwenden** für einen grafischen Effekt für den Avatar • **Text ausrichten** • die Textformatierungsoptionen, die Sie bereits von anderen Blöcken kennen Seitenleiste: • Im **Einstellungen**-Panel können Sie ändern, wer Autor*in des Beitrags ist, und einstellen, ob Sie den **Avatar anzeigen** möchten. Die **Avatargröße** können Sie in dem Fall auch anpassen, außerdem können Sie die **Bio anzeigen** und **Den Namen des Autors auf die Seite des Autors verlinken**. • Im **Erweitert**-Panel erwartet Sie die allseits bekannte Option **Zusätzliche CSS-Klasse(n)**. • Im **Stile**-Tab können Sie Einstellungen für **Farbe**, **Duoton**, **Typografie**, **Größe** und **Ränder** anpassen.

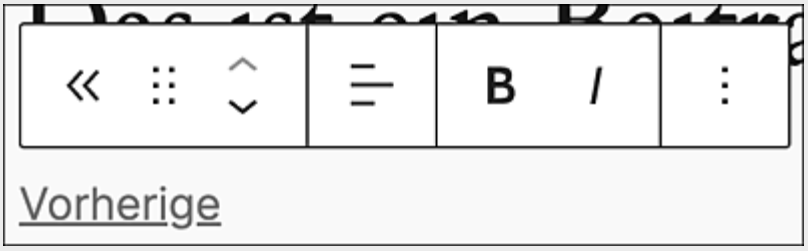
Name des Autors	Gibt den Namen der Autor*in aus, nicht mehr und nicht weniger.  Buttons: Option zur Textausrichtung Seitenleiste: • Sie können den Namen mit einem **Link zum Autorenarchiv** versehen und, wenn Sie möchten, **Zusätzliche CSS-Klasse(n)** vergeben. • Bei **Stile** haben Sie die Möglichkeit, Anpassungen für **Farbe**, **Typografie**, **Größe** und **Ränder** zu machen.
Datum	Block für die Anzeige des Veröffentlichungsdatums eines Beitrags, einer Seite oder eines Elements eines anderen Inhaltstyps. Ermöglicht auch die Änderung des Datums. 

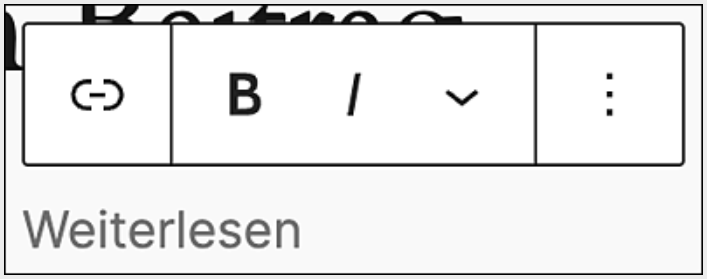
	Buttons: • Textausrichtung • **Datum ändern**, sodass Sie direkt mit dem Block das Datum des aktuellen Eintrags ändern können Seitenleiste: • Im **Einstellungen**-Panel können Sie wählen, ob das Datum das in den WordPress-Einstellungen vergebene **Standardformat** haben soll, oder ein individuelles Format festlegen. Außerdem können Sie das Datum als **Link** zur Einzelansicht ausgeben und, falls der Beitrag oder die Seite nach der Veröffentlichung bearbeitet wurde, **Das letzte Bearbeitungsdatum anzeigen**. Mit dieser Option wandeln Sie den Block in den **Änderungsdatum**-Block um. • **Erweitert** hält wieder die Option **Zusätzliche CSS-Klasse(n)** bereit. • Im **Stile**-Tab können Sie **Farbe**, **Typografie**, **Größe** und **Ränder** anpassen.

Änderungsdatum	Ist eine Variation des Datum-Blocks und zeigt das letzte Änderungsdatum eines Inhalts an. Wird nur angezeigt, wenn ein Inhalt nach der Veröffentlichung geändert wurde. Die Optionen und Einstellungen sind mit denen des Datum-Blocks identisch, abgesehen von der Block-Toolbar-Option zum Setzen des Datums, die es hier natürlich nicht gibt. 

Kategorien	Zeigt die dem aktuellen Beitrag zugeordneten Kategorien an.  Buttons: • Textausrichtung (links, zentriert, rechts) • Textformatierungsoptionen wie **Fett** und **Kursiv** Seitenleiste: • Der Block kann ganz oben in den Schlagwörter-Block umgewandelt werden – beide Blöcke sind Variationen eines Blocks. • Sie können aus unterschiedlichen **Stilen** wählen und Optionen in den Bereichen **Farbe**, **Typografie**, **Größe** und **Ränder** anpassen. • Unter **Erweitert** gibts wieder ein Feld für zusätzliche CSS-Klassen, aber auch eins für das Trennzeichen, das zwischen den Kategorien angezeigt wird.

Schlagwörter	Zeigt die Schlagwörter eines Beitrags an, ist in Optionen und Einstellungsmöglichkeiten mit dem Kategorien-Block identisch. 
Nächster Beitrag	Gibt einen Link zum nächsten Beitrag aus.  Buttons: • Option zur Änderung der Textausrichtung • **Fett** und **Kursiv**

Nächster Beitrag (Forts.)	Seitenleiste: • Sie können den Block in den Block **Vorheriger Beitrag** umwandeln, es handelt sich wieder um Variationen des gleichen Blocks, wie bei den beiden Datum-Blöcken. • **Den Titel als Link anzeigen** ermöglicht es, den Beitragstitel mit im Link anzuzeigen. • Sie können dem Link einen **Pfeil** folgen lassen. • **Erweitert**: **Zusätzliche CSS-Klasse(n)** sowie die Möglichkeit **Nach Taxonomie filtern**, womit Sie auf den nächsten Beitrag mit denselben Taxonomiebegriffen verweisen können. • Im **Stile**-Tab können Sie Einstellungen für **Farbe** und **Typografie** verändern.
Vorheriger Beitrag	Stellt einen Link zum vorherigen Beitrag dar und bietet dieselben Einstellungsmöglichkeiten wie der Block **Nächster Beitrag**. 

Weiterlesen	Block zur Anzeige des Links zur Einzelansicht eines Beitrags, einer Seite oder eines anderen Inhaltstyps.  Buttons: Möglichkeiten zur Textformatierung Seitenleiste: • **Einstellungen** • **In einem neuen Tab öffnen** sollte, wie schon erwähnt, nur in Ausnahmefällen genutzt werden, wenn etwa sonst Formulareingaben durch einen Seitenwechsel verloren gehen könnten. • **Erweitert** • **Zusätzliche CSS-Klasse(n)**: bekannt von anderen Blöcken • Im **Stile**-Tab haben Sie Optionen für **Farbe**, **Typografie**, **Größe** und **Ränder**.

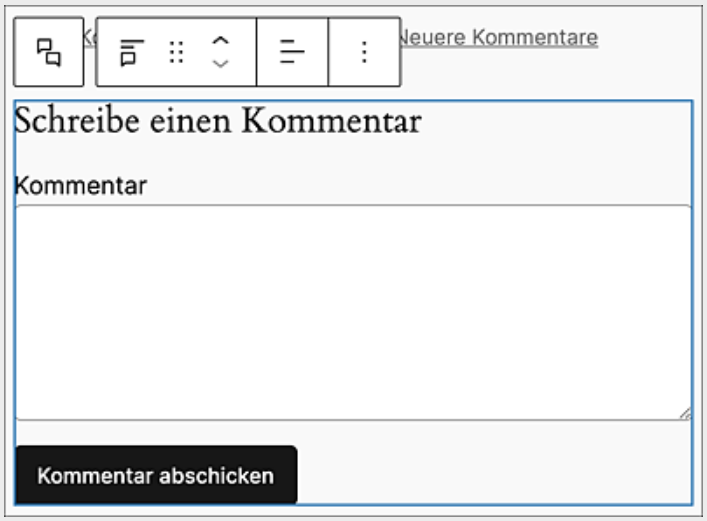
Kommentare	Wrapper-Block für die Anzeige der Kommentare zu einem Inhalt.  Buttons: Option zur **Ausrichtung** des Blocks Seitenleiste: • **Erweitert** mit den CSS-Klassen und der Option, das **HTML-Element** zu ändern • Im **Stile**-Tab können Sie Optionen für die Bereiche **Farbe**, **Typografie** und **Größe** anpassen.

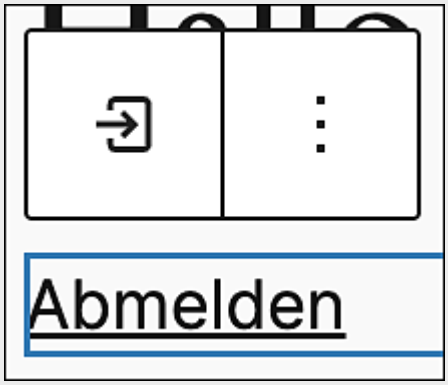
Kommentar-Template (nur im Kommentare-Block)	Ist wie der Block **Beitrags-Template** dafür zuständig, den Inhalt eines Kommentars anzuzeigen. Die Blöcke innerhalb des Kommentar-Template-Blocks werden für jeden Kommentar ausgegeben. Buttons: **Ausrichtung**-Option zur Änderung der Block-Ausrichtung Seitenleiste: • Stiloptionen für **Typografie**, **Größe** und **Ränder** sowie die CSS-Klassen-Option • Zeigt den Titel für den Kommentarbereich an.
Kommentartitel (nur im Kommentare-Block)	Buttons: • Option zur Block-**Ausrichtung** sowie zur Textausrichtung • Möglichkeit, die Überschriftenebene zu wechseln Seitenleiste: • Sie können optional als Teil der Überschrift den **Beitragstitel** sowie die **Anzahl der Kommentare anzeigen**. • **Erweitert** • **Zusätzliche CSS-Klasse(n)** • Im **Stile**-Tab finden Sie Optionen für **Farbe**, **Typografie**, **Größe** und **Ränder**.

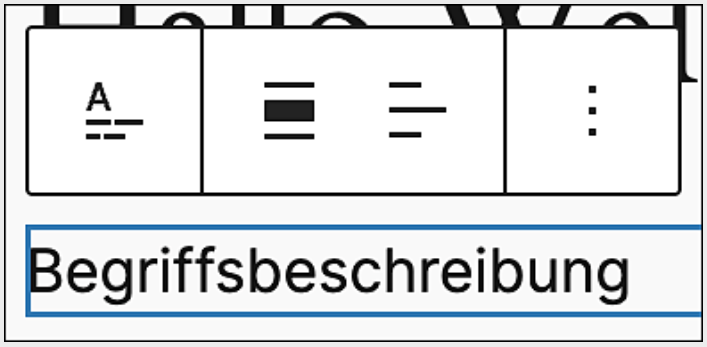
Kommentar-Autorennamen (nur im Kommentar-Template-Block)	Zeigt den Namen der Person an, die den Kommentar geschrieben hat. Buttons: Option zur Textausrichtung Seitenleiste: • **Link zur Autoren-URL**, um den Namen zu der URL zu verlinken, die im Kommentarformular angegeben wurde. Ist die Option aktiv, können Sie den Link **In einem neuen Tab öffnen** lassen. • **Erweitert** bietet erneut die zusätzlichen CSS-Klassen an. • Im **Stile**-Tab gibt es Optionen für **Farbe**, **Typografie**, **Größe** und **Ränder**.

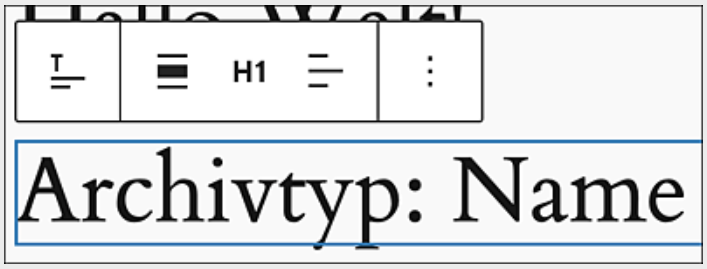
Kommentar-Datum (nur im Kommentar-Template-Block)	Block zur Anzeige des Kommentardatums Buttons: – Seitenleiste: • Sie können das **Standardformat** nutzen oder ein eigenes Format wählen und das Datum als **Link zum Kommentar** anzeigen. • Im **Erweitert**-Panel können Sie **Zusätzliche CSS-Klasse(n)** setzen. • Im **Stile**-Tab finden Sie Einstellungsmöglichkeiten für **Farbe**, **Typografie**, **Größe** und **Ränder**.
Kommentar-bearbeiten-Link (nur im Kommentar-Template-Block)	Zeigt für angemeldete User einen Bearbeiten-Link an, falls die Berechtigung zum Bearbeiten von Kommentaren gegeben ist. Buttons: Button zur Textausrichtung Seitenleiste: • Sie können den Link **In einem neuen Tab öffnen** und dem Block-Wrapper zusätzliche CSS-Klassen mitgeben. • Im **Stile**-Tab können Sie **Farbe**, **Typografie**, **Größe** und **Ränder** anpassen.

Kommentar-Inhalt (nur im Kommentar-Template-Block)	Block zur Darstellung des Kommentar-Inhalts. Buttons: Textausrichtung Seitenleiste: Stil-Optionen für **Farbe**, **Typografie**, **Größe** und **Ränder** sowie die Möglichkeit für CSS-Klassen im **Erweitert**-Panel
Kommentar-Antwort-Link (nur im Kommentar-Template-Block)	Zeigt den Link an, mit dem auf einen Kommentar geantwortet werden kann. Buttons: Textausrichtung Seitenleiste: wie beim Kommentar-Inhalt-Block
Seitennummerierung der Kommentare (nur im Kommentare-Block)	Wrapper-Block für die Blöcke zur Anzeige der Paginierung, wenn die Kommentarliste in mehrere Seiten umgebrochen wird. Bietet genau dieselben Optionen wie der **Seitennummerierung**-Block im **Abfrage-Loop**-Block.

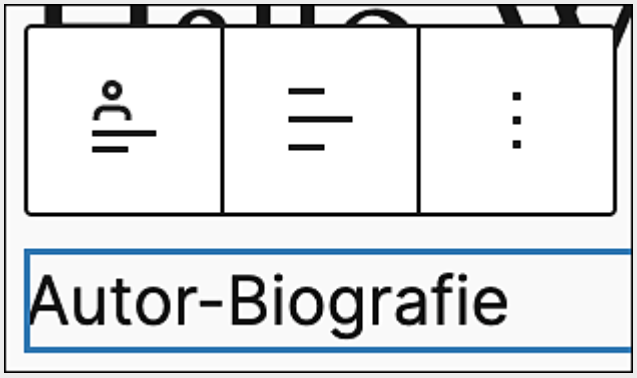
Vorherige und **nächste Seite der Kommentare**, **Kommentar-Seitennummern** (nur im Seitennummerierung-der-Kommentare-Block)	Blöcke zur Darstellung der einzelnen Paginierungsbestandteile. Haben dieselben Optionen wie die entsprechenden Blöcke für die Paginierung im »Abfrage-Loop«-Block, mit dem Unterschied, dass bei den Kommentar-Seitennummern nicht eingestellt werden kann, wie viele Seiten links und rechts neben der aktuellen Seite angezeigt werden sollen.
Formular für Kommentare	Block zur Darstellung des Formulars, mit dem Ihre Besucher*innen Kommentare schreiben können.  Buttons: Textausrichtungsoption Seitenleiste: Stil-Einstellungen für **Farbe**, **Typografie**, **Größe** und **Ränder** sowie die Option **Zusätzliche CSS-Klasse(n)** im **Erweitert**-Panel

An-/abmelden	Zeigt je nach Anmeldestatus der Besucher*in einen Anmelde- oder Abmelde-Link an.  Buttons: – Seitenleiste: • Sie können optional die **Anmeldung als Formular anzeigen** und einstellen, ob nach der An- oder Abmeldung wieder zur aktuellen URL weitergeleitet werden soll. • Unter **Erweitert** gibt es die altbekannte CSS-Klassen-Option. • Im **Stile**-Tab können Sie Einstellungen in den Bereichen Farbe, **Typografie**, **Größe** und **Ränder** vornehmen.

Begriffsbeschreibung	Zeigt die Beschreibung für Kategorien, Schlagwörter oder andere Taxonomien an, wenn die Archiv-Übersicht aufgerufen wird und eine Beschreibung hinterlegt ist.  Buttons: Optionen zur Block- und Textausrichtung Seitenleiste: Stil-Optionen für **Farbe**, **Typografie**, **Größe**, **Ränder** und die Option für zusätzliche CSS-Klassen

Archiv-Titel	Zeigt den Titel eines Archivs an.  Buttons: Optionen für Block- und Textausrichtung sowie die Überschriftenebene Seitenleiste: • Sie können **Den Archivtyp im Titel anzeigen** und **Zusätzliche CSS-Klasse(n)** angeben. • Im **Stile**-Tab gibt es Optionen für **Farbe**, **Typografie**, **Größe** und **Ränder**.

Titel der Suchergebnisse	Block für den Titel der Suchergebnisseite.  Buttons: wie beim **Archiv-Titel**-Block Seitenleiste: Sie können **Den Suchbegriff im Titel anzeigen**, zusätzliche CSS-Klassen angeben und im **Stile**-Tab Optionen für **Farbe**, **Typografie**, **Größe** und **Ränder** nutzen.

Biografie des Autors	Zeigt die Biografie des Autors aus den Profil-Einstellungen an.  Buttons: Textausrichtung Seitenleiste: Sie können Stil-Optionen für **Farbe**, **Typografie**, **Größe** und **Ränder** nutzen und im **Erweitert**-Panel zusätzliche CSS-Klassen angeben.

Einbettungen

Einbettungen verschiedener Plattformen	In der **EINBETTUNGEN**-Kategorie gibt es einen generischen **EINBETTUNGEN**-Block und viele Blöcke, um Inhalte von bestimmten Plattformen einzubinden, wie etwa YouTube oder Vimeo. Wenn Sie einen der Blöcke einbinden, bekommen Sie ein URL-Feld angeboten, in das Sie die URL des Inhalts einbauen. Buttons: Block-Ausrichtung und Änderung der URL Seitenleiste: je nach Plattform gegebenenfalls unterschiedliche Optionen, bei YouTube zum Beispiel die Möglichkeit, immer dasselbe Seitenverhältnis für das eingebettete Video beizubehalten

Tabelle 6.1 Die Standard-Blöcke von WordPress im Überblick

Jetzt haben Sie einen Überblick darüber, was der Block-Editor standardmäßig alles für Möglichkeiten bietet. Das Block-Angebot lässt sich über Plugins natürlich wie alle Aspekte in WordPress erweitern, und die Block-Auswahl von WordPress selbst wird auch immer größer.

Tipp: Inhalte aus anderen Programmen einfügen

Wenn Sie Ihre Texte lieber in anderen Programmen wie Word oder Google Docs schreiben, ist das natürlich problemlos möglich. Wenn Sie nicht gerade sehr spezielle Formatierungsoptionen nutzen, können Sie auch einfach probieren, den Text ohne

besondere Vorkehrung in den Block-Editor einfügen. Mindestens aus Google Docs funktioniert die Übernahme der Überschriften, Verlinkungen, Listen et cetera sehr gut.

Wenn es Probleme gibt, können Sie Text mit Strg/cmd + ⇧ + V einfügen, sodass die Formatierungen entfernt werden und nur der reine Text eingefügt wird.

6.1.4 Verkleinern-Funktion nutzen

In WordPress 6.7 wurde die sogenannte »Verkleinern«-Funktion eingefügt, die die Arbeit mit Block-Vorlagen im Editor vereinfachen soll. Dafür gibt es in der Editor-Toolbar auf großen Viewports ein Vergrößern-Icon, mit dem die Ansicht eingeschaltet werden kann (siehe Abbildung 6.15).

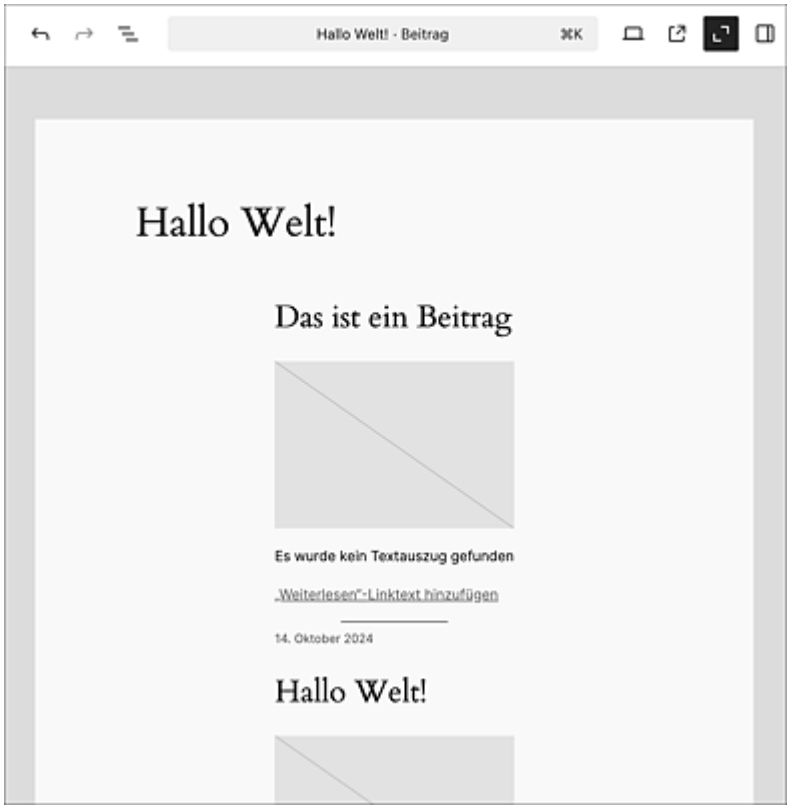


Abbildung 6.15 Wenn die Verkleinern-Funktion aktiv ist, wird der Editor-Bereich verkleinert und nicht mehr über die gesamte verfügbare Breite angezeigt, und Sie können nur noch Blöcke der obersten Ebene auswählen. Wenn Sie den Block-Inserter öffnen, wird direkt der Tab mit den »Vorlagen« ausgewählt.

6.1.5 Zwischenstände speichern

Wenn Sie an einer unveröffentlichten Seite oder einem Beitrag arbeiten, ist es nicht unüblich, nicht den gesamten Inhalt am Stück zu schaffen, und eigentlich ist es sogar eine gute Idee, einen Beitrag oder eine Seite nach der vermeintlichen Fertigstellung noch mal einen Tag liegen zu lassen, um ihn dann mit etwas Abstand erneut durchzulesen.

Egal, was der Grund für eine Pause ist, der aktuelle Stand soll natürlich nicht verloren gehen. Dafür gibt es bei Inhalten, die noch nicht veröffentlicht wurden, oben rechts im Editor den Button ENTWURF SPEICHERN (siehe Abbildung 6.16). Mit einem Klick auf den Button speichern Sie den aktuellen Zwischenstand. Falls Sie Revisionen nicht explizit deaktiviert haben, wird dabei auch eine Revision erstellt (mehr zu Revisionen im weiteren Verlauf des Kapitels).

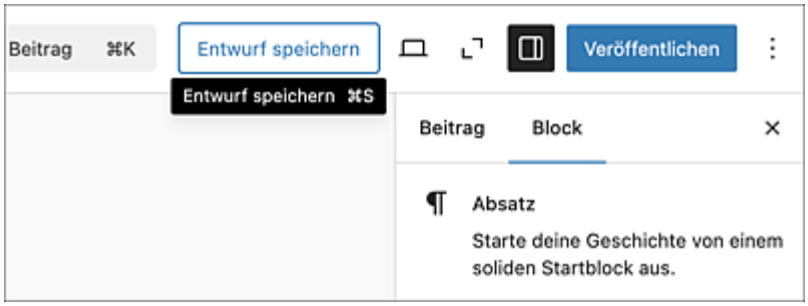


Abbildung 6.16 Sie können auch vor der Veröffentlichung jederzeit einen Zwischenstand mit Klick auf »Entwurf speichern« sichern.

Auch wenn WordPress eine Auto-Speichern-Funktion hat, die so manchen Entwurf gerettet hat, wenn etwa der Browser abgestürzt

ist, ist es eine gute Idee, auch unabhängig von längeren Pausen hin und wieder den aktuellen Stand zu speichern.

6.1.6 Vorschau anzeigen lassen

Wenn Sie sich den Text vor der Veröffentlichung im Frontend anschauen möchten, können Sie die Vorschau-Funktion von WordPress nutzen. Dafür gehen Sie im Editor auf den Button mit dem Laptop-Icon und klicken im Overlay den Punkt Vorschau in neuem Tab an (siehe Abbildung 6.17).

Wenn Sie einen Blick in die Adresszeile Ihres Browsers werfen, werden Sie sehen, dass die URL am Ende ein paar URL-Parameter enthält, sodass die URL einer Vorschau etwa so aussehen kann: *example.com/?p=137&preview=true*. Die Zahl, die für den *p*-Parameter angegeben ist, ist die ID des Beitrags in der Datenbank, und *preview=true* gibt an, dass es sich um eine Vorschau handelt.



Abbildung 6.17 Sie können sich jederzeit vor der Veröffentlichung den aktuellen Stand eines Beitrags, einer Seite oder eines anderen Inhalts über die Vorschau im Frontend anschauen.

6.1.7 Bilder/Dateien einfügen

Wenn Sie ein Bild in Ihren Beitrag oder Ihre Seite einfügen möchten, gibt es dafür den Bild-Block, den Sie eben schon in der Liste aller

Blöcke gesehen haben. So gehen Sie vor, um ein Bild einzufügen:

1. Wenn Sie per Drag-and-drop arbeiten möchten, können Sie einfach direkt ein Bild von Ihrem lokalen Dateisystem in den Editor ziehen und an der gewünschten Position loslassen. Dadurch wird das Bild in die Mediathek hochgeladen und ein Bild-Block mit dem Bild eingefügt.

   Andernfalls fügen Sie manuell einen Block vom Typ Bild in den Editor ein. Dazu können Sie entweder eins der Plus-Symbole im Editor nutzen oder den /-Befehl in einem leeren Absatz-Block. Nutzen Sie im eingefügten Bild-Block den Hochladen-Button, um ein Bild von Ihrem lokalen System hochzuladen, oder fügen Sie ein Bild aus der Mediathek oder von einer beliebigen URL ein. Direkt nach dem Einfügen könnten Sie alternativ auch noch per Drag-and-drop ein Bild auf den Block ziehen, um es hochzuladen und dem Block zuzuweisen (siehe Abbildung 6.18).



Abbildung 6.18 Sie können einfach ein Bild von Ihrem lokalen System auf den Bild-Block ziehen, um es in die Mediathek hochzuladen und im Block anzuzeigen.

2. Fügen Sie optional über den Block-Toolbar-Button Beschriftung hinzufügen (links neben Ersetzen) eine Beschriftung ein.
3. *Dringend empfohlen:* Schreiben Sie einen Alternativtext über das Feld in der Block-Seitenleiste, damit auch Menschen mit Sehbehinderung einen Eindruck vom Bildinhalt bekommen

können. Außerdem wird der Text angezeigt, wenn das Bild nicht geladen werden kann.

Jetzt können Sie sich die Vorschau anschauen und prüfen, ob Ihnen gefällt, wie das Bild angezeigt wird. Standardmäßig wird das Bild so groß dargestellt, wie es ist, es reicht also – wenn das Bild breit genug ist – über die gesamte Inhaltsbreite. Wenn Ihnen das nicht gefällt, können Sie im Backend mit den in der Block-Tabelle vorgestellten Block-Optionen die Breite ändern, das Bild bei Bedarf zuschneiden (siehe Abbildung 6.19) et cetera.

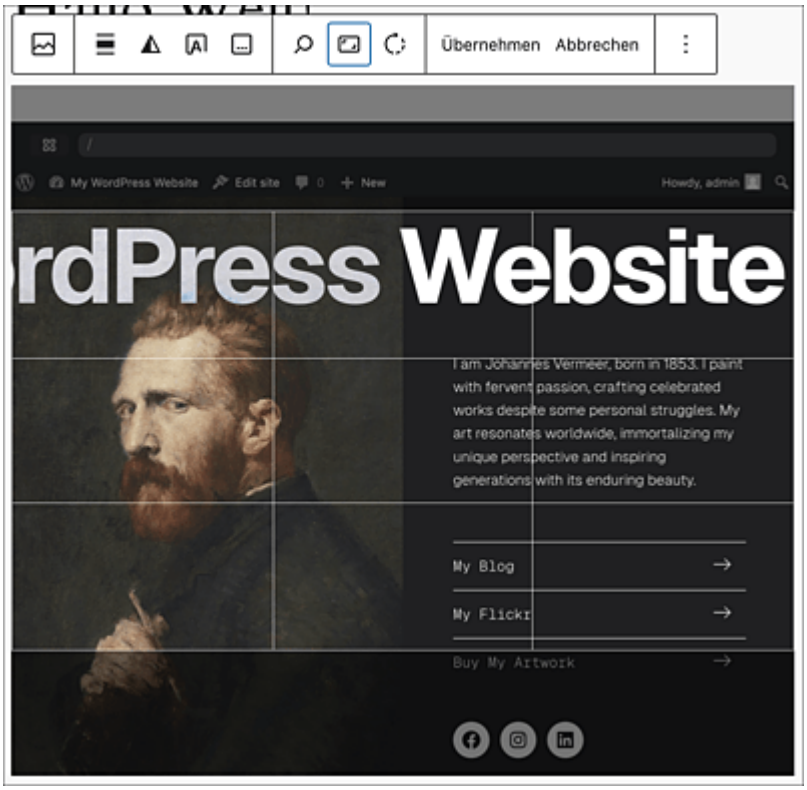


Abbildung 6.19 Beim Bild-Block bietet WordPress ein grundlegendes Werkzeug an, um das Bild zuzuschneiden und zu drehen.

Ein gängiges Verhalten ist, dass Bilder bei einem Klick auf das Bild in einer sogenannten *Lightbox* als vergrößerte Version in einem Overlay angezeigt werden. Diese Funktion gibt es mittlerweile direkt in WordPress integriert, Sie können sie bei einem Bild-Block in der

Werkzeugleiste über das Link-Symbol aktivieren (siehe Abbildung 6.20).

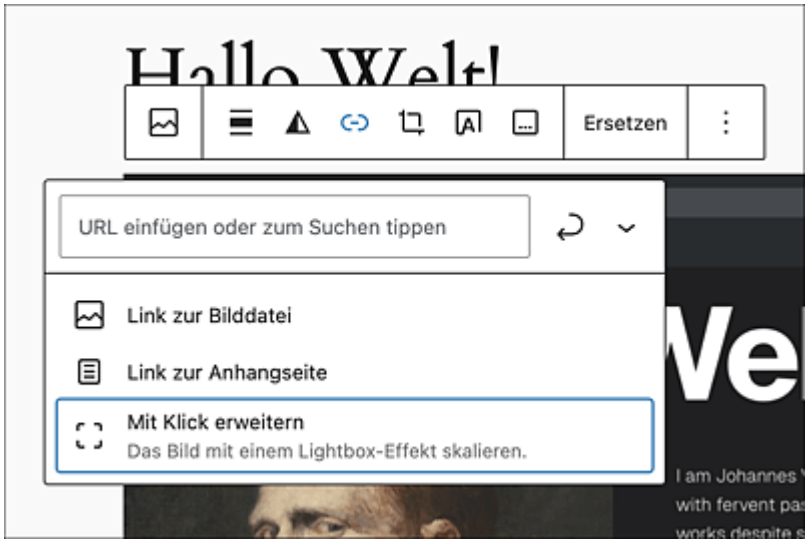


Abbildung 6.20 Sie können einstellen, dass das Bild bei Klick in einer Lightbox dargestellt wird.

6.1.8 Beitragsbild festlegen

Das Beitragsbild ist ein Bild, das Sie über die Dokument-Sidebar auf der rechten Seite festlegen können, die Option ist direkt mit oben angesiedelt (siehe Abbildung 6.21). Beitragsbilder werden häufig von Themes auf Übersichtsseiten genutzt, um sie bei dem Teaser für den Beitrag mit anzuzeigen. Oft werden sie auch auf der Einzelansicht angezeigt, das muss aber nicht unbedingt sein.

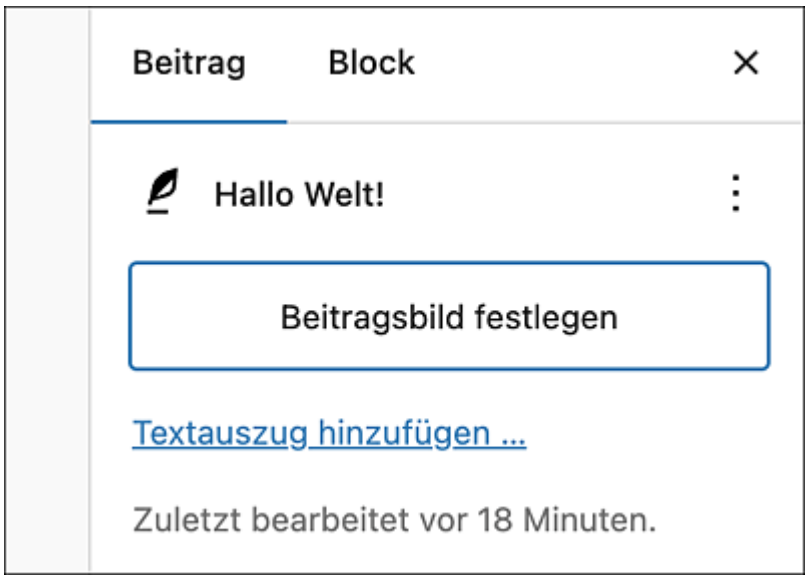


Abbildung 6.21 In der Dokumenten-Seitenleiste können Sie ein Beitragsbild festlegen, das vom Theme an unterschiedlichen Stellen genutzt werden kann, etwa in der Beitragsübersicht.

Nutzen Sie diese Funktion, wenn es Sinn ergibt, aber versuchen Sie nicht auf Biegen und Brechen, für jeden Beitrag ein Beitragsbild zu finden, und auch Stock-Material sollten Sie vermeiden. Wenn es kein sinnvolles Beitragsbild zu einem Beitrag gibt, dann ist das so, und die Seite lädt nebenher noch schneller.

6.1.9 Einen Inhalt veröffentlichen

Ist Ihr Beitrag oder Ihre Seite fertig, geht es an die Veröffentlichung (denken Sie vorher daran, den Permalink noch mal zu prüfen, wie am Anfang des Kapitels beschrieben). Der entsprechende Button Veröffentlichen ist oben rechts bei den Buttons zum Speichern und für die Vorschau zu finden (siehe Abbildung 6.22). Bevor Sie ihn aber anklicken, haben Sie ein paar verschiedene Möglichkeiten.



Abbildung 6.22 Schwer zu übersehen: der Button zur Veröffentlichung Ihres Beitrags oder Ihrer Seite

Sofort für alle sichtbar veröffentlichen

Wenn Sie den Inhalt direkt veröffentlichen und für alle sichtbar online stellen wollen, können Sie direkt auf den VERÖFFENTLICHEN-Button klicken. Vor der Veröffentlichung wird standardmäßig noch ein Prüfdialog angezeigt, der Ihnen ein paar wichtige Infos anzeigt, und zwar die Sichtbarkeit des Inhalts nach Veröffentlichung und wann die Veröffentlichung passieren soll (siehe Abbildung 6.23).

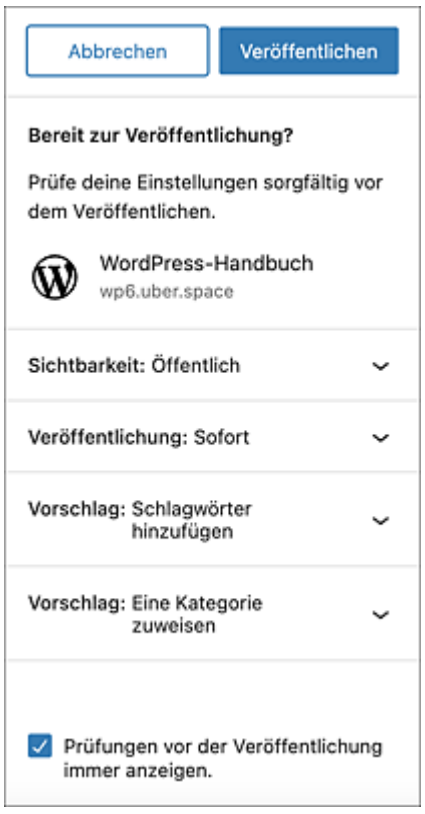


Abbildung 6.23 Vor der Veröffentlichung wird ein Dialog angezeigt, der für Sie zusammenfasst, wie der Inhalt veröffentlicht wird.

Gegebenenfalls finden Sie hier noch weitere Punkte, wie den Vorschlag, eine Kategorie zuzuweisen, wenn das noch nicht geschehen ist. Ganz unten im Dialog können Sie die Prüfung ausschalten.

Veröffentlichung zu einem späteren Zeitpunkt planen

Wenn Sie den Beitrag oder die Seite erst später veröffentlichen möchten, können Sie in der Dokumenten-Seitenleiste beim Punkt VERÖFFENTLICHEN auf die Standard-Einstellung SOFORT klicken, woraufhin sich ein Overlay mit Einstellungsmöglichkeiten für Tag und Zeit öffnet, wann der Eintrag online gehen soll (siehe Abbildung 6.24). Theoretisch können Sie so einen Beitrag oder eine Seite auch zurückdatieren, aber hier geht es jetzt um einen Zeitpunkt in der Zukunft.

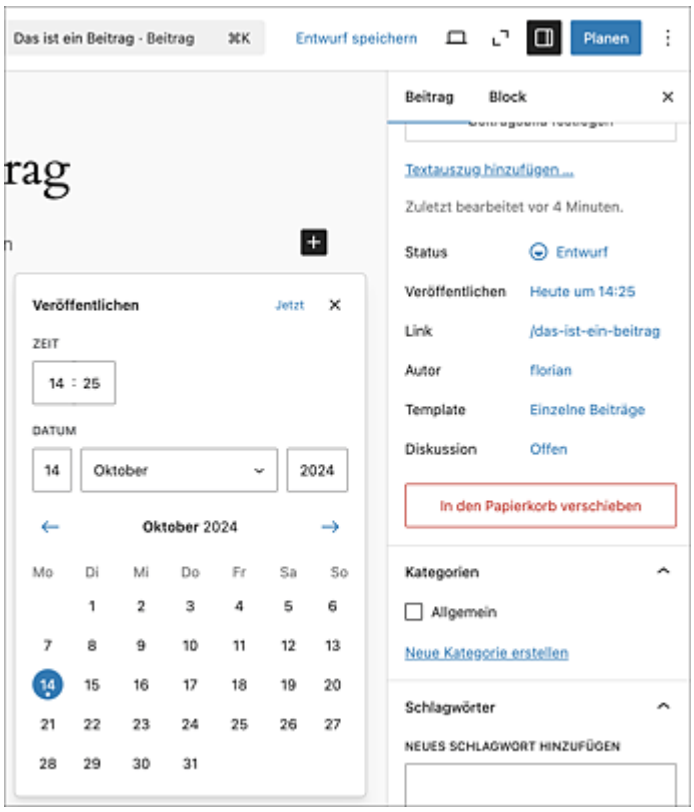


Abbildung 6.24 Sie können einen Inhalt auch für die Veröffentlichung planen, sodass er zu einem bestimmten Zeitpunkt online geht. Rechts oben ist die Beschriftung des Buttons dann nicht mehr »Veröffentlichen«, sondern »Planen«.

Stellen Sie beispielsweise ein, dass der Eintrag oder die Seite in fünf Minuten veröffentlicht werden soll, sehen Sie sofort, während Sie die Zeit verändern, dass oben rechts der Veröffentlichen-Button zu Planen wird (siehe Abbildung 6.24).

Nach einem Klick auf Planen taucht dieselbe Prüfung auf wie bei der sofortigen Veröffentlichung.

Status und Sichtbarkeit einstellen

Ein Inhalt in WordPress kann vor der Veröffentlichung unterschiedliche Status haben, und auch nach der Veröffentlichung gibt es nicht nur die Option »sichtbar für alle«. Um den Status zu verändern, gehen Sie in der Dokumenten-Seitenleiste über dem

Punkt Veröffentlichen auf den aktuellen Status des Inhalts: Standardmäßig ist der Status Entwurf. Nach einem Klick öffnet sich ein Overlay, in dem Sie Status und Sichtbarkeit einstellen können (siehe Abbildung 6.25).

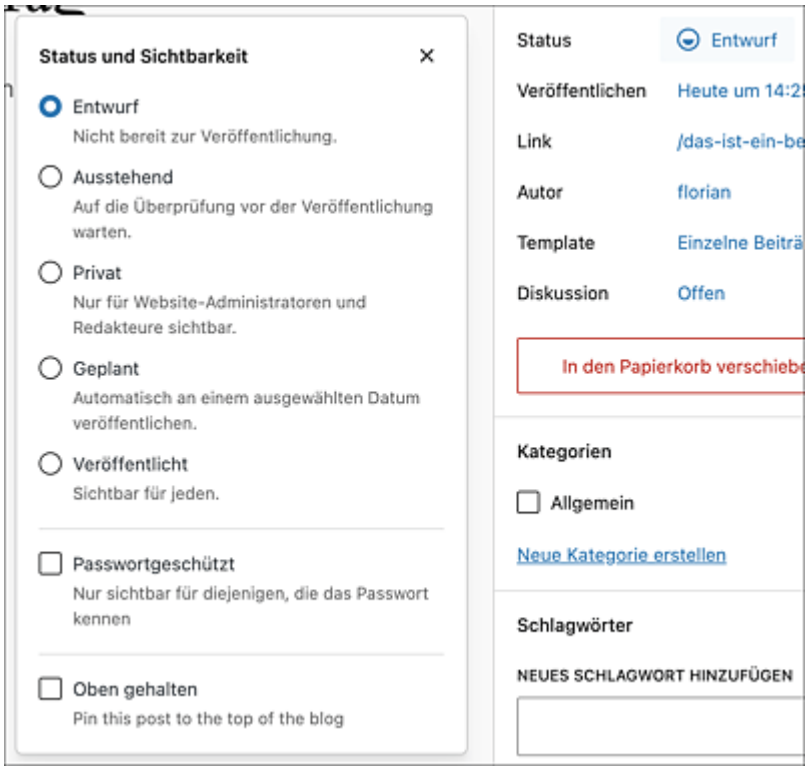


Abbildung 6.25 Beiträge und Seiten können unterschiedliche Status haben.

Sie haben folgende Möglichkeiten:

- Entwurf bedeutet, dass der Inhalt noch in Arbeit ist. Sie können einen Inhalt auch nach Veröffentlichung wieder in den Entwurfszustand bringen und dadurch die Veröffentlichung aufheben. Seien Sie sich dabei nur bewusst, dass – wenn die URL bereits bei anderen bekannt ist, etwa durchs Teilen auf Social Media – diese URL zu einer Fehlermeldung führt, wenn die URL nicht weitergeleitet wird.
- Ausstehend gibt an, dass noch Dinge vor der Veröffentlichung zu erledigen sind. Dieser Status ergibt wohl am meisten Sinn, wenn

mehrere Personen an einem Blog arbeiten und gegenseitig die Inhalte kontrollieren oder für unterschiedliche Aspekte eines Beitrags verantwortlich sind.

- Privat ist ein Zustand, in dem ein Inhalt zwar veröffentlicht ist (er also über die richtige URL ohne *preview*-Parameter oder Ähnliches aufgerufen werden kann), aber nur Administrator*innen und Redakteur*innen darauf zugreifen können. Im Frontend steht vor dem Titel des Inhalts »Privat:« (siehe Abbildung 6.26).



Abbildung 6.26 Ein privater Inhalt wird mit einem »Privat:«-Präfix vor dem Titel gekennzeichnet.

- Geplant ist der Status für die automatische Veröffentlichung zu einem späteren Zeitpunkt. Wenn Sie die Option wählen, erscheint unten im Overlay die Option zum Auswählen des Veröffentlichungszeitpunkts.
- Veröffentlicht ist der normale Status, wenn ein Inhalt online geht: sichtbar für alle.
- Passwortgeschützt: Bei allen Optionen außer Privat können Sie unten über eine Checkbox den Inhalt Passwortgeschützt machen. Wenn die Checkbox angehakt ist, erscheint darunter ein Eingabefeld, in dem Sie ein Passwort eingeben können. Dieses Passwort muss beim Aufrufen des Inhalts im Frontend eingegeben werden, bevor der Inhalt angezeigt wird. Der Titel erhält ein »Geschützt:«-Präfix (siehe Abbildung 6.27).



Abbildung 6.27 Bei einem passwortgeschützten Inhalt werden im Frontend nur der Titel und ein Passwortfeld angezeigt. Vor dem Titel wird ein »Geschützt:« eingefügt.

- Oben gehalten. Damit können Sie einstellen, dass der Beitrag immer ganz oben in der Übersicht aller Beiträge angezeigt wird (oder, falls es mehrere Beiträge mit der Option gibt, werden sie nach Datum sortiert oben angezeigt).

Nur für Beiträge: Kategorien und Schlagwörter

Wenn Sie vorhaben, regelmäßig Beiträge zu schreiben, kann es eine gute Idee sein, die Beiträge zu kategorisieren, damit Sie im Frontend Links auf die Artikel einer bestimmten Kategorie anbieten können.

In der Dokumenten-Seitenleiste im Editor finden Sie unten bei den Beiträgen das Panel Kategorien, in dem Sie sowohl bestehende Kategorien auswählen als auch direkt eine Neue Kategorie erstellen können (siehe Abbildung 6.28).

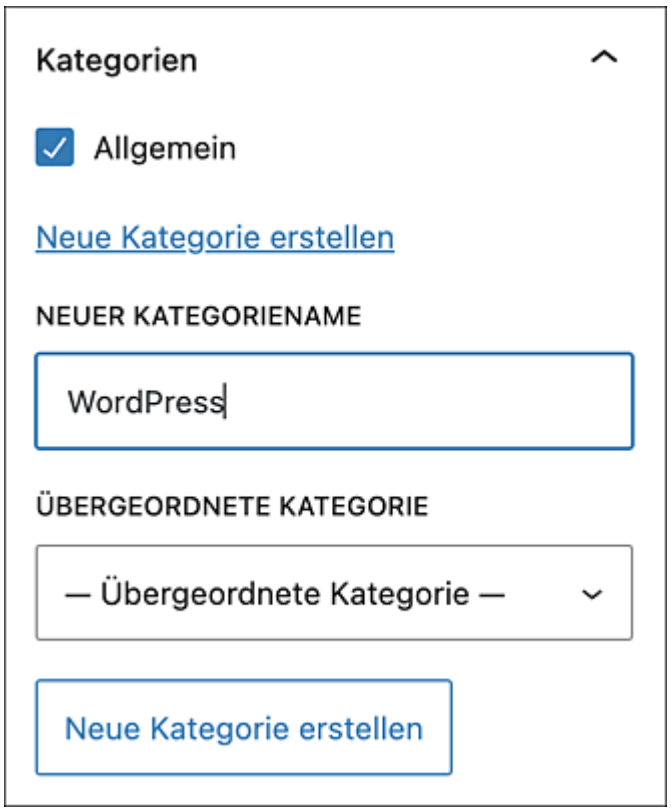


Abbildung 6.28 Kategorien können Sie nutzen, um Ihre Inhalte thematisch zu gruppieren. Sie können auch Unterkategorien anlegen.

Gleiches gilt für Schlagwörter, die eine zweite Systematik bieten, um Beiträge zu gruppieren. Die Zuordnung funktioniert hier ein bisschen anders: Es gibt ein Eingabefeld, und Sie können einfach durch Komma getrennt Schlagwörter eingeben. Wenn schon Schlagwörter existieren, werden sie beim Tippen zur Autovervollständigung angezeigt (siehe Abbildung 6.29).

Kategorien gibt es meist nur relativ wenige, während Schlagwörter (falls sie genutzt werden) oft eher wie Hashtags eingesetzt werden, also um Inhalte genauer zu umschreiben als mit groben Kategorien, weshalb es schnell deutlich mehr werden. Später in diesem Abschnitt geht es auch noch genauer um die Verwaltung von Kategorien und Schlagwörtern.

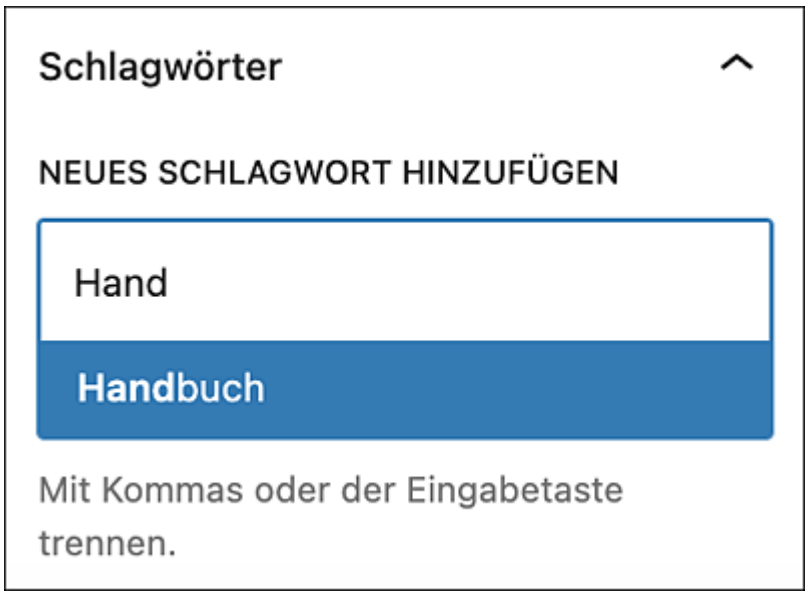


Abbildung 6.29 Schlagwörter geben Sie in ein Eingabefeld ein. Wenn bereits Schlagwörter existieren, wird Ihnen beim Tippen eine Autovervollständigung angezeigt.

Nur für Beiträge: Kommentare

Auch wenn sich heutzutage oft in Kommentarbereichen nicht mehr viel abspielt, schadet es ja nicht, die Möglichkeit trotzdem anzubieten. Außer wenn Sie im Admin-Menü unter Einstellungen • Diskussion eingestellt haben, dass Kommentare nicht aktiv sind, steht in der Dokumenten-Seitenleiste beim Punkt Diskussion standardmäßig Offen.

Wenn Sie auf den Punkt klicken, öffnet sich ein Overlay, in dem Sie den Status auf Geschlossen ändern und unabhängig davon Pingbacks und Trackbacks aktivieren können (sie sind standardmäßig aktiv). Pingbacks und Trackbacks sind Möglichkeiten, andere Websites darüber zu informieren, dass auf der eigenen Site beispielsweise ein Beitrag verlinkt wurde. Pingbacks werden automatisch gesendet.

Template ändern

Über den Punkt Template können Sie für einen Inhalt ein anderes Template auswählen. Bei Themes mit einer Sidebar ist es beispielsweise typisch, dass ein Template ohne Seitenleiste mitgeliefert wird. Neben der Option, ein vorhandenes Template auszuwählen, können Sie auch ein individuelles Template erstellen. Mehr dazu erfahren Sie in Kapitel 10.

Über den Punkt Template anzeigen können Sie die Ansicht im Editor umschalten, sodass der aktuelle Inhalt im ausgewählten Template angezeigt wird, also mit Header, Footer, Kommentarbereich et cetera (siehe Abbildung 6.30). Von hier können Sie auch in die Bearbeitungsansicht des Templates gelangen, wenn Sie versuchen, einen Block außerhalb des Inhalts zu bearbeiten.

Nur Beiträge: Textauszug

Über den Button Textauszug hinzufügen direkt unter der Beitragsbild-Option können Sie eine Kurzfassung des Textes hinzufügen (siehe Abbildung 6.31), der von Themes etwa in Beitragsübersichten angezeigt werden kann, wenn kein Mehr-Block eingesetzt wird (oder explizit der Textauszug angefragt wird).

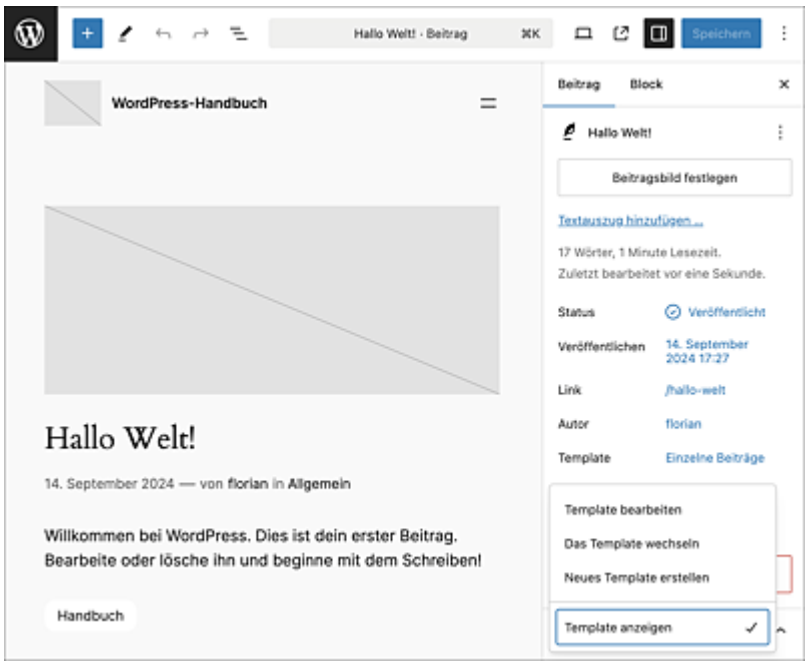


Abbildung 6.30 Wenn Sie »Template anzeigen« anklicken, wird der Inhalt so angezeigt, wie er im Template eingebunden ist. Sie können ihn in dieser Ansicht auch weiterbearbeiten.

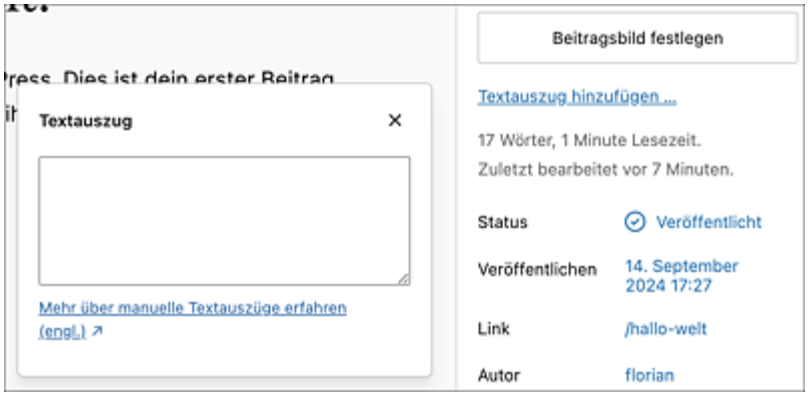


Abbildung 6.31 Der Textauszug ist meist ein kleiner Teaser-Text, der Lust auf den Beitrag machen soll.

6.2 Beiträge/Seiten verwalten

Nachdem Sie die Bearbeitung eines einzelnen Beitrags beziehungsweise einer einzelnen Seite kennengelernt haben, geht es jetzt eine Ebene höher um die Verwaltung von Beiträgen beziehungsweise Seiten.

6.2.1 Liste der Beiträge/Seiten anschauen

Der Einstiegspunkt in die Verwaltung sind die Übersichtsseiten für Beiträge beziehungsweise Seiten. Die erreichen Sie mit einem Klick auf den Punkt Beiträge/Seiten im linken Admin-Menü oder über die jeweiligen Unterpunkte Alle Beiträge und Alle Seiten (siehe Abbildung 6.32).

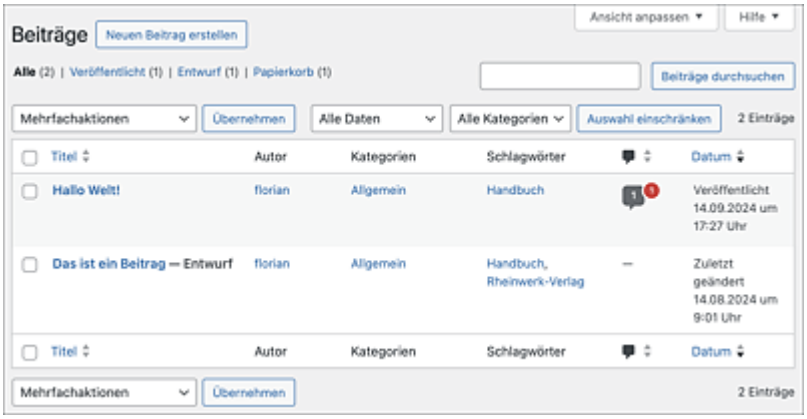


Abbildung 6.32 Beiträge und Seiten sind in WordPress sogenannte »Inhaltstypen«, die abgesehen von ein paar Ausnahmen in ähnlichen Übersichtslisten angezeigt werden.

Wenn Sie viele Seiten oder Beiträge haben, gibt es unterschiedliche Optionen, um das gesuchte Element zu finden.

- **Sortierung**
  Einige Spalten, wie die mit dem Titel oder dem Datum, können Sie nutzen, um die Liste danach zu sortieren. Die Sortierung

können Sie auch umkehren. Ob eine Spalte zur Sortierung genutzt werden kann, ist an den beiden Pfeilen neben dem Spaltentitel erkennbar und daran, dass die Spaltentitel ein Link sind.

- **Filter**
  Ganz oben finden Sie direkt unter dem Titel der Übersicht Links, die Sie die Einträge nach dem Status filtern lassen. So können Sie sich beispielsweise nur die Entwürfe anzeigen lassen. Welche Links konkret angezeigt werden, hängt von Ihren Inhalten ab: Ein Status wird nur angezeigt, wenn es mindestens einen Inhalt mit dem Status gibt. Hinweis: In dieser Liste finden Sie auch den Papierkorb, wenn es gelöschte Elemente gibt. Im Papierkorb können Sie die Elemente wiederherstellen oder endgültig löschen.

  Direkt über der Tabelle gibt es links neben dem Button Auswahl einschränken Dropdown-Felder (bei den Seiten nur eins), über das Sie die Liste auf einen bestimmten Veröffentlichungsmonat (Alle Daten) und bei Beiträgen auf eine bestimmte Kategorie (Alle Kategorien) einschränken können. Wenn Sie Ihre Auswahl getroffen haben, drücken Sie auf den Button Auswahl einschränken, und die Liste wird entsprechend gefiltert.

- **Suche**
  Rechts oben sehen Sie ein Suchformular, mit dem Sie die Beiträge oder Seiten durchsuchen können. Durchsucht wird nicht nur der Titel, sondern der gesamte Inhalt.

Sie können diese Optionen auch kombinieren, also sich beispielsweise nur die Entwürfe anzeigen lassen und diese durchsuchen.

6.2.2 Inhalte bearbeiten (QuickEdit und normal) oder löschen

Wenn Sie den Cursor über einen Eintrag in der Beitrags- oder Seitenliste bewegen, erscheinen verschiedene Aktionslinks (siehe Abbildung 6.33).



Abbildung 6.33 Wenn der Cursor über einem Eintrag in der Beitrags- oder Seitenliste platziert ist, erscheinen verschiedene Aktionslinks.

Der erste davon ist Bearbeiten, der Sie in die normale Bearbeitungsansicht bringt, um die es im vorherigen Abschnitt ging. Um den Block-Editor zu erreichen, müssen Sie aber nicht auf diesen Bearbeiten-Link klicken, sondern können einfach den Titel der Seite oder des Beitrags anklicken.

QuickEdit

Über einen Klick auf QuickEdit öffnet sich direkt in der Übersicht ein Formular, mit dem Sie grundlegende Informationen des Inhalts bearbeiten können (siehe Abbildung 6.34):

- Titel
- Titelform (auch bekannt als Slug, der letzte Teil im Permalink)
- Datum der Veröffentlichung
- Passwort oder Privat, um den Inhalt entweder mit einem Passwort zu schützen oder ihn nur für auf Ihrer Website registrierte

Personen mit mindestens Redakteurs-Rolle zugänglich zu machen

- Kategorien
- Template zur Änderung des Templates, mit dem der Inhalt dargestellt wird; nur sichtbar, wenn es mehrere Templates gibt
- Schlagwörter
- Kommentare erlauben
- Pings erlauben
- Status zur Änderung des Status
- Beitrag oben halten (in der Beitragsübersicht auf der Website)

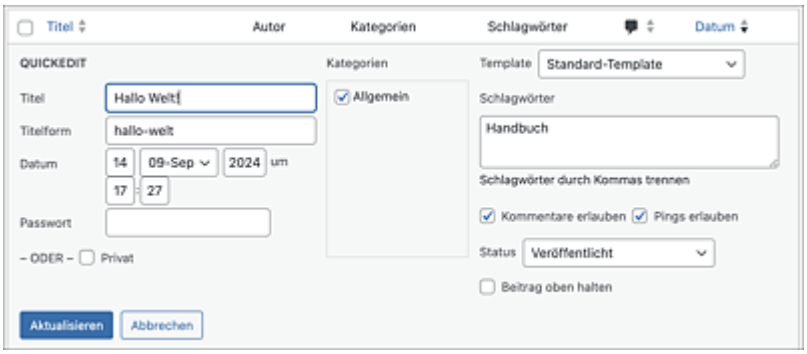


Abbildung 6.34 Über QuickEdit können Sie einige Aspekte eines Inhalts direkt in der Übersicht bearbeiten.

Achtung: Nachdem Sie Änderungen vorgenommen haben, müssen Sie unten auf Aktualisieren klicken.

Mehrfachbearbeitung mit QuickEdit

Mit QuickEdit können Sie auch mehrere Beiträge oder Seiten gleichzeitig bearbeiten. Wählen Sie dafür in der Spalte ganz links mit den Checkboxen mehrere Einträge aus, und wählen Sie oben

(oder unter der Tabelle) im Mehrfachaktionen-Dropdown den Punkt Bearbeiten aus, und klicken Sie danach auf Übernehmen.

Daraufhin öffnet sich ein Formular Mehrfachbearbeitung über den Einträgen in der Tabelle, über das Sie einige der eben besprochenen Punkte für alle ausgewählten Elemente gleichzeitig ändern können, wie den Status oder Kategorien (siehe Abbildung 6.35).

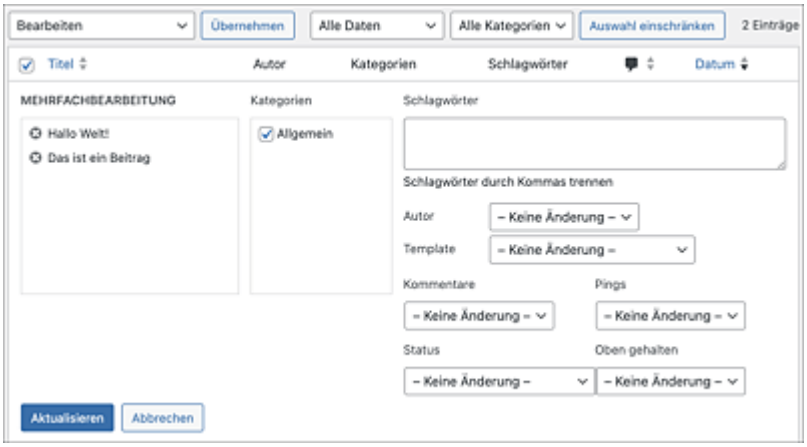


Abbildung 6.35 Sie können mit QuickEdit über die Mehrfachbearbeitung gleichzeitig mehrere Einträge bearbeiten.

Papierkorb

Neben dem Aktionslink QuickEdit sehen Sie den Link Papierkorb, mit dem Sie einen Beitrag oder eine Seite in den Papierkorb verschieben können. Wenn ein oder mehrere Einträge im Papierkorb sind, können Sie den Papierkorb oben über den Link in der Liste der Beitrags-/Seiten-Status aufrufen und einen Eintrag wiederherstellen oder dauerhaft löschen.

Ansehen/Vorschau

Der letzte Aktionslink Ansehen beziehungsweise Vorschau für unveröffentlichte Einträge führt ins Frontend und zeigt die Einzelansicht des entsprechenden Inhalts an.

6.2.3 Kategorien und Schlagwörter verwalten

Beiträge haben in WordPress zwei besondere Funktionen: Sie können Kategorien und Schlagwörtern zugeordnet werden, um sie zu gruppieren. Kategorien sind dabei eher die grobe thematische Einteilung Ihrer Blogbeiträge, während Schlagwörter genauer sind und eventuell auch mal nur für einen Beitrag genutzt werden.

Auf einem Blog rund um WordPress könnten Kategorien beispielsweise »Theme-Vorstellung«, »Plugin-Entwicklung« und »WordCamps« sein, während Schlagwörter »WordCamp Konstanz« und »Debugging« wären, weil die schon relativ genau beschreiben, worum es geht. Stellen Sie sich Schlagwörter wie Hashtags vor, die auf Social Media genutzt werden, um den Inhalt eines Posts zu beschreiben.

In der Bearbeitungsansicht eines Beitrags können Sie Schlagwörter und Kategorien über die Dokumenten-Seitenleiste hinzufügen und auch erstellen, wie Sie im vorherigen Abschnitt gesehen haben. Irgendwann wird die Pflege der Kategorien und Schlagwörter aber ein bisschen unübersichtlich, und es lassen sich auch nicht alle Dinge dort bearbeiten: So können Sie die Begriffe nicht umbenennen oder den Slug ändern.

Übersichtlicher geht es mit den dedizierten Ansichten für Kategorien und Schlagwörter, die Sie über das linke Admin-Menü erreichen: Beiträge • Kategorien beziehungsweise Beiträge • Schlagwörter (siehe Abbildung 6.36).

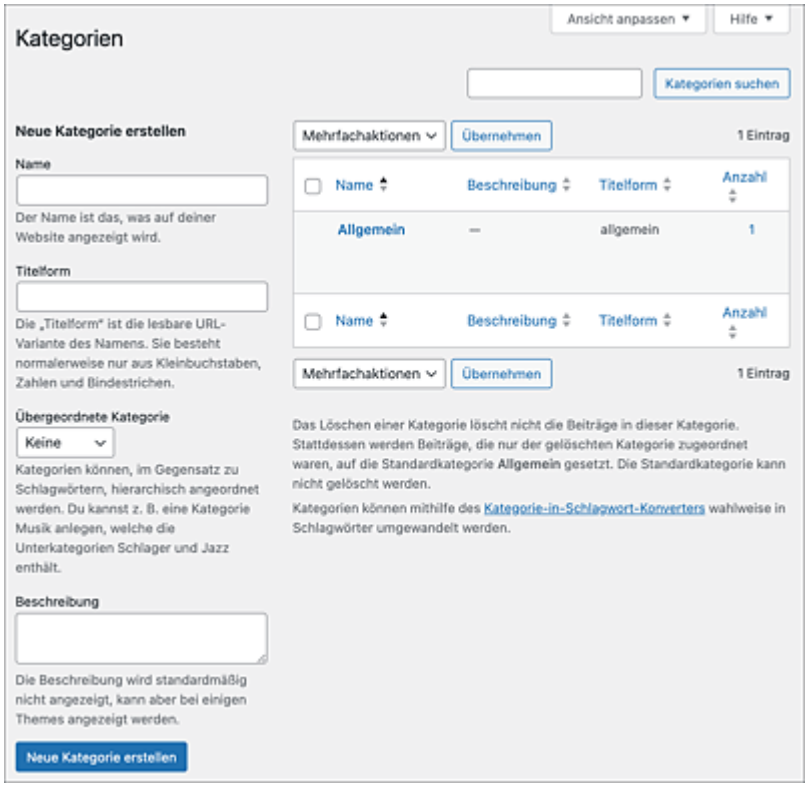


Abbildung 6.36 Unter »Beiträge« • »Kategorien« finden Sie eine Übersicht der Kategorien und können bestehende bearbeiten und neue Kategorien erstellen.

In der sich öffnenden Ansicht sehen Sie rechts die Liste der angelegten Kategorien beziehungsweise Schlagwörter und links ein Formular zum Anlegen neuer Kategorien oder Schlagwörter. Sie können hier die Titelform (den Slug, der beispielsweise in der URL beim Kategorie-/Schlagwort-Archiv angezeigt wird) anpassen sowie eine Beschreibung vergeben – zwei weitere Dinge, die direkt aus dem Editor nicht möglich sind.

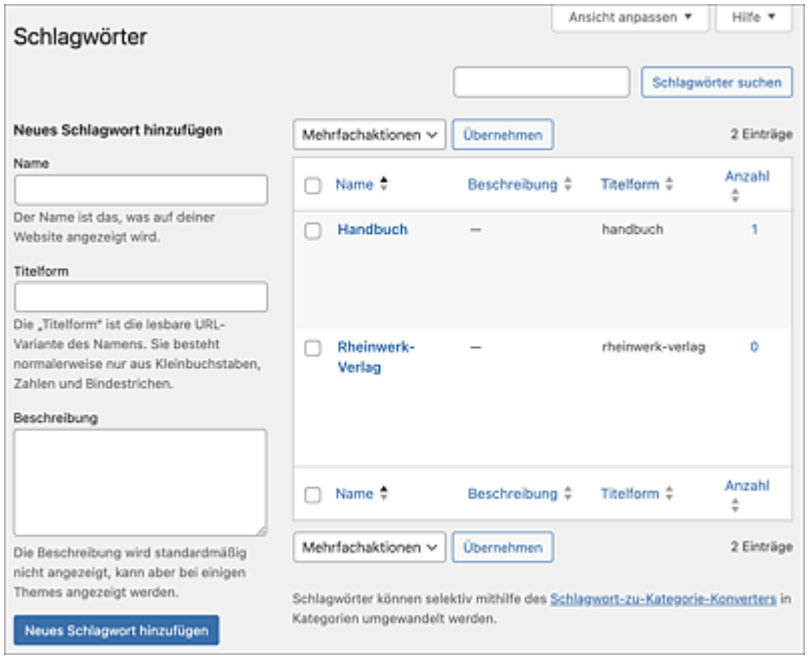


Abbildung 6.37 Die Schlagwörter lassen sich unter »Beiträge« • »Schlagwörter« anschauen und bearbeiten.

6.3 Mit Revisionen arbeiten

Immer, wenn Sie einen Beitrag oder eine Seite speichern, veröffentlichen oder aktualisieren, wird eine *Revision* angelegt (falls Sie die Option nicht deaktiviert haben). Diese Revision ist ein Zwischenstand des Inhalts, zu dem Sie später zurückgehen und den Sie bei Bedarf wiederherstellen können. Im Editor finden Sie die Revisionen in der Dokumenten-Seitenleiste (siehe Abbildung 6.38).

Wenn Sie auf die Anzahl der Revisionen klicken, öffnet sich eine Ansicht, in der unten zwei Versionen des Inhalts nebeneinandergestellt werden. Rechts ist der aktuelle Stand, links der vorherige. Über den Schieberegler oben können Sie zu älteren Revisionen wechseln, wobei Sie dann rechts immer den Stand haben, der wiederhergestellt werden würde, wenn Sie oben rechts auf den Button Diese Revision wiederherstellen klicken, und auf der linken Seite immer die vorherige Revision (siehe Abbildung 6.39).

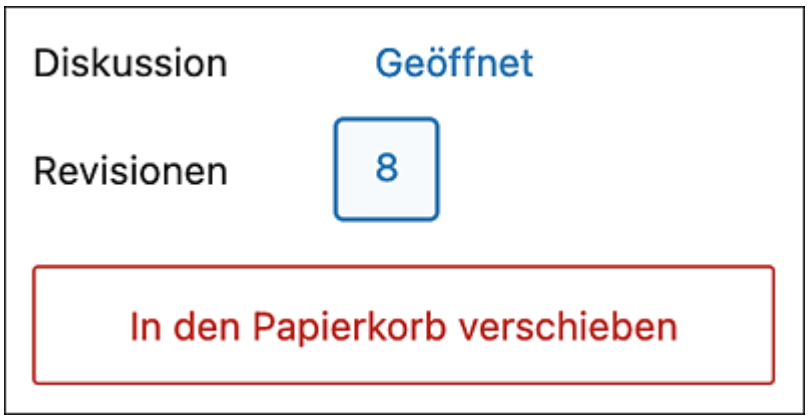


Abbildung 6.38 In der Dokumenten-Seitenleiste des Block-Editors finden Sie den Punkt »Revisionen«, über den Sie in die Revisionen-Ansicht gelangen.



Abbildung 6.39 In der Revisionsansicht können Sie unterschiedliche Stände eines Inhalts vergleichen und einen älteren Stand wiederherstellen.

Wenn Sie der Vergleich von zwei Revisionen, die direkt hintereinanderkommen, nicht weiterhilft, weil Sie weiter auseinanderliegende Revisionen vergleichen möchten, können Sie oben rechts die Checkbox Zwei beliebige Revisionen vergleichen nutzen. Ist die Checkbox aktiv, erscheint ein zweiter Schieberegler, sodass Sie mit dem linken Regler die Ursprungsversion wählen können, von der ausgehend Sie die Änderungen zu einer neueren Version anschauen möchten, die Sie mit dem rechten Regler auswählen.

Auch wenn Sie die Revisionen vielleicht nicht häufig brauchen werden, ist die Freude umso größer, dass es sie gibt, wenn Sie sie doch brauchen. Theoretisch können Sie die Revisionen deaktivieren, das ist aber nicht empfehlenswert. Wenn Sie die Anzahl der gespeicherten Revisionen verringern wollen (standardmäßig gibt es keine Begrenzung), können Sie das in der *wp-config.php* tun. Wie es geht, ist in Kapitel 5, »Administration und Konfiguration von WordPress«, beschrieben.

6.4 Individuelle Felder nutzen

Beiträge, Seiten und so ziemlich alle anderen Inhaltsarten auch können in WordPress mit sogenannten *Metadaten* angereichert werden. Das können Daten sein, in denen Plugins Informationen speichern, die beispielsweise in Blöcken oder der Seitenleiste des Block-Editors genutzt werden (ein Beispiel dazu finden Sie im Kapitel zur Plugin-Entwicklung).

Sie können aber auch manuell Metadaten anlegen, sei es nur für den internen Gebrauch oder um die Daten etwa als Inhalt von Blöcken zu nutzen. Dabei ist das Anlegen von Metadaten recht einfach, da der Block-Editor eine eigene Funktion dafür hat, die zwar standardmäßig ausgeblendet ist, aber einfach aktiviert werden kann.

6.4.1 Individuelles Feld anlegen

Um die Funktion für individuelle Felder im Editor zu aktivieren, führen Sie die folgenden Schritte aus:

1. Öffnen Sie oben rechts im Editor das Drei-Punkte-Menü.
2. Klicken Sie ganz unten im sich öffnenden Drop-Down-Menü auf Voreinstellungen.
3. Scrollen Sie in dem Overlay, das sich öffnet, im Allgemein-Tab ganz nach unten zum Abschnitt Erweitert, und aktivieren Sie die Option Individuelle Felder.
4. Sie werden unter der Option darüber informiert, dass für diese Änderung die Seite neu geladen werden muss. Klicken Sie auf

Anzeigen und Seite neu laden, und warten Sie, bis die Seite neu geladen ist (siehe Abbildung 6.40).

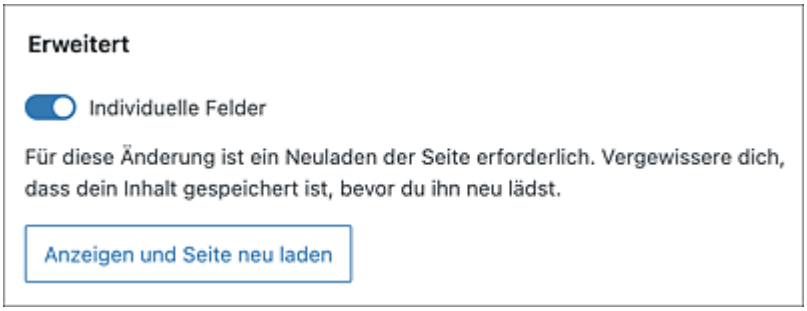


Abbildung 6.40 Die Funktion »Individuelle Felder« muss in den Block-Editor-Einstellungen aktiviert werden.

5. Wenn Sie nun im Editor ganz nach unten scrollen, finden Sie unter dem Editor-Bereich ein Panel Individuelle Felder (siehe Abbildung 6.41).

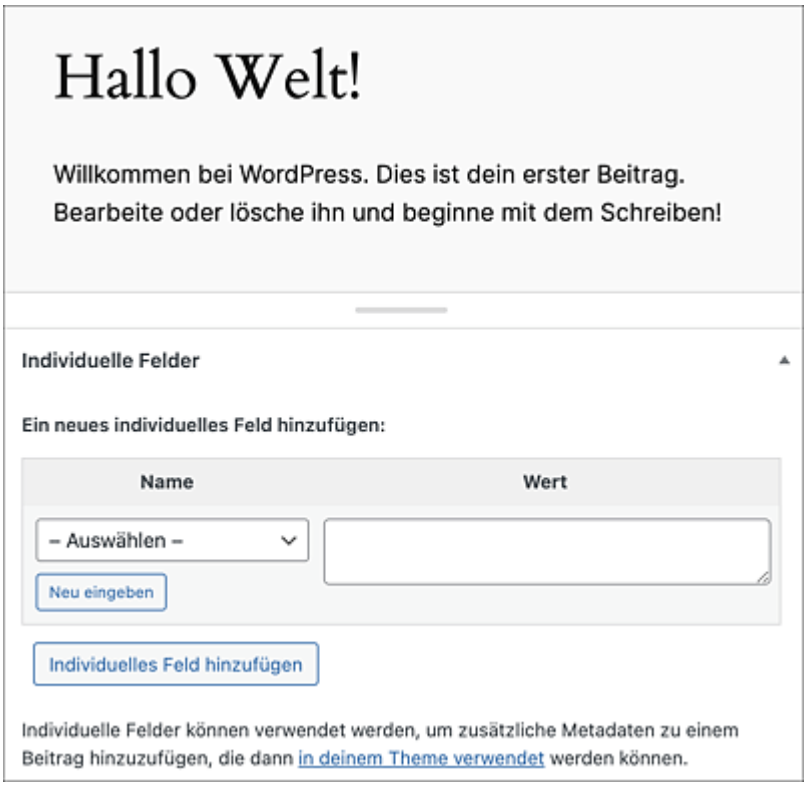


Abbildung 6.41 Nach der Aktivierung finden Sie das Panel für die individuellen Felder im Editor unter dem Bereich zur Inhaltsbearbeitung.

6. In das leere Name-Feld geben Sie den internen Bezeichner ein, über den Sie später auf den Wert des Felds zugreifen können. Der Name sollte deshalb keine Sonderzeichen, Leerzeichen oder Ähnliches enthalten. Beschränken Sie sich am besten auf Kleinbuchstaben und Unterstriche. Für das Beispiel nutzen Sie den Namen `overline`, um eine Dachzeile anzugeben.
7. Wenn Sie den Beitrag oder die Seite speichern, wird der Wert für das individuelle Feld in die `wp_postmeta`-Tabelle der Datenbank gespeichert.

Individuelles Feld in Plugin oder Classic Theme nutzen

Um ein individuelles Feld in einem Plugin oder Classic Theme auszugeben, können Sie die Funktion `get_post_meta()` nutzen. Dafür brauchen Sie lediglich die ID von einem Inhalt wie einer Seite oder einem Beitrag und den Namen des Felds:

```
get_post_meta( get_the_ID(), 'overline', true );
```

`get_post_meta()` ermöglicht Ihnen, einzelne oder auch alle Metadaten zu einem Inhalt aus der Datenbank zu holen.

Der erste Parameter, die ID des Inhaltselements, ist verpflichtend, die beiden anderen Parameter sind optional. Der zweite Parameter ist der Name des Metawerts, der dritte die Angabe, ob Sie nur einen einzelnen Wert zurückbekommen möchten (es können auch mehrere Werte für einen Namen existieren. Wenn der dritte Parameter `true` ist, wird der erste zurückgegeben). Wenn Sie nur den ersten Parameter übergeben, bekommen Sie alle Metadaten als Array zurück.

Aus dem `get_post_meta()`-Aufruf in diesem Beispiel bekommen Sie die Zeichenkette zurück, die Sie als individuelles Feld eingegeben

haben, und können es beliebig weiterverarbeiten.

Individuelles Feld in einem Block-Theme nutzen

Mit der *Block Bindings API*, auf die in Kapitel 23 noch mal eingegangen wird, können Sie Blöcke mit Postmeta-Einträgen verknüpfen, sodass sie immer den aktuellen Wert eines Postmeta-Eintrags anzeigen. Damit das funktioniert, muss der Metawert aber über die REST-API zugänglich sein, das heißt, sie muss extra registriert werden – nur die Definition über das Editor-Interface genügt nicht.

Um die `overline`-Postmeta zu registrieren, sodass sie in der Block Bindings API genutzt werden kann, ist folgendes Schnipsel in der *functions.php* notwendig:

```
add_action( 'init', function() {
  register_post_meta(
    'post',
    'overline',
    [
      'single' => true,
      'show_in_rest' => true
    ]
  );
} );
```

Listing 6.1 Registrierung der »overline«-Postmeta zur Nutzung mit der Block Bindings API

Auch dieses Code-Pattern kommt im Entwicklungskapitel 24 noch mal vor, daher nur kurz: `register_post_meta()` wird genutzt, um die `overline`-Postmeta für Beiträge (`post`) zu registrieren und in der REST-API anzuzeigen (`show_in_rest`).

Um den Metawert in einem Block anzuzeigen, gibt es zwei Möglichkeiten. Öffnen Sie für beide einen Beitrag oder eine Seite, in

die Sie unten über den Bereich Individuelle Felder eine `overline` eintragen.

1. **Über die Code-Ansicht**
   Wechseln Sie im Block-Editor in die Code-Ansicht (Drei-Punkte-Menü oben rechts • Code-Editor), und fügen Sie das folgende Markup ein:

```
<!-- wp:paragraph {
  "metadata":{
    "bindings":{
      "content":{
        "source":"core/post-meta",
        "args":{
          "key":"overline"
        }
      }
    }
  }
} -->
<p></p>
<!-- /wp:paragraph -->
```

Listing 6.2 Anzeige des Metawerts in einem Block über die Block Bindings API

   Achten Sie in der Code-Ansicht darauf, dass Sie das neue Markup außerhalb des Markups der gegebenenfalls schon vorhandenen Blöcke einfügen. Am einfachsten ist es, wenn Sie den Block ganz am Anfang oder ganz am Ende einfügen (verschieben können Sie ihn später in der visuellen Ansicht).

   Wechseln Sie über den Button Code-Editor verlassen oben rechts zurück in die visuelle Ansicht, sollten Sie einen Absatz-Block mit dem Postmeta-Wert vorfinden (siehe Abbildung 6.42).



Abbildung 6.42 Wenn Sie einen Block auswählen, der die Block Bindings API verwendet, wird durch Nutzung eines lila Farbtons angezeigt, dass der Block die API nutzt.

2. **Über das User-Interface**
   In WordPress 6.7 wurde ein User-Interface hinzugefügt, sodass ein Block ohne Arbeit im Code mit Postmetawerten verknüpft werden kann. Fügen Sie dafür einen leeren Absatz-Block in den Editor ein, und wählen Sie in der Block-Seitenleiste im ATTRIBUTE-Panel über das Plus-Icon erst das Attribut CONTENT und dann die OVERLINE aus (in den folgenden beiden Screenshots zu sehen). Das Ergebnis im Editor sollte dem Weg von Punkt 1 entsprechen.





Abbildung 6.43 Das Festlegen der Verknüpfung zwischen einem Block und einem Postmetawert über das User-Interface funktioniert über das »Attribute«-Panel und ist ein zweischrittiger Prozess.

Im Theme-Entwicklungs-Kapitel wird ein bisschen genauer auf die API eingegangen, unter anderem auch auf Beschränkungen.

6.5 Medien organisieren

»Medien« umfasst zwar alle Arten Dateien, in der Regel werden es aber wohl hauptsächlich Bilder sein. Videos sollten Sie am besten bei einer Plattform wie Vimeo oder YouTube hosten, damit Sie die Vorteile der Video-Optimierung solcher spezialisierter Plattformen nutzen können.

6.5.1 Medien verwalten, bearbeiten und löschen

Die Medienverwaltung in WordPress läuft über die *Mediathek*, die Sie im linken Admin-Menü über den Punkt Medien • Mediathek oder direkt mit einem Klick auf Medien erreichen (siehe Abbildung 6.44).

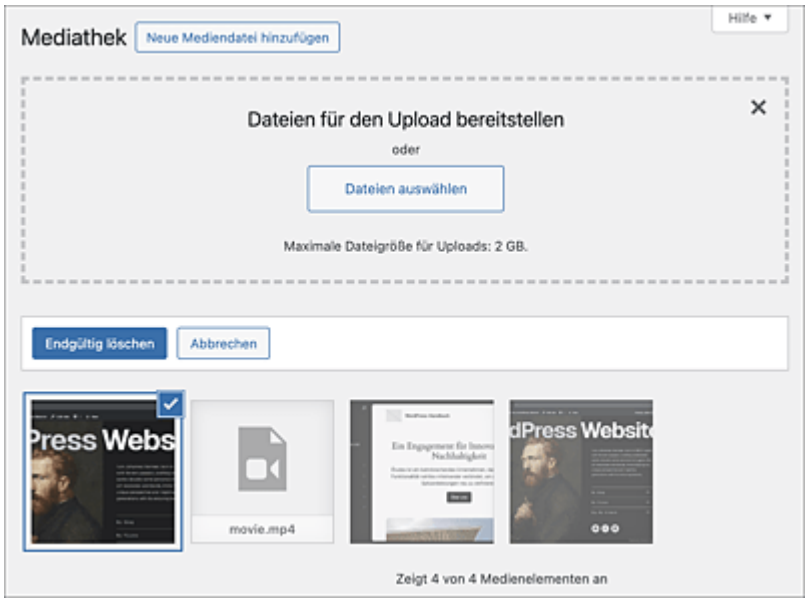


Abbildung 6.44 Die Mediathek zeigt Ihnen alle zu Ihrer Website hochgeladenen Dateien, egal ob der Upload direkt in der Mediathek erfolgt oder beispielsweise über den Bild-Block.

Die Ansichten der Mediathek

Schauen Sie sich im nächsten Schritt die Mediathek mit ihren unterschiedlichen Ansichten und Optionen an (siehe Abbildung 6.45).

❶ **Ansicht**
Hier können Sie die Darstellung der Mediathek zwischen einer Listen- und einer Rasteransicht umschalten. In der Listenansicht (siehe Abbildung) sehen Sie weniger Medien auf einen Blick, dafür sehen Sie nicht nur eine kleine Vorschau, sondern zusätzliche Informationen wie den Dateinamen oder in der Spalte Hochgeladen zu den Beitrag oder die Seite, zu der ein Medium hochgeladen wurde (falls ein Medium direkt in der Mediathek hochgeladen wurde oder der Beitrag/die Seite gelöscht wurde, zu der eine Datei hochgeladen wurde, wird (Nicht verknüpft) angezeigt).

Sie können die Liste nach den meisten der Spalten sortieren, wie es bei den Listen der Beiträge und Seiten möglich ist. Wenn Sie mit dem Cursor über einen Eintrag in der Liste hovern, sehen Sie Aktionslinks, wie Sie sie ebenfalls bereits aus den Übersichtslisten von Beiträgen und Seiten kennen, worüber Sie Dateien unter anderem Endgültig löschen können. Wenn Sie mehrere Dateien löschen möchten, können Sie die Möglichkeit im Auswahlfeld Mehrfachaktionen nutzen, wenn Sie ganz links in den Zeilen mehrere Dateien über die Checkbox markiert haben.

Als Raster (siehe Abbildung 6.44) sehen Sie bei Bildern nur ein Vorschaubild und bei anderen Medientypen zusätzlich zu einem Icon den Dateinamen. Um mehr Infos zu erhalten oder etwa für ein Bild einen Alternativtext anzugeben, müssen Sie auf den Eintrag klicken. Auch in dieser Ansicht ist eine Mehrfachauswahl zum Löschen möglich, und zwar über den Button über dem Raster. Ist die Mehrfachauswahl aktiv,

können Sie nacheinander die Bilder auswählen und diese dann Endgültig löschen.



Abbildung 6.45 In der Listenansicht sehen Sie direkt in der Übersicht auch nähere Details zu den Dateien. Dafür sehen Sie auf einen Blick weniger Dateien.

❷ Alle Medien/Alle Daten

Über diese beiden Auswahllisten können Sie die Medienübersicht nach Dateityp (Alle Medien) und/oder nach Datum filtern (Alle Daten). Die Liste wird direkt gefiltert, wenn Sie einen Eintrag in einer der Auswahllisten auswählen.

❸ Dateien suchen/Medien suchen

Ab einer gewissen Anzahl Dateien wird es etwas unübersichtlich, wenn Sie eine bestimmte Datei wiederfinden möchten, die Sie vor einiger Zeit hochgeladen haben, auch da es keine Möglichkeit gibt, Ordner zu erstellen, um die Dateien besser zu strukturieren (über Plugins können Sie diese Option nachrüsten). Wenn Sie Ihre Dateien sinnvoll benannt und bestenfalls auch noch eine Beschriftung hinzugefügt haben (siehe Abbildung 6.46), wird Ihnen die Suchfunktion aber gute Dienste leisten können, um gezielt nach Dateien zu suchen (siehe Abbildung 6.47).

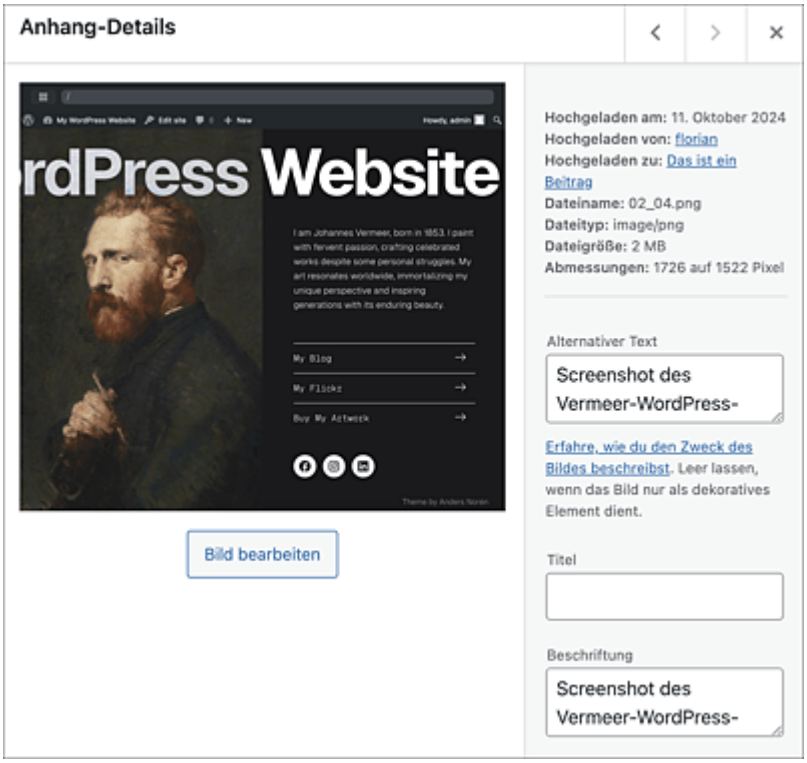


Abbildung 6.46 Die Pflege der Beschriftung ist für die Findbarkeit über die Suchfunktion nützlich.



Abbildung 6.47 Mit der Suche können Sie gezielt nach Dateien suchen. Wie gut das funktioniert, ist allerdings abhängig davon, wie gut die Dateien benannt beziehungsweise die Felder wie die Beschriftung gepflegt sind.

Achtung: Keine Änderungen per FTP an den Mediathek-Dateien

Die Dateien aus der Mediathek finden Sie auf dem Server im *wp-content/uploads*-Verzeichnis. Jetzt könnten Sie auf die Idee

kommen, per FTP Dateien hochzuladen, die werden aber nicht in der Mediathek angezeigt, da WordPress die Dateien beim Upload in der Datenbank registriert.

6.5.2 Bilder optimieren

Ja, auch in Zeiten von oft schnellen Internetverbindungen sollten Dateigrößen von Bildern so klein wie möglich sein. Zunächst haben nicht alle Personen Zugang zu schnellem Internet, oder sie sind vielleicht mit einer Verbindung unterwegs, bei der nach verbrauchten Daten bezahlt wird. Zum anderen tragen kleinere Bilddateien zu einer verbesserten Performance der Website bei, was für alle Besucher*innen gut ist und auch von Suchmaschinen positiv bewertet wird.

Ein erster Schritt bei der Optimierung ist, die Bilder nicht in der Originalgröße hochzuladen, in der sie aus der Kamera fallen, auch wenn das seit WordPress 5.3 eher dafür gut ist, nicht unnötig viel Platz auf Ihrem Server zu verbrauchen (siehe Kasten). Einen allgemeingültigen Maximalwert gibt es natürlich nicht, aber wenn Sie ein Bild zum Beispiel in einer Breite von 1000 Pixeln auf einer Seite einfügen, muss es wahrscheinlich nicht breiter als 2000 Pixel hochgeladen werden. Damit sollten auch höher auflösende Displays gut bedient sein.

Tipp: Maximale Bildmaße von Uploads seit WordPress 5.3

Mit WordPress 5.3 wurde eine neue Funktion eingefügt, um das Einbinden sehr großer Bilder zu verhindern. Dafür wird intern eine Bildgröße generiert, wenn ein hochgeladenes Bild höher oder breiter als 2560 Pixel ist, sodass die längere Seite des Bilds

maximal 2560 Pixel groß ist. Die Originalversion des Bilds wird jedoch nicht gelöscht, sondern bleibt auf dem Server erhalten, kann aber nicht versehentlich eingefügt werden. Über einen Filter lässt sich dieser Grenzwert verändern oder auch komplett deaktivieren.

Auch wenn die Verringerung der Bildmaße ein guter erster Schritt ist, kann die Dateigröße durch weitere Optimierungsmaßnahmen deutlich verkleinert werden. Am einfachsten geht das mit einem Plugin wie Imagify (*http://r-wrk.de/wp6_imagify*) oder Optimus (*http://r-wrk.de/wp6_optimus*). Diese Plugins schicken hochgeladene Bilder zu einem externen Server, der die Dateien optimiert und an Ihren Server zurückschickt, wo die optimierte Datei die nicht optimierte ersetzt. Gezahlt wird dabei ein Inklusivkontingent pro Zeiteinheit. Vorteil dieser Variante ist, dass auch die von WordPress generierten kleineren Größen optimiert werden. In Abschnitt 19.6 wird Ihnen Imagify näher vorgestellt.

Eine andere Variante ist, Bilder vor dem Upload durch Dienste wie JPEGmini (*http://r-wrk.de/wp6_jpegmini*) oder TinyPNG (*http://r-wrk.de/wp6_tinypng*) laufen zu lassen. Natürlich können Sie auch die Komprimierungsoptionen der gängigen Bildbearbeitungsprogramme nutzen, allerdings sind die Ergebnisse meist nicht so gut. Bei allen Varianten, bei denen die Bilddatei nur vor dem Upload optimiert wird, gibt es den Nachteil, dass die von WordPress generierten Größen nicht optimiert sind.

Tipp: Das richtige Bildformat

Je nach Art eines Bilds kann es geschickt sein, ein bestimmtes Bildformat einzusetzen, um die Dateigröße zu reduzieren.

- Für Fotos ist JPEG das richtige Format.

- Wenn Sie Illustrationen oder andere Grafiken mit nur wenigen Farben haben oder Transparenz zum Einsatz kommt, sollten Sie PNG verwenden.
- Es gibt verschiedene Weiterentwicklungen wie WebP oder AVIF, die Bilder bei gleicher Qualität noch kleiner machen können (aber nicht müssen). Zur Unterstützung gibt es unterschiedliche Lösungen, suchen Sie einfach mal mit der Suchmaschine Ihrer Wahl.

6.5.3 Videos bei YouTube oder Vimeo hosten

Videos werden Sie wahrscheinlich nicht so oft einsetzen wie Bilder, vielleicht sogar nie, aber wenn Sie eins in einen Beitrag oder eine Seite einbinden möchten, sollten Sie eine Sache beachten: Videos selbst zu hosten, ist selten eine gute Idee, es sei denn vielleicht, es handelt sich nur um ein paar Sekunden. Grund dafür ist, dass Videodateien von Haus aus recht groß sind und die Komprimierung ein bisschen tricky sein kann.

Bevor Sie sich in das Thema Video-Optimierung einarbeiten oder unkomprimierte Videos an Ihre Besucher*innen ausliefern, sollten Sie darüber nachdenken, das Hosting Ihrer Videos auf Plattformen wie Vimeo oder YouTube auszulagern. Vorteil dabei ist, dass diese Plattformen Ihr Video optimieren und in unterschiedlichen Größen bereitstellen, sodass die Version ausgeliefert wird, die am besten zum Endgerät und der Netzwerkverbindung Ihrer Besucher*innen passt. Wenn Sie möchten, haben Sie über die Videoplattform eine zusätzliche Möglichkeit, auf Ihre Videos aufmerksam zu machen.

6.6 Die mobilen Apps

Grundsätzlich ist das WordPress-Backend responsive, das heißt, Sie können es auch ohne Probleme mit dem Smartphone nutzen (siehe Abbildung 6.48). Trotzdem gibt es für Android und iOS je eine offizielle App für WordPress, die Ihnen die Nutzung der wichtigsten Funktionen optimiert für Smartphones bereitstellt und das Backend etwas aufräumt.

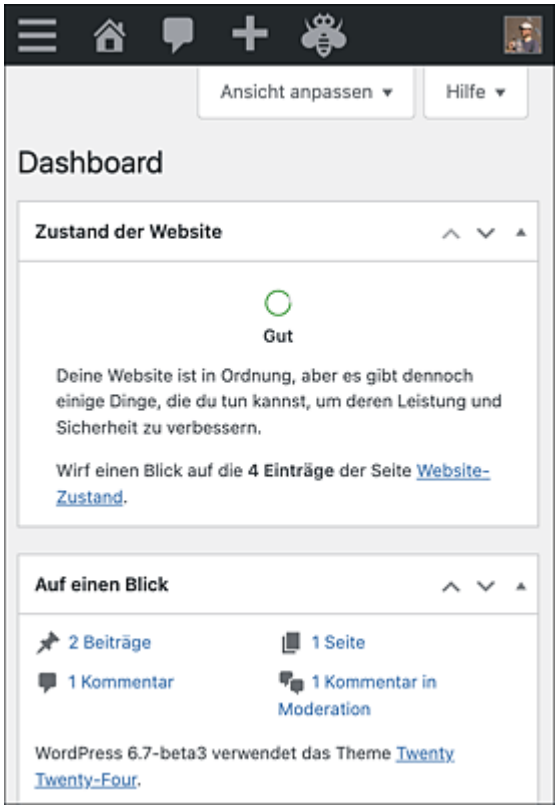


Abbildung 6.48 Da das WordPress-Backend responsive ist, lässt es sich generell auch mit Smartphones bedienen.

Wenn Sie regelmäßig über Ihr Smartphone Inhalte bearbeiten oder erstellen möchten, sollten Sie einmal einen Blick darauf werfen:

- Android-App: *http://r-wrk.de/wp6_app-android*

- iOS-App: *http://r-wrk.de/wp6_app-ios*

6.7 Inhalte planen

Abschließend kommen in diesem Kapitel noch ein paar Hinweise und Tipps zum eigentlichen Inhalt Ihrer Website. Das Vorgehen bei der Planung unterscheidet sich natürlich stark, je nachdem wie groß das Team ist, das an der Website arbeitet. Wenn es eine Person ist, ist der Organisationsaufwand deutlich geringer, als wenn mehrere Personen an der Inhaltserstellung beteiligt sind.

6.7.1 Konzept

Wenn es um Ihre private Website geht, auf der Sie etwas vorstellen, ohne unbedingt bestimmte Zielgruppen erreichen zu wollen, müssen Sie sich vorher vielleicht nicht unbedingt große Gedanken über ein Konzept machen. Wenn eine Website aber ein klares Ziel erfüllen soll, sollten Sie sich im ersten Schritt Gedanken machen, wie dieses Ziel am ehesten erreicht werden kann. Was braucht es für Inhalte und Arten von Inhalten, womit können Sie Ihre Zielgruppe(n) ansprechen, gibt es eventuell eine größere Redaktion, sodass Sie einen Redaktionsworkflow in WordPress abbilden und die zuständigen Personen schulen müssen, und so weiter.

6.7.2 Ohne gute Inhalte läuft nichts

Auch wenn Ihre Website wunderschön aussieht: Gute Inhalte sind das, worauf es ankommt. Ohne gute Inhalte bekommen Sie keine Leute auf Ihre Website und werden damit nicht das oder die Ziele erreichen, die Sie für Ihre Website haben. Sei es nur, dass Ihre Texte

gelesen und vielleicht kommentiert werden oder dass Produkte gekauft werden.

Ein paar Tipps:

- Schreiben Sie für Ihre Zielgruppe. Wenn Sie beispielsweise nicht nur Leser*innen ansprechen möchten, die schon tief in Ihrem Thema drin sind, setzen Sie nicht zu viel voraus.
- Lesen Sie Texte Korrektur, und lesen Sie sich den Text gegebenenfalls selbst laut vor. Wenn Sie Ihre Texte so schreiben möchten, wie Sie auch sprechen würden, kann das helfen zu erkennen, ob geschriebene Formulierungen sich seltsam anhören.
- Schreiben Sie Alternativtexte für alle Bilder, die nicht rein dekorativ sind, damit Menschen mit Sehbehinderung die Informationen des Bilds erfassen können. Nebenbei sind Alternativtexte auch für Suchmaschinen hilfreich.

6.7.3 Barrierefreiheit von Inhalten

Inhalte sind wie beschrieben das Wichtigste, und wichtig (und leider häufig nicht berücksichtigt) ist, dass möglichst viele Menschen diese Inhalte konsumieren können. Dafür müssen beispielsweise inhaltlich wichtige Bilder (also solche, die nicht rein dekorativ sind) mit dem `alt`-Attribut ausgestattet werden, in dem der Bildinhalt beschrieben wird, damit Menschen mit einer Sehbehinderung die Bildinformationen ebenfalls erfassen können. Bei dekorativen Bildern sollte ein leeres `alt`-Attribut gesetzt werden. Den Alternativtext für Bilder können Sie in der Detailansicht eines Bilds in der Mediathek pflegen oder über die Block-Seitenleiste im Editor, wenn Sie einen Bild-Block ausgewählt haben.

Für Videos sollten Untertitel bereitgestellt werden, für Audiodateien ein Transkript. Bei Formularen ist ein oft gesehenes Problem, dass Felder nicht richtig mit der Beschriftung (`label`) verknüpft sind und/oder statt einer richtigen Beschriftung nur ein Placeholder genutzt wird. Und es sollte sichergestellt werden, dass alle Inhalte und Aktionen nicht nur mit der Maus oder per Touch ausgeführt werden können, sondern beispielsweise auch mit der Tastatur.

Info: Richtlinien zur Barrierefreiheit im Web

Die *Web Content Accessibility Guidelines* (*WCAG*) sind von der *Web Accessibility Initiative* (*WAI*) entwickelte Standards für barrierefreie Websites. Unter *w3.org/WAI/WCAG21/quickref/* gibt es eine Liste mit sehr vielen Punkten, die nicht alle für jede Website relevant, aber eine sehr gute Anlaufstelle sind, um sich darüber bewusst zu werden und in Erinnerung zu rufen, welche Aspekte es zu berücksichtigen gibt.

Info: Gute Alternativtexte schreiben

Wie erwähnt, sind `alt`-Attribute dafür da, den Inhalt eines Bilds zu beschreiben, sodass auch Menschen mit Sehbehinderung, die mit einem Screenreader auf Ihre Webseite zugreifen, eine Idee davon bekommen, worum es im Bild geht und warum das Bild überhaupt eingefügt wurde. Der Alternativtext sollte keine lange, detaillierte Beschreibung sein, aber auch nicht zu kurz (»Fahrrad« ist kein guter Alternativtext für ein Bild von einem Fahrrad, etwas genauer darf es schon sein). Beachten Sie auch immer den Kontext, in dem das Bild vorkommt. Versuchen Sie, sich in eine Person hineinzuversetzen, die das Bild nicht sehen kann, und überlegen Sie, was sie für Informationen über das Bild benötigt. Und: Nutzen

Sie die Alternativtexte auf keinen Fall für SEO. Der Text ist für Menschen gedacht.

6.7.4 Struktur der Inhalte mit einer Sitemap planen

Um sich einen Überblick über die Inhaltsstruktur Ihrer Website zu machen, bietet es sich an, eine Sitemap zu erstellen. Darin können Sie die unterschiedlichen Navigationsebenen abbilden und so beispielsweise erkennen, ob Sie zu viele Seiten für die erste Menüebene eingeplant haben oder die Inhalte zu tief verschachtelt sind, als dass sie im Menü auf Ihrer Website abgebildet werden können.

Zur Erstellung einer Sitemap können Sie ein Programm wie Google Spreadsheet oder Excel nutzen: Jede Zeile ist eine Ebene der Inhaltsstruktur, in der obersten Zeile kommen also die Inhalte für die erste Menüebene, darunter diejenigen für die zweite und so weiter. Inhalte, die nicht im Hauptmenü auftauchen, können Sie als eigene Gruppen einfügen, beispielsweise die Seiten für rechtliche Hinweise, die Sie in einer Navigation im Footer anzeigen.

Tipp: Seitentitel mit oder ohne Website-Namen

SEO-Plugins bieten in der Regel unter anderem die Option an, die Ausgabe des Website-Titels zu verändern und beispielsweise den Namen der Website an den Seiten- oder Beitragstitel anzuhängen. Bei der Entscheidung, ob Sie das machen möchten oder nicht, hilft es vielleicht, sich zu fragen, ob diese Information für Nutzer*innen auf der Ergebnisseite einer Suchmaschine wichtig ist oder nicht. Wenn Sie eine Marke mit einem prägnanten Namen aufbauen möchten, kann es vielleicht nützlich sein, den Namen schon in den

Suchergebnissen im Titel zu sehen, aber der Namen nimmt auch immer einen Teil der verfügbaren Zeichen ein, die direkt in den Suchergebnissen sichtbar sind.

6.8 Classic Editor statt Block-Editor nutzen

Wenn Sie den Block-Editor nicht nutzen möchten/können, installieren Sie das offiziell von WordPress gepflegte Plugin »Classic Editor«: *https://de.wordpress.org/plugins/classic-editor/*.

Nach der Installation und Aktivierung des Plugins bearbeiten Sie die Inhalte im Classic Editor, bei dem Sie statt des Block-Editors das einzelne Feld des TinyMCE-Editors zur Verfügung haben (siehe Abbildung 6.49).

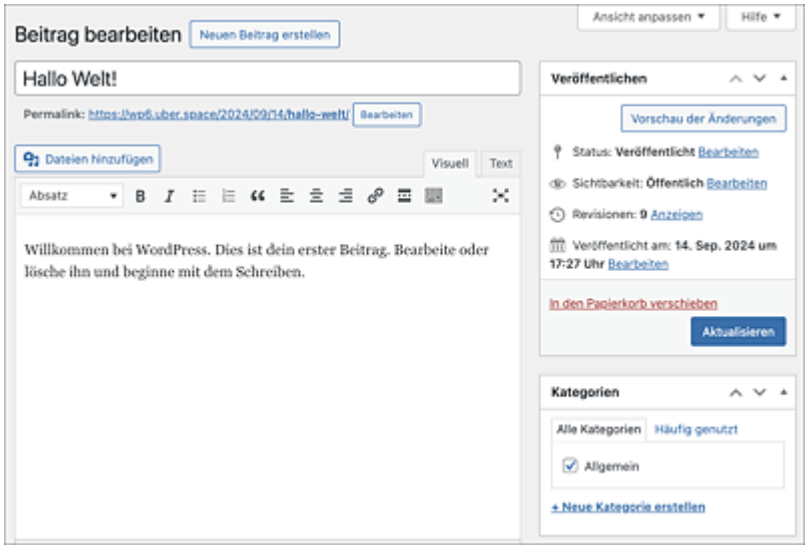


Abbildung 6.49 Mit dem »Classic Editor«-Plugin wird der Block-Editor gegen den TinyMCE-Editor ausgetauscht, der bis WordPress 5 standardmäßig genutzt wurde.

In den Einstellungen können Sie unter anderem anpassen, welcher Editor der Standard-Editor sein soll und ob Benutzer*innen den Editor auswählen dürfen (siehe Abbildung 6.50).

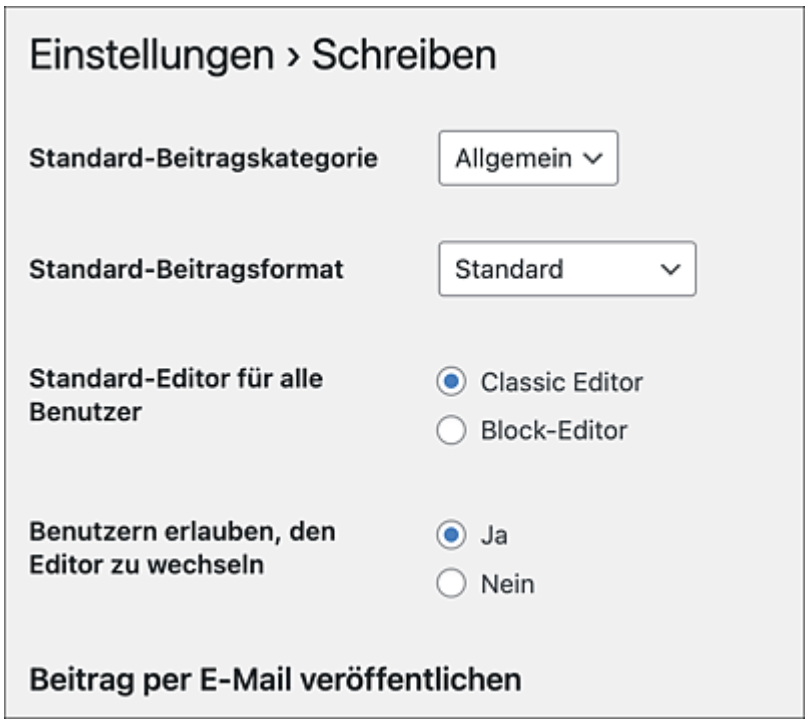


Abbildung 6.50 In den Einstellungen können Sie unter anderem anpassen, welcher Editor standardmäßig geöffnet wird.

6.9 Impressum und Datenschutzerklärung nicht vergessen

In der Regel braucht eine Website ein Impressum, und meist werden Sie auch schon durch Logs der Zugriffe auf Ihrer Website, die durch Ihren Webhoster erstellt werden, personenbezogene Daten verarbeiten (die IP-Adresse), weshalb auch eine Datenschutzerklärung oft notwendig ist (auch wenn diese relativ kurz sein kann, wenn Ihre Website wenig datenschutzrelevante Funktionen nutzt).

7 Benutzer*innen verwalten

*Wenn Sie nicht die einzige Person sind, die an einer WordPress-Instanz mitwirkt, ist es eine gute Idee, wenn Sie sich ein wenig mit den Berechtigungen und Rollen auseinandersetzen, die für Benutzer*innen in einer WordPress-Installation vergeben werden können.*

Im Rahmen der WordPress-Installation erstellen Sie einen Account für sich als Admin, womit Sie Zugriff auf alle Einstellungen und Möglichkeiten im WordPress-Backend haben. Sobald andere Personen dazukommen, stellt sich die Frage, welche Berechtigungen sie brauchen, um zu vermeiden, dass versehentlich oder absichtlich Einstellungen oder andere Aktionen vorgenommen werden, die nicht gewollt sind.

7.1 Rollen verstehen

Um verschiedene Arten von Benutzer*innen zu ermöglichen, gibt es in WordPress mehrere *Rollen*, die mit unterschiedlichen Berechtigungen einhergehen. Die folgenden Rollen bringt WordPress standardmäßig mit:

- Super-Administrator ist eine spezielle Rolle, die es nur in Multisite-Installationen gibt. Ein Super-Admin hat Zugang zur Netzwerk-Administration und zu allen Sites, kann neue Sites erstellen et cetera. Mehr zum Thema Multisite erfahren Sie in Kapitel 9.

- Administrator ist in einer Standard-Installation die höchste Rolle. Konten mit dieser Rolle haben Zugriff auf alle Einstellungen einer WordPress-Site. In einer Multisite kann es Benutzer*innen mit dieser Rolle geben, die nur für eine oder mehrere bestimmte Site(s) der Multisite Admin sind.
- Redakteur ist die richtige Rolle für Personen, denen es erlaubt sein soll, möglichst viele Dinge im redaktionellen Bereich zu erledigen. So kann diese Rolle Kommentare verwalten, Inhalte von anderen Personen bearbeiten, veröffentlichen, löschen …
- Autor ist eine Rolle, die neue Inhalte erstellen, bearbeiten und eigene Inhalte löschen kann, aber keinen Zugriff auf Inhalte anderer Personen hat.
- Abonnent bringt als Rolle keine besonderen Berechtigungen mit, außer das eigene Konto zu bearbeiten.

Rollen und Berechtigungen

Diese Rollen sind mit Berechtigungen verbunden. So hat zum Beispiel ein*e Benutzer*in mit der Rolle Super-Administrator die Berechtigung `create_sites`, die keine andere Rolle hat. Die `export`-Berechtigung ist eine, die auch normale Administrator*innen haben, `moderate_comments` haben auch Redakteur*innen und so weiter. Alle Rollen und Berechtigungen finden Sie auf *wordpress.org/documentation/article/roles-and-capabilities*.

Im Optimalfall wird bei der Beschränkung von Funktionen im Programmcode nicht auf eine bestimmte Rolle geprüft, sondern auf eine Berechtigung, also ob beispielsweise die `moderate_comments`-Berechtigung vorhanden ist.

Über unterschiedliche Plugins lassen sich die bestehenden Rollen auch anpassen, um beispielsweise einer bestimmten Rolle eine weitere Berechtigung zu geben und einer anderen Rolle Berechtigungen zu nehmen. Darüber hinaus können mit solchen Plugins weitere Rollen erstellt werden, und Plugins bringen teilweise auch eigene Rollen und Berechtigungen mit, wie etwa WooCommerce.

7.2 Benutzer*innen hinzufügen

Um neben Ihrem Adminstrationskonto weitere Benutzer*innen in WordPress anzulegen, gibt es zwei Möglichkeiten.

1. **Selbstregistrierung**

   Wenn Sie es über die Einstellungen erlaubt haben (Einstellungen • Allgemein • Jeder kann sich registrieren), können sich alle, die möchten, über die URL */wp-login.php?action=register* selbst ein Konto erstellen, indem E-Mail-Adresse und Benutzername angegeben werden.

   Nach Absenden des Formulars wird eine Bestätigungs-E-Mail an die eingegebene Adresse geschickt, in der ein Link enthalten ist, um den neuen Account zu aktivieren und ein Passwort zu vergeben. Nach einem Klick auf den Link gelangt die Person zu einem Formular, um ein Passwort zu vergeben, und wird danach auf die Login-Seite weitergeleitet.

2. **Über das Backend anlegen**
   Die zweite und vermutlich gängigere Variante ist, dass Sie eine*n Benutzer*in direkt über das WordPress-Backend anlegen. Dafür nutzen Sie den Punkt Benutzer • Neu hinzufügen im linken Admin-Menü. Wichtig sind beim Erstellen der Benutzername, die E-Mail-Adresse und die Rolle, da diese Informationen zum Anmelden genutzt werden können und Auswirkungen darauf haben, was nach dem Einloggen an Aktionen möglich ist.

   Alle anderen Informationen können Sie leer lassen, bei Bedarf kann das neu angelegte Profil von der Person, der es gehört, später selbst vervollständigt werden. Vor- und Nachname können dabei als Anzeigename genutzt werden, der auch im

Backend sichtbar wird, etwa bei der Autorenangabe für Beiträge oder Seiten. Alle anderen Angaben können von Themes genutzt werden, um zum Beispiel ein Kurzprofil unter Beiträgen anzuzeigen oder Ähnliches.

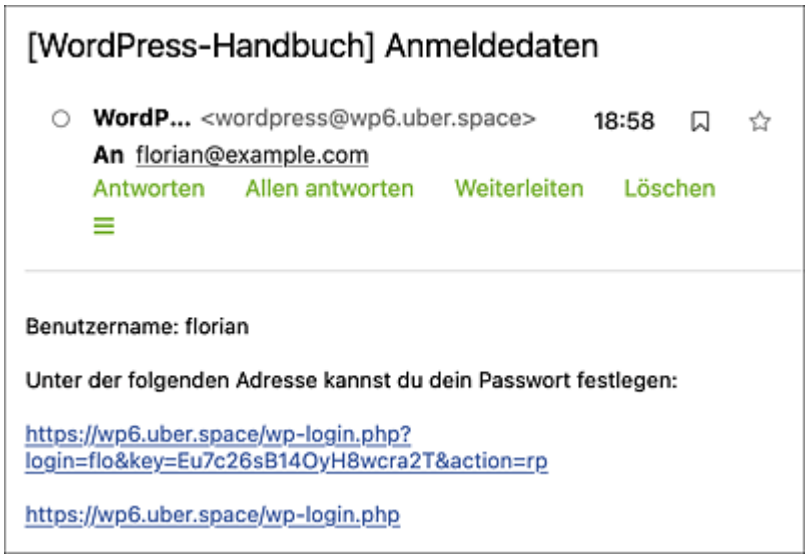


Abbildung 7.1 Die Bestätigung einer Anmeldung mit einem Link aus einer E-Mail wird »Double-Opt-in« genannt.

7.3 Benutzer*innen verwalten: sortieren, filtern, suchen

Wenn Sie sich einen Überblick über die auf Ihrer Website registrierten Benutzer*innen verschaffen möchten, können Sie das unter dem Admin-Menüpunkt Benutzer • Alle Benutzer tun. Hier finden Sie eine Liste, die alle Benutzerkonten der Website auflistet und Ihnen direkt in der Listenansicht ein paar nützliche Informationen zu jedem Eintrag anzeigt. Natürlich können Sie die Liste auch filtern, wie die meisten anderen Listen von Einträgen im WordPress-Backend. Gehen wir die Ansicht einmal durch.

❶ **Filterung nach Rolle**. Direkt unter dem Titel der Seite finden Sie Links, um die Liste nach Rollen zu filtern. Dabei werden nur die Rollen angezeigt, die auch verwendet werden, vielleicht haben Sie hier also nur Alle und Administrator stehen.

❷ Mehrfachaktionen. Über das Dropdown können Sie mehrere ausgewählte Benutzer*innen löschen oder die Passwort-vergessen-E-Mail an die hinterlegten E-Mail-Adressen versenden.

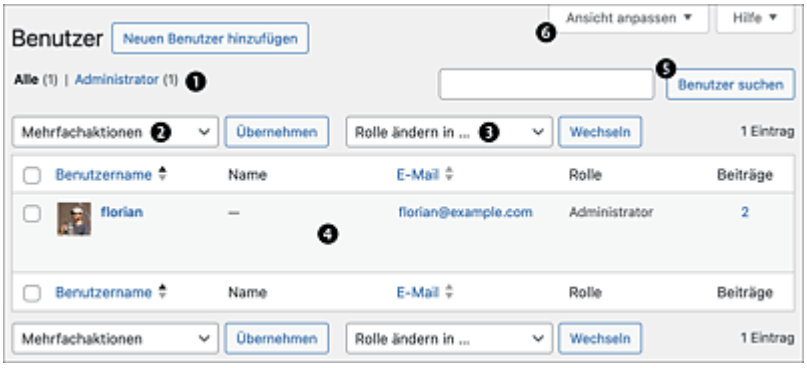


Abbildung 7.2 Von der Liste aller Benutzer*innen gelangen Sie in die Detailansicht zur Bearbeitung einzelner Accounts.

❸ Rolle ändern in. Damit können Sie für mehrere ausgewählte Benutzer*innen eine neue Rolle vergeben.

❹ **Liste der Benutzer*innen**. Hier sehen Sie standardmäßig die Spalten Benutzername, Name (der Anzeigename, den jede*r Benutzer*in im eigenen Profil festlegen kann, beispielsweise nur der Vorname oder Vor- und Nachname), E-Mail, Rolle und Beiträge (zeigt die Anzahl der Beiträge an, die von dem Account erstellt wurden).

Wenn Sie die Maus über einen Eintrag bewegen, sehen Sie in der Benutzername-Spalte Links zu Aktionen wie Bearbeiten (Sie können alternativ auch direkt auf den Namen klicken), Löschen und dem Versenden der Mail zum Zurücksetzen des Passworts. Bei Ihrem eigenen Profil ist der Link zum Löschen und Zurücksetzen des Passworts nicht vorhanden.

❺ Benutzer suchen erlaubt Ihnen, die Liste der Benutzer*innen zu durchsuchen.

❻ Ansicht anpassen finden Sie auch in anderen Backend-Ansichten – hier können Sie die angezeigten Informationen der Liste anpassen und Spalten ausblenden, die Sie nicht benötigen, beispielsweise die Anzahl der Beiträge.

7.4 Benutzerprofil bearbeiten

Wenn Sie in die Bearbeitungsansicht eines Profils wechseln, können Sie dort deutlich mehr Informationen bearbeiten als in dem Formular beim Erstellen eines neuen Profils.

❶ Visueller Editor • Beim Schreiben den visuellen Editor nicht benutzen. Wenn diese Checkbox gesetzt ist, hat der oder die Benutzer*in beim Bearbeiten eines Beitrags, einer Seite oder eines anderen Inhalts nur den Code-Editor zur Verfügung, nicht den visuellen Editor. Im Normalfall sollten Sie diese Option eigentlich nicht benötigen.

❷ Syntaxhervorhebung • Die Syntaxhervorhebung beim Bearbeiten von Code deaktivieren. Mit dieser Option wird das Syntax-Highlighting in dem Editor zur Bearbeitung von Theme- und Plugin-Dateien deaktiviert, die über Design • Theme-Datei-Editor beziehungsweise Plugins • Plugin-Datei-Editor erreichbar sind.

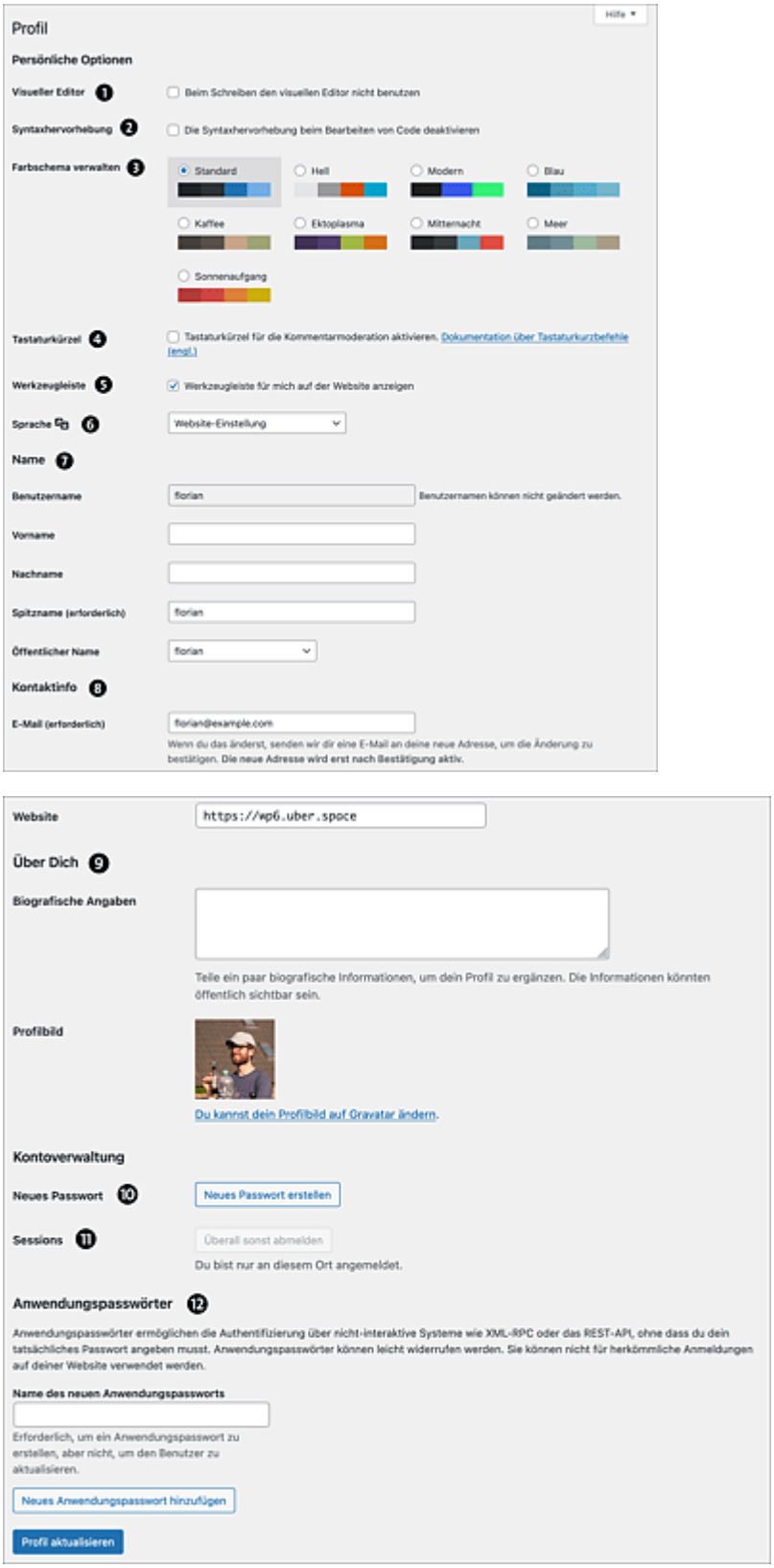


Abbildung 7.3 Die Bearbeitungsmöglichkeiten für ein Profil

Wichtig: Sie sollten diese beiden Datei-Editoren nur im Ausnahmefall nutzen und nur, wenn Sie genau wissen, was Sie tun. Sonst können Sie im schlimmsten Fall die Website zerschießen und sich danach nicht mehr einloggen. (Auch wenn es seit einigen WordPress-Versionen Vorkehrungen dagegen gibt, dass durch die Bearbeitung der Dateien über diese Editoren fatale Fehler entstehen, ist Vorsicht besser als Nachsicht.)

❸ Farbschema verwalten. Hier haben Sie die Möglichkeit, die Farben der oberen Adminbar und des linken Admin-Menüs im Backend anzupassen.

❹ Tastaturkürzel • Tastaturkürzel für die Kommentarmoderation aktivieren. Wenn Sie häufig neue Kommentare moderieren müssen, kann es hilfreich sein, Tastaturkürzel zur Verfügung zu haben, um schnell eine Liste von neuen Kommentaren abzuarbeiten. Mehr Informationen zum Managen von Kommentaren finden Sie in Kapitel 8.

❺ Werkzeugleiste • Werkzeugleiste für mich auf der Website anzeigen. Hiermit können Sie einstellen, ob Sie die oben angezeigte Adminbar auch im Frontend sehen möchten oder nur im Backend. Im Frontend nützlich ist die Leiste beispielsweise, um schnell von einem Beitrag oder einer Seite in die Bearbeitungsansicht springen zu können.

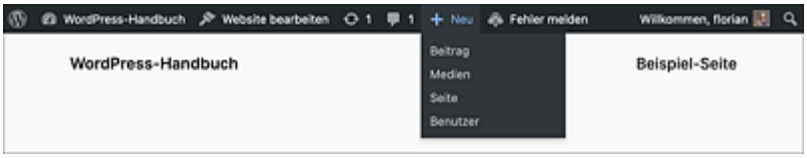


Abbildung 7.4 Die Adminbar bietet im Frontend ein paar nützliche Abkürzungen ins Backend, beispielsweise in die Bearbeitungsansicht des aktuellen Inhalts.

❻ Sprache. Hier stellen Sie die Sprache ein, in der Sie das Backend angezeigt haben möchten. Sie können dabei entweder mit dem Wert Website-Einstellung den Wert übernehmen, der in den allgemeinen Einstellungen für die Website festgelegt ist, oder eine andere Sprache wählen. Wenn ein Plugin oder Theme nicht in der entsprechenden Sprache vorliegt, wird auf Englisch zurückgefallen. Um das zu verhindern und beispielsweise für eine Website in formellem Deutsch zuerst informelles Deutsch als Fallback anzugeben, können Sie das Plugin »Preferred Languages« nutzen (*https://de.wordpress.org/plugins/preferred-languages/*).

❼ Name (Benutzername, Vorname, Nachname, Spitzname (erforderlich), Öffentlicher Name). Die wichtigste Angabe ist hier der Benutzername, der bei der Registrierung festgelegt wird und danach nicht einfach geändert werden kann. Dieser Name kann statt der E-Mail-Adresse für den Login genutzt werden. Die anderen Werte sind bis auf den Spitznamen optional und können für den Öffentlichen Namen genutzt werden. Das ist beispielsweise der Name, der im Backend als Autorenname angezeigt wird.

❽ Kontaktinfo. Hier finden Sie die bei der Registrierung angegebene E-Mail-Adresse, die im Gegensatz zum Username natürlich geändert werden kann. Zusätzlich gibt es noch die Möglichkeit, eine Website-URL anzugeben, die möglicherweise vom Theme genutzt wird, beispielsweise in einem Autorenprofil.

❾ Über Dich • Biografische Angaben. Hier können Sie einen kleinen Text über sich selbst schreiben, der – abhängig vom Theme – auf der Website dargestellt wird, genauso wie das Profilfoto, das Sie aber nicht direkt im WordPress-Backend ändern können, sondern standardmäßig nur über den Online-Dienst *Gravatar*, bei dem Sie Avatare zu E-Mail-Adressen zuordnen können.

❿ Kontoverwaltung • Neues Passwort • Neues Passwort erstellen. Wenn Sie auf diesen Button klicken, erscheint ein Eingabefeld, über das Sie ein neues Passwort vergeben können. Das Feld wird bereits mit einem Vorschlag vorausgefüllt, und unter dem Feld wird ein Indikator angezeigt, wie sicher das Passwort ist. Sie sollten – wie generell bei Passwörtern – kein Passwort nutzen, das einfach zu merken ist. Am besten nutzen Sie einen Passwortmanager und vergeben ein zufälliges, möglichst langes Passwort. Wenn Sie sich das Passwort merken müssen, ist es eine recht sichere Methode, vier oder mehr komplett zufällig gewählte Begriffe hintereinanderzuschreiben. Auch dafür gibt es Generatoren.

Ihr aktuelles Passwort können Sie hier nicht sehen. Wenn Sie das Passwort vergessen haben, schauen Sie in Abschnitt 21.2.6, »Ausgesperrt – Passwort wiederherstellen«, vorbei.

⓫ Kontoverwaltung • Sessions • Überall sonst abmelden. Dieser Button ist vielleicht ausgegraut und lässt sich nicht anklicken, aber wenn, können Sie darüber die Sessions in anderen Browsern beenden. Wenn Sie also beispielsweise auf einem fremden Rechner etwas an Ihrer Website geändert und vergessen haben, sich abzumelden, können Sie das nachträglich über diesen Button tun, egal ob Sie noch auf das Gerät Zugriff haben, auf dem Sie sich angemeldet hatten.

⓬ Anwendungspasswörter. Die Anwendungspasswörter-Funktion erlaubt es, dass sich ein Drittservice bei Ihrer Website authentifizieren kann. Im Optimalfall passiert das so, dass Sie bei dem externen Service die URL zu Ihrer WordPress-Website angeben und dann von dem Dienst eine URL in Ihr WordPress-Backend aufrufen. Nachdem Sie sich bei WordPress angemeldet haben, können Sie dort ein Anwendungspasswort für den Service erstellen (es kann nicht benutzt werden, um sich manuell im Backend anzumelden). Danach werden Sie zurück zum Service geleitet, und in der URL werden als Parameter der Username und das Anwendungspasswort übergeben. Mit diesen Daten kann der Service nun authentifizierte Anfragen an die REST-API stellen, um Aktionen auszuführen, die ohne die Authentifizierung nicht möglich wären.

In diesem Bereich des Benutzerkontos sehen Sie die angelegten Anwendungspasswörter und wann sie von welcher IP zuletzt verwendet wurden. Sie können sie von hier aus auch einfach »widerrufen«, sodass der Drittservice keinen Zugriff mehr hat.

7.5 Benutzer*in deaktivieren oder löschen

Wenn Sie verhindern möchten, dass jemand weiter Zugriff auf das Backend hat, haben Sie zwei Möglichkeiten:

- **Bearbeitungsmöglichkeiten entziehen**. Wenn Sie eine*n Benutzer*in zur Rolle Abonnent machen, kann er oder sie zwar noch das Backend aufrufen, aber außer den eigenen Profilinformationen nichts mehr bearbeiten. Dazu gehen Sie in das entsprechende Profil und ändern die Rolle über das gleichnamige Feld.
- **Endgültig löschen**. Um ein Konto zu löschen, gehen Sie in die Liste aller Benutzer*innen unter Benutzer • Alle Benutzer und fokussieren den entsprechenden Eintrag, sodass die Quick-Links auftauchen. Dort wählen Sie den Löschen-Link aus und werden anschließend gefragt, was mit den Inhalten geschehen soll, die mit diesem Account verknüpft sind. Hier können Sie Den gesamten Inhalt löschen oder Diesem Benutzer den gesamten Inhalt zuordnen. Bei der zweiten Option wird ein Dropdown angezeigt, aus dem Sie wählen können, wem der Inhalt zugeordnet werden soll.

8 Kommentare managen

Kommentare füllen Ihr Blog mit zusätzlichem Leben, mal »nur« mit positiven/kritischen Äußerungen, mal mit Fragen oder auch mit Informationen, die Beiträge verbessern. Hier lernen Sie, wie Sie mit den Kommentaren umgehen.

Auch wenn wahrscheinlich oft nur wenig bis keine Reaktionen in Form von Kommentaren auf Ihre Beiträge kommen werden, ist es doch umso schöner, wenn mal ein positiver Kommentar eintrudelt oder einer, der valide Kritikpunkte benennt und Ihnen ermöglicht, Ihre Inhalte zu verbessern, oder einfach eine Frage und wenn Sie jemandem weiterhelfen können. In diesem Kapitel lernen Sie unter anderem, welche Einstellungen Sie für Kommentare vornehmen können, wie Sie mit neuen Kommentaren umgehen und wie Sie sich gegen Spam wehren können.

8.1 Kommentar-Einstellungen

Unter Einstellungen • Diskussion finden Sie die Einstellungsmöglichkeiten für die Kommentarfunktion. Sie können hier zum Beispiel die Kommentarfunktion für zukünftige Beiträge deaktivieren oder einstellen, dass die Kommentarfunktion nach X Tagen geschlossen wird.

Da diese Einstellungen auf Beitragsebene überschrieben werden können, bezieht sich etwa die Einstellung für die Deaktivierung der

Kommentarfunktion nur auf noch nicht veröffentlichte Einträge, weshalb Sie sich am besten vor dem Start Ihrer Website diese Einstellungen schon mal durchschauen sollten. Deshalb finden Sie eine nähere Beschreibung der verfügbaren Optionen in Kapitel 5, »Administration und Konfiguration von WordPress«, in Abschnitt 5.2.4, »Diskussion«.

Tipp: Vorhandene Kommentare ausblenden

Wenn Sie sich irgendwann entscheiden, dass Sie die Kommentarfunktion komplett deaktivieren möchten und bereits Kommentare geschrieben wurden, die Sie auch nicht mehr anzeigen möchten, ist der vielleicht einfachste Weg, den Kommentarbereich aus dem Theme zu entfernen. Arbeiten Sie dafür, wenn Sie ein Classic Theme nutzen, unbedingt mit einem Child-Theme, wie in Abschnitt 23.2.1, »Bestehendes Theme mit einem Child-Theme anpassen«, beschrieben wurde.

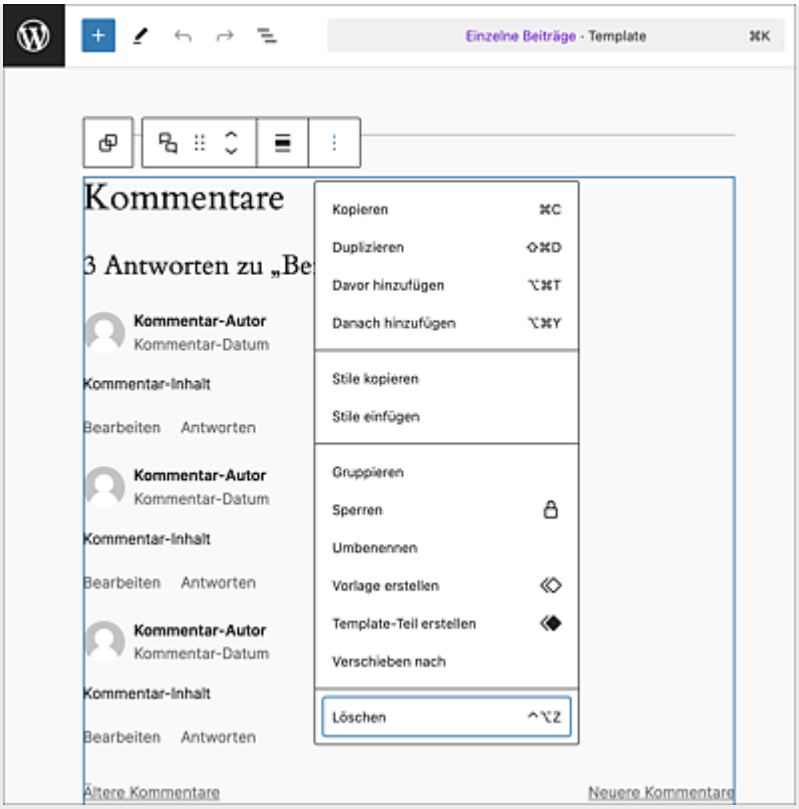


Abbildung 8.1 In einem Block-Theme können Sie den »Kommentare«-Block über den Website-Editor entfernen.

Suchen Sie im Theme nach der Funktion `comments_template()`. Am Beispiel von Twenty Twenty-One finden Sie den Aufruf in der *single.php*, *page.php* und *image.php*:

```
// If comments are open or there is at least one comment, load up the comment
template.
if ( comments_open() || get_comments_number() ) {
    comments_template();
}
```

Kopieren Sie die Dateien ins Child-Theme, und kommentieren Sie die Zeile mit `comments_template()` aus, sodass sie wie folgt aussieht:

```
// comments_template();
```

Laden Sie danach die Ansicht eines Beitrags oder einer Seite mit einem Kommentar neu, und der Kommentarbereich sollte

verschwunden sein.

Wenn Sie ein Block-Theme nutzen, können Sie den Kommentarbereich über den Website-Editor entfernen, ohne Code anpassen zu müssen. Das sehen Sie nachfolgend am Beispiel von Twenty Twenty-Four. Rufen Sie das Template für die Einzelansicht von Beiträgen auf, indem Sie im linken Admin-Menü über **Design** • **Website-Editor** den Website-Editor öffnen, in der linken Sidebar den Punkt **Templates** anklicken und rechts in der Templates-Liste das Template **Einzelne Beiträge** auswählen.

Im Template können Sie nun den **Kommentare**-Block entfernen und somit die Kommentarfunktion und Liste bestehender Kommentare löschen (siehe Abbildung 8.1). Wiederholen Sie den Vorgang für die anderen Templates, in denen der **Kommentare**-Block eingebunden ist.

8.2 Kommentare freigeben

Wenn Sie sich dazu entschieden haben, die Kommentarfunktion zu nutzen, wird irgendwann hoffentlich der Moment kommen, dass ein erster Kommentar geschrieben wird. Wenn Sie auf der Diskussion-Einstellungsseite die Option Bevor ein Kommentar erscheint,

• muss der Kommentar manuell freigegeben werden aktiviert haben (was empfehlenswert ist), werden Sie per Mail benachrichtigt, dass es einen neuen Kommentar gibt, der von Ihnen freigegeben werden muss. Wenn Sie jetzt die Seite Kommentare im linken Admin-Menü aufrufen, sehen Sie den neuen Kommentar in der Liste Alle farblich hervorgehoben.

Bewegen Sie den Cursor über den Eintrag, bekommen Sie unter dem Kommentarinhalt Aktionslinks angezeigt, wie Sie es auch schon aus den Backend-Übersichtsseiten für Seiten und Beiträge kennen (siehe Abbildung 8.2).

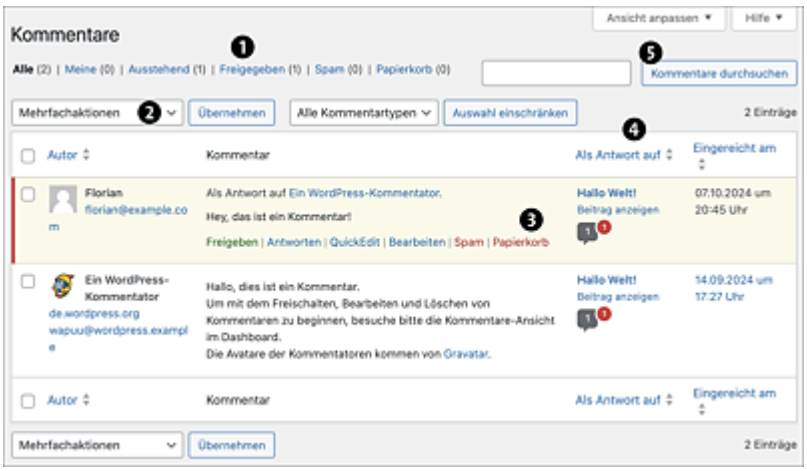


Abbildung 8.2 Die Liste der Kommentare ähnelt der für Seiten und Beiträge. Wenn ein Kommentar auf Freigabe wartet, wird er farblich hervorgehoben. Wenn Sie den Cursor über den Eintrag bewegen, erscheinen verschiedene Aktionslinks.

Ein Blick auf die Liste:

- **Kommentarfilter/-links ❶** und **❸**: Alle, Meine, Ausstehend, Freigegeben, Spam, Papierkorb

  Über diese Links können Sie, wie bei den Seiten- und Beitragsübersichten auch, die Liste nach Status filtern. Die Status bedeuten:

  - Alle zeigt alle Kommentare an, die nicht im Papierkorb liegen oder als Spam eingestuft sind.
  - Meine zeigt Ihre eigenen Kommentare an.
  - Ausstehend ist der Status für neue Kommentare, wenn die automatische Freischaltung nicht aktiv ist oder wenn fälschlich als Spam erkannte Kommentare aus dem Spam gerettet werden. Wenn Sie auf den grünen Freigeben-Link klicken, der erscheint, sobald Sie den Cursor über den Kommentar bewegen, wird der Kommentar in den Status Freigegeben geändert und ist live.
  - Freigegeben zeigt alle Kommentare an, die öffentlich auf Ihrer Website angezeigt werden. Bei freigegebenen Kommentaren gibt es statt des Freigeben-Links die Option Zurückweisen, um den Kommentar in den Ausstehend-Status zu schieben.
  - Im Spam landen Kommentare, die von einem entsprechenden Antispam-Plugin oder Ihnen manuell als Spam erkannt und markiert wurden. Wenn ein Kommentar fälschlicherweise als Spam eingestuft wurde, können Sie ihn über den Aktionslink Kein Spam in den Ausstehend-Status bewegen. Kommentare in diesem Status können von Antispam-Lösungen als Datenbasis für die Spam-Erkennung genutzt werden.
  - Im Papierkorb liegen die gelöschten Kommentare, die Sie über die Aktionslinks als Spam markieren oder Wiederherstellen können. Neben den beschriebenen Aktionen ist es in allen

Status auch möglich, den Kommentar in den Papierkorb zu verschieben oder aus Spam und Papierkorb endgültig zu löschen.

- Mehrfachaktionen/Kommentartypen ❷: Über das Dropdown Mehrfachaktionen unter der Liste zur Status-Filterung können Sie für mehrere Kommentare gleichzeitig den Status verändern. Über das Dropdown Alle Kommentartypen können Sie die Liste nach den Arten der Kommentare beziehungsweise Reaktionen filtern. Standardmäßig von WordPress unterstützt sind neben Kommentaren noch Pingbacks und Trackbacks, aber über Plugins können hier noch weitere Arten auftauchen, wie Webmentions (mehr dazu im IndieWeb-Kapitel).
- Als Antwort auf ❹ zeigt an, zu welchem Beitrag der Kommentar geschrieben wurde, und ermöglicht Ihnen, über zwei Links direkt in die Bearbeitungsansicht (verlinkter Titel des Beitrags) oder die Frontend-Ansicht des Beitrags (Beitrag anzeigen) zu gelangen. Außerdem sehen Sie, wie viele Kommentare zu dem Beitrag bereits online sind (die Zahl in der Sprechblase) und, optional, wie viele gerade ausstehend sind (Zahl im roten Kreis).
- Eingereicht am zeigt das Datum und die Uhrzeit des Kommentars an und ist bei freigegebenen Kommentaren ein Direktlink ins Frontend zu diesem Kommentar.
- Kommentare durchsuchen ❺: Wenn Sie einen Kommentar suchen, bei dem Sie Teile des Inhalts oder den Namen der Person kennen, die den Kommentar geschrieben hat, können Sie die Suchfunktion nutzen.

Tipp: Tastaturkürzel für die Bearbeitung von Kommentaren

Wenn Sie viele Kommentare zu moderieren haben, kann es schneller gehen, wenn Sie die Liste der ausstehenden

Kommentare mit Tastaturkürzeln abarbeiten. Diese Tastaturkürzel müssen Sie erst in Ihrem Profil aktivieren, indem Sie im linken Admin-Menü auf **Benutzer** • **Profil** gehen und im oberen Bereich **Persönliche Optionen** die Checkbox bei **Tastaturkürzel für die Kommentarmoderation aktivieren** setzen.

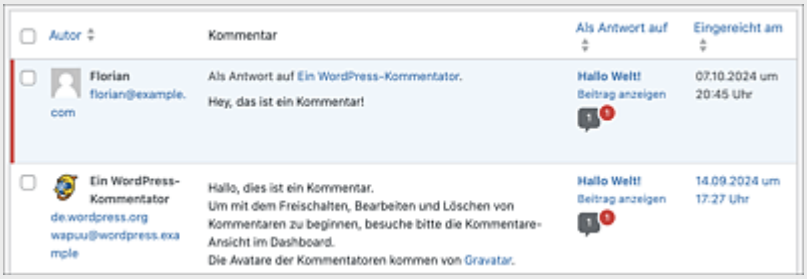


Abbildung 8.3 Der aktuell ausgewählte Kommentar wird mit einem blauen Hintergrund hervorgehoben.

Wechseln Sie über den Admin-Menüpunkt **Kommentare** zurück in die Kommentarübersicht und drücken Sie J oder K, um den ersten Kommentar auszuwählen, erkennbar am blauen Hintergrund. Mit K können Sie, wenn ein Kommentar ausgewählt ist, zum Kommentar darüber wechseln, mit J zum Kommentar darunter.

Wenn ein Kommentar markiert ist, stehen Ihnen folgende Tastaturkürzel zur Verfügung:

- A: **Freigeben** (approve)
- S: als **Spam** markieren
- D: in den **Papierkorb** verschieben oder **Endgültig löschen** (delete)
- Z: einen Kommentar aus dem **Papierkorb** wiederherstellen oder die **Rückgängig**-Aktion ausführen, die anstelle eines

Kommentars erscheint, der in den Papierkorb verschoben wurde

- U: in den **AUSSTEHEND**-Zustand **ZURÜCKWEISEN** (unapprove)
- R: auf den ausgewählten Kommentar antworten (reply); mit Esc können Sie das Antworten abbrechen, mit ⇆ zu anderen Elementen des Antwortformulars springen
- Q: öffnet die Quick-Edit-Ansicht für den Kommentar; mit Esc können Sie das Quick-Edit-Formular schließen und mit Tab durch die Elemente des Formulars navigieren
- E: öffnet die detailliertere Bearbeitungsansicht des ausgewählten Kommentars auf einer separaten Seite

8.3 Datenschutz-Bedenken und -Lösungen

Im Bereich der Kommentare gibt es ein paar Aspekte im Bereich Datenschutz zu beachten, auf die wir hier eingehen.

8.3.1 IP-Adresse bei Kommentaren nicht speichern

Wenn ein Kommentar geschrieben wird, speichert WordPress standardmäßig die *IP-Adresse*. Das ist aus Datenschutz-Aspekten ein Problem, da die IP-Adresse als personenbezogenes Datum angesehen wird und deshalb nicht einfach so gespeichert werden darf, wenn es nicht technisch notwendig ist.

Zum Glück lässt sich das recht einfach lösen, indem Sie den folgenden kleinen Code-Schnipsel entweder in einem eigenen Plugin einfügen oder in die *functions.php* eines Child-Themes (wie Sie ein eigenes kleines Plugin erstellen oder ein Child-Theme anlegen können, lesen Sie in Kapitel 23).

```
add_filter(
  'pre_comment_user_ip',
  function() {
      return '';
  }
);
```

Listing 8.1 Über den Filter »pre_comment_user_ip« können Sie die IP verändern, die zu einem Kommentar gespeichert wird.

Der Filter `pre_comment_user_ip` ermöglicht die Anpassung der IP-Adresse zu einem Kommentar, bevor der Kommentar gespeichert wird. Statt der IP wird ein leerer String zurückgegeben, und das ist der Wert, der statt der IP in der Datenbank gespeichert wird.

8.3.2 Profilbilder vom eigenen Server laden

Die Profilbilder, die viele Themes neben den Kommentaren anzeigen, werden standardmäßig direkt von den Servern des Dienstes »Gravatar« geladen, über den die Verknüpfung einer E-Mail-Adresse mit einem Bild funktioniert. Das ist problematisch, da so bei jedem Aufruf eines Ihrer Blogbeiträge die IP der Benutzer*in an die Server eines US-Unternehmens übertragen wird.

Im Kapitel »Administration und Konfiguration von WordPress« wurde in Abschnitt 5.2.4, »Diskussion«, bereits das Plugin »Avatar Privacy« (*https://de.wordpress.org/plugins/avatar-privacy/*) vorgestellt, das Abhilfe schafft: Das Plugin speichert die Profilbilder von Gravatar auf Ihrem Server zwischen, sodass sie als lokale Bilder eingebunden werden können und keine externen Requests mehr erzeugen.

8.3.3 DSGVO-Anfragen bearbeiten

Personen können auf Grundlage der Datenschutz-Grundverordnung Auskunft darüber verlangen, welche Daten bei einer Website über sie gespeichert sind, und/oder die Löschung der Daten beantragen. Bei der Kommentarfunktion werden neben dem Kommentartext der Name und die E-Mail-Adresse erfasst, und letztere Daten sind natürlich personenbezogen.

WordPress bringt Werkzeuge mit, um sowohl einen Export über personenbezogene Daten zu erstellen als auch die personenbezogenen Daten zu löschen beziehungsweise zu anonymisieren. Dafür gibt es im linken Admin-Menü unter Werkzeuge die Punkte Personenbezogene Daten exportieren (siehe Abbildung 8.4) und Personenbezogene Daten löschen (siehe

Abbildung 8.5). Die Seiten für Export und Löschen sehen ziemlich ähnlich aus, wie Sie in den beiden Abbildungen sehen können.

Wenn sich jemand bei Ihnen meldet, um Auskunft über die zu sich gespeicherten Daten zu erhalten oder diese löschen zu lassen, gehen Sie auf die entsprechende Seite und tragen im Feld Benutzername oder E-Mail-Adresse die E-Mail-Adresse oder den Benutzernamen der Person auf Ihrer Website ein.

Darunter gibt es die Checkbox Bestätigungs-E-Mail, mit der eine E-Mail an die E-Mail-Adresse geschickt wird, die in das Textfeld eingegeben wurde. In dieser E-Mail gibt es einen Link, um die Anfrage zu bestätigen. So können Sie sicherstellen, dass nicht jemand einfach eine Anfrage für eine fremde Mail-Adresse eingereicht hat.

Nachdem die Anfrage bestätigt wurde, können Sie in der Tabelle mit den Anträgen in der rechten Spalte Nächste Schritte einen Export-Link senden oder Personenbezogene Daten löschen.

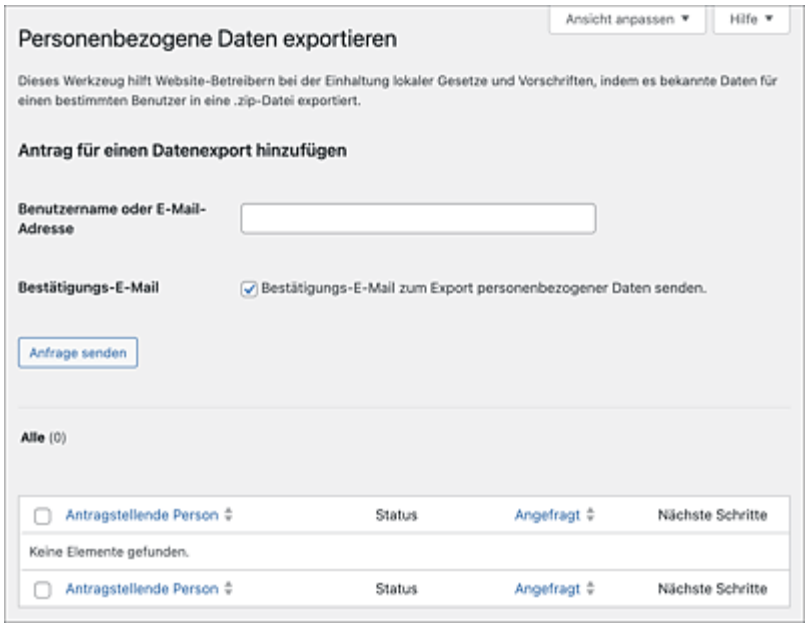


Abbildung 8.4 WordPress kommt mit einer Seite, auf der Sie einen Export von Daten einer Person durchführen können.

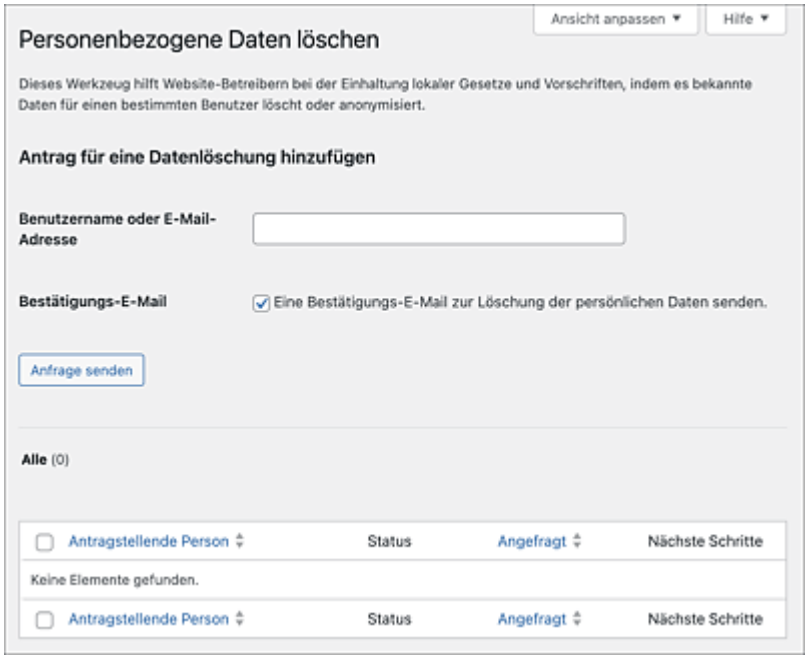


Abbildung 8.5 Auch für das Löschen beziehungsweise Anonymisieren von personenbezogenen Daten bietet WordPress ein Werkzeug.

8.4 Spamschutz mit Antispam Bee

Wenn Sie die Kommentarfunktion auf Ihrer Website aktiviert haben, ist es nur eine Frage der Zeit, bis die ersten Spam-Kommentare eintrudeln. WordPress bietet von Haus aus nur Optionen, Kommentare unter bestimmten Umständen nicht direkt freizugeben oder Kommentare auf Basis der »Kommentar-Sperrliste« direkt zu löschen.

Antispam-Plugins gibt es einige, und mit Akismet ist bei jeder normalen WordPress-Installation standardmäßig eins dabei, allerdings ist das Plugin aus Datenschutz-Sicht nicht unbedenklich, da die Daten der Kommentare zur Spam-Erkennung an Server in den USA gesendet werden.

Eine datenschutzfreundliche Alternative ist das Plugin »Antispam Bee«. Hier werden Kommentare standardmäßig nur lokal verarbeitet, und die Optionen, die Daten an Drittserver übertragen, müssen explizit aktiviert werden und sind ebenfalls mit dem Thema Datenschutz im Fokus entwickelt.

Nach der Installation von Antispam Bee finden Sie die Einstellungen unter Einstellungen • Antispam Bee (siehe Abbildung 8.6) – generell müssen Sie hier keine Änderungen vornehmen, da sinnvolle Standardwerte für die Optionen gesetzt sind.

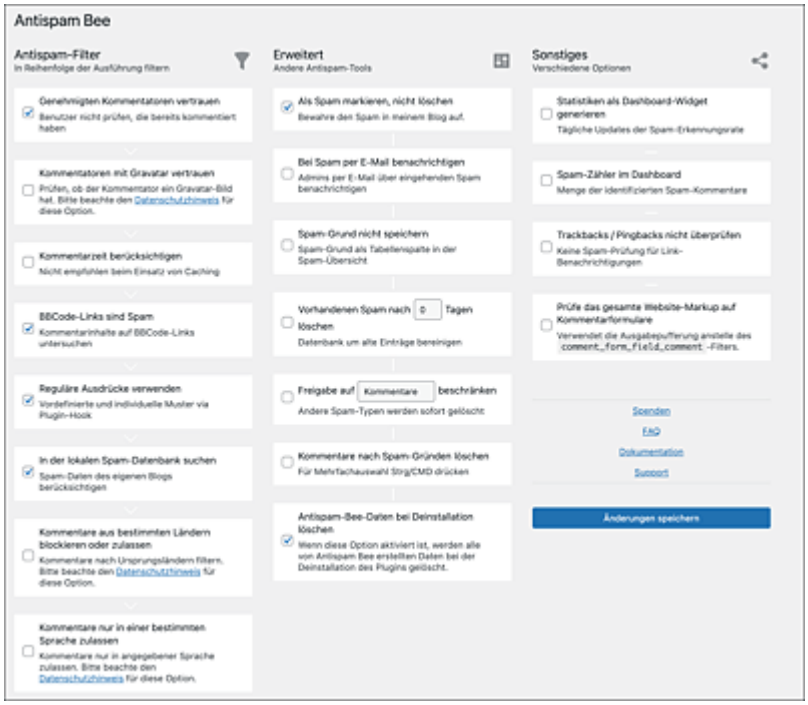


Abbildung 8.6 Die Einstellungen des Antispam-Plugins »Antispam Bee«

Der wohl wirkungsvollste Antispam-Mechanismus, den Antispam Bee mitbringt, ist der sogenannte Honeypot. Dabei handelt es sich um ein verstecktes Feld, das vom Plugin in das Kommentarformular eingefügt wird. Viele Spam-Bots erkennen nicht, dass das Feld versteckt ist, und füllen es dennoch aus – so können Kommentare klar als Spam erkannt werden.

Die weiteren Optionen zur Erkennung von Spam, die das Plugin anbietet:

- Autor*innen von bereits genehmigten Kommentaren vertrauen
- Kommentator*innen mit einem Gravatar vertrauen
- die Zeit berücksichtigen, nach der der Kommentar abgeschickt wurde
- Kommentare mit BBCode-Links als Spam einstufen

- Kommentare mit regulären Ausdrücken nach bestimmten Mustern durchsuchen (Antispam Bee bringt einige selbst mit, und über einen Hook können eigene definiert werden)
- die in der Datenbank vorhandenen Spamkommentare mit in die Erkennung einbeziehen
- Kommentare aus bestimmten Ländern zulassen/blockieren
- Kommentare nur in einer bestimmten Sprache erlauben

Neben den Antispam-Regeln bietet Antispam Bee noch verschiedene Aktionen, die mit erkanntem Spam durchgeführt werden können:

- erkannten Spam direkt löschen, statt ihn nur als Spam zu markieren
- per E-Mail bei Spam-Kommentaren benachrichtigen
- den Spam-Grund für einen Kommentar, den Antispam Bee standardmäßig speichert, nicht speichern
- vorhandenen Spam nach X Tagen löschen
- Spam bei bestimmten Spam-Gründen sofort löschen

8.5 Webmentions

Sogenannte Webmentions sind eine schöne Möglichkeit, den Kommentarbereich zu beleben, auch in Zeiten, in denen die Reaktionen auf einen Beitrag sich mehr auf Social-Media-Kanäle statt den Kommentarbereich verschoben haben. Mit Webmentions können Sie diese Reaktionen von sozialen Medien im Kommentarbereich abbilden. Dabei werden Antworten auf etwa einen Mastodon-Post auch als Kommentar angezeigt, Favoriten oder Reposts werden als kurze Info angezeigt.

Wie das genau funktioniert, lesen Sie in Kapitel 15 zum IndieWeb.

9 WordPress Multisite

Schon seit Version 3.0 ist WordPress ab Werk in der Lage, in einer einzigen Installation mehrere voneinander durchaus unabhängige Websites zu betreiben. In der Vergangenheit wurde dieser Modus oft »Blog-Netzwerk« genannt, heute ist eher der Begriff der »Multisite« gängig.

Auf den folgenden Seiten werden Sie lernen, wie Sie den Multisite-Modus in WordPress aktivieren können und warum Sie es in den meisten Fällen nicht tun sollten. Denn der Betrieb von Multisite-Netzwerken ist nicht ohne Tücken, und für viele Anwendungsgebiete gibt es mittlerweile deutlich praktischere Alternativen.

9.1 Was ist WordPress Multisite?

WordPress Multisite ermöglicht den Betrieb mehrerer Websites auf der Basis eines einzigen Sets an WordPress-Core-Dateien, Plugins und Themes. Die einzelnen Websites können nach außen komplett unabhängig wirken, müssen dies aber nicht sein. Sie können Websites in Ihrem Netzwerk unter jeweils eigenen Domains, unter verschiedenen Subdomains derselben Domain oder in Unterverzeichnissen derselben Domain betreiben. (Weitere Details zu den verschiedenen Modi finden Sie in Abschnitt 9.2.)

Als Betreiber*in einer Multisite können Sie Zugriffsrechte von Benutzer*innen auf die verschiedenen Websites im Netzwerk steuern, Plugins global oder für einzelne Sites aktivieren und die Theme-Auswahl der verschiedenen Websites festlegen.

Wie groß oder klein Sie das Ganze anlegen möchten, ist Ihnen weitgehend selbst überlassen. In den meisten Fällen besteht ein Multisite-Netzwerk aus einer einstelligen bis niedrigen zweistelligen Anzahl von Websites. Zumindest in der Theorie sind der Größe einer Multisite aber keine wirklichen Grenzen gesetzt. Das prominenteste Beispiel einer großen Multisite ist der *SaaS*-Dienst *WordPress.com*. Dort werden mehrere Millionen Websites aus einer einzigen Multisite heraus betrieben.

Achtung: Unterschied zwischen WordPress.com und WordPress

Durch die ähnlichen Namen von WordPress.com und WordPress kommt es regelmäßig zu Verwirrungen. Dieses Buch beschäftigt sich mit der Open-Source-Software *WordPress*, die Sie herunterladen und auf einem eigenen *Webspace* oder lokalen System installieren können, um eine Website zu betreiben und zu verwalten. WordPress.com ist eine kommerzielle Plattform, die von der Firma eines der WordPress-Mitgründer, Matt Mullenweg, betrieben wird. WordPress.com bietet allerlei Hosting-Dienste – kostenfreie wie kostenpflichtige – an. Dafür kommt zwar auch die Open-Source-Software WordPress zum Einsatz, allerdings in einer leicht modifizierten und beschränkten Version.

Die Größe Ihrer Multisite hängt am Ende also vor allem von Ihrem konkreten Anwendungsfall, Ihrem technischen Geschick und Budget für Server-Hardware ab. Je nach Komplexität der einzelnen

Websites, Zusammenstellung von Plugins und Last durch die Anzahl der Aufrufe kann der zuverlässige Betrieb einer Multisite durchaus anspruchsvoll sein und nicht nur tiefgreifendes Wissen zu PHP und MySQL erfordern, sondern auch einen souveränen Umgang mit der Server-Administration.

9.1.1 Vor- und Nachteile

Die wichtigsten Punkte zu Multisites haben Sie nun bereits kennengelernt. Schauen wir noch einmal genauer auf die wichtigsten Vor- und Nachteile.

Zentrale Installation

Das wesentliche Merkmal einer Multisite ist eine zentrale WordPress-Installation, aus der heraus mehrere Websites betrieben werden. Durch diesen Aufbau sparen Sie mit jeder weiteren Website, die Sie über eine Multisite betreiben, Speicherplatz und Dateien auf Ihrem Server. Denn für eine zweite traditionelle WordPress-Installation wären ja zumindest alle Dateien, aus denen der WordPress-Core besteht, ein zweites Mal nötig. Gleiches gilt für Plugins, die auch in beiden Websites installiert wären. Diese Ersparnis dürfte im größeren Kontext zweier Websites unerheblich sein. Bei größeren Anzahlen von Websites kann dieser Effekt aber durchaus ins Gewicht fallen.

Zentrale Verwaltung

Deutlich interessanter wird es, wenn wir uns mit der zentralen Verwaltung auf einen etwas offensichtlicheren Aspekt von WordPress-Multisites konzentrieren. Nach der Einrichtung können

Sie auf Netzwerkebene Plugins für alle Sites Ihres Netzwerks installieren, verwalten und natürlich auch entfernen. Wenn Sie also zum Beispiel das eine Spam-Schutz-Plugin auf allen Ihren Websites durch ein anderes ersetzen wollen oder überall ein neues Plugin für Kontaktformulare hinzufügen möchten, kann Ihnen eine Multisite eine Menge Zeit sparen. Gleiches gilt für Themes.

Die Verwaltung von Benutzer*innen gestaltet sich ähnlich und kann im täglichen Gebrauch sogar noch mehr Zeit sparen. Denn auch das Anlegen, Bearbeiten und Löschen von User-Accounts läuft innerhalb einer Multisite zentral ab. Sie können bestehende User zu Sites in Ihrem Netzwerk hinzufügen, sie mit anderen Berechtigungen ausstatten oder sie wieder entfernen. Für alle Benutzer*innen mit einem Account auf mehr als einer der Sites in Ihrem Netzwerk ergibt sich dabei der Vorteil, dass sie sich auf allen Sites mit demselben Username- und Passwort-Paar anmelden können.

Die Kehrseite der Medaille ist allerdings, dass Sie sich damit einen Flaschenhals, oder wie man im Englischen sagen würde, einen *single point of failure*, schaffen. Ein Problem beim Update Ihres WordPress-Core hat plötzlich nicht mehr nur Auswirkungen auf eine einzige Website, sondern gleich auf mehrere. Ein Plugin, dass nicht mit der neuesten PHP- oder WordPress-Version kompatibel ist, kann auf einmal ihr gesamtes Netzwerk lahmlegen. Gleiches gilt für Sicherheitslücken und Infektionen mit Schadcode. Befinden sich Ihre Websites hingegen in isolierten Umgebungen, hat eine Infektion einer Website nicht automatisch Auswirkungen auf eine andere. Bei der geteilten Code-Basis einer Multisite ist das fast unumgänglich.

Plugin-Kompatibilität

Im Hinblick auf Multisite lassen sich Plugins grundsätzlich in drei Kategorien sortieren: solche, die in Multisite-Installationen nicht funktionieren, solche, die weiter funktionieren wie auf Einzel-Installationen, und Plugins, die speziell für Multisites vorbereitet sind und Sie zum Beispiel netzwerkweite Einstellungen setzen lassen.

Die Erfahrung der letzten Jahre zeigt, dass die meisten Plugins mittlerweile in die mittlere Kategorie fallen und sich weitgehend problemlos auf Multisites nutzen lassen. Aber auch wenn Totalausfälle immer seltener werden, müssen Sie sich bewusst sein, dass der Einsatz von Multisite weitab vom Mainstream der WordPress-Nutzung ist. Es gibt immer noch eine Menge Entwickler*innen, die ihre Plugins nicht auf das Verhalten bei einem seltenen Anwendungsfall wie Multisite testen.

Medienverwaltung

In einer herkömmlichen Einzelinstallation haben Sie ein einziges Upload-Verzeichnis, in dem zum Beispiel verwendete Bilder oder PDF-Dateien von WordPress abgelegt werden. In einem Multisite-Setup legt WordPress pro Site in Ihrem Netzwerk ein solches Upload-Verzeichnis an. Sind die Sites in Ihrem Netzwerk inhaltlich komplett unabhängig voneinander, dann dürfte Sie dieses Verhalten kaum beeinflussen. Sind die Websites in Ihrer Multisite aber sehr ähnliche oder sogar nahezu identische Kopien, sammeln sich schnell viele Duplikate immer derselben Bilder auf Ihrem Server.

Eingeschränktes Hosting-Angebot

Wenn Sie von Anfang an planen, eine Multisite zu starten, sollten Sie direkt bei der Hosting-Auswahl (siehe auch Kapitel 4) darauf achten, dass das Hosting-Produkt Ihrer Wahl sich auf den Umgang mit Multisite versteht. Vor allem bei größeren deutschen Massenhostern gibt es da häufiger Probleme. Andere Anbieter werden dagegen explizit auf die Multisite-Kompatibilität ihrer Hosting-Produkte hinweisen.

9.1.2 Mögliche Anwendungsfälle

Auch wenn die Vorstellung, eine Multisite zu betreiben, im ersten Moment verlockend erscheinen mag, sollten Sie diesen einschneidenden Schritt nur dann gehen, wenn Sie tatsächlich auch eine Verwendung für die Werkzeuge haben, die WordPress Multisite Ihnen zur Verfügung stellt.

Mehrsprachigkeit

WordPress selbst ist aktuell noch nicht in der Lage, zeitgleich Inhalte ordentlich strukturiert in mehreren Sprachen bereitzustellen. Eine solche Funktion ist zwar geplant, wird aber noch einige Jahre auf sich warten lassen. In der Zwischenzeit bieten sich für mehrsprachige Websites allerlei Plugins, die das Verwalten und Ausspielen von verschiedensprachigen Inhalten ermöglichen und erleichtern sollen. Wie bei so vielen Problemen, denen Sie sich beim Betrieb einer WordPress-Website gegenübersehen können, finden sich auch für diesen Fall unzählige Plugins. Weil einige davon aber vor allem bei größeren Websites zu Performance-Problemen führen können, hat sich die Nutzung eines Multisite-Netzwerks als Alternative bewährt.

Besonders charmant ist dieser Ansatz, weil er – zumindest in der Theorie – komplett ohne zusätzliche Plugins auskommen kann. Jede Website in der Multisite bildet in diesem Setting eine eigene Sprache ab. Über Menüeinträge können Besucher*innen zwischen den verschiedenen Sprachversionen wechseln. Abrunden können Sie das Ganze dann aber noch durch Plugins wie das kostenfreie Multisite Language Switcher.

Plugin	Multisite Language Switcher
Aktive Installationen	5.000+
Download	*https://de.wordpress.org/plugins/multisite-language-switcher/*
Zweck	einfache Verknüpfung mehrsprachiger Inhalte in einer Multisite
Komplexität	+

Multisite Language Switcher macht das Verknüpfen von Inhalten, zum Beispiel über das Menü, deutlich komfortabler. Während Sie mit Bordmitteln entweder sehr komplex manuell Verknüpfungen pflegen und unzählige Versionen desselben Menüs pflegen müssten, ermöglicht MLS es, dynamisch Verlinkungen zu jeweils passenden Versionen eines Inhalts im Menü anzubieten.

Plugin	MultilingualPress
Download	*https://multilingualpress.org*
Zweck	erweiterte Verwaltung und Verknüpfung mehrerer Sprachversionen in einer Multisite
Komplexität	++

MultilingualPress geht noch einen Schritt weiter und gibt Ihnen allerlei Werkzeuge an die Hand, um mehrsprachige Inhalte in Ihrer Multisite noch effizienter zu verwalten.

Franchises

Die Stärke von Multisite liegt im Management vieler ähnlicher Websites. Zu große Unterschiede in den verwendeten Plugins und Themes können das Plus an Komfort, dass Sie durch eine Multisite gewinnen, schnell zunichtemachen. Wenn Sie aber zum Beispiel eine Website für ein Unternehmen pflegen, das als Franchise verschiedene Filialen hat, die halbwegs eigenständige Websites betreiben sollen, könnte Multisite der richtige Ansatz für Sie sein.

In diesem Szenario können Sie als Admin des Multisite-Netzwerks mit der Auswahl eines Themes ein Design für alle Sites vorgeben, Plugins zur Verfügung stellen, die auf den Sites genutzt werden sollen, und vielleicht sogar ein Set an Standard-Inhalten für jede neue Site bereitstellen. Jede Franchise-Nehmer*in erhält von Ihnen einen Admin-Account für die jeweils eigene Filial-Website. Auf ihren jeweiligen Websites können die Admins in einem gewissen Umfang schalten und walten, Inhalte anpassen, eigene schreiben und sogar weitere User anlegen, ohne sich jemals um Backups, Updates oder andere Verwaltung kümmern zu müssen. Denn das ist dann *Ihr Job*.

Kunden-Websites

Dieses Szenario wird in den meisten Fällen keines sein, für das Multisite zwingend ein guter Anwendungsfall ist. Weil es von dieser Regel aber einige Ausnahmen geben kann, wollen wir ihn hier trotzdem kurz behandeln.

Wie schon erwähnt, ist der Einsatz von Multisite vor allem dann sinnvoll, wenn Sie mehrere technisch sehr ähnliche Websites betreiben. Bei Sites für Kund*innen ist das in der Regel nicht der Fall. Gehen wir aber für einen Moment davon aus, dass Sie für eine Agentur arbeiten, die einfache Landingpages anbietet und diese möglichst effizient verwalten möchte.

Ein Punkt, den Sie in diesem Fall unbedingt beachten sollten, ist der, dass Sie in einer Multisite zwar eine User-Verwaltung haben, die grundsätzlich den Zugriff eines Admins auf eine (oder mehrere) definierte Website(s) beschränken kann, was aber nicht bedeutet, dass die Sites Ihrer Multisite in isolierten Umgebungen laufen. Die Daten aller Ihrer Nutzer*innen liegen in derselben Datenbank, alle Dateien auf demselben Dateisystem. Eine Sicherheitslücke in einem Plugin betrifft fast automatisch alle Websites in Ihrer Multisite. Prüfen Sie also genau, ob eine Multisite für Ihren Anwendungsfall wirklich eine gute Wahl ist.

Blog-Farmen

Besonders an Universitäten war es eine Zeit lang sehr populär, Studierenden und anderen Angehörigen der Uni eine Möglichkeit zu bieten, ihre eigenen Websites und Blogs zu betreiben. Das mag in den letzten Jahren etwas aus der Mode gekommen sein, ist aber immer noch ein spannender Anwendungsfall. Wie wäre es mit eigenen Blogs für alle Mitglieder des Leichtathletik-Vereins oder für die Schüler*innen der Klasse 7a?

Diese Beispiele sind am nächsten an *WordPress.com*, der vermutlich größten WordPress-Multisite, und erfordern daher in vielen Fällen keine besonderen Plugins.

9.2 Die Multisite-Einrichtung

9.2.1 Vorbereitung

Bevor Sie Ihre Multisite einrichten können, müssen Sie sich noch über einige Punkte Gedanken machen.

Neue Installation

Während Sie grundsätzlich jede WordPress-Installation in eine Multisite umwandeln können, empfiehlt es sich, für diesen Schritt zunächst eine komplett neue Installation anzulegen. Vor allem bei älteren Websites kommt es ansonsten gelegentlich zu Problemen mit bestehenden Plugins, Weiterleitungen oder exotischeren Inhaltstypen.

In älteren Installationen wird WordPress Ihnen deshalb die Möglichkeit zum Anlegen einer Unterverzeichnis-Installation nicht anbieten. Sollte genau das der Modus sein, den Sie verwenden möchten, so führt kein Weg um eine neue Installation herum. Aber auch wenn Sie die anderen Modi grundsätzlich auf älteren Websites aktivieren können, empfiehlt es sich, eine neue Installation dafür zu nutzen.

Sollten Sie sich dennoch entscheiden, eine bestehende WordPress-Installation als Grundlage für eine Multisite zu nutzen, so fertigen Sie unbedingt ein Backup der bestehenden Site an. Mehr zum Thema Backups finden Sie in Abschnitt 16.3.

Grundsatzfrage: Domain-Modus

WordPress bietet Ihnen im Rahmen des Multisite-Setups grundsätzlich zwei (eigentlich drei) verschiedene Modi an, aus denen sie wählen können – und müssen.

Subdomains

Sie können alle Websites in einem Netzwerk unter verschiedenen Subdomains betreiben. Bleiben wir bei unseren Beispielen aus 9.1.2 und läuft Ihr Netzwerk unter der Domain *example.com*, dann könnten die jeweiligen Unterseiten zum Beispiel *de. example.com*, *en.example.com* und *es.example.com* für verschiedene Sprachversionen oder *flensburg.example.com, bonn.example.com* und *konstanz.example.com* für Standorte Ihres Franchise-Unternehmens sein.

Tipp: Wildcard-Subdomain für einfachere Subdomain-Multisite

Damit eine Multisite mit Subdomains funktioniert, müssen die jeweiligen Subdomains auf das Verzeichnis Ihres Servers zeigen, in dem Sie WordPress installiert haben. Diese Konfiguration können Sie manuell für jede Subdomain vornehmen oder bei den meisten Anbietern auch durch eine sogenannte Wildcard. Diese *Wildcard* greift dann automatisch für alle Subdomains einer Domain, die nicht anderweitig konfiguriert sind, sodass Sie, wenn in der Zukunft eine weitere Subdomain zu Ihrem Multisite-Netzwerk hinzukommt, nicht noch einmal manuell die Konfiguration bemühen müssen.

Achten Sie bei diesem Setup unbedingt darauf, dass für jede Subdomain (ebenso wie für die Haupt-Domain) ein SSL-Zertifikat

genutzt wird, um die Daten Ihrer Websites verschlüsselt per https zu übertragen.

Unterverzeichnisse

Der zweite Modus, der Ihnen bei der Einrichtung zur Verfügung steht, legt die Sites Ihres Netzwerks in Unterverzeichnissen ab. Also *example.com/de*, *example.com/en*, *example.com/es* für verschiedene Sprachversionen oder *example.com/flensburg*, *example.com/bonn*, *example.com/konstanz* für Standorte. In diesem Modus kann eine Multisite nach außen wie eine einzige Website wirken, weil Ihre Besucher*innen nicht zwischen verschiedenen Subdomains hin und her wechseln.

Vor allem für Fälle, in denen Sie Probleme bei der Einrichtung von Domains in Ihrem Hosting haben oder die Multisite vorrangig zum Testen einsetzen wollen, kann dieser Modus nützlich sein.

Domains

Auch wenn während der Einrichtung nur die beiden bisher vorgestellten Optionen zur Verfügung stehen, sind Ihre Wahlmöglichkeiten in der Praxis noch mal etwas größer.

In der Vergangenheit waren noch zusätzliche Plugins nötig, um Multisites so umzubiegen, dass man in ihnen tatsächlich mehrere eigenständige Domains für die verschiedenen Sites benutzen konnte. Das ist mittlerweile nicht mehr der Fall. Stattdessen können Sie einen Kniff in der Oberfläche von WordPress-Multisite nutzen und nach dem Anlegen einer Site einfach deren Domain anpassen und individuell vergeben.

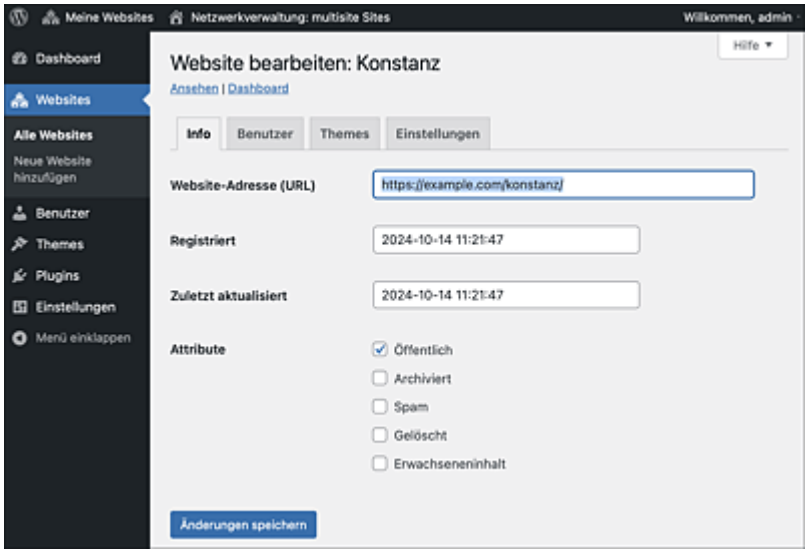


Abbildung 9.1 In den Detail-Einstellungen jeder Site Ihres Netzwerks können Sie nach der Installation die (Sub)domain anpassen (siehe Abschnitt 9.3).

Mit diesem zusätzlichen Schritt sind auch Mischformen möglich. Haben Sie Ihre Multisite zum Beispiel mit Unterverzeichnissen eingerichtet, so können Sie Sites Ihres Netzwerks in dieser Standard-Einstellung nutzen (*example.com/de* und *example. com/en*), während Sie gleichzeitig eine Subdomain einsetzen (*bonn.example.com*) oder direkt eine komplett eigene Domain für eine Site vergeben. Vermeiden Sie hier nach Möglichkeit aber unnötige Komplexität – Ihr zukünftiges Ich wird es Ihnen danken.

Auch bei der Verwendung von Domains müssen diese bei Ihrem Hosting-Anbieter so konfiguriert sein, dass sie auf das Verzeichnis auf Ihrem Server zeigen, in dem Ihre WordPress-Multisite liegt.

9.2.2 Aktivieren des Multisite-Modus

Nachdem Sie alle Vorbereitungen abgeschlossen haben, können Sie endlich daran gehen, Ihre neue Multisite einzurichten. Dafür benötigen Sie zunächst (S)FTP-Zugriff auf Ihren Webserver und einen Admin-Account für das betreffende WordPress.

Plugins deaktivieren

Zunächst loggen Sie sich als Admin auf der Website ein, die Sie in eine Multisite umwandeln möchten. Dort deaktivieren Sie alle Plugins. Besonders bei bereits bestehenden Sites ist dieser Schritt unumgänglich. Aber auch bei einer komplett neuen WordPress-Installation können aktive Plugins vorhanden sein – vor allem, wenn Sie die Installation nicht selbst durchgeführt haben, sondern zum Beispiel auf den Installer Ihres Hosting-Anbieters zurückgegriffen haben.

wp-config.php anpassen

Mit Ihrem FTP-Client können Sie anschließend die Konfigurationsdatei von WordPress *wp-config.php* herunterladen und bearbeiten. Dort finden Sie eine Zeile, die sie darüber informiert, dass Änderungen an dieser Datei nicht oberhalb stattfinden sollten. Nach dieser Zeile können Sie dieses Code-Schnipsel in einer neuen Zeile hinzufügen:

```
define( 'WP_ALLOW_MULTISITE', true );
```

Wenn Sie diese Änderung gespeichert und wieder auf Ihren Server hochgeladen haben, liest WordPress bei nächster Gelegenheit die neue Konfiguration und erkennt, dass Sie den Multisite-Modus aktiviert haben. Es wird entsprechend darauf reagieren.

Zurück im WordPress-Interface sollten Sie sich nun einmal aus- und wieder einloggen.

9.2.3 Konfiguration der Multisite

Einstellungen im Interface

Nach dem erfolgreichen Login sollten Sie im Admin-Menü auf der linken Seite unter WERKZEUGE einen Eintrag NETZWERK-EINRICHTUNG finden. Sollte Ihnen dieser Menüpunkt oder der WERKZEUGE-Menüpunkt fehlen, überprüfen Sie zunächst, ob Ihr User-Account in WordPress wirklich über Admin-Berechtigungen verfügt. Wenn das der Fall ist und Sie dennoch keine Netzwerk-Einrichtung angezeigt bekommen, überprüfen Sie noch einmal Ihre Anpassungen an der *wp-config.php* Datei und deren korrekte Übertragung auf den Webserver, auf dem Ihre WordPress-Website liegt.



Abbildung 9.2 Wenn Ihre Anpassungen erfolgreich waren, finden Sie unter dem Menüpunkt WERKZEUGE einen neuen Eintrag NETZWERK-EINRICHTUNG.



Abbildung 9.3 Hier können Sie Ihre Multisite einrichten und grundlegende Entscheidungen treffen.

Ist auch dieser Klick von Erfolg gekrönt, sollten Sie jetzt mit einer überschaubaren, aber folgenschweren Einstellungsseite konfrontiert sein. Neben einigen einleitenden Worten werden Sie direkt ins kalte Wasser geworfen und müssen sich entscheiden, wie Sie mit Adressen in ihrem Multisite-Netzwerk umgehen möchten. Wenn Sie dieses Kapitel bis hierher gelesen haben, sollte Sie diese Frage aber nicht mehr überraschen, und Sie können sich schnell für Subdomains oder Unterverzeichnisse entscheiden.

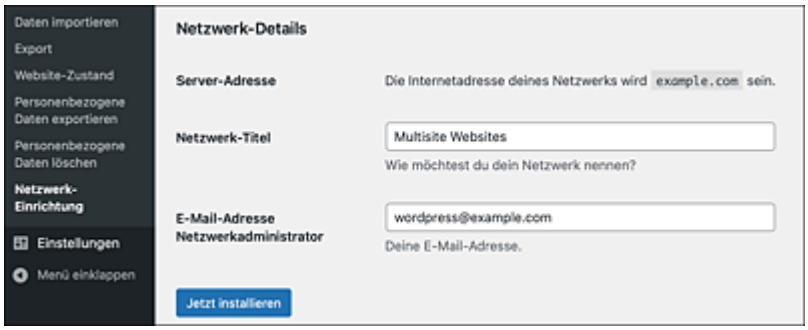


Abbildung 9.4 Zur Netzwerk-Einrichtung gehört auch die Angabe einiger Details, die von WordPress im laufenden Betrieb später verwendet werden.

Im zweiten Abschnitt der Multisite-Einrichtung informiert WordPress Sie zunächst über die Server-Adresse, die zukünftig für Ihre Multisite eingesetzt werden wird. Hierbei handelt es sich aller Voraussicht nach um die Domain, unter der Ihre Installation aufgesetzt wurde. Verstehen Sie diese Information einfach als

zusätzlichen Hinweis und eine Möglichkeit zu kontrollieren, dass Sie wirklich an der richtigen WordPress-Installation herumschrauben.

Bleiben noch zwei Felder auszufüllen, bevor Sie den nächsten Schritt auf dem Weg zum funktionierenden Multisite-Netzwerk geschafft haben. Der Netzwerk-Titel wird vor allem intern benutzt, um Ihrer Multisite einen Namen zu geben. Das Feld E-Mail-Adresse Netzwerkadministrator sollte mit einer E-Mail-Adresse gefüllt werden, unter der Sie oder eine beauftragte Person zukünftig Nachrichten zu Problemen des Netzwerks oder anstehenden Updates erhalten werden.

Beide Angaben können Sie auch später, nach der Einrichtung der Multisite, über die Netzwerk-Verwaltung (siehe Abschnitt 9.3) anpassen.

Überprüfen Sie Ihre Angaben noch einmal, bevor Sie diesen Schritt der Installation mit einem Klick auf Jetzt Installieren abschließen.

Anpassung der .htaccess

Basierend auf Ihren Eingaben aus dem vorherigen Schritt generiert WordPress Ihnen nun zwei Code-Schnipsel, die Sie – wie eingangs bereits geübt – mittels FTP-Client auf Ihren Server übertragen müssen.

Der erste Block von Netzwerkkonfigurationsregeln, den WordPress für Sie generiert, können Sie an dieser Stelle kopieren. Er enthält Angaben zur Netzwerk-Konfiguration, die Sie in die WordPress-Konfigurationsdatei *wp-config.php* einfügen. Diese Datei haben wir schon zum Start des Installationsprozesses bearbeitet.

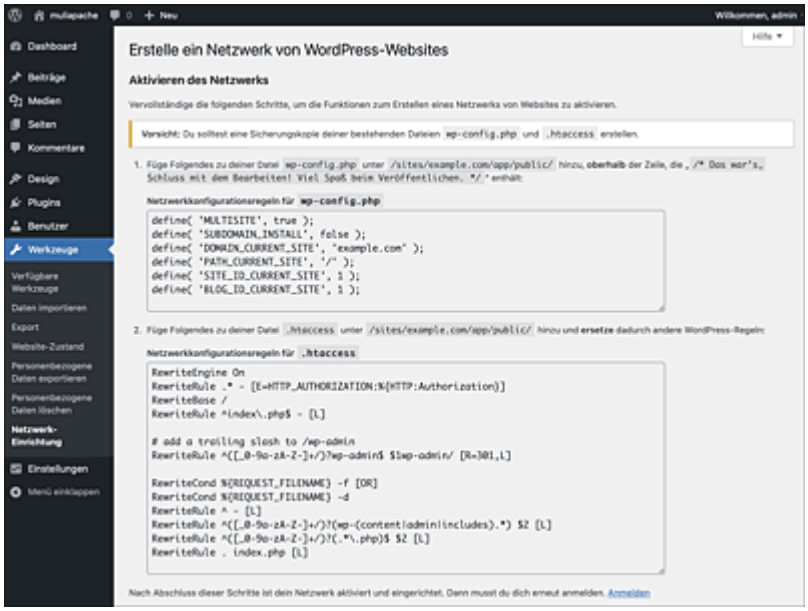


Abbildung 9.5 Auf der folgenden Seite präsentiert WordPress Ihnen zwei Code-Schnipsel, die Sie in verschiedene Dateien Ihrer WordPress-Installation einfügen können.

Der zweite Code-Block, den Sie kopieren und einfügen müssen, wird von WordPress direkt unter dem vorherigen angezeigt. Er enthält allerlei Weiterleitungsregeln für die *.htaccess*-Datei. Diese Datei finden Sie im Dateisystem Ihres Hosting-Pakets auf derselben Ebene wie die *wp-config.php*. Laden Sie nun die Änderungen an beiden Dateien hoch. Als abschließenden Schritt müssen Sie sich in Ihrer WordPress-Installation jetzt noch einmal ab- und wieder anmelden.

9.3 Bedienung des Netzwerks und Unterschiede zum Standard

Nach erfolgter Installation und erfolgreichem Einloggen sollten Sie sich nun vor einer fertigen Multisite-Installation sehen. Beleuchten wir also einmal kurz, wie sich Ihre Multisite von einer normalen WordPress-Installation unterscheidet.

9.3.1 Eine neue User-Rolle

In einer WordPress-Multisite unterscheiden sich die Berechtigungen, die einzelne Nutzer*innen haben, leicht von denen einer Standard-Installation.

Zunächst einmal ergänzt die Rolle des Super-Administrators die bestehenden User-Rollen Administrator, Redakteur, Autor, Abonnent. Super-Admins können Einstellungen auf Netzwerk-Ebene setzen, neue Sites in der Multisite anlegen und bestehende löschen. Außerdem können nur Super-Admins andere User mit Super-Admin-Rechten ausstatten.

Tipp: Sparsamer Umgang mit Berechtigungen

Behalten Sie bei der Vergabe der Rechte für Admins und Super-Admins immer im Hinterkopf, dass eine höhere Anzahl an Accounts mit diesen Rechten Ihre Website potenziell gefährdet, weil die Möglichkeiten für Hacking-Versuche und unsichere Passwörter mit jedem zusätzlichen Account wachsen.

In Kapitel 18 finden Sie effektive Mittel, um diese Gefahren etwas zu reduzieren. Dennoch lohnt es sich, vor der Vergabe von

Rechten zu überprüfen, was eine Person mit ihrem Account am Ende tatsächlich tun soll. Für den Mitarbeiter aus dem Vorzimmer, der regelmäßig neue Beiträge schreibt, mögen Redakteurs-Rechte ausreichen. Und auch für die Geschäftsführerin, die gern selbst an der Website mitarbeitet, müssen es nicht automatisch Admin-Rechte sein.

Außer Super-Admins müssen alle anderen User explizit zu einer Site im Netzwerk hinzugefügt werden, damit sie Zugriff auf deren Verwaltung erlangen. Die normalen WordPress-User-Rollen können einem Account je nach Site unterschiedlich zugewiesen werden. Während ich also Admin von Site A bin, kann ich auf Site B und C als Redakteur hinzugefügt sein.

Im Vergleich zu herkömmlichen WordPress-Installationen wurden die Berechtigungen von Admins leicht eingeschränkt. Diese können standardmäßig keine Plugins und Themes mehr installieren, sondern nur diejenigen aktivieren, die für die Multisite zur Verfügung gestellt wurden. Außerdem können sie Sites nicht löschen und auch keine neuen hinzufügen.

9.3.2 Das Netzwerk-Verwaltungs-Interface

Der Website-Switcher

Die erste Änderung, die Ihnen in der WP-Admin-Oberfläche ins Auge fallen könnte, ist eine Ergänzung in der Werkzeugleiste am oberen Bildschirmrand. Hier finden Sie links vom Namen der Website, der Sie mit einem Klick von WP-Admin auf die Website und zurück bringt, ein weiteres Element.

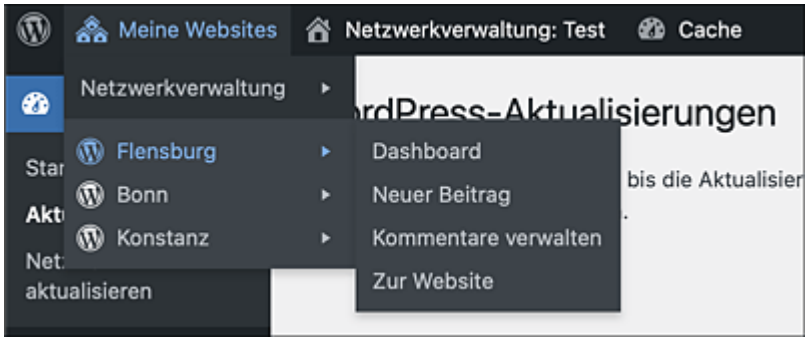


Abbildung 9.6 Als User mit Zugriff auf mehr als eine Site in einem Multisite-Netzwerk sehen Sie in der Werkzeugleiste oben links ein Element zum Wechsel zwischen den Sites.

Ein Klick auf MEINE WEBSITES befördert Sie zu einer Übersicht über alle Sites, auf die Sie Zugriff haben. Direkt und ohne Umweg können Sie zu diesen Sites aber auch wechseln, wenn Sie Ihre Maus über den Punkt MEINE WEBSITES bewegen. In einem Ausklappmenü listet WordPress Ihnen hier dieselben Websites auf und erlaubt Ihnen den schnellen Wechsel zu den jeweiligen WP-Admin-Interfaces. Als ersten Eintrag in diesem Untermenü sehen Sie außerdem NETZWERKVERWALTUNG nebst eigenen Unterpunkten. Ein Klick auf diesen Eintrag bringt Sie direkt in ein komplett neues Interface Ihres Multisite-Netzwerks.

Die Netzwerkverwaltung

Der größte Teil des neuen Multisite-Interface findet sich in einem gesonderten Bereich, auf den nur Super-Admins Zugriff haben. Sie erreichen diesen Bereich über den Punkt NETZWERKVERWALTUNG im zuvor beschriebenen Website-Switcher oder durch das Anhängen von */network/* an den Pfad zum WP-Admin-Interface Ihrer Site. Statt *example.com/wp-admin* finden Sie die Netzwerkverwaltung also unter *example.com/wp-admin/network*.

Einmal geöffnet präsentiert sich Ihnen hier ein Interface, das stark an WP-Admin angelehnt ist. Neben der Werkzeugleiste am oberen

Rand finden Sie links ein Admin-Menü und den eigentlichen Inhalt der einzelnen Einstellungsseiten rechts.

In der ersten Seite – dem Dashboard – werden Sie von Dashboard-Widgets empfangen, die Sie auch in einer normalen WordPress-Installation vielleicht schon gesehen haben, hier deutlich reduziert auf die beiden Widgets Auf einen Blick und WordPress-Veranstaltungen und Neuigkeiten. Während das Widget WordPress-Veranstaltungen und Neuigkeiten unverändert bleibt, präsentiert sich Auf einen Blick in einer Version speziell für Multisites. Hier bekommen Sie einen schnellen Überblick über die aktuelle Anzahl von Websites und User-Accounts in Ihrem Multisite-Netzwerk. Die beiden Links oben im Widget erlauben das Anlegen von neuen Sites und Usern. Zwei Suchfelder unten im Widget runden das übersichtliche Gesamtpaket ab. In Abschnitt 6.1.2 gehen wir noch einmal genauer auf die Widgets ein, die Sie an dieser Stelle in Einzelinstallationen sehen könnten.

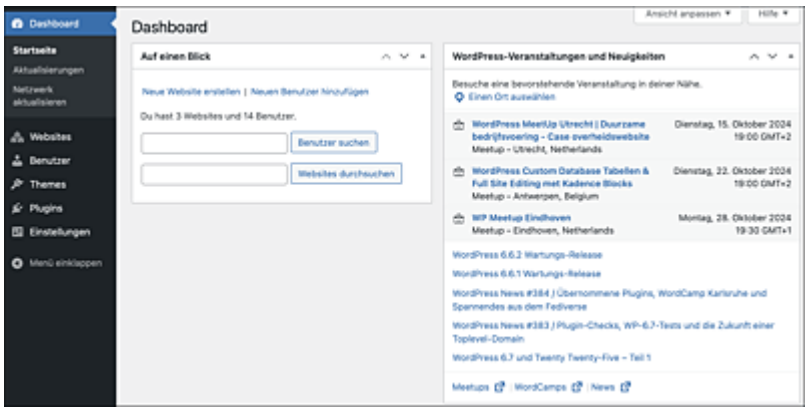


Abbildung 9.7 Das Dashboard der Netzwerkverwaltung ist stark an ein normales WordPress-Dashboard angelehnt.

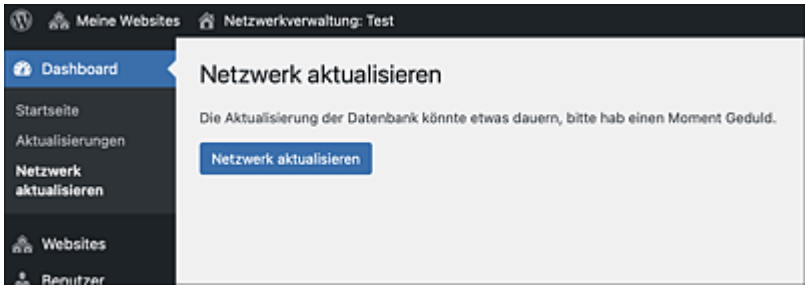


Abbildung 9.8 Auf der Seite »Netzwerk aktualisieren« können Sie sich nach größeren Updates um die Datenbank Ihrer Multisite kümmern.

Wenn wir diese Optionen für einen Moment links liegen lassen und uns am Admin-Menü entlangarbeiten, gelangen wir als Nächstes zur Unterseite AKTUALISIERUNGEN. Diese unterscheidet sich aber nicht von der entsprechenden Seite in Standardinstallationen. Der folgende Untermenüpunkt NETZWERK AKTUALISIEREN ist hingegen neu.

Diese Seite besteht beinahe nur aus einem einzigen großen Button NETZWERK AKTUALISIEREN. Im Rahmen größerer WordPress-Core-Updates kann es dazu kommen, dass auch eine Aktualisierung der Datenbank erforderlich wird. Auf Einzelinstallationen wird diese Aktualisierung einfach im Rahmen des Auto-Updates mit durchgeführt. Weil eine Multisite aber durchaus aus einer größeren Menge von Websites bestehen kann, wurde der Prozess des Datenbank-Updates hierfür vom eigentlich Update-Vorgang getrennt und muss manuell ausgelöst werden.

Nach einem Klick auf den Button arbeitet WordPress die Websites in Ihrem Netzwerk in Paketen von je fünf Sites ab. Entsprechend der Größe Ihres Netzwerks kann der ganze Vorgang wenige Sekunden bis zu mehreren Minuten dauern. In dieser Zeit lädt die Seite mehrfach neu, jeweils wenn die Datenbank-Einträge von fünf Sites fertig aktualisiert sind.

Sie können jederzeit auf diese Seite gelangen und den Button klicken. Wenn ein Datenbank-Update erforderlich ist, wird

WordPress Sie aber mit einem Banner im Dashboard gesondert darauf hinweisen.

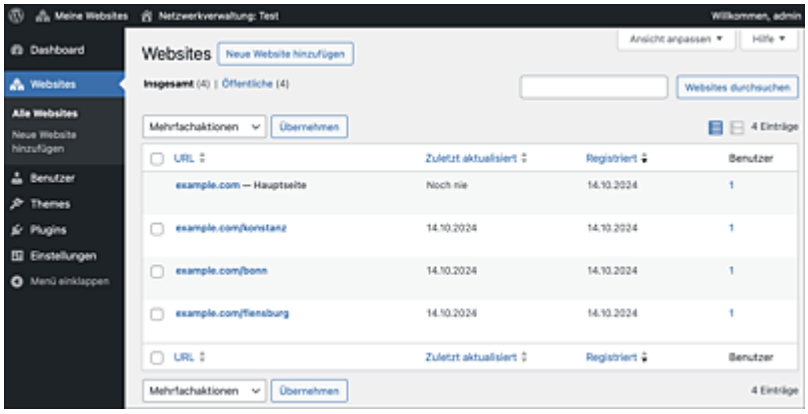


Abbildung 9.9 Unter dem Eintrag **WEBSITES** finden Sie alle Sites, die in Ihrem Multisite-Netzwerk angelegt sind, und können weitere hinzufügen.

Der nächste Menüpunkt nimmt die Websites Ihrer Multisite in den Fokus. Ein Klick auf den Menüeintrag WEBSITES befördert Sie in eine Auflistung aller Sites in Ihrem Netzwerk.

Die Tabelle auf dieser Seite listet alle Websites auf. Hier sehen Sie neben der jeweiligen (Sub-)Domain die Daten der initialen Registrierung und letzten Aktualisierung sowie die Anzahl der mit einer Site verknüpften User-Accounts.

Die erste Website in der Tabelle ist mit dem Zusatz HAUPTSEITE gekennzeichnet. Weil diese Site dieselbe Domain wie das Multisite-Netzwerk selbst nutzt, können Sie sie nicht löschen oder archivieren. Alle anderen Sites in ihrem Netzwerk – sollten Sie jetzt schon mehr als eine haben – können auch wieder entfernt werden. Die Einstellungen dafür können Sie erreichen, indem Sie Ihre Maus über die entsprechende Zeile in der Tabelle bewegen. Dann sollte Ihnen eine Reihe von Links angezeigt werden.

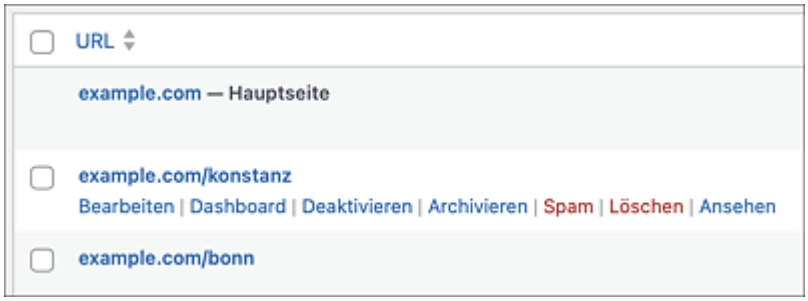


Abbildung 9.10 Wenn Sie Ihre Maus über Einträge in der Tabelle bewegen, zeigt WordPress Ihnen weitere Aktionen für die jeweilige Website in Ihrem Netzwerk an.

Bearbeiten, Dashboard und Ansehen stehen Ihnen bei allen Websites – auch der Hauptseite – zur Verfügung. Bearbeiten erlaubt das Setzen von Einstellungen. Dashboard bringt Sie zur jeweiligen WP-Admin-Ansicht und Ansehen auf die Website.

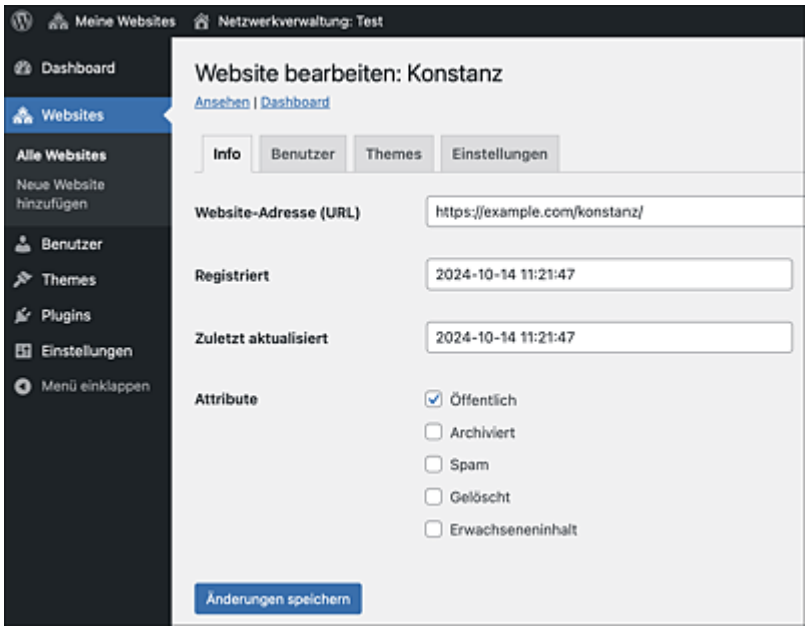


Abbildung 9.11 Die Bearbeiten-Ansicht einer Website kann von Plugins erweitert werden, ist aber immer aus verschiedenen Tabs zusammengesetzt.

Auf weiteren Sites stehen Ihnen in dieser Tabelle zusätzlich auch noch Links für Deaktivieren, Archivieren, Spam und Löschen zur Verfügung. Die ersten beiden Optionen verhalten sich nahezu identisch. In beiden Fällen wird die betreffende Website in der Übersicht rötlich markiert. Super-Admin-User können sie weiterhin

sehen und wiederherstellen. Alle Uploads und Daten bleiben erhalten. Admin-User und Besucher*innen sehen, wenn sie die Website besuchen, eine Meldung, dass die gewünschte Site nicht mehr verfügbar ist. Als Spam markierte Sites werden in der Übersicht gelb hinterlegt und sind ansonsten auch für Super-Admins weiterhin sichtbar, für alle anderen nicht. Ein Klick auf Löschen entfernt die Site und zugehörige Uploads aus Ihrer Installation.

Die Ansicht zum Bearbeiten von Websites in Ihrem Netzwerk setzt sich aus mehreren Tabs zusammen. Plugins können hier eigene Tabs hinzufügen, die hier vorgestellten entsprechen aber dem, was WordPress von Haus aus mitbringt.

Der Tab Info ist bereits aktiv und wird dargestellt. Hier können Sie unter Website-Adresse (URL) die vollständige URL Ihrer Website anpassen. Abhängig davon, welchen Modus Sie bei der Einrichtung Ihrer Installation ausgewählt haben, werden Sie hier entweder eine Subdomain oder ein Unterverzeichnis Ihrer Haupt-Domain vorfinden (siehe dazu auch Abschnitt 9.2.1, »Vorbereitung«).

In diesem Feld können Sie nach Belieben Anpassungen vornehmen, die WordPress ab dem Speichern der Einstellungen zu befolgen versuchen wird. Achten Sie aber in jedem Fall darauf, dass etwaige Domains und Sub-Domains auch wirklich auf das Verzeichnis zeigen müssen, in dem Ihre WordPress-Multisite installiert ist. Das ist eine Einstellung, die Sie in der Regel bei Ihrem Hosting-Anbieter vornehmen werden.

Die beiden folgenden Felder Registriert und Zuletzt aktualisiert enthalten Zeitstempel. Diese dienen eher der Dokumentation. Sie können sie bearbeiten, werden aber keinen wirklichen Mehrwert davon haben.

Die letzten Einstellungen in diesem Tab sind eine Reihe von Attributen. In der Regel wird hier bereits ein Haken beim ersten Attribut Öffentlich gesetzt sein. Die drei folgenden Optionen Archiviert, Spam und Gelöscht entsprechen den Einstellungen, die Sie beim Überblick über die Websites-Tabelle schon kennengelernt haben. Das letzte Attribut Erwachseneninhalt hat keinen wirklichen Nutzen im täglichen Gebrauch. Ist diese Option jedoch ausgewählt, kann das Auswirkungen auf die Indexierung Ihrer Website in Suchmaschinen haben.

Im Tab Benutzer wird Ihnen eine Tabelle mit allen Usern angezeigt, die der Site zugeordnet wurden. Hier sehen Sie auch, welche Rollen die jeweiligen User auf der ausgewählten Site haben.

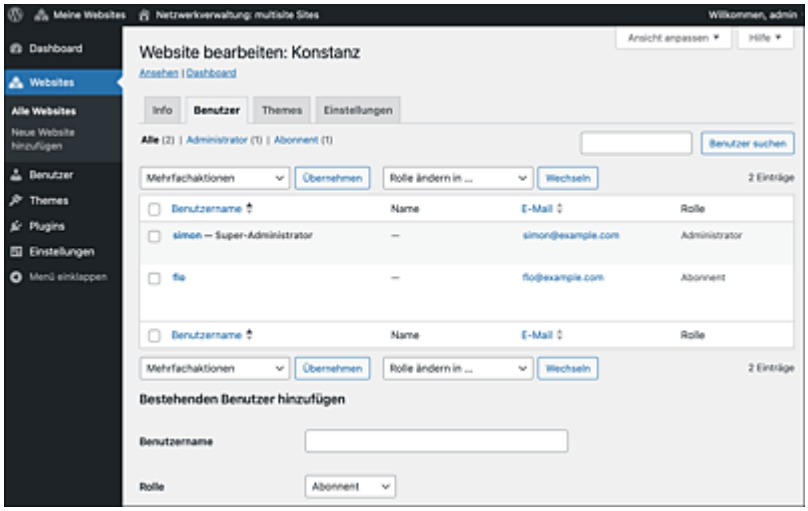


Abbildung 9.12 Der Benutzer-Tab gibt Ihnen eine Übersicht über alle Accounts mit Zugriff auf eine bestimmte Website in Ihrem Multisite-Netzwerk.

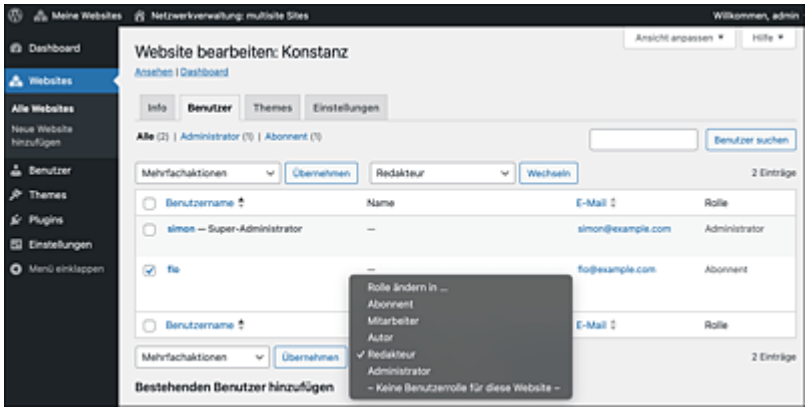


Abbildung 9.13 Direkt aus der Übersicht heraus können Sie die Rolle eines oder mehrerer Benutzer ändern.

Möchten Sie die Rolle eines Users hier ändern, so können Sie das in dieser Ansicht mit den Checkboxen in der ersten Tabellenspalte tun. Wählen Sie dort einfach einen oder mehrere User aus. Direkt unterhalb der Tabelle können Sie dann mit dem Auswahlfeld ROLLE ÄNDERN IN … eine neue Rolle auswählen und sie dem/den User(n) mit einem Klick auf den Button WECHSELN zuweisen.

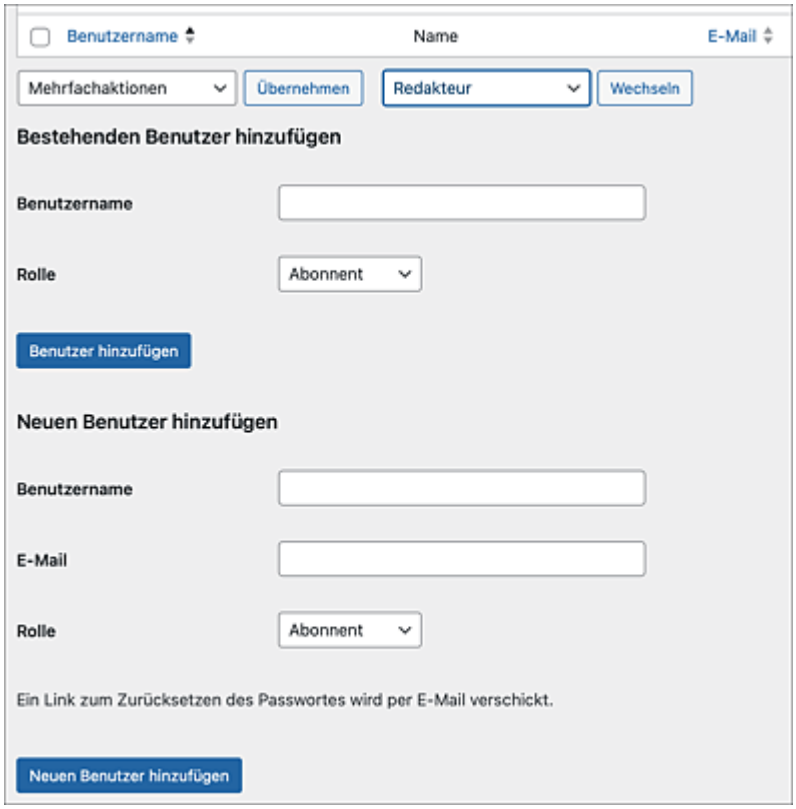


Abbildung 9.14 Unterhalb der Benutzer-Tabelle können Sie über die beiden Formulare neue User zu einer Website Ihrer Multisite hinzufügen.

Um neue User hinzuzufügen, finden Sie unterhalb der Tabelle zwei Formulare. Mit dem ersten können Sie einen Bestehenden Benutzer hinzufügen. Wenn ein User-Account, den Sie hinzufügen möchten, in Ihrem Multisite-Netzwerk schon existiert, können Sie ihn mit dem ersten Feld suchen. Die Suche läuft auch nach Eingabe der ersten Buchstaben schon und kann je nach Größe und Umfang der Multisite einige Sekunden dauern. Haben Sie den passenden User ausgewählt, können Sie im Auswahlfeld Rolle noch die Zugriffsrechte auswählen, die der User auf der Site haben soll. Ein Klick auf Benutzer hinzufügen fügt den Account zur ausgewählten Website hinzu und stattet ihn mit den entsprechenden Rechten aus.

Das zweite Formular verfolgt ein ähnliches Ziel, erlaubt Ihnen aber auch, Benutzer*innen hinzuzufügen, die bisher noch keinen Account in der Multisite haben. Im Feld Benutzername können Sie einen neuen Namen vergeben, im Feld E-Mail die E-Mail-Adresse eingeben, die dem Account zugeordnet werden soll. Analog zum ersten Formular müssen Sie auch hier noch eine Rolle auswählen, bevor Sie den User mit einem Klick auf Neuen Benutzer hinzufügen anlegen. Wie Sie vielleicht festgestellt haben, konnten Sie in diesem Prozess kein Passwort für den neuen Account vergeben. Stattdessen wird eine E-Mail zum Zurücksetzen und neuen Festlegen eines Passworts an die eben angegebene Adresse gesendet.

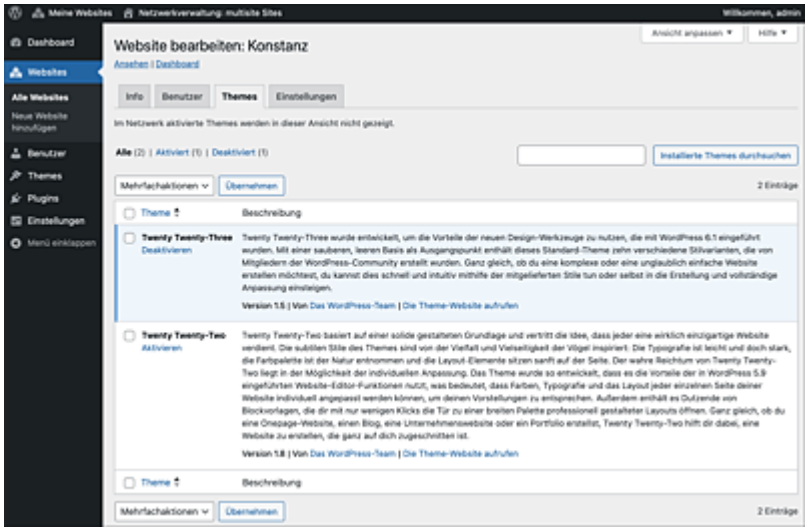


Abbildung 9.15 Im nächsten Tab können Sie auswählen, welche Themes auf dieser Website Ihres Netzwerks nutzbar sein sollen.

Sie haben die Möglichkeit, einzelne in Ihrem Netzwerk installierte Themes für die ausgewählte Website freizuschalten. In diesem Tab finden Sie eine weitere Tabelle vor. Dieses Mal sehen Sie hier eine Auflistung der verfügbaren Themes. Themes, die auf der Site genutzt werden können, sind leicht blau hinterlegt. Andere Themes können mit einem Klick auf Aktivieren zur Verfügung gestellt werden. Mit Deaktivieren kann ein Theme wieder aus dem Verkehr gezogen werden. Die Benennung dieser Optionen ist etwas irritierend. Denn durch das Aktivieren in dieser Ansicht wird das Theme nicht automatisch auch auf der Website eingesetzt. Hierzu müssen Sie auf das Dashboard der jeweiligen Website navigieren und dort ein Theme auswählen.

Themes, die Sie netzwerkweit freigeschaltet haben, tauchen in dieser Liste außerdem nicht auf (siehe dazu den nächsten Abschnitt, »Theme-Verwaltung im Netzwerk«).

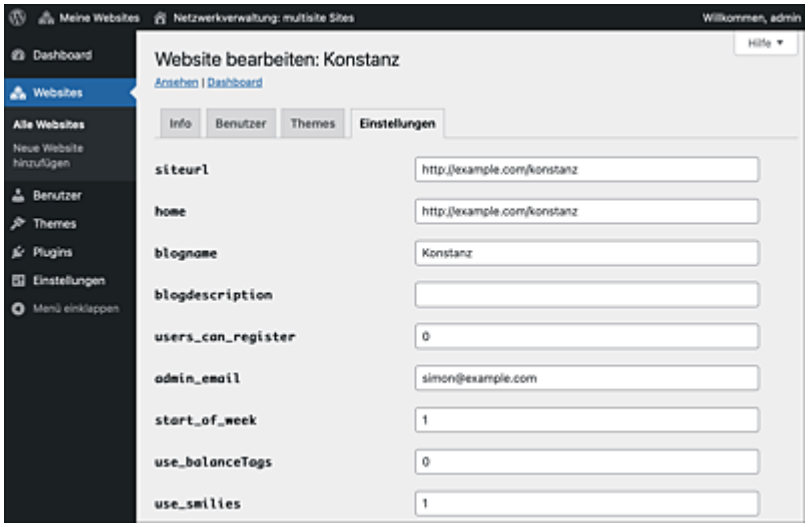


Abbildung 9.16 Im »Einstellungen«-Tab präsentiert sich Ihnen eine vollständige (und vollständig unformatierte) Liste aller Einstellungen einer Website.

Der Tab EINSTELLUNGEN enthält sämtliche Optionen, die WordPress für eine Site gespeichert hat. Die Darstellung dieser Optionen ist alles andere als anwenderfreundlich, aber zweckmäßig. Neben einigen offensichtlichen Optionen wie SITEURL oder BLOGNAME, die im Klartext gespeichert werden, gibt es einige Optionen, die als reine Zahlen oder sogar als unlesbare *serialisierte Daten* abgelegt sind. Bearbeiten Sie diese Einstellungen hier nur, wenn Sie sich sicher sind, was sie tun. Für nahezu alle diese Optionen finden Sie an anderer Stelle richtige Interfaces.

Die Benutzerverwaltung im Netzwerk

Nun verlassen wir wieder die Konfiguration einer einzelnen Website des Netzwerks und kehren zu den Punkten im Admin-Menü auf der linken Seite zurück. Die Einstellungen hier betreffen wieder das gesamte Multisite-Netzwerk.

Unter BENUTZER im Admin-Menü finden Sie eine Auflistung der User Ihres Multisite-Netzwerks. Diese Darstellung ist weitgehend identisch mit der User-Auflistung in Single-Site-Installationen. Der

einzige wirklich auffällige Unterschied zwischen der Single-Site- und der Multisite-Version ist die Spalte Websites. Hier sehen Sie, zu welchen Sites in Ihrem Netzwerk die verschiedenen User hinzugefügt wurden.

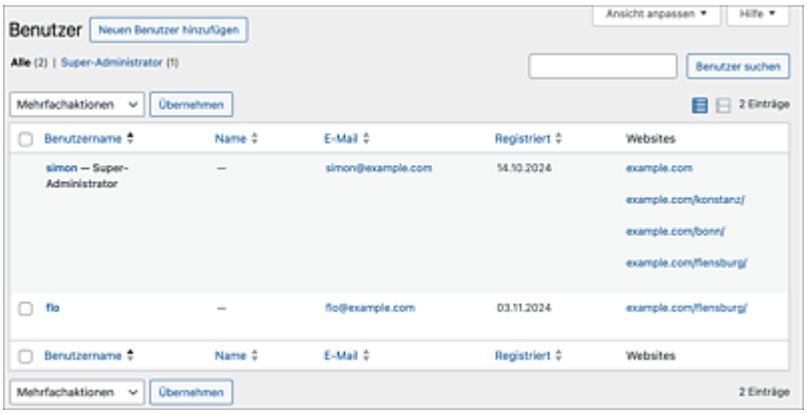


Abbildung 9.17 Auf dieser Seite erhalten Sie einen Überblick über alle Benutzer*innen auf Ihrer Website.

Vor allem bei größeren Multisites kann diese Darstellung sehr unübersichtlich werden, wenn mehrere User zu vielen Sites hinzugefügt sind. Grundsätzlich ermöglicht sie aber eine schnelle Übersicht über die Zuordnung von Usern zu Sites.

Wenn Sie es in den Netzwerk-Einstellungen erlauben, können Website-Admins selbstständig neue User zu ihren Websites hinzufügen bzw. neue User anlegen.

Theme-Verwaltung im Netzwerk

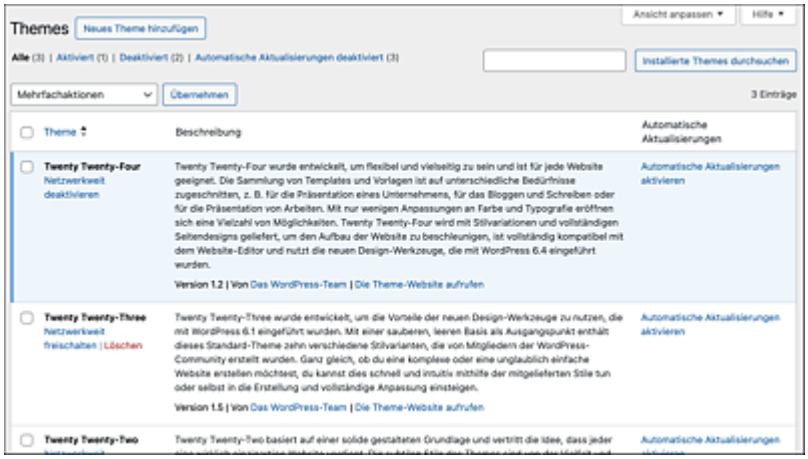


Abbildung 9.18 Als einfache Tabelle präsentieren sich Ihnen die Theme-Einstellungen Ihrer Multisite.

Im direkten Vergleich zu einfachen WordPress-Installationen fällt der Unterschied in der Theme-Verwaltung vermutlich am stärksten aus. In einer Single-Installation würden Sie hier eine Gitter-Darstellung von Screenshots der verschiedenen installierten Themes angezeigt bekommen.

In der Netzwerkverwaltung einer Multisite fällt diese Darstellung deutlich schlichter aus. Die Tabelle, die Sie in der Ansicht vor sich haben, zeigt jeweils Namen und Beschreibung für jedes installierte Theme Ihres Netzwerks an. Admin-User auf den Sites Ihrer Multisite können keine eigenen Themes installieren, sondern sind auf Ihre Vorauswahl angewiesen. Damit ein Site-Admin ein Theme zur Auswahl angezeigt bekommt, müssen Sie es entweder hier NETZWERKWEIT FREISCHALTEN oder über die Website-Einstellungen aktivieren.

Neue Themes können Sie wie in einer Single-Site über NEUES THEME HINZUFÜGEN im Admin-Menü installieren.

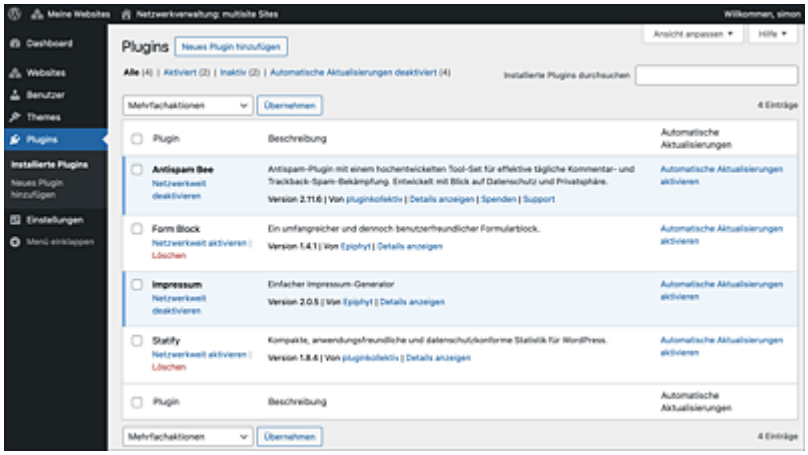


Abbildung 9.19 In der Plugin-Verwaltung erhalten Sie einen Überblick über die verschiedenen Plugins Ihrer Multisite und können weitere hinzufügen.

Ähnlich wie mit der Theme-Verwaltung verhält es sich mit Plugins. Auch auf dieser Seite bekommen Sie in einer Tabelle die installierten Plugins dargestellt. Auch hier können Sie Plugins NETZWERKWEIT AKTIVIEREN. Aber es gibt zwei entscheidende Unterschiede im direkten Vergleich mit der Theme-Verwaltung: Während *netzwerkweit aktiv* bei Themes bedeutet, dass sie für Site-Admins grundsätzlich zur Auswahl stehen, sind netzwerkweit aktive Plugins tatsächlich aktiv und werden ausgeführt. Ein SEO-Plugin, das Sie netzwerkweit aktivieren, würde ab diesem Zeitpunkt also tatsächlich auf allen Sites Ihrer Multisite laufen. Der zweite wichtige Unterschied liegt darin, dass es standardmäßig für Site-Admins gar keine Plugin-Verwaltung gibt. Über EINSTELLUNGEN, den nächsten Punkt im Admin-Menü, können Sie es Site-Admins aber ermöglichen. Diese können dann aus dem Plugin-Katalog auswählen, den Sie vorinstalliert haben.

Tipp: Plugin-Verwaltung in einer Multisite

Im Laufe der Zeit können aufwendige Setups mit vielen Plugins etwas unübersichtlich werden. Behalten Sie bei allen Plugins, die Sie installieren, daher im Hinterkopf, dass diese nicht ewig

existieren werden und eines Tages vielleicht ersetzt werden müssen.

Netzwerkweit installierte Plugins lassen sich nur für das gesamte Multisite-Netzwerk auf einmal wieder deaktivieren. Zu den netzwerkweit aktiven Plugins sollten daher eher solche gehören, die auf Netzwerkebene laufen müssen, zentrale Einstellungen für das gesamte Netzwerk haben oder als Hilfs-Plugins ohne große Konfiguration auskommen. Ein Kontaktformular-Plugin sollte zum Beispiel eher nicht netzwerkweit aktiv sein, weil Sie auf jeder Site noch mal Einstellungen setzen und vermutlich sogar Formulare in Ihre Inhalte einbinden müssen. Nach dem Austausch des bisherigen Kontaktformular-Plugins durch ein neues müssten Sie jede Site Ihres Netzwerks bearbeiten, um neue Formulare bereitzustellen.

Netzwerk-Einstellungen

Auf der letzten Seite, die standardmäßig im Menü Ihrer Netzwerkverwaltung angezeigt wird, können Sie einige grundlegende Einstellungen für Ihr gesamtes Multisite-Netzwerk setzen. Los geht es mit den Grundeinstellungen Ihres Netzwerks.

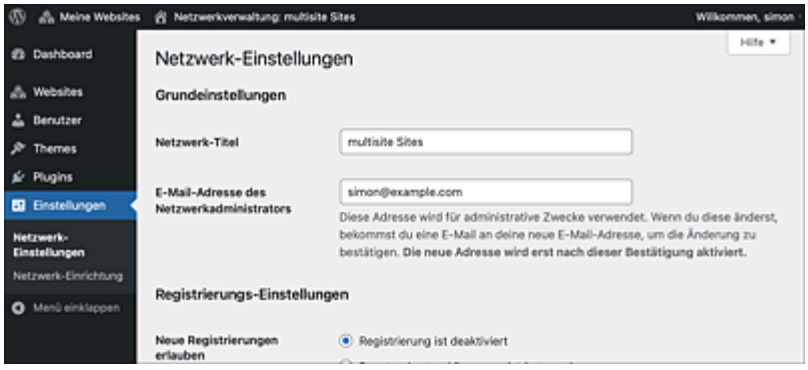


Abbildung 9.20 Der erste Abschnitt der Netzwerk-Einstellungen dreht sich um einige Grundeinstellungen.

Den Netzwerk-Titel und die E-Mail-Adresse des Netzwerkadministrators haben Sie eingangs beim Anlegen der Multisite festgelegt. Hier können Sie beide Werte anpassen.



Abbildung 9.21 Im nächsten Abschnitt der Netzwerkeinstellungen können Sie verschiedene Vorgaben zum Verhalten Ihrer Multisite rund um das Anlegen von neuen User-Accounts und Sites machen.

Standardmäßig ist das Anlegen neuer Websites und User in einer Multisite nur für Super-Admins möglich. Mit der Option Neue Registrierung erlauben können Sie dieses Verhalten anpassen. Registrierung ist deaktiviert bewirkt die bereits beschriebene Einschränkung auf Super-Admins. Benutzerkonten können registriert werden erlaubt das offene Registrieren neuer Accounts. Diese Option sollten Sie nur auswählen, wenn Sie einen wirklichen Anwendungsfall dafür haben. Die unkontrollierte Anmeldung von Usern birgt ein gewisses Sicherheitsrisiko. Angemeldete Benutzer können Websites registrieren lässt Ihnen ein wenig Kontrolle über das Anlegen neuer Sites. Nur User, die zuvor bereits zum Netzwerk hinzugefügt wurden, können so neue Websites hinzufügen. Sowohl Websites als auch Benutzerkonten können registriert werden öffnet Ihre Multisite für die Welt. So können Sie Ihre eigene kleine Blog-Farm betreiben und es anderen ermöglichen, ohne eigenes Hosting eine

Website zu betreiben. Wir sind keine Juristen, würden aber empfehlen, dass Sie sich mit eben solchen in Kontakt setzen, bevor Sie diese Option der breiten Öffentlichkeit zur Verfügung stellen.

Das Häkchen bei Registrierungs-Benachrichtigung stellt sicher, dass eine E-Mail an die Netzwerk-E-Mail-Adresse gesendet wird, wenn ein neuer User oder eine neue Website angelegt wird.

Neue Benutzer hinzufügen erlaubt es Website-Admins, selbstständig neue User anzulegen. Diese Einstellung ist unabhängig von Neue Registrierung erlauben und in vielen Fällen ein guter Mittelweg.

Verbotene Namen für Website-Adressen ist besonders dann interessant, wenn Sie anderen, in welcher Form auch immer, das Anlegen neuer Websites im Netzwerk erlauben. Durch Leerzeichen getrennt enthält dieses Feld bereits einige Begriffe, die nicht für Website-Adressen erlaubt sind. Damit soll vor allem die Kollision mit anderen gängigen Subdomains oder Verzeichnissen verhindert werden. Sie können die Liste anpassen und ergänzen. Wenn Sie zum Beispiel schon eine Subdomain *cloud. example.com* für eine Nextcloud-Installation verwenden, würden Sie `cloud` dieser Liste hinzufügen, damit niemand eine funktionslose Website anlegen kann.

Registrierung für E-Mail-Domains limitieren ist eine geschickte Möglichkeit, den Einstellungen aus Neue Registrierung erlauben etwas die Schärfe zu nehmen. Pro Zeile können Sie eine Domain angeben und damit zum Beispiel die User-Registrierung auf E-Mail-Adressen dieser Domain(s) limitieren. Mit wenig Aufwand ist es so zum Beispiel möglich, dass nur Mitarbeiter*innen mit einer Firmen-Adresse ein Blog anlegen.

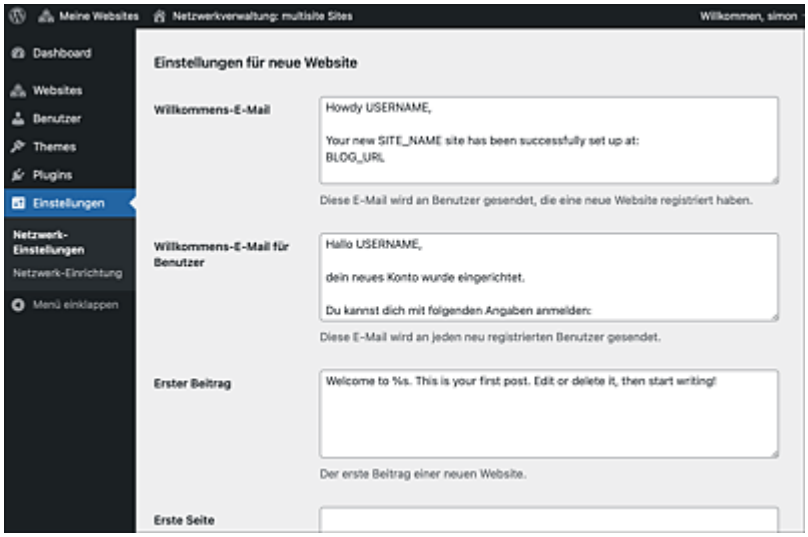


Abbildung 9.22 Der Abschnitt »Einstellungen für neue Websites« enthält allerlei Vorlagen, die beim Anlegen einer neuen Website in Ihrem Multisite-Netzwerk zum Einsatz kommen.

Umgekehrt funktioniert Geblockte E-Mail-Domains. Hier können Sie ganze E-Mail-Domains für die Registrierung blockieren.

Die beiden letzten Optionen sind nur dann wirklich relevant, wenn Sie zuvor User- oder Website-Registrierung aktiviert haben.

Dieser gesamte Abschnitt enthält anpassbare Vorlagen für Willkommens-E-Mails und Standard-Inhalte neuer Websites. Hier können Sie Ihrer Multisite eine persönliche Note geben, neue Benutzer*innen willkommen heißen und nützliche oder instruktive Standardinhalte vorgeben.

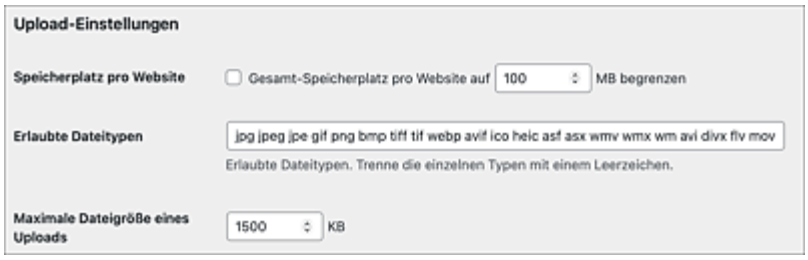


Abbildung 9.23 Die Upload-Einstellungen geben Ihnen Kontrolle darüber, wie viele und welche Dateien in Ihrer Multisite hochgeladen werden können.

Die Upload-Einstellungen sind ein wichtiges Instrument, um Ihre Multisite funktionsfähig und sicher zu halten. Mit Speicherplatz pro Website können Sie limitieren, wie viel Platz auf der Platte Ihres Servers jede einzelne Website belegen darf. Auch das ist besonders bei Multisites mit mehreren Usern sinnvoll, um zu verhindern, dass eine Site freiwillig oder unfreiwillig mit großen Dateien vollgestopft wird und den Server volllaufen lässt. Die standardmäßig vorgeschlagenen 100 MB werden in vielen Fällen ausreichen.

Erlaubte Dateitypen ist eine Einstellung, die Sie in Einzelinstallationen schmerzlich vermissen werden, wenn Sie eine Weile mit Multisites gearbeitet haben. In diesem Feld sind, durch Leerzeichen getrennt, verschiedene Dateiendungen aufgelistet, die in Ihrer Multisite von Usern hochgeladen werden können. Sie können eigene Dateiendungen wie `svg` hinzufügen, sollten dabei aber immer mögliche Sicherheitsimplikationen im Hinterkopf behalten. So erlaubt WordPress zum Beispiel bewusst keinen Upload von SVG-Dateien, weil in ihnen ausführbarer Code enthalten sein könnte, der ein Sicherheitsrisiko darstellen kann.

Die maximale Dateigrösse eines Uploads schlägt in eine ähnliche Kerbe wie der Speicherplatz pro Website. Hier können Sie verhindern, dass User zum Beispiel Bilder oder PDFs hochladen, die pro Stück eine gewisse Größe überschreiten.

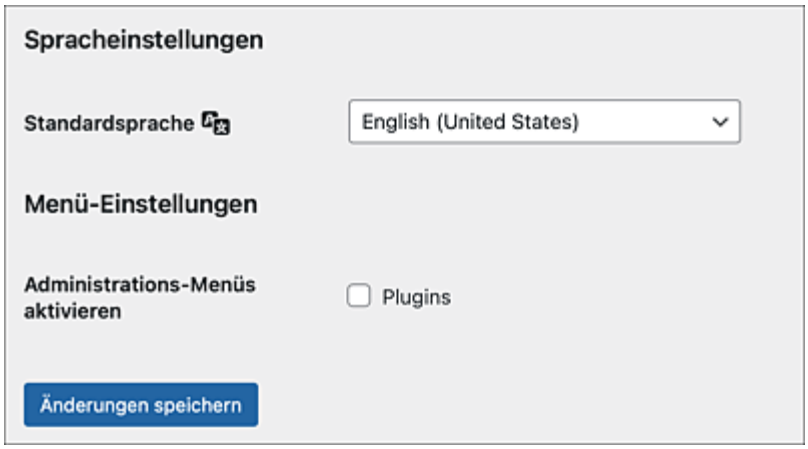


Abbildung 9.24 In den Netzwerk-Einstellungen können Sie außerdem festlegen, in welcher Sprache neue Websites in Ihrem Netzwerk angelegt werden sollen und ob Website-Admins Zugriff auf die Plugin-Verwaltung haben sollen.

Mit STANDARDSPRACHE können Sie festlegen, auf welche Sprache neue Websites in Ihrem Netzwerk voreingestellt werden sollen. Diese Einstellung kann anschließend für jede einzelne Website angepasst werden.

Ein Haken bei der Einstellung ADMINISTRATIONS-MENÜS AKTIVIEREN ermöglicht den Admins von Websites in Ihrer Multisite den Zugriff auf die Plugin-Verwaltung ihrer jeweiligen Website. Sie können ihnen damit erlauben, aus dem Katalog von vorinstallierten Plugins in Ihrem Netzwerk zu wählen und diese selbstständig zu aktivieren oder deaktivieren.

9.3.3 Upload-Verzeichnisse

Ein weiterer Unterschied, der Ihnen früher oder später auffallen könnte, findet sich in der Verzeichnisstruktur, in der WordPress etwa Bilder oder andere Medien speichert, die Sie hochladen. In einer einfachen Installation landen diese Dateien in einem Unterverzeichnis von *wp-content – /wp-content/uploads*. Nach der Aktivierung der Multisite trennt WordPress Medien-Uploads nach Sites im Netzwerk. Dadurch ergibt sich eine Verzeichnisstruktur, die zusätzlich auch noch eine ID enthält, die für Websites in der Multisite vergeben wird: *wp-content/uploads/sites/<site-id>*.

9.3.4 Datenbank-Tabellen

Wie alle WordPress-Installationen speichern auch Multisites Beitrags-, User- und Meta-Daten in der Datenbank. Zusätzlich zu den

Datenbank-Tabellen, die WordPress sowieso anlegt, ergänzt Multisite dabei einige Tabellen. Auf den täglichen Umgang mit Ihrem Multisite-Netzwerk sollte das keine Auswirkungen haben. Wenn Sie aber gelegentlich selbst mit der Datenbank hantieren, lohnt sich ein Blick in die Unterschiede.

```
wp_comments
wp_links
wp_options
wp_postmeta
wp_posts
wp_terms
wp_termmeta
wp_term_relationships
wp_term_taxonomy
wp_usermeta
wp_users
```

Zusätzlich zu diesen Standard-Tabellen fügt WordPress-Multisite einige netzwerkweite Tabellen hinzu.

```
wp_blogs
wp_blogs_versions
wp_registration_log
wp_signups
wp_site
wp_sitemeta
```

Zusätzlich dazu wird für jede Website, die in Ihrem Netzwerk angelegt wird, ein kleines Set an eigenen Tabellen angelegt, das jeweils die Site-ID als Präfix enthält.

```
wp_2_commentmeta
wp_2_comments
wp_2_links
wp_2_options
wp_2_postmeta
wp_2_posts
wp_2_terms
wp_2_termmeta
wp_2_term_relationships
wp_2_term_taxonomy
```

TEIL II

WordPress-Website anpassen und erweitern

10 Design anpassen

> *Das Design Ihrer Website ist zwar nicht das Wichtigste (das ist natürlich der Inhalt), aber das Design ist wahrscheinlich das Erste, was Besucher*innen auffällt, und ein erster guter Eindruck hat ja noch nie geschadet.*

Wenn Ihre Website gut aussieht, also das Design ansprechend ist, ist die Wahrscheinlichkeit zumindest nicht geringer, dass Personen ein bisschen länger auf Ihrer Website bleiben, als wenn das Design wenig ansprechend ist. Dabei muss aber immer sichergestellt werden, dass die Inhalte gut zu konsumieren sind, also beispielsweise Texte gut lesbar sind und nicht mit zu geringem Farbkontrast in einem zu feinen Schriftschnitt dargestellt werden, weil das (vermeintlich) gut aussieht.

In diesem Kapitel werden Sie erfahren, wo Sie *Themes* (Design für eine WordPress-Website) finden und wie Sie Themes anpassen können.

10.1 Classic Themes und Block-Themes: Was ist das?

Zum Einstieg in das Kapitel zunächst kurz ein Überblick über die zwei Arten von Themes, die es in WordPress gibt: *Classic Themes* und *Block-Themes*.

10.1.1 Classic Themes

Wie der Name verrät, sind Classic Themes die traditionellen Themes, die es bereits so lange gibt wie WordPress selbst. Diese Art Themes besteht aus PHP-Dateien, die sich um die Ausgabe der Inhalte kümmern, sowie aus CSS für das Styling und meist JavaScript für Funktionen, die der Website ein bisschen Dynamik verleihen, zum Beispiel mit einem Menü, das in der mobilen Ansicht hinter einem Hamburger-Menü verschwindet.

Diese Classic Themes lassen sich ohne Programmierkenntnisse nur anpassen, wenn sie selbst die Optionen dafür mitbringen und im Backend anzeigen, optimalerweise im *Customizer*. Für darüber hinausgehende Anpassungen reicht im besten Fall etwas CSS, das im Customizer eingefügt werden kann, andernfalls muss das Theme mit einem Child-Theme angepasst werden (wie das geht, lesen Sie in Abschnitt 23.2.1).

10.1.2 Block-Themes

Seit WordPress 5.9 werden im WordPress-Core sogenannte Block-Themes unterstützt. Bei diesen Themes wird die gesamte Website aus den Blöcken zusammengesetzt, die auch im Block-Editor nutzbar sind. Das heißt, es besteht nicht mehr nur der Inhalt von Beiträgen und Seiten aus Blöcken, sondern auch die Theme-Dateien enthalten Markup von Blöcken, um neben dem Inhalt noch Elemente wie den *Header* und *Footer* auszugeben.

Diese Themes lassen sich, da sie ja komplett aus Editor-Blöcken bestehen, mit dem ebenfalls in WordPress 5.9 eingeführten *Website-Editor* bearbeiten. So ist es nicht mehr nur möglich, einzelne vom Theme mitgebrachte Einstellungen zu verändern, wie zum Beispiel die Linkfarbe, sondern die gesamte Website umzubauen. Wenn

Ihnen der Header von einem Block-Theme also nicht gefällt, weil Sie die Navigation lieber unter dem Website-Titel hätten als daneben, können Sie das im Website-Editor anpassen, ohne ein Child-Theme erstellen zu müssen.

Passend dazu sind alle Standard-Themes seit dem mit WordPress 5.9 mitgelieferten Twenty Twenty-Two Block-Themes, die sich gut zum Ausprobieren dieser neuen Theme-Art eignen.

10.2 Installierte Themes vorab ansehen und aktivieren

Ein WordPress-Theme kümmert sich um die Ausgabe der von Ihnen erstellten Inhalte und kann im besten Fall ohne größere Probleme gegen ein anderes ausgetauscht werden, wenn Sie ein Theme gefunden haben, das Ihnen besser gefällt.

10.2.1 Trennung von Design und Inhalt

Worauf Sie bei der Auswahl eines Themes achten sollten, ist, dass es sich wirklich nur um die Darstellung der Inhalte kümmert, und keine eigenen Funktionen zur Inhaltserstellung mitbringt. Um Funktionen zur Inhaltserstellung sollten sich nur der WordPress-Core und Plugins kümmern.

Der Grund, warum die Trennung von Design und Inhalt wichtig ist: Es ist nicht unwahrscheinlich, dass Sie jetzt ein Theme auswählen, das Sie vielleicht schon in ein paar Monaten oder erst in ein paar Jahren wieder wechseln möchten, um das Design Ihrer Website zu ändern. Wenn Sie sich jetzt zum Start ein Theme ausgesucht haben, das beispielsweise die Möglichkeit mitbringt, ein Portfolio anzulegen, würden Sie diese Inhalte bei einem Wechsel verlieren (sogenannte »Premium«-Themes bringen häufig viele Funktionen mit, die eigentlich in Plugins gehören). Lassen Sie deshalb die Finger von Themes, die Funktionen mitbringen, mit denen Sie Inhalte erstellen können.

10.2.2 Standard-Themes

WordPress veröffentlicht einmal im Jahr mit einem WordPress-Release ein neues Theme, das bis zur Veröffentlichung des nächsten Themes das Standard-Theme neuer WordPress-Installationen ist. Diese Themes werden nach Jahreszahlen benannt, das neueste Theme ist aktuell Twenty Twenty-Five. Neben dem neuesten Theme werden auch ein paar der Themes aus den Vorjahren mitinstalliert.

Diese Standard-Themes nutzen in der Regel die neuesten Möglichkeiten, die WordPress für Themes bietet, weshalb sich eigentlich immer ein Blick auf das aktuelle Theme lohnt, um zu schauen, was gerade so alles möglich ist.

Die Themes finden Sie im Admin-Menü links unter dem Punkt Design • Themes (ein Klick auf Design bringt Sie auch auf die Seite), an erster Stelle steht immer das aktive Theme (siehe Abbildung 10.1).

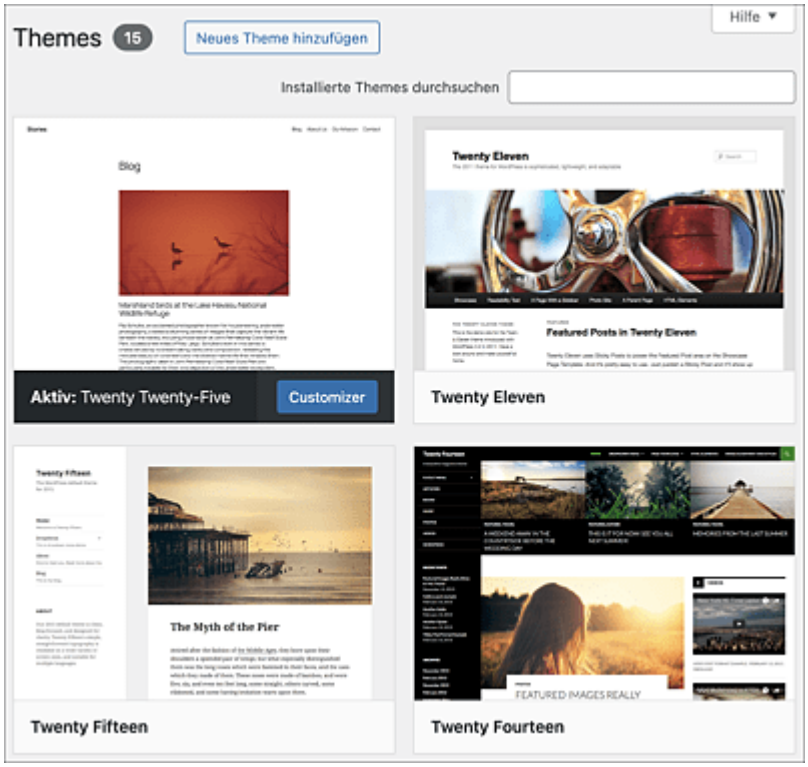


Abbildung 10.1 Eine frische WordPress-Installation kommt mit mehreren Standard-Themes von WordPress, von denen jedes Jahr ein neues veröffentlicht wird.

Wenn Sie den Cursor über eins der nicht aktiven Themes bewegen, erscheint ein Live-Vorschau-Button, mit dem Sie sich das Theme im Website-Editor (Block-Themes) beziehungsweise Customizer (Classic Themes) anschauen es und auch direkt verändern können. Wenn Sie mit dem Theme und Ihren Anpassungen zufrieden sind, können Sie es mit den Anpassungen aktivieren.

10.3 Block-Theme anpassen: den Website-Editor kennenlernen

Schauen wir uns zunächst die Anpassungsmöglichkeiten eines Block-Themes an. Dafür nutzen wir das WordPress-Standard-Theme Twenty Twenty-Five. Alle Anpassungen spielen sich bei Block-Themes im Website-Editor ab, den Sie über das linke Admin-Menü unter Design • Website-Editor erreichen können. Wenn Sie den Website-Editor öffnen, sollte die Ansicht mit dem Theme Twenty Twenty-Five ungefähr aussehen wie in Abbildung 10.2.

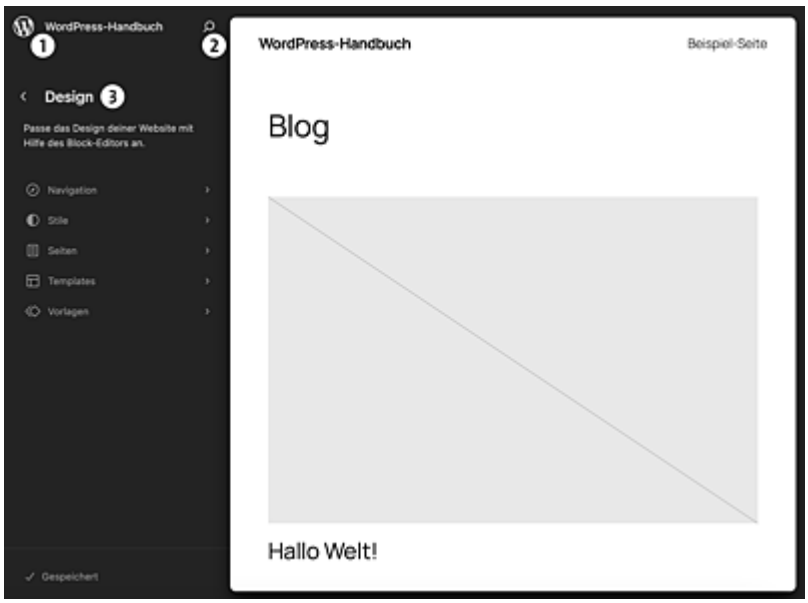


Abbildung 10.2 Der Website-Editor erlaubt die umfangreiche Bearbeitung von Websites mit Block-Themes direkt im Backend.

Der Website-Editor ist recht komplex und bietet viele Möglichkeiten und kann vor allem am Anfang etwas überwältigend wirken. Nachfolgend werden alle Bereiche besprochen, sodass Sie sich dann gut zurechtfinden sollten.

10.3.1 Die Sidebar links

Auf der linken Seite des Website-Editors sehen Sie eine Seitenleiste mit schwarzem Hintergrund. Diese Seitenleiste ist nicht dauerhaft da, sondern wird ausgeblendet, wenn Sie sich im Bearbeitungsmodus der Website befinden (siehe Abbildung 10.9). Wenn Sie in der Bearbeitungsansicht sind, können Sie die Seitenleiste über einen Klick oben links auf das Logo wieder einblenden.

Die Seitenleiste besteht aus den folgenden Elementen:

- Ganz oben links ❶ sehen Sie das WordPress-Logo (oder die Grafik, die Sie als Website-Icon unter Einstellungen • Allgemein festgelegt haben). Ein Klick auf das Icon bringt Sie ins Dashboard des Backends.

  Neben dem Icon ist der Titel der Website dargestellt, der auf die Startseite im Frontend verlinkt ist.

- In der rechten oberen Ecke der Seitenleiste ❷ wird eine Lupe angezeigt, die bei einem Klick die Befehlspalette öffnet, über die Sie durch die Suche Shortcuts zu vielen Aktionen im Editor erreichen können: vom Anlegen eines neuen Beitrags oder Öffnen einer bestehenden Seite bis zum Öffnen von Bereichen der Website-Editor-Seitenleiste.

- Etwas weiter unten in der Seitenleiste befindet sich ein mit Design überschriebener Bereich ❸ (ein Klick auf die Überschrift bringt Sie zum Dashboard). In dem Bereich gibt es fünf Buttons, mit denen Sie in verschachtelte Bereiche der Seitenleiste gelangen (wenn Sie sich in einem der Unterbereiche befinden, gelangen Sie über einen Klick auf den Pfeil nach links neben dem Titel des Bereichs wieder eine Ebene nach oben):

- Navigation bringt Sie zu einer Liste der angelegten Navigationsmenüs. Wenn Sie ein Menü anklicken, erscheint es rechts im Bearbeitungsbereich, sodass Sie es dort bearbeiten können (siehe Abbildung 10.3).
- Hinter Stile werden Ihnen alternative Stilkombinationen angezeigt, die das aktive Theme mitbringt. Diese Kombinationen erlauben es Ihnen, mit einem Klick das Aussehen Ihrer Website in eine stimmige andere Version zu verändern (siehe Abbildung 10.4). Vielleicht ändern sich nur Farben, vielleicht aber auch Schriften und die Breite des Inhalts – das hängt von Ihrem Theme ab.
- Über das Auge-Icon rechts von der Stile-Überschrift können Sie das sogenannte Stilbuch öffnen, das Ihnen alle Blöcke im aktuellen Design anzeigt, sodass Sie sich einen guten Überblick verschaffen können, ob alles passt. Über das Stift-Icon neben dem Auge können Sie in die Bearbeitungsansicht springen, um den Stil anzupassen. Auf diese Möglichkeit wird später genauer eingegangen.

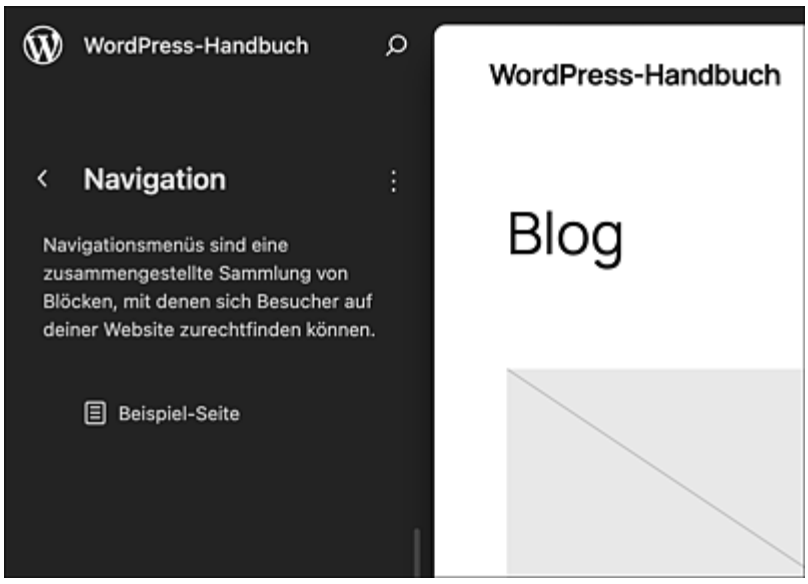


Abbildung 10.3 Über den »Navigation«-Menüpunkt in der Seitenleiste gelangen Sie zu einer Liste Ihrer Navigationsmenüs und können die Menüs bearbeiten.

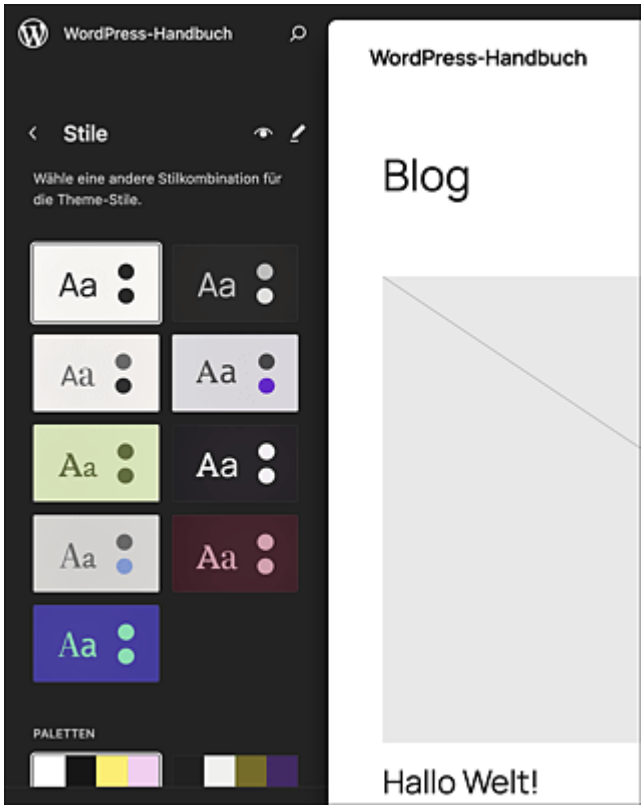


Abbildung 10.4 Themes können unterschiedliche »Stile« mitbringen, die Variationen des normalen Designs sind und von Ihnen einfach aktiviert werden können.

- Der Punkt SEITEN öffnet eine Liste der von Ihnen angelegten Seiten (siehe Abbildung 10.5). Die Funktionen doppeln sich mit denen der Ansicht, die Sie über den Punkt SEITEN im Admin-Menü erreichen, aber das wird vermutlich ein erster Blick darauf sein, wo es mit dem gesamten Design des Backends früher oder später hingeht. Direkt links in der Seitenleiste finden Sie die unterschiedlichen Status der Seiten, und durch einen Klick auf beispielsweise VERÖFFENTLICHT zeigen Sie nur die öffentlichen Seiten an.

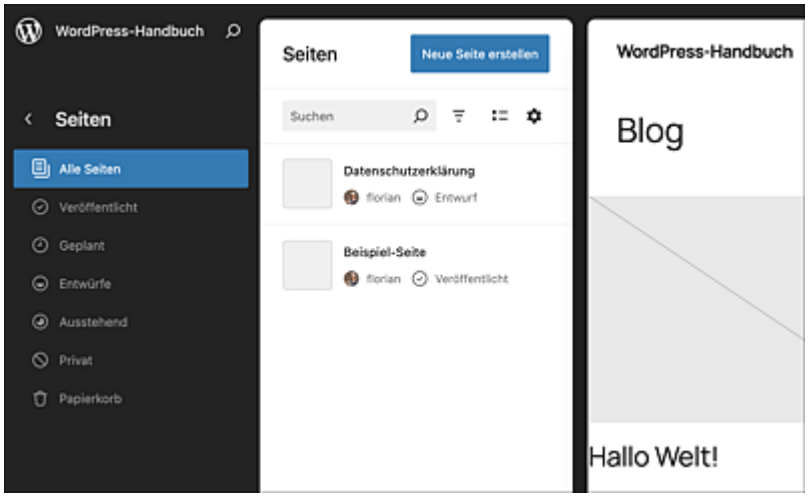


Abbildung 10.5 Im Website-Editor finden Sie auch eine Liste der Seiten Ihrer Installation, die Ihnen von den Funktionen bekannt vorkommen könnten.

- Über der Liste sehen Sie den Button Neue Seite erstellen, der ein Modal für den Titel der Seite öffnet und im Anschluss nach einem Klick auf Einen Entwurf erstellen diesen Entwurf im Website-Editor öffnet, sodass Sie ihn bearbeiten können.
- Über das Icon links neben dem Zahnrad unter dem Button zum Anlegen einer neuen Seite können Sie das Layout der Liste anpassen (die beiden zusätzlichen Optionen Raster und Tabelle zeigen die Seiten in der vollen Breite an, die die Seitenleiste übrig lässt). Das Icon spiegelt immer das aktuelle eingestellte Layout wieder.
- Mit dem Zahnrad-Icon können Sie die Sortierung ändern und einstellen, welche Eigenschaften angezeigt werden sollen. Außerdem können Sie die Elemente pro Seite einstellen.
- Mit der Suche können Sie die Elemente der Liste durchsuchen und über den Button Filter hinzufügen rechts neben der Suche nach Autor und Status filtern. Diese Filter können Sie auch kombinieren.

- Sie sehen schon, aktuell ist die Liste der Funktionen noch nicht gleichauf mit der klassischen Übersicht, die Sie direkt über das Admin-Menü erreichen. Wenn Sie sich aber sowieso gerade schon im Website-Editor befinden und eine Seite erstellen möchten, könnte es schon eine nützliche Abkürzung sein.
- Nach einem Klick auf Templates öffnet sich in dem rechten Bereich des Website-Editors eine Rasteransicht mit allen Templates, sowohl den Theme-eigenen als auch eventuell von Ihnen erstellten (siehe Abbildung 10.6). Jedes Template wird mit einer Vorschau angezeigt, unter der Vorschau steht der Titel des Templates, zum Beispiel Alle Archive für das Template für alle Archiv-Übersichten. Unter dem Titel und einer gegebenenfalls vorhandenen Beschreibung des Templates sehen Sie, von wem das Template erstellt wurde: Ist es ein Template vom Theme oder eins, das im Website-Editor von einer Benutzer*in erstellt wurde. Links in der Seitenleiste können Sie die Templates nach der Urheber*in filtern. Nach einem Klick auf ein Template gelangen Sie in die Bearbeitungsansicht des entsprechenden Templates.

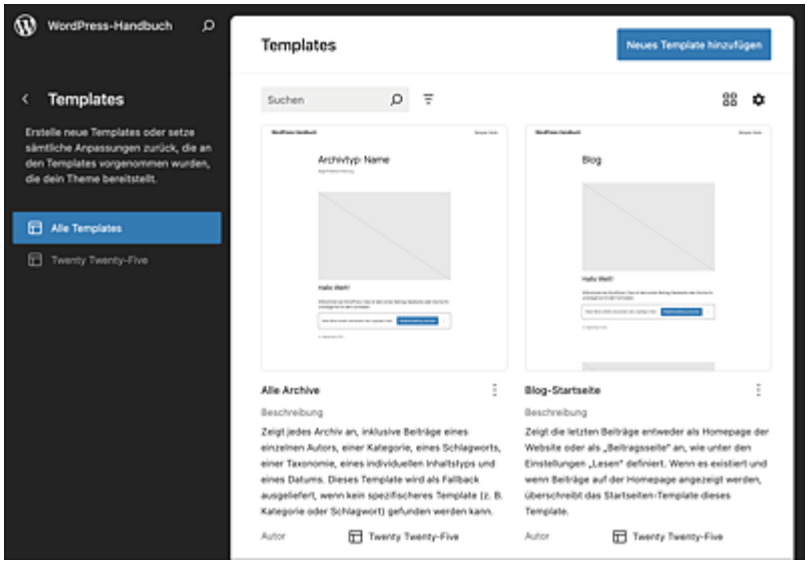


Abbildung 10.6 Auf der rechten Seite wird eine Liste aller Templates angezeigt.

- Wenn Sie ein Template bearbeitet haben, können Sie es wieder auf den Ursprungszustand zurücksetzen, indem Sie den Cursor über dem Template platzieren und die Checkbox oben links in der Ecke der Template-Vorschau anhaken. Anschließend sehen Sie unten neben der Anzahl der ausgewählten Einträge ein Icon zum Zurücksetzen der ausgewählten Templates (siehe Abbildung 10.7).

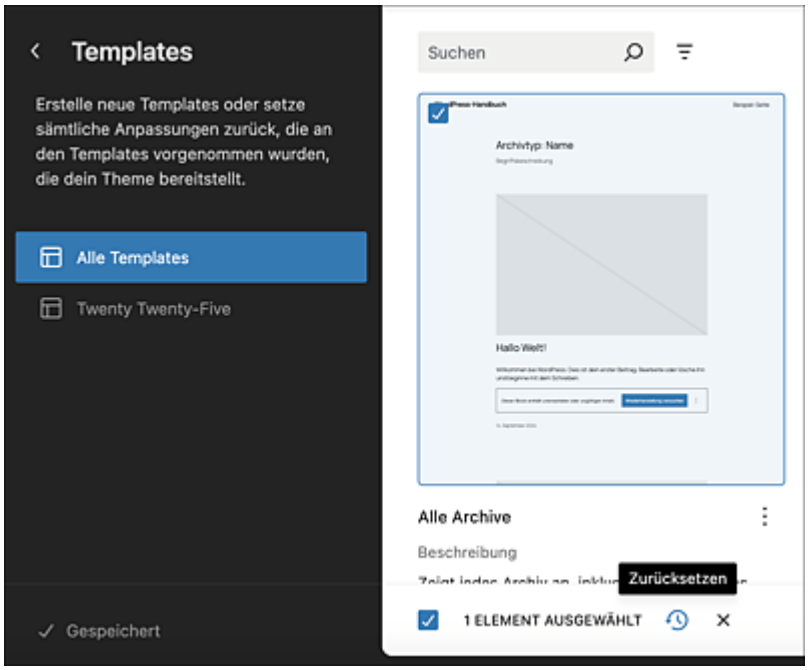


Abbildung 10.7 Veränderte Templates können wieder in den Ursprungszustand zurückgesetzt werden.

- Wenn Sie auf Vorlagen klicken, sehen Sie rechts eine ähnliche Rasteransicht wie bei den Templates (siehe Abbildung 10.8). Angezeigt werden hier aber die Block-Vorlagen und Template-Teile – sowohl die Theme-eigenen als auch die von Ihnen oder anderen Benutzer*innen direkt über den Website-Editor angelegten. Die linke Seitenleiste ist unterteilt in einen Bereich für die Template-Teile und einen für Vorlagen. Template-Teile sind in der Regel Header und Footer mit gegebenenfalls

vorhandenen Varianten. Diese Teile werden in alle oder die meisten Templates des Themes eingebunden.

- Block-Vorlagen sind Vorlagen aus einem oder mehreren Blöcken, die vorkonfiguriert sind und sich im Block-Editor einbauen lassen. Das ist besonders praktisch bei Elementen, die sich häufiger wiederholen und deren initiale Einrichtung ein bisschen dauert oder komplexer ist. Wie bei den Templates bringt Sie ein Klick auf eins der Elemente in der rechten Rasteransicht in die Bearbeitungsansicht der Vorlage oder des Template-Teils.

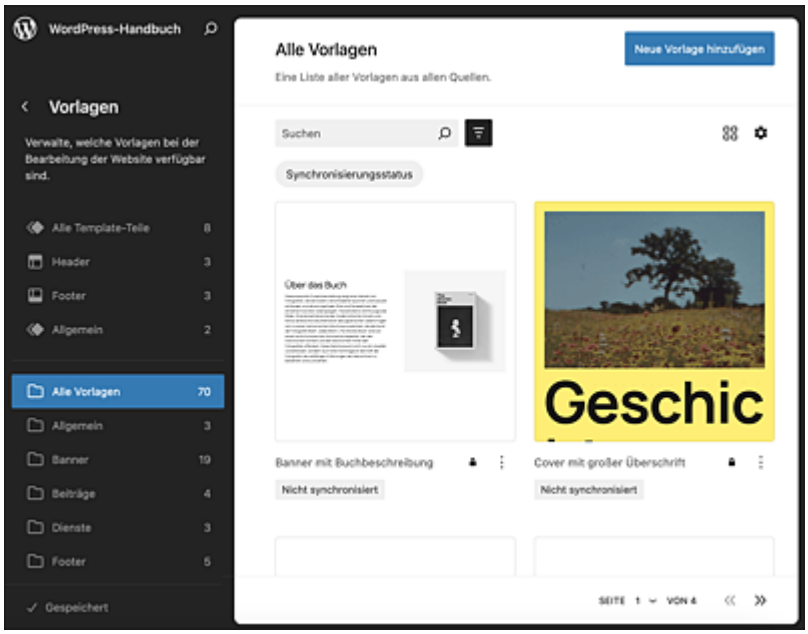


Abbildung 10.8 Unter dem Punkt »Vorlagen« finden Sie sowohl die Template-Teile, etwa den Header und Footer, als auch die Block-Vorlagen, die Block-Kombinationen zum einfachen Einfügen in den Editor bereitstellen.

10.3.2 Die Vorschau und Bearbeitungsansicht rechts

Wenn Sie irgendwo rechts auf die Vorschau klicken, öffnet sich die Bearbeitungsansicht, in der Sie das aktuelle Template bearbeiten können (siehe Abbildung 10.9).

Schauen wir uns auch diese Ansicht und ihre einzelnen Bestandteile in Ruhe an.

Die Toolbar

Oben sehen Sie eine Toolbar, wie sie auch im normalen Block-Editor für Ihre Inhalte angezeigt wird. Mittig wird Ihnen darin die Art des Templates angezeigt, das Sie gerade bearbeiten, in Abbildung 10.9 das Template Alle Archive. Wenn Sie auf den Namen des Templates klicken, öffnet sich die Befehlspalette, die Sie vorhin schon in der Seitenleiste gesehen haben.

Links vom Template-Titel sehen Sie Buttons, die Ihnen aus dem normalen Block-Editor bekannt vorkommen sollten: zum Einfügen neuer Blöcke, zum Umschalten des Interaktionsmodus, zur Anzeige der Blöcke-Hierarchie und zum Rückgängigmachen und Wiederholen vorheriger Schritte. Das WordPress-Logo/Site-Icon ganz links blendet wieder die Seitenleiste ein und wechselt rechts zurück in die Vorschau. Dabei bleiben eventuell vorhandene ungespeicherte Anpassungen erhalten.

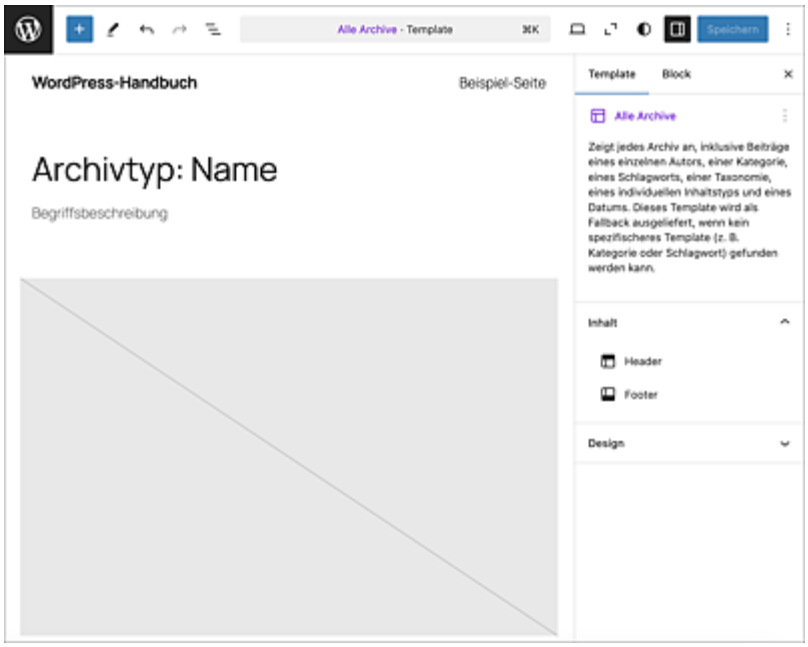


Abbildung 10.9 Diese Ansicht kommt Ihnen wahrscheinlich aus der Bearbeitungsansicht von Seiten und Beiträgen bekannt vor.

Rechts vom Template-Titel befinden sich ebenfalls aus dem Editor zur Inhaltsbearbeitung bekannte Buttons: Zur Anzeige einer Vorschau, zum Speichern der Änderungen, zum Ein- und Ausblenden der Sidebar mit den Einstellungen (statt des Beitrag- oder Seite-Tabs gibt es hier neben dem Block-Tab den Template-Tab), und ganz rechts sehen Sie das Drei-Punkte-Menü, über das Sie unter anderem in den Code-Editor wechseln können.

Neu ist das runde Icon links neben dem Drei-Punkte-Icon: ein Kreis, der halb weiß und halb schwarz gefüllt ist. Ein Klick auf das Icon öffnet die Stile-Sidebar, wie in Abbildung 10.11 zu sehen. Diese Seitenleiste ist dieselbe wie die, die Sie in der linken Seitenleiste unter dem Punkt Stile über das Stift-Icon erreichen konnten (siehe Abbildung 10.4).

Die »Stile«-Seitenleiste

Die Stile-Seitenleiste ermöglicht Ihnen die Anpassung vieler Design-Eigenschaften, die von Ihrem Theme vordefiniert wurden.

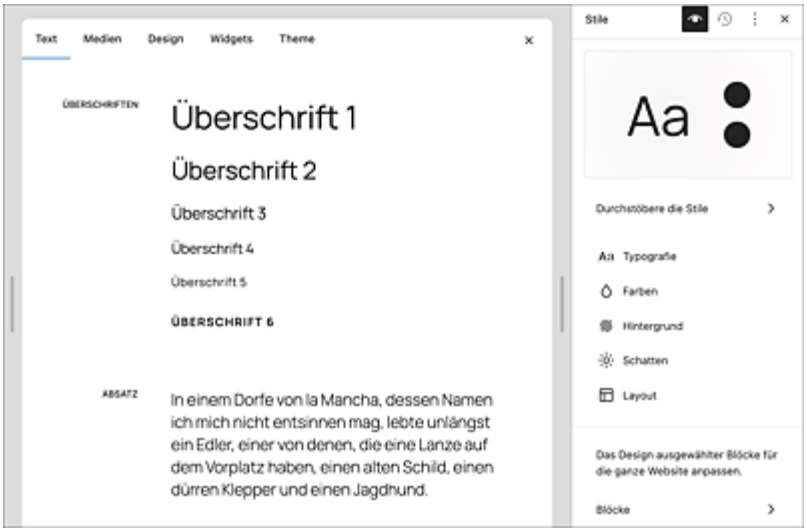


Abbildung 10.10 Das Stilbuch zeigt Ihnen alle Blöcke im aktuellen Design an.

Ganz oben haben Sie die Möglichkeit, über das Auge-Symbol zum Stilbuch zu wechseln ❶, das vorhin schon kurz angerissen wurde. Wie das aussieht, sehen Sie in Abbildung 10.10.

Rechts neben dem Auge sehen Sie einen Button mit einem Uhr-Icon, das von einem Pfeil umgeben ist ❷. Mit einem Klick auf das Icon werden Ihnen die Revisionen der Änderungen im Website-Editor angezeigt, falls Sie bereits Anpassungen vorgenommen haben. Über den Zeitstrahl, in dem bei jedem Punkt die gemachten Änderungen beschrieben werden, können Sie zu vorherigen Speicherständen springen, während sich links die Vorschau der Website entsprechend verändert. Wenn Sie zu einem anderen Stand zurückkehren möchten, klicken Sie auf Übernehmen (siehe Abbildung 10.12). Über das Drei-Punkte-Menü neben dem Revisionen-Button können Sie den Willkommensdialog noch einmal starten und ein Feld für zusätzliches CSS öffnen, wie es bei Classic Themes im Customizer zu finden ist.

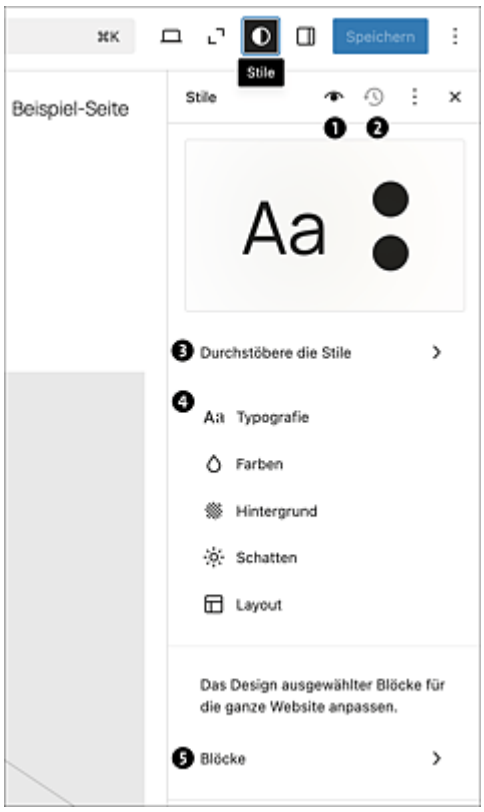


Abbildung 10.11 Die »Stile«-Seitenleiste erlaubt Ihnen die Anpassung des Designs vom Theme, wie beispielsweise Farben oder Schriften.

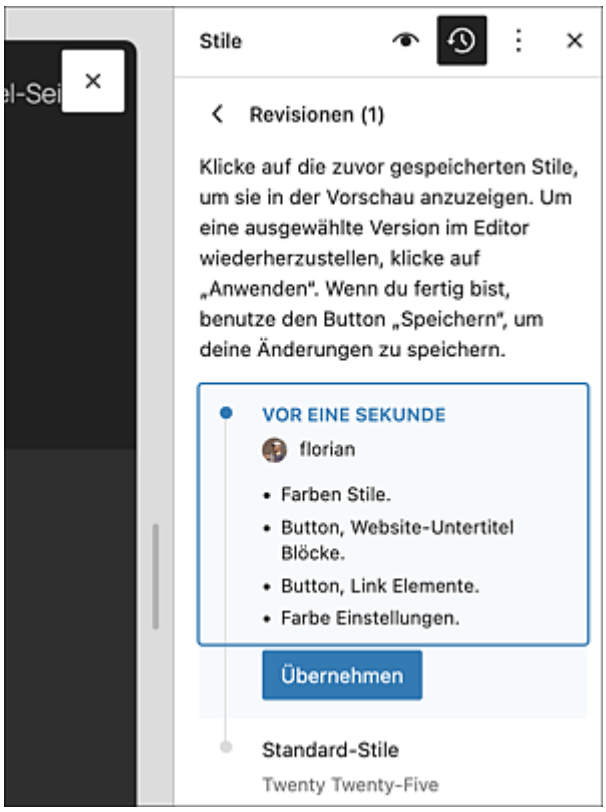


Abbildung 10.12 Nicht nur Beiträge und Seiten haben Revisionen, sondern auch die Anpassungen im Website-Editor.

Der Button Durchstöbere die Stile ❸ bringt Sie in die Übersicht der vom Theme vorkonfigurierten alternativen Stile, die Sie im vorherigen Abschnitt auch in der linken Seitenleiste gesehen haben.

Unter dem Button für die Stile finden Sie die Punkte Typografie, Farben, Hintergrund, Schatten und Layout ❹. Hinter diesen Buttons befinden sich Einstellungen, die sich global auf alle passenden Elemente auswirken, etwa die Farbe der Überschriften. Anpassungen für bestimmte Blöcke können über den Punkt Blöcke vorgenommen werden, der sich im nächsten Abschnitt der Stile-Seitenleiste befindet ❺.



Abbildung 10.13 Im Bereich »Typografie« der »Stile«-Seitenleiste können Sie die verfügbaren Schriften verwalten und Typografie-Optionen für bestimmte Elemente anpassen.

Ein Klick auf Typografie bietet Ihnen die Möglichkeit, die im Theme verfügbaren Schriften zu verändern, die beispielsweise beim Absatz- oder Überschriften-Block ausgewählt werden können, sowie die Typografie-Einstellungen für bestimmte »Elemente« anzupassen (siehe Abbildung 10.13).

Wenn Sie über der Liste der Schriften auf das Regler-Icon klicken ❻, öffnet sich ein Overlay mit den drei Tabs Bibliothek, Hochladen und Schriften installieren (siehe Abbildung 10.14). Im Bibliothek-Tab sehen Sie die installierten Schriften und wie viele Varianten aktiv sind. Wenn Sie auf eine Schrift klicken, können Sie Varianten deaktivieren oder aktivieren. Unter Hochladen können Sie Schriftdateien hochladen, und über Schriften installieren haben Sie die Möglichkeit, Schriften von Google Fonts zu installieren.

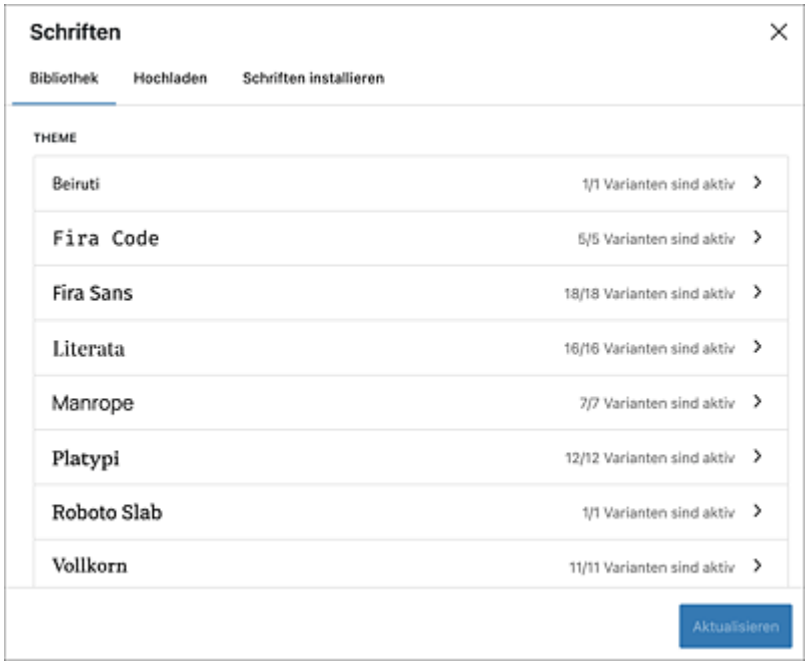


Abbildung 10.14 In der Schriften-Bibliothek sehen Sie, welche Schriften mit wie vielen Varianten installiert sind, und können Varianten deaktivieren oder aktivieren. Außerdem haben Sie über die Tabs »Hochladen« und »Schriften installieren« die Möglichkeit, zusätzliche Schriften hinzuzufügen.

Wenn Sie im Typografie-Bereich der Stile-Seitenleiste direkt auf eine der Schriften klicken, öffnet sich ebenfalls das Schriften-Overlay, und Sie landen in der Bibliothek direkt bei der entsprechenden Schrift.

Unter Elemente ❼ sehen Sie die Punkte Text, Links, Überschriften, Beschriftungen und Buttons. Hinter jedem der Punkte finden Sie Optionen zur Anpassung der Typografie-Einstellungen, in Abbildung 10.15 ist beispielhaft die Einstellungsseite für Überschriften zu sehen. Sie können unter anderem die Schrift und Zeilenhöhe anpassen sowie für die einzelnen Überschriften-Ebenen die Größe anpassen.



Abbildung 10.15 In den Typografie-Einstellungen können Sie für bestimmte Elemente wie beispielsweise Überschriften Dinge wie die Schriftart, Schriftgröße und Zeilenhöhe anpassen.

Zurück in die oberste Ebene der Stile-Sidebar: Im Bereich Farben ❹ können Sie die Palette an Farben bearbeiten sowie Farbeinstellungen für Elemente vornehmen, wie Sie es eben schon für die Typografie-Optionen kennengelernt haben (siehe Abbildung 10.16). Ein Klick auf Palette bringt Sie in eine Ansicht, die in die Tabs Farbe und Verlauf unterteilt ist (siehe Abbildung 10.17). Im Farbe-Tab können Sie die Theme-Farben anpassen und optional aus unterschiedlichen vorgefertigten Paletten wählen. Über das Drei-Punkte-Menü bei Theme haben Sie immer die Möglichkeit, die Farben zurücksetzen. Im Verlauf-Tab haben Sie ähnliche Möglichkeiten für die Farbverläufe und Duoton-Effekte.

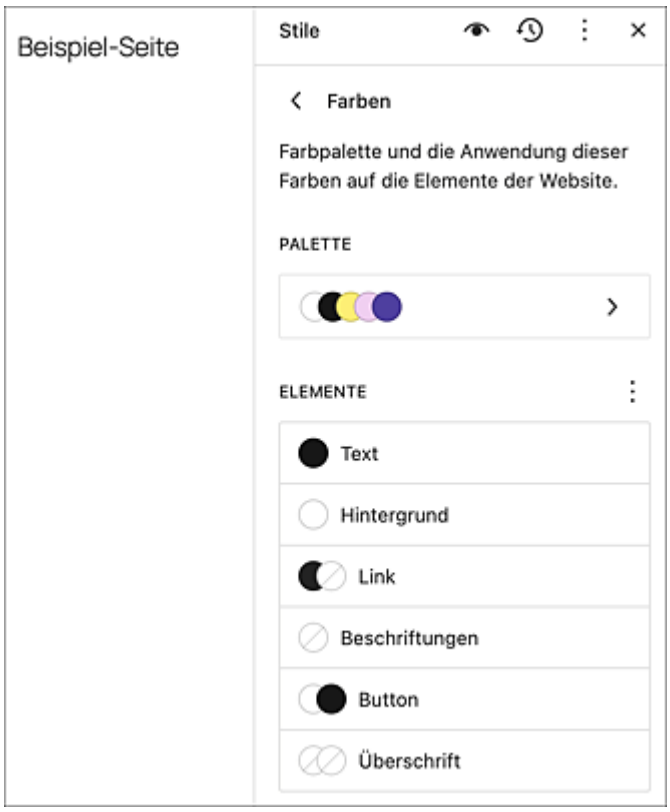


Abbildung 10.16 Im »Farben«-Bereich können Sie die Farbpalette des Themes anpassen sowie Farbeinstellungen für verschiedene Elemente global setzen.

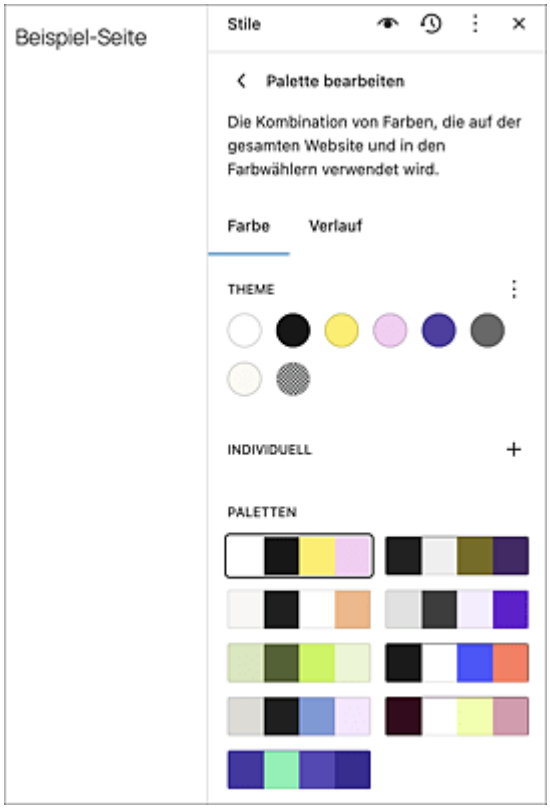


Abbildung 10.17 Im »Farbe«-Tab können Sie die Farbpaletten des Themes anpassen, im »Verlauf«-Tab können Sie die Verläufe und Duoton-Effekte verändern.

Im Bereich HINTERGRUND ❹ können Sie ein Hintergrundbild für die Website festlegen (siehe Abbildung 10.18). Wenn Sie ein Hintergrundbild festgelegt haben, bekommen Sie bei einem Klick auf das Bild alle möglichen Optionen, um die Darstellung und Position anzupassen.

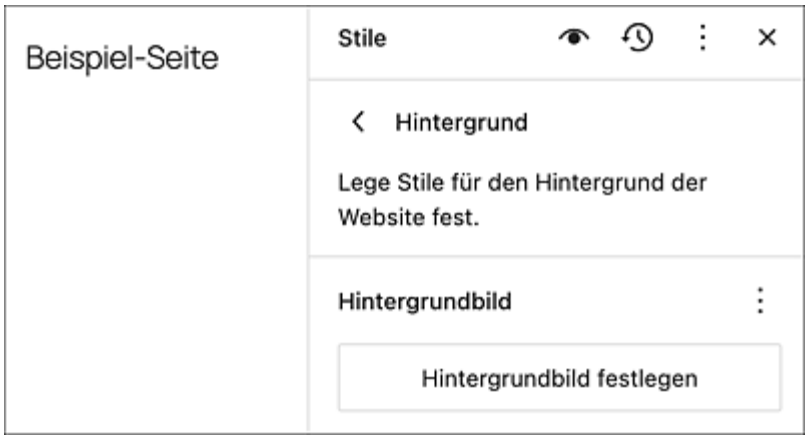


Abbildung 10.18 Unter dem Eintrag »Hintergrund« haben Sie die Möglichkeit, ein Hintergrundbild für die Website festzulegen.

Im Schatten-Bereich ❹ können Sie die Schatten-Effekte anpassen, die bei einigen Blöcken eingesetzt werden können (siehe Abbildung 10.19). Es gibt einige Standard-Schatten, die Sie durch einen Klick individualisieren können (siehe Abbildung 10.20). Unter den vorgefertigten Schatten können Sie auch individuelle Schatten erstellen.

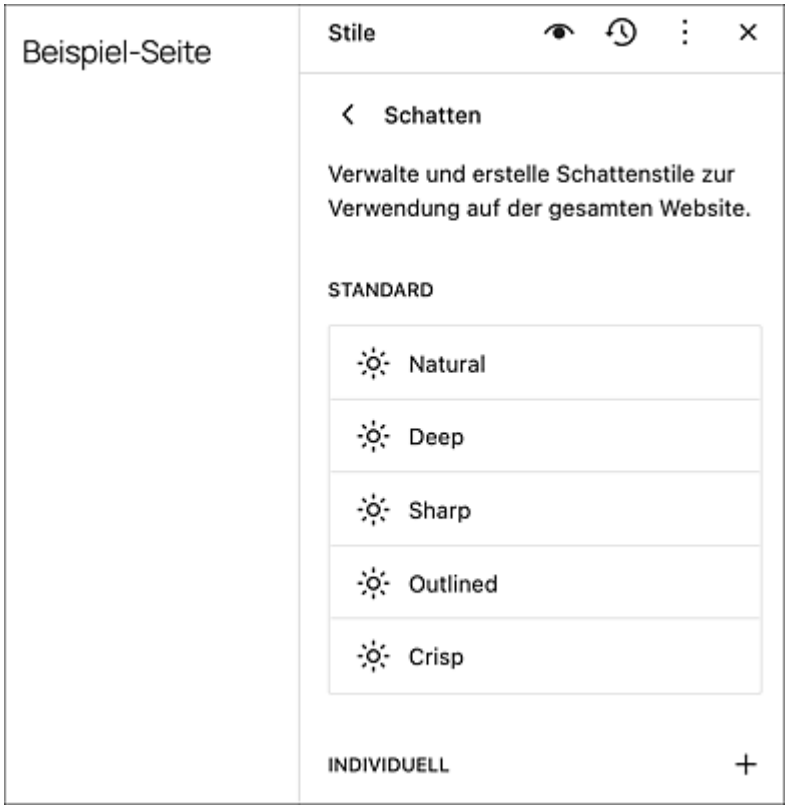


Abbildung 10.19 Blöcken wie dem Bild-Block können Sie einen Schatten-Effekt zuweisen. In der »Stile«-Seitenleiste können Sie die vorhandenen Schatten bearbeiten und individuelle Schatten hinzufügen.

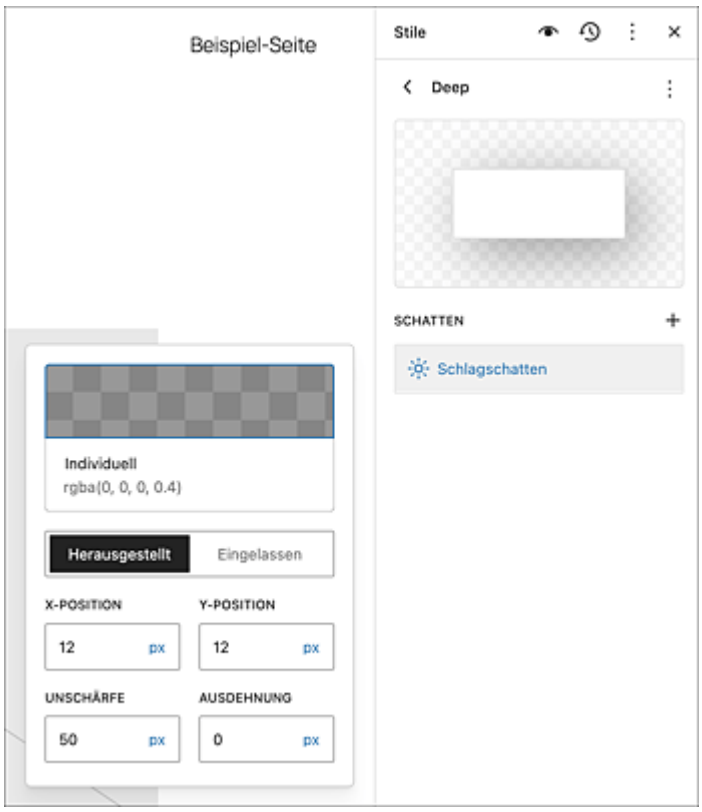


Abbildung 10.20 In der Bearbeitungsansicht eines Schattens können Sie alle Werte verändern, die sich auch über CSS setzen lassen würden.

Hinter dem letzten Punkt, Layout ❹, verbergen sich Einstellungsmöglichkeiten für die normale und weitere Inhaltsbreite, Abstände der gesamten Website sowie die Standard-Abstände zwischen Blöcken, wie die Spalten in einem Spalten-Block (siehe Abbildung 10.21).

Wenn Sie wieder in die oberste Ebene der Stile-Seitenleiste wechseln und den Punkt Blöcke ❼ anklicken, landen Sie in einer Liste mit allen Blöcken der Website. Ein Klick auf einen der Blöcke bringt Sie in eine ähnliche Ansicht wie eben bei den Typografie-Einstellungen: Viele Optionen, mit denen Sie das Aussehen des entsprechenden Blocks anpassen können. Beim Absatz-Block können Sie beispielsweise Farben und Abstände anpassen (siehe Abbildung 10.22). Diese Anpassungen wirken sich dann auf alle

Blöcke des entsprechenden Typs aus, die bereits eingefügt sind oder die Sie später noch einfügen, bei denen Sie für die veränderten Designeigenschaften keinen abweichenden Wert gesetzt haben. In diesem Fall würde dieser Wert den aus der Stile-Seitenleiste überschreiben. Welche Eigenschaften Sie verändern können, unterscheidet sich von Block zu Block.

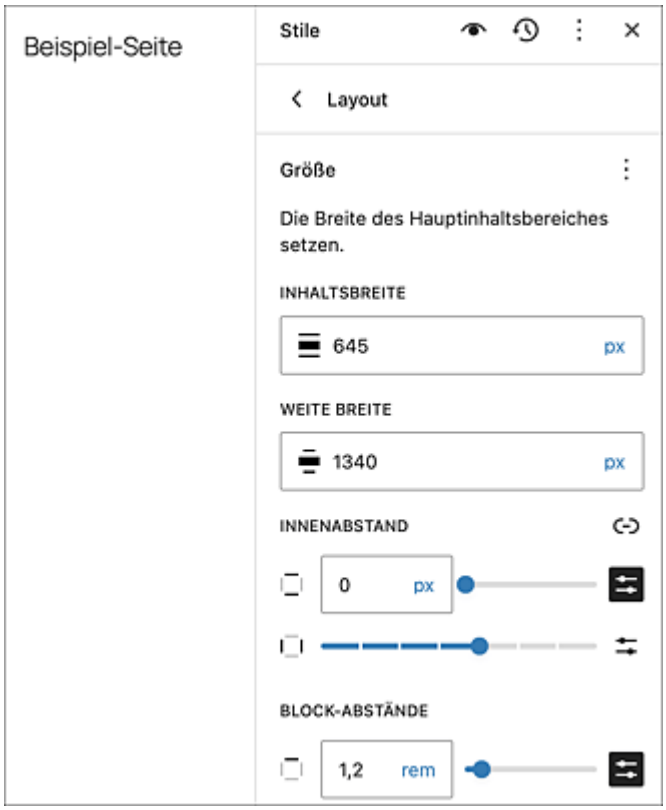


Abbildung 10.21 Im Bereich »Layout« lässt sich unter anderem die Inhaltsbreite anpassen.

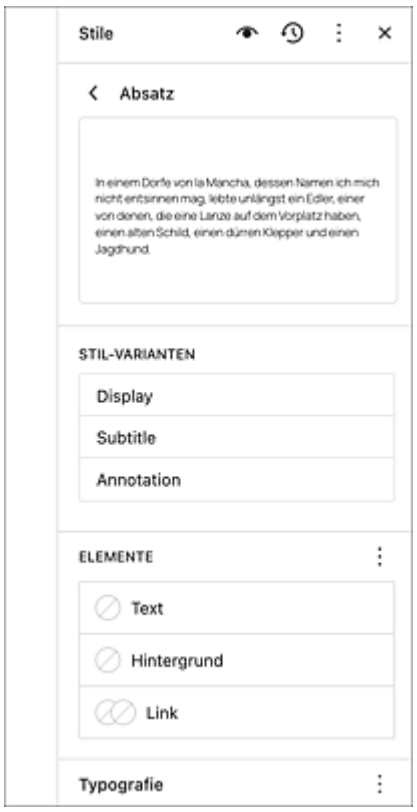


Abbildung 10.22 In der »Blöcke«-Liste können Sie Stile für alle verfügbaren Blöcke anpassen. Beim Absatz-Block haben Sie beispielsweise Einstellungsmöglichkeiten für Farbe, Typografie und Abstände.

Am Anfang sind diese ganzen Optionen und Möglichkeiten vielleicht etwas überwältigend. Nehmen Sie sich ein wenig Zeit, und machen Sie sich mit den vorhandenen Einstellungen vertraut, spielen Sie ein bisschen mit den Werten herum, und schauen Sie sich links im Editor die Auswirkungen an.

Template bearbeiten

Den größten Bereich im Website-Editor nimmt eine Vorschau Ihrer Website ein, in der Sie anpassen können, aus welchen Blöcken Ihre Website zusammengesetzt wird. Die Mechanik ist identisch mit der des Block-Editors für Ihre Inhalte: Sie können Blöcke auswählen, deren Einstellungen über die Block-Toolbar oder die Seitenleiste

rechts verändern, den Block-Inhalt anpassen, Blöcke entfernen oder neue Blöcke hinzufügen (siehe Abbildung 10.23).



Abbildung 10.23 Das Interface zum Bearbeiten der Blöcke, aus denen Ihre Website besteht, gleicht dem des Editors für die Inhalte Ihrer Beiträge und Seiten.

Wenn Sie sich einmal die eingesetzten Blöcke anschauen, finden Sie hier einige der Blöcke, die besonders nützlich für das Full-Site-Editing sind, zum Beispiel den Abfrage-Loop-Block, der die Liste der neuesten Beiträge anzeigt. Der Header und der Footer bestehen aus Template-Teil-Blöcken, das heißt, diese Inhalte können Sie in beliebigen Templates einbinden und wiederverwenden. Wenn Sie in einem Template-Teil eine Änderung vornehmen, zum Beispiel im Footer einen Absatz hinzufügen, wird das auch in allen anderen Templates angezeigt, die den Footer-Template-Teil nutzen.

Andere Templates bearbeiten

Bisher haben Sie das Archive-Template betrachtet, aber vielleicht (oder wahrscheinlich) möchten Sie auch mal ein anderes Template

verändern, zum Beispiel das zur Anzeige von Seiten. Um das Template zu wechseln, klicken Sie links oben auf das WordPress-Logo (oder Ihr Website-Icon) und wählen in der Seitenleiste den Eintrag Templates aus. Hier landen Sie in der bereits vorhin vorgestellten Raster-Ansicht aller Templates und können ein anderes Template zur Bearbeitung auswählen.

Templates hinzufügen

Vermissen Sie in Ihrem Theme eins der Standard-WordPress-Templates, können Sie es aus der Liste der Templates heraus erstellen, die Sie in der linken Seitenleiste des Website-Editors und dann per Klick auf Templates erreichen. Jetzt finden Sie oben rechts den Button Neues Template hinzufügen, und nach einem Klick darauf wird Ihnen eine Liste der Templates angezeigt, die im Theme nicht vorhanden sind und die von Ihnen erstellt werden können (siehe Abbildung 10.24).

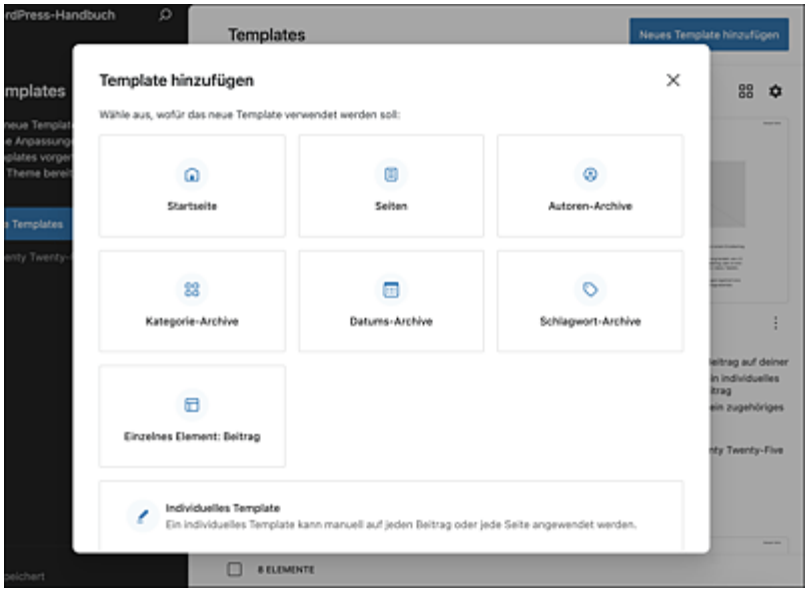


Abbildung 10.24 Wenn Sie sich in der Templates-Liste befinden, gibt es oben rechts einen Button zum Hinzufügen neuer Templates. Neben den Standard-WordPress-Templates, die im aktuellen Theme fehlen, können Sie auch ein »individuelles Template« erstellen, das Sie manuell für Beiträge und Seiten auswählen können.

Neben den Standard-Templates können Sie ganz unten auch ein Individuelles Template erstellen, das Sie anschließend manuell bei der Bearbeitung einer Seite oder eines Beitrags zuweisen können.

Klicken Sie beispielsweise auf den Eintrag Autoren-Archive, um ein Template für die Liste der Beiträge einer Autor*in zu erstellen, müssen Sie erst noch spezifizieren, dass es ein Template Alle Autoren sein soll, nicht nur für eine*n bestimmten. Anschließend öffnet sich wie gewohnt der Website-Editor und präsentiert Ihnen gegebenenfalls eine oder mehrere Vorlagen für das Template (siehe Abbildung 10.25).

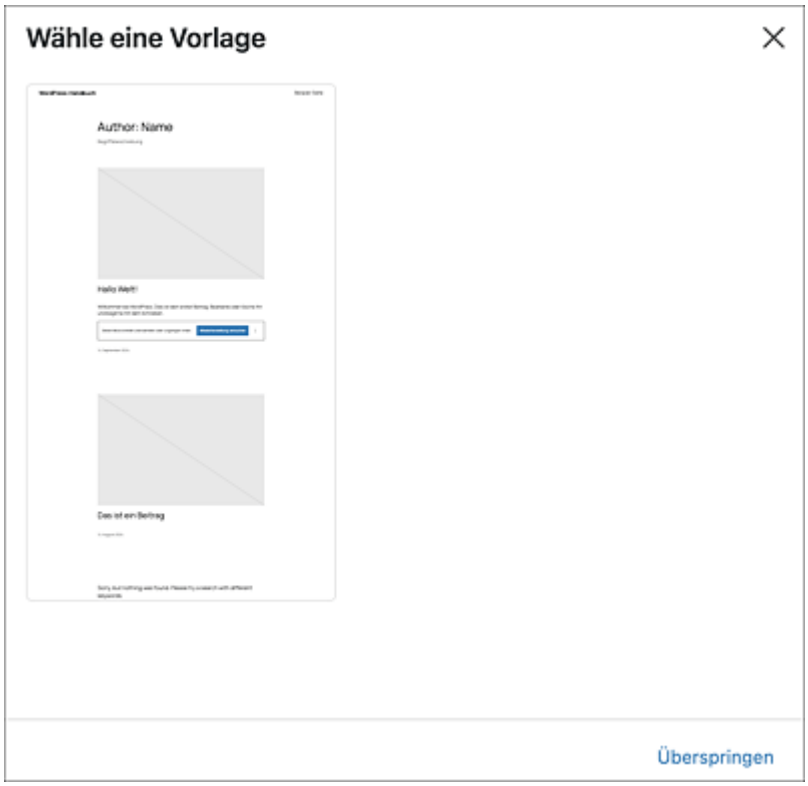


Abbildung 10.25 Bei der Erstellung eines Templates werden Ihnen gegebenenfalls Vorlagen angeboten, von denen Sie starten können.

Möchten Sie ohne Vorlage starten, klicken Sie einfach unten rechts auf Überspringen oder oben rechts auf das X, und Sie kommen in eine komplett leere Bearbeitungsansicht, in der Sie das Template zusammenbauen können (siehe Abbildung 10.26).



Abbildung 10.26 Bei einem neuen Template starten Sie von null, wenn Sie keine der Vorlagen auswählen. Aber auch wenn Sie keine Vorlage nutzen, können Sie sich per Copy-and-paste zumindest Dinge wie Header und Footer schnell aus anderen Templates rüberkopieren.

Wenn keine passende Vorlage für Ihr Template dabei ist, möchten Sie oft wahrscheinlich trotzdem nicht komplett von vorne anfangen, sondern die Struktur eines bestehenden Templates übernehmen, in diesem Fall zum Beispiel die des Templates Alle Archive. Dazu wechseln Sie einfach zu dem entsprechenden Template und markieren alle Blöcke. Manchmal ist das etwas hakelig, am besten suchen Sie sich am Rand einen Bereich, bei dem Ihre Maus keine Markierungen im Editor auslöst, und klicken dort hinein. Anschließend drücken Sie Strg/cmd + A, um alle Blöcke zu markieren, und anschließend kopieren Sie die Blöcke mit Strg/cmd + C (alternativ können Sie über das Drei-Punkte-Menü rechts oben im Website-Editor gehen, dort finden Sie einen Button Alle Blöcke kopieren, siehe Abbildung 10.27). Wechseln Sie nun zurück zum neuen Template, klicken Sie in den Editor, sodass Sie durch Lostippen einen Absatz einfügen könnten, und drücken Strg/cmd + V, um die Blöcke einzufügen.

Eine sinnvolle Erweiterung des Templates für die Ansicht aller Beiträge von Autor*innen könnte das Profilbild, der Avatar, sein. Fügen Sie dafür den Avatar-Block beispielsweise zentriert über dem Titel des Archivs ein, und geben Sie ihm vielleicht noch einen Radius von 100 %. Wenn Sie das Template nun speichern, sollte das

Archiv mit den Beiträgen eines Autors oder einer Autorin aussehen wie das in Abbildung 10.28. Das Archiv rufen Sie über eine URL im Format *example.com/author/IhrBenutzername* auf.

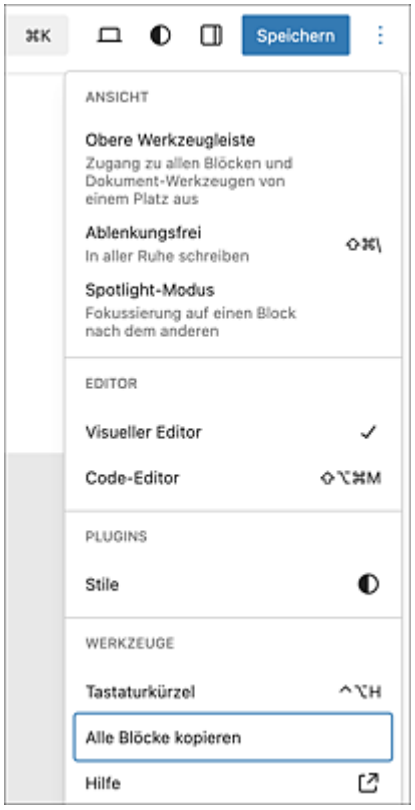


Abbildung 10.27 Die Blöcke aus einem bestehenden Template können gut als Ausgangslage für ein neues Template genutzt werden, um nicht bei null starten zu müssen, wenn es keine passende Vorlage gibt.



Abbildung 10.28 Das neue Template für Seiten von Autor*innen in Aktion

Neben der Template-Liste im Website-Editor gibt es eine zweite Möglichkeit, Templates hinzuzufügen (und auch bestehende zu bearbeiten): den Template-Editor. Den erreichen Sie aus der Bearbeitungsansicht eines Beitrags oder einer Seite heraus, und zwar rechts in der Dokumenten-Seitenleiste über den Eintrag Template, wo Sie das Template für die Seite oder den Beitrag festlegen können. Wenn Sie auf das aktuelle Template klicken, öffnet sich ein Overlay, das Ihnen ermöglicht, das Template zu wechseln oder ein neues Template zu erstellen (siehe Abbildung 10.29).

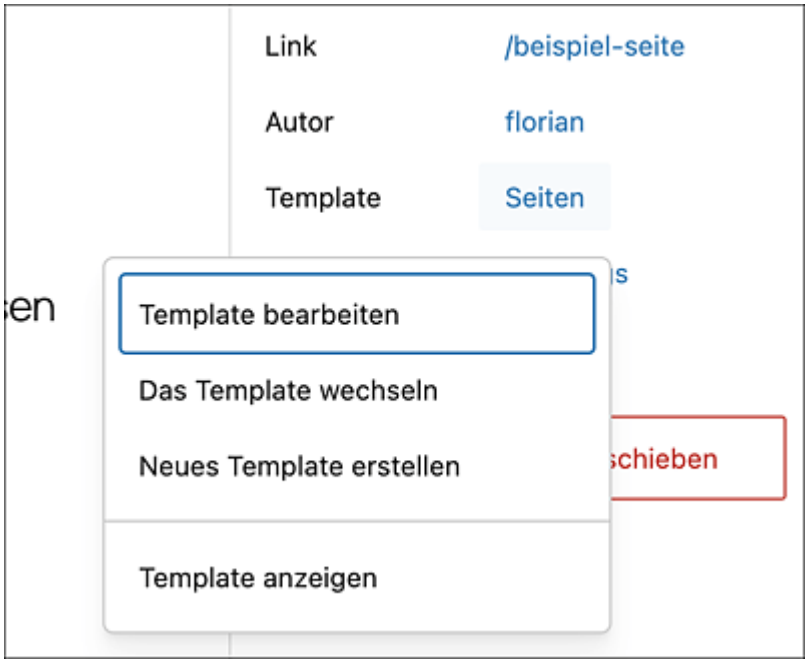


Abbildung 10.29 Durch einen Klick auf das aktuelle Template eines Beitrags oder einer Seite öffnet sich ein Overlay, das Ihnen einen Template-Wechsel ermöglicht oder Gelegenheit gibt, ein komplett neues Template zu erstellen.

Nach einem Klick auf Template bearbeiten oder Neues Template erstellen öffnet sich der Template-Editor, der nicht die gleichen Funktionen anbietet wie der Site-Editor – so können Sie von hier aus nicht die Stile bearbeiten. Aber der Bereich zum Bearbeiten des Templates ist gleich (siehe Abbildung 10.30).

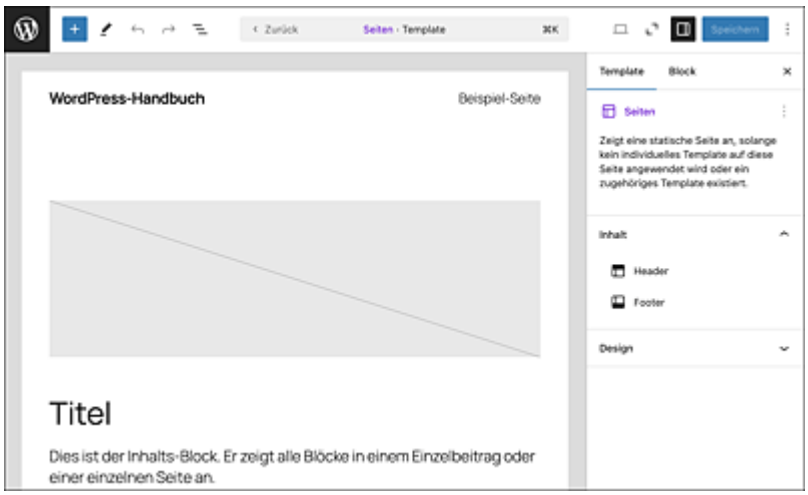


Abbildung 10.30 Mit dem Template-Editor können Sie Templates bearbeiten, die für Seiten und Beiträge zum Einsatz kommen. Er bietet nicht alle Funktionen, die der Site-Editor mitbringt, aber die Bearbeitungsansicht für die Blöcke des Templates ist gleich.

Template-Teile bearbeiten

Neben den Templates, die für die Darstellung ganzer Webseiten zuständig sind, gibt es noch das Konzept der *Template-Teile*. Wie der Name schon sagt, handelt es sich dabei nicht um vollständige Templates, sondern nur um Teilbereiche, zum Beispiel den Header oder Footer. Über den Template-Teil-Block können diese Teile in Templates (oder auch in andere Template-Teile) eingebunden werden. So müssen Sie beispielsweise den Footer Ihrer Website nicht in jedem Template pflegen, sondern binden in die Templates den Template-Teil »Footer« ein, und Änderungen an dem Template-Teil werden dann in allen Templates angezeigt, die den Template-Teil einbinden.

Die bereits vom Theme mitgelieferten Template-Teile finden Sie im Website-Editor in der Sidebar links, die Sie über das Logo oben links öffnen. Dort klicken Sie auf Vorlagen und finden dann im oberen Bereich der Seitenleiste Buttons für die Template-Teile (siehe Abbildung 10.31). Hier haben Sie die Möglichkeit, entweder einen

bestehenden Template-Teil zu verändern oder über den Hinzufügen-Button oben rechts einen neuen zu erstellen.

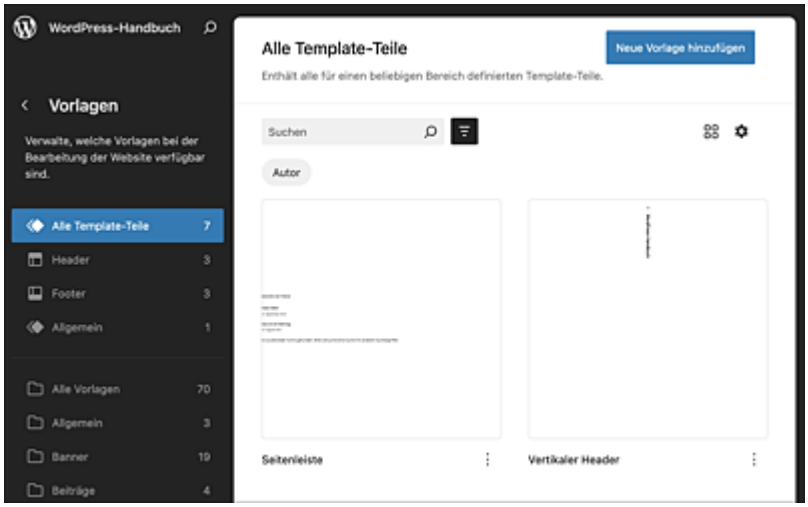


Abbildung 10.31 Über »Vorlagen« • »Alle Template-Teile« erreichen Sie eine Raster-Ansicht der Template-Teile, die vom Theme mitgebracht oder von Ihnen erstellt wurden.

Klicken Sie hier auf einen Template-Teil, öffnet sich erneut der Website-Editor, der diesmal aber keine Vorschau einer kompletten Webseite anzeigt, sondern nur den ausgewählten Template-Teil, beispielhaft in Abbildung 10.32 der Header-Template-Teil von Twenty Twenty-Five. Über die Anfasser rechts und links von der Vorschau des Template-Teils können Sie den Viewport schmaler schieben, in dem der Template-Teil angezeigt wird, um zu sehen, wie er sich bei anderen Viewport-Größen verhält.

Neben der Möglichkeit, die Template-Teile über diese Ansicht zu bearbeiten, können Sie die Inhalte von Template-Teilen auch direkt bearbeiten, wenn Sie sich ein Template anschauen, das sie einbindet.

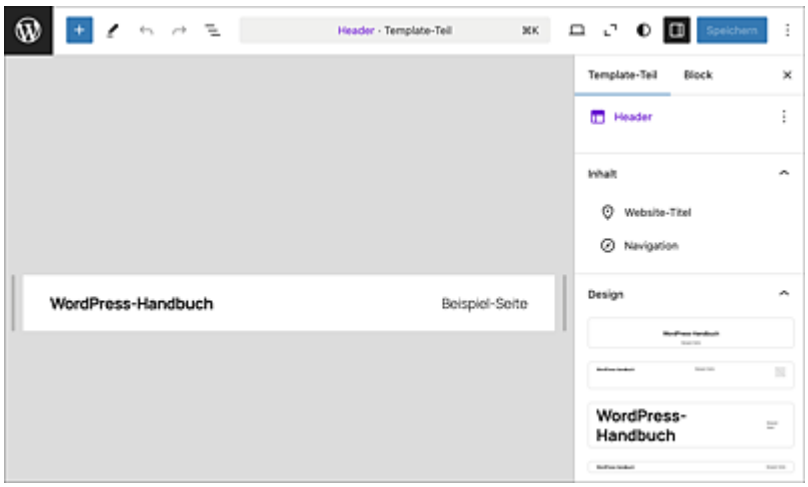


Abbildung 10.32 Für die Bearbeitung eines Templates-Teils öffnet sich wieder der Website-Editor, diesmal aber nur mit dem Template-Teil ohne den Rest einer kompletten Webseite drum herum.

Template-Teile hinzufügen

Wie bei den Templates können Sie auch neue Template-Teile über den Hinzufügen-Button in der Template-Teile-Liste erstellen. Nach einem Klick auf den Button müssen Sie noch auswählen, dass Sie ein Neues Template-Teil hinzufügen möchten und keine Vorlage. Danach öffnet sich ein Overlay, in dem Sie einen Namen für den Template-Teil vergeben und einen Bereich auswählen sollen (siehe Abbildung 10.33). Über den Namen finden Sie den Template-Teil später wieder, er sollte also beschreiben, wofür der Template-Teil da ist. Der Bereich ist eine Art Kategorisierung. Wenn es sich bei Ihrem Template-Teil um einen Header oder Footer handelt, wählen Sie den entsprechenden Bereich, ansonsten kommt er in den Bereich Allgemein.

Wenn Sie einen Namen und einen Bereich ausgewählt haben und auf das Plus klicken, öffnet sich der Website-Editor mit einem leeren Bearbeitungsbereich, in den Sie die Blöcke für den Template-Teil einfügen können.

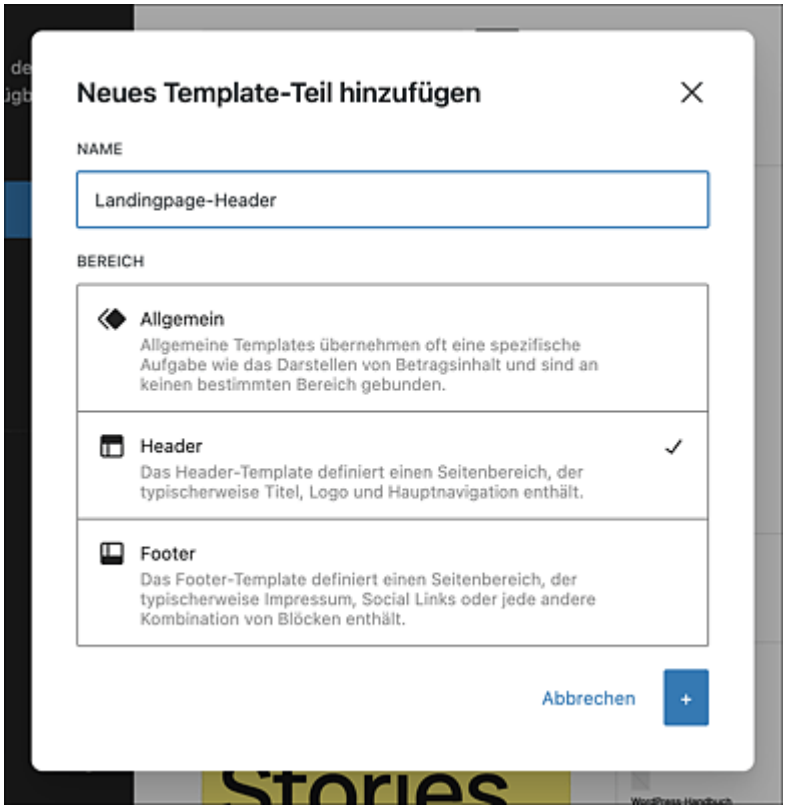


Abbildung 10.33 Beim Hinzufügen eines Template-Teils werden zuerst Name und Bereich festgelegt.

Eine andere Möglichkeit, einen Template-Teil zu erstellen, führt über den Template-Teil-Block. Wenn Sie in ein Template einen Template-Teil-Block einbinden, bekommen Sie nämlich die Wahl, ob Sie einen bestehenden Template-Teil nutzen oder einen neuen Template-Teil erstellen möchten. Abbildung 10.34 zeigt den Schritt, wenn ein Template-Teil-Block eingefügt und in dem Block auf Leer beginnen geklickt wurde: Es öffnet sich ein Overlay zum Eingeben eines Template-Teil-Titels, und nach dem Klick auf Erstellen bleiben Sie in der Bearbeitungsansicht des Templates und haben darin einen neuen Template-Teil, in den Sie Blöcke einfügen können. In der Block-Seitenleiste können Sie bei Bedarf anpassen, ob es sich bei dem Template-Teil nicht um einen allgemeinen Teil, sondern einen Header oder Footer handelt.

Template-Teile einbinden

Nachdem Sie einen Template-Teil erstellt und gespeichert haben, können Sie ihn über den Template-Teil-Block einbinden oder direkt über den vergebenen Namen im Block-Inserter finden.

Template-Teile lassen sich nicht innerhalb jedes Blocks einfügen, so ist es beispielsweise nicht möglich, Template-Teile innerhalb des Beitrags-Template-Blocks zu haben, innerhalb von anderen Template-Teilen geht es aber zum Beispiel sehr wohl.

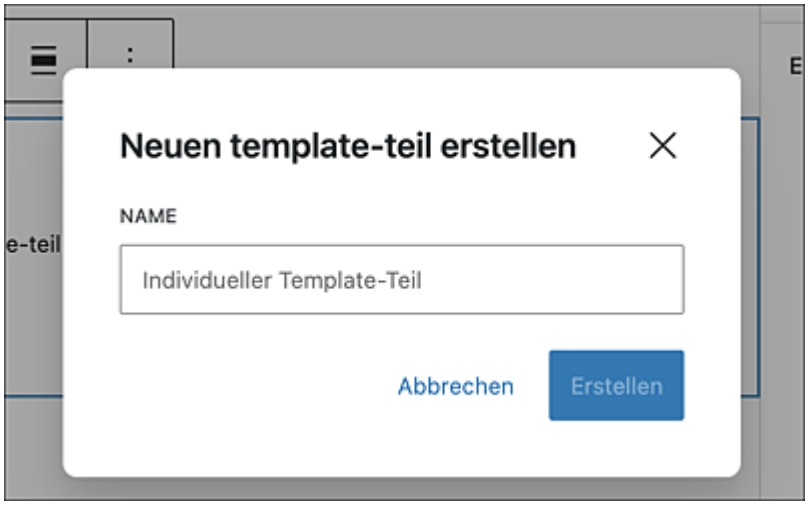


Abbildung 10.34 Template-Teile können auch direkt aus den Blöcken heraus erstellt werden, mit denen bestehende Template-Teile angezeigt werden.

Änderungen im Website-Editor speichern

Vielleicht haben Sie schon ein paar kleine Anpassungen gemacht und gespeichert? Falls ja, ist Ihnen wahrscheinlich aufgefallen, dass sich der Dialog, der nach dem Klick auf den Speichern-Button erscheint, von dem bei Beiträgen und Seiten unterscheidet. Im Website-Editor bekommen Sie vor dem finalen Abspeichern noch mal angezeigt, in welchen Komponenten Sie Änderungen vorgenommen haben: zum Beispiel welche Templates Sie bearbeitet haben, welche Template-Teile, ob Dinge in der Stile-Sidebar geändert

oder Menüs angepasst wurden. Sie können, wenn Sie einzelne Anpassungen nicht übernehmen möchten, hier den Haken bei dem entsprechenden Eintrag entfernen (siehe Abbildung 10.35).

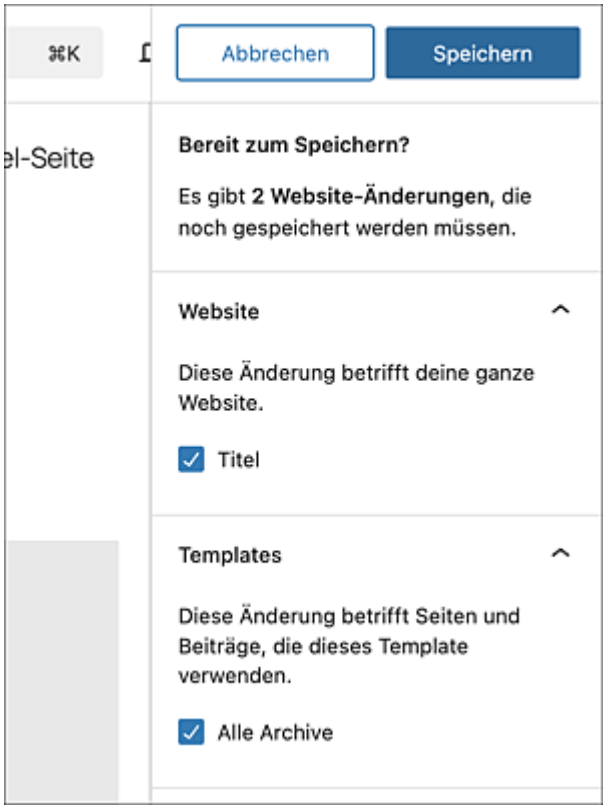


Abbildung 10.35 Nach dem Klick auf den »Speichern«-Button im Website-Editor werden alle Komponenten aufgelistet, die Sie verändert haben. Hier haben Sie dann die Möglichkeit, Änderungen an einzelnen Komponenten doch nicht zu speichern.

Block-Vorlagen erstellen

Unter dem Punkt Vorlagen in der linken Seitenleiste des Website-Editors finden Sie nicht nur die Template-Teile, sondern auch die Block-Vorlagen. Diese Vorlagen sind dazu gedacht, das Einfügen komplexerer oder einfach häufig wiederkehrender Block-Kombinationen zu erleichtern. Eine Block-Vorlage kann über den Block-Inserter eingefügt werden, den Sie in Kapitel 6, »Inhalte veröffentlichen«, näher kennengelernt haben.

Wenn Sie eine Block-Vorlage erstellen möchten, haben Sie zwei Möglichkeiten:

1. **Eine Block-Vorlage aus dem Vorlagen-Bereich des Website-Editors erstellen:** Wenn Sie sich in der linken Website-Editor-Seitenleiste in den Vorlagen-Bereich begeben, können Sie oben rechts eine Neue Vorlage hinzufügen (darüber hatten Sie vorhin bereits einen Template-Teil erstellt). Nach Klick auf den Button können Sie auswählen, dass Sie eine Neue Vorlage hinzufügen möchten. Nach einem Klick auf diesen Punkt öffnet sich ein Overlay, in dem Sie einen Namen und optional eine Kategorie für die Vorlage auswählen. Außerdem können Sie hier angeben, ob die Vorlage Synchronisiert sein soll, das bedeutet, wenn Sie die Vorlage auf mehreren Seiten eingebunden haben und an einer Stelle den Inhalt ändern, wird er auch auf allen anderen Seiten aktualisiert (siehe Abbildung 10.36).

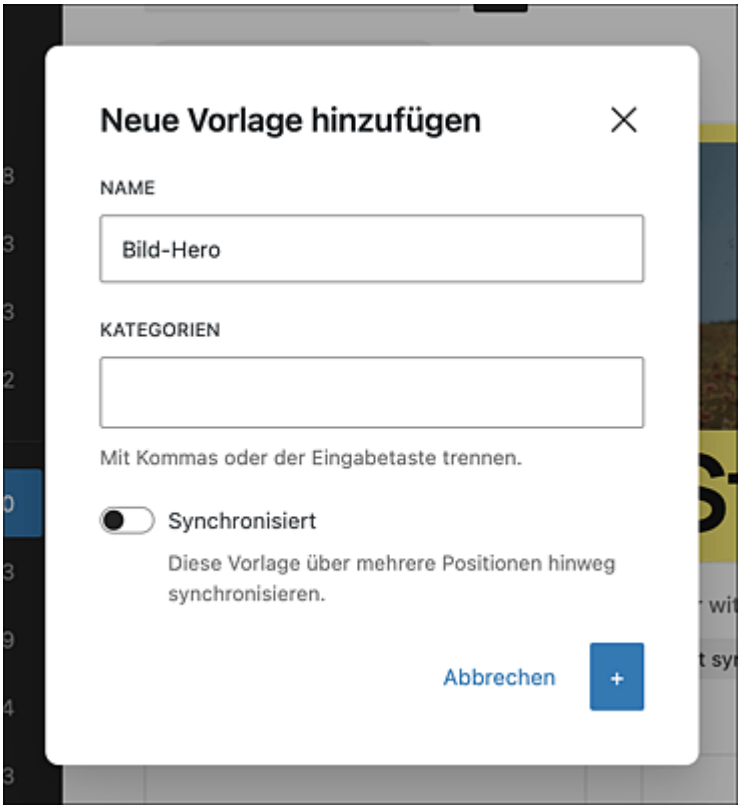


Abbildung 10.36 Eine Vorlage braucht einen Namen und optional eine Kategorie. Sie können außerdem festlegen, ob die Vorlage an allen Stellen, wo sie eingefügt wird, immer denselben Inhalt haben soll.

Nachdem Sie auf das Plus-Icon geklickt haben, landen Sie in einem ähnlichen Editor wie für die Template-Teile: Sie sehen nur den Inhalt der Vorlage, kein Template darum herum (siehe Abbildung 10.37).

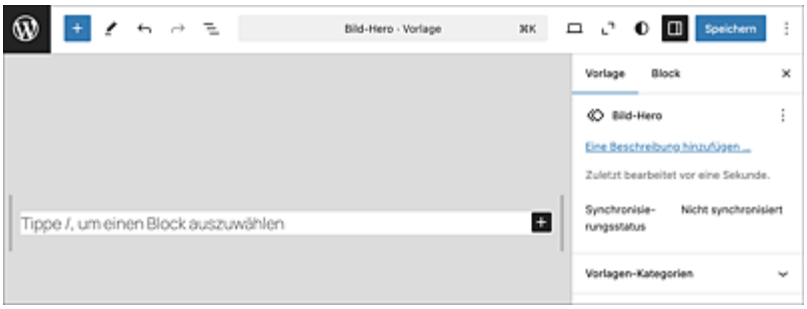


Abbildung 10.37 Der Editor für die Vorlage ähnelt dem für Template-Teile.

2. Die zweite Option ist, **Blöcke in eine Vorlage umzuwandeln**. Statt über den Vorlagen-Bereich zu gehen, können Sie auch ganz einfach aus dem Block-Editor heraus eine Vorlage erstellen aus Blöcken, die dort eingefügt wurden. Wenn Sie zum Beispiel ein Teaser-Element als Vorlage erstellen möchten, das für dieses vereinfachte Beispiel aus einem Gruppe-Block sowie einer darin enthaltenen Überschrift, einem Absatz und einem Button bestehen soll, können Sie wie folgt vorgehen:

   Fügen Sie die Blöcke im Editor ein (ganz egal, wo), und wählen Sie danach den Gruppe-Block aus. Gehen Sie anschließend über das Drei-Punkte-Menü in der Werkzeugleiste des Blocks auf den Punkt Vorlage erstellen (siehe Abbildung 10.38).

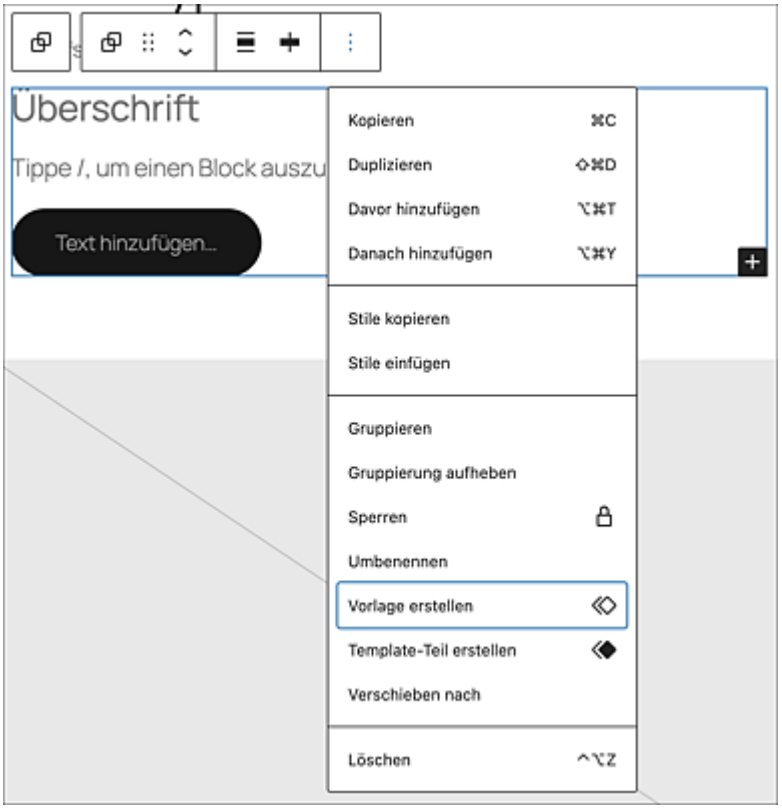


Abbildung 10.38 Sie können einfach direkt aus dem Editor heraus einen oder mehrere Blöcke in eine Vorlage umwandeln, sodass Sie den oder die Blöcke wiederverwenden können.

Nach einem Klick auf VORLAGE ERSTELLEN öffnet sich dasselbe Overlay, das Sie schon bei der Erstellung einer Vorlage gesehen haben (siehe Abbildung 10.36).

Tipp: Blöcke in Vorlagen »sperren«

Wenn Sie sicherstellen möchten, dass nur genau die Blöcke in einer Vorlage genutzt werden, aus denen Sie die Vorlage erstellt haben, und nicht etwa die Reihenfolge der Blöcke geändert wird oder Blöcke gelöscht werden, können Sie die Vorlage sperren. Folgende Varianten sind möglich:

- Umsortieren von Blöcken verbieten
- Entfernen von Blöcken verbieten

- Einfügen von Blöcken verbieten (kann nur über die Code-Ansicht eingestellt werden)
- Nutzung der Block-Einstellungen verbieten (auch nur über die Code-Ansicht möglich)

Wenn Sie eine Vorlage beispielsweise mit einem Gruppe-Block haben, der weitere Blöcke enthält, wie der einfache Teaser-Block aus dem Beispiel zum Anlegen einer Vorlage aus dem Block-Editor heraus, können Sie den Gruppe-Block auswählen und auf das Drei-Punkte-Menü klicken. Dort finden Sie den Punkt **Sperren**, der ein Overlay öffnet und Ihnen zwei Optionen bietet: **Bewegen deaktivieren**, um das Verschieben von Blöcken zu verbieten, und **Entfernen verhindern**, um das Löschen von Blöcken zu verbieten (siehe Abbildung 10.39).

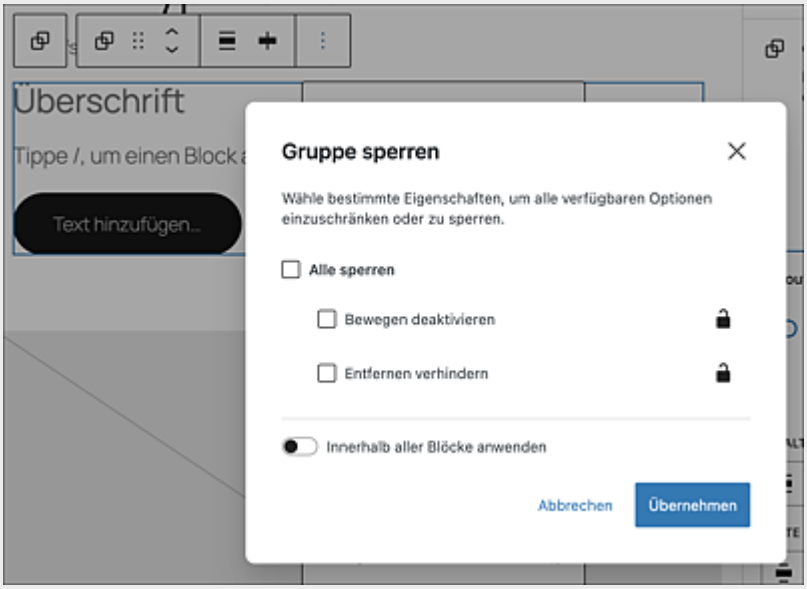


Abbildung 10.39 Durch das »Sperren« von Blöcken kann beispielsweise verhindert werden, dass die Reihenfolge von Blöcken verändert wird.

Über den Toggle **Innerhalb aller Blöcke anwenden** können Sie die Einstellung auf alle Kindblöcke anwenden. Wie Sie Blöcke durch Veränderung des Markups sperren können, sodass nur

noch die Inhalte verändert werden können, lesen Sie in Kapitel 23 zur Theme-Entwicklung in Abschnitt 23.4.4.

10.4 Classic Theme anpassen

Bei Classic Themes haben Sie im Backend ein paar unterschiedliche Anlaufstellen, wenn Sie Anpassungen vornehmen möchten. Wenn ein Classic Theme keine eigene Einstellungsseite mitbringt, sollten Sie im Customizer alle Optionen finden. Die Verwaltung von Menüs und Widgets ist in ihren eigenen Backend-Bereichen aber gegebenenfalls etwas komfortabler.

10.4.1 Classic Theme über den Customizer konfigurieren

Wenn Sie die Live-Vorschau eines Themes geöffnet haben, befinden Sie sich im sogenannten *Customizer*. Im Customizer nehmen Sie Einstellungen vor, um das Classic Theme weiter anzupassen. Was veränderbar ist, ist von Theme zu Theme verschieden. Von WordPress selbst kommen die Punkte Website-Informationen, Menüs, Widgets, Homepage-Einstellungen und Zusätzliches CSS.

Sie erreichen den Customizer nicht nur über die Live-Vorschau eines Themes, sondern für das gerade aktive Theme über Design • Customizer.

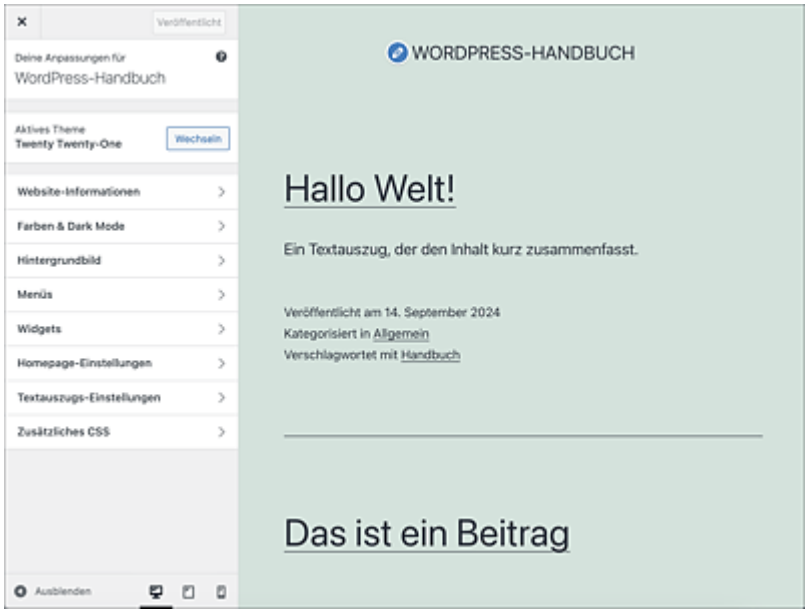


Abbildung 10.40 Der Customizer ermöglicht Ihnen die Anpassung von Classic Themes mit einer Live-Vorschau der Änderungen.

Auf den folgenden Seiten werden Sie den Customizer und die Anpassung eines Themes im Customizer anhand des letzten WordPress-Standard-Themes kennenlernen, das als Classic Theme umgesetzt wurde: Twenty Twenty-One. Für andere Classic Themes werden sich vermutlich die verfügbaren Optionen unterscheiden, aber Sie werden die Mechanismen hoffentlich wiedererkennen. Es gibt auch Classic Themes, die nicht den Customizer nutzen, sondern eine eigene Backend-Seite mit Optionen mitbringen, oft als Unterpunkt unter Design. Dort gibt es dann zwar mehr Platz für die Einstellungen, aber keine Vorschau.

Responsive Design testen

Beginnen wir zunächst mit einer kleinen Option, die ganz am unteren Ende der linken Seitenleiste des Customizers zu finden ist: Über drei Icons können Sie die Größe der Vorschau auf der rechten Seite steuern (siehe Abbildung 10.41):

- Das Computer-Bildschirm-Icon steht dafür, dass die Vorschau den gesamten verfügbaren Platz einnimmt. Durch Änderung der Größe Ihres Browser-Fensters können Sie die Breite und Höhe in diesem Modus anpassen.
- Über das Tablet-Icon können Sie die Vorschau auf die Maße eines generischen Tablets einschränken.
- Über das Smartphone-Symbol wechselt die Vorschau auf die Größe, die einem Smartphone entsprechen könnte.



Abbildung 10.41 Am unteren Ende der Sidebar des Customizers finden Sie die Option, die Vorschau auf unterschiedliche Viewport-Größen zu verändern.

Auch wenn die Funktion da unten etwas versteckt ist, ist sie doch ein nützliches Hilfsmittel, um schnell zu testen, wie die Website auf kleineren Viewports aussieht. Das ersetzt kein Testen der Website auf echten Geräten, und es deckt nur einen sehr eingeschränkten Bereich an Viewport-Größen ab, aber für einen ersten Eindruck tut's das.

Theme anpassen

In diesem Abschnitt geht es um die Einstellungsmöglichkeiten des Customizers. Je nachdem, welches Theme Sie einsetzen, können mehr oder weniger als die hier beschriebenen Optionen vorhanden sein.

Hinweis: Wenn Sie sich in einem Unterbereich des Customizers befinden, kommen Sie oben links über den Pfeil eine Ebene höher. Das X-Icon ganz links oben bringt Sie an die Stelle, von der aus Sie den Customizer geöffnet haben.

- **Website-Informationen**
  Hier können Sie Logo, Titel der Website, Untertitel und Website-Icon anpassen. Titel der Website und Untertitel entsprechen den Werten, die Sie im Admin-Menü auch unter Einstellungen • Allgemein verändern können. Das Website-Icon wird auch *Favicon* genannt.
- **Farben & Dark-Mode**
  In diesem Bereich können Sie die Farbe des Hintergrunds anpassen und mit der Option Dark-Mode-Unterstützung aktivieren, dass das Theme sich je nach Einstellung der Benutzer*in im normalen hellen oder einem dunklen Design mit heller Schrift zeigt.
- **Hintergrundbild**
  Erlaubt Ihnen, ein Bild festzulegen, das als Hintergrund der Website angezeigt wird. Wenn ein Bild ausgewählt ist, können Sie noch die Bildposition und Bildgrösse anpassen und einstellen, ob sich das Hintergrundbild wiederholen und/oder ob es mit der Seite mitscrollen soll (siehe Abbildung 10.42).

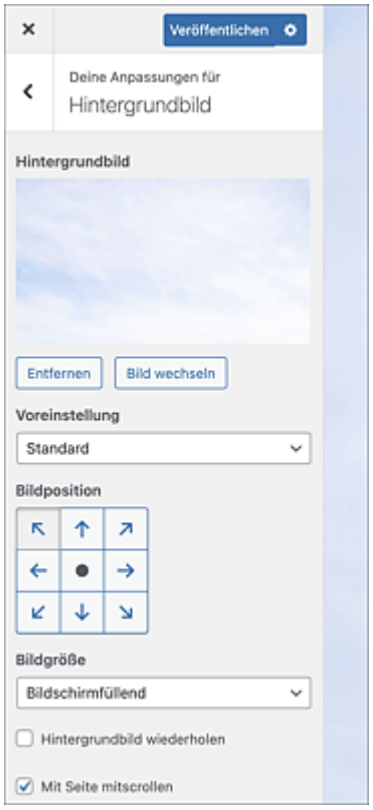


Abbildung 10.42 Wenn Sie ein Hintergrundbild festgelegt haben, erscheinen zusätzliche Optionen, um die Darstellung des Bilds zu steuern.

- **Menüs**
  Hier können Sie bestehende Menüs bearbeiten oder neue Menüs erstellen und diese Menüs den Menü-Positionen des Themes zuweisen. Twenty Twenty-One bietet zwei Positionen an: die Hauptmenü-Position oben rechts im Header und die Untermenü-Position, die über der durchgezogenen Linie im Footer angezeigt wird. Wenn Sie auf den Punkt Menü-Positionen klicken, sehen Sie eine Liste der Menü-Positionen des Themes und können über die Dropdown-Liste bestehende Menüs auswählen (siehe Abbildung 10.43).

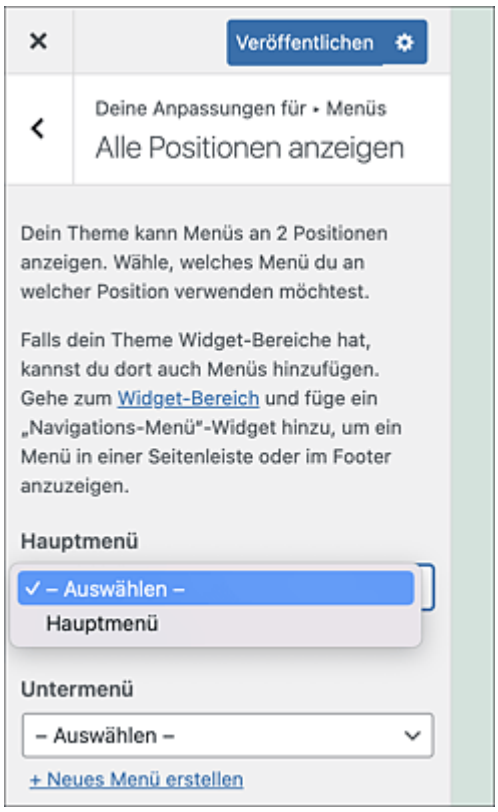


Abbildung 10.43 Wenn Sie Menüs erstellt haben, können Sie diese einer Menüposition zuweisen und dadurch an der entsprechenden Stelle im Theme anzeigen.

- Wenn Sie noch kein Menü haben, können Sie über den Button Neues Menü erstellen in der ersten Ebene des Menü-Panels, in dem Sie auch zu den Positionen gelangen, ein neues Menü anlegen. Einfacher geht das zwar gegebenenfalls auf der eigenen Seite für Menüs, die Sie im Admin-Menü unter Design • Menüs erreichen. Aber es geht auch hier, und Sie haben direkt eine Vorschau des Ergebnisses. Sie müssen dem Menü einen Namen geben und können optional im ersten Schritt auch schon eine Menü-Position auswählen (siehe Abbildung 10.44).
- Nach einem Klick auf Weiter gelangen Sie in eine Ansicht, in der Sie Einträge hinzufügen oder bestehende Einträge umsortieren und entfernen können (siehe Abbildung 10.45). Wie bei den anderen Einstellungsmöglichkeiten im Customizer sehen Sie auch hier die Änderung direkt rechts in der Vorschau, wenn das Menü einer

Position zugeordnet ist oder über ein Navigations-Widget eingebunden wird. Diese zweite Ansicht ist dieselbe, die auch bei der Bearbeitung eines bestehenden Menüs angezeigt wird.



Abbildung 10.44 Im ersten Schritt für ein neues Menü im Customizer legen Sie einen Namen und optional die Menü-Position fest.

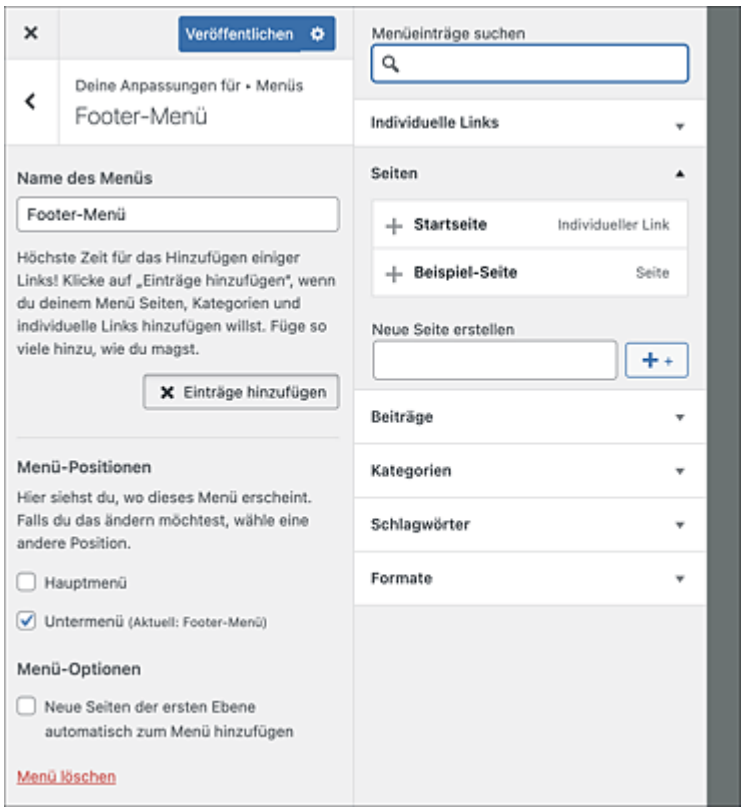


Abbildung 10.45 Im zweiten Schritt der Menü-Erstellung gelangen Sie in die Ansicht, in der Sie die Menüpunkte hinzufügen und verändern können.

- **Widgets**
  Bis WordPress 5.8 waren Widgets ein eigenes Konzept von statischen oder dynamischen Elementen, die Sie in Widget-Bereichen platzieren konnten. Mit WordPress 5.8 wurden die Widget-Bereiche zu weiteren Instanzen des Block-Editors und die Widgets abgelöst durch die Blöcke, die Sie auch für Seiten und Beiträge nutzen. Twenty Twenty-One bietet einen Widget-Bereich im Footer an. Wie die Menüs können Sie auch die Widgets in einem eigenen größeren Bereich bearbeiten, den Sie im Admin-Menü über Design • Widgets erreichen können. Allerdings verlieren Sie da wieder den Vorteil der Live-Vorschau.
- **Homepage-Einstellungen**
  Hier können Sie dieselben Einstellungen für die Startseite Ihrer Website vornehmen wie unter Einstellungen • Lesen im Admin-

Menü: Sie können festlegen, ob die Startseite Deine letzten Beiträge oder Eine statische Seite anzeigen soll. Ist Eine statische Seite ausgewählt, müssen Sie noch festlegen, welche Seite die Homepage und welche die Beitragsseite sein soll.

- **Textauszugs-Einstellungen**
  Dieses Panel ermöglicht Ihnen zu steuern, ob auf Archivseiten Eine Kurzfassung der Beiträge oder der vollständige Text zu sehen sein soll.

- **Zusätzliches CSS**
  Wenn Sie bei Ihrem Theme an den Punkt kommen, dass Sie gerne ein paar visuelle Anpassungen machen würden, die nicht über die verfügbaren Optionen machbar sind, können Sie in diesem Feld CSS einfügen (siehe Abbildung 10.46).



Abbildung 10.46 Sie können im Customizer das Aussehen des Themes über die verfügbaren Optionen hinaus anpassen, indem Sie eigenes CSS in das Feld »Zusätzliches CSS« einfügen.

CSS steht für *Cascading Style Sheets* und ist die Styling-Sprache des Webs. Die hier eingetragenen Regeln bleiben bei Theme-Updates erhalten, da sie in der Datenbank gespeichert werden. Sie sind allerdings an das Theme gekoppelt, werden also bei einem Theme-Wechsel nicht mit übernommen (was meist auch wenig sinnvoll ist, da sie sich ja auf ein bestimmtes Theme beziehen).

Tipp: Öffentliche Vorschau von Customizer-Änderungen und Veröffentlichungsplanung

Neben der sofortigen Veröffentlichung der Änderungen im Customizer über den Button **Veröffentlichen** oben rechts in der Customizer-Seitenleiste können Sie über das Zahnrad rechts neben dem Button ein Overlay öffnen, das Ihnen weitere Optionen anbietet (bei der Auswahl einer der Optionen verändert sich die Beschriftung des **Veröffentlichen**-Buttons entsprechend):

- Sie können den aktuellen Stand **Speichern**, um später weiterzuarbeiten. Wenn Sie die Änderungen gespeichert haben, können Sie diesen gespeicherten Zustand auch über einen **Vorschau-Link teilen**, den Sie unten in dem Bereich sehen, der sich bei einem Klick auf das Zahnrad geöffnet hat.
- Sie können die Veröffentlichung **Planen**, sodass die Änderungen zu einem bestimmten Zeitpunkt in der Zukunft online gehen. Wenn Sie diese Option wählen, öffnet sich unter dem **Planen**-Punkt ein Bereich, in dem Sie das Datum einstellen können (siehe Abbildung 10.47).

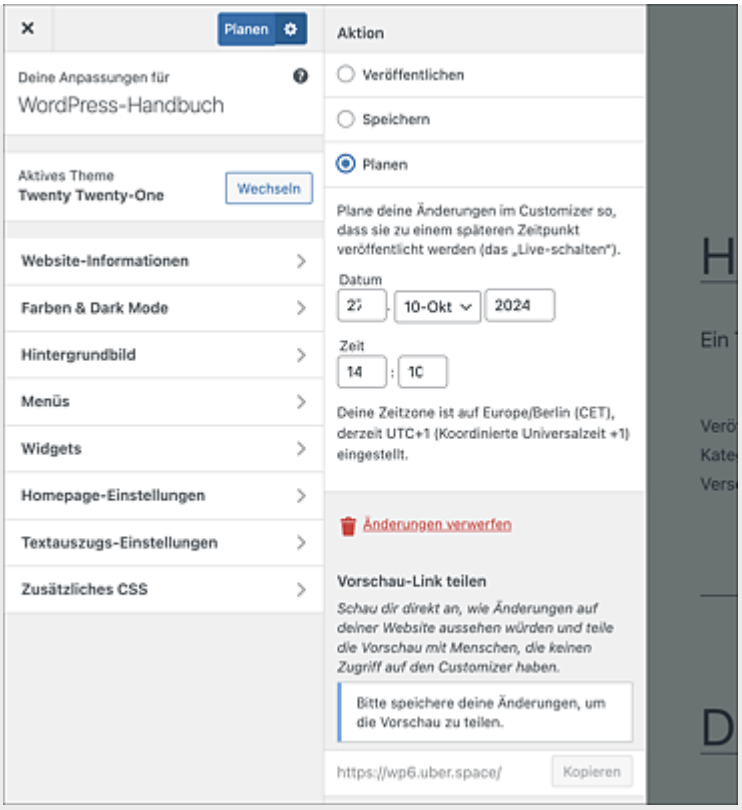


Abbildung 10.47 Sie können Ihre Änderungen im Customizer auch für die Zukunft planen, sodass sie automatisch zu einem späteren Zeitpunkt veröffentlicht werden.

10.4.2 Navigationsmenü erstellen

Eben haben Sie bereits gesehen, wie Sie im Customizer ein Menü erstellen können. Nun lernen Sie die dedizierte Admin-Seite kennen, die die Erstellung und Bearbeitung von Menüs gegebenenfalls etwas vereinfacht, da mehr Platz vorhanden ist. Öffnen Sie die Seite über das linke Admin-Menü unter Design • Menüs.

Viele Seiten haben mehr als ein Menü, weil es Seiten wie das Impressum und die Datenschutzerklärung gibt, die zwar aus rechtlichen Gründen sehr wichtig sind, aber für normale Besucher*innen meist recht uninteressant. Darum sind solche

Seiten oft nicht im prominent platzierten Hauptmenü zu finden, sondern etwa in einer Navigation im Footer. In Classic Themes registriert das Theme sogenannte Menü-Positionen, an denen Sie ein Menü ausgeben können. Darüber hinaus gibt es den Navigationsmenü-Block, mit dem Sie ein Menü in einem Widget-Bereich anzeigen können.

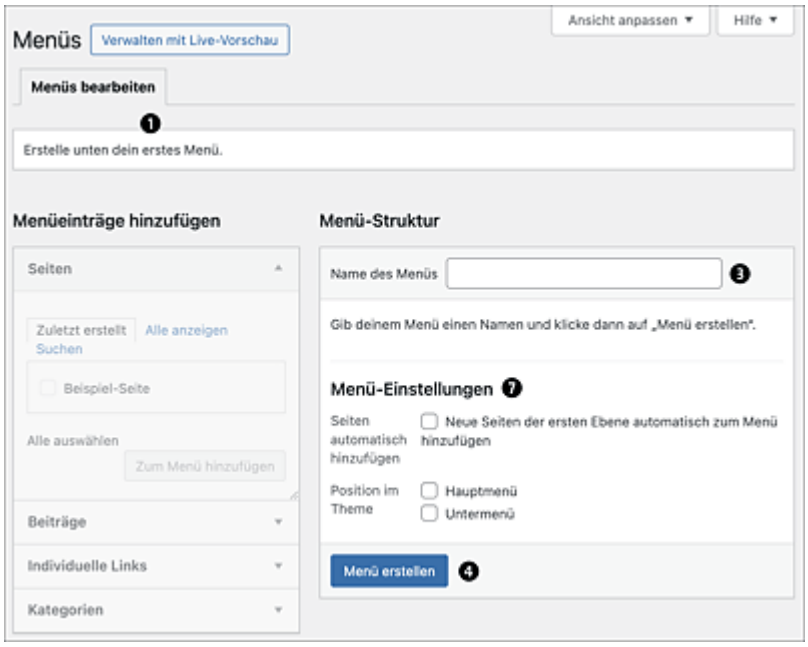


Abbildung 10.48 Unter »Design« • »Menüs« können Sie neue Menüs erstellen und bestehende Menüs bearbeiten.

Wenn Sie die Seite Design • Menüs geöffnet haben, werden Sie von einer Ansicht begrüßt, die ähnlich der in Abbildung 10.48 sein sollte. Wenn Sie bereits ein Menü erstellt haben, gibt es ein paar mehr Schaltflächen. Diese veränderte Ansicht sehen Sie in Abbildung 10.49. Die Verwaltung von Menüs besteht aus folgenden Teilen:

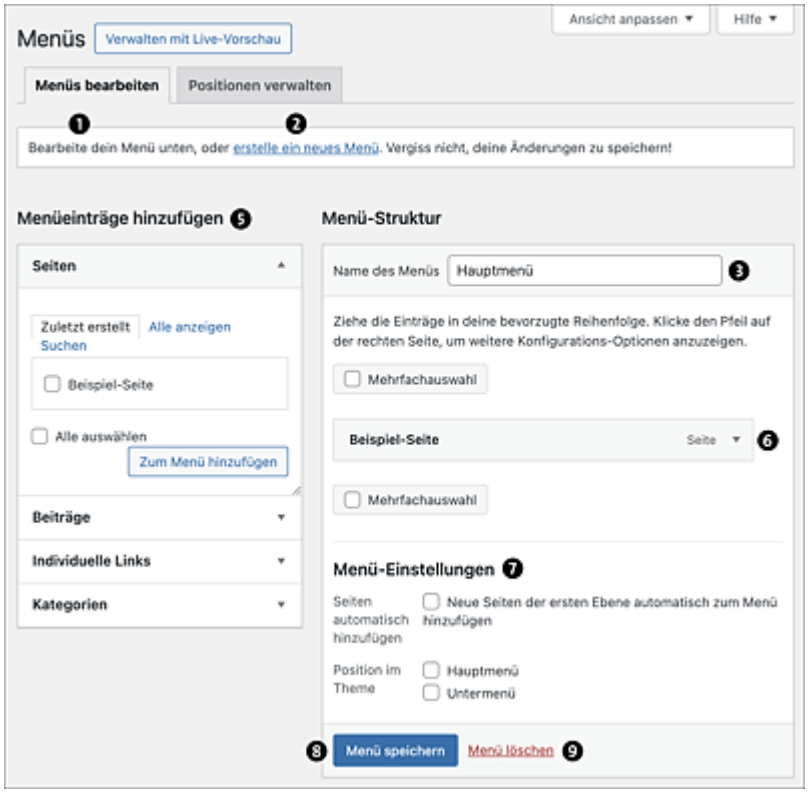


Abbildung 10.49 Diese Ansicht begrüßt Sie unter »Design« • »Menüs«, wenn Sie bereits ein Menü erstellt haben.

1. Wenn ein Menü erstellt wurde, gibt es oben zwei Tabs: Menüs bearbeiten ❶ und Positionen verwalten ❷. Wenn noch kein Menü erstellt ist, ist nur Menüs bearbeiten sichtbar.

   Im Tab Menüs bearbeiten können Sie Menüs erstellen und bearbeiten. Geben Sie dafür einen Namen des Menüs ❸ ein, und klicken Sie auf Menü erstellen ❹.

2. Nachdem Sie ein Menü erstellt haben, können Sie links Menüeinträge hinzufügen ❺. Dieser Bereich ist unterteilt in mehrere Unterbereiche, jeweils für eine andere Art Menüeintrag: Seiten, Beiträge und Kategorien für Inhaltsarten von WordPress sowie Individuelle Links als Möglichkeit, beliebige URLs zum Menü hinzuzufügen. Je nachdem, welche Plugins Sie installiert haben, können hier weitere Bereiche vorhanden sein.

Markieren Sie die gewünschten Inhalte mit der Checkbox, die bei jedem Eintrag vorhanden ist, und klicken Sie auf Zum Menü hinzufügen. Wenn Sie einen individuellen Link einfügen möchten, geben Sie die URL ein und einen für Besucher*innen sichtbaren Link-Text.

3. Die Einträge erscheinen jetzt rechts im Menü und können über das kleine Dreieck oben rechts aufgeklappt werden ❻, um den angezeigten Namen des Menüpunkts zu ändern, der für Besucher*innen im Menü angezeigt wird (das hat keinen Einfluss auf den Titel der Seite oder des Beitrags im Backend).

   Neben der Umbenennung von Menüpunkten können Sie per Drag-and-drop die Menüpunkte umsortieren und auch verschachteln, indem Sie einen Punkt ein wenig nach rechts ziehen. Über einen gestrichelt umrandeten Kasten wird Ihnen angezeigt, wo der Eintrag angeordnet werden würde, wenn Sie ihn loslassen (siehe Abbildung 10.50).



Abbildung 10.50 Per Drag-and-drop können Sie die Menüeinträge umsortieren und auch untereinander verschachteln.

4. In den Menü-Einstellungen ❼ unter den Menü-Einträgen können Sie einstellen, ob Sie Neue Seiten der ersten Ebene automatisch hinzufügen möchten und in welcher Position im Theme das Menü angezeigt werden soll. Die Zuordnung zu einer Position können Sie oben auch über den Reiter Positionen verwalten anpassen.
5. Am Ende müssen Sie das Menü speichern ❽, damit die Änderungen wirksam werden und das Menü dann beispielsweise im Header angezeigt wird (siehe Abbildung 10.51).
6. Wenn mehrere Menüs angelegt sind, finden Sie oben unter den beiden Tabs ein Dropdown, mit dem Sie das Menü auswählen können, das Sie bearbeiten möchten.
7. Wenn Sie ein Menü löschen möchten, finden Sie den entsprechenden Link unten neben dem Button zum Speichern eines Menüs ❾.

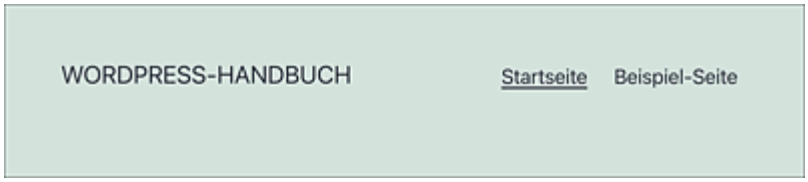


Abbildung 10.51 Das neue Menü als Hauptmenü im Theme Twenty Twenty-One.

10.4.3 Widget-Bereiche füllen

Kommen wir nun zum letzten Standard-WordPress-Element bei Classic Themes, den Widget-Bereichen. Wie bereits kurz erwähnt waren Widgets bis WordPress 5.8 eine eigene Art Inhaltselement, bis sie dann mit WordPress 5.8 von den Blöcken abgelöst wurden, die Sie auch im Editor für Ihre Seiten und Beiträge nutzen.

Widget-Bereiche werden wie Menü-Positionen vom Theme registriert, oft gibt es einen oder mehrere Bereiche im Footer oder ganz klassisch einen in einer Seitenleiste. Hier können Sie dann zum Beispiel einen kurzen About-Text anzeigen oder eine Liste Ihrer Beitragskategorien. Im Admin-Menü finden Sie die Seite zur Verwaltung der Bereiche unter Design • Widgets (siehe Abbildung 10.52).

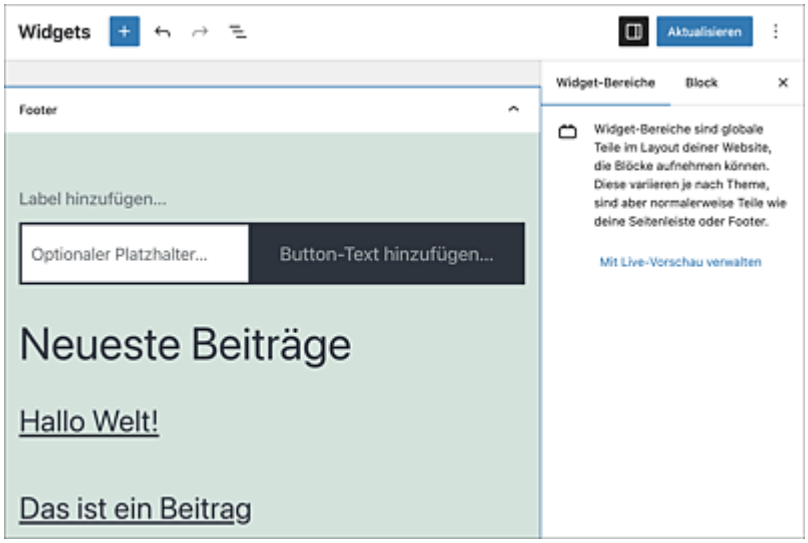


Abbildung 10.52 Unter »Design« • »Widgets« können Sie für Classic Themes die Widget-Bereiche füllen.

Wenn Sie die Seite öffnen, sehen Sie pro Widget-Bereich einen aufklappbaren Bereich, in dem sich eine Instanz des Block-Editors befindet. Twenty Twenty-One hat nur den Footer-Widget-Bereich, und eventuell sind hier bereits ein paar Blöcke eingefügt, wie der Suche-Block oder der Block für die Liste der neuesten Beiträge. Fügen Sie hier die Blöcke ein, die Sie möchten, und bestätigen Sie die Änderungen oben rechts mit Aktualisieren. Wenn Sie jetzt das Frontend aufrufen, sollten Sie den veränderten Widget-Bereich im Footer sehen (siehe Abbildung 10.53).

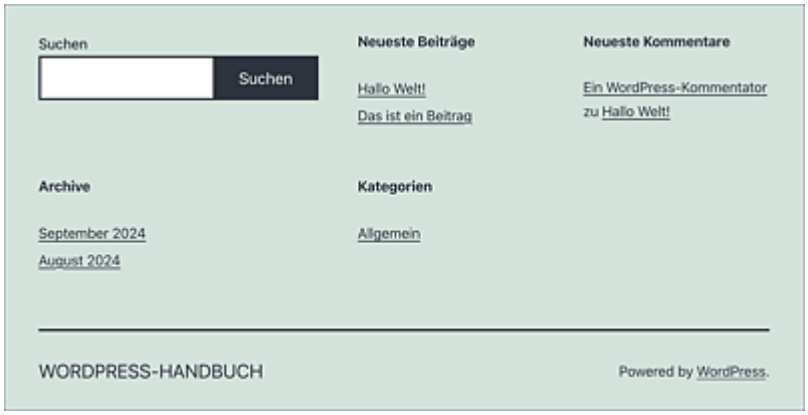


Abbildung 10.53 Der Widget-Bereich von Twenty Twenty-One im Footer

10.5 Neues Theme suchen und installieren

Für einige passt eins der vorinstallierten Standard-Themes von WordPress zur geplanten Website, für andere ist nicht das Richtige dabei – der nächste Schritt ist also die Suche nach einem passenden Theme.

10.5.1 Ein Theme finden

Beginnen Sie mit der Suche nach einem neuen Theme direkt aus WordPress heraus: Im linken Admin-Menü können Sie über Design • Themes oben auf den Punkt Neues Theme hinzufügen klicken und gelangen so zu einer Ansicht, in der Sie die Themes des WordPress.org-Theme-Verzeichnisses durchsuchen können (siehe Abbildung 10.54).

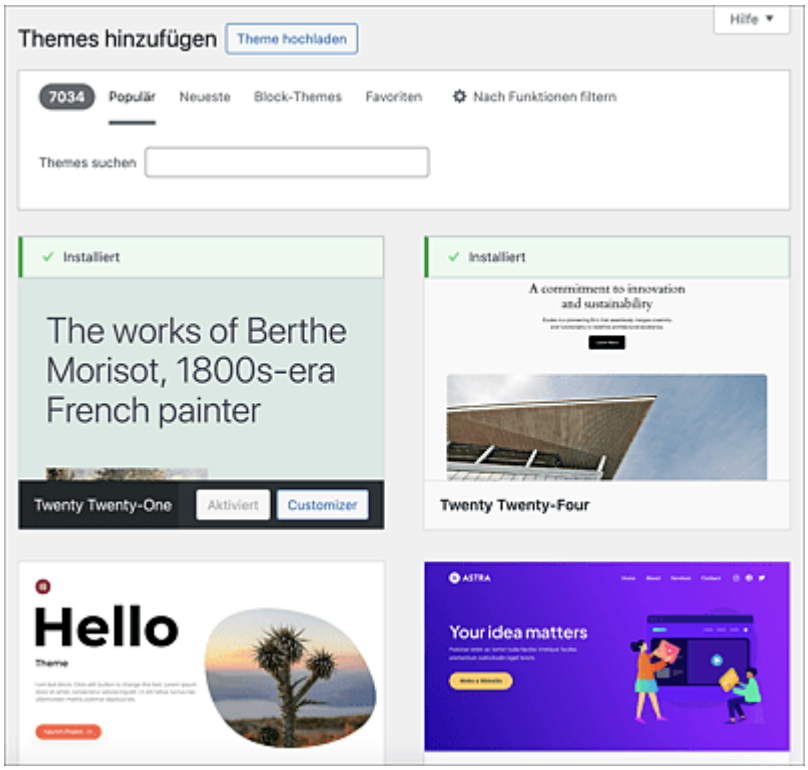


Abbildung 10.54 Die Themes aus dem WordPress.org-Theme-Verzeichnis können Sie direkt aus dem WordPress-Backend heraus durchsuchen. Dafür gehen Sie auf die Seite »Design« • »Themes« und klicken oben auf »Neues Theme hinzufügen«.

Die Ansicht besteht aus unterschiedlichen Tabs:

- Unter Populär finden Sie gerade beliebte Themes.
- Neueste zeigt die Themes-Liste sortiert nach dem Datum an, an dem sie in das Verzeichnis aufgenommen wurden.
- Block-Themes listet nur Block-Themes auf, keine Classic Themes.
- Unter Favoriten können Sie sich die Theme-Favoriten zu einem WordPress.org-Account anzeigen lassen (wenn Sie auf WordPress.org eingeloggt sind, können Sie unter *WordPress.org/themes* Themes als Favoriten kennzeichnen).
- Nach Funktionen filtern erlaubt Ihnen, die Themes nach bestimmten Funktionen zu filtern, die Themes für sich selbst angeben können.

Scrollen Sie einfach ein wenig durch die Themes, und schauen Sie sich die Screenshots an. Wenn ein Theme passend für Sie aussieht, können Sie mit einem Klick auf das Bild eine Vorschau öffnen. Abhängig vom Theme ist diese Demo mehr oder weniger hilfreich. Falls Sie nur eine sehr generische Vorschau vorfinden, können Sie das Theme unter *de.wordpress.org/themes* aufrufen, um die Theme-Homepage herauszufinden (siehe Abbildung 10.55). Mit etwas Glück gibt es dort eine bessere Demo.

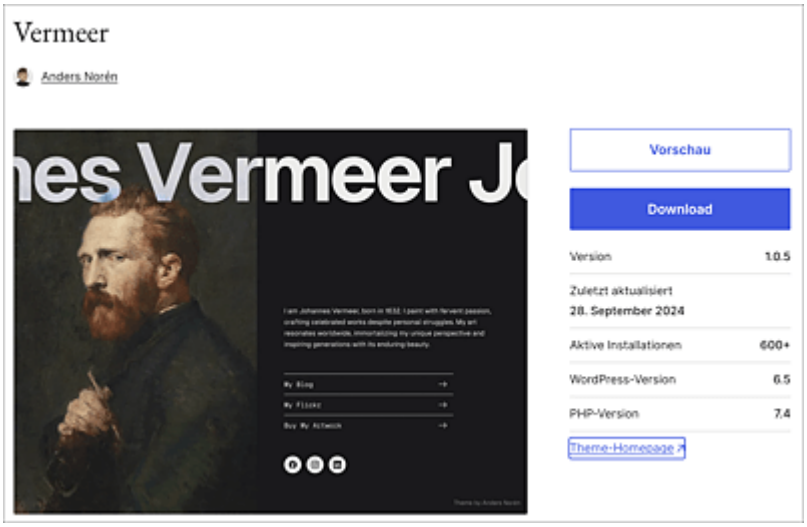


Abbildung 10.55 Über den Link »Theme-Homepage« gelangen Sie zu der Seite, die von den Theme-Autor*innen als Theme-Homepage hinterlegt wurde. Oft ist dort eine Demo-Site verlinkt, die eine bessere Vorschau bietet als die Vorschaufunktion des Theme-Verzeichnisses.

Die Zahl der Themes im WordPress.org-Verzeichnis ist groß, und wie überall, wo Sie eine große Zahl Themes von vielen Herstellern zusammengefasst finden, ist die Qualität der Themes unterschiedlich. Es gibt sowohl sehr tolle Themes (optisch wie auch technisch) als auch welche, die eher nicht so super aussehen und/oder technisch zu wünschen übrig lassen. Klicken Sie sich einfach ein bisschen durch das Angebot, und nutzen Sie die Filter der Suche, um nach bestimmten Eigenschaften zu filtern.

Werden Sie hier nicht fündig, können Sie zum Beispiel nach Theme-Listen suchen, zum Beispiel nach den besten WordPress-Themes für Fotograf*innen (suchen Sie auf Englisch, das ist erfolgversprechender). Auch hier ist es Glücksache, ob Sie ein Theme finden, das Ihnen erstens gefällt und das zweitens gut ist. Eine weitere Möglichkeit besteht darin, bei Websites, die Ihnen gefallen, zu schauen, ob sie zufällig ein WordPress-Theme nutzen, das entweder kostenlos oder kostenpflichtig heruntergeladen werden kann. Um das zu testen, können Sie die URL der Website auf

whatwpthemeisthat.com eingeben und bekommen, wenn es sich um eine WordPress-Website handelt, die Informationen zum verwendeten Theme ausgegeben.

Bei Ihrer Recherche werden Sie vermutlich auch über Marktplätze wie Envato stolpern, auf denen kostenpflichtige Themes angeboten werden, die sich gegenseitig mit der Menge an verfügbaren Einstellungsmöglichkeiten, Zusatzfunktionen und Layoutoptionen übertrumpfen und am besten noch einen Page-Builder mitbringen. **Hier ist Vorsicht geboten.** Auf den ersten Blick sind solche Themes, die quasi Funktionen für alles mitbringen, was man sich nur wünschen kann, sehr verlockend (sie werden auch »Multipurpose-Themes« genannt). Machen Sie aber nicht den Fehler, dieser Verlockung nachzugeben, sondern machen Sie einen großen Bogen um solche Themes, um sich später Ärger zu ersparen, wenn Sie das Theme einmal nicht mehr nutzen und zu einem anderen wechseln möchten.

Achtung: Vorsicht vor Multipurpose-Themes

Bei der Recherche nach Themes werden Sie vermutlich früher oder später auf Kandidaten wie Avada treffen, das sich auf der Plattform Themeforest mit »Website Builder For WordPress & eCommerce« beschreibt und als Funktionen unter anderem mit dem »Avada Website Builder« einen Page-Builder auflistet und auch noch eigene Inhaltstypen mitbringt, etwa um ein Portfolio zu erstellen. Auf den ersten Blick klingt das nach: »Wow, mit dem Theme kann ich alles umsetzen, was ich möchte, und brauche keine Plugins für weitere Funktionen.« Und das mag stimmen. Aber wenn Sie ein bisschen in die Zukunft denken und davon ausgehen, dass Sie nicht für immer dasselbe Theme nutzen werden (oder Avada irgendwann nicht weiterentwickelt wird), dann

stehen Sie vor einem Problem: Wenn Sie Avada deaktivieren, verlieren Sie Inhalte wie die Portfolioeinträge, und Sie verlieren die Layouts, die Sie mit dem Page-Builder gebaut haben.

Das bedeutet, Sie müssten je nach Umfang Ihrer Website einen enormen Aufwand betreiben, um das Theme zu wechseln, da Sie alle Page-Builder-Inhalte von den Shortcodes befreien und die Layouts mit anderen Lösungen nachbauen müssen, und Inhalte wie die Portfolioeinträge dürfen Sie komplett neu anlegen.

Um dieses Problem zu umgehen, sollten Sie sich für ein Theme entscheiden, das sich nur um das Design kümmert, nicht darum, möglichst viele Funktionen mitzubringen, am besten noch welche, mit denen sich Inhalte erstellen lassen (die dann bei einem Theme-Wechsel weg sind). Funktionserweiterungen, die sich um die Erstellung von Inhalten kümmern, sollten Sie nur über Plugins installieren.

Tipp: Worauf Sie bei der Theme-Auswahl achten sollten

Theme-Demos sehen in der Regel gut aus. Das muss aber nicht heißen, dass Ihre Website mit Ihren Inhalten in dem Theme auch gut aussieht. Wenn eine Theme-Demo zum Beispiel sehr viele Bilder präsentiert, Sie aber kaum Bilder auf Ihrer Website nutzen, kann der Eindruck vom Theme auf Ihrer Website dann ein ganz anderer sein als in der Demo. Sie sollten also versuchen, Ihre Inhalte in das Theme hineinzudenken.

Was sich aber ganz gut an einer Demo testen lässt, ist, ob das Theme performant ist. Wenn die Demo schon mehrere Sekunden zum Laden braucht, suchen Sie nach einem anderen Theme. Sie können die Theme-Demo zur Performanceanalyse auch durch Tools wie PageSpeed Insights (*http://r-*

wrk.de/wp6_pagespeedinsights) oder den Pingdom-Website-Speed-Test (*http://r-wrk.de/wp6_pingdom*) laufen lassen, um einen Eindruck von Metriken zu bekommen, die wichtig für die Performance sind (zum Beispiel Anzahl und Größe der geladenen Dateien).

Ein weiterer diesmal einfacherer Test an der Theme-Demo ist, wie sich das Theme in unterschiedlichen Viewport-Größen verhält. Rufen Sie die Theme-Demo mal auf Ihrem Smartphone auf oder verkleinern Sie einfach das Browserfenster. Sieht es auch auf kleineren Größen gut aus, oder wurde nur viel Wert auf die größeren Viewports gelegt?

Zudem sollten Sie bei Bedarf ermitteln, ob das Theme die Bearbeitung von Designaspekten erlaubt, die Sie gerne ändern würden (wenn es sich um ein Block-Theme handelt, haben Sie hier natürlich bessere Chancen).

Ein weiterer wichtiger Punkt ist, ob das Theme in der Sprache Ihrer Website verfügbar ist und, wenn nicht, ob es für die Übersetzung vorbereitet ist, sodass Sie zur Not die Übersetzung selbst vornehmen können.

Darüber hinaus – es wurde schon oben im Kasten zu Multipurpose-Themes erwähnt – noch mal zur Sicherheit: Halten Sie großen Abstand von Themes, die Funktionen zur Inhaltserstellung mitbringen, seien es Slider oder gleich komplette Page-Builder. Damit handeln Sie sich Probleme ein, die Sie spätestens ereilen, wenn Sie das Theme einmal wechseln möchten und die Inhalte dann weg sind, die über die Theme-Funktionen erstellt wurden.

10.5.2 Theme-Installation

Wenn Sie ein passendes Theme gefunden haben, entweder im Theme-Verzeichnis von WordPress oder über eine andere Quelle, geht es an die Installation.

Tipp: Backup nicht vergessen und erst auf einem Testsystem testen

Auch wenn oft alles gut geht, ein Backup vor einer größeren Umstellung oder Änderung, wie dem Wechsel eines Themes, schadet nie. Und testen Sie am besten zuerst auf einem eigenen Testsystem, nicht auf der Live-Website.

Ein Theme aus dem Theme-Verzeichnis installieren

Falls Ihr Wunsch-Theme im WordPress-Verzeichnis ist, können Sie es einfach direkt über das Backend installieren:

1. Gehen Sie im linken Admin-Menü auf den Punkt Design • Themes und auf der Seite oben auf den Punkt Neues Theme hinzufügen.
2. Suchen Sie am besten über die Suchfunktion nach dem Theme.
3. Haben Sie das Theme in der Liste vor sich, bewegen Sie den Cursor über die Kachel, und klicken Sie auf Jetzt installieren (siehe Abbildung 10.56).

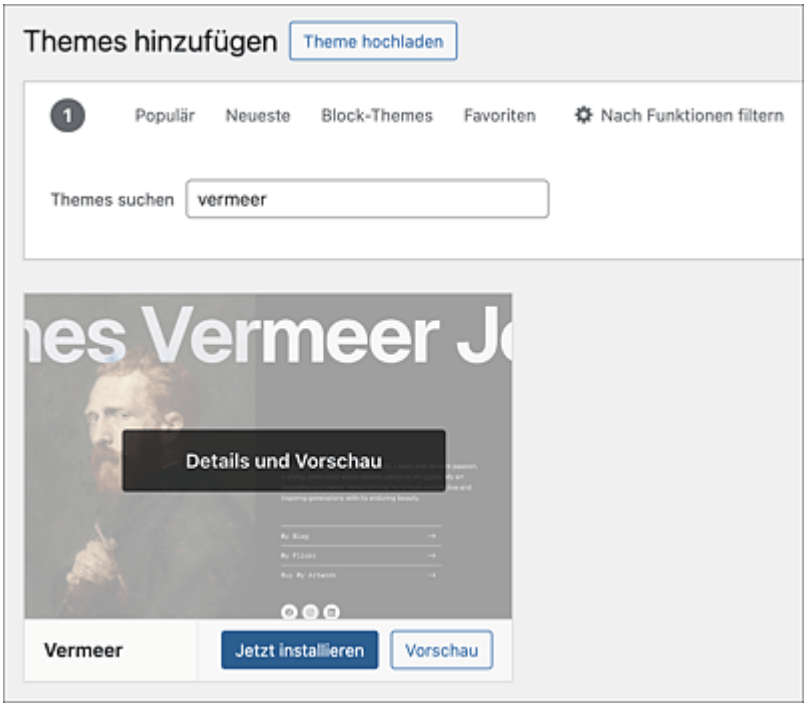


Abbildung 10.56 Ein Theme aus dem WordPress.org-Theme-Verzeichnis kann einfach direkt über das WordPress-Backend installiert werden.

4. Nach einigen Sekunden bestätigt Ihnen ein INSTALLIERT auf einem grünen Balken am oberen Rand des Theme-Screenshots, dass die Installation abgeschlossen wurde. Mit einem Klick auf AKTIVIEREN können Sie das Theme direkt aktivieren oder in der Liste der installierten Themes über den Button LIVE-VORSCHAU Anpassungen im Customizer (für Classic Themes) oder Website-Editor (für Block-Themes) vornehmen, bevor die Website mit dem neuen Theme online geht (siehe Abbildung 10.57).

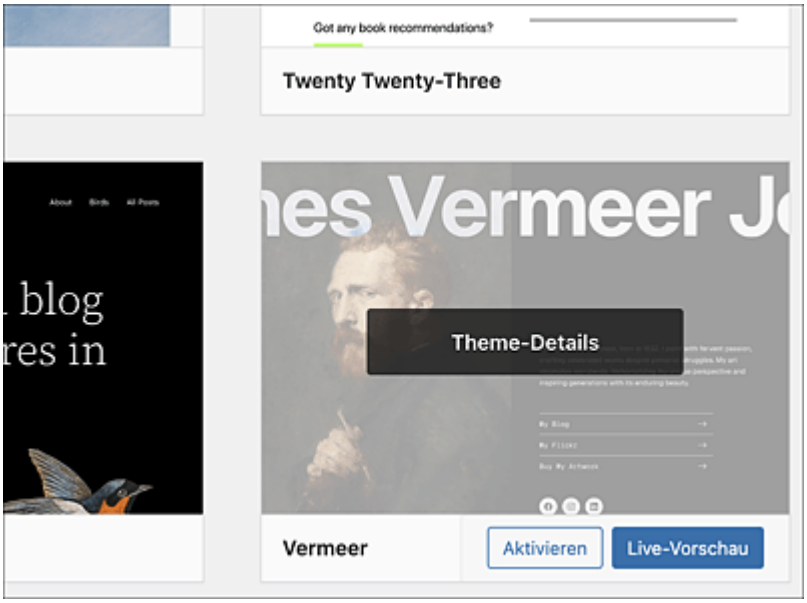


Abbildung 10.57 Nachdem das Theme installiert wurde, können Sie es direkt aktivieren oder erst mal in der Live-Vorschau anschauen und gegebenenfalls Änderungen vornehmen, bevor Sie es aktivieren.

Das bisher aktive Theme bleibt natürlich installiert, und Sie können jederzeit dahin zurückwechseln, indem Sie es unter DESIGN • THEMES auswählen.

Ein Theme aus einer anderen Quelle installieren

Themes, die nicht aus dem WordPress.org-Verzeichnis stammen, können Sie ebenfalls recht einfach installieren, indem Sie ein ZIP-Archiv mit dem Theme im Backend hochladen. Dieses Archiv muss Ihnen von der Quelle des Themes, etwa nach dem Kauf, bereitgestellt werden.

1. Wenn Sie das ZIP heruntergeladen haben, entpacken Sie es zur Sicherheit einmal, um zu schauen, ob es sich dabei direkt um das Theme handelt oder ob vielleicht das Theme-ZIP mit weiteren Dingen wie einer Dokumentation darin liegt. Die Theme-Dateien erkennen Sie an einer *style.css*.

2. Haben Sie das richtige ZIP zur Hand, gehen Sie im Backend zur Seite Design • Themes und klicken oben auf Neues Theme hinzufügen.
3. Auf der sich öffnenden Seite mit den Themes aus dem WordPress-Theme-Verzeichnis sehen Sie oben den Button Theme hochladen.
4. Nach einem Klick öffnet sich ein Bereich mit einem Datei-Feld, über das Sie das ZIP-Archiv hochladen können (siehe Abbildung 10.58).

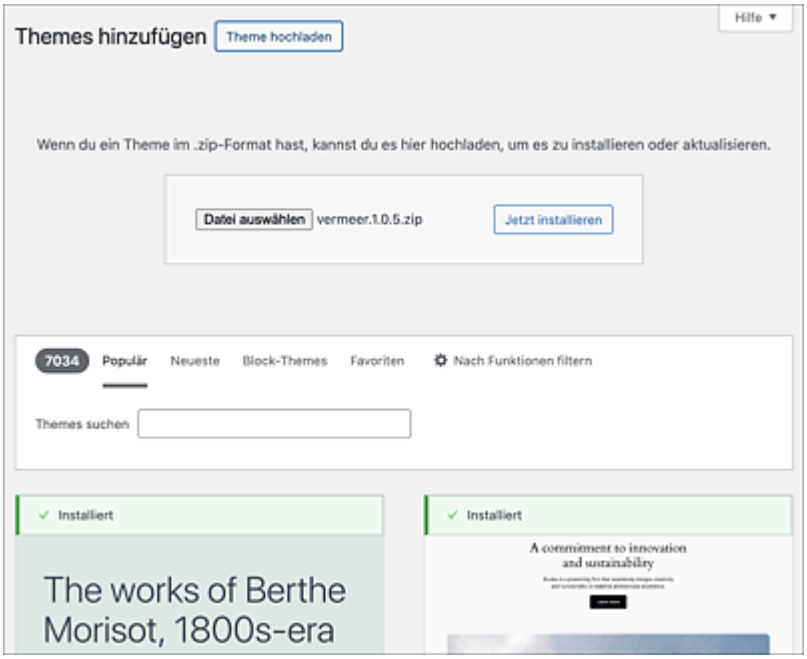


Abbildung 10.58 Ein Theme aus einer anderen Quelle als dem WordPress-Verzeichnis können Sie als ZIP hochladen.

5. Nach einem Klick auf Jetzt installieren wird das Theme hochgeladen und installiert, was je nach Größe des Archivs und Internetgeschwindigkeit ein bisschen dauern kann. Während der Installation wird Ihnen der aktuelle Status angezeigt und am Ende mit Das Theme wurde erfolgreich installiert die Installation bestätigt (siehe Abbildung 10.59).

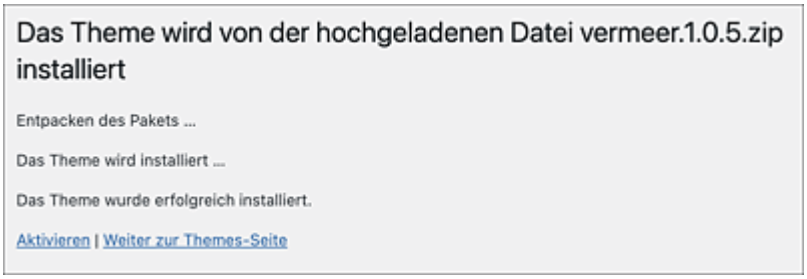


Abbildung 10.59 Bei der Installation durch Hochladen eines ZIP-Archivs werden Ihnen die durchgeführten Schritte angezeigt.

10.6 Barrierefreiheit von Themes testen

Das Thema Barrierefreiheit haben viele immer noch nicht auf dem Schirm oder finden es unwichtig. Wenn Sie aber nicht darauf achten, dass Ihre Website möglichst barrierefrei oder barrierearm ist, kann das schnell dazu führen, dass Menschen von der Nutzung Ihrer Website ausgeschlossen werden. Beim Thema Barrierefreiheit gibt es verschiedene Aspekte zu berücksichtigen, die nicht alle für jede Website zutreffen und von denen einige einfacher und andere schwieriger zu überprüfen sind. Eine umfangreiche Liste mit Punkten und Beschreibungen finden Sie unter *http://r-wrk.de/wp6_barrierefrei*. Einige Punkte, die Sie recht einfach testen können (beispielsweise in der Demo des Themes, das in Ihrer näheren Auswahl ist), sind:

10.6.1 Tastaturbedienbarkeit

Öffnen Sie eine Website, die das Theme aktiv hat, und nutzen Sie die ⇆-Taste, um die fokussierbaren Elemente der Website durchzugehen. Dazu gehören alle Links, Buttons und Formularelemente. Auf diese Dinge sollten Sie achten:

- Ist das aktuell fokussierte Element visuell so hervorgehoben, dass Sie es klar identifizieren können? Das bedeutet beispielsweise, dass es nicht nur durch einen Wechsel der Farbe hervorgehoben ist, sondern etwa durch einen Rahmen oder eine Unterstreichung.
- Lassen sich nur interaktive Elemente ansteuern, die wirklich etwas tun (also beispielsweise keine inaktiven Buttons, die aus irgendwelchen Gründen gerade gar keine Funktion haben)?

- Wenn Sie ein Menü mit mehreren Ebenen nutzen möchten, probieren Sie, ob sich Untermenüs mit der Tastatur bedienen lassen. Das ist ein Aspekt, der oft zu beobachten ist: Entweder lassen sich die Untermenüpunkte gar nicht erreichen, oder sie werden zwar fokussiert, aber das Untermenü öffnet sich nicht. Auch ist es nicht optimal, wenn die Untermenüpunkte automatisch fokussiert und durchgegangen werden. Das Standardverhalten sollte sein, dass nur die Menüpunkte der obersten Ebene durchgegangen werden und bei einem Menüpunkt mit Unterpunkten mit einem Extra-Button (oder ähnlichen Lösungen) die Möglichkeit besteht, das Untermenü anzuzeigen.
- Entspricht die Fokusreihenfolge der visuellen Reihenfolge der Elemente? Ohne Anpassungen im Code werden die fokussierbaren Elemente so durchgegangen, wie sie im Quellcode stehen, nicht, wie sie auf der Website dargestellt werden. Über CSS lassen sich aber schnell Änderungen umsetzen, die die visuelle Reihenfolge so ändern, dass erst ein weiter unten auf der Seite angezeigtes Element fokussiert wird, bevor eins darüber den Fokus erhält. Die Firefox-Devtools kommen mit einer Funktion, um die Tab-Reihenfolge anzuzeigen (siehe Abbildung 10.60). Erreichbar sind diese Entwicklerwerkzeuge über einen Rechtsklick in die Website und den Punkt Untersuchen. Oder Sie können über den Punkt Barrierefreiheit-Eigenschaften untersuchen direkt in den Barrierefreiheit-Tab der Entwicklerwerkzeuge springen.

Neben der Tab-Reihenfolge können Sie auch weitere Punkte, wie die Fokussierbarkeit inaktiver Elemente oder die mangelnde Hervorhebung fokussierter Elemente, in den Barrierefreiheit-Tools von Firefox überprüfen, indem Sie aus der Auswahlliste Überprüfung auf Probleme den Wert Tastatur wählen.

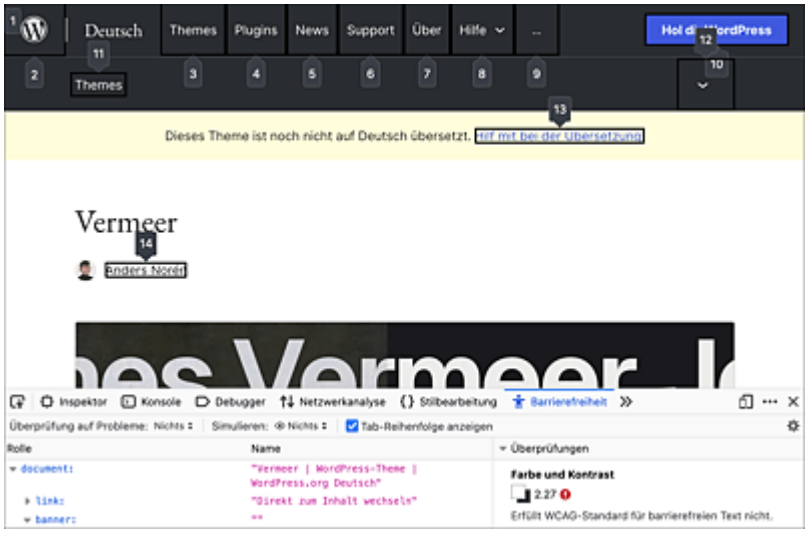


Abbildung 10.60 In Firefox kann mit den Webentwicklertools die Tab-Reihenfolge angezeigt werden, sodass einfach geprüft werden kann, ob sie sinnvoll ist.

10.6.2 Farbkontraste

Es gibt in den *Web Content Accessibility Guidelines* (WCAG) klare Angaben dazu, wie der Kontrast zwischen Textfarbe und Hintergrundfarbe sein muss, um die unterschiedlichen Level (aufsteigend nach Anforderungen: A, AA und AAA) der WCAG zu erreichen. Um Level AA zu erreichen, muss der Kontrast von Text zu Hintergrund bei 4,5 : 1 liegen, es sei denn, es ist rein dekorativer Text oder Teil eines Logos, dort gibt es keine Vorgaben. Bei großem Text muss der Kontrast mindestens 3:1 sein.

Es gibt unterschiedliche Werkzeuge, um die Farbkontraste auf einer Webseite zu überprüfen. Der Browser Firefox hat so ein Werkzeug beispielsweise direkt an Bord (siehe Abbildung 10.61), und es gibt mit *axe* (*http://r-wrk.de/wp6_axe*) ein nützliches Tool, das Browsererweiterungen etwa für Chrome und Firefox anbietet (das für Firefox wird auf der Website nicht erwähnt, lässt sich mit einer schnellen Suche aber finden). Einige dieser Tools können auch Farbfehlsichtigkeiten simulieren.



Abbildung 10.61 Über das Barrierefreiheit-Werkzeug in den Firefox-Entwicklertools lassen sich Farbkontraste auf einer Website überprüfen.

10.6.3 Lesbarkeit von Texten

Sehr beliebt waren und sind teilweise immer noch sehr dünne Schriftschnitte. Die mögen auf hochauflösenden Displays gut aussehen, auf Displays mit normaler Auflösung können dann aber selbst für gut sehende Menschen Probleme bei der Lesbarkeit entstehen. Generell sollten sie besser vermieden oder nur für sehr große Texte eingesetzt werden, auf keinen Fall für Fließtexte.

Auch andere typografische Entscheidungen können Probleme machen, etwa lange Texte in kursiv gesetztem Text, was die Lesbarkeit erschwert.

10.6.4 Labels für Formularfelder

Viele Themes haben zumindest irgendwo ein Suchformular, manchmal auch weitere Formulare, und manchmal wird als einzige Beschriftung für Formularfelder der Placeholder-Text genutzt (ein meist etwas ausgegrauter Text, der verschwindet, sobald etwas in das Feld eingegeben wird). Das ist für alle suboptimal, denn sobald etwas in dem Feld steht, müsste der Inhalt wieder entfernt werden, um die Beschriftung zu sehen.

Zur richtigen Beschriftung von Formularfeldern gibt es das `label`-Element, das über das `for`-Attribut mit dem Formularfeld verknüpft wird, indem es den Wert des `id`-Attributs des Formularfelds übergeben bekommt, wie in Listing 10.1 beispielhaft dargestellt. Ob diese Verknüpfung funktioniert, ist daran zu erkennen, dass nach einem Klick auf die Beschriftung das Formularfeld ausgewählt ist oder dass bei einer Checkbox auch der Klick auf die Beschriftung das Häkchen setzt.

```
<label for="search">Suche</label>
<input type="text" id="search" name="s">
```

Listing 10.1 Die Beschriftung »Suche« ist über das »for«-Attribut mit dem Textfeld verbunden, da der Wert des for-Attributs der »id« des Textfelds entspricht.

11 Neue Funktionen mit Plugins hinzufügen

> *WordPress selbst – der WordPress-Core – ist recht schlank gehalten. Ein entscheidender Teil des Erfolgs des CMS liegt aber in seiner Erweiterbarkeit. Seit vielen Jahren kann WordPress daher mithilfe sogenannter Plugins im Funktionsumfang erweitert werden.*

Plugins sind kleine Software-Pakete, die mehr oder weniger vollständige Funktionen mitbringen und damit Ihre Website erweitern können. Funktionen wie Kalender, Gästebücher, Kontaktformulare oder ganze Shop-Systeme (in Kapitel 12 setzen Sie mit dem Plugin WooCommerce einen Online-Shop auf) können von Plugins ebenso bereitgestellt werden wie Ergänzungen zu anderen Plugins oder Funktionen von *WordPress-Core*.

Der reinen WordPress-Lehre folgend kümmern sich Plugins ausschließlich um die funktionale Erweiterung einer WordPress-Website. Themes (mehr zu Themes erfahren Sie in Kapitel 10) sollen sich dieser Lehre folgend gänzlich auf die Darstellung einer Website fokussieren. In der Praxis wird dieser Grundsatz gelegentlich gebrochen, grob sollten sich alle Plugins und Themes, die Sie auf Ihrer WordPress-Website einsetzen, aber an diesen Grundsatz halten.

Der WordPress-Core ist durch unzählige Schnittstellen dafür vorbereitet, durch Plugins erweitert zu werden. Aber auch große Teile der Community-Infrastruktur zielen darauf ab, Ihnen den Umgang mit Plugins möglichst einfach zu machen. Auf *de.wordpress.org/plugins* finden Sie einen riesigen Katalog von knapp 60.000 kostenfreien WordPress-Erweiterungen, die Sie mit wenigen Klicks auf Ihrer Website installieren können.

In diesem Kapitel möchten wir Ihnen eine Einführung in die wunderbare Welt der Plugins geben und Sie bei den ersten Schritten durch diese Welt begleiten.

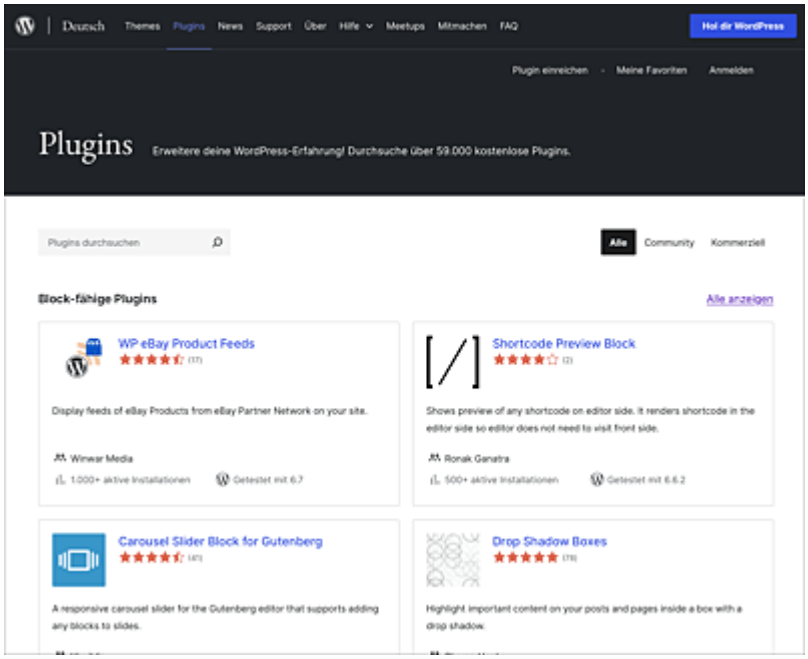


Abbildung 11.1 Im Plugin-Verzeichnis auf WordPress.org finden Sie knapp 60.000 kostenfreie Plugins.

11.1 Plugins testen – sicher

Mit Admin-Rechten für eine WordPress-Website ausgestattet können Sie in der Regel ohne große Hürden Plugins installieren. Doch bevor wir das im nächsten Abschnitt tun, sollten wir uns

einen Moment Zeit nehmen und über die Konsequenzen wilder Plugin-Installationen sprechen.

Abseits von möglichen Sicherheitslücken oder Schadcode in Plugins geht die größte unmittelbare Gefahr für Ihre Website von Datenmüll aus, den einmal installierten Plugins oft auch nach der Deinstallation noch hinterlassen. Deshalb ist es nicht unbedingt empfehlenswert, einfach auf gut Glück Plugins auf Ihrer produktiven WordPress-Website zu installieren.

Stattdessen sollten Sie, wenn Ihr Webhoster dies anbietet, eine gesonderte *Staging*-Umgebung nutzen, in der sie bedenkenlos Plugins testen können, ohne sich wirklich um Konsequenzen Gedanken machen zu müssen. Wenn Staging für Ihr Hosting keine Option ist, können Sie sich auch mit einer lokalen Testumgebung behelfen. In den Kapitel 2 und Kapitel 3 lernen Sie unterschiedliche Möglichkeiten kennen, eine lokale Installation aufzusetzen.

Beim Testen gilt es, vor allem zwei Aspekte genauer zu berücksichtigen:

1. Erfüllt ein Plugin Ihre konkreten Anforderungen?

   Mit beinahe 60.000 Plugins im offiziellen Verzeichnis auf WordPress.org und unzähligen weiteren in kostenpflichtigen Shops verschiedener Plugin-Entwickler*innen wird die Wahl schnell zur Qual.

   Bei der Auswahl sollten Sie sich im ersten Schritt über Ihre Anforderungen klar werden und damit schon einmal Plugins vorauswählen, bevor Sie an das genauere Testen gehen. Mit dem Definieren und Prüfen von Anforderungen beschäftigten wir uns in Abschnitt 11.2.

2. Verträgt sich das Plugin, das Sie ausgewählt haben, mit Ihrem bestehenden Setup?

   Wenn Sie Ihre Auswahl dann auf einen oder einige wenige potenzielle Kandidaten eingegrenzt haben, sollten Sie abschließend überprüfen, ob diese in Ihrem aktuellen Setup funktionieren.

   Hier können verschiedene Faktoren eine Rolle spielen: Ihre WordPress-Version, Ihre PHP-Version, andere installierte Plugins und in Ausnahmefällen das eingesetzte Theme.

Tipp: Wenn es doch einmal klemmt …

Wenn Sie sich auf einer produktiven Website an einem neuen Plugin versuchen und der Versuch in einem Fehler endet, der Ihre Site unerreichbar macht, haben wir in Kapitel 21, »Notfallmaßnahmen«, einige hilfreiche Tipps für Sie gesammelt.

11.2 Von der Anforderung zum Plugin

Gehen wir noch einmal einen Schritt zurück, und widmen wir uns der Frage nach den Anforderungen, die Sie an ein Plugin stellen. Lassen Sie uns als Beispiel einmal die Suche nach einem Kontaktformular-Plugin durchspielen. Was muss so ein Plugin können?

Laufen Sie nicht in die Falle, diese Frage einfach nur mit »Kontaktformulare einbinden« zu beantworten. Der Teufel steckt im Detail.

Sollen Daten aus Ihrem Kontaktformular zum Beispiel nur per E-Mail versendet werden oder auch in WordPress direkt aufrufbar sein? Legen Sie Wert auf einen ausgefeilten Formular-Builder, um das Kontaktformular zu bauen? Oder genügt es Ihnen, ein vorgefertigtes Template mit etwas HTML anzupassen?

Sollen Ihre Besucher*innen im Kontaktformular Dateien hochladen können? Und sind Sie bereit, für Ihr Kontaktformular-Plugin Geld zu bezahlen, wenn es alle Ihre Anforderungen erfüllt?

Abhängig davon, welche dieser Fragen Sie bejahen, bieten sich komplett unterschiedliche Kandidaten aus dem breiten Sortiment der Kontaktformular-Plugins an. Es lohnt sich also, wenn Sie zunächst etwas Zeit investieren, um Ihre Anforderungen zu sortieren und sich ein Bild davon zu machen, wie ein Plugin aussehen muss, um für Sie und Ihren Anwendungsfall wirklich nützlich zu sein.

11.3 Gute Plugins erkennen

Nachdem Sie nun wissen, was ein Plugin für Ihren Anwendungsfall können muss, stellt sich die Frage, welche Plugins tatsächlich gut sind.

Dafür haben wir für Sie eine kleine Checkliste zusammengestellt, die Sie anschließend in Abschnitt 11.3.2 finden. Widmen wir uns zunächst aber noch kurz den verschiedenen Quellen, aus denen Sie Plugins beziehen können, und solchen, aus denen Sie sie vielleicht nicht beziehen sollten.

11.3.1 Quellen für Plugins

Die einfachsten WordPress-Plugins bestehen aus einer einzigen PHP-Datei. Die meisten Plugins setzen sich aber aus mehreren Dateien zusammen, die in verschiedenen Verzeichnissen organisiert sind. Sie können von wenigen Kilobyte bis zu mehreren Megabyte Dateigröße variieren. Was sie alle gemeinsam haben, ist, dass Sie sie in der Regel als ZIP-Datei herunterladen und entweder direkt über das WP-Admin-Interface in WordPress installieren oder per (S)FTP auf Ihren Server legen können. Wie die Installation Schritt für Schritt abläuft, lernen Sie in Abschnitt 11.4.

Die verschiedenen Quellen, von denen Sie Plugins beziehen können, lassen sich grob in fünf Kategorien einteilen. Deren verschiedene Vor- und Nachteile werden wir nun einmal näher betrachten.

WordPress.org

Das offizielle Plugin-Verzeichnis auf WordPress.org enthält knapp 60.000 kostenfreie Plugins.

Theoretisch kann jede*r eigene Plugins über das Verzeichnis von WordPress.org anbieten. Bevor diese Einreichungen aber veröffentlicht werden, werden sie von einem Team Freiwilliger überprüft. Alle hier angebotenen Plugins müssen die Regeln und Konventionen von WordPress.org einhalten. Auch nach dem Eintritt ins offizielle Verzeichnis können Plugins, die auffällig werden oder für die Sicherheitsmängel bekannt werden, aus dem Katalog auch wieder entfernt werden.

Das WordPress.org-Verzeichnis ist so aufgebaut, dass jedes Plugin eine eigene Seite hat, auf der Plugin-Entwickler*innen neben eigenem Erklärtext mit einem Logo und einer größeren Grafik auch etwas Design einbringen können. Auf diesen Plugin-Seiten finden Sie auch Bewertungen von anderen Benutzer*innen und Zugang zum Support-Forum des jeweiligen Plugins. Die Entwickler*innen sind jedoch nicht verpflichtet, das Support-Forum eines Plugins auch wirklich aktiv zu betreuen.

Alle Plugins von WordPress.org können kostenfrei installiert werden.

Shops der Plugin-Anbieter

Entwickler*innen, die sich nicht auf das Anbieten von kostenfreien Plugins über WordPress.org beschränken wollen, richten sich oftmals einfach selbst einen Shop auf ihrer eigenen Website ein. Auf ihren eigenen Websites haben diese Anbieter dann alle nur denkbaren Freiheiten. Bis auf einige Lizenz-Bestimmungen, die von WordPress vorgegeben werden, gibt es keine Vorgaben oder Richtlinien, an die Entwickler*innen sich halten müssen.

Das ist nicht zwingend ein Nachteil, bedeutet aber natürlich deutlich weniger Kontrolle durch eine zentrale Stelle.

Info: Sind kostenpflichtige Plugins besser?

Diese Frage kommt in Gesprächen rund um kostenfreie und kostenpflichtige Plugins schnell auf. Und eine wirklich eindeutige Antwort gibt es darauf nicht. Sie werden auf WordPress.org viele qualitativ hochwertige Plugins finden und mindestens genauso viele schlecht gepflegte oder schlampig programmierte.

Im großen Angebot der kostenpflichtigen Plugins auf der anderen Seite finden Sie ebenfalls viele hochwertige Plugins. Aber Ihnen werden auch Plugins über den Weg laufen, die sich über Wert verkauft haben, eher auf Marketing als auf echte Substanz setzen und qualitativ mit einigen kostenfreien Plugins auf WordPress.org nicht mithalten können.

Behalten Sie das stets im Hinterkopf, und vergleichen Sie die verschiedenen Angebote.

Plugin-Marktplätze

Viele Entwickler*innen scheuen – zu Recht – den Betrieb eines eigenen Shops. Vor allem kleinere Anbieter*innen oder Einzelkämpfer*innen suchen sich daher gelegentlich Marktplätze wie CodeCanyon oder WooCommerce.com, um ihre Plugins zu verkaufen und gegebenenfalls Support anzubieten.

Die Regeln und Vorgaben, die diese Plattformen durchsetzen, können sich sehr stark unterscheiden.

GitHub

GitHub dient vor allem als Entwicklungsplattform für Entwickler*innen. Weil viele Anbieter an ihren Plugins hier aber im

Sinne von Open Source öffentlich arbeiten, können Sie GitHub oft auch als Quelle für Plugins nutzen.

»Nulled«

Ob Sie ein Plugin von WordPress.org, direkt vom Anbieter, über eine Plattform oder über GitHub beziehen, ist Ihnen überlassen. Ob Sie eher kostenfreie oder kostenpflichtige Plugins einsetzen wollen, hängt von Ihrem Budget und Ihren Anforderungen ab.

Der eine Grundsatz, an den Sie sich halten sollten, ist der, Plugins nur von deren tatsächlichen Anbieter*innen zu beziehen.

Wegen der Lizenzbestimmungen von WordPress gibt es einen gewissen Graubereich, in dem Dritte die Plugins (und Themes) von anderen weiterverbreiten können. Das ist grundsätzlich legal. Mindestens unter moralischen Aspekten ist die Verwendung dieser Piraten-Websites mindestens fragwürdig.

Zu allem Überfluss sind Plugins und Themes, die Sie von sogenannten »Nulled«-Websites bekommen, aber auch oft noch mit Schadcode versetzt und können damit Ihre gesamte Website in Gefahr bringen.

11.3.2 Checkliste zur Plugin-Auswahl

Mit dieser Vielzahl von verschiedenen Plugin-Quellen, Entwickler*innen und verschiedenen Feature-Sets sind viele WordPress-Anwender*innen bei der Auswahl eines neuen Plugins überfordert.

Wir werden uns an dieser Stelle nicht in einer Doktorarbeit zur Qualitätskontrolle von WordPress-Plugins verlieren. Im Kontext eines größeren Unternehmens würden Sie vor dem Einsatz eines

Plugins vermutlich eine komplette Überprüfung des Programmcodes eines Plugins durchführen (lassen). Für kleinere Projekte oder im persönlichen Kontext behelfen wir uns derweil eher damit, den Eindruck eines Plugins zu interpretieren, um auf dessen Qualität zu schließen.

Dafür haben wir im Folgenden einige Aspekte gesammelt, die Sie in Kombination zu einem Gesamtbild zur Hand nehmen können, um eine erste Einschätzung zur Plugin-Auswahl treffen zu können.

Wichtig ist aber, dass die folgenden Kriterien in den seltensten Fällen für sich genommen als Entscheidungsfaktoren genutzt werden sollten. Sie helfen uns vielmehr dabei, uns ein umfassenderes Bild von einem Plugin zu machen. Wenn Sie also mehrere Plugins gefunden haben, die Ihren Anforderungen entsprechen, können Sie die folgenden Punkte nutzen, um die Auswahl weiter einzuschränken, bevor Sie sich daran machen, die Plugins tatsächlich zu installieren und auszuprobieren.

Funktionsumfang

Mit Ihren Anforderungen an ein Plugin haben wir uns schon beschäftigt. Überprüfen Sie aber auch, ob der Funktionsumfang eines Plugins Ihre Anforderungen nicht vielleicht sogar übersteigt. Ein Plugin, das möglichst schlank bleibt und nur wirklich benötigte Funktionalitäten bereitstellt, ist oft eine bessere Wahl als All-in-One-Plugins, die versuchen, viele Dinge zu tun, und dafür nicht selten bei Benutzbarkeit und Performance Abstriche machen.

Auftritt

Egal ob auf WordPress.org oder in einem Plugin-Shop – sobald Sie auf die Detail-Seite eines Plugins kommen, werden Sie einen Eindruck davon haben, wie professionell die Darstellung ist. Werden Funktionsumfang und Funktionsweise des Plugins gut erläutert?

Gibt es Screenshots des Plugin-Interfaces oder (auf WordPress.org) eine Live-Vorschau? Hat sich vielleicht sogar jemand die Mühe gemacht, einige häufige Fragen in einer FAQ-Sektion zu beantworten?

Wenn Sie diese Fragen mit »ja« beantworten können und auch inhaltlich einen guten Eindruck haben, können Sie das als Pluspunkt für das betreffende Plugin werten. Plugins ohne ausreichende Beschreibung oder mit überbordenden Marketing-Texten schneiden in diesem Punkt eher schlechter ab.

Letzte Änderungen

Wie das ganze Web entwickelt auch WordPress sich ständig weiter. Ein aktiv gepflegtes Plugin erhält daher einigermaßen regelmäßige Aktualisierungen. Auf WordPress.org finden Sie die passende Angabe dazu in der rechten Seitenleiste.

Bei vielen Plugins ist es noch kein Grund zur Sorge, wenn zwischen Updates einige Monate vergehen. Natürliche Ausnahmen wären sicherheitskritischere Plugins wie Online-Shops.

Vor allem einfache, kleine Plugins funktionieren auch mehrere Jahre ohne ein Update. Wenn eine wichtige Funktion Ihrer Website davon abhängt, wollen Sie das Risiko, dass ein WordPress-Core-Update ein veraltetes Plugin außer Gefecht setzt, aber vermutlich nicht eingehen.



Abbildung 11.2 In der Einzelansicht auf WordPress.org finden Sie hilfreiche Informationen zu jedem Plugin.

Aktive Installationen

WordPress.org und auch einige größere Plugin-Shops geben Ihnen Informationen zur Anzahl der aktiven Installationen oder (im Fall von kostenpflichtigen Plugins) verkauften Lizenzen. Für sich allein genommen ist dieser Wert nicht sehr aussagekräftig.

Ein Plugin mit 10 aktiven Installationen kann von einer erstklassigen Entwicklerin erstellt worden sein und über Jahre gepflegt werden, während der Entwickler eines Plugins mit 900.000 aktiven Installationen morgen das Handtuch werfen und einen anderen Karriereweg einschlagen könnte.

Grundsätzlich vermittelt eine größere Zahl hier aber ein gewisses Vertrauen und kann durchaus ein Indikator für Beständigkeit sein.

Kompatibilität

Sie sollten natürlich immer eine aktuelle Version von WordPress einsetzen, und auch Ihre PHP-Version bewegt sich hoffentlich in einem gewissen Rahmen.

Weil aber Plugins manchmal bestimmte Versionen nicht mehr oder noch nicht unterstützen, sollten Sie auch darauf ein Auge haben. Auf WordPress.org finden Sie diese Information ebenfalls in der rechten Seitenleiste. Bei vielen Shops befinden sich diese Angaben oft in der Beschreibung oder einem Extra-Tab.

Bewertungen

Das umfassende Bewertungssystem auf WordPress.org gibt Ihnen nützliche Hinweise auf die Qualität des Plugins und den Service dahinter. Nehmen Sie sich einen Moment Zeit, und betrachten Sie zunächst das Verhältnis der verschiedenen Bewertungen zueinander. Viele negative Bewertungen geben ein eindeutiges Bild, viele positive Bewertungen ebenfalls. Skepsis ist angebracht, wenn 5-Sterne-Bewertungen vor allem von Accounts gegeben wurden, die am Tag der Bewertung angelegt wurden und danach nie wieder zum Einsatz kamen. Beachten Sie außerdem die Gesamtzahl der Bewertungen. Ein Schnitt von 3 Sternen bei nur 5 Bewertungen hat eine andere Aussagekraft als ein 4,7-Sterne-Schnitt bei über 500 Bewertungen.

Auch in Shops gibt es gelegentlich die Möglichkeit, Bewertungen einzusehen. Besonders wenn die Shops sich in Kontrolle der Entwickler*innen eines Plugins befinden, sind diese Daten natürlich mit Vorsicht zu genießen.

Support-Aktivitäten

Während Sie in den Shops vieler Anbieter die Support-Aktivität nicht direkt sehen können, ist das auf WordPress.org sehr transparent. Hier können Sie in der rechten Seitenleiste auf einen Blick erkennen, ob es ungelöste Fragen zu einem Plugin gibt.



Abbildung 11.3 Auf WordPress.org können Sie die Support-Anfragen im öffentlichen Forum jedes Plugins einsehen. Gelöste Anfragen werden mit einem grünen Haken markiert.

Wenn Sie auf den Eintrag SUPPORTFORUM ANZEIGEN in der Seitenleiste klicken, gelangen Sie auf eine Liste der vergangenen und aktuellen Support-Anfragen über das jeweilige Forum auf WordPress.org. Hier können Sie nicht nur die Fragen und Antworten in den einzelnen Konversationen lesen, sondern auch sehen, ob sich die Entwickler*innen um aktiven Support bemühen. Unbeantwortete Fragen im Forum können ein Zeichen dafür sein, dass ein Plugin nicht mehr aktiv betreut wird.

Gleichzeitig gilt es zu beachten, dass Entwickler*innen auf WordPress.org nicht verpflichtet sind, das Supportforum zu nutzen. Viele, die Plugins in ihrer Freizeit entwickeln und betreuen, entscheiden sich aktiv gegen die Verwendung des Forums.

Tipp: Persönliche Plugin-Empfehlungen

Die persönliche Empfehlung eines Plugins durch erfahrene WordPress-Anwender*innen kann Ihnen einiges Kopfzerbrechen und Recherche bei der Plugin-Auswahl ersparen. Passende Gesprächspartner*innen finden Sie zum Beispiel auf lokalen *WordPress-Meetups* (*https://wpmeetups.de/*). Aber auch in diesem Buch bemühen wir uns, Plugins zu empfehlen, mit denen wir in der Vergangenheit gute Erfahrungen gemacht haben.

Nach dieser Liste von Kriterien haben Sie hoffentlich ein etwas besseres Bild davon, ob Ihnen ein Plugin vertrauenswürdig erscheint. Nun geht es daran, Ihre Auswahl einmal zu installieren – egal ob nur noch ein einziges Plugin im Rennen ist oder ob Sie sich zwischen einer Handvoll Kandidaten entscheiden müssen.

In einer lokalen Testumgebung oder einer Playground-Umgebung können Sie ein neues Plugin testen, bevor Sie es auf Ihre Website loslassen.

11.4 Plugins installieren

Egal ob in Ihrer tatsächlichen, produktiven Website oder in einer Test-Installation: Um ein Plugin zu verwenden, müssen Sie es zunächst installieren. Für Plugins, die Sie von WordPress.org installieren, können Sie das in Ihrer komfortablen WP-Admin-Oberfläche erledigen. Bei Plugins aus anderen Quellen können Sie eine ZIP-Datei hochladen.

11.4.1 Installation über das WordPress-Interface

Als Admin-User gelangen Sie in WP-Admin im Menü auf der linken Seite über den Punkt Plugins auf eine Liste Ihrer bereits installierten Plugins. Je nach Konfiguration können hier standardmäßig bereits einige Plugins wie »Hello Dolly« und »Akismet« vorinstalliert sein. Wie Sie diese Plugins entfernen können (und das sollten Sie), sehen wir uns in Abschnitt 11.6 an. Doch vorher installieren wir aber ein neues Plugin.

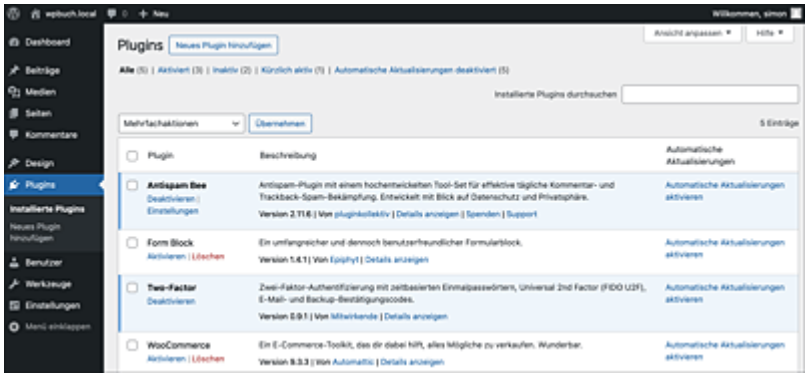


Abbildung 11.4 Auf der Seite »Plugins« erhalten Sie eine Übersicht über die Plugins, die auf Ihrer Website aktuell installiert sind.

Klicken Sie dafür entweder oben oder links im Menü auf Neues Plugin hinzufügen.

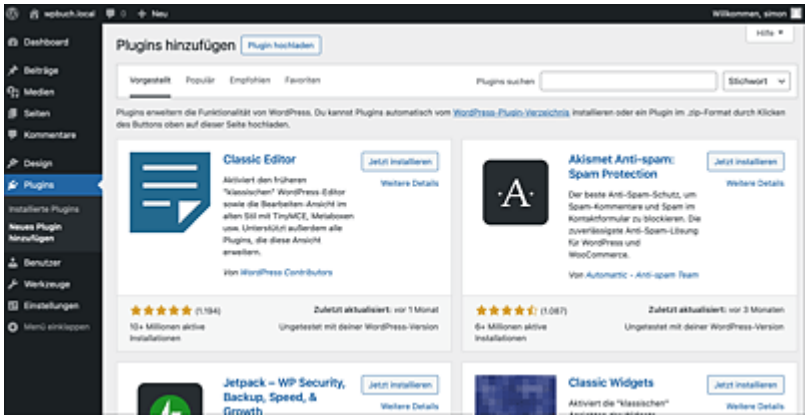


Abbildung 11.5 Unter »Neues Plugin hinzufügen« können Sie nach neuen Plugins suchen und diese installieren.

Auf der Seite PLUGINS HINZUFÜGEN sehen Sie oben eine Leiste mit verschiedenen Reitern. Standardmäßig ist hier der Reiter VORGESTELLT ausgewählt. Wenn Sie möchten, können Sie zu den anderen Reitern POPULÄR und EMPFOHLEN wechseln. Tatsächlich zeigen alle Reiter aber jeweils nur einen winzigen Bruchteil der knapp 60.000 Plugins, die im offiziellen Verzeichnis auf WordPress.org verfügbar sind.

Tipp: Unsere Plugin-Favoriten ansehen

Unter dem Punkt **FAVORITEN** können Sie sich eine Liste der Plugins ansehen, die von einem beliebigen User-Account auf WordPress.org als Favoriten gespeichert wurden.

Um hier eine Liste aller Plugins, die wir in diesem Buch empfehlen, anzusehen, können Sie zum Beispiel »WPHandbuch« in das Feld für den Benutzernamen eintragen.

Wenn Sie ein bestimmtes Plugin oder eine bestimmte Funktionalität suchen, können Sie aber auch direkt die Suchfunktion bemühen, die sich rechts neben den Reitern befindet. Sobald Sie hier einen Suchbegriff eingegeben haben und die ↵-Taste drücken, sucht WordPress im offiziellen Verzeichnis nach

einem passenden Treffer. Die Suchfunktion ist notorisch schlecht. Wenn Sie nach einem bestimmten Plugin suchen, kann es passieren, dass Sie es nicht unter den Top-Ergebnissen haben, auch wenn Sie den exakten Plugin-Namen eingeben. In einigen Fällen lohnt sich also auch noch eine parallele Suche mit der Suchmaschine Ihrer Wahl.

In den Suchergebnissen erhalten Sie einen schnellen Überblick über Namen, Beschreibung, Bewertung, Anzahl der Installationen und den Zeitpunkt des letzten Updates. Mit einem Klick auf Weitere Details können Sie eine Mini-Version der Plugin-Beschreibung inklusive Screenshots betrachten.

Ein Klick auf Installieren installiert das Plugin. Nach einigen Sekunden ist die Installation abgeschlossen, und der Installieren-Button wird durch einen blauen Aktivieren-Button ersetzt.

Installierte Plugins – egal ob aktiviert oder deaktiviert – tauchen in der Plugin-Übersicht (Menüeintrag Plugins) auf.

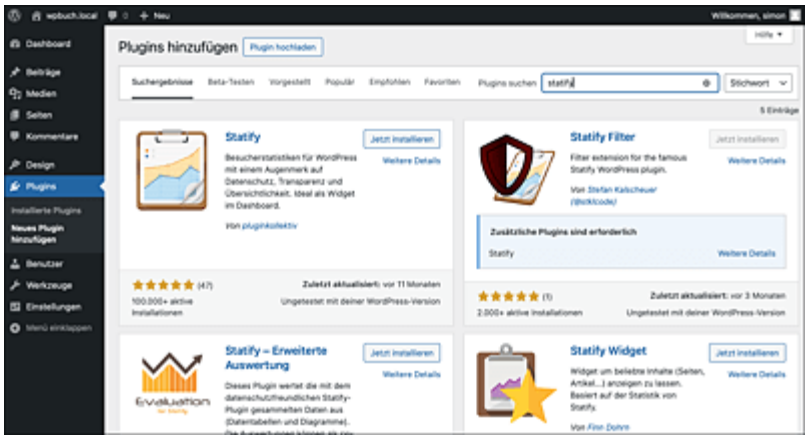


Abbildung 11.6 Über die Suche finden Sie neue Plugins von WordPress.org und können diese direkt installieren und aktivieren.

11.4.2 Manuelle Installation per ZIP-Datei

Auch Plugins aus einer anderen Quelle als WordPress.org können Sie im relativen Komfort der WordPress-Admin-Oberfläche installieren. Dafür erhalten Sie in der Regel eine ZIP-Datei des Plugins. Entpacken Sie diese testweise auf Ihrem Computer. Bei manchen Anbietern findet sich in der Datei mehr als nur das Plugin. Wenn hier nicht nur der Plugin-Code, sondern erst mal weitere Unterverzeichnisse zu finden sind, wird das die Installation in WordPress stören.

Wenn Ihre ZIP-Datei bereit ist, starten Sie die manuelle Installation wie auch die vorherige Variante auf der Unterseite Plugin hinzufügen. Statt sich jedoch mit den verschiedenen Reitern oder der Suche zu beschäftigen, können Sie direkt oben neben dem Seitentitel auf Plugin hochladen klicken.

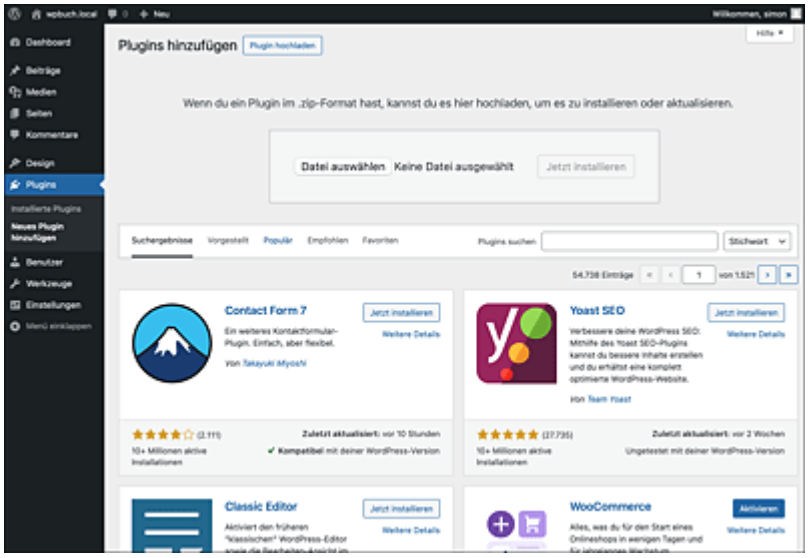


Abbildung 11.7 Im Upload-Dialog können Sie eine ZIP-Datei von Ihrem Computer auswählen und hochladen.

In dieser Ansicht haben Sie zusätzlich zu den eben bereits beschriebenen Bedienelementen eine Box zum Hochladen eines Plugins. Über diese Box können Sie eine ZIP-Datei auf Ihrem Computer auswählen und mit einem Klick auf Jetzt installieren hochladen.

War der Upload erfolgreich, werden Sie auf eine neue Seite weitergeleitet, auf der Sie neben einem knappen Protokoll der durchgeführten Aktionen einen Button Plugin aktivieren und einen Link Weiter zur Plugin-Installation angezeigt bekommen.

Während der Button das Plugin direkt aktiviert, würde Sie der Link wieder zurück in die Ansicht zum Installieren eines Plugins leiten.

11.4.3 Manuelle Installation per (S)FTP

In der Praxis werden Sie alle Plugins über die beiden bisher vorgestellten Wege installieren können. Für die wenigen Fälle, in denen das nicht möglich oder nicht gewünscht ist, gibt es aber eine weitere, weniger elegante Möglichkeit.

Hierfür benötigen Sie einen (S)FTP-Zugang zu Ihrem Webserver. Verwenden Sie eine (S)FTP-Software Ihrer Wahl, um eine Verbindung mit Ihrem Server aufzubauen.

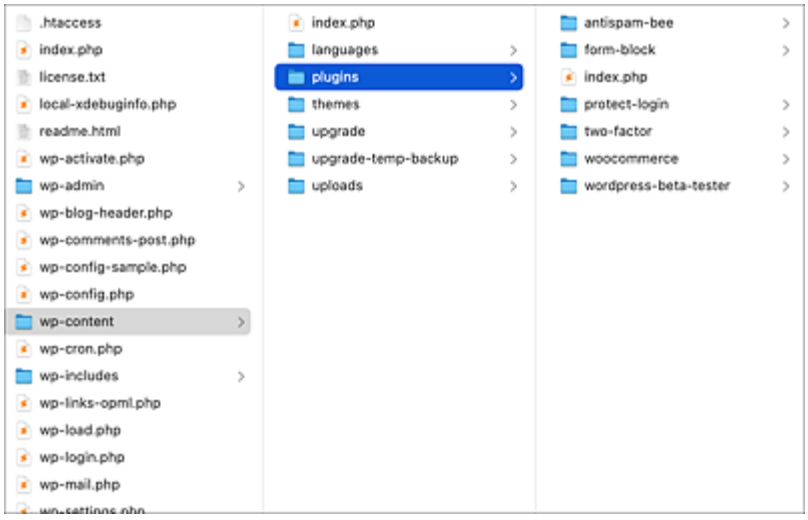


Abbildung 11.8 Im Verzeichnis `/wp-content/plugins/` finden Sie alle installierten Plugins in ihren jeweiligen Unterverzeichnissen.

Tipp: Manuelle Installation in lokaler Testumgebung

Die meisten lokalen Testumgebungen können Sie einfach über den Explorer oder Finder Ihres Computers als normales Verzeichnis erreichen. In diesem Fall benötigen Sie also keinen gesonderten (S)FTP-Client. Die folgenden Schritte bleiben dieselben.

Navigieren Sie in das Verzeichnis, in dem Ihre WordPress-Installation abgelegt ist. Dort sollten Sie ein `wp-content` Verzeichnis vorfinden. Zu den Unterverzeichnissen von `wp-content` gehört `plugins`. Wenn bereits Plugins installiert sind, werden sie in der Regel ordentlich in Unterverzeichnissen von `plugins` abgelegt.

Bevor Sie Ihr neues Plugin in diese Sammlung einreihen können, müssen Sie lokal einmal die ZIP-Datei des Plugins entpacken. Nun können Sie das beim Entpacken entstandene Verzeichnis in den Ordner `plugins` auf Ihrem Server bewegen.

Wenn der Upload erfolgreich war, erkennt WordPress das neue Plugin und fügt es der Plugin-Übersichtstabelle hinzu, die Sie nach einem Wechsel in Ihren Browser unter dem Menüpunkt PLUGINS Ihrer WP-Admin-Oberfläche finden.

11.5 Plugin aktualisieren

Genau wie für Apps auf Ihrem Smartphone gibt es auch für WordPress-Plugins regelmäßig Aktualisierungen. Um sicherzustellen, dass Ihre Website dauerhaft sicher und erreichbar bleibt, sollten Sie diese Updates auch tatsächlich durchführen.

Ähnlich wie bei der Plugin-Installation gibt es auch hierfür verschiedene Methoden.

11.5.1 Aktualisierungen mit dem WordPress-Updater

Über die zentrale Update-Seite

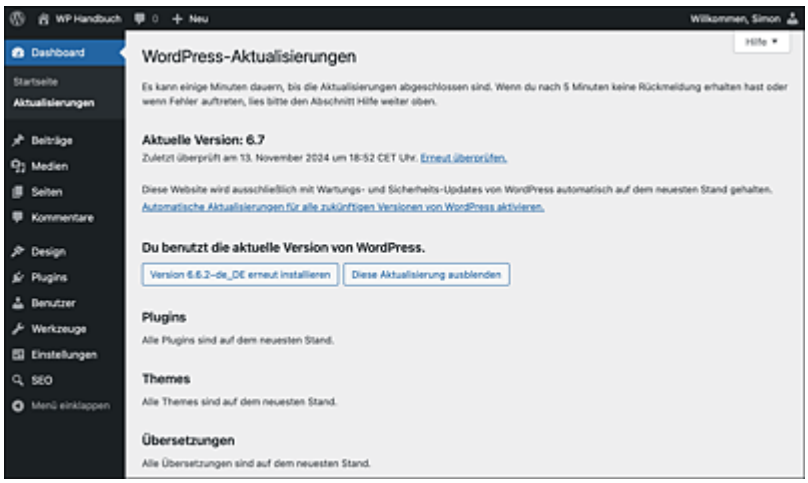


Abbildung 11.9 Unter »Dashboard • Aktualisierungen« erhalten Sie einen schnellen Überblick über alle verfügbaren Updates.

Die mit Abstand einfachste Methode ist die Aktualisierung über das WordPress-Interface. Wenn Sie als Admin-User angemeldet sind, finden Sie einen Untermenüpunkt Aktualisierungen unter dem ersten Punkt Dashboard im Admin-Menü auf der linken Seite.

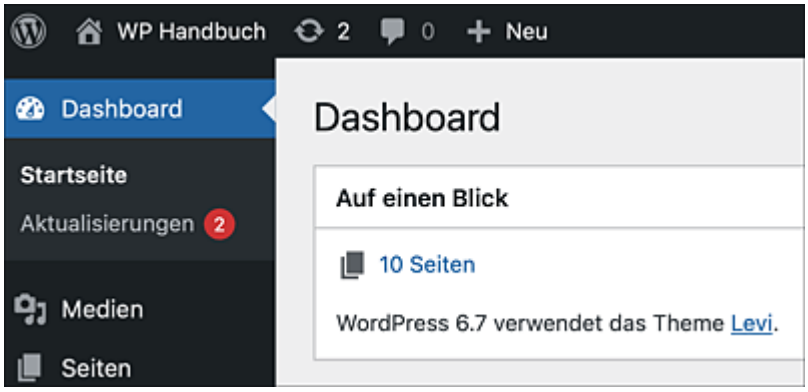


Abbildung 11.10 Wenn Updates verfügbar sind, zeigt WordPress im Admin-Menü und oben in der Werkzeugleiste eine farbig hinterlegte Zahl.

Wenn Updates für Ihre Plugins (und Themes, WordPress-Core und Übersetzungen) verfügbar sind, wird Ihnen an dieser Stelle im Menü und oben in der Werkzeugleiste zusätzlich auch ein Indikator angezeigt.

Über die Seite WordPress-Aktualisierungen können Sie sämtliche Updates durchführen, die direkt von WordPress.org kommen. Das betrifft den WordPress-Core, Plugins und Themes aus den offiziellen Verzeichnissen und die Übersetzungen für alle diese Software-Pakete, die von Freiwilligen gepflegt werden.



Abbildung 11.11 Im Bereich Plugins finden Sie die verfügbaren Updates Ihrer installierten Plugins.

Wenn für Ihre Plugins Aktualisierungen verfügbar sind, werden Sie im Abschnitt Plugins auf dieser Seite eine Tabelle mit dem oder den installierbaren Updates angezeigt bekommen. Für jedes Plugin

können Sie hier sowohl die aktuell auf Ihrer Website genutzte Version als auch die neueste verfügbare Version einsehen. Ein Klick auf Details der Version 1.3.0 anzeigen blendet das Änderungsprotokoll des jeweiligen Plugins ein. Entwickler*innen, die etwas von ihrem Handwerk verstehen, werden Ihnen in diesem Protokoll Hinweise darauf geben, welche Neuerungen und ggf. wichtigen Änderungen es in einem Update gibt.

Für jedes Plugin, das Sie aktualisieren möchten, setzen Sie zunächst einen Haken links in der Tabelle. Wenn Sie alle Plugins auf einmal auswählen möchten, können Sie ganz oben oder unten in der Tabelle auf Alle auswählen klicken. Ein anschließender Klick auf den Button Plugins aktualisieren unterhalb der Tabelle startet das Update.

Während des Update-Vorgangs werden Sie auf eine vorübergehend leere Seite weitergeleitet, auf der nach einigen Sekunden, nach erfolgter Aktualisierung, ein kurzes Protokoll der durchgeführten Operationen durch einen Klick auf Weitere Details angezeigt werden kann.

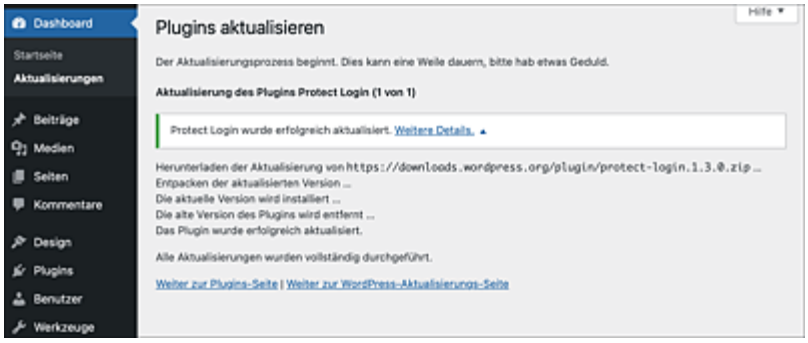


Abbildung 11.12 Im Update-Protokoll können Sie die Schritte nachvollziehen, die WordPress beim Aktualisieren durchgeführt hat.

Zwei Links unterhalb des Protokolls erlauben Ihnen, entweder mit einem Klick auf Weiter zur Plugins-Seite direkt in die Plugin-Übersicht zu springen oder durch Klicken auf Weiter zur WordPress-

Aktualisierungs-Seite zurück auf die zentrale Update-Seite zu gelangen.

Direkt aus der Plugin-Übersicht

Wenn Sie sich für den Moment nur mit Plugin-Updates beschäftigen möchten, finden Sie diese auch direkt in der Plugin-Übersicht.



Abbildung 11.13 Auch aus der Plugin-Übersicht können Sie Plugin-Updates direkt ausführen.

Plugins, für die eine Aktualisierung verfügbar ist, werden in der Plugin-Übersicht mit einem gelben Kasten markiert. Für gewöhnlich werden Sie hier informiert, welche neue Version verfügbar ist. Ein Klick auf den Link Details der Version 1.3.0 anzeigen zeigt Ihnen auch hier das Änderungsprotokoll des Plugins an. Mit einem Klick auf jetzt aktualisieren wird das Update im Hintergrund durchgeführt. Wenn es erfolgreich abgeschlossen wurde, wird die gelbe Box durch eine grüne Erfolgsmeldung ersetzt.

Info: Updates kostenpflichtiger Plugins

Das WordPress-Update-System, das hier im Hintergrund die Fäden zusammenhält, wird außerdem auch von vielen (guten) Entwickler*innen kostenpflichtiger Plugins mitgenutzt. Diese betreiben dafür eigene Update-Server, die sich in den Update-Mechanismus von WordPress mit einklinken und Ihnen Updates über dasselbe komfortable Interface erlauben.

11.5.2 Automatische Updates einstellen

Seit einiger Zeit bietet WordPress Ihnen außerdem die Möglichkeit, Plugin-Updates automatisch auszuführen. Standardmäßig ist diese Option für alle Plugins deaktiviert.

Wenn Sie automatische Updates für ein Plugin aktivieren möchten, finden Sie in der Plugin-Übersicht eine Spalte Automatische Aktualisierungen. Plugins, die noch nicht automatisch aktualisiert werden, präsentieren in dieser Spalte den Text Automatische Aktualisierungen aktivieren. Nach einem Klick wechselt dieser Text auf Automatische Aktualisierungen deaktivieren. Sobald nun eine Aktualisierung für dieses Plugin verfügbar wird, kümmert sich WordPress automatisch um das Update. Durch einen Klick auf Automatische Aktualisierungen deaktivieren können Sie automatische Updates für ein Plugin wieder deaktivieren.



Abbildung 11.14 In der Plugin-Übersicht sehen Sie in einer gesonderten Spalte den Status der Automatischen Aktualisierungen des jeweiligen Plugins.

Hinweis: Automatische Plugin-Updates oder lieber nicht?

Während automatische Updates des WordPress-Core in den meisten Fällen weitgehend bedenkenlos laufen können, ist bei Plugins etwas mehr Fingerspitzengefühl gefragt.

Sehr vorsichtig sollten Sie bei komplexen Plugins sein oder solchen, die von anderen Plugins erweitert oder ergänzt werden. Auch auf Websites, auf denen viele (mehr als 20) Plugins aktiv sind, sollten Sie automatische Plugin-Aktualisierungen nur sehr vorsichtig einsetzen.

11.5.3 Manuelle Aktualisierungen

In Fällen, in denen eine Aktualisierung über die WordPress-Oberfläche mehrfach fehlschlägt, oder wenn ein Plugin-Anbieter keine Möglichkeit anbietet, direkt innerhalb von WordPress Updates auszuführen, kann es auch notwendig werden, die Updates händisch durchzuführen. Hier haben Sie grundsätzlich die Möglichkeiten, entweder per (S)FTP die Änderungen direkt im Dateisystem Ihres Servers abzulegen oder eine ZIP-Datei über das WordPress-Interface hochzuladen.

Manuelle Aktualisierungen per (S)FTP

Verbinden Sie sich mit dem (S)FTP-Programm Ihrer Wahl zunächst mit Ihrem Server. Wählen Sie das Verzeichnis, in dem Ihr WordPress gespeichert ist, und suchen Sie dort das `wp-content` Verzeichnis. In diesem finden Sie ein Unterverzeichnis `plugins`, das wiederum für jedes installierte Plugin ein eigenes Unterverzeichnis enthält.

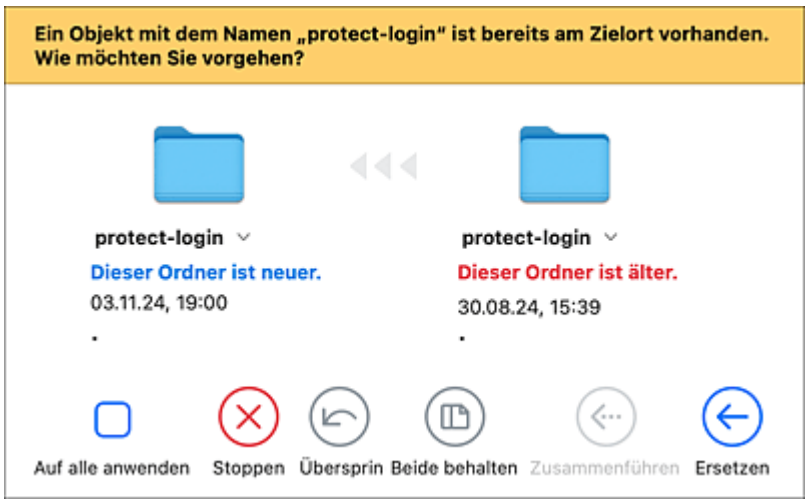


Abbildung 11.15 Mit einem (S)FTP-Client können Sie auch manuell Plugin-Updates durchführen. Ob und welche Meldung Sie in diesem Fall angezeigt bekommen, hängt von Ihrem jeweiligen (S)FTP-Client ab.

Die aktualisierte Version Ihres Plugins haben Sie vermutlich bereits auf Ihren Computer heruntergeladen. Entpacken Sie die aktuelle Version, falls sie noch als ZIP-Datei vorliegt. Stellen Sie sicher, dass der Ordnername der neuen lokalen Kopie des Plugins identisch mit dem Ordnernamen des betreffenden Plugins auf Ihrem Server ist.

Wenn Sie nun die neue, lokale Version auf den Server verschieben, wird Ihr FTP-Client Sie fragen, ob Sie die bestehende Version ersetzen möchten. Bestätigen Sie diese Frage, und warten Sie, bis die Aktualisierung abgeschlossen ist.

Wenn Sie anschließend die Plugin-Übersicht in der WordPress-Oberfläche aufrufen, sollten Sie hier das weiterhin aktive und aktualisierte Plugin vorfinden.

Manuelle Aktualisierungen per Upload

Seit wenigen Versionen ist es außerdem möglich, eine ZIP-Datei direkt über die WordPress-Oberfläche hochzuladen. Der Prozess ist identisch mit dem in Abschnitt 11.4.2 vorgestellten Ablauf zum Installieren eines Plugins mittels ZIP-Datei.

Wählen Sie im Admin-Menü Plugins und Neues Plugin hinzufügen. Klicken Sie auf der Folgeseite oben auf Plugin hochladen, und wählen Sie in der Upload-Box die entsprechende ZIP-Datei aus. Wenn Sie nun den Upload starten, erkennt WordPress, dass es sich um eine andere Version eines bereits installierten Plugins handelt.

Sie werden auf eine Seite weitergeleitet, auf der Sie von WordPress informiert werden, dass das Plugin, das Sie hochgeladen haben, bereits installiert wurde. Auf diese Information folgt eine kurze Tabelle, die Unterschiede in Versionsnummer und Kompatibilität zwischen beiden Versionen auflistet.

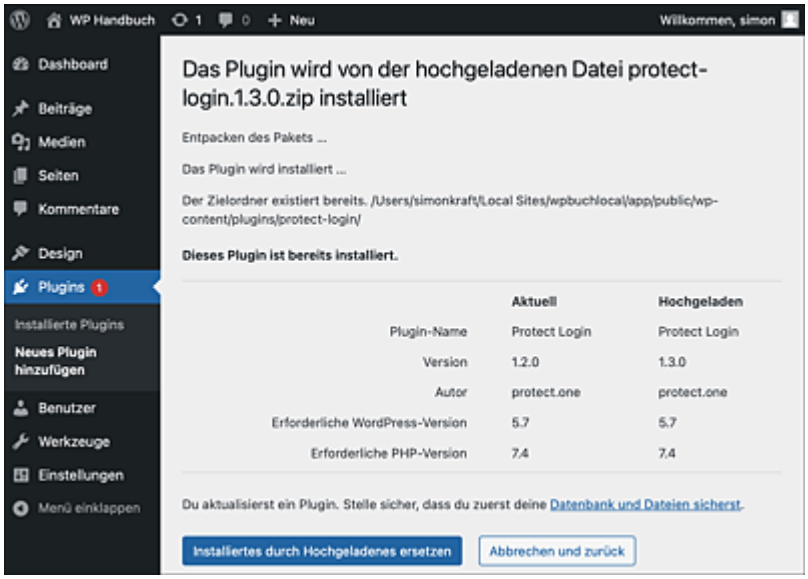


Abbildung 11.16 Wenn WordPress erkennt, dass Sie die neue Version eines bereits installierten Plugins hochgeladen haben, können Sie auch auf diesem Weg ein Update ausführen.

Mit dem blauen Button Installiertes durch Hochgeladenes ersetzen können Sie das Update aus der ZIP-Datei durchführen, mit Abbrechen und zurück unterbinden Sie den Vorgang und gelangen wieder in die Ansicht zum Installieren eines neuen Plugins.

11.6 Plugins deaktivieren und löschen

Egal, ob Sie ein Plugin nicht länger benötigen oder es nur vorübergehend deaktivieren möchten – auf der Plugin-Übersichtsseite finden Sie die passende Option dafür.

Wenn Sie also die Plugin-Übersicht über den Admin-Menüpunkt Plugins öffnen, sehen Sie in der Plugin-Tabelle unterhalb des jeweiligen Plugin-Namens mindestens eine Option Deaktivieren. Durch einen Klick auf diese Option lädt die gesamte Plugin-Übersicht neu, und das betreffende Plugin sollte nicht länger bläulich hinterlegt sein. Nun finden Sie unterhalb des betreffenden Plugin-Namens zwei Optionen: Aktivieren, um das nun deaktivierte Plugin wieder zu aktivieren, und Löschen, um es endgültig aus Ihrer WordPress-Website zu entfernen.

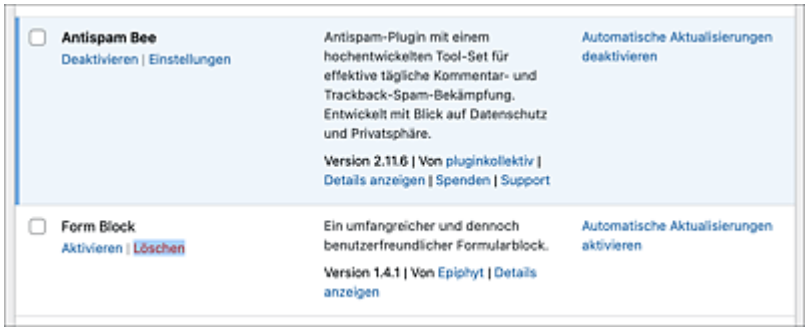


Abbildung 11.17 Nur inaktive Plugins können über das WordPress-Interface deinstalliert werden.

Tipp: Unbenutzte Plugins nicht nur deaktivieren, sondern tatsächlich löschen

In einigen Fällen kann es vorkommen, dass ein Plugin auch im deaktivierten Zustand als Angriffsweg auf eine WordPress-Installation genutzt werden kann. Deshalb ist es empfehlenswert,

unbenutzte Plugins nicht nur zu deaktivieren, sondern tatsächlich auch komplett aus Ihrer WordPress-Website zu löschen.

TEIL III

Beispiel-Projekte

12 Einen Online-Shop mit WooCommerce betreiben

Ein Anwendungsbereich, in dem sich wie in wenigen anderen die Erweiterbarkeit von WordPress zeigt, ist der E-Commerce. Die richtige Planung vorausgesetzt, können Sie mit Ihrer WordPress-Website und WooCommerce die verschiedensten Online-Shops realisieren.

In diesem Kapitel lernen Sie das WooCommerce-Plugin und das dazugehörige Ökosystem von *Erweiterungen*, *Themes* und Diensten kennen. Mit diesen Werkzeugen können Sie, basierend auf Ihrer WordPress-Website, einen eigenen Online-Shop einrichten und betreiben.

WooCommerce ist zu Recht die populärste *E-Commerce*-Lösung für WordPress und bewegt sich auch außerhalb der WordPress-Blase auf Augenhöhe mit Branchengrößen wie Magento und Shopify. Dennoch gibt es einige Fallstricke und Herausforderungen, derer Sie sich bewusst sein sollten, bevor Sie sich Hals über Kopf in ein WooCommerce-Projekt stürzen. Mit diesen Themen und ihren Auswirkungen wollen wir uns kurz beschäftigen, bevor wir in Abschnitt 12.2 in die Einrichtung von WooCommerce und die ersten Schritte mit Ihrem neuen Shop einsteigen.

12.1 Die Vor- und Nachteile von WooCommerce

Bevor Sie sich für eine E-Commerce-Lösung auf Basis von WordPress und WooCommerce entscheiden, sollten Sie sich der vollen Bandbreite von Einschränkungen und Möglichkeiten bewusst sein, die das nach sich zieht.

12.1.1 Selbst gehostet vs. SaaS

Wenn Sie mit dem Gedanken spielen, einen eigenen Online-Shop zu erstellen, haben Sie grundsätzlich die Wahl zwischen zwei recht unterschiedlichen Ansätzen. Auf der einen Seite stehen Software-as-a-Service-Angebote (kurz *SaaS*), bei denen Sie eher Mieter*in einer vorgefertigten Lösung sind, die sich im Rahmen abgesteckter Parameter nutzen lässt. Solche geschlossenen Systeme kommen in der Regel aus einer Hand, haben von Anfang bis Ende durchdachte Nutzungskonzepte und einen zentralen Support. Die Einfachheit der Software erkaufen Sie sich nicht nur durch die eingeschränkten Anpassungsmöglichkeiten, sondern oft auch durch regelmäßig anfallende Kosten oder anteilmäßige Provisionen auf Ihre Verkäufe.

Im Gegensatz dazu stehen offene Systeme wie WooCommerce, aber auch andere Open-Source-Shop-Software. Hier kommen neben der zentralen Komponente in der Regel nur einige wenige *Erweiterungen* von einem einzigen Hersteller. Fehlende Funktionen können in der Regel nachinstalliert werden – im Fall von WooCommerce als Plugins. Diese Erweiterungen, kostenfreie wie kostenpflichtige, werden von einer breiten Community von Interessierten erdacht, entwickelt und betreut. Und Erweiterungen, die nicht verfügbar sind, können mithilfe offener Schnittstellen und frei verfügbarer Dokumentation mit ein wenig Geschick und

Sachverstand nachgerüstet werden. Ein Nachteil dieses Ansatzes ist, dass Bedienungsoberflächen teilweise uneinheitlich wirken oder nicht durchgehend übersetzt sein können. Wenn sich nicht alle Anbietenden von Erweiterungen an einem gemeinsamen Interface-Design orientieren oder auch nur leicht abweichende Begriffe verwenden, können sich vor allem Einsteiger*innen schnell schwertun. Weit ärgerlicher als diese oberflächlichen Probleme sind inkompatible Erweiterungen. Erweiterungen, die nicht regelmäßig gepflegt werden, die Programmschnittstellen nicht ganz wie vorgesehen nutzen oder sich schlicht und ergreifend nicht mit anderen Drittanbietererweiterungen vertragen, können zu fatalen Problemen an Ihrem Shop führen. Eine regelmäßige Überprüfung aller installierten Plugins ist also empfehlenswert. Durch diese zusätzliche Komplexität in der Bedienung und Wartung Ihres Shops erkaufen Sie sich aber Freiheiten und Flexibilität, die Sie mit einem SaaS-Dienst nicht hätten.

Sie sehen also, sowohl selbst gehostete als auch fremd betriebene Shops haben Ihre Daseinsberechtigung und können Stärken und Schwächen in unterschiedlichen Situationen ausspielen.

Wenn Sie der zusätzliche Aufwand nicht abschreckt, werden Sie mit der Wahl von WooCommerce als E-Commerce-Lösung für WordPress mit einigen zusätzlichen Vorteilen belohnt.

12.1.2 Das WordPress-Ökosystem

Weil WooCommerce in erster Linie ein WordPress-Plugin ist, profitieren Sie mit Ihrem WooCommerce-Shop von den vielen Vorzügen des offenen Systems. So fügt sich WooCommerce fast nahtlos in Ihre bestehende Website ein und ist mit den meisten Themes grundsätzlich kompatibel. Einige Entwickler*innen haben

ihre Themes zusätzlich speziell für WooCommerce vorbereitet. Ganz ähnlich verhält es sich mit WordPress-Plugins. Die meisten der knapp 60.000 WordPress-Erweiterungen können problemlos neben WooCommerce genutzt werden. Einige Plugins berücksichtigen die Präsenz eines Shops in ihrer Funktionalität. So kann ein SEO-Plugin zum Beispiel spezielle Werkzeuge zur Optimierung von Produktseiten bereitstellen oder ein Spamschutz-Plugin gezielt auch Produktrezensionen abdecken. Aber wie schon erwähnt, gibt es hier eine ganze Unterkategorie von Plugins, die speziell als Erweiterungen von WooCommerce konzipiert sind. Mit einigen dieser Erweiterungen werden wir uns in Abschnitt 12.4, »Zusätzliche Erweiterungen für Ihren Shop« genauer beschäftigen. Sollten Sie mit Ihrem WooCommerce-Shop einmal weitergehende Hilfe benötigen, finden Sie in der WordPress- und WooCommerce-Community mehr lokale Meetups, Dienstleister und einfache Tutorials als für vermutlich jedes andere Shop-System.

Tipp: Die WordPress-Community finden

Ganz egal, ob Sie Fragen rund um WooCommerce und WordPress haben, mit Gleichgesinnten über aktuelle Trends bei Plugins und Themes fachsimpeln oder einfach nur spannende Vorträge zu aktuellen Themen rund um das WordPress-Ökosystem hören wollen – lokale Usergroups, sogenannte Meetups, sind in vielen Städten die erste Anlaufstelle.

Organisiert von Freiwilligen, treffen sich diese Gruppen in der Regel monatlich, setzen auf lokalen und direkten Austausch und sind ein wunderbarer und noch dazu kostenfreier Einstieg in die Welt von WordPress.

Im deutschsprachigen Raum finden Sie Treffen in Ihrer Nähe auf *https://wpmeetups.de*.

12.1.3 Skalierbarkeit

Wenn Sie gelegentlich lesen, dass sich WooCommerce eher für kleinere Shops eignet, hat die Sache in einigen Fällen sicherlich einen wahren Kern. Wird Ihr Shop zu groß, hat er zu viele zeitgleiche Zugriffe durch potenzielle Kundinnen und Kunden oder sind Ihre technischen Anforderungen so komplex, dass Sie eine sehr große Zahl von Erweiterungen einsetzen müssen, stößt das System schnell an seine Grenzen. In diesen Fällen lässt sich durch das Bereitstellen zusätzlicher Ressourcen im Hosting oder mit weiteren Detailoptimierungen gegensteuern.

Es gibt keine magische Grenze, an der ein WooCommerce-Shop nicht mehr funktioniert. Allen Unkenrufen zum Trotz gibt es sehr große Shops mit vielen Tausend Produkten und komplexen Geschäftsprozessen im Hintergrund, die mit WooCommerce und WordPress umgesetzt wurden. Nach meist professioneller Einrichtung und unter regelmäßiger Fürsorge und Wartung verrichten diese Shops problemlos über viele Jahre ihren Dienst. Gleichzeitig könnte Ihr Shop für Häkelwaren mit nur einer Handvoll von Produkten unter einem moderaten Ansturm von Kund*innen schnell die weiße Flagge hissen. Um solchen Problemen vorzubeugen, lohnt es sich, bei der Planung und Einrichtung ein offenes Auge zu wahren und neue Funktionen sowie deren Sinn zu hinterfragen. Je weniger komplex Ihr Shop aufgebaut ist, je weniger Plugins Sie verwenden und je schneller die grundsätzliche Performance Ihrer Website ist, desto besser sind Sie für zukünftige Erfolge in der E-Commerce-Welt aufgestellt.

12.1.4 Anbindung an andere Geschäftssysteme

Auch wenn es leicht wäre, vor dem Hintergrund eines flexibel wachsenden und modular aufgebauten Shop-Systems ins Schwärmen zu geraten und von einer unbegrenzten Flexibilität zu fantasieren, müssen wir uns noch einem weiteren Problem stellen: der Anbindung von WooCommerce an *Warenwirtschaftssysteme*, *Buchhaltungssoftware* & Co.

Durch die Popularität von WooCommerce gibt es zu vielen moderneren Geschäftssystemen einfach zu nutzende Verbindungen. Für einen neuen Shop, den Sie auf der grünen Wiese ohne vorherige Altlasten errichten, empfiehlt sich die Wahl von zeitgemäßen Geschäftssystemen, für die es bereits Integrationen in WooCommerce gibt.

Wenn Sie zum Beispiel bereits ein Ladengeschäft betreiben und dieses um einen Online-Shop erweitern wollen, ist eine Integration in Ihr bestehendes System vermutlich weit oben auf Ihrer Wunschliste. Eine Umstellung aller Geschäftsprozesse zugunsten des neuen Online-Shops mag in einigen wenigen Fällen gewünscht sein, umgekehrt wird das Ganze aber meist deutlich günstiger sein.

Eine einfache Suche mit der Suchmaschine Ihrer Wahl nach dem Namen zum Beispiel Ihrer Warenwirtschaftssoftware zusammen mit »WooCommerce« wird schnell Klarheit bringen. Bietet der Hersteller Ihrer Software eine eigene WooCommerce-Erweiterung an oder lässt sie über eine Drittanbietererweiterung integrieren, versuchen Sie Ihr Glück – in Abschnitt 1.2 erfahren Sie, wie Sie damit loslegen.

Wenn Sie nicht fündig werden, bleiben Ihnen zwei Optionen: Sie können sich nach einem anderen Shop-System umsehen, oder Sie machen sich auf die Suche nach einer Agentur oder nach freien Entwickler*innen, die eine auf Sie zugeschnittene WooCommerce-Anbindung programmieren. Wegen der Komplexität der Anbindung

und der vermutlich sensiblen Natur der übermittelten Daten empfiehlt sich eine Eigenentwicklung auch für versierte Entwickler*innen hier nicht unbedingt, eine kleine Investition in eine professionelle Lösung kann sich aber durchaus lohnen.

12.2 WooCommerce installieren und einrichten

Wenngleich dieses Kapitel keine umfängliche WooCommerce-Anleitung ersetzt, wird es Sie doch bei den ersten Schritten begleiten, Sie sicher durch die Einrichtung und das erste Befüllen Ihres Shops führen und bei der Auswahl einiger nützlicher Erweiterungen unterstützen.

Bevor wir nun also endlich mit der Installation von WooCommerce starten, empfiehlt es sich, eine Sicherheitskopie Ihrer bestehenden Website anzufertigen oder für die ersten Gehversuche auf einer (lokalen) Entwicklungsumgebung zu arbeiten.

Plugin	WooCommerce
Aktive Installationen	8.000.000+
Download	*http://r-wrk.de/woocommerce*
Zweck	umfangreiche Online-Shop-Lösung mit robustem Funktionsumfang, der nachträglich erweitert werden kann
Komplexität	+++

Zunächst installieren und aktivieren Sie WooCommerce wie bereits geübt über Plugins • Installieren. Nach der Aktivierung begrüßt Sie der Einrichtungsassistent. In diesem Prozess werden Sie durch die wichtigsten Grundeinstellungen des WooCommerce-Plugins geführt. All diese Einstellungen können Sie auch später noch verändern. Um den Start zu erleichtern, folgen wir aber den Schritten des Assistenten.

Sollten Sie den Assistenten nicht nutzen wollen, können Sie mit einem Klick auf Geführte Einrichtung überspringen zum gewohnten WordPress-Interface zurückkehren und das Setup später manuell durchlaufen.



Abbildung 12.1 Nach der Installation von WooCommerce startet der Einrichtungsassistent automatisch.

12.2.1 Der WooCommerce-Einrichtungsassistent

Der Einrichtungsassistent startet mit einigen Grundkonfigurationen. Zuerst ist die Hauptkomponente WooCommerce mit der Konfiguration an der Reihe – zunächst anhand einiger rudimentärer Schritte. Sie starten den Assistenten mit einem Klick auf Meinen Shop einrichten.

Schritt 1 – Die Marketing-Abfrage

Gleich im ersten Schritt versucht der Einrichtungsassistent, besser auf Ihre Bedürfnisse einzugehen und zu verstehen, ob Sie gerade erst mit Ihrem Unternehmen starten, bereits einen Shop haben oder einen WooCommerce-Shop für einen Kunden einrichten wollen.

Schritt 2 – Shop-Details

Im nächsten Schritt geben Sie einige Details zu Ihrem Shop an. Hier können Sie Ihrem Shop einen Namen geben, der von WooCommerce von nun an genutzt wird, später aber auch noch geändert werden kann.

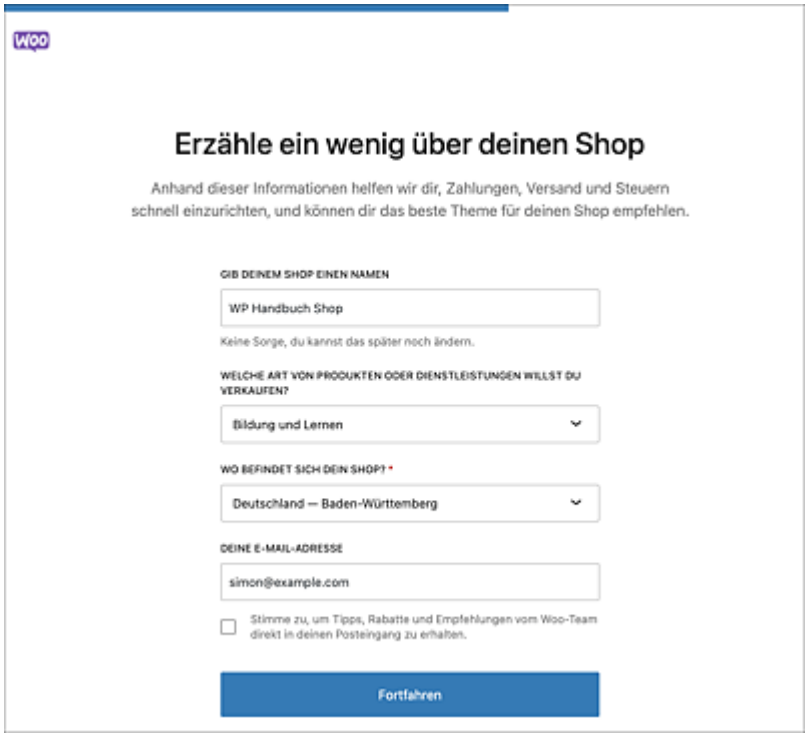


Abbildung 12.2 In welchen Bereich passt Ihr Shop am besten?

Hier können Sie aus einer Liste mit verschiedenen Bereichen wählen. Sollten Sie mit Ihrem Shop Produkte aus unterschiedlichen Bereichen anbieten wollen, ist eine Mehrfachauswahl möglich.

Wenn keiner der genannten Bereiche auf Ihren Shop zutrifft, können Sie Andere wählen.

Tipp: WooCommerce-Marketing-E-Mails erhalten

Wenn Sie den Einrichtungsassistenten zum ersten Mal starten, werden Sie sehen, dass Sie unter Ihrem bereits vorausgefüllten E-Mail-Feld einen Haken bei Stimme zu, um Tipps, Rabatte und Empfehlungen vom Woo-Team direkt in deinen Posteingang zu erhalten setzen können. Vor allem als Einsteiger*in in die Welt von WooCommerce können diese E-Mails nützlich sein. Sollten Sie grundsätzlich kein Fan von Marketing-E-Mails sein, sich anderweitig auf dem Laufenden rund um WooCommerce halten oder vielleicht sogar mehrere WooCommerce-Shops betreiben, können Sie den Haken entfernen, um Ihr E-Mail-Postfach etwas zu entlasten.

Ein Klick auf Fortfahren bringt Sie zum nächsten Schritt.

Schritt 3 – Kostenlose Plugins

Im nächsten Schritt unterbreitet Ihnen der Einrichtungsassistent allerlei kostenfreie Plugins, die praktischerweise auch alle schon vorausgewählt sind. Hier ist etwas Fingerspitzengefühl gefragt, denn hier versucht der Einrichtungsassistent, Ihnen einige zusätzliche Plugins aus dem Hause Automattic zu installieren, die Sie nicht unbedingt brauchen oder in der EU benutzen sollten.

WooPayments ist eine Anbindung an einen Zahlungsdienst, ein sogenanntes *Payment-Gateway*. Sie können WooPayments nutzen, auf PayPal zurückgreifen, das sich in den letzten Jahren großer Beliebtheit erfreut hat, oder aus den vielen anderen Bezahloptionen

wählen. Wenn Sie möchten, können Sie den Haken zu diesem Punkt entfernen.

Das Jetpack-Plugin richtet sich nicht explizit nur an WooCommerce-Kund*innen, ist aber eine bunte Sammlung verschiedener Funktionen. Auch wenn Aussichten und Funktionsumfang des Plugins verlockend sind, sollten Sie sich dessen Einsatz in der EU genau überlegen, stehen einige seiner Funktionen doch schon seit Jahren in der Kritik, nicht kompatibel mit der Datenschutz-Grundverordnung (*DSGVO*) zu sein. Auch hier sollten Sie den Haken in der Auflistung entfernen.

Meta Ads and Pixel ist das nächste Plugin, das Ihnen hier vorgeschlagen wird. Es richtet sich an Shop-Betreiber*innen, die ihren Shop mit Facebook und Instagram verbinden wollen, um dort Werbekampagnen zu schalten. Zum Thema möglicher Probleme mit der DSGVO sollten Sie auch hier vermutlich eher vorsichtig sein und im Zweifelsfall juristischen Rat suchen, bevor Sie ein solches Plugin aktivieren.

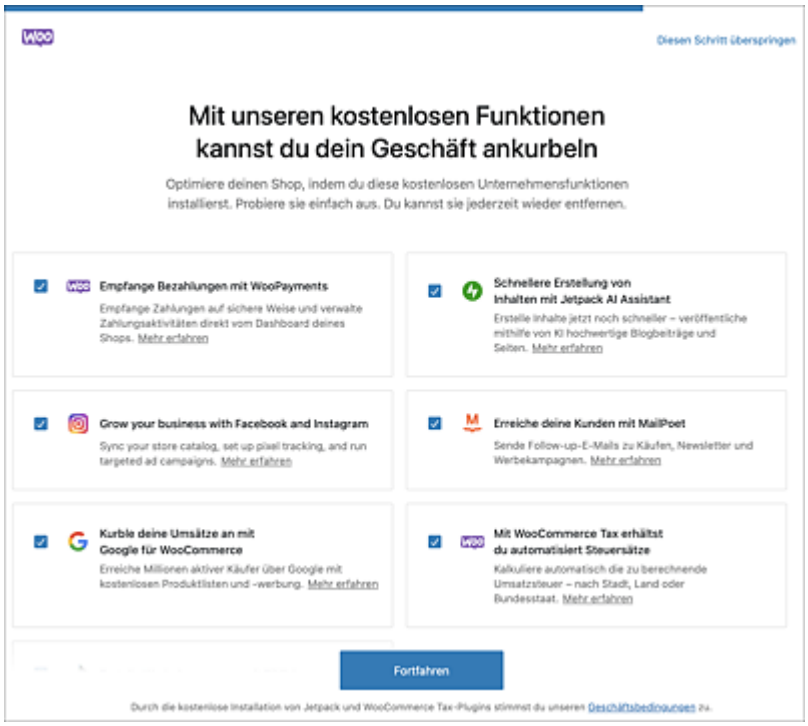


Abbildung 12.3 WooCommerce bietet Ihnen schon bei der Einrichtung eine Reihe kostenfreier Erweiterungen an. Prüfen Sie aber genau, welche davon Sie benutzen dürfen und benutzen wollen, bevor Sie fortfahren.

MailPoet ist ein Plugin zum Erstellen von Newslettern und automatisierten Mailings aus der WordPress-Oberfläche heraus. Sollte dieses Themengebiet für Sie interessant sein, lohnt sich vielleicht auch ein Blick auf das kostenpflichtige, aber sehr gute Plugin Newsletter Glue (*https://newsletterglue.com/*), das im Vergleich zu MailPoet durch eine deutlich bessere Integration in den Block-Editor überzeugt. Wenn Newsletter und E-Mail-Kampagnen Sie für den Moment nicht interessieren oder Sie eine Alternative zu MailPoet im Kopf haben, können Sie auch dieses Häkchen entfernen.

Google Listings und Ads ist das vorletzte Plugin in dieser Liste. Es kümmert sich, wie der Titel nahelegt, um die Integration Ihrer Produkte in den Shopping-Tab der Suchmaschine.

WooCommerce Tax ist ein Plugin zur korrekten Ausweisung von Steuern in Ihrem Shop. Wenn Sie Ihren Shop mit einem Plugin wie Germanized für WooCommerce einsetzen, um Ihren Shop rechtssicher zu machen – und das sollten Sie –, benötigen Sie WooCommerce Tax nicht und sollten den Haken in der Auflistung entfernen.

Wenn Sie sich um alle diese Plugins erst mal nicht kümmern möchten, können Sie durch einen Klick auf Diesen Schritt überspringen oben rechts ohne Installation fortfahren. Wenn Sie einzelne oder alle dieser Plugins installieren möchten, klicken Sie auf Fortfahren, um die ausgewählten Plugins zu installieren.

Sie können auch später jederzeit diese und viele weitere Erweiterungen für WooCommerce installieren.

Wenn Sie diesen Schritt abgeschlossen haben, ist der Einrichtungsassistent auch schon fertig. Doch die Konfiguration Ihres Shops geht jetzt erst richtig los. Sie wurden automatisch auf die Seite WooCommerce • Startseite geleitet.

12.2.2 WooCommerce anpassen

Auch für diese Anpassungen nimmt WooCommerce Sie ein Stück weit an der Hand und schlägt eine Reihe von Aktionen vor, um Sie mit Ihrem Shop vertraut zu machen.

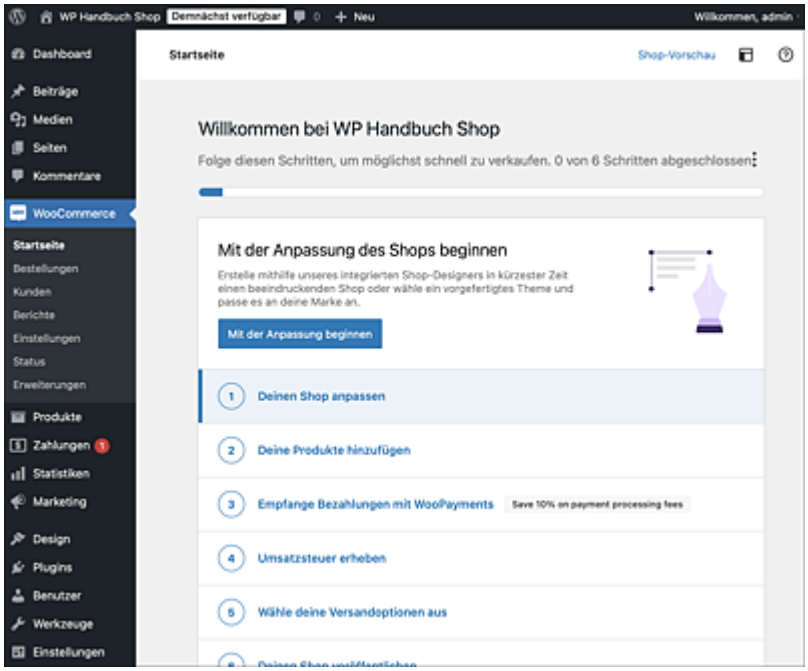


Abbildung 12.4 Auch nach dem Durchlaufen des Einrichtungsassistenten sind Sie mit der Einrichtung Ihres WooCommerce-Shops noch nicht fertig.

Schritt 1 – Deinen Shop anpassen

Vor allem wenn Sie WooCommerce auf einer bereits existierenden Website installieren, wollen Sie vermutlich bei Ihrem aktuellen Theme bleiben. Dann können Sie diesen ersten Schritt ignorieren. Ansonsten haben Sie nun die Qual der Theme-Wahl. Während

WooCommerce mit nahezu jedem Theme, das mit aktuellen WordPress-Versionen läuft, kompatibel ist, gibt es einige Theme-Entwickler*innen, die ihre Themes explizit auf die Unterstützung der WooCommerce-Gestaltungselemente optimiert haben.

Ein Klick auf Deinen Shop anpassen bringt Sie in eine Auflistung möglicher kostenfreier und kostenpflichtiger Themes.

Tipp: Theme-Auswahl

In diesem Kapitel beschäftigen wir uns nicht näher mit der Auswahl eines WooCommerce-Themes, auch wenn sich hierzu ein komplettes eigenes Kapitel schreiben ließe.

Grundsätzlich können Sie fast jedes WordPress-Theme auch für Ihren WooCommerce-Shop einsetzen. Einige Anbieter*innen von Themes optimieren ihr Angebot noch mal gesondert auf das Shop-Plugin, um auch exotischere Elemente wie den Warenkorb oder eine Produktübersicht ordentlich und vor allem im gestalterischen Einklang mit dem restlichen Webdesign darzustellen.

Besonders wenn Sie Ihren Shop als zusätzlichen Teil einer bereits bestehenden Website nutzen wollen, ist ein Theme-Wechsel unter Umständen ein aufwendiges und damit wenig attraktives Unterfangen. Ob Sie sich ein spezielles WooCommerce-Theme zulegen, ist daher ganz Ihnen überlassen. WooCommerce wird sich in den nächsten Jahren besser auf den Block-Editor einstellen, stärker auf eigene Blöcke setzen und damit noch modularer aufgebaut sein. Die Kompatibilität zu anderen Themes dürfte damit noch einmal verbessert werden.

Auf *https://woocommerce.com/de/product-category/themes* finden Sie eine streng überprüfte Auswahl kostenpflichtiger Themes.

Schritt 2 – Produkte hinzufügen

Die Liste der möglichen Produkttypen, die WooCommerce unterstützt, ist etwas länger. In den meisten Fällen werden Sie sich aber zwischen zwei Optionen entscheiden:

Wenn Sie planen, echte, physische Produkte anzubieten, sei es im stationären Verkauf vor Ort, per Versand oder über einer Kombination aus beidem, ist Physisches Produkt die Option Ihrer Wahl.

Sollten Ihre Produkte weniger greifbarer Natur sein, bietet sich die Option Digitales Produkt an. Solche virtuellen Produkte haben den charmanten Vorteil, dass Sie sich um lästige Punkte wie Verpackung, Versand und Rücksendung keine Gedanken machen müssen – und WooCommerce Ihnen dafür entsprechend auch keine Einstellungen anbieten muss.

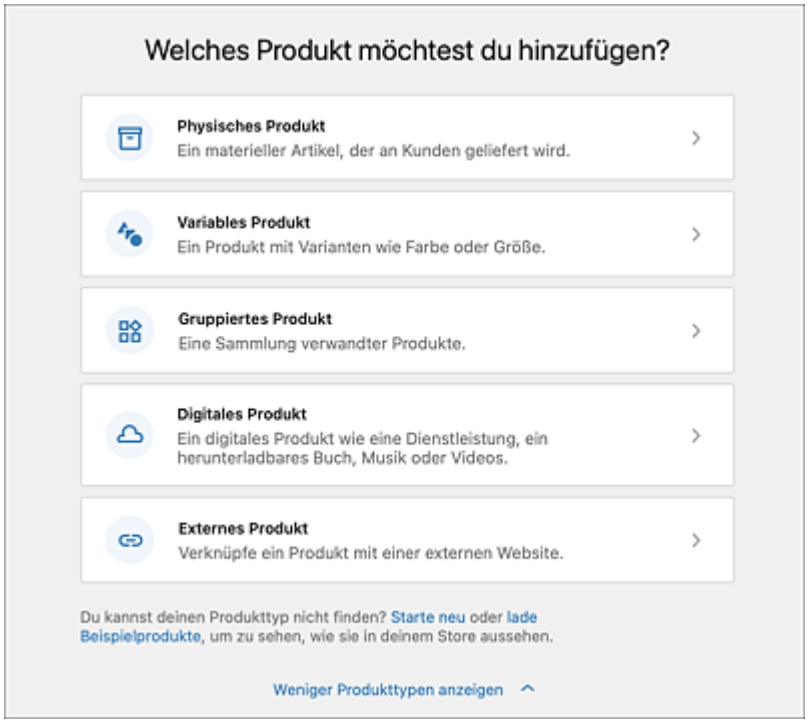


Abbildung 12.5 Sie haben die Wahl, welche Art von Produkt Sie hinzufügen möchten.

Ihr Shop muss sich nicht auf einen der beiden Produkttypen beschränken. Wenn Sie zusätzlich zu Ihren gehäkelten Topflappen und Zierdeckchen auch noch digitale Häkelanleitungen zum Download anbieten möchten, können Sie also auch Produkte beider Typen anlegen. Ebenso steht es Ihnen später jederzeit frei, Produkttypen ins oder aus dem Sortiment zu nehmen.

Wenn Ihre Bedürfnisse mit diesen beiden Basisprodukttypen bereits befriedigt sind, können Sie mit einem Klick auf Fortfahren zum nächsten Punkt weitergehen. Sollten Sie bei einem Blick auf die zusätzlichen Optionen neugierig geworden sein, finden Sie in Abschnitt 12.4, »Zusätzliche Erweiterungen für Ihren Shop«, Informationen zu einigen zusätzlichen Plugins. Wenn Sie planen, sich Ihrem Shop tatsächlich ernsthaft zu widmen und ihn für mehr als ein Hobbyprojekt zu nutzen, ist die Investition in die teilweisen recht hochpreisigen Erweiterungen oft eine lohnende Entscheidung. Allein mit den Optionen aus dieser Liste könnten Sie Ihren Shop für Häkelwaren um ein Topflappen-Abo, attraktive Bundle-Angebote (drei Topflappen zum Preis von zwei), Angebote für zahlende Mitglieder und kostenpflichtige Termine für persönliche Häkelkurse erweitern. Die Optionen und Kombinationsmöglichkeiten sind nahezu endlos.

Sie können von hier aus direkt Produkte anlegen. Alternativ haben wir den Prozess in Abschnitt 12.2.4 auch noch einmal genauer erläutert.

Schritt 3 – WooPayments

Über dieses Plugin sind Sie schon in Abschnitt 12.2.1 gestolpert. Wenn Sie es nach wie vor nicht benötigen, können sie es auch hier getrost ignorieren.

Schritt 4 – Umsatzsteuer

Ähnlich verhält es sich mit dem Punkt Umsatzsteuer erheben. Hier verbirgt sich das Plugin WooCommerce Tax, dass Sie auch nicht zwingend installieren müssen.

Schritt 5 – Versand

Wenn Sie sich an den Verkauf physischer Produkte machen, können Sie hier einige Details für den Versand vorbereiten. Nach Eingabe Ihrer Adresse schlägt WooCommerce Ihnen verschiedene Erweiterungen vor, mit denen Sie den Druck von Versandetiketten und die Versandabwicklung automatisieren können.

Schritt 6 – Veröffentlichen

Abschließend könnten Sie jetzt Ihren Shop veröffentlichen und damit aus dem Vorschau-Modus befreien. Bevor Sie diesen Schritt gehen, widmen wir uns aber noch kurz der Vorbereitung Ihres Shops für den deutschsprachigen Markt.

12.2.3 Rechtssicherheit mit »Germanized für WooCommerce«

WooCommerce ist, ebenso wie die meisten seiner Erweiterungen, nicht primär für den europäischen Markt entwickelt worden. Infolgedessen legt es, wenn man es ohne zusätzliche Erweiterungen einsetzt, eine wunderbare Ignoranz gegenüber den hier geläufigen rechtlichen Anforderungen und Erwartungen an den Tag.

Dank seiner Natur als quelloffene, flexible und erweiterbare Software ist es dennoch möglich, auch in der EU WooCommerce-

Shops rechtssicher zu betreiben und potenziellen Käuferinnen und Käufern alle Annehmlichkeiten zu bieten, die sie von einem modernen Online-Shop erwarten würden. Dafür haben sich in den letzten beiden Jahren die Erweiterungen zweier deutschsprachiger Anbieter als zuverlässig erwiesen: Germanized für WooCommerce und German Market. Wer in Deutschland einen WooCommerce-basierten Online-Shop betreibt, sollte sich zwischen diesen beiden Plugins entscheiden und eines davon dann auch tatsächlich nutzen.

In diesem Abschnitt werden wir mit Germanized für WooCommerce arbeiten. Die beiden konkurrierenden Angebote sind in ihrem Funktionsumfang aber vergleichbar und auch in der Bedienung ähnlich, sodass Sie die freie Wahl zwischen beiden haben.

Plugin	Germanized für WooCommerce
Aktive Installationen	80.000+
Download	http://r-wrk.de/germanized
Zweck	für deutsche Online-Shops spezialisierte Erweiterung von WooCommerce mit rechtsrelevanten Funktionen; zusätzlich zur kostenfreien Version ist eine Pro-Version mit Buchhaltung, Mustertexten und Support verfügbar (und empfehlenswert)
Komplexität	+++

Installieren und aktivieren Sie zunächst das Plugin Germanized für WooCommerce. Ähnlich wie bei WooCommerce selbst werden Sie auch hier nach der Aktivierung direkt zu einem Einrichtungsassistenten geleitet. Auch diesen Assistenten gehen wir nun einmal gemeinsam durch, um alle relevanten Einstellungen in

Germanized für WooCommerce zu setzen, bevor wir mit der weiteren Einrichtung des Shops fortfahren.

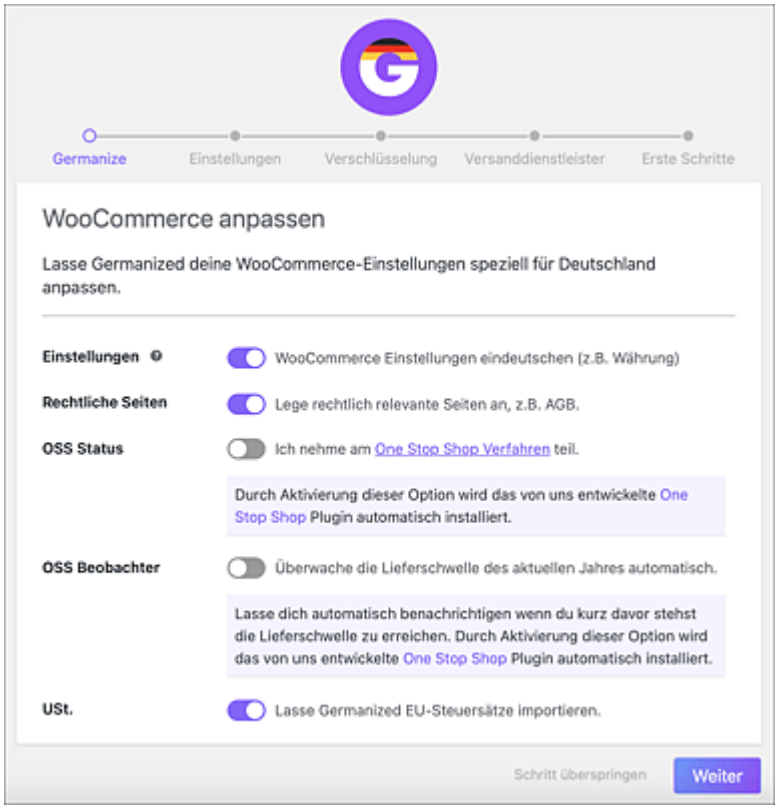


Abbildung 12.6 Auch »Germanized für WooCommerce« führt Sie mit einem Einrichtungsassistenten durch die wichtigsten Schritte.

Schritt 1 – Grundeinstellungen

Im ersten Schritt sind einige sinnvolle Punkte bereits vorausgewählt. In den wenigsten Fällen werden Sie das Plugin einsetzen, aber auf die Konfiguration der WooCommerce-Einstellungen und den Import von EU-Steuersätzen verzichten wollen.

Schritt 2 – Weitere Einstellungen

Wenn Sie nach § 19 UStG unter die Kleinunternehmerregelung fallen, können Sie die Option Kleinunternehmerregelung aktivieren, damit WooCommerce auch in diesem Spezialfall, den es von sich aus nicht kennen würde, korrekt mit Preis und Steuerberechnung umgeht.

Den Double-Opt-in sollten Sie auf jeden Fall aktivieren. Diese Option bewirkt, dass neue Konten für Kundinnen und Kunden nur angelegt werden, wenn diese ihre E-Mail-Adressen durch den Klick auf einen Link in einer E-Mail bestätigen. Auf diesem Wege können Sie sowohl sicherstellen, dass E-Mail-Adressen korrekt sind, als auch, dass nur von den tatsächlichen Eigentümer*innen einer Adresse Konten angelegt werden und dass das nicht gegen deren Willen geschehen kann.

Über Weiter kommen Sie auch hier zum nächsten Schritt.

Schritt 3 – Verschlüsselte Daten

Um sensible Daten nicht im Klartext in Ihrer Datenbank abzulegen, bietet Germanized die Option an, einen Schlüssel in Ihrer *wp-config.php*-Datei abzulegen, mit dessen Hilfe das Plugin sensible Daten wie etwa Zugangsdaten verschlüsseln kann.

Wenn Sie auf Schlüssel einfügen klicken, legt das Plugin den entsprechenden Eintrag an, und Sie gelangen zum nächsten Schritt. Alternativ können Sie auch mit einem Klick auf Schritt überspringen fortfahren.

Schritt 4 – Versandoptionen

Ähnlich wie bei Payment-Gateways ist auch die Auswahl an Versanddienstleistern je nach Land eine ganz andere. Für

Deutschland bietet Germanized für WooCommerce eigene Integrationen mit der Deutschen Post und DHL an. In der kostenpflichtigen Version ist auch eine DPD-Integration möglich.

Wenn Ihr Shop nur digitale Produkte anbietet, ist die Auswahl eines Versanddienstleisters natürlich unnötig, in den meisten anderen Fällen werden Sie vermutlich mindestens einen der hier gelisteten Anbieter nutzen. Doch auch für weitere Anbieter finden sich passende Erweiterungen.

Treffen Sie Ihre Auswahl und klicken Sie auf Weiter.

Tipp: »Germanized für WooCommerce Pro«

Die kostenfreie Version des Plugins legt den Grundstein für einen rechtssicheren WooCommerce-Shop. Weil in der Entwicklung kostenfreier Plugins aber keine besonderen Reichtümer liegen, hat sich der Entwickler des Plugins für ein sogenanntes *Freemium-Modell* entschieden. Zusätzlich zum kostenfreien Kern-Plugin gibt es daher eine kostenpflichtige Pro-Version, die einige attraktive Features mitbringt.

- *Mustertexte* füllen die Seiten für Impressum, Datenschutz, AGBs & Co, die während der Plugin-Einrichtung nur als leere Platzhalter angelegt wurden, mit belastbaren Rechtstexten. Allein für dieses Feature lohnt sich für viele Anwender*innen der Kauf der Pro-Version vermutlich schon.
- *Buchhaltung* stellt neben mächtigen Integrationen in die populären Online-Dienste *Lexoffice* und *Sevdesk* auch das automatische Anlegen von PDF-Rechnungen zur Verfügung.

Werfen Sie einen Blick auf diese und viele weitere Pro-Funktionen und entscheiden Sie dann, ob Sie ein wenig Geld für das Plugin

ausgeben möchten. Sollten Sie sich dagegen entscheiden, müssen Sie sich im Folgenden noch um die passenden Rechtstexte kümmern.

Schritt 5 – Erste Schritte

Damit ist die einleitende Konfiguration von Germanized für WooCommerce abgeschlossen. Bevor Sie zurück zum WordPress-Interface wechseln oder das Tutorial starten, um die wichtigsten Einstellungen kennenzulernen, sollten Sie über den Button Entdecke die Pro-Version noch einen Blick auf die Möglichkeiten werfen, die sich in der kostenpflichtigen Version des Germanized-Plugins ergeben.

12.2.4 Produkte hinzufügen

Wenden wir uns nun wieder den praktischeren Aspekten Ihrer Shop-Einrichtung zu, und legen wir ein paar erste Produkte an. Navigieren Sie dafür wieder zurück zu WooCommerce • Startseite.

Lassen wir diesen Schritt aber für den Moment außer Acht, steht das Anlegen der ersten Produkte (oder auch nur eines einzigen Produkts) als Nächstes auf unserer Liste.

Ein Klick auf Meine Produkte hinzufügen stellt Sie vor die Wahl, ein Produkt entweder aus einer Vorlage zu erstellen, es manuell anzulegen oder es zu importieren.

Wählen Sie Beginne mit einer Vorlage und fahren Sie mit Physisches Produkt fort. Sie werden nach einem Klick auf Los direkt in den Produkt-Editor geleitet, in dem Beispielinhalte auf Sie warten.

Im Editor angekommen, werden Sie von einer kleinen Tour begrüßt, die Ihnen die wichtigsten Felder und Einstellungsmöglichkeiten des Produkt-Editors nahebringt.

Los geht es mit dem Produktnamen. Dieser kann über das Feld direkt unter dem Titel Neues Produkt hinzufügen eingetragen werden.

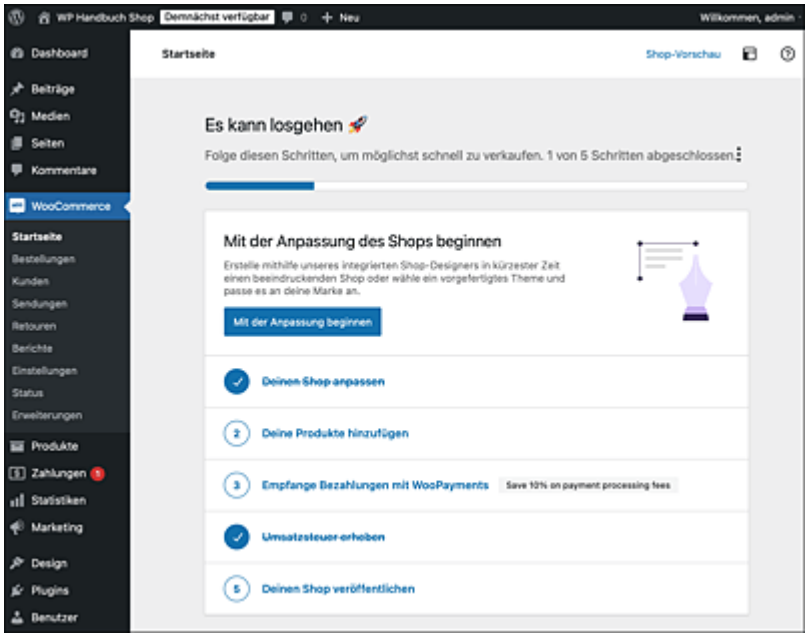


Abbildung 12.7 Die Stichpunkte, mit denen WooCommerce Sie zu leiten versucht, müssen nicht unbedingt in dieser Reihenfolge bearbeitet werden, sind aber eine gute Gedächtnisstütze.

Über das große Editorfeld direkt unter dem Produktnamen können Sie die Produktbeschreibung einfügen. Dieses Feld bietet genug Platz für detaillierte Ausführungen zu Ihrem Produkt. Nehmen Sie sich etwas Zeit, um hier einen ansprechenden und informativen Text zu formulieren, bevor Sie fortfahren. Der Editor bietet Ihnen grundlegende Möglichkeiten, die Produktbeschreibung zu formatieren, die Ausrichtung anzupassen und zum Beispiel die Schriftstärke zu verändern.

Unterhalb des Editors für die Produktbeschreibung finden Sie eine Meta-Box mit verschiedenen Einstellungsmöglichkeiten rund um

die verschiedenen Produktdaten.

In den meisten Fällen werden Sie mit dem Produkttyp Einfaches Produkt arbeiten. Mit einer Reihe von Checkboxen können Sie dem Produkt verschiedene Eigenschaften zuschreiben. Ist es vielleicht Virtuell und Herunterladbar, oder handelt es sich um eine Dienstleistung? In diesen Fällen würde etwa die Logik zur Berechnung von Versandkosten nicht zum Einsatz kommen.

Andere Mechanismen kommen dagegen zum Einsatz, wenn es sich um eine Gebrauchtware, ein Mängelexemplar oder ein Produkt handelt, das Differentialbesteuert ist. Diese Optionen werden Sie vermutlich nicht auswählen müssen, wenn es sich bei Ihren Produkten um (hoffentlich) mängelfreie Neuware handelt.



Abbildung 12.8 Die Meta-Box unterhalb des Produkt-Editors enthält die Kontrollelemente für alle wichtigen Metadaten rund um Ihr Produkt.

Wenn wir diese erste Zeile der Meta-Box hinter uns lassen, können Sie auf der linken Seite in der Box eine Navigationsleiste mit verschiedenen Punkten finden. Die Punkte können sich je nach eingesetzter Kombination aus Plugins und WooCommerce-Erweiterungen unterscheiden.

- Der erste Punkt, Allgemein, sollte bereits geöffnet sein und präsentiert Ihnen in der Produktdaten-Meta-Box Einstellungen rund um Preise und Preisauszeichnungen. Mindestens das Feld Regulärer Preis sollten Sie ausfüllen, damit Ihr neu angelegtes Produkt auch gekauft werden kann. Sollten Sie sich in der Zukunft umentscheiden und Ihr Produkt zu einem anderen Preis

anbieten wollen, können Sie entweder den regulären Preis bearbeiten oder, sollte es sich um eine Preissenkung handeln, das Feld Angebotspreis bemühen, um die Preissenkung in Ihrem Shop werbewirksam neben dem durchgestrichenen regulären Preis anzuzeigen.

- Der zweite Punkt auf der linken Seite ist der Lagerbestand. Wählen Sie diesen Punkt aus, ändern sich die Kontrollelemente auf der rechten Seite der Box. Nun haben Sie die Möglichkeit, eine Artikelnummer zu vergeben, um das Produkt eindeutig identifizierbar zu machen. Außerdem können Sie hier bestimmen, wie WooCommerce mit Ihrem Lagerbestand umgehen soll. In der einfachsten Version ist die Checkbox Lagerbestand verwalten? nicht ausgewählt. In diesem Fall können Sie mit der Einstellung Lagerbestand einfach die Lieferbarkeit des Produkts selbst angeben, ohne eine detaillierte Lagerhaltung bemühen zu müssen. Wenn Sie für dieses Produkt Vorrätig auswählen, wird es Ihren Kundinnen und Kunden als lieferbar angezeigt werden, wenn Sie Nicht Vorrätig oder Auf Lieferrückstand auswählen, wird auch das in Ihrem Shop entsprechend dargestellt. Sollten Sie auf eine detaillierte Lagerhaltung angewiesen sein, setzen Sie einen Haken in der Checkbox Lagerbestand verwalten? und können in den anschließend eingeblendeten Feldern die Lagerbestandsmenge angeben. Sie müssen außerdem entscheiden, ob Lieferrückstände erlaubt werden und ob das Produkt damit auch noch verkauft werden soll, wenn es nicht mehr vorrätig ist. Und Sie können mit Schwellwert für »geringer Lagerbestand« bestimmen, ab wann im Shop angezeigt werden soll, dass der Lagerbestand langsam zur Neige geht.

Hinweis: Nutzung von Warenwirtschaftssoftware

Wie bereits weiter oben in diesem Kapitel erwähnt, gibt es eine Reihe von Warenwirtschaftssystemen, die an WooCommerce angebunden werden. Während die Verwaltung des Lagerbestands in WooCommerce für kleine bis mittlere Shops problemlos funktioniert und nur einer gelegentlichen Inventur bedarf, um nicht dramatisch vom tatsächlichen Lager abzuweichen, stößt sie doch irgendwann an ihre Grenzen.

Sollten Sie also einen Shop mit vielen verschiedenen Produkten oder vielen Produktvarianten (mein persönlicher Albtraum waren T-Shirts in verschiedenen Farben und Größen) planen, denken Sie frühzeitig über den Einsatz einer echten Warenwirtschaft nach, und prüfen Sie, welche gut mit WooCommerce genutzt werden kann und zu Ihren Anforderungen passt.

- Versand ist der dritte Punkt auf der linken Seite. Wenn Sie Ihr Produkt vorher als virtuell deklariert haben, wird dieser Punkt nicht angezeigt, weil sich virtuelle Produkte nur schwer in Päckchen stecken lassen. Hier können Sie Abmessung und Gewicht des Produkts angeben. Diese Informationen können genutzt werden, um die Versandkosten zu berechnen. In vielen Fällen ist eine Regelung der Versandkosten über das Feld Versandklasse aber deutlich einfacher. Aktuell können Sie hier noch keine Klasse auswählen, wir kommen später zurück zu diesem Punkt, nachdem wir uns in Abschnitt 12.2.6 dem Einrichten von Versandklassen gewidmet haben.
- Unter dem Punkt Verlinkte Produkte finden Sie eine pfiffige Lösung, sogenannte *Cross*- und *Upsell*-Produkte zu definieren. Beim Anlegen des ersten Produkts in Ihrem neuen Shop ist diese Sektion noch nicht sinnvoll nutzbar, wenn Sie Ihre virtuellen Regale aber erst mal befüllt haben, können Sie die Produktnamen anderer Produkte in die beiden Felder eingeben. Auf diesem Weg

können Sie Ihren Kunden Zusatzverkäufe schmackhaft machen – etwa um Interessenten der Häkeldecke für ein etwas teureres Modell mit ausgefallenem Muster zu begeistern oder um Querverkäufe passend zum ausgewählten Produkt zu empfehlen – wie wäre es etwa mit einem passenden Tischset zu dieser Häkeldecke?

- Eigenschaften ist der vorletzte Punkt auf der linken Seite. Hier haben Sie die Möglichkeit, über die Felder hinauszugehen, die WooCommerce Ihnen standardmäßig zur Verfügung stellt. Sie können an dieser Stelle eigene Produkteigenschaften erstellen und Ihrem Produkt zuweisen. Beispielhaft sind hier Marke und Farbe (Color) als Eigenschaften eingetragen. Sind bereits Werte (Begriffe) für eine Eigenschaft definiert, können Sie diese durch einfaches Tippen in das Wert(e)-Feld einfügen. Möchten Sie zusätzliche Werte hinzufügen, klicken Sie auf den Button Hinzufügen und geben den Namen Ihres gewünschten neuen Werts an.



Abbildung 12.9 In der Übersicht zu den Produkteigenschaften finden Sie alle gespeicherten Begriffe und können sie bearbeiten.

- Sollten Sie diese Daten später für alle Produkte in Ihrem Shop verwalten wollen, finden Sie im Admin-Menü den Punkt Produkte • Eigenschaften. Dort werden alle angelegten Eigenschaften aufgelistet, können bearbeitet und gelöscht werden. Produkteigenschaften eignen sich hervorragend zum Beispiel zur Größenangabe von Kleidung und können für variable Produkte auch von Ihren Kundinnen und Kunden im Bestellprozess ausgewählt werden.

- Der letzte Punkt in dieser Box ist Erweitert. Hier finden Sie eine Reihe von Funktionen, für die sich kein passenderer Platz finden wollte. Hier können Sie einen Hinweis zum Kauf angeben, der Ihren Kunden nach dem Kauf des Produkts angezeigt wird. Außerdem können Sie eine Menüreihenfolge festlegen. Diese Einstellung können Sie in der Praxis aber getrost ignorieren, WordPress hat schon seit Jahren ein ausgefeilteres Menüsystem. Darüber hinaus könnten Sie hier die standardmäßig aktive Möglichkeit für Ihre Kunden, Produktbewertungen für dieses Produkt zu hinterlassen, deaktivieren.

Unterhalb der Produktdaten-Meta-Box finden Sie die Produkt Kurzbeschreibung. Der Name ist hier Programm. Die meisten Themes zeigen diese Kurzbeschreibung unterhalb des Produktnamens oder unter dem Preis an. Beschränken Sie sich daher auf zwei bis drei Sätze und verzichten Sie eher auf das Einfügen von Bildern, dazu kommen wir sowieso in einem der nächsten Schritte.

Mit der Warenkorb Kurzbeschreibung verhält es sich ganz ähnlich. Sie ist im Editor das nächste Feld unterhalb der Kurzbeschreibung. Auch hier sollten Sie sich kurzfassen, damit die Warenkorb-Übersicht, in der diese Beschreibung angezeigt werden kann, kompakt bleibt.

Werfen wir als Nächstes einen Blick auf die Seitenleiste am rechten Rand des Produkt-Editors. Die oberste Box entspricht im Umfang dem, was Sie im Editor für Beiträge und Seiten auch schon kennengelernt haben. Hier können Sie das Produkt sofort veröffentlichen, einen Entwurf speichern oder eine Vorschau betrachten. Sollten Sie ein bestehendes Produkt als Vorlage für weitere Produkte nutzen wollen, finden Sie hier außerdem den praktischen Link In neuen Entwurf kopieren.

Auch die beiden folgenden Boxen, Produkt Kategorien und Produkt Schlagwörter, verhalten sich wie Kategorien und Schlagwörter in Beiträgen. Hier können Sie existierende Kategorien und Schlagwörter einsetzen, um Ihr Sortiment besser zu strukturieren.

In den meisten Fällen lebt Ihr Shop von visuellen Elementen. Daher sollten Sie die folgende Box Produktbild auf keinen Fall vernachlässigen. Das Bild, das Sie hier auswählen, wird von WooCommerce und Ihrem Theme an den verschiedensten Orten angezeigt. Es ist in der Regel in Produktauflistungen im Shop sichtbar, kann auf der Produktseite im Detail betrachtet werden und findet sich auch in vielen Warenkorbdesigns sowie in der Bestellübersicht an der Kasse wieder. Wenn für ein Produkt bereits ein Produktbild hinterlegt wurde, wird es in dieser Box angezeigt. Durch einen Klick auf das Bild gelangen Sie in die gewohnte Medienverwaltung, die Sie auch für Beiträge und Seiten nutzen.

Ganz ähnlich verhält sich die letzte Box in der Seitenleiste: Produktgalerie. Hier können mehrere Produktbilder eingefügt werden. Beachten Sie aber, dass nicht alle Themes diese Galerie auch tatsächlich anzeigen.

Vergewissern Sie sich noch einmal, dass Sie alle Felder korrekt ausgefüllt haben, und klicken Sie abschließend auf den großen Veröffentlichen-Button oben in der Seitenleiste, um Ihr erstes Produkt in den Shop aufzunehmen.

Wechseln Sie nach dem Veröffentlichen im Admin-Menü auf der linken Seite in Produkte • Alle Produkte, sollten Sie Ihr eben angelegtes Produkt in einer Listenansicht mit einigen Details sehen. Wenn Sie diese Arbeitsschritte einige Male wiederholen, um alle Produkte anzulegen, die Sie anbieten wollen, wird sich diese Liste schnell mit Leben füllen.

Bevor Sie zur Tat schreiten können und einen ersten Testkauf in Ihrem Shop durchführen, müssen wir uns nur noch den letzten beiden Punkten der Konfiguration widmen.

12.2.5 Erste Zahlungsmethoden einrichten

Das Entgegennehmen von Zahlungen ist ein zentraler Bestandteil der meisten Online-Shops. Wie bereits beschrieben, stehen Ihnen in WooCommerce die verschiedensten *Payment-Gateways* zur Verfügung. Je nach Zielgruppe Ihres Shops lohnt sich hier ein genauerer Blick auch auf die Alternativen, die nicht in diesem Kapitel vorgestellt werden.

Die gute Nachricht ist, dass Sie sich nicht für eine einzige Bezahlmethode entscheiden müssen. WooCommerce bietet Ihnen die Möglichkeit, beliebige Methoden parallel zu nutzen und Ihre Kundinnen und Kunden entscheiden zu lassen, wie sie bezahlen möchten. Weil die Anbindung vieler verschiedener Zahlungsmethoden nicht nur einen erheblichen organisatorischen Aufwand bedeutet, sondern eine zu große Auswahlmöglichkeit Ihre Kund*innen auch schnell überfordern kann, sollten Sie sich für den Moment auf maximal zwei bis drei verschiedene Methoden beschränken. Sie können hier später zu jeder Zeit Anpassungen vornehmen. Zahlungsmethoden, die nicht genutzt werden oder für Sie als Verkäufer*in unrentabel sind, können deaktiviert oder durch andere ersetzt werden.

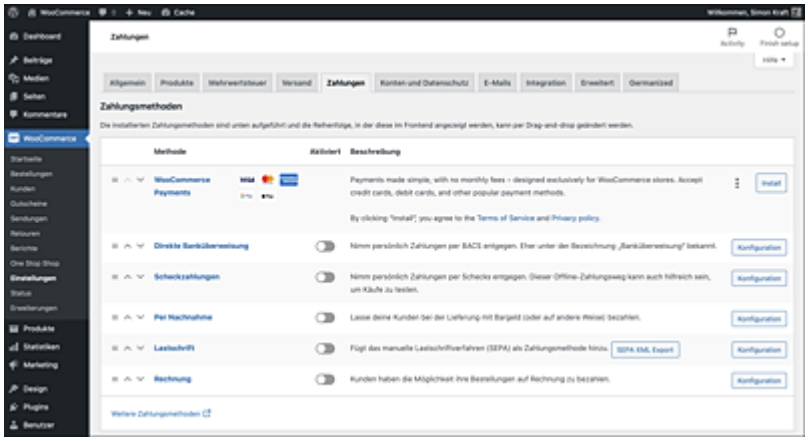


Abbildung 12.10 In einer kompakten Übersicht sehen Sie, welche Zahlungsmethoden in Ihrem Shop aktuell zur Verfügung stehen und welche tatsächlich aktiviert sind.

Hinweis: Augen auf bei Kosten für Zahlungsdienste

Wie im stationären Handel sind auch bei Online-Shops mit der Abwicklung von Zahlungen Kosten verbunden. Selbst wenn Payment-Gateways, die WooCommerce-Erweiterungen zur Anbindung externer Dienste, oft kostenfrei nutzbar sind: Die Dienste an sich erheben in der Regel Gebühren.

Abgesehen einigen wenigen Diensten mit einer festen (monatlichen) Grundgebühr, ist es vor allem üblich, einen fixen Betrag pro Transaktion sowie teilweise auch einen prozentualen Anteil am Gesamtbetrag zu erheben. Besonders bei vielen kleinen Beträgen oder in bestimmten Konstellationen von Zahlungsdienst und Währung können die Kosten erheblich ins Gewicht fallen.

Neben der einfachen Einbindung in WooCommerce und der Akzeptanz einer Zahlungsmethode in Ihrer Zielgruppe sollten Sie daher auch die Kosten, die Ihnen entstehen, in Ihre Entscheidungen mit einfließen lassen. Studieren Sie die Konditionen der verschiedenen Anbieter eingehend, bevor Sie sich entscheiden.

Die Einstellungsseite für den Zahlungsmethoden finden Sie über das Admin-Menü unter WooCommerce • Einstellungen und dort im Register Zahlungen. Auch wenn die Tabelle auf dieser Seite auf den ersten Blick umfangreich wirkt, sehen Sie an den Schiebereglern in der Spalte Aktiviert, dass aktuell keine der Methoden aktiv ist. Werfen wir einen kurzen Blick auf die Methoden, die WooCommerce von Haus aus unterstützt.

- WooPayments: Der Name dieser Erweiterung ist Ihnen vielleicht aus der WooCommerce-Einrichtung noch ein Begriff. In Abschnitt 12.2.1 wurde Ihnen diese hauseigene Lösung von Automattic, dem Unternehmen hinter WooCommerce, bereits angepriesen. Die Erweiterung bindet Möglichkeiten zum Zahlen mit Kreditkarte, Apple Pay und Google Pay an. Haben Sie die Erweiterung bei der Einrichtung bewusst nicht mit installiert, handelt es sich auch hier lediglich um Werbung, die Sie mit einem Klick auf die drei Punkte und dann auf Ausblenden entfernen können. Alternativ können Sie das Plugin hier installieren und konfigurieren.
- Direkte Banküberweisung: Hierbei, wie bei den folgenden Methoden, handelt es sich nun tatsächlich um Funktionen von WooCommerce selbst.

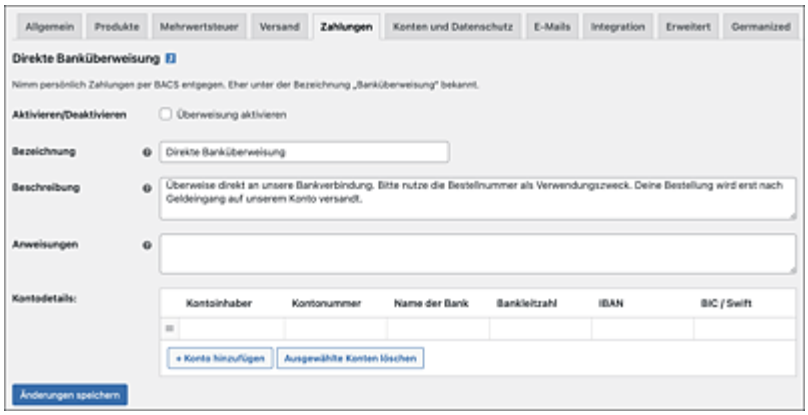


Abbildung 12.11 Konfigurieren können Sie das Überweisungs-Gateway direkt in WooCommerce, um die Abwicklung von eingehenden Zahlungen müssen Sie sich später aber selbst kümmern.

Zahlung per Überweisung (gern auch Vorkasse genannt) hat für Sie als Verkäufer*in den Vorteil, dass Sie Ihre Ware erst versenden, sobald Sie das Geld Ihrer Käuferin oder des Käufers erhalten haben. Außerdem wird eine Zahlung per Überweisung für Sie in vielen Fällen die kostengünstigste Option sein. Weil Überweisungen in vielen Fällen nicht in Echtzeit durchgeführt werden und Sie jede eingehende Zahlung manuell in WooCommerce als eingegangen markieren müssen, kann es bei dieser Zahlungsmethode aber auch zu einigen Verzögerungen zwischen Bestellung und Versand der Ware kommen.

Möchten Sie Überweisungen für Ihren Shop nutzen, klicken Sie in der Tabelle auf den Button Konfiguration in der entsprechenden Zeile.

Auf einer neuen Seite öffnen sich die Einstellungen. Setzen Sie zunächst einen Haken bei Aktivieren/Deaktivieren, um die Methode zu aktivieren. Die Felder Bezeichnung und Beschreibung enthalten bereits vorbereitete Texte. Diese können Sie nach Belieben anpassen und im Feld Anweisungen noch weiter ergänzen. All diese Informationen werden Ihren Kundinnen und Kunden auf der Kassenseite bei der Auswahl einer Zahlungsmethode angezeigt. Unter Kontodetails sollten Sie mindestens eine Bankverbindung angeben, die anschließend für Überweisungen genutzt werden soll. Mit einem Klick auf den Tab Zahlungen gelangen Sie zurück zur vorherigen Übersicht.

- Scheckzahlung: Diese Zahlungsmethode kann nützlicher sein, als Sie auf den ersten Blick vielleicht vermuten würden. Eigentlich für die Annahme von Schecks gedacht, können Sie diese

Zahlungsmethode auch wunderbar zweckentfremden, wenn Sie statt Schecks lieber Bargeld von Ihren Kund*innen nehmen möchten, wenn diese die bestellte Ware vor Ort abholen. Um Scheckzahlung oder Barzahlung vor Ort für Ihren Shop zu nutzen, klicken Sie in der Tabelle auf den Button Konfiguration in der entsprechenden Zeile. Wie bei der direkten Banküberweisung können Sie auch bei Scheckzahlung eigenen Text für Bezeichnung, Beschreibung und Anweisungen angeben. Wenn Sie die Bezeichnung der Methode zu »Barzahlung bei Abholung« anpassen und in den anderen beiden Feldern erklären, wie die Abholung funktioniert und wo abgeholt werden kann, haben Sie eine praktische zusätzliche Option. Natürlich müssen Sie auch hier einen Haken bei Aktivieren/Deaktivieren setzen, um die Methode zu aktivieren.

- Per Nachnahme: Wenn Ihre Kundinnen und Kunden direkt bei der Postbotin oder dem Paketboten bezahlen können sollen, ist die Zahlung per Nachnahme eine attraktive Option. Das Geld, das auf diesem Weg von Ihrem Logistikpartner für Sie eingesammelt wird, landet in der Regel per Überweisung auf Ihrem Konto, und Sie können sicher sein, dass Ihre Ware erst nach erfolgter Bezahlung übergeben wird. Dieser Service hat aber seinen Preis. Klicken Sie auf den Button Konfiguration, um die Einstellungen der Nachnahmezahlung zu bearbeiten und die Zahlungsmethode zu aktivieren.

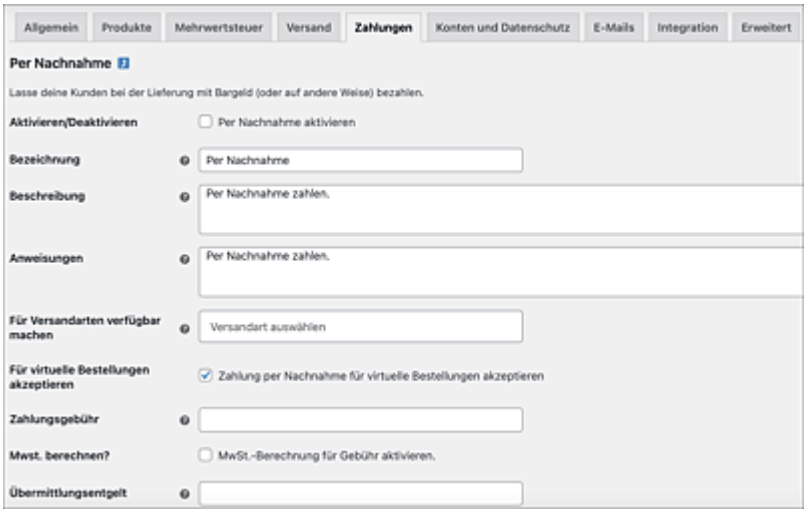


Abbildung 12.12 Vor allem im deutschsprachigen Raum ist das Zahlen per Nachnahme über den Versanddienstleister immer noch ein gängiger Weg.

Neben den grundlegenden Einstellungen, die Sie bei der direkten Überweisung bereits kennengelernt haben, können Sie mit der Auswahl Für Versandarten verfügbar machen Zahlungen per Nachnahme auf bestimmte Versandarten einschränken. Das ist etwa dann nützlich, wenn nicht alle ihre Versanddienstleister einen solchen Service anbieten. In Abschnitt 12.2.6 widmen wir uns den Einstellungen von Versandarten und -methoden. Um diese Zahlungsmethode abschließend zu konfigurieren, müssen Sie nach dem Bearbeiten des nächsten Abschnitts noch einmal hierher zurückkehren. Zusätzlich können Sie entscheiden, ob Sie Zahlungen per Nachnahme Für virtuelle Bestellungen akzeptieren, können eine zusätzliche Zahlungsgebühr festlegen und Mehrwertsteuerberechnung sowie ein gesondertes Übermittlungsentgelt festlegen.

Wenn Sie in Abschnitt 12.2.3 das Plugin Germanized für WooCommerce nicht installiert haben, waren das alle Zahlungsmethoden, die Ihnen im Moment zur Verfügung stehen. Bevor wir betrachten, wie Sie zusätzliche Payment-Gateways

installieren, werfen wir noch einen Blick auf die beiden Methoden, die das Germanized-Plugin mitbringt.

- Lastschrift: Wenn Sie über Ihre Bank die Möglichkeit haben, einen Lastschrifteinzug zu veranlassen, könnte diese Zahlungsmethode für Sie infrage kommen. Ähnlich wie bei Überweisungen sind auch hier die Gebühren bei den meisten Banken überschaubar. Anders als bei Überweisungen sind Sie aber nicht auf die Disziplin Ihrer Kundinnen und Kunden angewiesen. Stattdessen ist es an Ihnen, ausgestellte Lastschriftmandate zu nutzen, um Geld einzuziehen. Wenn Sie Germanized für WooCommerce installiert und aktiviert haben, finden Sie den Eintrag Lastschrift in der Liste der möglichen Bezahlmethoden und können auf den Button Konfiguration klicken, um die Einstellungen zu bearbeiten und die Zahlungsmethode zu aktivieren. Auch die Seite, auf der Sie an dieser Stelle landen, fängt mit der Grundkonfiguration an, die Sie bei der Überweisungszahlungsmethode bereits kennengelernt haben. Zusätzlich müssen Sie Ihre Daten ins Feld Gläubiger Informationen eintragen und die Bankverbindung angeben. Die weiteren Felder müssen Sie nicht bearbeiten, können aber den Text, der für das Lastschriftmandat generiert wird, im entsprechenden Textfeld anpassen.
- Rechnung: Der Kauf auf Rechnung ist in Deutschland keine Seltenheit. Und mit dieser recht grundlegenden Bezahlmethode können Sie Ihren Kunden ebenfalls die Zahlung auf Rechnung anbieten. Auch hier bringt Sie ein Klick auf den Button Konfiguration in die Einstellungen. Neben den Optionen, die Sie in den Einstellungen der direkten Banküberweisung bereits kennengelernt haben, finden Sie hier wieder einige Besonderheiten. So können Sie den Bestellstatus festlegen, den eine Bestellung haben soll, nachdem die Bestellung aufgegeben wurde. Jede Bestellung, die in Ihrem Shop eingeht, hat einen

Status, um zum Beispiel den Bearbeitungszustand, ausstehende Zahlungen oder eine Stornierung zu kennzeichnen. Weil bei dieser Lösung für den Kauf auf Rechnung keine Bonitätsprüfung stattfindet, sind die Optionen Registrierte Kunden und Kunden eingrenzen besonders nützlich. Sie erlauben es, diese Zahlungsmethode nur angemeldeten Benutzer*innen anzubieten, und sorgen damit etwas mehr für Ihre Sicherheit. Sollten Bezahlmethoden mit Bonitätsprüfung für Sie interessant sein, finden Sie auch dafür passende Payment-Gateways von Drittanbietern. Oben auf der aktuellen Einstellungsseite ist Ihnen vielleicht der Hinweis des Plugins Germanized für WooCommerce für dessen Bezahlversion aufgefallen. Sollten Sie sich entscheiden, hierfür Geld in die Hand zu nehmen, übernimmt das Plugin sogar das Generieren und Versenden von PDF-Rechnungen für Sie und macht diese Option damit noch etwas runder.

Nach dem Speichern kehren Sie zur Übersicht der verschiedenen Zahlungsmethoden zurück. Wenn Sie bereits eine oder sogar mehrere der verfügbaren Methoden aktiviert haben, sehen Sie deren Aktivierung auch hier in der Tabelle. Mit den Pfeilen auf der linken Seite können Sie die Reihenfolge der verschiedenen Zahlungsmethoden anpassen. Bringen Sie sie einfach mittels Drag-and-drop in die Reihenfolge, in der sie auch Ihren Kundinnen und Kunden angezeigt werden sollen.

Wenn Sie weitere Zahlungsmethoden hinzufügen möchten, finden Sie am unteren Ende der Tabelle links den Link Weitere Zahlungsmethoden. Ein Klick auf diesen Link bringt Sie auf den WooCommerce-eigenen Marktplatz. Hier können Sie die verschiedensten Erweiterungen finden und installieren. Den Marktplatz erreichen Sie alternativ auch über das Admin-Menü WooCommerce • Erweiterungen. Wenn Sie von der Übersichtsseite der

Zahlungsmethoden kommen, sind im Marktplatz eben diese bereits vorausgewählt.

Aufgrund der großen Verbreitung PayPals und dessen einfacher Nutzung bietet sich PayPal Payments an, Klarna und dessen Marke Sofort sind als in Europa interessant, aber darüber hinaus gibt es unzählige weitere Anbieter und Dienste zu entdecken.

12.2.6 Versand leicht gemacht mit Versandklassen

Gehen wir das letzte große Thema an, bevor Sie mit Ihrem Shop durchstarten können: den Versand. Diesen Abschnitt können Sie selbstverständlich überspringen, wenn sich Ihr Shop ausschließlich auf den Verkauf digitaler Produkt beschränkt und Sie niemals in die Verlegenheit kommen, physische Güter von A nach B transportieren (lassen) zu müssen.

Sie werden im WooCommerce-Marktplatz einige Erweiterungen finden, die die Versandlogistiken, mit denen WooCommerce von Haus aus gesegnet ist, komplett ersetzen oder zumindest so sehr erweitern, dass Sie sie fast nicht wiedererkennen werden. Auch wenn mit den Erweiterungen teilweise wirklich komplexe Lösungen möglich sind, wollen wir es für dieses bespiel trotzdem weiter praxisnah und einfach halten und beschränken uns auf den Standardfunktionsumfang von WooCommerce.

Navigieren Sie zunächst zu WooCommerce • Einstellungen und von dort weiter ins Register Versand. Hier können Sie in einem ersten Schritt verschiedene Versandzonen definieren. Über diese Zonen können Sie im nächsten Schritt bestimmen, wo welche Versandmethoden zur Verfügung stehen sollen. Wollten Sie es ganz einfach halten, kämen Sie in vielen Fällen vermutlich mit einer einzigen Zone zurecht, nennen wir sie »National«. Für dieses

Beispiel, und um die Möglichkeiten hinter Versandzonen etwas zu verdeutlichen, entscheiden wir uns aber für drei Zonen:

Lokal wird unsere Zone für Kund*innen, die in unmittelbarer Nähe zu unserem Geschäft wohnen. Sie sollen ihre Lieferungen neben dem normalen Versand per Paketdienstleister auch selbst abholen können und zusätzlich eine Same-Day-Delivery-Option per Fahrradkurier zur Verfügung haben.

National ist die Zone für alle anderen Kund*innen unseres Online-Shops, die ihre Bestellungen innerhalb des Landes empfangen möchten. Hier werden wir uns auf einfachen Paketversand beschränken. In der Praxis könnten Sie aber auch hier selbstverständlich verschiedene Methoden anbieten und zum Beispiel auf die Angebote diverser Logistikpartner zurückgreifen.

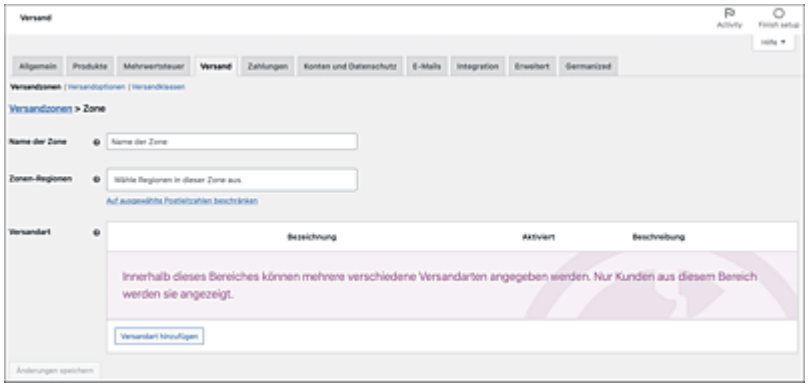


Abbildung 12.13 Als mehr oder weniger leere Seite präsentiert sich die Versandzonenverwaltung bei Ihrem ersten Besuch. Doch mit dieser unscheinbaren Funktion können Sie auch komplexere Versandszenarien abdecken.

EU ist der logische nächste Schritt, wenn Sie sich nicht auf die einheimische Kundschaft beschränken wollen. Folglich landen in dieser Zone alle Kund*innen, die sich innerhalb der EU, aber außerhalb unseres eigenen Landes (in unserem Beispiel: außerhalb Deutschlands) aufhalten.

Klicken Sie auf den Button Versandzone hinzufügen. Auf der nun aufgerufenen Seite können Sie einen internen Namen für die neue Zone in das Feld Name der Zone eintragen. Um die Ausdehnung der Zone festzulegen, nutzen Sie das Feld Zonen-Regionen, um ein einzelnes Land (Deutschland), Bundesland (Baden-Württemberg), eine einzelne Region (Europa) oder mehrere Länder, Bundesländer oder Regionen zu einer Zone zusammenzuführen. Ein Klick auf Auf ausgewählte Postleitzahlen beschränken stellt Ihnen ein zusätzliches Eingabefeld zur Verfügung, in dem Sie Postleitzahlen oder Postleitzahlenbereiche angeben können, um die Auswahl weiter zu verfeinern.

Unter Versandart wird Ihnen eine Tabelle der eingerichteten Versandarten angezeigt – bei der ersten Einrichtung ist sie vermutlich leer. Ein Klick auf den Button Versandart hinzufügen öffnet ein modales Fenster, mit dessen Hilfe Sie zusätzliche Versandmethoden und -tarife angeben können. Dabei haben Sie die Wahl zwischen den drei Optionen Versandkostenpauschale, Kostenlose Lieferung und Abholung vor Ort. Fügen Sie eine oder mehrere dieser Optionen hinzu, um Sie in der aktuellen Versandzone zu aktivieren. Nach dem Hinzufügen der neuen Versandarten sehen Sie diese in der Übersichtstabelle. Aus dieser Tabelle heraus können Sie die Parameter der einzelnen Versandarten steuern, Preise festlegen oder sie einzeln deaktivieren. Wenn Sie Ihre Maus über die Tabelle bewegen, sehen Sie einen Link Bearbeiten unter dem Titel der Versandmethode. Ein Klick auf diesen Link öffnet ein weiteres modales Fenster. Hier können Sie die jeweilige Methode benennen. Für das Beispiel Lokal haben Sie die Versandmethode Versandkostenpauschale hinzugefügt und können diese nun in Methodenbezeichnung zum Beispiel in »Fahrradkurier« umbenennen. Wiederholen Sie diesen Schritt, bis alle Versandarten, die Sie in Ihrem Shop für diese Zone anbieten wollen, angelegt sind. Den

Sonderfall des lokalen Versands durch einen ortsansässigen Kurierdienst einmal außen vor gelassen: Haben Sie sich bereits für einen Versanddienstleister entschieden? Informieren Sie sich bei den einschlägigen Anbietern, vergleichen Sie Konditionen, Preise, aber auch die Reputation der verschiedenen Marktteilnehmer und überprüfen Sie ebenfalls, ob es WooCommerce-Erweiterungen gibt, die Ihnen die Arbeit in diesem Schritt abnehmen können.



Abbildung 12.14 Mit einigen Anpassungen ist die pauschale Versandmethode erstaunlich flexibel einsetzbar.

Wenn Sie, wie oben beschrieben, mehrere Zonen nutzen möchten, können Sie den beschriebenen Prozess so oft wie nötig wiederholen. Auch bei diesem einfachen Ansatz ist das System erstaunlich flexibel und kann an verschiedene Bedürfnisse angepasst werden.

Sollten Sie Anforderungen haben, die über einfache Versandzonen nicht gelöst werden können – etwa zusätzlich Sendungsverfolgung anbieten, Barcodes generieren oder detaillierte Packlisten für jede Bestellung anlegen wollen –, gelangen Sie über den Link Weitere Optionen anzeigen unten auf der Seite mit den Versandzonen wieder direkt auf den WooCommerce-Marktplatz.

12.3 Letzter Schliff und Testkauf

Mit dem fertig eingerichteten Versand (oder für rein virtuelle Produkte auch ohne Versand) ist Ihr Shop nun im Grunde startbereit. Aber lassen Sie uns noch einen kurzen Blick auf Ihren Shop werfen, bevor Sie sich aufmachen, Ihre erste Testbestellung abzusetzen. Wenn Sie das Storefront-Theme oder ein anderes WooCommerce-Theme einsetzen, werden Sie auf der Startseite Ihres Shops eine Übersicht Ihrer Produkte finden. Außerdem hat WooCommerce direkt nach seiner Installation eine Reihe von Seiten angelegt und jeweils mit einem passenden Block ausgestattet, der beim Aufrufen Inhalte wie Warenkorb, Produktübersicht oder eine Kassenseite anzeigen kann. Einige dieser Seiten wollen Sie vermutlich dem Menü Ihrer Website hinzufügen, um Ihren Kundinnen und Kunden das Navigieren durch Ihren Shop zu erleichtern.

Jetzt wird es endlich Zeit für einen Testlauf. Um mit einem möglichst realitätsnahen Zustand zu starten, können Sie nun entweder in einen anderen Browser wechseln, in dem Sie nicht bei Ihrem Shop eingeloggt sind, oder Sie öffnen ein privates Fenster Ihres Browsers. Je nachdem, wie Sie Ihren Shop zuvor aufgebaut haben, sind hier durchaus mehrere Durchläufe sinnvoll, grundsätzlich ist auf diese Punkte zu achten:

- Shop-Darstellung und Nutzbarkeit: Sind alle wichtigen Teile des Shops tatsächlich erreichbar und funktionieren richtig? Behalten Sie dabei im Auge, welche Aktionen Ihre Kundinnen und Kunden durchführen müssen, welche Informationen sie interessieren könnten und wo Sie sie vermuten werden.

- Kontoerstellung und E-Mail-Versand: Wenn Sie bei Ihrem Shop das Anlegen von Benutzerkonten erlauben, sollten Sie überprüfen, ob dieser Prozess auch wirklich so abläuft, wie Sie es sich vorgestellt haben. Was passiert zum Beispiel, wenn eine falsche E-Mail-Adresse angegeben wird? Und da wir schon bei E-Mails sind: Werden diese korrekt zugestellt, und sehen sie so aus, wie Sie es sich vorgestellt haben?
- Payment-Gateways und gegebenenfalls Versandschnittstellen: Viele Zahlungsdienstleister bieten einen Sandkastenmodus an, in dem Sie testweise Bestellungen abwickeln können, ohne wirkliche Transaktionen zu verursachen. Sollte das in Ihrem konkreten Setup nicht funktionieren oder zu kompliziert sein, lohnt es sich dennoch, einmal eine Bestellung (pro Payment-Gateway) durchzuspielen, um zu prüfen, ob wirklich alles funktioniert. Wenn Sie keine größeren Beträge herumschieben wollen, können Sie hierfür auch ein 1-€-Testprodukt anlegen.
- Wenn Sie den Versand Ihrer Waren über einen externen Dienst automatisiert haben, sollten Sie auch diese Schnittstellen einmal testen.
- Eingang von Bestellungen im WooCommerce-Backend: Und zu guter Letzt die Probe auf Exempel: Bestellungen, die Sie im Rahmen Ihres Tests getätigt haben, sollten, wenn Sie zurückkehren in den Browser oder das Fenster, in dem Sie in WordPress angemeldet sind, auch im WooCommerce-Backend auftauchen. Werfen Sie dafür einen Blick in WooCommerce • Bestellungen.

Sollten im Rahmen dieser Tests kleine Fehler auftreten, ist das ganz normal. Die meisten Shops sind im ersten Wurf nicht perfekt und bedürfen daher etwas Nachbesserung. Noch wichtiger als diese wenigen Handgriffe, die Sie noch erledigen müssen, um Fehler zu

beseitigen, ist die dauerhafte Aufmerksamkeit und Pflege, die ein Shop braucht. Planen Sie regelmäßig ein paar Stunden Zeit ein, um Funktionstests und Aktualisierungen durchzuführen. So halten Sie alles auf dem neuesten und sichersten Stand und stellen sicher, dass Sie den Shop, in dessen Einrichtung Sie viel Zeit und vielleicht auch Geld investiert haben, lange nutzen können.

Sie sollten jetzt einen voll funktionsfähigen Shop Ihr Eigen nennen können. Weil es für WooCommerce aber deutlich mehr zu erfahren gibt, als in ein einziges Kapitel passt, möchte ich Sie nicht ohne zumindest ein paar Empfehlungen für interessante Erweiterungen verlassen.

12.4 Zusätzliche Erweiterungen für Ihren Shop

Wie schon einige Male im Laufe dieses Kapitels angedeutet, ist der Markt für WooCommerce-Erweiterungen groß. Neben den vielen verfügbaren und meist kostenpflichtigen Lösungen im offiziellen Marktplatz gibt es einige nützliche Erweiterungen auch im Plugin-Verzeichnis auf WordPress.org und viele weitere auf den Plattformen der jeweiligen Anbieter*innen. Hier nur ein kleiner Auszug:

12.4.1 Geld entgegennehmen mit »PayPal Payments for WooCommerce«

Lange Zeit gehörte PayPal zum Lieferumfang von WooCommerce. Heute müssen Sie eine (kostenlose) Erweiterung suchen und installieren, um es zu nutzen. Doch der zusätzliche Schritt lohnt sich immer noch, gehört PayPal doch zu den beliebtesten und verbreitetsten Online-Zahlmethoden.

Plugin	PayPal Payments for WooCommerce
Download	*https://woocommerce.com/de/products/woocommerce-paypal-payments/*
Komplexität	+

12.4.2 Produkt-Abos mit »WooCommerce Subscriptions«

Abo-Produkte sind nicht nur für Streaming-Plattformen ein interessantes Format. Auf Basis wiederkehrender Zahlungen lassen

sich die verschiedensten Geschäftsmodelle aufbauen. Ganz gleich also, ob Sie WordPress-Plugins, monatlich neue Kuchenrezepte oder automatisch vor die Tür gelieferte Zahnbürstenköpfe verkaufen wollen, das WooCommerce-Subscriptions-Plugin ist einen Blick wert.

Plugin	WooCommerce Subscriptions
Download	*https://woocommerce.com/de/products/woocommerce-subscriptions/*
Komplexität	++

12.4.3 Komplex konfigurierbare Produkte mit »Composite Products for WooCommerce«

WooCommerce kann out of the box variable Produkte verwalten. Das ist perfekt, wenn Sie ein T-Shirt in verschiedenen Farben und Größen anbieten wollen. Für komplexere Produkte ist Composite Products eine spannende Wahl. Es erlaubt das Anlegen von Kits, die aus verschiedenen von Ihren Kund*innen wählbaren Komponenten bestehen. Perfekt, wenn Sie etwa konfigurierbare Computer, Fahrräder mit verschiedenen Bauteilen oder frei kombinierbare Müslimischungen verkaufen wollen.

Plugin	Composite Products for WooCommerce
Download	*https://woocommerce.com/de-de/products/composite-products/*
Komplexität	++

12.4.4 Vermieten und Planen mit »WooCommerce Bookings«

Ganz egal, ob Sie das nächste AirBnB aufmachen, Termine in Ihrer Praxis unters Volk bringen oder Workshops anbieten möchten: WooCommerce Bookings ist ein treuer Begleiter und kümmert sich für Sie um die Verwaltung Ihrer Ressourcen.

Plugin	WooCommerce Bookings
Download	*https://woocommerce.com/de/products/woocommerce-bookings/*
Komplexität	++

12.4.5 Mehrere Produkte verkaufen mit »Product Bundles for WooCommerce«

Die Gründe dafür, dass Sie Produkte als Bundle verkaufen möchten, können vielfältig sein. Vielleicht haben Sie entdeckt, dass Ihre Kunden immer eine bestimmte Kombination von Produkten kaufen. Oder Sie wissen, dass es sinnvoll ist, zusätzlich zu einer Trinkflasche auch gleich ein Reinigungsset zu verkaufen. Ganz gleich aus welchem Grund, Product Bundles ermöglicht das Zusammenstellen und gemeinsame Verkaufen von Produkten.

Plugin	Product Bundles for WooCommerce
Download	*https://woocommerce.com/de/products/product-bundles/*
Komplexität	+

12.4.6 Eröffnen Sie einen Marktplatz mit »Product Vendors for WooCommerce«

Wenn Sie gleich einen Schritt weiter gehen wollen und nicht nur als Verkäufer*in, sondern direkt als Marktplatz auftreten möchten, auf der auch andere ihre Produkte anbieten können, sollte Product Vendors die Erweiterung sein, nach der Sie suchen.

Plugin	Product Vendors for WooCommerce
Download	*https://woocommerce.com/de/products/product-vendors/*
Komplexität	+++

13 Das Blog

»Blog« oder auch Weblog ist eine Wortkreuzung aus den Worten Web und Log. In seiner ursprünglichen Form versteht sich ein Blog also als Internet-Tagebuch. Die Idee des Bloggens ist beinahe so alt wie das Web selbst. Tim Berners-Lee, der Begründer des World Wide Web, hat sein Blog 1990 gestartet und gilt damit als erster Blogger der Welt. In diesem Kapitel lernen Sie die wichtigsten Schritte, Konzepte und Plugins kennen, um mit Ihrem Blog erfolgreich zu starten.

In einer Zeit vor den allgegenwärtigen Social-Media-Plattformen erlebte das Blogging seine Blüte. In dieser Zeit – zwischen den frühen 2000er- bis in die 2010er-Jahre – wurden Blogs von Enthusiast*innen eifrig gepflegt. Egal ob Mode, Lifestyle, Reise oder Programmierung, für praktisch alles fand sich eine Handvoll Autor*innen. In dieser Zeit hat auch WordPress seine Wurzeln.

Und auch wenn die ersten Wellen des Bloggens inzwischen abgeklungen sind, kann es sich immer noch lohnen, ein *Blog* zu starten und ein eigenes Publikum aufzubauen. Egal, ob Sie Ihre Fortschritte im Stricken dokumentieren, für die Familie das Heranwachsen des ersten Kindes begleiten oder sich professionell als Meinungsführer*in in Ihrem Feld etablieren wollen: Ein Blog ist dafür eine solide Basis. Und mit WordPress haben Sie alle Mittel zum Start einer erfolgreichen Online-Präsenz bereits in Händen.

Der besondere Kniff, wenn Sie sich für ein Blog entscheiden und Ihre Beiträge nicht einfach einem aktuellen sozialen Netzwerk in den Rachen werfen: Sie selbst behalten die Kontrolle über Ihre Inhalte. Nicht nur in dem Sinne, dass Sie nicht unter die Jurisdiktion fragwürdiger Geschäftsbedingungen fallen, Sie selbst haben auch unter Kontrolle, wie lange Ihre Inhalte verfügbar – oder eben nicht verfügbar – sein sollen. Und weil WordPress Open Source ist und diesen Punkt ernst nimmt, werden Sie auch in 15 Jahren noch die Hoheit über Ihre Daten haben und diese im Zweifelsfall exportieren und zu einer anderen Software übertragen können.

Wagen wir also einen Schritt in die wunderbare und etwas chaotische Welt der Blogs. Achten Sie besonders auf Unterkapitel 13.4. Dort haben wir einige wertvolle Plugin-Tipps für Sie gesammelt.

13.1 Zielgruppe und Plan

Ohne konkrete Planung im Vorfeld kann eine neue Website schnell im Chaos versinken. Besonders im Umgang mit WordPress sollten Sie sich regelmäßig daran erinnern, dass zum Beispiel eine nahezu unbegrenzte Auswahl an Plugins und Themes nicht bedeutet, dass man diese auch alle installieren muss. Wir sind oft geneigt, neue Funktionen oder Design-Elemente hinzuzufügen, nicht weil sie wirklich gebraucht würden, sondern weil sie auf Demoseiten interessant aussahen.

Es ist nicht verwerflich, wenn Sie sich gelegentlich von Vorgaben Ihres Themes inspirieren lassen oder von Zeit zu Zeit ein neues Plugin installieren und damit Ihr Blog erweitern. Behalten Sie aber immer im Hinterkopf, dass Sie sich selbst und Ihren Besucher*innen einen Bärendienst erweisen, wenn Sie sie mit einem chaotischen

Design, konfusen Inhalten oder unnötigen Funktionen konfrontieren.

Stellen Sie sich daher die Frage: Müssen Seitenleisten wirklich mit Widgets überladen sein? Hilft die Bilder-Slideshow im Header-Bereich jedes Beitrags Ihren Besucher*innen wirklich? Und ist es wirklich zielführend und nötig, ein Pop-up mit einer Aufforderung zur Newsletter-Anmeldung bei jedem neuen Seitenaufruf erscheinen zu lassen? Behalten Sie stets den Einfluss solcher Elemente und Funktionen auf den Lesefluss im Hinterkopf. In Ihrem Blog sollen schließlich Ihre Inhalte die zentrale Rolle spielen.

Bevor Sie sich also Hals über Kopf in Ihr Blog-Abenteuer stürzen, ist es ratsam, Antworten auf einige zentrale Fragen zu finden, bevor Sie sich an die Details machen. Damit schaffen Sie die Grundlage für einen durchdachten und langfristig erfolgreichen Online-Auftritt mit Ihrem Blog.

13.1.1 Welche Grundmotivation?

Ihre Motivation ist die Basis, auf der Ihr gesamtes Blog-Konzept aufbaut. Was treibt Sie an zu bloggen? Ist es die Leidenschaft, Ihre Gedanken und Ideen mit der Welt zu teilen? Oder haben Sie vielleicht eine Geschäftsidee im Sinn und wollen mit Ihrem Blog über *Affiliate*-Partner und Sponsoring Einnahmen generieren? Vielleicht sehen Sie das Blogging auch als Möglichkeit, sich als Expert*in in Ihrem Feld zu positionieren.

Egal ob Hobby oder Profession, Ihre Motivation wird bestimmen, wie ambitioniert Sie dieses Projekt verfolgen, wie viel Energie und Zeit Sie in Ihr Blog investieren und welche Ziele Sie sich realistisch setzen können. Es lohnt sich, einen Plan zu erstellen, der Ihre langfristigen Ziele und Meilensteine festlegt.

13.1.2 Welche Inhalte?

Was möchten Sie Ihrem Publikum vermitteln? Die Möglichkeiten sind vielfältig. Von Rezepten über Reiseberichte oder Tech-Rezensionen – Ihre Themenwahl ist praktisch nur von Ihrer Fantasie begrenzt. Wichtig ist, dass Sie sich Themen aussuchen, über die Sie leidenschaftlich schreiben können. Ein Blog funktioniert am besten, wenn der Inhalt wirklich authentisch ist und Ihre Leser*innen Ihre Begeisterung für das Thema fühlen können. In einer Welt, in der kein Mangel an Inhalten besteht, haben Sie mit gut verständlichen und vor allem einzigartigen Inhalten einen echten Stein im Brett und sind schon den ersten Schritt auf dem Weg zu einer treuen Leserschaft gegangen.

13.1.3 Besonderheiten Ihres Blogs?

Die Vielfalt von Themen im Netz ist riesig. Wie Sie sich aus der Masse abheben können? Denken Sie darüber nach, wie Sie eine Nische finden oder ein Thema aus einer neuen Perspektive behandeln können. Ein Bekannter hat vor einigen Jahren zum Beispiel einen spannenden Ansatz gefunden, um über seinen Auslandsaufenthalt in Indien zu schreiben: Er nahm für jeden Blog-Beitrag eine Zahl als Aufhänger und erzählte davon ausgehend eine Geschichte. Eine solche herausstechende Herangehensweise oder eine einzigartige Kombination von Themen kann Ihr Blog interessanter machen.

Und auch wenn Sie grundsätzlich niemand davon abhält, in Ihrem Blog über die verschiedensten Themen zu schreiben, sollten Sie dem Ganzen doch einen gewissen Rahmen geben und nicht sowohl über neueste Trends in der virtuellen Realität als auch Ihre liebsten Wanderrouten durch Island berichten. Ein wenig Vorhersehbarkeit

erleichtert es einem potenziellen Publikum, sich auf Sie und Ihre Inhalte einzulassen.

13.1.4 Haben Sie das nötige Durchhaltevermögen?

Bloggen ist ein Marathon, kein Sprint. Ein regelmäßiger Rhythmus beim Veröffentlichen von Beiträgen ist empfehlenswert. Aber anders als bei Plattformen wie YouTube, die von Algorithmen dominiert werden, wird es Ihnen Ihr Publikum eher verzeihen, wenn Sie für einen Urlaub ein paar Wochen Auszeit nehmen.

Haben Sie aus dem Stegreif genug Themenideen, um mit einer gewissen Regelmäßigkeit interessante Beiträge zu veröffentlichen? Haben Sie in Ihrem Alltag genug Zeit dafür, diese Inhalte auch tatsächlich zu verfassen?

13.1.5 Was ist Ihre Zielgruppe?

Für wen schreiben Sie? Mit der Wahl Ihres Themas definieren Sie zwar schon grob Ihre Zielgruppe, aber nehmen Sie sich noch mal ein wenig Zeit, darüber nachzudenken, wie Sie diese Gruppe noch gezielter ansprechen können. Alter, Interessen und Bedürfnisse Ihrer Leser*innen sollten sowohl die Inhalte, die Sie schreiben, als auch die Gestaltung des Blogs an sich beeinflussen.

13.1.6 Wie steht es um Monetarisierung?

Während es in den letzten Jahren definitiv schwerer geworden ist, auf einer unabhängigen Plattform wie einem Blog wirklich Geld zu verdienen, ist es doch noch immer möglich. Ihnen muss aber klar sein, die Monetarisierung ist und bleibt ein längerfristiges Ziel. Um

damit wirklich durchstarten zu können, müssen Sie zunächst eine gewisse Reichweite aufbauen. Je größer Ihr Publikum, desto eher tun sich Möglichkeiten auf, mit Ihrem Blog Geld zu verdienen. Die Möglichkeiten von klassischer Werbeplatzierung über die Teilnahme an Affiliate-Programmen bis hin zu bezahlten Kooperationen sind vielfältig. Ein Blog kann Ihnen mit der Zeit zu einem ansehnlichen Nebeneinkommen verhelfen. Wichtig ist dafür aber, dass Sie konsequent weiter daran arbeiten, die passenden Strategien anwenden und vor allem regelmäßig neue Inhalte veröffentlichen.

Tipp: Privates oder internes Blog?

Auch wenn wir in diesem Kapitel grundsätzlich davon ausgehen, dass Sie ein öffentliches Blog starten wollen und der ganzen Welt Ihre Inhalte präsentieren möchten: Scheuen Sie sich nicht vor einem etwas unkonventionelleren Ansatz. Wie wäre es zum Beispiel mit einem Familien-Blog, in dem Sie über Ihr Leben schreiben, Fotos vom Weihnachtsfest mit der Familie posten oder die Großeltern über das Heranwachsen des jüngsten Enkels im Bilde halten? Oder wenn wir eher im Kontext eines Unternehmens denken: Wie wäre es zum Beispiel mit einem internen Blog, in dem Mitarbeiter*innen sich über die aktuellen Themen im Unternehmen informieren können? Für beide Fälle finden Sie in Abschnitt 13.4 passende Plugins.

Wenn Sie sich klar darüber werden, was Ihre Antworten auf diese Fragen sind, legen Sie damit das Fundament für ein strukturiertes Konzept für Ihr Blog. Im Laufe der Zeit können Sie regelmäßig überprüfen, ob Sie sich noch auf Kurs befinden oder ob sich Ihr Projekt weiterentwickelt hat und Sie das Konzept anpassen müssen.

13.2 Design und Layout

Die Wurzeln von WordPress als Blog-Software werden schnell klar, wenn Sie einen genaueren Blick auf eine neue WordPress-Installation werfen. Mit dem richtigen Theme zeigt die Homepage Ihres Blogs schon einen ersten Beispielinhalt. Von Haus aus sind die Kommentarfunktion sowie das Track- und Pingback-Feature aktiv und machen Interaktionen mit Ihrem Publikum und den Autor*innen anderer Blogs möglich. Im Prinzip könnten Sie direkt loslegen.

Tipp: Entdecken Sie ganz neue Möglichkeiten mit dem Fediverse

In Kapitel 15 dieses Buchs können Sie sich mit den Konzepten von föderierten Websites vertraut machen, die gemeinsam ein soziales Netzwerk bilden. Das sogenannte Fediverse bietet neuen und bestehenden Blogs einen ganzen Strauß an Möglichkeiten zur Interaktion und Verteilung von Inhalten.

Wenn Sie eine neue WordPress-Installation durchführen, ist immer erst mal ein sogenanntes Standard-Theme installiert. Nach dem jeweiligen Jahr benannt, war das Erste der neueren Standard-Themes im Jahr 2010 »Twenty Ten«.

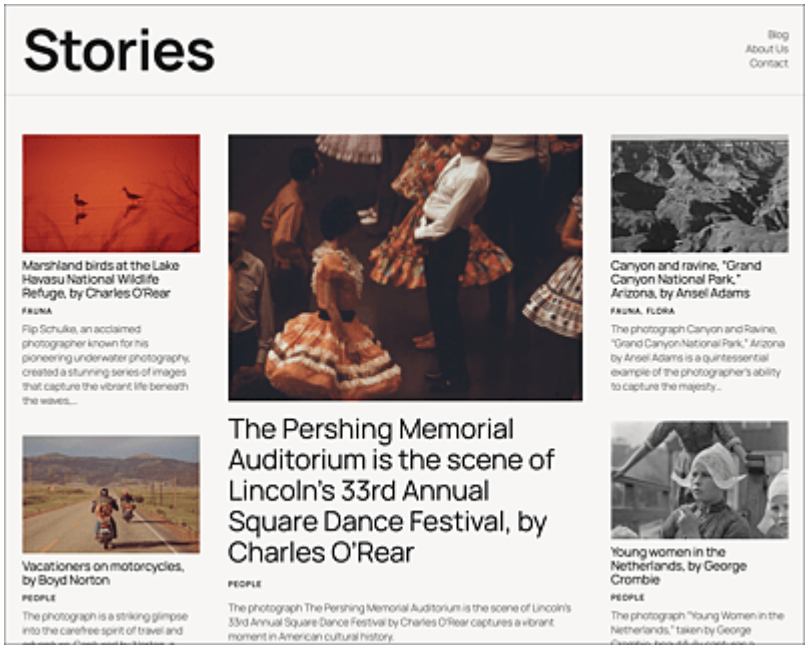


Abbildung 13.1 Das Standard-Theme »Twenty Twenty-Five« hat einen klaren Fokus auf Blogs.

Mit WordPress 6.7 wurde das neue Standard-Theme »Twenty Twenty-Five« ausgeliefert. Die Standard-Themes in den letzten 15 Jahren haben sich durchaus unterschiedlich gut für den Praxiseinsatz geeignet. Sie wurden aber vor allem in den letzten Jahren immer praktischere Begleiter, die noch dazu mit dem Ziel gebaut werden, den aktuellen Stand der Technik zu demonstrieren. Auch wenn Sie sich also für ein anderes Theme entscheiden können, machen Sie mit dem Standard-Theme vermutlich nicht allzu viel falsch. Machen wir uns einmal mit den Elementen vertraut, die Ihnen im Theme »Twenty Twenty-Five« zur Verfügung stehen:

1. **Header und Navigation**



Abbildung 13.2 Der Header-Bereich des Themes »Twenty Twenty-Five« Themes glänzt durch Schlichtheit.

Große, plakative Header-Bilder sind in den letzten Jahren etwas aus der Mode gekommen. Nicht zuletzt deshalb verzichtet Twenty Twenty-Five auf Bilder im *Header*. Stattdessen finden Sie hier klassisch Website-Titel und eine Navigation.

2. **Hauptinhalt**



Abbildung 13.3 Der Inhaltsbereich in »Twenty Twenty-Five« setzt für Blogbeiträge auf ein einspaltiges Design. Für komplexere Seiten sind aber auch aufwendigere Vorlagen enthalten.

Ebenfalls sehr auf Linie mit aktuellen Design-Trends verzichtet das Standard-Theme auf eine zusätzliche Seitenleiste und hat ein weitgehend einspaltiges Design mit viel Platz für die Inhalte. Für die Beitragsübersicht gibt es verschiedene Vorlagen. Egal ob diese nur Beitragsbilder und Titel oder kurze Auszüge

aus den jeweiligen Beiträgen anzeigen: Die neuesten Beiträge sind oben gelistet. Ältere Beiträge wandern weiter nach unten und verschwinden gegebenenfalls auf weiter hinten gelegenen Seiten.

Besonders an diesem Theme ist auch, dass es mit verschiedenen Designs für verschiedene Arten von Blogs kommt. Neben einem klassischen Standard-Design für Blogs kommt es mit einer gesonderten Variation für Foto-Blogs und einer weiteren Design-Variation für Nachrichten-Websites. Wenn Sie Ihre Inhalte also etwas stärker auf Fotos oder auf das Zusammenspiel verschiedener Inhalte ausrichten wollen, haben Sie hier einen interessanten Ausgangspunkt.

3. **Footer**



Abbildung 13.4 In der einfachsten Form enthält der Footer neben Namen und Beschreibung des Blogs einfach ein paar Links zur Navigation. Dank des Website-Editors können Sie Ihrer Fantasie hier aber auch freien Lauf lassen und andere Blöcke einfügen.

Früher wurden die *Footer* oder Fußzeilen von Themes häufig für Widgets genutzt. Während Widgets in WordPress nach und nach von Blöcken abgelöst werden, erweitern sich auch die Möglichkeiten, welche Inhalte Sie in den Footer Ihres Blogs packen können – egal ob begleitende Informationen, die Suche nach Beiträgen, Links zu Kategorien oder, ganz klassisch, als *Blogroll* zu anderen Blogs.

Dieses moderne und sehr schlichte Design hat den Vorteil, dass Sie sich keinen Kopf darüber machen müssen, was mit einer Seitenleiste auf Mobilgeräten passiert. Dort rutschen Inhalte, wenn es eng wird, zusammen. Weil »Twenty Twenty-Five« schon einspaltig ist, ändert sich daran nicht viel.

Der grundlegende Aufbau eines Blogs ist in der Regel sehr ähnlich und wird sich auch nicht dramatisch von dem hier gezeigten unterscheiden, wenn Sie sich für ein andere Theme entscheiden sollten. In Kapitel 10 haben Sie vermutlich bereits die wichtigsten Punkte über die Theme-Auswahl gelesen. Blogs sind in der Regel deutlich anspruchsloser als andere, oft sehr komplexe Websites. Bei der Suche nach einem passenden Theme sollten Sie aber dennoch ein besonderes Augenmerk auf die Aspekte legen, die für ein Blog zentral sind: Darstellung von Blog-Beiträgen, Kommentare und gegebenenfalls andere Punkte, die für Ihr Blog im Speziellen wichtig sind: Bilder-Galerien, Videos, Info-Kästen.

13.3 Struktur und Navigation

Ein gut strukturiertes Blog ist der Schlüssel zu einer angenehmen Benutzererfahrung und wird zu einer höheren Bindung Ihres Publikums beitragen. Die Gestaltung Ihres Blogs sollte einfach und verständlich sein: die Texte gut lesbar und das Layout klar und übersichtlich. Und behalten Sie dabei immer im Hinterkopf, dass Ihr Blog für *Sie* zwar der Dreh- und Angelpunkt sein mag, für die meisten Ihrer Leser*innen aber nur ein Zwischenstopp. Sie können auf den verschiedensten Wegen auf Ihr Blog gestoßen sein:

- über ein Google-Suchergebnis
- durch einen Social-Media-Beitrag
- von der Verlinkung eines anderen Blogs
- als Ergebnis einer Werbeanzeige

Stellen Sie sich vor, wie Ihre Besucher*innen über diese verschiedenen Wege auf Ihrem Blog landen: Was erwarten sie dort, und wie können Sie Ihr Blog nach Möglichkeit so aufbauen, dass es den Erwartungen Ihres Publikums entspricht?

Besucher*innen, die direkt auf einem Ihrer Beiträge landen, interessieren sich vermutlich primär für den aktuellen Inhalt. Das Menü spielt für sie eine untergeordnete Rolle – sie sind zum Lesen hier, nicht um das gesamte Blog zu durchforsten. Am Ende des Beitrags besteht dann aber die Gefahr, dass sie Ihr Blog wieder verlassen. Genau da können Sie ansetzen und zum Beispiel weitere, thematisch passende Beiträge vorschlagen. Um so eine Empfehlungs-Funktion anzubieten, können Sie auf sehr komplexe Plugins zurückgreifen. Oder Sie wählen einen Weg, der etwas weniger auf perfekte Empfehlungen getrimmt ist, aber auch

deutlich einfacher umzusetzen: Nutzen Sie für Ihre Leseempfehlungen einfach die Kategorie oder Schlagworte des aktuellen Beitrags. Empfehlungen basierend auf solchen verwandten Inhalten funktionieren hervorragend, bieten Leser*innen einen echten Mehrwert und halten sie noch dazu ein wenig länger auf Ihrem Blog.

Auf der Homepage und auf den verschiedenen Archivseiten (Kategorien, Datum etc.) erwarten die meisten Besucher*innen die klassische umgekehrt chronologische Anordnung von Beiträgen. Das hat sich über die letzten Jahre als Standard etabliert und macht es einfach, neue Inhalte zu finden. Zusätzliche Elemente wie ein Slider im Header, der auf die beliebtesten oder andere hervorgehobene Artikel hinweist, mag im ersten Moment verlockend erscheinen. Es hat sich aber gezeigt, dass solche Layoutelemente von vielen Besucher*innen meist übersehen werden und eher wertvollen Platz im sichtbaren Bereich Ihres Blogs verschwenden, statt die Aufmerksamkeit effektiv zu lenken.

Das Menü Ihres Blogs ist zweifellos von zentraler Bedeutung, wird aber erst dann wichtig, wenn sich Ihre Besucher*innen tiefer in die Struktur Ihrer Seite hineinarbeiten oder sich orientieren möchten. Achten Sie darauf, dass das Menü klar strukturiert und leicht verständlich ist. Es sollte den Leser*innen helfen, sich in Ihrem Blog zurechtzufinden und nach Möglichkeit weitere spannende Inhalte zu finden. Wenn Sie sie gezielt einsetzen, können Untermenüs hier eine große Hilfe sein. Übertreiben Sie es aber nicht, und halten Sie die Struktur Ihres Menüs stets übersichtlich. Von einem Mega-Menü, wie man es auf vielen großen E-Commerce-Websites findet, sollten Sie für Ihr Blog lieber Abstand nehmen.

Sie können in Ihrem Menü problemlos auch Verweise auf externe Inhalte wie zum Beispiel Ihre Social-Media-Profile unterbringen.

Aber Sie sollten sich überlegen, ob Sie sich damit wirklich einen Gefallen tun oder potenzielle Leser*innen nicht einfach nur auf einen anderen Kanal verweisen. Eventuell sind Social-Media-Links besser in einem gesonderten Menü in Ihrem Footer aufgehoben?

13.4 Funktionen und Plugins

Eine klare Struktur, ein übersichtliches Design und hochwertige Inhalte – sehr viel mehr brauchen Sie für Ihr neues Blog eigentlich nicht. WordPress bringt alle wichtigen Basis-Funktionen mit, die Sie zum Betrieb Ihres Blogs brauchen. An einigen Stellen werden Sie dennoch etwas mehr benötigen als das, was WordPress Ihnen standardmäßig bietet. Dafür haben wir Ihnen im Folgenden einige empfehlenswerte Plugins zusammengestellt, die für Sie und Ihr Blog interessant sein könnten. Auf keinen Fall benötigen Sie alle vorgestellten Plugins, aber vielleicht finden Sie ja ein wenig Inspiration.

Für unsere Plugin-Auswahl haben wir uns vorrangig auf kostenfreie Plugins aus dem offiziellen Plugin-Verzeichnis von WordPress.org konzentriert. Aber auch einige interessante kostenpflichtige Lösungen haben ihren Weg in unsere Liste gefunden.

Info: Weitere interessante Plugins

Weil Ihr Blog am Ende des Tages auch nur eine WordPress-Website ist, gelten viele der Plugin-Empfehlungen, die wir an anderer Stelle in diesem Buch geben, auch für Ihr Blog-Projekt. Beachten Sie insbesondere die Empfehlungen zu Performance (Kapitel 19) und Suchmaschinenoptimierung (Kapitel 20).

13.4.1 Statify

Es gibt eine Vielzahl von Plugins und Services, mit denen Sie messen können, wie viele Besucher*innen Ihre Beiträge aufgerufen

haben. Der Platzhirsch *Google Analytics* wird Ihnen eine derartige Fülle an Daten liefern, dass Sie in den meisten Fällen maximal an der Oberfläche kratzen werden. Spätestens seit der Einführung der DSGVO im Jahr 2018 müssen Sie vor der Datenerhebung mit umfassenden Tools wie Google Analytics aber die Zustimmung Ihrer Besucher*innen einholen.

Wenn Sie sich mit dieser Problematik nicht herumschlagen wollen und außerdem auch keine übermäßig tiefgreifenden Daten zur Nutzung Ihres Blogs benötigen, könnte das schlanke Tracking-Plugin Statify etwas für Sie sein. Statify beschränkt sich auf einige wichtige Eckdaten zu Ihren Seitenaufrufen: Welche Inhalte (Beiträge, Seiten etc.) wurden wie oft abgerufen? Woher kamen die Besucher*innen? Wie verteilen sich die Aufrufe im Zeitverlauf? In den meisten Fällen sind diese Informationen absolut ausreichend, um zu erfahren, ob ein bestimmter Blog-Beitrag besonders beliebt ist, dass montags die meisten Besuche auf Ihrer Website stattfinden und dass der Beitrag über Sie in der Lokalzeitung ein stetiger Quell von Zugriffen ist.

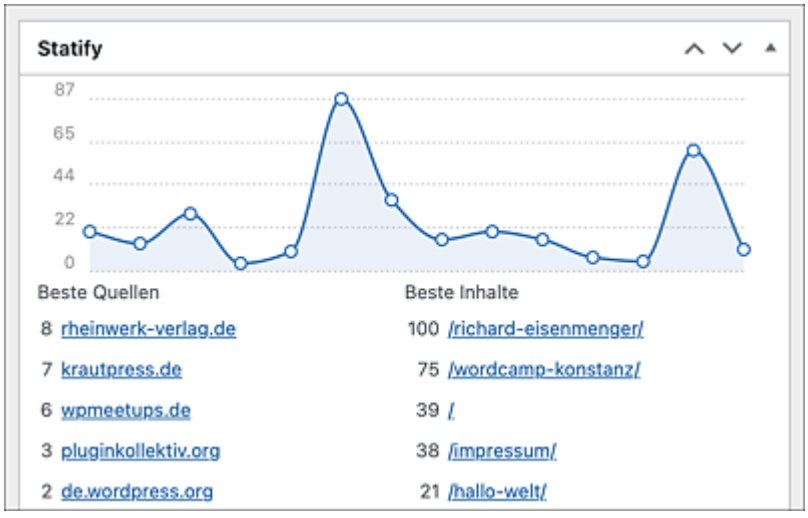


Abbildung 13.5 Statify zeigt die Statistik über die Zugriffszahlen als Dashboard-Widget an.

Plugin	Statify
Aktive Installationen	100.000+
Download	*https://de.wordpress.org/plugins/statify/*
Zweck	datenschutzkonformes User-Tracking ohne personenbezogene Daten
Komplexität	+

Nach der Installation und Aktivierung ist das Plugin direkt funktionsbereit. Wenn Sie als Admin-User angemeldet sind, sehen Sie auf Ihrem Dashboard ein zusätzliches Statify-Widget, in dem nach einiger Zeit die ersten Tracking-Daten angezeigt werden. Sie können das Widget an einer anderen Position auf Ihrem Dashboard platzieren, wenn Sie möchten. Wenn Sie Ihre Maus über das Widget bewegen, erscheint in der Leiste am oberen Rand der Text Konfigurieren. Wenn Sie auf diesen klicken, wechselt das Widget in eine Ansicht, in der Sie die wichtigsten Einstellungen setzen können.

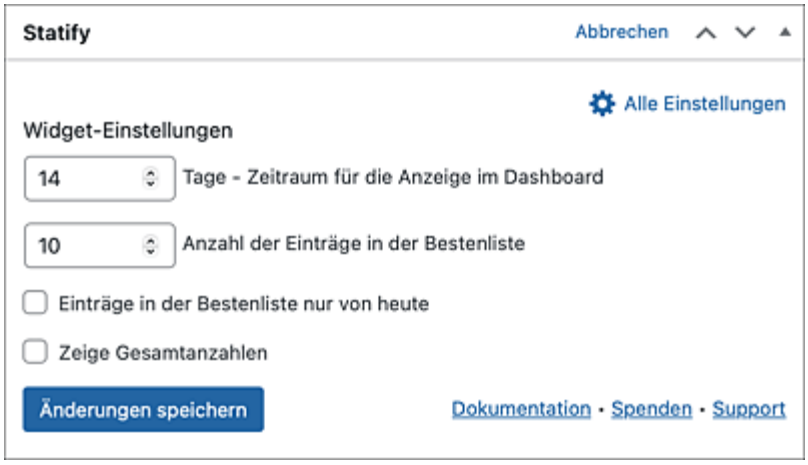


Abbildung 13.6 Sie können einige Einstellungen direkt im Widget verändern.

Tage – Zeitraum für die Anzeige im Dashboard: Mit der ersten Option legen Sie den Zeitraum fest, für den das Plugin Daten im Dashboard-

Widget anzeigt. 14 Tage ist die Standardeinstellung, einige Anwender*innen bevorzugen einen 30-Tage-Zeitraum. Wenn Sie hier einen längeren Zeitraum einstellen, sollten Sie anschließend die umfassende Einstellungsseite besuchen und dort auch noch den Zeitraum der Datenaufbewahrung anpassen (siehe Abbildung 13.7).

Anzahl der Einträge in der Bestenliste: Die zweite Option gibt Ihnen Kontrolle über die Anzahl der Einträge in der Liste der erfolgreichsten Inhalte.

Mit Einträge in der Bestenliste nur von heute können Sie diese Einträge auf einer Tagesbasis anzeigen lassen. Zeige Gesamtanzahlen blendet Ihnen zusätzlich zu den Quellen und besten Inhalten noch die heutigen Zugriffe und die aufsummierten Zugriffe der letzten 30 Tage ein.

Damit Änderungen, die Sie gemacht haben, greifen, müssen Sie abschließend noch auf den Button Änderungen speichern klicken. Vor allem wenn Sie das Plugin erst vor kurzen installiert haben, kann es einige Tage dauern, bis Sie den Effekt der Änderungen sehen. Haben Sie den Erfassungszeitraum zum Beispiel von 14 auf 30 Tage hochgesetzt, wird die Darstellung jeden Tag erweitert, bis die 30 Tage erreicht sind.

Wenn Sie weitere Einstellungen setzen möchten, finden Sie die betreffende Seite unter Einstellungen • Statify im Menü auf der linken Seite oder unter Alle Einstellungen, wenn Sie im Dashboard-Widget im Konfigurations-Modus sind.

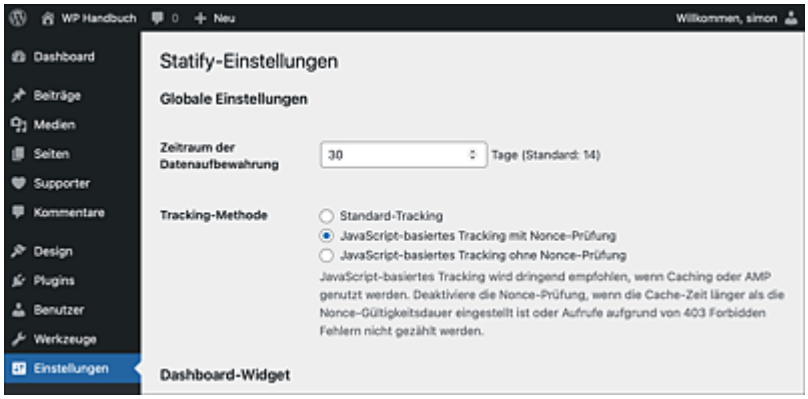


Abbildung 13.7 Die globalen Einstellungen auf der Statify-Einstellungsseite

Auf dieser Seite können Sie mit Zeitraum der Datenaufbewahrung die Vorhaltezeit Ihrer Tracking-Daten konfigurieren.

Die Option Tracking-Methode ist besonders dann interessant, wenn Sie für Ihr Blog ein Caching-Plugin nutzen. Während das Standard-Tracking nämlich auf einen Aufruf in Echtzeit setzt, können Sie bei den beiden JavaScript-basierten Optionen dank eines kleinen Code-Schnipsels, das automatisch eingefügt wird, auch dann Zugriffe tracken, wenn Ihre Website aus dem Zwischenspeicher eines Caching-Plugins ausgeliefert wird.

Tipp: Caching-Plugin einsetzen

Auch wenn Sie Caching in den wenigsten Fällen als Allheilmittel sehen sollten, lohnt es sich, wenn Sie sich für Ihr Blog ein wenig mit dem Thema Performance-Optimierung beschäftigen. In Kapitel 19 haben wir einige der wichtigsten Informationen für Sie zusammengestellt.

Im folgenden Abschnitt Dashboard-Widget können Sie, vergleichbar mit den Einstellungen im Widget selbst, die Darstellung Ihrer Daten anpassen. Zeitraum für die Anzeige im Dashboard beeinflusst die Darstellung der Statistik im Dashboard-Widget. Anzahl der Einträge in

der Bestenliste beeinflusst, wie viele Inhalte unterhalb des Diagramms im Dashboard-Widget angezeigt werden. Auch Bestenlisten nur für heute und Zeige Gesamtanzahlen verhält sich entsprechend den Optionen, die Sie aus den Einstellungen direkt im Widget bereits kennen.

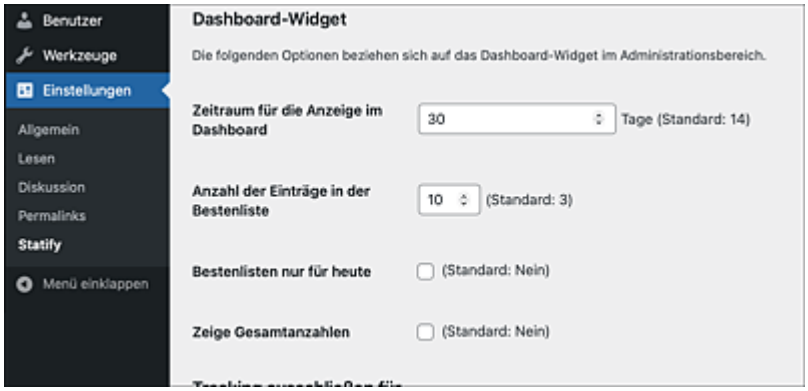


Abbildung 13.8 Die Einstellungen für das Dashboard-Widget

Die letzten beiden Optionen auf dieser Seite erlauben Ihnen, ein wenig anzupassen, wer oder was vom Tracking erfasst werden soll.

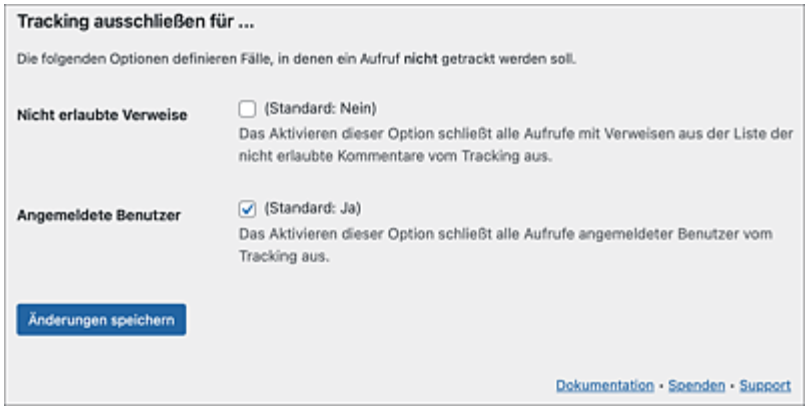


Abbildung 13.9 Optionen zum Ausschließen vom Tracking

Nicht erlaubte Verweise unterdrückt die Darstellung bestimmter Websites, die auf Ihre Inhalte verweisen. Unter bestimmten Umständen kann es vorkommen, dass offensichtliche Spam-Websites in Ihrer Liste auftauchen – in diesem Fall können Sie diese Option aktivieren. In den meisten Fällen ist das nicht nötig.

Angemeldete Benutzer ist standardmäßig aktiv. Besuchen Benutzer*innen, die zeitgleich mit ihrem Account auf Ihrer Website angemeldet sind, eine Seite oder einen Beitrag, so wird dies nicht als Extraaufruf gezählt, um die Statistiken nicht zu verfälschen. Wenn Sie für Ihr Blog zum Beispiel auch für Leser*innen eine Anmeldung ermöglichen, sollten Sie diese Option deaktivieren, damit deren Aufrufe dann trotzdem aufgezeichnet werden können.

13.4.2 Public Post Review

Wenn Sie einen Beitrag vor der Veröffentlichung von anderen Personen gegenlesen lassen möchten, die keine Benutzer*innen Ihrer WordPress-Installation sind, können Sie mit dem Plugin Public Post Preview einen Link erstellen, mit dem eine öffentliche Vorschau aufgerufen werden kann.

Plugin	Public Post Review
Aktive Installationen	100.000+
Download	*https://de.wordpress.org/plugins/public-post-preview/*
Zweck	ermöglicht die Erstellung eines Links, über den ein Beitrag vor der Veröffentlichung auch von Personen ohne Account in der WordPress-Installation angeschaut werden kann
Komplexität	+

Nach der Aktivierung finden Sie in der Dokumenten-Seitenleiste von Beitragsentwürfen die Checkbox Öffentliche Vorschau aktivieren (siehe Abbildung 13.10). Wenn Sie die Checkbox anhaken, wird ein

Feld mit einem Link eingeblendet, mit dem Sie die öffentliche Vorschau teilen können.

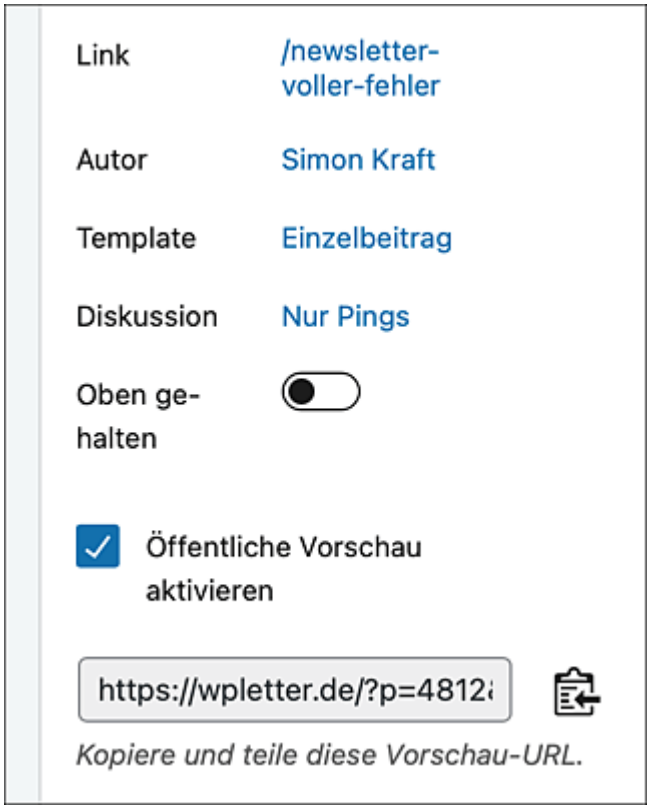


Abbildung 13.10 Das Plugin Public Post Review ermöglicht Ihnen das öffentliche Teilen einer Beitragsvorschau.

13.4.3 Real Cookie Banner

Wenn Sie auf Ihrer Website Dienste einbinden, die Cookies setzen oder personenbezogene Daten an Drittanbieter übertragen, wie Google Analytics, müssen Sie vorher die Einwilligung Ihrer Besucher*innen einholen, solange es sich nicht um technisch notwendige Cookies wie etwa für einen Warenkorb in einem Online-Shop handelt. Ihnen wird wahrscheinlich selbst schon des Öfteren ein Element begegnet sein, das sich beim Öffnen einer Website über den Inhalt legt und Ihnen unterschiedliche Einwilligungsoptionen

anbietet, was Cookies und andere Dienste angeht, die personenbezogene Daten verarbeiten.

Mit Real Cookie Banner können Sie einen solchen Cookie- oder Consent-Banner (Consent für Einwilligung) erstellen und verhindern, dass von Ihrem Theme oder von Plugins vor der Einwilligung Ressourcen von externen Servern geladen werden oder Skripte ausgeführt werden, die Daten an Drittanbieter schicken.

Plugin	Real Cookie Banner
Aktive Installationen	100.000+
Download	*https://de.wordpress.org/plugins/real-cookie-banner/*
Zweck	Erstellung eines Consent-/Cookie-Banners, um die Einwilligung Ihrer Besucher*innen in das Setzen von Cookies und Verarbeiten personenbezogener Daten einzuholen; ermöglicht darüber hinaus das Blocken von Inhalten, bis die Einwilligung gegeben wurde
Komplexität	++

13.4.4 Avatar Privacy

Wenn Sie die Kommentarfunktion in Ihrem Blog nutzen und Profilbilder bei den Kommentaren angezeigt werden, werden diese standardmäßig von dem externen Dienst *Gravatar* geladen. Das Plugin Avatar Privacy sorgt dafür, dass die Bilder auf Ihrem Webserver zwischengespeichert und von dort ausgegeben werden. Außerdem wird beim Kommentarformular eine Checkbox angezeigt, mit der eine Person vor dem Abschicken eines

Kommentars angeben kann, ob ein Gravatar für den Kommentar angezeigt werden soll oder nicht.

Die Einstellungen des Plugins finden Sie nach der Aktivierung ganz unten auf der Seite Einstellungen • Diskussion.

Plugin	Avatar Privacy
Aktive Installationen	1.000+
Download	*https://de.wordpress.org/plugins/avatar-privacy/*
Zweck	speichert die Avatare von Gravatar lokal auf dem Webserver zwischen, sodass bei der Anzeige der Avatare bei Kommentaren im Frontend keine Anfragen an die Gravatar-Server erfolgen
Komplexität	+

13.4.5 Embed Privacy

Der Block-Editor von WordPress bietet eine ganze Fülle von Blöcken zum Einbetten externer Inhalte, etwa von YouTube oder Spotify. Wenn Sie solche Inhalte einbinden, kann das eine deutliche Auswirkung auf die Ladegeschwindigkeit der Website haben, da teils einiges an externen Ressourcen geladen wird, um die eingebetteten Inhalte anzuzeigen. Außerdem wird durch die eingebetteten Inhalte ohne Einwilligung die IP-Adresse Ihrer Besucher*innen an Drittanbieter übertragen, was aus Datenschutz-Sicht bedenklich ist.

Mit dem Plugin Embed Privacy können Sie beiden Problemen begegnen: Das Plugin ersetzt die Einbettungen durch einen Platzhalter mit einem von Ihnen anpassbaren Hinweistext. Wenn Besucher*innen auf den Platzhalter klicken, wird der externe Inhalt

nachgeladen. Über eine Checkbox unter dem Hinweis kann eine Person einstellen, dass sie Inhalte von beispielsweise YouTube immer sehen möchte, sodass sie bei anderen YouTube-Einbettungen nicht mehr die Einwilligung geben muss.

Die Einstellungen des Plugins finden Sie unter Einstellungen • Embed Privacy.

Plugin	Embed Privacy
Aktive Installationen	10.000+
Download	*https://de.wordpress.org/plugins/embed-privacy/*
Zweck	verhindert das Laden von eingebetteten externen Inhalten wie YouTube-Videos, was einerseits die Ladezeit der Website verbessert und andererseits verhindert, dass ohne Zustimmung personenbezogene Daten der Besucher*innen an die externen Websites übertragen werden
Komplexität	+

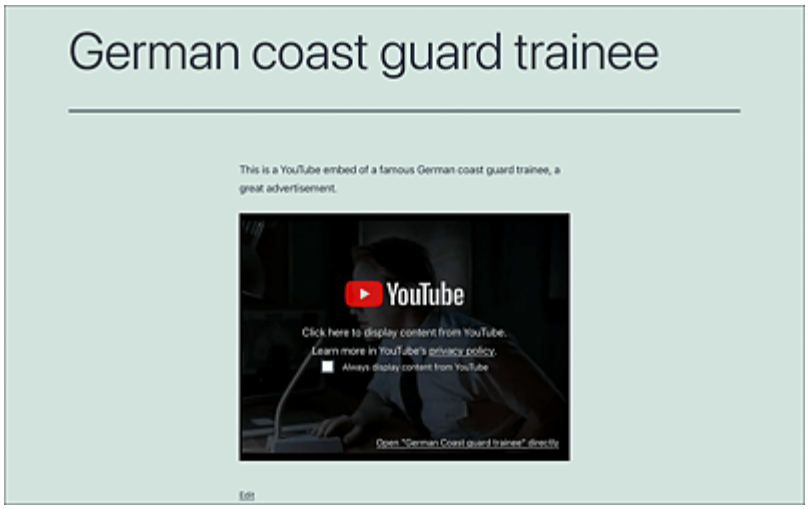


Abbildung 13.11 Embed Privacy verhindert, dass Einbettungen wie YouTube-Videos direkt geladen werden.

13.4.6 Worthy (VG-Wort-Integration)

Wenn Sie Texte im Internet schreiben, wie beispielsweise Blogbeiträge, können Sie bei einer ausreichend hohen Abrufzahl eines Beitrags in einem Jahr von der *VG Wort* dafür vergütet werden. Dafür müssen Sie sich bei der VG Wort registrieren (*https://www.vgwort.de*) und in Ihren Blogbeiträgen Zählmarken einbinden, damit die Aufrufe gezählt werden können.

Das Plugin Worthy unterstützt Sie dabei, indem es in der kostenlosen Version den Import von VG-Wort-Zählmarken ermöglicht und sich darum kümmert, dass Beiträgen Zählmarken zugewiesen werden. In der Bezahlvariante kann auch die Meldung der Artikel an die VG Wort direkt aus dem WordPress-Backend heraus getätigt werden.

Das gesamte Thema VG Wort ist ein bisschen komplex, aber wenn Sie regelmäßig Inhalte veröffentlichen und/oder Artikel haben, die längerfristig gut abgerufen werden, schauen Sie sich das Thema mal genauer an. Es ist ein gutes Gefühl, wenn einmal im Jahr die Ausschüttung kommt und Sie Geld für einen oder mehrere Blogbeiträge bekommen (reich werden Sie damit aber nicht, nur um keine falsche Hoffnung aufkommen zu lassen).

Plugin	Worthy
Aktive Installationen	1.000+
Download	*https://de.wordpress.org/plugins/wp-worthy/*
Zweck	Verwaltung von VG-Wort-Zählmarken und Einbindung der Marken bei Blogbeiträgen. In der Bezahl-Version kann unter anderem auch die Meldung von Artikeln an die VG Wort aus dem WordPress-Backend heraus geschehen.

Komplexität	+

13.4.7 Aged Content Message

Manchmal sind Inhalte relativ zeitlos, aber manchmal sind sie früher oder später nicht mehr aktuell und damit nur noch bedingt nützlich. Mit dem Plugin Aged Content Message können Sie für Beiträge, die älter als X Jahre sind, im Frontend automatisch eine Nachricht anzeigen lassen, mit einem Hinweis darauf, dass der Inhalt vielleicht nicht mehr aktuell ist.

Die Einstellungen zu dem Plugin, in denen Sie unter anderem die Nachricht und das Styling der Nachricht anpassen können, finden Sie unter Einstellungen • Veraltete Inhalte.

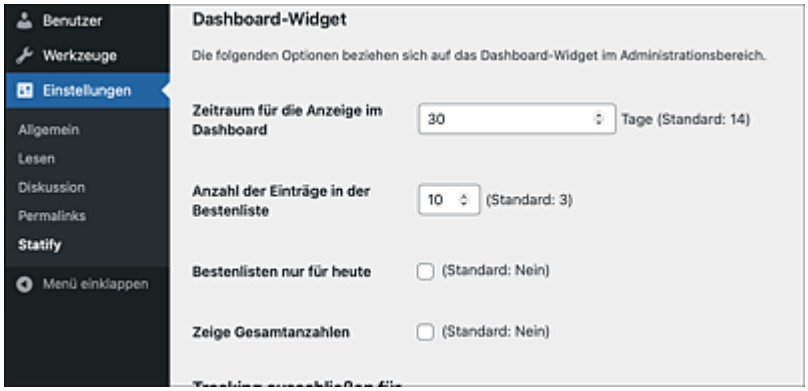


Abbildung 13.12 Mit dem Plugin Aged Content Message können Sie bei älteren Beiträgen automatisch eine Nachricht anzeigen lassen, dass der Inhalt gegebenenfalls nicht mehr aktuell ist.

Plugin	Aged Content Message
Aktive Installationen	10+
Download	*https://wordpress.org/plugins/aged-content-message*

Zweck	zeigt eine Nachricht in älteren Beiträgen an, mit dem Hinweis, dass der Inhalt gegebenenfalls nicht mehr aktuell ist
Komplexität	+

13.5 Blogging-Tipps

Wenn Sie schließlich mit Ihrem Blog erst mal fertig sind, beginnt die eigentliche Arbeit: Neue und interessante Beiträge wollen erstellt werden. Bestehende Inhalte brauchen von Zeit zu Zeit vielleicht ein wenig Pflege. Denn beinahe noch mehr als in anderen Bereichen gilt im Blogging: »Content is King.« Wenn Sie sich also ein treues Publikum aufzubauen wollen und außerdem auch neue Leser*innen anziehen möchten, ist ein konstanter Strom neuer Inhalte ideal.

Aber das Schreiben allein reicht leider oft nicht aus. Um Ihr Blog bekannter zu machen und Ihre Inhalte einer möglichst breiten Öffentlichkeit zu präsentieren, müssen Sie die Kunst des Marketings beherrschen. Hier einige Tipps aus der Schreibpraxis, mit denen Sie das Beste aus Ihrem Blog herausholen können:

13.5.1 Ideen sammeln

Halten Sie Ihre Augen und Ohren stets nach neuen Ideen für Blog-Beiträge offen. Denn gute Ideen für Inhalte sind ein kostbares Gut. Machen Sie es sich deshalb zur Gewohnheit, Ideen gleich festzuhalten, wenn Sie ihnen begegnen. Ob Sie dafür ein Notizbuch mit sich führen, eine extra App verwenden oder einfach auf die Notizen-App Ihres Smartphones zurückgreifen, macht keinen Unterschied. Wichtig ist, dass Sie Gedankenblitze sofort aufschreiben. Selbst eine flüchtige Idee oder eine interessante Anekdote könnten die Grundlage für einen großartigen Blogbeitrag sein.

13.5.2 Ablenkungen vermeiden

Für die meisten von uns erfordert Schreiben ein hohes Maß an Konzentration. Jede Unterbrechung – egal ob durch soziale Medien, E-Mails oder andere Störungen – unterbricht den Schreibfluss und macht es ein wenig schwieriger, einen Beitrag tatsächlich fertigzustellen. Ein möglichst ruhiger Arbeitsplatz, an dem Sie ungestört arbeiten können, ist daher empfehlenswert. Finden Sie heraus, zu welcher Tageszeit Sie besonders gut und konzentriert schreiben können. Egal, ob früh am Morgen im vollen Zug auf dem Weg zur Arbeit, in der Mittagspause oder abends, nachdem Hund und Kinder versorgt sind: Die meisten Menschen haben ein Setting, in dem sie besonders produktiv schreiben können.

13.5.3 Regelmäßige Beiträge – ein Redaktionsplan hilft

Wie eingangs bereits erwähnt ist Konsistenz ein Schlüssel zum Erfolg für Ihr Blog. Versuchen Sie, so regelmäßig wie möglich neue Inhalte zu veröffentlichen. Das Führen eines *Redaktionsplans* kann dabei ausgesprochen hilfreich sein. Neben Tools wie Trello oder Todoist bieten fertige Excel-Dokumente für die Redaktionsplanung einfache Möglichkeiten, Ihre Ideen, Aufgaben und anstehenden Beiträge zu organisieren. Das macht es ganz nebenbei auch deutlich einfacher, wenn Sie irgendwann nicht mehr alleine an Ihrem Blog arbeiten und plötzlich Inhalte mehrerer Autor*innen koordinieren müssen.

13.5.4 Aufgaben gruppieren

Das Schreiben eines Blog-Beitrags umfasst verschiedene Schritte: Themenrecherche, Schreiben, Korrekturlesen, in einigen Fällen

Bilderauswahl und Formatierung. Ganz abhängig davon, wie Ihr persönlicher kreativer Prozess aussieht, kann es hier helfen, eine Art Fließbandarbeit zu nutzen. Statt andauernd zwischen den verschiedenen Aufgaben hin und her zu wechseln, können Sie diese gruppieren und nacheinander abarbeiten.

13.5.5 Artikel attraktiv gestalten

Neben dem reinen Inhalt eines Beitrags spielt auch ein ansprechendes Layout eine wichtige Rolle für die Wahrnehmung Ihres Blogs in der Öffentlichkeit. Überlegen Sie, welchen Stil Sie verfolgen möchten, nutzen Sie die Möglichkeiten, die Ihnen ein Theme wie Twenty Twenty-Five bietet, und entwickeln Sie Ihre eigene Bild- und Textsprache weiter. Die Auswahl einer stimmigen Schriftart, passende – und wirklich nützliche – Bilder und eine durchdachte Struktur können den Unterschied zwischen einem ganz gewöhnlichen und einem herausragenden Blog-Beitrag ausmachen.

13.5.6 Social Media

Das Teilen Ihrer Beiträge auf Social Media kann die Reichweite Ihres Blogs deutlich erhöhen. Entscheidend für den Erfolg dieser Strategie ist aber, dass Sie das richtige Timing für die Veröffentlichung in den verschiedenen Netzwerken finden. Nutzen Sie Plattformen wie das Fediverse, Threads, Bluesky, LinkedIn, Instagram und Pinterest gezielt. Passen Sie Ihre Inhalte an die jeweiligen Erwartungen des Publikums an, und betreiben Sie kein stupides Cross-Posting, bei dem identische Beiträge in allen Netzwerken publiziert werden. Services wie Buffer helfen Ihnen, Ihre Posts zu planen und zur

jeweils optimalen Zeit in den verschiedenen Netzwerken zu veröffentlichen.

13.5.7 SEO im Blick behalten

Suchmaschinenoptimierung (SEO) ist eine unerlässliche Praxis, um Ihre Blogbeiträge für Suchmaschinen und damit für ein breites Publikum sichtbar zu machen. Achten Sie auf einige grundlegende Dinge wie saubere Überschriftenstrukturen und gut strukturierte Inhalte. Weitere Details zur Suchmaschinenoptimierung finden Sie auch in Kapitel 20.

14 Eine Community-Website

WordPress bietet Ihnen deutlich mehr als nur die Möglichkeit, ein Blog oder eine Geschäfts-Website zu betreiben. Mit seiner grundlegenden Benutzerverwaltung können Sie zum Beispiel verschiedenen Personen individuelle Konten mit unterschiedlichen Zugriffsrechten zum Lesen, für redaktionelle oder administrative Tätigkeiten anlegen. Anhand einer Vereins-Website wollen wir in diesem Kapitel betrachten, wie sich WordPress als Community-Plattform nutzen lässt. Ein durchdachtes Konzept und die geschickte Auswahl der passenden Plugins sind dabei für den Erfolg essenziell.

Ob zur Verwaltung und Koordination der Angelegenheiten eines tatsächlichen Vereins oder als Plattform für den Betrieb einer bloßen Online-Community – WordPress ist dank seines Plugin-Systems flexibel und kann für die verschiedensten Anwendungsfälle angepasst werden. Auch wenn wir den »Verein« als Beispiel zur Hand nehmen, sind viele Fälle vorstellbar, in denen Sie eine ähnliche Website etwa für eine lockere Gruppe von Interessierten betreiben oder vielleicht sogar Ihre Großfamilie und deren jährliches Treffen koordinieren.

Wie im vorherigen Kapitel werden wir auch hier WordPress schrittweise ausbauen und als Grundlage für eine kleine Community-Website nutzen. Diese Website werden wir um

sinnvolle Funktionen erweitern, die für solche Szenarien nützlich sein können.

14.1 Aufgaben einer Vereins-Website

Die Website eines Vereins kann die verschiedensten Aufgaben übernehmen. Zum einen ist sie das Aushängeschild der Gemeinschaft. Ähnlich wie die meisten anderen Websites beantwortet sie Fragen wie »Wer wir sind«, »Hier finden Sie uns«. Zeitgleich kann sie aber auch interne Funktionen erfüllen, über das aktuelle Vereinsleben berichten und die Kommunikation der Vereinsmitglieder untereinander fördern.

14.1.1 Externe Darstellung

Für die meisten Inhalte, die Sie der breiten Öffentlichkeit präsentieren möchten, werden Sie auf einfache WordPress-Inhaltstypen zurückgreifen können, Seiten für statische Inhalte, um Ihren Verein zu präsentieren.

14.1.2 Neuigkeiten aus dem Vereinsleben

Zusätzlich könnten Sie ein Blog nutzen (siehe Kapitel 13) und über aktuelle Pläne, Beschlüsse des Vorstands oder die Ergebnisse Ihrer letzten Spendensammlung berichten. Auch diese Inhalte können Sie direkt mit WordPress-Bordmitteln erstellen und anzeigen und müssen keine aufwendige Anpassung Ihres Themes vornehmen.

14.1.3 Bessere innere Vernetzung

Das ist der Bereich, in dem Sie einige spannende Funktionen umsetzen können – seien es ein Kalender, ein interner und passwortgeschützter Downloadbereich, ein Diskussionsforum oder ein E-Mail-Newsletter. Mit einigen der vielen verfügbaren WordPress-Plugins können Sie diese Funktionen direkt auf Ihrer Website umsetzen, ohne dafür auf Dienste Dritter oder fragwürdige Social-Media-Apps zurückgreifen zu müssen.

14.2 Design, Struktur und Navigation

Bei einer Vereins- oder Community-Website sollten Sie auf ein klassisches Website-Layout setzen. Während reine Blogs hier etwas anspruchsloser sind und auf eine große Navigation oder Seitenstruktur beinahe komplett verzichten können, ist für Ihren Verein eine gute Organisation der Inhalte Pflicht. Daher sollten Sie hier auch von extravaganten Designs eher die Finger lassen und stattdessen auf ein übersichtliches Layout und klare Strukturen Wert legen.

Hier könnte sich ein klassisches Design mit einer Seitenleiste neben dem Inhalt als nützlich erweisen. Aber auch moderne Gestaltungen, die auf eine Seitenleiste verzichten und stattdessen auf ein gut strukturiertes Menü und eine übersichtliche Fußzeile setzen, bieten sich an.

Wenn Sie einen geschlossenen Bereich für angemeldete Mitglieder anbieten wollen, sollten Sie sich von Anfang an überlegen, welche Informationen öffentlich für alle Besucher*innen Ihrer Website verfügbar sein sollen und welche nur für angemeldete Mitglieder.

Bedenken Sie immer, dass unter Umständen nicht alle Mitglieder Ihres Vereins oder Ihrer Gruppe einen Computer als primäres Endgerät nutzen. Bei der Auswahl eines Themes sollten Sie deshalb darauf achten, dass es auch auf Smartphones und Tablets einfach zu bedienen ist. Diesen Aspekt sollten Sie auch nach der Auswahl des Themes nicht aus den Augen verlieren, weil vor allem bei komplexeren Funktionen wie einem Veranstaltungskalender oder einem Forum die mobile Nutzung mit Schwierigkeiten verbunden sein kann. Nehmen Sie selbst ein Smartphone oder Tablet zur Hand,

wenn Sie große Änderungen an der Website vorgenommen haben, und überprüfen Sie deren Nutzbarkeit.

14.3 Funktionalitäten und Plugins

Die Plugins in diesem Unterkapitel sind praktische zusätzliche Funktionen, mit denen Sie Ihre WordPress-Installation bereit für den Einsatz als Vereins- oder Community-Website machen.

Doch neben den Plugins, die wir hier vorstellen, gibt es einige weitere Kandidaten, die den Rahmen sprengen würden, die wir der Vollständigkeit aber dennoch kurz erwähnen möchten.

14.3.1 BuddyPress

Seit über 15 Jahren ist BuddyPress als Plugin verfügbar. Es kann seinerseits mit einer Vielzahl von Plugins erweitert werden, bringt aber auch für sich genommen die wichtigsten Funktionen eines kleinen sozialen Netzwerks in Plugin-Form. Als Facebook-Abklatsch aus den 10er-Jahren können Sie mit BuddyPress erweiterte Profile, Gruppen, private Nachrichten, interne Foren-Bereiche und mehr umsetzen. Wegen seiner Komplexität und der vielen Erweiterungsmöglichkeiten gestaltet sich die Integration von BuddyPress in eine Website mit anderen Funktionen aber oft etwas komplizierter. Auch wenn die Optionen verlockend sind, sollten Sie sie nicht leichtfertig auf Ihrer Website integrieren, wenn Sie nicht sicher sind, dass die Funktionen in Ihrem Verein auch wirklich genutzt werden. Nichts ist langweiliger als ein Social Network, das einer Geisterstadt gleicht.

Plugin	BuddyPress
Aktive Installationen	100.000+

Download	*https://de.wordpress.org/plugins/buddypress/*
Zweck	liefert Funktionen eines einfachen sozialen Netzwerks als Plugin für WordPress
Komplexität	+++

14.3.2 Newsletter versenden – Mailjet

Wenn Sie einen Newsletter versenden möchten, gibt es viele verschiedene Anbieter. Einer davon ist Mailjet, mit dem Sie kostenlos bis zu 6000 E-Mails im Monat bei maximal 200 E-Mails am Tag versenden können. Von dem Anbieter gibt es ein WordPress-Plugin, mit dem Sie die Kampagnen aus dem WordPress-Backend heraus verwalten und ein Formular zur Anmeldung zum Newsletter erstellen können.

Plugin	Mailjet
Aktive Installationen	20.000+
Download	*https://de.wordpress.org/plugins/mailjet-for-wordpress/*
Zweck	ermöglicht die Verwaltung der Mailjet-Newsletter-Kampagnen über das WordPress-Backend sowie die Erstellung eines Anmeldeformulars für Ihre Website
Komplexität	++

14.3.3 Kontaktformulare – Form Block

Wenn Sie ein oder mehrere Formulare auf der Website einbinden möchten, um beispielsweise die Kontaktaufnahme für Besucher*innen mit Ihnen zu ermöglichen, können Sie das Plugin Form Block nutzen. Anders als andere Formular-Plugins erstellen Sie die Formulare mit diesem Plugin direkt im Block-Editor statt in einem separaten User-Interface. So müssen Sie sich nicht an ein neues Bearbeitungskonzept gewöhnen, wenn Sie sich schon mit dem Block-Editor auskennen.

Um ein Formular zu erstellen, öffnen Sie nach der Plugin-Installation einfach die Seite oder den Beitrag, wo Sie das Formular einfügen möchten, und wählen den Formular-Block aus.

Plugin	Form Block
Aktive Installationen	100+
Download	*https://de.wordpress.org/plugins/form-block/*
Zweck	ermöglicht Ihnen die Erstellung von Formularen direkt über den Block-Editor
Komplexität	+

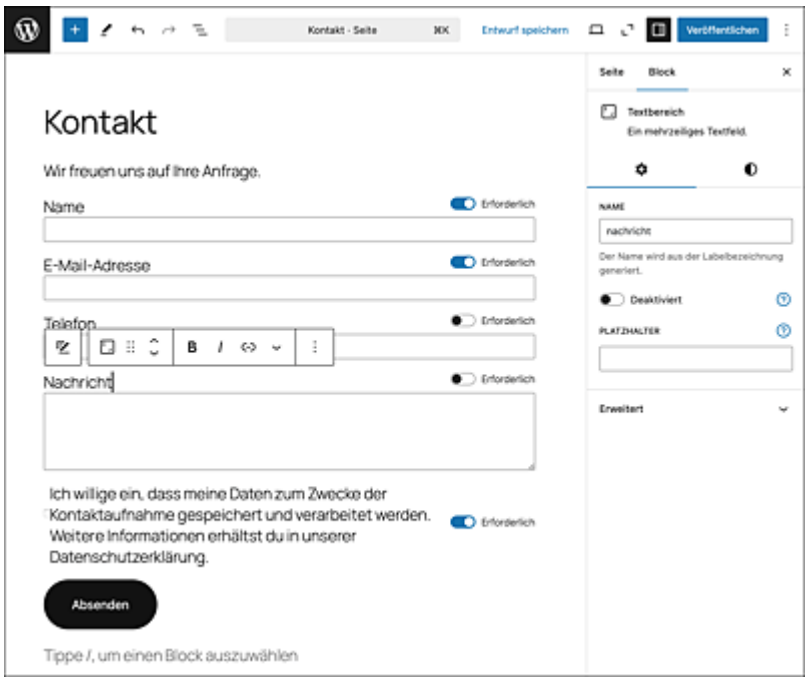


Abbildung 14.1 Das Form-Block-Plugin lässt Sie Formulare direkt im Block-Editor erstellen und anpassen.

14.3.4 Zugriffsrechte und Rollen justieren – User Role Editor

Für die meisten Anwendungsfälle wird die relativ übersichtliche Anzahl an Rollen von WordPress ausreichen. Wenn Sie aber zusätzliche Rollen benötigen oder eine bestehende Rolle verändern möchten, um ihr zum Beispiel mehr Berechtigungen geben, hilft das Plugin User Role Editor. Mit dem Plugin haben Sie die Möglichkeit, sowohl bestehende Rollen zu verändern als auch komplett neue Rollen hinzuzufügen.

Plugin	User Role Editor
Aktive Installationen	700.000+
Download	*https://de.wordpress.org/plugins/user-role-editor/*

Zweck	lässt Sie neue Rollen anlegen und bestehende Rollen bearbeiten
Komplexität	++

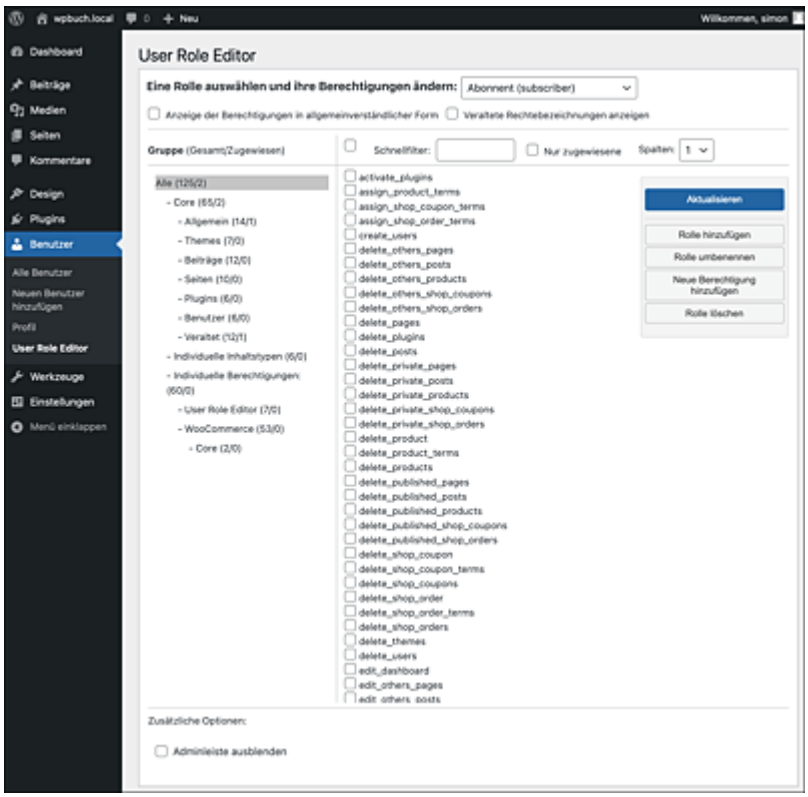


Abbildung 14.2 Mit dem Plugin User Role Editor können Sie bestehende Rollen bearbeiten und neue hinzufügen.

15 IndieWeb und Fediverse

*Soziale Netzwerke gewinnen zunehmend an Bedeutung, auch für Blogger*innen und Website-Betreiber*innen, da sie mehr Reichweite und Engagement schaffen. In diesem Kapitel erfahren Sie, wie Sie durch IndieWeb und Fediverse die Vorteile sozialer Netzwerke direkt in WordPress nutzen und soziale Interaktionen wie Kommentare, Likes und Shares auf Ihre eigene Website holen können.*

Soziale Netzwerke sind aus dem heutigen Internet nicht mehr wegzudenken. Sie bieten eine Plattform für soziale Interaktionen, den Austausch von Inhalten und die Vernetzung mit anderen Benutzer*innen. Für Blogger*innen und Website-Betreiber*innen sind soziale Netzwerke ein wichtiger Kanal, um ihre Inhalte zu verbreiten, eine größere Reichweite zu erzielen und mit ihrer Zielgruppe in Kontakt zu treten.

Autor des Kapitels



Matthias Pfefferle ist ein WordPress-Nutzer der ersten Stunde. Sein erstes Blog betrieb er 2002 mit b2, dem Vorgänger von WordPress, und seit 2016 widmet er sich hauptberuflich der Arbeit an und mit WordPress.

Seit 2006 beschäftigt er sich intensiv mit Technologien rund um ein offenes, portables, interoperables, unabhängiges und dezentralisiertes Internet. Er hat zahlreiche WordPress-Plugins in diesen Bereichen entwickelt und teilt sein Wissen und seine Erfahrungen regelmäßig in Vorträgen, Fachartikeln und als Gast in Podcasts.

Darüber hinaus beteiligt er sich aktiv in der IndieWeb- und ActivityPub-Community, um Standards wie Webmentions und ActivityPub mitzugestalten. Kontaktmöglichkeiten finden Sie unter: notiz.blog/contact

Allerdings haben zentrale soziale Netzwerke wie Facebook, Twitter und Instagram auch ihre Nachteile. Sie sind oft undurchsichtig in Bezug auf ihre Algorithmen und Richtlinien, kontrollieren den Zugang zu Inhalten und Daten, und es gelten immer die Regeln des jeweiligen Betreibers. Dies hat dazu geführt, dass viele Benutzer*innen und Entwickler*innen nach Alternativen suchen, die ihnen mehr Kontrolle über ihre Daten und ihre Online-Identität geben.

In diesem Kapitel werden wir uns mit zwei Bewegungen befassen, die sich für eine dezentrale und offene Webinfrastruktur einsetzen: dem *IndieWeb* und dem *Fediverse*. Diese Bewegungen fördern die Verwendung von offenen Standards und Protokollen, um soziale Interaktionen und den Austausch von Inhalten zwischen verschiedenen Websites zu ermöglichen. Wir werden uns ansehen, wie Sie die Prinzipien des IndieWeb in WordPress integrieren können oder Teil des Fediverse werden, um direkt von den Vorteilen sozialer Netzwerke zu profitieren, ohne sich von ihnen abhängig zu machen.

Achtung: Sie bauen hier kein soziales Netzwerk

Sie werden in diesem Kapitel nicht lernen, wie Sie ein eigenes soziales Netzwerk (das erfahren Sie aber in Kapitel 14) auf Basis von WordPress aufbauen. Wir werden nicht auf Plugins wie BuddyPress oder BuddyBoss eingehen, sondern erläutern, wie Sie WordPress direkt mit klassischen oder modernen dezentralen sozialen Netzwerken verbinden können.

15.1 Das IndieWeb

Das IndieWeb (*https://indieweb.org/*) wurde 2010 von Tantek Çelik und Aaron Parecki ins Leben gerufen. Die Gründer waren frustriert von der zunehmenden Abschottung der großen sozialen Netzwerke (Stichwort: Walled Gardens) einerseits und der wachsenden Komplexität der offenen Standards und Protokolle andererseits und wollten eine Gegenbewegung schaffen. Tantek fasst das Problem in einer Präsentation folgendermaßen zusammen: »Mich interessieren keine dezentralen Protokolle, mich interessieren meine Daten, und mich interessieren meine Freunde.«

Um nicht die gleichen Fehler zu begehen, sieht sich das IndieWeb in erster Linie als eine Bewegung mit der Idee, sich unabhängig von großen sozialen Netzwerken zu machen. Im Vordergrund stehen nicht Technologie und Protokolle, sondern Prinzipien und Werte.

Die Grundsätze des IndieWeb sind:

- *Ihre Inhalte (Texte, Bilder, Videos, ...) gehören Ihnen* – wenn Sie etwas im Web veröffentlichen, sollte es Ihnen gehören, nicht einer Firma. Zu viele Unternehmen mussten Konkurs anmelden, was den kompletten Verlust aller Daten ihrer Nutzer*innen zur Folge hatte. Mit dem IndieWeb bleibt Ihr Inhalt Ihr Eigentum und unter Ihrer Kontrolle.
- *Sie sind besser vernetzt* – Ihre Artikel und Statusmeldungen können auf allen Plattformen geteilt werden, nicht nur auf einer. So haben Sie die Freiheit, überall zu interagieren. Antworten und Likes von anderen Diensten werden auf Ihre Website zurückgeführt, sodass alles zentral an einem Ort gesammelt wird.
- *Sie haben die Kontrolle* – Sie können veröffentlichen, was und in welcher Form Sie möchten, ohne dass jemand Sie überwacht.

Eine Liste aller Prinzipien finden Sie unter *https://indieweb.org/principles*.

15.1.1 Konferenzen und Meetups

Die Bewegung ist durch ihre Camps gewachsen, weshalb Konferenzen nach wie vor ein zentraler Bestandteil des IndieWeb sind. Die Gruppe versteht sich als menschenorientiert, nicht projektorientiert. Zu den regelmäßigen Treffen sind alle willkommen. Man muss keine Website haben, um teilzunehmen, sollte jedoch Interesse an einer solchen zeigen. Die Konferenzen sind ähnlich aufgebaut und strukturiert wie die Veranstaltungen der WordPress-Community.

IndieWebCamps

Ein *IndieWebCamp* ist eine mehrtägige Veranstaltung, ähnlich wie ein *WordCamp*, bei der Entwickler*innen, Designer*innen und Blogger*innen zusammenkommen, um an Projekten zu arbeiten, Ideen auszutauschen und neue Technologien auszuprobieren. Diese Camps finden weltweit in verschiedenen Städten statt und stehen allen offen, die sich für das Thema interessieren.

Der Aufbau eines IndieWebCamps folgt klassischerweise einem zweitägigen Format (es gibt aber auch Ausnahmen):

- **Tag 1: Vorträge und Diskussionen.** Der erste Tag des Camps gleicht einem Barcamp oder einer Unkonferenz. Er beginnt mit einer Vorstellungsrunde, gefolgt von einer Themensammlung und der Planung der Sessions. Das Format der Sessions ist offen – es kann sich um Vorträge oder Diskussionsrunden handeln. Auch die Themen sind flexibel und reichen von Technik und Programmierung über Design und Datenschutz bis hin zu kulturellen und sozialen Themen. Der Tag endet mit einer Abschlussrunde, in der die Teilnehmer*innen ihre Erfahrungen

und Erkenntnisse teilen können, und einem abschließenden Abendessen.

- **Tag 2: Hackathon und Workshops.** Der zweite Tag ist der »Create Day«. Die Teilnehmer*innen haben die Möglichkeit, an ihren Projekten oder Webseiten zu arbeiten, neue Technologien auszuprobieren und sich gegenseitig zu unterstützen. Der Tag endet mit einer Abschlusspräsentation, bei der die Teilnehmer*innen ihre Projekte vorstellen und Feedback erhalten.

Eine Liste der anstehenden IndieWebCamps finden Sie unter *https://indieweb.org/IndieWebCamps*.

Homebrew Website Clubs

Der *Homebrew Website Club* ist eine regionale Gruppe, die sich ein- bis zweimal im Monat trifft, um gemeinsam an ihren Webseiten zu arbeiten. Man könnte diese Treffen mit einem *WordPress-Meetup* vergleichen, jedoch mit dem Unterschied, dass es hier stärker um praktische Themen geht. Sollten Sie Probleme mit IndieWeb-Plugins oder Fragen zu allgemeinen Themen haben, ist der Homebrew Website Club die ideale Anlaufstelle. Homebrew Website Clubs gibt es in vielen, vor allem größeren Städten weltweit. Die Treffen sind offen für alle und finden oft in Cafés oder Coworking Spaces statt.

Info: Homebrew Website Club

Der Name »Homebrew Website Club« ist eine Hommage an den »Homebrew Computer Club«, einen Verein von Hobbybastler*innen und Elektronikbegeisterten, die sich Mitte der 1970er-Jahre trafen, um Teile, Schaltungen und Wissen zum Bau

von Computern auszutauschen. Das bekannteste Mitglied war Steve Wozniak, Mitbegründer von Apple Computer.

Unter *https://indieweb.org/Homebrew_Website_Club* können Sie herausfinden, ob es auch in Ihrer Nähe einen Homebrew Website Club gibt.

15.1.2 Technik und Protokolle

Die Gruppe stellt aber nicht nur Thesen auf, sondern arbeitet auch an Lösungen, um diese bestmöglich umzusetzen. Technisch gesehen könnte man das IndieWeb als eine Weiterentwicklung des Web 2.0 sehen. Der Fokus liegt auf einfachen Formaten, ähnlich wie RSS-Feeds, Pingbacks/Trackbacks, und einer engen Verbundenheit zwischen Blogs/Webseiten und sozialen Netzwerken.

Info: Web 2.0

Der Begriff »Web 2.0« entstand Anfang der 2000er-Jahre und beschreibt die Weiterentwicklung des Internets im Vergleich zu dem Web 1.0. Während das Web 1.0 vor allem aus statischen Webseiten bestand, auf denen Nutzer*innen Inhalte lediglich passiv konsumieren konnten, ermöglichte das Web 2.0 eine aktive Teilnahme der Nutzer*innen. Im Web 2.0 konnten Menschen Inhalte erstellen, teilen und in sozialen Netzwerken interagieren, wodurch das Internet zu einer dynamischen und kollaborativen Plattform wurde. Im Gegensatz zu Web 1.0, das hauptsächlich Informationen einseitig bereitstellte, förderte das Web 2.0 eine stärkere Vernetzung und Mitgestaltung der Nutzer*innen.

Viele der Prinzipien des IndieWeb lassen sich mit dem Webmention-Protokoll und der Idee des *POSSE* (Publish (on your) Own Site, Syndicate Elsewhere) umsetzen, deshalb werden wir uns mit diesen beiden Themen im Folgenden genauer beschäftigen.

Webmentions

Webmentions (*https://webmention.net/*) sind eine sehr vereinfachte Version von Pingbacks bzw. Trackbacks und funktionieren nach dem gleichen Prinzip. Alle Links eines Artikels werden »angepingt«, um den verlinkten Seiten mitzuteilen, dass sie gerade erwähnt wurden. Im Gegensatz zu XML-RPC (Pingback) oder embedded RDF (Trackback) basiert eine Webmention auf einem simplen HTTP Post. Damit folgt die Gruppe ihrem Leitsatz: »Bei Bedarf übernehmen, verbessern und schaffen wir offene Standards für gutes Design, Benutzererfahrung und einfache Implementierung.« Sie nehmen ein vorhandenes Protokoll wie Pingback und vereinfachen und erweitern es.

Info: Pingbacks

Pingbacks sind eine automatische Methode, um auf Blogbeiträge hinzuweisen und eine Verbindung zwischen ihnen herzustellen. Wenn Sie in Ihrem Blogbeitrag auf einen anderen Beitrag verlinken, sendet Ihr Blog automatisch eine Pingback-Benachrichtigung an den Blog, auf den Sie verlinken. Der empfangende Blog überprüft die Pingback-Nachricht und fügt, falls sie gültig ist, einen Link zu Ihrem Beitrag in den Kommentaren seines eigenen Beitrags hinzu. Dadurch wird eine Verbindung zwischen den beiden Beiträgen hergestellt, und Leser*innen können leicht sehen, welche Beiträge miteinander verknüpft sind.

Der Unterschied beider Formate lässt sich am besten anhand eines Beispiels erklären. Der Aufbau eines Pingbacks sieht wie folgt aus:

```
POST /pingback-endpoint HTTP/1.1
Host: example.com
Content-Type: text/xml
<?xml version="1.0" encoding="iso-8859-1"?>
<methodCall>
 <methodName>pingback.ping</methodName>
 <params>
  <param>
   <value>
    <string>https://example.org/source</string>
   </value>
  </param>
  <param>
   <value>
    <string>https://example.com/target</string>
   </value>
  </param>
 </params>
</methodCall>
```

Durch das spezielle XML-Format ist das Verhältnis zwischen Auszeichnung und eigentlicher Information (die URL des verlinkenden Beitrags (source) und die URL des verlinkten Beitrags (target)) nicht wirklich ausgeglichen. Webmention hingegen ist ein simpler HTTP-Post, der nur die beiden URLs enthält:

```
POST /webmention-endpoint HTTP/1.1
Host: example.com
Content-Type: application/x-www-form-urlencoded

source=https://example.org/source&
target=https://example.com/target
```

Das Protokoll ist dadurch nicht nur deutlich einfacher, sondern bietet auch den Vorteil, dass Pings (oder Mentions) ganz leicht über ein simples HTML-Formular gesendet werden können.

```
<form method="post"
 action="https://example.com/webmention-endpoint">
 <input name="source" type="url" />
 <input name="target" type="hidden"
  value="https://example.com/target" />
 <input type="submit" value="Ping!" />
</form>
```

Ein Anwendungsbeispiel für ein solches Formular finden Sie in Abschnitt 15.1.3 über das Webmention-Plugin.

Auch wenn die Webmention-Spezifikation nicht klar definiert, wie eine solche »Mention« dargestellt werden soll und welche Informationen sie enthalten kann, gibt es eine Reihe von Best Practices, die von der IndieWeb-Community entwickelt wurden. Während Pingbacks versuchen, einen »Auszug aus dem Seiteninhalt rund um den Link des Beitrag« zu extrahieren, versucht die IndieWeb-Community über Webmentions, den Inhalt der verlinkenden Seite so gut wie möglich zu »verstehen« und zu interpretieren.

Das präferierte Format für Webmentions sind Microformats. Microformats gehörten zu den Web-Semantiken und nutzen bestehende HTML-Tags, um Daten zu strukturieren und auszuzeichnen.

Info: Microformats

Microformats sind eine einfache Möglichkeit, strukturierten Daten in HTML zu integrieren, um Informationen besser verständlich zu machen, sowohl für Menschen als auch für Maschinen (z. B. Suchmaschinen). Sie verwenden Standard-HTML-Tags, um bestimmte Arten von Informationen (wie Veranstaltungen, Personen, Orte oder Bewertungen) zu kennzeichnen, ohne zusätzliche, komplexe Formate zu verwenden.

Ein Beispiel für ein Microformat wäre das Auszeichnen von Kontaktinformationen:

```
<div class="h-card">
 <span class="p-name">Max Mustermann</span>
 <a class="u-email" href="mailto:max@example.com">max@example.com</a>
</div>
```

Im Beispiel werden der Name und die E-Mail-Adresse von Max Mustermann durch spezielle Klassen wie `p-name` und `u-email` ausgezeichnet.

Der Präfix `x-` definiert, wie ein Attibut interpretiert werden soll:

`h-*` – der »Container« eines Microformats, er definiert den Typ des Formats (z. B. `h-card`)

`p-*` – für Fließtexteigenschaften ohne HTML-Code (z. B. `p-name`)

`u-*` – für URL-Eigenschaften (z. B. `u-photo`)

`dt-*` – für Datums-/Uhrzeiteigenschaften (z. B. `dt-bday`)

`e-*` – ähnlich wie `p-*`, aber mit HTML-Unterstützung (z. B. `e-note`)

Über Microformats können Daten, die für die Besucher*innen einer Website relevant sind, auch für Suchmaschinen und andere Anwendungen besser interpretierbar gemacht werden.

Mehr Informationen zu Microformats finden Sie unter *https://microformats.org/wiki/microformats2*.

Über das `h-entry`-Microformat lässt sich ein Blog-Post mit all seinen Attributen auszeichnen:

```
<article class="h-entry">
 <h1 class="p-name">Mein Blog-Post</h1>
 <p class="e-content">Hier ist der Inhalt meines Blog-Posts.</p>
 <a class="u-url" href="https://example.com/source">Permalink</a>
</article>
```

Das `h-entry`-Element kennzeichnet den gesamten Blog-Post, `p-name` den Titel, `e-content` den Inhalt und `u-url` die URL des Beitrags. Microformats ermöglichen Ihnen aber auch, die Verlinkung einer Seite auszuzeichnen. Darüber kann die verlinkte Seite unterscheiden, ob es sich bei Ihrem Artikel nur um eine Erwähnung,

einen Kommentar zu einem Artikel oder einfach nur einen simplen »Like« handelt.

```
<article class="h-entry">
 <!-- Der Text ist eine Antwort an den verlinkten Beitrag -->
 <a class="u-in-reply-to" href="https://example.org/target">Verlinkter Beitrag</a>
 <!-- Der Text ist ein "Like" des verlinkten Beitrags -->
 <a class="u-like-of" href="https://example.org/target">Verlinkter Beitrag</a>
</article>
```

Über diese Formatierung lässt sich nicht nur der exakte Text des Beitrags ermitteln, sondern auch die Intention des Verfassers bzw. der Verfasserin abbilden. Der erste Link zeigt, dass der Artikel eine direkte Antwort auf den verlinkten Beitrag ist, der zweite Link bildet einen »Like« ab. Mit diesen Informationen kann die verlinkte Seite dann entscheiden, wie sie mit der Webmention umgehen bzw. wie sie die Erwähnung darstellen soll.

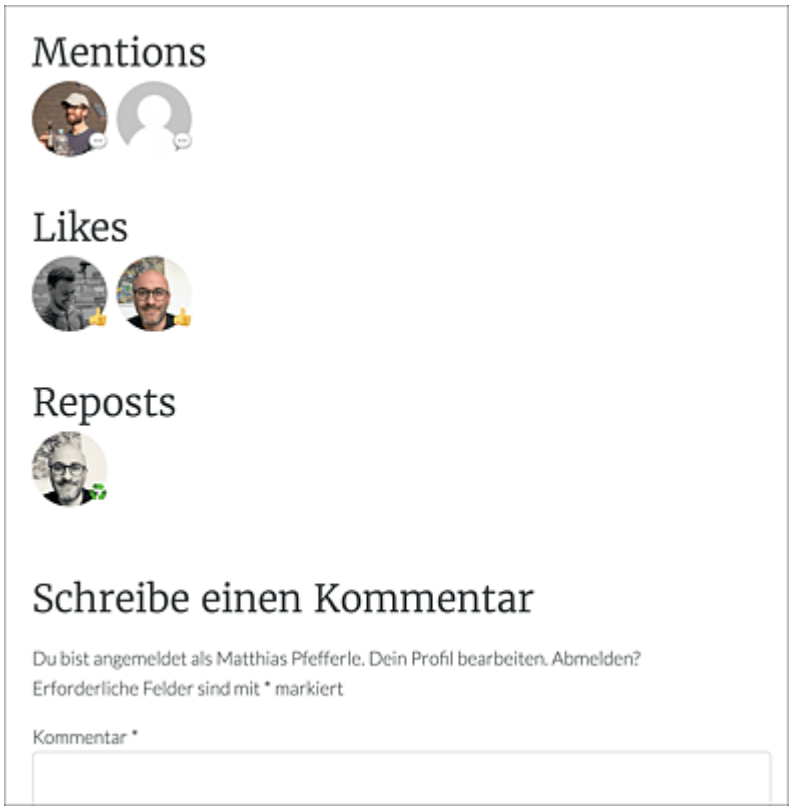


Abbildung 15.1 Beispiel für die Darstellung der unterschiedlichen Webmention-Typen

Tipp: Webmentions ohne manuelle Theme-Anpassungen nutzen

Nicht alle sind in der Lage, HTML zu schreiben oder ihr WordPress-Theme entsprechend anzupassen. Wir werden im Kapitel zum Webmention-Plugin erklären, welche Alternativen es gibt und wie Sie auch ohne HTML-Kenntnisse Webmentions auf Ihrer Website nutzen können.

Webmentions in Kombination mit Microformats sind exemplarisch für die Herangehensweise des IndieWeb. Alle Protokolle sind einfach gehalten und bieten trotzdem oder gerade deshalb Möglichkeiten, die weit komplexeren Protokollen in nichts nachstehen.

Mit einem simplen HTTP-Post und etwas HTML bietet die Community die Möglichkeit einer dezentralen Kommunikation zwischen Blogs, und in Kombination mit POSSE lässt sich dieses Prinzip auch auf soziale Netzwerke anwenden.

POSSE (und PESOS)

Um eine Brücke zwischen Webseiten/Blogs und sozialen Netzwerken zu schlagen, hat das IndieWeb zwei Vorgehensweisen definiert:

- **POSSE** – Publish (on your) Own Site, Syndicate Elsewhere. Dieser Ansatz besagt, dass Sie Ihre Inhalte zuerst auf Ihrer eigenen Website veröffentlichen und sie dann auf sozialen Netzwerken und anderen Plattformen teilen. Auf diese Weise behalten Sie die Kontrolle über Ihre Inhalte und können sie überall dort teilen, wo Ihre Zielgruppe ist. *https://indieweb.org/POSSE*

- **PESOS** – Post Elsewhere, Syndicate (to your) Own Site. Das Gegenstück zu POSSE: Sie veröffentlichen Ihre Inhalte zuerst auf sozialen Netzwerken und anderen Plattformen und aggregieren sie dann auf Ihrer eigenen Website. Viele Netzwerke wie z. B. Instagram bieten keine Möglichkeit außer ihrer eigenen App, um Bilder zu veröffentlichen. Über PESOS kommen Sie dennoch an Ihre Bilder und können sie an einem zentralen Ort (Ihrer Webseite) sammeln und archivieren. *https://indieweb.org/PESOS*

In der WordPress-Welt ist dieses Vorgehen nicht neu und eher unter dem Begriff »Crossposting« bekannt. Es gibt aber einen wesentlichen Unterschied zwischen Crossposting und POSSE. Crossposting wird meist über Plugins realisiert, welche Ihre Beiträge direkt über die API des jeweiligen Netzwerks veröffentlichen. POSSE setzt hingegen auf externe Dienste, die auch oft als »Bridges« bezeichnet werden. Der Vorteil einer solchen »Brücke« ist die generische Implementierung in WordPress. POSSE benötigt nicht für jedes weitere soziale Netzwerk ein neues Plugin, da die eigentliche Logik in der Bridge liegt.

Eines der bekanntesten Beispiele ist *Brid.gy*. Der Dienst ermöglich Ihnen das Crossposten zu flickr, GitHub, Mastodon, reddit, Bluesky, tumblr, WordPress.com, Blogger und Medium. Die Bridge ist so aufgebaut, dass Sie auf Ihrem WordPress-Blog nur das Webmention-Plugin installieren müssen.



Abbildung 15.2 Die Brid.gy Startseite mit Anmeldemöglichkeiten für verschiedene soziale Netzwerke

Das nächste Mal, wenn Sie einen Beitrag veröffentlichen, wird Brid.gy automatisch benachrichtigt und veröffentlicht Ihren Beitrag dann auf den von Ihnen gewählten sozialen Netzwerken. Neben POSSE unterstützt Brid.gy auch PESOS. Kommentare und andere Interaktionen wie z. B. Likes, die auf dem entsprechenden Netzwerk stattfinden, werden über Brid.gy gesammelt und wieder als Webmention zurück an Ihren Blog gesendet.

Sie finden eine Liste weiterer Bridges unter *https://indieweb.org/bridge_all_the_things*.

Der Ansatz von POSSE und PESOS hat den Vorteil, dass Sie letztendlich nur das Webmention-Plugin installieren müssen, um sowohl mit anderen Blogs als auch mit sozialen Netzwerken kommunizieren zu können. Der einzige (kleine) Mehraufwand ist die Konfiguration der Bridges, was in der Regel aber relativ einfach ist.

15.1.3 Plugins

Um die Prinzipien des IndieWeb in WordPress zu integrieren, gibt es eine Reihe von Plugins. Die bekanntesten sind das Webmention-Plugin und das IndieWeb-Plugin.

Webmention-Plugin

Das Webmention-Plugin erfordert keine zusätzlichen Einstellungen und ist direkt nach der Installation einsatzbereit. Ähnlich wie bei Pingbacks durchsucht es neue Beiträge nach Links und versucht, diese zu »pingen«. Zudem stellt es einen API-Endpunkt bereit, über den andere Webseiten Webmentions an Ihre Seite senden können.

Wenn eine Webmention empfangen wird, versucht das Plugin, den Inhalt der verlinkten Seite mithilfe verschiedener Methoden zu extrahieren. Neben Microformats verwendet es auch OpenGraph, Twitter Cards, Schema.org und die WordPress API, um den Inhalt zu interpretieren. Falls es Darstellungsprobleme mit Webmentions auf anderen Seiten gibt, empfiehlt es sich, eines der genannten Formate per Plugin nachzurüsten oder das IndieBlocks-Plugin zu installieren.

Zusätzlich zur Pingback-Funktionalität ermöglicht das Webmention-Plugin, Kommentare über Webmentions zu beantworten. Antworten Sie über das WordPress-Kommentarformular auf eine Webmention, wird diese ebenfalls versendet und als Teil eines Kommentar-Threads angezeigt.

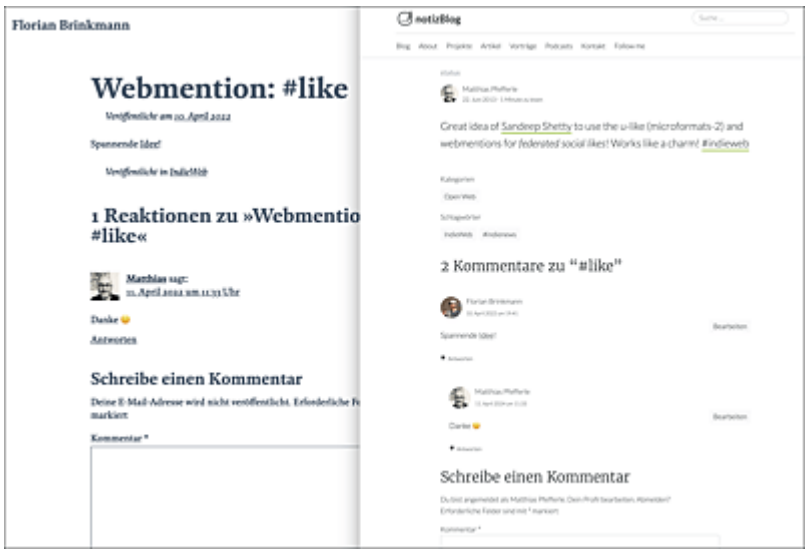


Abbildung 15.3 Ein Webmention-Beispiel für einen Kommentar-Thread

Das Plugin speichert Webmentions mithilfe desselben Mechanismus wie klassische Kommentare. Das bedeutet, dass Spamschutz-Plugins auch Webmentions prüfen und Ihnen die gleichen Moderationsmöglichkeiten zur Verfügung stehen.

Die Einstellungen des Plugins finden Sie im WP-Admin Bereich unter Einstellungen • Webmention oder unter IndieWeb • Webmention, falls Sie das IndieWeb-Plugin benutzen.

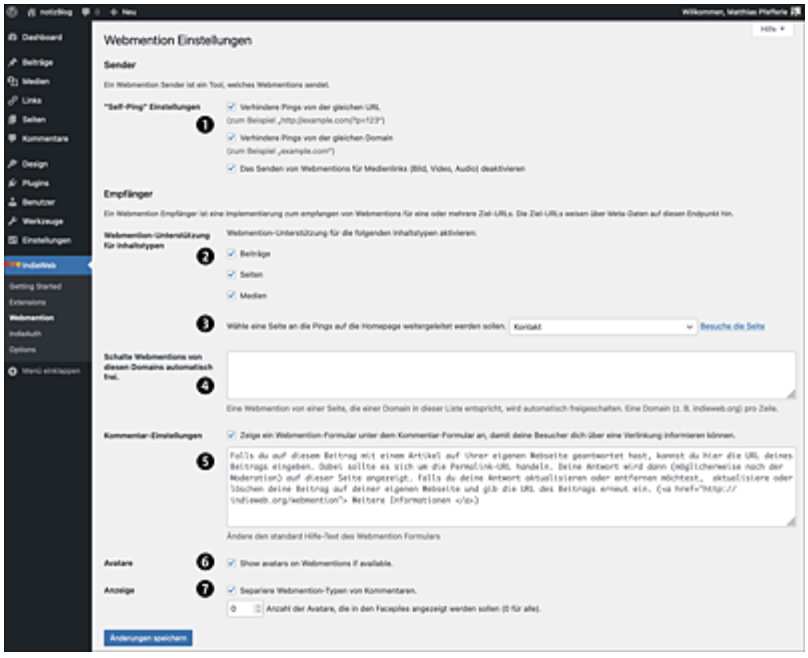


Abbildung 15.4 Die Einstellungen des Webmention-Plugins

Der Bereich Sender beinhaltet alle Einstellungen, die den Versand von Webmentions betreffen. Über die "Self-Ping" Einstellungen ❶ können Sie entscheiden, ob Sie Webmentions auch an Ihr eigenes Blog senden möchten. Die Beschreibungen sind hier selbsterklärend und sollten keine weiteren Fragen aufwerfen.

Im Bereich Empfänger können Sie verschiedene Einstellungen vornehmen, die beim Empfang von Webmentions angepasst werden sollen:

❷ Webmention-Unterstützung für Inhaltstypen: Legen Sie fest, für welche Beitragstypen Webmentions gesendet und empfangen werden sollen. Standardmäßig sind Beiträge und Seiten aktiviert.

❸ Wähle eine Seite, an die Pings auf die Homepage weitergeleitet werden sollen. Über eine Auswahlbox können Sie bestimmen, ob Pings an Ihre Homepage an eine bestimmte Seite weitergeleitet werden sollen, und an welche.

❹ Schalte Webmentions von diesen Domains automatisch frei. Webmentions von bestimmten, hier eingetragenen Domains werden automatisch freigeschaltet.

❺ Kommentar-Einstellungen: Sie können ein Webmention-Formular unter den Kommentaren anzeigen lassen, damit Ihre Leser*innen manuell Webmentions senden können. Außerdem lässt sich die Beschreibung im Formular individuell anpassen.

❻ Avatare: Bestimmen Sie, ob Avatare an Webmentions angezeigt werden sollen.

❼ Anzeige: Entscheiden Sie, ob bestimmte Mentions wie »Likes« oder »Reposts« gesondert als Facepile-Liste angezeigt werden sollen und nach wie vielen Mentions die Liste gekürzt wird.



Abbildung 15.5 »Facepile« Liste von »Like«-Webmentions

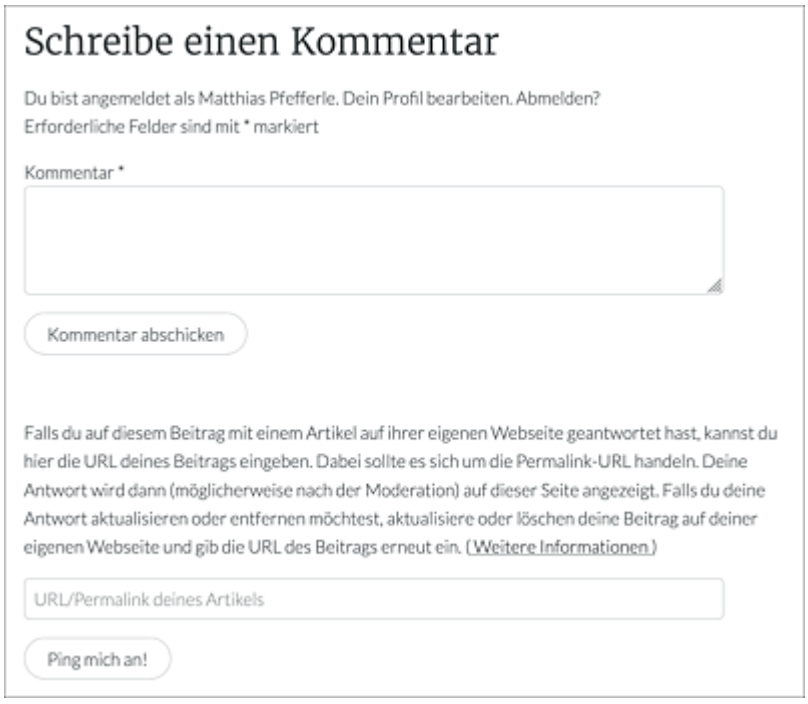


Abbildung 15.6 Das Webmention-Formular unter einem Kommentarfeld

Das Plugin arbeitet unauffällig im Hintergrund, sodass Sie Ihre gewohnte Routine beim Veröffentlichen von Beiträgen beibehalten können. Sie können auch weiterhin Apps oder andere Tools zum

Schreiben Ihrer Texte nutzen – Webmentions werden dadurch nicht eingeschränkt.

Plugin	Webmention
Aktive Installationen	900+
Download	*https://de.wordpress.org/plugins/webmention/*
Komplexität	++

IndieWeb-Plugin

Wenn Sie gut mit Webmentions klarkommen und Lust auf weitere Funktionen haben, bietet das IndieWeb-Plugin einen schönen Einstieg. Das Plugin ist eine Art Installer für andere, von der IndieWeb-Community entwickelte oder empfohlene Plugins. Darüber hinaus bietet es einen einfachen Einstieg in die Materie und eine Reihe von Einstellungsmöglichkeiten.

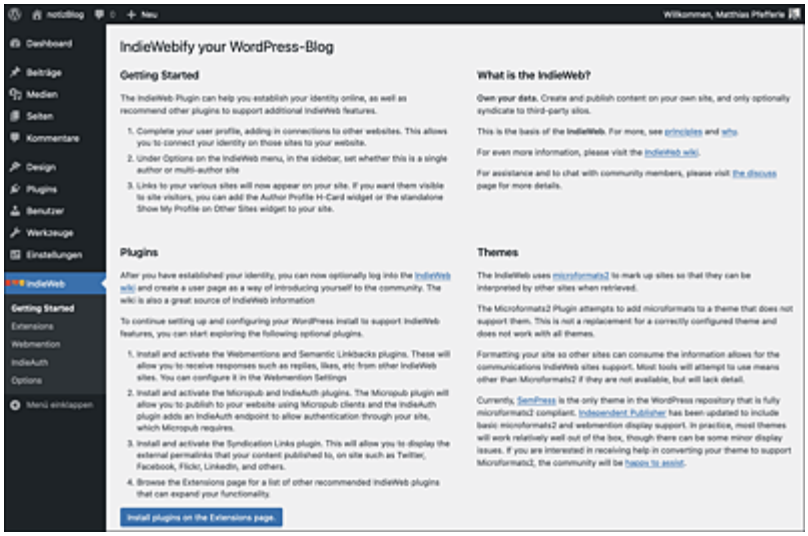


Abbildung 15.7 Die »Getting Started«-Seite des IndieWeb-Plugins

Getting Started erklärt, wie Sie das Plugin am besten einrichten und welche Schritte Sie als Nächstes unternehmen sollten. Außerdem beschreibt die Seite, wie Sie die weiteren Plugins installieren können und was Sie bei Themes berücksichtigen sollten.

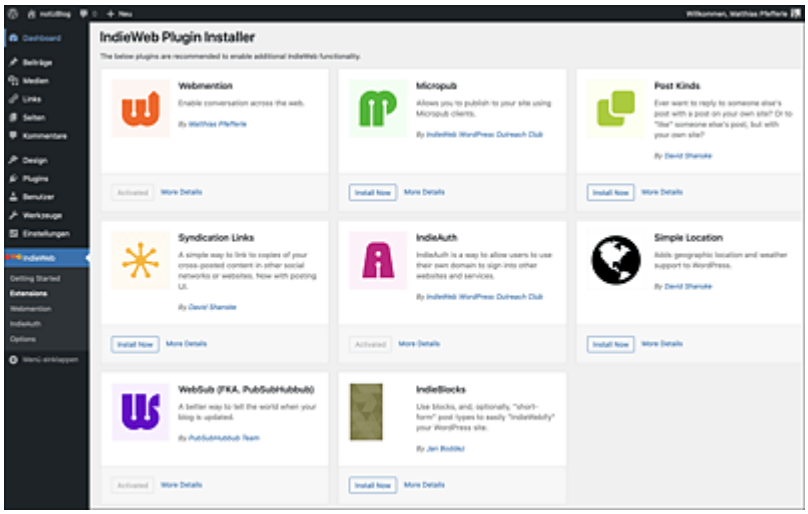


Abbildung 15.8 Die »Extensions«-Seite des IndieWeb-Plugins

Der Menüpunkt Extensions beinhaltet die Liste an empfohlenen Plugins mit einer kurzen Beschreibung und der Möglichkeit, sie direkt über die Seite zu installieren und aktivieren.

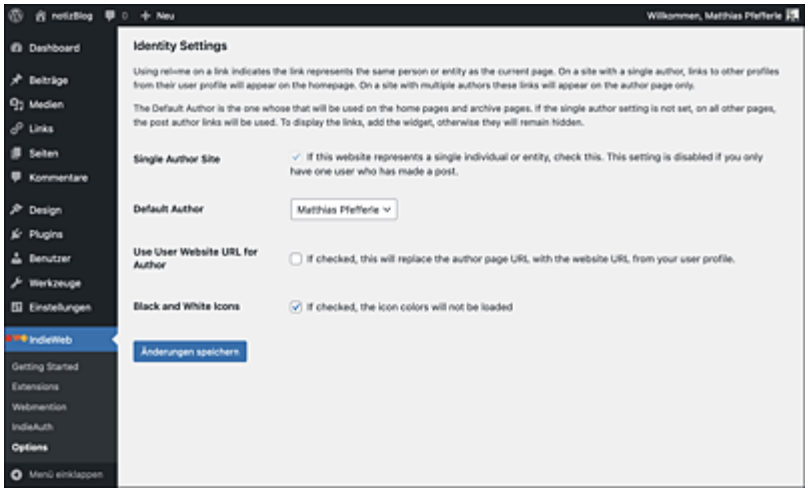


Abbildung 15.9 Die »Options«-Seite des IndieWeb-Plugins

Options hilft Ihnen bei ein paar Einstellungen, die die Nutzung der IndieWeb-Plugins vereinfachen und optimieren.

Plugin	IndieWeb
Aktive Installationen	600+
Download	*https://de.wordpress.org/plugins/indieweb/*
Komplexität	+

Weitere Plugins

Neben dem IndieWeb- und Webmention-Plugin gibt es eine Reihe weiterer Plugins. Sie finden alle folgenden Plugins aber auch über das IndieWeb-Plugin und können sich dort noch einmal alle Beschreibungen durchlesen oder die einzelnen Plugins einfach installieren und ausprobieren.

IndieAuth ist ein föderiertes Anmeldeprotokoll, das auf bewährten Technologien wie OAuth und OpenID basiert, jedoch den Prozess vereinfacht. Mit dem Plugin können Sie sich einerseits mit Ihrer eigenen Domain bei verschiedenen Diensten und Anwendungen anmelden und andererseits die Anmeldung und den Zugriff auf die WordPress API auf Ihrer Website über IndieAuth sichern.

Plugin	IndieAuth
Aktive Installationen	500+
Download	*https://de.wordpress.org/plugins/indieauth/*
Komplexität	++

Micropub ist ein offener API-Standard, mit dem Sie Beiträge auf Ihrer Website über Drittanbieter-Clients erstellen können. Web-Apps und native Apps (z. B. iPhone, Android) können Micropub verwenden, um kurze Notizen, Fotos, Ereignisse oder andere Beiträge auf Ihrer eigenen Seite zu veröffentlichen, ähnlich wie ein

X-Client auf *x.com* postet. Zur Authentifizierung wird zusätzlich das IndieAuth-Plugin benötigt.

Plugin	Micropub
Aktive Installationen	300+
Download	*https://de.wordpress.org/plugins/micropub/*
Komplexität	++

IndieBlocks bietet verschiedene Blöcke, mit denen Sie Microformats zu Beiträgen hinzufügen können, ohne HTML anpassen oder kennen zu müssen. Außerdem fügt es »Kurzform«-Beitragstypen hinzu, um Ihre Seite leichter als ein »Microblog« zu benutzen. Wenn Sie ein Block-Theme verwenden, wird IndieBlocks versuchen, das Theme mit »Microformats« anzureichern. In Kombination mit dem Webmention-Plugin ermöglicht es Ihnen, Notizen und Likes (sowie andere von Ihnen gewählte Beitragstypen) zu versenden.

Plugin	IndieBlocks
Verbreitung	60+
Download	*https://de.wordpress.org/plugin/indieblocks/*
Komplexität	+++

15.2 Das Fediverse

IndieWeb und Fediverse verfolgen das gleiche Ziel: die Unabhängigkeit von zentralen und geschlossenen Systemen und eine dezentrale und offene Kommunikation. In der Umsetzung dieses Ziels gehen beide Bewegungen aber unterschiedliche Wege. Während das IndieWeb auf die Kombination einfacher Protokolle und einer engen Verbindung zwischen Blogs und sozialen Netzwerken setzt, definiert das Fediverse ein eigenes Netzwerk, das bestehende Communitys nicht berücksichtigt, solange sie nicht das gleiche Protokoll verwenden.

Das Fediverse (*https://fediverse.info/*) ist die Idee eines dezentralen sozialen Netzwerks, das aus einer Vielzahl von unabhängigen Plattformen besteht, die miteinander kommunizieren und Inhalte austauschen können. Im Gegensatz zu zentralisierten sozialen Netzwerken wie Facebook oder Twitter, bei denen alle Benutzer*innen auf einer einzigen Plattform agieren, ermöglicht das Fediverse Benutzer*innen die Wahl zwischen verschiedenen Servern und Diensten, die auf unterschiedlichen Prinzipien und Werten basieren.

Eine der bekanntesten Plattformen im Fediverse ist Mastodon, ein dezentraler Mikroblogging-Dienst, der es Benutzer*innen ermöglicht, Nachrichten, Bilder und Videos in kurzen Beiträgen zu teilen. Mastodon basiert auf dem ActivityPub-Protokoll, das es verschiedenen Servern ermöglicht, miteinander zu kommunizieren und Inhalte auszutauschen. Nicht zuletzt durch Mastodon hat sich ActivityPub als Protokoll durchgesetzt, das Fediverse ist aber eigentlich agnostisch und wesentlich älter als Mastodon und ActivityPub.

Fediverse ist ein Kofferwort aus »federation« und »universe«, was man am besten als »föderiertes Universum« übersetzen kann. Der Begriff tauchte 1984 das erste Mal auf und wurde um das Jahr 2000 auch für diese Art von Netzwerken verwendet, zum Vergleich: Mastodon wurde 2016 veröffentlicht und ActivityPub erst 2018.

Die Seite *fediverse.party* bietet eine Übersicht über die verschiedenen Plattformen und Dienste im Fediverse. Hier sind einige Zahlen, die die Größe und Vielfalt des Fediverse verdeutlichen:

- ~ 12.000.000 Accounts
- ~ 2.000.000 aktive Benutzer*innen
- > 17.000 Server
- > 200 verschiedene Anwendungen

Auch wenn man ActivityPub nicht mit der Idee des Fediverse gleichsetzen kann, so ist es doch das Protokoll, das sich in den letzten Jahren durchgesetzt hat und mittlerweile von den meisten Plattformen im Fediverse unterstützt wird. Diese Entwicklung ist wichtig, da sie die Interoperabilität zwischen verschiedenen sozialen Netzwerken und Diensten im Fediverse gewährleistet.

Wegen der weiten Verbreitung von ActivityPub im Allgemeinen und in der WordPress-Community im Besonderen werden wir uns im Folgenden ausschließlich mit diesem Protokoll beschäftigen.

15.2.1 ActivityPub

ActivityPub (*https://activitypub.rocks/*) ist, wie schon angedeutet, ein offenes Protokoll, das für die dezentrale soziale Interaktion zwischen Benutzer*innen auf dem gleichen oder einem anderen

Server entwickelt wurde. Der Server und die Art der Plattform spielen dabei keine Rolle. ActivityPub wird mittlerweile von einer Vielzahl von Web-Anwendungen unterstützt, darunter soziale Netzwerke, Blogs, Foren, Bildergalerien, Wikis und mehr. Sie sind dadurch in der Lage, sowohl den Service-Anbieter als auch die Software relativ frei zu wählen, ohne sich dadurch einzuschränken. Durch ActivityPub können Sie Ihrem Blog auf Mastodon folgen, einen Kommentar auf Ihrem Blog über einen Forum-Post erhalten oder ein veröffentlichtes Bild auf Pixelfed liken.

ActivityPub basiert auf dem reinen Kommunikationsprotokoll und dem Format Activity Streams 2.0, das die Struktur und den Inhalt der Aktivitäten und Interaktionen definiert.

Protokoll

Man kann sich ActivityPub am besten als eine Art »E-Mail für soziale Netzwerke« vorstellen, Server stellen allen Benutzer*innen eine Inbox und eine Outbox bereit, und Aktivitäten werden über diese beiden Kanäle ausgetauscht.

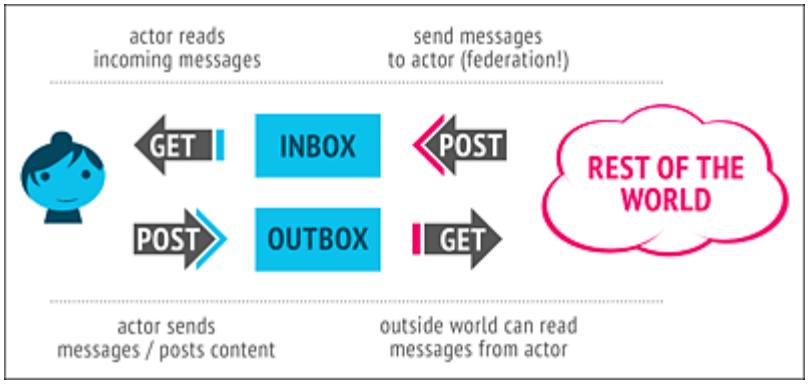


Abbildung 15.10 Schaubild zur Funktionsweise von ActivityPub

Die eigentliche Kommunikation erinnert dabei stark an das Abonnieren eines Newsletters, um bei dem Beispiel E-Mail zu bleiben. Wenn eine Person jemandem folgt, so abonniert sie die

Aktivitäten und erhält diese in ihrer Inbox. Anders als bei einem Newsletter bietet ActivityPub aber auch einen Rückkanal, über den die Person auf die Aktivitäten ihrer Follower*innen reagieren kann, die dann über ihre Outbox an die Inbox der anderen Person gesendet werden.

Wenn Sie sich das im Folgenden beschriebene WordPress-Plugin installieren, ist es Ihren Leser*innen möglich, Ihre Beiträge auf einer beliebigen, ActivityPub-fähigen Plattform zu »abonnieren«, um sie im Folgenden zu kommentieren, zu liken oder zu teilen. Diese Interaktionen werden dann wiederum auf Ihrer Website angezeigt.

Activity Streams 2.0

Activity Streams (*https://activitystrea.ms/*) hat seinen Ursprung im Web 2.0 und wurde 2011 als Erweiterung für RSS-/Atom-Feeds entwickelt, um diese mit mehr Informationen anzureichern. Eines der ersten Projekte, die Activity Streams als Basis für eine dezentrale Kommunikation einsetzten, war die Open-Source-Plattform StatusNet. StatusNet gilt als eines der ersten Fediverse-Projekte und Wegbereiter für das ActivityPub-Protokoll. Die Version 2.0 wurde 2017 veröffentlicht und 2018 als Basis für ActivityPub spezifiziert.

Der Aufbau einer Aktivität ist dabei relativ einfach und lässt sich auf drei wesentliche Informationen reduzieren:

- **Actor:** Der*die Akteur*in ist die Person oder Entität, die die Aktivität ausführt. Dies kann ein*e Benutzer*in, eine Gruppe oder eine Anwendung sein.
- **Type:** Der Typ beschreibt die Art der Aktivität, die ausgeführt wird, z. B. »posten«, »liken« oder »folgen«.

- **Object:** Das Objekt ist das Ziel der Aktivität, z. B. ein Beitrag, ein Kommentar oder ein Bild.

Darüber hinaus hat eine Aktivität auch eine Reihe von weiteren Eigenschaften, wie z. B. den Zeitpunkt der Aktivität, Tags, Anhänge oder eine Liste von Empfänger*innen, diese sind für das Verständnis des Protokolls aber nicht wichtig.

Ein Beispiel für eine Aktivität könnte wie folgt aussehen:

```
{
 "actor": "https://example.org/user",
 "type": "Like",
 "object": "https://example.com/post"
}
```

`actor` beinhaltet die URL der Person, die die Aktion ausgeführt hat, `type` gibt den Typ der Aktivität an (in diesem Fall ein »Like«), und `object` beinhaltet die URL des Beitrags, der geliket wurde.

Dieses Prinzip lässt sich auf alle möglichen Aktivitäten anwenden, von einfachen »Likes« und »Shares« bis hin zu komplexeren Interaktionen wie »Kommentieren« oder »Folgen«.

Ein weiteres Beispiel für eine Aktivität:

```
{
 "actor": "https://example.org/user",
 "type": "Follow",
 "object": "https://example.org/author"
}
```

Auch in diesem Beispiel beinhaltet das `actor`-Attribut die URL der Person, die die Aktion ausgeführt hat, `type` zeigt, dass es sich um ein »Follow« handelt, und `object` beinhaltet die URL der Person, der gefolgt wurde.

Sie können sich ActivityPub bzw. Activity Streams also wortwörtlich als einen »Stream« oder Fluss von Aktivitäten vorstellen, der zwischen den verschiedenen Inboxen und Outboxen der

Benutzer*innen hin- und herfließt und von der jeweiligen Plattform interpretiert und abgearbeitet wird.

15.2.2 WebFinger

Twitter (heute X) ist Vorreiter von vielen Mechanismen, die wir heute in fast allen modernen sozialen Netzwerken finden. Zwei der populärsten sind wahrscheinlich das Hashtag und das @-Reply. Um eine Person auf X anzusprechen, reicht es, ihren Benutzernamen mit einem @-Zeichen zu versehen. Hier ein Beispiel:

> *Ich schreibe gerade an einem Kapitel für das WordPress-Buch von @flobrinkmann und @krafit.*

Diese Art der Erwähnung funktioniert in dezentralen Netzwerken wie Mastodon natürlich nicht, da der Benutzername nicht eindeutig ist und auf verschiedenen Servern vorkommen kann. Um dieses Problem zu lösen, benutzt man auf Mastodon zusätzlich noch die Domain des jeweiligen Servers. Das obige Beispiel würde also auf Mastodon wie folgt aussehen:

> *Ich schreibe gerade an einem Kapitel für das WordPress-Buch von @florian@dewp.space und @simon@dewp.space.*

Man kann das @-Reply wie folgt interpretieren: Erwähne den Benutzer »florian« und den Benutzer »simon« auf dem Server »dewp.space«.

Diese Art von Benutzername, der im Prinzip wie eine E-Mail Adresse aufgebaut ist und dadurch auch wieder sehr gut zu unserer Analogie passt, nennt man eine WebFinger-»Ressource«. WebFinger ist die Technologie, die es ermöglicht, von einer solchen Ressource auf die eigentliche URL der Person zu schließen.

Hier ein Beispiel für eine WebFinger-Abfrage über den WebFinger.net-Service:

```
GET /lookup/?resource=florian%40dewp.space HTTP/1.1
Host: webfinger.net
```

Als Ergebnis erhält man unter anderem den Mastodon-Server, auf dem der Benutzer »florian« registriert ist:

```
{
 "subject": "acct:florian@dewp.space",
 "links": [{
  "rel": "self",
  "type": "application/activity+json",
  "href": "https://dewp.space/users/florian"
 }]
}
```

WebFinger kann noch mehr Informationen liefern, diese würden aber den Rahmen dieses Kapitels sprengen und eher für Verwirrung sorgen. Wenn Sie mehr über WebFinger erfahren möchten, empfehlen wir Ihnen, die Community-Seite *https://webfinger.net/* zu besuchen.

Info: WebFinger

WebFinger ist kein fester Bestandteil von ActivityPub, wird aber von vielen Plattformen wie z. B. Mastodon genutzt, um @-Replies über verschiedene Server hinweg abzubilden.

15.2.3 Plugins

Um WordPress mit dem Fediverse zu verbinden, gibt es eine Reihe von Plugins, die Ihnen helfen, die Vorteile sozialer Netzwerke direkt auf Ihrer Website zu nutzen.

ActivityPub-Plugin

Das Plugin verbindet Ihr WordPress-Blog mit beliebten sozialen Plattformen wie Mastodon, Pixelfed oder Threads und macht es damit einem größeren Publikum zugänglich. Einmal installiert, können Ihnen Benutzer*innen auf diesen Plattformen folgen, wodurch Ihre neuen Beiträge in ihren »Feeds« erscheinen.

Die Einstellungen des Plugins finden Sie im WP-Admin-Bereich unter Einstellungen • ActivityPub. Auf der Willkommen-Seite finden Sie eine kurze Einführung in das Plugin und eine Übersicht der wichtigsten Informationen.

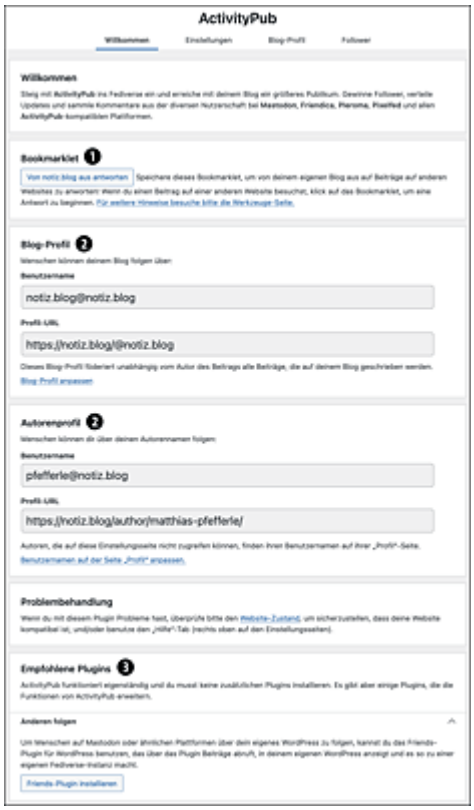


Abbildung 15.11 Die »Willkommen«-Seite des ActivityPub-Plugins

❶ Der Verweis auf ein Bookmarklet, mit dessen Hilfe Sie auf Beiträge, die Sie im Fediverse entdecken, reagieren können. Ausführliche Informationen finden Sie unter Werkzeuge • Verfügbare Werkzeuge.

❷ Abhängig von Ihren Einstellungen können Sie die Benutzernamen des Blog-Profils und darunter Ihren eigenen sehen, mit dem Verweis, wie Sie sie jeweils anpassen können.

❸ Am Ende der Seite finden Sie außerdem noch eine Liste an weiteren optionalen Plugins, die die Funktionalität des ActivityPub-Plugins erweitern.



Abbildung 15.12 Die »Einstellungen«-Seite des ActivityPub-Plugins

Unter dem Punkt Einstellungen können Sie das Plugin konfigurieren und an Ihre Bedürfnisse anpassen:

1. Profile nach Typ aktivieren: Legen Sie fest, welche Art von Profilen Sie zulassen wollen.
    - Nur Autorenprofile aktivieren: Die Autor*innen auf Ihrem Blog werden für ActivityPub freigeschaltet, und Blogposts werden im Namen des*der Verfasser*in geteilt.
    - Nur Blog-Profil aktivieren: Ihr Blog erhält ein zentrales Profil, und alle Beiträge werden darüber geteilt.
    - Beide Profile aktivieren: Artikel werden im Namen des*der Autor*in veröffentlicht, und das Blog-Profil teilt diese Beiträge zusätzlich noch einmal mit seinen Follower*innen.

2. Activity-Objekttyp: Legen Sie fest, welcher Objekttyp für die Aktivitäten verwendet werden soll.
    - Standardmäßig ist dies »Note«, was einem kurzen Text entspricht und mit den meisten sozialen Netzwerken kompatibel ist.
    - Über die Einstellung WordPress-Beitragsformat versucht das Plugin, den besten ActivityPub-Objekttyp anhand des WordPress-Beitragsformats zu ermitteln.

3. Unterstützte Inhaltstypen: Legen Sie fest, welche Beitragsarten für ActivityPub freigeschaltet werden sollen.

4. Hashtags: Legt fest, ob im Beitrag nach Hashtags gesucht werden soll, um diese dann als WordPress-Tags an den Beitrag anzuhängen.

Unter dem Punkt Profil können Sie Ihr Blog-Profil anpassen. Hier können Sie ein Profilbild, eine Beschreibung und eine Liste von Zusatzfeldern hinzufügen. Diese Informationen werden auf den Plattformen im Fediverse angezeigt, wenn Benutzer*innen Ihnen

folgen. Der Reiter Followers listet alle Benutzer*innen auf, die Ihrem Blog-Profil folgen. Sie können nach Follower*innen suchen oder diese löschen.

Die Einstellungen für Ihr eigenes Profil gleichen weitestgehend denen des Blog-Profils und sind unter Benutzer • Profil zu finden. Ihre Follower*innen finden Sie unter Benutzer • Follower und die Zusatzfelder unter Benutzer • Zusatzfelder.

Achtung: Besondere Anforderungen ans Hosting

Das ActivityPub-Protokoll hat ein paar Anforderungen, die nicht von jedem (Shared-) Hosting-Anbieter unterstützt werden. Das Plugin versucht, viele dieser bekannten Probleme zu erkennen und Ihnen entsprechende Lösungsmöglichkeiten anzubieten. Beobachten Sie nach der Installation den **Website-Zustand** unter **Werkzeuge** • **Website-Zustand**, und melden Sie sich gegebenenfalls im ActivityPub-Support-Forum: *https://wordpress.org/support/plugin/activitypub/*.



Abbildung 15.13 Die »Profil«-Seite des ActivityPub-Plugins

Neben diversen Anpassungsmöglichkeiten stellt das Plugin auch drei Blöcke bereit.



Abbildung 15.14 Der Block »Föderierte Antwort« im Block-Editor

Der Block Föderierte Antwort ist das Gegenstück des zuvor beschriebenen Bookmarklets. Über diesen Block können Sie auf andere Beiträge im Fediverse mit einem Blogpost reagieren. Fügen Sie dazu die URL des Beitrags in das vorgesehene Formularfeld ein, oder nutzen Sie einfach das Bookmarklet.

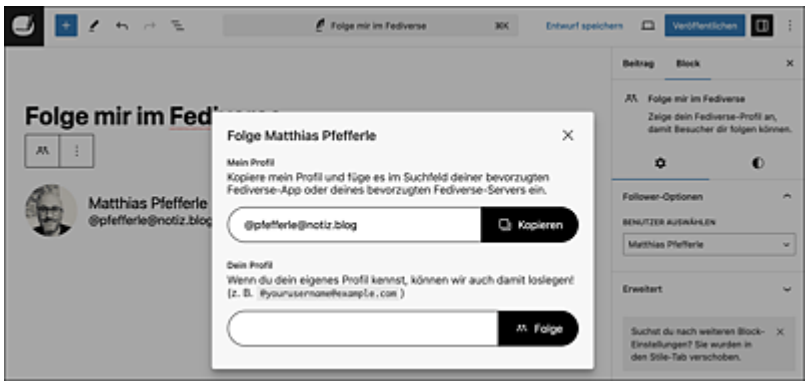


Abbildung 15.15 Der Block »Folge mir im Fediverse« mit Overlay im Block-Editor

Folge mir im Fediverse ist ein Block, der Ihre Leser*innen dabei unterstützen soll, Ihrem Blog auf Mastodon oder einer anderen Fediverse-Plattform zu folgen. Über den Block-Editor können Sie wählen, für welches Blog-Profil der Block generiert werden soll. Wählen Sie Dynamic User, versucht der Block, den Benutzer automatisch aus dem Kontext, in den er eingefügt wurde, zu erkennen. Das macht vor allem dann Sinn, wenn Sie den Block im Beitrags-Template oder auf der Autorenseite einsetzen wollen.

Der Fediverse-Follower-Block zeigt eine Liste Ihrer Follower*innen im Fediverse an. Sie können auch für diesen Block den*die Benutzer*in über den Block-Editor festlegen oder automatisch aus dem Kontext erkennen lassen.

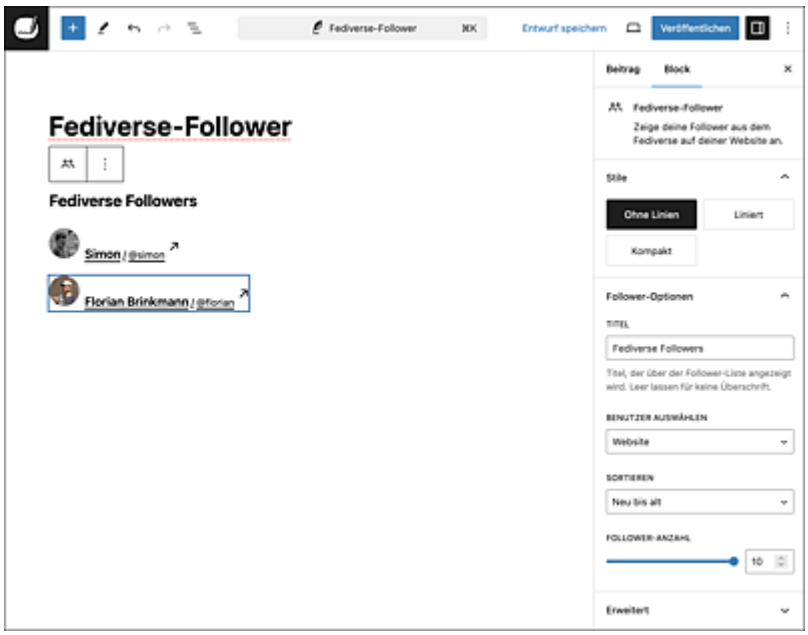


Abbildung 15.16 Der Block »Fediverse-Follower« im Block-Editor

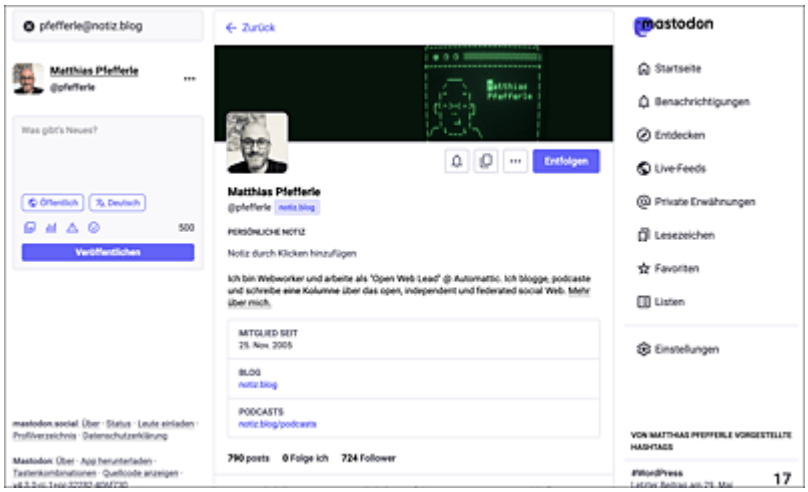


Abbildung 15.17 Ein Beispiel für die Anzeige eines Blog-Profils auf Mastodon

Um zu sehen, wie Ihr Profil auf Mastodon aussieht, geben Sie die Profil-URL oder die WebFinger-ID Ihres Profils (Sie finden diese auf der Willkommen-Seite des ActivityPub-Plugins) in die Suche der jeweiligen Seite ein. Klicken Sie in den Suchergebnissen auf Ihr Profil und dann auf den Folgen-Button.

Info: Leicht verzögerte Veröffentlichung

Es kann bis zu rund 15 Minuten dauern, bis der neue Beitrag in Ihrem Fediverse-Feed erscheint. Das liegt daran, dass die Nachrichten über einen verzögerten Cron an die verschiedenen Plattformen gesendet werden. Das soll verhindern, dass das Veröffentlichen für Benutzer*innen mit vielen Follower*innen unterbrochen wird. In den meisten Fällen wird der Post aber binnen weniger Minuten erscheinen.

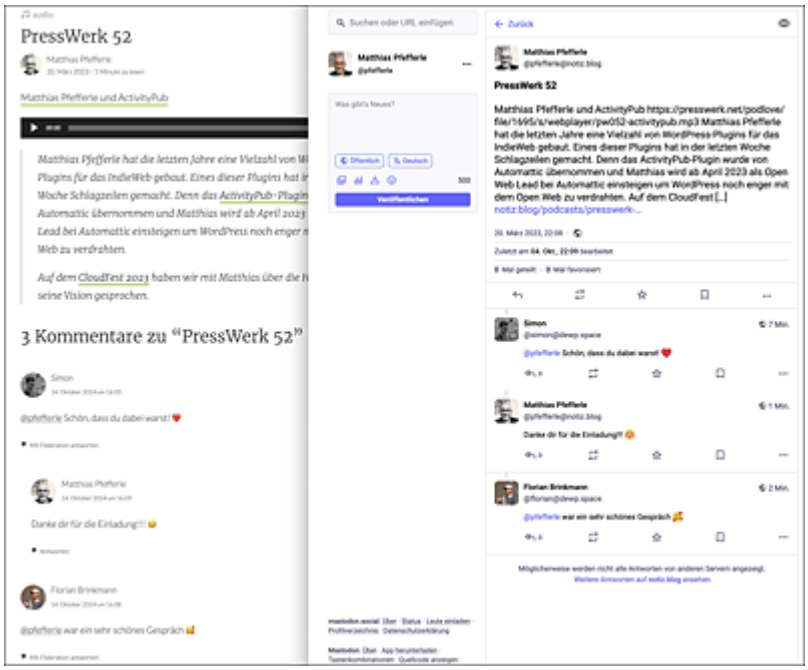


Abbildung 15.18 Ein Beispiel für föderierte Kommentare. Rechts die Reaktionen auf Mastodon und links die entsprechenden Kommentare auf der WordPress-Seite.

Ähnlich wie bei Webmentions erscheinen Reaktionen auf Ihre Beiträge im Fediverse als Kommentare auf Ihrer Website. Sie können direkt auf Ihrem Blog oder mit Ihrem Mastodon-Account darauf antworten, und diese Reaktionen werden dann an die entsprechenden Plattformen zurückgesendet. Aktuell unterstützt das Plugin nur Kommentare, aber zukünftig sollen auch Likes und Shares integriert werden.

Plugin	ActivityPub

Aktive Installationen	6.000+
Download	*https://de.wordpress.org/plugins/activitypub/*
Komplexität	++

Weitere Plugins

Prinzipiell benötigen Sie keine weiteren Plugins, da das ActivityPub-Plugin alle nötigen Grundfunktionen bereitstellt. Es gibt jedoch eine Reihe von Plugins, welche die Funktionalität von ActivityPub erweitern oder verbessern. Hier finden Sie eine Liste an Empfehlungen:

Das **Friends**-Plugin ermöglicht es Benutzer*innen, Freundschaften und Verbindungen zwischen verschiedenen Websites zu verwalten. Sie können Freund*innen per RSS folgen und in Kombination mit dem ActivityPub-Plugin auch Personen auf Mastodon sowie anderen ActivityPub-kompatiblen sozialen Netzwerken. Das ActivityPub-Plugin bietet aktuell noch keine Möglichkeit, anderen Fediverse-Accounts zu folgen. Friends schließt diese Lücke und macht aus Ihrem Blog eine vollwertige ActivityPub-Plattform.

Plugin	Friends
Aktive Installationen	1.000+
Download	*https://de.wordpress.org/plugins/friends/*
Komplexität	+++

Mit dem Plugin **Enable Mastodon Apps** können Sie sich über eine Mastodon-App direkt mit Ihrem WordPress-Blog verbinden und Ihre eigenen Beiträge anzeigen lassen – ideal, wenn Sie in einem

Blog mit mehreren Autor*innen arbeiten. Sie können auch einfach Beiträge mit Text und Anhängen erstellen, was Ihnen eine praktische Alternative zur WordPress-App bietet. In Kombination mit ActivityPub und Friends können Sie Ihren Feed einsehen und Personen auf Mastodon folgen sowie entfolgen. Das Plugin unterstützt zahlreiche Mastodon-Clients, darunter Ivory, Mammoth und die offizielle Mastodon-App.

Plugin	Enable Mastodon Apps
Aktive Installationen	400+
Download	*https://de.wordpress.org/plugins/enable-mastodon-apps/*
Komplexität	++

Das ActivityPub-Plugin bietet eine grundlegende Unterstützung für **WebFinger**. Wenn Sie jedoch mehr Kontrolle über diese Funktion benötigen oder mehrere WordPress-Plugins den WebFinger-Endpunkt verwenden, ist es ratsam, zusätzlich das WebFinger-Plugin zu installieren. Es erfordert keine Konfiguration und funktioniert sofort nach der Installation.

Plugin	WebFinger
Verbreitung	1.000+
Download	*https://de.wordpress.org/plugins/webfinger/*
Komplexität	+

NodeInfo ist ein einfaches Plugin, das Informationen über Ihren WordPress-Server im NodeInfo-Format bereitstellt. NodeInfo ist ein Standardformat für soziale Netzwerke im Fediverse, das es Servern ermöglicht, Informationen über ihre Benutzer*innen, Aktivitäten

und Funktionen auszutauschen. Ähnlich wie WebFinger bietet das ActivityPub-Plugin schon eine grundlegende Unterstützung für NodeInfo, sollten Sie aber das volle Potenzial von NodeInfo ausschöpfen wollen oder mehrere Plugins verwenden, empfiehlt es sich, das NodeInfo-Plugin zusätzlich zu installieren.

Plugin	NodeInfo
Verbreitung	1.000+
Download	*https://de.wordpress.org/plugins/nodeinfo/*
Komplexität	+

15.3 IndieWeb oder Fediverse?

Dieses Kapitel bietet lediglich einen kurzen Überblick über beide Themen, die jeweils genug Stoff für ein eigenes Buch bieten. Falls es Ihnen schwerfällt, sich für eines der beiden zu entscheiden, hier noch einmal eine kurze Empfehlung:

- Das **IndieWeb** ist ideal, wenn Sie aktiv auf klassischen sozialen Netzwerken wie Facebook, Twitter oder Instagram sind und diese nahtlos mit Ihrem Blog verknüpfen möchten.
- **ActivityPub** und das **Fediverse** sind die richtige Wahl, wenn Sie eine offene, dezentrale Alternative zu den etablierten sozialen Netzwerken suchen, weil Sie deren Geschäftsmodelle oder deren Umgang mit Ihren Daten kritisch sehen.

Sie müssen sich jedoch nicht für einen Ansatz entscheiden, denn beide Bewegungen (und auch alle Plugins) lassen sich problemlos kombinieren. Sie können also sowohl Webmentions als auch ActivityPub auf Ihrer Website nutzen und das Beste aus beiden Welten vereinen. Die Communitys überschneiden sich stark, und viele Plugin-Entwickler*innen sind in beiden aktiv oder stehen in regem Austausch miteinander.

Let's democratize the independent and federated social web!

TEIL IV

WordPress optimieren und warten

16 WordPress-Wartung und -Pflege

Wenn der initiale Stand Ihrer Website fertig und veröffentlicht ist, können beziehungsweise sollten Sie die Website nicht sich selbst überlassen. Regelmäßige Backups und Updates sollten eingeplant werden und benötigen nicht viel Zeit.

Wenn eine Website mit einem CMS betrieben wird, kann sie nie als wirklich fertig betrachtet werden, da für das System und eventuell genutzte Erweiterungen Updates veröffentlicht werden, die regelmäßig installiert werden sollten, da darunter auch hin und wieder Sicherheits-Updates sind. Für WordPress gilt da natürlich keine Ausnahme, weshalb Sie Ihrer Website regelmäßig ein wenig Aufmerksamkeit schenken sollten.

16.1 WordPress, Plugins und Themes aktualisieren

Die weite Verbreitung von WordPress hat nicht nur positive Seiten, sondern sorgt auch dafür, dass WordPress-Websites ein beliebtes Ziel für Hacker*innen sind. Bei einer großen Zahl Angriffsziele steigt die Wahrscheinlichkeit, dass Websites mit Sicherheitslücken darunter sind. Dabei ist das Sicherheitsproblem nur selten WordPress selbst, sondern es sind meist Plugins.

Aus diesen Gründen ist es eine gute Idee, alle Komponenten Ihrer Website auf einem aktuellen Stand zu halten, also sowohl den Core als auch Plugins und Themes. Wenn in Plugins oder Themes, die auf WordPress.org gehostet sind, schwere Sicherheitslücken auftreten, werden sogenannte Forced Updates ausgeführt, das heißt, es wird ein Update erzwungen und automatisch auf Websites installiert, auch wenn automatische Plugin- oder Theme-Updates nicht aktiv sind. Bei Produkten von anderen Quellen gibt es das nicht.

16.1.1 Kleine WordPress-Updates

Kleine Updates, die die dritte Zahl in der Versionsnummer betreffen, also beispielsweise 6.7.1 auf 6.7.2, macht WordPress seit Version 3.7 automatisch, wenn Sie dieses Verhalten nicht angepasst haben. Damit sind Sie auf diese Updates in der Regel gut vorbereitet.

16.1.2 Größere WordPress-Updates

Wenn alle paar Monate ein größeres Update für WordPress erscheint, ist bei älteren Installationen normalerweise ein manuelles Upgrade notwendig, da diese Updates nicht automatisch installiert werden. Das wurde in WordPress 5.6 geändert, sodass WordPress-Websites, die mit 5.6 oder neuer aufgesetzt sind (also nicht auf die Version aktualisiert, sondern damit neu erstellt wurden), auch für größere Updates die Aktualisierung automatisch vornehmen. Dieses Verhalten lässt sich anpassen, mehr dazu in Abschnitt 16.1.4, »Update-Verhalten konfigurieren«.

Tipp: Sicherstellen, dass Hintergrundupdates zeitnah ausgeführt werden

Automatische Aktionen wie das Installieren von Hintergrund-Updates oder das Veröffentlichen von geplanten Beiträgen werden in WordPress vom WP-Cron ausgeführt. Dieses System ist kein echter *Cronjob*, der auf Serverebene registriert ist und in einem definierten Intervall läuft – WP-Cron wird bei einem Seitenaufruf ausgeführt. Das bedeutet, ob und wie zeitnah ein solches Event ausgeführt wird, ist abhängig davon, wie regelmäßig Ihre Website aufgerufen wird.

Wenn Sie ein Hosting-Paket haben, das richtige Cronjobs enthält, können Sie den WP-Cron mit einem solchen Cronjob triggern und die Abhängigkeit von Seitenaufrufen entfernen. Wie ein Cronjob eingerichtet werden kann, hängt von Ihrem Hoster ab, hier wird deshalb nur ein möglicher Inhalt für einen Cronjob gezeigt und nicht darauf eingegangen, wie er konkret erstellt wird.

Cronjobs werden in einer Textdatei gespeichert und folgen dem Format, dass zuerst mit fünf Angaben das Intervall angegeben wird (Minuten, Stunden, Tage, Monate, Wochentage), gefolgt von dem Befehl, der ausgeführt werden soll. Auf der entsprechenden WordPress-Handbook-Seite (*https://developer.wordpress.org/plugins/cron/hooking-wp-cron-into-the-system-task-scheduler/*) wird empfohlen, mit dem Cron einfach die */wp-cron.php* aufzurufen, beispielsweise mit `wget`:

```
*/5 * * * * wget --delete-after https://example.com/wp-cron.php
```

Dieser Cronjob würde alle 5 Minuten die */wp-cron.php* aufrufen und damit fällige Jobs von WP-Cron ausführen.

Jetzt können Sie in der *wp-config.php* noch deaktivieren, dass WP-Cron bei Seitenaufrufen ausgeführt wird:

```
define( 'DISABLE_WP_CRON', true );
```

Damit haben Sie WP-Cron an einen System-Cronjob gehängt und sind nicht mehr abhängig davon, dass Ihre Website regelmäßig aufgerufen wird.

16.1.3 WordPress-6-Update für alte Systeme

Wenn Sie eine ältere WordPress-Instanz auf den neuesten Stand bringen möchten, an der schon länger nichts mehr gemacht wurde, ist das gegebenenfalls eine leicht spannende Aufgabe. Wenn Sie eine Staging-Umgebung haben, ist es eine gute Idee, die im Folgenden beschriebenen Schritte dort auszuführen. Haben Sie keine Staging-Umgebung, sollten Sie eventuell, je nachdem, wie kritisch ein Ausfall der Website durch ein Problem bei den Updates wäre, eine Staging-Umgebung aufsetzen. Wie das geht, wird in Kapitel 17 beschrieben.

Wenn es sich um eine recht kleine Website mit wenigen Plugins handelt, die nicht viel besucht wird, können Sie ein Backup machen (wie das geht, erfahren Sie später in diesem Kapitel), dann über Dashboard • Aktualisierungen alle Plugins und Themes aktualisieren und am Ende die aktuelle WordPress-Version installieren. Prüfen Sie danach im Frontend und Backend, ob alles funktioniert, und stellen Sie zur Not das Backup wieder her.

Wenn die Installation ein bisschen komplexer ist oder ein Ausfall der Website durch ein Problem bei den Updates nicht passieren sollte, gehen Sie schrittweise vor (wie erwähnt am besten erst in einer Staging-Umgebung):

1. **Backup**
   Damit Sie im Fall der Fälle auf den alten Stand zurückkönnen, wenn etwas schief geht. Testen Sie das Backup nach der

Erstellung, um sicherzugehen, dass Sie es auch wiederherstellen können.

2. **Plugins und Themes updaten und testen**
   Bevor Sie sich dem WordPress-Update widmen, aktualisieren Sie unter Dashboard • Aktualisierungen die installierten Plugins und Themes auf die neueste Version. Wenn hier etwas schiefgeht, versuchen Sie, in den Detailinformationen herauszufinden, ob das damit zusammenhängen kann, dass das Plugin die alte WordPress-Version nicht mehr unterstützt, sich das Problem also nach der Aktualisierung von WordPress erledigen könnte. Ist das nicht der Fall und das Plugin aus einem anderen Grund kaputt, müssen Sie sich wohl oder übel nach einer Alternative umschauen. Je nachdem, um was für ein Plugin es sich handelt, kann das einigen Aufwand nach sich ziehen, wenn Inhalte neu erstellt werden müssen.

3. **Backup**
   Haben Sie einen funktionierenden Stand mit aktuellen Plugins und Themes, erstellen Sie wieder ein Backup, damit Sie bei einem ab jetzt auftretenden Problem nicht mehr auf den Stand vor den Plugin- und Theme-Updates zurückkehren müssen.

4. **WordPress aktualisieren und testen**
   Aktualisieren Sie WordPress über Dashboard • Aktualisierungen auf die aktuelle Version. Ist das Update installiert, stellen Sie sicher, dass alles im Frontend und Backend weiter funktioniert wie geplant.

5. **Backup**
   Jetzt steht gegebenenfalls noch ein Update der PHP-Version aus. Machen Sie deshalb noch mal ein Backup.

6. **PHP-Version aktualisieren und testen**
   Im WordPress-Backend können Sie über Werkzeuge • Website-

Zustand im Tab Bericht den Bereich Server öffnen, um herauszufinden, welche PHP-Version aktiv ist (siehe Abbildung 16.1). Aktuell (November 2024) sind PHP 8.2 und 8.3 aktiv mit Updates unterstützte Versionen, PHP 8.1 bekommt noch bis Ende 2025 Sicherheitsupdates. Sie sollten also mindestens auf PHP 8.1 aktualisieren, besser aber noch auf PHP 8.2 oder 8.3.

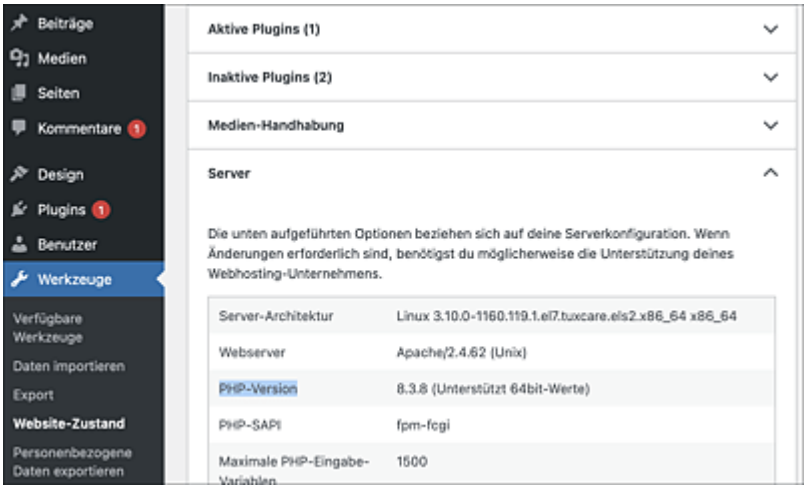


Abbildung 16.1 Die PHP-Version des Servers können Sie im »Website-Zustand«-Werkzeug einsehen.

Wie genau Sie die PHP-Version ändern können, hängt von Ihrem Hoster ab. Tasten Sie sich am besten schrittweise heran, und ändern Sie die Version nicht direkt auf PHP 8.3, sondern beginnen Sie mit 8.1. Läuft da noch alles ohne Probleme, können Sie 8.2 probieren und dann 8.3.

Sollte es schon bei 8.1 Probleme geben, versuchen Sie anhand der Fehlermeldung herauszufinden, welches Plugin oder Theme die Ursache ist. Wenn Sie den Grund gefunden haben, können Sie versuchen, im Support-Forum oder über ähnliche Kontaktmöglichkeiten nachzufragen, ob ein Update für die Unterstützung aktueller PHP-Versionen geplant ist. Falls nicht, sollten Sie sich nach einer anderen Lösung umschauen.

16.1.4 Update-Verhalten konfigurieren

Wer die automatischen Updates anders konfigurieren möchte als von der Standardeinstellung vorgesehen, kann zwischen automatischen Updates für kleine oder alle WordPress-Releases seit WordPress 5.6 über einen Link auf der Seite DASHBOARD • AKTUALISIERUNGEN umschalten (siehe Abbildung 16.2). Die Funktion für automatische Updates ist eine praktische Sache und kann zumindest für Minor-Updates eigentlich bedenkenlos aktiviert werden. So brauchen Sie sich beispielsweise keine Gedanken zu machen, Sicherheits-Updates schnell zu installieren, da es sich dabei um Minor-Versionen handelt und Sie diese dann automatisch bekommen.

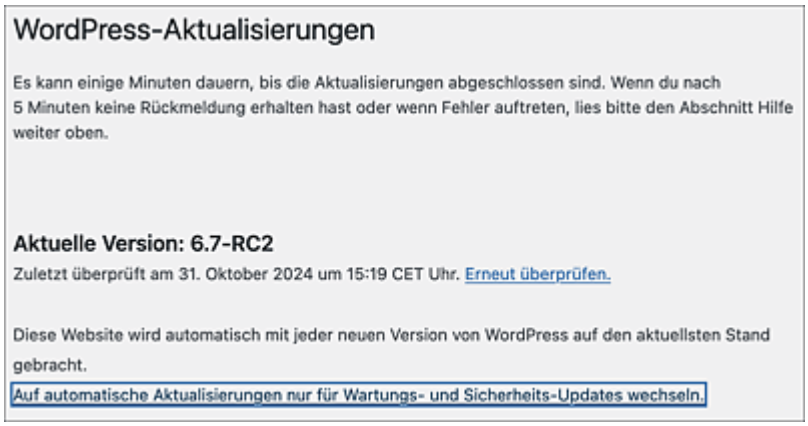


Abbildung 16.2 Über den hervorgehobenen Link am unteren Rand des Screenshots können die automatischen WordPress-Updates zwischen kleinen und allen Versionen umgestellt werden.

Wer die automatischen Updates lieber komplett deaktivieren oder den Modus der automatischen Updates via Code festlegen möchte, kann die Konfigurationskonstante `WP_AUTO_UPDATE_CORE` in der Datei `wp-config.php` im Hauptverzeichnis der WordPress-Installation setzen:

- `define( 'WP_AUTO_UPDATE_CORE', false );` deaktiviert alle automatischen WordPress-Updates.

- `define( 'WP_AUTO_UPDATE_CORE', 'minor' );` aktiviert nur die Updates, die die letzte Zahl der Version betreffen, also Sicherheits-Updates und kleine Wartungs-Updates.
- `define( 'WP_AUTO_UPDATE_CORE', true );` aktiviert alle Core-Updates, sogar für Entwicklungsversionen, wenn Sie sich dafür freigeschaltet haben (siehe Kasten). Wenn Sie Ihre WordPress-Installation nach WordPress 5.6 aufgesetzt haben, ist das der Standard-Wert Ihrer Installation.

Info: Beta-Versionen installieren

Für WordPress wird jeden Tag eine neue Entwicklungsversion veröffentlicht, eine sogenannte Nightly-Version, die den aktuellen Stand der Entwicklung abbildet. Der Prozess zur Veröffentlichung einer neuen stabilen WordPress-Version ist in eine Alpha-, eine Beta- und eine Release-Candidate-Phase eingeteilt, die einen zunehmend fertigen Stand der Software markieren (in der Alpha kann sich noch relativ viel ändern, Beta-Versionen sind für öffentliche Tests gedacht, um Fehler zu finden, und Release-Candidates sind Software-Stände, bei denen davon ausgegangen wird, dass sie so veröffentlicht werden könnten, aber noch abschließende Tests gemacht werden).

Plugin	WordPress Beta Tester
Verbreitung	2.000+
Download	*https://de.wordpress.org/plugins/wordpress-beta-tester/*
Zweck	ermöglicht die Installation von Entwicklungsversionen von WordPress
Komplexität	+

Durch die Installation des Plugins »WordPress Beta Tester« (*http://r-wrk.de/wp6_beta*) können Sie diese Vorabversionen auf Ihrer eigenen Installation testen. Hinweis: Das sollten Sie nicht auf einer Live-Website machen, da bei den Vorabversionen immer auch Fehler enthalten sein könnten. Das Plugin bietet auf der Einstellungsseite unter **Werkzeuge** • **Beta Testing** zwei unterschiedliche Aktualisierungs-Kanäle an:

- **Point Release (Minorversions-Update)** für den Entwicklungsstand des aktuell installierten Versionszweigs, also bei installierter Version 6.6.2 beispielsweise für 6.6.3.
- **Bleeding Edge (Hauptversions-Update)** für größere WordPress-Updates wie 6.6 auf 6.7.

Wenn Sie einen Kanal ausgewählt haben, können Sie unten in den Einstellungen eine der folgenden Stream-Optionen auswählen:

- **Nightlies** installiert das täglich erscheinende Nightly-Update mit dem neuesten Code. Bei diesen Versionen können schnell auch mal Fehler vorkommen.
- **Nur Beta/RC** (nur beim Kanal **Bleeding Edge**) installiert nur Updates, die den Beta- oder Release-Candidate-Status haben, also schon recht stabil sind.
- **Nur Release-Kandidaten** (nur beim Kanal **Bleeding Edge**) installiert nur Release-Candidate-Updates. Die sicherste Variante, was eventuelle Fehler in der Vorabversion angeht.

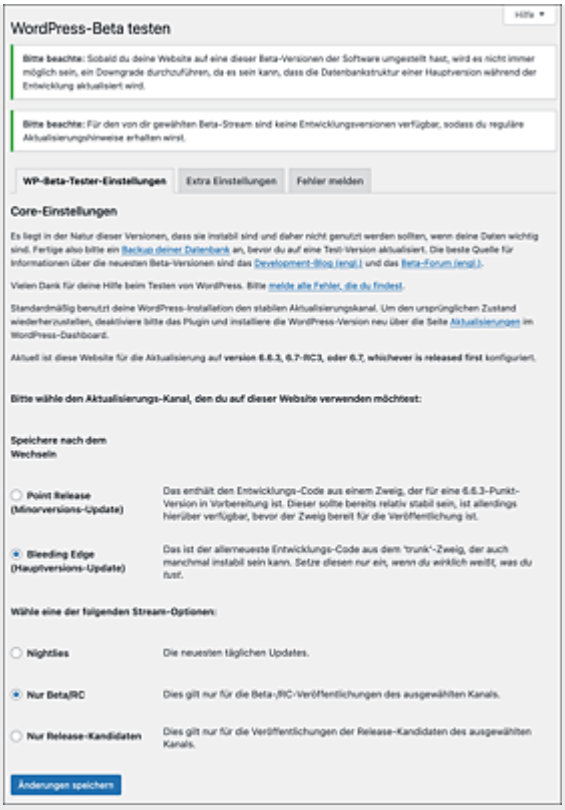


Abbildung 16.3 Das Plugin »WordPress Beta Tester« ermöglicht Ihnen recht einfach, Vorabversionen von WordPress zu testen.

Wenn Sie Plugins und/oder Themes entwickeln, sollten Sie in das Testing schon während der Beta-Phase einsteigen, um Zeit zu haben, auf Fehler zu reagieren und sie zu melden, falls es Probleme sind, die nicht auf Ihrer Seite liegen, sondern vom Core kommen.

Wenn Sie wieder zurück auf eine stabile Version wechseln möchten, können Sie das Plugin einfach deaktivieren und über **Dashboard • Aktualisierungen** die letzte stabile Version installieren. Unter Umständen ist das nicht möglich, wenn bei einer Vorabversion die Datenbankstruktur angepasst wurde. Dann müssen Sie auf die offizielle Version der Vorabversion warten oder ein Backup wiederherstellen.

Anstelle der `WP_AUTO_UPDATE_CORE`-Konstante können Sie auch Filter benutzen, um die automatischen Updates zu konfigurieren. Hier haben Sie auch die Möglichkeit, die Development-Versionen einzeln anzusprechen, nicht wie bei der Konstante nur zusammen mit den Major-Versionen.

- `add_filter( 'auto_update_core', '__return_true' );` aktiviert alle WordPress-Core-Updates.
- `add_filter( 'allow_major_auto_core_updates', '__return_true' );` aktiviert die Updates für Hauptversionen.
- `add_filter( 'allow_minor_auto_core_updates', '__return_true' );` aktiviert die Updates für die Minor-Versionen.
- `add_filter( 'allow_dev_auto_core_updates', '__return_true' );` aktiviert die Development-Versionsupdates, wenn Sie das Beta-Testing-Plugin installiert haben.

Tipp: Alle automatischen Updates deaktivieren

Wenn Sie automatische Updates komplett deaktivieren möchten, sowohl für Core als auch Plugins und Themes, gibt es zwei Varianten.

Einmal über eine Konstante:

```
define( 'AUTOMATIC_UPDATER_DISABLED', true );
```

Und über einen Filter:

```
add_filter( 'automatic_updater_disabled', true );
```

16.1.5 WordPress-Downgrade

Normalerweise geht es bei der Änderung einer Software-Version immer darum, eine neuere Version zu installieren. In bestimmten Fällen kann es aber notwendig sein, auf eine ältere Version zurückzuwechseln, etwa weil ein Plugin mit der neuen Version noch nicht funktioniert und nach einer Lösung gesucht werden muss. Dafür gibt es das Plugin »WP Downgrade«, und nach Installation und Aktivierung finden Sie die Einstellungen unter EINSTELLUNGEN • WP DOWNGRADE (siehe Abbildung 16.4).

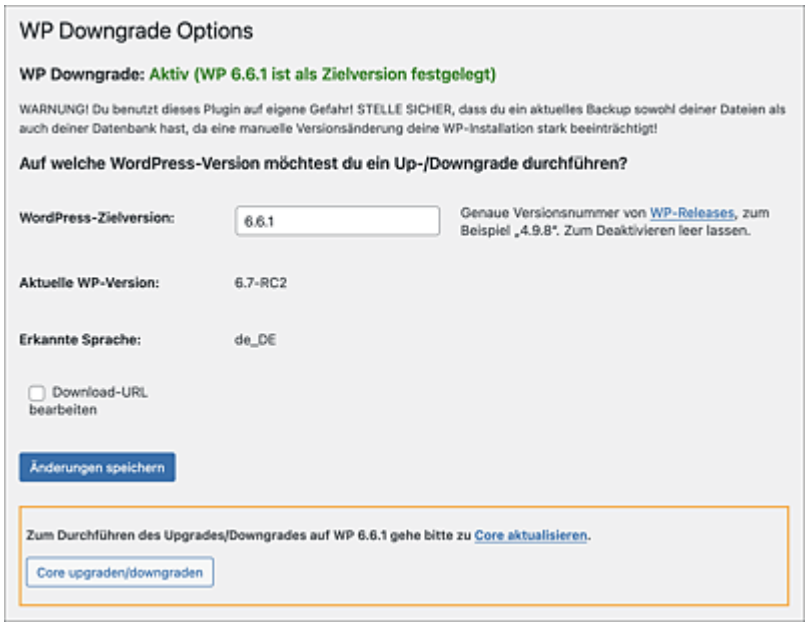


Abbildung 16.4 Mit dem Plugin »WP Downgrade« können Sie zu einer bestimmten WordPress-Version zurückwechseln.

Plugin	WP Downgrade
Verbreitung	100.000+
Download	*https://de.wordpress.org/plugins/wp-downgrade/*
Zweck	bietet die Möglichkeit, eine bestimmte ältere Version von WordPress zu installieren
Komplexität	+

Hier wählen Sie die exakte WordPress-Zielversion aus, auf die Sie wechseln möchten, und klicken auf Änderungen speichern. Danach erscheint unten auf der Einstellungsseite ein Button Core upgraden/downgraden, der Sie auf die Seite für die Aktualisierungen bringt, auf der Sie dann das Downgrade durchführen können, indem Sie auf Version X erneut installieren klicken.

16.1.6 Plugin- und Theme-Updates

Seit WordPress 5.5 gibt es ein User-Interface zum Aktivieren und Deaktivieren von automatischen Updates für Themes und Plugins. In der Plugin-Liste unter Plugins • Installierte Plugins gibt es in der rechten Spalte für jedes Plugin, das den normalen Update-Mechanismus von WordPress unterstützt, Links zum Aktivieren beziehungsweise Deaktivieren von automatischen Updates, wie in Abbildung 16.5 dargestellt.

Auch für Themes gibt es diese Funktion, zu finden in der Detailansicht eines Themes unter Design • Themes, wie in Abbildung 16.6 zu sehen.

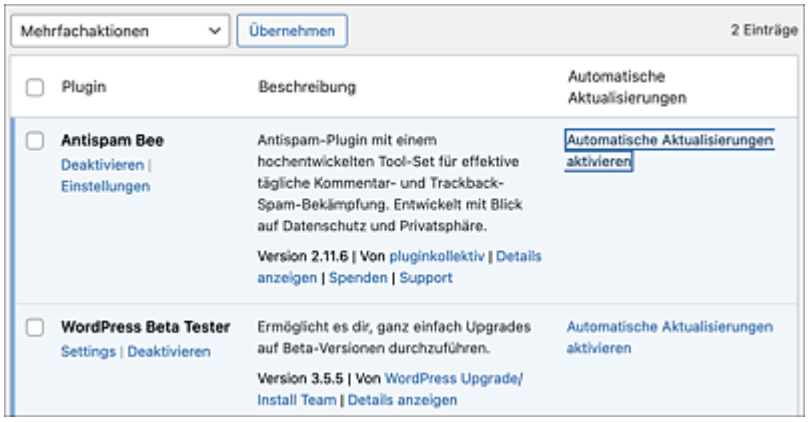


Abbildung 16.5 In der rechten Spalte der Plugin-Liste kann das Verhalten der automatischen Updates gesteuert werden.

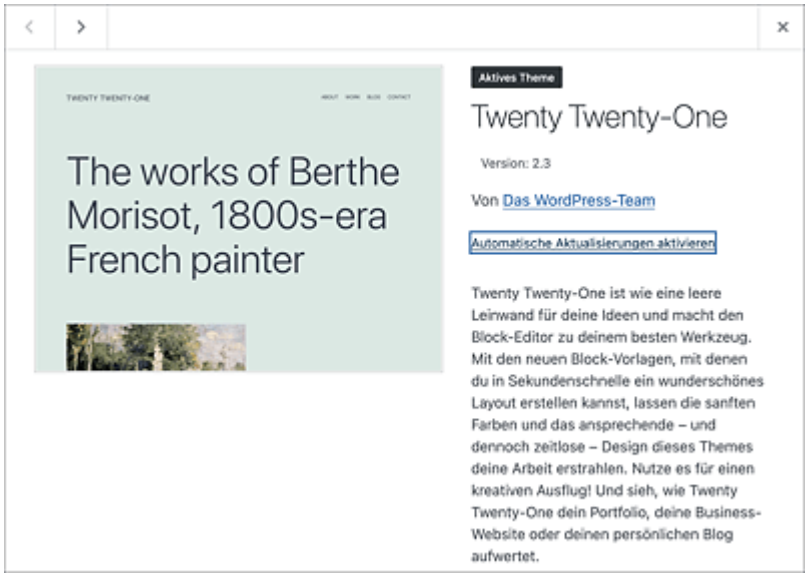


Abbildung 16.6 In dem Overlay mit Details zu einem Theme kann die automatische Aktualisierung für das Theme aktiviert oder deaktiviert werden.

Daneben können Sie das Verhalten auch über Filter steuern, wie im letzten Abschnitt die Core-Updates. Sie verwenden dabei folgende Hooks:

- `add_filter( 'auto_update_theme', '__return_true' );` aktiviert automatische Theme-Updates.
- `add_filter( 'auto_update_plugin', '__return_true' );` aktiviert Plugin-Updates.
- `add_filter( 'auto_update_translation', '__return_true' );` aktiviert Übersetzungs-Updates.

Mit `__return_false` als zweitem Parameter können Sie die jeweilige Einstellung deaktivieren.

16.1.7 Update-Benachrichtigungen steuern

WordPress verschickt bei verschiedenen Ereignissen um Updates herum E-Mails, um Sie zu informieren, ohne dass Sie einen Blick ins Backend werfen müssen:

- Ein WordPress-Update wurde durchgeführt.
- Ein Update ist verfügbar, wird aber nicht automatisch installiert.
- Es gab ein Problem bei einem automatischen Update.

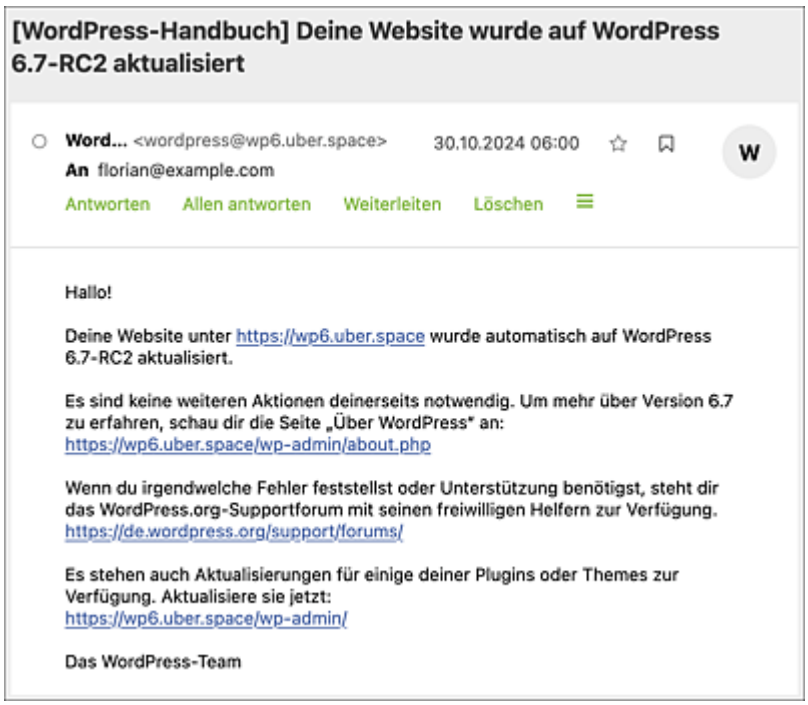


Abbildung 16.7 WordPress verschickt unter anderem E-Mail-Benachrichtigungen, wenn Updates durchgeführt wurden, damit Sie immer auf dem Laufenden sind.

Sie können über Filter einstellen, ob Sie bestimmte dieser E-Mails nicht mehr bekommen möchten:

- `add_filter( 'auto_core_update_send_email', '__return_false' );` deaktiviert die Benachrichtigungen nach automatischen WordPress-Updates.
- `add_filter( 'send_core_update_notification_email', '__return_false' );` deaktiviert die Benachrichtigungen über verfügbare Updates.
- `add_filter( 'automatic_updates_send_debug_email', '__return_false' );` deaktiviert E-Mails mit Debug-

Informationen, die standardmäßig nach Auto-Updates auf Entwicklungsversionen verschickt werden.

Über `__return_true` können Sie die Benachrichtigungen explizit aktivieren.

16.2 Wartungsmodus aktivieren – Plugin »Slim Maintenance Mode«

Wenn Sie größere Änderungen am Frontend vornehmen, die ihre Zeit brauchen, kann es sinnvoll sein, das Frontend in einen Wartungsmodus zu versetzen, sodass Besucher*innen eine Meldung angezeigt bekommen, dass die Website sich im Wartungsmodus befindet und bald wieder erreichbar ist. Bei einer bestehenden Website sollten Sie diesen Modus nicht allzu lange nutzen, da sich das auch auf das Ranking in Suchmaschinen auswirken könnte.

Plugin	Slim Maintenance Mode
Verbreitung	10.000+
Download	*https://de.wordpress.org/plugins/slim-maintenance-mode/*
Zweck	zeigt statt des Frontends Ihrer Website eine Wartungsmeldung an
Komplexität	+

Plugin-Lösungen zur Erstellung einer Wartungsseite gibt es einige. »WP Maintenance« beispielsweise bietet Ihnen viele Möglichkeiten, um eine schöne Wartungsseite zu gestalten, den Text anzupassen und mehr. In erster Linie soll diese Wartungsseite ja aber nur darüber informieren, dass hier momentan nichts zu sehen ist, und schnell wieder überflüssig sein, weil die Arbeiten abgeschlossen wurden. Daher wird hier die schlanke Lösung »Slim Maintenance Mode« vorgestellt (*https://de.wordpress.org/plugins/slim-maintenance-mode/*).

Wenn Sie das Plugin installieren und aktivieren, ist der Wartungsmodus bereits aktiv und erlaubt nur Benutzer*innen mit Admin-Berechtigungen den Zugriff auf die Website. Für alle anderen wird eine kurze Info angezeigt (siehe Abbildung 16.8).

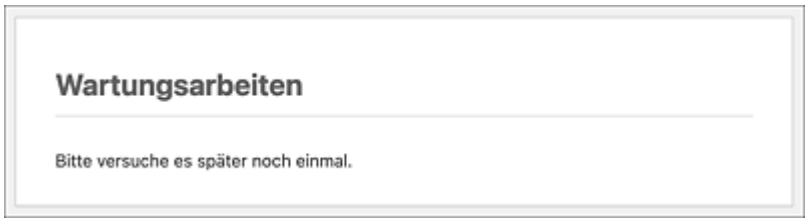


Abbildung 16.8 Das Plugin »Slim Maintenance Mode« bringt keine Einstellungsmöglichkeiten mit, sondern zeigt einfach eine kurze Info an, dass hier momentan nichts zu sehen ist.

Einstellungen oder Möglichkeiten der Anpassung der Nachricht im Frontend gibt es nicht. Im Backend wird eine Admin-Notice angezeigt, wenn das Plugin aktiv ist, damit es nicht versehentlich vergessen wird.

16.3 Backups planen und durchführen

Backups, also Kopien der Website, werden gern vernachlässigt, weil sie im Normalfall nicht gebraucht werden. Wenn aber doch einmal etwas bei einem Update schiefgeht oder Sie aus einem anderen Grund einen älteren Stand wiederherstellen müssen, sollte besser ein Backup vorhanden sein. Und dank Plugin-Lösungen für automatische Backups müssen Sie auch nicht ständig daran denken, Backups anzulegen, sondern sich nur einmal um die Einrichtung kümmern.

Achtung: Backups testen

Testen Sie regelmäßig, ob die Backups auch funktionieren, indem Sie sie lokal oder woanders einmal wiederherstellen. Wenn die Backups aus irgendeinem Grund defekt sind, sind sie genauso hilfreich wie gar kein Backup. *Ebenfalls wichtig ist*, dass Sie Backups nicht auf demselben Server speichern sollten, auf dem Ihre Website läuft. Andernfalls haben Sie keine Backups, wenn die Daten vom Server verloren gehen.

16.3.1 Wann Backups erstellt werden sollten

Am einfachsten ist es, wenn Sie ein regelmäßiges Backup einrichten, das beispielsweise einmal pro Woche die Daten der Installation sichert. Wenn Sie oft Änderungen an Ihrer Website vornehmen, kann es auch sinnvoll sein, täglich ein Backup zu machen, damit Sie im Fall der Fälle nicht zu viele Inhaltsänderungen verlieren.

Manuelle Backups können Sie dann beispielsweise noch anstoßen, wenn Sie ein größeres Update oder eine andere größere Änderung an der Website vornehmen.

16.3.2 Wie lange Backups aufbewahrt werden sollten

Bei der Überlegung, wie lange Sie Backups aufbewahren möchten, sollte eine Rolle spielen, wie lang es schlimmstenfalls dauert, bis Sie einen Fehler an der Website bemerken, der nicht so offensichtlich ist wie »Die gesamte Website ist nicht mehr erreichbar«.

Wenn Sie nur selten Änderungen an Ihrer Website machen und damit ein versteckter Fehler im Backend vielleicht erst nach einem Monat auffällt, lohnt es sich, so viele Backups aufzubewahren, dass Sie zur Not vielleicht eineinhalb Monate zurückkönnen. Wenn Sie sehr aktiv an Ihrer Website arbeiten und damit Fehler wahrscheinlich recht schnell auffallen, kann es reichen, Backups der letzten Woche aufzubewahren.

16.3.3 Backups anlegen und verwalten – Plugin »BackWPup«

Backup-Plugins gibt es viele, hier lernen Sie die kostenlose Version von BackWPup kennen. Das Plugin hat auch eine Bezahlversion, die unter anderem weitere Backup-Ziele anbietet.

Plugin	BackWPup
Verbreitung	600.000+
Download	*https://de.wordpress.org/plugins/backwpup/*
Zweck	Plugin zum Erstellen und Planen von Backups

Komplexität ++

Nachdem Sie das Plugin installiert und aktiviert haben, wechseln Sie im linken Admin-Menü auf den Punkt BackWPup • Dashboard. Hier finden Sie unterschiedliche Informationen rund um das Plugin, aber auch einen hilfreichen Button, um direkt ein Datenbank-Backup herunterladen zu können (siehe Abbildung 16.9), was manchmal recht nützlich sein kann und deutlich schneller geht, als vorher einen Backup-Auftrag zu erstellen.

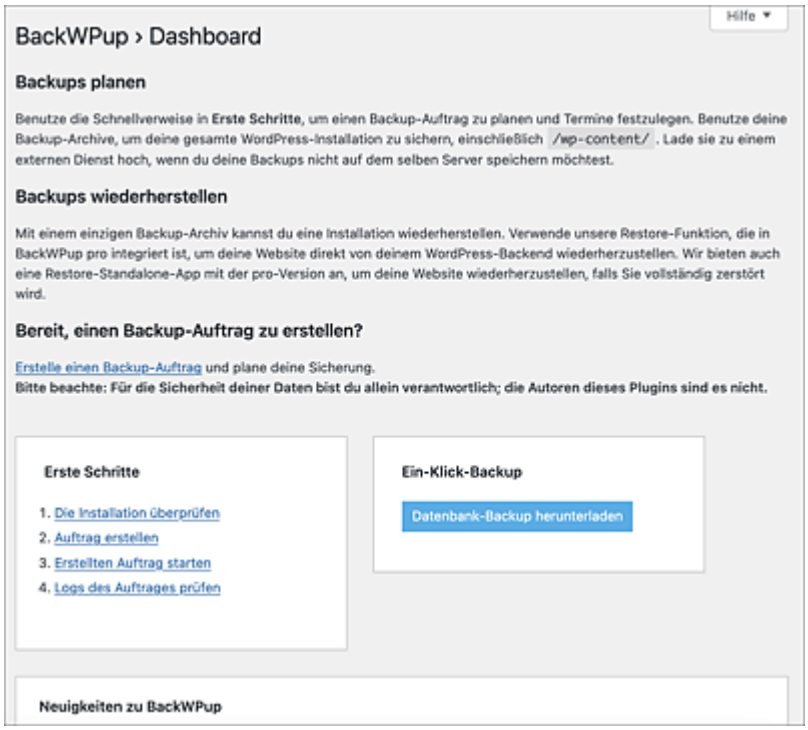


Abbildung 16.9 Funktionen bietet das Dashboard kaum, aber einen nützlichen Button, um direkt ein Datenbank-Backup herunterzuladen.

Interessanter wird es auf der Seite BackWPup • Neuer Auftrag. Hier legen Sie einen Backup-Auftrag an, indem Sie sich durch ein paar Tabs klicken.

- Zuerst geöffnet ist der Tab Allgemein, in dem Sie allgemeine Einstellungen zum Backup vornehmen (siehe Abbildung 16.10). So vergeben Sie einen Namen für den Auftrag und stellen ein, was

genau gesichert werden soll. Hier haben Sie die Wahl zwischen Datenbank-Backup, Dateien-Backup, WordPress-XML-Export, Liste der installierten Plugins und Datenbank-Tabellen prüfen. Besonders wichtig ist ein Backup der Datenbank und der Dateien, hier besonders des *wp-content*-Ordners. Haken Sie also zumindest diese beiden Optionen an. Sie können natürlich auch separate Backup-Aufträge anlegen, also einen für die Datenbank und einen für die Dateien.



Abbildung 16.10 Im Tab »Allgemein« legen Sie allgemeine Aspekte des Backup-Auftrags fest, wie etwa, wo das Backup gespeichert wird.

Im Abschnitt Erstellen der Backup-Datei können Sie das Format des Backup-Dateinamens anpassen und auswählen, in welchem Format das Archiv gespeichert werden soll. Vorausgewählt ist hier Zip, was wahrscheinlich die unkomplizierteste Variante ist, wenn Sie das Archiv entpacken müssen.

Unter Zielverzeichnis des Auftrags können Sie einstellen, wo das Backup gespeichert werden soll. Wie weiter oben kurz erwähnt sollten Sie das Backup nicht auf dem Webserver speichern, um zu verhindern, dass bei einem Datenverlust auch die Backups weg sind. Für das Beispiel nutzen wir die Option Backup in die Dropbox.

Im unteren Bereich Protokoll-Dateien können Sie eine E-Mail angeben, an die ein Protokoll des Backups geschickt werden soll, und steuern, ob das Protokoll nur im Fehlerfall verschickt werden soll. Drücken Sie auf Änderungen speichern, und wechseln Sie in den nächsten Tab.

- Der Planen-Tab ermöglicht Ihnen einzustellen, ob das Backup automatisch ausgeführt werden soll und, falls ja, was der Auslöser ist (siehe Abbildung 16.11).



Abbildung 16.11 Im »Planen«-Tab können Sie festlegen, ob das Backup automatisch ausgeführt werden soll.

Standardmäßig wird der Auftrag NUR MANUELL ausgeführt, Sie können das Backup aber auch über den WP-Cron ausführen lassen (wie Sie die Ausführung des WP-Cron mit einem echten Cronjob verbinden, erfahren Sie weiter vorne in diesem Kapitel im Abschnitt zu WordPress-Updates) oder über einen Link. Wählen Sie die Option MIT WORDPRESS CRON aus, stellen Sie in dem Date-Picker, der unten erscheint, das gewünschte Intervall ein, und speichern Sie die Änderung.

- Im Tab DATENBANK-BACKUP können Sie einstellen, welche Datenbank-Tabellen gesichert werden sollen, wie die Backup-Datei heißen soll und ob die Datei komprimiert werden soll (siehe Abbildung 16.12). In der Regel möchten Sie alle Tabellen sichern, es sei denn, es liegen Tabellen in der Datenbank, die nicht mit zu der WordPress-Installation gehören.

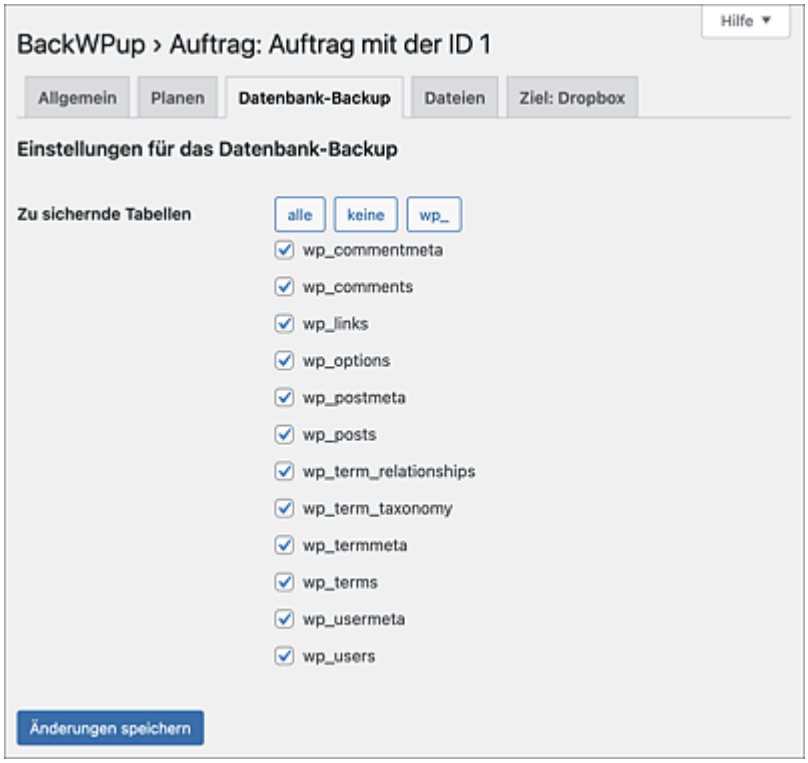


Abbildung 16.12 Hier können Sie einstellen, welche der Datenbank-Tabellen gesichert werden sollen.

- Unter Dateien können Sie festlegen, welche Dateien mitgesichert werden sollen (siehe Abbildung 16.13). Die WordPress-Installations-Ordner sind nicht unbedingt notwendig, da Sie WordPress im Fall der Wiederherstellung neu installieren können, aber einfacher ist es natürlich, wenn Sie die Dateien mitsichern. Die Einstellungen können hier eigentlich bleiben, wie sie von BackWPup vorgeschlagen werden. Schauen Sie aber etwa bei den Punkten Sicherung des Ordners wp-content und Uploads-Verzeichnis sichern, ob dort von Plugins erstellte Ordner aufgelistet werden, die Sie nicht benötigen, zum Beispiel von Duplicator oder WP Staging.



Abbildung 16.13 Im Tab »Dateien« können Sie anpassen, welche Verzeichnisse und Dateien im Dateien-Backup enthalten sein sollen.

Weiter unten haben Sie noch die Optionen, bestimmte Dateien und/oder Ordner nach Namen auszuschließen, und zwei spezielle Optionen, nämlich ein paar wichtige Dateien (*wp-config.php*, *.htaccess* etc.) mitzusichern, auch wenn das Hauptverzeichnis von

der Sicherung ausgeschlossen sein sollte. Außerdem können Sie Einen Ordner höher als WP-Installations-Ordner festlegen, was bedeutet, dass BackWPup nach Speichern der Änderungen auch den Ordner auf dem Webserver mit einbezieht, in dem das Verzeichnis mit WordPress liegt. So können Sie auch Dateien in diesem Verzeichnis mitsichern, die dort liegen und mit der WordPress-Installation zu tun haben.

- Im Tab Ziel: Dropbox müssen Sie Ihre Website mit Ihrem Dropbox-Konto verknüpfen (siehe Abbildung 16.14).

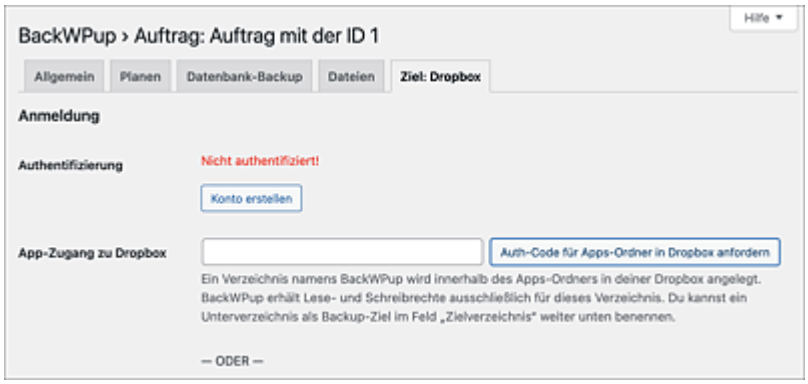


Abbildung 16.14 Im Tab zum Speichern in der Dropbox gibt es den Button »Auth-Code für Apps-Ordner in Dropbox anfordern«, mit dem Sie den Prozess starten können, um Ihrer Website Zugriff auf die Dropbox zu geben.

Klicken Sie dafür auf den Button Auth-Code für Apps-Ordner in Dropbox anfordern, der Sie zu Dropbox leitet. Melden Sie sich an, falls noch nicht geschehen, und klicken Sie im Dialog auf Zulassen, der Sie fragt, ob Sie der App Inpsyde BackWPup Zugriff auf den Ordner Apps • InpsydeBackWPup in Ihrer Dropbox gewähren möchten (siehe Abbildung 16.15). Nach einem Klick auf Zulassen bekommen Sie einen Code angezeigt, den Sie in dem BackWPup-Tab in das Feld App-Zugang zu Dropbox eintragen. Wenn Sie nun die Änderungen speichern, sollte bei Authentifizierung der Hinweis Authentifiziert erscheinen (siehe Abbildung 16.16).

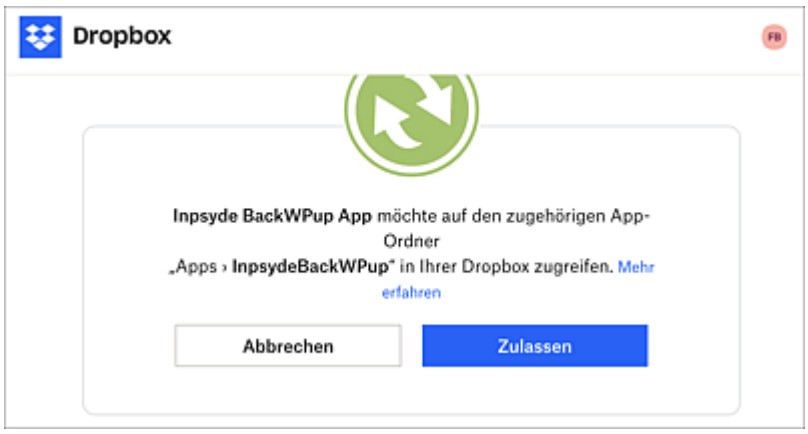


Abbildung 16.15 Sie müssen Ihrer Website Zugriff auf Ihre Dropbox geben, damit die Backups von Ihrer Website in der Dropbox gespeichert werden können.

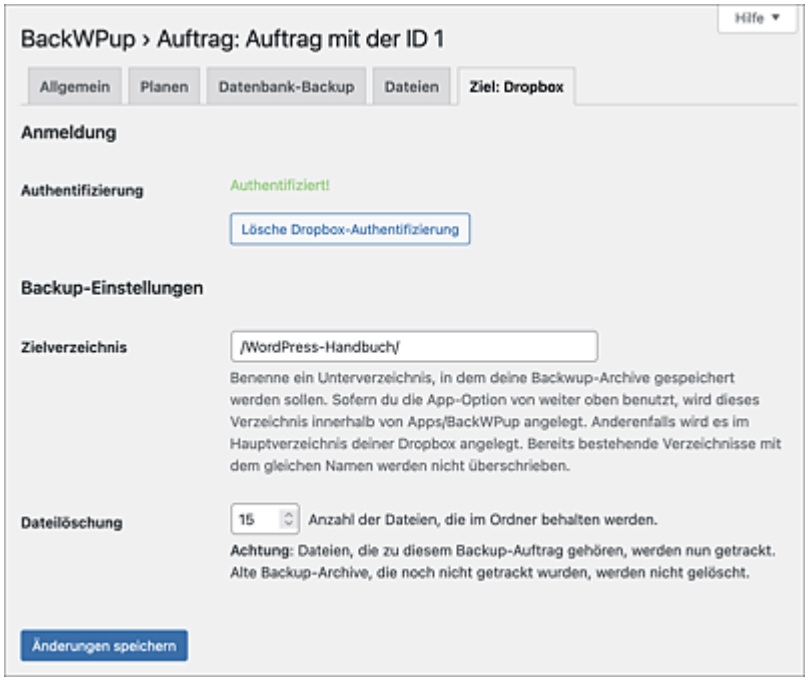


Abbildung 16.16 So oder so ähnlich sollte der Dropbox-Tab nach der erfolgreichen Verbindung mit Ihrem Dropbox-Konto aussehen.

Wenn Sie im ALLGEMEIN-Tab noch weitere Dinge angehakt haben, die gesichert werden sollen, finden Sie gegebenenfalls noch weitere Tabs vor. Für unser Beispiel sind alle Einstellungen vorgenommen, und Sie können links im Admin-Menü auf den Punkt BACKWPUP • AUFTRÄGE wechseln. Dort sollten Sie den neu erstellten Auftrag finden (siehe Abbildung 16.17).

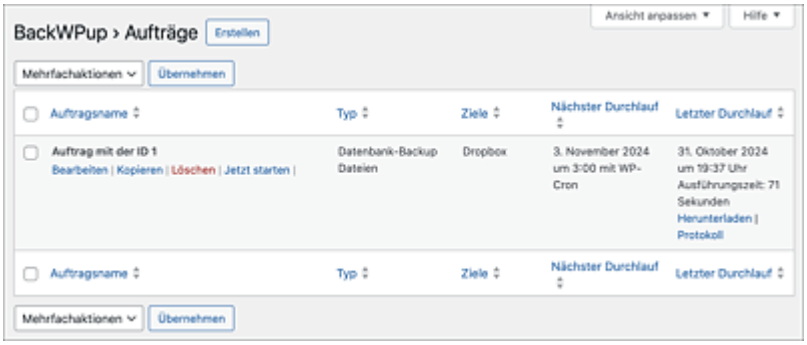


Abbildung 16.17 In der Übersicht der Aufträge können Sie unter anderem einen Auftrag direkt starten, auch wenn er regelmäßig automatisch ausgeführt wird.

Wenn Sie den Cursor über den Eintrag bewegen, sehen Sie Aktionslinks, wie Sie es schon aus den anderen Übersichtsseiten im WordPress-Backend kennen, etwa von den Seiten und Beiträgen. Hier finden Sie den Punkt Jetzt starten, mit dem Sie den Auftrag direkt starten können. In der Spalte Nächster Durchlauf wird Ihnen bei einem geplanten Auftrag angezeigt, wann er das nächste Mal ausgeführt wird, und daneben sehen Sie in der Spalte Letzter Durchlauf, wann der Auftrag zum letzten Mal ausgeführt wurde und wie lange er gebraucht hat. Außerdem können Sie mit einem Klick auf Herunterladen in die Backups-Liste springen (Sie erreichen diese Liste auch links im Admin-Menü über BackWPup • Backups), wo Ihnen die erstellten Backups angezeigt werden (siehe Abbildung 16.18). Hier können Sie die Backups herunterladen oder löschen.

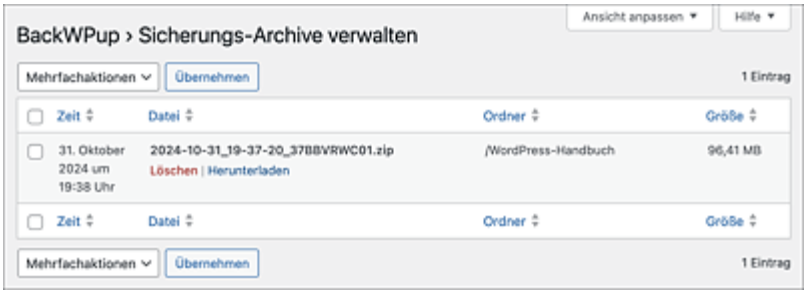


Abbildung 16.18 Unter »BackWPup« • »Backups« können Sie sich die gemachten Backups anschauen und Backups herunterladen.

Seit Kurzem gibt es auch in der kostenlosen Version des Plugins die Möglichkeit, ein Backup über das Backend wiederherzustellen. Dafür können Sie eine Backup-Datei über den Menüpunkt BackWPup • Wiederherstellen hochladen.

17 Migrationen, Synchronisationen und Deployments

Früher oder später werden Sie an einen Punkt kommen, an dem Sie den Stand einer WordPress-Installation woanders hin übertragen möchten – sei es, um Ihren lokalen Stand einer Website online zu stellen, eine Kopie einer Installation zu erstellen oder den Hoster zu wechseln. Dieses Kapitel zeigt Ihnen unterschiedliche Methoden, wie Sie dabei vorgehen können.

Bei einer einfachen persönlichen Website wird es dazu vermutlich seltener Grund geben, aber schon wenn Sie eine größere Änderung nicht auf der Live-Website ausprobieren möchten, müssen Sie eine Kopie Ihrer Installation erstellen, um damit testen zu können. Dabei möchten Sie die Testinstallation bei den Tests mit der Live-Website synchronisiert haben, also auf dem gleichen Stand. Wenn eine Änderung für die Live-Website bereit ist, steht das *Deployment* an (die Änderung auf ein anderes System übertragen), und manchmal möchten oder müssen Sie vielleicht eine *Migration* starten, um etwa den Hoster zu wechseln oder von einem System auf WordPress oder andersrum zu wechseln.

17.1 Testen und entwickeln mit Entwicklungs-, Staging- und Live-Umgebungen

Wenn Ihre Website ein wenig komplexer ist oder häufiger größere Änderungen anstehen, sollten Sie neben der Live-Umgebung noch zwei weitere Instanzen bereithalten:

- Die **Entwicklungsumgebung** ist Ihre lokale Version der Website, mit der Sie unkompliziert an Änderungen arbeiten und diese ausprobieren können, ohne die Anpassungen auf einen anderen Server übertragen zu müssen (wie Sie eine lokale Installation aufsetzen können, erfahren Sie in den Kapitel 2 und Kapitel 3, ein paar Tipps zur Anpassung der Installation für Entwicklungszwecke gibt es in Kapitel 22).
- Die **Staging-Umgebung** ist eine Umgebung auf einem Server, auf die Anpassungen deployt werden, bevor sie auf das Live-System kommen. Hier können sich alle Beteiligten die Änderungen anschauen, sie testen und entscheiden, ob etwas bereit für den Live-Gang ist oder ob noch eine Entwicklungsrunde notwendig ist. Optimalerweise sollte diese Umgebung technisch der Live-Umgebung entsprechen, um zu vermeiden, dass etwa eine andere PHP-Version dafür sorgt, dass Dinge anders funktionieren als dann nach dem Deployment auf Live. Oft befindet sich die Staging-Umgebung auf demselben Server wie die Live-Umgebung und läuft unter einer Subdomain, beispielsweise *staging.example.com*.
- Die **Live-Umgebung** ist die Umgebung, auf der die öffentliche Version Ihrer Website läuft.

17.1.1 Einrichten einer Staging-Umgebung

Um eine Staging-Umgebung einzurichten, gibt es verschiedene Möglichkeiten. Sie können ein Plugin wie »WP Staging« nutzen, das in der Pro-Version auch die Möglichkeit bietet, eine Kopie einer Website auf einer Subdomain zu erstellen (in der kostenlosen Version, die am Ende dieses Kapitels vorgestellt wird, ist nur eine Kopie als Unterverzeichnis-Installation möglich, was nicht optimal ist, wenn Ihre Live-Website nicht auch in einem Unterverzeichnis liegt, um die Änderungen des Setups möglichst gering zu halten). Oder Sie gehen den manuellen Weg, der in Abschnitt 17.2.1 beschrieben ist.

Unabhängig davon, wie Sie die Umgebung einrichten, sollten Sie die Installation mit einem Passwortschutz versehen, damit nur berechtigte Personen darauf zugreifen können. Dafür können Sie beispielsweise das Plugin »Password Protected« verwenden (*https://de.wordpress.org/plugins/password-protected/*) oder einen Basic-Auth-Schutz über die *.htaccess* einrichten (siehe Kasten).

Info: Passwortschutz über die .htaccess-Datei einrichten

Setzen Sie als Webserver Apache ein (wenn Sie ein Shared-Hosting-Paket bei einem größeren Hoster verwenden, dürfte das meist der Fall sein), können Sie relativ einfach einen Passwortschutz über die *.htaccess*-Datei einrichten, die auch dazu benötigt wird, dass die schöneren Permalinks funktionieren, die nicht nur die ID des Beitrags oder der Seite enthalten.

Dafür müssen Sie zunächst eine Datei mit den Zugangsdaten der erlaubten Benutzer*innen erstellen (der Name der Datei ist egal, wir nennen sie *.htpasswd*). *Platzieren Sie diese Datei außerhalb*

des Verzeichnisses, auf das Ihre Domain zeigt, sodass sie nicht direkt über das Web aufgerufen werden kann.

Um die Datei zu erstellen, können Sie den `htpasswd`-Befehl nutzen, der mit Apache installiert wird. Dafür müssten Sie sich per SSH auf den Server einloggen und den folgenden Befehl ausführen, um die Datei zu erstellen und einen Eintrag für `username` anzulegen:

```
htpasswd -c /pfad/zur/.htpasswd username
```

Nach Eingabe des Befehls werden Sie nach dem Passwort für `username` gefragt. *Wenn Sie einen weiteren Eintrag in die Datei einfügen möchten, lassen Sie das* `-c` *weg, sonst wird die Datei mit einer neuen Datei überschrieben.*

Wenn Sie keinen Zugriff per SSH haben oder lieber nicht in der Kommandozeile arbeiten möchten, können Sie sich die Zeichenkette für die Datei mit einem Online-Generator wie *https://www.entwicklertools.de/tools/htaccess/htpasswd-generator/* erstellen lassen. Kopieren Sie die Zeile, die aus dem Generator kommt (am Anfang steht immer der Name der Benutzer*in, gefolgt von einem Doppelpunkt und dem Hash des Passworts), und fügen Sie ihn in eine leere Datei ein, die Sie als *.htpasswd* auf Ihren Server hochladen.

Jetzt müssen Sie noch die *.htaccess*-Datei anpassen, die sich im Root-Verzeichnis Ihrer WordPress-Installation befindet. Fügen Sie am Anfang die folgenden Zeilen ein:

```
AuthType Basic
AuthName "Staging"
AuthUserFile /pfad/zur/.htpasswd
Require valid-user
```

- Über `AuthType` setzen Sie die Authentifizierungsmethode auf `Basic`.

- `AuthName` ist eine Zeichenkette, die in dem Formular für User und Passwort angezeigt wird.
- `AuthUserFile` ist ein absoluter Pfad zu der Datei mit den Benutzer*innen und dazugehörigen Passwort-Hashes, die Sie im vorherigen Schritt angelegt haben. Den absoluten Pfad können Sie über die Kommandozeile mit dem Befehl `pwd` herausfinden, der den Pfad zu dem Verzeichnis ausgibt, in dem Sie sich gerade befinden. Wenn Sie keinen Zugriff per SSH auf den Webserver haben oder sich nicht damit auskennen, müsste in der Administrationsoberfläche Ihres Webhosters irgendwo festgehalten sein, wie der Pfad zu Ihrem Web-Verzeichnis lautet.
- `Require` ermöglicht, nur bestimmte Benutzer*innen aus der Datei zuzulassen. Wir möchten alle erlauben und setzen den Wert deshalb auf `valid-user`.

Wenn Sie jetzt die Änderungen an der `.htaccess` speichern, sollten Sie beim Aufrufen der WordPress-Installation ein Formular sehen, um User und Passwort einzugeben.

17.1.2 Nutzung von Git für eigenen Code und Deployment über CI

Wenn Sie für die Website ein eigenes Theme oder eigene Plugins entwickeln, sollten Sie mit *Git* arbeiten. Das ist eine Software zur Versionsverwaltung, mit der Sie quasi Zwischenstände Ihres Codes speichern können und so Änderungen nachverfolgen können. Mit GitHub (*github.com*) und GitLab (*gitlab.com*) gibt es zwei große Plattformen, die Ihnen die Nutzung erheblich vereinfachen können.

Dieses Buch kann Ihnen keinen Einstieg in Git geben, aber ein paar Gründe, warum Sie sich damit beschäftigen sollten:

- Wenn Sie feststellen, dass Änderungen von vor ein paar Tagen, Wochen oder Monaten ein komisches Problem verursachen und deshalb rückgängig gemacht oder zumindest angepasst werden sollten, Sie aber nicht mehr genau wissen, was da alles angepasst wurde, können Sie sich die Historie Ihres »Repositorys« (jedes Projekt in Git ist ein Repository) anschauen und genau nachverfolgen, in welchen Dateien welche Änderungen vorgenommen wurden.
- Sie haben eine zentrale Stelle, an der Ihr Code liegt. Das ist nicht nur hilfreich, wenn Sie mit mehreren Entwickler*innen an einem Projekt arbeiten, sondern auch, wenn ein Projekt lange nicht angefasst wurde, Sie vielleicht zwischendurch Ihr lokales System gewechselt haben und dann nicht mehr sicher sind, ob Sie beim Systemwechsel alles übertragen haben oder doch lieber die Daten vom Live-Server herunterladen sollten, um auf Nummer sicher zu gehen.
- Die Arbeit mit einem Git-Repository als zentrale Verwaltungsstelle Ihres Codes kann auch zum Deployment Ihres Codes auf die Staging- und Live-Umgebung genutzt werden. Dafür gibt es beispielsweise bei GitHub die sogenannten »Actions« und bei GitLab »GitLab CI«. Beide ermöglichen Ihnen, vollautomatisch auf bestimmte Ereignisse im Repository zu reagieren, wie etwa den »Push« von neuem Code, und dann eine Aktion auszuführen, beispielsweise den Code auf den Staging-Server zu übertragen.

  Diesen Weg sollten Sie dem manuellen Deployment über SFTP vorziehen. So können Sie sicherstellen, dass es nicht zu versehentlichen Überschreibungen kommt oder Stände

auseinanderlaufen. Im Git-Repository ist immer der Stand, der auch auf dem Server liegt.

17.2 Eine Website migrieren

Wenn Sie eine Website migrieren möchten, beispielsweise um zu einem anderen Hoster zu wechseln, haben Sie generell zwei unterschiedliche Möglichkeiten: Sie führen die Migration von Hand durch, oder Sie nutzen ein Plugin dafür. Wir beginnen hier mit dem manuellen Weg, danach wird ein Plugin vorgestellt, das die Arbeit vereinfacht.

17.2.1 Manuell

Eine WordPress-Website besteht aus den Dateien auf dem Webserver und den Daten in der Datenbank. Diese Dinge müssen Sie also exportieren/herunterladen und beim neuen Hoster importieren/hochladen.

Dateien migrieren

Der einfachste Weg zur Migration ist, die Dateien der Installation per SFTP herunterzuladen und anschließend beim neuen Hoster wieder per SFTP hochzuladen. Dazu können Sie beispielsweise das kostenlose Programm FileZilla nutzen (*https://filezilla-project.org/*) und wie folgt vorgehen:

1. Sie benötigen SFTP/SSH-Zugangsdaten zu dem alten und dem neuen Server. Diese Zugangsdaten können Sie über das Administrations-Interface der entsprechenden Webhoster herausfinden/bearbeiten.
2. Haben Sie die Zugänge zur Hand, öffnen Sie FileZilla (siehe Abbildung 17.1). Der größte Teil der Programmoberfläche ist in

zwei Spalten aufgeteilt: Links sehen Sie Ihre lokalen Verzeichnisse und Dateien, rechts sehen Sie die Dateien des Webservers, sobald Sie sich verbunden haben.

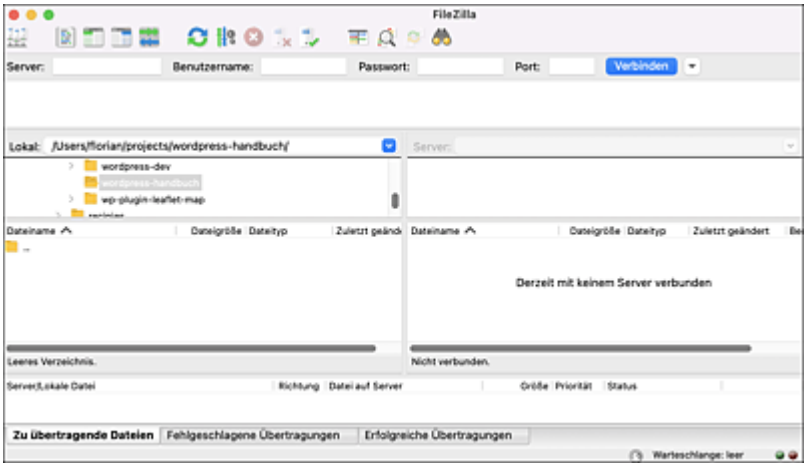


Abbildung 17.1 Auf der linken Seite sehen Sie in FileZilla Ihre lokalen Dateien und Ordner, rechts wird erst etwas angezeigt, wenn Sie sich mit einem Server verbunden haben.

Direkt oben unter der Icon-Werkzeugleiste können Sie die Verbindungsdaten zu Ihrem Server eingeben. Wir gehen für das Beispiel davon aus, dass Sie die Verbindung zu dem alten Server nicht mehr oft brauchen und diese deshalb nicht zur Wiederverwendung speichern möchten. Geben Sie in die Felder die entsprechenden Informationen von Ihrem Webhoster ein, und klicken Sie auf VERBINDEN.

3. Jetzt sollten Sie in der rechten Spalte Verzeichnisse und Dateien vom Webserver sehen (siehe Abbildung 17.2). Wechseln Sie links in Ihren lokalen Verzeichnissen zu einem leeren Verzeichnis, in das Sie die Dateien der Website herunterladen möchten, und gehen Sie auf dem Webserver in das Verzeichnis, in dem die WordPress-Dateien liegen.

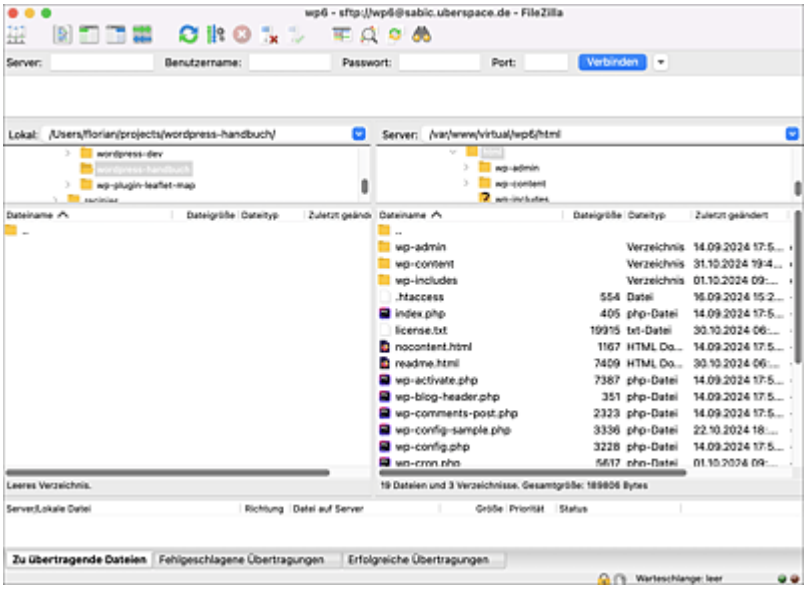


Abbildung 17.2 Nachdem Sie sich mit dem Server verbunden haben, wechseln Sie lokal zu einem leeren Verzeichnis und auf dem Server zum WordPress-Verzeichnis.

4. Markieren Sie jetzt auf dem Webserver alle Verzeichnisse und Dateien im Root-Verzeichnis von WordPress (klicken Sie das erste Element an und danach mit gedrückter ⇧-Taste das letzte, um alle Elemente auszuwählen). Anschließend klicken Sie mit der rechten Maustaste auf die markierten Einträge und wählen im erscheinenden Dialog den Eintrag HERUNTERLADEN (siehe Abbildung 17.3).
5. Warten Sie, bis der Download abgeschlossen ist. Je nachdem, wie schnell Ihre Internetverbindung ist und wie viele Medien die WordPress-Installation umfasst, kann das etwas dauern.
6. Verbinden Sie sich mit dem Server, auf den Sie die Daten hochladen möchten. Wenn Sie sich voraussichtlich häufiger mit diesem Server verbinden müssen, können Sie ihn in FileZilla als Server anlegen, indem Sie ganz links oben auf das Icon für den SERVERMANAGER klicken. Haben Sie den Manager geöffnet, können Sie einen neuen Server anlegen und dort die Verbindungsinformationen hinterlegen (siehe Abbildung 17.4).

Wenn die Verbindung nicht über einen SSH-Key, sondern über ein Passwort abläuft, ist es eine gute Idee, dieses Passwort nicht zu speichern, sondern bei der VERBINDUNGSART den Wert NACH PASSWORT FRAGEN zu wählen. So müssen Sie zwar das Passwort neu eingeben, wenn Sie sich wieder mit dem Server verbinden möchten, aber falls mal jemand anders Zugriff auf Ihr System haben sollte, kann die Person sich nicht direkt auch mit dem Webserver verbinden.

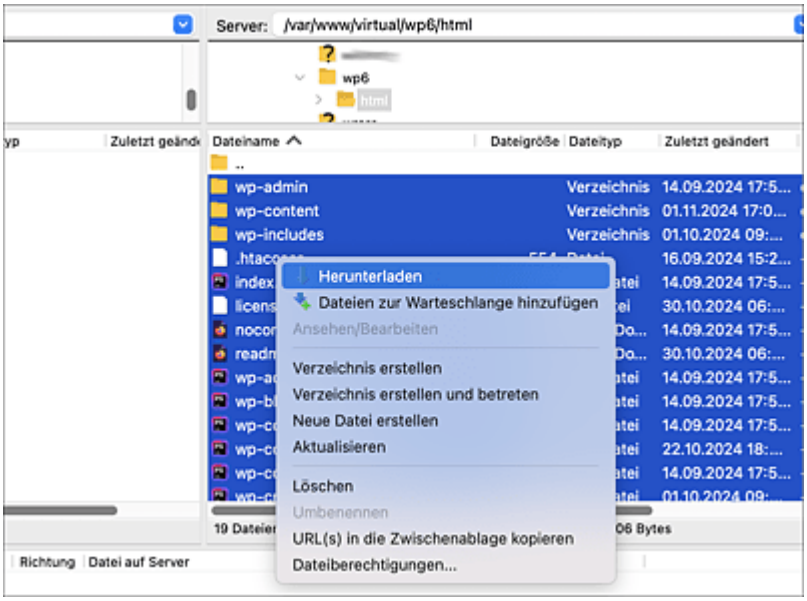


Abbildung 17.3 Über einen Rechtsklick können Sie die markierten Dateien vom Webserver auf Ihren lokalen Server herunterladen.

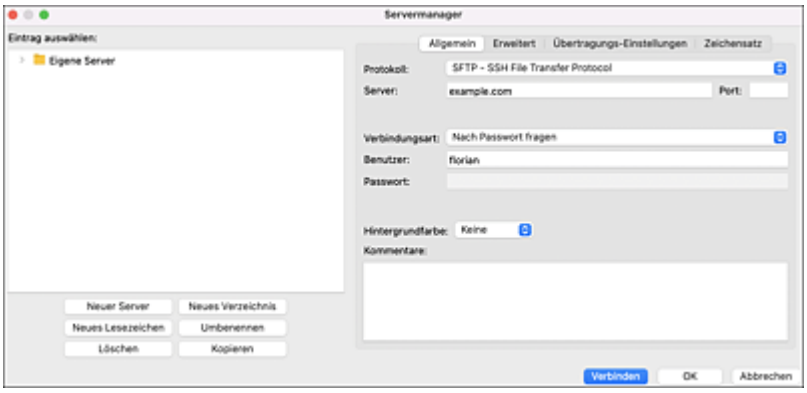


Abbildung 17.4 Im Servermanager von FileZilla können Sie Verbindungseinstellungen speichern.

7. Haben Sie sich mit dem neuen Server verbunden, navigieren Sie auf dem Webserver zu dem Verzeichnis, in dem die WordPress-Instanz liegen soll. Markieren Sie dann auf der linken Seite die Dateien der Installation, und laden Sie die Dateien hoch, wieder über einen Rechtsklick und diesmal den Punkt Hochladen.

Tipp: Schnellerer Umzug mit SSH-Zugang

Wenn Sie bei beiden Hostern einen SSH-Zugang und bereits ein bisschen Erfahrung damit haben, können Sie sich den Umweg über Ihr lokales System sparen und direkt die Dateien zwischen den beiden Servern übertragen. Dafür loggen Sie sich auf dem neuen Server per SSH ein und wechseln in das Verzeichnis, in das Sie die Dateien vom anderen Server übertragen möchten. Nun können Sie mit dem `rsync`-Befehl die Dateien vom anderen Server in dieses Verzeichnis synchronisieren:

```
rsync -av user@example.com:/pfad/zur/alten/installation/. ./
```

`-av` sind zwei Flags für den `rsync`-Befehl. `a` ist dabei ein Wrapper für mehrere andere Optionen und sorgt unter anderem dafür, dass auch Unterverzeichnisse berücksichtigt und alle Berechtigungen und Änderungsdaten vom alten System beibehalten werden. Mit `v` werden beim Synchronisieren die Dateien angezeigt, die übertragen werden.

Statt `user` müssen Sie den SSH-Benutzer des alten Servers einsetzen und *example. com* mit dem Servernamen oder der IP des alten Servers ersetzen (diese Infos bekommen Sie von Ihrem alten Webhoster). */pfad/zur/alten/installation/* ist der absolute Pfad auf dem Server zu dem Verzeichnis, in dem Ihre WordPress-Installation liegt. Mit *./* geben Sie an, dass die Dateien in das

Verzeichnis übertragen werden sollen, in dem Sie sich gerade befinden.

Wenn Sie den Befehl ausführen, sollten Sie im Terminal eine lange Liste an Dateien sehen, die vom alten auf den neuen Server übertragen werden (siehe Abbildung 17.5).

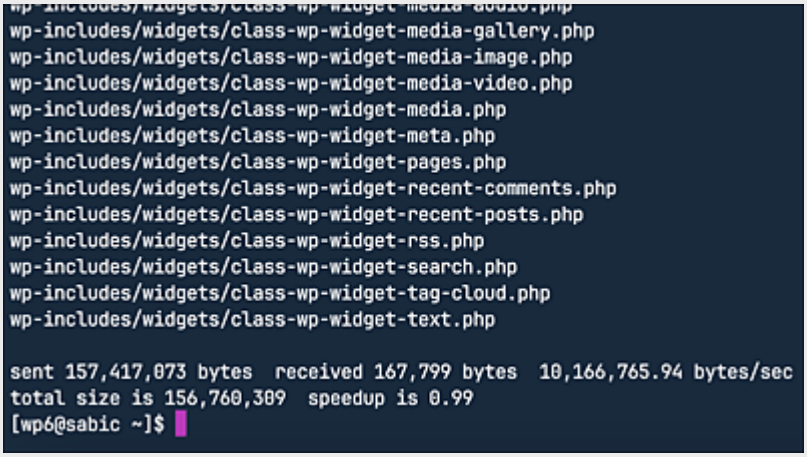


Abbildung 17.5 Mit »rsync« können Sie Dateien direkt zwischen zwei Servern übertragen, wenn Sie zu beiden einen SSH-Zugang haben.

Datenbank migrieren

Wie Sie die Datenbank exportieren und importieren können, hängt von Ihrem Hoster ab. Wenn Sie WP-CLI bei Ihrem Hoster installiert haben und sich damit auskennen, können Sie den Befehl `wp db export` benutzen, um die Datenbank zu exportieren (*https://developer.wordpress.org/cli/commands/db/export/*).

Bei vielen Hostern werden Sie ein Tool wie phpMyAdmin zur Verfügung haben, um auf die Datenbank zuzugreifen. Starten Sie das entsprechende Tool bei dem Hoster, den Sie verlassen möchten, wählen Sie – falls noch nicht automatisch geschehen – die Datenbank mit den WordPress-Tabellen aus, und gehen Sie in die

EXPORTIEREN-Ansicht (siehe Abbildung 17.6). Die Voreinstellungen passen für den Export, bestätigen Sie mit OK.

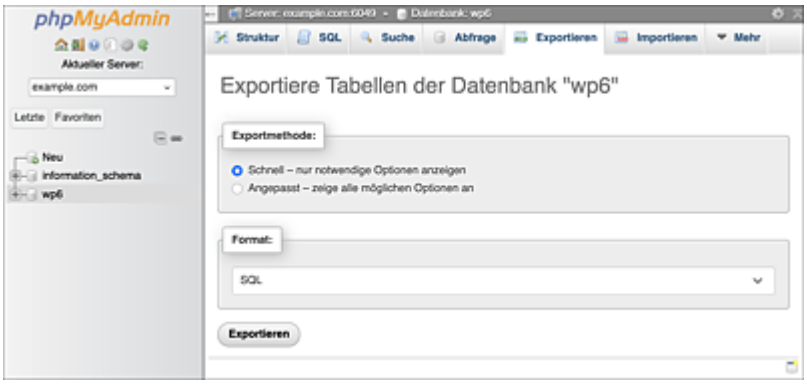


Abbildung 17.6 Die Standard-Einstellungen sollten für den Export meist passen, wie hier bei phpMyAdmin.

Wechseln Sie jetzt zu dem neuen System, und erstellen Sie, falls noch nicht geschehen, eine leere Datenbank für die WordPress-Installation. Öffnen Sie nun phpMyAdmin oder das Äquivalent, wählen Sie die neue Datenbank aus, und gehen Sie in den IMPORTIEREN-Bereich (siehe Abbildung 17.7). Für den Import wählen Sie die Datei, die bei dem Export entstanden ist, und starten Sie den Vorgang.

Jetzt müssen Sie in der *wp-config.php* auf dem neuen Server noch die Datenbankparameter anpassen. Laden Sie die Datei dafür per FTP herunter (wie Sie es im vorherigen Abschnitt beispielsweise mit FileZilla gesehen haben), und öffnen Sie sie. Die Werte, die Sie anpassen müssen, sind:

- `DB_NAME` – der Name der Datenbank
- `DB_USER` – der Nutzername für die Datenbank
- `DB_PASSWORD` – das Passwort für den Datenbanknutzer
- `DB_HOST` – der Datenbankserver

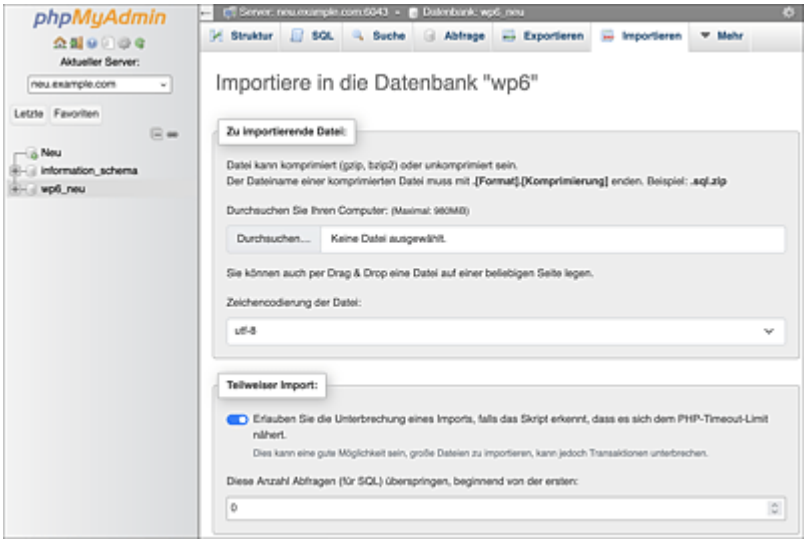


Abbildung 17.7 Im Import-Tab laden Sie die exportierte SQL-Datei hoch und importieren die Daten in eine leere Datenbank.

Diese Informationen sollten Sie bei Ihrem Hoster in dem Bereich finden, wo Sie auch die neue Datenbank erstellt haben.

Domain anpassen

Jetzt haben Sie auf dem neuen Server denselben Stand von Dateien und Datenbank-Inhalten wie auf dem alten Server und die Verbindung zwischen WordPress und der Datenbank vorbereitet. Was jetzt noch fehlt, ist die Domain. Wenn Sie dieselbe Domain weiter verwenden möchten, müssen Sie lediglich die Domain auf die neue Instanz zeigen lassen.

Wie Sie dabei vorgehen müssen, unterscheidet sich danach, wie die Domain-Situation ist:

1. Ist die Domain Teil des Webhosting-Pakets, von dem Sie umziehen möchten, müssen Sie die Domain auch umziehen. Wenn der neue Hoster Domain-Hosting anbietet, können Sie die Domain dorthin umziehen (dafür müssen Sie bei Ihrem alten Hoster die Domain für einen Umzug freigeben, um einen

sogenannten *Auth-Code* zu erhalten, den Sie bei der Registrierung der Domain beim neuen Hoster angeben, um nachzuweisen, dass Sie die Domain umziehen dürfen).

2. Sie haben die Domain unabhängig vom Hosting der Website und können sie einfach dort liegen lassen. In diesem Fall müssen Sie die DNS-Einstellungen anpassen. Welche genau und auf welche Werte, erfahren Sie bei Ihrem neuen Hoster.

Wenn Sie bei der Migration die Domain wechseln, müssen Sie die Domain auch in Ihrer WordPress-Installation anpassen. Öffnen Sie dafür noch mal die *wp-config.php*, und schauen Sie, ob bereits Werte für `WP_SITEURL` und `WP_HOME` vergeben sind. Falls ja, passen Sie diese auf die neue Domain an. Falls nicht, fügen Sie die Zeilen neu ein:

```
define( 'WP_SITEURL', 'https://example.com' );
define( 'WP_HOME', 'https://example.com' );
```

Damit sollte die Installation unter der neuen Domain erreichbar sein, wenn sie auf den neuen Server zeigt.

Jetzt bleibt als letzter Schritt noch, alle Vorkommen der alten Domain in der Datenbank durch die neue zu ersetzen. Loggen Sie sich dazu in der neuen Instanz ein, und installieren Sie das Plugin »Better Search Replace«. Nach Aktivierung des Plugins gehen Sie zu den Einstellungen, die Sie über das linke Admin-Menü unter WERKZEUGE • BETTER SEARCH REPLACE erreichen (siehe Abbildung 17.8).

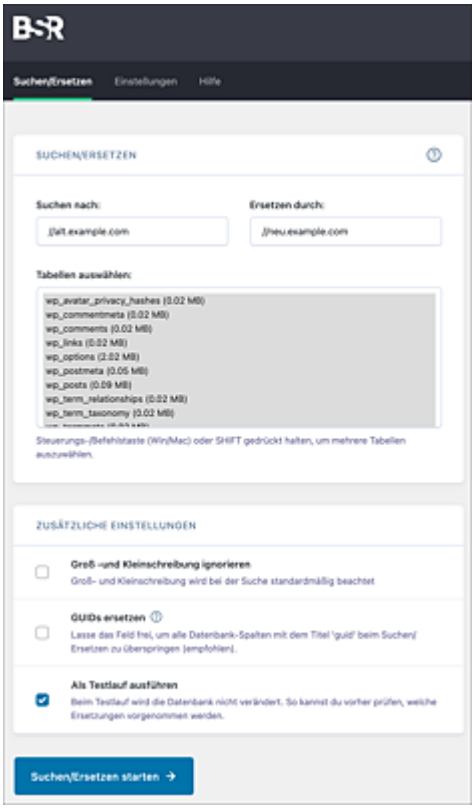


Abbildung 17.8 Mit »Better Search Replace« können Sie Werte in der Datenbank suchen und durch andere ersetzen, in unserem Fall die alte Domain durch die neue.

Geben Sie in das Feld SUCHEN NACH die alte Domain ein, am besten ohne Protokoll, also nach dem Muster *//alt.example.com*, und in das Feld ERSETZEN DURCH die neue Domain: *//neu.example.com*.

Plugin	Better Search Replace
Verbreitung	1.000.000+
Download	*https://de.wordpress.org/plugins/better-search-replace/*
Zweck	erlaubt das einfache Suchen und Ersetzen von Werten in der Datenbank (Sie sollten allerdings wissen, was Sie tun, bevor Sie es nutzen.)
Komplexität	+

Wählen Sie anschließend alle Tabellen in dem Auswahlbereich darunter aus, und klicken Sie unten auf Suchen/Ersetzen starten (standardmäßig wird es als Testlauf ausgeführt). Wenn Dinge zum Ersetzen gefunden werden, führen Sie den Prozess ohne Testlauf-Option aus.

Damit ist der manuelle Umzug abgeschlossen.

17.2.2 Mit dem Plugin »Duplicator«

Um WordPress-Installationen mithilfe eines Plugins umzuziehen, gibt es viele verschiedene Möglichkeiten. Duplicator ist eine davon, die auch in der kostenlosen Version die einfache Migration einer Installation ermöglicht.

Plugin	Duplicator
Verbreitung	1.000.000+
Download	*https://de.wordpress.org/plugins/duplicator/*
Zweck	Plugin, das unter anderem bei der Migration einer Website helfen kann
Komplexität	++

Nachdem Sie das Plugin installiert haben, finden Sie die Einstellungen links im Admin-Menü als eigenen Duplicator-Menüpunkt. Unter Duplicator • Backups können Sie ein Paket von Ihrer Installation erstellen, um es bei Ihrem neuen Hoster wieder zu installieren. Klicken Sie dafür oben rechts auf Neues erstellen. Daraufhin öffnet sich eine neue Ansicht, in der Sie Einstellungen zu dem Paket machen können (siehe Abbildung 17.9).

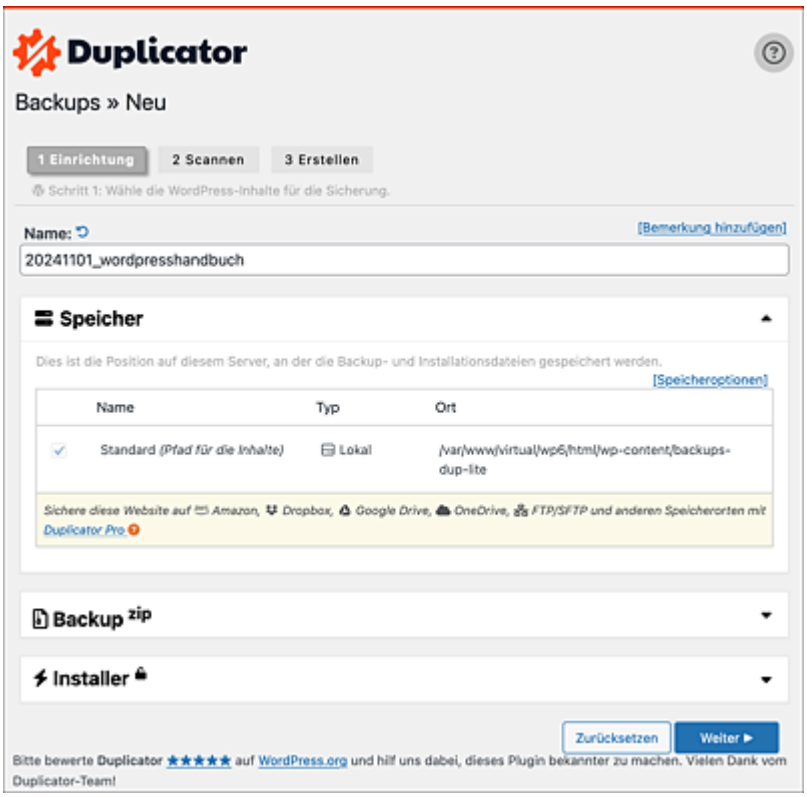


Abbildung 17.9 Duplicator gibt Ihnen die Möglichkeit, Einstellungen für die Erstellung eines Backups zu ändern, etwa Dateien oder Ordner auszuschließen. Hier können Sie bei den Standard-Einstellungen bleiben.

Die Standard-Einstellungen können Sie hier aber unangetastet lassen, klicken Sie also auf Weiter, um zum Scannen-Schritt zu kommen. Hier prüft Duplicator, ob es Probleme gibt, die die Erstellung des Backups verhindern könnten (siehe Abbildung 17.10). Wenn alles Okay ist, klicken Sie auf Erstellen und warten ab, bis das Archiv erstellt wurde.

In der Bestätigungsansicht bekommen Sie jetzt die Möglichkeit, Installer und Archiv herunterzuladen. Der Installer ist wichtig, wenn Sie wie in unserem Fall vorhaben, das Archiv an anderer Stelle als eigene Instanz wieder zu nutzen (siehe Abbildung 17.11).

Wenn Sie Archiv und Installer heruntergeladen haben, können Sie beide per SFTP beim neuen Webhoster in das Verzeichnis hochladen, in dem Ihre WordPress-Installation liegen soll (wie Sie

Dateien per SFTP mit FileZilla übertragen können, wurde im vorherigen Abschnitt erklärt).

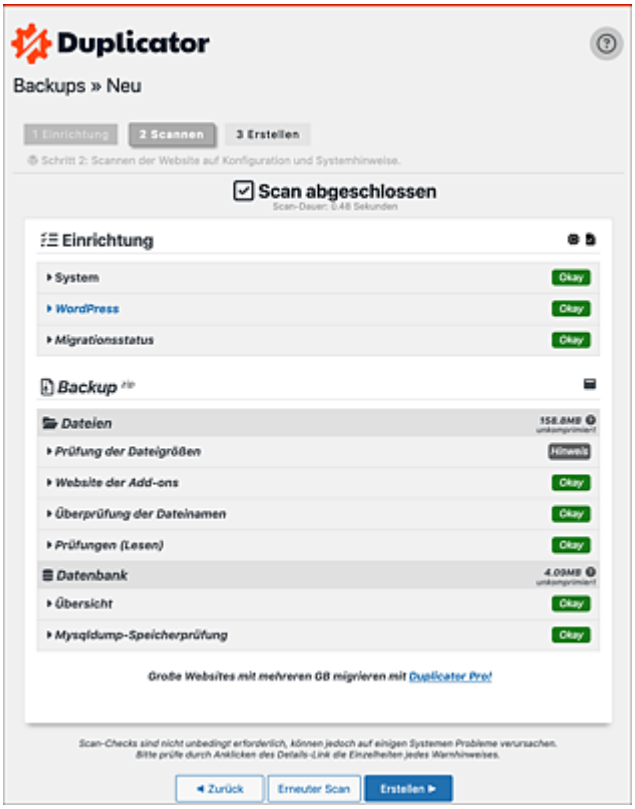


Abbildung 17.10 Vor der Erstellung des Backups prüft Duplicator, ob es irgendwelche Probleme gibt, die einer problemlosen Erstellung im Weg stehen könnten.

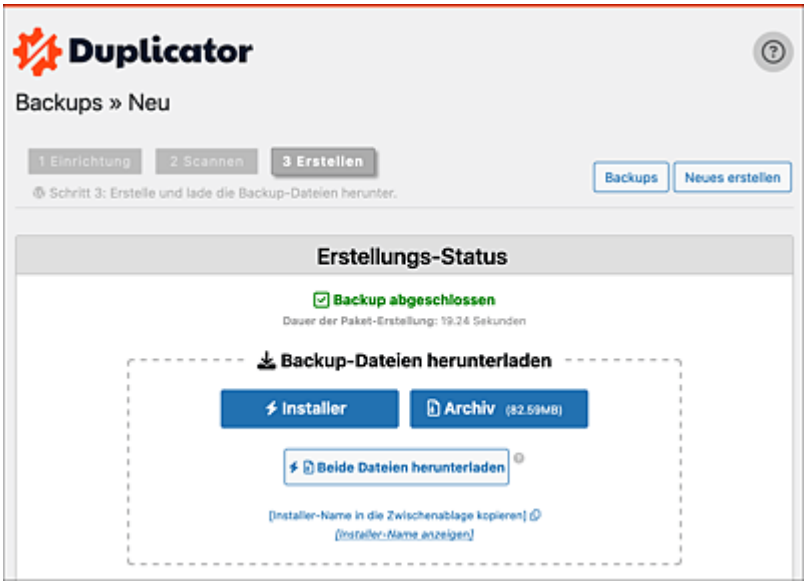


Abbildung 17.11 Laden Sie sich sowohl den »Installer« als auch das »Archiv« herunter.

Jetzt müssen Sie die *installer.php* im Browser aufrufen, also beispielsweise *example. com/installer.php* (siehe Abbildung 17.12). Im ersten Schritt des Installers geben Sie die Verbindungsdaten der Datenbank an (wenn die Datenbank nicht leer ist, werden bestehende Tabellen gelöscht) – die neue URL der Site übernimmt Duplicator automatisch von der Domain, von der Sie den Installer aufrufen.

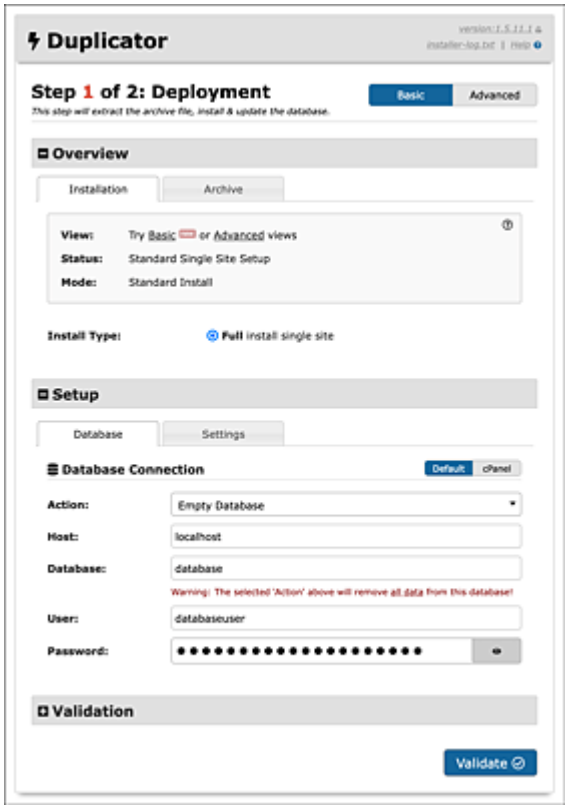


Abbildung 17.12 Im Installer von Duplicator geben Sie die neuen Datenbank-Daten an.

Danach wird bereits die Installation wiederhergestellt, und Sie können im nächsten Schritt über den Klick auf den Button Admin Login ins Backend wechseln (siehe Abbildung 17.13). Lassen Sie die Checkbox neben dem Button angehakt, damit der Installer entfernt wird.

Und damit hätten Sie auch schon eine Website mit Duplicator erfolgreich umgezogen.

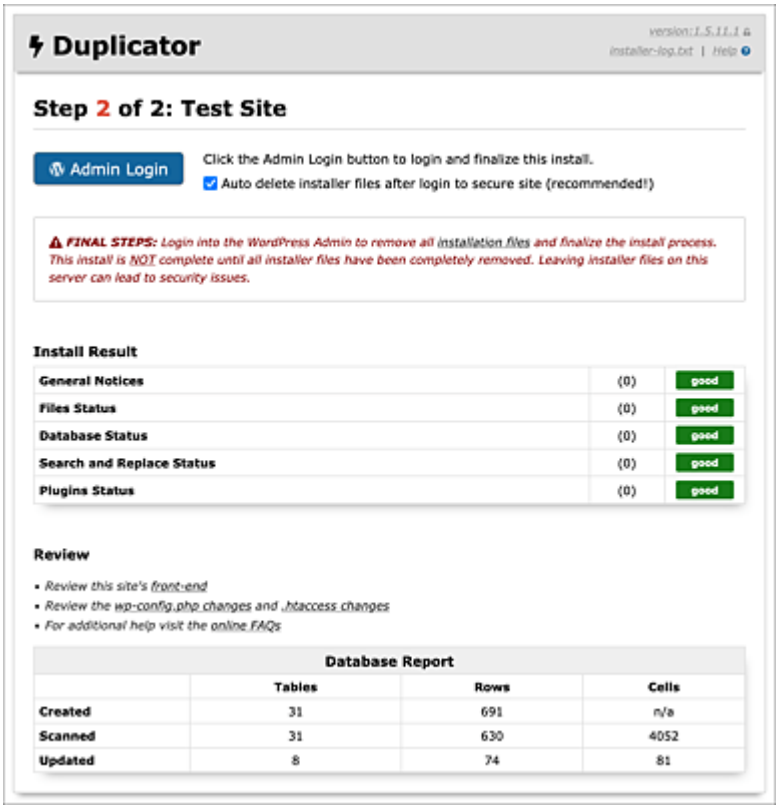


Abbildung 17.13 Und das war es auch schon: Die Website ist wiederhergestellt, und Sie können ins Backend wechseln.

Abschließend seien hier noch BackWPup und WP Migrate genannt, die ähnliche Funktionen bieten wie Duplicator, allerdings teilweise erst in der Bezahlversion.

17.2.3 Schnell eine Kopie einer Instanz in einem Unterverzeichnis mit »WP Staging«

Plugin	WP Staging
Verbreitung	100.000+
Download	*https://de.wordpress.org/plugins/wp-staging/*
Zweck	Plugin zum schnellen Erstellen einer Staging-Website

Komplexität +

Auch »WP Staging« ist ein Plugin, mit dem Sie WordPress-Sites migrieren können. Besonders praktisch ist es aber für den Fall, wenn Sie keine dedizierte Staging-Umgebung für Ihre Website haben, denn Sie können damit sehr einfach eine Kopie Ihrer Website in einem Unterverzeichnis erstellen. Und das geht so:

1. Installieren Sie das Plugin »WP Staging« über das WordPress-Backend, und gehen Sie im linken Admin-Menü zum Punkt WP Staging • Staging-Websites (siehe Abbildung 17.14).

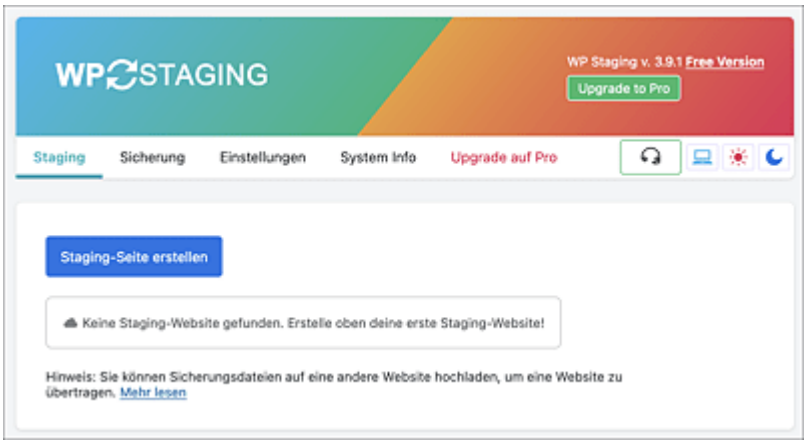


Abbildung 17.14 Mit dem Plugin »WP Staging« können Sie einfach eine Kopie Ihrer Website in einem Unterverzeichnis auf demselben Server erstellen.

2. Klicken Sie auf Staging-Seite erstellen, und geben Sie einen Namen ein. Optional können Sie vor dem Erstellen der Kopie noch anpassen, welche Dateien und Datenbank-Tabellen mitkopiert werden, aber bei einer normalen WordPress-Installation passen die Voreinstellungen des Plugins, klicken Sie also auf Klonen starten (siehe Abbildung 17.15).
3. Nachdem Sie auf Klonen starten geklickt haben, bekommen Sie in einem Modal den aktuellen Stand des Vorgangs angezeigt und nach Vollendung des Prozesses einen Link zu der Kopie angeboten (siehe Abbildung 17.16).

4. Wenn Sie den Link öffnen, müssen Sie Ihre Zugangsdaten eingeben, um die Kopie ansehen zu können. Haben Sie das gemacht, landen Sie im Frontend der Kopie. Die Admin-Bar ist zur Kenntlichmachung der Kopie orange eingefärbt.

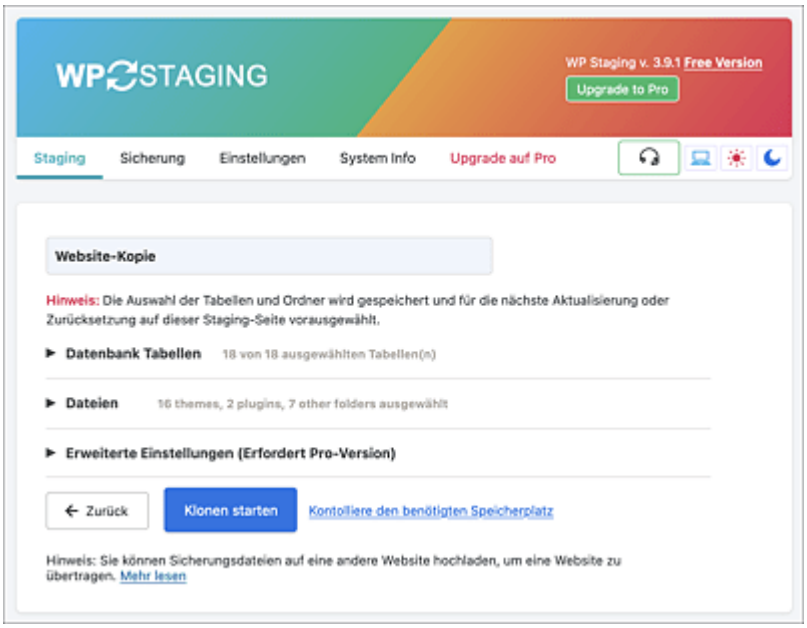


Abbildung 17.15 Sie können einen Namen für die Kopie angeben und danach direkt auf »Klonen starten« klicken, um die Kopie zu starten.

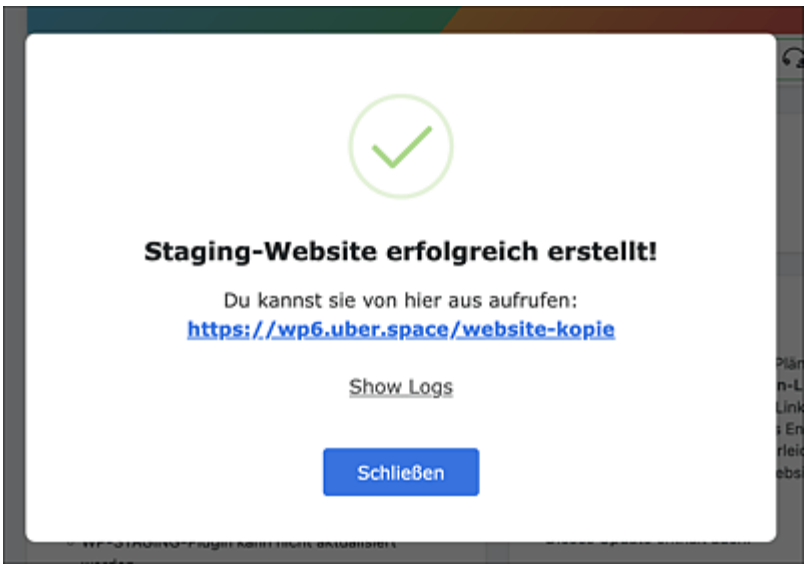


Abbildung 17.16 Nach erfolgreichem Erstellen der Kopie bekommen Sie in einem Modal den Link zur Kopie angezeigt.

In den WP-Staging-Einstellungen können Sie später die Kopie der Website über den Aktionen-Button auf den aktuellen Stand der Live-Website Aktualisieren, Zurücksetzen oder Löschen, wenn Sie den Klon nicht mehr benötigen (siehe Abbildung 17.17).

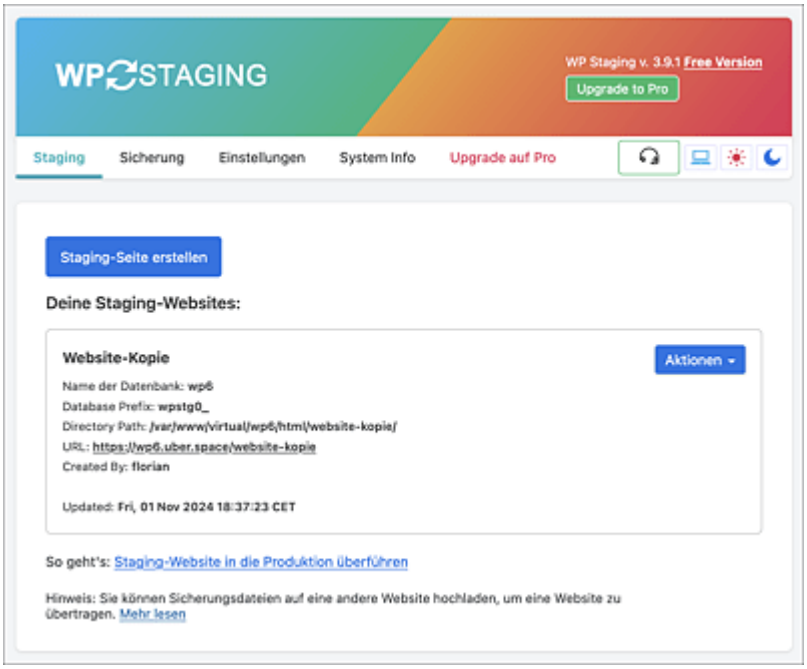


Abbildung 17.17 Sie können die Website-Kopie über den »Aktionen«-Toggle löschen oder auf den aktuellen Stand der Live-Site aktualisieren.

17.2.4 Von einem anderen System zu WordPress

Wenn Sie Ihre Website bisher mit einem anderen System als WordPress betreiben, aber gerne wechseln möchten, gibt es keine allgemeingültige Lösung, wie Sie am besten vorgehen können. Sie können über die Suchmaschine Ihres Vertrauens schauen, ob es vielleicht Importer für WordPress gibt (vorausgesetzt natürlich, Sie können die Daten aus Ihrem aktuellen System exportieren). Für Joomla! gibt es beispielsweise ein solches Plugin.

Wenn Sie eins der unter Werkzeuge • Daten importieren aufgelisteten Systeme einsetzen (Blogger, LiveJournal, Movable Type/TypePad

oder Tumblr), können Sie über diese Seite entsprechende Importer installieren.

17.2.5 Von WordPress zu einem anderen System

Wenn Sie WordPress den Rücken kehren möchten, können Sie Ihre Daten entweder als SQL exportieren (wie das geht, haben Sie weiter vorne im Kapitel gesehen) oder über WERKZEUGE • EXPORT alle oder nur bestimmte Inhaltsarten als XML-Dateien exportieren (siehe Abbildung 17.18).

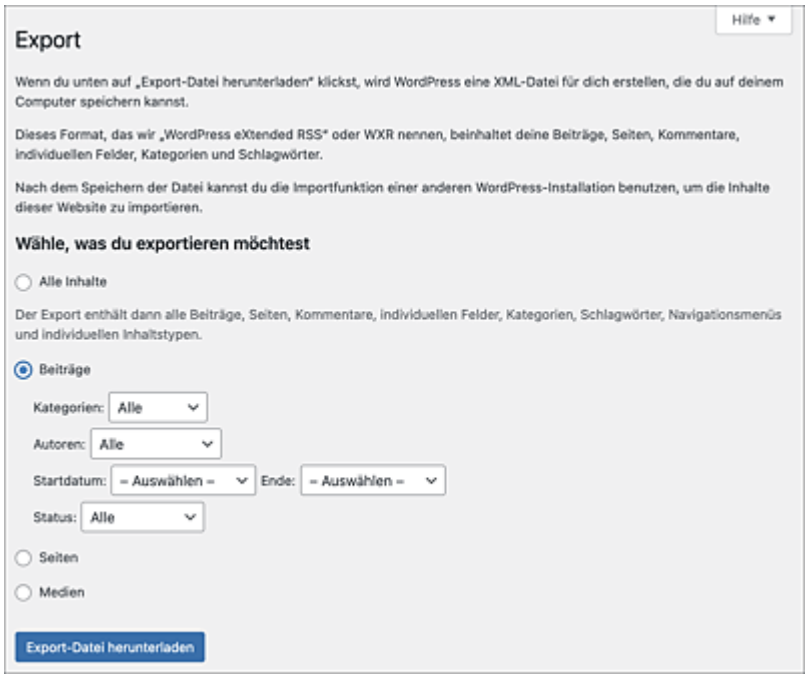


Abbildung 17.18 WordPress bietet Ihnen die Möglichkeit, Ihre Daten oder nur Teile davon als XML zu exportieren.

Dabei werden allerdings nicht die Uploads aus der Mediathek mit exportiert, Sie müssten diese über FTP herunterladen oder über ein Backup-Plugin ein Backup der Dateien erstellen und sie so herunterladen.

18 WordPress absichern

Obwohl es mit Blick auf die Sicherheit manchmal etwas verschrien ist, ist WordPress an sich ziemlich sicher. Wenn Sicherheitslücken direkt am CMS bekannt werden, gibt es in der Regel in kürzester Zeit ein Sicherheits-Update. Durch eine aktive Community, viele Augen auf dem Code und solide Update-Mechanismen sind Sie mit WordPress auf der relativ sicheren Seite. Plugins, Themes und das Verhalten der Menschen, die eigene User-Konten auf Ihrer Website haben, stellen die größte Gefahr dar. Doch keine Sorge, hier können Sie mit einigen Handgriffen vorsorgen.

Bei der Verwaltung Ihrer Website ist die Sicherheit ein Thema, das Sie auf keinen Fall ignorieren sollten – seien es Hacker-Angriffe, Datenlecks, unsichere Passwörter oder Sicherheitslücken in Plugins. In diesem Kapitel erfahren Sie mehr darüber, wie Sie sich und Ihre WordPress-Website vor den meisten Sicherheitsproblemen schützen können. Auch wenn Sie Ihre Website natürlich nie absolut sicher bekommen können, solange Sie nicht den Stecker ziehen, können Sie mit überschaubarem Aufwand doch einen effektiven Schutz vor den meisten Bedrohungen erreichen.

Bevor wir uns den tatsächlichen Sicherheitsthemen widmen, kurz zwei wichtige Grundlagen: Stellen Sie sicher, dass sie stets ein aktuelles Backup Ihrer Website haben. Je nachdem, wie häufig Sie neue Inhalte veröffentlichen oder bestehende verändern, sprechen wir hier nicht zwingend von einem täglichen Backup. Bei vielen

Websites ist auch eine wochenaktuelle Sicherheitskopie absolut ausreichend. Wichtig ist vor allem, dass es überhaupt ein Backup gibt und dass dieses nicht (nur) auf demselben Server gespeichert wird, auf dem auch Ihre Website läuft. Außerdem schadet es nicht, gelegentlich zu prüfen, ob ein Backup für den Ernstfall auch wirklich eine lauffähige Kopie Ihrer Website enthält, und das Ganze einmal als Trockenübung wiederherzustellen. Und Sie sollten einen Rhythmus finden, in dem Sie sich um Aktualisierungen für den WordPress-Core, Plugins und Themes kümmern.

18.1 Login-Sicherheit

Neben Sicherheitslücken in WordPress oder den Plugins, die Sie einsetzen, und allgemeinen Verbesserungen der Sicherheit (siehe Abschnitt 18.3) gibt es mit dem Login, den mindestens Sie selbst als Administrator*in Ihrer Website benutzen, einen Bereich, dem wir uns zuerst widmen sollten. Beginnen wir also mit einem Thema, das sich auf Ihren WordPress-Zugang genauso anwenden lässt wie auf viele andere Zugangsdaten, die Sie regelmäßig online verwenden: Passwort-Sicherheit.

18.1.1 Passwort-Sicherheit

Was ist das Problem mit Passwörtern? Die meisten Benutzer*innen nutzen Passwörter, die zu kurz, zu einfach und zu oft im Einsatz sind. Wenn Sie sich ein Passwort einfach merken können, ist es vermutlich unsicher.

Eine minimale Länge ist ein wunderbarer erster Schritt. Außerdem wollen wir für Passwörter einfache Begriffe vermeiden und personenbezogene Daten wie den Namen eines Haustiers, der/des

Verlobten und offensichtliche Zeichenfolgen wie »Passwort123« vermeiden.



Abbildung 18.1 WordPress gibt Ihnen eine direkte Rückmeldung dazu, wie sicher ein Passwort ist, das Sie verwenden möchten.

Wenn Sie in WordPress ein neues Passwort für sich oder einen anderen Account vergeben, zeigt WordPress Ihnen direkt die Stärke des eingetippten Passworts an. Wenn Sie ein schwaches Passwort verwenden möchten, müssen Sie das zusätzlich bestätigen. Von sich aus ist WordPress nicht in der Lage, ein sicheres Passwort zu erzwingen. In Abschnitt 18.1.2 stellen wir Ihnen ein Plugin vor, das unter anderem dies für Sie tun kann.

Tipp: Passwortmanager

Sie haben vermutlich mehr als zwei oder drei Zugänge zu Websites und Diensten, für die Sie Passwörter benötigen. Gefährlich wäre es jetzt, nur ein einziges Passwort für alle Dienste nutzen, um sich das Leben leichter zu machen. Wenn bei einem Dienst Login-Daten nicht sicher gespeichert werden und mal abhandenkommen, wäre direkt die Sicherheit aller Ihrer anderen Accounts gefährdet, für die Sie das Passwort wiederverwenden. Die Lösung ist recht einfach und mindestens ebenso komfortabel: Nutzen Sie einen Passwortmanager (1Password, Bitwarden, KeePass, Passbolt). Diese Apps bieten Ihnen die Möglichkeit, Ihre Login-Daten abzuspeichern und über mehrere Geräte hinweg zu synchronisieren und zu nutzen. Dienste wie 1Password bieten Ihnen über Browser-Erweiterungen auch die Möglichkeit, Logins

direkt in Login-Formulare auf Webseiten einzufügen. Auch Passwortgeneratoren bringen die meisten dieser Programme direkt mit. So können Sie für jeden Dienst ein anderes Passwort verwenden, das möglichst lang ist, ohne sich diese individuellen Passwörter merken zu müssen.

18.1.2 Sichere Passwörter durchsetzen und Login schützen

Wenn Sie eine Website betreiben, auf die mehrere Benutzer*innen gemeinsam Zugriff haben, um Inhalte zu schreiben und zu modifizieren, wird es langsam schwer, die Verwendung sicherer Passwörter sicherzustellen. Von Haus aus bietet WordPress keine Möglichkeit, Passwortregeln – im Fachjargon *Password Policies* – durchzusetzen. Aber wie so oft können Sie sich auch hier mit Plugins behelfen.

Zusätzlich kann es helfen, Ihren WordPress-Login vor sogenannten *Brute-Force-Attacken* zu schützen. Bei einer solchen Attacke versucht ein Angreifer automatisiert, das Passwort eines User-Accounts für Ihre Website systematisch zu erraten. Abhängig davon, wie komplex ein Passwort ist, dauert es länger, es auf diesem Wege zu knacken. Ein einfaches, vierstelliges Passwort wäre zum Beispiel in kürzester Zeit geknackt. Ein komplexeres, 20-stelliges Passwort wird einen Angreifer hingegen eine Menge Zeit kosten.

Doch auch hier können wir noch eine weitere Sicherung einführen, in dem wir ein Plugin nutzen, das das Durchprobieren – eben eine sogenannte Brute-Force-Attacke – von Passwörtern nach mehreren fehlgeschlagenen Anmeldeversuchen verlangsamt. Sie kennen diesen Mechanismus vermutlich von Ihrem Smartphone – nach 5 fehlgeschlagenen Anmeldeversuchen dauert es einige Minuten, bis

Sie es erneut versuchen können. Das kann für Sie als die tatsächliche Person hinter einem Account etwas unkomfortabler sein, bremst Angreifer*innen aber recht zuverlässig aus.

Hinweis: Warum sollten ausgerechnet Sie angegriffen werden?

Wer sollte sich genug für Ihre Website interessieren, um sie anzugreifen? Das mag im ersten Moment nach einer ganz berechtigten Frage klingen. In der Praxis ist es aber egal, ob Sie eine kleine WordPress-Website für den Hannoveraner Kleingartenverein oder für das örtliche Repair Café betreiben. Egal ob Ihre Website täglich 3 oder 3000 Zugriffe verzeichnet, automatisierte Skripte und Bots durchstreifen Tag und Nacht das Netz und versuchen die verschiedensten Angriffe auf alle möglichen Websites. Mit dem Marktanteil von WordPress gehört es unter Angreifern quasi zum guten Ton, auch die wichtigsten WordPress-Schwachstellen zu testen und an den Türen zu kratzen. Die Wahrscheinlichkeit, dass so ein Automatismus erfolgreich ist, ist nicht hoch, aber trotzdem hoch genug, dass sich dieses Vorgehen für Angreifer lohnt.

Es ist daher ratsam, einige effektive Sicherheitsmaßnahmen in Stellung zu bringen, um zumindest diese Angriffe abzuwehren. Einen speziell auf Sie zugeschnittenen Angriff werden Sie deutlich schwerer abwehren können, das Risiko dafür ist aber auch deutlich geringer.

Für diese und weitere Funktionen bieten viele All-in-One-Sicherheits-Plugins Konfigurationsmöglichkeiten. Weil diese Plugins aber oft mit einer Menge zusätzlichem Ballast daherkommen, empfiehlt es sich in vielen Fällen, auf ein

schlankeres Plugin zu setzen, das sich auf einzelne Funktionen konzentriert. Mit dem recht neuen Plugin *Protect Login* haben Sie genau diese beiden Funktionen – das Durchsetzen von Password Policies und den Schutz vor *Brute-Force-Attacken*.

Plugin	Protect Login
Verbreitung	200+
Download	*https://de.wordpress.org/plugins/protect-login/*
Zweck	kleines, nützliches Plugin zur Verbesserung der Login-Sicherheit durch verschiedene Techniken
Komplexität	+

Einmal installiert und aktiviert greift der Brute-Force-Schutz umgehend. Unter Einstellungen • Protect Login finden Sie außerdem die Möglichkeit, Passwort-Regeln zu definieren und für alle neuen Passwörter auf Ihrer Website verpflichtend zu machen.

18.1.3 Zwei-Faktor-Authentifizierung

Wie viele andere Dienste und Websites gibt es auch bei WordPress die Möglichkeit, eine sogenannte *Zwei-Faktor-Authentifizierung* (Two-Factor Authentication, auch *2FA*) zu aktivieren. Auch hierfür ist aktuell leider noch ein Plugin erforderlich. Bei der Zwei-Faktor-Authentifizierung wird die Eingabe von Benutzername (E-Mail-Adresse) und Passwort durch einen zweiten Sicherheitsfaktor ergänzt. Damit soll sichergestellt werden, dass nur die berechtigte Person Zugriff auf ein Konto hat. Dafür wird Ihnen zum Beispiel an die im Account hinterlegte E-Mail-Adresse ein Code geschickt oder eine SMS an die hinterlegte Mobilnummer, wenn Benutzername und Passwort richtig eingegeben worden sind. Diesen Code müssen

Sie dann ebenfalls eingeben, bevor Sie sich einloggen können. So können sich Angreifer*innen nicht einfach in Ihren Account einloggen, selbst wenn sie die richtigen Zugangsdaten haben, solange sie keinen Zugriff auf den zweiten Faktor haben (im Beispiel Ihren E-Mail-Account oder Ihr Mobiltelefon). Passwortmanager wie 1Password oder Bitwarden bringen Unterstützung für Zwei-Faktor-Authentifizierung mit, sodass Sie sie zur Generierung der Codes nutzen können und sich die Codes nicht per Mail oder SMS schicken lassen müssen.

Plugin	Two Factor
Verbreitung	80.000+
Download	*https://de.wordpress.org/plugins/two-factor/*
Zweck	kleines Plugin zur Absicherung von WordPress-Benutzer-Accounts mit einem zweiten Faktor
Komplexität	++

Auch hier werden Sie in den verschiedensten großen Security-Plugins die Möglichkeit finden, Zwei-Faktor-Authentifizierung zu aktivieren. Seit vielen Jahren ist die klare Empfehlung hier jedoch das Plugin *Two Factor*. Es wird von einer Gruppe von Enthusiasten entwickelt, um eines Tages Teil des WordPress-Core zu werden, sollte also den höchsten Ansprüchen der WordPress-Community genügen. Außerdem ist Two Factor wieder eines dieser Plugins, die sich auf eine Aufgabe konzentrieren und diese sehr gut erledigen.

Einmal installiert und aktiviert, können Sie unter Benutzer • Profil ganz unten auf der Seite eine Zwei-Faktor-Methode für Ihren Account definieren. Als Admin können Sie diese zusätzliche Absicherung auch für andere Benutzer*innen aktivieren. Hier bietet sich die E-Mail-Methode an. Wenn Sie das erledigt haben, können

sich Ihre User nur noch anmelden, wenn sie zusätzlich zu Username und Passwort auch einen Code eingeben, den sie per E-Mail zugeschickt bekommen. Das ist nicht die sicherste Methode, die das Plugin Ihnen zur Verfügung stellt, kann aber von allen genutzt werden, auch wenn die betreffenden Personen weder einen Passwortmanager noch eine andere Authenticator-App verwenden.

18.2 Die Wichtigkeit von SSL/TLS

Wenn Sie in Online-Shops einkaufen und Ihre Kundendaten und die Kreditkartennummer eingeben, nutzen Sie eines der wichtigsten Protokolle im Internet: *Hypertext Transfer Protocol Secure* (HTTPS) bzw. den darin gekapselten Secure Sockets Layer (SSL), in neuesten Versionen auch in Transport Layer Security (TLS) umgetauft. Diese Protokolle verschlüsseln die zwischen Server und Browser übertragenen Daten, sodass weder Ihr Internetanbieter noch ein Angreifer, der den Datenstrom mitschneidet, der zu Ihnen fließt, die übertragenen Daten auch tatsächlich mitlesen kann. Die sichere Verbindung erkennen Sie an diversen Schlüssel- oder Schlosssymbolen in der Adresszeile Ihres Browsers, auf jeden Fall aber an der vorangestellten Protokollabkürzung `https://`. Seit einigen Jahren zeigt zum Beispiel der Chrome-Browser das Schloss bei gesicherten Verbindungen nicht mehr an, sondern warnt, wenn eine Verbindung nicht derart gesichert ist. Auch wenn es keine hundertprozentige Sicherheit gibt, ist SSL ein Must-have für eine sichere Website – und zwar nicht nur für Online-Shops und Websites, die mit kundenspezifischen oder persönlichen Daten arbeiten. Mittlerweile sollten alle Website-Betreiber*innen ihre Websites auf https umstellen bzw. umgestellt haben.

18.2.1 Kostenloses SSL aktivieren

Es ist erst wenige Jahre her, da war das Erzeugen und Einrichten eines SSL-Zertifikats eine wirklich kostspielige und gegebenenfalls sogar aufwendige Angelegenheit. Viele Hosting-Anbieter boten die Zertifikate gegen einen gehörigen Preis als Ergänzung zu einer Domain an. Alternativ gab (und gibt) es professionelle Anbieter, die

Zertifikate mit aufwendigerer Prüfung (zum Beispiel von Personen- oder Unternehmensdaten) anbieten. Diese ermöglichen neben der Verschlüsselung der Daten auch noch einen Nachweis, dass besagte Daten tatsächlich von der fraglichen Gegenstelle kommen und nicht von einem Hochstapler. Diese Zertifikate mit erweiterter Validierung gibt es noch heute. So lange Sie keine Bank-Website betreiben wollen, können Sie deren Existenz aber getrost ignorieren. Stattdessen kommen Sie in den Genuss des Authentifizierungsprojekts *Let's Encrypt*. Als Zusammenschluss von großen Namen der Branche wie Mozilla, Google, Cisco und Akamai wurde eine Initiative gestartet, um SSL-Zertifikate einfach und vor allem kostenfrei zur Verfügung zu stellen. Bald 10 Jahre nach dem Start von Let's Encrypt sollten Sie keinen Hosting-Tarif mehr kaufen, in dem Sie keine kostenfreien Zertifikate zu jeder Domain bekommen. Bei den meisten Hosting-Anbietern sollte SSL bereits bei der Einrichtung vorkonfiguriert sein. Bei bestehenden Websites sollte die Aktivierung nur wenige Minuten und ein paar Klicks in der Verwaltungsoberfläche Ihres Anbieters erfordern. Sollten Sie hier nicht weiterkommen, kann Ihnen der Support Ihres Hosting-Anbieters sicher weiterhelfen. Sollte dieser allerdings versuchen, Ihnen Geld für ein einfaches SSL-Zertifikat abzunehmen, wird es Zeit, Ihre Koffer zu packen und einen zeitgemäßeren Anbieter zu suchen.

18.2.2 SSL erzwingen

Nachdem Sie SSL für Ihre WordPress-Installation aktiviert haben und einen ersten Testaufruf Ihrer Website mit einem vorangestellten `https://` anstelle des `http://` erfolgreich war, wird es Zeit für den nächsten Schritt. Dieser Tage gibt es praktisch keinen Grund dafür, eine Nicht-SSL-Verbindung für eine Website

anzubieten. Wenn Sie alle Besucher automatisch auf die SSL-Version Ihrer Website leiten, vermeiden Sie Probleme mit mehreren parallel existierenden Links und gehen noch dazu sicher, dass nicht versehentlich doch einmal die Übertragung aus einem Kontaktformular über eine unsichere Nicht-SSL-Verbindung erfolgt.

Machen wir uns also daran, SSL für alle Aufrufe Ihrer Website zu erzwingen. Dafür bemühen wir einmal mehr die beiden Dateien *wp-config.php* und *.htaccess*.

1. **SSL in der WordPress-Konfiguration erzwingen**
   Öffnen Sie die Datei *wp-config.php* im Hauptverzeichnis Ihrer WordPress-Installation. Über der Zeile `/* Das war's, Schluss mit dem Bearbeiten! Viel Spaß beim Bloggen. */ /* That's all, stop editing! Happy blogging. */` fügen Sie nun zwei zusätzliche Zeilen ein:

   ```
   define('FORCE_SSL', true);
   define('FORCE_SSL_ADMIN', true);
   ```

   Listing 18.1 Erzwingen von SSL-Verbindungen in der »wp-config.php«-Datei

   Wenn Sie diese Änderung gespeichert und wieder auf den Server hochgeladen haben, weisen Sie WordPress an, alle Verbindungen zu Ihrer Website und dem Admin-Interface über SSL abzuwickeln.

2. **SSL über die Serverkonfiguration *.htaccess* erzwingen**
   Öffnen Sie die Datei *.htaccess* im Hauptverzeichnis Ihrer WordPress-Installation zur Bearbeitung. Ergänzen Sie ganz oben, noch über der Zeile `# BEGIN WordPress`, diese Konfiguration:

   ```
   RewriteEngine On
   RewriteCond %{HTTPS} !=on
   RewriteRule ^(.*)$ https://%{HTTP_HOST}%{REQUEST_URI} [L,R=301]
   ```

Listing 18.2 Umschreiben von http- zu https-Verbindungen in der ».htaccess«-Datei

Auch diese Änderung sollten Sie speichern und wieder auf Ihren Server hochladen.

Einmal an der richtigen Stelle platziert, wird diese Regel zukünftig abgerufene URLs Ihrer Website umschreiben, und zwar dann, wenn HTTPS nicht aktiv ist. Ist das der Fall, greift die Regel (`RewriteRule`), die angeforderte URL (bestehend aus `{HTTP_HOST}` und `{REQUEST_URI}`) mit vorangestelltem HTTPS aufzurufen. Hierbei handelt es sich um die letzte zu berücksichtigende Regel (`L`), die Seite solle sofort noch mal unter Berücksichtigung dieser neuen Regel geladen werden, und zwar ab jetzt immer (`R=301`, permanenter Redirect bzw. permanente Weiterleitung). Hakt es nach Einstellen dieser beiden Maßnahmen immer noch, liegt eine Besonderheit in der Serverkonfiguration vor. Setzen Sie sich auch dafür am besten mit dem Support Ihres Hosting-Anbieters in Kontakt.

18.2.3 Unsaubere HTTP-Elemente aufräumen

Vor allem bei bereits bestehenden WordPress-Websites kann es zu Problemen kommen, wenn für diese nachträglich SSL aktiviert wird. Weil WordPress die Referenzen zu Dateien wie Bildern oder extern eingebundene Inhalte mit komplettem Pfad in der Datenbank ablegt, ändert es auch die Angabe zur Nutzung von http oder https nicht in allen Fällen sofort. Wenn eine Website also zwar grundsätzlich angibt, per https aufgerufen werden zu wollen, aber einzelne Teile ihres Inhalts weiterhin über unsichere http-Verbindungen anbietet, warnen moderne Browser beim Seitenaufruf. Um diesen Warnungen vor Mixed-Content vorzubeugen, müssen Sie sich auf die Spurensuche nach all den

Webseitenelementen begeben, die unverschlüsselt übertragen wurden. WordPress wurde im vorherigen Schritt komplett auf SSL umgestellt, es handelt sich also entweder um Links, die fix z. B. in einem Plugin oder dem Theme verwendet werden, um auf Ressourcen zu referenzieren – zum Beispiel auf Schriften, JavaScript-Bibliotheken oder CSS-Frameworks. In vielen Fällen ist das Rätsel schnell gelöst: Öffnen Sie die Quelltextansicht der betreffenden Webseite, und suchen Sie nach »http:«. Überlegen Sie dann, welches Plugin, welche Website-Komponente den betreffenden Link produziert. Für die Website in diesem Beispiel war eine von Google Fonts nachgeladene Schrift verantwortlich (die Sie aus Datenschutzgründen nicht mehr direkt von Google-Servern einbinden sollten, sondern besser lokal hosten).

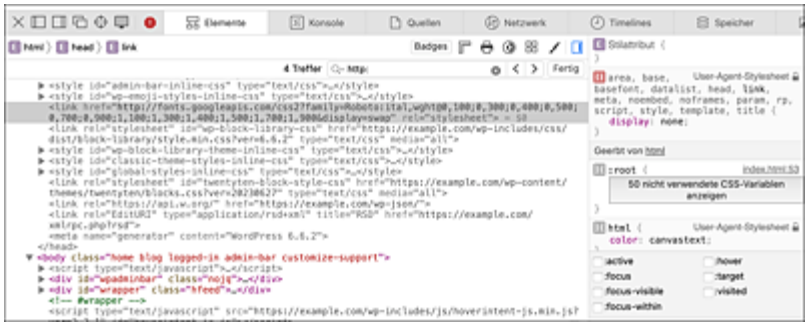


Abbildung 18.2 Zur Identifizierung der problematischen Elemente werfen Sie einen Blick in den Quelltext und suchen nach »http:«.

Falls Sie keine HTTP-Links im Quelltext ausfindig machen, verstecken sich die unsicheren Elemente etwas tiefer, beispielsweise in nachgeladenen CSS-Dateien oder von JavaScript generiertem HTML-Code. Nutzen Sie dann die Ansicht Web-Entwickler von Firefox, die Entwicklertools/Devtools von Chrome oder die Element-Informationen in Safari, um eine klare Fehlermeldung zu erhalten. Zu diesen Tools gelangen Sie entweder über das Menü oder einen Rechtsklick auf ein Element der Webseite und den Kontextmenüpunkt Untersuchen bzw. Element untersuchen.

Haben Sie den betreffenden Link lokalisiert, können Sie sich an die Korrektur machen. Bei Bildern und eingebundenen Inhalten wie Videos oder Social-Media-Posts werden Sie die entsprechenden Änderungen über den Seiten- oder Beitrags-Editor anpassen können. Bei Themes und Plugins könnten Sie selbst aktiv werden und deren Source-Code anpassen. In der Praxis führt das zu Problemen, weil Ihre Änderungen beim nächsten Plugin- oder Theme-Update wieder überschrieben würden. Für Themes gibt es dafür sogenannte Child-Themes. Mit Problemen dieser Art können Sie sich aber auch an die jeweiligen Anbieter des betreffenden Plugins oder Themes wenden.

18.3 WordPress abhärten

Neben diesen grundlegenden Maßnahmen gibt es eine ganze Reihe kleiner Stellschrauben und Konfigurationen, die Sie verändern können, um Ihre WordPress-Website noch etwas sicherer zu machen. Die folgenden Maßnahmen können Sie einzeln einbauen, entweder jetzt sofort oder später.

18.3.1 XML-RPC-Schnittstelle deaktivieren

Die *XML-RPC-Schnittstelle* wurde in WordPress früher genutzt, um zum Beispiel die WordPress-Apps mit WordPress nutzen zu können. Seit es die REST-API von WordPress gibt, ist das die bessere Alternative und die XML-RPC-Schnittstelle eigentlich nicht mehr notwendig, darüber hinaus birgt sie Sicherheitsrisiken. So wird durch Anfragen an die *xmlrpc.php* einiges an Last auf dem Server erzeugt, sodass ein gezielter Angriff auf die Schnittstelle mit vielen Anfragen (DDoS) einen Webserver in die Knie zwingen kann. Außerdem kann die XML-RPC-Schnittstelle für Brute-Force-Attacken genutzt werden, wie wir sie früher in diesem Kapitel auch schon kennengelernt haben. Diese Art von Angriff auf die Zugangsdaten eines WordPress-Users ist über XML-RPC noch einfacher und schneller, weil hier gleich mehrere Username-Passwort-Paare in einer einzigen Anfrage geprüft werden können.

In den meisten Fällen werden Sie die Schnittstelle über einen kleinen Codeschnipsel, den Sie in Ihre *.htaccess*-Datei einfügen, deaktivieren können.

```
<Files "xmlrpc.php">
  Require all denied
</Files>
```

Listing 18.3 Deaktivieren der XML-RPC-Schnittstelle in der ».htaccess«-Datei

Wenn Sie jetzt *example.com/xmlrpc.php* aufrufen (wobei Sie *example.com* durch Ihre Domain ersetzen), sollten Sie einen Zugriffsfehler erhalten. Damit kann diese Datei künftig nicht mehr aufgerufen werden, die Schnittstelle ist damit praktisch inaktiv.

Hinweis: Benutzen Sie die WordPress-App?

Die offizielle WordPress-App für Android und iOS nutzt nach aktuellem Stand (November 2024) noch die XML-RPC-Schnittstelle. Wenn Sie diese App also aktuell im Einsatz haben und auch weiterhin nutzen wollen, können Sie diese Schnittstelle also noch nicht deaktivieren. Kurz vor Redaktionsschluss dieses Buchs hat das Team hinter den offiziellen Apps aber erste Pläne diskutiert, zukünftig von XML-RPC zur moderneren REST-API zu wechseln, um die Kommunikation mit WordPress-Installationen sicherzustellen.

Je nachdem, wann Sie dieses Kapitel lesen, ist es also möglich, dass Sie auch als Benutzer*in der offiziellen App auf XML-RPC verzichten können.

18.3.2 REST-API prüfen

Die WordPress-REST-API ist seit vielen Jahren fester Bestandteil von WordPress. Für den Block-Editor ist sie essenziell, damit er seine Dienste verrichten kann, und auch für viele andere Anwendungsfälle ist die API sehr nützlich. Sie stellt Daten in strukturierter und einfach zu verarbeitender Weise (als JSON) bereit, was gleichzeitig Vor- und Nachteil ist. Denn viele dieser Daten sind nicht nur im eingeloggten Zustand sichtbar, sondern auch für alle

öffentlich zugänglich – so zum Beispiel die Liste der Beiträge (über *example.com/wp-json/wp/v2/posts*) oder Medien (über *example.com/wp-json/wp/v2/media*). Das ist nicht unbedingt direkt ein Sicherheitsproblem, erlaubt aber natürlich theoretisch jedem recht komfortabel, strukturiert an Daten Ihrer WordPress-Installation zu kommen. Um das zu verhindern, könnten Sie die API für nicht eingeloggte Besucher*innen deaktivieren. Dabei müssen Sie nur aufpassen, dass Sie keine Plugins einsetzen, die die REST-API nutzen, zum Beispiel Formular-Plugins, die Anfragen dann an die REST-API schicken, damit WordPress sie weiterverarbeitet. Mit Lösungen wie dem Plugin *Disable REST API* können Sie aber auch nur Teile der API deaktivieren.

Plugin	Disable REST API
Aktive Installationen	90.000+
Download	*https://de.wordpress.org/plugins/disable-json-api/*
Zweck	erlaubt es, Zugriff auf bestimmte Teile der REST-API einzuschränken
Komplexität	++

Sie können die REST-API auch vollständig deaktivieren, riskieren damit aber, dass essenzielle Teile von WordPress nicht mehr korrekt funktionieren. Daher ist dieser Schritt nicht unbedingt empfehlenswert.

18.3.3 Verzeichnisschutz per Basic Authentication

Auch mit sicheren Passwörtern und einer Zwei-Faktor-Authentifizierung kann sich eine weitere Sicherheitsmaßnahme rund um WP-Admin als sinnvoll erweisen. Eine solche Maßnahme

ist eine zusätzliche Passwortsperre auf Serverebene, noch bevor das Anmeldesystem von WordPress greift. Diese Passwortsperre bringt allerdings einen gewissen Komfortverlust mit sich. Bei der Sperre, von der wir hier sprechen, handelt es sich um die sogenannte Basic Authentication des Apache Webservers. Dieser Mechanismus öffnet im Webbrowser ein einfaches Benutzername/Passwort-Fenster, bevor der Inhalt der Seite geladen wird (siehe Abbildung 18.3). Wer ein falsches Passwort eingibt, wird vom Webserver zurückgewiesen und erhält eine Fehlermeldung.

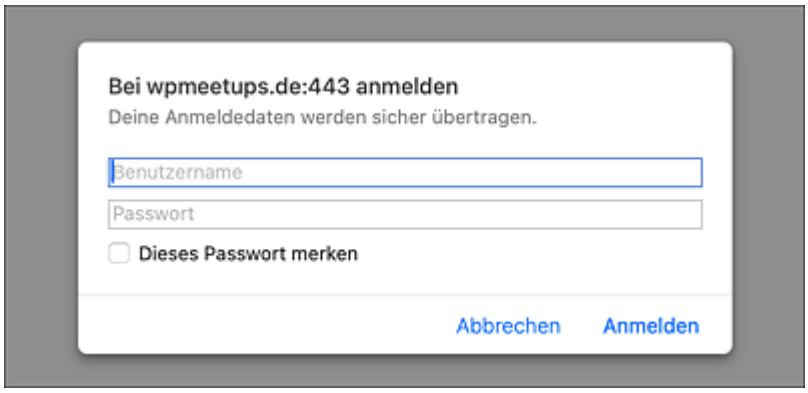


Abbildung 18.3 Über die »Basic Authentication« des Apache Webservers setzen Sie eine weitere Passwortsperre noch vor den Login-Mechanismus von WordPress.

Für die Basic Authentication ergänzen Sie eine kleine Konfiguration im Administratorverzeichnis von WordPress und erzeugen eine Datei, die ein verschlüsseltes Passwort enthält.

1. Erzeugen Sie im Verzeichnis */wp-admin/* eine neue leere Datei namens *.htaccess*, und öffnen Sie sie zur Bearbeitung mit Ihrem FTP-Programm. (In einer lokalen Installation: Sie legen eine leere Datei an und speichern diese unter */wordpresstest/wp-admin/.htaccess*. Unter Windows speichern Sie sie zunächst als *htaccess.txt* und benennen sie nach der Bearbeitung über den Explorer um, indem Sie am Anfang und Ende einen Punkt setzen.)
2. Fügen Sie folgendes Konfigurationsfragment in die Datei ein:

```
AuthUserFile /AbsoluterPfadZumIhremWordPressAdminOrdner/.htpasswd
AuthType Basic
AuthName "Passwort Abfrage"
Require valid-user
```

Listing 18.4 Erzwingen von SSL-Verbindungen in der ».htaccess«-Datei

Die erste Zeile bezieht sich mit der Direktive `AuthUserFile` auf eine Passwortdatei, die in Kürze erzeugt und im selben Verzeichnis wie die *.htaccess*-Datei gespeichert wird. Tragen Sie vor `/.htpasswd` den absoluten Serverpfad inklusive */wp-admin/*-Verzeichnis ein. In einer lokalen Testumgebung unter Windows ist das beispielsweise *c:\xampp\htdocs\wordpresstest\wp-admin/* (unter macOS */Applications/XAMPP/xamppfiles/htdocs/wordpresstest/wp-admin/* und unter Linux */opt/lampp/htdocs/wordpresstest/wp-admin/*). Der vollständige Pfad in einem angemieteten Webspace ist in der Regel länger und enthält z. B. Ihre Kundennummer. Sie erhalten den Pfad beispielsweise per PHP. Die zweite Zeile mit der Direktive `AuthType` beschreibt das Authentifizierungsverfahren, in diesem Fall die Basic Authentication. Es folgen die im Passwortfenster angezeigte Textmitteilung (`AuthName`) und schließlich die Mitteilung an den Webserver, dass ausschließlich authentifizierte Benutzer*innen (`valid-user`) Webseiten in diesem Verzeichnis aufrufen dürfen.

3. Speichern Sie die *.htaccess*-Datei, und erzeugen Sie eine neue Datei *.htpasswd* im selben Verzeichnis (bzw. *htpasswd.txt* unter Windows, Umbenennung dann wieder über den Explorer).

4. Das Passwort ist verschlüsselt, Sie benötigen deshalb ein Tool für die Erzeugung. Davon gibt es zahlreiche im Internet, beispielsweise *https://htpasswdgenerator.de*: einfach **Username** und **Password** eingeben und den erzeugten Code – genau eine

Zeile – in die Zwischenablage befördern, dann in die im Editor offene *.htpasswd*-Datei kopieren.

5. Speichern Sie die Passwortdatei, und prüfen Sie, ob nun bei Aufruf der Administrations-URL die HTTP-Authentifizierungsabfrage erfolgt.

Hinweis: Beliebige Verzeichnisse absichern

Auf diese Weise sichern Sie beliebige Serververzeichnisse ab. Dazu passen Sie lediglich den absoluten Pfad hinter `AuthUserFile` an und achten darauf, die erzeugte *.htpasswd*-Datei an die korrekte Stelle zu legen. Das funktioniert auch für die gesamte WordPress-Installation, z. B. für ein Testsystem – allerdings erzeugen Sie dann keine neue *.htaccess*-Datei, sondern fügen die vier Authentifizierungsdirektiven am Anfang der vorhandenen ein, denn WordPress hat solch eine Datei bereits für seine eigenen Zwecke angelegt (nichts löschen, sondern den `Auth`-Abschnitt darübersetzen).

Falls Sie sich jetzt nicht einloggen können, hat sich entweder beim Eingeben der *.htaccess*-Direktiven oder beim Schreiben des Passworts ins Login-Fenster oder in den *.htpasswd*-Generator ein Tippfehler eingeschlichen. Wiederholen Sie dann den gesamten in diesem Abschnitt vorgestellten Prozess, und löschen Sie Ihren Browser-Cache vor dem nächsten Login-Versuch; gegebenenfalls arbeiten Sie in einem neuen privaten oder Inkognito-Browserfenster.

Info: Serverpfad zur WordPress-Installation herausfinden

Für bestimmte programminterne Zwecke benötigen Sie den absoluten Pfad zu Ihrem WordPress-Installationsverzeichnis, z. B. um die Serverkonfiguration auf die Datei zu verweisen, die das Verzeichnispasswort enthält. Diesen Pfad finden Sie im WordPress-Backend unter **Werkzeuge** • **Website-Zustand** im Tab **Bericht** und dann in dem Bereich **Verzeichnisse und Größen**.

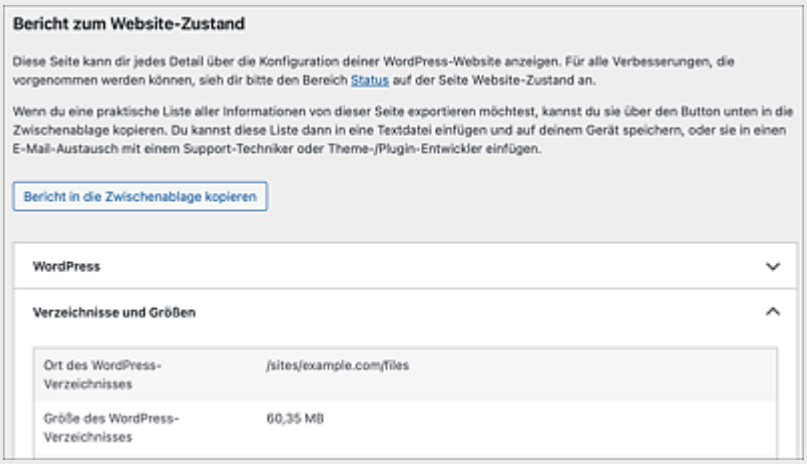


Abbildung 18.4 Im Tool »Website-Zustand« lässt sich der Serverpfad zur WordPress-Installation einfach herausfinden.

Den absoluten Pfad, den Sie hier herausgefunden haben, setzen Sie als Pfad für die *.htpasswd*-Datei in die *.htaccess* ein, beispielsweise so: `AuthUserFile /sites/example.com/files/wp-admin /.htpasswd`.

18.3.4 Datei-Editor deaktivieren

Ein neu installiertes WordPress ermöglicht Ihnen das Bearbeiten von Plugin- und Theme-Dateien direkt aus dem WP-Admin-Interface heraus. Unter Plugins • Datei-Editor können Sie zum Beispiel den Quelltext eines Plugins einsehen und direkt manipulieren (siehe Abbildung 18.5).

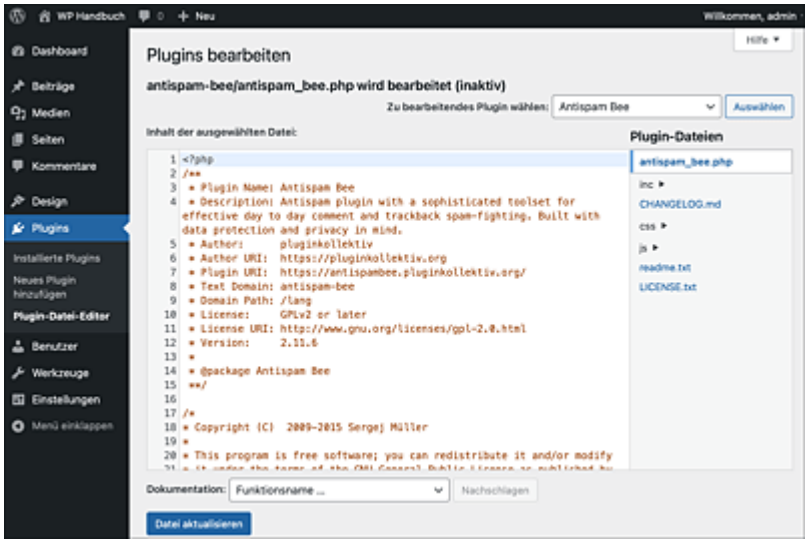


Abbildung 18.5 Im Datei-Editor können Sie Plugin- und Theme-Dateien bearbeiten und damit Probleme verursachen.

Neben der Bedrohung durch mutwillige Veränderung von Code durch einen Angreifer mit Admin-Rechten (gegen so ein Szenario wird eine effektive Absicherung schwer) gibt es vor allem den deutlich wahrscheinlicheren Fall der versehentlichen Bearbeitung von Code. Gegen den können Sie sich durchaus absichern.

Während es mit diesem Editor bis vor einigen Jahren noch möglich war, mit unbedachten Änderungen direkt schwerwiegende Fehler in einer WordPress-Installation zu erzeugen, ist der Editor in den letzten Jahren aber deutlich sicherer geworden. Mehrere Sicherheitsmechanismen sollen verhindern, dass Plugins schwer beschädigt werden und damit die ganze Website gegen die Wand fährt. Aber weniger schwerwiegende oder weniger offensichtliche Fehler sind nach wie vor nicht ausgeschlossen.

Wenn Sie den Editor auf Ihrer Website deaktivieren wollen, öffnen Sie die Datei *wp-config.php* im Hauptverzeichnis Ihrer WordPress-Installation. Über der Zeile `/* Das war's, Schluss mit dem Bearbeiten! Viel Spaß beim Bloggen. */ /* That's all, stop editing! Happy blogging. */` fügen Sie nun eine zusätzliche Zeile ein:

```
define( 'DISALLOW_FILE_EDIT', true );
```

Listing 18.5 Deaktivieren des Datei-Editors in der »wp-config.php«-Datei

Nachdem Sie diese Änderung gespeichert und die Datei wieder auf den Server hochgeladen haben, sollten die Menü-Einträge des Editors beim nächsten Neuladen der WP-Admin-Oberfläche verschwunden sein.

18.3.5 Plugin- und Theme-Installation verhindern

Eine weitere Anpassung, die nur in bestimmten Szenarien zusätzlichen Schutz bringt, dreht sich um die Installation und Deinstallation neuer Plugins und Themes. Um zu verhindern, dass Benutzer*innen mit Admin-Rechten selbstständig über die WP-Admin-Oberfläche Plugins oder Themes nachinstallieren, können Sie auch diese Option deaktivieren.

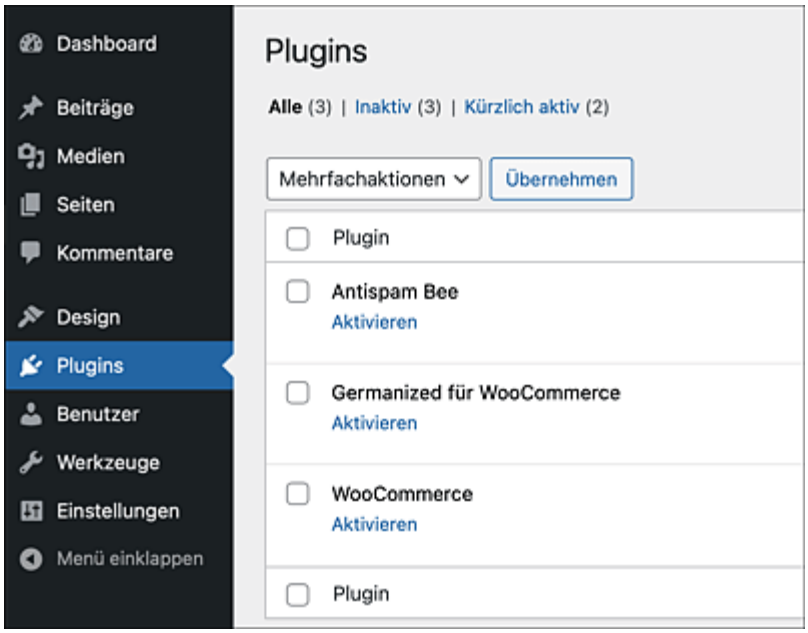


Abbildung 18.6 Wenn Sie das Installieren von Plugins und Themes deaktivieren, wird auch das Löschen installierter Plugins und Themes unmöglich.

Um für alle Benutzer*innen Ihrer Website das Installieren und Deinstallieren von Plugins und Themes zu deaktivieren, öffnen Sie die Datei *wp-config.php* im Hauptverzeichnis Ihrer WordPress-Installation. Über der Zeile `/* Das war's, Schluss mit dem Bearbeiten! Viel Spaß beim Bloggen. */ /* That's all, stop editing! Happy blogging. */` fügen Sie nun eine zusätzliche Zeile ein:

```
define( 'DISALLOW_FILE_MODS', true );
```

Listing 18.6 Deaktivieren von Plugin- und Theme-Veränderungen in der »wp-config.php«-Datei

Auch hier sollten die entsprechenden Optionen nach dem Speichern und Hochladen der *wp-config.php*-Datei auf den Server und einem erneuten Laden der WP-Admin-Oberfläche verschwunden sein (siehe Abbildung 18.6).

18.3.6 Zugriff auf IP-Adresse begrenzen?

Über die *.htaccess*-Datei lässt sich ein weiteres interessantes Sicherheits-Feature aktivieren, mit dem Sie so gut wie alle Unbefugten abblocken: eine IP-Sperre. Dazu müssen Sie Ihre öffentliche IP-Adresse kennen und den Apache-Webserver so konfigurieren, dass er nur Anfragen von dieser Adresse entgegennimmt. Alle anderen Anfragen würden ignoriert werden. Diese Einstellung ist sehr effektiv, aber nur dann sinnvoll, wenn Sie eine statische IP-Adresse haben. Weil das auf viele Internet-Anschlüsse nicht zutrifft und weil Sie Ihre Website vielleicht auch mal von unterwegs auf Ihrem Smartphone oder aus dem Co-Working-Space in Norddeutschland bearbeiten möchten, ist diese Option oftmals nicht wirklich praxistauglich.

Sollte Ihnen ein zusätzlicher Schutz Ihrer Website aber besonders wichtig sein oder möchten Sie zum Beispiel aus einem Firmen-

Netzwerk mit fester IP-Adresse auf den Admin-Bereich Ihrer Website zugreifen, so können Sie auch diese Änderung über die *.htaccess*-Datei im Hauptverzeichnis Ihrer WordPress-Installation einbauen. Soll nur der Administrationsbereich abgeriegelt werden, bearbeiten Sie die *.htaccess*-Datei aus dem Ordner */wp-admin/*. (Erzeugen Sie gegebenenfalls eine neue leere Datei, falls sie noch nicht existiert.) Ergänzen Sie folgende Zeilen:

```
Require all denied
Require local
Require ip 123.456.789.000
```

Listing 18.7 Verbietet Zugriffe, die nicht von einer festgelegten IP-Adresse stammen, über die ».htaccess«-Datei

Die erste Zeile sperrt für ausnahmslos alle den Zugriff auf das Verzeichnis (`all denied` – für alle verweigert). Die weiteren Zeilen heben dieses strikte Verbot für einige Ausnahmen auf:

- `Require local` stellt sicher, dass Sie lokal Zugriff auf das Verzeichnis haben.
- Die dritte Zeile (und beliebig viele darauffolgende Zeilen) dienen der Freischaltung einer weiteren IP. Anstelle der Beispiel-IP `123.456.789.000` setzen Sie Ihre öffentliche IP, die Sie herausfinden, indem Sie in Google »what is my ip« eingeben oder die Website *https://www.whatismyip.com* besuchen.

Speichern Sie die *.htaccess*-Datei, und testen Sie die IP-Sperre z. B. mit Ihrem Smartphone. Schalten Sie dazu WLAN aus, sodass Ihr Telefon sich in das Netzwerk des Mobilfunkanbieters einbucht. Die öffentliche IP des Telefonbrowsers ist damit eine andere als die Ihres Büro- bzw. Heimnetzwerks, sodass eine IP-Sperre erfolgt.

Tipp: Website im Intranet

Mit einer solchen *.htaccess*-Anpassung können Sie auch die komplette Website nur über eine (oder mehrere) definierte IP-Adressen abrufbar machen. Das ist ein einfacher Weg, um zum Beispiel ein digitales Schwarzes Brett für Ihr Unternehmen über ein Intranet verfügbar zu machen. Kolleg*innen im Büro-Netzwerk oder VPN (jeweils mit statischer IP) können dann auf die Installation zugreifen, während die Website von außerhalb nicht erreichbar ist.

18.4 No-Gos der WordPress-Sicherheit

Wenn Sie sich mit WordPress-Sicherheit beschäftigen und einschlägige Beiträge und Tutorials zum Thema lesen, werden Ihnen immer wieder Empfehlungen begegnen, die als effektiver Schutz vor Angriffen beworben werden.

In einigen Fällen sind diese Tipps aber eher für das gute Gefühl und sorgen nicht wirklich für ein Plus an Sicherheit. Einige solche Beispiele haben wir hier gesammelt, damit Sie nicht unnötig Zeit mit deren Umsetzung verbringen müssen.

18.4.1 Login verstecken

Viele Security-Plugins, aber auch einige der genannten Tutorials werden Ihnen anbieten oder empfehlen, Ihren WordPress-Login nicht länger unter *example.com/wp-login.php* beziehungsweise *example.com/wp-admin* erreichbar zu machen, sondern ihn an einen anderen Ort – etwa */anmeldung/* – zu verschieben.

In der Praxis bringt das für manuelle Angriffe auf Ihre Website vielleicht einen gewissen Schutz. Für alle Angriffe, die sich nicht auf den Login-Bereich konzentrieren, sondern stattdessen Schwachstellen in Plugins oder Themes ausnutzen, ist die Maßnahme aber komplett nutzlos. Wenn Sie sich Ihre WordPress-Installation als Haus vorstellen, würde das in etwa dem Verstecken der Eingangstür entsprechen. Weil sie irgendwo sein muss, wird auch eine versteckte Tür für einen Einbrecher kein wirkliches Hindernis darstellen. Und sollte er sie gar nicht finden, steigt er eben über ein Fenster oder den Dachboden ins Haus ein.

18.4.2 WordPress-Version verstecken

In dieselbe Kategorie der *Security by Obscurity* oder »Sicherheit durch Unklarheit« fällt auch der Versuch, die Versionsnummer von WordPress – oder allgemein die Nutzung von WordPress – zu verbergen. Die Idee dahinter klingt erst mal einleuchtend: Wenn ein Angreifer nicht weiß, dass Sie WordPress in Version X.Y einsetzen, kann er eine bestimmte Sicherheitslücke in dieser Version nicht gezielt angreifen. Wie wir bereits in der Info-Box in Abschnitt 18.1.2 zu Beginn dieses Kapitels erklärt haben, wird aber in den seltensten Fällen eine gezielte Attacke nur auf bestimmte WordPress-Versionen durchgeführt. Vielmehr probieren Angreifer automatisiert einfach eine ganze Batterie von bekannten Lücken bei jeder Website, die ihnen unterkommt.

Sie können also davon ausgehen, dass jede bekannte Attacke auf eine WordPress-Installation früher oder später auch auf Ihre Website angewendet werden wird. Ob Sie die Versionsnummer verbergen oder nicht, macht dabei keinen Unterschied. Stattdessen sollten Sie sich um schnelle Aktualisierungen bemühen, um so rasch wie möglich in den Genuss von Sicherheits-Updates zu kommen.

18.4.3 Datenbank-Präfix ändern

Mindestens genauso lange wie das zu versteckten Versionsnummern hält sich schon das Gerücht, das Ändern der Tabellenpräfixe, unter denen WordPress Daten in der Datenbank ablegt, könnte die Sicherheit einer Website erhöhen. Tatsächlich bringt diese Maßnahme keine Verbesserung der Sicherheit für Ihre Website und ist dagegen sogar mit einigen Risiken behaftet.

Die Idee ist, SQL-Injection-Angriffe zu unterbinden, bei denen ein Angreifer Zugriff auf Ihre Datenbank erlangt, indem er zum Beispiel eine Schwachstelle in einem Plugin ausnutzt. Tatsächlich ist das Sicherheitsspiel praktisch verloren, wenn ein Angreifer wirklich Zugriff auf Ihre Datenbank hat. Egal wie Tabellen benannt sind, sie können mit einem einfachen Befehl komplett gelöscht werden, und auch geänderte Tabellenpräfixe lassen sich mit einer SQL-Abfrage abrufen. Diese Maßnahme verzögert einen Angriff maximal um wenige Millisekunden, bietet aber keinen wirklichen Schutz.

Um bei der Metapher des Hauses zu bleiben, dem unsere Website entspricht: Das Ändern der Tabellenpräfixe hat einen ähnlichen Effekt, als würden Sie in Ihrem Haus das Licht ausschalten, um einen Angreifer davon abzuhalten, Ihre Wertsachen zu finden.

Stattdessen laufen Sie bei Manipulationen am Tabellen-Präfix auf einer Live-Website Gefahr, die Website unerreichbar zu machen, wenn Sie bei der Änderung einen Fehler machen. In vielen Fällen muss dann ein Backup bemüht werden, um den vorherigen Stand wiederherzustellen.

19 Performance-Optimierung

*Eine Website sollte möglichst schnell laden, um die Geduld der Besucher*innen nicht unnötig zu strapazieren oder im schlimmsten Fall Personen zu verlieren, die die Website wegen der langen Ladezeit direkt wieder schließen.*

Die Ladezeit einer Website ist ein Merkmal, das mitentscheidet, ob jemand längere Zeit auf einer Website verbringt oder sie schnell wieder verlässt. Fragen Sie sich selbst, wie viel Spaß es macht, sich durch eine Website zu klicken, wenn ein Seitenwechsel mehrere Sekunden dauert. Falls diese Website nicht gerade die einzige Anlaufstelle für die Informationen ist, beginnt vielleicht schnell die Suche nach einer performanteren Alternative.

Darüber hinaus ist eine schnell ladende Website auch aus Sicht der Suchmaschinenoptimierung vorteilhaft, da beispielsweise Google die Ladezeit mit als Rankingfaktor berücksichtigt.

In diesem Kapitel werden Sie lernen, wie Sie die Performance Ihrer Website testen können und welche Stellschrauben es gibt, um die Performance zu verbessern. Und warum es auch für den Klimaschutz gut ist, eine performante Website zu haben.

19.1 Grundlagen zur Optimierung

Wie schnell eine Website lädt, hängt im Wesentlichen von zwei Faktoren ab:

1. wie gut die Website optimiert ist und wie schnell der Webserver reagiert
2. wie gut die Internetverbindung des*der Benutzer*in beim Aufruf der Seite ist

An dem ersten Punkt können Sie etwas ändern, an dem zweiten Punkt nicht. Sie sollten aber, da nicht alle Menschen immer eine schnelle Internetverbindung haben, die Performance Ihrer Website auch mit gedrosselter Internetgeschwindigkeit testen. Denn bei der Performance-Optimierung kommt es nicht unbedingt darauf an, wie schnell eine Website objektiv fertig geladen ist, sondern wie schnell sie subjektiv für die Person lädt, die eine Website aufruft.

Das klingt jetzt vielleicht erst etwas abstrakt, lässt sich aber leicht veranschaulichen: Eine Website, die insgesamt eine Sekunde zum Laden braucht, bei der die ersten Elemente aber erst nach einer halben Sekunde für die Besucher*innen sichtbar sind, lädt subjektiv wahrscheinlich langsamer als eine Website, bei der die ersten Elemente schon nach einer viertel Sekunde auftauchen, der gesamte Ladevorgang aber zwei Sekunden dauert.

Wenn Sie sich ein Bild von der aktuellen Performance Ihrer Website machen möchten, können Sie einfach die Entwicklungswerkzeuge in Ihrem Browser öffnen (über einen Rechtsklick und dann einen Punkt Untersuchen oder ähnlich, abhängig vom Browser). In den Werkzeugen sollte es einen Netzwerk-Tab geben, oder Netzwerkanalyse, wie er in Firefox heißt. Wenn Sie diesen Tab öffnen und Ihre Website neu laden, sehen Sie eine Liste der *Requests*, die für die Anzeige der Website gemacht werden müssen, inklusive der Zeit,

die die einzelnen Requests brauchen. Außerdem sehen Sie unten die Gesamtzeit des Website-Aufbaus (siehe Abbildung 19.1).

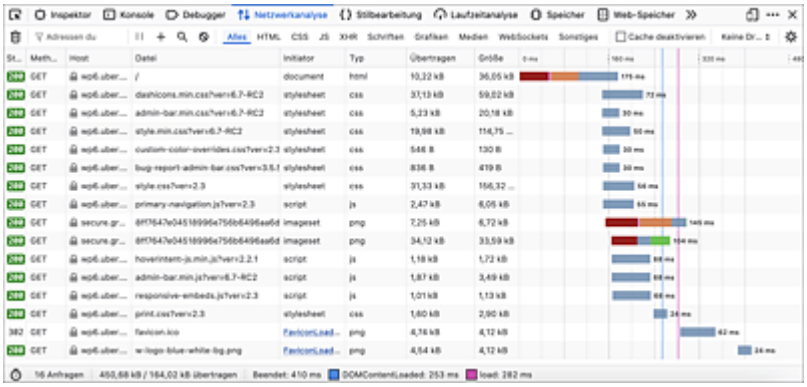


Abbildung 19.1 Sie können über den Netzwerk-Tab (Benennung je nach Browser unterschiedlich) einen Einblick in das Ladeverhalten Ihrer Website erhalten.

Firefox bietet auch an, die Seite mit simulierter gedrosselter Geschwindigkeit aufzurufen. Dafür können Sie rechts in dem Netzwerkanalyse-Tab bei dem Dropdown mit dem Wert Keine Drosselung einen anderen Wert einstellen und die Website neu laden. Darüber hinaus können Sie den Cache deaktivieren. So können Sie sich gut ein Bild davon machen, wie der erste Aufruf Ihrer Website mit einer langsameren Internetverbindung aussehen kann.

Sie können die Anfragen nach der Art der angefragten Ressourcen filtern, also sich zum Beispiel nur die Requests für JavaScript-Dateien anzeigen lassen. Sie können die Aufrufe auch anders sortieren, beispielsweise nach der Grösse, um sich die Anfragen mit den größten Antworten oben anzeigen zu lassen.

In dieser Ansicht erhalten Sie schon erste Anhaltspunkte für eine Optimierung. Wenn es zum Beispiel relativ lange dauert, bis das eigentliche HTML-Dokument übertragen ist (sagen wir 250 Millisekunden), auch wenn es von der Dateigröße nur relativ klein ist, kann das ein Hinweis darauf sein, dass etwas beim Zusammenbauen der HTML-Antwort auf dem Server länger braucht

als vielleicht notwendig. Dafür können Sie sich in Firefox die Zeit der Übertragung genauer anzeigen lassen, sodass sie nach den unterschiedlichen Phasen des Requests aufgeschlüsselt wird. Wenn die Phase Warten lange dauert, ist das die Zeit, in der der Request vom Server bearbeitet wird.

Info: Caching-Plugins als letztes Mittel der Optimierung

Wenn Ihre Website langsam oder zumindest nicht schnell ist, kommt vielleicht schnell die Idee auf, es mit einem Caching-Plugin zu versuchen. Ein Caching-Plugin sollte aber eingesetzt werden, um eine sonst bereits optimierte Seite noch schneller auszuliefern. Wenn Sie eine nicht optimierte Seite gecacht ausliefern, lädt sie zwar vermutlich deutlich schneller als ohne Caching-Plugin, aber nicht so schnell, wie sie im optimierten Zustand laden könnte. Kümmern Sie sich also erst um die Optimierungen abseits vom Caching, bevor Sie sich ein Caching-Plugin suchen.

19.2 PageSpeed Insights – Missstände verstehen und bei Bedarf abarbeiten

Ein beliebtes Tool zum Testen der Performance von Websites ist *PageSpeed Insights* von Google (*https://pagespeed.web.dev/*). Nicht alle Punkte, die bei der Analyse gefunden werden, müssen zwangsläufig für eine schnelle Website behoben werden, aber Sie erhalten einen guten ersten Anhaltspunkt zur Optimierung.

Öffnen Sie also die Website des Tools, und geben Sie die URL der Startseite Ihrer Website in das URL-Feld ein. Bei der Ergebnisansicht kommen zuerst fünf Metriken für die Leistung der Website (siehe Abbildung 19.2), gefolgt von einer Auflistung konkreter Probleme und möglicher Lösungswege. Wichtig: Testen Sie unterschiedliche Seiten Ihrer Website, neben der Startseite zum Beispiel eine Übersicht der Beiträge, eine Einzelansicht einer Seite und eines Beitrags, und, wenn vorhanden, andere Arten von Seitentypen.

Die Bewertung von PageSpeed Insights wird in die vier Bereiche Leistung, Barrierefreiheit, Best Practices und SEO aufgeteilt. Im Folgenden werden nach einer Einleitung zu den fünf von PageSpeed prominent dargestellten Messwerten die durchgeführten Tests durchgegangen, wobei wir uns auf die wichtigsten beschränken.

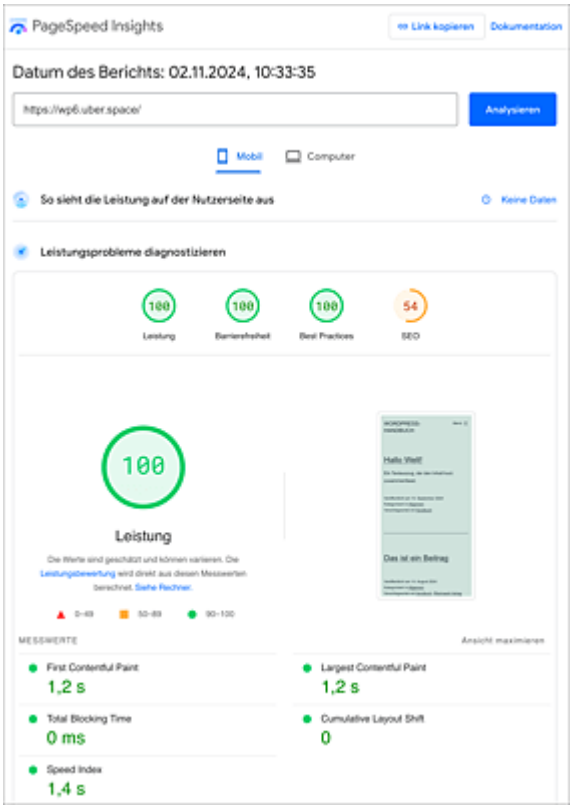


Abbildung 19.2 Der erste Teil der Performanceanalyse mit PageSpeed Insights zeigt die Messwerte für die Bewertung der »Leistung«.

19.2.1 Core Web Vitals und weitere Messwerte

Die Core Web Vitals sind eine 2020 von Google vorgestellte Reihe von Messwerten (die kontinuierlich weiterentwickelt werden), um die User Experience einer Website zu bewerten, und seit Juni 2021 werden diese Werte auch in das Ranking von Google einbezogen. Die Web Vitals bestehen aktuell aus den Werten Largest Contentful Paint (LCP), Cumulative Layout Shift (CLS) und Interaction to Next Paint (INP).

Die Core Web Vitals werden in der PageSpeed-Auswertung angezeigt, aber nicht alle in jedem Fall. Die Bewertung kann auf Daten von echten Nutzer*innen basieren (Felddaten) oder auf Daten

eines Tests von Google (Lab-Daten). INP ist ein Wert, der nur in den Auswertungen nach Felddaten auftaucht.

- **First Contentful Paint (FCP)**: Die Zeit, die vom Beginn des Ladens einer Seite bis zum ersten angezeigten Element vergeht. Zeit für eine gute Bewertung: 1,8 Sekunden oder weniger.
- **Largest Contentful Paint (LCP)**: Die Zeit vom Start des Ladens einer Seite, bis das größte Bild- oder Textelement im sichtbaren Bereich gerendert ist. Ein guter Wert sind 2,5 Sekunden oder schneller.
- **Interaction to Next Paint (INP)**: Misst die Zeit, die eine Website während der Nutzung durch Nutzer*innen braucht, um auf Aktionen zu reagieren. Der Wert dieser Metrik ist die Dauer, unter der auf (fast) alle Interaktionen reagiert wurde. Ein guter Wert sind hier 200 Millisekunden oder weniger.
- **Total Blocking Time (TBT)**: Die Zeit, die eine Website nach dem *First Contentful Paint* blockiert ist, bevor sie auf Aktionen von Benutzer*innen reagieren kann, wie etwa Mausklicks oder Tastatureingaben. Maximal 200 Millisekunden sind hier ein guter Wert.
- **Cumulative Layout Shift (CLS)**: Ein etwas komplexerer Wert, der die größte Summe an Scores von Layout-Shifts angibt, die in weniger als einer Sekunde in einem maximal fünf Sekunden dauernden Zeitfenster aufeinanderfolgen. Ein Layout-Shift bedeutet, dass sich die Startposition eines bereits sichtbaren Elements nachträglich verändert, beispielsweise weil ein Bild nachgeladen wurde und die Elemente darunter weiter nach unten verschoben werden. Jeder Layout-Shift erhält einen Score, der sich aus dem prozentualen Anteil errechnet, den das verschobene Element im aktuell sichtbaren Bereich (Viewport) einnimmt, der sogenannten Impact Fraction, sowie aus dem

prozentualen Wert, um den das Element im Viewport verschoben wird (hierbei wird die längste Distanz, um die sich das Element im Viewport verschoben hat, durch die längste Seite des Viewports geteilt), der sogenannten Distance Fraction. Ein guter Wert ist 0,1 oder weniger.

- **Speed Index**: Misst, wie schnell der Inhalt während des Ladens der Seite sichtbar ist. Ein Wert schneller als 3,4 Sekunden ist gut.

Entscheidend für die Bewertung bei den Metriken ist das 75. Perzentil der Seitenaufrufe. Das bedeutet, wenn mindestens 75 % der Seitenaufrufe für eine Metrik den guten Wert erreichen oder unterschreiten, wird die Metrik als gut bewertet.

Info: Mehr Informationen zu den Metriken

Auf *web.dev* gibt es mehr Informationen zu den Core Web Vitals und anderen Metriken, unter anderem gibt es dort zu jedem der kurz vorgestellten Werte eine eigene Seite (*web.dev/fcp/*, *web.dev/lcp/*, *web.dev/inp/*, *web.dev/tbt/*, *web.dev/cls/* und *web.dev/si/*), auf der genau beschrieben ist, wie der Wert berechnet wird und welche Maßnahmen ergriffen werden können, um bessere Werte zu erreichen. Viele der dort aufgelisteten Maßnahmen werden im Folgenden genauer erläutert.

19.2.2 PageSpeed-Tests im Bereich »Leistung«

Die eben schon beschriebenen fünf Metriken sind der Beginn der Auswertungen für den Bereich Leistung. Unter den Metriken finden Sie eine Liste unter der Überschrift Diagnose, die Punkte auflistet, in denen PageSpeed Insights Verbesserungspotenzial sieht (siehe

Abbildung 19.3). Sortiert sind die Punkte nach einer von dem Tool vorgenommenen Gewichtung.

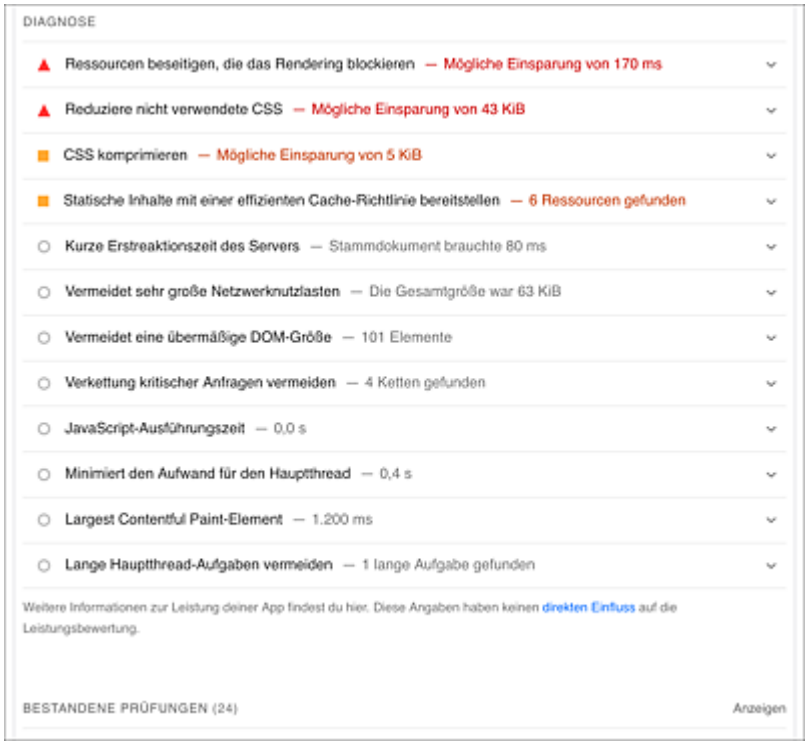


Abbildung 19.3 Unter »Diagnose« werden Punkte aufgelistet, bei denen Sie aus Sicht von PageSpeed Insights tätig werden könnten.

Unter den Punkten zu Diagnose sehen Sie Bestandene Prüfungen, also Tests, bei denen die Website bereits gut abgeschnitten hat.

Große Netzwerknutzlasten vermeiden

Eigentlich der naheliegendste Ansatz zum Verbessern der Performance: die Menge der Daten reduzieren, die geladen werden. Neben Performanceproblemen sorgen »schwere« Websites nämlich auch für höhere Kosten bei Nutzer*innen ohne Daten-Flatrate oder ein schneller aufgebrauchtes Datenvolumen.

Einige der hier vorgestellten anderen Optionen sorgen bereits dafür, dass weniger Daten übertragen werden. Daneben haben Sie aber

eine einfache weitere Möglichkeit, die Größe Ihrer Website stark zu reduzieren: überflüssige Elemente entfernen. Das klingt erst mal vielleicht etwas krass, aber schauen Sie sich einmal jedes Element Ihrer Website an und stellen Sie sich die Frage, ob der Mehrwert des Elements für die Benutzer*innen vielleicht gar nicht so groß ist und ob beispielsweise ein Slider wirklich notwendig ist. Vielleicht tut es auch ein statisches Bild, oder Sie kommen direkt zum wesentlichen Inhalt und sparen so die Daten der Bilder, des Slider-Scripts und der Styles für den Slider ein.

Rendering-blockierende Ressourcen beseitigen

Externe JavaScript- und CSS-Dateien blockieren das Rendering, während der Browser das JavaScript beziehungsweise CSS interpretiert. Als Render-blockierend werden externe JavaScript-Ressourcen innerhalb des `head` angesehen, die weder das `defer`- noch das `async`-Attribut gesetzt haben. Mit `defer` können Sie angeben, dass das Skript erst ausgeführt werden soll, wenn das gesamte Dokument geparst ist. `async` lädt ein Skript parallel zum Parsen eines Dokuments und führt es aus, sobald es geladen ist.

Seit WordPress 6.3 gibt es die Möglichkeit, bei dem Einbinden von Skripten über die Scripts-API von WordPress auch diese beiden Attribute zu definieren, und seit WordPress 6.4 wird das `defer`-Attribut für Skripte des WordPress-Core und von Standard-Themes genutzt. Mit etwas Glück müssen Sie hier also nicht groß tätig werden, wenn auch die von Ihnen eingesetzten Plugins und das eingesetzte Theme diese neuen Möglichkeiten nutzen. Sollte es trotzdem große Einsparmöglichkeiten wegen Render-blockierenden Skripten geben, können Sie mit einem der im späteren Verlauf des Kapitels vorgestellten Optimierungs-Plugins wie Autoptimize oder WP Rocket gegensteuern.

Bei Stylesheets ist die Lösung etwas schwieriger, weil Sie diese nicht einfach über `async` oder `defer` anders laden können. Stylesheets werden als Render-blockierend angesehen, wenn sie nicht das `disabled`-Attribut haben oder keine spezifische `media`-Angabe, damit sie nur unter bestimmten Umständen geladen werden. Die vorgeschlagenen Lösungen von Google sind:

- Kritisches CSS inline einbinden, das für die Darstellung der direkt beim Laden einer Seite sichtbaren Elemente zuständig ist. Dafür gibt es zwar Tools, Sie sollten aber genau testen, ob das wirklich Vorteile bringt oder nur sehr viel CSS inline eingebunden wird und damit Ihr HTML-Dokument deutlich größer wird.
- Die Stylesheets in unterschiedliche Stylesheets aufteilen, gruppiert nach Media-Queries, und beim Einbinden der Dateien für jede Datei den passenden Media-Query über das `media`-Attribut angeben. Auch das ist je nach Website ein komplexes bis hoffnungsloses Unterfangen, erst recht für Stylesheets von Dritt-Plugins oder -Themes.
- Das CSS minifizieren, um nur so wenig Dateigröße wie möglich zu übertragen. Das ist zumindest eine recht einfach umzusetzende Möglichkeit. Plugins, die dabei behilflich sind, lernen Sie später noch kennen.

Nicht verwendete Ressourcen (JS und CSS) entfernen

Eigentlich würde man annehmen, dass auf einer Website nur Dinge geladen werden, die auch notwendig sind. Es kommt aber beispielsweise vor, dass ein Plugin genutzt wird, das auf allen Seiten Stylesheets und JavaScripts lädt, obwohl das Plugin nur auf einzelnen Seiten zum Einsatz kommt. In solchen Fällen kann bei populären Plugins oder Themes eine Suche in der Suchmaschine

Ihrer Wahl zum Erfolg führen, vielleicht hat sich schon jemand dieses Problems angenommen und die Lösung ins Internet gestellt.

Andernfalls müssten Sie nachforschen, ob es eine Möglichkeit gibt, zu erkennen, auf welchen Seiten die Komponente, für die Style und/oder Script gebraucht wird, zum Einsatz kommt, um es in allen anderen Fällen mit `wp_dequeue_style()` und `wp_dequeue_script()` zu entfernen. Die Analyse und Umsetzung der Lösung ist aber eher etwas für Fortgeschrittene.

CSS und JavaScript minifizieren

Die Minifizierung von Textdateien wie JavaScript und CSS meint, dass alle unnötigen Leerzeichen, Umbrüche und Leerzeilen entfernt werden. Darüber hinaus kann bei CSS und JavaScript, in der Regel durch ein automatisiertes Tool beim Umwandeln der Quell-Dateien in die Version für die Produktiv-Umgebung, auch eine Minifizierung der Werte stattfinden, etwa indem Farbwerte im CSS in kürzerer Schreibweise angegeben werden oder Namen von Funktionen, Variablen und anderen Bezeichnern in JavaScript auf einen oder zwei Buchstaben gekürzt werden.

Tools wie das später vorgestellte WP Rocket können Ihnen dabei helfen, die Ressourcen Ihrer Website zu minifizieren, allerdings kann es unter Umständen durch die Minifizierung zu Problemen kommen. Testen Sie daher nach Aktivierung der Option gründlich alle Funktionen, und deaktivieren Sie die Minifizierung gegebenenfalls wieder.

Aufwand für Haupt-Thread minimieren und Ausführungszeit von JavaScript reduzieren

Der Haupt- oder Main-Thread ist im Browser standardmäßig der Prozess, der sich um die komplette Darstellung der Website sowie um die Verarbeitung der Aktionen von Nutzer*innen kümmert. Der Thread parst zuerst das HTML und baut das DOM, interpretiert das CSS und führt JavaScript aus. Und sobald der Thread mit so etwas beschäftigt ist, kann er keine User-Aktionen verarbeiten, etwa Klicks auf Links oder Buttons.

Daher sollte die Zeit, die der Haupt-Thread mit der Verarbeitung der Website verbringt, so kurz wie möglich gehalten werden. Offensichtlich der einfachste Weg zu weniger Last für den Haupt-Thread wäre zu prüfen, ob alle geladenen Scripts und Stylesheets wirklich notwendig sind. Durch die schon angesprochene Minifizierung der Ressourcen lässt sich zumindest die Zeit reduzieren, die für die Anfragen notwendig ist.

Ein weiterer Weg ist, Ressourcen nachzuladen, die nicht direkt beim Aufruf der Seite benötigt werden, zum Beispiel eine JavaScript-Datei, die erst nach einem Klick auf einen Button ausgeführt werden muss. Für diesen Fall könnte das Script beim Klick auf den Button nachgeladen werden, sodass es den initialen Aufbau der Website nicht bremst. Auch die bereits angesprochene Nutzung von `async`- und `defer`-Attributen für Scripts kann helfen. Insgesamt ist das aber eher ein Optimierungspunkt für Fortgeschrittene.

Bilder optimieren und richtig dimensionieren

Am Ende dieses Kapitels wird darauf eingegangen, wie Sie die Dateigröße von Bildern vor oder noch besser beim Upload optimieren können.

Daneben ist es wichtig, dass Bilder ungefähr in der richtigen Größe geladen werden, also auf einem Smartphone nicht das Bild in 2000

Pixel Breite, sondern eine deutlich kleinere Version. Das wird von WordPress schon automatisch durch die Nutzung der Technik *Responsive Images* vorbereitet. Dadurch können mehrere Varianten eines Bilds angeboten werden, und der Browser sucht sich die richtige zum aktuellen Kontext der Nutzer*in heraus.

Ebenfalls wichtig ist, dass bei Bildern das `width`- und `height`-Attribut gesetzt wird, sodass der Browser schon den korrekten Platz für das Bild berechnen und »freihalten« kann, auch wenn es noch nicht geladen ist.

Lazy Loading von Bildern (und anderen Medien)

Bilder und andere Medien wie Videos oder iFrames sollten nicht direkt beim initialen Laden der Seite mit angefragt werden, sondern erst versetzt, wenn das Medium in die Nähe des sichtbaren Bereichs kommt.

In WordPress werden Bilder mit einem `width`- und `height`-Attribut seit WordPress 5.5 standardmäßig mit dem `loading`-Attribut ausgestattet, sodass Browser die Bilder nachladen. Seit WordPress 5.7 werden auch iFrames mit `width` und `height` mit dem Attribut versehen. In WordPress 5.9 wurde das Verhalten leicht angepasst, sodass das erste Bild oder iFrame im Inhalt nicht mehr nachgeladen wird, da es damit leichte Verschlechterungen des Werts für den »Largest Contentful Paint« gab.

Wenn Sie Videos einsetzen, sollten Sie diese möglichst über eine Plattform wie Vimeo oder YouTube einbinden. Dafür werden iFrames genutzt, weshalb Sie sich dann nicht um das Lazy Loading kümmern müssten. Bei selbst gehosteten Videos können Sie beim Video-Block angeben, welche Daten vom Video vorgeladen werden.

Damit können Sie einstellen, dass nicht direkt die Video-Datei geladen wird, sondern beispielsweise erst mal nur die Metadaten.

Caching

Statische Ressourcen wie Bilder, Schriften, JavaScript- und CSS-Dateien sollen vom Server mit Caching-Informationen für den Client (also Browser) ausgeliefert werden, sodass der Browser beispielsweise weiß, dass die Schriften für einen Monat im lokalen Browser-Cache gecacht werden sollen. Wenn die URL für die Schriften dann bei einem neuen Request angefragt wird, holt sich der Browser die Dateien aus dem Cache.

Ob ein Request aus dem Cache beantwortet wurde, können Sie in den Netzwerk-Tools der Browser-Dev-Tools sehen. In Firefox wird in der Übertragen-Spalte angezeigt, ob die Ressource aus dem Cache kommt (siehe Abbildung 19.4), bei Chrome sehen Sie es in der Size-Spalte.

Wenn Sie auf einen Request klicken, der nicht aus dem Cache kommt, und sich die Response-Header anschauen (siehe Abbildung 19.5), wird im Eintrag »Cache-Control« festgelegt, wie lange die Ressource in Sekunden gecacht werden soll.

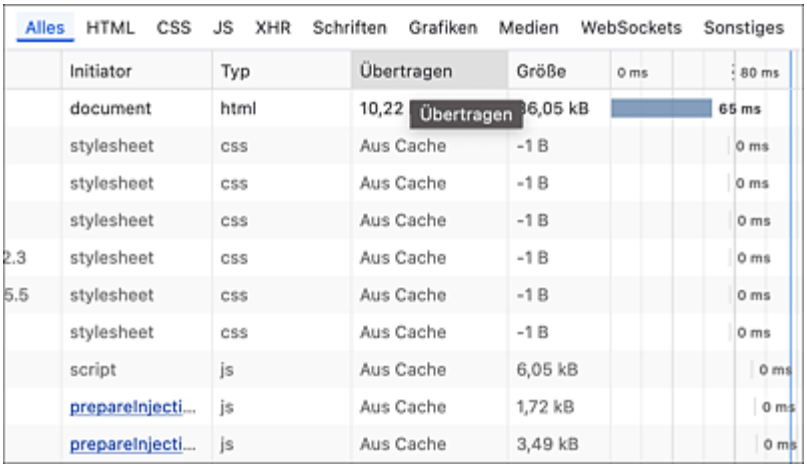


Abbildung 19.4 In Firefox können Sie in der Spalte »Übertragen« sehen, ob eine Anfrage aus dem Cache beantwortet wurde.

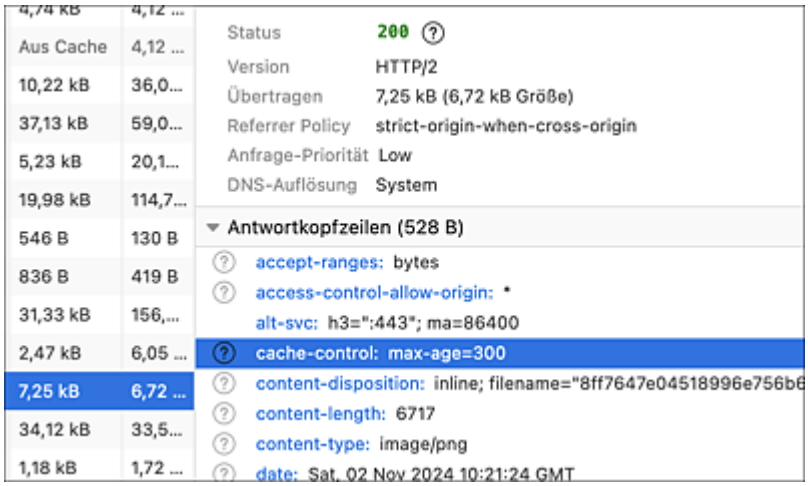


Abbildung 19.5 Anhand des Response-Headers »Cache Control« können Sie erkennen, ob vom Server eine Cache-Policy für eine bestimmte Ressource angegeben wurde. Das ist der Teil, auf den PageSpeed Insights prüft.

Für Ressourcen wie Schriften kann eine lange Cache-Zeit meist ohne Probleme eingestellt werden, da sich diese Dateien selten ändern. Wenn Sie aber beispielsweise eine Cache-Zeit von 30 Tagen für CSS-Dateien einstellen und dann Änderungen an dieser Datei gemacht werden, kriegen das im schlimmsten Fall Nutzer*innen, die direkt vorher die Seite besucht haben, erst nach 30 Tagen mit.

Umgehen lässt sich das, wenn sich der Dateiname der CSS-Datei mit der Änderung auch verändert. Das ist die Methode, die Plugins wie das später vorgestellte WP Rocket nutzen. Aber an dem Beispiel sehen Sie, dass lange Cache-Zeiten nicht immer unproblematisch sind.

Wenn Sie für eine Ressource eine `no-cache`-Policy angeben, wird sie von PageSpeed Insights nicht mit in die Prüfung aufgenommen. `no-cache` bedeutet aber nicht, dass die Ressource gar nicht gecacht wird, sondern nur, dass Browser vor dem Lesen aus dem Cache prüfen müssen, ob die Ressource vom Server sich geändert hat.

Wenn Sie kein Plugin für Caching einsetzen, können Sie die Caching-Regeln manuell über die *.htaccess* definieren:

```
<IfModule mod_expires.c>
  ExpiresActive On
  ExpiresDefault "access plus 1 month"
  ExpiresByType text/css "access plus 3 months"
  ExpiresByType application/rss+xml "access plus 1 hour"
  # …
</IfModule>
```

Listing 19.1 Beispielhaftes Setzen von Cache-Headern

In dem Beispiel wird mit `ExpiresDefault` festgelegt, dass die Standard-Cache-Dauer einen Monat betragen soll. Für CSS-Dateien und RSS-Feeds überschreiben wir diese Werte mit `ExpiresByType`.

Request-Anfragen mit Gzip oder Brotli komprimieren

Sie können die Übertragungsgröße der Anfragen verringern, indem Sie die Antworten vom Server komprimiert übertragen lassen. Dabei gibt es unterschiedliche Möglichkeiten der Komprimierung. Gzip kann wohl als der Standard bezeichnet werden, der schon lange existiert und auch in sehr alten Browsern unterstützt ist.

Ebenfalls schon fast 10 Jahre dabei ist Brotli, ein Kompressionsalgorithmus von Google aus dem Jahr 2015. Verglichen mit Gzip liefert Brotli noch einmal kleinere Dateigrößen, und der Browser-Support ist ebenfalls gut.

Da Brotli aber im Gegensatz zu Gzip bei Webhostern nicht unbedingt verfügbar ist, sehen Sie hier die Methode zur Komprimierung mit Gzip. Wenn Ihr Webhoster Brotli unterstützt, können Sie Brotli als Default nutzen und Gzip als Fallback, wenn der Browser kein Brotli kann. Dafür kann der `Accept-Encoding`-Anfrage-Header nach »br« durchsucht werden, was angibt, ob der Browser Brotli unterstützt.

Im nachfolgenden Beispiel wird die Gzip-Komprimierung für CSS- und JavaScript-Dateien sowie HTML aktiviert. Ob die Komprimierung funktioniert, können Sie wie beim Cache über die Netzwerk-Tools nachschauen und dort über den Antwort-Header `Content-Encoding` (siehe Abbildung 19.6).

```
<IfModule mod_deflate.c>
    AddOutputFilterByType DEFLATE text/css
    AddOutputFilterByType DEFLATE text/javascript
    AddOutputFilterByType DEFLATE text/html
</IfModule>
```

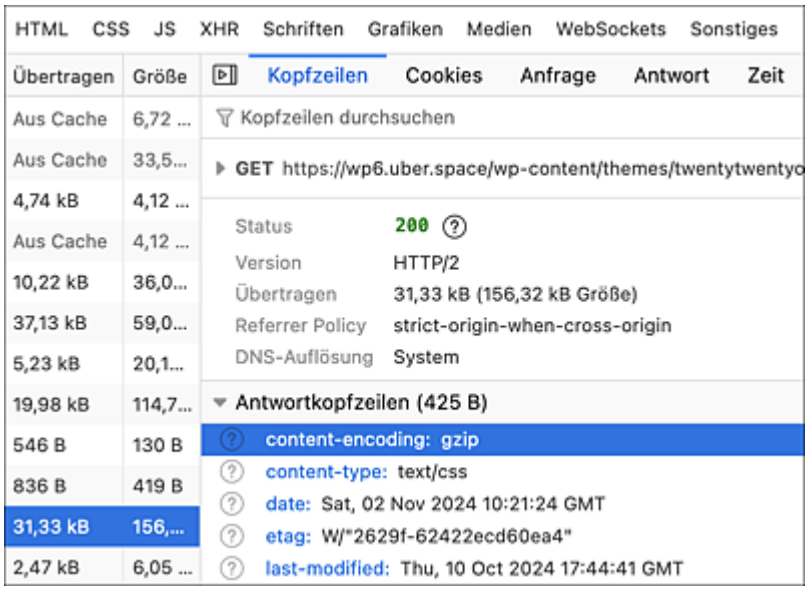


Abbildung 19.6 Der Antwort-Header »Accept-Encoding« gibt an, ob die Antwort komprimiert ist.

Übermäßige DOM-Größe vermeiden

Je größer das DOM einer Webseite ist (das Document Object Model, das die Struktur einer Webseite darstellt und für Änderungen beispielsweise durch JavaScript zugänglich macht), desto mehr Aufwand ist zur Darstellung der Seite notwendig. Daher sollte das HTML-Markup der Website möglichst einfach sein, ohne unnötig tiefe Verschachtelungen.

Schnelle Reaktion des Servers

Wenn der Browser schon lange auf die Auslieferung des HTML-Dokuments warten muss, ist das ein Zeichen dafür, dass Optimierungspotenzial besteht. Der Grund für eine lange Wartezeit des Browsers auf den Server könnte damit zusammenhängen, dass in WordPress oder einem Plugin zeitintensive Aufgaben ablaufen, die optimiert werden können.

Zum Finden und Beheben dieser Probleme sind allerdings Kenntnisse in der Entwicklung notwendig, und wenn das Problem im Code eines Dritt-Plugins oder -Themes liegt, können Sie selbst sowieso schlecht tätig werden, außer eventuell einen Hinweis auf das Problem im Support-Forum oder in einem Issue-Tracker schreiben, gegebenenfalls mit einem Lösungsansatz.

Mit einem Caching-Plugin wie Cachify oder WP Rocket kann das Problem umgangen werden, indem eine statische Version einer Seite generiert und ausgeliefert wird, sodass WordPress nur noch beim ersten Seitenaufruf loslaufen muss und alle nachfolgenden Besucher*innen die statische Version bekommen. Wie am Anfang des Abschnitts im Kasten angemerkt, ist dieses Erschlagen eines Performance-Problems mit einem Cache-Plugin nicht die ideale Lösung, aber manchmal bleibt Ihnen nichts anderes übrig.

Verkettung kritischer Anfragen vermeiden

Sie können über das `preload`-Attribut angeben, dass Ressourcen mit einer höheren Priorität geladen werden sollen als andere. Das kann hilfreich sein, um beispielsweise schnell die richtige Schrift anzeigen zu können. Wenn Sie aber zu viele solche Ressourcen haben und eventuell aus diesen Dateien noch andere Dateien geladen werden, kann das den Seitenaufbau verlangsamen.

Vorverbindungen zu wichtigen Drittquellen herstellen

Nach Möglichkeit sollten Sie keine externen Drittquellen einbinden, aber falls Sie doch welche haben, können Sie ein `link`-Attribut mit `rel="preconnect"` einbinden, damit der Browser weiß, dass die Website eine Verbindung zu der als `href` übergebenen Domain aufbauen will. Das kann so aussehen:

```
<link rel="preconnect" href="https://example.com">
```

Mehrere Weiterleitungen vermeiden

Wenn jemand Ihre Website als *http://example.com* aufruft und dann erst eine Weiterleitung auf *https://example.com* stattfindet, gefolgt von einer Weiterleitung auf *https://www.example.com*, kann die mittlere Weiterleitung übersprungen werden, um Zeit zu sparen.

Wenn solche Weiterleitungen erkannt werden, können Sie sie beispielsweise durch eine bessere Organisation der Weiterleitungen in der *.htaccess* optimieren.

Videoformate für animierte Inhalte verwenden

GIF-Dateien können schnell groß werden, weshalb die Empfehlung lautet, Animationen als Video bereitzustellen.

Veraltetes JavaScript in modernen Browsern vermeiden

Eher ein Tipp für Profis: Während der Entwicklung kann modernstes JavaScript geschrieben werden, das dann über einen Build-Step umgewandelt wird, sodass es mithilfe von Polyfills und anderen Schreibweisen auch in älteren Browsern verstanden wird. Wenn

aber ein moderner Browser den modernen Code versteht, braucht er diesen zusätzlichen Code nicht.

Je nach verwendetem Tool lassen sich hier Lösungen einbauen, um Polyfills zum Beispiel nachträglich im Browser zu laden, wenn ein Browser sie benötigt.

Text bleibt während des Ladens von Webfonts sichtbar

Es gibt unterschiedliche Strategien zum Laden von Webfonts. Ein »Problem« ist die Anzeige von Text, bevor der Webfont geladen ist, wodurch der Text in einer Fallback-Schrift dargestellt wird. Ein Ansatz ist, diesen »ungestylten« Text zu verstecken, bis der Webfont geladen wird, was zu einer Seite ohne Text führt.

Das sollten Sie vermeiden und lieber nach einem oder mehreren Fallback-Fonts suchen, die beispielsweise auf macOS oder Windows installiert sind und Ihrem Font ähnlich sehen, sodass der Unterschied nicht so groß ist, wenn der Fallback zur richtigen Schrift wechselt. Und in jedem Fall sollten Sie die Webfont-Dateien optimieren, also in `woff2` ausliefern und am besten »subsetten«, sodass nur die Zeichen in der Schrift vorhanden sind, die Sie auch brauchen.

Umfangreiche Layoutverschiebung vermeiden

Layoutverschiebungen während des Ladens einer Seite entstehen zum Beispiel, wenn Sie weit oben auf der Seite ein Bild eingebunden haben, das kein `width`- und `height`-Attribut hat. Dadurch weiß der Browser nicht, welche Größe das Bild haben wird, wenn es geladen ist, und das `img`-Element nimmt keinen Platz ein. Sobald es geladen wird, ändert sich das natürlich, und der gesamte Inhalt unter dem Bild wird nach unten geschoben.

Im besten Fall ist das nur irritierend, im schlimmsten Fall könnte es ungewollte Aktionen von den Benutzer*innen zur Folge haben, wenn nämlich in dem Moment, in dem auf ein Element geklickt wird, die Layoutverschiebung passiert und dadurch ein anderes Element unter dem Finger oder dem Cursor ist und angeklickt wird. Wenn das der finale Kaufen-Button ist und die eigentliche Intention war, den Kauf doch abzubrechen, ist die Nutzer*innen-Erfahrung – vorsichtig formuliert – extrem schlecht.

Nutzung von Drittanbietern minimieren

Aus Datenschutzgründen ist die Einbindung von externen Quellen sowieso eher fragwürdig, aber auch die Ladezeit wird dadurch negativ beeinflusst. Laden Sie also alles, was geht, lokal von Ihrem Server. Und überdenken Sie bei Ressourcen, die nicht lokal eingebunden werden können, ob Sie sie wirklich brauchen. Falls ja, versuchen Sie, sie nachzuladen.

19.2.3 PageSpeed-Tests im Bereich »Barrierefreiheit«

Das Thema Barrierefreiheit ist keins, das komplett über automatisierte Tests abgedeckt werden kann. Tests wie PageSpeed Insights können Anhaltspunkte geben, aber mehr nicht. Und PageSpeed Insights führt selbst eine Liste mit Dingen auf, die manuell überprüft werden müssen, unter anderem:

- Interaktive Elemente sind mit der Tastatur fokussierbar.
- Interaktive Elemente sind als solche erkennbar, und der aktuelle Status ist erkennbar/hervorgehoben.
- Die fokussierbaren Elemente auf einer Seite sind in einer logischen Reihenfolge angeordnet, also so, wie sie auf der Seite

angezeigt werden.

- Der Fokus für Tastatur oder Screen-Reader wird nicht in einem Bereich gefangen, ohne dass es einen Ausweg gibt.
- Wenn neuer Inhalt zur Seite hinzugefügt wird, wird der User darüber benachrichtigt.
- HTML5-Landmark-Elemente wie `header`, `footer` und `main` werden genutzt, um die Navigation zu verbessern.
- Nicht sichtbarer Inhalt ist vor Hilfstechnologien wie Screen-Readern versteckt.
- Selbst entwickelte Steuerelemente wurden auf Barrierefreiheit getestet.

Kurz: Wenn Sie einen Wert von 100 im Bereich Barrierefreiheit haben, sagt das im Grunde wenig darüber aus, ob Ihre Website besonders barrierefrei ist. Es heißt nur, dass einige grundlegende Punkte eingehalten sind, die sich programmatisch überprüfen lassen.

Im Folgenden wird deshalb nur auf Tests eingegangen, bei denen es sinnvoll erscheint, dass sie automatisch getestet werden. Es gibt beispielsweise auch einen Test, ob `aria-*`-Attribute gültige Werte enthalten. Aber nur weil die Werte gültig sind, bedeutet das nicht, dass sie richtig eingesetzt wurden.

Klickbare Bereiche müssen groß genug sein und ausreichend Abstand zu anderen Elementen haben

Wenn ein klickbarer Bereich, beispielsweise von einem Link, zu klein ist oder zu dicht an einem anderen klickbaren Bereich, können etwa Nutzer*innen Schwierigkeiten haben, den Link anzuklicken, die

zwar eine Maus benutzen, aber Probleme mit der Feinmotorik haben.

Und spätestens wenn Touch-Geräte im Spiel sind, profitieren alle von einer größeren Klickfläche beziehungsweise genug Abstand, wenn sie versuchen, ein klickbares Element zwischen anderen klickbaren Elementen anzutippen.

Bilder haben einen Alternativtext

Wenn Sie Bilder einbinden, sollten diese immer ein `alt`-Attribut haben. Mit diesem Attribut beschreiben Sie, was auf dem Bild zu sehen ist, damit auch Personen mit einer Sehbehinderung den Inhalt des Bilds erfassen können. Zusätzlich wird der Text angezeigt, wenn das Bild nicht geladen werden kann.

Wenn ein Bild rein dekorativ ist, wird ein leeres `alt`-Attribut verwendet: `alt=""`. Ob die Alternativtexte sinnvoll sind, muss wieder manuell geprüft werden.

Korrekte Überschriften-Hierarchie

Die Überschriften-Elemente `h1` bis `h6` sollten nicht wild durcheinander verwendet werden. Pro Seite sollte es eine `h1` geben, und auf die `h1` dürfen `h2` folgen, aber nicht direkt `h3`. `h3` sind dafür da, einen Bereich von einer `h2` weiter zu unterteilen, und so weiter.

Sinnvolle Linktexte

Sie sind bestimmt auch schon mal über Links gestolpert, die nur aus »hier« oder »Mehr lesen« bestehen. Das ist wenig hilfreich, um zu erkennen, wo der Link hinführt, besonders nicht, wenn Sie sich wie mit einem Screen-Reader eine Liste von Links anzeigen lassen

können, also nicht den Text-Kontext vor und hinter den Links kennen.

Erstellen Sie also Ihre Links mit sinnvollen Linktexten, beispielsweise »Mehr zu unseren Leistungen«, oder verlinken Sie den Titel eines externen Artikels, auf den Sie verweisen möchten.

Farbkontraste sind ausreichend groß

Der Kontrast zwischen Vorder- und Hintergrund muss eine gewisse Stärke haben.

Die Sprache der Website wird im html-Element angegeben

Über das `lang`-Attribut kann für das `html`-Element die Sprache der Website angegeben werden, zum Beispiel `lang="de-DE"`.

Zoomen erlaubt

Im `viewport`-Meta-Tag wurde das Zoomen durch Benutzer*innen nicht deaktiviert.

19.2.4 PageSpeed-Tests im Bereich »Best Practices«

Im Bereich Best Practices werden unterschiedlichste allgemeine Aspekte geprüft, einige davon werden im Folgenden erwähnt.

HTTPS aktiv

Mit HTTPS wird die Kommunikation zwischen dem Webserver und dem Browser verschlüsselt, sodass unter anderem verhindert wird, dass andere die Kommunikation mitlesen oder verändern können.

Außerdem ist HTTPS Voraussetzung für verschiedene neue Web-APIs.

HTTPS sollte immer und auf allen Seiten aktiv sein, und wenn Sie bei einem Webhoster sind, der Geld für ein SSL-Zertifikat verlangt, sollten Sie sich nach einem anderen Hoster umschauen (kostenlose SSL-Zertifikate sind dank Let's Encrypt schon lange möglich und werden von vielen Hostern auch unterstützt).

Das Einfügen in Formularfelder erlauben

Es gibt Websites, die beispielsweise verhindern, dass Inhalt aus der Zwischenablage in ein Passwortfeld eingefügt werden kann. Solche Eingriffe in die Nutzung des Users sollten Sie unter allen Umständen vermeiden.

Beim Seitenaufbau werden keine besonderen Berechtigungen angefordert

Websites können über entsprechende APIs auf Ihren ungefähren Standort zugreifen oder Ihnen im Browser Benachrichtigungen schicken. Für beides müssen Sie die Zustimmung der Nutzer*innen einholen, und manchmal wird danach direkt nach dem Laden der Website gefragt.

Schriftgröße gut lesbar

Dieser Test prüft, ob der Text der Website mindestens 12 Pixel groß ist. 12 Pixel ist auch noch ziemlich klein, orientieren Sie sich besser an der Browser-Standard-Einstellung 16 Pixel, und gehen Sie von dort vielleicht maximal bis 14 Pixel nach unten.

Keine Fehler in der Konsole

In den Entwicklungstools des Browsers gibt es die Konsole, in der unter anderem JavaScript-Fehler angezeigt werden, wenn es welche gibt.

19.2.5 PageSpeed-Tests im Bereich »SEO«

Auch für den Bereich SEO handelt es sich um allgemeinere Tests und natürlich nicht um eine tiefgehende Analyse. Hier einige der Punkte, die PageSpeed Insights prüft:

- Seite kann indexiert werden.
- Es gibt ein `title`-Element.
- Links haben beschreibende Linktexte.
- Bilder haben `alt`-Attribute.
- Es gibt eine Meta-Beschreibung.

19.3 Optimierungs-Plugin »Autoptimize«

Das Plugin »Autoptimize« bietet unterschiedliche Funktionen, um die Auslieferung Ihrer Website zu optimieren. Dabei geht es nicht um das Caching der Dateien, sondern um das Optimieren des Ladevorgangs und der Dateigröße. Einige Optionen des Plugins sind auch in dem später vorgestellten Caching-Plugin WP Rocket enthalten. Wenn Sie aber lieber auf ein einfaches Caching-Plugin wie das im Nachgang vorgestellte Cachify setzen möchten, ist Autoptimize ein gutes zusätzliches Plugin.

Plugin	Autoptimize
Verbreitung	1.000.000+
Download	*http://r-wrk.de/wp6_ao*
Zweck	Tool zur Optimierung des JavaScript-, CSS- und HTML-Codes sowie der Auslieferung externer Ressourcen
Komplexität	++

Nach der Installation finden Sie die Optionen von Autoptimize unter Einstellungen • Autoptimize. Die Einstellungen sind in unterschiedliche Tabs aufgeteilt, die Sie gleich nacheinander kennenlernen werden.

19.3.1 JS, CSS & HTML

Im ersten Tab JS, CSS & HTML finden Sie Optionen, um die Auslieferung von JavaScript, CSS und HTML zu optimieren.

JavaScript-Optionen

In der Box JavaScript-Optionen haben Sie die folgenden Einstellungsmöglichkeiten:

- JavaScript-Code optimieren
  Wenn Sie Ihr JavaScript nicht optimieren möchten, haken Sie diese Option ab. Dann können alle anderen Optionen in der Box nicht mehr aktiviert werden. Durch Anhaken der Option wird das JavaScript minifiziert.
- JS-Dateien zusammenfügen
  Diese Option ist standardmäßig nicht aktiv. Früher war eine empfohlene Technik, möglichst wenig Dateien auszuliefern, um die Anzahl der HTTP-Requests niedrig zu halten. Das ist seit HTTP/2 nicht mehr notwendig, da die neue HTTP-Version mehr parallele Anfragen tätigen kann als noch HTTP/1.1. Und da inzwischen HTTP/2 in den meisten Fällen aktiv sein sollte, ist es besser, mehr kleinere Dateien parallel auszuliefern als weniger große.
- Nicht zusammenfügen aber verzögern
  Die einzelnen JS-Dateien werden minifiziert und verzögert geladen, sodass sie das Rendering nicht blockieren. Seit WordPress 6.4 werden Skripte von WordPress-Core und von Standard-Themes standardmäßig verzögert geladen, aber so haben Sie auch Plugins und Themes mit an Bord, die diese Strategie selbst noch nicht einsetzen.
- Inline-JS auch verzögert laden
  Mit dieser Option können Sie auch das Ausführen von Inline-Skripten verzögern.
- Folgende Skripte von Autoptimize ausschliessen
  Hier können Sie eine kommagetrennte Liste von Dateien

angeben, die nicht von Autoptimize bearbeitet werden soll. Das kann nützlich sein, wenn es durch die Optimierung durch Autoptimize zu Fehlern in einem Skript kommt.

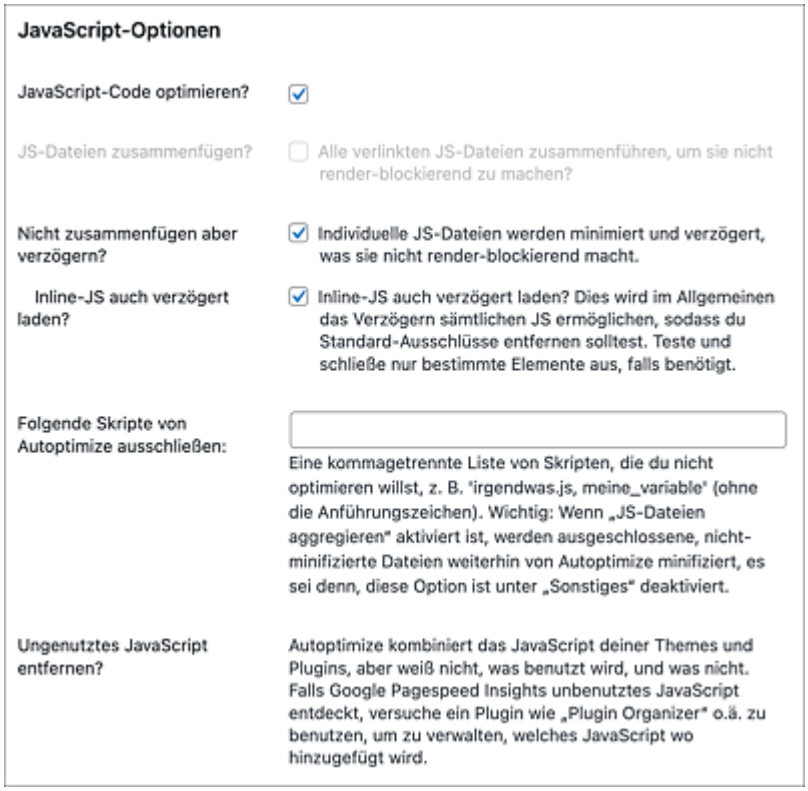


Abbildung 19.7 Die Optimierungsmöglichkeiten für JavaScript in Autoptimize

CSS-Optionen

Für CSS können Sie die folgenden Einstellungen vornehmen:

- CSS-Code optimieren
  Wie bei JavaScript geben Sie hiermit an, ob Sie CSS mit Autoptimize optimieren möchten oder nicht.
- CSS-Dateien zusammenfügen
  Ebenfalls wie bei JavaScript haben Sie die Option, die einzelnen CSS-Dateien zusammenzufassen. Das sollten Sie bei Nutzung von HTTP/2 nicht machen, da in der Regel mehrere kleine Dateien schneller parallel heruntergeladen werden können als eine große.

- **Auch Inline-CSS zusammenfügen**
  Wenn Sie die Option zum Zusammenfügen aktiviert haben, können Sie auch Inline-CSS zusammenführen.
- **Data: URIs für Bilder generieren (Inline Images)**
  Diese Option ist ebenfalls nur aktiv, wenn Sie die Option zum Zusammenführen von CSS-Dateien angehakt haben. Hierbei geht es darum, dass in CSS als externe Hintergrundbilder verlinkte Dateien inline eingebunden werden können, um die externen Requests zu sparen.
- **Render-blockierendes CSS entfernen**
  Wenn Sie diese Option anhaken, erscheint ein Feld, in das Sie CSS-Regeln für die Elemente einfügen können, die direkt beim Laden der Website sichtbar sind. Dieses CSS wird dann inline eingefügt und das restliche externe CSS nach dem Laden der Website nachgeladen. So kann der Seitenaufbau beschleunigt werden, allerdings ist es auch nicht gut, wenn das inline eingebundene CSS zu groß ist, da dadurch auf jeder Seite die Größe des HTML-Dokuments vergrößert wird und das, anders als die externen CSS-Dateien, nicht gecacht werden kann. Um das CSS für diese Option zu generieren, gibt es unterschiedliche Tools, mit »CriticalCSS« ist auch eins davon direkt in Autoptimize integriert.
- **Gesamten CSS-Code inline einfügen**
  Das sollten Sie nicht tun, außer das gesamte CSS Ihrer Website besteht nur aus 10 bis 20 Regeln.
- **Folgende CSS-Dateien von Autoptimize ausschliessen**
  Hier können Sie wie bei den JavaScript-Einstellungen Dateien angeben, die nicht von Autoptimize optimiert werden sollen.

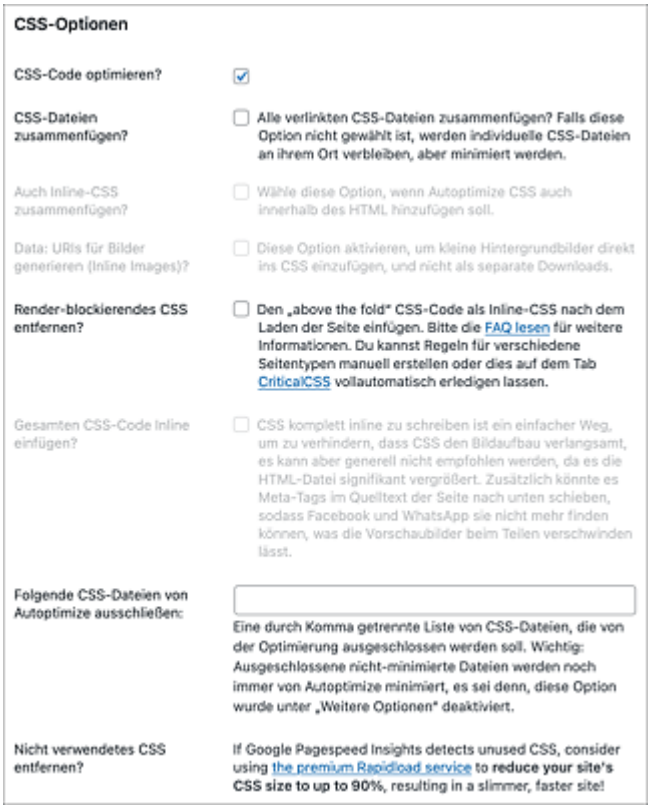


Abbildung 19.8 Die Optimierungsmöglichkeiten für CSS in Autoptimize

HTML-Optionen

Neben dem CSS und JavaScript kann natürlich auch das HTML minifiziert werden. Hier gibt es nur einige wenige Optionen.

- HTML-Code optimieren
  ob das HTML von Autoptimize minifiziert werden soll
- Auch Inline-JS/CSS minimieren
  Hier können Sie einstellen, ob inline eingebundener JavaScript- und CSS-Code ebenfalls minifiziert werden soll.
- HTML-Kommentare beibehalten
  Wenn Sie möchten, können HTML-Kommentare in der minifizierten Version behalten werden.

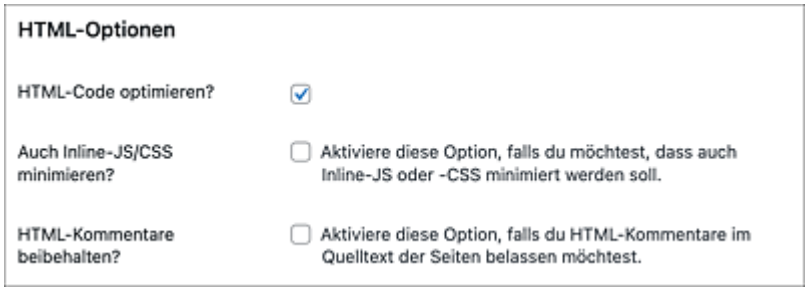


Abbildung 19.9 Auch das HTML kann minifiziert werden.

CDN-Optionen

Sie können optional eine CDN-Basis-URL angeben, um die Autoptimize-Dateien über ein CDN zu laden, also zum Beispiel *files.example.com*. Können Sie in der Regel ignorieren.

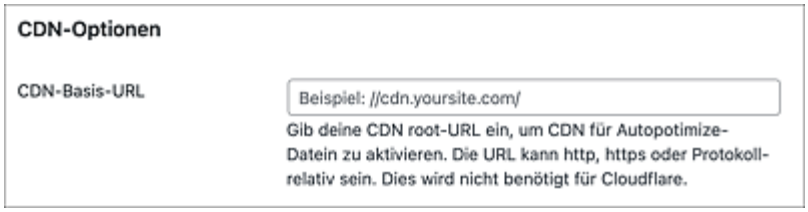


Abbildung 19.10 Wenn Sie ein CDN nutzen, können Sie auch die Autoptimize-Dateien darüber einbinden.

Cache-Informationen

Hier werden Ihnen ein paar Informationen zum Autoptimize-Cache angezeigt:

- Cache-Ordner zeigt den Ordnerpfad an, in dem die gecachten Dateien von Autoptimize gespeichert werden.
- Schreibberechtigung gibt an, ob die Ordnerberechtigungen passen, sodass das Plugin in das Verzeichnis schreiben kann.
- Styles und Skripte im Cache zeigt die Anzahl und Größe der Cache-Dateien an.



Abbildung 19.11 Die Cache-Informationen zeigen Ihnen ein paar interessante Informationen zum Autoptimize-Cache an.

Weitere Optionen

Hier haben Sie ein paar unterschiedliche Optionen, die normalerweise so bleiben können, wie sie von Autoptimize vorgesehen sind.

- Zusammengefügte CSS-/Skript-Dateien als statische Dateien speichern

  Diese Option ist standardmäßig aktiv. Wenn der Webserver Probleme bei der Auslieferung der Dateien hat, kann die Option deaktiviert werden.

- 404-Fallbacks aktivieren

  Wenn in einer gecachten Seite noch Verweise auf inzwischen ungültige Autoptimize-CSS- und -JS-URLs vorhanden sind, kann Autoptimize versuchen, Fallbacks einzubinden, damit die Website weiterfunktioniert.

- Auch für angemeldete Redakteure/Administratoren optimieren

  Standardmäßig ist Autoptimize auch für angemeldete Benutzer*innen aktiv. Wenn es dadurch Probleme gibt, können Sie die versuchen, die Option zu deaktivieren.

- Konfiguration pro Beitrag/Seite aktivieren

  Autoptimize zeigt standardmäßig bei Beiträgen und Seiten in der Seitenleiste ein Panel an, über das Sie die Einstellungen für

einzelne Einträge anpassen können (siehe Abbildung 19.12).

- Extra-Logik für Kompatibilität deaktivieren
  Autoptimize versucht direkt, Probleme bei der Optimierung von JavaScript zu verhindern. Sie können versuchen, ob diese Extra-Logik deaktiviert werden kann. Testen Sie dann aber genau, ob noch alles funktioniert.

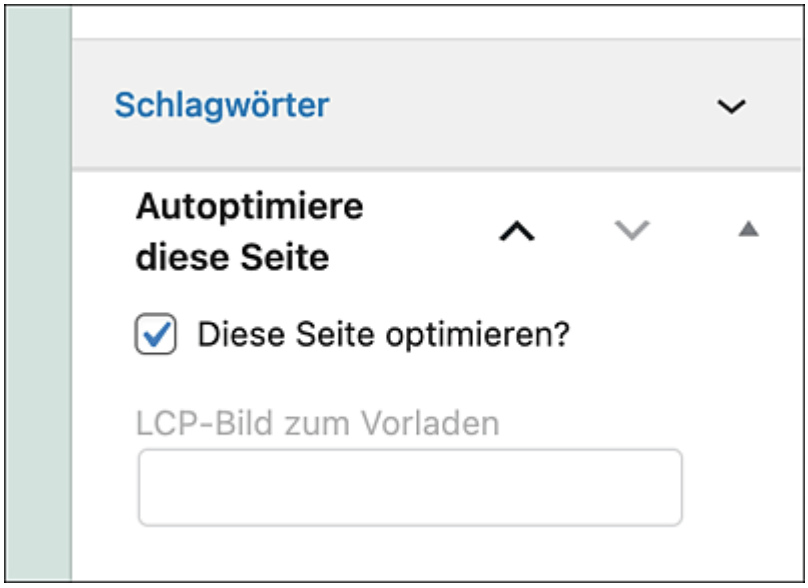


Abbildung 19.12 Sie können die Autoptimize-Einstellungen für einzelne Seiten oder Beiträge anpassen.

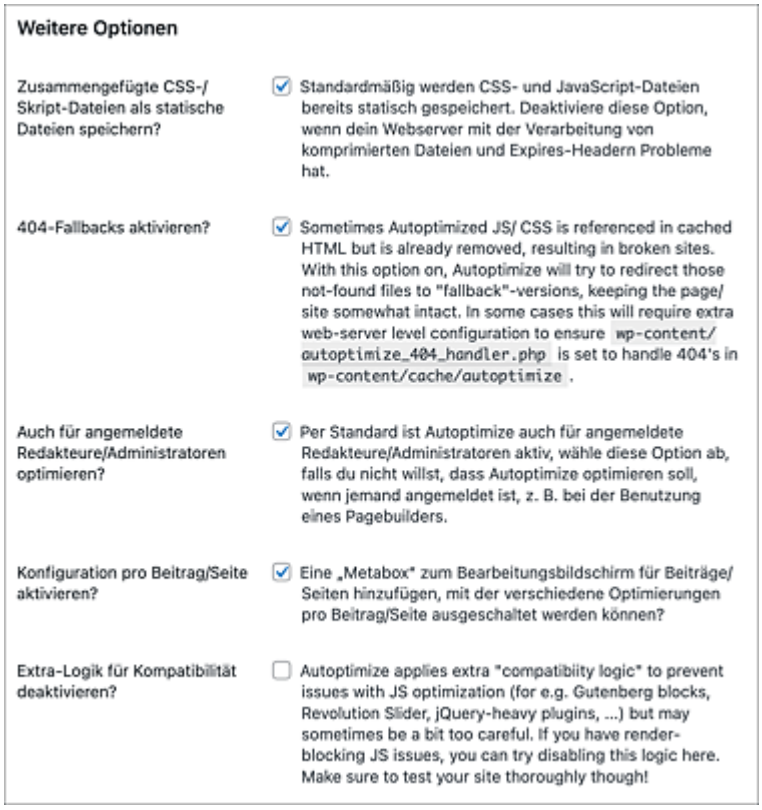


Abbildung 19.13 In der Box »Weitere Optionen« sind unterschiedlichste Optionen zusammengefasst, die in der Regel unangetastet bleiben können.

19.3.2 Bilder

Hier haben Sie die Option, den Dienst ShortPixel zur Bildoptimierung zu nutzen und Bilder über das ShortPixel-CDN auszuliefern. Aus Datenschutzgründen sollten Sie sich genau überlegen, ob Sie das tun möchten. Nutzen Sie für die Bildoptimierung lieber ein dediziertes Plugin, bei dem die Bilder dann letztlich wieder lokal von Ihrem Server ausgeliefert werden.

Darüber hinaus können Sie Bilder verzögert laden, was WordPress inzwischen aber standardmäßig selbst macht. Wenn Sie also nicht gerade aus irgendeinem Grund mehr Optionen brauchen, etwa um bestimmte Bilder mit einer Klasse vom Lazy Loading

auszuschließen, sollten Sie die Option wahrscheinlich nicht benötigen.

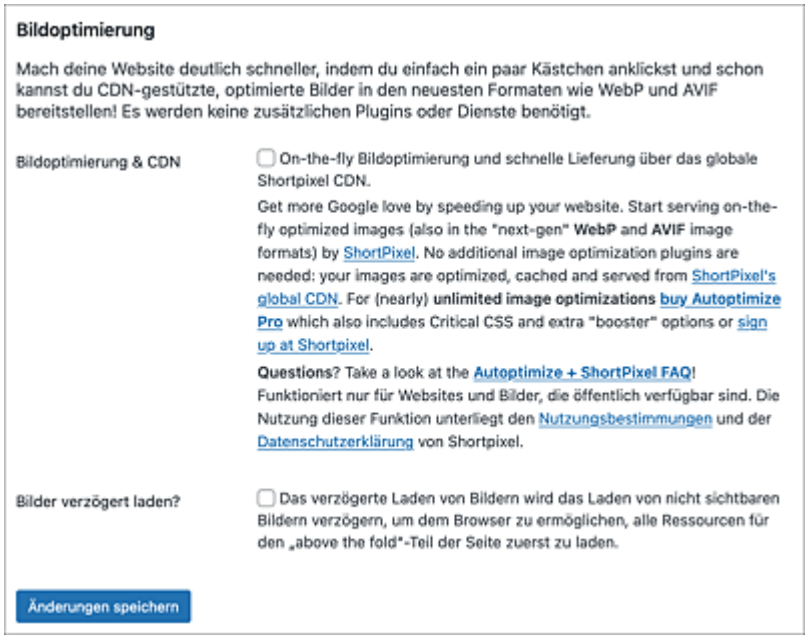


Abbildung 19.14 Den Tab mit den Bilder-Optionen können Sie wahrscheinlich einfach überspringen.

19.3.3 Kritisches CSS

Dieser Tab hängt zusammen mit der Option im Tab JS, CSS & HTML, bei dem Sie Render-blockierendes CSS inline einfügen konnten, indem Sie das CSS zur Darstellung der direkt sichtbaren Elemente angeben. Hier können Sie die CSS-Regeln bearbeiten und neue Regeln hinzufügen, die bei Bedarf auch nur für bestimmte Templates oder Seiten gelten. Zusätzlich können Sie das Plugin optional mit dem Dienst »criticalcss.com« verbinden, der sich um die Generierung des CSS kümmert.

Wie vorhin schon angedeutet, ist diese Art der Optimierung ein schmaler Grat und nicht gerade einfach. Sie müssen aufpassen, dass das inline eingebundene CSS nicht zu groß wird, sodass der Effekt ins Negative geht. Konzentrieren Sie sich lieber darauf, dass auf

Ihrer Website nur das CSS geladen wird, das benötigt wird, und nicht für viele Optionen und Elemente, die Sie sowieso nicht nutzen.

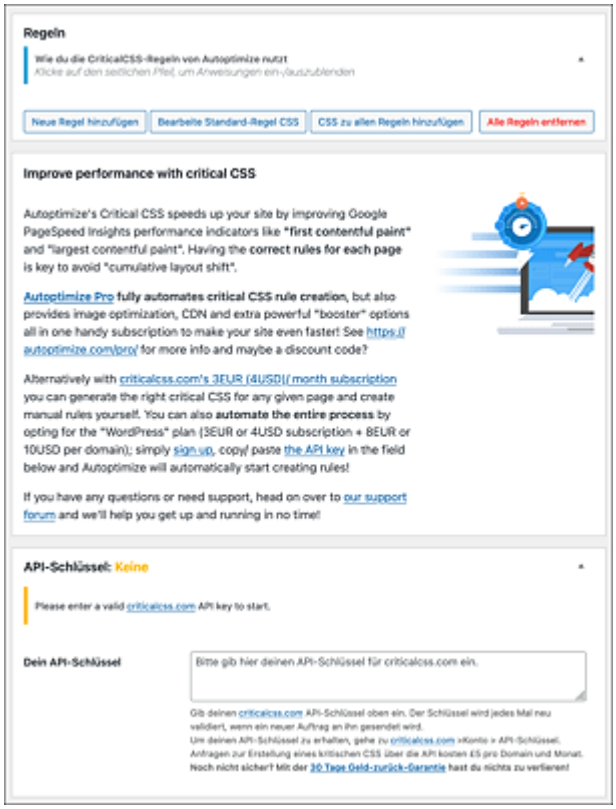


Abbildung 19.15 Auch den Tab für kritisches CSS können Sie in der Regel links liegen lassen. Diese Optimierung ist nicht gerade einfach und kann sich auch negativ auf die Website-Performance auswirken.

19.3.4 Extras

Hier finden Sie unterschiedliche Einstellungsmöglichkeiten, die die Performance verbessern können.

- Google Fonts
  Diese Einstellungen sollten Sie überspringen können, da Google Fonts aus Datenschutzgründen nur lokal eingebunden und nicht von Googles Servern geladen werden sollten.
- Emojis entfernen
  WordPress bindet Code ein, damit Emojis auch in Browsern

angezeigt werden, die die Bildzeichen nicht unterstützen. Da alte Browser ohne Emoji-Support immer seltener genutzt werden, können Sie diese Styles und Skripte entfernen.

- **Abfragezeichenfolgen von statischen Ressourcen entfernen**
  Sie können Query-Strings von CSS- und JavaScript-URLs entfernen (WordPress fügt standardmäßig die WordPress-Versionsnummer hinzu, wenn keine andere explizite Angabe von einem Plugin oder Theme gemacht wird). Das Entfernen der Strings führt zwar nicht zu einer besseren Ladezeit, kann gegebenenfalls aber Performance-Scores verbessern.
- **WordPress-Block-CSS entfernen**
  WordPress fügt Standard-Stile für seine Blöcke ein. Wenn Sie diese Styles nicht brauchen, können Sie die Einbindung hier deaktivieren.
- **Zu Dritt-Domains vorverbinden**
  Wenn Ihre Website Verbindungen zu Drittanbietern aufbauen muss, können Sie hier die entsprechenden Domains angeben, sodass der Verbindungsaufbau schneller stattfindet.
- **Spezifische Anfragen im Voraus laden**
  Wenn Sie URLs zu CSS- und JavaScript-Dateien haben, die Sie besonders schnell laden möchten, können Sie diese hier angeben. Wichtig: Geben Sie hier nur URLs an, die wirklich wichtig sind. Wenn Sie zu viele Daten über diese Methode laden, wirkt sich das negativ auf die Website-Performance aus.
- **Async Javascript-files**
  Hier können Sie JavaScript-Dateien angeben, die mit dem `async`-Attribut ausgestattet und damit asynchron geladen werden sollen.

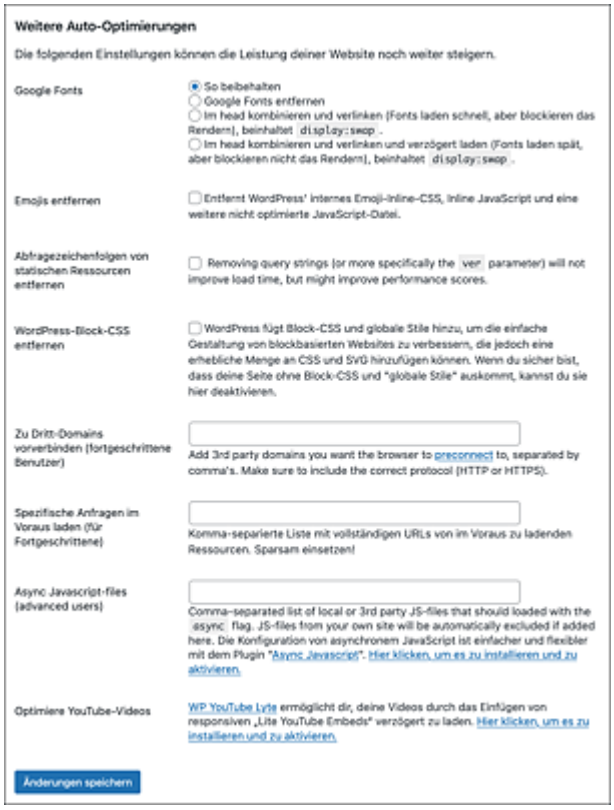


Abbildung 19.16 Im Tab »Extras« finden Sie unterschiedlichste Einstellungsmöglichkeiten, die die Performance verbessern können.

19.3.5 »Mehr optimieren« und »Seiten-Cache«

Auf den verbleibenden zwei Tabs verweist Autoptimize auf andere Plugins, die Sie noch ausprobieren könnten, sowie auf die Pro-Version.

19.4 Einfaches Caching-Plugin ohne Schnickschnack – »Cachify«

Cachify ist ein sehr übersichtliches Caching-Plugin, das nicht viele Optionen und Funktionen mitbringt, sondern sich wirklich auf das Caching der Website konzentriert. Wenn Sie das Plugin aktiviert haben, haben Sie bereits den Cache mit sinnvollen Standard-Werten aktiv.

Plugin	Cachify
Verbreitung	10.000+
Download	*https://de.wordpress.org/plugins/cachify/*
Zweck	ein einfaches Caching-Plugin mit wenigen Optionen – aktivieren: fertig
Komplexität	+

Die Einstellungen von Cachify finden Sie unter Einstellungen • Cachify (siehe Abbildung 19.17). Folgende Dinge können Sie anpassen:

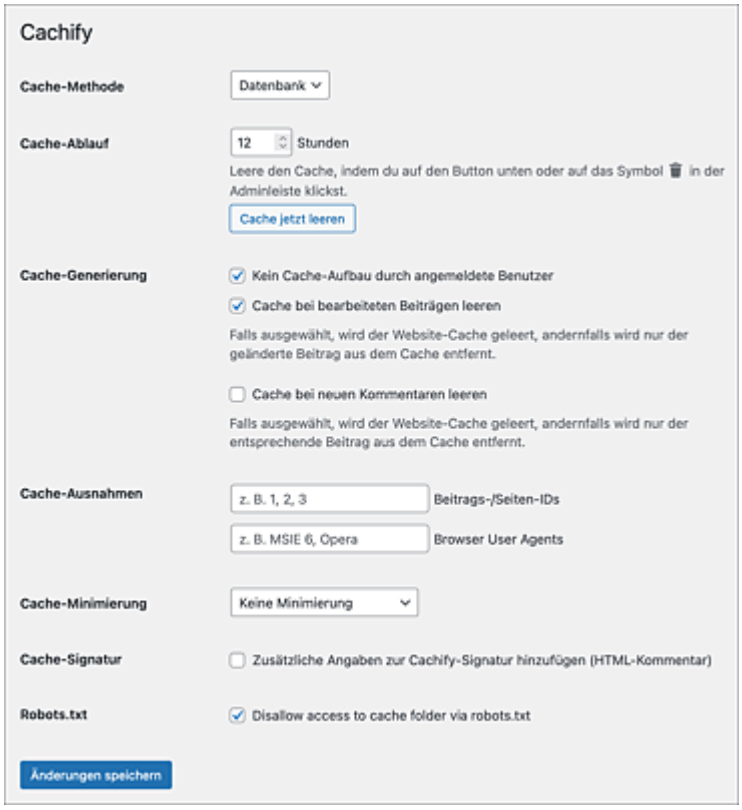


Abbildung 19.17 Die Cachify-Einstellungen sind verglichen mit anderen Caching-Plugins sehr übersichtlich.

- Cache-Methode
  Hier haben Sie die Auswahl an unterschiedlichen Methoden der Speicherung des Caches. Wie viele Optionen hier auswählbar sind, hängt von Ihrem System ab, mindestens sollten Datenbank und Festplatte vorhanden sein, mit Datenbank als Standardwert. Bei der Datenbank-Methode wird der Cache in der Datenbank gespeichert und von WordPress ausgelesen.

  Bei dem Wert Festplatte wird von Cachify eine statische Version der Inhalte als HTML-Dateien auf dem Server abgelegt. Wenn Sie diese Option ausgewählt haben, müssen Sie Ihre Server-Konfiguration (*.htaccess* oder nginx-Konfiguration) anpassen, damit diese Dateien direkt vom Webserver ausgeliefert werden. So haben Sie den Vorteil, dass WordPress gar nicht loslaufen muss, wenn eine Datei schon gecacht ist, was einen deutlichen

Geschwindigkeitsvorteil bringt. Wie Sie die Konfiguration anpassen können, wird in einem Extra-Tab angezeigt, wenn die Option »Festplatte« ausgewählt ist (siehe Abbildung 19.18).



Abbildung 19.18 Wenn die Cache-Methode auf »Festplatte« gestellt ist, müssen Sie die Webserver-Einstellung anpassen, um alle Vorteile davon nutzen zu können.

- Cache-Ablauf
  Steuert, wann der Cache automatisch abläuft. Außerdem können Sie über den Button Cache jetzt leeren den Cache löschen.
- Cache-Generierung
  Hier können Sie anpassen, wann der Cache generiert und wann der gesamte Cache geleert wird. Standardmäßig wird durch angemeldete Benutzer*innen kein Cache aufgebaut.

  Sie können einstellen, ob der gesamte Cache beim Bearbeiten von Beiträgen geleert wird, oder, wenn die Option nicht angehakt ist,

nur der Cache von dem entsprechenden Beitrag. Gleiches können Sie einstellen, wenn neue Kommentare veröffentlicht werden.

- **Cache-Ausnahmen**
  Hier können Sie IDs von Inhalten angeben, die nicht gecacht werden sollen. Außerdem können Sie das Caching für bestimmte Browser-User-Agents deaktivieren.
- **Cache-Minimierung**
  Ermöglicht, das gecachte HTML und optional auch Inline JavaScript zu minimieren.
- **Cache-Signatur**
  Cachify fügt am Ende des gecachten HTML einen HTML-Kommentar ein, der anzeigt, welches Tool für das Caching verantwortlich ist. Hier können Sie als zusätzliche Info in den Kommentar den Zeitpunkt der Cache-Erstellung einfügen.

19.5 Profi-Cache mit WP Rocket

Das Plugin WP Rocket gibt es nur als Bezahlversion. Es bietet deutlich mehr Optionen als Cachify, da es neben dem Caching auch Funktionen für die generelle Optimierung der Auslieferung einer Website mitbringt und damit unter anderem Überschneidungen mit Autoptimize hat. Wenn Sie WP Rocket nutzen, sollten Sie genau schauen, ob Sie ein anderes Optimierungs-Plugin brauchen, damit keine Probleme entstehen, wenn sich die Plugins ins Gehege kommen.

Plugin	WP Rocket
Download	*https://wp-rocket.me/*
Zweck	Caching- und Performance-Optimierungs-Plugin mit umfangreichen Optionen zur Beschleunigung Ihrer Website und einer Funktion zum Vorladen des Caches
Komplexität	++

Nach der Installation von WP Rocket finden Sie die Einstellungsmöglichkeiten des Plugins über das linke Admin-Menü Einstellungen • WP Rocket. Über die Adminbar haben Sie die Möglichkeit, schnell auf Funktionen wie das Leeren oder Vorladen des Caches zuzugreifen.

Die Einstellungen sind in mehrere Tabs unterteilt, deren Möglichkeiten im Folgenden beschrieben werden.

19.5.1 Dashboard

Im Dashboard sehen Sie den Status Ihrer Lizenz für WP Rocket und ob Sie ein Abo für das RocketCDN haben. Außerdem finden Sie im Dashboard Links zum Leeren und Vorladen des Website-Caches und zum Löschen des Caches für Vorrangige Elemente.

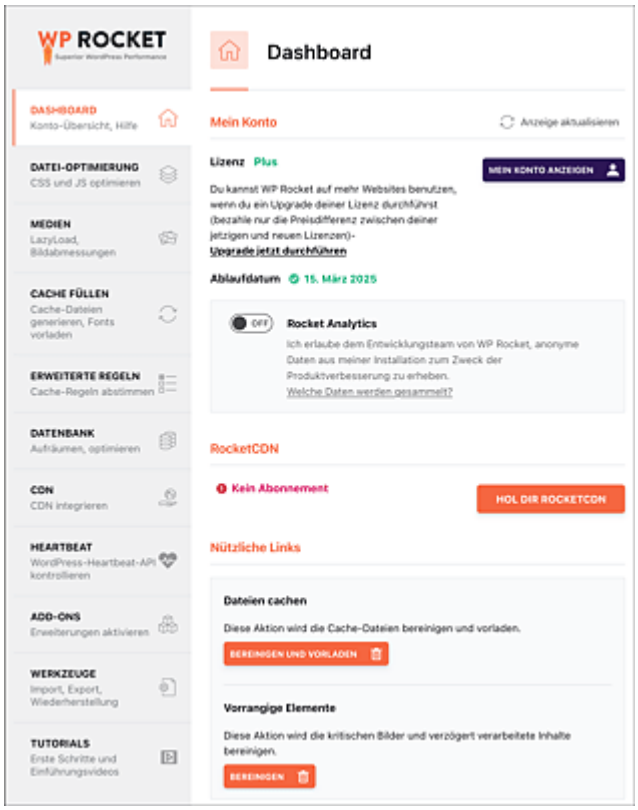


Abbildung 19.19 Das Dashboard zeigt Informationen zu Ihrer WP-Rocket-Lizenz und bietet rechts Buttons zum Löschen vom Cache.

»Vorrangige Elemente« sind Bilder, die direkt beim Laden der Website im sichtbaren Bereich sind und deshalb vom Lazy Loading ausgeschlossen werden und mit Priorität geladen werden, um positiven Einfluss auf den »Largest Contentful Paint«-Wert zu nehmen. Außerdem gibt es die Funktion »Lazy Render Content«, die ebenfalls unter die »vorrangigen Elemente« fällt. Diese Funktion verzögert beim Laden der Seite das Rendering von Elementen, die außerhalb des sichtbaren Bereichs sind, damit der Seitenaufbau schneller abläuft.

19.5.2 Datei-Optimierung

Im Datei-Optimierung-Tab können Sie die Auslieferung von CSS und JavaScript optimieren.

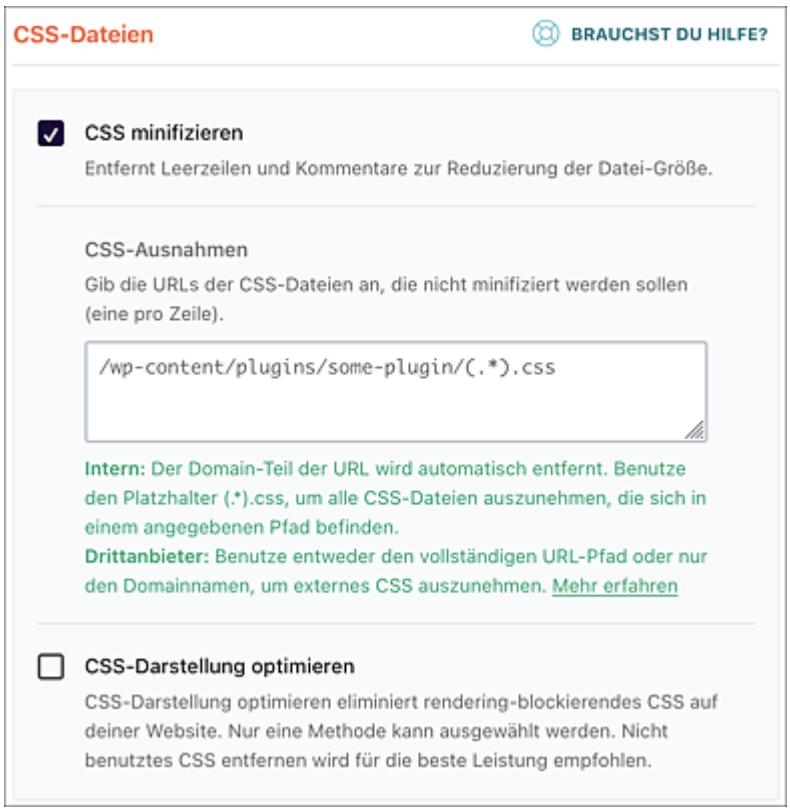


Abbildung 19.20 Die Optionen von WP Rocket für CSS-Dateien

CSS-Dateien

Für CSS-Dateien können Sie die folgenden Optionen nutzen:

- CSS minifizieren
  Diese Option ist standardmäßig aktiv und ist dafür da, dass die CSS-Dateien minifiziert werden. Über CSS-Ausnahmen können Sie URLs von CSS-Dateien angeben, die nicht minifiziert werden sollen.
- CSS-Darstellung optimieren
  Hier haben Sie die Auswahl aus zwei Optionen: Remove unused CSS

und CSS-Dateien asynchron laden.

- Unbenutztes CSS entfernen ist die empfohlene Einstellung und funktioniert so, dass WP Rocket einen Dienst bei Ihren Seiten vorbeischickt, der analysiert, welches CSS nicht gebraucht wird und aus der gecachten Version entfernt werden kann. Wichtig: Damit wird natürlich nicht das CSS vom Theme oder Plugins direkt verändert. Wenn die Option deaktiviert wird, ist wieder alles so wie vorher.
- CSS-Dateien asynchron laden generiert CSS für die Elemente, die direkt beim Laden sichtbar sind, und lädt das restliche CSS asynchron.

JavaScript-Dateien

Für JavaScript bringt WP Rocket die folgenden Optionen mit:

- JavaScript minifizieren
  Minifiziert den Inhalt von JavaScript-Dateien.

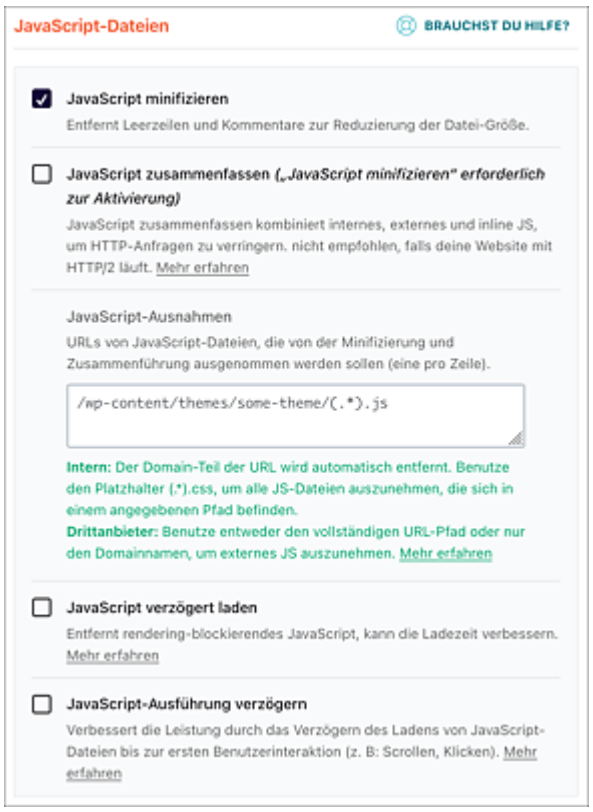


Abbildung 19.21 Die Möglichkeiten zur Optimierung von JavaScript durch WP Rocket

- JavaScript zusammenfassen
  Fasst mehrere JavaScript-Dateien zusammen, um die Anzahl der Anfragen zu verringern. Wenn Ihr Server HTTP/2 verwendet, was inzwischen eigentlich der Standard sein sollte, ist das nicht empfohlen, da mit HTTP/2 mehr Dateien parallel übertragen werden können als mit HTTP/1.1, wodurch mehrere kleine Dateien schneller übertragen werden können als weniger große.
- JavaScript-Ausnahmen
  Hier können Sie URLs zu Dateien angeben, die von der Minifizierung und Zusammenfassung ausgeschlossen werden sollen.
- JavaScript verzögert laden
  Durch die Option wird JavaScript erst geladen, wenn das

Dokument fertig geparst ist. Dadurch wird verhindert, dass das Laden und Ausführen von JavaScript das Rendering blockiert.

- JavaScript-Ausführung verzögern
  Mit dieser Option werden die JavaScript-Dateien erst geladen, wenn die erste Interaktion des*der Benutzer*in stattfindet, etwa durch Bewegung der Maus. Wenn die Option angehakt ist, können Ausnahmen definiert werden.

19.5.3 Medien

Im Medien-Tab können Sie Lazy Loading für (Hintergrund-)Bilder, iFrames und Videos aktivieren und Ausnahmen definieren.

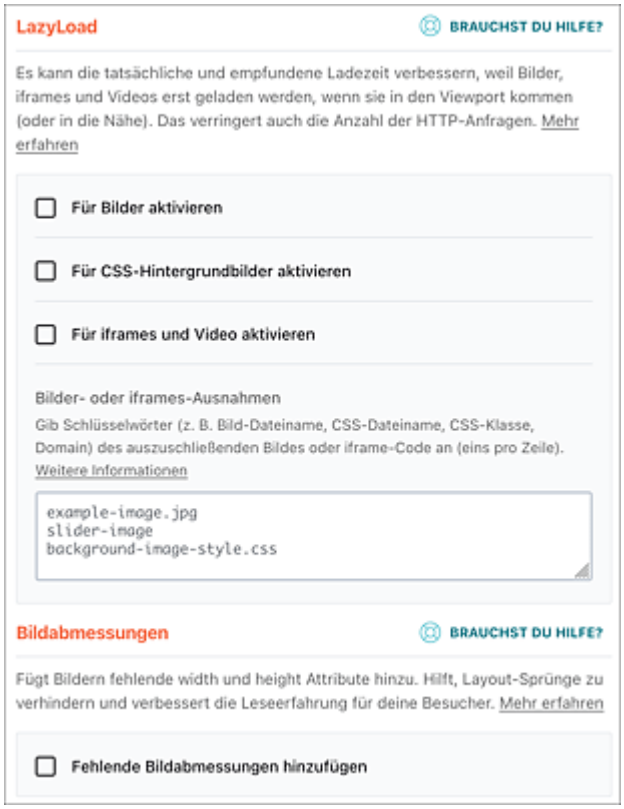


Abbildung 19.22 Die Optionen für Medien in WP Rocket beinhalten Lazy Loading und Nachrüsten von »width« und »height« für Bilder.

Wenn Sie keine Ausnahmen brauchen, reicht vielleicht auch das Standard-Lazy-Loading durch den Browser für Bilder und iFrames, das von WordPress aktiviert wird.

Darüber hinaus können Sie bei Bildern ohne `width`- und `height`-Attribute diese Attribute nachrüsten lassen. Dadurch können Sie vermeiden, dass das Layout springt, wenn Bilder geladen werden.

19.5.4 Cache füllen

Eine sehr schöne Funktion von WP Rocket ist das Vorladen des Caches. Das bedeutet, dass nach dem Leeren des Caches die Seiten automatisch angefragt werden, damit sie schon gecacht sind, wenn die erste Person die Seite aufruft. Diese Funktion können Sie mit VORLADEN AKTIVIEREN einschalten und bei Bedarf URLs davon ausschließen.



Abbildung 19.23 WP Rocket ermöglicht das Vorladen des Caches, sodass Besucher*innen auch nach dem Leeren des Caches eine schon gecachte Version aufrufen.

Weitere Optionen sind:

- Links vorladen, was die wahrgenommene Ladezeit verkürzt, da eine Seite schon geladen wird, wenn der Cursor über einem Link ist.
- DNS-Prefetch ermöglicht Ihnen die Angabe externer Domains, die von Ihrer Website angefragt werden. Durch das Prefetch funktioniert der Verbindungsaufbau schneller.
- Fonts vorladen lässt Sie Pfade zu Schriften angeben, damit diese vorgeladen werden und so schneller angezeigt werden können.

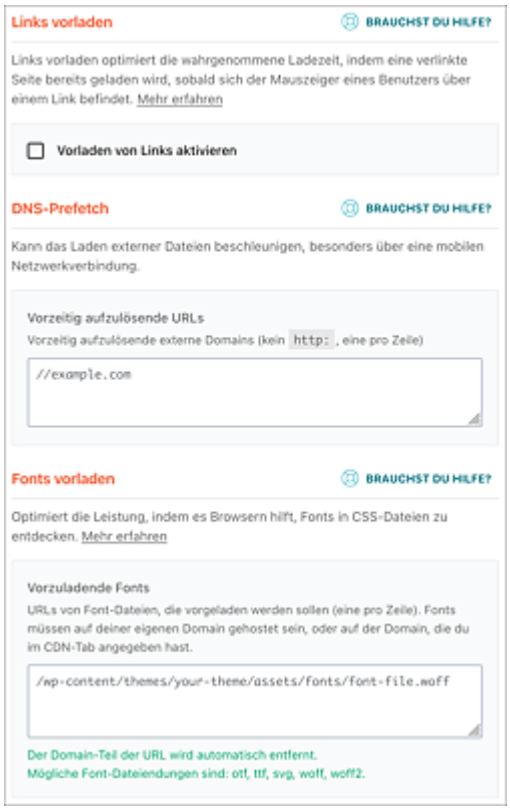


Abbildung 19.24 Die weiteren Optionen im Tab »Cache füllen« sind beispielsweise das Vorladen von Inhalten anderer Seiten, wenn sich der Cursor über einem Link zu dem Inhalt befindet.

19.5.5 Erweiterte Regeln

Im Tab Erweiterte Regeln können Sie weitere Einstellungen für den Cache vornehmen:

- Cache-Dauer
  Gibt an, nach wie vielen Stunden der Cache automatisch neu generiert werden soll.
- Nie cachen (URLs)
  Hier können Sie URLs angeben, die nie gecacht werden sollen.
- Nie cachen (Cookies)
  Wenn Cookies bei den Benutzer*innen gesetzt sind, die hier angegeben wurden, wird das Caching verhindert.
- Nie Cachen (User Agents)
  Hier können Sie das Caching für bestimmte User Agents deaktivieren.
- Immer löschen (URLs)
  Hier haben Sie die Möglichkeit, URLs anzugeben, bei denen der Cache immer gelöscht wird, wenn Beiträge oder Seiten aktualisiert werden.
- Cache für Query Strings
  Normalerweise werden Seiten mit Query-Strings von WP Rocket nicht gecacht. Hier können Sie Query-Strings angeben, für die ein Cache erstellt werden soll.



Abbildung 19.25 In den »Erweiterten Regeln« können Sie Sonderregeln fürs Caching einstellen.

19.5.6 Datenbank

Hier gibt WP Rocket Ihnen unterschiedliche Optionen zur Datenbank-Optimierung. Machen Sie zur Sicherheit ein Datenbank-Backup, bevor Sie die Optimierung ausführen (wie Sie ein Backup anlegen können, lernen Sie in Abschnitt 16.3).

- Beiträge/Seiten bereinigen
  Hier können Sie Revisionen, Automatisch erstellte Entwürfe und Gelöschte Einträge aus der Datenbank entfernen. Beim Entfernen der Revisionen sollten Sie vorsichtig sein, da Sie dann keinen Zugriff mehr auf ältere Stände Ihrer Beiträge und Seiten haben.
- Kommentare bereinigen
  Sie können auswählen, ob Sie Spam-Kommentare und/oder

Gelöschte Kommentare entfernen möchten. Antispam-Plugins wie Antispam Bee nutzen die als Spam markierten Kommentare als lokale Vergleichsdatenbank für eingehende Kommentare, wenn Sie also dieses oder ein anderes Plugin mit so einer Funktion nutzen, sollten Sie die Spam-Kommentare nicht einfach entfernen.

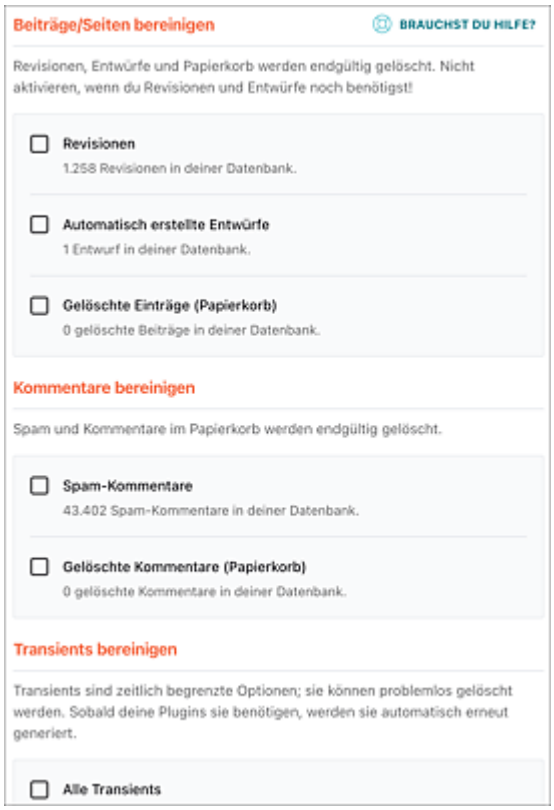


Abbildung 19.26 WP Rocket bietet unterschiedliche Aktionen an, um die Größe der Datenbank zu verringern. Hier sollten Sie aber nicht einfach alles anhaken und durchführen, da Dinge wie Revisionen und Spam-Kommentare hilfreich sein können.

- Transients bereinigen

  Das Löschen von Transients sollte keine negativen Auswirkungen haben. In Transients werden von Plugins und dem Core temporär Daten gespeichert, und wenn sie nicht vorhanden sind, werden sie neu erstellt.

- Datenbank bereinigen
  Die Option Tabellen optimieren führt das MySQL-Statement `OPTIMIZE TABLE` durch, mit dem die physische Speicherung der Daten optimiert wird, um den benötigten Speicherplatz zu verringern und Lese- und Schreibzugriffe effizienter zu machen.
- Automatische Bereinigung
  Hier können Sie einstellen, ob die eingestellten Aktionen regelmäßig ausgeführt werden sollen.

19.5.7 CDN

Hier können Sie das WP-Rocket-eigene CDN RocketCDN einrichten oder die URLs für statische Assets wie CSS und JavaScript auf eine beliebige andere CDN-URL ändern. Für die meisten Websites ist die Nutzung eines CDNs nicht wirklich sinnvoll.

19.5.8 Heartbeat

Die Heartbeat-API ist eine API von WordPress, mit der durch regelmäßige Anfragen (alle 15 bis 120 Sekunden) an einen AJAX-Endpunkt nahezu Echtzeit-Updates für Plugins umgesetzt werden können.

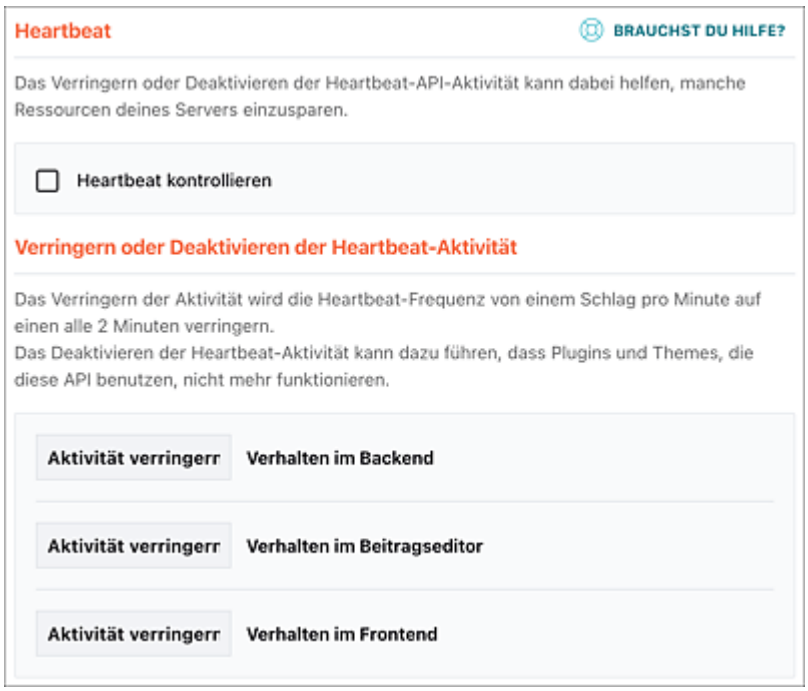


Abbildung 19.27 WP Rocket kann das Intervall der Heartbeat-API verringern, um die Serverlast zu reduzieren.

Das Intervall des »Herzschlags« kann von WP Rocket verringert werden, um die Serverlast zu reduzieren. Wenn Sie Plugins einsetzen, die die API nutzen, müssen Sie testen, ob sie danach noch funktionieren.

19.5.9 Add-ons

WP Rocket kommt mit ein paar Add-ons, die Sie direkt aktivieren und ohne weitere Konfiguration nutzen können. So kann beispielsweise für jede*n angemeldete Benutzer*in ein eigener Cache angelegt werden.

Darüber hinaus gibt es ein paar weitere Add-ons, etwa für die Nutzung mit Cloudflare, die eine extra Konfiguration benötigen.

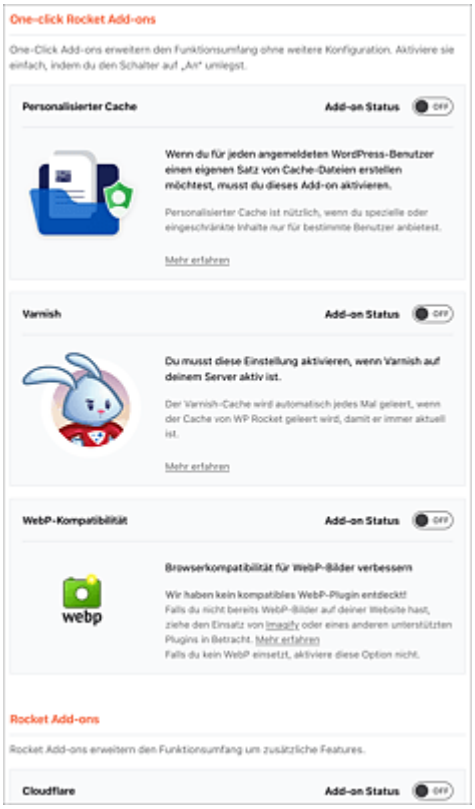


Abbildung 19.28 Im Tab »Add-ons« finden Sie ein paar Add-ons, die Sie ohne zusätzliche Konfiguration direkt aktivieren können, und ein paar, die konfiguriert werden müssen.

19.5.10 Werkzeuge

Hier können Sie WP-Rocket-Einstellungen exportieren und importieren und zur vorherigen Version von WP Rocket zurückkehren, wenn Sie mit dem neuesten Update ein Problem feststellen. Außerdem können Sie die Inklusions- und Exklusions-Listen aktualisieren, die zentral von WP Rocket gepflegt und einmal wöchentlich von Ihrer Installation aktualisiert werden. In den Listen sind Informationen für unterschiedliche Features von WP Rocket, um die Kompatibilität der Features zu verbessern.

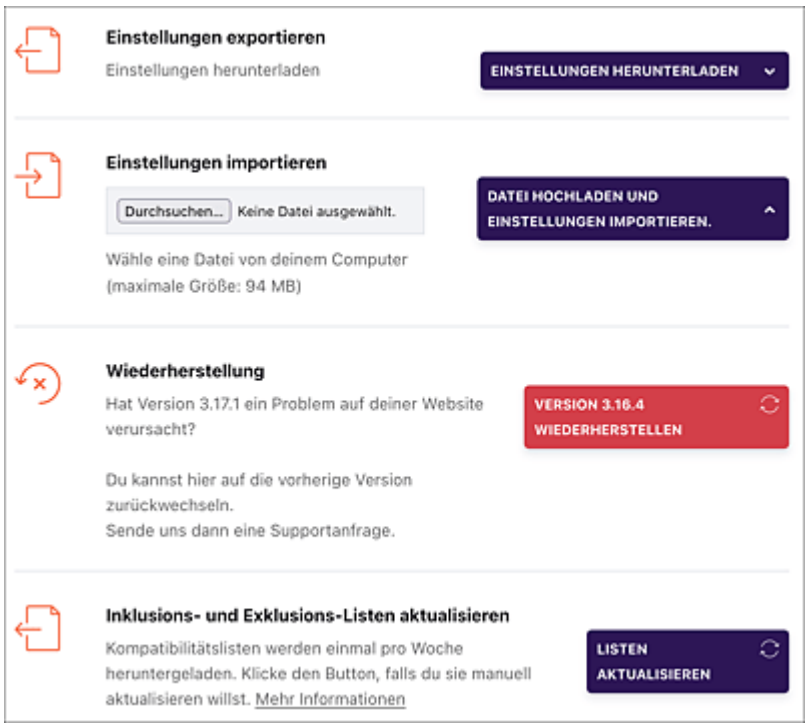


Abbildung 19.29 Sie können Ihre Einstellungen von WP Rocket exportieren, um sie auf einer anderen Instanz wieder zu importieren.

19.6 Bilder optimieren

Je nach Website machen Bilder einen sehr großen Teil der Daten aus, die an Besucher*innen übertragen werden, und gleichzeitig gibt es bei Bildern ein großes Optimierungspotenzial. Die vielleicht offensichtlichste Maßnahme ist, Bilder möglichst in der passenden Größe auszuspielen (darum kümmert sich WordPress dank Markup für *Responsive Images* halbwegs automatisch) und beispielsweise für Fotos keine PNG-Dateien, sondern JPGs zu verwenden. Doch auch wenn Bilder in der halbwegs richtigen Größe ausgeliefert werden, bleibt noch einiges an Potenzial bei der Komprimierung.

Info: Responsive Images

Vielleicht denken Sie bei dem Begriff Responsive Images zuerst daran, dass sich Bilder mit einer flexiblen `max-width` und der Angabe `height: auto` an den ihnen zur Verfügung stehenden Platz in einem responsiven Website-Layout anpassen.

Es gibt allerdings für das `img`-Element, das zur Darstellung von Bildern genutzt wird, ein paar Attribute, mit denen dem Browser mehrere Bilddateien angeboten werden können, sodass er sich die am besten passende aussucht. So kann sich ein Desktop-Browser mit einem Viewport von 2000 Pixel Breite die Originalgröße des Bildes laden, und auf einem Smartphone würde eine der kleinen Bildgrößen geladen, die WordPress beim Hochladen eines Bildes erstellt. Die entscheidenden Attribute sind `srcset` und `sizes`. In `srcset` können mehrere Bilddateien verlinkt werden, in `sizes` wird angegeben, wie groß das Bild auf der Website dargestellt wird.

Ein Beispiel könnte so aussehen:

```
<img srcset="kleines-bild.jpg 480w, grosses-bild.jpg 900w"
  sizes="(max-width: 900px) 100vw, 900px"
  src="grosses-bild.jpg"
  alt="Ein tolles Bild">
```

In dem `srcset`-Attribut werden die URLs oder Pfade zu den Bildern aufgelistet und mit Kommata getrennt. Hinter jeder URL oder einem Pfad steht die Breite des Bildes gefolgt von einem `w`; `kleines-bild.jpg 480w` bedeutet also, dass das kleine Bild 480 Pixel breit ist. In `sizes` wird angegeben, dass das Bild bis zu einer Viewport-Breite von 900 Pixeln in einer Breite von `100vw` angezeigt wird, also über die gesamte Viewport-Breite geht (`vw` ist eine zum Viewport relative Einheit und steht für die Breite des Viewports (*viewport width)*). Bei breiteren Viewports wird das Bild nicht weiter vergrößert, sondern bleibt bei 900 Pixeln Breite. In `src` ist ein Fallback-Bild für alte Browser angegeben, die `srcset` und `sizes` nicht unterstützen (die Browser-Unterstützung für Responsive Images ist sehr gut).

Wenn ein Browser mit Unterstützung für Responsive Images das oben dargestellte Markup vorfindet, schaut er sich zunächst den Wert aus `sizes` an und sucht die erste Bedingung, die zutrifft. Wenn der Viewport des Browsers schmaler als 900 Pixel ist, weiß der Browser, dass das Bild in `100vw` ausgegeben wird. Nach der Analyse von `sizes` macht sich der Browser auf zum `srcset`-Attribut und sucht das Bild, dessen `w`-Wert am besten zu der `sizes`-Angabe passt. Dabei ist nicht nur die Viewport-Breite ausschlaggebend, sondern beispielsweise auch die Auflösung des Displays, mit dem der Browser angezeigt wird. Wenn es sich um ein hochauflösendes Display handelt, müsste bei einem 480 Pixel breiten Viewport schon das 900 Pixel breite Bild angezeigt werden, damit es nicht unscharf ist.

Aber keine Sorge, Sie müssen Bilder in WordPress nicht manuell anpassen, WordPress stattet die Bilder im Inhalt automatisch mit `srcset` und `sizes` aus. Das `sizes`-Attribut ist dabei mit einem Standardwert von `(max-width: bildBreite) 100vw, bildBreite` ausgestattet, was bedeutet, dass der Browser davon ausgeht, dass das Bild bei einem Viewport, der schmaler als das Bild ist, über die gesamte Viewport-Breite angezeigt wird und auf breiteren Viewports in seiner vollen Größe. Das ist natürlich oft nicht exakt so, und manchmal sogar recht weit von der Wirklichkeit entfernt, wenn zum Beispiel in mehreren Spalten Teaser zu Blogbeiträgen mit dem Beitragsbild angezeigt werden, das dort in derselben Größe eingebunden ist wie auf der Detailseite des Blogbeitrags, wo das Bild aber deutlich größer dargestellt wird. Aber es ist ein Ausgangswert, der okay ist, und für die meisten Fälle dürfte sich eine Optimierung dieses Werts über Filter nicht lohnen.

Die Option, ein Bild für das Web zu speichern, die Sie vielleicht aus Bildbearbeitungsprogrammen kennen, ist dabei eher wenig hilfreich. Besser geeignet sind Werkzeuge wie TinyPNG (trotz des Namens kann es auch JPGs verarbeiten). Laden Sie ein Bild versuchsweise einmal bei *tinypng.com* hoch und schauen Sie, was das Tool an Ersparnis bringt, ohne die Qualität des Bildes merklich zu verringern. Mit diesem und ähnlichen Tools können Sie Ihre Bilder vor dem Upload optimieren und die Dateigröße teils deutlich verringern. Zwei Nachteile: Es ist ein zusätzlicher Schritt vor dem Upload notwendig, und es wird nur die Originaldatei optimiert, nicht aber die kleineren Bildgrößen, die WordPress aus diesem Bild generiert. Für beide Nachteile gibt es unterschiedliche Services, die sich über WordPress-Plugins mit dem normalen Upload-Prozess verbinden lassen, sodass die Bilder direkt beim Upload optimiert und auch die kleineren Bildgrößen behandelt werden. Der Nachteil hier: In aller Regel sind sie kostenpflichtig, wobei die

unterschiedlichen Preispläne auf verarbeiteten Dateigrößen oder der Dateianzahl basieren. Zum Teil gibt es auch kleine kostenlose Pläne. Ein empfehlenswerter Dienst in dem Bereich ist *Imagify* (*imagify.io*), das entsprechende WordPress-Plugin ist auf WordPress.org verfügbar (*de.wordpress.org/plugins/imagify/*). Imagify bietet einen kostenlosen Plan an, mit dem Sie monatlich 20 MByte an Bildern optimieren können, wobei ein Bild maximal 2 MByte groß sein darf (in den Bezahlplänen gibt es kein Limit der Dateigröße).

Plugin	Imagify
Verbreitung	900.000+
Download	*https://de.wordpress.org/plugins/imagify/*
Zweck	Plugin zur Optimierung Ihrer Bilder, sowohl direkt beim Upload als auch für bestehende Medien in der Mediathek
Komplexität	+

Nach der Installation des Plugins finden Sie die Einstellungen unter Einstellungen • Imagify (siehe Abbildung 19.30).



Abbildung 19.30 Die Einstellungen von Imagify finden sich alle auf einer Seite wieder.

Imagify bietet Ihnen die folgenden Optionen:

- Allgemeine Einstellungen • Bilder beim Hochladen automatisch optimieren. In der Regel sollte hier ein Haken gesetzt sein, es sei denn, Sie laden häufig Bilder hoch, die Sie aus irgendwelchen Gründen nicht optimieren möchten. Nachträglich können Sie Bilder in der Mediathek optimieren; so können Sie auch Bilder optimieren, die Sie vor der Installation von Imagify hochgeladen haben (siehe Abbildung 19.31).

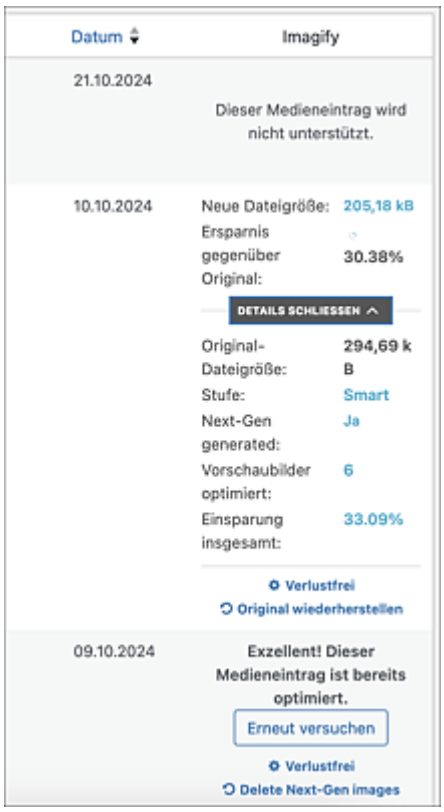


Abbildung 19.31 In der Listenansicht der Mediathek gibt es eine Spalte »Imagify«, in der Sie die Originalversion eines Bilds wiederherstellen oder nicht optimierte Bilder optimieren können.

- Allgemeine Einstellungen • Backup der Original-Bilder. Sollte zur Sicherheit aktiv sein. Damit können Sie die unveränderten Bilder wiederherstellen.
- Allgemeine Einstellungen • Lossless compression. Wenn Sie möchten, dass auch kein noch so winziger sichtbarer Unterschied zwischen Original und optimierter Version vorhanden ist, ist das Ihre Option.
- Optimierung • Next-gen image format. Hier können Sie auswählen, ob Sie für unterstützende Browser die Bilder in moderneren Formaten bereitstellen möchten. Dabei haben Sie die Auswahl zwischen AVIF und WebP. WebP ist ein von allen modernen Browsern unterstütztes neueres Bildformat, das häufig kleinere Bildgrößen ermöglicht als JPEG und PNG. AVIF ist noch nicht so

weit verbreitet, bietet unter Umständen aber noch kleinere Dateigrößen (das muss aber nicht sein). Haben Sie eins der modernen Formate gewählt, gibt es zwei Möglichkeiten, die Bilder an Ihre Benutzer*innen auszuspielen: über eine Änderung in der *.htaccess* oder über das `picture`-Element.

- Optimierung • Grössere Bilder verkleinern. Wenn Sie diese Option aktivieren, werden hochgeladene oder nachträglich optimierte Bilder, die breiter als der eingegebene Wert sind, auf diesen Wert verkleinert. Seit WordPress 5.3 kümmert sich der Core von WordPress bereits darum, dass bei sehr großen hochgeladenen Bildern eine kleinere Version erstellt wird, die dann ausgegeben wird (standardmäßig liegt der Maximalwert bei 2560 Pixeln Höhe oder Breite). Die Originalversion wird dabei aber nicht verändert, sondern beibehalten.
- Optimierung • Datei-Optimierung. Hier können Sie auswählen, welche der von WordPress generierten Bildgrößen neben der Originalgröße von Imagify optimiert werden sollen.
- Optimierung • Eigene Ordner. In diesem Bereich können Sie zusätzliche Ordner auf dem Webspace auswählen, aus denen die Bilder optimiert werden sollen. Es ist auch möglich, die Bilddateien des Themes zu optimieren.
- Anzeige-Optionen. Hier können Sie einstellen, ob oben in der Admin-Leiste ein Imagify-Eintrag vorhanden sein soll, mit dem Sie unter anderem direkt zu den Einstellungen oder der Mengenoptimierung springen können.

Nachdem Sie Imagify eingerichtet haben, müssen Sie sich von diesem Zeitpunkt an eigentlich keine Gedanken mehr um die Bildoptimierung machen. Die Originalbilder vor dem Upload zu skalieren, bleibt aber eine gute Idee – allein schon, weil sonst das Datenvolumen von Imagify unnötig schnell aufgebraucht ist.

Info: Moderne Bildformate WebP, AVIF und JPEG XL

WebP, AVIF und JPEG XL sind modernere Bildformate, die kleinere Dateigrößen als die »alten« Formate JPEG und PNG ermöglichen. AVIF und JPEG XL können dabei noch einmal mehr als WebP, sind aber auch deutlich neuer und werden damit schlechter von Browsern unterstützt. Während der Einsatz von WebP heute bereits eigentlich bedenkenlos möglich ist, ist AVIF bei den neuesten Browser-Versionen zwar auch schon sehr gut unterstützt, in Opera Mini allerdings noch gar nicht, in Edge erst seit Anfang 2024 und in Safari erst seit Version 16.1 auf macOS 13. Bei AVIF kann also noch halbwegs schnell ein älteres System ohne Unterstützung vorkommen. Bei JPEG XL sieht die Unterstützung deutlich schlechter aus, das Format ist bisher nur in Safari unterstützt und hinter einer Feature-Flag in Firefox.

JPEG XL hat gegenüber AVIF einige Vorteile, so können beispielsweise normale JPEGs verlustfrei in JPEG XL transformiert werden, um Dateigröße einzusparen. Diese Bilder lassen sich für Systeme, die kein JPEG XL unterstützen, aber wieder in normale JPEGs umwandeln. Es braucht also nur eine JPEG-XL-Datei, um sowohl JPEG als auch JPEG XL ausspielen zu können. Die Umwandlung soll dabei deutlich performanter sein als bei AVIF. Ein weiterer Vorteil ist die Unterstützung von Progressive Decoding, was bedeutet, dass ein Bild schon in einer qualitativ schlechten Version angezeigt werden kann, bevor es vollständig heruntergeladen wurde.

19.7 Optimieren für den Klimaschutz

Jede Website, egal ob klein oder groß, mit WordPress gebaut und selbst gehostet oder auf einer Software-as-a-Service-Plattform betrieben, erzeugt mit jedem Aufruf CO_2-Emissionen. Was im ersten Moment nicht wirklich logisch klingt, wird schnell klarer, wenn wir betrachten, was bei einem Aufruf einer beliebigen Webseite passiert:

Auf dem Webserver, der die Website beherbergt, laufen Prozesse, Datenbankabfragen und Speicherzugriffe. Oftmals dynamisch erzeugtes HTML wird ausgeliefert. Um zu dem Gerät zu gelangen, das den Aufruf gestartet hat, müssen die übertragenen Daten nicht nur durch das Netzwerk des Rechenzentrums des Hosting-Anbieters wandern, sondern auch eine gewisse Distanz durch die physische Welt und mehrere kleine und große Netzwerkknoten zurücklegen, bevor sie im Netzwerk ankommen, in dem sich das Empfangsgerät befindet. Handelt es sich hier um ein Mobilgerät, so findet oft noch ein Transport der Daten über einen drahtlosen Übertragungsstandard wie LTE oder 5G statt. Viele Geräte hinter privaten Internetanschlüssen sind per WLAN mit dem Internet verbunden. Und auch auf dem empfangenen Gerät finden Prozesse statt, die neben dem Anfragen und Empfangen der Daten auch noch die Interpretation und Darstellung des Website-Codes umfassen.

Jeder Einzelne dieser Schritte benötigt Strom. Und genau wie der Strom, mit dem Sie Ihren Kaffee oder Tee kochen, wird bei der Produktion dieses Stroms eine gewisse Menge CO_2 freigesetzt. Für jeden Aufruf werden in den meisten Fällen relativ kleine Mengen Energie aufgewendet. Dadurch bewegen wir uns auch beim CO_2-Ausstoß oft im Bereich von ≤1 g pro Aufruf. Doch wenn Sie das über

die Anzahl der Seitenaufrufe im Laufe eines Monats addieren, können bei gut besuchten Websites schnell mehrere Hundert Kilo CO_2-Ausstoß pro Jahr zusammenkommen. Und natürlich fallen hier nicht nur Aufrufe von Menschen ins Gewicht, sondern auch die von Bots und Suchmaschinen, die das Web Tag und Nacht scannen.

Die gute Nachricht ist, dass viele der Maßnahmen und Werkzeuge, die Sie in diesem Kapitel bereits kennengelernt haben, Ihre Website nicht nur schneller machen, sondern auch den negativen Einfluss auf die Umwelt reduzieren.

19.7.1 CO_2-Ausstoß ermitteln

Bevor Sie Ihre Website im Hinblick auf den CO_2-Ausstoß optimieren können, müssen Sie ein Gefühl dafür bekommen, wo Sie im Moment stehen – und im besten Fall sogar dafür, was die Problemfelder sind, an denen Sie arbeiten können.

Diesen Wert können Sie mit einem relativ einfachen Werkzeug schätzen lassen. Ähnlich wie Performance-Analyse-Tools, die Sie in diesem Kapitel bereits kennengelernt haben, geben Sie auch hier einfach eine URL ein, um diese überprüfen zu lassen.

Hinweis: Methodik

Unzählige Faktoren können einen Einfluss auf den konkreten Stromverbrauch – und damit den CO_2-Ausstoß – Ihrer Website haben. Alle aktuellen Werkzeuge zur Ermittlung dieses Ausstoßes betrachten vorrangig die Menge der übertragenen Daten und nehmen diese als Berechnungsgrundlage für den Stromverbrauch. Daraus ergibt sich ein etwas unvollständiges, aber durchaus tragfähiges Bild, weil tatsächlich ein Großteil der nötigen Energie

zum Betrieb einer Website für Speicherung und Übertragung der Daten aufgewendet werden muss.



Abbildung 19.32 Mit dem Website Carbon Calculator können Sie den CO_2-Ausstoß Ihrer Website schätzen.

Während es eine ganze Reihe von Tools und Websites gibt, die Ihnen Schätzungen zum CO_2-Ausstoß Ihrer Website geben, nutzen wir in diesem Beispiel *websitecarbon.com*. Von einer Agentur aus Großbritannien betrieben, ist Website Carbon seit vielen Jahren ein Vorreiter auf diesem Gebiet und liefert kostenfrei zuverlässige Ergebnisse.

Wenn Sie *websitecarbon.com* in Ihrem Browser aufrufen, können Sie direkt in das Feld Your web page Address die URL zur Webseite eingeben, die Sie testen möchten. Ein Klick auf Calculate startet den

Vorgang, der je nach Umfang der zu testenden Website 5–30 Sekunden dauern kann.

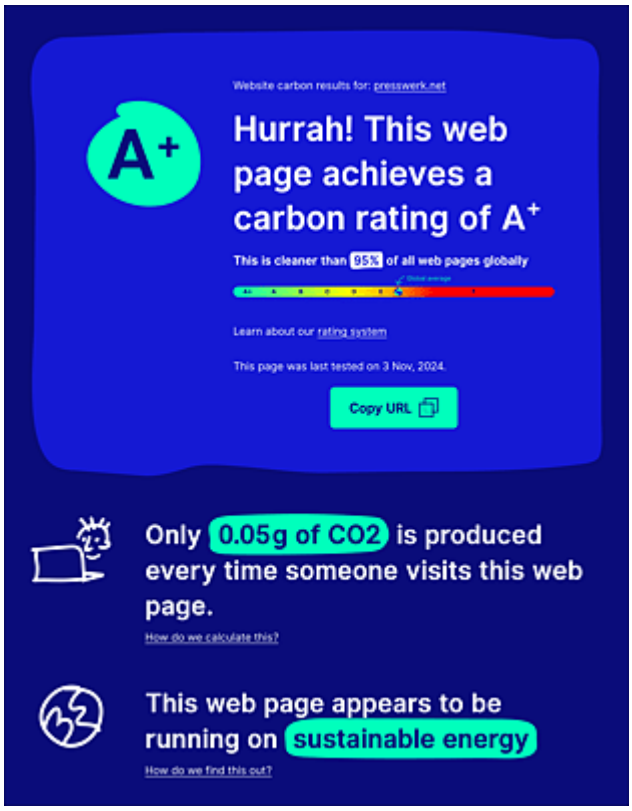


Abbildung 19.33 Nach erfolgtem Test erhalten Sie drei wichtige Werte: eine Art Energie-Label, einen CO_2-Ausstoß pro Seitenaufruf und Daten zur Nachhaltigkeit des verwendeten Stroms Ihres Hosting-Anbieters.

Sobald die Analyse abgeschlossen ist, bekommen Sie eine vereinfachte Auswertung in Form eines Energie-Labels. Diese Labels kennen Sie vielleicht von Kühlschränken, Wasserkochern oder Glühbirnen. Auf der Skala von A+ bis F soll Ihnen diese Angabe einen schnellen Eindruck geben, wie effizient – und damit klimafreundlich – die getestete Webseite abschneidet. Der globale Durchschnitt liegt 2024 knapp unter E. Für eine ambitionierte Optimierung können Sie sich aber durchaus vornehmen, mindestens einen Wert von B oder C zu erreichen.

Der nächste Wert ist der CO_2-Ausstoß pro Seitenaufruf. Für sich genommen ist das eine sehr abstrakte Zahl. Ein Wert unter 0,5 g/Aufruf wären hier das mindeste Ziel, das Sie anstreben sollten – dieser Wert ist in etwa deckungsgleich mit einem Energie-Label von C. Weiter unten auf der Ergebnisseite können Sie die Anzahl der monatlichen Website-Aufrufe eingeben, um direkt eine Hochrechnung der gesamten monatlichen CO_2-Menge zu erhalten.

Zusätzlich *versucht* das Tool, herauszufinden, ob Ihr Hosting-Anbieter zum Betrieb Ihres Servers erneuerbare Energien einsetzt. Leider gibt es keine Möglichkeit, das zweifelsfrei zuzuordnen. Sollten Sie hier also eine Warnung angezeigt bekommen, weil vermeintlich fossile Energien zum Betrieb der Website verwendet werden, setzen Sie sich mit Ihrem Hosting-Anbieter in Kontakt und fragen Sie dort nach den konkreten Maßnahmen, die das Unternehmen zur Reduktion von Klimaschäden durch seinen eigenen Betrieb unternimmt.

Hinweis: Schätzungen

Es liegt leider in der Natur der Sache, dass Tools wie *websitecarbon.com* keine wirklichen Messwerte Ihrer Website ergeben können. Wie bereits beschrieben sind neben den Systemen Ihres Hosters andere Netzwerke und schließlich die Empfänger-Geräte und Hausanschlüsse Teile dieses Systems. Welchen Weg Daten nehmen und welche Hardware konkret Teil der Übertragung ist, variiert und ist nahezu unvorhersehbar. Daher arbeiten alle diese Tools mit Schätzungen und Näherungswerten. Solange Sie Ihre Website immer mit demselben Tool überprüfen und die Ergebnisse im Zeitverlauf vergleichen, sollte das aber kein großes Problem darstellen, weil die Methodik, mit der die Daten ermittelt werden, gleich bleibt.

19.7.2 CO_2-Ausstoß reduzieren

Ähnlich wie bei der Optimierung Ihrer Website für schnelle Ladezeiten ist es auch bei der Optimierung für geringe CO_2-Emissionen eines der obersten Ziele, die übertragene Dateigröße für einen Website-Aufruf zu reduzieren.

Hosting optimieren

Doch bevor Sie sich damit beschäftigen, gilt es, das eigene Hosting auf Herz und Nieren zu überprüfen. Energie- und Wasserverbrauch durch den Betrieb von Rechenzentren sind in den letzten Jahren enorm gestiegen. Ohne zu tief in die Materie einzusteigen, können Sie zumindest sichergehen, dass für den Betrieb Ihrer Website erneuerbare Energien (Öko-Strom) eingesetzt werden. Man geht davon aus, dass der Wechsel von fossilen zu erneuerbaren Energien bei uns in Mitteleuropa eine CO_2-Reduktion von fast 90 % bedeuten. Wenn Sie es noch etwas genauer nehmen wollen, informieren Sie sich über weitere Maßnahmen, die Ihr Hosting-Anbieter unternimmt. Wird etwa die Abwärme der Server zum Heizen verwendet? Gibt es Pläne zur Nachnutzung ausgedienter Hardware? Oder zieht Ihr Anbieter sich darauf zurück, sich durch den Kauf von CO_2-Zertifikaten einen grünen Anstrich zu geben?

Bilder optimieren

Eine der wichtigsten Maßnahmen für die tatsächliche Reduktion der übertragenen Dateigröße haben Sie mit der Bildoptimierung in Abschnitt 19.6 schon kennengelernt. Durch den Einsatz eines Bildoptimierungs-Plugins, passender Bildmaße und moderner

Dateiformate können Sie einen Großteil der Dateigröße, die auf Ihrer Website für Bilder anfallen, reduzieren. Und weil Bilder auf einer durchschnittlichen Website knapp die Hälfte der übertragenen Dateigröße ausmachen, hat diese Maßnahme allein schon einen großen Einfluss. Auch der Einsatz von Lazy Loading von Bildern, wie bereits in Abschnitt 19.2.2 beschrieben, reduziert die übertragene Dateigröße deutlich.

Browser-Caching

Zusätzlich können Techniken wie Browser-Caching, bei dem einige statische Dateien wie CSS oder Schriftarten lokal im Browser zwischengespeichert werden, die Menge der übertragenen Daten deutlich herabsetzen. Ist Browser-Caching für Ihre Website aktiv, werden die betreffenden Dateien für jedes Gerät, das Ihre Website aufruft, einmal beim ersten Aufruf übertragen und dann lokal gespeichert. Für folgende Aufrufe müssen dieselben Daten dann nicht erneut den Weg durch das Netz antreten. Das reduziert die Ladezeit und übertragene Dateigröße deutlich.

19.7.3 Weitreichendere Verbesserungen

Ganz im Sinne einer minimalistischen und eleganten Website ist weniger oftmals mehr. Das bedeutet natürlich nicht, dass Sie sich auf eine asketische Website ohne Bilder, Farben und Styling beschränken sollen.

Aber es gibt einige Aspekte zu beachten, die sich tatsächlich positiv auf die CO_2-Bilanz und die Ladegeschwindigkeit Ihrer Website auswirken werden.

Struktur

Neben der unmittelbaren Reduzierung der Größe Ihrer Website können Sie auch die Anzahl der nötigen Aufrufe verringern und damit unterm Strich übertragene Daten und damit CO_2-Emissionen einsparen. Durch eine optimierte Struktur und Navigation tragen Sie dazu bei. Versetzen Sie sich dafür in Ihre Besucher*innen hinein. Warum kommen diese auf Ihre Website? Wonach suchen sie, und wie können Sie Ihnen dabei helfen? Wenn pro Besucher*in statt 4–5 künftig zum Beispiel nur noch 2 Seiten aufgerufen werden müssen, reduzieren Sie außerdem nicht nur den CO_2-Ausstoß Ihrer Website, sondern verbessern auch noch die Zufriedenheit Ihrer Besucher*innen.

Diese Optimierung bietet sich vor allem für Geschäfts-Websites an, auf denen Ihr Publikum gezielt nach Informationen sucht. Wenn Sie zum Beispiel ein Blog betreiben, ist es vermutlich eher in Ihrem Interesse, wenn Leser*innen noch einen zweiten oder dritten Beitrag lesen wollen.

Große Bilder im Header oder Hintergrund

Auch wenn Sie Bilddateien schon mittels eines Plugins optimieren, sollten Sie den Einsatz von (großen) Bildern an einigen Stellen gesondert hinterfragen.

Die Zeiten, in denen große Bilder im Header von Websites im Trend waren, sind schon seit einigen Jahren vorbei. Stattdessen haben Sie damit weit oben auf jeder Seite ein Bild, das in jedem Fall geladen werden muss. Gehen Sie doch einmal in sich, und überlegen Sie, ob Sie bereit wären, Ihre Website hier ein wenig zu optimieren, indem Sie auf ein großes Header-Bild verzichten.

Ähnlich verhält es sich mit Hintergrundbildern. Auch diese sind in den letzten Jahren etwas aus der Mode gekommen und wurden von einfarbigen oder sogar weißen Website-Hintergründen abgelöst.

Videos

Aber alle Hintergrundbilder sind harmlos im Vergleich zu Videos, die im Hintergrund der Website laufen. In besonders eindrücklichen Fällen können hier auch mal über 100 MB Datenübertragung beim einfachen Betrachten einer Webseite zusammenkommen. Gleichzeitig sind diese bewegten Hintergründe in den seltensten Fällen informativ und haben eher negative Konsequenzen, wenn wir an Bereiche wie die Barrierefreiheit denken. Aus all diesen und vielen weiteren Gründen sollten Sie auf Hintergrundvideos eher verzichten.

Wenn Sie an anderer Stelle auf Ihrer Website Videos nutzen, um zum Beispiel Aspekte eines Produkts vorzustellen, ist die erste Maßnahme, diese Videos nicht direkt zu laden, wenn die Website geladen wird. Unter Umständen genügt ein Link zum zugehörigen YouTube-Video? Oder, wenn Sie ein Video direkt auf der Seite anzeigen wollen, eine 2-Klick-Lösung wie das Embed-Privacy-Plugin, das wir in Kapitel 13 vorgestellt haben? Damit müssen Ihre Besucher*innen explizit zustimmen, bevor das Video nachgeladen wird. Gut für den Datenschutz und für die übertragene Dateimenge.

20 Suchmaschinenoptimierung

*Wenn Sie nicht gerade eine reine Hobby-Website betreiben und es Ihnen nicht so wichtig ist, ob und wie viele Besucher*innen Sie haben, sollten Sie sich zumindest ein wenig mit dem Thema Suchmaschinenoptimierung auseinandersetzen.*

Das Thema *Suchmaschinenoptimierung* (SEO), also die Optimierung Ihrer Website für eine möglichst gute Auffindbarkeit in Suchmaschinen bei für Sie wichtigen Suchbegriffen, ist ein sehr breites Feld. Am wichtigsten ist, wie für menschliche Besucher*innen: der Inhalt.

Grob gesagt teilt sich SEO in zwei Bereiche auf: Onpage- und Offpage-Optimierung. Onpage-Optimierung betrifft alles, was Sie auf der Website selbst machen, also angefangen vom Inhalt bis hin zu technischen Optimierungen, die beispielsweise mit Plugins vorgenommen werden wie denen, die Sie in diesem Kapitel kennenlernen werden. Offpage-SEO betrifft Maßnahmen, die abseits von der Ziel-Website getroffen werden – sei es das Aufbauen von externen Links, Aktivitäten auf Social Media oder andere Marketing-Maßnahmen.

Hier werden Sie keine konkreten SEO-Strategien oder Aspekte der Offpage-Optimierung finden. Sie werden zwei Plugins kennenlernen, die Ihnen helfen, die technische Voraussetzung zu schaffen, dass Ihre Inhalte gut indexiert werden können. Aber das

allein reicht natürlich nicht aus, es kommt wie gesagt besonders auf die Inhalte selbst an. In jedem Fall ist SEO kein Thema, bei dem von heute auf morgen sichtbare Erfolge erzielt werden können, sondern eine langfristige Aufgabe.

20.1 Schlank mit allen wichtigen Funktionen – Plugin »The SEO Framework«

Das Plugin »The SEO Framework« ist ein vergleichsweise schlanker Vertreter eines SEO-Plugins, das sich auf technisches SEO konzentriert, also darauf, dass die Inhalte gut indexiert werden können und an passenden Stellen das Markup um für Suchmaschinen nützliche Informationen ergänzt oder erweitert wird.

Es bietet von Haus aus keine Funktionen, die Sie bei dem Schreiben von Inhalten für Suchmaschinen unterstützen, Sie können solche Funktionen aber über Plugins nachrüsten. Ebenfalls solche Funktionen bietet direkt in der Grundfunktion das Plugin Yoast SEO, das im nächsten Abschnitt vorgestellt wird.

Plugin	The SEO Framework
Verbreitung	100.000+
Download	*https://de.wordpress.org/plugins/autodescription/*
Zweck	ein übersichtliches SEO-Plugin für technisches SEO in einem aufgeräumten Interface
Komplexität	++

Wenn Sie das Plugin installiert und aktiviert haben, finden Sie die Einstellungen links im Admin-Menü unter dem Punkt **SEO**. Das

Plugin hat nur eine Einstellungsseite, die in unterschiedliche Boxen unterteilt ist (die Elemente kommen Ihnen von den Dashboard-Widgets her vielleicht bekannt vor), die Sie bei Bedarf über die Pfeile rechts oben umsortieren können. Generell kommt das Plugin bei den meisten Einstellungen mit sinnvollen Standardwerten, aber einige Dinge wie die Einstellungen für die Startseite sollten Sie sich auf jeden Fall anschauen.

20.1.1 Allgemeine Einstellungen

Die Box für Allgemeine Einstellungen beinhaltet Einstellungen, die in fünf unterschiedlichen Tabs aufgeteilt sind.

Gestaltung

In dem Tab Gestaltung haben Sie die folgenden Einstellungsmöglichkeiten:

- SEO-Leisteneinstellungen bietet Optionen zur Anzeige einer Leiste mit Hinweisen zu SEO-Aspekten für Seiten und Beiträge in den Übersichtstabellen und in der SEO-Metabox, die unten im Inhaltseditor angezeigt wird.
- Zähler-Einstellungen erlaubt Ihnen zu aktivieren, ob für den Titel und die Beschreibung eines Inhalts die Zeichen gezählt werden sollen und geprüft werden soll, ob die Eingabe auf die Suchergebnisseite passt.

Leistung

Im Leistung-Tab können Sie einstellen, ob es möglich sein soll, Inhalte von der Website-internen Suchfunktion und/oder Archivseiten zu entfernen, und ob diese Veränderung der

WordPress-Querys in der Datenbank oder auf der Website geschehen soll.

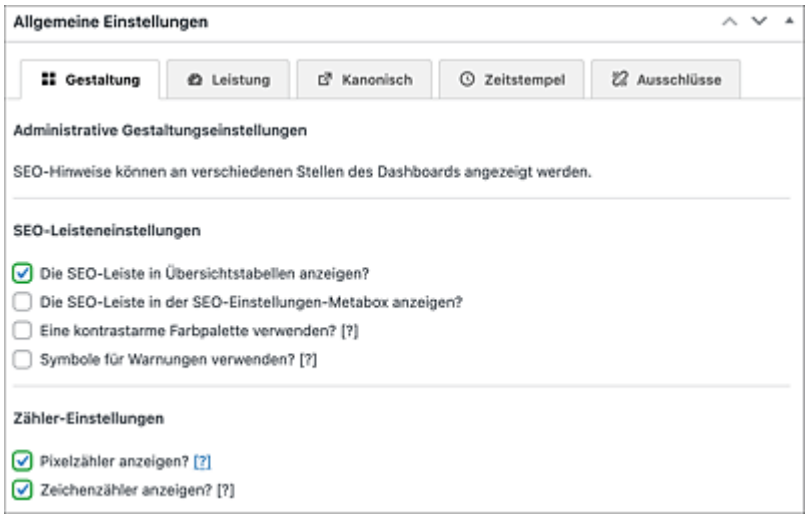


Abbildung 20.1 Im Tab »Gestaltung« der Box für die »Allgemeinen Einstellungen« können Sie unter anderem festlegen, ob in der Übersicht von Seiten und Beiträgen eine Spalte mit Hinweisen zum SEO-Stand für die Inhalte angezeigt werden soll.

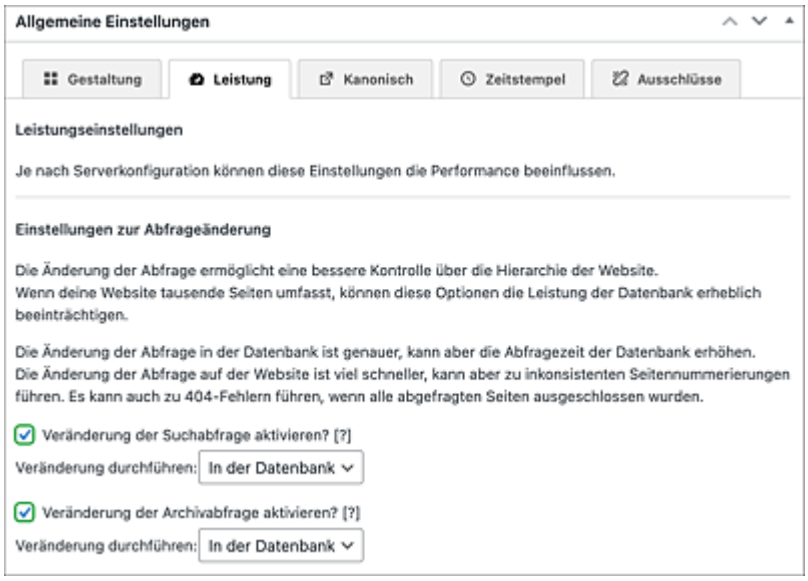


Abbildung 20.2 Sie können Seiten von der Suchfunktion auf der Website ausschließen und Inhalte aus den Archiv-Listen entfernen.

Kanonisch

Im Tab Kanonisch können Sie unter Schema-Einstellungen das bevorzugte URL-Schema für die kanonische URL angeben (hier sollte HTTPS ausgewählt werden, wenn Ihre Website HTTPS unterstützt, was sie in jedem Fall tun sollte).



Abbildung 20.3 Die kanonische oder »canonical« URL ist eine Angabe für Suchmaschinen, welche URL für die Indexierung eines Inhalts verwendet werden soll, wenn ein Inhalt unter mehreren URLs verfügbar ist.

Darüber hinaus können Sie bei Einstellungen zu Linkbeziehungen bei Seitennummerierung die Ausgabe von `rel`-Links aktivieren, um bei Seiten oder Archiven mit Paginierung einen Hinweis für Suchmaschinen auszugeben, welche URL die richtige für die Indexierung ist.

Zeitstempel

Im Zeitstempel-Tab gibt es lediglich eine Einstellung für das Format des Zeitstempels zur Ausgabe vom Veröffentlichungs- und Änderungsdatum.

Ausschlüsse

Unter Ausschlüsse haben Sie die Option, für nicht WordPress-eigene Inhaltstypen oder Taxonomien die Optimierung durch das Plugin zu deaktivieren. Das verhindert nicht die Indexierung durch Suchmaschinen, sondern deaktiviert lediglich die Integration des SEO-Plugins für die angehakten Inhaltsarten und Taxonomien.



Abbildung 20.4 Hier können Sie das Datumsformat anpassen, das im Quelltext für das Veröffentlichungs- und Änderungsdatum eines Inhalts angezeigt wird.

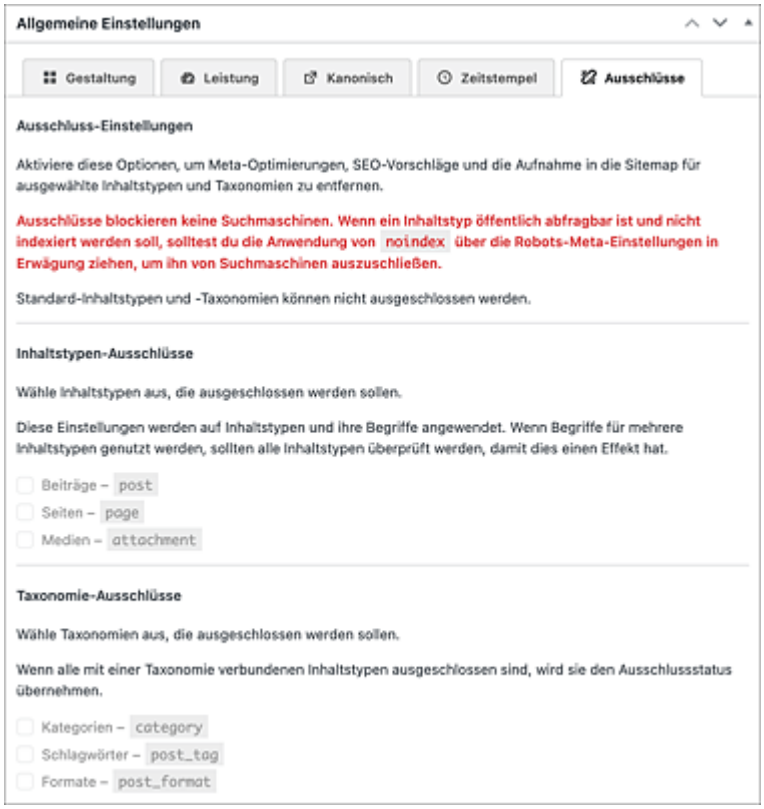


Abbildung 20.5 Sie können die Optimierung durch »The SEO Framework« für Inhaltstypen deaktivieren, die nicht standardmäßig von WordPress mitgebracht werden.

20.1.2 Titel-Einstellungen

In der Titel-Einstellungen-Box können Sie Einstellungen vornehmen, die den im `title`-Tag einer Seite ausgegebenen Titel betreffen. Diese Zeichenkette wird beispielsweise im Browser-Tab angezeigt, aber auch von Suchmaschinen als Titel in den Suchergebnislisten.

Im oberen Bereich der Box sehen Sie eine Beispielausgabe des Seitentitels und eine Beispielausgabe des Archivtitels, die direkt die gemachten Änderungen widerspiegeln (siehe Abbildung 20.6).



Abbildung 20.6 Oben in den »Titel-Einstellungen« wird eine Vorschau angezeigt.

Der untere Bereich der Box ist in die drei Tabs Allgemein, Ergänzungen und Präfixe unterteilt.

Allgemein

Unter Allgemein können Sie angeben, welches Titeltrennzeichen zwischen den einzelnen Bestandteilen des Website-Titels verwendet werden soll. Darüber hinaus können Sie HTML-Tags aus generierten Titeln entfernen.

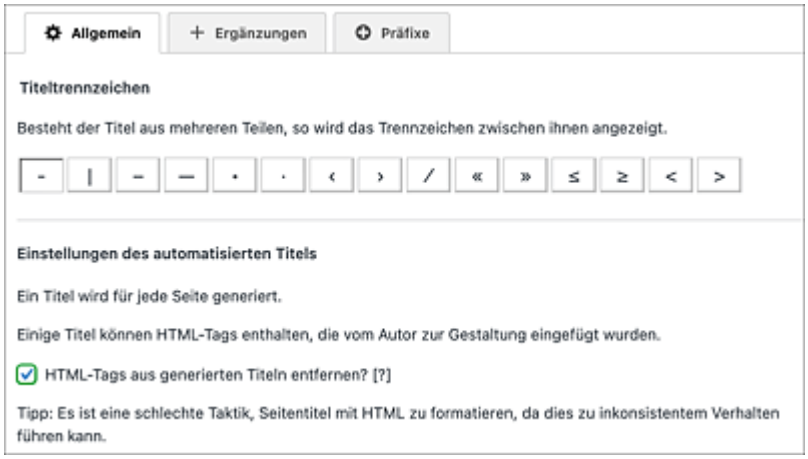


Abbildung 20.7 Hier können Sie das Trennzeichen für den Titel festlegen und steuern, ob HTML aus dem Titel entfernt werden soll.

Ergänzungen

Es ist nicht unüblich, dass der Titel der Website nicht nur im `title`-Element der Startseite angezeigt wird, sondern auch als Ergänzung aller oder der meisten anderen Titel einer Website. Hier können Sie einstellen, ob Sie genau das möchten, bei Bedarf diese Ergänzung anpassen, wenn Sie nicht den Website-Titel nutzen möchten, und den Ort des Website-Titels einstellen (Sie sollten die Ergänzung hinter dem eigentlichen Titel anzeigen, nicht davor, damit die wichtige Information vorne steht – das hat unter anderem positive Auswirkungen auf die Barrierefreiheit).

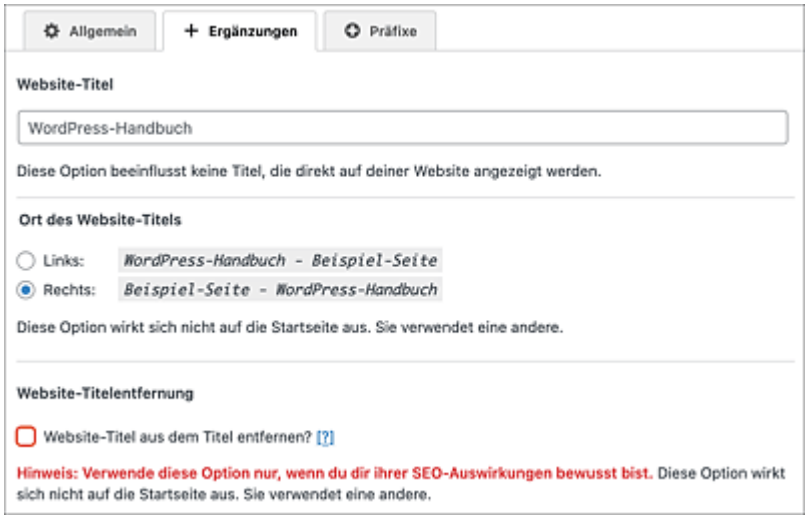


Abbildung 20.8 Im »Ergänzungen«-Tab können Sie einstellen, ob zusätzlich zum Titel beispielsweise einer Seite noch eine Ergänzung angezeigt werden soll.

Wenn der Titel kein Teil des `title`-Elements sein soll, können Sie den WEBSITE-TITEL AUS DEM TITEL ENTFERNEN.

Präfixe

Wenn Sie in WordPress eine Archiv-Seite einer Kategorie aufrufen, wird im sichtbaren Titel auf der Website standardmäßig ein »Kategorie«-Präfix angezeigt. Wenn Sie nicht möchten, dass dieses Präfix auch im `title` auftaucht, können Sie das BEGRIFFSARTPRÄFIX AUS GENERIERTEM ARCHIVTITEL ENTFERNEN.

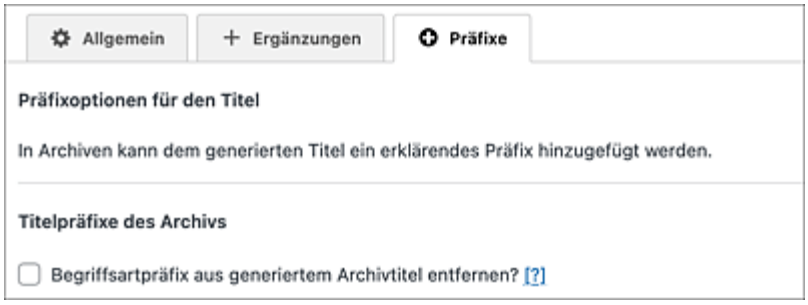


Abbildung 20.9 Hier finden Sie eine Einstellungsmöglichkeit für Titel von Archiv-Seiten.

20.1.3 Meta-Description-Einstellungen

In der Box zu den Meta-Description-Einstellungen können Sie die Option Automatische Beschreibungen generieren aktivieren, sodass Meta-Beschreibungen automatisch generiert werden, wenn Sie im Inhaltseditor keine Beschreibung eingepflegt haben.

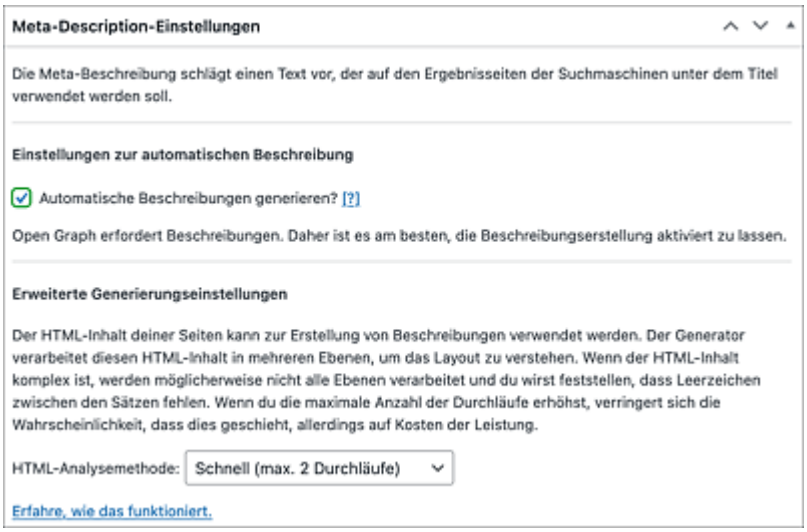


Abbildung 20.10 The SEO Framework bietet eine Option zur automatischen Generierung von Meta-Beschreibungen.

Die Einstellung HTML-Analysemethode ermöglicht Ihnen, die Anzahl der Durchläufe einzustellen, die der HTML-Parser vom Plugin nutzt, um die verschachtelten HTML-Ebenen des Inhalts durchzugehen. Mehr Durchläufe können zu besseren Ergebnissen führen, brauchen aber auch länger.

20.1.4 Startseiten-Einstellungen

Hier können Sie Einstellungen für die Startseite der Website vornehmen. Die Einstellungen sind in die Tabs Allgemein, Ergänzungen, Sozial und Robots unterteilt.

Allgemein

Im Allgemein-Tab können Sie den Titel und die Beschreibung der Startseite setzen.

Ergänzungen

Für die Startseite können Sie hier gesondert Einstellungen für Ergänzungen vornehmen. So können Sie einen Text als Ergänzung eingeben (der Standardwert ist die Website-Beschreibung) und einstellen, ob eine Ergänzung angezeigt werden soll und wo, also links oder rechts vom Titel (wie bei der generellen Ergänzung von weiter oben sollten Sie die Ergänzung nach dem Titel anzeigen).

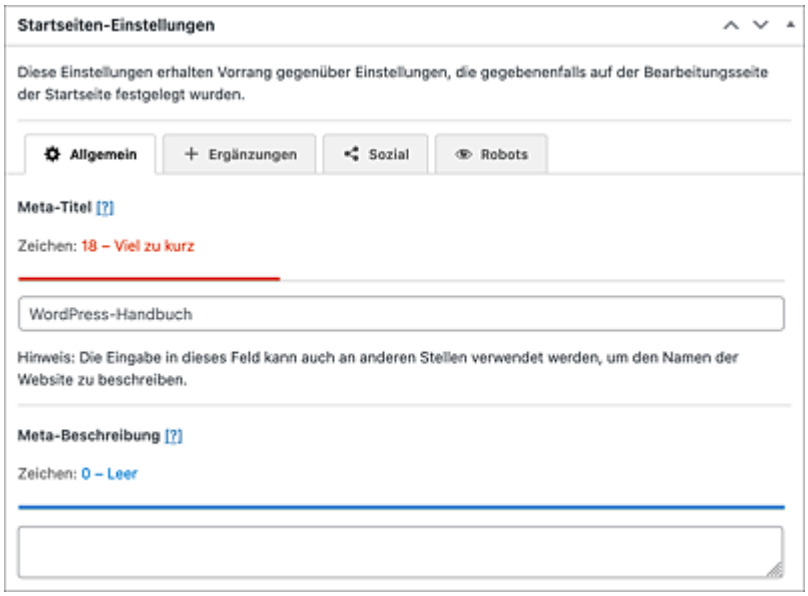


Abbildung 20.11 Hier können Sie Titel und Beschreibung der Startseite setzen.

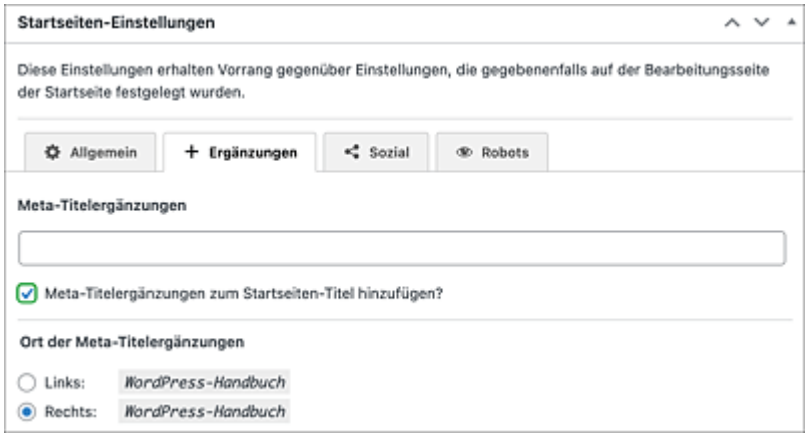


Abbildung 20.12 Einstellungen für Ergänzungen im Titel der Startseite

Sozial

Im Tab Sozial können die Meta-Angaben angepasst werden, die auf der Startseite für soziale Netzwerke ausgegeben werden. *Open Graph* ist dabei ein Standard, der initial von Facebook entwickelt wurde und inzwischen von vielen Diensten als Quelle für Vorschauen von Links verwendet wird, die geteilt werden.

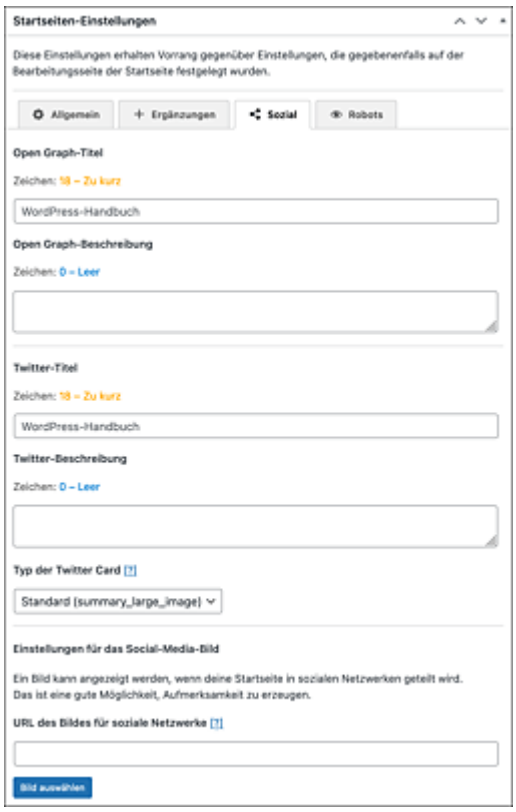


Abbildung 20.13 Hier können Sie die Werte für die Meta-Angaben für soziale Netzwerke anpassen.

Neben Titel und Beschreibung in den Open-Graph-Meta-Tags können Sie spezifische Werte für Twitter/X festlegen und den Typ der Twitter Card festlegen. Twitter Card ist die Vorschau eines Links auf Twitter. Außerdem können Sie ein Vorschaubild für soziale Netzwerke festlegen, standardmäßig wird das Website-Icon verwendet.

Robots

In dem Bereich Robots können Sie Werte für das `robots`-Meta-Tag für die Startseite anpassen. Sie werden hier schon von einem Warnhinweis des Plugins begrüßt, dass keine öffentliche Website `noindex` oder `nofollow` für die Startseite setzen sollte.

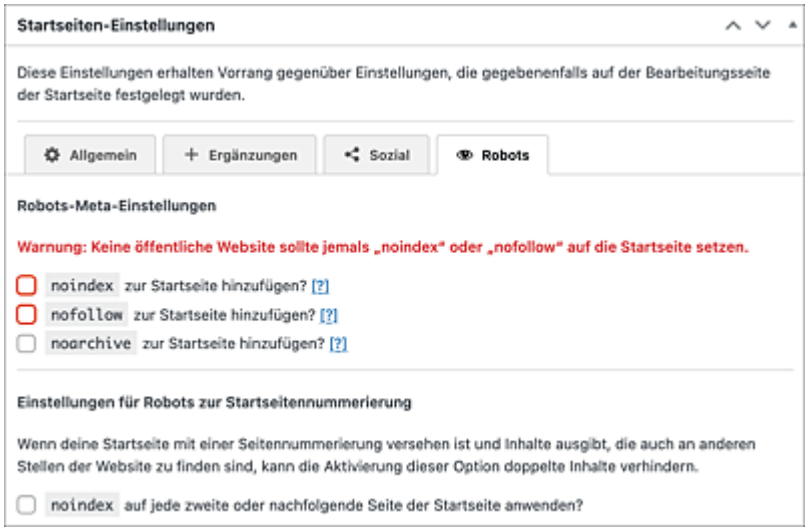


Abbildung 20.14 Hier finden Sie ein paar potenziell gefährliche Einstellungen, was die Sichtbarkeit Ihrer Website angeht.

Wenn Ihre Startseite eine Paginierung hat und auf den hinteren Seiten Inhalte ausgegeben werden, die auch woanders zu finden sind, kann mit der Einstellung NOINDEX AUF JEDE ZWEITE ODER NACHFOLGENDE SEITE DER STARTSEITE ANWENDEN verhindert werden, dass Inhalte doppelt indexiert werden.

20.1.5 Soziale Netzwerk-Meta-Einstellungen

Hier können Sie für die ganze Website gültige Einstellungen für Angaben machen, die beim Teilen eines Links beispielsweise in Social Media oder Messengern genutzt werden. Die Einstellungen sind in die Tabs ALLGEMEIN, FACEBOOK, TWITTER, OEMBED und BEITRAGSDATUM unterteilt, wobei sich je nach Einstellungen im ALLGEMEIN-Tab unterscheidet, welche Tabs angezeigt werden.

Allgemein

- EINSTELLUNGEN FÜR SOCIAL META TAGS
  Hier können Sie einstellen, ob Sie Open-Graph-Meta-Tags, Tags

für Facebook und Tags für Twitter/X ausgeben möchten. Open Graph wird von vielen anderen Diensten verwendet, auch als Fallback von welchen, für die es selbst speziellere Tags gibt. Außerdem können Sie einstellen, ob Sie sogenannte oEmbed-Skripte ausgeben möchten. Damit ist es möglich, beispielsweise Links zu einem Beitrag von Ihnen auf einer anderen WordPress-Website oder einer Plattform, die oEmbed unterstützt, als Vorschau einzufügen.

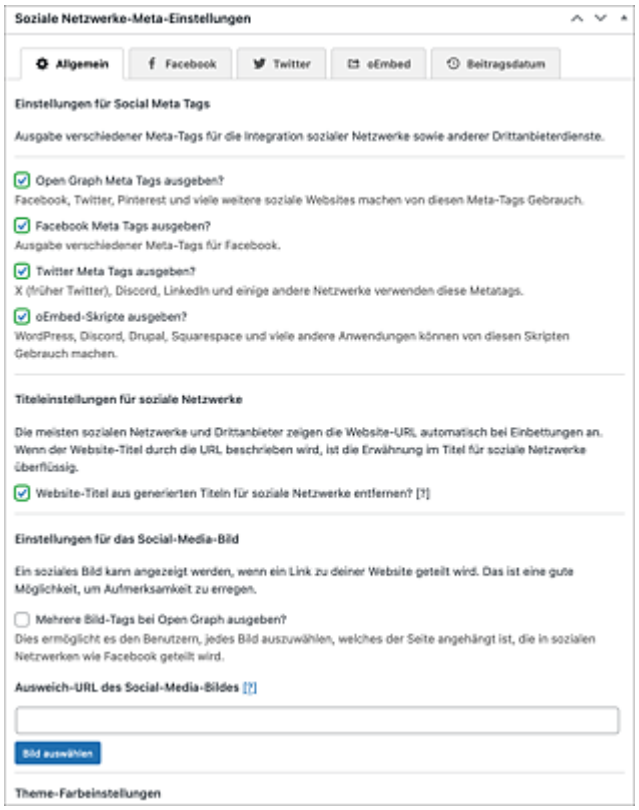


Abbildung 20.15 In den allgemeinen Einstellungen zu den Meta-Einstellungen für soziale Netzwerke können Sie unter anderem die Ausgabe verschiedener Meta-Tag-Arten für unterschiedliche Netzwerke aktivieren.

- Titeleinstellungen für soziale Netzwerke
  ermöglicht Ihnen, den Website-Titel aus generierten Titeln für soziale Netzwerke zu entfernen

- Einstellungen für das Social-Media-Bild
  Bei den Vorschauen in sozialen Netzwerken kann auch ein Bild Teil der Vorschau sein. Sie können optional Mehrere Bild-Tags bei Open Graph ausgeben, sodass Benutzer*innen beim Teilen aus den Bildern der Seite auswählen können. Außerdem können Sie über Ausweich-URL des Sozial-Media-Bildes einen Fallback angeben, wenn ein Inhalt kein Bild hat.
- Theme-Farbeinstellungen
  Hier können Sie eine zu Ihrem Design passende Farbe wählen, die als Metatag ausgegeben wird und beispielsweise von Browsern teilweise zur Anpassung von Farben im User-Interface genutzt wird.
- Website-Kurzlink-Einstellungen
  Hier können Sie einstellen, ob ein Kurzlink-Tag ausgegeben werden soll.

Facebook

Für auf Facebook geteilte Links können Sie hier einen Link zu Ihrer Facebook-Page sowie Ihrem Facebook-Profil eingeben. Die Einstellung für das Facebook-Profil kann in den Profileinstellungen der auf Ihrer Website registrierten Benutzer*innen angepasst werden.



Abbildung 20.16 Hier können Sie Facebook-spezifische Informationen zu Ihrer Website hinterlegen.

Twitter

Auch für Twitter können Sie Ihr zur Website passendes Twitter-Profil sowie Ihr persönliches Profil angeben, Letzteres können andere Benutzer*innen wieder in ihrem eigenen Profil überschreiben.

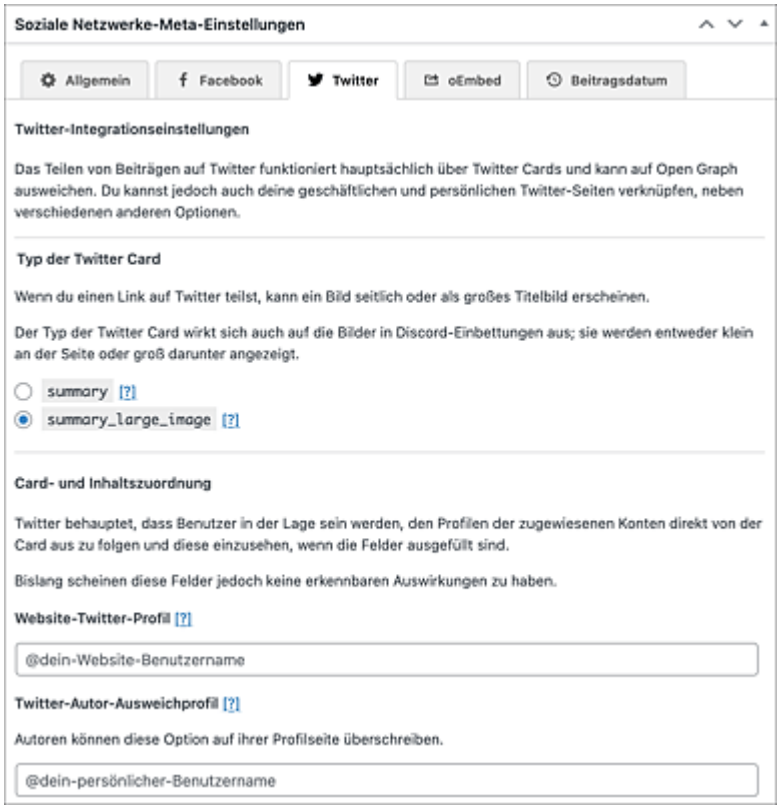


Abbildung 20.17 Für Twitter können Sie unter anderem den Twitter-Card-Typ definieren.

Außerdem haben Sie die Möglichkeit, den Typ der Twitter Card auszuwählen. Das ist die Vorschau, die beim Teilen eines Links bei Twitter/X angezeigt wird.

oEmbed

Wie schon kurz beschrieben, ist oEmbed dafür da, dass Sie beispielsweise eine Vorschau für einen Link auf einen anderen Ihrer Beiträge oder Seiten anzeigen können, wenn Sie die URL im Block-Editor in eine eigene Zeile einfügen.

Hier können Sie einstellen, ob statt des Seitentitels der Open-Graph-Titel verwendet werden soll, ob statt des Beitragsbilds auf LinkedIn das Social-Media-Image genutzt und für Discord der Autorenname entfernt werden soll.

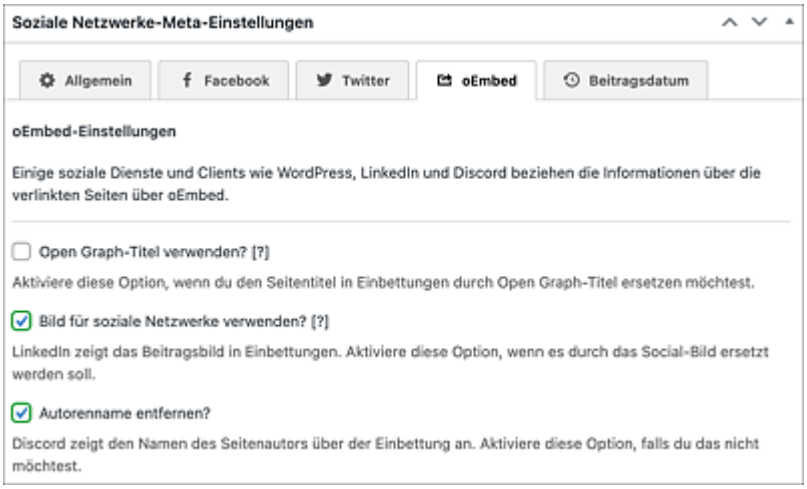


Abbildung 20.18 In den oEmbed-Einstellungen können Sie ein paar Eigenschaften verändern, die das Embed betreffen, wenn beispielsweise auf einer WordPress-Site ein Inhalt einer anderen WordPress-Site mittels eines oEmbeds als Vorschau angezeigt wird.

Beitragsdatum

Um die sozialen Netzwerke zu unterstützen, die das Veröffentlichungs- beziehungsweise Änderungsdatum eines Inhalts mit in der Vorschau anzeigen können, haben Sie hier die Möglichkeit, die entsprechenden Tags ausgeben zu lassen.



Abbildung 20.19 Hier steuern Sie, ob das Veröffentlichungs- und Änderungsdatum über entsprechende Tags ausgegeben wird.

20.1.6 Schema.org-Einstellungen

Schema.org-Markup kann als verstecktes Markup für Ihre Inhalte ausgegeben werden, um von Suchmaschinen für eine verbesserte

Anzeige der Suchergebnisse genutzt zu werden. Es gibt beispielsweise Schema.org-Markup für Bewertungen, durch die in den Suchergebnissen dann bereits die Bewertung angezeigt werden kann.

Die Einstellungen sind in die Tabs Allgemein und Presence aufgeteilt.

Allgemein

Im Tab Allgemein können Sie zuerst entscheiden, ob Sie Strukturierte Daten ausgeben möchten. Wenn die Option aktiv ist, können Sie im Bereich Erweiterte strukturierte Daten angeben, ob Sie eine Authorized Presence hinzufügen möchten, über die Sie Informationen zu Inhaber*in und Social-Media-Kanälen angeben können. Außerdem können Sie eine versteckte Brotkrümelnavigation hinzufügen, um Suchmaschinen dabei zu helfen, die Website-Hierarchie zu erfassen, und Sie können Sitelinks-Suchfelder hinzufügen, womit direkt in den Suchergebnissen ein Suchfeld für die Inhalte Ihrer Website angezeigt werden kann.

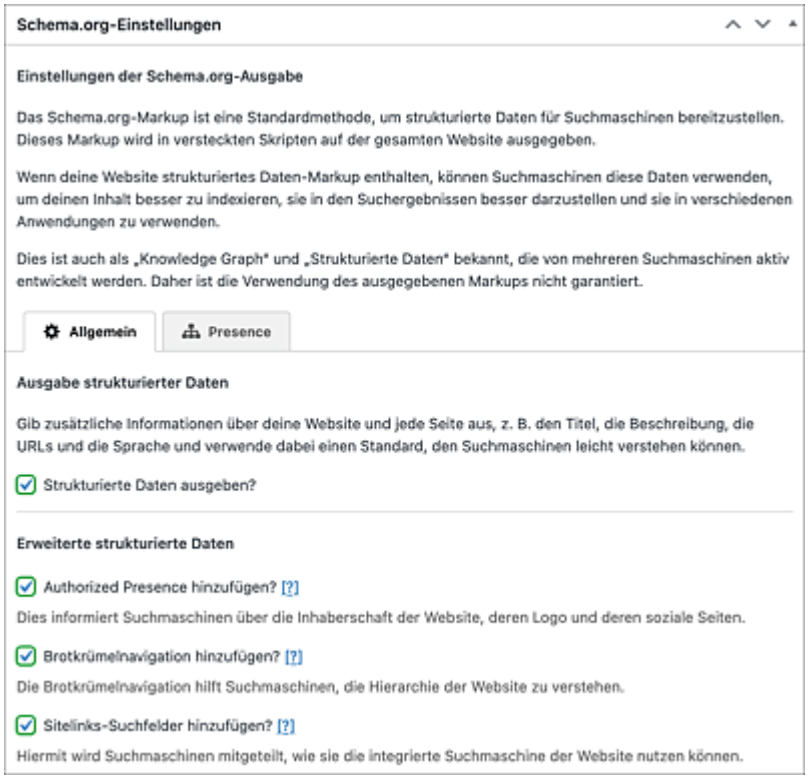


Abbildung 20.20 Hier können Sie unter anderem angeben, ob überhaupt Schema.org-Markup für strukturierte Daten ausgegeben werden soll.

Presence

Hier haben Sie die Möglichkeit anzugeben, ob die Website für eine Person oder Organisation steht, wie die Person/Organisation heißt und welche Social-Media-Kanäle es gibt.

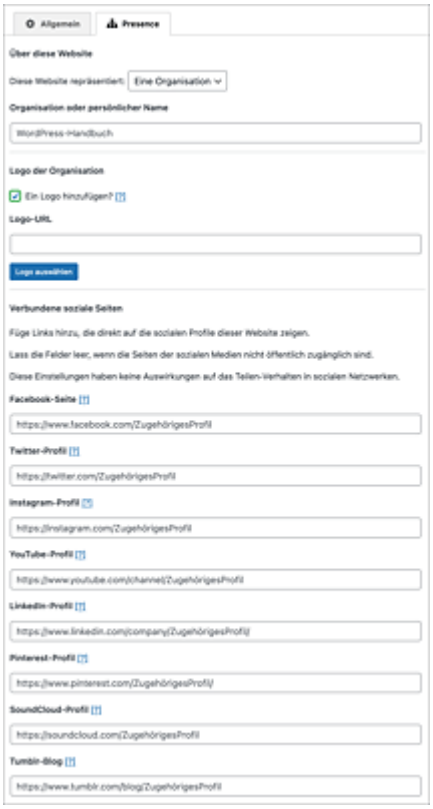


Abbildung 20.21 Im »Presence«-Tab können Sie angeben, ob die Site eine Person oder Organisation repräsentiert.

20.1.7 Robots-Meta-Einstellungen

Über den Robots-Meta-Tag können Sie beispielsweise angeben, ob eine Seite indexiert werden soll und ob der Suchmaschinen-Crawler den Links auf der Seite folgen soll. Die Einstellungen sind in die Tabs Allgemein, Indexierung, Verfolgung und Archivierung aufgeteilt.

Allgemein

Im Allgemein-Tab haben Sie die folgenden Einstellungsmöglichkeiten:

- Erweiterten Abfrageschutz aktivieren
  Einige URL-Abfragen können dazu führen, dass WordPress falsche

Archiv-Ansichten anzeigt. Durch Aktivierung dieser Option wird verhindert, dass diese Ansichten von Suchmaschinen indexiert werden.

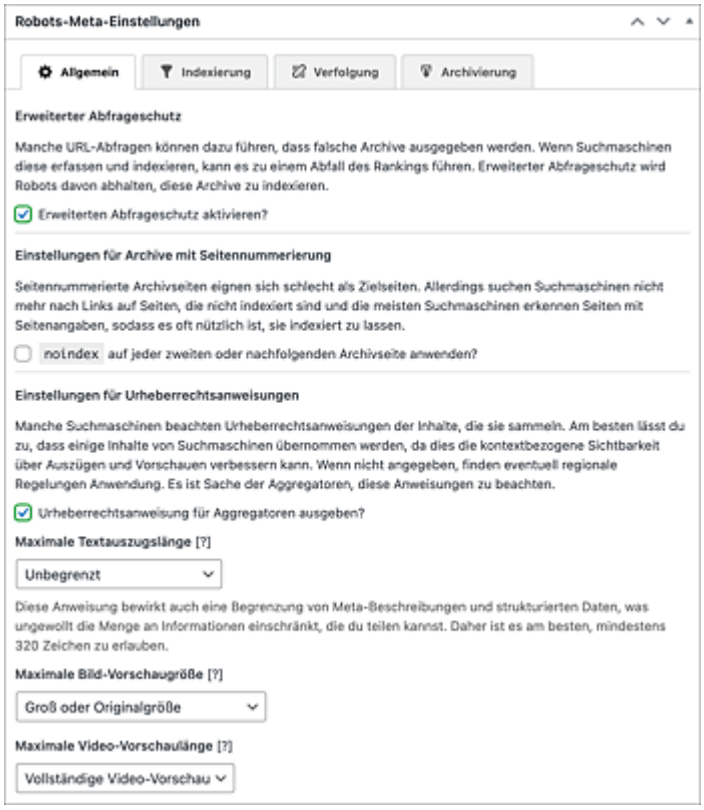


Abbildung 20.22 Die allgemeinen Einstellungen für die Robots-Metadaten betreffen beispielsweise das Setzen von »noindex« für paginierte Archive.

- NOINDEX AUF JEDER ZWEITEN ODER NACHFOLGENDEN ARCHIVSEITE ANWENDEN
  Hiermit können Sie einstellen, dass nur die erste Seite eines Archivs indexiert werden soll.
- EINSTELLUNGEN FÜR URHEBERRECHTSANWEISUNGEN
  Das sind einige Optionen, mit denen Sie Angaben darüber machen können, ob etwa in Suchmaschinenergebnissen Textauszüge angezeigt werden dürfen. Es ist Sache der Suchmaschinen, ob sie sich daran halten.

Indexierung

Unter INDEXIERUNG finden Sie unterschiedliche Optionen, um die Indexierung für bestimmte Inhaltsarten oder Übersichtsansichten zu verhindern.

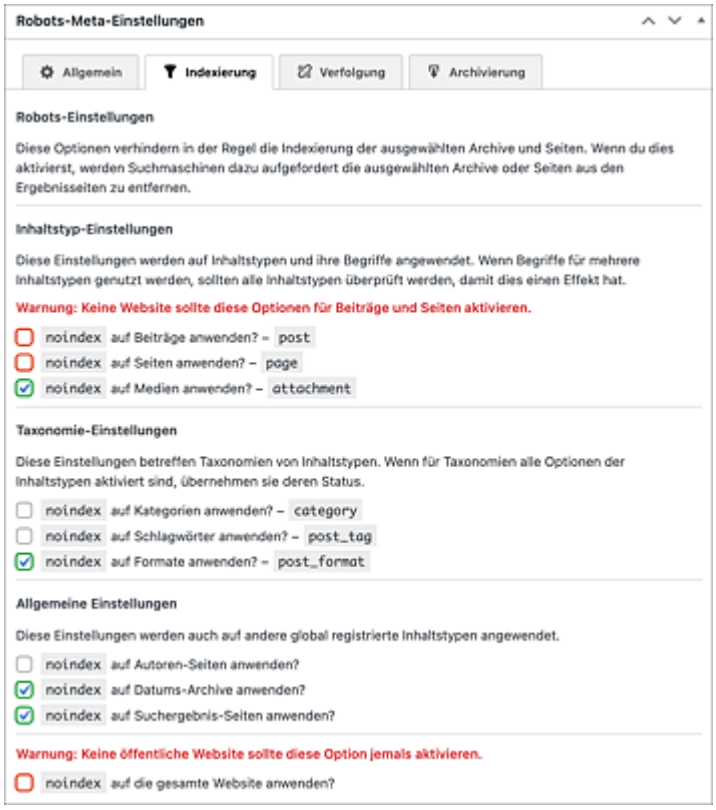


Abbildung 20.23 Hier können Sie je Inhaltsart und Taxonomie einstellen, ob diese generell nicht indexiert werden sollen.

So können Sie die Indexierung einstellen für

- jeden registrierten Inhaltstyp,
- die unterschiedlichen Taxonomien,
- Autoren-Archive,
- Datums-Archive und
- Suchergebnisseiten.

Außerdem können Sie die Indexierung der gesamten Website deaktivieren. *Wichtig:* Es wird schon durch Warnmeldungen vom

Plugin angezeigt, aber hier auch der Hinweis: Wenn Sie hier versehentlich einen falschen Haken setzen und beispielsweise die Indexierung Ihrer Seiten verbieten, kann das erhebliche Auswirkungen auf die Findbarkeit Ihrer Website in Suchmaschinen haben.

Verfolgung

Hier können Sie für dieselben Inhaltsarten die Einstellung für `nofollow` vornehmen. Diese Angabe gibt an, ob Suchmaschinen-Bots den Links auf einer Seite folgen sollen.

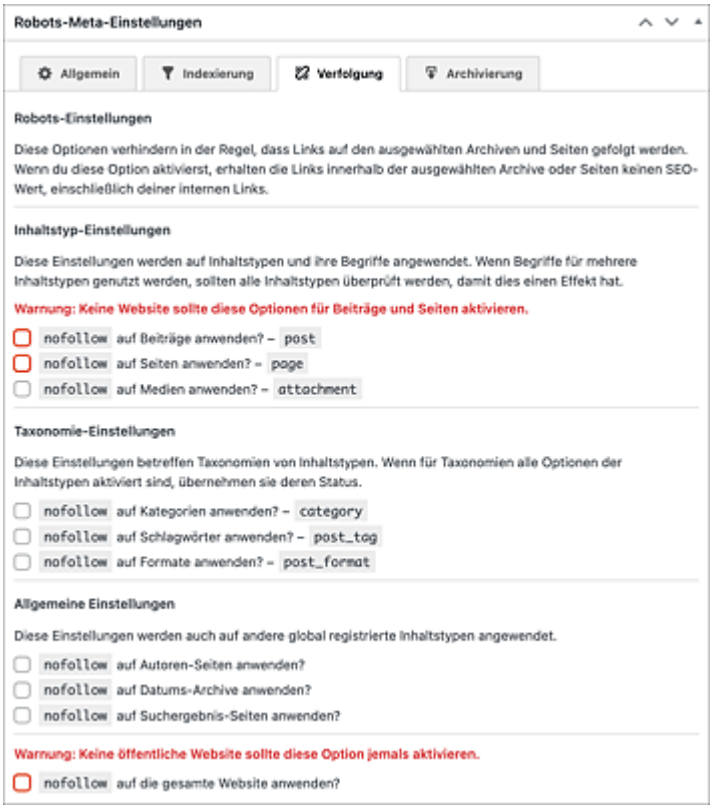


Abbildung 20.24 Im »Verfolgung«-Tab finden Sie die entsprechenden Einstellungen von der »Noindex«-Seite für den »nofollow«-Wert.

Archivierung

In dem Tab Archivierung können Sie schließlich die Ausgabe des `noarchive`-Werts für den Robots-Meta-Tag steuern. Mit der Angabe `noarchive` können Sie angeben, dass beispielsweise Google keine gecachte Version der Seite in den Suchergebnissen anbieten soll.

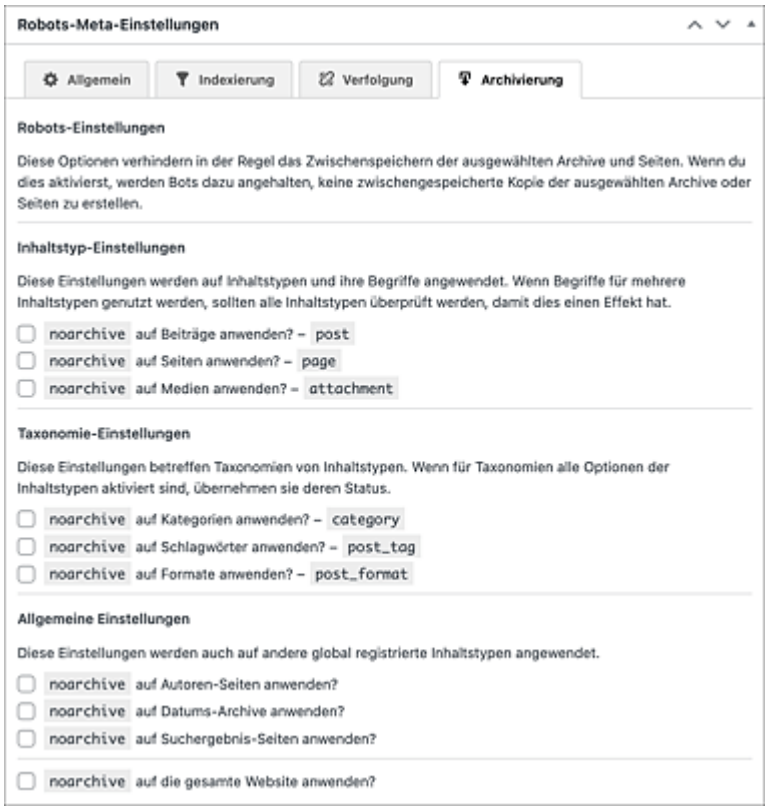


Abbildung 20.25 Hier können Sie die Einstellungen für den Wert »noarchive« anpassen.

20.1.8 Webmaster-Meta-Einstellungen

Suchmaschinen wie Google und Bing haben Tools, mit denen Sie Einblicke in die Performance Ihrer Website in den Suchergebnissen nehmen können. So können Sie zum Beispiel schauen, für welche Keywords welche Inhalte gefunden werden et cetera.

Wenn Sie Ihre Website bei den Tools eintragen, müssen Sie bestätigen, dass die Website Ihnen gehört. Das kann über einen Code passieren, der im `head` der Website eingebunden wird. Diese

Codes können Sie einfach in der Box Webmaster-Meta-Einstellungen hinterlegen.

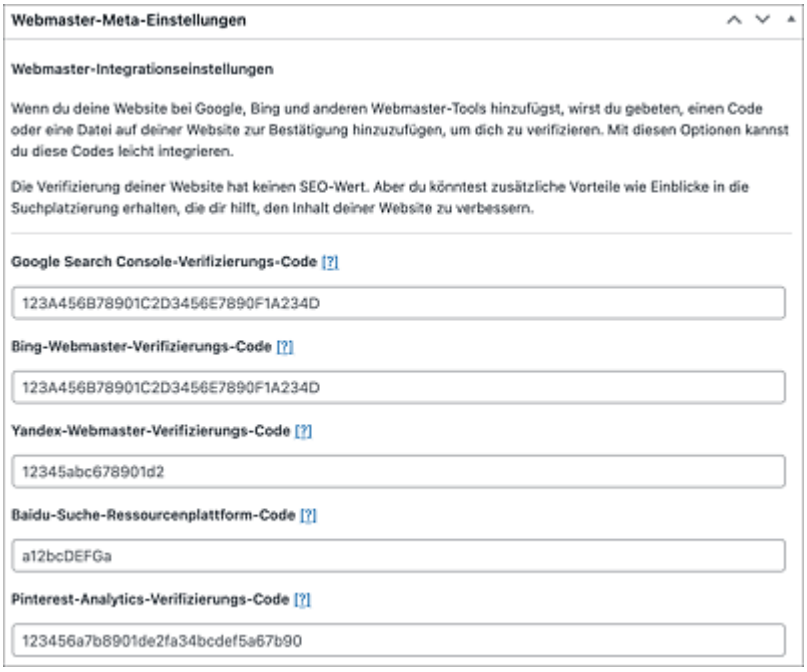


Abbildung 20.26 Wenn Sie sich bei einem oder mehreren der Webmaster-Tools von Suchmaschinen angemeldet haben, können Sie hier die Bestätigungscodes eingeben.

20.1.9 Sitemap-Einstellungen

Hier finden Sie Einstellungsmöglichkeiten, um die Inhalte der Sitemap zu verändern, die von Suchmaschinen dafür genutzt werden kann, schneller neue Inhalte zu finden. Die Einstellungen bestehen aus den Tabs Allgemein, Robots.txt, Metadaten und Erscheinungsbild (abhängig von den Einstellungen).

Allgemein

Im Tab Allgemein können Sie einstellen, ob Sie eine Optimierte Sitemap ausgeben möchten. In der optimierten Variante sind nur die neueren Inhalte enthalten und keine Archive, da kleinere Sitemaps schneller gecrawlt werden. Über die Begrenzung der Sitemap-Abfrage können Sie

eine Maximalanzahl von Elementen einstellen, die pro Abfrage aus der Datenbank ausgelesen werden. Wenn Sie Probleme mit der Sitemap haben, kann es helfen, den Wert zu verringern.

Wenn Sie die optimierte Sitemap aktiviert haben, können Sie in den Transient-Zwischenspeicher-Einstellungen angeben, ob die Ausgabe in der Datenbank zwischengespeichert werden soll und ob die Sitemap über den WP-Cron vorgeneriert werden soll.

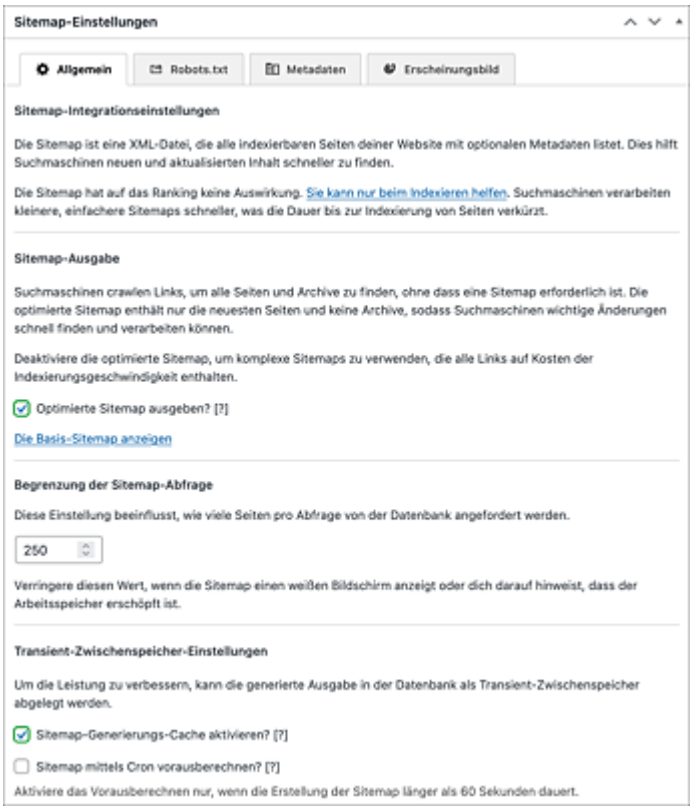


Abbildung 20.27 Hier können Sie allgemeine Einstellungen für die Sitemap vornehmen.

Robots.txt

Hier können Sie einstellen, ob die Sitemap in der Datei *robots.txt* angegeben werden soll, damit die Suchmaschinen direkt darauf hingewiesen werden, dass es eine Sitemap gibt und wo sie zu finden ist.

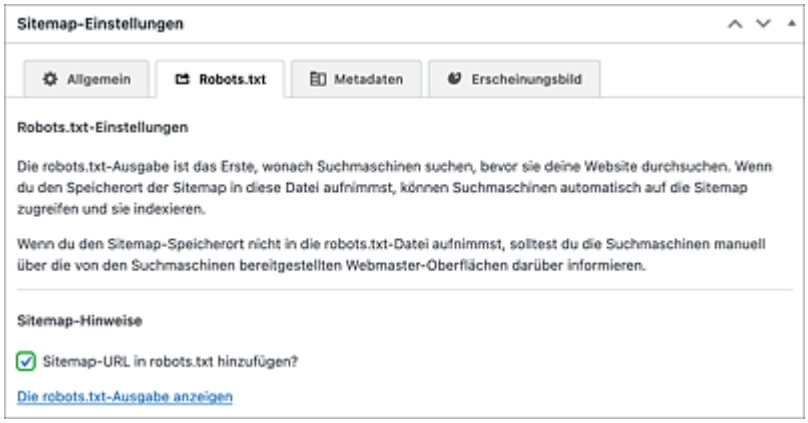


Abbildung 20.28 Die Sitemap kann in der robots.txt verlinkt werden.

Metadaten

Über das `lastmod`-Element können Sie für Inhalte das letzte Aktualisierungsdatum mit in der Sitemap ausgeben. So können Suchmaschinen darauf hingewiesen werden, wenn sich Inhalte verändert haben, damit sie hoffentlich schnell noch mal vorbeischauen, um die neue Version zu indexieren.



Abbildung 20.29 Sie können optional den Zeitpunkt der letzten Aktualisierung eines Inhalts in die Sitemap aufnehmen.

Erscheinungsbild

Normalerweise verlaufen sich Besucher*innen nicht auf die Sitemap, weshalb es auch ziemlich egal ist, wie die Sitemap gestaltet ist. Trotzdem können Sie, wenn die optimierte Sitemap-Ausgabe aktiv ist, über den Tab ERSCHEINUNGSBILD ein bisschen das Styling der Sitemap anpassen und auch ein Logo anzeigen.

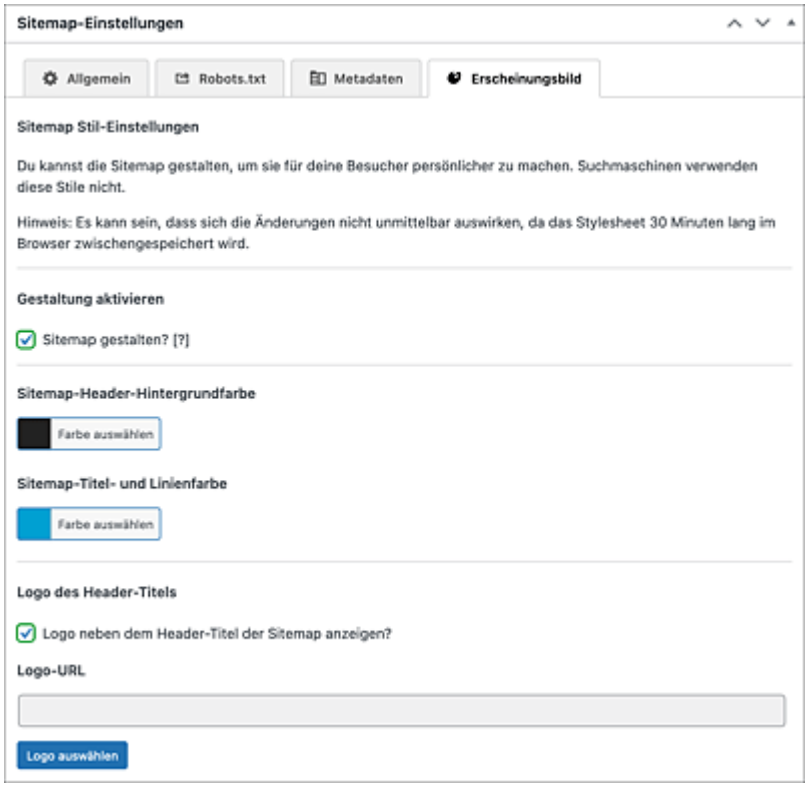


Abbildung 20.30 Hier können Sie die Sitemap ein bisschen stylen, falls sich mal menschliche Besucher*innen darauf verirren sollten.

20.1.10 Feed-Einstellungen

WordPress generiert unterschiedliche RSS-Feeds, beispielsweise für die Blogbeiträge. Mit diesen Feeds können Ihre Inhalte in Feed-Readern abonniert und gelesen werden. An sich ist es gute Praxis, wenn Sie den gesamten Inhalt im Feed zugänglich machen, damit die Nutzer*innen von Feed-Readern ihre Reader nicht verlassen müssen.

Theoretisch ist es über die Feeds aber auch möglich, den Inhalt recht einfach automatisiert auf anderen Websites zu veröffentlichen, was nicht nur insgesamt ärgerlich ist, sondern im SEO-Kontext zu Problemen mit doppeltem Inhalt führen kann. Wenn Sie so etwas

feststellen, können Sie die Feed-Einträge in Textauszüge umwandeln und den Link zur Quelle hinzufügen.

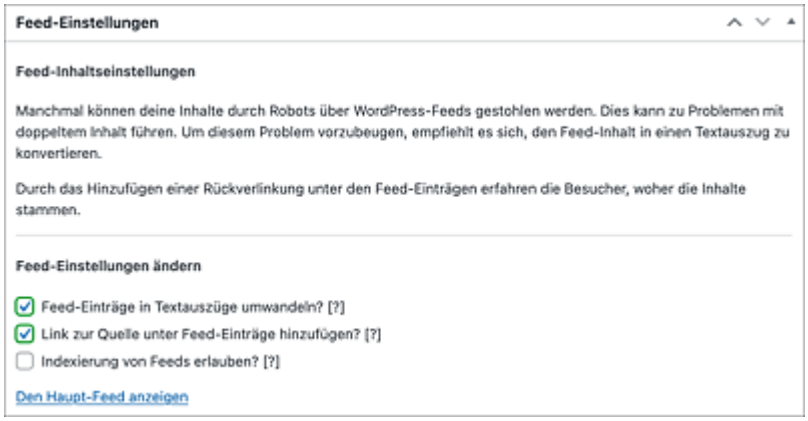


Abbildung 20.31 In den Feed-Einstellungen haben Sie Optionen für den Inhalt der RSS-Feeds, die WordPress für Ihre Website generiert.

20.1.11 Einstellungsmöglichkeiten im Editor

Nachdem Sie die Grundeinstellungen vorgenommen haben, können Sie pro Beitrag oder Seite noch Anpassungen vornehmen. Wenn Sie eine Seite im Editor öffnen, finden Sie unten einen Bereich »SEO-Einstellungen« (siehe Abbildung 20.32).

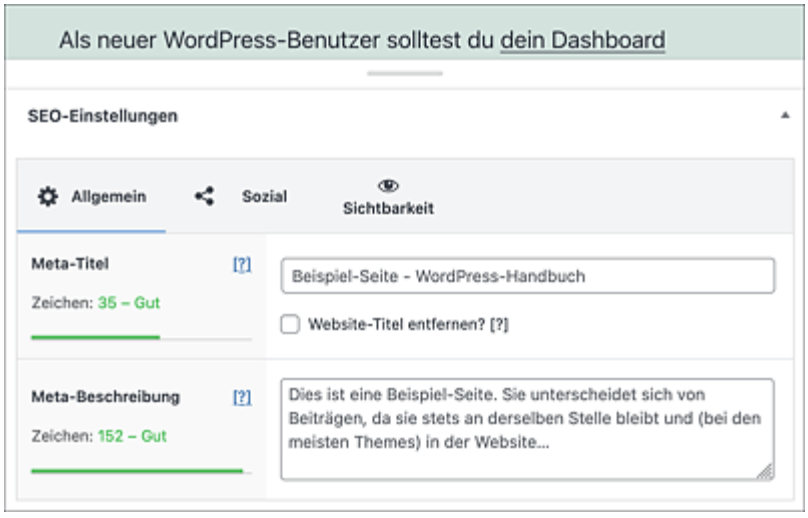


Abbildung 20.32 Unter dem Inhaltsbereich im Editor können Sie verschiedene SEO-Einstellungen auf Beitrags- beziehungsweise Seitenebene anpassen.

- Allgemein
  Hier können Sie den Titel und die Meta-Beschreibung für den Inhalt setzen. Der Titel wird sonst vom Beitrags- oder Seitentitel nach den entsprechenden Einstellungen in den Plugin-Optionen gebildet. Die Meta-Beschreibung kann durch Aktivieren in den Optionen automatisch beim Seitenaufruf generiert werden, mit dem Feld Meta-Beschreibung können Sie selbst einen Wert eingeben.
- Sozial
  In diesem Tab können Sie die Infos wie Titel, Beschreibung und Bild für die sozialen Medien anpassen.
- Sichtbarkeit
  Erlaubt das Setzen einer kanonischen URL, was hilfreich ist, wenn Sie beispielsweise einen Beitrag für eine andere Online-Publikation geschrieben haben, ihn aber auf Ihrem eigenen Blog auch veröffentlichen dürfen. Dann geben Sie hier die URL zu dem Beitrag bei dem anderen Medium an, damit es keine Probleme mit doppelten Inhalten gibt.

  Darüber hinaus können Sie im Sichtbarkeit-Tab die Robots-Meta-Einstellungen anpassen sowie die für das Erscheinen in Website-Suchergebnissen. Außerdem können Sie den Inhalt per 301-Weiterleitung auf eine andere URL weiterleiten lassen.

20.1.12 Erweiterungen und Bezahlfunktionen

Das SEO-Framework bietet unterschiedliche Erweiterungen an, die Sie unter *https://theseoframework.com/extensions/* finden. Dabei sind beispielsweise Lösungen zur Ausgabe von strukturierten Daten für Ihre Blogbeiträge oder Hilfsmittel für das Schreiben von Inhalten für bestimmte Fokus-Keywords.

Viele der Erweiterungen können Sie nur nutzen, wenn Sie einen der Bezahlpläne für das Plugin abschließen. Diese starten aber schon bei sieben Dollar monatlich, wenn Sie also die Funktionen viel nutzen, rechnet sich das recht schnell.

20.2 Viele SEO-Maßnahmen in einem Plugin mit »Yoast SEO«

Das zweite Plugin, das Sie hier kennenlernen, ist Yoast SEO. Dieses Plugin bringt anders als »The SEO Framework« neben den grundlegenden Einstellungsmöglichkeiten und Funktionen für technisches SEO auch Analyse-Tools für Ihre Inhalte mit. Teile davon sind in der kostenlosen Version verfügbar, für andere muss die Bezahlversion gekauft werden.

Plugin	Yoast SEO
Verbreitung	10.000.000+
Download	*https://de.wordpress.org/plugins/wordpress-seo/*
Zweck	ein mächtiges SEO-Plugin, das von Haus aus auch schon einige Funktionen zur Analyse der Inhalte mitbringt
Komplexität	++

20.2.1 Der Konfigurationsassistent

Nach der ersten Aktivierung wird Ihnen angeboten, die grundlegenden Einstellungen als Teil eines Einführungsdialogs zu vorzunehmen. Starten Sie den Assistenten, und klicken Sie den Upsell-Hinweis weg, gelangen Sie in eine Ansicht für die erste Konfiguration, wie in Abbildung 20.33 zu sehen.

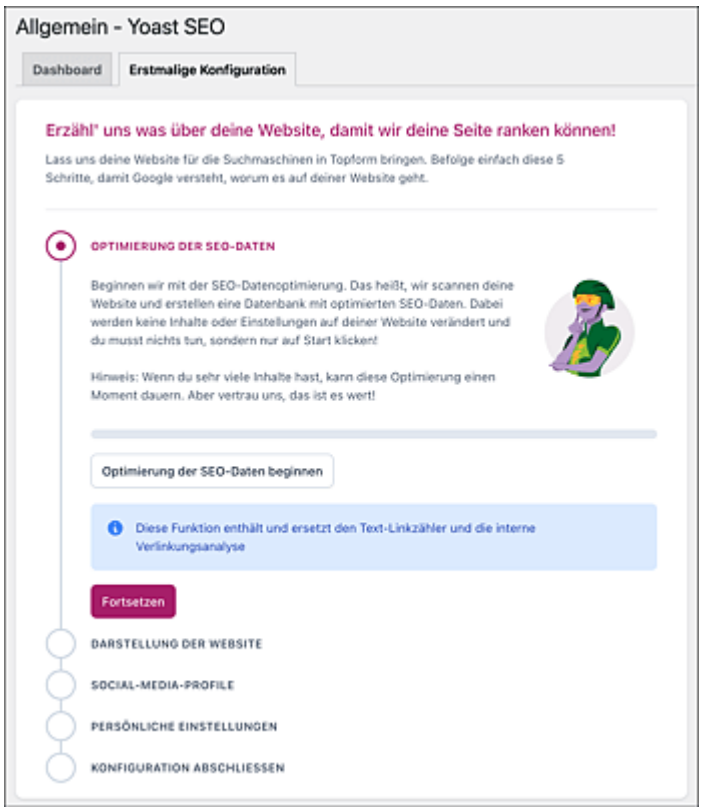


Abbildung 20.33 Wenn Sie Yoast SEO aktivieren, wird ein Konfigurationsassistent gestartet.

Im ersten Schritt des Assistenten werden die Inhalte der Website gescannt, und es wird eine SEO-Datenbank erstellt. Wenn die Website viele Inhalte hat, kann das eine Zeit dauern, Sie können den Assistenten aber währenddessen weiter durchführen.

Im zweiten Schritt werden Informationen zur Darstellung der Website in der Suche abgefragt (siehe Abbildung 20.34). So müssen Sie angeben, ob die Website eine Firma oder Person repräsentiert und wie die Website und die Firma oder Person heißen. Zudem geben Sie das Logo der Organisation oder Ihren Avatar/Ihr persönliches Logo an.

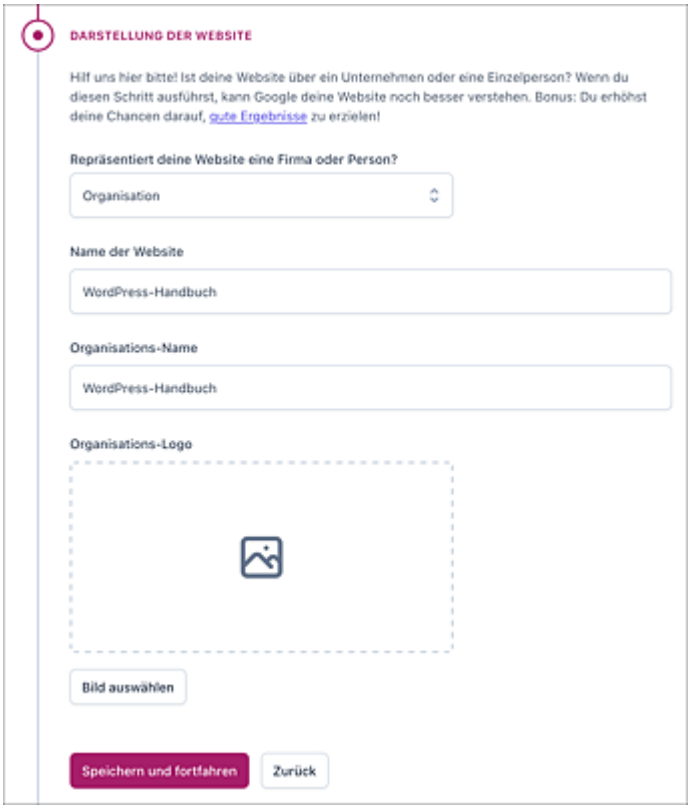


Abbildung 20.34 Im zweiten Schritt des Assistenten geben Sie Informationen zur Website an.

Im nächsten Schritt können Sie Links zu Ihren Social-Media-Profilen oder denen der Organisation einfügen (siehe Abbildung 20.35). Im darauffolgenden Schritt können Sie sich optional für einen Yoast-SEO-Newsletter anmelden und anonyme Informationen mit Yoast teilen, die zur Verbesserung des Produkts genutzt werden sollen.



Abbildung 20.35 Im Einführungsdialog können Sie Social-Media-Profile angeben.

Anschließend wechseln Sie über einen Link auf der letzten Seite des Assistenten in das SEO-Dashboard, in dem Probleme und Benachrichtigungen angezeigt werden, wenn es welche gibt.

20.2.2 Einstellungen

Die Einstellungen von Yoast SEO finden Sie über das linke Admin-Menü unter Yoast SEO • Einstellungen. Die Einstellungen sind in unterschiedliche Seiten aufgeteilt, Sie landen zunächst auf der Seite Allgemein • Website-Funktionen.

Allgemein • Website-Funktionen

Auf dieser Seite finden Sie eine lange Liste an Funktionen, die Yoast mitbringt und die Sie hier aktivieren oder deaktivieren können. Funktionen, die sich auf einzelne Inhalte beziehen, wie die SEO- oder Lesbarkeits-Analyse, können Sie nach Aktivierung in der Bearbeitungsansicht von Seiten und Beiträgen über die Box von Yoast nutzen, die unter dem Editor angezeigt wird.

- Schreiben • SEO-Analyse
  Tool zur Analyse der SEO-Freundlichkeit von Inhalten
- Schreiben • Lesbarkeits-Analyse
  gibt Tipps zur Verbesserung der Lesbarkeit von Inhalten

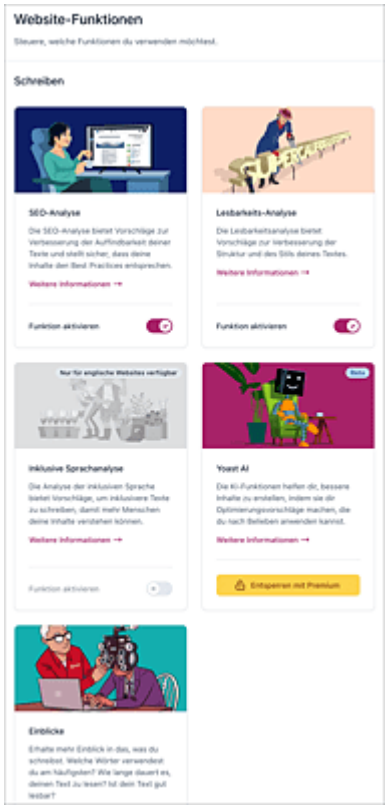


Abbildung 20.36 Hier können Sie unterschiedlichste Funktionen von Yoast aktivieren oder deaktivieren.

- Schreiben • Inklusive Sprachanalyse
  funktioniert nur für englische Websites und analysiert, wie inklusiv Texte geschrieben wurden
- Schreiben • Yoast AI – Bezahlversion
  Funktion, die Vorschläge zur Optimierung von Inhalten bietet
- Schreiben • Einblicke
  zeigt Dinge an wie die Lesezeit und Anzahl der Wörter und in der Bezahlversion markante Wörter

- Website-Struktur • Cornerstone-Inhalt
  Sogenannte Cornerstone-Inhalte sind die für Sie wichtigsten Seiten oder Beiträge, die Sie besonders gut in den Suchmaschinen platzieren möchten. Dieses Tool lässt Sie Cornerstone-Inhalte markieren und hilft dabei, den Inhalt entsprechend hervorzuheben.
- Website-Struktur • Text Link Zähler
  zählt die internen Links von und zu Inhalten
- Website-Struktur • Vorschläge zur internen Verlinkung – Bezahlversion
  gibt Ihnen Vorschläge, welche relevanten Inhalte Sie verlinken könnten
- Teilen über soziale Netzwerke • Open Graph-Daten
  steuert, ob Open-Graph-Daten zur verbesserten Vorschau von Links auf Social Media ausgegeben werden
- Teilen über soziale Netzwerke • X Kartendaten
  steuert die Ausgabe von Tags speziell für X/Twitter
- Teilen über soziale Netzwerke • Slack-Freigabe
  erweitert die Vorschau eines bei Slack geteilten Links um Autorenzeile und voraussichtliche Lesedauer
- Werkzeuge • Adminbar-Menü
  Hier können Sie einstellen, ob in der Adminbar das Yoast-SEO-Menü angezeigt werden soll, erkennbar an dem Logo des Plugins.
- APIs • REST-API-Endpunkt
  bietet die Yoast-Daten über die REST-API an, sodass beispielsweise auch Websites, die WordPress nur als Backend, aber nicht für das Frontend nutzen (Headless WordPress), einfach auf die Inhalte zugreifen können

- APIs • XML-Sitemaps
  aktiviert die XML-Sitemap-Funktion von Yoast
- APIs • IndexNow – Bezahlversion
  benachrichtigt Suchmaschinen bei der Veröffentlichung/Bearbeitung/Löschung von Inhalten

Allgemein • Website-Grundlagen

Hier können Sie Dinge anpassen wie den von Yoast genutzten Namen der Website (unabhängig vom Website-Titel, der im Admin-Menü unter Einstellungen • Allgemein festgelegt ist) und das Trennzeichen für die einzelnen Bestandteile des Werts für das `title`-Element.



Abbildung 20.37 Auf der Einstellungsseite »Website-Grundlagen« können Sie unter anderem das Trennzeichen für den Titel ändern.

- Website-Informationen • Name der Website/Alternativer Name der Website
  Hier können Sie den Namen der Website angeben, wenn er vom Website-Titel abweichen soll, und optional eine kürzere Version des Namens.
- Website-Informationen • Untertitel/Slogan
  Hier können Sie den Website-Untertitel ändern. Dieses Feld hat Auswirkungen auf den unter Einstellungen • Allgemein festgelegten Website-Untertitel.
- Website-Informationen • Titel-Trennzeichen
  das Trennzeichen für die einzelnen Bestandteile des Titels
- Website-Informationen • Website-Bild
  ein Fallback-Bild für den Fall, dass für einen Inhalt keine Bilder festgelegt sind
- Website-Voreinstellungen • Erweiterte Einstellungen für Autoren einschränken
  Mit dieser Checkbox können Sie einstellen, ob nicht nur Admins und Redakteur*innen Zugriff auf gewisse Abschnitte in den Yoast-Einstellungen einzelner Beiträge und Seiten haben, sondern auch Autor*innen.
- Website-Voreinstellungen • Nutzungs-Tracking
  aktiviert das Teilen von Daten mit Yoast zum Zweck der Verbesserung des Plugins
- Website-Richtlinien – Bezahlversion
  Hier können Sie unterschiedliche Seiten mit Richtlinien auswählen, wie beispielsweise die Ethikrichtlinien. Das soll die Vertrauenswürdigkeit von Sites bei Google stärken.

Allgemein • Darstellung der Website

Hier tragen Sie Informationen zu der Website ein, die für den *Knowledge Graph* von Google gedacht sind. Das sind zusätzliche Informationen direkt auf der Suchergebnisseite, die auf der Desktop-Version rechts von der Liste angezeigt werden.

- Organisation/Person
  Hier legen Sie fest, ob es bei der Website um eine Organisation oder eine Person geht.

Wenn Organisation ausgewählt wurde, sind die folgenden Einstellungen verfügbar.

- Organisation • Organisations-Name/Alternativer Organisationsname
  der Name der Organisation und eine alternative kürzere Variante
- Organisation • Organisations-Logo
  Logo der Organisation
- Andere Profile
  Hier können Links zu Profilen auf Social Media hinzugefügt werden.
- Zusätzliche Informationen zur Organisation – Bezahlversion
  verschiedene Felder, um etwa eine Beschreibung und Kontaktdaten für die Organisation zu hinterlegen
- Identifikatoren der Organisation
  Identifikationsnummern wie die Steuer- und Umsatzsteuer-ID

Wurde Person ausgewählt, gibt es die folgenden Einstellungen:

- Persönliche Daten • Benutzer auswählen
  Hier wählen Sie den WordPress-User aus, der hinter der Site steht.
- Persönliche Daten • Persönliches Logo oder persönlicher Avatar
  Logo oder Avatar der Person, die im Auswahlfeld ausgewählt wurde

Allgemein • Website-Verbindungen

Unter dem Punkt Website-Verbindungen bietet Yoast Felder für die Codes zur Verifizierung bei Services wie den Bing Webmaster Tools oder der Google Search Console an.

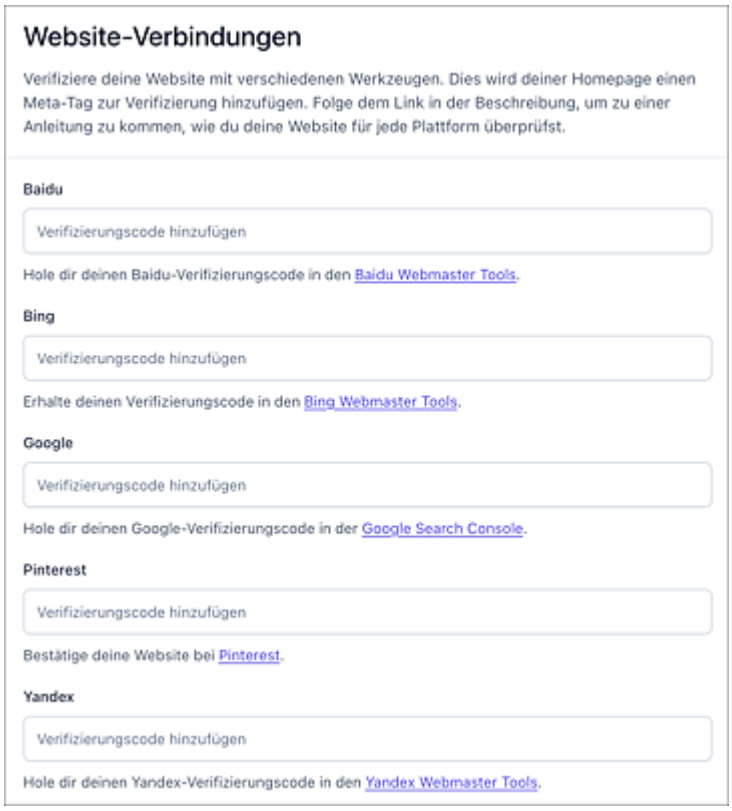


Abbildung 20.38 Unter »Website-Verbindungen« können Sie Codes für die Webmaster-Tools unterschiedlicher Suchmaschinen eintragen.

Die Codes werden dann als Metadaten auf der Website ausgegeben, sodass die Anbieter der Services nachprüfen können, dass Ihnen die Website gehört oder Sie zumindest Zugriff auf die Site haben, die Sie bei den Services nutzen möchten.

Inhaltsarten • Homepage

- Darstellung in der Suche • SEO-Titel und SEO-Beschreibung
  Legen Sie den Titel und die Beschreibung fest, die für die

Startseite von Suchmaschinen genutzt werden soll.

- Social-Media-Auftritt
  Hier können Sie ein Bild sowie Titel und Beschreibung angeben, die beim Teilen des Links zur Startseite in Social Media angezeigt werden sollen.



Abbildung 20.39 Hier nehmen Sie Einstellungen für die Startseite vor.

Inhaltsarten • Beiträge

- Darstellung in der Suche
  Hier können Sie festlegen, ob Beiträge in den Suchergebnissen angezeigt oder ausgeschlossen werden sollen. Außerdem können Sie ein Format für Meta-Titel und Meta-Beschreibung angeben, indem Sie Variablen nutzen, um beispielsweise den Titel eines Beitrags und das Trennzeichen auszugeben.

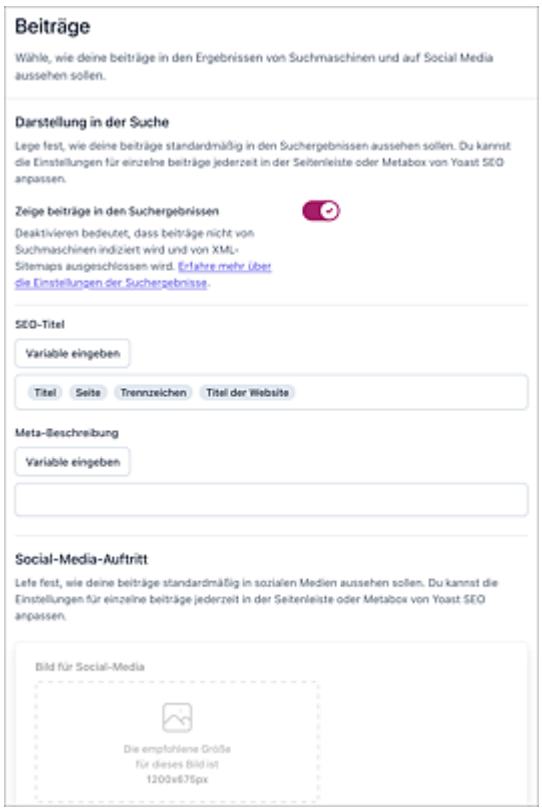


Abbildung 20.40 Hier können Sie Einstellungen für die Darstellung der Beiträge in den Suchergebnissen vornehmen.

- Social-Media-Auftritt – Bezahlversion
  ermöglicht es Ihnen, Informationen für die Anzeige von Beiträgen in Social Media anzugeben, wie etwa das Titelformat
- Schema
  Hier können Sie Standardwerte für Seiten- und Artikeltyp für die Schema.org-Auszeichnung von Beiträgen festlegen.
- Zusätzliche Einstellungen
  Dieser Bereich erlaubt Ihnen, die Werkzeuge von Yoast im Beitragseditor auszublenden und in der Bezahlversion auch individuelle Felder in die Seitenanalyse mit einzubeziehen.

Inhaltsarten • Seiten

Unter dem Punkt Seiten gibt es die gleichen Einstellungen wie für Beiträge, nur diesmal für den Inhaltstyp der Seiten.

Kategorien und Schlagwörter • Kategorien

Hier können Sie einstellen, ob und wie Kategorieseiten in den Suchergebnissen angezeigt werden.

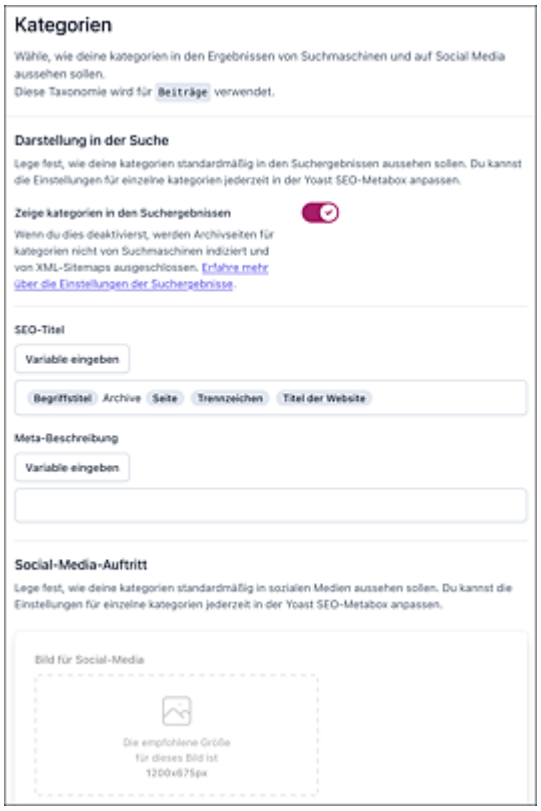


Abbildung 20.41 Auf der Einstellungsseite für Kategorien steuern Sie die Darstellung von Kategorien in Suchergebnissen.

Dafür haben Sie ähnliche Einstellungsmöglichkeiten für Titel und Beschreibung sowie die Social-Media-Angaben wie bei Beiträgen und Seiten. Zusätzlich können Sie die Basis für Kategorien (standardmäßig *category*) aus der URL entfernen, sodass die Kategorie mit dem Slug `allgemein` nicht mehr unter */category/allgemein* zugänglich ist, sondern unter */allgemein*.

Kategorien und Schlagwörter • Schlagwörter

Hier finden Sie bis auf die Basis in der URL die gleichen Einstellungen für Schlagwörter wie für Kategorien.

Erweitert • Crawl-Optimierung

- Unerwünschte Metadaten entfernen
  Hier können Sie eine Reihe von Meta-Angaben entfernen, die standardmäßig in den `head` eingebunden werden, beispielsweise Links zu REST-API-Endpunkten und einen Tag, der angibt, mit welcher WordPress-Version die Website läuft.
- Unerwünschte Inhaltsformate deaktivieren
  Hier können Sie einzelne oder alle RSS-Feeds deaktivieren, die WordPress generiert. RSS-Feeds sind eine bequeme Möglichkeit, Inhalte unterschiedlicher Quellen in Feed-Readern zu lesen, deshalb sollten Sie das nicht leichtfertig tun.
- Ungenutzte Ressourcen entfernen
  Unter diesem Punkt haben Sie die Möglichkeit, Fallback-Skripte für Emojis für ältere Browser zu entfernen und das Crawling der REST-API über die *robots.txt* zu verbieten.
- Unerwünschte Bots blockieren – teilweise Bezahlversion
  In der kostenlosen Version können Sie hier verhindern, dass Ihre Website durch den Google-Ads-Bot gecrawlt wird. In der Bezahlversion haben Sie Optionen für unterschiedliche weitere Bots, wie den von ChatGPT.
- Bereinigung der internen Website-Suche
  Hier können Sie unterschiedliche Einstellungen vornehmen, um zu verhindern, dass die interne Suche in Suchergebnissen auftaucht oder von SEO-Spammern als Angriffspunkt genutzt wird.

- Erweitert: URL-Bereinigung
  In diesem Bereich können Sie einstellen, ob nicht registrierte URL-Parameter von den URLs entfernt werden sollen. Diese Funktionen sollten Sie nur nutzen, wenn Sie wissen, was Sie tun.



Abbildung 20.42 Hier können Sie Funktionen deaktivieren, die Sie nicht nutzen.

Erweitert • Breadcrumbs

- Aussehen der Breadcrumbs
  Hier können Sie unterschiedliche Einstellungen für das Aussehen der Breadcrumb vornehmen, wie das Trennzeichen oder den Linktext für die Startseite.
- Breadcrumbs für Inhaltstypen
  Sie können optional bei Beiträgen Taxonomien mit in der Breadcrumb ausgeben.

- Breadcrumbs für Taxonomien
  Hier können Sie Einstellungen für Taxonomie-Breadcrumbs vornehmen.
- Wie du Breadcrumbs für Dein Theme einbinden kannst
  Verweis auf die Doku zum Anzeigen der Breadcrumb sowie Einstellung zur Aktivierung der Breadcrumb



Abbildung 20.43 Breadcrumbs helfen bei der Navigation innerhalb einer Website-Struktur und können Suchmaschinen bei der Erkennung der Struktur unterstützen.

Erweitert • Autor-Archiv

Auf der Einstellungsseite Erweitert • Autor-Archiv finden Sie Optionen für die Archive, die Beiträge einer bestimmten Person anzeigen. So können Sie die Archive komplett deaktivieren, sodass es auf die Startseite umleitet. Außerdem können Sie – ähnlich wie für Beiträge, Seiten und die Taxonomien – Einstellungen für die

Anzeige in den Suchergebnissen und Social Media (das nur in der Bezahlversion) vornehmen.

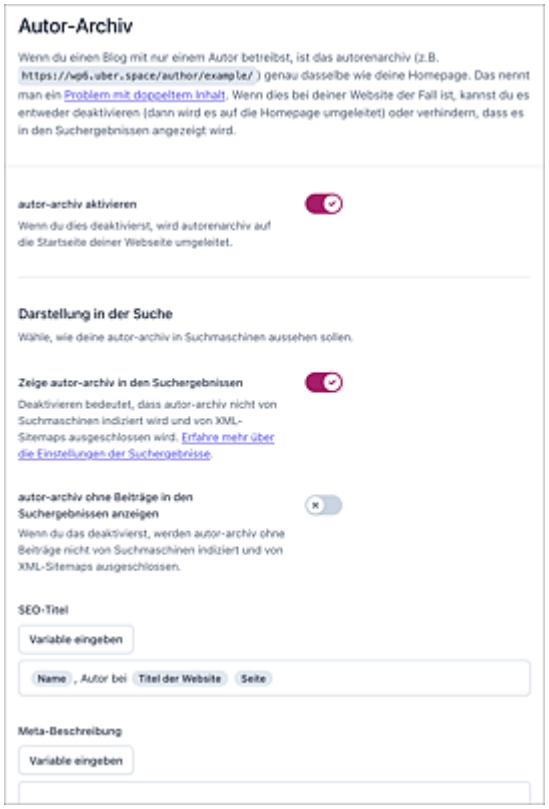


Abbildung 20.44 Hier können Sie Einstellungen für die Autor-Archive vornehmen.

Erweitert • Datums-Archive

In dieser Ansicht haben Sie bis auf wenige Ausnahmen die gleichen Optionen, die Sie eben bei der Seite für Autoren-Archive gesehen haben, nur beziehen sie sich auf die Archive, die Beiträge aus einem bestimmten Zeitraum anzeigen, beispielsweise die aus März 2024.

Erweitert • Format Archive

Format-Archive zeigen die Beiträge mit einem bestimmten Post-Format an. Hier haben Sie die gleichen Einstellungsmöglichkeiten wie für die Datums-Archive.

Erweitert • Spezielle Seiten

Unter Spezielle Seiten haben Sie die Option, den Titel für die Suchergebnisseiten sowie die 404-Fehlerseite anzupassen, die angezeigt wird, wenn ein Inhalt nicht gefunden wurde.

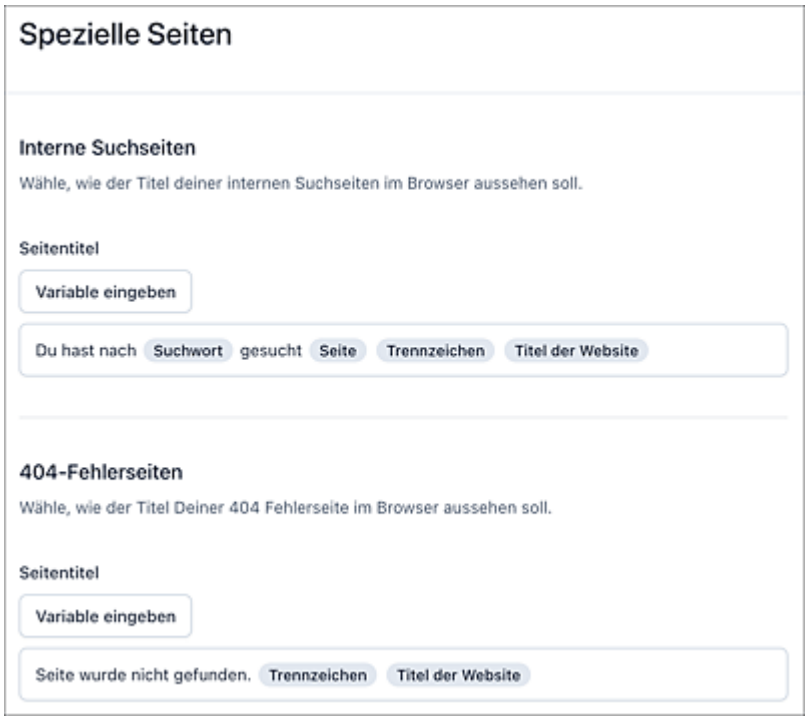


Abbildung 20.45 Hier können Sie Seitentitel für die Suchergebnisseiten und die 404-Fehlerseite vergeben.

Erweitert • Medienseiten

Standardmäßig werden von Yoast die Anhangseiten deaktiviert, die WordPress für hochgeladene Medien erstellt. Stattdessen wird auf das Medium selbst umgeleitet.

Hier können Sie die Anhangseiten aktivieren, die Darstellung in der Suche sowie die Schema-Daten anpassen und aktivieren, dass die SEO-Tools im Editor für die Medien angezeigt werden.

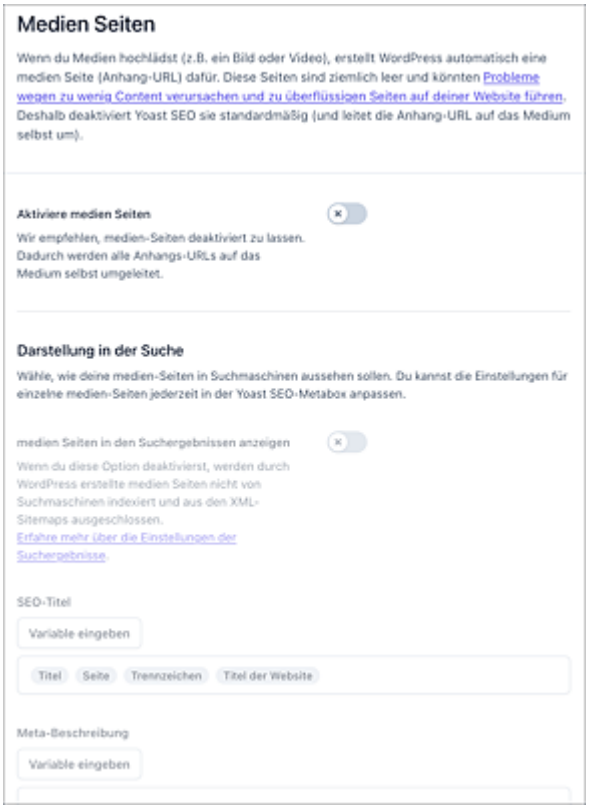


Abbildung 20.46 WordPress generiert für hochgeladene Medien eigene Anhangseiten, für die Sie hier Einstellungen vornehmen können.

Erweitert • RSS

Im RSS-Bereich haben Sie die Option, sowohl vor als auch nach jedem Eintrag in den RSS-Feeds Inhalte anzuzeigen, um zum Beispiel zurück auf das Blog zu verlinken, um die Originalquelle anzugeben.



Abbildung 20.47 Hier können Sie Inhalte vor oder nach jedem Beitrag im Feed einfügen.

20.2.3 Integration

Unter Yoast SEO • Integration können Sie unterschiedliche Integrationen für Yoast aktivieren, etwa um bestimmte Inhaltstypen wie Podcasts mit den entsprechenden Schema-Daten auszuzeichnen.

20.2.4 Werkzeuge

Wenn Sie Yoast SEO • Werkzeuge aufrufen, finden Sie eine Reihe von nützlichen Werkzeugen, die Ihnen das Leben erleichtern können.

- Import und Export
  die Möglichkeit, die Yoast-SEO-Einstellungen zu exportieren und zu importieren

- Datei-Editor
  Tool zur Bearbeitung von wichtigen Dateien wie der *robots.txt*
- Massenbearbeitung
  erlaubt Ihnen die Bearbeitung von SEO-Titel und -Beschreibung von Beiträgen und Seiten in einer Listenansicht, ohne dass Sie die Bearbeitungsansicht aufrufen müssen
- SEO-Daten optimieren
  Diesen Schritt haben Sie wahrscheinlich schon während des Einrichtungsassistenten durchgeführt. Falls nicht, können Sie ihn hier starten, um die Speicherung der SEO-Daten zu optimieren.

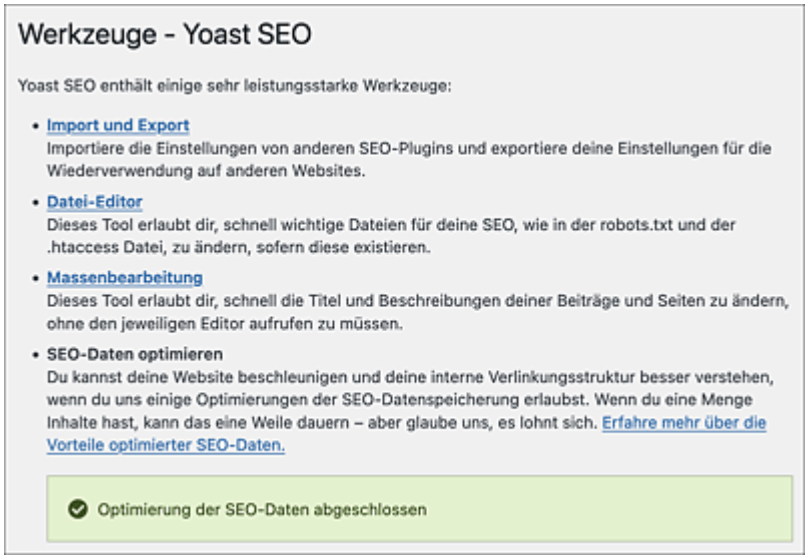


Abbildung 20.48 Yoast bringt ein paar nützliche Werkzeuge mit, beispielsweise zum Import und Export der Einstellungen.

20.2.5 Verschiedene Bezahlfunktionen

Die wichtigsten Seiten in dem Yoast-SEO-Menü haben Sie jetzt kennengelernt, hinter den anderen finden Sie beispielsweise Trainings-Videos, die teilweise kostenlos und teilweise Teil der Bezahlversion sind, oder die Möglichkeit, Erweiterungen für Yoast SEO zu kaufen, um etwa speziell für Local SEO zu optimieren, also

Suchmaschinenoptimierung mit lokalem Bezug. Ebenfalls der Bezahlversion vorbehalten ist die Verwaltung von Weiterleitungen.

20.2.6 Optionen im Inhaltseditor

Wenn Sie den Editor zur Bearbeitung eines Beitrags oder einer Seite öffnen, finden Sie wie bei »The SEO Framework« unter dem Editorbereich die Einstellungsmöglichkeiten für Yoast auf Beitrags- beziehungsweise Seitenebene (siehe Abbildung 20.49).

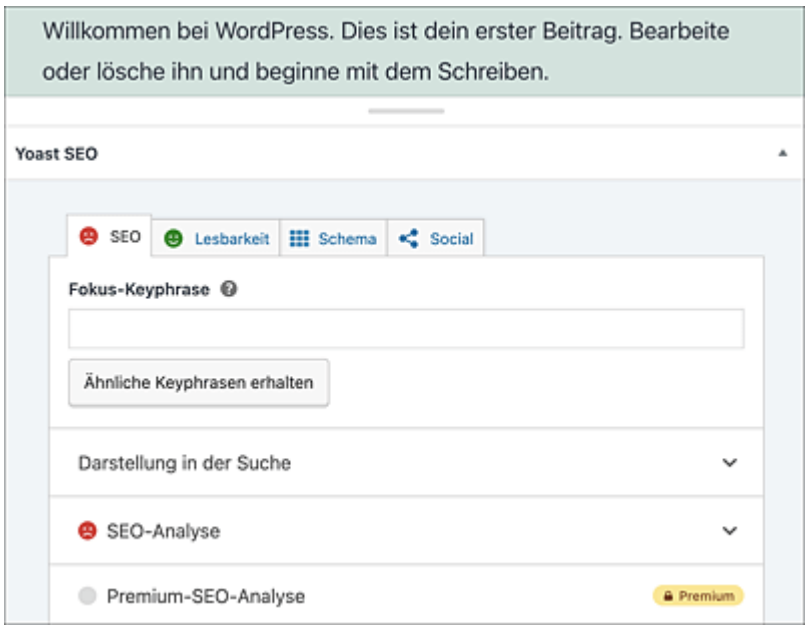


Abbildung 20.49 Unter dem Inhaltseditor finden Sie die Funktionen und Einstellungen von Yoast SEO.

Rechts in der Dokumenten-Seitenleiste finden Sie ebenfalls einen kleinen Yoast-SEO-Bereich, der den Stand der Lesbarkeits-Analyse und SEO-Analyse angibt (siehe Abbildung 20.50).

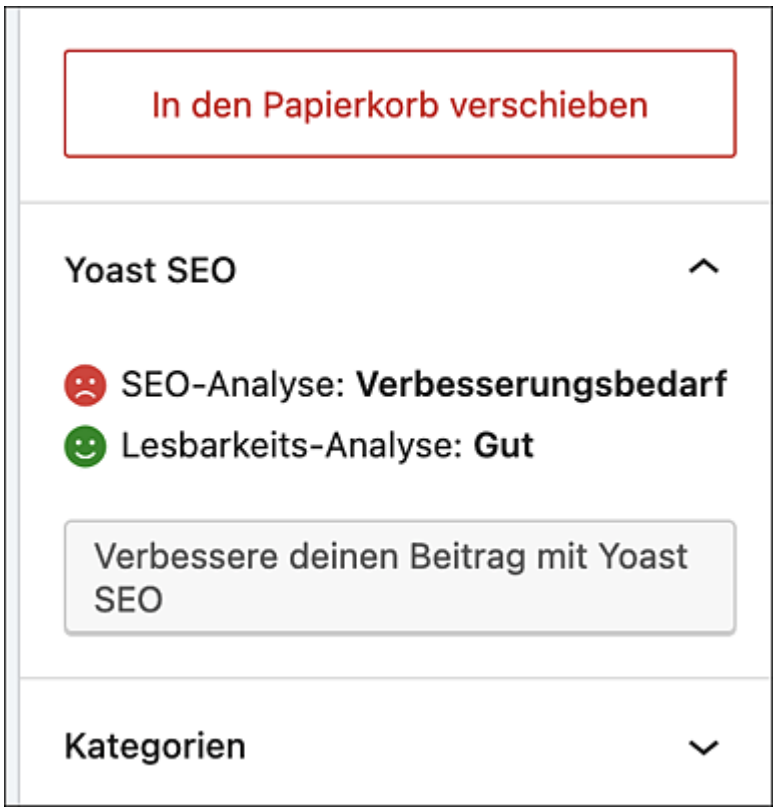


Abbildung 20.50 In der Dokumenten-Seitenleiste sehen Sie den aktuellen Stand von Lesbarkeits- und SEO-Analyse.

Die unteren Einstellungen teilen sich in die vier Tabs SEO, Lesbarkeit, Schema und Social auf. Hier wird nur auf die Punkte eingegangen, die in der kostenfreien Version verfügbar sind.

SEO

- Fokus-Keyphrase

  das Keyword oder die Phrase, für die der Inhalt in den Suchmaschinen ranken soll

- Darstellung in der Suche

  Hier können Sie Titel und Beschreibung für die Anzeige in den Suchmaschinen anpassen.

- SEO-Analyse

gibt Ihnen Hinweise, ob es Probleme im Hinblick auf SEO mit dem aktuellen Inhalt gibt, etwa ob der Text lang genug ist

- Cornerstone-Inhalt

  Hier können Sie den Beitrag oder die Seite als Cornerstone-Inhalt markieren, also als einen der wichtigsten Inhalte Ihrer Website, die in Suchmaschinen besonders gut ranken sollen.

- Erweitert

  Einstellungsmöglichkeiten für den robots-Meta-Tag, den Breadcrumb-Titel und die Canonical URL

- Einblicke

  gibt die voraussichtliche Lesezeit, die Anzahl der Wörter und eine Lesbarkeitsbewertung aus

Lesbarkeit

Hier sehen Sie eine Analyse der Lesbarkeit Ihres Beitrags- oder Seiteninhalts. Es wird beispielsweise geprüft, ob Absätze und Sätze nicht zu lang sind und genug Zwischenüberschriften genutzt werden.

Schema

Lässt Sie den Seitentyp und Artikeltyp für die Schema.org-Angaben verändern.

Social

Ermöglicht Ihnen das Setzen eines Bilds und eines gesonderten Titels und einer Beschreibung zur Nutzung von allen möglichen

Social-Media-Plattformen und zusätzlich optional spezifisch für X/Twitter.

20.3 Sitemap bei Google und Bing einreichen

Eine Sitemap für Suchmaschinen ist eine XML-Datei (oder auch mehrere) mit einer Liste der URLs Ihrer Website, die indexiert werden sollen. Die Sitemap können Sie bei den Suchmaschinen einreichen, um zu versuchen, eine schnellere Indexierung der Inhalte zu erreichen.

Seit WordPress 5.5 wird von WordPress standardmäßig eine Sitemap generiert, die unter *example.com/wp-sitemap.xml* erreichbar ist. Aber auch wenn Sie ein Plugin dafür benutzen – Yoast liefert beispielsweise eine eigene Sitemap-Funktion mit –, unterscheidet sich der Prozess des Einreichens bei Suchmaschinen nicht.

20.3.1 Einreichung bei Google über die Search Console

Öffnen Sie die URL *search.google.com/search-console*. Um die Search Console nutzen zu können, um eine Sitemap einzureichen, müssen Sie Ihre Website erst zur Search Console hinzufügen (wählen Sie den Property-Typ **URL PREFIX**). Dabei müssen Sie nachweisen, dass Sie berechtigt sind, die Website hinzuzufügen, und dafür bietet Google unterschiedliche Wege an. Einer davon ist das Einfügen eines Codes in das Markup der Website über einen Meta-Tag. Beide hier vorgestellten SEO-Plugins bieten die Möglichkeit, den Code über die Einstellungen einzufügen, wenn Sie also eins davon nutzen, ist das wohl der einfachste Weg.

Ist die Property angelegt und verifiziert, werden Sie mit der Zeit interessante Informationen zu Ihrer Website in den Suchergebnissen von Google in der Search Console vorfinden, zum

Beispiel Suchanfragen, über die Ihre Website gefunden wird (Menüpunkt Performance). Über den Menüpunkt Sitemaps können Sie sich die eingereichten XML-Sitemaps anschauen und neue Sitemaps einreichen. Kopieren Sie dafür einfach die URL der Sitemap in das Textfeld über der Liste, und klicken Sie Submit (siehe Abbildung 20.51).

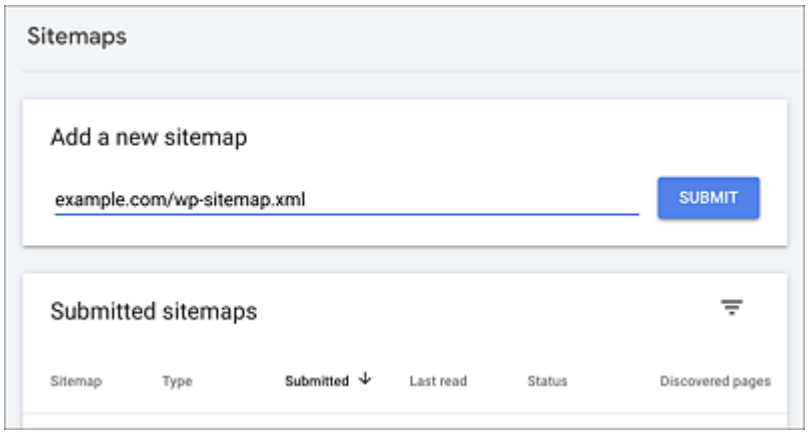


Abbildung 20.51 Die Sitemap(s) für Ihre Website reichen Sie bei Google über die Search Console ein.

Google wird die Sitemap nun prüfen und Ihnen dann hoffentlich einen Success-Status anzeigen. Bis die Seiten aber im Index von Google aufgenommen werden, dauert es etwas.

20.3.2 Einreichung bei Bing

Die Webmaster-Tools von Bing finden Sie unter *https://www.bing.com/webmasters*. Beim Hinzufügen einer Website wird Ihnen angeboten, die Sites aus der Google Search Console zu importieren, was auch die eingereichten Sitemaps einschließt.

Starten Sie also einfach den Import, und gewähren Sie Bing Zugriff auf Ihr Google-Konto, damit die in der Search Console hinterlegten Websites importiert werden können. Vor dem Import können Sie

auswählen, welche der in der Search Console eingetragenen Sites importiert werden sollen.

Wenn Sie nicht den Weg über den Import gehen möchten, fügen Sie eine neue Website in den Bing Webmaster Tools hinzu, und wählen Sie die für Sie passende Überprüfungsmethode aus. Wenn Sie eins der beiden vorgestellten SEO-Plugins verwenden, können Sie über deren Einstellungen den Code von der HTML-Meta-Tag-Methode einfügen.

Nachdem Sie die Website hinzugefügt haben, finden Sie links in der Navigation den Punkt SITEMAPS und können über den Punkt SITEMAP SENDEN oben rechts eine neue Sitemap hinzufügen (siehe Abbildung 20.52). Wie in der Search Console haben Sie in der Liste der eingereichten Sitemaps eine Spalte für den Status, in der Sie sehen können, ob die Sitemap erfolgreich verarbeitet werden konnte.

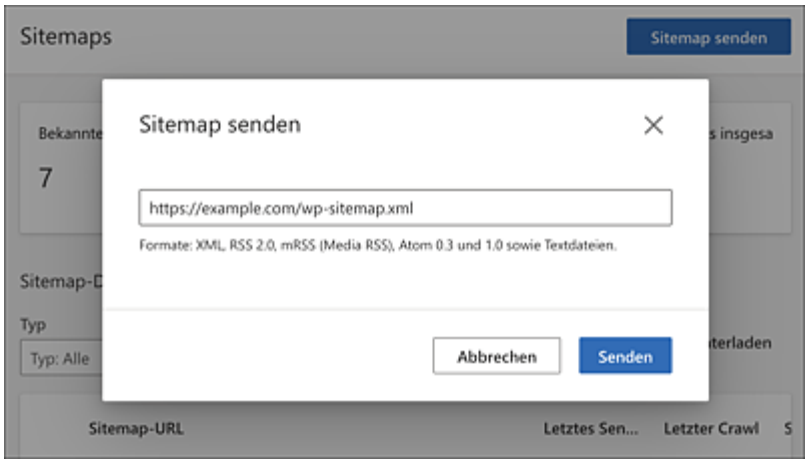


Abbildung 20.52 Hinzufügen einer Sitemap in den Bing Webmaster Tools

21 Notfallmaßnahmen

Wenn nach einer Plugin-Aktualisierung nur noch eine weiße Seite angezeigt wird oder Ihre Website gehackt wurde, finden Sie in diesem Kapitel Tipps, wie Sie vorgehen können.

Normalerweise wird auf mögliche Probleme in diesem Handbuch direkt bei den Themen eingegangen, wo sie auftreten können. Auf den folgenden Seiten werden einige Aspekte besprochen, die entweder von allgemeiner Natur oder so wichtig sind, dass es sich lohnt, sie in einem eigenen Kapitel hervorzuheben.

Zunächst erfahren Sie, was Sie tun können, wenn Hacker*innen einen erfolgreichen Angriff auf Ihre Website unternommen haben. Danach geht es um allgemeine Fehlermeldungen und Probleme, die bei der täglichen Arbeit mit WordPress auftauchen können. Natürlich können hier nicht alle möglichen Probleme besprochen werden. Wenn Sie mit Ihrer Installation ein Problem haben, für das Sie hier keine Lösung finden, werfen Sie die Suchmaschine Ihres Vertrauens an, und suchen Sie, am besten auf Englisch, nach dem Fehler. Die Wahrscheinlichkeit ist groß, dass schon jemand anders das Problem hatte.

21.1 Gehackt, was nun?

Wenn Sie den Verdacht haben oder wissen, dass Ihre Website gehackt wurde (beispielsweise durch eine Mail von Ihrem Hoster),

sollten Sie schnell tätig werden, denn im schlechtesten Fall ist Ihre Website jetzt eine Bedrohung für Ihre Besucher*innen und wird demnächst von Suchmaschinen und Browsern als gefährlich eingestuft.

21.1.1 Sofortmaßnahmen-Checkliste

Unter *https://sitecheck.sucuri.net/* können Sie einen Malware-Check für Ihre Website ausführen, um sich ein Bild davon zu machen, wie ernst die Lage ist.

Anschließend führen Sie zügig die folgenden Schritte aus, um Ihr Publikum zu schützen.

1. **Beschränken Sie den Zugriff auf die Website**
   Richten Sie einen Passwortschutz per *.htaccess* ein, wie in Abschnitt 17.1 beschrieben. Damit haben Sie schon mal dafür gesorgt, dass niemand außer Ihnen mehr auf die möglicherweise gefährliche Website zugreifen kann.
2. **Passwörter ändern**
   Ein möglicher Angriffspunkt für Hacker*innen sind einfach die »normalen« Zugänge. Ändern Sie deshalb alle Passwörter, die mit der Website zu tun haben: SFTP, Datenbank, WordPress-Benutzer*innen, Webhoster-Passwort. Damit können Sie sicherstellen, dass eventuell geknackte Passwörter nicht mehr funktionieren.
3. **Backup erstellen**
   Sichern Sie sich den aktuellen Stand der Dateien per SFTP und die Datenbank per phpMyAdmin oder mit einem anderen Tool, das Ihnen von Ihrem Webhoster zur Verfügung gestellt wird. So können Sie später in Ruhe nachforschen, wo das Problem lag,

wenn der unerlaubte Zugriff über eine Sicherheitslücke in einem Plugin oder in WordPress erfolgt ist.

4. **Website deaktivieren oder sauberes Backup einspielen**
   Wenn Sie ein Backup haben, bei dem Sie sicher sind, dass es vor dem Angriff erstellt wurde, können Sie das Backup wiederherstellen. Wenn Sie das Backup wiederhergestellt haben, aktualisieren Sie alle Bestandteile der Website auf die neueste Version. *Achtung:* Schadcode kann auch schon längere Zeit auf der Website eingeschleust sein, bevor offensichtliche Aktionen wie Spam-E-Mailversand oder Ähnliches passieren.

   Wenn Sie nicht sicher sind, ob Sie ein sauberes Backup haben, sorgen Sie dafür, dass keine Skripte mehr ausgeführt werden können, auch nicht per Cronjob oder ähnlich. Sie könnten dafür das Verzeichnis auf dem Webspace umbenennen, so sollten eventuell eingerichtete Cronjobs nicht mehr funktionieren.

21.1.2 Schadcode suchen

Nachdem Sie jetzt sichergestellt haben, dass die Website keinen Schaden mehr anrichten kann, gehen Sie auf die Suche nach dem Schadcode, um zu beurteilen, ob die Website gerettet werden kann oder neu aufgebaut werden muss. Um zu prüfen, ob und wenn ja wo Schadcode eingebaut wurde, müssen Sie die Dateien Ihrer Installation mit den Originalversionen von WordPress, den Plugins und Themes vergleichen.

Tipp: WP-CLI zum Prüfen geänderter Dateien nutzen

Wenn Sie das Glück haben, einen SSH-Zugang zu Ihrem Webspace zu haben, können Sie das WP-CLI nutzen, das

Kommandozeilen-Interface für WordPress (*https://make.wordpress.org/cli/handbook/*). Loggen Sie sich per SSH ein (die Zugangsdaten gibt es beim Hoster), und wechseln Sie in das Verzeichnis, in dem die gehackte WordPress-Instanz liegt.

Wenn das WP-CLI noch nicht installiert ist, können Sie das mit dem folgenden Befehl tun:

```
curl -O https://raw.githubusercontent.com/wp-cli/builds/gh-pages/phar/wp-cli.phar
```

Führen Sie danach den folgenden Befehl aus, um zu testen, ob das WP-CLI korrekt installiert ist:

```
php wp-cli.phar --info
```

Hat alles geklappt, können Sie mit dem folgenden Befehl prüfen, ob die Dateien des WordPress-Core unverändert sind – dabei wird die Prüfsumme der Dateien und Inhalte von den lokalen Dateien mit der Prüfsumme von den WordPress-Servern verglichen:

```
php wp-cli.phar core verify-checksums
```

Mit den optionalen Argumenten `--locale` und `--version` können Sie angeben, gegen welche WordPress-Version in welcher Sprache getestet werden soll, zum Beispiel:

```
php wp-cli.phar core verify-checksums --locale=de_DE --version=6.6
```

Diesen Abgleich können Sie auch für die installierten Plugins machen (das funktioniert aber nur für Plugins aus dem WordPress.org-Verzeichnis):

```
php wp-cli.phar plugin verify-checksums --all
```

Wenn es bei der Prüfung ein Problem gibt, bekommen Sie die Datei(en) ausgegeben, die sich von den Originalen unterscheiden.

Wenn Sie das WP-CLI nicht nutzen können oder auch Plugins von außerhalb des WordPress.org-Verzeichnisses nutzen, müssen Sie den Vergleich der Dateien lokal machen. Laden Sie sich die Installationsarchive von WordPress und alle Plugins in den Versionen herunter, die Sie installiert hatten, und entpacken Sie sie lokal in einen Ordner, so als würden sie eine funktionierende WordPress-Installation darstellen (die Plugins also in den Ordner *wp-content/plugins*). Anschließend nutzen Sie ein Programm zum Vergleich von diesem Ordner mit den Originaldateien und Ihrer per SFTP heruntergeladenen Installation (nehmen Sie dafür temporär die Ordner wie *wp-content/uploads* raus, damit das nicht als Unterschied erkannt wird). Beispiele für solche Programme sind WinMerge (Open Source, *http://winmerge.org*) und BeyondCompare (*http://www.scootersoftware.com*). *Tipp:* Schauen Sie bei den Vergleichstools in die Einstellungen, um nicht auch nach den Änderungsdaten der Dateien zu prüfen. Sie brauchen nur einen Vergleich nach dem Inhalt der Datei.

In Abbildung 21.1 sehen Sie beispielhaft einen Vergleich in BeyondCompare.

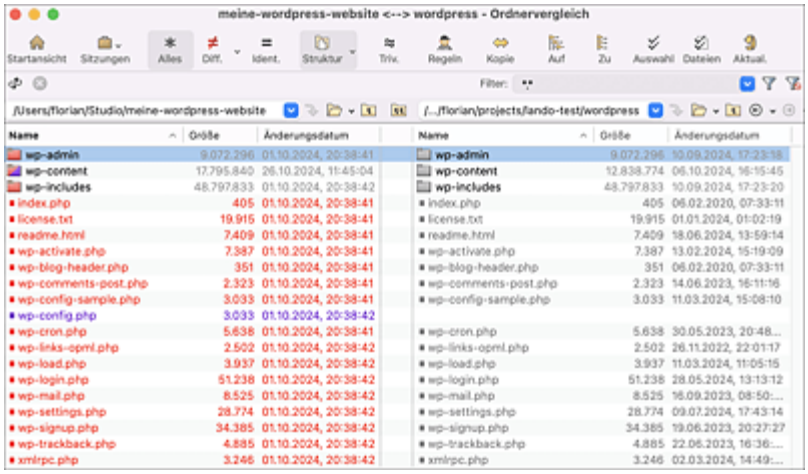


Abbildung 21.1 Beim Vergleich des Ordners Ihrer Installation mit dem Ordner der WordPress- und Plugin-Originaldateien werden Ihnen in entsprechenden Programmen Unterschiede angezeigt, wie hier in BeyondCompare.

Wenn Ihnen Unterschiede angezeigt werden, wissen Sie, dass eine Infektion vorliegt und wo der Schadcode eingebunden ist. Meist entstehen Sicherheitslücken durch Plugins und/oder durch nicht ausgeführte Updates.

21.1.3 Website reparieren oder neu aufbauen?

Sie wissen jetzt, wo der Schadcode ist, und können entscheiden, welchen Weg Sie nun gehen:

- Entfernen des Schadcodes durch Überschreiben mit den Originaldateien
- Neuaufbau der gesamten Website – zeitaufwendig, aber die sicherste Methode

Durch den Vergleich mit den Originaldateien wissen Sie genau, welche Dateien infiziert wurden, weshalb das Entfernen von Schadcode aus Dateien meist recht einfach geht: Sie überschreiben die entsprechenden Dateien mit den Originaldateien. Allerdings bleibt die Datenbank als Unsicherheitsfaktor, denn auch hier können Änderungen vorgenommen worden sein. Beachten Sie dazu die Hinweise im Kasten.

Tipp: Aufräumen nach der Kompromittierung

- Schauen Sie in die `.htaccess`-Datei, ob hier eventuell auch Schadcode eingeschleust wurde.
- Prüfen Sie in der Datenbank die `wp_users`-Tabelle (gegebenenfalls mit einem anderen Präfix) auf Einträge, die Ihnen nichts sagen. Löschen Sie alles, was Ihnen unbekannt vorkommt.

- Setzen Sie die Passwörter für die User zurück, indem Sie irgendeinen anderen Wert in das Passwortfeld in der `wp_users`-Tabelle einsetzen und die Benutzer*innen bitten, die Funktion **Passwort vergessen?** zu nutzen.
- Durchsuchen Sie die `wp_posts`-Tabelle nach Stellen mit JavaScript- und PHP-Code (`LIKE '%<script%` und `LIKE '%<?%)` sowie iFrames (`LIKE '%<iframe%`). Dafür können Sie den folgenden SQL-Code nutzen und den Teil nach `LIKE` ersetzen:
  `SELECT * FROM `wp_posts` WHERE post_content LIKE '%<script%';`
- Prüfen Sie die Verzeichnis- und Dateirechte: `750` für Verzeichnisse (oder `755` bei Problemen), `640` für Dateien (bei Problemen `644`) und `600` für die `wp-config.php`.

21.2 Die häufigsten Standardproblemlösungen

Wie Sie bereits gelernt haben, ist die Ursache für Probleme mit einer Website meist nicht WordPress selbst, sondern eins der installierten Plugins. Auf den folgenden Seiten lernen Sie Lösungen für Probleme kennen, die bei einer WordPress-Installation mal auftreten können.

21.2.1 Standardproblemlösung

Wenn es ein Problem mit Ihrer Website gibt, können Sie die folgenden Schritte als Erstes ausführen (am besten in einer Testumgebung, die Sie beispielsweise mit WP Staging erstellen können, wie in Abschnitt 17.2.3 beschrieben).

- Deaktivieren Sie alle Plugins, und wechseln Sie auf ein Standard-Theme von WordPress, um auszuschließen, dass es sich um ein Problem von WordPress selbst handelt.
- Aktivieren Sie wieder Ihr Theme und danach Schritt für Schritt die Plugins, und testen Sie zwischendurch immer, ob der Fehler wieder auftritt. So können Sie herausfinden, welches Plugin oder Theme für das Problem verantwortlich ist. Haben Sie die Ursache gefunden und liegt für das Theme oder Plugin ein Update vor, installieren Sie das Update und schauen, ob das Problem damit behoben wird. Falls nein, können Sie noch einmal alle Plugins außer dem problematischen deaktivieren, um auszuschließen, dass das Problem nur mit einem anderen Plugin auftritt.

Wenn Sie ein Plugin oder Theme als Grund für das Problem ausfindig gemacht haben, schauen Sie im Support-Forum des

Plugins/Themes vorbei, ob vielleicht schon jemand anders dasselbe Problem hatte. Falls nicht, können Sie ein neues Thema aufmachen.

21.2.2 Website langsam mit vielen Zugriffen auf xmlrpc.php

Symptom

Ihre Website ist plötzlich langsam, und in den Zugriffslogs des Webhosters finden Sie viele Anfragen an die */xmlrpc.php*-Datei.

Erklärung

Die */xmlrpc.php* ist eine Datei, die beispielsweise von anderen Programmen wie mobilen Apps dafür benutzt werden kann, Inhalte in WordPress zu veröffentlichen. Inzwischen gibt es dafür zwar auch die REST-API von WordPress, aber es gibt weitere Apps, die diese Datei benutzen. Durch viele Anfragen an die Datei kann die Last auf dem Server erhöht werden und dadurch die Website verlangsamt werden.

Lösung

Sie können die XML-RPC-Schnittstelle deaktivieren, indem Sie in der `functions.php` oder einem Plugin folgenden Code einfügen:

```
add_filter( 'xmlrpc_enabled', '__return_false' );
remove_action( 'wp_head', 'rsd_link' );
```

Und in der `.htaccess` können Sie noch die folgenden Zeilen ergänzen, um Abfragen auf die Datei direkt vom Webserver blocken zu lassen:

```
<Files xmlrpc.php>
order allow,deny
deny from all
</Files>
```

Achtung: Damit blockieren Sie die Schnittstelle natürlich für alle, auch für eventuell gewollte Anfragen. Prüfen Sie also, ob eins Ihrer Plugins die Schnittstelle verwendet oder Sie einen Dienst oder eine App nutzen, die über XML-RPC mit Ihrer Installation kommuniziert. Wenn das der Fall ist, müssen Sie die Schnittstelle aktiv lassen. Sie können über die *.htaccess* dann versuchen, IP-Adressbereiche zu blocken, von denen die Angriffe kommen. Wenn es sich um ein Bot-Netz handelt, können sich die aber auch schnell ändern. Ersetzen Sie dafür die Zeile `deny from all` durch `deny from x.y.z.`, wobei `x.y.z` die ersten drei Zahlen der IP-Adresse sind, die Sie blockieren möchten.

21.2.3 404-Fehler – Seite nicht gefunden

Symptom

Nach dem Anlegen eines neuen Inhaltstyps oder der Installation eines Plugins mit neuem Inhaltstyp erscheinen auf den Seiten des neuen Inhaltstyps 404-Fehler.

Erklärung

Wenn ein neuer Inhaltstyp registriert wird, müssen die Rewrite-Rules des neuen Inhaltstyps erst in WordPress aufgenommen werden. Dafür kann das Plugin selbst eine Vorkehrung treffen, damit das nach der Aktivierung automatisch passiert, aber wenn es vergessen wurde oder Sie selbst einen neuen Inhaltstyp angelegt haben, müssen die Regeln in der Datenbank manuell neu generiert werden.

Lösung

Rufen Sie die Seite Einstellungen • Permalinks auf. Das genügt schon, Sie müssen die Einstellungen nicht mal neu speichern.

21.2.4 Änderung nicht sichtbar

Symptom

Sie haben eine Änderung an einem Beitrag, einer Plugin-Einstellung oder im Website-Editor gemacht, sehen aber die Änderung nicht im Frontend.

Erklärung

Falls Sie eine Staging-Instanz haben, prüfen Sie, dass Sie sich wirklich auf dem Live-System befinden (Sie würden sich wundern, wie oft diese Verwechslung passiert, dass in einer Umgebung die Anpassung vorgenommen und in einer anderen Umgebung das Frontend geprüft wird). Andernfalls kann es am Cache des Browsers liegen, oder ein Caching-Plugin hat nicht auf die Änderung reagiert.

Lösung

Gehen Sie die folgenden Schritte durch:

1. Haben Sie die Änderungen in der Umgebung vorgenommen, in der Sie das Frontend prüfen oder versehentlich die Staging-Umgebung verändert, aber auf der Live-Umgebung getestet?
2. Laden Sie die Webseite ohne Cache neu: Strg + F5 oder cmd + ⇧ + R.

3. Löschen Sie den Browser-Cache in den Einstellungen, und laden Sie die Website neu.
4. Rufen Sie die Permalink-Einstellungen unter Einstellungen • Permalinks auf, um die Einstellungen neu zu generieren.
5. Löschen Sie den Cache von einem gegebenenfalls installierten Caching-Plugin.
6. Prüfen Sie, ob es von Ihrem Hoster gegebenenfalls serverseitig noch Caches gibt, die gelöscht werden können.

21.2.5 Keine Suchmaschinen-Indexierung

Symptom

Obwohl Ihre Website seit einigen Wochen online ist, taucht sie noch nicht in den Suchergebnissen auf.

Erklärung

Dass es einige Zeit dauert, dass eine neue Website in den Index von Suchmaschinen aufgenommen wird, ist ganz normal. Durch die Einreichung einer Sitemap in den Webmaster Tools der entsprechenden Suchmaschinen kann das eventuell beschleunigt werden. Aber auch dann werden Sie nicht unbedingt direkt weit oben in den Suchergebnissen auftauchen.

Lösung

Prüfen Sie zunächst, ob Ihre Website nicht doch schon indiziert ist, indem Sie nach `site:example.com` suchen (`example.com` ersetzen Sie

natürlich durch Ihre URL). Dadurch werden Ihnen die Seiten angezeigt, die indexiert sind.

Wenn keine Ergebnisse erscheinen, prüfen Sie:

1. Ob Ihre Website indexiert werden darf oder ob unter Einstellungen • Lesen • Sichtbarkeit für Suchmaschinen ein Haken gesetzt wird, der Suchmaschinen signalisiert, dass die Website nicht indexiert werden soll. Wenn da kein Haken gesetzt ist, Sie aber ein SEO-Plugin installiert haben, schauen Sie, ob dieses Plugin eventuell die Funktion bietet oder Sie dort versehentlich die Indexierung verhindert haben.
2. Ob die Seiten wirklich öffentlich zugänglich sind oder noch ein Passwortschutz aktiv ist oder Ähnliches, der Ihnen nur nicht angezeigt wird, weil Sie schon angemeldet sind.

21.2.6 Ausgesperrt – Passwort wiederherstellen

Symptom

Sie haben das Passwort nicht mehr zur Hand, mit dem Sie sich anmelden können, und kommen nicht mehr ins Backend.

Erklärung

Sie haben das Passwort vergessen. Denken Sie über die Nutzung eines Passwortmanagers nach, damit müssen Sie sich nur noch ein Passwort zum Entsperren des Passwort-Tresors merken und können beliebig komplexe Passwörter für Ihre Logins verwenden.

Lösung

Klicken Sie unter dem Login-Formular auf Passwort vergessen?, und folgen Sie den Anweisungen. Wenn das fehlschlägt, etwa weil kein E-Mail-Versand eingerichtet ist, können Sie zwei Wege gehen: direkt über die Datenbank oder über das WP-CLI.

1. Wenn Sie per SSH Zugriff auf den Webspace haben, nutzen Sie das WP-CLI, um ein neues Passwort zu setzen. Loggen Sie sich dafür per SSH ein, wechseln Sie in das Website-Verzeichnis, und installieren Sie das WP-CLI, falls es noch nicht installiert ist (wie das geht, lesen Sie in Abschnitt 21.1.2, »Schadcode suchen«). Führen Sie danach den folgenden Befehl aus:

   ```
   wp user update username --user_pass=o2i4ugh3tbio2g3ne4rg
   ```

   Ersetzen Sie `username` durch den Usernamen des Accounts, für den Sie das Passwort zurücksetzen möchten (Sie können auch die E-Mail-Adresse oder ID nutzen). Hinter `--user_pass=` geben Sie das Passwort ein.

2. Wenn Sie keinen SSH-Zugriff haben oder ihn nicht nutzen möchten, begeben Sie sich über phpMyAdmin oder ein ähnliches Tool von Ihrem Hoster in die Datenbank und die `wp_users`-Tabelle (eventuell ist das Präfix anders).

   Gehen Sie in die Bearbeitungsansicht für die gewünschte Zeile, und wählen Sie bei dem Feld `user_pass` als Funktion MD5 aus. Das ist der Verschlüsselungsalgorithmus, mit dem die Passwörter verschlüsselt werden. Geben Sie dann Ihr neues Passwort in das Feld Wert ein, und speichern Sie. Wenn Sie jetzt in der `wp_users`-Tabelle die Liste der Einträge anschauen, sehen Sie statt des von Ihnen eingegebenen Passworts einen Hash – das ist die MD5-Entsprechung des Werts, den Sie eingegeben haben.

21.2.7 Ausgesperrt – kein Zugriff auf die Website

Symptom

Wenn Sie die Domain umgestellt haben oder von HTTP auf HTTPS gewechselt sind, ohne die alte Domain vorher in der Datenbank durch die neue zu ersetzen, versucht WordPress weiter, die Site unter der alten Domain anzuzeigen. Ein anderer Grund für ein Problem beim Login kann sein, dass etwas mit den Cookies nicht stimmt.

Lösung

Wenn Sie etwas an der Domain angepasst haben, können Sie WordPress die Änderung nachträglich einfach über zwei Zeilen in der `wp-config.php` mitteilen:

```
define( 'WP_HOME', 'https://example.com' );
define( 'WP_SITEURL', 'https://example.com' );
```

Jetzt sollten Sie wieder die Website aufrufen und sich ins Backend einloggen können. Wenn Sie in den Einstellungen • Allgemein bei den beiden Feldern Website-Adresse und WordPress-Adresse nachschauen, sehen Sie, dass sie ausgegraut sind und den neuen Wert anzeigen.

Wenn Sie eine Fehlermeldung wegen Cookie-Problemen bekommen, versuchen Sie es in einem privaten Browser-Fenster.

21.2.8 Serverfehler

Symptom

Beim Aufrufen einer beliebigen Seite im Front- oder Backend erscheint eine Fehlermeldung »Internal Server Error«.

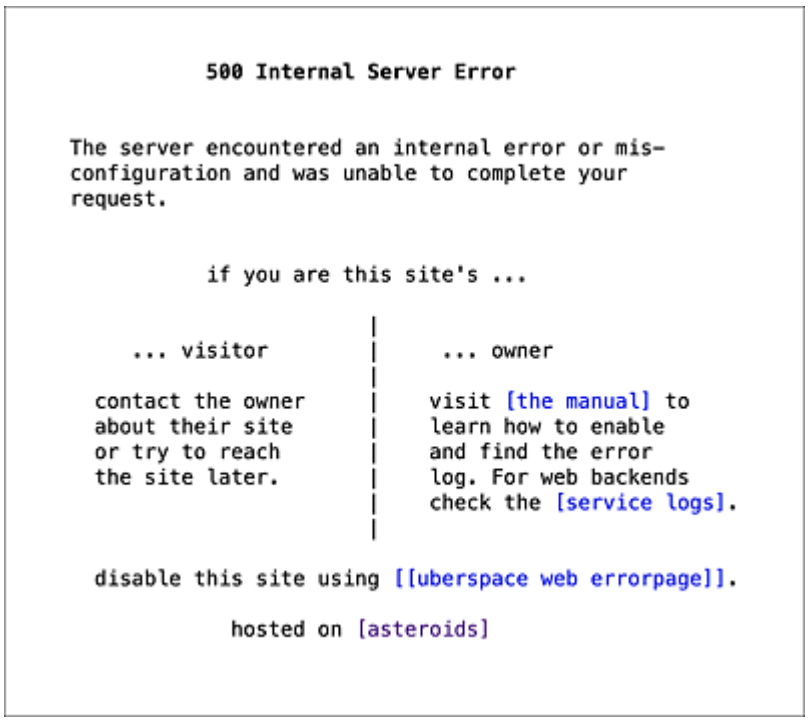


Abbildung 21.2 Wenn bei einer Fehlermeldung der Webserver-Name mit ausgegeben wird (Apache oder nginx), liegt die Ursache nicht bei WordPress, sondern auf Webserver-Ebene.

Erklärung

Wenn Sie solch eine Fehlermeldung sehen, die direkt vom Webserver kommt, ist der Grund für das Problem nicht WordPress, sondern eine Konfiguration im Webserver.

Lösung

Wenn Sie nicht gerade Ihren eigenen Webserver betreiben, ist ein Fehler in der globalen Serverkonfiguration unwahrscheinlich. Meist liegt das Problem in den lokalen Anpassungsdateien wie der *.htaccess* und kommt dort von einem Flüchtigkeitsfehler bei der Bearbeitung oder durch die Anpassung der Datei durch ein neu installiertes Plugin. Gehen Sie wie folgt vor, um nach dem Fehler zu suchen:

1. Loggen Sie sich per SFTP oder SSH auf dem Server ein, und öffnen Sie die `.htaccess` entweder direkt auf dem Server oder laden Sie sie herunter, und öffnen Sie sie lokal.
2. Kommentieren Sie mit einem `#` am Anfang der Zeile alle Zeilen aus, die nicht zwischen `BEGIN WordPress` und `END WordPress` (den Standardregeln von WordPress) stehen. Speichern Sie die Datei, und laden Sie sie auf den Server hoch, wenn Sie sie lokal bearbeitet haben.
3. Wenn das Problem gelöst ist, kommentieren Sie nach und nach die anderen Blöcke wieder ein. Hier können Sie nicht Zeile für Zeile vorgehen, da teilweise mehrere Zeilen zusammen vorkommen müssen, damit es zu keinem Fehler kommt. Prüfen Sie nach jeder Änderung, ob der Fehler wieder auftritt, und falls ja, haben Sie den problematischen Code gefunden.

   Wenn der Fehler auch auftritt, wenn nur der WordPress-Block aktiv ist, wurde Code innerhalb dieses Blocks verändert. Um wieder die korrekten Standardregeln von WordPress zu erhalten, benennen Sie die Datei um, beispielsweise in `_.htaccess`, loggen sich ins Backend ein und rufen die Seite EINSTELLUNGEN • PERMALINKS auf, um eine neue `.htaccess` generieren zu lassen. Kopieren Sie dann die zusätzlichen Regeln aus der `_.htaccess` in die Datei, und aktivieren Sie die Regeln nach und nach wieder.

21.2.9 Datenbankproblem

Symptom

Beim Aufrufen der Website erscheint die Fehlermeldung »Fehler beim Aufbau einer Datenbankverbindung«.

Erklärung

WordPress kann aus irgendeinem Grund keine Verbindung zur Datenbank herstellen. Da alle Inhalte und Optionen (bis auf die aus der `wp-config.php`) in der Datenbank gespeichert sind, kann WordPress ohne die Verbindung wenig anfangen.

Lösung

Wenn sich WordPress nicht mit der Datenbank verbinden kann, kann das zwei Gründe haben: Entweder sind die Zugangsdaten falsch, oder die Datenbank ist nicht erreichbar.

Prüfen Sie zunächst die Zugangsdaten zur Datenbank in der `wp-config.php`. Diese sollten Sie in der Datei relativ weit oben finden, die Konstanten heißen `DB_NAME` (Name der Datenbank), `DB_USER` (Datenbank-User), `DB_PASSWORD` (Passwort für die Datenbank) und `DB_HOST` (der Datenbank-Server). Die Zugangsdaten können Sie bei Ihrem Webhoster einsehen.

Wenn die Zugangsdaten stimmen, prüfen Sie, ob Sie die Datenbank über phpMyAdmin oder ein ähnliches Tool öffnen können, das von Ihrem Webhoster bereitgestellt ist. Wenn das funktioniert, ist die Datenbank an sich erreichbar, schreiben Sie eine Mail an den Support des Webhosters, und fragen Sie nach Unterstützung. Das ist ebenfalls das nächste Vorgehen, wenn die Datenbank nicht erreichbar sein sollte.

21.2.10 Fatal Error: Allowed Memory exhausted

Die Meldung Fatal Error: Allowed Memory Size [...] exhausted erscheint, inklusive Angabe des Speichers und der Datei.

Erklärung

Eine Aktion benötigt mehr Speicher, als PHP zur Verfügung hat. Passiert meist bei einer Aktion im Backend.

Lösung

Manchmal hilft es, die Aktion noch mal durchzuführen. Falls nicht, müssen Sie den für PHP reservierten Speicher vergrößern. Ob das bei Ihrem Webhoster möglich ist, erfahren Sie vom Support.

Den Speicher können Sie über eine Konstante in der `wp-config.php` erhöhen. In der Fehlermeldung wird der benötigte Speicher in Bytes angegeben; ersetzen Sie bei der Zahl die letzten sechs Ziffern durch ein `M` für Megabyte, und erhöhen Sie den Wert noch etwas, um Puffer zu haben. Schreiben Sie den Wert dann wie folgt in die `wp-config.php`:

```
define( 'WP_MEMORY_LIMIT', '256M' );
```

Falls das Problem weiter auftritt, versuchen Sie, den Speicher über die `.htaccess` zu erhöhen:

```
php_value memory_limit 256M
```

21.2.11 Verbindung ist nicht sicher

Symptom

Sie haben Ihre Website vorbildlich auf HTTPS umgestellt, aber Browser geben an (oft in der Adressleiste bei dem Schloss, das eine gesicherte Verbindung anzeigt), dass die Verbindung nicht sicher ist.

Erklärung

Wenn auf einer Website, die über HTTPS ausgeliefert wird, Ressourcen über HTTP eingebunden werden, wird die Verbindung als unsicher angezeigt. Das kann bei einer Umstellung von HTTP auf HTTPS schnell passieren, wenn in der Datenbank, einem Theme oder Plugin noch Ressourcen über HTTP eingebunden werden.

Lösung

HTTP-URLs in Beiträgen und Seiten können Sie über ein Suchen und Ersetzen in der Datenbank schnell korrigieren. Dafür können Sie das Plugin Better Search Replace nutzen, das in Abschnitt 17.2 vorgestellt wurde.

Wenn die Meldung danach immer noch kommt, prüfen Sie, welche URLs dafür verantwortlich sind, indem Sie die Webdeveloper-Tools öffnen (Rechtsklick und dann Untersuchen) und den Netzwerk-Tab öffnen. Laden Sie die Seite nun ohne Cache neu, und schauen Sie, welche der URLs `http` als Protokoll nutzt. Haben Sie die URL(s) gefunden und kommen sie aus einem Plugin oder Theme, wenden Sie sich an den Support des Plugins oder Themes.

21.2.12 White Screen of Death

Bis WordPress 5.2 kam es vor, dass Sie im Fehlerfall einfach eine weiße Seite angezeigt bekamen. Mit WordPress 5.2 wurde das Error-Handling verbessert, sodass Sie eine Fehlermeldung sehen sollten und/oder eine E-Mail mit der Fehlermeldung an die Admin-E-Mail-Adresse bekommen.

TEIL V

WordPress für Entwickler*innen

22 Grundwissen für WordPress-Entwickler*innen

Bisher haben Sie kennengelernt, wie Sie WordPress mit fertigen Themes und Plugins an Ihre Bedürfnisse anpassen können. Darüber hinaus können Sie natürlich auch selbst tätig werden und Plugins oder Themes entwickeln. Bevor Sie aber loslegen, finden Sie hier ein bisschen Grundwissen. Grundlegende Kenntnisse in HTML, CSS, JavaScript und PHP werden allerdings vorausgesetzt, da eine Programmiereinführung den Rahmen sprengen würde.

Dieses und die folgenden Kapitel machen Sie mit wichtigen Grundlagen vertraut, die Sie für die Entwicklung von WordPress-Plugins und -Themes benötigen. Dabei geht es mit ein bisschen Theorie über die Coding-Standards und Entwicklungskonzepte von WordPress los. Im Anschluss richten Sie Ihre Entwicklungsumgebung ein, und danach wird es mit ersten Code-Beispielen praktisch. Sie müssen kein Profi im Bereich Webentwicklung sein, um die Kapitel des Entwicklungsteils zu verstehen, aber ganz ohne Vorkenntnisse wird es schwierig, den Code-Beispielen im Buch zu folgen.

22.1 Coding-Standards

Bevor es mit der Programmierung losgeht, hier erst mal ein bisschen Theorie. Coding-Standards sind dafür gedacht, dass für ein bestimmtes Projekt geschriebener Code einheitlich formatiert ist. WordPress hat eigene Coding-Standards für PHP, HTML, CSS und JavaScript definiert, die für den Core von WordPress gelten und am besten auch von Themes und Plugins eingehalten werden sollten.

Natürlich ist es nicht schlimm, wenn Sie nicht direkt alle Standards behalten und richtig anwenden können. Mit der Zeit werden Sie sicherer in der Anwendung der Standards werden, und es gibt Tools, die Ihnen bei der Einhaltung helfen. Schauen Sie dafür im GitHub-Repository für die WordPress-Coding-Standards für das Tool PHP_CodeSniffer vorbei (*https://github.com/WordPress/WordPress-Coding-Standards*).

Im Folgenden werden die im November 2024 aktuellen Coding-Standards durchgegangen. Die aktuelle Version finden Sie unter *https://developer.wordpress.org/coding-standards/wordpress-coding-standards/*.

22.1.1 Accessibility

Die Coding-Standards für *Accessibility* (Barrierefreiheit) sind keine einfach zu erfüllenden Standards wie die für die unterschiedlichen Programmiersprachen. Diese Standards besagen, dass der Code des WordPress-Ökosystems, unter anderem WordPress-Core, die WordPress.org-Websites und offizielle Plugins, die *Web Content Accessibility Guidelines* (*WCAG*) Version 2.1 im Level AA erfüllen müssen.

Die WCAG sind eine umfangreiche Liste an Richtlinien, die dabei helfen, eine Website möglichst barrierefrei zu gestalten. Die Richtlinien sind in die drei Level A, AA und AAA unterteilt, wobei A

am einfachsten zu erfüllen ist und AAA am schwierigsten. Der Grad der Zugänglichkeit ist genau umgekehrt: Eine Website, die Level AAA erfüllt, ist zugänglicher als eine Site, die nur Level A einhält. Detaillierte Informationen zu den Leveln gibt es beim W3C (*https://www.w3.org/WAI/WCAG21/Understanding/conformance#levels*).

Das Thema Barrierefreiheit ist enorm wichtig, wird aber leider von vielen nicht mit der nötigen Priorität berücksichtigt. Und es braucht auch einige Zeit, sich in das Thema einzuarbeiten. Auf der Seite der Accessibility-Coding-Standards von WordPress gibt es Verweise auf Seiten mit weiteren Informationen zum Thema (*https://developer.wordpress.org/coding-standards/wordpress-coding-standards/accessibility/*). Schauen Sie da einmal vorbei, und beschäftigen Sie sich mit dem Thema. Hier im Buch finden Sie in Abschnitt 6.7.3, »Barrierefreiheit von Inhalten«, und in Abschnitt 10.6, »Barrierefreiheit von Themes testen«, weitere Informationen.

22.1.2 CSS

Für die Styling-Sprache CSS hat WordPress die folgenden Coding-Standards definiert:

Struktur

- Tabs statt Spaces für die Einrückung verwenden
- zwei Leerzeilen zwischen Bereichen und eine Leerzeile zwischen Blocks eines Bereichs
- Jeder Selektor steht in einer eigenen Zeile mit einem Komma oder einer öffnenden geschwungenen Klammer am Ende. Die

Eigenschaften-Wert-Paare stehen ebenfalls allein in einer Zeile, haben ein Semikolon am Ende und sind um einen Tabstopp eingerückt. Die schließende geschwungene Klammer ist wieder auf demselben Level eingerückt wie der oder die Selektoren.

Richtig:

```
h1,
h2 {
    line-height: 1.2;
    color: #000;
}
```

Falsch:

```
h1, h2 {
color: #000;
}
h1, h2 { color: #000; }
```

Selektoren

- Selektoren werden kleingeschrieben und Begriffe mit Bindestrichen getrennt.
- lesbare Selektoren verwenden, die beschreiben, was für Elemente gestylt werden
- Attribut-Selektoren werden mit doppelten Anführungszeichen um die Werte geschrieben.
- unnötig spezifische Selektoren vermeiden, beispielsweise statt `ul.menu` einfach `.menu`

Richtig:

```
.primary-nav {
    display: flex;
}

input[type="text"] {
    border: 1px solid #000;
}
```

Falsch:

```
/* Kein Camel-Case nutzen */
.primaryNav {
    display: flex;
}

/* Keine Unterstriche verwenden */
.primary_nav {
    display: flex;
}

/* Unnötig spezifische Selektoren vermeiden */
nav.primary-nav {
    display: flex;
}

/* Kein verständlicher Name */
.rh3 {
    display: flex;
}

/* Keine doppelten Anführungszeichen */
input[type=text] {
    border: 1px solid #000;
}
```

Eigenschaften

- Nach einer Eigenschaft folgen ein Doppelpunkt und ein Leerzeichen (`background: red` und nicht `background:red` oder ähnlich).
- Für Farben sollen Hex-Codes (wenn möglich die Kurzversion der Farbe) oder `rgba` genutzt werden, wenn Transparenz benötigt wird. Werte werden kleingeschrieben.
- Wenn möglich, sollen Kurzformen genutzt werden, also beispielsweise `border: thin solid black`, statt einzeln `border-width`, `border-style` und `border-color` anzugeben. Wenn andere Werte überschrieben werden müssen, ist die Nutzung der spezifischen Eigenschaft aber meist die bessere Wahl und

verhindert ungewollte Nebeneffekte bei Eigenschaften mit vielen möglichen Werten, etwa `background`.

Reihenfolge der Eigenschaften

Bei der Reihenfolge der Eigenschaften gibt es unterschiedliche Ansätze. In erster Linie sollte die Reihenfolge einen Sinn haben und in einer Art semantisch sein. Im WordPress-Core werden die Eigenschaften-Gruppen sortiert: Display, Positioning, Box-Model, Farben und Typografie, gefolgt von anderen Eigenschaften.

Werte

- Vor dem Wert kommt ein Leerzeichen.
- hinter einer öffnenden Klammer und vor einer schließenden Klammer keine Leerzeichen einfügen
- Nach einem Wert kommt immer ein Semikolon.
- Wenn Anführungszeichen benötigt werden, nutzen Sie doppelte statt einfacher Anführungszeichen.
- Das Schriftgewicht sollte mit einer Zahl angegeben werden (beispielsweise `600` statt `semibold` und `400` statt `normal`).
- Bei einem `0`-Wert soll keine Einheit verwendet werden (also `height: 0` statt `height: 0em`).
- Als Zeilenhöhe sollten Werte ohne Einheit genutzt werden. Warum das besser ist, wird hier beschrieben: *https://meyerweb.com/eric/thoughts/2006/02/08/unitless-line-heights/*.
- Für Dezimalwerte wird eine führende `0` angegeben, also `0.9` statt nur `.9`.

- Mehrere durch Komma getrennte Werte für eine Eigenschaft sollten entweder mit einem Leerzeichen oder einem Zeilenumbruch getrennt werden. Wenn Sie Werte auf mehrere Zeilen aufteilen, fügen Sie auch vor dem ersten Wert einen Zeilenumbruch ein und rücken die Werte um einen Tabstopp mehr ein als die Eigenschaft. Zum Beispiel:

```
box-shadow:
    0 0 0 0.05em blue,
    0 0 0.3em 0.05em red;
```

Media-Queries

Allgemein gesprochen sollten Media-Queries am Ende der CSS-Datei nach ihren Medien-Typen gruppiert werden. Regeln innerhalb eines Media-Queries sind um einen Tabstopp eingerückt.

Kommentare

- Schreiben Sie Kommentare, und lieber zu viele als zu wenige. Wenn es Bedenken wegen der Dateigröße gibt, stellen Sie die minifizierte Version einer Datei bereit und die unminifizierte mit Kommentaren, die bei Nutzung der `SCRIPT_DEBUG`-Konstante eingebunden wird. Lange Kommentare sollten bei maximal 80 Zeichen umgebrochen werden.
- Für längere Stylesheets kann es sinnvoll sein, am Anfang ein Inhaltsverzeichnis zu schreiben.
- Kommentare sollten größtenteils wie PHPDoc formatiert sein.

Best Practices

Mit der Zeit werden Stylesheets meist eher größer und unübersichtlicher als andersrum. Mit ein paar Best Practices können

Sie zumindest versuchen, dem entgegenzuwirken.

Probieren Sie beispielsweise, ob sich ein Problem nicht durch das Entfernen von Code lösen lässt, statt direkt mit neuem Code zu starten. Weitere Tipps finden Sie in den CSS-Coding-Standards (*https://developer.wordpress.org/coding-standards/wordpress-coding-standards/css/*).

22.1.3 HTML

Die HTML-Coding-Standards von WordPress (*https://developer.wordpress.org/coding-standards/wordpress-coding-standards/html/*) beziehen sich teilweise noch auf den XHTML-Standard des W3C, beispielsweise wird nicht erwähnt, dass selbstschließende Tags keinen Schrägstrich brauchen (`<br>` statt `<br />`).

Hier wird deshalb nicht auf Regeln eingegangen, die Aspekte des aktuellen HTML-Standards außen vor lassen.

Validierung

Der Browser ist sehr gut darin, auch kaputtes HTML noch sinnvoll darzustellen. Eine Website geht also nicht gleich kaputt, wenn das HTML nicht den Standards folgt. Trotzdem ist es eine gute Idee, das HTML-Markup Ihrer Website mit dem W3C-Validator zu prüfen (*https://validator.w3.org/*).

Schreibweise von Attributen und Tags

Attribute und Tags sollte kleingeschrieben werden, genauso die Werte von Attributen, sofern sie nicht für Menschen gedacht sind

(wie `alt`, `aria-label` oder `title`).

Einrückungen

HTML-Einrückung sollte die logische Struktur widerspiegeln, und zur Einrückung sollen Tabs und nicht Leerzeichen genutzt werden. Wenn HTML mit PHP gemischt ist, soll PHP die Einrückung von HTML berücksichtigen.

```
<?php if ( ! have_posts ) { ?>
<article class="post">
    <h1 class="entry-title">Nichts zu sehen</h1>
    <?php get_search_form(); ?>
</article>
<?php } ?>
```

JavaScript

Seit der Einführung des Block-Editors in WordPress 5.0 spielt JavaScript eine immer wichtigere Rolle. Die WordPress-Coding-Standards für JavaScript stammen aus dem jQuery-JavaScript-Style-Guide (*https://contribute.jquery.org/style-guide/js/*), abgesehen von den folgenden Punkten:

- WordPress nutzt für Zeichenketten einfache Anführungszeichen.
- `case`-Statements in `switch`-Blöcken sind eingerückt.
- Der Inhalt einer Funktion ist einheitlich eingerückt, auch bei Datei-füllenden Closure-Wrappern.
- Einige Weißraum-Regeln sind unterschiedlich, um mit den PHP-Standards von WordPress im Einklang zu sein.
- Das 100-Zeichen-Limit für eine Zeile wird empfohlen, aber nicht erzwungen.

Spacing

Wie bei den anderen Programmiersprachen wird Spacing großzügig verwendet.

- Einrückung mit Tabs
- keine Leerzeichen am Ende einer Zeile oder in einer Leerzeile
- Zeilen sollten in der Regel nicht länger als 80 Zeichen sein und 100 nicht überschreiten. Das ist keine feste Regel, aber zu lange Zeilen erschweren die Lesbarkeit.
- `if`-, `else`-, `for`-, `while`- und `try`-Blöcke sollten immer Klammern nutzen und in mehreren Zeilen geschrieben werden.
- Vor Operatoren wie `++` soll kein Leerzeichen stehen.
- Vor `,` und `;` darf kein Leerzeichen stehen.
- Ein `;` muss am Ende einer Zeile stehen, wenn es ein Statement abschließt.
- In Objekt-Definitionen darf vor `:` kein Leerzeichen stehen.
- Vor und nach `?` und `:` in einer Ternary-Bedingung muss ein Leerzeichen sein.
- kein Leerzeichen in leeren Objekten oder Ähnlichem (beispielsweise `{}` und `[]`)
- Am Ende jeder Datei sollte eine Leerzeile stehen.
- Nach einem `!`-Operator sollte ein Leerzeichen stehen.
- Alle Funktions-Bodys sind um einen Tab eingerückt, auch wenn die ganze Datei von einem Closure umschlossen ist.
- Leerzeichen können Code in Dokumentationsblöcken oder innerhalb einer Zeile ausrichten, am Anfang einer Zeile sollten aber nur Tabs genutzt werden.

Objekte und Arrays

Objekte und Arrays können einzeilig geschrieben werden, wenn sie kurz sind. Sonst steht jeder Wert in einer eigenen Zeile und hat ein Komma am Ende, wobei die Objektschlüssel nur dann in Anführungszeichen stehen, wenn es notwendig ist, beispielsweise weil sie Sonderzeichen enthalten.

```
const travel = {
    transportMethod: 'train',
    destination: 'Turin',
    purpose: 'WordCamp Europe',
};
```

Arrays und Funktionsaufrufe

Sowohl bei dem Ansprechen von Array-Werten als auch beim Aufrufen von Funktionen mit Argumenten wird nach der öffnenden und vor der schließenden Klammer je ein Leerzeichen verwendet.

```
setDestination( 'Turin' );
const transfers = [ 'Hamburg', 'Basel' ];
console.log( transfers[ 1 ] );
```

Semikolons

Semikolons sollten immer genutzt werden, statt auf das automatische Einfügen der Semikolons (die »Automatic semicolon insertion«) zu vertrauen.

Einrückungen und Zeilenumbrüche

Wie überall im WordPress-Kontext soll zur Einrückung der Tab statt des Leerzeichens verwendet werden.

```
( function() {
    // Some code here.
```

```
    function isTravelling() {
        // Some code here.
    }
} )();
```

Statements wie `if` oder `for` sollen immer in der Schreibweise mit geschwungenen Klammern geschrieben werden.

```
if ( isTravelling() ) {
    console.log( '\o/' );
} else if ( hasPlannedTrip() || justCameBack() ) {
    console.log( '<3' );
} else {
    console.log( '/o\' );
}
```

Mehrzeilige Statements

Wenn Sie ein Statement in mehrere Zeilen umbrechen, sollte der Umbruch nach einem Operator erfolgen. Bei einem `if`-Statement oder ähnlich steht jede Bedingung in einer eigenen Zeile, wenn es umbrochen wird.

```
if (
    isTravelling() &&
    wantsToTravel()
) {
    beHappy();
}

const isHappy = isTravelling() &&
    wantsToTravel();
```

Verkettete Methodenaufrufe

Wenn eine Kette von Methodenaufrufen zu lang für eine Zeile ist, muss jede Methode in einer eigenen Zeile stehen. Die Methoden werden gegenüber dem Objekt um einen Tab eingerückt.

```
journey
    .addTraveller( 'Florian' )
    .addDestination( 'Turin' )
    .addDuration( '2024-06-12', '2024-06-16' );
```

Variablen-Deklarationen

`const` und `let` sollten statt `var` verwendet werden, um Variablen zu deklarieren. `let` wiederum sollte nur verwendet werden, wenn `const` nicht möglich ist.

Variablen werden nicht direkt alle am Anfang deklariert, sondern erst wenn sie benötigt werden.

Namenskonventionen

- Variablen- und Funktionsnamen sollten ganze Wörter sein und in Camel-Case geschrieben werden, mit einem kleinen Anfangsbuchstaben. Hier weicht der JavaScript-Standard von den PHP-Coding-Standards ab.
- Namen sollten sinnvoll gewählt werden, außer beispielsweise bei Iteratoren für eine `for`-Schleife.
- Akronyme werden komplett in Großbuchstaben geschrieben, außer sie stehen am Anfang des Namens, dann werden sie kleingeschrieben (`nextDOMNode`, `domNode`).
- Andere Abkürzungen folgen den normalen Regeln, beispielsweise `userId`.
- Klassennamen werden am Anfang großgeschrieben (`class TrainJourney`).
- Eine Ausnahme von Camel-Case gilt für Konstantennamen, die der `SCREAMING_SNAKE_CASE`-Schreibweise folgen.

Kommentare

Kommentare stehen immer vor dem Code, auf den sie sich beziehen, und vor einem Kommentar kommt eine Leerzeile. Der

Anfang des Kommentars wird großgeschrieben, und am Ende steht ein Punkt.

```
journey.start();

// First stop: Hamburg.
journey.changeTrain( 'RE', 'ICE' );

// Second stop is in Basel, and this
// comment is long enough for two lines.
```

JSDoc-Kommentare nutzen die `/**`-Schreibweise:

```
/**
 * Class that describes all aspects of a
 * nice journey by train.
 */
class TrainJourney
```

Strikte Vergleiche

Es müssen strikte Vergleiche für Gleich- und Ungleichheit verwendet werden (`===` statt `==` und `!==` statt `!=`).

Typ-Überprüfungen

Zur Überprüfung eines Variablen-Typs (beispielsweise ob eine Variable ein String oder Objekt ist) gibt es Vorgehen, die auf *https://developer.wordpress.org/coding-standards/wordpress-coding-standards/javascript/#type-checks* beschrieben werden.

Strings

Strings stehen in einfachen Anführungszeichen.

```
const journeyDescription = 'Zugfahrt zum #WCEU';
```

Wenn ein String `'` als Zeichen enthält, muss er mit einem Backslash escaped werden.

```
const wrongQuote = 'Für\'s Beispiel hier mit falschem Apostroph.';
```

Switch-Statements

Bei `switch`-Statements muss für jeden Fall außer `default` ein `break` genutzt werden. Wenn ein Fall den nächsten Fall mitnutzen soll, erwähnen Sie das explizit in einem Kommentar.

```
switch ( journey.destinationStation ) {
    case 'Turin Porta Nouva':
    case 'Turin Porta Susa':
        welcome();
        break;
    case 'Athens':
        console.warning( 'Falsches Jahr!' );
        break;
    default:
        console.log( 'Wo geht’s hin?' );
}
```

Innerhalb eines `switch`-Statements sollten keine Werte mit `return` zurückgegeben werden. Weisen Sie stattdessen innerhalb des Statements den zurückzugebenden Wert einer Variablen zu, und geben Sie diese nach dem Statement zurück.

Best Practices

Zusätzlich zu den beschriebenen Regeln gibt es in den Docs noch einen Best-Practices-Bereich (*https://developer.wordpress.org/coding-standards/wordpress-coding-standards/javascript/#best-practices*), in dem ein paar Tipps festgehalten sind.

22.1.4 PHP

Abschließend lernen Sie nun noch die Coding-Standards für PHP kennen.

Öffnende und schließende PHP-Tags

Wenn PHP und HTML gemischt werden, stehen die öffnenden und schließenden PHP-Tags in einer eigenen Zeile, außer bei einzeiligen Statements.

```
function do_something() {
    ?>
    <article <?php post_class(); ?>>
        <h2>
            <?php the_title(); ?>
        </h2>
    </article>
    <?php
}
```

Die Kurzform der PHP-Tags (`<?` und `<?=`) wird nicht verwendet.

Anführungszeichen

Nutzen Sie einfache oder doppelte Anführungszeichen, je nach Anwendungsfall. Wenn möglich, verwenden Sie die einfachen Anführungszeichen.

include- und require-Statements

`require`/`require_once` und `include`/`include_once` brauchen keine Klammer um die Pfadangabe. Vor der Angabe des Pfads steht genau ein Leerzeichen.

```
require_once dirname( __FILE__ ) . '/vendor/autoload.php';
```

Sie sollten `require` oder `require_once` nutzen, wenn eine Datei unbedingt eingebunden werden muss, da `include` und `include_once` nur eine Warnung werfen, wenn eine Datei nicht vorhanden ist, statt eines Fehlers.

Namenskonventionen

Nutzen Sie Kleinbuchstaben in den Namen für Variablen, Funktionen/Methoden und Filter/Actions, keinen Camel-Case. Wörter werden mit Unterstrichen getrennt, verzichten Sie auf unnötige Abkürzungen, damit der Code besser verständlich ist.

```
function do_something( $attr ) {
    // The code that does something.
}
```

Bei der Benennung von Funktionsparametern verzichten Sie am besten auf reservierte Begriffe, und denken Sie daran, dass seit PHP 8.0 die Umbenennung eines Parameters von einer Funktion, die Teil der öffentlichen API ist, eine Breaking Change darstellt, da Parameter beim Funktionsaufruf über ihren Namen angesprochen werden können.

Namen für Klassen, Traits, Interfaces und Enums werden ebenfalls mit Unterstrichen getrennt, aber je mit einem Großbuchstaben am Anfang des Worts.

```
interface Journey_Interface {}
class Train_Journey implements Journey {}
enum Journey_Status {}
```

Konstanten werden, wie in JavaScript auch, in Screaming Snake Case geschrieben:

```
define( 'ABTISPAM_BEE_DEBUG', true );
```

Dateinamen werden kleingeschrieben und Wörter mit Bindestrichen getrennt.

```
antispam-bee.php
```

Bei Dateinamen von Klassen wird ein `class-`-Präfix genutzt, gefolgt vom Klassennamen in Kleinschreibung mit Bindestrichen statt Unterstrichen. Eine Ausnahme sind Test-Klassen, die dem PSR4-

Standard folgen, also genauso heißen wie die Klasse, damit die Tests mit allen unterstützten Versionen von PHPUnit funktionieren.

```
class-train-journey.php
```

Interpolation für Benennung dynamischer Hooks

Bei der Nutzung von dynamischen Actions oder Filtern sollte der Name mit Interpolation statt Concatenation erstellt werden.

```
do_action( "{$new_status}_{$post->post_type}", $post->ID, $post );
```

Leerzeichen

Nutzen Sie Leerzeichen ausgiebig, ähnlich wie in den JavaScript-Standards. Nur am Ende einer Zeile haben Leerzeichen nichts verloren.

```
if ( wants_to_travel() && is_sunny() ) {
    bike_tour( 'Flensburg', 'Hamburg' );
} else {
    book_writing();
}

$array = array( 'test', 3 );
```

Weitere Beispiele gibt es in den Docs (*https://developer.wordpress.org/coding-standards/wordpress-coding-standards/php/#space-usage*).

Einrückungen

Wie auch bei den anderen Sprachen werden Tabs für die Einrückung verwendet, außer es geht um die Ausrichtung von Code innerhalb einer Zeile, etwa in einem Kommentar-Block – dann werden Leerzeichen verwendet.

Geschweifte Klammern

Geschweifte Klammern sollten bei Bedingungen und anderen Statements immer verwendet werden, auch wenn sie theoretisch weggelassen werden könnten (das heißt nicht, dass die Konstrukte mit `if`/`endif` usw. nicht genutzt werden dürfen).

```
if ( check() ) {
    do_something();
}
```

Wenn Sie merken, dass ein Block innerhalb einer Bedingung sehr lang wird, schauen Sie, ob Sie ihn in mehrere Blöcke aufteilen können.

```
// Unübersichtlich.
if ( check_one() ) {
    if ( check_two() ) {
        // …
    }

    if ( check_three() ) {
        if ( check_three_one ) {
            // …
        }
    }
}

// Besser.
if ( check_one() && check_two() ) {
    // …
}

if ( check_one() && check_three() ) {
    if ( check_three_one() ) {
        // …
    }
}
```

Arrays

Arrays sollen in der Langform (`array()` statt `[]`) deklariert werden, da diese Schreibweise einfacher zu verstehen ist.

Mehrzeilige Funktionsaufrufe

Wenn Sie einen Funktionsaufruf über mehrere Zeilen schreiben, muss jeder Parameter in eine eigene Zeile:

```
$test_array = array(
    'first' => true,
    'meta_key' => 'field_name',
);

do_something(
    $test_array,
    true,
    null
);
```

Typ-Deklarationen

Vor und hinter einer Typ-Deklaration steht genau ein Leerzeichen, der Nullability-Operator `?` steht ohne Leerzeichen direkt vor dem Typ.

Beim Return-Typ steht der Doppelpunkt direkt hinter der schließenden Klammer der Funktionsdeklaration.

```
function do_something( Action $action, ?bool $force ): bool
```

Magische Konstanten

Die PHP-Konstanten wie `__DIR__` sollten in Großbuchstaben geschrieben werden.

`::class` zur Auflösung eines Klassennamens wird kleingeschrieben und ohne Leerzeichen vor oder nach den Doppelpunkten verwendet.

```
Action::class
```

Spread-Operator

Vor dem Spread-Operator kommt ein Leerzeichen oder ein Zeilenumbruch mit entsprechender Einrückung, je nach Kontext. Zwischen dem Operator und der Variablen/Funktion, auf die er sich bezieht, kommt kein Leerzeichen, und wenn der Referenz-Operator `&` mit verwendet wird, steht er ohne Leerzeichen direkt vor dem Spread-Operator.

```
function do_something( &...$params ) {}
```

Namespaces

Jeder Namespace-Teil sollte aus Wörtern mit großem Anfangsbuchstaben bestehen, getrennt von Unterstrichen. Vor der Deklaration eines Namespace kommt eine Leerzeile, danach mindestens eine.

```
namespace Handbuch\Entwicklung_Grundwissen\Coding_Standards;
```

Pro Datei darf es nur eine Namespace-Deklaration geben, die ganz am Anfang einer Datei steht. Deklarationen mit geschwungenen Klammern sind nicht erlaubt.

Im Core gibt es aktuell noch keine Namespaces, die Nutzung in Plugins und Themes wird aber dringend empfohlen. Achten Sie bei der Nutzung von Namespaces darauf, dass sie einzigartig sein sollten. Ein projektspezifisches Präfix ist beispielsweise eine gute Idee.

use-Statements

Alle `use`-Statements sollten am Anfang einer Datei nach dem optionalen Namespace stehen. Die Statements sollten basierend auf dem Typ des Imports in folgender Reihenfolge stehen:

1. `use`-Statements für Namespaces, Klassen, Interfaces, Traits und Enums
2. `use`-Statements für Funktionen
3. `use`-Statements für Konstanten

Eine Objekt-Struktur pro Datei

Wenn Sie zum Beispiel eine Klasse oder ein Interface erstellen, kann nichts anderes in der jeweiligen Datei stehen.

Trait-use-Statements

`use`-Statements für Traits sollten am Anfang der Klasse stehen. Vor den Statements kommt eine Leerzeile, danach mindestens eine, es sei denn, die Klasse besteht nur aus Trait-Imports.

Sichtbarkeit immer definieren

Für alle Elemente, bei denen es möglich ist, sollte immer die Sichtbarkeit definiert werden, also beispielsweise für Methoden und Eigenschaften.

Reihenfolge von Sichtbarkeit und Modifiers

- Wenn Sie mehrere Modifiers für eine Klassen-Deklaration nutzen, kommt zuerst `abstract` oder `final`, gefolgt von dem optionalen `readonly`-Modifier.
- Bei mehreren Modifiers für Konstanten in einer objektorientierten Struktur kommt zuerst der optionale `final`-Modifier, gefolgt vom Sichtbarkeits-Modifier.

- Wenn Sie mehrere Modifier für eine Eigenschaft verwenden, kommt zuerst der Sichtbarkeits-Modifier, dann der optionale `static`- oder `readonly`-Modifier und abschließend die optionale Typ-Deklaration.
- Bei einer Methoden-Deklaration mit mehreren Modifiers kommt erst der optionale `abstract`- oder `final`-Modifier, dann die Sichtbarkeit und abschließend optional der `static`-Modifier.

Objekt-Instanziierung

Bei der Erstellung eines neuen Objekts werden immer Klammern genutzt, auch wenn sie nicht notwendig sind. Zwischen der schließenden Klammer und dem Klassennamen kommt kein Leerzeichen.

Kontrollstrukturen

- Nutzen Sie `elseif` statt `else if`.
- Es sollen Yoda-Conditions genutzt werden, das heißt, bei logischen Vergleichen mit Variablen sollen die Konstanten, Literale oder Funktionsaufrufe an erster Stelle stehen, damit nicht versehentlich eine Zuweisung an die Variable passiert.

  Bei Größer/Kleiner-Vergleichen sollten Yoda-Conditions vermieden werden, da sie deutlich schwerer zu lesen sind.

Operatoren

- Beim Ternary-Operator sollte immer geprüft werden, ob die Bedingung wahr ist (eine Ausnahme wäre `! empty()`).
- Der Error-Control-Operator `@` sollte nicht verwendet werden.

- Bei Increment- und Decrement-Operatoren sollte die Variante mit vorherigem Increment/Decrement genutzt werden (beispielsweise `++$count`), da sie etwas performanter ist und Bugs verhindern kann, wenn Code nachträglich verschoben wird.

Datenbankanfragen

Vermeiden Sie, die Datenbank direkt anzufassen. Nutzen Sie stattdessen die von WordPress bereitgestellten Funktionen, um Daten auszulesen und zu verändern.

Diese Abstraktion hilft dabei, Ihren Code vorwärtskompatibel zu machen, und ermöglicht das Cachen von Ergebnissen.

Formatierung von SQL-Statements

Komplexe SQL-Statements können bei Bedarf mit der entsprechenden Einrückung in mehrere Zeilen umgebrochen werden. Die SQL-Keywords wie `UPDATE` und `WHERE` sollten in Großbuchstaben geschrieben werden.

Funktionen, die die Datenbank ansprechen, sollten davon ausgehen, dass SQL-Escaping noch nicht durchgeführt wurde. Escaping sollte am besten mit `$wpdb->prepare()` durchgeführt werden.

```
$wpdb->query(
    $wpdb->prepare(
        "UPDATE $wpdb->posts
            SET post_title = %s
            WHERE ID = %d",
        'Neuer Beitragstitel',
        12
    )
);
```

Weitere Empfehlungen

Zusätzlich zu diesen Regeln gibt es weitere Empfehlungen in den Docs (*https://developer.wordpress.org/coding-standards/wordpress-coding-standards/php/#recommendations*).

22.1.5 Versionierung

Zur Versionierung eines Plugins oder Themes gibt es zwar keinen Coding-Standard von WordPress, es ist aber eine gute Idee, hierbei den Regeln des *Semantic Versioning* (*semver.org*) zu folgen. Der Vorteil ist, dass direkt aus der Änderung der Versionsnummer hervorgeht, um welche Art Release es sich bei der neuen Version handelt, denn im Semantic Versioning besteht jede Versionsnummer aus drei Stellen `Majorversion.Minorversion.Patchversion`, die jeweils die Art des Releases kennzeichnen:

- Die *Majorversion* muss erhöht werden, wenn eine neue Version Breaking Changes enthält. Breaking Changes bedeutet, dass sich etwas an der API eines Programms geändert hat, also gegebenenfalls andere Programme kaputtgehen könnten, die genau diese Funktion nutzen.
- Die Erhöhung der *Minorversion* gibt an, dass neue Funktionen hinzugefügt wurden, die neue Version aber rückwärtskompatibel ist.
- Die *Patchversion* wird erhöht, wenn eine neue Version nur Bugfixes enthält.

Info: WordPress und semantische Versionierung

Leider nutzt der WordPress-Core selbst keine semantische Versionierung, und auch bei Plugins und Themes können Sie sich

nicht darauf verlassen. Trotzdem ist es eine gute Idee, wenn Sie sich selbst an diesen Standard halten und nicht relativ willkürlich beispielsweise ein Major-Update (die erste Zahl erhöht) herausbringen, nur weil neue Funktionen dazugekommen sind. Das wäre ein Fall für ein Minor-Update (zweite Nummer erhöhen), es sei denn, Sie haben auch Dinge verändert, die die Rückwärtskompatibilität betreffen.

22.2 WordPress-Konzepte

Im Folgenden lernen Sie ein paar der Konzepte kennen, die WordPress zugrunde liegen.

22.2.1 Hooks

Das wichtigste Konzept, das Sie als Entwickler*in im WordPress-Kontext kennen müssen, sind die sogenannten Hooks. Hooks erlauben Ihnen, sich an bestimmte Stellen in WordPress einzuhängen und zusätzliche Aktionen auszuführen (*Action*-Hooks) oder Werte zu verändern (*Filter*-Hooks).

Actions

Bei einer Action müssen Sie keinen Wert zurückgeben. Sie können beispielsweise mittels `add_action` an die WordPress-Action `wp_head` eine Funktion anhängen, um im `head`-Bereich der Website zusätzliche Informationen auszugeben.

Das kann beispielhaft so aussehen, um einen HTML-Kommentar in den `head` einzufügen:

```
add_action( 'wp_head', function() {
    echo '<!-- Hi! -->';
} );
```

Der erste Parameter für `add_action` ist der Name der Action, der zweite ein Callback, der ausgeführt wird. Viele Actions übergeben optional noch Parameter, die in der Funktion genutzt werden können.

In der Doku zur `activated_plugin`-Action (*https://developer.wordpress.org/reference/hooks/activated_plugin/*) ist beispielsweise sichtbar, dass zwei Parameter übergeben werden. Ob Sie keinen davon, nur den ersten oder beide nutzen möchten, liegt an Ihnen. Wenn Sie keinen oder nur den ersten nutzen wollen, unterscheidet sich der Aufruf von `add_action` nur geringfügig:

```
add_action(
    'activated_plugin',
    function() {
        // …
    }
);

add_action(
    'activated_plugin',
    function( $plugin ) {
        // …
    }
);
```

Wenn Sie bei einem Callback mehr als einen Parameter der Action (oder auch eines Filters) nutzen möchten, müssen Sie die Anzahl der Parameter beim `add_action`-Aufruf als vierten Parameter angeben.

```
add_action(
    'activated_plugin',
    function( $plugin, $network_wide ) {
        // …
    },
    10,
    2
);
```

Der dritte Parameter im Beispiel, die `10`, legt die Priorität fest, mit der der Callback ausgeführt wird. Je niedriger der Wert, desto früher wird er ausgeführt, `10` ist der Standardwert. Über die `2` als vierten Parameter geben Sie an, dass der Callback zwei Parameter der Action bekommen soll.

Filter

Anders als bei Actions muss bei Filtern ein Wert zurückgegeben werden, da der Wert im weiteren Ablauf des WordPress-Codes weiterverwendet wird.

Ein Filter-Hook ist beispielsweise `body_class`, der es erlaubt, die Klassen für das `class`-Attribut des `body`-Elements zu verändern oder zu erweitern:

```
add_filter( 'body_class', function( $classes ) {
    $classes[] = 'example-class';
    return $classes;
} );
```

Im Beispiel wird dem Klassen-Array `$classes` die Klasse `example-class` hinzugefügt und das Array zurückgegeben. Wie bei Actions gibt es auch Filter mit mehreren Parametern, die über Angabe des vierten Parameters an `add_filter` genutzt werden können.

22.2.2 WordPress-APIs, Template-Tags (Classic Themes)

WordPress bringt unterschiedliche APIs mit, die Sie als Entwickler*in nutzen können, um zum Beispiel eine Einstellungsseite für Ihr Plugin zu erstellen (Settings- und Options-API), um die REST-API von WordPress zu erstellen oder HTTP-Anfragen zu stellen (HTTP-API). Dokumentation zu den APIs finden Sie unter *developer.wordpress. org/*.

Bei der Theme-Entwicklung gibt es seit einiger Zeit neben den sogenannten Classic Themes das Konzept der Block-Themes, die über den Website-Editor von WordPress erstellt werden können und komplett aus Blöcken bestehen. Auch wenn die Block-Themes seit ein paar Jahren existieren, gibt es immer noch Punkte, die nicht oder nur umständlich umzusetzen sind, weshalb Classic Themes immer noch ihre Berechtigung haben. Bei der Erstellung eines

Classic Themes kommen Sie an den sogenannten Template-Tags nicht vorbei, mit denen Sie beispielsweise den Loop erstellen können, um eine Liste von Beiträgen auszugeben.

22.2.3 Übersetzungen (Internationalisierung)

Wenn Sie ein Plugin oder Theme erstellen, das Sie veröffentlichen möchten, sodass es von anderen genutzt werden kann, ist es eine gute Idee, Ihr Produkt für die Übersetzung in unterschiedliche Sprachen vorzubereiten.

Dafür stellt WordPress sogenannte *Gettext*-Funktionen bereit, die Sie sowohl in PHP als auch in JavaScript verwenden können. Im einfachsten Fall besteht eine Gettext-Funktion aus der übersetzbaren Zeichenkette und einer eindeutigen Text-Domain. Typischerweise wird als Text-Domain der Ordnername des Plugins oder Themes verwendet.

Der Aufruf einer Gettext-Funktion kann wie folgt aussehen:

```
_e( 'Click here!', 'text-domain' );
```

`_e()` ist im Beispiel die Gettext-Funktion, die den als ersten Parameter übergebenen Wert oder dessen Übersetzung direkt ausgibt (`e` wie `echo`). Der zweite Parameter ist die Text-Domain, anhand derer die Übersetzungen geladen werden.

Neben der Auszeichnung der zu übersetzenden Zeichenketten mit den Gettext-Funktionen muss als weiterer Schritt die Übersetzung geladen werden. Hier muss unterschieden werden zwischen Übersetzungen für PHP-Code und Übersetzungen für JavaScript-Code. Sie werden hier an einem minimalen Beispiel der Einfachheit halber erst mal nur den Weg für PHP-Code kennenlernen, am Ende

dieses Abschnitts finden Sie einen Kasten zum Laden von JavaScript-Übersetzungen.

Der folgende Code zeigt das gesamte Plugin, das als Beispiel dient. Erstellen Sie im *wp-content/plugins*-Ordner ein Verzeichnis *wh-hello-world* und darin eine Datei *wh-hello-world.php* mit dem Code aus dem Listing 22.1.

```
<?php
/**
 * Plugin Name: WH Hello World
 * Description: Alert in Front- und Backend
 * Author: Ihr Name
 * Version: 1.0.0
 * Text Domain: wh-hello-world
 */

function wh_hello_world_load_textdomain() {
    load_plugin_textdomain( 'wh-hello-world', false, basename(
    dirname( __FILE__ ) ) . '/languages' );
}

function wh_hello_world() {
    echo '<script>alert( "' . __( 'Hello World!', 'wh-hello-world' ) .
    '" );</script>';
}

add_action( 'admin_head', 'wh_hello_world' );
add_action( 'wp_head', 'wh_hello_world' );
add_action( 'init', 'wh_hello_world_load_textdomain' );
```

Listing 22.1 Ein sehr einfaches, aber übersetzbares Plugin

Die Hauptdatei eines WordPress-Plugins beginnt mit einem Kommentar, in dem Meta-Informationen zum Plugin stehen, wie der Plugin-Name und die Beschreibung. Wichtig für die Übersetzbarkeit ist, dass Sie eine `Text Domain` angeben. Den Wert, den Sie hier vergeben, müssen Sie in den Gettext-Funktionen wiederverwenden.

Das Plugin besteht nur aus zwei Funktionen: `wh_hello_world_load_textdomain()` wird an den `init`-Action-Hook gehängt und ruft die WordPress-Funktion `load_plugin_textdomain()`

auf, womit die Plugin-Übersetzungen geladen werden. Als ersten Parameter übergeben Sie die Text-Domain, der zweite Parameter ist veraltet, und der dritte Parameter gibt den Verzeichnispfad an, in dem die Übersetzungsdateien liegen – in diesem Beispiel ist das der Ordner *languages* in dem Verzeichnis, in dem auch die Plugin-Datei liegt.

Wenn Sie dieses Plugin bei WordPress.org veröffentlichen würden, würde der erste Parameter, die Text-Domain, ausreichen, da die Übersetzungen dann automatisch von *translate.WordPress.org* geladen werden.

Die zweite Funktion `wh_hello_world()` fügt im Backend und Frontend im `head` der Website ein kleines JavaScript-Snippet ein, das beim Aufrufen der Seite eine Alert-Meldung mit der Nachricht »Hello World!« anzeigt, und dieses »Hello World!« wird über die Nutzung der `__()`-Funktion übersetzbar gemacht.

Info: Weitere Übersetzungsfunktionen und Best Practices

Neben `_e()` gibt es noch einige weitere Gettext-Funktionen. Mit `__()` (ja, zwei Unterstriche) können Sie Text übersetzbar machen, ohne ihn direkt auszugeben:

```
__( 'Click here!', 'text-domain' );
```

Wenn ein Begriff mehrfach vorkommt, aber unterschiedliche Bedeutungen hat, können Sie einen Kontext angeben, sodass der Begriff im Übersetzungsprogramm mehrfach angezeigt und unterschiedlich übersetzt werden kann.

```
_x( 'guide', 'verb', 'text-domain' );
_x( 'guide', 'subject', 'text-domain' );
```

Der zweite Parameter ist der Kontext. Dieser Kontext wird im Übersetzungsprogramm angezeigt, sodass der*die Übersetzer*in weiß, wie korrekt übersetzt werden muss. Wenn Sie den String direkt ausgeben möchten, gibt es auch hier wieder die Variante mit e: `_ex()`.

Manchmal kommt es vor, dass Sie innerhalb eines übersetzbaren Strings eine Variable einsetzen möchten, beispielsweise um den Weiterlesen-Link auf einen Beitrag anzuzeigen, »Continue reading ›Hello World‹«. Spontan könnte die Idee aufkommen, das aus zwei Teilen zusammenzusetzen: erst der übersetzbare String »Continue reading«, gefolgt vom Titel des Beitrags.

Das ist allerdings keine gute Idee, da die Reihenfolge des Strings je nach Sprache unterschiedlich sein kann, sodass es möglich sein muss, die Variable (den Titel) an einer anderen Position als am Ende zu nutzen (im Deutschen würde die Reihenfolge beispielsweise »›Hello World‹ weiterlesen« sein).

Dafür können Sie die Funktionen `printf()` und `sprintf()` nutzen, die es Ihnen erlauben, Platzhalter in einen Text einzusetzen, die dann mit den Werten von beispielsweise Variablen ersetzt werden:

```
printf(
    __( 'Continue reading »%s«', 'text-domain' ),
    get_the_title()
);
```

Bei der Übersetzung kann über den Platzhalter `%s` die Position des Titels jetzt frei geändert werden.

Wenn Sie mehrere Platzhalter in einem String haben, können Sie die Ersetzungsreihenfolge der Platzhalter markieren, sodass sich die Reihenfolge der Platzhalter im String verändern kann und trotzdem die richtigen Platzhalter mit den richtigen Werten ersetzt werden. (In einem String mit `%s %s` würde immer der erste

Platzhalter mit dem ersten Wert nach der Übersetzungsfunktion ersetzt werden, Sie hätten keine Möglichkeit, den zweiten Wert vor dem ersten anzuzeigen.)

Das sieht wie folgt aus:

```
printf(
    __( '%1$d comments on »%2$s«', 'text-domain' ),
    get_comments_number(),
    get_the_title()
);
```

Über die Zahlen in den Platzhaltern `%1$d` und `%2$s` wird festgelegt, dass der Platzhalter `%1$d` immer mit dem ersten Parameter nach der Übersetzungsfunktion ersetzt wird, egal ob er vor oder nach `%2$s` steht.

Anhand dieses Beispiels mit der Anzahl der Kommentare lernen Sie noch eine weitere Übersetzungsfunktion kennen, denn es wäre ja praktisch, wenn bei einem Kommentar nicht »comments«, sondern »comment« erscheint und vielleicht statt »1 comment« lieber »One comment«. Das geht mit der Funktion `_n()`, der Sie eine Singular- und Plural-Form übergeben können sowie als letzten Parameter eine Anzahl, die darüber entscheidet, welche Form genutzt wird:

```
$comment_count = get_comments_number();
printf(
    _n(
        'One comment on »%2$s«',
        '%1$d comments on »%2$s«',
        'text-domain',
        $comment_count
    ),
    $comment_count,
    get_the_title()
);
```

Darüber hinaus gibt es noch weitere Gettext-Funktionen, wie etwa `esc_attr__()`, eine Kombination aus `esc_attr()` und `__()`, mit der Sie übersetzbare Zeichenketten sicher in HTML-Attributen

verwenden können, ohne sich selbst um das Escaping von speziellen Zeichen kümmern zu müssen.

22.2.4 Übersetzungen anlegen

Wenn Sie Ihr Plugin oder Theme auf die Übersetzung vorbereitet haben, kommt im nächsten Schritt die eigentliche Übersetzung. Hier haben Sie mehrere Möglichkeiten:

Wenn Sie Ihr Plugin oder Theme im offiziellen Plugin- beziehungsweise Theme-Verzeichnis auf WordPress.org veröffentlichen, wird die Übersetzung über die Plattform *translate.WordPress.org* angelegt. Sie laden also zunächst das Plugin oder Theme ins Verzeichnis hoch und können danach die Übersetzungen anlegen.

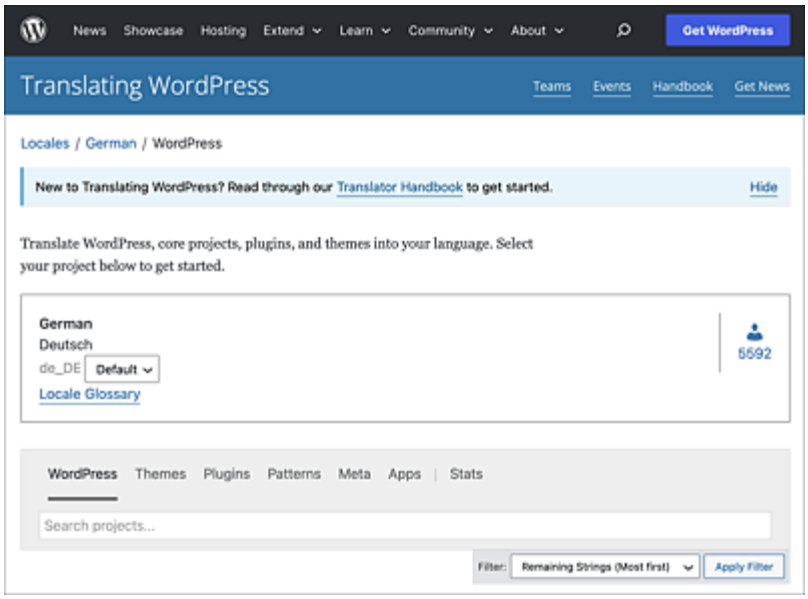


Abbildung 22.1 Auf translate.WordPress.org werden unter anderem die Übersetzungen für alle Plugins und Themes verwaltet, die auf WordPress.org gehostet werden.

Wenn Ihr Plugin/Theme nicht im offiziellen Verzeichnis veröffentlicht wird, müssen Sie die Übersetzungen mit den Dateien

des Plugins oder Themes ausliefern (es gibt an sich auch hier die Möglichkeit, die Übersetzungen extern zu verwalten, aber das ist um einiges umständlicher in der Einrichtung, weshalb es hier nicht näher erklärt wird).

Ein beliebtes Tool zum Anlegen von Übersetzungen ist Poedit (*poedit.net*), ein Programm zur Erstellung von Übersetzungen für Zeichenketten in Programm-Code, das spezielle Unterstützung für WordPress-Plugins und -Themes mitbringt, hierfür ist allerdings die Bezahlversion notwendig.

Im ersten Teil dieses Abschnitts haben Sie bereits den Code eines kleinen Plugins gesehen, das jetzt als Beispiel zum Anlegen einer Übersetzung herhält. Installieren Sie dafür Poedit für Ihr System, und starten Sie es.

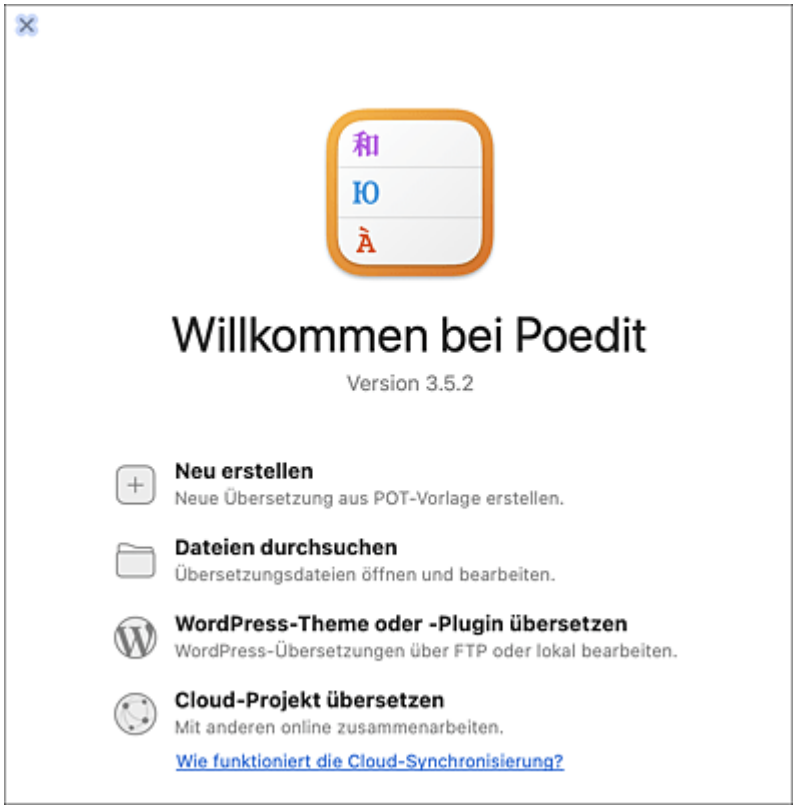


Abbildung 22.2 Poedit ist eine Software zur Übersetzung von Zeichenketten in Programmcode.

Wenn Sie die Pro-Version von Poedit besitzen, können Sie im Startbildschirm den Punkt WordPress-Theme oder -Plugin übersetzen auswählen. Hier wird der Weg mit der kostenlosen Version beschrieben:

1. Erstellen Sie in dem Verzeichnis, in dem die Plugin-Datei liegt, einen *languages*-Ordner.
2. Vor dem Anlegen einer Übersetzung muss eine Übersetzungsvorlage erstellt werden, was über Poedit ohne die Pro-Version etwas hakelig ist.
3. Wählen Sie im Menü Datei • Neu und im erscheinenden Dialog English für die englischsprachige Vorlage, und bestätigen Sie mit OK.
4. Klicken Sie nun auf den Button Aus Quellcode extrahieren in der Fenstermitte, und nehmen Sie die folgenden Einstellungen vor.
5. Im Tab Übersetzungseigenschaften geben Sie in das Feld Projektname und -version den Wert `wh-hello-world` ein.
6. Im Tab Schlüsselwörter aus Quelltexten können Sie Übersetzungsfunktionen angeben, nach denen Poedit im Quellcode suchen soll. Im Beispiel-Plugin gibt es nur die Funktion `__()`, deshalb muss diese Funktion eingefügt werden – klicken Sie dazu auf das Plus-Symbol, und geben Sie die beiden Unterstriche ein.

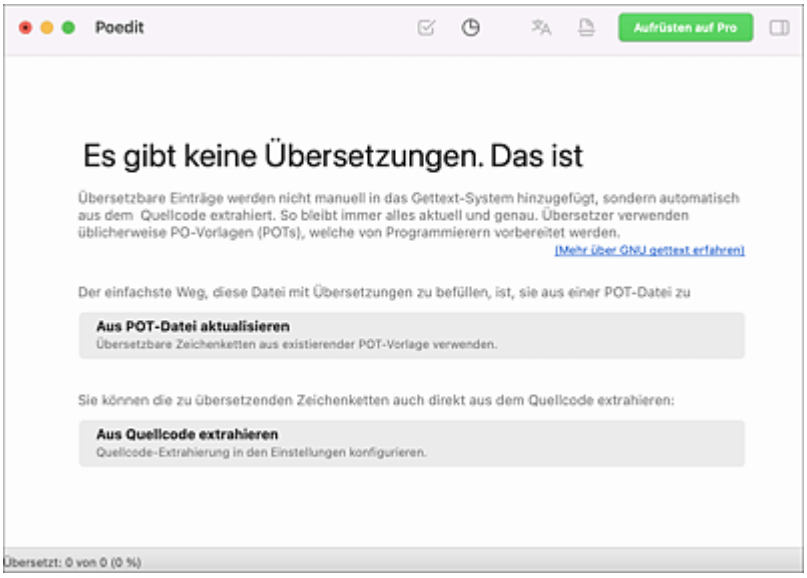


Abbildung 22.3 Poedit-Dialog zum Anlegen einer neuen Übersetzung

7. Bestätigen Sie die Änderungen mit einem Klick auf OK, und speichern Sie die Datei über Datei • Speichern unter im Ordner *languages* in dem Plugin-Verzeichnis als *wh-hello-world.pot*. Poedit wird Ihnen hier nur erlauben, als Endung *.po* zu verwenden – bestätigen Sie das, und benennen Sie die Datei im Anschluss manuell in *wh-hello-world.pot* um.
8. Öffnen Sie die umbenannte Datei mit Poedit, und klicken Sie wieder auf Aus Quellcode extrahieren. Hier können Sie im Tab Quell-Pfade jetzt den Plugin-Ordner im Pfade-Fenster angeben und mit OK bestätigen.
9. Jetzt durchsucht Poedit die Dateien des Ordners und zeigt Ihnen danach den einzigen zu übersetzenden String »Hello World!« im Hauptfenster an.
10. Wenn keine Übersetzungen gefunden werden, überprüfen Sie noch einmal die Einstellungen der Quell-Pfade und der Schlüsselwörter aus Quellen.
11. Speichern Sie jetzt die *.pot*-Datei mit Strg/cmd + S ab, ohne schon weiter etwas auszufüllen – Sie haben erst mal die

Übersetzungsvorlage erstellt.

Hinweis: Keine Gettext-Funktionen im Buch

In den Code-Beispielen in diesem Buch werden der Lesbarkeit halber weiterhin Zeichenketten ohne Gettext-Funktionen genutzt.

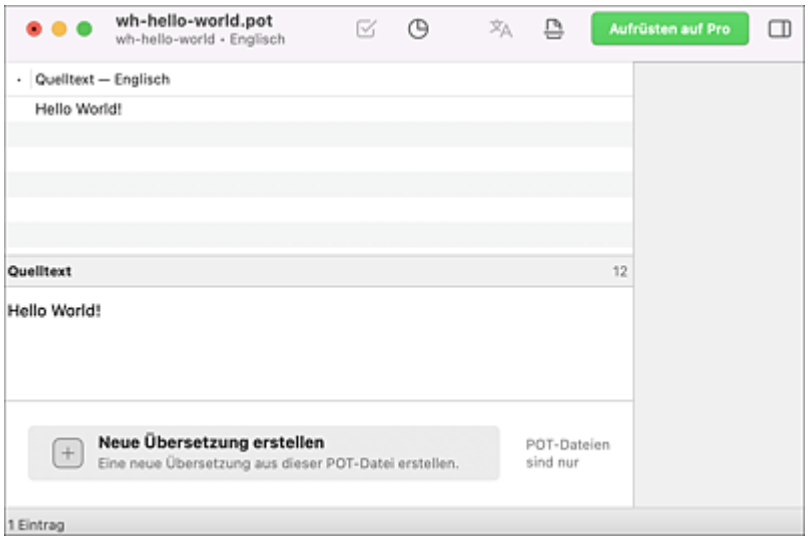


Abbildung 22.4 Das Übersetzungstemplate mit dem »Hello World!«-String

Jetzt legen Sie die konkrete Übersetzung an:

1. Wählen Sie im Menü Datei • Neu aus POT/PO-Datei, und wählen Sie die gerade erstellte Datei *wh-hello-world.pot* aus.

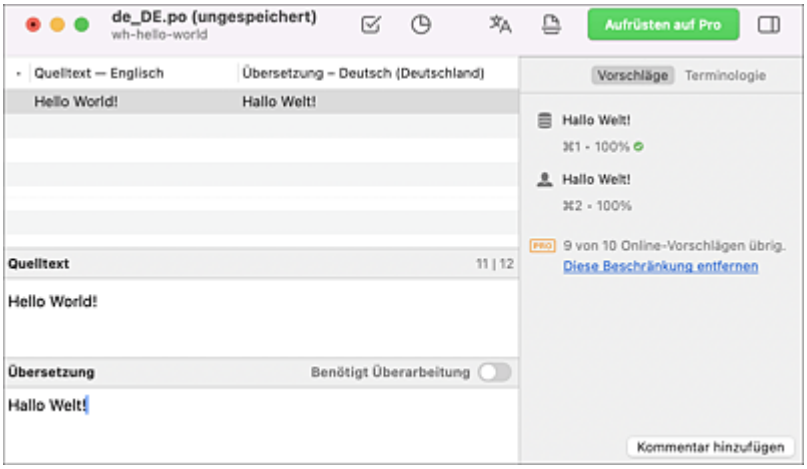


Abbildung 22.5 Ansicht in Poedit, in der Sie die Übersetzungen erstellen können

2. In der sich öffnenden Ansicht wählen Sie die Sprache Deutsch (Deutschland) aus. Das ist die Sprachkombination, nach der von WordPress gesucht wird, wenn die Website auf Deutsch eingestellt ist.
3. Wählen Sie die »Hello World!«-Zeile aus, und tragen Sie unten die Übersetzung »Hallo Welt!« ein.
4. Speichern Sie die Übersetzung über Strg/cmd + S unter dem Dateinamen *wh-hello-world-de_DE.po* im *languages*-Verzeichnis.

Jetzt können Sie Poedit schließen und sollten im *languages*-Verzeichnis drei Dateien vorfinden (oder vier, aber die *wh-hello-world.mo* können Sie löschen):

- die Vorlagendatei *wh-hello-world.pot*
- die deutsche Übersetzung *wh-hello-world-de_DE.po* im Klartext
- die deutsche maschinenlesbare Übersetzung *wh-hello-world-de_DE.mo*, die von WordPress für das Laden der Übersetzung genutzt wird

Wenn Sie nun das Plugin auf einer deutschen WordPress-Website aktivieren, sollten Sie beim Neuladen einer Seite im Frontend oder Backend eine Alert-Meldung »Hallo Welt!« sehen.

Auch wenn von WordPress nur die *.mo*-Datei verwendet wird, sollten Sie keinesfalls die *.pot*- oder *.po*-Datei wegwerfen, damit Sie später einfach die Übersetzung bearbeiten oder erweitern können (Button Aktualisieren aus Quellcode).

Info: Alternativer Weg zur POT-Datei mit WP-CLI

Wenn Sie nichts dagegen haben, mit der Kommandozeile zu arbeiten, können Sie auch das Kommandozeilen-Interface von WordPress nutzen, um eine POT-Datei zu erstellen. Dafür müssen Sie WP-CLI zunächst installieren, was aber in ein paar Schritten erledigt ist. Schauen Sie für die zu Ihrem System passende Installationsanleitung auf *https://make.wordpress.org/cli/handbook/guides/installing/*. Nachdem WP-CLI installiert ist, können Sie mit dem folgenden Befehl eine POT-Datei für das Projekt erstellen, in dem Sie sich in der Kommandozeile gerade befinden:

```
wp i18n make-pot . languages/slug.pot
```

Der Befehl erstellt eine *slug.pot* im *languages*-Verzeichnis mit allen übersetzbaren Strings. Mit dem Argument `--exclude` können Sie Verzeichnisse ausschließen, was zum Beispiel nützlich ist, wenn JavaScript-Übersetzungsfunktionen enthalten sind, diese aber nur aus den gebauten Dateien extrahiert werden sollen und nicht aus den Dateien, die später nicht mit auf dem Webserver bei den Benutzer*innen landen:

```
wp i18n make-pot . languages/slug.pot -exclude=src
```

Mit diesem Befehl werden alle Übersetzungen extrahiert, außer denen aus dem `src`- Verzeichnis.

Info: Übersetzung für JavaScript-Translation-Strings

Das werden Sie zwar erst in Kapitel 24 im Rahmen der Scripts für den Block-Editor kennenlernen, aber WordPress bietet äquivalent zu den PHP-Übersetzungsfunktionen auch welche für die Übersetzung von Strings in JavaScript an. Diese Strings können nicht einfach mit PO-Dateien übersetzt werden, die

Übersetzungen werden aus JSON-Dateien ausgelesen. Es ist mit WP-CLI allerdings möglich, Übersetzungen für JavaScript-Strings aus einer PO-Datei in eine solche JSON-Datei umzuwandeln, sodass Sie für die Übersetzung trotzdem Tools wie Poedit nutzen können.

Bei der Generierung einer POT-Datei mit dem Befehl `wp i18n make-pot`, der im vorherigen Kasten gezeigt wurde, werden auch bereits Strings aus JavaScript-Übersetzungsfunktionen in die POT-Datei geschrieben. Wenn Sie aus dieser Datei wie im Kapitel beschrieben eine PO-Datei mit Übersetzungen erstellen, können Sie mit dem folgenden Befehl die Strings für JavaScript-Übersetzungsfunktionen in JSON extrahieren:

```
wp i18n make-json languages --no-purge
```

Mit dem Befehl werden alle PO-Dateien im *languages*-Ordner nach Strings aus JavaScript-Dateien durchsucht, und diese werden in JSON-Dateien extrahiert. Es wird pro JavaScript-Datei, aus der Strings in den PO-Dateien liegen, eine eigene JSON-Datei erstellt, die nach dem Muster `${domain}-${locale}-${md5}.json` benannt sind. Der Hash wird aus dem relativen Pfad zu der JavaScript-Datei generiert. Statt des Hashes kann auch das Handle genutzt werden, mit dem das entsprechende Skript in WordPress registriert wurde.

Das `--no-purge`-Argument sorgt dafür, dass die extrahierten Strings nicht aus der PO-Datei gelöscht werden, standardmäßig werden sie nämlich entfernt.

Nachdem Sie die Übersetzungsdateien vorliegen haben, müssen Sie diese noch extra laden, denn über `load_plugin_textdomain()` werden nur PHP-Übersetzungen geladen. Für JavaScript-Übersetzungen gibt es die Funktion

`wp_set_script_translations()`, die wie folgt eingesetzt werden kann (in der Hauptdatei eines Plugins):

```
function set_script_translations() {
    wp_set_script_translations(
        'script-handle',
        'wh-talks',
        plugin_dir_path( __FILE__ ) . '/languages/'
    );
}
add_action( 'init', 'set_script_translations', 11 );
```

Als ersten Parameter erhält die Funktion das Handle des Scripts, für das die Übersetzung ist, als zweiten die Text-Domain und als dritten den Pfad zum Ordner mit den Übersetzungsdateien. Jetzt sollten die Übersetzungen auch für JavaScript-Übersetzungsfunktionen geladen werden.

22.3 Entwicklungs- und Testumgebung einrichten

Den wichtigsten Teil der lokalen Entwicklungsumgebung haben Sie bereits im zweiten Kapitel des Buchs kennengelernt: den lokalen Webserver. Mit wp-now haben Sie beispielsweise die Möglichkeit, recht einfach lokale Instanzen zu erstellen, die für die Theme- oder Plugin-Entwicklung genutzt werden können.

22.3.1 Code-Editor auswählen

Neben dem lokalen Server brauchen Sie eigentlich nur noch einen Editor und können loslegen. Theoretisch könnten Sie den Text-Editor nutzen, der mit Ihrem Betriebssystem ausgeliefert wird. Angenehmer ist es allerdings, wenn der Editor Dinge wie Syntax-Highlighting und Code-Vervollständigung unterstützt.

Darüber hinaus ist es bei Code-Editoren oft eine Frage des persönlichen Geschmacks. Soll es ein schlankerer Editor sein oder eine *IDE* (Integrated Development Environment), die mehr Möglichkeiten bietet (wobei sich auch viele der schlankeren Editoren über Plugins erweitern lassen, sodass sie einen großen Funktionsumfang erreichen können)? Wenn Sie noch keine Präferenz für einen Editor haben, ist es vielleicht eine gute Idee, mit einem schlankeren Editor anzufangen und gegebenenfalls später zu einer IDE zu wechseln.

Beliebte halbwegs schlanke Editoren sind zum Beispiel:

- Visual Studio Code (*https://code.visualstudio.com/*)
- Sublime Text (*https://www.sublimetext.com/*)

Bei den IDEs ist wahrscheinlich PhpStorm (*https://www.jetbrains.com/de-de/phpstorm/*) der bekannteste Vertreter.

22.3.2 Programmierhilfen finden

Unabhängig vom Kenntnisstand: Wer programmiert, muss früher oder später Dinge nachschlagen, weil niemand alles im Kopf haben kann. Für WordPress gibt es im Dev-Hub-Bereich unter *developer.wordpress.org/reference/* eine Referenz aller Funktionen, Hooks, Klassen und Methoden, die aus dem WordPress-Quellcode generiert werden. Zusätzlich zu den Infos aus dem Quellcode gibt es aber weiterführende Informationen und Beispiele.

Wenn Sie für den Block-Editor entwickeln, ist *developer.wordpress.org/block-editor/* eine gute Anlaufstelle, dort finden Sie unter anderem Tutorials und Referenzen.

Falls Sie trotzdem keine Lösung finden, können Sie mit folgenden Ansätzen weiterforschen: die Suche im Code von WordPress oder Plugins/Themes, in Foren oder Beispiele in Blogbeiträgen oder Ähnlichem.

Code vom Core und/oder Plugins und Themes durchsuchen

Wenn Sie eine Stelle in WordPress oder einem Plugin oder Theme kennen, die genau (oder ungefähr) das macht, was Sie umsetzen möchten, können Sie die Stelle im Code suchen und sich anschauen, wie die Funktionalität umgesetzt wurde. Dazu können Sie sich beispielsweise WordPress einfach herunterladen und lokal auf die Suche gehen – mit Plugins und Themes geht das natürlich genauso.

Solange die Plugins und Themes den offiziellen Vorgaben folgen und eine Lizenz verwenden, die mit der von WordPress kompatibel ist, können Sie auch Code kopieren und wiederverwenden (Quellen angeben ist natürlich trotzdem lieb).

Im Fall des Block-Editors lohnt sich ein Blick in das beziehungsweise eine Suche im GitHub-Repository (*https://github.com/wordpress/gutenberg*). Dort liegen die Dateien in der Ursprungsform vor, nicht minifiziert, wie sie im Core ausgeliefert werden.

Ein paar Tipps, wie Sie am besten den Code zu einer Funktion finden, die Sie sich anschauen möchten:

- Wenn es um eine Funktion geht, die sichtbare Ausgaben erzeugt, beispielsweise ein Formular, schauen Sie sich im Browser den Quelltext an, und suchen Sie nach Dingen, die relativ eindeutig sind, wie `id`- oder `class`-Werte. Durchsuchen Sie den Code nun nach diesen Werten, und mit etwas Glück haben Sie die Stelle für die Ausgabe gefunden, von der Sie dann weiter kommen.
- Wenn Sie herausfinden möchten, wie etwas funktioniert, was durch eine API-Anfrage ausgelöst wird, suchen Sie nach der URL, die angefragt wird, und suchen Sie im Quellcode nach einem Teil davon. Wenn Sie finden, wo diese URL beziehungsweise der Endpunkt definiert wurden, müssten Sie auch die dahinterliegende Funktionalität finden, da sie in dem Bereich angesprochen werden muss.
- Wenn Sie den Namen einer Klasse oder Funktion wissen und mehr darüber herausfinden möchten, suchen Sie nach genau diesem Namen, sodass Sie die Stelle der Deklaration finden und sich anschauen können, was für Code ausgeführt wird. Namen

von Variablen sind selten hilfreich für die Suche in einem ganzen Projekt.

22.3.3 Debuggen

Debuggen beschreibt den Vorgang, einen Fehler im Programmcode zu finden und zu beheben. Dafür gibt es sehr nützliche spezielle Tools wie Xdebug, die es erlauben, den Durchlauf eines Programms an einem bestimmten Punkt (am besten kurz vor der Fehlerstelle) zu pausieren, sich alle Werte von Variablen, Globals und Attributen zu diesem Zeitpunkt anzuschauen und dann schrittweise – bei Bedarf wirklich Zeile für Zeile – den Code weiter durchzugehen, um zu schauen, wo etwas aus welchem Grund nicht wie erwartet funktioniert.

WordPress selbst bringt einen Debug-Modus mit, der damit nicht viel zu tun hat, aber deutlich einfacher einzurichten ist und eine Voraussetzung für das Debuggen liefert: die Fehlermeldungen. Der Debug-Modus von WordPress ermöglicht nämlich das Anzeigen und/oder Logging von im PHP-Code auftretenden Problemen, und ohne diese Fehlermeldungen wäre auch kein sinnvolles Debugging möglich, weil unbekannt ist, wo der Fehler auftritt. Häufig reicht die Fehlermeldung auch schon als Information, um den Fehler beheben zu können. Neben dem Debug-Modus und Xdebug werden Sie noch das Plugin *Query Monitor* kennenlernen, das beim Aufspüren von Problemen bei WordPress-Websites hilfreich ist.

Debug-Mode von WordPress

Der Debug-Modus von WordPress wird über drei Konstanten gesteuert, die in der *wp-config.php* gesetzt werden können:

1. `WP_DEBUG`. Diese Konstante werden Sie in Ihrer *wp-config.php* bereits finden, vermutlich auf den Wert `false` gesetzt (die ganze Zeile sieht so aus: `define( 'WP_DEBUG', false );`). Mit dieser Konstante steuern Sie grundsätzlich, ob der Debug-Mode an ist oder nicht.

   Wenn Sie den Wert auf `true` setzen, werden Sie beim Neuladen Ihrer Website, sofern es Fehler oder Warnungen gibt, diese direkt auf der Website angezeigt bekommen.

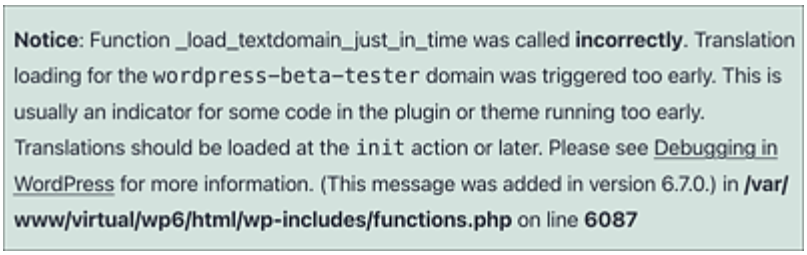


Abbildung 22.6 Ausgabe einer PHP-Warnung auf der Website

2. `WP_DEBUG_LOG` ermöglicht es zu steuern, ob die Fehlermeldungen in einer Log-Datei gespeichert werden sollen, die standardmäßig unter *wp-content/debug.log* angelegt wird. Wenn die Konstante fehlt, wird keine Log-Datei geschrieben.
3. `WP_DEBUG_DISPLAY` erlaubt es Ihnen, die Anzeige von Fehlermeldungen auf der Website zu unterdrücken, auch wenn der Debug-Modus an ist. Das ist eigentlich immer eine gute Idee, weil Fehlermeldungen und Warnungen nicht öffentlich sichtbar sein sollten. Sie müssen dafür den Wert auf `false` setzen.

Wenn Sie auf einer Live-Website einen Fehler haben und die entsprechende Fehlermeldung erhalten möchten, bietet sich die folgende Kombination von Werten an, damit die Fehler geloggt, aber nicht öffentlich angezeigt werden:

```
define( 'WP_DEBUG', true );
define( 'WP_DEBUG_LOG', true );
```

```
define( 'WP_DEBUG_DISPLAY', false );
```

Da die Log-Datei unter *wp-content/debug.log* gespeichert wird, können Sie sie über *example.com/wp-content/debug.log* direkt im Browser aufrufen.

Achtung: Fehler-Logging nicht auf Live-Sites aktiv lassen

Wie oben schon einmal kurz erwähnt, sollten Sie das Debug-Log auf öffentlich zugänglichen Sites nicht einfach aktiv lassen. Die Fehler werden durch die Angabe von `define( 'WP_DEBUG_DISPLAY', false );` zwar nicht mehr direkt im Frontend ausgegeben, aber die *debug.log*-Datei ist von allen aufrufbar und damit ein potenzielles Sicherheitsrisiko. Wenn Sie sich auf einer Live-Site die Fehler zu einem konkreten Problem ausgeben lassen möchten, ist das temporäre Aktivieren des Error-Logs ein valider Weg, allerdings sollten Sie ihn danach wieder deaktivieren und die *debug.log*-Datei löschen.

Wichtige Hintergrundinfos mit dem Plugin »Query Monitor«

Das Plugin Query Monitor (*https://de.wordpress.org/plugins/query-monitor*) ist mehr als nur ein Monitoring-Tool für Datenbankabfragen (Querys).

Wenn Sie das Plugin installieren und aktivieren, finden Sie es im Frontend und Backend in der Werkzeugleiste oben (falls Sie sie fürs Frontend ausgeblendet haben, können Sie sie in Ihrem Profil wieder einblenden). Direkt auf den ersten Blick bekommen Sie hier die Ladezeit (mehr als zwei Sekunden sollten das nicht sein, besser wäre um oder unter eine Sekunde), die maximale Speichernutzung, die Dauer sowie die Anzahl der Datenbankabfragen gezeigt.

Ein Klick auf den Bereich der Werkzeugleiste öffnet den Query-Monitor-Bereich, der am unteren Viewport-Rand angedockt ist. Der Bereich ist in unterschiedliche Tabs unterteilt, die Sie links wechseln können. Die folgenden Tabs gibt es:

- Überblick. Hier sehen Sie ein paar Informationen im Überblick, wie noch mal die Daten, die in der Werkzeugleiste angezeigt werden, und Daten zum Cache.
- Datenbankabfragen. Hier sehen Sie alle Datenbankabfragen für die aktuell angezeigte Seite, zuzüglich der Informationen, von wo die Abfrage ausgeführt wurde, wie lange sie gedauert hat und wie viele Ergebnisse zurückgeliefert wurden. Diese Ansicht ist sehr praktisch, wenn Sie schauen möchten, ob langsame Datenbankabfragen existieren.

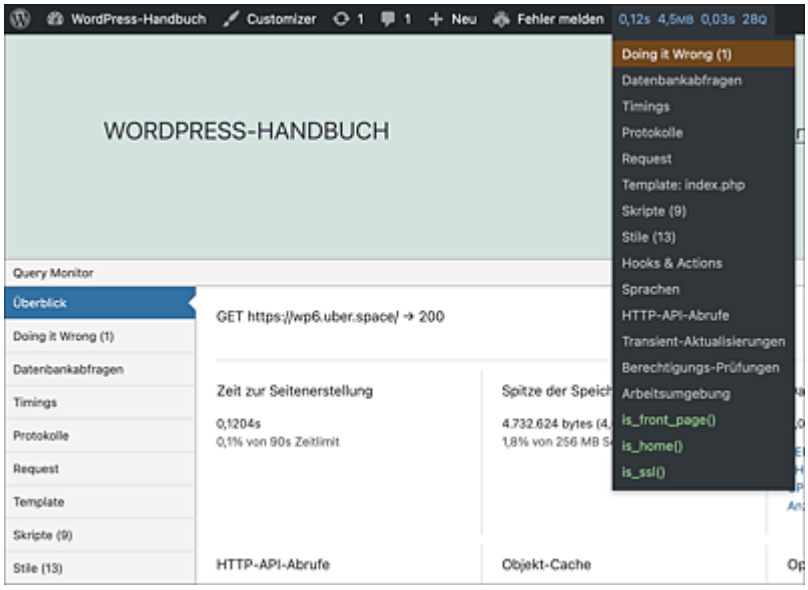


Abbildung 22.7 Das Query-Monitor-Plugin gibt Ihnen nützliche Einblicke hinter die Kulissen eines Requests.

- Timings. Hier werden Sie standardmäßig nichts finden, außer einen Hinweis auf die Docs von Query Monitor (*https://querymonitor.com/wordpress-debugging/profiling-and-logging/*). Sie können selbst im Code Funktionsaufrufe verteilen,

um bestimmte Passagen von Query-Monitor messen zu lassen. Diese Ergebnisse sehen Sie dann im Timings-Tab.

- Protokolle. Ähnlich wie der Timings-Tab ist auch hier normalerweise nichts zu sehen. Sie können über von Query Monitor mitgelieferte Actions-Werte aus dem Code loggen, ähnlich wie `console.log` in JavaScript, die dann hier angezeigt würden. Wie das geht, steht auf derselben Seite in den Docs wie die Informationen zum Timings-Tab.
- Request. Hier finden Sie Informationen zum Seiten-Request, wie welches Objekt angefragt wurde.
- Blocks. Hier sehen Sie bei Block-Themes eine Liste der Blöcke, aus der die Seite besteht, inklusive der Attribute, eines möglichen Kontexts und optionaler Render-Callbacks.
- Template. Zeigt Ihnen Informationen zum aktuellen Theme an, zum geladenen Template und zur Template-Hierarchie für die aktuelle Ansicht.
- Skripte. Listet alle JavaScripte auf, die eingebunden wurden.
- Stile macht dasselbe für alle CSS-Dateien.
- Unter Hooks & Actions sehen Sie die ausgeführten Hooks und die Aktionen, die daraufhin ausgeführt wurden.
- Sprachen liefert wichtige Informationen, wenn Sie Probleme mit nicht ladenden Übersetzungen haben.
- HTTP-API-Abrufe zeigt an, welche Aufrufe an APIs gemacht wurden.
- Unter Transient-Aktualisierungen finden Sie neu gesetzte Transienten.

- Berechtigungsprüfungen werden aus Performance-Gründen standardmäßig nicht geloggt. Wenn Sie das benötigen, können Sie es über die Konstante `QM_ENABLE_CAPS_PANEL` aktivieren.
- Hinter Arbeitsumgebung verbergen sich Infos zum Webserver und der Konfiguration, beispielsweise die PHP-Version.
- Bedingungen zeigt Ihnen an, welche WordPress-Bedingungen für die aktuelle Seite erfüllt werden, beispielsweise `is_single()` für die Einzelansicht eines Beitrags.

Kurz vorgestellt: mächtiges Debugging-Tool Xdebug

Xdebug (*https://xdebug.org*) ist eine PHP-Erweiterung, die unterschiedliche Funktionen fürs Debugging mitbringt. Ich (Florian) nutze fast ausschließlich das »Step Debugging«. Diese Funktion ermöglicht das Setzen von sogenannten Breakpoints im PHP-Code, an dem die Ausführung des Codes angehalten wird, sodass Sie sich unter anderem anschauen können, welche Werte Variablen und Globals zu diesem Zeitpunkt haben, und Sie können von einem Breakpoint auch Schritt für Schritt weiter durch den Code gehen.

Die Installation des Tools kann sich je nach Umgebung als etwas komplizierter herausstellen, für `wp-now` gibt es aktuell (November 2024) noch keine Unterstützung, aber zumindest schon mal ein Issue (*https://github.com/WordPress/wordpress-playground/issues/314*).

23 Theme entwickeln

Wenn Sie eine Website erstellen und es um die Umsetzung des Designs geht, gibt es zwei Wege: Sie finden ein Theme, das Ihren Vorstellungen (fast) entspricht und das Sie anpassen. Oder Sie erstellen Ihr eigenes Theme.

Wenn Sie ein Theme finden, das Ihren Wünschen genau oder beinahe entspricht, ist es wahrscheinlich deutlich einfacher, das Theme als Ausgangspunkt zu nutzen, als ein eigenes Theme zu entwickeln. Allerdings werden Sie in der Regel kein Theme finden, das nur genau die Anforderungen erfüllt, die Sie haben – meist werden für Ihren Fall »überflüssige« Elemente mit an Bord sein.

Wenn Sie Ihre Website sehr stark auf Performance, also eine schnelle Ladezeit, optimieren möchten, kann es dann schnell doch aufwendiger werden. Eine generelle Empfehlung kann hier nicht gegeben werden, Sie müssen von Fall zu Fall schauen, ob eine bestehende Lösung genutzt und angepasst werden kann oder ob eine Eigenentwicklung sinnvoller ist.

Wie bereits an anderer Stelle erwähnt, kann dieses Buch neben den WordPress-spezifischen Entwicklungsthemen nicht auch eine Einleitung in die Grundlagen von HTML, CSS, JavaScript und PHP geben, weshalb Grundkenntnisse vorausgesetzt werden.

23.1 Block-Theme erstellen

Block-Themes sind ein nicht mehr ganz so neues Konzept, das in WordPress 5.9 mit dem ersten großen Schritt Richtung *Full-Site-Editing* eingeführt wurde, das die Bearbeitung der gesamten Site über das Backend erlaubt, ohne Code anpassen zu müssen. In Abschnitt 10.1.2 wurde bereits auf die Besonderheiten des Block-Theme-Konzepts eingegangen, und in Abschnitt 10.3, »Block-Theme anpassen: den Website-Editor kennenlernen«, wurde der Website-Editor vorgestellt, mit dem Block-Themes verändert werden können. Deshalb geht es in diesem Abschnitt nur um die Erstellung eines Block-Themes.

Wenn Sie ein Classic Theme anpassen oder erstellen möchten, lesen Sie in Abschnitt 23.2, »Classic Theme anpassen und erstellen«, weiter.

23.1.1 Grundlagen zur Theme-Entwicklung

Anders als ein Classic Theme kann ein Block-Theme fast ohne das Schreiben von Code direkt über das Backend erstellt werden. Die notwendigen Dateien für ein Block-Theme, damit WordPress es erkennt, sind eine *style.css* und eine *templates/index.html*, die das Index-Template beinhaltet. Die großen Unterschiede zu Classic Themes:

- Block-Theme-Templates bestehen aus HTML-Markup von Blöcken. PHP gibt es nur in der optionalen *functions.php* oder dort eingebundenen PHP-Dateien.
- Block-Themes kümmern sich nur um das, was im `body` passiert, das HTML-Gerüst aus Doctype, `html`, `head` und `body` wird von WordPress generiert, und auch um die Ausführung der Action-Hooks `wp_head`, `wp_body_open` und `wp_footer` kümmert sich

WordPress automatisch. Darüber hinaus erstellt WordPress automatisch einen Skip-to-Content-Link.

- Template-Dateien werden im *templates*-Verzeichnis abgespeichert, Template-Teile in *parts* (wie Sie in einem Kasten im Classic-Theme-Abschnitt lernen werden, können seit WordPress 6.1 auch Classic Themes die *parts*-Template-Teile nutzen). In beiden Fällen werden HTML-Dateien verwendet, die Benennung der Template-Dateien gleicht aber ansonsten der bei den Classic Themes, es greift dieselbe Template-Hierarchie (mehr Infos dazu im Abschnitt zur Classic-Theme-Entwicklung).

Kurz: Sie können sich bei einem Block-Theme ganz auf die Ausgabe der Inhalte konzentrieren. Und auch hierfür müssen Sie eigentlich keinen Code schreiben, denn Sie können die Templates und Template-Teile über den Website-Editor erstellen, nachdem Sie ein minimales Grundgerüst erstellt haben, damit das Block-Theme in WordPress aktiviert werden kann. Danach können Sie die Templates und Template-Teile, die von WordPress in der Datenbank gespeichert werden, als Theme-Dateien exportieren und in Ihr Theme integrieren. Und genau so werden wir im nächsten Abschnitt vorgehen.

23.1.2 Block-Theme umsetzen

In diesem Abschnitt wird es darum gehen, ein einfaches WordPress-Block-Theme zu erstellen. Seit WordPress 6.2 gibt es das »Stilbuch«, eine Funktion in der Stil-Sidebar im Website-Editor, mit der Sie sich alle verfügbaren Blöcke im aktuellen Design anschauen können. Daher brauchen Sie keine Seite oder einen Beitrag mit allen Blöcken erstellen, wie es beim Classic Theme eine gute Idee wäre.

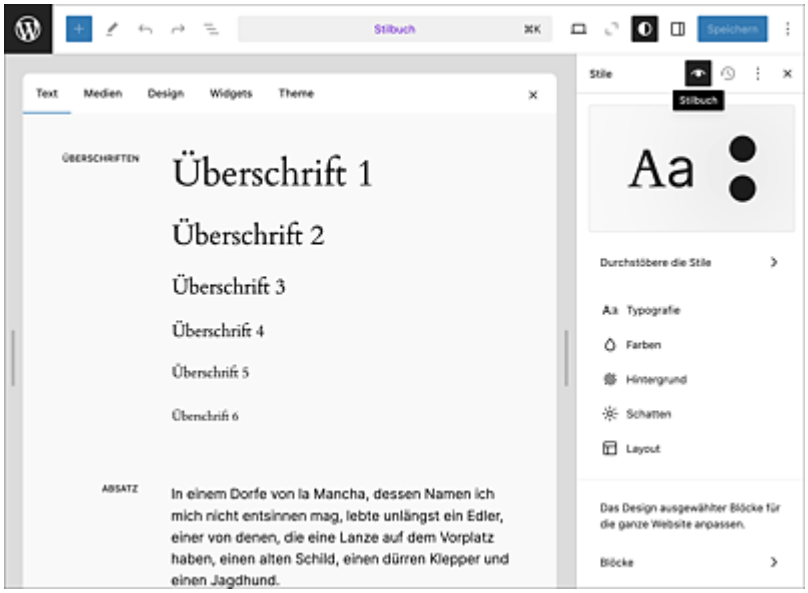


Abbildung 23.1 Mit dem Stilbuch haben Sie eine schöne Übersicht über alle verfügbaren Blöcke und deren Aussehen.

Theme-Grundgerüst aus style.css und templates/index.html

Ein Block-Theme braucht mindestens eine *style.css* und das Index-Template *templates/index.html* (bis WordPress 6 war zusätzlich eine *index.php* notwendig). Erstellen Sie unter *wp-content/themes* einen neuen Ordner *wh-block* (Sie können natürlich einen anderen Ordnernamen wählen), und legen Sie die zwei Dateien an, wie in Abbildung 23.2 zu sehen.

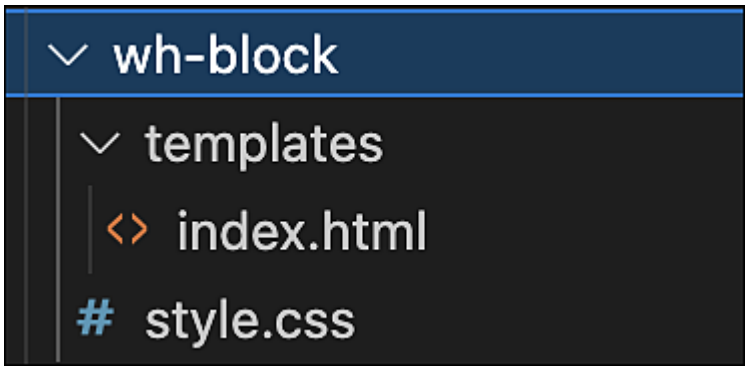


Abbildung 23.2 So muss die Datei- und Ordnerstruktur des Block-Themes aussehen, damit WordPress es erkennen kann.

Die *index.html* bleibt leer, nur die *style.css* benötigt den Kommentar mit den Metainformationen, damit das Theme im WordPress-Backend angezeigt wird.

```
/*
Theme Name: WH Block
Author: Ihr Name
Description: Simple theme used to show how to create a block theme.
Requires at least: 6.0
Tested up to: 6.6
Requires PHP: 7.0
Version: 1.0
License: GNU General Public License v2 or later
License URI: http://www.gnu.org/licenses/gpl-2.0.html
Text Domain: wh-block
*/
```

Listing 23.1 Die Metadaten, die in der »style.css« angegeben werden können, sind dieselben wie bei einem Classic Theme.

Jetzt sollten Sie das Theme im Backend unter Design • Themes sehen und aktivieren können (siehe Abbildung 23.3).

Nach der Aktivierung sehen Sie im Admin-Menü unter Design den Eintrag Website-Editor, ein sicheres Zeichen dafür, dass das Anlegen des Themes als Block-Theme funktioniert hat. In diesem Editor spielt sich die Erstellung des Themes ab.

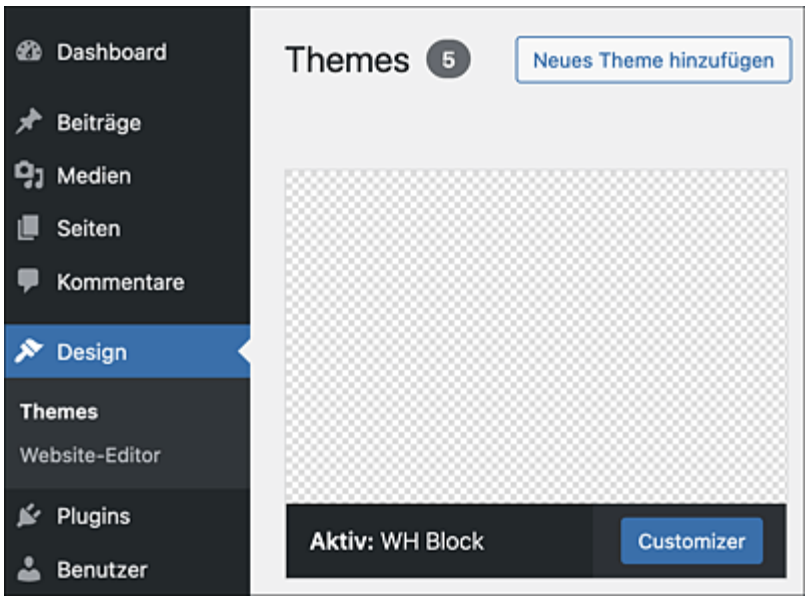


Abbildung 23.3 Das Theme-Grundgerüst kann bereits in WordPress aktiviert werden. Dass es ein Block-Theme ist, fällt direkt aufgrund des Eintrags »Design« • »Website-Editor« links im Admin-Menü auf.

Index-Template erstellen

Wenn Sie den Website-Editor öffnen, werden Sie ein leeres Index-Template sehen, das darauf wartet, mit Blöcken gefüllt zu werden (siehe Abbildung 23.4).

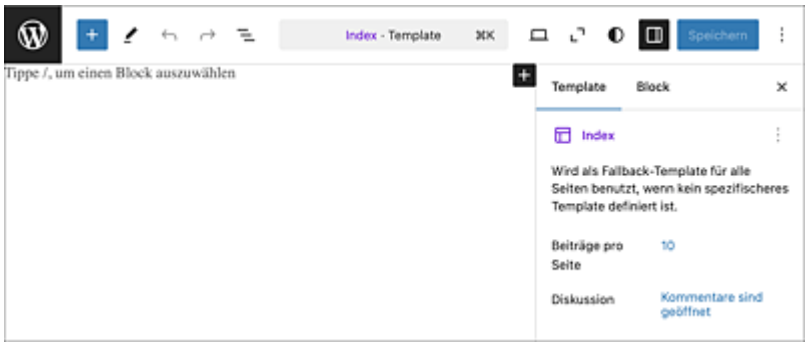


Abbildung 23.4 Nach dem Öffnen des Website-Editors werden Sie von einem leeren Index-Template begrüßt.

1. **Header-Template-Teil einfügen**
   Fangen wir beim Befüllen des Templates oben an, beim Header. Der wird in alle Templates eingebunden, das bedeutet, dass er am besten mit einem Template-Teil-Block umgesetzt wird. Den Inhalt solcher Blöcke speichert WordPress gesondert ab, sodass sie wiederverwendet werden können. Wenn Sie »Template-Teil« in den Block-Inserter eintippen, bekommen Sie neben diesem Block noch den Header- und Footer-Block angezeigt, die ebenfalls Template-Teile erstellen und anzeigen können, die dann der entsprechenden Gruppe Header oder Footer zugeteilt werden (dadurch wird beispielsweise das Wrapper-Element des Template-Teils auf das richtige Element eingestellt, `header` bzw. `footer`). Wir wählen einen Header-Block aus und fügen ihn ein (siehe Abbildung 23.5).



Abbildung 23.5 Mit dem »Header«-Block wird der Template-Teil für den Kopfbereich der Website erstellt.

Nachdem der Block eingefügt wurde, klicken Sie im Block auf den Button Leer beginnen (würden schon Template-Teile existieren, könnten Sie die hier auswählen) und vergeben dem Template-Teil den Namen »Header (h1)«. In den Template-Teil fügen Sie die folgenden Blöcke hinzu:

- Gruppe, um den Blöcken innerhalb des Headers die Breite zu geben, die später in der *theme.json* definiert wird. Die weiteren Blöcke gehören in den Gruppen-Block.
- Website-Titel zur Anzeige und zum Setzen des Website-Titels (wählen Sie als Element für den Titel eine `h1`).
- Website-Untertitel zur Darstellung und Anpassung der Website-Beschreibung.
- Navigation. Wenn Sie den Block einfügen, wird ein Menü mit der Seitenliste eingefügt. Dieses Menü können Sie im Editor bearbeiten, der Block und die Optionen sind in Kapitel 6 beschrieben worden.

Richten Sie alle Blöcke mittig aus, und es sollte ungefähr aussehen wie in Abbildung 23.6 (für die Navigation ist das Overlay-Menü deaktiviert, da für die konkrete Website nur wenige Menüpunkte auf einer Ebene eingeplant sind).



Abbildung 23.6 Der Header-Template-Teil im Website-Editor

Speichern Sie das Ergebnis und öffnen Sie das Frontend, sollten Sie ungefähr dasselbe sehen wie im Editor.

2. **Beitragsliste anzeigen**
   Der nächste Schritt ist die Ausgabe der Beitragsliste. Dafür gibt es den Abfrage-Loop-Block, bei dem Sie nach Einfügen auf LEER BEGINNEN klicken und aus den dann angezeigten Varianten TITEL, DATUM UND TEXTAUSZUG auswählen (siehe Abbildung 23.7).

   In dem eingefügten Block können Sie direkt dort, wo der Weiterlesen-Link angezeigt wird, im entsprechenden Block den Text ändern und ganz unten einen Text eingeben, der angezeigt wird, wenn keine Beiträge gefunden wurden. Darüber hinaus sollten Sie für den Titel- und Datum-Block einstellen, dass sie verlinkt ausgegeben werden.



Abbildung 23.7 Mit dem »Abfrage-Loop«-Block erstellen Sie die Liste der neuesten Beiträge.

Beim Abfrage-Loop-Block selbst müssen Sie in der Seitenleiste den ABFRAGETYP auf STANDARD stellen (die Option ist vorausgewählt), sodass passend zur aufgerufenen Ansicht im Frontend die richtigen Beiträge ausgegeben werden. Darüber hinaus können Sie im Bereich ERWEITERT in der Block-Seitenleiste das HTML-Element des Blocks auf `main` ändern, um den Hauptinhalt der Webseite zu kennzeichnen.

Fügen Sie nun im Abfrage-Loop-Block unter dem Beitragsdatum noch den Block für das Beitragsbild ein.

3. **Footer-Template-Teil einfügen**
   Fehlt noch der Footer zum kompletten Index-Template. Dafür fügen Sie einen Footer-Block ein, wählen wie beim Header LEER BEGINNEN, vergeben einen Namen und ergänzen zunächst wieder einen Gruppen-Block, der das HTML-Element `footer` nutzen soll . Innerhalb der Gruppe fügen Sie dann beliebige Blöcke für den Footer ein. Im Frontend sieht die Startseite nun hoffentlich ungefähr aus wie der Screenshot in Abbildung 23.8.



Abbildung 23.8 Das Index-Template mit Header, Beitragsliste und Footer im Frontend

Template für Seiten

Kümmern wir uns als Nächstes um das Template für die Einzelansicht von Seiten. Dazu wechseln Sie im Website-Editor in die Liste der Templates (links oben das WordPress-Logo anklicken und in der Seitenleiste Templates auswählen) und klicken oben rechts auf Neues Template hinzufügen, um ein neues Template zu erstellen. In dem aufploppenden Menü wählen Sie den Punkt Seiten aus und im folgenden Dialog Alle Seiten, da Sie kein Template nur für eine bestimmte Seite erstellen möchten. Ist das geschafft, öffnet sich erneut der Website-Editor mit leerem Layout (überspringen Sie die Wahl einer Vorlage).

Da Header und Footer als Template-Teile angelegt wurden, könnten Sie beim Template-Teil-Block jetzt über den Button Auswählen einfach die erstellten Template-Teile auswählen und einfügen. Schneller geht es, die Teile über ihren Namen im Inserter zu suchen und einzufügen, als wären es eigene Blöcke. Der allgemeine Header-Block wird, zumindest in WordPress 6.7-RC4, nicht mehr angezeigt.

Da dies ein Template für eine Einzelansicht ist, soll der Website-Titel im Header nicht als `h1` ausgegeben werden, sondern als einfacher Absatz. Dafür müssen Sie einen neuen Template-Teil erstellen, können den Inhalt des bestehenden Template-Teils aber als Basis nehmen. Wählen Sie dazu den Template-Teil im Editor aus, klicken Sie in der Block-Toolbar auf das Icon mit den drei Punkten ganz rechts, und wählen Sie den Punkt Loslösen aus (siehe Abbildung 23.9).

Jetzt haben Sie die einzelnen Blöcke im Editor – ohne den Template-Teil-Block drum herum. Wählen Sie den Gruppen-Block aus, der die Blöcke des Headers umgibt, öffnen Sie wieder das Drei-Punkte-Menü, und klicken Sie diesmal auf den Punkt Template-Teil erstellen (siehe Abbildung 23.10). Vergeben Sie den Namen »Header (p)«, und wählen Sie als Bereich Header aus. Ist das geschafft, wählen Sie den

Website-Titel-Block aus und ändern das HTML-Element über die Block-Toolbar zu einem Absatz.

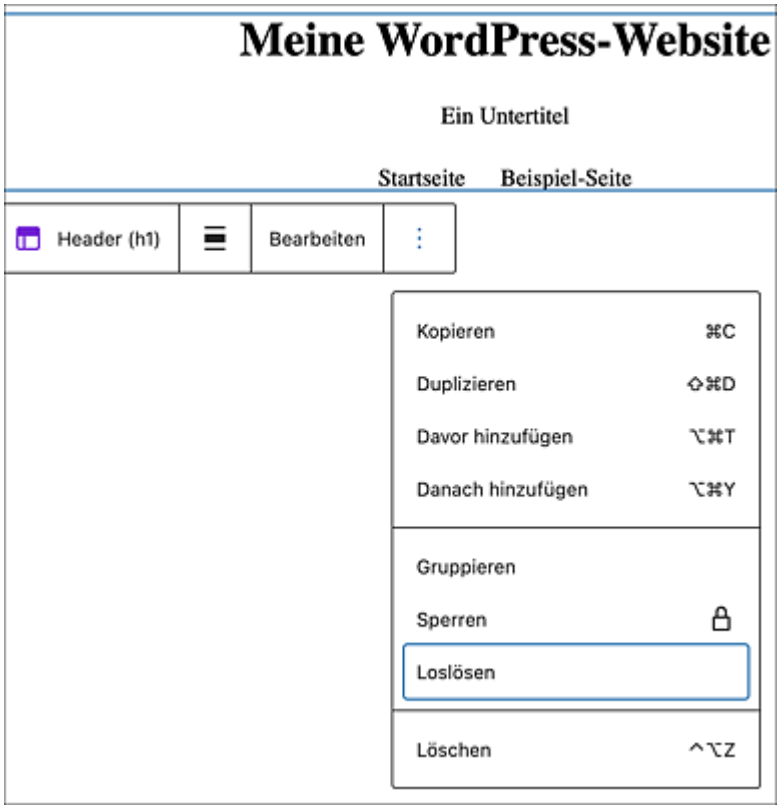


Abbildung 23.9 Ein Template-Teil kann recht einfach in seine Blöcke aufgelöst werden, sodass sie verändert werden können, ohne den Template-Teil selbst anpassen zu müssen.

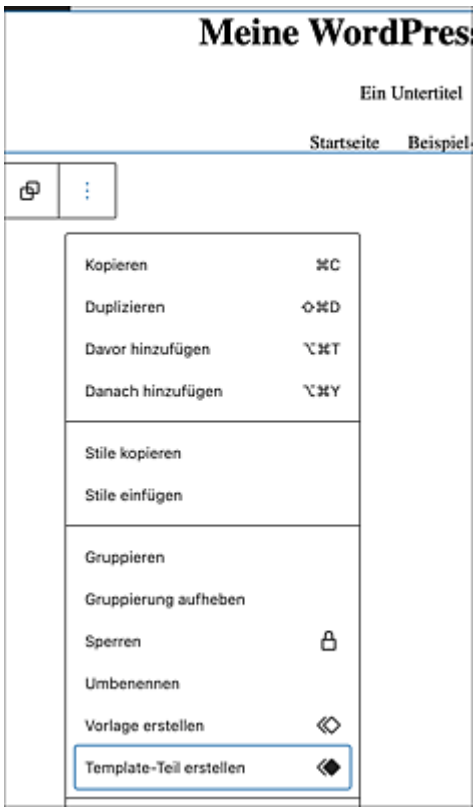


Abbildung 23.10 Ausgewählte Blöcke können in einen Template-Teil umgewandelt werden.

Zur Darstellung des Seitentitels, -inhalts und des Kommentarbereichs kommen wir nun zu einer kleinen Verschachtelungsaufgabe:

- Erstellen Sie zwischen Header und Footer einen Gruppen-Block und wählen Sie rechts in der Block-Seitenleiste unter ERWEITERT als HTML-ELEMENT den Wert `<main>` aus.
- In die Gruppe fügen Sie zwei weitere Gruppen-Blöcke ein (wählen Sie für den ersten als HTML-ELEMENT `article` aus, für den zweiten behalten Sie den Standard bei). Es ist etwas tricky, den zweiten Block auf dieselbe Ebene zu bekommen wie den `article`-Gruppen-Block. Am einfachsten geht es, wenn Sie den Gruppen-Block auswählen und über das Drei-Punkte-Menü die Aktion DANACH EINFÜGEN auswählen.

Tipp: Listenansicht der Blöcke im Editor anzeigen

Wenn es an mehrfach verschachtelte Blöcke geht, kann die Übersichtlichkeit im Editor leiden. Ein nützliches Werkzeug ist die Listenansicht der Blöcke, die im Editor eingefügt sind, in der auch die Hierarchie abgebildet wird. Anzeigen lassen können Sie sich die Liste mit einem Klick auf das Icon mit den drei zueinander versetzten Strichen rechts neben dem »Wiederholen«-Pfeil in der Toolbar.

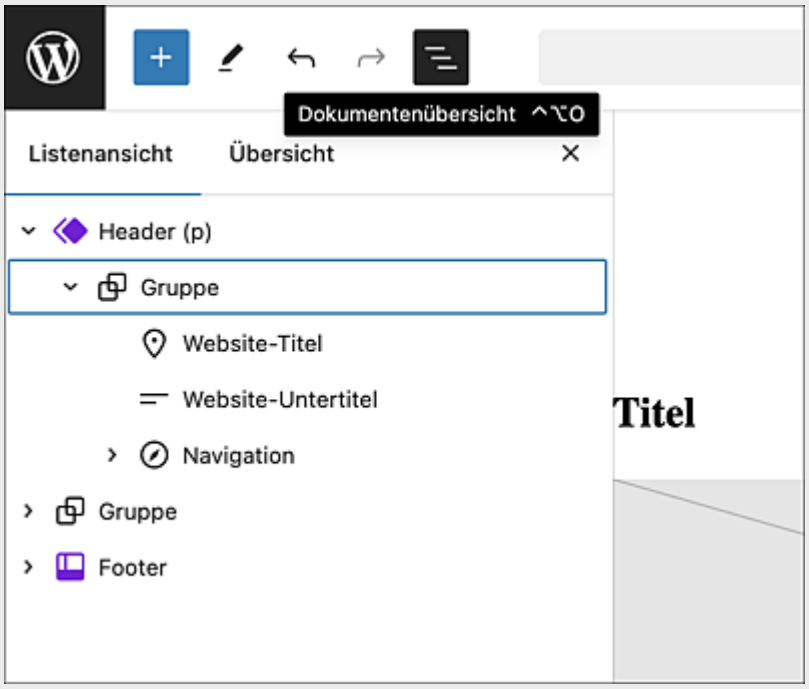


Abbildung 23.11 Die Listenansicht der eingefügten Blöcke hilft bei der Übersicht mit verschachtelten Blöcken.

- Fügen Sie in den `article`-Gruppen-Block einen weiteren Gruppen-Block ein, für den Sie das `header`-Element auswählen, und den Block INHALT.
- In den `header`-Gruppen-Block fügen Sie jetzt noch die Blöcke TITEL und BEITRAGSBILD ein. Wählen Sie für den Titel-Block über die Block-Toolbar die `h1` als Überschriftenebene.

- In den Gruppen-Block, den Sie auf derselben Ebene wie den `article`-Gruppen-Block eingefügt haben, fügen Sie abschließend den Block Kommentare ein.

Nachdem Sie das Template gespeichert haben und im Frontend eine Seite aufrufen, sollten Sie etwas sehen, das Abbildung 23.12 ähnelt.



Abbildung 23.12 Das Template für Seiten in Aktion

Template für die Einzelansicht von Beiträgen

Da das Template für die Einzelansicht der Beiträge ähnlich ist wie das zur Anzeige einer Seite, öffnen Sie zunächst im Editor noch einmal das Seiten-Template und kopieren alle Blöcke des Templates. Öffnen Sie dazu oben rechts das Drei-Punkte-Menü, und wählen Sie die Aktion Alle Blöcke kopieren.

Jetzt erstellen Sie das leere Template für die Einzelansicht von Beiträgen und gehen dabei genauso vor wie bei dem Template für Seiten:

1. Klicken Sie innerhalb des Website-Editors links oben auf das Logo, und navigieren Sie über Templates • Alle Templates

(vielleicht befinden Sie sich auch direkt bei Alle Templates) zur Liste der vorhandenen Templates.

2. Legen Sie über den Button Neues Template hinzufügen rechts oben ein Template für Einzelne Beiträge an.
3. Im sich öffnenden Editor-Fenster überspringen Sie wieder die Auswahl einer Vorlage und haben das leere Template vor sich.
4. Fügen Sie die kopierten Blöcke des Seiten-Templates mit Strg/cmd+V in den Editor ein.

Fügen Sie nach dem Beitragstitel jetzt den Block Datum ein (auf derselben Ebene wie den Titel, also innerhalb des `header`-Gruppen-Blocks) und nach dem Inhalt einen neuen Gruppen-Block, der ein `footer` sein soll und in den Sie den Block Kategorien einfügen. Wenn Sie nun einen Beitrag im Frontend öffnen, sollten Sie in etwa das in Abbildung 23.13 Gezeigte sehen.



Abbildung 23.13 Die Einzelansicht eines Beitrags im Frontend

Die Templates für Archivübersichten und Suchergebnisse

Erstellen Sie aus der Templates-Liste heraus das Template Alle Archive, und füllen Sie es mit einer Kopie der Blöcke aus dem Index-

Template. Da der Archivtitel auf der Seite die `h1` ist, muss der Header-Template-Teil ausgetauscht werden. Wählen Sie dazu den Template-Teil im Editor aus, und klicken Sie im Drei-Punkte-Menü auf Ersetzen, um den Template-Teil Header (p) auszuwählen.

Jetzt müssen noch der Archivtitel und die gegebenenfalls vorhandene Beschreibung des Begriffs angezeigt werden, für den das Archiv ist. Fügen Sie dafür im Abfrage-Loop-Block vor dem Beitrags-Template-Block eine Gruppe ein, für die Sie als HTML-Element das `header`-Element wählen, und fügen Sie in die Gruppe die Blöcke Archiv-Titel und Begriffsbeschreibung ein.

Kopieren Sie nun die Blöcke aus dem Archiv-Template in die Zwischenablage, damit Sie sie als Ausgangspunkt für das Such-Template nutzen können. Erstellen Sie danach aus der Templates-Liste im Website-Editor das Template für die Suchergebnisse, und fügen Sie die Blöcke dort ein. Fügen Sie statt des Archivtitels und der Begriffsbeschreibung den Block Titel der Suchergebnisse ein. Gehen Sie nun noch ans Ende des Abfrage-Loop-Blocks, und fügen Sie einen Suchen-Block mit in den Teil ein, der angezeigt wird, wenn keine Ergebnisse gefunden wurden (siehe Abbildung 23.14).

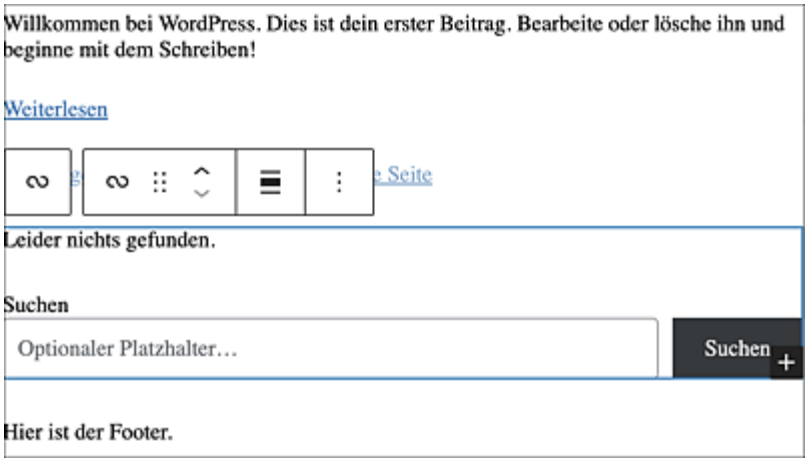


Abbildung 23.14 Wenn keine Beiträge für eine Suche gefunden wurden, wird ein Suchformular angeboten, um es noch einmal zu versuchen.

Inhalt nicht vorhanden, 404

Als Letztes fehlt noch das 404-Template, das angezeigt wird, wenn zu einer Anfrage kein passender Inhalt gefunden werden kann. Kopieren Sie als Basis dafür die Blöcke aus dem Seiten-Template, erstellen Sie das 404-Template aus der Templates-Liste heraus und fügen Sie die Blöcke ein. Ersetzen Sie den Beitragstitel-Block durch eine Überschrift des Typs h1, die etwas enthält wie »Nichts gefunden«, und löschen Sie den Beitragsbild-Block.

Ersetzen Sie außerdem den Block mit dem Seiteninhalt durch eine Gruppe, der Sie im ERWEITERT-Panel der Block-Seitenleiste die Klasse `entry-content` mitgeben, und fügen Sie in die Gruppe alle Blöcke ein, die Sie anbieten möchten, um doch noch den gewünschten Inhalt zu finden, beispielsweise ein Suchformular. Entfernen Sie abschließend den Block mit dem Kommentarbereich, und speichern Sie das Template (siehe Abbildung 23.15).

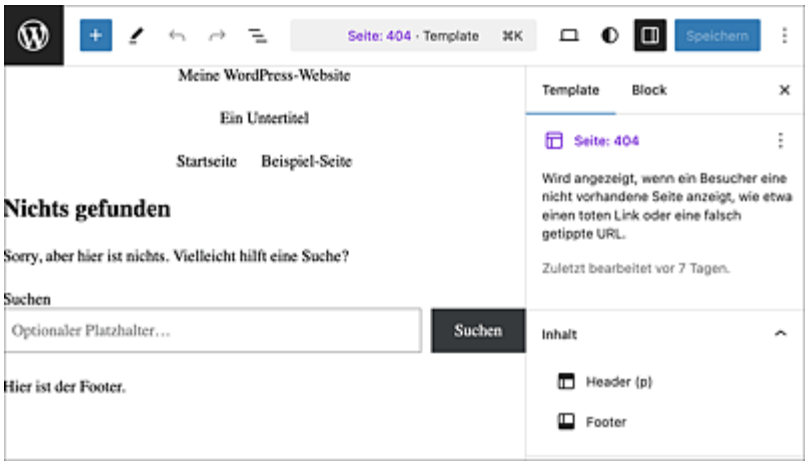


Abbildung 23.15 Mit dem 404-Template sind alle Templates erstellt, die unser Theme haben soll.

Wenn Sie die Website jetzt im Frontend anschauen und durch die unterschiedlichen Inhalte springen, sollte für alle ein sinnvolles Template existieren.

Theme-Dateien exportieren

Nachdem jetzt alle Templates erstellt sind, ist die Arbeit im Website-Editor getan. Bisher sind die Templates und Template-Teile nur in der Datenbank gespeichert, was die Nutzung des Themes für andere Websites schwierig macht. Um das Problem zu lösen und die Templates als Dateien in den Theme-Ordner einfügen zu können, nutzen Sie jetzt die Exportfunktion des Website-Editors. Gehen Sie dafür in die Bearbeitungsansicht eines beliebigen Templates, öffnen Sie oben rechts das Menü mit den drei Punkten, und wählen Sie den Punkt Exportieren aus (siehe Abbildung 23.16).

Der Export besteht aus einem ZIP-Archiv, das die Template- und Template-Teil-Dateien enthält sowie eine *style.css* und eine *theme.json*.

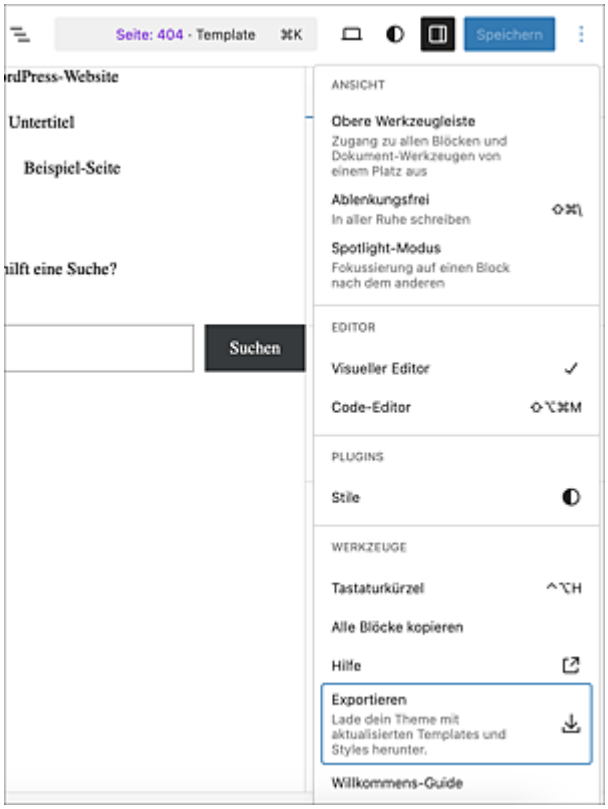


Abbildung 23.16 Der aktuelle Stand des Themes lässt sich über den Website-Editor exportieren.

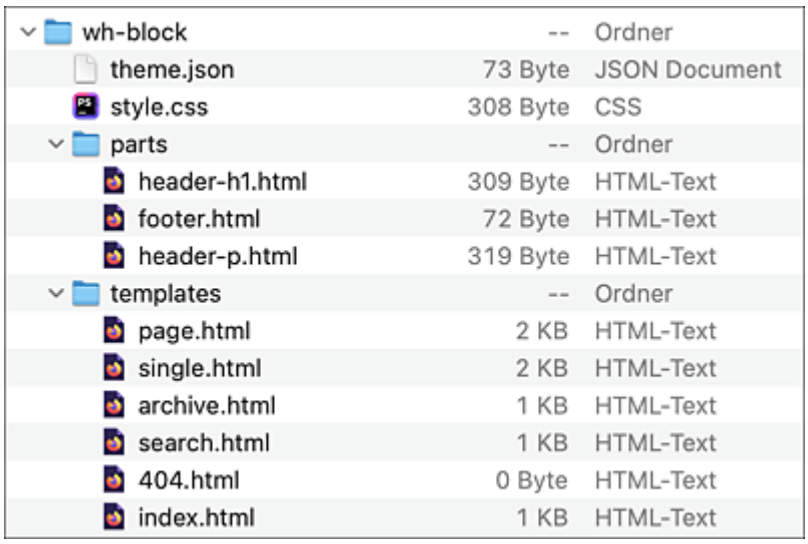


Abbildung 23.17 Der entpackte Ordner enthält alle Dateien, die für ein Block-Theme notwendig sind.

Kopieren Sie alle Dateien in den *wh-blocks*-Ordner im Theme-Verzeichnis, und überschreiben Sie die vorhandenen Dateien (*style.css* und die leere Template-Datei *index.html*). Jetzt könnten Sie das Theme als ZIP in einer anderen WordPress-Installation hochladen und es mit den erstellten Templates nutzen.

Referenzen aus den Template-Dateien entfernen

Wenn Sie die Template-Dateien heruntergeladen und in Ihr Theme-Verzeichnis kopiert haben, müssen noch ein paar Referenzen auf konkrete Inhalte entfernt werden. Andernfalls wird nach der Installation Ihres Themes auf Websites, auf denen dieser Inhalt nicht existiert, ein Fehler ausgegeben, wie in Abbildung 23.18 am Beispiel eines Navigations-Blocks zu sehen ist, der in der Template-Datei mit einem Menü verknüpft ist, das es auf der Website nicht gibt, auf der das fertige Theme installiert wurde.



Abbildung 23.18 So sieht ein Navigations-Block aus, für den in der Template-Datei eine Menü-ID angegeben wurde, die es auf dieser Website nicht gibt.

In dem hier erstellten Block-Theme ist die einzige Referenz tatsächlich die Navigation in den beiden Header-Template-Teilen. Wenn Sie eine der Dateien öffnen und sich das Markup für den Navigations-Block anschauen, sehen Sie so etwas:

```
<!-- wp:navigation {"ref":1, "overlayMenu":"never", "layout":{"type":"flex",
"justifyContent":"center"}} /-->
```

Listing 23.2 Im Navigations-Block wird mit dem »ref«-Attribut auf eine Menü-ID verwiesen.

Problematisch in dem Code ist `"ref":1`, damit verweist der Block auf ein konkretes Menü, das wahrscheinlich auf anderen Websites nicht vorliegen wird. Entfernen Sie den Teil aus beiden Header-Template-Teilen. Alternativ können Sie vor dem Exportieren des Themes aus dem Site-Editor natürlich auch dort alle externen Referenzen durchgehen und entfernen, damit sie gar nicht in den Template-Dateien landen.

Theme mit der theme.json konfigurieren

Eine weitere recht neue Funktion im Bereich Themes ist die *theme.json*, mit der ein Theme viele Aspekte des Block-Editors konfigurieren kann, für die vorher `add_theme_support()`-Aufrufe verwendet wurden, die aber die Möglichkeiten der *theme.json* nicht komplett abbilden.

Sie können in der Datei unter anderem granular Block-Optionen setzen, beispielsweise die verfügbaren Farbpaletten oder Layoutoptionen, aber auch die Layoutbreite bestimmen und Schriften einbinden. Beim Export des Themes aus dem Editor wurde bereits eine *theme.json*-Datei erstellt, die zwei Einträge enthält:

```
{
    "$schema": "https://schemas.wp.org/wp/6.7/theme.json",
    "version": 3
}
```

Listing 23.3 Die beim Theme-Export aus dem Website-Editor generierte »theme.json«-Datei

Mit dem `$schema`-Key wird das JSON-Schema angegeben, das für die Konfigurationsdatei gilt. Dieses Schema definiert, welche Einträge vorkommen können und welche Werte sie haben dürfen. Es gibt Editoren, beispielsweise Visual Studio Code, die diese Angabe nutzen, um eine Autovervollständigung anzubieten. Die `version` der *theme.json* ist die aktuelle Version 3.

Neben diesen Einträgen spielt sich das meiste innerhalb der Bereiche `settings` und `styles` ab. In `settings` legen Sie Werte für Optionen fest, beispielsweise die Inhaltsbreite im Editor, welche Optionen für Blöcke mit welchen Auswahlmöglichkeiten vorhanden sind und anderes. In `styles` geben Sie Standardwerte für Blöcke an, also beispielsweise welche Textfarbe ein Block standardmäßig haben soll. Viele der Konfigurationsmöglichkeiten generieren direkt passendes CSS und setzen Klassen, sodass Sie sich nicht darum kümmern müssen.

Wenn nicht unterhalb des `blocks`-Schlüssels definiert (weiter hinten in diesem Abschnitt wird es ein Beispiel dafür geben), beziehen sich die Konfigurationen innerhalb von `settings` und `styles` auf alle Blöcke. In den nachfolgenden Abschnitten werden Sie verschiedene Optionen kennenlernen, auch wenn nicht alle in der *theme.json*

möglichen Einstellungen gezeigt werden. Im Block-Editor-Handbuch gibt es eine Seite mit den verfügbaren Optionen und Werten (*https://developer.wordpress.org/block-editor/how-to-guides/themes/theme-json/*).

- **Layoutbreite definieren**
  Dank der häufigen Nutzung des Gruppen-Blocks zum Umschließen von Inhaltsblöcken kann jetzt über die *theme.json* die Layoutbreite eingestellt werden, sodass sie sich auch auf den Editor auswirkt (neben dem Website-Editor auch auf den normalen Inhaltseditor). Erweitern Sie die *theme.json* dazu wie in Listing 23.4 gezeigt.

```
{
  "$schema": "https://schemas.wp.org/wp/6.7/theme.json",
  "version": 3,
  "settings": {
    "layout": {
      "contentSize": "30rem",
      "wideSize": "40rem"
    }
  }
}
```

Listing 23.4 Über »settings.layout« kann die normale Inhaltsbreite (»contentSize«) und die für Blöcke mit weiter Breite (»wideSize«) eingestellt werden.

Durch die Angabe von `contentSize` und `wideSize` unter `settings.layout` werden die Werte für die normale Inhaltsbreite und die Breite für Blöcke mit weiter Breite eingestellt. Wenn Sie den Website-Editor und das Frontend aufrufen, sollten Sie bereits sehen, dass sich die Breite der Blöcke verändert hat.

Info: Weitere Layout-Einstellungsmöglichkeiten

Neben der `contentSize` und `wideSize` gibt es aktuell (WordPress 6.7) noch zwei weitere Optionen unter `settings.layout`:

- Mit `allowEditing` können Sie einstellen, ob die Layout-Optionen beispielsweise beim Gruppen-Block angezeigt werden sollen (Standardwert), oder nicht (mit dem Wert `false`).
- Mit `allowCustomContentAndWideSize` können Sie die standardmäßig sichtbaren Felder deaktivieren, die die Angabe eigener Werte für die normale und breite Inhaltsbreite ermöglichen.

- **Schriften einbinden**
  Seit WordPress 6 können Schriften komplett ohne PHP über die *theme.json* eingebunden werden. Das Theme verwendet die »IBM Plex Serif«, die unter anderem bei Google Fonts zu finden ist (*https://fonts.google.com/specimen/IBM+Plex+Serif*). Google bietet die Möglichkeit, die Dateien direkt vom Google-eigenen Server einzubinden, was aus Datenschutzgründen keine gute Idee ist. Ein Tool, das die lokale Einbindung von Schriften aus den Google Fonts stark erleichtert, ist der *google-webfonts-helper* (*https://gwfh.mranftl.com/fonts*). Hier können Sie eine Schrift auswählen, welchen Zeichensatz Sie benötigen (für Deutsch ist `latin` richtig) und welche Schriftschnitte Sie einsetzen möchten (siehe Abbildung 23.19).

  Nachdem Sie die Styles REGULAR, ITALIC und 700 für die IBM PLEX SERIF ausgewählt haben, scrollen Sie ein bisschen nach unten und finden dort ein Feld mit CSS-Code zum Einbinden der Schrift (siehe Abbildung 23.20). Wählen Sie hier die Variante MODERN BROWSERS aus, um nicht mehr alle möglichen Formate anzeigen zu lassen, sondern nur *.woff2*-Dateien – es sei denn, Sie müssen noch Internet Explorer unterstützen (das sollte hoffentlich bei keinem Projekt mehr der Fall sein). Sie können auch das Verzeichnis anpassen, aus dem die Schriften in dem Code geladen werden.

Das Verzeichnis kommt bei der Verwendung im Theme aber über eine WordPress-Funktion, daher kann hier der Standard erst mal einfach stehen bleiben.

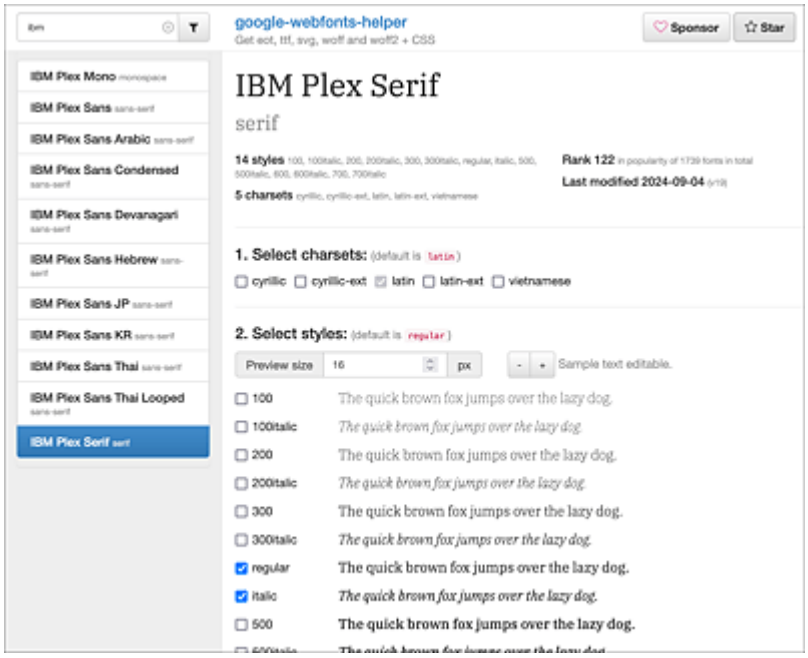


Abbildung 23.19 Mit dem »google-webfonts-helper« lassen sich Schriftdateien für das lokale Hosting auf dem Webserver der eigenen Website einfach herunterladen.

Nachdem Sie die Einstellungen vorgenommen haben, können Sie unten im Abschnitt DOWNLOAD FILES ein ZIP mit den Dateien herunterladen. Entpacken Sie es und kopieren Sie die Schriftdateien in den Ordner *assets/fonts* innerhalb des Theme-Ordners.

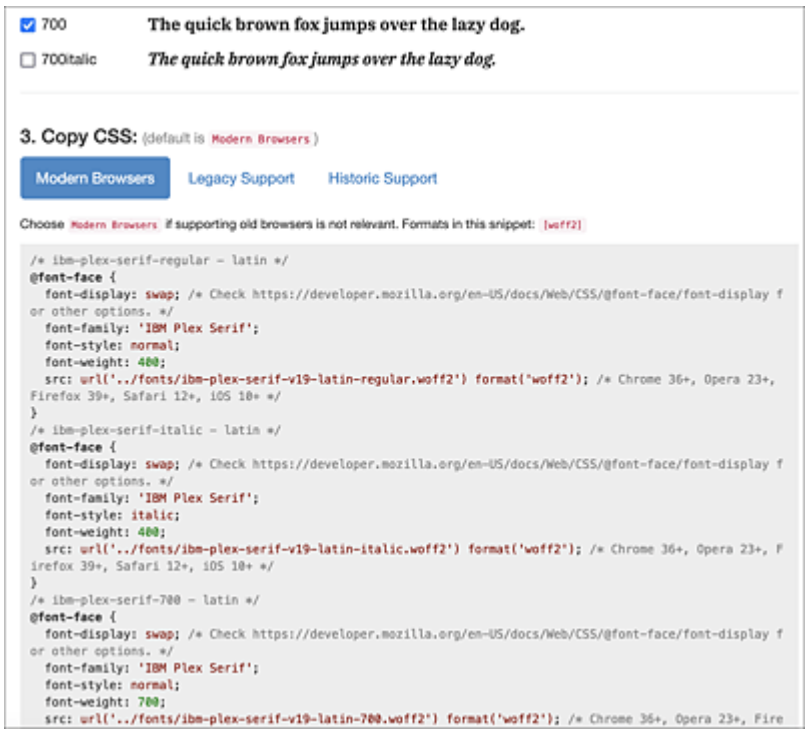


Abbildung 23.20 Die Option »Modern Browsers« sollte eigentlich für alle Websites ausreichen.

Die Einbindung der Schrift gelingt über das folgende Snippet in der *theme.json*:

```
{
    "$schema": "https://schemas.wp.org/wp/6.7/theme.json",
    "version": 3,
    "settings": {
        "layout": {
            "contentSize": "30rem",
            "wideSize": "40rem"
        },
        "typography": {
            "fontFamilies": [
                {
                    "fontFamily": "\"IBM Plex Serif\", serif",
                    "name": "IBM Plex Serif",
                    "slug": "ibm-plex-serif",
                    "fontFace": [
                        {
                            "fontFamily": "IBM Plex Serif",
                            "fontWeight": "400",
                            "fontStyle": "normal",
                            "fontStretch": "normal",
                            "src": [
                                "file:./assets/fonts/ibm-plex-serif-v15-latin-
```

```
regular.woff2"
                            ]
                        },
                        {
                            "fontFamily": "IBM Plex Serif",
                            "fontWeight": "700",
                            "fontStyle": "normal",
                            "fontStretch": "normal",
                            "src": [
                                "file:./assets/fonts/ibm-plex-serif-v15-latin-
700.woff2"
                            ]
                        },
                        {
                            "fontFamily": "IBM Plex Serif",
                            "fontWeight": "400",
                            "fontStyle": "italic",
                            "fontStretch": "normal",
                            "src": [
                                "file:./assets/fonts/ibm-plex-serif-v15-latin-
italic.woff2"
                            ]
                        }
                    ]
                }
            ]
        }
    },
    "styles": {
        "typography": {
            "fontFamily": "var:preset|font-family|ibm-plex-serif"
        }
    }
}
```

Neu sind `settings.typography` und `styles.typography`.

- Unter `settings.typography` können mit `fontFamilies` Schriften angegeben werden, die bei Block-Schriftart-Optionen zur Auswahl stehen. `fontFamily` gibt darin den String an, wie er für die `font-family`-CSS-Eigenschaft genutzt werden muss, um die Schrift anzuzeigen, `name` ist der im Editor sichtbare Name, und der `slug` wird für die Benennung einer CSS-Variablen genutzt, die den `font-family`-Wert enthält (diese Variable setzen wir gleich ein, um die Schrift als Standard im Editor zu setzen).

- Die eigentliche Einbindung der Schriftdateien passiert über `fontFace`. Aus den hier übergebenen Werten erzeugt WordPress eine CSS-`@font-face`-Regel, mit der die angegebene Schrift eingebunden wird. `fontFamily` ist der Name der Schrift, wie sie in CSS angesprochen werden kann (derselbe Wert wie der erste Wert bei dem `fontFamily`-Eintrag eine Ebene höher), `fontWeight` und `fontStyle` definieren den Schriftschnitt, `font-stretch` gibt an, ob eine `condensed`- (schmalere Buchstaben) oder `expanded`-Variante (breitere Buchstaben) der Schrift genutzt werden soll statt der normalen, und `src` beinhaltet eine Liste der Pfade zu den Dateien.
- Damit die Schrift nicht nur im Editor auswählbar, sondern auch die Standardschrift ist, wird diese in `styles.typography` als `fontFamily` angegeben. Dabei wird nicht der Variablennamen genutzt, den WordPress aus der Angabe generiert, sondern eine Pfad-Schreibweise (`var:preset|font-family|ibm-plex-serif`), über die WordPress dann den Variablennamen generiert.

Wenn Sie jetzt den Editor oder die Site im Frontend öffnen, sollten Sie sehen, dass die IBM Plex Serif als Schrift angezeigt wird (siehe Abbildung 23.21).



Abbildung 23.21 Nach Anpassen der Inhaltsbreite und Hinzufügen der Schrift sieht das Template im Editor schon besser aus.

- **Schriftgrößen und Zeilenhöhen definieren**
  Inzwischen lassen sich viele Styles (die Inhaltsbreite haben Sie bereits gesehen) direkt über die *theme.json* definieren, sodass sie nicht mehr separat in einem Stylesheet angegeben werden müssen. Im Folgenden werden die unterschiedlichen Schriftgrößen des Themes über die *theme.json* konfiguriert und direkt auf die entsprechenden Elemente angewendet, ebenso die Zeilenhöhen.

  WordPress generiert aus Angaben in der *theme.json* zu Aspekten wie Farben oder Schriftgrößen automatisch CSS-Variablen. Für die Zeilenhöhe gibt es zumindest aktuell diese Möglichkeit nicht, weshalb wir dafür den `settings.custom`-Schlüssel nutzen, über den Sie selbst eigene Werte angeben können, für die WordPress dann Variablen bereitstellt:

```
{
  [...],
  "settings": {
    "custom": {
      "lineHeight": {
        "default": "1.4",
        "h5": "1.2",
        "h4": "1.2",
        "h3": "1.2",
        "h2": "1.2",
        "h1": "1.1"
      }
    },
    [...]
```

  Aus diesen Angaben generiert WordPress CSS-Variablen nach dem folgenden Schema:

```
--wp--custom--line-height-default: 1.4
--wp--custom--line-height-h-5: 1.2
```

In der *theme.json* können Sie auf diese Werte über eine eigene Schreibweise zugreifen:

```
var:custom|line-heigh|default
var:custom|line-height|h5
```

Im nächsten Schritt erweitern wir das Objekt unter `settings.typography`:

```
{
  [...],
  "settings": {
    [...],
    "typography": {
      "defaultFontSizes": false,
      "customFontSize": false,
      "letterSpacing": false,
      "fluid": true,
      "fontSizes": [
        {
          "name": "Small",
          "slug": "small",
          "size": "0.889rem",
          "fluid": {
            "min": "0.889rem",
            "max": "1.11rem"
          }
        },
        {
          "name": "Default body size",
          "slug": "default",
          "size": "1em",
          "fluid": {
            "min": "1rem",
            "max": "1.25rem"
          }
        },
        {
          "name": "H5",
          "slug": "h5",
          "size": "1.266rem",
          "fluid": {
            "min": "1.266rem",
            "max": "1.58rem"
          }
        },
        {
          "name": "H4",
          "slug": "h4",
          "size": "1.424rem",
          "fluid": {
```

```
          "min": "1.424rem",
          "max": "1.78rem"
        }
      },
      {
        "name": "H3",
        "slug": "h3",
        "size": "1.602rem",
        "fluid": {
          "min": "1.602rem",
          "max": "2rem"
        }
      },
      {
        "name": "H2",
        "slug": "h2",
        "size": "1.802rem",
        "fluid": {
          "min": "1.802rem",
          "max": "2.252rem"
        }
      },
      {
        "name": "H1",
        "slug": "h1",
        "size": "2.027rem",
        "fluid": {
          "min": "2.027rem",
          "max": "2.534rem"
        }
      }
    ],
    [...]
```

Listing 23.5 Definition von Schrifteinstellungen in der »theme.json«

- Wir setzen `defaultFontSizes` auf `false` und deaktivieren dadurch die von WordPress bereitgestellte Liste von Standard-Schriftgrößen.
- Über `"customFontSize": false` deaktivieren Sie die Möglichkeit, eigene Schriftgrößen anzugeben.
- Auch der Zeichenabstand soll nicht individuell geändert werden können, weshalb wir diese Option mit `"letterSpacing": false` entfernen.

- `"fluid": true` aktiviert die sogenannte »Fluid Typography«, einen Mechanismus, bei dem die Schriftgröße nicht fix angegeben wird, sondern ein Bereich aus Mindest- und Maximalgröße. Die Anpassung der Schriftgröße geschieht dann automatisch stufenlos.
- Unter `fontSizes` geben wir jetzt die Schriftgrößen an, die im Editor auswählbar sind. In dem `fontSizes`-Array wird jede Schriftgröße innerhalb eines Objekts definiert, das die Keys `name`, `slug` und `size` benötigt, die selbsterklärend sein sollten. Optional ist die Angabe von fluid, die entweder auf `false` gesetzt werden kann, um die fluide Typografie für eine bestimmte Schriftgröße zu deaktivieren. Der andere mögliche Wert ist ein Objekt mit `min`- und `max`-Angabe, mit der Sie die Mindest- und Maximalschriftgröße setzen können. Wenn Sie den `fluid`-Key bei einer Schriftgröße nicht setzen, aber `settings.typography.fluid` auf `true` gesetzt ist, berechnet WordPress automatisch einen Größenbereich auf Basis von `size`.

Als Nächstes kümmern wir uns darum, die Schrift-Styles direkt auf die entsprechenden Elemente anzuwenden:

```
{
  [...]
  "styles": {
    "typography": {
      [...]
      "fontSize": "var:preset|font-size|default",
      "lineHeight": "var:custom|line-height|default"
    },
    [...]
    "elements": {
      [...]
      "heading": {
        "typography": {
          "fontWeight": "700"
        }
      },
      "h1": {
```

```
        "typography": {
          "fontSize": "var:preset|font-size|h-1",
          "lineHeight": "var:|custom|line-height|h-1"
        }
      },
      "h2": {
        "typography": {
          "fontSize": "var:preset|font-size|h-2",
          "lineHeight": "var:custom|line-height|h-2"
        }
      },
      "h3": {
        "typography": {
          "fontSize": "var:preset|font-size|h-3",
          "lineHeight": "var:custom|line-height|h-3"
        }
      },
      "h4": {
        "typography": {
          "fontSize": "var:preset|font-size|h-4",
          "lineHeight": "var:custom|line-height|h-4"
        }
      },
      "h5": {
        "typography": {
          "fontSize": "var:preset|font-size|h-5",
          "lineHeight": "var:custom|line-height|h-5"
        }
      }
    }
  }
}
```

Listing 23.6 Anwendung der Schrifteinstellungen in der »theme.json«

- Unter `styles.typography` haben Sie bereits in `fontFamily` die Schrift für das `body`-Element definiert. Jetzt kommt noch die Standard-Schriftgröße mit `fontSize` hinzu, für die Sie als Wert die Referenz auf die vorher neu definierte `default`-Schriftgröße angeben: `vars|font-size|default`. Darüber hinaus legen Sie mit `lineHeight` die Standard-Zeilenhöhe fest, für die mit `var:custom|line-height| default` eine der anfangs festgelegten eigenen Variablen aus `settings.custom` genutzt wird.
- Jetzt geht es an die Überschriften. Der Pfad `styles.elements.heading` ermöglicht Ihnen, Werte festzulegen,

die für alle Überschriften-Elemente gelten, unabhängig von der Ebene. Diese Möglichkeit nutzen wir zum Festlegen des `font-weight`-Werts und gehen dafür über den Pfad `styles.elements.heading.typography. fontWeight`.

- Mit `styles.elements.h[1-6]` können Angaben gemacht werden, die gezielt nur für eine der sechs Überschriften-Ebenen gelten. Für die Level `h1` bis `h5` legen wir jeweils mit `typography.fontSize` und `typography.lineHeight` die Schriftgröße und Zeilenhöhe fest.

Info: Zusätzliche Möglichkeiten unter settings.typography

Neben den gezeigten Möglichkeiten gibt es noch weitere Einstellungen unter `settings.typography`. Aktuell (WordPress 6.6) sind das die Folgenden.

- `fontStyle` gibt an, ob im Editor beispielsweise bei Absätzen die Möglichkeit gegeben sein soll, bei der Schriftschnitt-Auswahl kursive Schnitte zu wählen.
- `fontWeight` ermöglicht es einzustellen, ob bei den Schriftoptionen das Schriftgewicht geändert werden können soll, also beispielsweise dünn oder medium.
- `lineHeight` ist standardmäßig auf `false` gestellt und erlaubt Ihnen, den Benutzer*innen zu ermöglichen, eine eigene Zeilenhöhe festzulegen.
- `textAlign` erlaubt es, die Textausrichtung zu ändern. Ohne Angabe ist diese Option aktiv.
- `textColumns` ermöglicht es, die Option für mehrere Textspalten bei Blöcken zu aktivieren, die diese Funktion unterstützen. Soweit ich sehe, hat aktuell (WordPress 6.6) kein Core-Block

die Unterstützung für diese Funktion. Der Standardwert ist `false`.

- `textDecoration` können Sie nutzen, um zu verhindern, dass `text-decoration`-Werte für Texte angepasst werden, um Text beispielsweise durchzustreichen.
- `writingMode` ist normalerweise deaktiviert und kann aktiviert werden, sodass im Editor eingestellt werden kann, ob Text horizontal oder vertikal ausgerichtet sein soll.
- `textTransform` ist standardmäßig aktiv und ermöglicht den Anwender*innen, Text beispielsweise komplett in Großbuchstaben anzuzeigen.
- `dropCap` ist automatisch aktiv und steuert, ob bei Blöcken wie dem Absatzblock die Option für einen Initialbuchstaben angezeigt wird.

- **Farbeinstellungen anpassen**

  Wenn ein Theme keine Farbpalette definiert, wird eine Standardfarbpalette von WordPress angezeigt, und Benutzer*innen mit ausreichend Berechtigung können diese im Website-Editor in der Stile-Seitenleiste anpassen oder Farben zu einer individuellen Farbpalette hinzufügen. Über die *theme.json* kann eine Theme-Farbpalette erstellt und die Standard-Farbpalette kann deaktiviert werden (siehe Listing 23.7 – ab jetzt werden in den Listings nur noch die neuen Codeteile mit ein bisschen Referenzcode der bestehenden Datei davor und dahinter gezeigt, um es übersichtlich zu halten).

```
{
[...]
  "settings": {
    [...]
    "color": {
```

```
      "background": false,
      "link": true,
      "defaultPalette": false,
      "custom": false,
      "defaultGradients": false,
      "customGradient": false,
      "gradients": [],
      "palette": [
        {
          "color": "#333",
          "name": "Dark",
          "slug": "dark"
        },
        {
          "color": "#fff",
          "name": "White",
          "slug": "light"
        }
      ]
    }
  },
  [...]
```

Listing 23.7 Mit »settings.color« lassen sich unterschiedliche Einstellungen für die Farboptionen im Editor vornehmen.

- `settings.color.background` legt fest, ob die Option zum Auswählen einer Hintergrundfarbe für Blöcke angezeigt werden soll, die diese Option unterstützen (zum Beispiel der Überschriften- oder Gruppen-Block). Mit `false` wird diese Option deaktiviert (im nächsten Listing werden wir sie für ausgewählte Blöcke aktivieren).
- `settings.color.link` aktiviert die Farboption für Links innerhalb von Block-Inhalten.
- `settings.color.defaultPalette` bestimmt, ob die Standardpalette von WordPress angezeigt werden soll (das soll nicht passieren, daher wird als Wert `false` übergeben).
- `settings.color.custom` aktiviert oder deaktiviert die Möglichkeit, dass Benutzer*innen im Inhaltseditor auch eine individuelle Farbe festlegen können.

- `Settings.color.defaultGradients` steuert, ob die Standard-Farbverläufe von WordPress angezeigt werden.
- `settings.color.customGradient` macht dasselbe für die Funktion, eigene Farbverläufe festzulegen.
- `settings.color.gradients` enthält die Farbverläufe, die im Editor ausgewählt werden können. Sollen keine Verläufe angezeigt werden, wird ein leeres Array angegeben.
- `settings.color.palette` definiert die Theme-Farbpalette. Jedes Farbobjekt innerhalb des Arrays benötigt die Einträge `color` (Farbwert in einem beliebigen für Browser verständlichen Format), `name` (der für die Farbe angezeigte Name) und `slug` (wird für den Klassennamen genutzt, der Elementen mit der Farbe mitgegeben wird, und für den CSS-Variablennamen). Es werden zwei Farben definiert: ein dunkles Grau und Weiß.

Wenn Sie nun einen beliebigen Block auswählen, der Farboptionen anbietet, sollten Sie die neue Farbpalette sehen und nicht mehr die Standardfarben (siehe Abbildung 23.22).

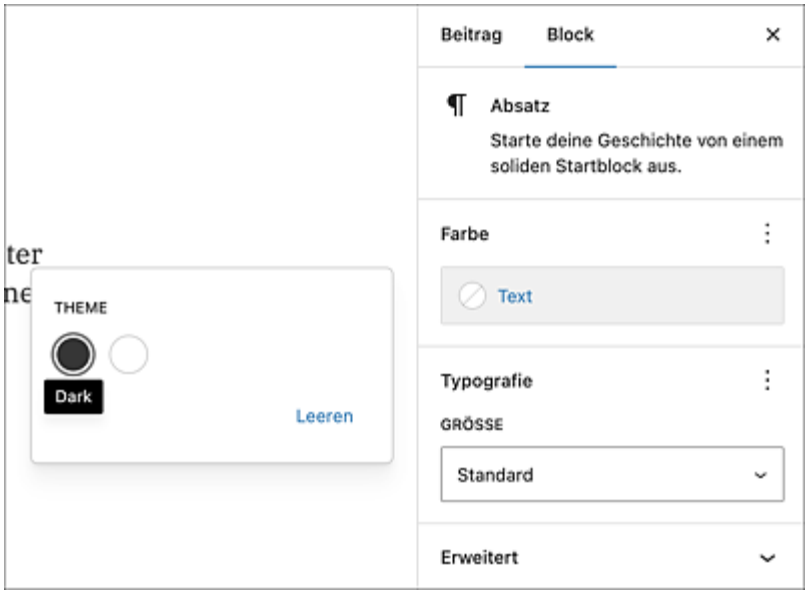


Abbildung 23.22 Die Theme-Farbpalette wird bei allen Blöcken mit Farboptionen angezeigt.

Nachdem jetzt generell für alle Blöcke die Option für eine Hintergrundfarbe deaktiviert ist, soll sie für Gruppen-Blöcke aber doch nutzbar sein.

```
{
  [...]
  "settings": {
    [...]
    "blocks": {
      "core/group": {
        "color": {
          "background": true
        }
      }
    }
    [...]
```

Listing 23.8 Innerhalb von »settings.blocks« können die ganzen Einstellungen explizit für einzelne Blöcke vorgenommen werden.

Sowohl bei `settings` als auch bei `styles` gibt es den Unterpunkt `blocks`, in dem alle Optionen noch mal auf Block-Ebene angepasst werden können. Dafür folgen in der Ebene unter `blocks` der Bezeichner des Blocks und dann die Optionen nach demselben Schema wie auf der Ebene unter `settings` bzw. `styles`.

Folgerichtig wird in `settings.blocks.core/group.color.background` die Hintergrundfarbenoption für diesen einen Block aktiviert.

Als letzte Farbeinstellung passen Sie die Farbe des Textes und der Links an. Der Text soll automatisch die dunkle Farbe der Farbpalette erhalten, und Links sollen dieselbe Farbe haben wie das Elternelement, in dem sie eingefügt sind. Dafür ist wieder eine Anpassung im `styles`-Abschnitt notwendig.

```
{
  [...]
  "styles": {
    [...]
    "color": {
      "text": "var:preset|color|dark"
```

```
    },
    "elements": {
      "link": {
        "color": {
          "text": "currentColor"
        }
      }
    }
  }
}
```

Listing 23.9 Im »styles«-Abschnitt legen Sie Standardfarben fest.

Mit `styles.color.text` legen Sie den Farbwert für das `body`-Element fest, das dann auf die anderen Elemente vererbt wird. Als Wert wird die CSS-Variable genutzt, die WordPress für die Farbe generiert hat, weil sie in der Farbpalette eingefügt wurde. Unter `styles.elements` lassen sich Standard-Styles für die unterschiedlichen Überschriftenelemente (`h1` bis `h6`) sowie Links angeben. Durch den Wert `currentColor` für `styles.elements.link.color.text` wird das CSS `a { color: currentColor; }` generiert.

Info: Weitere Möglichkeiten zur Farbeinstellung

- `customDuotone` erlaubt Nutzer*innen, eigene Duotone-Filter für Blöcke wie den Bildblock zu erstellen – Standardwert ist `true`.
- `defaultDuotone` ist ebenfalls standardmäßig aktiv und gibt an, ob die Standardpalette an Duotone-Filtern angezeigt werden soll.
- `duotone` erlaubt es Ihnen, eigene Duotone-Filter zu definieren, die zu Ihrem Theme passen. Als Wert wird ein Array von Objekten angegeben, die neben `name` und `slug` ein `colors`-Array enthalten, das aus HEX- oder RGB-Farbwerten von dunkel nach hell besteht.

- `text` erlaubt die Änderung der Textfarbe und ist standardmäßig aktiv.
- `heading` gibt an, ob die Farbe für Überschriften angepasst werden kann. Wenn Sie den Wert nicht explizit setzen, ist das möglich.
- `button` ist ebenfalls by default aktiv und erlaubt das Setzen einer Farbe von Buttons.
- `caption` ermöglicht die Anpassung der Farbe für Beschriftungen – auch hier ist der Standardwert `true`.

- **Weitere Möglichkeiten innerhalb von settings**
  Sie haben jetzt bereits einige Einstellungsmöglichkeiten kennengelernt, aber damit ist noch lange nicht das Ende der Optionen erreicht, die Sie unter `settings` angeben können. Folgende Keys sind aktuell (WordPress 6.7) noch möglich:
  - `appearanceTools` erlaubt das Aktivieren von gleichzeitig mehreren UI-Werkzeugen im Editor, unter anderem Hintergrundbild und -größe und Einstellungsmöglichkeiten für Rahmen.
  - `background` ist ein Objekt mit den beiden möglichen Keys `backgroundImage` und `backgroundSize`. Damit können Sie bei Blöcken, die die Funktionen unterstützen, Optionen für ein Hintergrundbild und für die Größe des Bilds aktivieren.
  - `border` ist ebenfalls ein Objekt und kann die Schlüssel `color`, `radius`, `style` und `width` enthalten, die die entsprechenden Einstellungsmöglichkeiten für Rahmen aktivieren.
  - `dimensions` dreht sich um Optionen, bei denen Sie das Seitenverhältnis von Elementen sowie die Mindesthöhe setzen können. Dafür können Sie im Objekt mit `aspectRatio`

aktivieren, dass die Optionen für das Seitenverhältnis angezeigt werden. Mit `dimensions.defaultAspectRatios` können Sie deaktivieren, dass die Standard-Seitenverhältnisse ausgewählt werden können. `dimensions.aspectRatios` erlaubt die Angabe eigener Seitenverhältnisse, als Wert müssen Sie ein Array aus Objekten übergeben, die aus `name`, `slug` und `ratio` bestehen. Über `dimensions.minHeight` können Sie den Nutzer*innen die Möglichkeit geben, für bestimmte Blöcke eine Mindesthöhe zu setzen.

- `lightbox` ermöglicht die Konfiguration der Lightbox-Funktion für Bilder. Mit `lightbox.enabled` geben Sie an, ob die Lightbox aktiv ist oder nicht, und mit `lightbox.allowEditing` können Sie den Nutzer*innen ermöglichen, die Option über den Editor anzupassen.
- `position` erlaubt es über `position.sticky`, die Sticky-Positionierungs-Option zu aktivieren.
- `shadow` können Sie nutzen, um bei unterstützenden Blöcken die Optionen für Schatten zu aktivieren. `shadow.defaultPresets` ist ein boolscher Wert, über den Sie die Standard-Schattenoptionen von WordPress aktivieren oder deaktivieren können. Mit `shadow.presets` können Sie ein Array von Objekten angeben, die die Eigenschaften `name`, `slug` und `shadow` enthalten, wobei `shadow` einen Wert für die CSS-Eigenschaft `box-shadow` hat.
- `spacing` beinhaltet unterschiedliche Einstellungen für Abstände.
  Mit `spacing.blockGap` können Sie einstellen, ob beispielsweise beim Spalten-Block die Einstellungsmöglichkeit für den Spaltenabstand angezeigt wird (`true`) oder nicht (`false`). Wenn Sie `false` als Wert angeben, können Sie in der *theme.json* im

`styles`-Bereich trotzdem noch Werte für die Abstände angeben. Der dritte mögliche Wert ist null (das ist auch der Standardwert), der nicht nur die Option für die Nutzer*innen deaktiviert, sondern auch keine CSS-Regeln für Abstände generiert.
`spacing.margin` und `spacing.padding` ermöglicht Ihnen zu steuern, ob im Editor die Außen- und Innenabstände bei Blöcken geändert werden können, wenn sie die Einstellung unterstützen.
`spacing.units` ist ein Array von CSS-Einheiten, die im Editor ausgewählt werden können, und `spacing.customSpacingSize` gibt an, ob ein Feld für eigene Größenangaben angezeigt werden soll.
Über `spacing.defaultSpacingSizes` legen Sie fest, ob die Standard-Abstände von WordPress auswählbar sind, und eigene Werte können Sie mit `spacing.spacingSizes` definieren, einem Array von Objekten, die die Eigenschaften `name`, `slug` und `size` erwarten – `size` muss eine CSS-Größenangabe inklusive Einheit sein.
Mit `spacing.spacingScale` können Sie eine Skalierung angeben, nach der sich der Abstand der vorauswählbaren Optionen etwa für die Größe des Abstand-Blocks richtet. Dabei wird als Wert ein Objekt mit den folgenden Eigenschaften erwartet: `operator` (`+` oder `*`), `increment` (der Wert, um den je Schritt erhöht wird), `steps` für die Anzahl der Schritte, `mediumStep` für den mittleren Schritt und `unit` für die zu verwendende Einheit.

- **Standardwerte innerhalb von styles definieren**

  Im Abschnitt oben haben Sie bereits ein paar Beispiele kennengelernt, wie Sie Standardwerte für Schrifteigenschaften setzen können. Genauso können Sie für die anderen Einstellungsmöglichkeiten vorgehen. So können Sie

beispielsweise wie folgt die Zwischenabstände etwa des Spalten-Blocks definieren, wenn Sie für `settings. spacing.blockGap` die Werte `true` oder `false` und nicht `null` angegeben haben:

```
{
  "version": 3,
  "styles": {
    "spacing": {
      "blockGap": "1.5rem"
    }
  }
}
```

- **Weitere Möglichkeiten mit der theme.json**
  Neben `settings` und `styles` gibt es noch drei andere Bereiche, die in der *theme.json* genutzt werden können:
  - `customTemplates` ermöglicht das Hinterlegen von Informationen für Custom Templates, die in der Einzelansicht von Seiten und Beiträgen im Bereich »Template« der Block-Editor-Seitenleiste ausgewählt werden können. Ohne diese Angabe wird das Template mit dem Slug des Dateinamens aufgeführt (*custom.html* wird zum `custom`-Template) und für alle Inhaltstypen angeboten. In Listing 23.10 sehen Sie, wie so ein Eintrag in der *theme.json* für das Template *custom.html* aussehen kann.

```
{
  "version": 3,
  "customTemplates": [
     {
       "name": "custom",
       "title": "Ein Template",
       "postTypes": [
         "post",
         "page"
       ]
     }
   ]
}
```

Listing 23.10 Unter »customTemplates« können Informationen über die Templates des Themes hinterlegt werden, die keine Standard-Templates sind.

- Das `customTemplates`-Array kann mehrere Objekte beinhalten, von denen jedes ein Template beschreibt. Die Eigenschaften `name` und `title` sind zwingend notwendig, mit der optionalen Angabe von `postTypes` kann eingestellt werden, für welche Inhaltstypen das Template auswählbar ist (im Beispiel für Seiten und Beiträge).
- `templateParts` funktioniert ähnlich wie `customTemplates`, nur für die Template-Teile. Sie geben pro Template-Teil `name` (muss dem Dateinamen ohne Endung entsprechen) und `title` sowie optional den Bereich des Template-Teils an (`area`). In Listing 23.11 ist das für die Template-Teile des Themes umgesetzt.

```
    "templateParts": [
    {
      "name": "header-h1",
      "title": "Header (h1)",
      "area": "header"
    },
    {
      "name": "header-p",
      "title": "Header (p)",
      "area": "header"
    },
    {
      "name": "footer",
      "title": "Footer",
      "area": "footer"
    }
  ]
}
```

Listing 23.11 In »templateParts« können Namen und Bereiche für die Template-Teile des Themes festgelegt werden.

- `patterns` erlaubt Ihnen, Block-Vorlagen (dazu später mehr) aus dem WordPress.org-Vorlagenverzeichnis direkt mit einzubinden (*https://de.wordpress. org/patterns/*). Dafür geben Sie den Slug einer oder mehrere Vorlagen an, die Sie verfügbar machen möchten – der Slug ist der letzte Teil der URL, bei

https://de.wordpress.org/patterns/pattern/callout-over-a-patterned-background/ zum Beispiel `calloutover-a-patterned-background`. Um diese Vorlage direkt auswählbar zu machen, gehen Sie vor wie im folgenden Listing gezeigt.

```
{
  "version": 3,
  "patterns": [ "callout-over-a-patterned-background" ]
}
```

Stylesheet einbinden

Nachdem über die *theme.json* bereits ein paar grundlegende Styles eingebunden wurden, sind aber auch herkömmliche CSS-Regeln in einer CSS-Datei notwendig, um das Aussehen der Website weiter zu verfeinern. Die konkreten CSS-Regeln werden hier nicht erläutert (Sie können das Stylesheet hier herunterladen: *http://r-wrk.de/wp6_block_stylesheet*), es geht nur um die Einbindung des Stylesheets. Legen Sie dazu eine *functions.php* im Theme-Ordner an, und schreiben Sie den Code aus Listing 23.12 hinein.

```
<?php
if ( ! function_exists( 'wh_block_enqueue_assets' ) ) {
  function wh_block_enqueue_assets() {
    wp_enqueue_style(
      'wh-block-style',
      get_stylesheet_uri(),
      [],
      filemtime( get_stylesheet_directory() . '/style.css' )
    );
  }
}
add_action( 'wp_enqueue_scripts', 'wh_block_enqueue_assets' );
```

Listing 23.12 Die Einbindung eines Stylesheets über die functions.php funktioniert beim Block-Theme noch genauso wie bei einem Classic Theme.

Mit `wp_enqueue_style()` wird ein Stylesheet eingebunden, indem als erster Parameter das Handle angegeben wird, als zweiter die URL des Stylesheets, gefolgt von Abhängigkeiten (hier gibt es keine) und

einer Version, die als Query-Parameter an die Stylesheet-URL angehängt wird, um die Datei nach einer Änderung verlässlich bei allen Besucher*innen neu zu laden, statt eine veraltete Version aus dem Cache zu nehmen.

Alternative Stilkombinationen erstellen

Im *styles*-Ordner eines Block-Themes können Sie weitere JSON-Dateien mitliefern, die nach dem Muster der *theme.json* `styles` und `settings` definieren und damit die Werte aus der *theme.json* überschreiben. In der Stile-Seitenleiste des Website-Editors können diese anderen Stilkombinationen (»Styles variations«) dann aktiviert werden. Wenn beispielsweise eine solche *styles/blue.json* existiert, wie in Listing 23.13 abgedruckt, wird das in der Seitenleiste dargestellt, wie in Abbildung 23.23 zu sehen.

```
{
  "version": 3,
  "title": "Blue variation",
  "settings": {
    "color": {
      "palette": [
        {
          "color": "#002547",
          "name": "Dark",
          "slug": "dark"
        },
        {
          "color": "#f2f9ff",
          "name": "Light",
          "slug": "light"
        }
      ]
    }
  },
  "styles": {
    "color": {
      "background": "var:preset|color|light",
      "text": "var:preset|color|dark"
    }
  }
}
```

Listing 23.13 Definition einer alternativen Stilkombination, die im Website-Editor ausgewählt werden kann

Die Werte aus der JSON-Datei werden mit denen aus der *theme.json* zusammengeführt, und es können neben der optionalen Angabe eines `title` nur Änderungen in `settings` und `styles` vorgenommen werden. Array-Werte (etwa die Farbpalette) können nicht erweitert werden, sondern werden komplett überschrieben. Der Name der Datei fungiert als Name der Stilkombination im Backend.

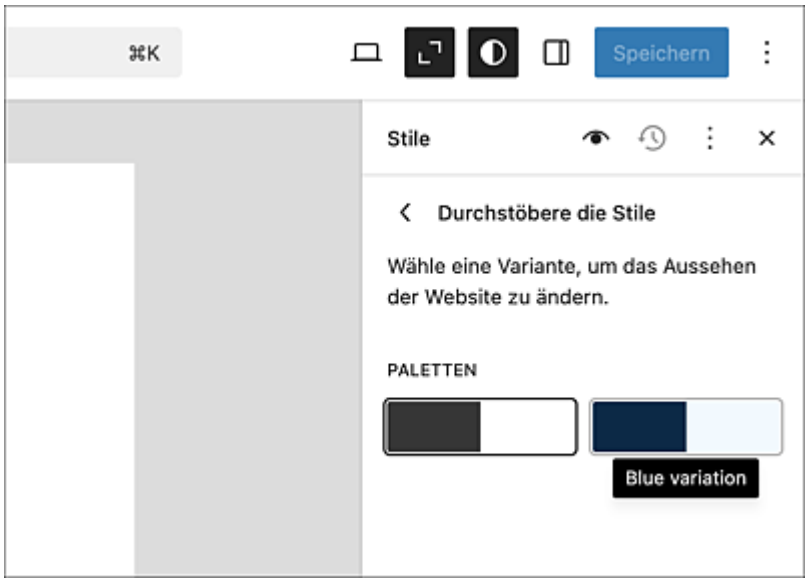


Abbildung 23.23 In der »Stile«-Seitenleiste können vorhandene andere Stilkombinationen über den Punkt »Durchstöbere die Stile« ausgewählt werden.

Vorlagen erstellen

Eine Vorlage (oder ein Pattern) ist das Markup eines oder mehrerer Blöcke, das so registriert wird, dass es im Block-Inserter ausgewählt und eingefügt werden kann, ohne dass die Block-Konstruktion manuell nachgebaut werden muss. Das ist sehr praktisch für aufwendige und/oder häufig verwendete Inhaltselemente, und es gibt unterschiedliche Wege, eine Vorlage zu erstellen.

1. In jedem Kontext (egal ob Block-Theme, Classic Theme oder Plugin) mit der Funktion `register_block_pattern()`. Der Code sieht dabei beispielsweise wie der in Listing 23.14 aus, um eine Vorlage »Color inverted group« zu erstellen.

```
register_block_pattern(
  'wh-block/color-inverted',
  [
    'title' => __( 'Color inverted group', 'wh-block' ),
    'content' => '<!-- wp:group {"backgroundColor":"dark",
"textColor":"white"} -->
    <div class="wp-block-group has-white-color has-dark-background-color has-
text-color has-background"><!-- wp:paragraph -->
    <p>Text</p>
    <!-- /wp:paragraph --></div>
    <!-- /wp:group -->',
  ]
);
```

Listing 23.14 Registrierung einer Block-Vorlage mit der »register_block_pattern()«-Funktion

   Als ersten Parameter übergeben Sie an die Funktion den Bezeichner der Vorlage, der eindeutig sein muss. Als zweiter Parameter wird ein Array übergeben, das mindestens den Eintrag `title` (der für Benutzer*innen sichtbare Titel im Block-Inserter) und `content` enthalten muss. In `content` ist das Markup des oder der Blöcke gespeichert, die beim Nutzen der Vorlage eingefügt werden sollen. Am einfachsten kommen Sie an das Markup, wenn Sie die Blöcke wie gewünscht im Block-Editor zusammenbauen und konfigurieren und danach das Markup aus dem Editor kopieren.

2. In Block-Themes können Vorlagen seit WordPress 6 mit PHP-Dateien im *patterns*-Verzeichnis registriert werden (WordPress sucht automatisch nach dem Verzeichnis und enthaltenen Dateien, Sie müssen die Dateien nicht in der *functions.php* einbinden). Dabei wird nicht `register_block_pattern()` genutzt, sondern ein Datei-Header-Kommentar mit den

Metainformationen zur Vorlage und danach das Markup. Das ist in Listing 23.15 beispielhaft für dieselbe Vorlage wie in Listing 23.14 dargestellt.

```
<?php
/**
 * Title: Color inverted group
 * Slug: wh-block/color-inverted
 */
?>
<!-- wp:group {"backgroundColor":"dark","textColor":"white"} -->
<div class="wp-block-group has-white-color has-dark-background-color has-text-
color has-background"><!-- wp:paragraph -->
<p>Text</p>
<!-- /wp:paragraph --></div>
<!-- /wp:group -->
```

Listing 23.15 In Block-Themes lassen sich Vorlagen ab WordPress 6 mit Dateien im »patterns«-Verzeichnis registrieren.

Im Dateikommentar müssen Sie mindestens den sichtbaren Namen der Vorlage (`Title`) sowie den `Slug` angeben. Wenn ein Child-Theme denselben Slug angibt, kann es damit die Block-Vorlage überschreiben. Nach dem Kommentar folgt dann direkt das Markup für die Block-Vorlage.

Es gibt noch ein paar weitere Konfigurationsmöglichkeiten für Vorlagen, so kann zum Beispiel die Breite des Viewports der Vorlagenvorschau festgelegt werden, die beim Durchsuchen der Vorlagen im Inserter zu sehen ist. Alle verfügbaren Optionen gibt es auf der Seite zur Registrierung von Vorlagen im Themes-Handbuch (*https://developer.wordpress.org/themes/patterns/registering-patterns*). Einfügen können Sie die Vorlagen über den Block-Inserter. Da sind sie über die Suche und im Bereich VORLAGEN zu finden (siehe Abbildung 23.24).

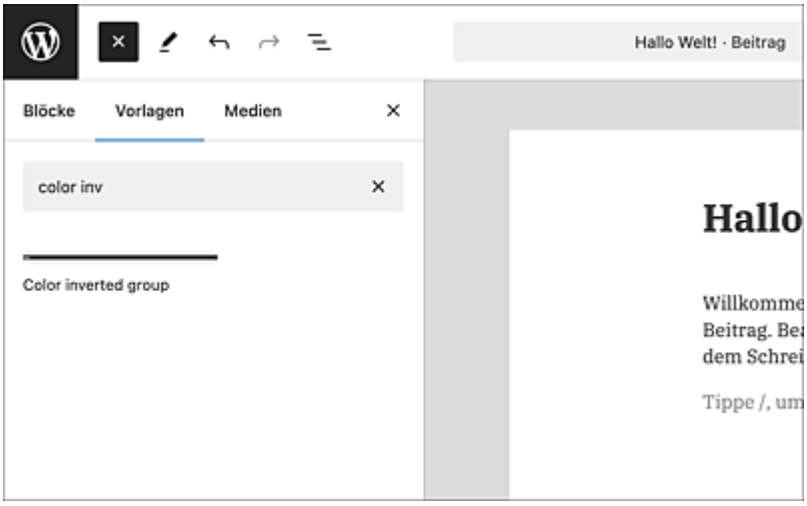


Abbildung 23.24 Im Block-Inserter können die vorhandenen Block-Vorlagen durchgeschaut und eingefügt werden.

Seitenvorlagen

Seitenvorlagen sind Vorlagen, die Ihnen beim Anlegen einer neuen Seite oder eines anderen Inhalts angeboten werden, sodass Sie nicht bei null starten müssen.

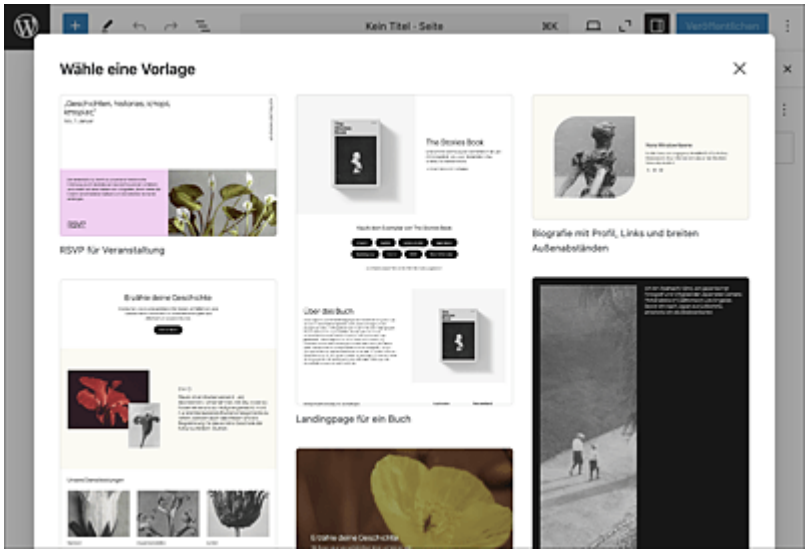


Abbildung 23.25 Seitenvorlagen in dem Theme Twenty Twenty-Five

Eine Block-Vorlage zu einer Seitenvorlage zu machen, ist einfach: Sie müssen lediglich den Block-Types-Parameter auf `core/post-content`

setzen und können über Post Types die Inhaltstypen angeben.

Mit `register_block_pattern()` sieht das wie folgt aus:

```
register_block_pattern(
  'wh-block/color-inverted',
  [
    'title' => __( 'Color inverted group', 'wh-block' ),
    'blockTypes' => ['core/post-content'],
    'postTypes' => ['page'],
    'content' => '<!-- wp:group {"backgroundColor":"dark","textColor":"white"} -->
    <div class="wp-block-group has-white-color has-dark-background- color has-text-
color has-background"><!-- wp:paragraph -->
    <p>Text</p>
    <!-- /wp:paragraph --></div>
    <!-- /wp:group -->',
  ]
);
```

Listing 23.16 Registrierung einer Vorlage mit der »register_block_pattern()«-Funktion als Seiten-Vorlage

Wenn Sie keine Inhaltstypen mit `postTypes` angeben, wird es standardmäßig für alle Inhaltstypen aktiviert.

In Block-Themes sieht das in einer Datei im *patterns*-Verzeichnis wie folgt aus:

```
<?php
/**
 * Title: Color inverted group
 * Slug: wh-block/color-inverted
 * Block Types: core/post-content
 * Post Types: page
 */
?>
<!-- wp:group {"backgroundColor":"dark","textColor":"white"} -->
<div class="wp-block-group has-white-color has-dark-background-color has-text-color
has-background"><!-- wp:paragraph -->
<p>Text</p>
<!-- /wp:paragraph --></div>
<!-- /wp:group -->
```

Listing 23.17 Registrierung einer Seiten-Vorlage in einem Block-Theme

Template-Vorlagen

Neben den Seiten-Vorlagen gibt es Template-Vorlagen. Diese Vorlagen werden angezeigt, wenn Sie im Website-Editor ein neues Template anlegen.

Um eine Vorlage zu einer Template-Vorlage zu machen, gibt es die Angaben `Template Types`, über die Sie angeben können, für welche Template-Arten die Vorlage angezeigt werden soll. Zudem können Sie über `Inserter: no` einstellen, dass die Vorlage nicht im Block-Inserter auftaucht.

Im folgenden Listing sehen Sie ein Beispiel des veränderten Headers.

```
<?php
/**
 * Title: Front page template
 * Slug: wh-block/front-page-template
 * Template Types: front-page
 * Inserter: no
 */
?>
<!-- Block markup -->
```

Listing 23.18 Über eine zusätzliche Angabe im Datei-Header kann eine Vorlage zu einer Template-Vorlage gemacht werden.

Block-Typ-Vorlagen

Sie haben eben bei den Seitenvorlagen schon die Meta-Angabe `Block Types` kennengelernt. Die Angabe ist aber für einen weiteren Anwendungsfall gut: Sie können darüber angeben, ob eine Vorlage für einen bestimmten Block gedacht ist.

So können Sie beispielsweise eine Vorlage für den Cover-Block mit `Block Types: core/cover` so kennzeichnen, dass die Vorlage bei einem unveränderten Cover-Block im Editor ausgewählt werden kann:

```php
<?php
/**
 * Title: Cover mit Überschrift und Text
 * Slug: wh-block/cover
 * Block Types: core/cover
 * Post Types: page
 */
?>
<!-- wp:cover {"overlayColor":"contrast","isUserOverlayColor":true,"metadata":
{"categories":["banner"],"patternName":"themeslug/cover-contrast","name":"Cover With
Contrast Background"},"align":"full","style":{"elements":{"link":{"color":
{"text":"var:preset|color|light"}}}},"textColor":"light","layout":{"type":
"constrained"}} -->
<div class="wp-block-cover alignfull has-light-color has-text-color has-link-color">
<span aria-hidden="true" class="wp-block-cover__background has-contrast-background-
color has-background-dim-100 has-background-dim"></span><div class="wp-block-
cover__inner-container"><!-- wp:heading {"textAlign":"left"} -->
<h2 class="wp-block-heading has-text-align-left">Eine Überschrift</h2>
<!-- /wp:heading -->

<!-- wp:paragraph -->
<p>Ein kurzer Absatz.</p>
<!-- /wp:paragraph --></div></div>
<!-- /wp:cover -->
```

Listing 23.19 Sie können Vorlagen für bestimmte Blöcke erstellen, sodass diese Blöcke nach dem Einfügen in den Editor in die Vorlage umgewandelt werden können.

Wenn Sie diese Vorlage als *patterns/cover.php* speichern, können Sie nach dem Einfügen eines Cover-Blocks in einer Seite diesen Block in die Vorlage umwandeln, indem Sie oben links auf das Block-Icon klicken und dann VORLAGEN auswählen, wie in Abbildung 23.26 zu sehen.

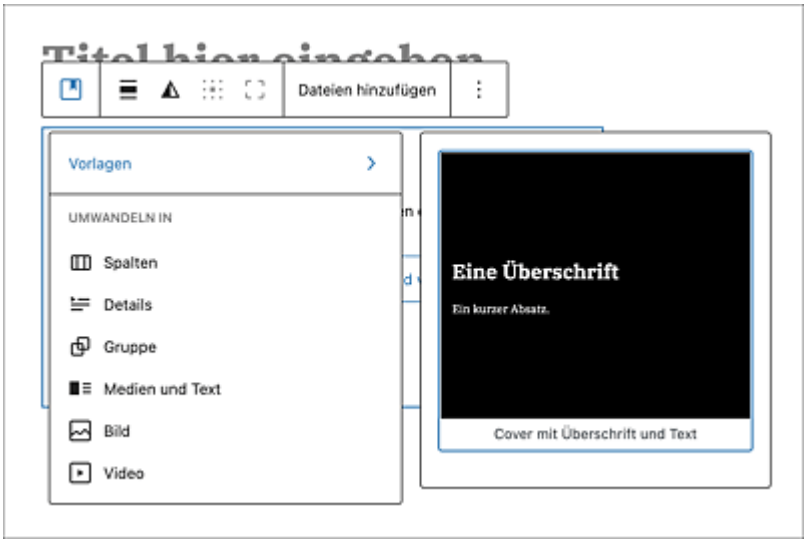


Abbildung 23.26 Block-Vorlagen können als Variante für bestimmte Blöcke markiert werden.

Übersetzungen

Das Thema Übersetzung ist bei Block-Themes ein bisschen schwierig. Die Theme-Metadaten können ganz normal durch das Erstellen einer Übersetzungsdatei und das Laden der Übersetzung in der *functions.php* übersetzt werden. Um die Übersetzungen zu laden, fügen Sie den folgenden Code in die *functions.php* ein.

```
if ( ! function_exists( 'wh_block_load_textdomain' ) ) {
  function wh_block_load_textdomain() {
    load_theme_textdomain( 'wh-block, get_template_directory() .
'/languages' );
  }
}
add_action( 'after_setup_theme', 'wh_block_load_textdomain' );
```

Listing 23.20 Das Laden der Übersetzungen ist in Block- und Classic Theme gleich.

Mit `load_theme_textdomain()` werden die vorhandenen Übersetzungen geladen. Der erste Parameter ist die Text-Domain, für die die Übersetzungen angezeigt werden sollen, der zweite, optionale Parameter gibt den Pfad zum Verzeichnis mit den Übersetzungen an. Wenn Sie Ihr Theme im WordPress.org-Theme-

Verzeichnis veröffentlichen, müssen Sie diese Pfadangabe weglassen, da die Übersetzung der dort gehosteten Themes über *translate.wordpress.org* erfolgt.

Wie Sie eine Übersetzung erstellen können, wird in Abschnitt 22.2.3, »Übersetzungen (Internationalisierung)«, erläutert.

Die Strings allerdings, die in den Templates vorkommen, liegen als Block-Attribute oder Werte von HTML-Elementen in den HTML-Template-Dateien, für die es keine Übersetzungsmöglichkeit gibt. Ein Umweg ist die Nutzung von Vorlagen. Da diese in PHP-Dateien definiert werden, können die normalen WordPress-Funktionen zur Übersetzung genutzt werden, und eine Block-Vorlage kann über `<!-- wp:pattern {"slug":"wh-block/pattern-slug"} /-->` in ein Template eingebunden werden. Es ist sicher nicht die schönste Lösung, aber eine andere gibt es momentan nicht.

Eine Vorlage kann bei der Registrierung über die Angabe `'inserter' => false` (klassische Registrierung) oder `Inserter: no` (bei der Registrierung über das *patterns*-Verzeichnis in einem Block-Theme) aus dem Inserter entfernt werden, sodass solche Vorlagen, die nur für die Übersetzbarkeit existieren, nicht wie normale andere Vorlagen eingefügt werden können.

23.2 Classic Theme anpassen und erstellen

In diesem Abschnitt werden Sie zunächst lernen, wie ein *Classic Theme* anzupassen ist, und danach, wie Sie ein eigenes Theme entwickeln können.

23.2.1 Bestehendes Theme mit einem Child-Theme anpassen

Der wichtigste Aspekt bei der Anpassung eines Classic Themes ist, dass Sie nicht direkt den Code des Themes verändern, sondern Änderungen in einem sogenannten *Child-Theme* vornehmen. So verhindern Sie, dass die Änderungen bei einem Update des Themes verloren gehen.

Ein Child-Theme ist aber keine Kopie des Themes, das Sie verändern möchten, sondern es beinhaltet nur die Anpassungen am Parent-Theme, der Rest wird vererbt. WordPress lädt zunächst die PHP-Dateien des Parent-Themes und überschreibt die Dateien, die auch im Child-Theme vorhanden sind, mit den Dateien aus dem Child-Theme. Dabei gibt es die Ausnahme der *functions.php*, die nicht überschrieben, sondern zusätzlich geladen wird.

Falls der Hersteller des Themes, das Sie anpassen möchten, ein Child-Theme anbietet, verwenden Sie ruhig das, eventuell werden dabei direkt Besonderheiten berücksichtigt, die es beim Theme zu beachten gilt. Allgemein ist das Vorgehen beim Anlegen eines Child-Themes aber immer gleich. Das folgende Beispiel zeigt Ihnen das Anlegen eines Child-Themes zum letzten WordPress-Standard-Theme, das als Classic Theme umgesetzt wurde: Twenty Twenty-One.

1. Legen Sie unter *wp-content/themes* ein Unterverzeichnis für Ihr Child-Theme an. Sie können es grundsätzlich beliebig benennen, im Beispiel wird der Verzeichnisname *twentytwentyone-child* genutzt.
2. Erstellen Sie eine *style.css* im gerade angelegten Verzeichnis mit dem folgenden Inhalt:

```
/*
Theme Name: Twenty Twenty-One Child
Template: twentytwentyone
*/
```

   Das sind die mindestens notwendigen Angaben, damit WordPress das Design als Child-Theme unter Design • Themes angezeigt wird. `Template` ist der Slug des Themes, das Sie mit dem Child-Theme erweitern möchten. *Hinweis:* Die vollständige Liste an Metadaten, die Sie einem Theme mitgeben können, finden Sie in Abschnitt 23.2.3, »Theme-Aufbau«.

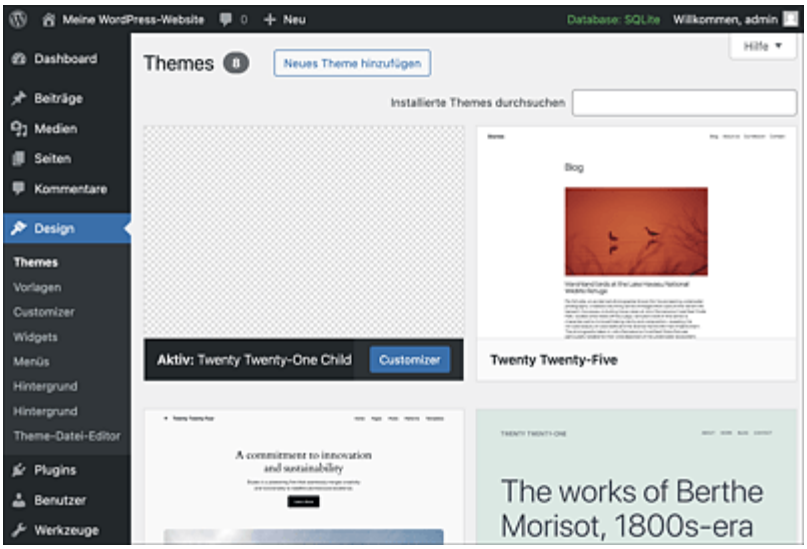


Abbildung 23.27 Nach Anlegen der Datei »style.css« erscheint das Child-Theme bereits im WordPress-Backend und kann aktiviert werden. Jetzt kann es losgehen mit den Anpassungen. (Soll unter »Design« • »Themes« ein Vorschaubild erscheinen, legen Sie einen Screenshot namens »screenshot.png« ins Child-Theme-Verzeichnis.)

3. Wenn Sie das Child-Theme aktivieren und das Frontend aufrufen, werden Sie sehen, dass die Website aussieht wie mit einem aktiven Twenty Twenty-One. Das liegt daran, dass Twenty Twenty-One seine Styles so einbindet, dass sie auch bei Nutzung eines Child-Themes automatisch geladen werden. Bei anderen Themes kann das anders sein, dann müssen Sie das Stylesheet des Parent-Themes selbst über die *functions.php* laden. Hier gibt es eine Einschränkung: Wenn Sie Änderungen im Customizer vorgenommen haben, werden diese nicht mit übernommen, da sie mit dem Theme verknüpft sind.

Damit haben Sie bereits ein minimales Child-Theme erstellt. Wenn Styles überschrieben werden sollen, müssen Sie die *style.css* des Child-Themes einbinden, was nur ein paar Zeilen Code benötigt. Erstellen Sie dazu eine *functions.php* im Child-Theme-Ordner, und fügen Sie folgenden Code ein:

```
<?php
function ttoc_enqueue_style() {
    wp_enqueue_style(
        'twentytwentyone-child-style',
        get_stylesheet_uri(),
        [ 'twenty-twenty-one-style' ],
        filemtime( get_stylesheet_directory() . '/style.css' )
    );
}
add_action( 'wp_enqueue_scripts', 'ttoc_enqueue_style' );
```

Listing 23.21 Laden der »style.css«-Datei in der »functions.php«-Datei im Child-Theme

In der Funktion wird mit `wp_enqueue_style()` die WordPress-Funktion aufgerufen, die sich um das Einbinden von Stylesheets kümmert. Als ersten Parameter erwartet die Funktion ein eindeutiges Handle, mit dem sich beispielsweise andere Styles an dieses Stylesheet hängen können. Der zweite Parameter ist die URL zum Stylesheet, hierfür bietet WordPress die handliche Funktion `get_stylesheet_uri()` an, die die URL zur *style.css* des Themes

zurückgibt. Der dritte Parameter ist optional, hier können Abhängigkeiten angegeben werden (die Werte im Array sind die Handles der Stylesheets, wie sie als erster Parameter übergeben wurden), sodass unser Stylesheet erstens immer zusammen mit den Abhängigkeiten geladen und zweitens danach eingefügt wird (wichtig für die CSS-Kaskade). `twenty-twenty-one-style` ist das Handle der *style.css* von Twenty Twenty-One. Wichtig: Wenn Sie hier versehentlich ein Handle angeben, das nicht existiert, wird Ihr Stylesheet nicht geladen. Als letzten Parameter übergeben wir schließlich einen Zeitstempel (Unix-Timestamp) des Zeitpunkts, an dem die *style.css*-Datei zuletzt gespeichert wurde, sodass Änderungen Ihren Besucher*innen direkt angezeigt und nicht eventuell von im Cache liegenden älteren Versionen verschluckt werden.

Am Ende hängen wir unsere Funktion über den Action-Hook `wp_enqueue_scripts` in WordPress ein und haben es geschafft: Unsere CSS-Datei sollte jetzt mit geladen werden, und Sie können Ihre CSS-Anpassungen in dieser Datei vornehmen.

Möchten Sie einzelne Template-Dateien verändern, beispielsweise die *header.php*, kopieren Sie sie einfach vom Parent-Theme-Ordner in das Child-Theme-Verzeichnis, und WordPress wird diese Datei statt die aus dem Parent-Theme nutzen. Die einzige Template-Datei, die sich so nicht überschreiben lässt, ist die *functions.php*, hier wird die Parent-Datei durch die Child-Datei erweitert.

Achtung: Änderungen im Parent-Theme gegebenenfalls ins Child-Theme übernehmen

Auch wenn Sie mit einem Child-Theme Ihre Änderungen bei einem Update des Parent-Themes nicht verlieren, sollten Sie sich bei jedem Update darüber informieren, welche Änderungen

gemacht wurden. Zum einen kann es sein, dass sich Dinge ändern, die Sie in Ihr Child-Theme übernehmen müssen (beispielsweise wenn eine Template-Datei verändert wurde, die Sie im Child-Theme überschrieben haben), und zum anderen können auch in Themes Sicherheitslücken auftreten, die bei Updates behoben werden. Wenn Sie jetzt gerade die Datei in Ihr Child-Theme übernommen haben, die eine Sicherheitslücke enthält, ist diese Sicherheitslücke auch nach dem Update des Parent-Themes noch da und muss durch Übernahme des Fixes in die Child-Theme-Datei übernommen werden. Um Änderungen an einem Theme vor und nach dem Update zu vergleichen, können Sie unterschiedlich vorgehen.

- Erste Möglichkeit, wenn das Theme aus dem WordPress.org-Verzeichnis ist (was bei Twenty Twenty-One der Fall ist): Jedes Theme hat dort ein SVN-Repo, in dem Sie sich die geänderten Dateien zwischen zwei Versionen anschauen können. Rufen Sie dazu *https://themes.trac.wordpress.org/log/twentytwentyone/2.3* auf und ersetzen Sie `twentytwentyone` durch den Slug Ihres eingesetzten Parent-Themes und `2.3` durch die neue Version. Klicken Sie auf der Seite auf den Button **View Changes**, und es öffnet sich eine Ansicht, in der Sie die Änderungen zwischen der Version aus der URL und der Version davor sehen können.
- Zweite Möglichkeit, wenn das Theme öffentlich auf GitHub oder einer anderen Plattform zur Versionskontrolle liegt: Sie können sich die Unterschiede zwischen zwei Versionen anschauen. Wie das aussehen kann, sehen Sie unter *https://github.com/WordPress/twentytwentytwo/compare/v0.5.0...v1.0* am Vergleich zwischen Twenty Twenty-Two Version 1.0 und 0.5.0 (das Repository ist archiviert, weil das Theme nach

der Aufnahme im Core über ein anderes System gepflegt wird, aber dem Beispiel tut das ja keinen Abbruch).

- Wenn das Theme nirgendwo öffentlich zur Verfügung steht, müssen Sie wohl oder übel die alte und neue Version auf Ihren Rechner laden und dann mit einem Dateivergleichsprogramm die Ordner vergleichen. Achten Sie darauf, das Datum der Dateien aus dem Vergleich auszuschließen, das ist wahrscheinlich bei allen Dateien neu.



Abbildung 23.28 Für Themes aus dem WordPress.org-Verzeichnis können Sie sich Änderungen zwischen Versionen anschauen, hier beispielsweise zwischen Twenty Twenty-One 2.2 und 2.3.

Welche Template-Dateien sind involviert?

Der in Abschnitt 22.3.3 vorgestellte Query Monitor zeigt Ihnen neben vielen weiteren nützlichen Informationen auch an, welche Template-Datei geladen wurde und welche weiteren Template-Dateien in diese Datei eingebunden wurden. Dazu rufen Sie mit aktiviertem Plugin einfach eine Seite im Frontend auf und wählen oben in der Werkzeugleiste über den Punkt mit den Performance-Informationen zur angezeigten Seite den Eintrag Template, um das Query-Monitor-Fenster am unteren Viewport-Rand zu öffnen und mehr Infos zu den verwendeten Template-Dateien zu erhalten.

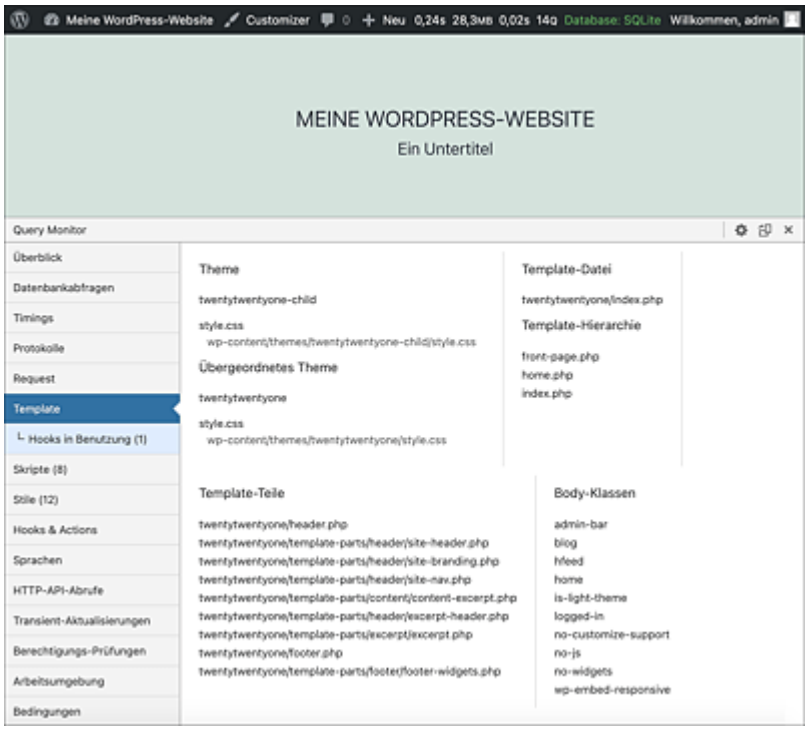


Abbildung 23.29 Das Plugin Query Monitor gibt Ihnen unter anderem Auskunft darüber, welche Template-Datei und welche Template-Teile geladen wurden.

Template-Datei anpassen

Im Footer von Twenty Twenty-One wird der Website-Titel (oder, falls gesetzt, das Website-Logo) angezeigt sowie ein Powered-by-WordPress-Link (siehe Abbildung 23.30). Lassen Sie uns im Child-

Theme den Footer des Themes anpassen und diese beiden Elemente entfernen.



Abbildung 23.30 Den Website-Titel und den Hinweis auf WordPress wollen wir im Footer nicht haben.

Dafür kopieren Sie zunächst die *footer.php* aus dem *twentytwentyone*-Verzeichnis in Ihr Child-Theme-Verzeichnis und suchen in der Datei den entsprechenden Teil, der entfernt werden soll. Ungefähr in der Mitte der Datei (Zeile 42 in Twenty Twenty-One 1.5) finden Sie ein `<div class="site-info">`, dessen erste Zeilen so aussehen wie in Listing 23.22:

```
<div class="site-info">
    <div class="site-name">
        <?php if ( has_custom_logo() ) : ?>
            <div class="site-logo"><?php the_custom_logo(); ?></div>
        <?php else : ?>
```

Listing 23.22 Die ersten Zeilen des »div«-Elements im Footer von Twenty Twenty-One 2.3, das die beiden zu entfernenden Elemente enthält

Neben dem Titel/Logo und dem Hinweis zu WordPress wird in diesem `div` auch noch ein Link zu der Datenschutzerklärungsseite ausgegeben (`the_privacy_policy_link()`). Da Sie diesen Link vermutlich in einem Menü zusammen mit dem Impressum unterbringen werden, können Sie diesen Teil ebenfalls entfernen und einfach das gesamte div mit der Klasse `site-info` löschen.

Nachdem Sie die Datei gespeichert haben und die Webseite aktualisieren, sollte der Teil aus dem Footer verschwunden sein.

23.2.2 Ein Theme von Grund auf neu entwickeln

Wenn Sie Anforderungen haben, die von keinem bestehenden Theme erfüllt werden können, muss ein eigenes Theme her. In den folgenden Abschnitten geht es zunächst ein bisschen um theoretische Aspekte der Theme-Entwicklung. Eine gute Idee ist, sich bei der Lektüre parallel Dateien von beispielsweise dem Theme Twenty Twenty-One durchzuschauen. Nach den Grundlagen folgt die Erstellung eines einfachen Themes. Dabei wird es darum gehen, ein ähnliches Ergebnis umzusetzen wie im Abschnitt zur Block-Theme-Entwicklung.

Grundlagen zur Theme-Entwicklung

Verglichen mit dem Aufwand der Anpassung eines gut passenden, fertigen Themes ist die Entwicklung eines eigenen Themes natürlich aufwendig. Allerdings haben Sie bei einem eigenen Theme nicht nur den Lerneffekt, sondern am Ende auch ein Theme, das nur genau das kann, was Sie benötigen, und keinen unnötigen Ballast mitbringt.

Auf den folgenden Seiten wird der theoretische Aufbau besprochen, für den Praxisteil blättern Sie zum Folgeabschnitt, in dem Sie ein einfaches Theme von Grund auf neu entwickeln werden.

Für ein Classic Theme brauchen Sie normalerweise Folgendes:

- ein Theme-Verzeichnis in *wp-content/themes*
- eine *style.css*, die wie bei einem Child-Theme die Metadaten des Themes enthält und eine der beiden zwingend notwendigen Dateien eines Classic Themes ist
- die Datei *functions.php*, in der beispielsweise das Stylesheet eingebunden und angegeben wird, ob das Theme WordPress-

Funktionen wie Beitragsbilder unterstützt

- Template-Dateien, die sich um die Ausgabe der Inhalte kümmern; die *index.php* ist eine davon und neben der *style.css* die zweite Datei, die für ein Classic Theme zwingend notwendig ist
- optional: Bilder, wenn Sie etwa Icons einsetzen
- optional: JavaScript-Dateien

23.2.3 Theme-Aufbau

Bevor es an die Entwicklung des Themes geht, kommen normalerweise Konzeption und Design. Hier werden die Anforderungen definiert, die Sie dann mit dem Theme umsetzen müssen. Wenn Sie also nicht allein an dem Projekt arbeiten, ist es wichtig, in der Phase schon dabei zu sein, um zu verhindern, dass unrealistische Anforderungen definiert werden.

Info: Keine Entwicklung an der Live-Website

Egal, wie klein oder groß ein Projekt ist: Sie sollten nie direkt an einer Website arbeiten, die öffentlich zugänglich ist, sondern immer Änderungen lokal oder auf einer anderen Test-Instanz vornehmen, bevor Sie sie auf live deployen.

Die Mindestanforderung an ein Theme

Das einfachstmögliche Theme besteht aus einer *style.css* und einer *index.php*. Wie bereits angerissen übernimmt die *style.css* dabei einen wichtigen Part, denn sie enthält oben als Kommentar die Metadaten des Themes. Ohne diese Informationen weiß WordPress nicht, dass es sich um ein Theme handelt, und mindestens

notwendig ist die Angabe des Theme Name, des Namens des Themes. Wenn Sie das Theme beispielsweise im WordPress.org-Verzeichnis veröffentlichen möchten, sind daneben noch weitere Angaben verpflichtend, wie der Name der Autor*innen, die Angabe der Lizenz, unter der das Theme veröffentlicht wird, und eine Text-Domain für die Übersetzbarkeit (*https://developer.wordpress.org/themes/basics/main-stylesheet-style-css/#explanations*).

Die Datei *index.php* ist das Fallback-Template, auf das immer zurückgefallen wird, wenn keine spezifischeren Templates im Theme vorhanden sind, um beispielsweise eine Seite anzuzeigen.

Theme-Metadaten

In der *style.css*, die im Hauptverzeichnis jedes Themes liegt, werden wie beschrieben nicht nur die CSS-Styles eingefügt, sondern die Theme-Metadaten. Diese werden am Anfang der Datei als CSS-Kommentar (zu erkennen mit `/*` am Anfang und `*/` am Ende) eingefügt.

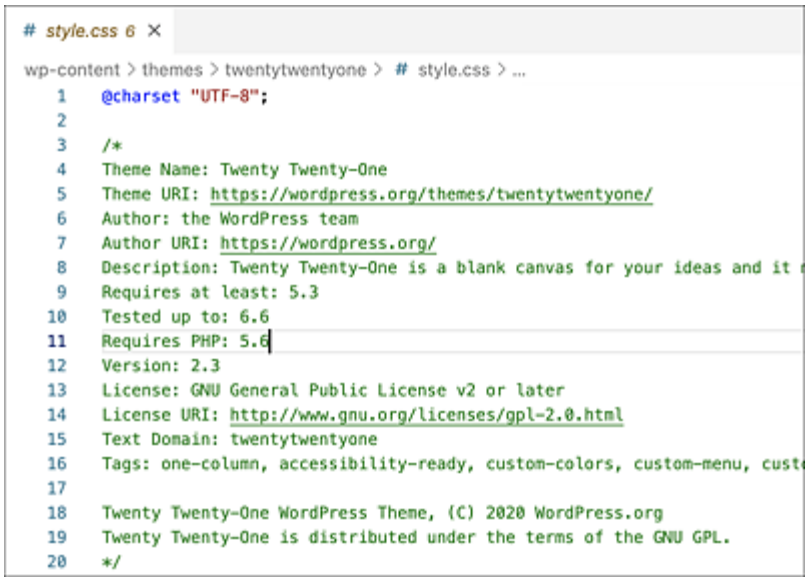


Abbildung 23.31 Der Theme-Meta-Kommentar am Anfang der »style.css« des Standard-Themes Twenty Twenty-One

1. `Theme Name`: Name des Themes, wie er in Backend und Theme-Verzeichnissen angezeigt wird
2. `Theme URI`: Website des Themes, auf der optimalerweise Verweise auf eine Dokumentation, Download-Möglichkeit und Hinweise zum Support zu finden sind
3. `Author`: Name der Entwickler*in oder des Unternehmens dahinter
4. `Author URI`: Website der Entwickler*in
5. `Description`: die Beschreibung des Themes
6. `Version`: sehr wichtig, damit Updates installiert werden können
7. `License`: Die Lizenz, unter der das Theme veröffentlicht ist, muss mit der GNU General Public License v2 kompatibel sein, die WordPress nutzt. Das bezieht sich auf alle Bestandteile des Themes, also auch beispielsweise Bilder und Skripte.
8. `License URI`: Seite, auf der die Lizenz nachzulesen ist
9. `Text-Domain`: Bezeichner für die Übersetzungen des Themes, entspricht in der Regel dem Ordnernamen des Themes
10. `Tags`: Begriffe für die Suche im WordPress.org-Theme-Verzeichnis

Die Template-Hierarchie

Wenn im Frontend einer WordPress-Instanz ein Inhalt aufgerufen wird, sei es die Startseite, eine Einzelansicht eines Beitrags oder die Beiträge einer bestimmten Kategorie, hat WordPress eine Liste von

Template-Dateien, die im Theme gesucht werden, um den Inhalt darzustellen. Dabei wird zuerst nach sehr spezifischen Template-Dateien gesucht, und dann nach immer allgemeineren, bis am Ende die *index.php* steht, der letzte Fallback in der sogenannten *Template-Hierarchie*, weshalb diese *index.php* auch das einzig zwingend notwendige Template eines Themes ist (*https://developer.wordpress.org/themes/basics/template-hierarchy/*).

Auf den folgenden Seiten lernen Sie spezielle Template-Dateien für unterschiedliche Website-Bereiche kennen, es gibt aber darüber hinaus noch weitere für noch speziellere Fälle.

Startseiten-Templates

Für die Startseite Ihrer Website probiert WordPress unterschiedliche Templates durch, wobei die Liste der Templates abhängig davon ist, ob auf der Startseite die Liste der letzten Beiträge oder eine statische Seite angezeigt wird (Konfiguration unter Einstellungen • Lesen).

Template-Datei	Zweck
front-page.php	Wenn diese Datei im Theme existiert, wird sie in jedem Fall für die Startseite genutzt.
home.php	Wenn es keine *front-page.php* gibt und die neuesten Beiträge auf der Startseite angezeigt werden sollen, greift die *home.php*.

Template-Datei	Zweck
page.php, *singular.php* und die spezifischeren Varianten	Wird statt der neuesten Beiträge eine statische Seite als Startseite genutzt, wird die Template-Hierarchie für eine normale Seite durchlaufen, die Sie weiter unten detaillierter beschrieben bekommen. Die letzten beiden Dateien vor der *index.php* sind hier die *page.php* und die *singular.php*.
index.php	Der letzte Fallback.

Einzelansichten von Seiten und Beiträgen

Wenn eine Seite oder ein Beitrag aufgerufen wird, greift die folgende Fallback-Kette.

Template-Datei	Zweck
Individuelles Page-Template	Sie können bei einem Theme zusätzlich zum normalen Template für Seiten noch weitere Templates mitliefern, die im Backend dann individuell pro Seite festgelegt werden können.
page-{id|slug}.php	Template für eine bestimmte Seite, angesprochen über die ID oder den Slug.
page.php	Allgemeines Template für die Einzelansicht einer Seite.

Template-Datei	Zweck
single-{inhaltstyp}-{slug}.php und *single-{inhaltstyp}.php*	Template für die Einzelansicht eines bestimmten Eintrags eines bestimmten Inhaltstyps. Die *single-portfolio-hannover.php* würde zum Beispiel greifen, wenn der Eintrag mit dem Slug `hannover` und dem Inhaltstyp `portfolio` angezeigt wird. Als nächste Abstufung gibt es die *single-{inhaltstyp}.php*, um ein spezielles Template für alle Einträge eines Inhaltstyps festzulegen.
single.php	Allgemeines Template für die Darstellung der Einzelansicht eines Beitrags.
singular.php	Letztes Template vor der *index.php* zur Darstellung eines Beitrags oder einer Seite.

Sonderseiten

Zusätzlich zu den beschriebenen Fällen gibt es einige Sonderseiten bei einer WordPress-Website, zum Beispiel die Übersicht aller Beiträge einer bestimmten Kategorie oder aus einem bestimmten Jahr. Bei den Ansichten für mehrere Beiträge ist die *archive.php* der allgemeine Fallback vor der *index.php*.

Template-Datei	Zweck
archive-{inhaltstyp}.php	Template zur Anzeige der Beitragsübersicht eines bestimmten Inhaltstyps.
category-{id|slug}.php und *category.php*	Templates zur Anzeige der Beiträge einer Kategorie. Über die Nutzung der Kategorie-ID oder des Kategorie-Slugs als Suffix im Dateinamen kann für eine bestimmte Kategorie auch eine andere Darstellung verwendet werden.

Template-Datei	Zweck
tag-{id|slug}.php und *tag.php*	Äquivalent zu den Template-Dateien für Kategorien, nur für Tags.
author-{nicename|id}.php und *author.php*	Darstellung der Seite mit den Beiträgen einer bestimmten Autor*in
search.php	Template für die Suchergebnisseite.
taxonomy-{taxonomie}-{begriff}.php, *taxonomy-{taxonomie}.php* und *taxonomy.php*	Templates für die Beiträge einer Kategorie. Sie können auch ein separates Template für eine bestimmte Taxonomie oder sogar einen Begriff (Term) einer Taxonomie erstellen.
archive.php	Allgemeines Fallback-Template für die vorgenannten Archiv-Ansichten vor dem Fallback auf die *index.php*.
{mime-type}.php und *attachment.php*	Templates zur Ausgabe von Datei-Anhangseiten. Sie können medienspezifisch sein, beispielsweise eine *image.php* für alle Bilder, oder noch spezifischer, beispielsweise eine *image-png.php* für PNG-Bilder. Mit der *attachment.php* haben Sie einen Fallback vor der *single.php* für alle Medienarten.
404.php	Template für den Fall, dass eine Seite aufgerufen wird, die es nicht gibt.

Auslagern von Header, Footer und Widget-Bereichen

In vielen Classic Themes finden Sie eine *header.php* und *footer.php*, eventuell auch eine *sidebar.php*. Die Auslagerung dieser Bereiche,

die in der Regel auf der gesamten Website unverändert bleiben, ist so gängig, dass es in WordPress eigene Funktionen zum Einbinden dieser Dateien gibt: `get_header()`, `get_footer()` und `get_sidebar()`. Über die Angabe eines Parameters können Sie auch spezialisierte Templates einbinden, beispielsweise eine Header-Datei für einen Portfolio-Bereich über `get_header( 'portfolio' )`, um die Datei *header-portfolio.php* zu laden.

Modulare Inhaltselemente in Template-Teilen

Ebenfalls gängige Praxis ist die Auslagerung von Template-Teilen in einen *template-parts*-Ordner. Hier könnte zum Beispiel eine Datei liegen, die sich um die Ausgabe des Menüs und des Logos kümmert, oder eine Datei für die Ausgabe von Teaser-Elementen für Beiträge in einer Übersichtsliste.

Diese Teile können Sie dann in unterschiedlichen Template-Dateien einbinden und so verhindern, dass Code mehrfach in Dateien steht. Um *Template-Parts* einzubinden, gibt es von WordPress die Funktion `get_template_part()`. Als ersten Parameter übergeben Sie den Slug des Teils an, so können Sie mit `get_template_part( 'template-parts/content' )` die Datei *template-parts/content.php* einbinden. Als zweiten Parameter können Sie optional einen Namen für einen spezielleren Template-Teil angeben, um beispielsweise mit `get_template_part( 'template-parts/content', 'single' )` die *template-parts/content-single.php* einzubinden. Wenn diese Datei nicht vorhanden sein sollte, wird die *template-parts/content.php* eingebunden.

Info: Block-basierte Template-Teile

Seit WordPress 6.1 ist es möglich, die Template-Part-Systematik von Block-Themes auch in Classic Themes zu verwenden. Dafür muss lediglich die Unterstützung im Theme aktiviert werden:

```
add_theme_support( 'block-template-parts' );
```

Danach können Sie im *parts*-Ordner des Themes HTML-Dateien mit dem Markup von Blöcken erstellen und diese mit der `block_template_part()`-Funktion einbinden. Bearbeiten lassen sich diese Template-Teile anschließend im Backend unter **Design** • **Vorlagen** • **Alle Template-Teile**.

Das Theme umsetzen

Sie haben jetzt viele verschiedene mögliche Template-Dateien kennengelernt, Sie werden in der Regel aber nur einen Bruchteil davon benötigen, besonders die sehr spezifischen Templates kommen doch eher selten zum Einsatz.

In diesem Abschnitt werden Sie ein Theme von Grund auf erstellen. Es wird ein sehr einfaches Theme sein – mit einem Header, einem Inhaltsbereich und einem Footer, in den Widgets eingefügt werden können. Aber auch durch dieses sehr einfache Beispiel werden Sie viele Aspekte der Theme-Entwicklung kennenlernen und dieses Wissen als Basis für komplexere Themes nutzen können.

Um bei der Entwicklung das Theme gut testen zu können, sollten Sie sich mehrere Beiträge und Seiten mit unterschiedlichen Inhalten anlegen, inklusive Bildern. Verwenden Sie dabei am besten nicht die vielgenutzten Lorem-Ipsum-Texte, denn die deutsche Sprache hat deutlich längere Wörter als die Wörter in diesen Dummy-Texten. Es ist auch eine gute Idee, eine Seite mit allen verfügbaren Blöcken anzulegen, um an einer Stelle prüfen zu

können, ob Styles für alle Blöcke vorhanden sind, die Sie unterstützen möchten.

Einfach haben Sie es natürlich, wenn eine bestehende Website und damit auch bestehende Inhalte vorhanden sind, die Sie nutzen können.

Theme-Grundgerüst aus style.css und index.php anlegen

Wie in diesem Abschnitt beschrieben, benötigt ein Theme mindestens eine *style.css*- und eine *index.php*-Datei, damit es von WordPress korrekt erkannt wird und aktiviert werden kann. Legen Sie also ein Verzeichnis unter *wp-content/themes* für Ihr neues Theme an (im Beispiel heißt das Theme-Verzeichnis *wh-classic*) und erstellen Sie darin die beiden Dateien. Später wird die *style.css* auch die CSS-Regeln beinhalten, erst mal fügen Sie aber nur den Kommentar mit den Metainformationen ein, die WordPress für die Anzeige im Backend verwendet:

```
/*
Theme Name: WH Classic
Author: Ihr Name
Requires at least: 6.0
Tested up to: 6.7
Requires PHP: 7.0
Version: 1.0
License: GNU General Public License v2 or later
License URI: http://www.gnu.org/licenses/gpl-2.0.html
Text Domain: wh-classic
*/
```

Listing 23.23 Metadaten in der »style.css«-Datei

Wirklich notwendig ist nur der Name des Themes. Daneben geben Sie aber noch den*die Autor*in des Themes an, welche WordPress-Version mindestens zur Nutzung notwendig ist (`Requires at least`), mit welcher WordPress-Version es getestet wurde (`Tested up to`), welche PHP-Version mindestens vorhanden sein muss und um

welche Theme-Version es sich handelt. Dazu dann Infos zur Lizenz (alles, was sich im Theme-Ordner befindet, muss mit dieser Lizenz kompatibel sein) und die Text-Domain zur Übersetzbarkeit.

Tipp: Update URI-Angabe für Themes außerhalb des W.org-Repos

Wenn Ihr Theme nicht im offiziellen WordPress.org-Theme-Verzeichnis eingereicht werden soll, können Sie über die Angabe des `Update URI`-Keys in den Metadaten verhindern, dass Ihr Theme versehentlich durch ein Theme mit demselben Slug aus dem WordPress.org-Verzeichnis überschrieben wird.

Wenn ein installiertes Theme einen Slug hat, den es auch im WordPress.org-Verzeichnis gibt, geht WordPress nämlich normalerweise davon aus, dass es sich dabei um dieses Theme handelt. Wenn das nicht der Fall ist, aber das Theme im Verzeichnis eine höhere Version hat, würde in Ihrer Installation ein Update angeboten werden, was Ihr Theme dann mit dem aus dem Verzeichnis überschreibt.

Seit WordPress 6.1 können Sie das durch die Angabe einer Update-URI in den Metadaten verhindern, den gleichen Header gibt es für Plugins bereits seit WordPress 5.8. Das kann dann wie folgt aussehen:

```
Update URI: https://example.com/ihr-theme
```

Eine weitere Neuerung in WordPress 6.1 ist die Einführung des `update_themes_{$hostname}`-Filters. Über diesen Filter können Sie Updates für Themes bereitstellen, die diese Update-URI-Angabe nutzen, sodass in der Installation wie gewohnt Updates installiert werden können, auch wenn das Theme nicht im WordPress.org-Verzeichnis gehostet ist.

Die *index.php*-Datei lassen Sie erst einmal leer und aktivieren das Theme im Backend unter Design • Themes. Wenn Sie sich jetzt die Website anschauen, sollten Sie eine weiße Seite sehen.

Der letzte Fallback: die index.php

Beginnen wir nun, die *index.php* mit Inhalt zu füllen. Wie Sie gelernt haben, handelt es sich dabei um das letzte Fallback-Template in der Template-Hierarchie. In unserem Theme wird sie nur dazu da sein, die Blogübersicht anzuzeigen und daneben natürlich den Website-Header und -Footer einzubinden. Die komplette *index.php* ist in Listing 23.24 abgebildet.

```
<?php
get_header();
if ( is_home() && ! is_front_page() ) {
  printf(
    '<header class="screen-reader-text">%s</header>',
    single_post_title( '', false )
  );
}
if ( have_posts() ) {
  while ( have_posts() ) {
    the_post();
    get_template_part(
      'template-parts/content/content',
      get_post_type()
    );
  }
} else {
   get_template_part(
     'template-parts/content/content',
     'none'
  );
}

the_posts_pagination(
  [
    'type'      => 'list',
    'prev_text' => __( 'Previous', 'wh-classic' ),
    'next_text' => __( 'Next', 'wh-classic' ),
  ]
);
get_footer();
```

Listing 23.24 Der komplette Inhalt der »index.php«-Datei des Themes

Schritt für Schritt erklärt:

1. `get_header()` bindet die *header.php* des Themes ein. Theoretisch könnte der Header auch direkt mit in die *index.php*, aber erstens müsste der gesamte Code dann in jedem Template wiederholt werden, und zweitens würde die Datei recht lang und unübersichtlich.
2. Anschließend muss ein Fall behandelt werden, in dem sonst keine `h1` ausgegeben würde, nämlich wenn die Blogseite über die Einstellungen in WordPress auf eine andere Seite als Startseite verlegt wurde. In diesem Fall greift dieses Template, und für den Fall muss der Titel der Seite als `h1` ausgegeben werden, die als Blogseite gesetzt wurde. Dafür wird mit `is_home() && ! is_front_page()` geprüft, ob zwar die Blogseite angezeigt wird, es sich aber nicht um die Startseite handelt. Für den Fall geben wir ein visuell verstecktes `header`-Element aus (das CSS für die `screen-reader-text`-Klasse wird vom Core mitgeliefert und eingebunden), in das über `single_post_title()` der Titel der Seite innerhalb einer `h1`-Überschrift ausgegeben wird.
3. Danach wird der Loop ausgeführt, von dem Sie schon im Buch gelesen haben, um die Beiträge, die zur aufgerufenen URL passen, zu durchlaufen und anzuzeigen. Dazu wird zunächst geprüft, ob Beiträge vorhanden sind (`if ( have_posts() )`), und wenn das gegeben ist, werden die Beiträge mit einer `while`-Schleife durchlaufen. Innerhalb der Schleife wird mit `the_post()` der nächste Post in der Schleife ausgewählt, und dessen Daten werden zugänglich gemacht, sodass sie im Template einfach ausgegeben werden können, ohne einen spezifischen Beitrag als Parameter an Funktionen übergeben zu müssen, etwa um

den Titel des Beitrags anzuzeigen. Nachdem das erledigt ist, wird mit der Funktion `get_template_part()` eine andere Datei des Themes eingebunden, in der Sie später die konkrete Ausgabe der Beitragsinformationen vornehmen werden. Über den ersten Parameter geben Sie den Slug der Datei an, der in diesem Fall auf die *content.php* im Verzeichnis *template-parts/content* zeigt. Über den zweiten Parameter kann optional ein Name angegeben werden, sodass WordPress vor der *content.php* nach einer Datei *content-{name}.php* sucht und erst die *content.php* lädt, wenn die spezifischere Datei nicht existiert. Im Code wird mit `get_post_type()` der Inhaltstyp als Name ausgegeben, das heißt, für die Anzeige von Beiträgen wird erst nach einer *template-parts/content/content-post.php* gesucht, bevor die *template-parts/content/content.php* eingebunden wird.

4. Wenn keine Beiträge zum Anzeigen vorhanden sind, wird ein Template eingebunden, das die Besucher*innen darüber informiert (*template-parts/content/content-none.php*).
5. Für den Fall, dass so viele Beiträge vorhanden sind, dass sie auf mehrere Seiten umbrochen werden, wird mit `the_posts_pagination()` eine Paginierung angezeigt, für die wir den Typ sowie den Text der Weiter- und Zurück-Links festlegen (hier wie bei allen weiteren für User sichtbaren Strings werden Übersetzungsfunktionen verwendet, auf die in Abschnitt 22.2.3, »Übersetzungen (Internationalisierung)«, näher eingegangen wird).
6. Abschließend wird mit `get_footer()` die *footer.php* des Themes eingebunden.

Wenn Sie das Frontend neu laden, sollte sich nichts verändert haben und immer noch die weiße Seite zu sehen sein. Als Nächstes

kümmern Sie sich um den Kopfbereich der Seite, die *header.php*.

Der Kopf der Seite: die header.php und die Header-Template-Teile

In der *header.php* wird der Anfang des HTML-Dokuments definiert, aus dem die Website besteht. Das heißt, hier findet nicht nur der sichtbare Header Platz, sondern auch wichtige Teile, die zwar für die Besucher*innen nicht sichtbar, aber unverzichtbar für eine funktionierende Website sind. Beginnen wir wieder mit dem Code der Datei (siehe Listing 23.25).

```
<!DOCTYPE html>
  <html <?php language_attributes(); ?>>
  <head>
    <meta charset="utf-8">
    <meta
      name="viewport"
      content="width=device-width, initial-scale=1, minimum-scale=1"
    >
    <?php wp_head(); ?>
  </head>
  <body <?php body_class(); ?>>
    <?php wp_body_open(); ?>
    <a class="skip-link screen-reader-text" href="#content">
      <?php esc_html_e( 'Skip to content', 'wh-classic' ); ?>
    </a>
    <header class="site-header inverted-link-style">
      <?php
      get_template_part(
        'template-parts/header/branding'
      );
      get_template_part(
        'template-parts/header/nav'
      );
      ?>
    </header>
    <main id="content">
```

Listing 23.25 Der Kopfbereich der Website wird in der »header.php« definiert.

1. Nach dem Doctype wird das `html`-Element eingefügt, in dem mittels `language_attributes()`-Aufruf das `lang`-Attribut mit dem

Wert ausgegeben wird, der zur Spracheinstellung von WordPress passt (wenn Deutsch als Website-Sprache eingestellt ist, sieht das so aus: `lang="de-DE"`).

2. Innerhalb des `head`-Elements wird der Zeichensatz der Site (`charset`) auf UTF-8 festgelegt, damit alle Zeichen problemlos angezeigt werden, und das Viewport-Meta-Element legt fest, dass der Viewport der Screen-Breite entspricht und der Zoomfaktor initial `1` sein soll. Diese Angabe ist wichtig, da sonst Responsive Design auf mobilen Geräten nicht greift und die Website im Layout wie auf großen Geräten angezeigt wird (rausgezoomt, damit es alles auf das kleinere Display passt). Mit `wp_head()` wird der `wp_head`-Action-Hook ausgeführt, an den sich alle WordPress-Funktionen hängen, die Dinge im `head` ausgeben, beispielsweise Stylesheets oder das `title`-Element.
3. Nach dem schließenden `</head>`-Tag wird das `body`-Element geöffnet, und auch hier wird wieder innerhalb des Tags eine WordPress-Funktion aufgerufen, um ein Attribut auszugeben: Mit `body_class()` wird das `class`-Attribut generiert und mit nützlichen Klassen gefüllt, die beispielsweise Informationen über den angezeigten Inhaltstyp liefern und genutzt werden können, um CSS-Regeln auf einzelne Templates einzuschränken.
4. `wp_body_open()` triggert die seit WordPress 5.2 vorhandene Action `wp_body_open`, über die WordPress beispielsweise die Werkzeugleiste einbindet sowie verschiedene Inline Styles für Block-Effekte. Außerdem können Plugins damit einfach Inhalte direkt nach dem öffnenden `body`-Element einbauen, Sie sollten es in Ihren Themes also integrieren.
5. Nach dem Funktionsaufruf wird ein versteckter Link eingefügt, der später beim Fokus sichtbar wird und erlaubt, den Header

mit seinen Links direkt zu überspringen und zum Inhalt zu gelangen.

6. Es folgt mit dem `header`-Element der sichtbare Header, der Logo/Website-Titel und ein Menü beinhalten wird. Wie in der *index.php* binden wir mit `get_template_part()` zwei Template-Teile ein, um die *header.php* schlank zu halten. Danach wird der Header geschlossen und das `main`-Element geöffnet, damit das in den Template-Dateien nicht immer gemacht werden muss (das Element erhält die `id content`, die wir bei dem Sprunglink am Anfang von `body` als Ziel nutzen).

Schauen wir uns als Nächstes die beiden Template-Teile an, die Branding-Informationen und die Navigation ausgeben sollen. Wir starten mit der *template-parts/header/branding.php*, deren Inhalt in Listing 23.26 zu sehen ist.

```
<?php
$has_custom_logo   = has_custom_logo();
$site_description  = get_bloginfo( 'description', 'display' );
$shows_description = ! $has_custom_logo && ! empty( $site_description );
?>
<div class="branding <?php echo $shows_description ? 'with-description' : ''; ?>">
  <?php
  $title_tag = is_home() && is_front_page() ? 'h1' : 'p';
  if ( $has_custom_logo ) {
    $logo = get_custom_logo();
    echo "<$title_tag class='site-logo'>$logo</$title_tag>";
  }
  if ( ! $has_custom_logo ) {
    $site_title    = get_bloginfo( 'name', 'display' );
    $is_front_page = is_front_page();
    $site_url      = get_site_url();
    printf(
      '<%1$s class="site-title">%2$s%3$s%4$s</%1$s>',
      $title_tag,
      $is_front_page ? '' : "<a href='$site_url'>",
      $site_title,
      $is_front_page ? '' : '</a>'
    );
    if ( ! empty( $site_description ) ) {
      echo "<p class='site-description'>$site_description</p>";
    }
  }
  ?>
```

```
</div>
```

Listing 23.26 Die »branding.php« kümmert sich um die Ausgabe des Website-Logos oder des Titels und der Beschriftung.

1. Direkt am Anfang müssen ein paar Variablen definiert werden, um dem `div` mit der Klasse `branding` einen Hinweis mitzugeben, ob die Beschreibung angezeigt wird (nur aus Styling-Gesichtspunkten relevant). Zunächst wird mit `has_custom_logo()` ermittelt, ob im Customizer ein Logo festgelegt wurde. Anschließend speichern wir die in den WordPress-Einstellungen eingegebene Website-Beschreibung ab, die mit `get_bloginfo( 'description', 'display' )` aus der Datenbank ausgelesen wird. Danach wird geprüft, ob kein Logo vorhanden ist (in dem Fall werden Titel und Beschreibung angezeigt) und ob eine Beschreibung vorliegt. Ist das der Fall, bekommt das `div` zusätzlich zu `branding` noch die Klasse `with-description`.
2. Innerhalb des `div` wird festgelegt, in was für ein HTML-Element das Logo oder der Titel gewrappt werden soll. Wenn auf der Startseite die neuesten Beiträge angezeigt werden, soll es die `h1` der Seite werden, in allen anderen Fällen haben wir bereits vom Inhalt eine `h1` und umschließen es mit einem Absatz (`p`).
3. Anschließend wird geprüft, ob ein Logo festgelegt wurde, und in diesem Fall wird das Logo in dem entsprechenden Wrapper-Element ausgegeben (die Logo-Funktion im Customizer wird im nächsten Schritt aktiviert, nachdem diese beiden Header-Template-Teile besprochen wurden).
4. Wenn hingegen kein Logo gesetzt ist, wird der Website-Titel ausgegeben (über `get_bloginfo( 'name', 'display' )` wird er aus der Datenbank ausgelesen), und falls eine Beschreibung vorhanden ist, wird diese darunter als Absatz angezeigt. Für die

Ausgabe des Titels wird die Funktion `printf()` genutzt, die als ersten Parameter einen String mit Platzhaltern erwartet (im Beispiel `%1$s` etc.), die mit den Werten ersetzt werden, die als weitere Parameter übergeben werden.

5. Am Ende der Datei wird das `branding-div` wieder geschlossen.

Als Nächstes kommt die Datei *template-parts/header/nav.php* an die Reihe, in der die Header-Navigation ausgegeben wird (siehe Listing 23.27).

```
<?php
wp_nav_menu(
  [
    'theme_location'  => 'primary',
    'menu_class'      => 'primary-menu',
    'container'       => 'nav',
    'container_class' => 'primary-menu-container',
    'depth'           => 1,
  ]
);
```

Listing 23.27 Die Ausgabe des Header-Menüs geschieht in der »template-parts/branding/nav.php«.

Das ist doch mal übersichtlich. Mit dem Aufruf von `wp_nav_menu()` wird das Menü von der Theme-Position (`location`) `primary` ausgegeben (diese Position wird im nächsten Schritt definiert, sodass ihr im Backend unter Design • Menüs ein Menü zugewiesen werden kann), und die Klasse des Menüelements sowie ein Container-Element werden festgelegt. Außerdem wird mit `'depth' => 1` festgelegt, dass nur eine Menüebene ausgegeben wird. Wenn Sie jetzt die Webseite neu laden, sollten Sie den Titel und die Beschreibung (falls Sie sie nicht gelöscht haben) sehen, und auch die Werkzeugleiste sollte bereits vorhanden sein (siehe Abbildung 23.32).



Abbildung 23.32 Nicht schön, aber immerhin schon mal eine Ausgabe nach Anlegen der »header.php« mit den Template-Teilen für Branding-Inhalte und Navigation

Menüposition registrieren und Theme-Support für Titel, Logo und Feed-Links hinzufügen

Nachdem im Header bzw. in den Template-Teilen nun schon die Ausgabe von Logo und der `primary`-Menüposition geregelt ist, wird im nächsten Schritt die Menüposition registriert, und der Theme-Support für das Logo, das `title`-Element sowie Links zu den RSS-Feeds wird hinzugefügt. Dafür erstellen Sie die neue Datei *functions.php*, die WordPress automatisch lädt, und füllen sie mit dem Code aus dem folgenden Listing.

```
<?php
if ( ! function_exists( 'wh_classic_add_theme_support' ) ) {
  function wh_classic_add_theme_support() {
    add_theme_support(
      'custom-logo',
      [
        'unlink-homepage-logo' => true,
      ]
    );
    add_theme_support( 'title-tag' );
    add_theme_support( 'automatic-feed-links' );
  }
}
add_action( 'after_setup_theme', 'wh_classic_add_theme_support' );
if ( ! function_exists( 'wh_classic_add_logo_alt' ) ) {
  function wh_classic_add_logo_alt( array $attrs ) {
    $alt = $attrs['alt'] ?? '';
    if ( '' !== $alt ) {
      return $attrs;
    }
    $attrs['alt'] = get_bloginfo( 'name', 'display' );
```

```
    return $attrs;
  }
}
add_filter( 'get_custom_logo_image_attributes', 'wh_classic_add_logo_alt' );
if ( ! function_exists( 'wh_classic_register_nav_menu' ) ) {
  function wh_classic_register_nav_menu() {
    register_nav_menu(
      'primary',
      __( 'Header menu', 'wh-classic' )
    );
  }
}
add_action( 'after_setup_theme', 'wh_classic_register_nav_menu' );
```

Listing 23.28 Das Hinzufügen der Theme-Unterstützung für die Logo-Funktion, das »title«-Tag, die Anpassung des Logo-Markups und die Registrierung der Menüposition geschehen in der »functions.php«.

Info: Theme-Funktionen »pluggable« machen

Wenn Sie in Ihrem Theme die Funktionen aus der *functions.php* (oder anderen Dateien, die Sie in der *functions.php* laden) mit einer `function_exists()`-Prüfung umschließen, um zu testen, ob diese Funktion bereits definiert wurde, können Sie Child-Themes damit erlauben, diese Funktionen zu überschreiben, da die *functions.php* des Child-Themes direkt vor der des Parent-Themes geladen wird. So machen Sie die Funktion *pluggable*, also austauschbar. Diesen Weg werden wir für das Beispiel-Theme gehen, es werden also alle Funktionen mit einem `function_exists()`-Aufruf umschlossen. Eine andere und vielleicht etwas elegantere Variante wäre, in jeder Funktion die genutzten Attribute oder die Ausgabe über einen Filter veränderbar zu machen, der Einfachheit halber gehen wir hier den anderen Weg.

Für den Start fügen Sie drei Funktionen in die *functions.php* ein, die alle mit `function_exists()`-Prüfungen umschlossen sind, sodass die Funktionen von einem Child-Theme überschrieben werden könnten (siehe Kasten »Theme-Funktionen ›pluggable‹ machen«).

1. In der Theme-Funktion `wh_classic_add_theme_support()` werden zwei `add_theme_support()`-Aufrufe eingefügt. Diese Funktion erlaubt es, WordPress-Funktionen für Themes zu aktivieren, und neben den in diesem Theme genutzten gibt es noch ein paar weitere (*https://developer.wordpress.org/reference%2Ffunctions%2Fadd_theme_support%2F/*).

   `add_theme_support( 'custom-logo' );` aktiviert die Website-Logo-Funktion im Customizer. Über den optionalen zweiten Parameter geben wir an, dass das Logo auf der Startseite nicht als Link ausgegeben werden soll, da es in dem Fall auf die Seite verlinkt, auf der sich die Besucher*innen bereits befinden. Mit `add_theme_support ( 'title-tag' );` geben Sie an, dass WordPress sich um die Erstellung des `title`-Elements im `head` kümmern soll, damit im Browser-Tab statt der URL der Titel der aktuell angezeigten Seite erscheint und eben nicht die URL (abgesehen vom Tab wird das noch für weitere Situationen genutzt, zum Beispiel beim Setzen eines Lesezeichens, beim Teilen des Beitrags über Social Media oder in Suchmaschinen-Ergebnislisten). Der Aufruf von `add_theme_support( 'automatic-feed-links' );` fügt dem `head`-Element Links zu den Feeds hinzu, die WordPress erstellt. Das ist immer ein Link zum Feed der neuesten Beiträge sowie zu einem Kommentar-Feed. Abhängig von der gerade aufgerufenen Ansicht sind weitere Feeds enthalten, auf Archivseiten wird beispielsweise noch ein Feed verlinkt, der die Beiträge des entsprechenden Archivs enthält.

   Um WordPress über die `wh_classic_add_theme_support()`-Funktion zu informieren, wird sie an den `after_setup_theme`-Action-Hook angehängt.

2. Wenn für das Logo die Option gesetzt ist, dass es auf der Startseite nicht verlinkt sein soll, wird dort das `alt`-Attribut des Logos leer gelassen, das normalerweise den Titel der Website enthält. Das ist aus Gründen der Barrierefreiheit nicht gerade optimal, und wir hätten dann auf der Startseite eine leere `h1` (der `alt`-Text wird auch als Teil des normalen Textinhalts für das umschließende Element angesehen), also muss dieses Verhalten umgangen werden. Dazu wird der Filter `get_custom_logo_image_attributes` verwendet, mit dem die Attribute des Logos verändert werden können, bevor es ausgegeben wird. Die an den Filter gehängte Funktion `wh_classic_add_logo_alt()` bekommt als Parameter das Array der Attribute, in dem unter anderem der Schlüssel alt mit dem Wert für das `alt`-Attribut existieren sollte. Wir gehen aber lieber auf Nummer sicher und prüfen bei der Zuweisung des Werts an eine Variable, ob der Schlüssel vorhanden ist, ansonsten speichern wir einen leeren String. Wenn die Variable `$alt` etwas anderes als einen leeren String enthält, wird das Array unverändert zurückgegeben, da ein Alternativtext vorhanden ist. Ansonsten weisen wir dem Array-Eintrag für den `alt`-Text (`$attrs['alt']`) den Website-Titel zu und geben das veränderte Array zurück.
3. Die Funktion `wh_classic_register_nav_menu()` kümmert sich um die Registrierung der Menüposition `primary` und ruft dafür die WordPress-Funktion `register_nav_menu()` auf, die als ersten Parameter den eindeutigen Kennzeichner für die Position erwartet (das ist der Wert, der in der *template-parts/header/nav.php* beim Aufruf von `wp_nav_menu()` für die Option `theme_location` angegeben wurde) und als zweiten die im Backend sichtbare Beschriftung, die für die Menüposition

angezeigt wird. Auch diese Funktion wird an den `after_setup_theme`-Hook gehängt.

Wenn Sie nun die Seite neu laden, sollten Sie sehen, dass im Tab nicht mehr die URL angezeigt wird, sondern der Website-Titel und die Beschreibung (wenn Sie sich auf der Startseite befinden) oder der Titel des aufgerufenen Inhalts. Öffnen Sie nun den Customizer (über Design • Customizer oder aus dem Frontend über den Customizer-Link in der Werkzeugleiste) und dort den Bereich Website-Informationen, sollten Sie die Option Logo vorfinden, mit der Sie Ihr Site-Logo festlegen können (siehe Abbildung 23.33).

Und ebenfalls im Customizer oder über Design • Menüs können Sie jetzt ein Menü anlegen und der neuen Menüposition zuweisen (wie Sie mit Menüs arbeiten, wurde in Abschnitt 10.4.2, »Navigationsmenü erstellen«, behandelt), wie in Abbildung 23.34 zu sehen.

Von den Funktionen ist der Header damit fertiggestellt (ein bisschen Styling fehlt, aber das kommt ganz am Schluss und wird auch nicht detailliert in diesem Buch behandelt, da es den Rahmen sprengen würde).

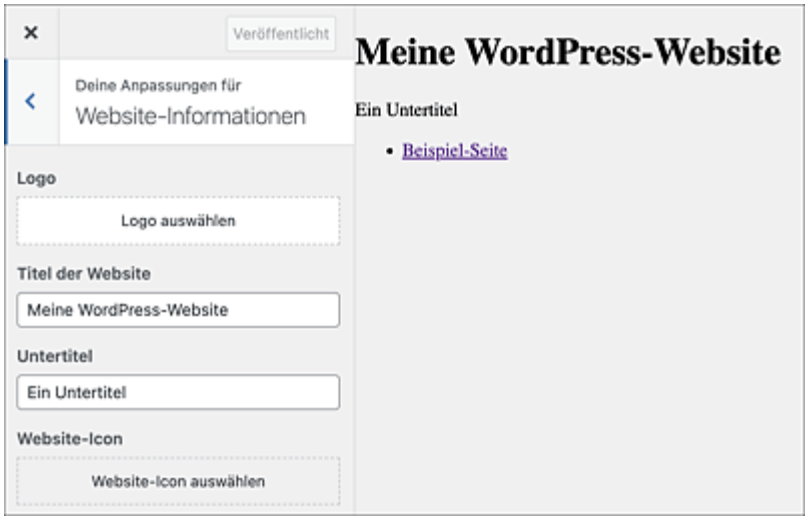


Abbildung 23.33 Nach der Aktivierung des Theme-Supports für die Logo-Funktion ist die Option dafür im Customizer zu finden.

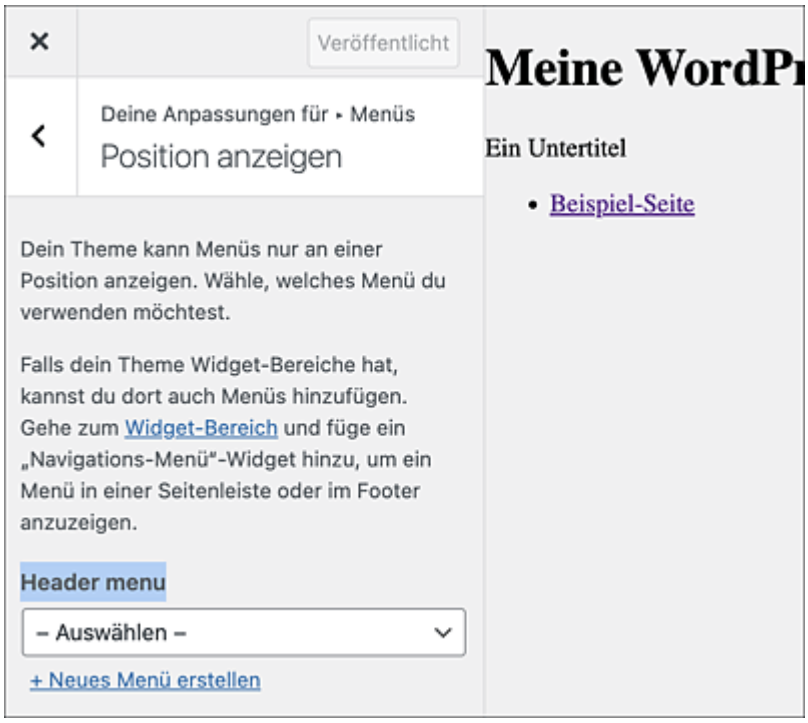


Abbildung 23.34 Die registrierte Menüposition ist in der Menüverwaltung auswählbar.

Das Ende des HTML-Dokuments: die footer.php

Nach dem Header kommt erst mal der Footer dran, damit das HTML-Gerüst vollständig wird und weil WordPress eine ähnliche Funktion wie `wp_head()` bereitstellt, über die Scripts und andere Dinge am Ende einer Webseite eingefügt werden können. Erstellen Sie dafür eine *footer.php* mit dem Code aus Listing 23.29.

```
      </main>
    <?php wp_footer(); ?>
  </body>
</html>
```

Listing 23.29 In der »footer.php« wird die Ausgabe des Website-Endes geregelt.

Am Anfang der Datei wird zunächst das `main`-Element geschlossen, das am Ende der *header.php* geöffnet wurde. Danach folgt der

Aufruf von `wp_footer()`, dem Äquivalent von `wp_head()` für den Bereich am Ende des HTML-Quelltextes. Den Abschluss der Datei und damit auch der Seite bilden das schließende `body`- und das schließende `html`-Element, die auch jeweils in der *header.php* geöffnet wurden.

Die Template-Teile für die Blogübersicht

Nachdem Header und Footer (dieser wird nachher noch erweitert, aber das Wichtigste ist erst mal da) so weit stehen, wenden Sie sich jetzt den Template-Teilen zu, die in der *index.php* geladen werden, sodass danach die Startseite neben dem Header auch die Inhalte der neuesten Beiträge anzeigt.

Starten wir mit der *template-parts/content/content.php*, in der die Inhalte eines Beitrags für eine Übersicht, wie sie z. B. auf Archivseiten, Kategorieseiten oder anderen Beitragsübersichten verwendet werden, ausgegeben werden (siehe Listing 23.30).

```
<article <?php post_class(); ?>>
  <header class="entry-header">
    <?php
    $permalink = get_the_permalink();
    the_title( "<h2 class='entry-title'><a href='$permalink'>", '</a></h2>' );
    printf(
      '<p class="entry-date"><a href="%s">%s</a></p>',
      $permalink,
      get_the_date()
    );
    if ( ! is_search() && has_post_thumbnail() ) {
      printf(
        '<figure class="post-thumbnail">%s</figure>',
        get_the_post_thumbnail( null, 'large' )
      );
    }
    ?>
  </header>
  <div class="entry-content">
    <?php
    if ( has_excerpt() && ! is_singular() ) {
      printf(
        '<p class="excerpt">%s</p>',
```

```
        get_the_excerpt()
      );
    } else {
      wh_classic_the_content();
    }
    ?>
  </div>
</article>
```

Listing 23.30 Die »content.php«-Datei im »template-parts/content«-Ordner kümmert sich um die Ausgabe von Beiträgen für Beitragsübersichten.

1. Im öffnenden `article`-Tag setzen Sie mit `post_class()` wieder eine Funktion ein, die einen `class`-String mit hilfreichen Klassen erzeugt, ähnlich wie `body_class()`.
2. Innerhalb des `header`-Elements wird der Permalink zum Beitrag ermittelt und danach der verlinkte Titel als `h2` ausgegeben (die optionalen Parameter für die Funktion `the_title()` ermöglichen die Angabe von Inhalten und/oder Markup, die vor und hinter dem Titel ausgegeben werden sollen). Nach dem Titel wird innerhalb eines Absatzes das ebenfalls auf den Beitrag verlinkende Datum ausgegeben. So wird sichergestellt, dass auch für Beiträge ohne Titel ein Link auf die Detailseite vorhanden ist, denn ein Veröffentlichungsdatum hat ein Beitrag immer (normalerweise auch einen Titel, aber man weiß ja nie). Nach dem Datum wird geprüft, ob es sich nicht um eine Suchergebnisseite handelt (`! is_search()`) und ob ein Beitragsbild vorhanden ist (`has_post_thumbnail()`). Wenn beide Bedingungen zutreffen, wird innerhalb eines `figure`-Elements mit der Funktion `get_the_post_thumbnail()` das Beitragsbild ausgegeben. Der erste Parameter an die Funktion kann optional eine Post-ID oder ein Post-Objekt sein; falls kein Parameter übergeben wird (oder wie hier `null`), wird das Beitragsbild zum aktuellen Post gesucht. Der zweite Parameter gibt die Bildgröße an, die ausgegeben werden soll, hier entscheiden Sie sich mit

large für die große Variante, damit das Bild so breit dargestellt werden kann wie der Inhalt.

3. Nach dem `header` wird in einem `div` der Beitragsinhalt ausgegeben. Zunächst wird geprüft, ob der Beitrag einen explizit festgelegten Auszug hat (`has_excerpt()`) und ob wir uns gerade nicht auf einer Einzelansicht befinden (`! is_singular()`). Ist das der Fall, wird der Auszug ausgegeben. Andernfalls rufen wir die gleich noch zu erstellende Funktion `wh_classic_the_content()` auf, die den Beitragsinhalt ausgibt, entweder bis zum Mehr-Block oder den gesamten Inhalt, wenn es keinen Mehr-Block gibt oder die Einzelansicht eines Beitrags angezeigt wird.

Die Funktion `wh_classic_the_content()` wird, vielleicht haben Sie es schon geahnt, in die *functions.php* geschrieben. Fügen Sie also den Code aus Listing 23.31 am Ende der *functions.php* ein.

```
if ( ! function_exists( 'wh_classic_the_content' ) ) {
  function wh_classic_the_content() {
    the_content(
      sprintf( /* translators: visible text for the more tag */
        '<span aria-hidden="true">%1s</span><span class="screen-reader-
text">%2s</span>',
        __( 'Continue reading', 'wh-classic' ),
        sprintf( /* translators: continue reading text for screen reader
 users. s=post title */
          __( 'Continue reading %s', 'wh-classic' ),
          the_title( '', '', false )
        )
      )
    );
  }
}
```

Listing 23.31 Die Theme-eigene Funktion zur Ausgabe des Beitragsinhalts kümmert sich um einen zugänglicheren Weiterlesen-Link.

Das sieht im ersten Moment aufgrund der vielen Verschachtelungen etwas komplex aus, aber eigentlich ist es recht einfach. Sichtbar auf der Website soll der String »Continue reading«

ausgegeben werden. Dieser Linktext allein würde aber ein Problem bei der Barrierefreiheit darstellen, da dann die Linkliste der Website viele »Continue reading«-Links ohne weitere Informationen enthalten würde, was für Nutzer*innen von Screenreadern wenig hilfreich ist. Deshalb verstecken wir diesen für sehende Personen sichtbaren String vor Screenreadern (mit `aria-hidden="true"`) und geben direkt dahinter einen visuell versteckten String aus (mit der Klasse `screen-reader-text`), der aber Screenreader-Benutzer*innen vorgelesen wird. Dieser übersetzbare String enthält nach dem »Continue reading« einen Platzhalter für den Titel des Beitrags, sodass es zum Beispiel heißt: »Continue reading Hello World!«. Wir benötigen diese Funktion im finalen Theme zwar nur an einer Stelle, aber falls Sie das Ganze als Ausgangspunkt für weitere Arbeiten nutzen möchten und vielleicht weitere Template-Teile für Inhaltsteaser anlegen, können Sie es da einfach wiederverwenden.

Mit diesen Änderungen sollten Sie jetzt auf der Startseite bereits die letzten Beiträge angezeigt bekommen (sofern Sie die Einstellung für die Inhalte der Startseite nicht auf eine statische Seite geändert haben). Wie die Startseite nun ungefähr aussehen sollte, ist in Abbildung 23.35 zu sehen (die Liste zwischen Website-Beschreibung und der Überschrift des ersten Beitrags ist ein Menü in unserer `primary`-Menüposition).



Abbildung 23.35 Das sieht doch gut aus (vom fehlenden Design abgesehen): Jetzt werden schon die Beiträge in der Blogübersicht angezeigt.

Der zweite Template-Teil, der in der *index.php* eingebunden wird, ist die *template-parts/content/content-none.php* für den Fall, dass eine existierende Liste von Blogbeiträgen aufgerufen wird, in der aber kein Blogbeitrag vorhanden ist (eigentlich sollte das nie vorkommen, es sei denn, Sie haben eine Blogseite eingerichtet, aber keinen einzigen Beitrag geschrieben). Der Inhalt der Datei ist in Listing 23.32 abgedruckt.

```
<article class="hentry">
  <header class="entry-header">
    <h1 class="entry-title">
      <?php _e( 'No posts yet.', 'wh-classic' ); ?>
    </h1>
  </header>
  <div class="entry-content">
    <p>
      <?php _e( 'We do not have posts here yet, sorry.', 'wh-classic' ); ?>
    </p>
  </div>
</article>
```

Listing 23.32 Dieser Template-Teil wird nur ausgegeben, wenn für eine existierende Liste von Beiträgen kein Beitrag vorhanden ist, also wahrscheinlich nie.

Auch wenn hier kein Beitragsteaser angezeigt wird, stellen wir das HTML-Markup nach, das für die normalen Teaser in der Beitragsliste ausgegeben wird, damit beim Styling mit CSS kein Sonderfall beachtet werden muss, sondern die Regeln auch einfach hier greifen. Titel und Inhalt des »Beitrags« weisen die Besucher*innen darauf hin, dass es hier nichts zu sehen gibt.

Bevor wir uns der Einzelansicht von Beiträgen zuwenden, passen wir noch ein (jedenfalls in diesem Theme) unerwünschtes Verhalten an: Standardmäßig werden die Weiterlesen-Links mit einer Sprungmarke ausgestattet, durch die in der Einzelansicht direkt zu dem Punkt gesprungen wird, an dem der Weiterlesen-Block eingefügt wird. Dafür müssen wir einen Filter nutzen, und deshalb

wechseln wir in die *functions.php* und fügen den Code aus Listing 23.33 am Ende ein.

```
if ( ! function_exists( 'wh_classic_remove_more_link_scroll' ) ) {
  function wh_classic_remove_more_link_scroll( $link ) {
    $link = preg_replace( '/#more-[0-9]+/', '', $link)
    return "<p class='read-more'>$link</p>";
  }
}
add_filter( 'the_content_more_link', 'wh_classic_remove_more_link_scroll' );
```

Listing 23.33 Wenn Besucher*innen auf einen Weiterlesen-Link klicken, sollen sie in der Einzelansicht ganz normal am Anfang der Webseite landen und nicht direkt zu der Position des Weiterlesen-Blocks springen.

Die Funktion `wh_classic_remove_more_link_scroll()` wird an den Filter `the_content_more_link` gehängt und erhält dadurch als Parameter das Markup des Weiterlesen-Links. Der Sprunglink zum Weiterlesen-Block ist nach dem Muster `#more-$post_id` aufgebaut, den wir über einen regulären Ausdruck finden und durch einen leeren String ersetzen. Danach gibt die Funktion den Weiterlesen-Link innerhalb eines Absatzes zurück.

Damit hätten wir auch das Index-Template und die zugehörigen Template-Teile und -Funktionen abgeschlossen. Im nächsten Schritt wird die Beitragsbild-Funktion aktiviert, und danach werden die Templates für die Einzelansicht von Beiträgen erstellt sowie für andere Inhaltstypen, die keine Seiten sind.

Theme-Support für Beitragsbilder

Mit der im vorherigen Abschnitt genutzten `get_the_post_thumbnail()`-Funktion geben Sie das Beitragsbild eines Inhalts aus, allerdings muss bei Classic Themes die Unterstützung dafür explizit aktiviert werden. Dafür erweitern Sie die Funktion `wh_classic_add_theme_support()` in der *functions.php* um die Zeile aus dem folgenden Listing.

```
add_theme_support( 'post-thumbnails')
```

Jetzt sollten Sie im Editor beispielsweise für einen Beitrag in der Editor-Seitenleiste ein Beitragsbild festlegen können (siehe Abbildung 23.36).

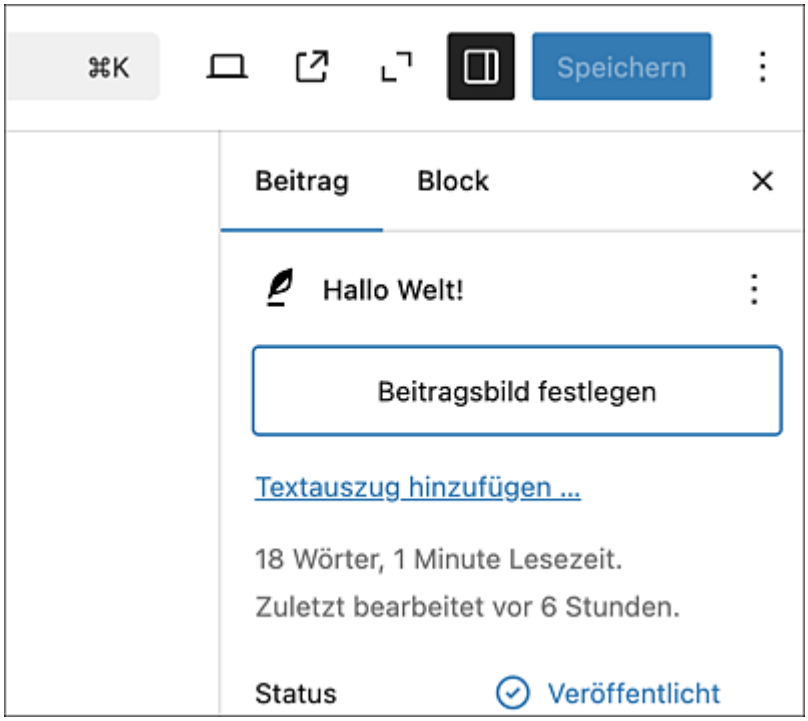


Abbildung 23.36 Wurde die Unterstützung aktiviert, ist die Beitragsbildfunktion im Editor zu sehen.

Die Einzelansicht von Beiträgen und anderen Nicht-Seiten-Inhaltstypen: singular.php

Das *singular.php*-Template ist das Fallback-Template (vor der *index.php*) für die Einzelansicht aller Inhaltstypen, egal ob Beiträge, Seiten oder eigene Inhaltstypen. Da sich die Ansicht der Seiten etwas davon unterscheiden wird, erstellen wir dafür ein eigenes Template, sodass sich die *singular.php* in unserem Theme in einer Standard-WordPress-Installation ohne weitere Inhaltstypen erst mal nur um die Darstellung der Detailansicht von Beiträgen kümmert. Der Code der Datei ist in Listing 23.34 zu sehen.

```php
<?php
get_header();

while ( have_posts() ) {
  the_post();
  get_template_part(
    'template-parts/content/content-single',
    get_post_type()
  );
}
comments_template();
get_footer();
```

Listing 23.34 Die »singular.php« ist in unserem Theme für die Einzelansicht von Beiträgen zuständig sowie für eventuell weitere hinzugefügte Inhaltstypen.

1. Am Anfang der Template-Datei wird mit `get_header()` die *header.php* eingebunden, wie Sie es schon von der *index.php* kennen.
2. In diesem Einzel-Template braucht es vor dem Loop keine Prüfung, ob Beiträge vorhanden sind, da dieses Template nur genutzt wird, wenn der Eintrag des Inhaltstyps (beispielsweise ein Beitrag) existiert. In dem Loop laden wir den Template-Teil *template-parts/content/content-single.php* mit optionaler Erweiterung durch den Inhaltstyp-Slug.
3. Nach dem Loop wird über `comments_template()` die *comments.php* des Themes geladen, in der die Ausgabe des Kommentarformulars und der Liste der Kommentare vorgenommen wird.

Werfen wir nun einen Blick auf die *template-parts/content/content-single.php*, die den Inhalt für die Einzelansicht eines Beitrags ausgibt (siehe Listing 23.35).

```php
<article <?php post_class(); ?>>
  <header class="entry-header">
    <?php
    the_title( '<h1 class="entry-title">', '</h1>')
    printf(
      '<p class="entry-date">%s</p>',
      get_the_date()
```

```
    );
    if ( has_post_thumbnail() ) {
      printf(
        '<figure class="post-thumbnail">%s</figure>',
        get_the_post_thumbnail( null, 'large' )
      );
    }
    ?>
  </header>
  <div class="entry-content">
    <?php the_content(); ?>
  </div>
  <footer class="entry-footer">
    <p class="post-categories">
      <?php
      _e( 'Categories: ', 'wh-classic' );
      echo get_the_category_list( ', ' );
      ?>
    </p>
  </footer>
</article>
```

Listing 23.35 Die »template-parts/content/content-single.php« kümmert sich um die Anzeige der Inhalte für die Einzelansicht eines Beitrags.

Dieser Template-Teil ähnelt der *template-parts/content/content.php*, deshalb werden hier nur die Unterschiede behandelt. Der Titel des Beitrags wird hier als `h1` ausgegeben und nicht verlinkt (das Datum auch nicht), da die Besucher*innen sich ja bereits auf der Einzelansicht befinden. Für die Anzeige des Inhalts wird direkt die Funktion `the_content()` genutzt, da es hier keinen Weiterlesen-Link gibt. In der Einzelansicht des Beitrags wird in einem Footer die Liste der Kategorien angezeigt, denen der Beitrag zugeordnet ist, wofür die Funktion `get_the_category_list()` verwendet wird, um eine kommagetrennte Liste der Beitragskategorien zu erhalten.

Jetzt fehlt für die Einzelansicht von Beiträgen noch die *comments.php*, denn unabhängig davon, ob Sie später die Kommentarfunktion aktiviert lassen oder deaktivieren, sollte ein Theme die Funktion auf jeden Fall bereitstellen. Den Code für diese Datei finden Sie in Listing 23.36.

```
<?php
if ( ! comments_open() && ! get_comments_number() ) {
  return;
}
if ( post_password_required() ) {
  return;
}
?>

<aside
  aria-label="
  <?php
  esc_html_e(
    'Area for comments and other reactions',
    'wh-classic'
  );
  ?>
  "
  class="comments-area"
>

  <?php
  if ( have_comments() ) {
    $comments_heading = sprintf(
      _n( /* translators: 1=post/page title, 2=number of comments */
        'One reaction to »%1$s«',
        '%2$s reactions to »%1$s«',
        get_comments_number(),
        'wh-classic'
      ),
      get_the_title(),
      get_comments_number()
    );
    echo "<h2 class='comments-area-heading'>$comments_heading</h2>";
    $comments = wp_list_comments(
      [
        'echo'     => false,
        'style'    => 'ul',
        'type'     => 'comment',
        'per_page' => ( isset( $comment_args['number'] ) ?
 $comment_args['number'] : - 1 ),
      ]
    );

    if ( ! empty( $comments ) ) {
      echo "<ul class='reactions-list comments'>$comments</ul>";
    }

    $pings = wp_list_comments(
      [
        'style'    => 'ul',
        'type'     => 'pings',
        'per_page' => ( isset( $comment_args['number'] ) ?
 $comment_args['number'] : - 1 ),
```

```
      ]
    );

    if ( ! empty( $pings ) ) {
      echo "<ul class='reactions-list pings'>$pings</ul>";
    }

    the_comments_navigation();

    if ( ! comments_open() && get_comments_number() ) {
      ?>
      <p class="nocomments">
        <?php
        esc_html_e( 'Comments are closed.', 'wh-classic' );
        ?>
      </p>
      <?php
    }
  }

  comment_form( [ 'label_submit' => __( 'Submit Comment', 'wh-classic' ) ] );
  ?>
</aside>
```

Listing 23.36 In der »comments.php« werden die Kommentare und andere Reaktionen auf einen Beitrag ausgegeben, und das Kommentarformular wird angezeigt.

Ja, das ist ein bisschen umfangreicher, aber bitte bleiben Sie mir erhalten und laufen Sie nicht weg, wir gehen das gemeinsam durch:

1. Am Anfang der Datei wird geprüft, ob die Reaktionen und das Formular überhaupt angezeigt werden müssen. Wenn Kommentare geschlossen sind und vorher keine Kommentare geschrieben wurden, können wir die Abarbeitung der Datei direkt beenden (`return`). Gleiches gilt, wenn der Beitrag mit einem Passwort geschützt ist und das richtige Passwort noch nicht eingegeben wurde.
2. Innerhalb des `aside`-Elements wird mit `have_comments()` geprüft, ob Reaktionen für den Beitrag vorliegen. Ist das der Fall, wird die Überschrift für die Reaktionen zusammengebaut, die am Anfang die Anzahl der Reaktionen beinhalten soll und am Ende den Titel, wobei Singular und Plural unterschieden werden.

Dafür wird die Übersetzungsfunktion `_n()` genutzt, der zuerst der Singular- gefolgt vom Plural-String übergeben wird. Der dritte Parameter ist die Zahl, anhand deren über Singular oder Plural entschieden wird, und am Ende kommt wie gewohnt die Text-Domain. Das Ergebnis der Übersetzungsfunktion wird mit `sprintf()` von den Platzhaltern befreit, die durch den Titel des Beitrags und, im Fall des Plural-Strings, durch die Anzahl der Reaktionen ersetzt werden. Danach wird die Überschrift als `h2` ausgegeben.

3. In `$comments` wird das Ergebnis von `wp_list_comments()` gespeichert, der wir über das Parameter-Array mitteilen, dass es die Liste der Reaktionen nicht direkt ausgeben soll (`'echo' => false`), dass wir eine `ul`-Liste haben möchten und dass nur die Reaktionen vom Typ `comment` Teil der Liste sein sollen (also über das Kommentarformular abgeschickte Kommentare, keine Ping-/Trackbacks oder andere Reaktionen, die über Plugins ermöglicht werden, wie etwa Webmentions). Außerdem sollen so viele Kommentare pro Seite angezeigt werden, wie in den Einstellungen von WordPress festgelegt wurde. Wenn `$comments` nicht leer ist, also Kommentare als Liste zurückgegeben wurden, wird das Ergebnis mit einem `ul`-Element umschlossen (das kommt von `wp_list_comment()` nämlich nicht zurück) und ausgegeben.
4. Im nächsten Schritt passiert dasselbe für `pings`, also Pingbacks und Trackbacks (Benachrichtigungen von anderen Websites, wenn diese auf Sie verlinken).
5. Nach der Pings-Liste werden mit `the_comments_navigation()` Links zur nächsten/vorherigen Seite von Reaktionen angezeigt, wenn die Reaktionen in mehrere Seiten umbrochen sind.

6. Wenn Kommentare geschlossen sind und Kommentare existieren (eigentlich ist die zweite Prüfung überflüssig, da ganz am Anfang schon sichergestellt wird, dass nicht beides unwahr ist, aber sicher ist sicher), wird ein Hinweis angezeigt, dass die Kommentarfunktion geschlossen ist.
7. Am Schluss wird mit `comment_form()` das Kommentarformular angezeigt (wenn die Kommentarfunktion deaktiviert ist, wird es automatisch ausgeblendet), über den Parameter wird der Text des Absendebuttons angepasst.

Wenn Sie jetzt die Einzelansicht eines Beitrags aufrufen, sollte die *singular.php* greifen mit dem Template-Teil und der *comments.php*.



Abbildung 23.37 Nach Erstellen der »singular.php«, des »content-single.php«-Template-Teils und der »comments.php« sieht die Einzelansicht des »Hallo-Welt!«-Beitrags ungefähr so aus.

Jetzt kann das Theme schon die Blogübersicht und Einzelansichten von Beiträgen richtig darstellen (zumindest was den reinen Inhalt angeht). Als nächstes Template schauen wir uns das für die Seiten an.

Darstellung von Seiten mit der page.php

Die Einzelansicht von Seiten ist ein bisschen anders als die von Beiträgen, so sollen das Veröffentlichungsdatum und der Footer mit den Kategorien nicht angezeigt werden (die man für Seiten nicht vergeben kann). Der Code der *page.php* ist in Listing 23.37 zu sehen.

```
<?php
get_header();
while ( have_posts() ) {
  the_post();
  get_template_part(
    'template-parts/content/content',
    'page'
  );
}
comments_template();
get_footer();
```

Listing 23.37 Die »page.php« sieht der »singular.php« sehr ähnlich, der einzige Unterschied besteht in dem Template-Teil, der eingebunden wird.

Der Unterschied zur *singular.php* liegt in dem Template-Teil, der eingebunden wird: Die *template-parts/content/content-page.php* kümmert sich um die Ausgabe, ihren Inhalt sehen Sie in Listing 23.38.

```
<article <?php post_class(); ?>>
  <header class="entry-header">
    <?php
    if ( is_search() ) {
      $permalink = get_the_permalink();
      the_title( "<h2 class='entry-title'><a href='$permalink'>", '</a>
</h2>' );
    } else {
      the_title( '<h1 class="entry-title">', '</h1>' );
      if ( has_post_thumbnail() ) {
        printf(
```

```
            '<figure class="post-thumbnail">%s</figure>',
            get_the_post_thumbnail( null, 'large' )
          );
        }
      }
      ?>
    </header>
    <div class="entry-content">
      <?php
      if ( is_search() ) {
        the_excerpt();
      } else {
        the_content();
      }
      ?>
    </div>
</article>
```

Listing 23.38 Die Ausgabe des Seiteninhalts wird von der »template-parts/content/content-page.php« übernommen.

Der Template-Teil ähnelt dem für Beiträge, jedoch mit dem Unterschied, dass hier das Datum und der Footer nicht ausgegeben werden. Zudem muss bei Titel und Inhalt die Unterscheidung gemacht werden, ob die Datei für die Einzelansicht einer Seite oder für die Suchergebnisseite genutzt wird (da wird sie im nächsten Abschnitt eingebunden). Wenn sie für Suchergebnisse bestimmt ist (`is_search()`), wird der Titel als `h2` ausgegeben und verlinkt auf die Einzelansicht, und statt `the_content()` wird `the_excerpt()` aufgerufen, um die ersten 55 Wörter des Inhalts anzuzeigen statt des ganzen Inhalts. Wenn die Datei für die Einzelansicht genutzt wird, ist der Titel eine unverlinkte `h1`, und der Inhalt wird mit `the_content()` komplett ausgegeben.

Suchergebnisseite und Archivübersicht: search.php und archive.php

Wenn auf der Website ein Suchformular eingebaut wird, beispielsweise in einem Widget-Bereich oder nachher im 404-

Template, wird für die Suchergebnisse nach einer *search.php* gesucht. Legen Sie diese Datei an und füllen Sie sie mit dem Code aus Listing 23.39.

```
<?php
get_header();
?>
<header class="archive-header">
  <h1 class="archive-title">
    <?php
    printf( /* translators: s=search query. */
      esc_html__( 'Search Results for: %s', 'wh-classic' ),
      get_search_query()
    );
    ?>
  </h1>
</header>
<?php
if ( have_posts() ) {
  while ( have_posts() ) {
    the_post();
    get_template_part(
      'template-parts/content/content',
      get_post_type()
    );
  }
} else {
  get_template_part(
    'template-parts/content/content',
    'no-search-result'
  );
}

the_posts_pagination(
  [
    'type'      => 'list',
    'prev_text' => __( 'Previous', 'wh-classic' ),
    'next_text' => __( 'Next', 'wh-classic' ),
  ]
);

get_footer();
```

Listing 23.39 In der »search.php« wird die Ergebnisliste für eine Suche aufbereitet.

Das Template sieht der *index.php* recht ähnlich, es gibt aber ein paar wichtige Unterschiede:

1. Nach dem Einbinden des Headers (`get_header()`) wird ein `header`-Element mit dem Titel für die folgende Beitragsliste ausgegeben. In die Überschrift wird mittels `printf()` und `get_search_query()` der gesuchte String eingebaut.
2. Danach folgt der Ihnen bekannte Loop, und für jeden gefundenen Beitrag wird die *template-parts/content/content.php* eingebunden (oder, wenn vorhanden, eine für Inhaltstypen spezifischere Datei).
3. Wenn keine Beiträge gefunden wurden, die zu der Suche passen, wird der Template-Teil *template-parts/content/content-no-search-result.php* genutzt, um darüber zu informieren.
4. Es folgt die Paginierung und die Einbindung des Footers.

Der Template-Teil, der für gefundene Beiträge genutzt wird, ist bereits angelegt, den Teil für die Nachricht, dass keine Ergebnisse gefunden wurden, erstellen wir jetzt mit der Datei *template-parts/content/content-no-search-result.php*, deren Inhalt in Listing 23.40 abgedruckt ist.

```
<article class="hentry">
  <div class="entry-content">
    <p>
      <?php _e( 'Nothing found for your search.', 'wh-classic' ); ?>
    </p>
    <?php get_search_form(); ?>
  </div>
</article>
```

Listing 23.40 Hier wird kurz und knapp mitgeteilt, dass nichts gefunden werden konnte.

Der Inhalt ähnelt grob der *content-none.php*, nur dass keine Überschrift mit ausgegeben wird und dass zusätzlich zu dem Hinweis, dass nichts gefunden wurde, noch ein Suchformular erscheint, damit direkt die nächste Suche gestartet werden kann.

Die *archive.php* ist, wie im Abschnitt zur Template-Hierarchie beleuchtet, das letzte Fallback-Template vor der *index.php* für die Anzeige von Beitragsarchiven, etwa Kategorie- oder Datum-Archiven:

```
<?php
get_header();
?>
<header class="archive-header">
  <h1 class="archive-title">
    <?php the_archive_title(); ?>
  </h1>
  <?php
  $archive_description = get_the_archive_description();
  if ( ! empty( $archive_description ) ) {
    echo "<p class='archive-description'>$archive_description</p>";
  }
  ?>
</header>
<?php
if ( have_posts() ) {
  while ( have_posts() ) {
    the_post();
    get_template_part(
      'template-parts/content/content',
      get_post_type()
    );
  }
} else {
  get_template_part(
    'template-parts/content/content',
    'none'
  );
}

the_posts_pagination(
  [
    'type'      => 'list',
    'prev_text' => __( 'Previous', 'wh-classic' ),
    'next_text' => __( 'Next', 'wh-classic' ),
  ]
);

get_footer();
```

Listing 23.41 Die »archive.php« wird genutzt, um Beitragsarchive darzustellen, zum Beispiel eine Liste der Beiträge aus einer Kategorie.

Der untere Teil der Datei ist identisch mit der *index.php*. Oben werden nach dem Laden des Headers der Titel des Archivs sowie, falls vorhanden, die Beschreibung ausgegeben. Der Titel ist beispielsweise bei einem Kategorie-Archiv »Kategorie: $kategoriename«, und die Beschreibung ist der Beschreibungstext, der im Backend optional für Kategorien angegeben werden kann (es gibt nicht für alle Archivarten Beschreibungen, für die Datum-Archive kann zum Beispiel keine angegeben werden).

Inhalt nicht gefunden: die 404.php

Wenn ein aufgerufener Inhalt nicht (mehr) existiert, wird ein 404-Fehler geworfen, und in Classic Themes wird das *404.php*-Template geladen.

```
<?php get_header(); ?>
<article <?php post_class( 'hentry' ); ?>>
  <header class="entry-header">
    <h1 class="entry-title">
      <?php _e( 'Nothing found.', 'wh-classic' ); ?>
    </h1>
  </header>
  <div class="entry-content">
    <p>
      <?php _e( 'The content you are looking for does not exist, sorry. Maybe a
search helps?', 'wh-classic' ); ?>
    </p>
    <?php get_search_form(); ?>
  </div>
</article>
<?php
get_footer();
```

Listing 23.42 Die »404.php« kommt zum Einsatz, wenn ein Inhalt nicht gefunden wurde.

Nach dem Einbinden des Headers wird quasi das Markup des Template-Teils nachgestellt, der für die Darstellung des Seiteninhalts zuständig ist, und den Usern wird darin mitgeteilt, dass nichts gefunden werden konnte. Außerdem wird mit

`get_search_form()` das Suchformular ausgegeben (hier wird standardmäßig der Suchstring als Wert vorausgefüllt).

Widget-Bereich in den Footer einbauen

Damit die Benutzer*innen des Themes auch einen Widget-Bereich zur Verfügung haben, wird das Theme einen im Footer mitbringen. Dafür wird zunächst in der *functions.php* der Widget-Bereich registriert, der Code aus Listing 23.43 kann einfach am Ende der Datei eingefügt werden.

```
if ( ! function_exists( 'wh_classic_register_widgets_area' ) ) {
  function wh_classic_register_widgets_area() {
    $heading = __( 'Footer widgets' );
    register_sidebar(
      [
        'name'          => __( 'Footer widgets area', 'wh-classic' ),
        'id'            => 'footer-widgets-area',
        'before_widget' => '<aside id="%1$s" class="widget %2$s">',
        'after_widget'  => "</aside>\n",
        'before_title'  => '<h3 class="widget-title">',
        'after_title'   => '</h3>',
        'before_sidebar' => "<footer class='site-footer'><h2 class='screen-reader-
text'>$heading</h2>",
        'after_sidebar'  => '</footer>',
      ]
    );
  }
}
add_action( 'widgets_init', 'wh_classic_register_widgets_area' );
```

Listing 23.43 Damit im Backend ein Widget-Bereich angezeigt wird, in den Widgets eingefügt werden können, muss der Bereich registriert werden.

Die Funktion wird an den `widgets_init`-Hook gehängt, und darin wird mit `register_sidebar()` ein Widget-Bereich registriert. Diesem Bereich können in einem Array verschiedene Optionen mitgegeben werden; so wird ein Name angegeben, der im Backend sichtbar ist, eine ID, die zur Ausgabe im Theme benötigt wird, sowie Änderungen an dem vom Core standardmäßig vor und hinter Widgets eingefügten Markup.

Nach der Registrierung müsste im Backend unter Design • Widgets der Widget-Bereich angezeigt werden, wie in Abbildung 23.38 zu sehen.

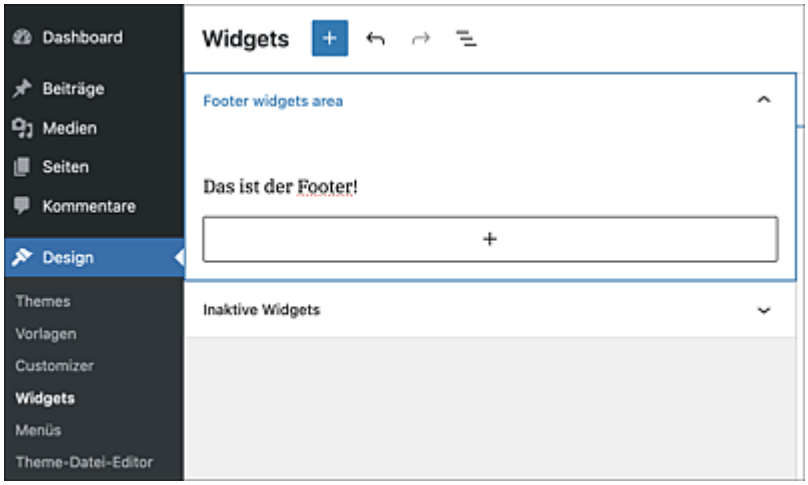


Abbildung 23.38 Der neu erstellte Widget-Bereich im Backend, hier bereits mit einem Absatz-Block befüllt

Als Nächstes geht es an die Ausgabe, dafür muss noch einmal die *footer.php* angepasst werden. Listing 23.44 zeigt das gesamte Markup der angepassten Datei.

```
    </main>
    <?php
    if ( is_active_sidebar( 'footer-widgets-area' ) ) {
      dynamic_sidebar( 'footer-widgets-area' );
    }
    wp_footer();
    ?>
  </body>
</html>
```

Listing 23.44 Die finale »footer.php« mit der Ausgabe des Widget-Bereichs

Neu ist der Teil nach dem schließenden `main`-Tag. Mit `is_active_sidebar( 'footer-widgets-area' )` wird geprüft, ob der gerade registrierte Widget-Bereich Widgets (bzw. Blöcke) enthält, und falls das der Fall ist, wird der Widget-Bereich mit der Funktion `dynamic_sidebar()` ausgegeben, der wieder die ID der Sidebar übergeben wird.

In Abbildung 23.39 wird das beispielhaft mit dem Absatz-Block aus dem Backend-Screenshot gezeigt.



Abbildung 23.39 Das ist nicht nur ein Absatz, sondern ein Absatz in dem neu registrierten und im Footer eingebundenen Widget-Bereich.

Jetzt sind die Anpassungen an den Template-Dateien geschafft, und es folgen nur noch ein paar Arbeiten an der *functions.php*. Beginnen wir damit, das Stylesheet einzubauen und die Schriften zu laden, damit die Webseite etwas schöner aussieht.

HTML5-Markup für WordPress-Komponenten aktivieren und Stylesheets einbinden

Einige Komponenten, die in Classic Themes vorkommen, können über einen `add_theme_support()`-Aufruf auf HTML5-Markup umgestellt werden. Wir wollen das für die Liste der Reaktionen auf einen Beitrag sowie für das Kommentar- und das Suchformular machen und passen deshalb in der *functions.php* die Funktion `wh_classic_add_theme_support()` an, wie in Listing 23.45 zu sehen.

```
    add_theme_support( 'automatic-feed-links' );
    add_theme_support( 'html5', [ 'comment-list', 'comment-form', 'search- form' ]
);
    add_theme_support( 'editor-styles' );
  }
}
add_action( 'after_setup_theme', 'wh_classic_add_theme_support' );
```

Listing 23.45 Erweiterung der Funktion »wh_classic_add_theme_support()« um zwei »add_theme_support()«-Aufrufe

Die erste Zeile mit `automatic-feed-links` ist in der Datei bereits vorhanden, genauso wie die letzten drei Zeilen, also die schließenden Klammern und der `add_action()`-Aufruf, sie dienen nur der Orientierung. Mit der Übergabe von `html5` als ersten Parameter wird das HTML5-Markup für die im zweiten Parameter als Array übergebenen Komponenten aktiviert. Den zweiten neuen Aufruf brauchen wir eigentlich erst gleich, wenn die Editor-Styles hinzugefügt werden, aber da wir die Datei sowieso gerade bearbeiten: `add_theme_support( 'editor-styles' );` aktiviert die Unterstützung für Editor-Styles.

Wie schon einmal kurz erwähnt, wird es hier nicht um die Erstellung des CSS-Codes gehen (den finden Sie hier: *http://r-wrk.de/wp6_css*). Die Styles sind fertig und in die *style.css* eingefügt, jetzt geht es um die Einbindung in WordPress. Dafür muss wieder die functions.php erweitert werden, den Code zeigt Listing 23.46:

```
if ( ! function_exists( 'wh_classic_enqueue_assets' ) ) {
  function wh_classic_enqueue_assets() {
    wp_enqueue_style(
      'wh-classic-style',
      get_stylesheet_uri(),
      [],
      filemtime( get_stylesheet_directory() . '/style.css' )
    );
  }
}
add_action( 'wp_enqueue_scripts', 'wh_classic_enqueue_assets' );
```

Listing 23.46 In der »functions.php« wird das Stylesheet des Themes eingebunden.

Hier kommt wieder die Funktion `wp_enqueue_style()` zum Einsatz, die bereits aus dem Abschnitt mit der Child-Theme-Erstellung bekannt ist. Der erste Parameter ist das eindeutige Handle des Stylesheets, der zweite die URL (hier können wir einfach `get_stylesheet_uri()` verwenden, das die URL zur *style.css* des Themes zurückgibt), den dritten Parameter lassen wir leer, da keine Abhängigkeiten zu anderen Stylesheets bestehen, und als vierten Parameter übergeben wir als Version-String den Timestamp der letzten Dateiänderung.

Jetzt sollte die Webseite schon gestylt aussehen, es fehlt aber noch die richtige Schrift. Wie Sie die IBM Plex Serif mit dem google-webfonts-helper herunterladen können, wurde im Abschnitt zur Block-Theme-Entwicklung erläutert – wir nutzen dieselben drei Schriftschnitte wie für das Block-Theme. Wenn Sie die Schriftdateien heruntergeladen haben, verschieben Sie sie in das *assets/fonts*-Verzeichnis des Themes. Die Einbindung nehmen wir über die *theme.json*-Datei vor, eine Datei zur Konfiguration von Themes, die im Abschnitt zur Block-Theme-Entwicklung genauer vorgestellt und umfangreicher genutzt wurde. Erstellen Sie dafür eine Datei namens *theme.json* im Theme-Verzeichnis, und füllen Sie sie mit dem folgenden Inhalt:

```
{
    "$schema": "https://schemas.wp.org/wp/6.7/theme.json",
    "version": 3,
    "settings": {
        "layout": {
            "contentSize": "30rem",
            "wideSize": "40rem"
        },
        "typography": {
            "fontFamilies": [
                {
                    "fontFamily": "\"IBM Plex Serif\", serif",
                    "name": "IBM Plex Serif",
                    "slug": "ibm-plex-serif",
                    "fontFace": [
                        {
```

```
                    "fontFamily": "IBM Plex Serif",
                    "fontWeight": "400",
                    "fontStyle": "normal",
                    "fontStretch": "normal",
                    "src": [
                        "file:./assets/fonts/ibm-plex-serif-v15-
                         latin-regular.woff2"
                    ]
                },
                {
                    "fontFamily": "IBM Plex Serif",
                    "fontWeight": "700",
                    "fontStyle": "normal",
                    "fontStretch": "normal",
                    "src": [
                        "file:./assets/fonts/ibm-plex-serif-v15-
                         latin-700.woff2"
                    ]
                },
                {
                    "fontFamily": "IBM Plex Serif",
                    "fontWeight": "400",
                    "fontStyle": "italic",
                    "fontStretch": "normal",
                    "src": [
                        "file:./assets/fonts/ibm-plex-serif-v15-
                         latin-italic.woff2"
                    ]
                }
            ]
        }
    ]
  }
}
```

Listing 23.47 Einbinden der Schriften über die theme.json-Datei und Festlegen der Inhaltsbreite.

Die *theme.json* wurde in dem Abschnitt zur Block-Theme-Entwicklung umfangreicher erklärt, deshalb fassen wir uns hier kurz: Mit `settings.layout.contentSize` und `settings.layout.wideSize` definieren wir die Breite des Inhalts. In `settings. typography.fontFamilies` ist ein Array aus Objekten angegeben, die jeweils einen der Schriftschnitte der Schrift enthalten. Wenn Sie jetzt die Startseite laden und alles geklappt hat,

sollten Sie vom Design her ungefähr das Ergebnis in Abbildung 23.40 erhalten.

Um das Thema Styling abzurunden, wird das Editor-Layout ein wenig angepasst. Theoretisch ist hier viel möglich, im Rahmen dieses Projekts wird sich darauf beschränkt, die Breite des Editors ungefähr auf das Niveau des Frontends zu bringen und die IBM Plex Serif im Editor zu nutzen. Die Voraussetzung für die Nutzung der Schrift im Editor wurde bereits durch die Einbindung über die *theme.json* im letzten Listing geschaffen, jetzt muss noch das Editor-Stylesheet eingebunden werden (Sie müssen dafür die Datei *assets/css/editor.css* anlegen, das CSS finden Sie hier: *http://r-wrk.de/wp6_classic_stylesheet*) und der Theme-Support muss aktiviert werden, das wurde schon in Listing 23.45 erledigt. Fehlt also nur noch die Einbindung der Datei. Den entsprechenden Code finden Sie im folgenden Listing, und er gehört wieder in die *functions.php*.



Abbildung 23.40 Die Startseite im gestylten Gewand

```
if ( ! function_exists( 'wh_classic_add_editor_style' ) ) {
  function wh_classic_add_editor_style() {
    add_editor_style( 'assets/css/editor.css' );
  }
}
add_action( 'admin_init', 'wh_classic_add_editor_style' );
```

Listing 23.48 Kurz und knapp das Einbinden des Editor-Stylesheets

Mit der Funktion `add_editor_style()` wird die übergebene Datei (der Pfad ist relativ zum Theme-Root-Verzeichnis) im Editor eingebunden. Den Aufruf der Funktion hängen wir an den `admin_init`-Hook.

Da am Anfang des Abschnitts bereits der Aufruf von `add_theme_support( 'editor-styles' );` eingefügt wurde, sollte der Editor aussehen wie in Abbildung 23.41. Der `add_theme_support()`-Aufruf sorgt unter anderem dafür, dass die Styles, die mit `add_editor_style()` übergeben werden, mit einem zusätzlichen Klassen-Selektor versehen werden, sodass sie sich nur auf den Editor-Bereich mit den Blöcken auswirken und nicht versehentlich auf das Admin-Menü oder die Editor-Seitenleiste.

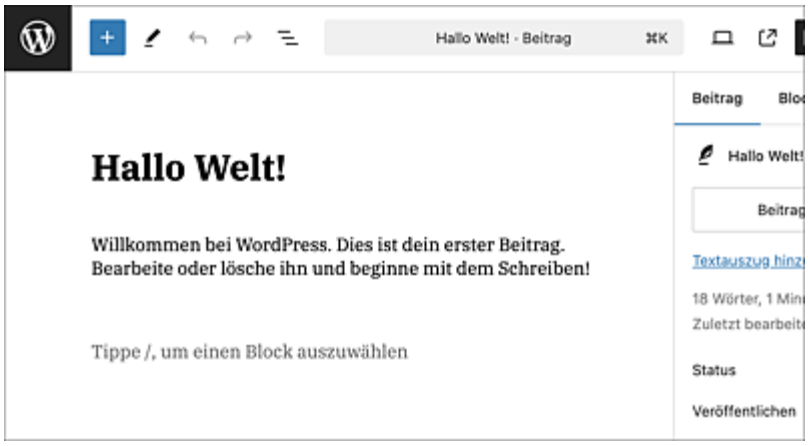


Abbildung 23.41 Jetzt ist auch die Editor-Ansicht zumindest ein bisschen an das Theme angepasst.

Theme für Übersetzungen vorbereiten

Das meiste ist eigentlich schon getan, da die ganzen sichtbaren Strings mit Übersetzungsfunktionen ausgegeben werden. Jetzt müssen in der *functions.php* noch die Übersetzungen geladen werden:

```
if ( ! function_exists( 'wh_classic_load_textdomain' ) ) {
  function wh_classic_load_textdomain() {
    load_theme_textdomain( 'wh-classic', get_template_directory() .
'/languages' );
  }
}
add_action( 'after_setup_theme', 'wh_classic_load_textdomain' );
```

Listing 23.49 Damit Übersetzungen angezeigt werden können, müssen sie geladen werden.

Der Code gleicht bis auf die Text-Domain und das Präfix im Funktionsnamen dem, der in dem Abschnitt zum Übersetzen eines Block-Themes genutzt wurde. Kurz: `load_theme_textdomain()` ist für das Laden der Übersetzungen zuständig und bekommt die Text-Domain des Themes sowie das Verzeichnis übergeben, in dem die Übersetzungen liegen. Wie Sie eine Übersetzung erstellen können, wird in Abschnitt 22.2.3, »Übersetzungen (Internationalisierung)«, erläutert.

Damit haben Sie es geschafft und Ihr erstes kleines Theme entwickelt.

Verbesserungs- und Erweiterungsmöglichkeiten

Wie gesagt, handelt es sich bei diesem Theme nur um ein sehr einfaches Theme, um die Grundlagen der Theme-Entwicklung zu veranschaulichen. Momentan wird die *theme.json* nur für die Einbindung der Schriften und das Festlegen der Inhaltsbreite genutzt, aber natürlich könnten Sie hier auch weitere Optionen nutzen, die in dem Block-Theme-Abschnitt vorgestellt wurden, um

zum Beispiel die Farbpalette im Block-Editor zu verändern. Auch beinhaltet das Theme aktuell keine Vorlagen oder alternative Templates für Seiten und/oder Beiträge. Aber Sie sollten jetzt eine gute Basis haben, um von hier komplexere Themes zu entwickeln. Aber denken Sie daran: Bauen Sie keine Funktionen ein, über die Benutzer*innen Inhalte anlegen können. Das Theme ist nur zur Gestaltung gedacht, alles andere gehört in Plugins.

23.3 Theme veröffentlichen

Egal ob Classic oder Block-Theme, wenn Sie es fertiggestellt haben, könnte es sein, dass Sie es gern anderen für ihre WordPress-Sites zur Verfügung stellen möchten. Dabei müssen Sie sich erst einmal überlegen, ob Sie das Theme kostenlos oder kostenpflichtig anbieten wollen. (Machen Sie sich nicht zu viele Hoffnungen, mit einem Theme viel Geld verdienen zu können: Es ist sehr schwierig, heute auf dem Theme-Markt zu starten und erfolgreich zu sein, da das bestehende Angebot extrem groß ist und sich auf Theme-Marktplätzen einige Themes etabliert haben, die zwar nicht immer gut sein müssen, weil viele davon Funktionen mitbringen, die in Plugins gehören, was aber Benutzer*innen häufig nicht wissen.) Wenn Sie Kauf-Themes anbieten möchten, versuchen Sie, sich eine Nische zu suchen, vielleicht haben Sie damit Glück.

Je nachdem, ob das Theme kostenlos oder kostenpflichtig sein soll, gibt es unterschiedliche Möglichkeiten der Veröffentlichung.

23.3.1 Ein kostenloses Theme veröffentlichen

Am einfachsten erscheint erst mal wahrscheinlich die Veröffentlichung als Link zum ZIP auf einer eigenen Website, falls Sie eine Website haben. Ebenfalls recht unkompliziert wäre die Nutzung eines GitHub-Repos (oder GitLab), auf dem Ihr Theme-Code liegt und heruntergeladen werden kann.

Dann müssten Sie aber beispielsweise noch überlegen, wie neue Versionen des Themes den Weg auf die Instanzen der Nutzer*innen finden.

Darum müssten Sie sich keine Gedanken machen, wenn Sie das Theme in das offizielle WordPress.org-Theme-Verzeichnis einreichen würden, was nebenbei bemerkt noch ein paar weitere Vorteile hat:

- Das Theme kann direkt aus dem WordPress-Backend heraus installiert werden.
- Wie schon angesprochen, müssen Sie sich nicht um einen Update-Mechanismus kümmern. Das übernimmt WordPress, sobald eine neue Version ins Theme-Verzeichnis hochgeladen wird.
- Sie brauchen Übersetzungen nicht lokal zu pflegen, sondern können die Plattform *translate.wordpress.org* verwenden. Damit können Sie auch Übersetzungen anpassen, ohne gleich ein Theme-Update veröffentlichen zu müssen.
- Für jedes Theme auf WordPress.org gibt es ein eigenes Support-Forum, in dem Fragen zum Theme gestellt und Sie antworten können.

Wenn Sie Ihr Theme bei WordPress.org veröffentlichen möchten, ist Ihre Anlaufstelle zum Upload *https://wordpress.org/themes/getting-started/*. Nach dem Upload durchläuft Ihr Theme einen Review-Prozess, in dem auf die Einhaltung unterschiedlicher Punkte geprüft wird, die im Theme-Review-Handbuch aufgelistet sind (*https://make. wordpress.org/themes/handbook/review/required/*). Diese Punkte sollten Sie unbedingt vor dem Upload durchgehen, da damit der Review-Prozess verkürzt wird und außerdem Ihr Theme bei Verletzung von mehr als drei Regeln direkt als nicht genehmigt abgelehnt wird (Sie können es dann später neu einreichen, wenn Sie die Punkte korrigiert haben, aber es spart beiden Seiten Zeit, wenn Sie es vorher machen). Auf folgende Punkte müssen Sie achten (nähere Informationen zu den Punkten finden Sie über die URL).

1. **Lizenz und Copyright**. Themes müssen unter einer Lizenz veröffentlicht werden, die kompatibel mit der Lizenz »GNU General Public License« ist. Das betrifft den gesamten Code und alles andere, was in der Theme-ZIP ist. Für alle Drittressourcen (Schriften, Bilder etc.) müssen die Lizenzen und Copyright-Informationen angegeben werden.
2. **Datenschutz**. Tracking und Sammeln von Daten dürfen nur nach Aktivierung erfolgen.
3. **Barrierefreiheit**. Ein Theme muss einen Sprunglink zum Inhalt enthalten, mit dem der Header-Bereich übersprungen werden kann. Alle Inhalte müssen nur mit der Tastatur erreichbar und bedienbar sein, und das fokussierte Element muss optisch hervorgehoben sein. Außerdem müssen Links innerhalb von Inhalten unterstrichen sein.
4. **Code**. Themes dürften keine PHP- oder JavaScript-Fehler/-Warnungen/-Notices werfen. Nicht vertrauenswürdige Eingaben müssen vor dem Speichern in der Datenbank validiert und bereinigt werden, vor der Anzeige müssen Daten escapet werden. Es dürfen keine veralteten Funktionen oder Konstanten genutzt werden, und alles, was das Theme im öffentlichen Namespace definiert (Funktionen, Variablen, Optionsnamen, Metadaten etc.), muss ein eindeutiges Präfix erhalten.
5. **Funktionen**. Unter diesem Punkt wird aufgelistet, was in einem Theme verboten ist, wie beispielsweise das Verstecken von WordPress-Funktionen hinter einer Bezahlschranke, die Registrierung von Inhaltstypen, die Erstellung von Blöcken, eigenen Rollen etc.
6. **Plugins**. Themes dürfen nur Plugins empfehlen, die auf WordPress.org gehostet werden, und Plugins dürfen nicht

automatisch installiert werden.

7. **Benennung, Schreibweise und Marken**. Das Themes-Team kann Themes aufgrund unangebrachter Theme-Namen ablehnen oder wenn der Name anderen Themes zu ähnlich ist. Außerdem sind ein paar Begriffe als Teil des Namens verboten, WordPress muss überall richtig geschrieben werden, und es dürfen keine Markenrechte verletzt werden.
8. **Sprache und Internationalisierung**. Alle Textstrings müssen mit Gettext-Funktionen übersetzbar sein, temporäre Ausnahmen sind Texte in HTML-Templates von Block-Themes. Der Theme-Slug muss als Text-Domain genutzt werden, und wenn das Theme ein Framework einsetzt, dürfen nicht mehr als zwei Text-Domains genutzt werden.
9. **Dateien**. Themes müssen fertig sein, wenn sie eingereicht werden, sie dürfen keinen Code enthalten, der nicht für die kostenlose Version gedacht ist, und entwicklungsspezifische Verzeichnisse wie .git müssen entfernt werden. Außerdem muss über alle Dateien hinweg derselbe Typ von Zeilenende genutzt werden (DOS oder UNIX).
   Ohne die Zustimmung der Theme-Nutzer*innen ist das Laden externer Ressourcen verboten, eine Ausnahme bilden Google Fonts (die Sie aber aus Datenschutzgründen auch nicht nutzen sollten). Darüber hinaus gibt es hier noch einige weitere Vorgaben.
10. **Classic Themes**. Es gibt einige Vorgaben, die nur Classic Themes betreffen, etwa dass bestimmte Funktionen aufgerufen werden müssen, zum Beispiel `language_attributes()`, `wp_head()`, `body_class()`, `wp_footer()` und `wp_body_open()`.
11. **Block-Themes**. Block-Themes müssen ein paar Dateien auf jeden Fall mitbringen (*style.css*, *templates/index.html*,

theme.json und *readme.txt*) und vollständig sein.

12. **Theme-Einstellungen und -Onboarding**. Hier sind einige Punkte aufgelistet, die bei der Erstellung von Themes zu berücksichtigen sind, wenn Änderungen im Backend vorgenommen werden. So sind Admin-Notices nur erlaubt, wenn sie einmalig auftauchen oder dauerhaft ausgeblendet werden können. Eine eigene Seite für Einstellungen ist unterhalb des Design-Menüpunkts erlaubt, es sollten aber die WordPress-User-Interface-Elemente genutzt werden. Weitere Punkte finden Sie auf der verlinkten Seite.
13. **Verkaufen, Credits, Links und Spam**. Themes dürfen im Frontend einen Credits-Link einbauen, der auf eine der beiden URLs verweisen muss, die als `Author URI` oder `Theme URI` in der *style.css* gesetzt sind. Wenn ein Hinweis auf eine kostenpflichtige Version des Themes eingebaut ist, darf das nicht im Frontend oder aufdringlich sein. Außerdem dürfen im Theme keine Spam- oder SEO-Inhalte vorhanden sein.
14. **Upload-Einschränkungen**. Sie können zeitgleich nur ein Theme in der Review-Schlange haben (auch wenn Sie mehrere WordPress.org-Accounts haben). Wenn Sie Themes außerdem noch anderswo anbieten, müssen auch diese Themes kompatibel mit der GPL sein.

Werden alle diese Regeln eingehalten, können Sie Ihr Theme hochladen und gespannt auf das Review-Ergebnis einer freiwilligen Person aus dem Themes-Team warten. Sobald das erfolgreich vorliegt, ist Ihr Theme live auf WordPress.org und damit für alle Benutzer*innen von WordPress direkt über das Backend installierbar.

23.3.2 Ein kostenpflichtiges Theme veröffentlichen

Wenn Sie die Nutzung Ihres Themes an eine Bezahlung koppeln möchten, haben Sie zwei Möglichkeiten: Entweder Sie verkaufen über einen eigenen Shop oder über einen Marktplatz wie Creative Market.

- **Eigener Shop**. Vorteile des eigenen Shops sind, dass Sie einen größeren Anteil des Gewinns pro Verkauf behalten können. Je nach Zahlungsart entstehen unterschiedlich hohe Gebühren, die aber alle niedriger sein dürften als der Anteil, den Sie an einen Marktplatz zahlen müssten. Außerdem haben Sie die volle Freiheit bei der Gestaltung des Shops. Nachteile sind, dass Sie sich zunächst mal selbst um die Einrichtung und den Betrieb des Shops kümmern und alle rechtlichen Aspekte beachten müssen. Außerdem müssen Sie einen Weg implementieren, über den Ihre Themes aktualisiert werden können, wenn Sie nicht möchten, dass Ihre Käufer*innen immer die ZIP der neuesten Version manuell im Backend installieren müssen. Und wenn Sie nicht schon ziemlich bekannt in der Zielgruppe sind, an die Sie das Theme verkaufen möchten, müssen Sie sich bzw. den Shop bekannt machen.
- **Marktplatz**. Vorteile eines Marktplatzes sind, dass Sie sich außer um den Upload des Themes um nicht viel kümmern müssen und dass die Marktplätze für viele Benutzer*innen eine Anlaufstelle sind, um nach Themes zu suchen. Außerdem gibt es bei spezialisierten Marktplätzen eine Chance, dass sie einen Update-Mechanismus anbieten, den Sie nutzen können, um Updates für neue Theme-Versionen auszuliefern. Nachteile sind, dass Sie pro Verkauf einen gewissen Anteil an den Marktplatz abgeben müssen und dass die Konkurrenz anderer Themes auf derselben Plattform vermutlich ziemlich hoch ist (bei Plattformen wie

ThemeForest sehr hoch). Egal für welche Variante Sie sich entscheiden: Es ist unwahrscheinlich, in diesem sehr vollen und unübersichtlichen Markt heute den Einstieg zu schaffen und erfolgreich zu werden. Wenn Sie etwas ganz Besonderes bieten, kann das natürlich gelingen, aber machen Sie sich keine falschen Vorstellungen und erwarten Sie nicht, in ein paar Monaten nur von Theme-Verkäufen leben zu können.

23.4 Weitere nützliche Funktionen für Themes

Natürlich bietet WordPress einige weitere Funktionalitäten, die Themes nutzen können. In den folgenden Abschnitten werden Sie verschiedene davon kennenlernen. Zum Teil gibt es Unterschiede dabei, ob Funktionen von Classic oder von Block-Themes unterstützt werden; das wird dann jeweils erwähnt.

23.4.1 Zusätzliche Bildgrößen registrieren

Zusätzlich zu den Bildgrößen, die WordPress beim Upload erstellt, können Sie eigene Bildgrößen anlegen. Das kann beispielsweise sinnvoll sein, wenn Sie in einem Theme an einer bestimmten Stelle Bilder ausgeben und keine der vorhandenen Bildgrößen diesen Maßen auch nur annähernd entspricht. In Listing 23.50 sehen Sie den Code zur Registrierung einer zusätzlichen `teaser`-Bildgröße, den Sie in die *functions.php* einfügen müssen oder in eine Datei, die in die *functions.php* eingebunden wird.

```
add_action(
  'after_setup_theme',
  function() {
    add_image_size( 'teaser', 300, 200 );
  }
);
```

Listing 23.50 Mit »add_image_size()« können Sie eine zusätzliche Bildgröße registrieren.

Innerhalb der Funktion, die an die `after_setup_theme`-Action gehängt wird, fügen Sie mit `add_image_size()` eine neue Bildgröße ein. Der erste Parameter ist der interne Bezeichner, den Sie auch

angeben müssen, wenn Sie die Bildgröße ausgeben möchten, zum Beispiel mit `the_post_thumbnail( 'teaser' )`. Die folgenden Bezeichner sind reserviert und dürfen nicht für eigene Größen genutzt werden: `thumb`, `thumbnail`, `medium`, `medium_large`, `large` und `post-thumbnail`. Der zweite Parameter ist die Breite des Bildes, der dritte die Höhe, und mit dem vierten Parameter könnten Sie angeben, ob das Bild genau auf diese Maße zugeschnitten werden soll. Andernfalls wird es in die angegebenen Maße eingepasst.

Die neue Bildgröße wird für bestehende Bilder nicht automatisch generiert, das heißt, wenn Sie bereits Bilder in der Mediathek haben, müssen Sie für diese die kleinen Bildgrößen neu generieren. Wenn Sie WP-CLI zur Hand haben, können Sie einfach den folgenden Befehl ausführen, um die Bilder für die neue Größe zu erstellen:

```
wp media regenerate --image_size=teaser
```

Alternativ gibt es Plugins, die sich um die Erstellung der Bildgrößen kümmern, zum Beispiel Regenerate Thumbnails *https://de.wordpress.org/plugins/regenerate-thumbnails/*.

Diese Bildgröße können Sie jetzt in Funktionen angeben, um Bilder auszugeben, aber für Benutzer*innen ist es nicht möglich, die neue Größe in den Editor einzufügen. Und vielleicht ist das auch genau das, was Sie möchten, weil die Größe nur an bestimmten Stellen genutzt und nicht einfach in den Inhalt eingefügt werden soll. In Listing 23.51 sehen Sie den Code, um die gerade erstellte Bildgröße im Editor auswählbar zu machen.

```
add_filter(
  'image_size_names_choose',
  function ( $sizes ) {
    $sizes['teaser'] = __( 'Teaser', 'wh-block' );
    return $sizes;
  }
);
```

Listing 23.51 Mit dem Filter »image_size_names_choose« kann eine neue Bildgröße für Benutzer*innen im Editor auswählbar gemacht werden.

Der Filter `image_size_names_choose` übergibt seiner Callback-Funktion ein Array der Bildgrößen, die im Editor ausgewählt werden können. Innerhalb der Funktion wird dem Array der Größen die eigene Größe hinzugefügt, indem als Array-Key der Größen-Bezeichner (`teaser`) und als Wert die für die Benutzer*innen sichtbare Beschriftung festgelegt wird. Das Ergebnis sehen Sie in Abbildung 23.42.

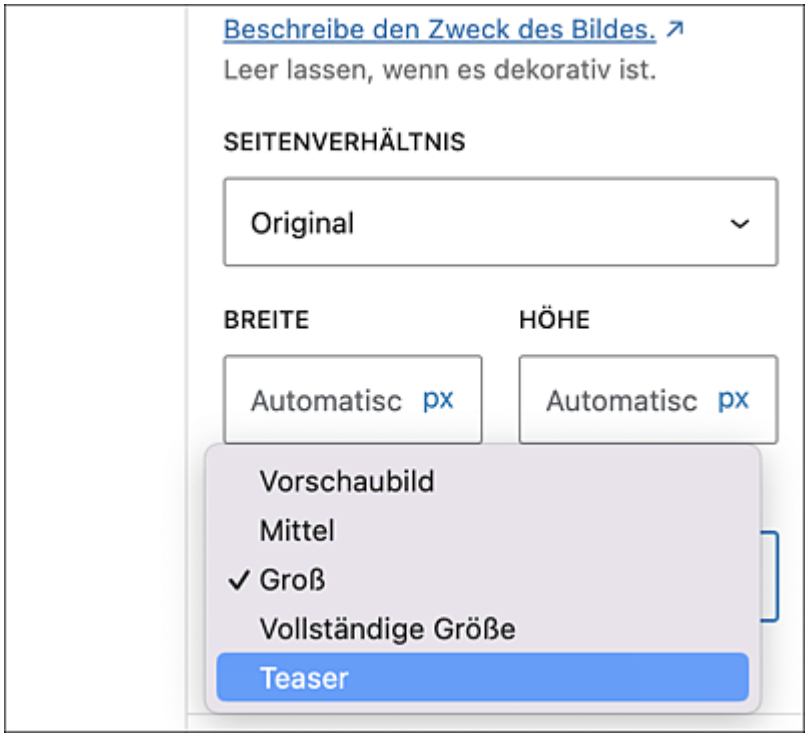


Abbildung 23.42 Nach dem Hinzufügen der neuen Größe zu den auswählbaren Bildgrößen taucht sie in der Bild-Block-Option zum Setzen der Größe auf.

23.4.2 Block-Stile registrieren

WordPress bietet die Möglichkeit, für Blöcke zusätzliche Stile zu registrieren. Wird dieser Stil im Block-Editor ausgewählt, wird dem Block-Element eine zusätzliche Klasse hinzugefügt, die Sie dann per

CSS ansprechen können, um den Stil entsprechend zu gestalten. In Listing 23.52 sehen Sie den Code, der einen Stil zum Gruppen-Block hinzufügt.

```
add_action(
  'after_setup_theme',
  function() {
    register_block_style( 'core/group', [
      'name'          => 'highlight',
      'label'         => __( 'Highlight', 'wh-blocks' )
    ] );
  }
);
```

Listing 23.52 Ein Block-Stil für den Gruppen-Block

Zur Registrierung des Block-Stils wird die Funktion `register_block_style()` verwendet, an die als erster Parameter der Block-Bezeichner übergeben wird, für den der Stil gedacht ist, und als zweiter Parameter ein Array mit weiteren Parametern. Notwendig sind hier `name` und `label`. Ersteres ist der interne Name, der für die Generierung des Klassennamens verwendet wird, und das `label` ist der sichtbare Name. Wie das in der Editor-Seitenleiste aussieht, sehen Sie in Abbildung 23.43.

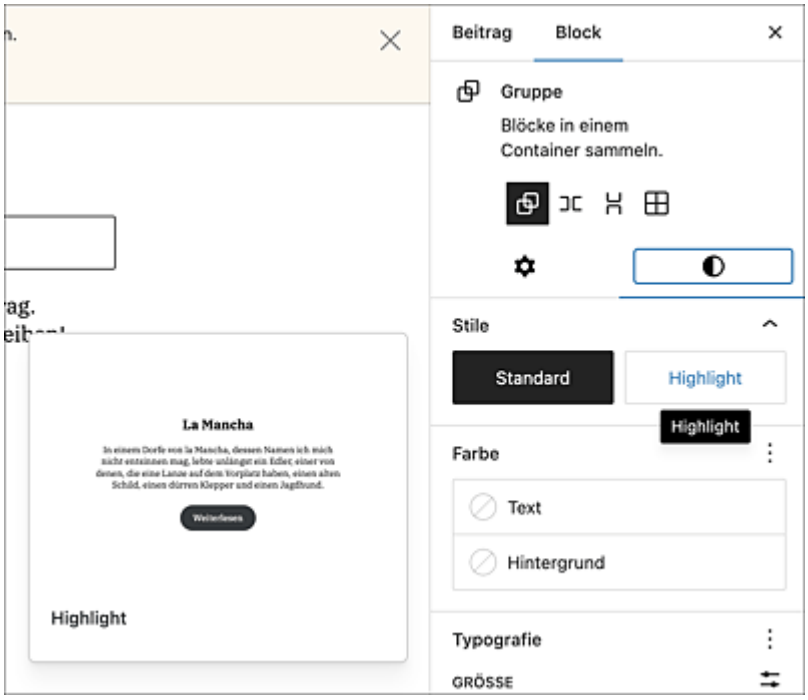


Abbildung 23.43 In der Block-Sidebar kann der neue Stil ausgewählt werden.

Wie beschrieben, wird bei der Auswahl eines Stils dem Block eine Klasse zugewiesen. Wie das im Frontend aussieht, zeigt Abbildung 23.44. Als Präfix für Stilklassen wird `is-style-` verwendet, danach folgt der `name` des Stils, für den `highlight`-Stil heißt die Klasse dementsprechend `is-style-highlight`.

```
<div class="entry-content">
  <div class="wp-block-group is-style-highlight is-l
  d"> == $0
    <p>Ein Absatz im Highlight-Gruppe-Block.</p>
  </div>
```

Abbildung 23.44 Die HTML-Klasse für Stile wird aus dem Präfix »is-style-« und dem »name« des Stils gebildet.

Block-Stil-Variationen erstellen

In WordPress 6.6 wurde die Möglichkeit der Block-Stile erweitert, unter anderem kann sie für mehrere Blöcke gleichzeitig registriert werden und direkt Stil-Definitionen mit bei der Registrierung

angeben. Außerdem sind neue Wege der Registrierung möglich: Sie können die Variationen jetzt auch über JSON-Dateien im *styles*-Verzeichnis erstellen, ähnlich wie die Theme-Style-Variationen.

Dafür wurden auf erster Ebene der JSON-Datei zwei neue Eigenschaften eingeführt: `blockTypes` und `slug`. Slug ist ein eindeutiger Wert der Variation, `blockTypes` ist ein Array der Blöcke, für die die Variation gilt. Folgende Limitierungen gibt es:

- Aktuell können Sie keine `settings` für die Variationen angeben, sondern nur `styles` definieren.
- Es können momentan nur die Root-Styles im Styles-Bereich des Website-Editors verändert werden, nicht die Styles der inneren Blöcke.
- Eigene Block-Stil-Variationen können nicht im Stilbuch angeschaut werden.
- Variationen können keine Variationen von verschachtelten Blöcken neu definieren oder verändern.

So kann eine einfache Variation für den Gruppe-Block aussehen:

```
{
  "$schema": "https://schemas.wp.org/wp/6.7/theme.json",
  "version": 3,
  "title": "Dark",
  "blockTypes": [
    "core/group"
  ],
  "styles": {
    "color": {
      "text": "var:preset|color|light",
      "background": "var:preset|color|dark"
    },
    "spacing": {
      "padding": {
        "top": "1.5rem",
        "right": "1.5rem",
        "bottom": "1.5rem",
        "left": "1.5rem"
      }
    }
```

```
  }
}
```

Listing 23.53 Eine Stil-Variation für Gruppen-Blöcke, angelegt als JSON-Datei innerhalb des »styles«-Verzeichnisses

Wenn Sie diese Datei zum Beispiel als *styles/dark.json* in Ihrem Block-Theme anlegen und danach im Editor einen Gruppe-Block einfügen, haben Sie in der Block-Sidebar den neuen »Dark«-Stil, der bei Auswahl direkt auch die Style-Angaben anwendet.

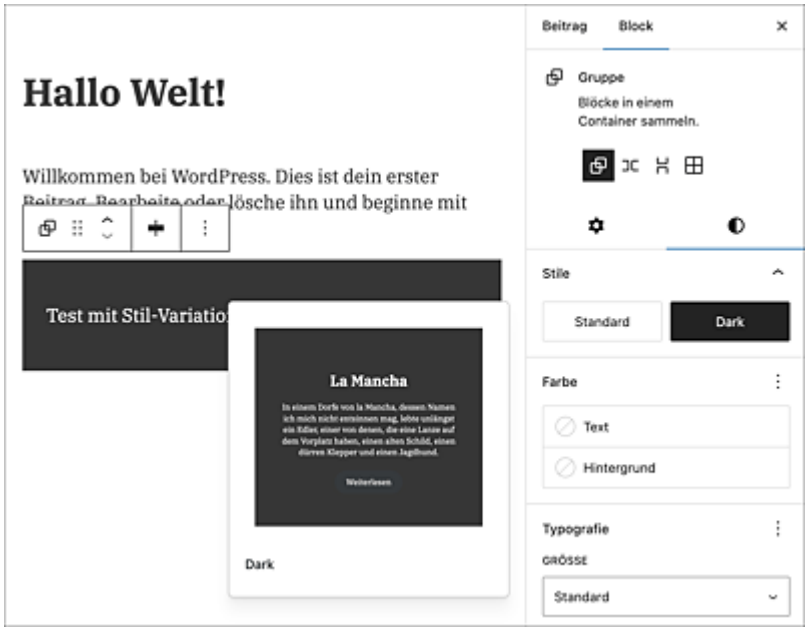


Abbildung 23.45 Die neue Variation inklusive Styles im Editor

23.4.3 Core-Block-Stile entfernen

WordPress bringt von Haus aus bereits für einige Blöcke zusätzliche Stile mit, zum Beispiel für den Zitat-Block. Wenn Sie diese Stile nicht benötigen, können Sie sie mithilfe eines Filters entfernen, wie in Listing 23.54 gezeigt wird.

```
add_filter( 'block_type_metadata', function( $metadata ) {
  if ( strpos( $metadata['name'], 'core/' ) !== 0 ) {
    return $metadata;
  }
```

```
  if ( empty( $metadata['styles'] ) ) {
    return $metadata;
  }

  unset( $metadata['styles'] );

  return $metadata;
} );
```

Listing 23.54 Mit ein paar Zeilen Code lassen sich die Block-Stile entfernen, die WordPress mitbringt.

Der `block_type_metadata`-Filter wird vor der Registrierung jedes Blocks aufgerufen, der über eine *block.json* konfiguriert wird, und übergibt an die Callback-Funktion ein Array mit der Konfiguration des Blocks. Im Callback prüfen Sie zunächst, ob es sich um einen Core-Block handelt, also ob der Name mit dem Namespace `core/` beginnt. Ist das nicht der Fall, werden die Metadaten unverändert zurückgegeben. Gleiches gilt, wenn `$metadata['styles']` leer ist, weil als Wert für diesen Schlüssel die Block-Stile festgelegt werden. Blöcke, die hier keinen Wert haben, haben somit auch keine Block-Stile.

Wenn Stile vorhanden sind, wird der Array-Key entfernt und das Ergebnis zurückgegeben. Damit wurden die Block-Stile, die WordPress direkt mitbringt, von den Core-Blöcken entfernt.

23.4.4 Sperren (Locking) von Blöcken und Templates

Im Block-Editor können Blöcke auf unterschiedliche Arten gesperrt werden, sodass bestimmte Aktionen nicht mehr erlaubt sind. Folgende Dinge können verhindert werden:

- Umsortieren von Blöcken
- Entfernen von Blöcken

- Einfügen von Blöcken (nicht über das User-Interface möglich)
- Nutzung der Block-Einstellungen über die Änderung von Inhalten und Medien hinaus (nicht über das User-Interface möglich)

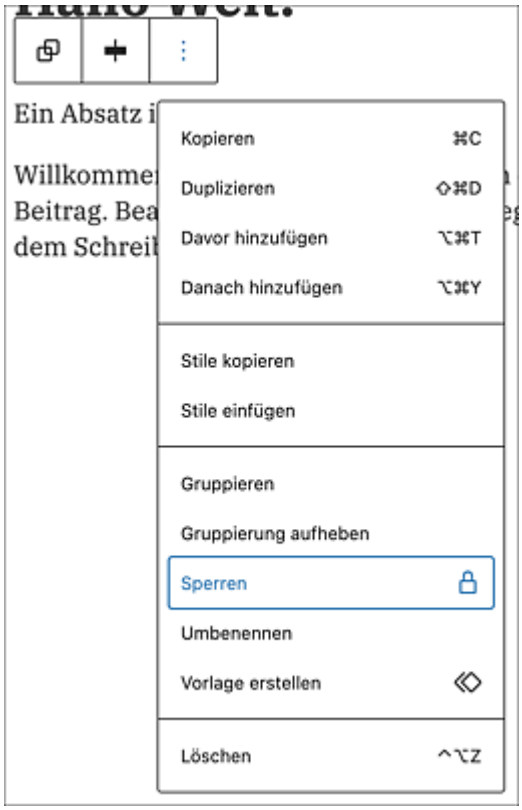


Abbildung 23.46 Über das Drei-Punkte-Menü eines Blocks können im Editor unterschiedliche Optionen zum Sperren genutzt werden.

Diese Optionen sind besonders hilfreich, wenn Sie beispielsweise Templates in Ihrem Block-Theme anlegen und bei bestimmten Bereichen oder beim gesamten Template etwa verhindern möchten, dass versehentlich Blöcke gelöscht oder umsortiert werden.

Wenn Sie beispielsweise einen Gruppe-Block mit einem Absatz und Bild erstellen möchten, und zwar so, dass Benutzer*innen nur den Inhalt vom Absatz und das Bild ändern können, müssen Sie manuell im Block-Kommentar des Gruppe-Blocks die Einstellung `"templateLock":"contentOnly"` ergänzen:

```
<!-- wp:group {"layout":{"type":"constrained"}, "templateLock":"contentOnly"} -->
<div class="wp-block-group">
<!-- wp:paragraph -->
<p>Absatz</p>
<!-- /wp:paragraph -->
<!-- wp:image -->
<figure class="wp-block-image"><img alt=""/></figure>
<!-- /wp:image --></div>
<!-- /wp:group -->
```

Listing 23.55 Markup eines Gruppe-Blocks mit Bild und Absatz, bei dem nur der Inhalt des Absatzes und das Bild verändert werden können

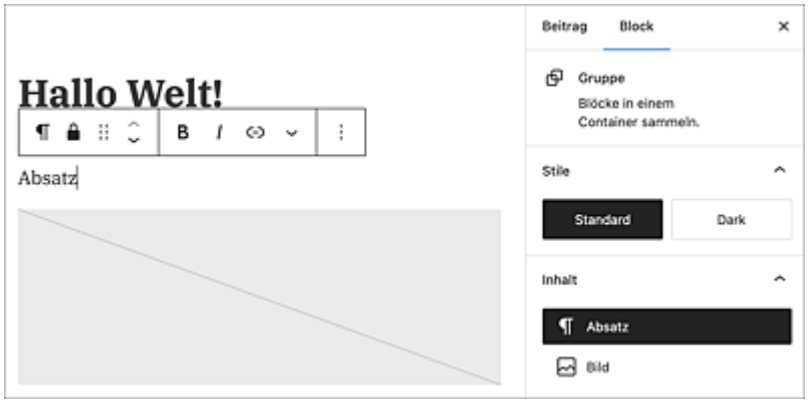


Abbildung 23.47 Bei dem Gruppe-Block kann nur noch der Inhalt verändert werden. Die Einstellungen des Absatz- und Bild-Blocks sind nicht mehr zugänglich.

Die Einstellungen zum Sperren ausblenden

Wenn Sie ein Theme für ein spezielles Projekt erstellen und nicht für den freien Download, möchten Sie in einigen Fällen wahrscheinlich verhindern, dass die Einstellungen für das Sperren im Editor angepasst werden können.

Um das zu tun, können Sie den Filter `block_editor_settings_all` verwenden, um die Einstellungen des Editors zu verändern und die Sperren-Option zu entfernen:

```
add_filter( 'block_editor_settings_all', function( $settings, $context ) {
    $settings['canLockBlocks'] = false;

    return $settings;
}, 10, 2 );
```

Listing 23.56 Ausblenden der Option zum Sperren und Entsperren von Blöcken

24 Plugin und Block entwickeln

Wenn Sie kein passendes Plugin finden, bleibt nur, jemanden für die Programmierung zu bezahlen oder sich selbst daran zu versuchen. In diesem Kapitel werden Sie ein Plugin entwickeln, das verschiedene Aspekte von WordPress berührt, und damit für Ihren weiteren Weg in der Plugin-Entwicklung gerüstet sein.

Dieses gesamte Kapitel wird sich um die Entwicklung eines Plugins drehen, mit dem Sie Vorträge in WordPress einpflegen können. Grundlage der Verwaltung wird ein eigener *Inhaltstyp* sein, der, ähnlich wie bei Beiträgen und Seiten, das Anlegen und Bearbeiten der Vorträge ermöglichen wird. Zu jedem Vortrag wird es möglich sein, Metadaten über einen eigenen Bereich in der Editor-Seitenleiste zu verwalten, unter anderem um die Sprache und die Länge des Vortrags anzugeben. Um diese Informationen auf der Website auszugeben, werden zwei Blöcke für den Block-Editor entwickelt: einer, der eine Liste aller gesetzten Metadaten mit einem Label ausgibt, und einer, der die Anzeige einzelner Metadaten mit jeweils einem beliebigen Label erlaubt. So wird es auch möglich sein, die Metadaten für Templates in einem Block-Theme zu verwenden.

24.1 Installation von Node.js

Da auch für den Block-Editor entwickelt wird, benötigt die Entwicklungsumgebung ein bisschen Vorbereitung. Sie brauchen auf Ihrem System Node.js (*https://nodejs. org/en/*), damit über das mitgelieferte *npm* (*node package manager*) später ein Paket installiert werden kann, das für die Entwicklung für den Block-Editor erstellt wurde und die Entwicklung erheblich vereinfacht.

Es gibt eigentlich für alle Betriebssysteme ein passendes Paket, mit dem Sie Node.js installieren können. Für Windows und Mac werden Installer angeboten, für Linux können Sie sich die Binarys herunterladen oder Node.js über verschiedene Package-Manager für Ihre Linux-Distribution installieren. Wenn Sie die Installation beendet haben, können Sie das Grundgerüst des Plugins erstellen.

24.2 Das Grundgerüst des Plugins

Viele Voraussetzungen für ein Plugin gibt es nicht, es sind sogar ein paar weniger als für ein Theme. Damit ein Plugin in WordPress auftaucht und aktiviert werden kann, braucht es nur eine PHP-Datei innerhalb des *wp-content/plugins*-Verzeichnisses mit einem Header-Kommentar, in dem der Titel des Plugins definiert wird. Wenn Sie in einer frischen WordPress-Installation in dieses Verzeichnis schauen, werden Sie mit der *hello.php* ein Eine-Datei-Plugin vorfinden. Üblicher und übersichtlicher ist es aber, einen Unterordner zu erstellen und dort die Plugin-Dateien zu platzieren, selbst wenn es nur eine einzige Datei sein sollte.

Beginnen Sie also das Plugin mit diesem mindestens notwendigen Gerüst. Erstellen Sie in *wp-content/plugins* den Ordner *wh-talks* und darin die Datei *wh-talks.php*. Öffnen Sie die Datei im Code-Editor Ihrer Wahl, und schreiben Sie den Inhalt aus Listing 24.1 hinein.

```
<?php
/**
 * Plugin Name: Talks
 * Description: Managing talks you gave.
 * Author: Ihr Name
 * Author URI: https://example.com /
 * Version: 1.0.0
 * Requires at least: 6.6
 * Requires PHP: 7.0
 * Text Domain: wh-talks
 */
```

Listing 24.1 Ähnlich wie bei Themes ist auch bei Plugins ein Kommentar mit Metadaten dafür zuständig, dass WordPress das Plugin erkennt.

Dieser PHP-Kommentar sorgt dafür, dass das Plugin im Backend unter Plugins • Installierte Plugins angezeigt wird. Wirklich notwendig ist nur die Angabe von `Plugin Name`, dem Namen des Plugins. Mit den übrigen Angaben geben Sie eine Beschreibung, den

Namen und die URL der Entwicklerin oder des Entwicklers und die Version an. `Requires at least` gibt die mindestens benötigte WordPress-Version an, mit der das Plugin funktioniert, und `Requires PHP`, welche PHP-Version benötigt wird. Die `Text Domain` ist für die Übersetzung wichtig und entspricht dem Slug des Plugins (Name des Ordners). Wenn Sie nun in die Plugins-Liste schauen, sollte Ihnen das Plugin angezeigt werden, wie in Abbildung 24.1 zu sehen. Sie können es ruhig schon mal aktivieren, auch wenn es bisher nichts tut.

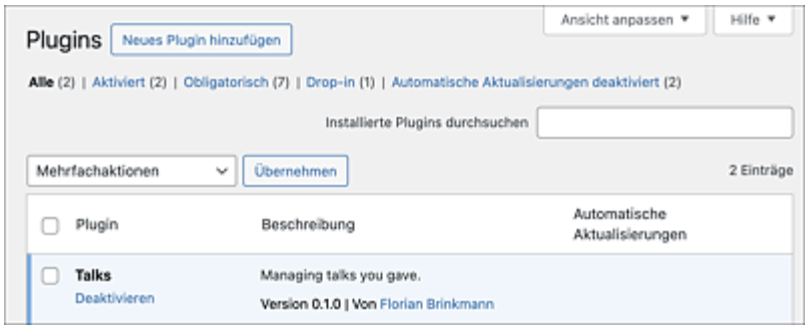


Abbildung 24.1 Mehr als ein PHP-Kommentar mit Metaangaben ist für ein WordPress-Plugin nicht notwendig.

Damit haben Sie das Plugin-Grundgerüst bereits fertiggestellt. Jetzt geht es an die Erstellung des Inhaltstyps, damit das Plugin auch etwas Mehrwert bietet.

24.3 Inhaltstyp erstellen

Beiträge und Seiten sind Inhaltstypen (Post-Types) von WordPress, und WordPress ermöglicht es, eigene Inhaltstypen (Custom Post-Types) zu erstellen, die von der Art und Weise der Verwaltung dann ziemlich genau so funktionieren wie Seiten und Beiträge. Diese Möglichkeit wird von sehr vielen Plugins genutzt, um eigene Inhalte zu verwalten. Das E-Commerce-Plugin WooCommerce beispielsweise bringt mehrere eigene Inhaltstypen mit, einen für die Produkte des Shops, einen für Bestellungen und so weiter.

Die Erstellung eines Inhaltstyps eröffnet Ihnen also viele Möglichkeiten, und wie Sie gleich sehen werden, ist es trotzdem gar nicht so schwer. Um die Hauptdatei des Plugins (die *wh-talks.php*) übersichtlich zu halten, werden Sie die Funktionen, die den Inhaltstyp betreffen, in eine Datei *inc/cpt.php* auslagern (*cpt* steht für Custom Post-Type). Diese Datei können Sie schon mal anlegen; bevor dort aber die ersten Zeilen Code geschrieben werden, muss die *wh-talks.php* um den Code aus Listing 24.2 erweitert werden.

```
namespace WH\Talks;

require_once 'inc/cpt.php';
```

Listing 24.2 Mehr kommt in die »wh-talks.php« erst mal zusätzlich nicht rein.

Zunächst wird ein *Namespace* definiert, um die Notwendigkeit von Präfixen für Funktionsnamen zu umgehen, und danach wird die Datei *inc/cpt.php* eingebunden. Das war es fürs Erste auch schon mit der Anpassung dieser Datei.

Als Nächstes geht es in der *inc/cpt.php* weiter, in der die eigentliche Registrierung des Inhaltstyps für die Verwaltung der Vorträge vorgenommen wird. Wie das aussieht, ist in Listing 24.3 gezeigt.

```php
<?php
namespace WH\Talks;

function register_cpt() {
 $labels = [
  'name'                     => __( 'Talks', 'wh-talks' ),
  'singular_name'            => __( 'Talk', 'wh-talks' ),
  'add_new_item'             => __( 'Add new talk', 'wh-talks' ),
  'edit_item'                => __( 'Edit talk', 'wh-talks' ),
  'new_item'                 => __( 'New talk', 'wh-talks' ),
  'view_item'                => __( 'View talk', 'wh-talks' ),
  'view_items'               => __( 'View talks', 'wh-talks' ),
  'search_items'             => __( 'Search talks', 'wh-talks' ),
  'not_found'                => __( 'No talks found', 'wh-talks' ),
  'not_found_in_trash'       => __( 'No talks found in trash', 'wh-talks' ),
  'all_items'                => __( 'All talks', 'wh-talks' ),
  'archives'                 => __( 'Talk archives', 'wh-talks' ),
  'attributes'               => __( 'Talk attributes', 'wh-talks' ),
  'insert_into_item'         => __( 'Insert into talk', 'wh-talks' ),
  'uploaded_to_this_item'    => __( 'Uploaded to this talk', 'wh-talks' ),
  'filter_items_list'        => __( 'Filter talks list', 'wh-talks' ),
  'items_list_navigation'    => __( 'Talks list navigation', 'wh-talks' ),
  'items_list'               => __( 'Talks list', 'wh-talks' ),
  'item_published'           => __( 'Talk published.', 'wh-talks' ),
  'item_published_privately' => __( 'Talk published privately', 'wh-talks' ),
  'item_reverted_to_draft'   => __( 'Talk reverted to draft.', 'wh-talks' ),
  'item_scheduled'           => __( 'Talk scheduled.', 'wh-talks' ),
  'item_updated'             => __( 'Talk updated.', 'wh-talks' ),
  'item_link'                => __( 'Talk link', 'wh-talks' ),
  'item_link_description'    => __( 'A link to a talk', 'wh-talks' ),
 ];

 $args = [
  'labels'       => $labels,
  'supports'     => [
   'title',
   'editor',
   'custom-fields',
   'excerpt',
   'thumbnail',
   'revisions',
  ],
  'hierarchical' => false,
  'public'       => true,
  'show_in_rest' => true,
  'has_archive'  => true,
  'rest_base'    => 'talks',
  'rewrite'      => [ /* translators: slug for permalink of CPT */
   'slug' => __( 'talks', 'wh-talks' ),
  ],
  'menu_icon'    => 'dashicons-megaphone',
 ];
 register_post_type( 'talk', $args );
```

```
}
add_action( 'init', __NAMESPACE__ . '\\register_cpt' );
```

Listing 24.3 Die Registrierung des »talk«-Inhaltstyps in der »inc/cpt.php«

1. Innerhalb der `register_cpt()`-Funktion wird zunächst ein großes Array mit den für Benutzer*innen sichtbaren Beschriftungen rund um den Inhaltstyp erstellt. Das geht von der Beschriftung des Menüpunkts links im Admin-Menü bis zu den Nachrichten, die angezeigt werden, wenn ein Eintrag des Inhaltstyps für die Veröffentlichung geplant oder ein veröffentlichter Eintrag zurück in den Entwurfsstatus geändert wurde. Das sind alle Labels, in denen der Name des Inhaltstyps auftaucht, daneben gibt es noch ein paar weitere (*https://developer.wordpress.org/reference/functions/get_post_type_labels/*).
2. In `$args` wird ein Array mit den Argumenten für den Inhaltstyp gespeichert. Das ist zunächst das Array mit den ganzen Beschriftungen (`$labels`), gefolgt von dem `supports`-Schlüssel, dem ein Array mit Editorfunktionen übergeben wird, die der Inhaltstyp unterstützt. Die meisten der Einträge sind selbsterklärend. `custom-fields` ist wichtig, damit später das Speichern der Metadaten funktioniert, und `revisions` bedeutet, dass Revisionen der Inhalte gespeichert werden sollen, so wie es für Seiten und Beiträge auch gemacht wird, wenn es nicht über die *wp-config.php* deaktiviert wurde (was keine gute Idee ist, Revisionen sind ein nützliches kleines Sicherheitsnetz).

   Der Inhaltstyp soll nicht `hierarchical` sein, das bedeutet, dass keine Eltern-Kind-Beziehungen zwischen Talks erstellt werden können, wie das bei Seiten möglich ist. Sie möchten, dass der Inhaltstyp öffentlich (`public`) ist und dass er in der REST-API angezeigt wird (`show_in_rest`). Das ist die Voraussetzung dafür,

dass der Block-Editor für den Inhaltstyp aktiviert wird, andernfalls würde der Classic Editor angezeigt werden. Mit `has_archive` aktivieren Sie die Archivübersicht für den Inhaltstyp (also die Übersicht der neuesten Talks), mit `rest_base` legen Sie die Basis für den Inhaltstyp in der REST-API fest, und an `rewrite` übergeben Sie in einem Array die Informationen darüber, wie die Links für den Inhaltstyp aussehen sollen. Durch die Angabe `'slug' => __( 'talks', 'wh-talks' )`, legen Sie fest, dass die Archivseite der Vorträge über *example.com/talks* und die Einzelansicht über *example.com/talks/spannender-vortrag* erreichbar sein werden. Da der Slug übersetzbar ist, kann er beispielsweise für Deutsch auf `vortraege` geändert werden, um die URL in *example. com/vortraege* zu ändern.

   Als letzter Eintrag im Array wird mit `'menu_icon' => 'dashicons-megaphone'`, das Menü-Icon für den Inhaltstyp auf das Megafon-Icon der WordPress-Dashicons geändert (*https://developer.wordpress.org/resource/dashicons/*). Statt eines Dashicons kann hier auch ein Base64-encodiertes SVG angegeben werden.

3. Jetzt wird `register_post_type()` aufgerufen, um den Inhaltstyp zu registrieren. Der erste Parameter, der an die Funktion übergeben wird, ist der interne Bezeichner des Inhaltstyps, der beispielsweise genutzt wird, um mit `is_post_type()` zu prüfen, ob ein Inhalt zu einem bestimmten Inhaltstyp gehört. Als zweiter Parameter wird das Argumente-Array `$args` übergeben.
4. Abschließend wird die Funktion `register_cpt()` an den `init`-Action-Hook von WordPress angehängt.

Wenn Sie jetzt ins Backend wechseln oder es neu laden, sollten Sie im Admin-Menü den neuen Punkt Talks sehen, wie in

Abbildung 24.2 gezeigt (vorausgesetzt, Sie hatten vorhin das Plugin bereits aktiviert).

Hier sehen Sie schon das angepasste Icon und ein paar der eingetragenen Beschriftungen. Wenn Sie jetzt aber testweise einen Eintrag anlegen, veröffentlichen und im Frontend anschauen möchten, landen Sie auf einer 404-Fehlerseite. Das liegt daran, dass WordPress die ganzen Regeln dazu, wie es URLs zu Inhalten auflöst, in der Datenbank speichert und dieser Eintrag nicht automatisch neu erstellt wird, wenn ein neuer Inhaltstyp vorhanden ist. Zum Testen lässt sich das beheben, indem Sie die Seite EINSTELLUNGEN • PERMALINKS aufrufen (Sie müssen die Einstellungen gar nicht speichern, die Seite aufzurufen reicht schon). Danach sollten Sie den Vortrag im Frontend sehen können.

Das ist eine etwas unschöne Situation, deshalb werden Sie im nächsten Schritt eine Lösung dafür kennenlernen.

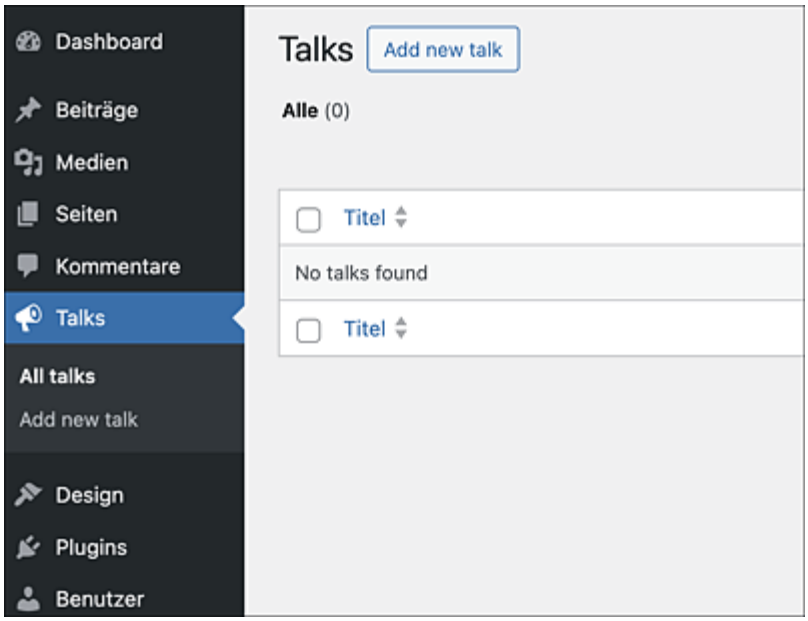


Abbildung 24.2 Nach der Registrierung des Inhaltstyps können Vorträge erstellt werden.

24.3.1 Permalink-Struktur automatisch aktualisieren

Wenn Sie ein Plugin schreiben, das *Rewrite-Rules* (die Regeln, wie WordPress URLs auflöst) hinzufügt oder ändert (beispielsweise weil ein Inhaltstyp registriert wird), sollten Sie beim Aktivieren und Deaktivieren des Plugins die Rewrite-Rules neu generieren, so als würden Sie manuell die Permalinks-Einstellungen aufrufen. Das ist recht schnell gemacht, Sie schreiben einfach den folgenden Code in die *wh-talks.php* (vor dem Einbinden der *inc/cpt.php*-Datei):

```
register_activation_hook( __FILE__, function() {
 register_cpt();
 flush_rewrite_rules();
} );

register_deactivation_hook( __FILE__, function() {
 flush_rewrite_rules();
} );
```

Listing 24.4 Wenn ein Plugin Anpassungen der Rewrite-Rules vornimmt, ist es eine gute Idee, diese bei Aktivierung und Deaktivierung neu generieren zu lassen.

`register_activation_hook()` wird bei der Aktivierung eines Plugins ausgeführt, `register_deactivation_hook()` bei der Deaktivierung. Als ersten Parameter erwarten beide Funktionen den Dateinamen der Haupt-Plugin-Datei inklusive des Pfads. Dafür kann die PHP-Konstante `__FILE__` verwendet werden, die die aktuelle Datei inklusive Pfad zurückgibt. Der zweite Parameter ist die Funktion, die ausgeführt werden soll – in beiden Fällen wird hier eine anonyme Funktion genutzt, statt eine extra Funktion zu definieren und deren Namen zu übergeben. Im Activation-Hook wird zuerst die Funktion `register_cpt()` aufgerufen, um WordPress über den Inhaltstyp und dessen Rewrite-Einstellungen zu informieren, gefolgt von `flush_rewrite_rules()`, um die Rewrite-Rules in der Datenbank neu zu generieren. In `register_deactivation_hook()` werden nur die Rewrite-Rules neu erstellt, damit die Regeln des Inhaltstyps gelöscht werden, da sie nicht länger gebraucht werden.

Wenn Sie jetzt das Plugin deaktivieren und wieder neu aktivieren, sollten Sie direkt den Vortrag im Frontend aufrufen können, ohne die Permalinks neu generieren zu müssen (um sicherzugehen, dass nicht einfach die eben manuell über das Aufrufen der Permalinks-Seite neu erstellten Regeln noch vorhanden sind und es deshalb funktioniert, können Sie mit deaktiviertem Plugin die Permalinks-Einstellungen aufrufen, um die Regeln des Inhaltstyps wieder zu entfernen, und danach das Plugin aktivieren und testen).

24.3.2 Metadaten für den Inhaltstyp registrieren

Um die Vorträge mit zusätzlichen Daten anzureichern, die beispielsweise in der Übersicht der Vorträge mit angezeigt werden können oder an einer bestimmten Stelle in der Einzelansicht, werden Sie diese Daten als *Post-Meta* speichern. Die Daten liegen in einer eigenen Datenbanktabelle und können so bequem unabhängig vom eigentlichen Inhalt aus dem Block-Editor angezeigt werden.

Diese Metadaten sollen aus dem Block-Editor heraus aktualisiert werden können, weshalb es wichtig ist, dass die Metadaten explizit registriert werden und in der REST-API sichtbar sind, damit aus dem Block-Editor heraus auf sie zugegriffen werden kann. Die folgenden Metadaten sollen für Vorträge festgelegt werden können:

- Name der Veranstaltung, auf der der Vortrag gehalten wurde
- Sprache des Vortrags
- Länge des Vortrags
- URL zu einem Video des Talks
- URL zu den Slides des Talks

Diese Metadaten und eine für Benutzer*innen sichtbare Beschriftung werden später an mehreren Stellen verwendet, weshalb ein Array mit den Informationen der Metadaten von einer eigenen Funktion zurückgegeben wird, die in Listing 24.5 abgedruckt ist und in die *inc/cpt.php* gehört.

```
function get_string_metas() {
 return [
  [
   'key'   => 'wh_talks_event_name',
   'label' => __( 'Event', 'wh-talks' ),
  ], [
   'key'   => 'wh_talks_language',
   'label' => __( 'Language', 'wh-talks' ),
  ],
  [
   'key'   => 'wh_talks_duration',
   'label' => __( 'Duration', 'wh-talks' ),
  ], [
   'key'   => 'wh_talks_video_link',
   'label' => __( 'Video', 'wh-talks' ),
  ],
  [
   'key'   => 'wh_talks_slides_link',
   'label' => __( 'Slides', 'wh-talks' ),
  ],
 ];
}
```

Listing 24.5 Die Definition der Metadatenbezeichner und -beschriftungen wird in eine eigene Funktion ausgelagert, da die Informationen an mehreren Stellen gebraucht werden.

Die Funktion `get_string_metas()` gibt ein Array von Arrays zurück, von denen jedes einen `key` und ein `label` enthält. Der `key` wird als Name des Metadatums genutzt, mit dem es in der Datenbank abgespeichert wird, und das `label` ist die sichtbare Beschriftung, die bei der Bearbeitung im Editor oder der Anzeige der Metadaten eingefügt wird.

In Listing 24.6 finden Sie den Code, der sich um die Registrierung der Metadaten aus diesem Array kümmert.

```
function register_meta() {
 foreach ( get_string_metas() as $meta ) {
```

```
  $meta_key = $meta['key'] ?? null;
  if ( empty( $meta_key ) ) {
   continue;
  }

  register_post_meta(
   'talk',
   $meta_key,
   [
    'single'       => true,
    'show_in_rest' => true,
   ]
  );
 }
}
add_action( 'init', __NAMESPACE__ . '\\register_meta' );
```

Listing 24.6 In der Funktion »register_meta()« werden die Metadaten in WordPress registriert, sodass sie im Block-Editor genutzt und verändert werden können.

1. Mit `foreach( get_string_metas() as $meta )` wird das Array durchlaufen, das in Listing 24.5 definiert und von der Funktion zurückgegeben wurde.
2. Innerhalb der Schleife wird geprüft, ob es einen `key`-Eintrag für das Metadatum gibt, und andernfalls wird direkt der nächste Schleifendurchlauf gestartet.
3. Mit `register_post_meta()` wird eine Post-Meta registriert. Der erste Parameter ist der Bezeichner des Inhaltstyps, dem das Metadatum zugewiesen sein soll (das ist der Wert, der in Listing 24.3 als erster Parameter an `register_post_type()` übergeben wurde). Der zweite Parameter ist der Schlüssel des Metadatums, über den die Werte aus der Datenbank ausgelesen werden können, und als dritter Parameter werden in einem Array nähere Informationen über das Metadatum angegeben, nämlich dass der Metawert aus einem einzelnen Wert besteht (`'single' => true,`) und dass die Post-Meta in der REST-API sichtbar sein soll.

4. Abschließend wird die Funktion `register_meta()` an den `init`-Action-Hook gehängt, und damit wird die Registrierung der Post-Metas vorgenommen.

Wenn Sie nun den REST-Endpunkt für die Vorträge aufrufen (*example.com/wp-json/wp/v2/talks*) und bereits einen Vortrag veröffentlicht haben, sollten Sie in der JSON-Antwort der REST-API die Metadaten unter dem Punkt `meta` finden, der für jeden Eintrag existieren müsste (siehe Abbildung 24.3). Daran können Sie erkennen, dass die Registrierung der Metadaten wie gewünscht funktioniert hat.

Im nächsten Abschnitt werden Sie den Editor anpassen, sodass die Metadaten für einen Vortrag bearbeitet werden können.



Abbildung 24.3 In der REST-API-Antwort für einen Vortrag sind bereits die registrierten Post-Meta-Einträge zu sehen (Screenshot aus dem Firefox-Browser, der JSON-Antworten etwas schöner darstellt als beispielsweise Chrome).

24.4 Panel in der Editor-Seitenleiste zur Bearbeitung der Metadaten

Die Seitenleiste rechts im Editor besteht aus zwei Ansichten: der für das gesamte Dokument, in der Sie beispielsweise die Option zum Veröffentlichen oder für den Permalink finden, und der für einen im Editor ausgewählten Block. Der Dokumentenbereich lässt sich an bestimmten Stellen erweitern (durch sogenannte *SlotFills*; eine Liste aller SlotFills finden Sie in der Dokumentation *https://developer.wordpress.org/block-editor/reference-guides/slotfills/*), und es können auch komplett neue Bereiche (*Panels*) hinzugefügt werden. Diese Variante werden Sie jetzt nutzen, um einen Bereich mit Formularfeldern zu erstellen, die zur Bearbeitung der Metadaten verwendet werden können.

Um die Entwicklung komfortabler zu gestalten, ist noch eine kleine Vorarbeit notwendig.

24.4.1 Das @wordpress/scripts-Paket installieren

Der Block-Editor ist mit dem JavaScript-Framework *React* erstellt worden (*https://reactjs.org/*). Das bedeutet, dass Sie unterschiedliche Dinge beachten müssen, wenn Sie für den Block-Editor entwickeln, Sie sollten zum Beispiel nicht einfach direkt per `document.querySelector()` ein DOM-Element ansprechen und verändern wollen. Für alle Anpassungen, die vorgenommen werden können, gibt es Funktionen, die entweder von WordPress oder direkt von React kommen.

Das macht die Entwicklung für den Block-Editor nicht gerade einfacher, wenn Sie sich bisher nicht mit React oder vergleichbaren

Frameworks auseinandergesetzt haben, da einige Konzepte doch etwas anders sind als in normalem JavaScript. Damit zumindest die Einrichtung für die Entwicklung für den Editor möglichst unkompliziert ist, gibt es das npm-Paket `@wordpress/scripts` (*https://www.npmjs.com/package/@wordpress/scripts*), das viele unterschiedliche Funktionen mitbringt, die Sie bei der Entwicklung unterstützen. Erstellen Sie im Plugin-Ordner eine Datei *package.json* und füllen Sie sie mit dem Inhalt aus Listing 24.7.

```
{
  "name": "wh-talks",
  "version": "1.0.0",
  "scripts": {
    "build": "wp-scripts build",
    "start": "wp-scripts start",
    "lint:js": "wp-scripts lint-js"
  },
  "license": "GPL-2.0-or-later",
  "devDependencies": {
    "@wordpress/scripts": "^30.4.0"
  }
}
```

Listing 24.7 Die »package.json« wird von npm genutzt und enthält unter anderem die npm-Pakete, die für die Entwicklung notwendig sind (»devDependencies«).

In der *package.json* werden die Metadaten für Pakete hinterlegt, die auf npmjs.org veröffentlicht werden (wie `@wordpress/scripts`). Eigentlich müssten Sie Name, Version und Lizenz hier nicht angeben, da Sie das Plugin nicht als npm-Paket veröffentlichen werden. Wichtig ist der Bereich `"scripts"`, in dem Scripts definiert werden, die Sie von der Kommandozeile mit `npm run scriptname` ausführen können. `wp-scripts` ist ein Befehl, der vom `@wordpress/scripts`-Paket mitgebracht wird. Dessen Unterbefehle `build` und `start` kümmern sich um die Verarbeitung des JavaScript-Codes, den Sie gleich schreiben werden, damit er vom WordPress-Plugin eingebunden werden kann. Mit `start` wird ein Prozess gestartet, der die Dateien auf Änderungen überwacht und diese

direkt verarbeitet, also praktisch für die Entwicklung, und `build` baut die Dateien für den Produktiveinsatz, ohne Dateiänderungen zu überwachen. Mit dem Befehl `wp-scripts lint-js` wird geprüft, ob das JavaScript den Coding-Standards von WordPress folgt, und mit `wp-scripts format` werden die Coding-Standards angewendet – standardmäßig für JavaScript-, JSON-, TypeScript- und YAML-Dateien aktiviert.

Der zweite wichtige Bereich in dieser *package.json* ist `"devDependencies"` mit dem Eintrag `@wordpress/scripts`. Damit geben Sie an, dass das Paket `@wordpress/scripts` installiert werden soll in einer Version, die nach *Semantic Versioning* kompatibel mit der Version 30.4.0 ist (mehr zum Thema Versionierung lesen Sie in Abschnitt 22.1.5, »Versionierung«). Das bedeutet: Wird eine Version 30.5.0 oder 30.4.1 veröffentlicht, wird diese installiert, nicht aber eine Version 31.0.0, da diese nicht komplett rückwärtskompatibel zur Version 30 ist und deshalb Dinge kaputtmachen könnte.

Wenn Sie jetzt in der Kommandozeile in das Plugin-Verzeichnis wechseln und `npm install` ausführen (siehe Abbildung 24.4), sollten Abhängigkeiten installiert werden und danach ein *node_modules*-Ordner und eine *package-lock.json* im Plugin-Ordner existieren. *node_modules* beinhaltet die installierten Abhängigkeiten, und wenn Sie einen Blick hineinwerfen, werden Sie sehen, dass darin deutlich mehr ist als nur ein Verzeichnis für `@wordpress/scripts`.

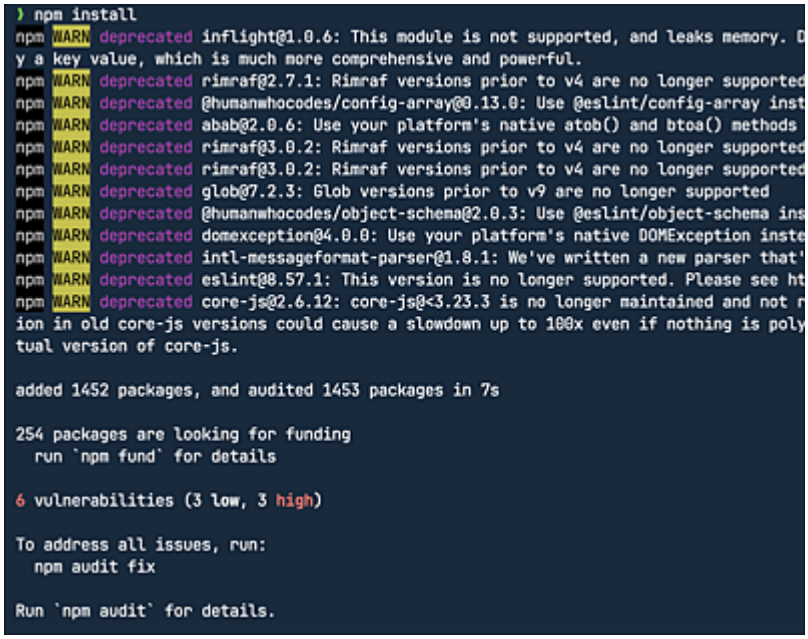


Abbildung 24.4 Nach dem Ausführen von »npm install« werden einige Meldungen und Warnungen angezeigt, aber die Installation läuft erfolgreich durch.

Das liegt daran, dass auch dieses Paket verschiedene Abhängigkeiten in seiner `package.json` angibt, die dann bei der Installation mit installiert werden, genauso wie die Abhängigkeiten dieser Abhängigkeiten – und so weiter, bis alles installiert ist.

Info: Das Schweizer Taschenmesser für die Block-Editor-Entwicklung: @wordpress/scripts

Mit dem Paket `@wordpress/scripts` können Sie eigentlich alles machen, was Sie bei der Entwicklung eines Blocks oder einer anderen Anpassung am Block-Editor benötigen. Sie können Ihren geschriebenen JavaScript-Code kompilieren, sodass er im Browser lauffähig ist, Sie können SASS-Styles in CSS umwandeln, die Einhaltung von Coding-Standards überprüfen, Tests laufen lassen und mehr. Und für all das bringt es bereits eine Standardkonfiguration mit, die Sie bei Bedarf anpassen können.

Weil sich dieses eine Paket darum kümmert, die Abhängigkeiten aufzulösen, die für diese ganzen Funktionalitäten notwendig sind, müssen Sie sich nicht mehr direkt um das Installieren der unterschiedlichen Abhängigkeiten kümmern und – sehr wichtig – nicht um die Konfiguration, da diese zumindest am Anfang ziemlich komplex und zeitaufwendig sein kann. Mehr über die ganzen Möglichkeiten, die das Paket mitbringt, finden Sie auf *https://www.npmjs.com/package/@wordpress/scripts*.

Nachdem Sie das Paket erfolgreich installiert haben, kann es mit der Entwicklung weitergehen.

24.4.2 Das Panel erstellen

Die `wp-scripts build`- und `wp-scripts start`-Befehle suchen als Einstiegspunkt zunächst nach *block.json*-Dateien innerhalb des Plugin-Verzeichnisses, die seit WordPress 5.8 der empfohlene Weg sind, um die Metadaten eines Blocks zu definieren und auch die Pfade zur JavaScript-Datei anzugeben. Diese Angabe nutzt *wp-scripts* als Einstiegspunkte, was bedeutet, dass die so angegebenen Dateien von den Scripts kompiliert werden. Wenn keine *block.json*-Dateien vorhanden sind, wird die *src/index.js* als Einstiegspunkt genommen.

src/block.json-Datei und src/index.js erstellen

Nun ist das Panel für die Seitenleiste kein Block, allerdings ist es auch nicht möglich, die Angabe der Einstiegspunkte zu mischen, also *block.json*-Dateien und eine JavaScript-Datei. Und da später noch zwei Blöcke angelegt werden, die über *block.json*-Dateien konfiguriert werden, legen Sie im nächsten Schritt eine Pseudo-*block.json*-Datei an, um die Datei *src/index.js* als Datei anzugeben,

die von den Scripts verarbeitet werden soll. Legen Sie dafür das Verzeichnis `src` im Plugin-Ordner an und darin die beiden Dateien *block.json* und *index.js*. Die *index.js* kann erst mal leer bleiben, in die *block.json* schreiben Sie den Code aus Listing 24.8.

```
{
    "apiVersion": 3,
    "name": "wh-talks/editor-modifications",
    "editorScript": "file:./index.js"
}
```

Listing 24.8 Mit »editorScript« geben Sie an, welche Datei das Editor-Script für den Block (oder hier eher für die Komponente) enthält. Der Name ist später für die Übersetzung wichtig.

Die API-Version wird angegeben, da WordPress den Editor als iFrame lädt, wenn alle Blöcke die Version 3 unterstützen. Auch wenn das hier kein Block wird, soll es natürlich nicht der Teil sein, an dem das scheitert. Und auch der Name ist wichtig, trotzdem es kein Block ist, um später Übersetzungen laden zu können. Der Name setzt sich zusammen aus einem Namespace und dem Block-Namen oder hier `editor-modifications`. `editorScript` gibt an, welche Datei im Editor geladen werden soll, als Wert übergeben Sie mit `file:./index.js` den Pfad zur leeren *index.js*-Datei. Wenn Sie nun in der Kommandozeile den Befehl `npm run start` ausführen, sollte ein `build`-Verzeichnis mit den drei Dateien *block.json*, *index.asset.php* und *index.js* erstellt werden. Die *index.js* beinhaltet das Kompilierungsergebnis aus der *src/index.js* und ist die Datei, die im Editor eingebunden werden wird. Die *index.asset.php* ist eine Datei mit zusätzlichen Informationen über die Datei bzw. die Inhalte, sie beinhaltet ein Array mit Abhängigkeiten sowie einen Versions-String. Die *block.json* wurde einfach kopiert und enthält denselben Inhalt.

Script in den Editor einbinden

Auch wenn das Script bisher nichts macht, schadet es nicht, es direkt in den Editor einzubinden. Erstellen Sie dafür eine *inc/blocks.php* und fügen Sie den Code aus Listing 24.9 ein.

```
<?php
namespace WH\Talks;
function register_blocks() {
 register_block_type(
  __DIR__ . '/../build'
 );
}
add_action( 'init', __NAMESPACE__ . '\\register_blocks' );
```

Listing 24.9 Der Aufruf von »register_block_type()« sorgt dafür, dass das Script eingebunden wird.

Innerhalb von `register_blocks()` wird die Funktion `register_block_type()` aufgerufen, der als Parameter der Pfad zum *build*-Ordner mit der *block.json*-Datei übergeben wird.

Im Hintergrund kümmert sich die Funktion unter anderem darum, dass das Editor-Script eingebunden wird. Die Funktion wird an den `init`-Action-Hook angehängt.

Jetzt muss die Datei noch in die *wh-talks.php* eingebunden werden, bevor es mit der Erstellung des Panels losgehen kann. Fügen Sie dazu die folgende Zeile am Ende der Datei ein.

```
require_once 'inc/blocks.php';
```

Panel-Script erstellen

Nachdem alle Vorbereitungen getroffen wurden, kann jetzt mit der Umsetzung des Panels begonnen werden. Ähnlich wie bei der *wh-talks.php* soll die *src/index.js* übersichtlich bleiben, erstellen Sie deshalb eine *src/sidebar-panel.js*, die den eigentlichen Code für das Panel enthalten wird. Um den Inhalt dieser Datei in die *src/index.js* zu bekommen, reicht die Zeile aus dem folgenden Listing. Bei dem

`import`-Statement muss die Dateiendung nicht mit angegeben werden.

```
import './sidebar-panel';
```

Zur Erstellung des Panels ist einiges an Code nötig, weshalb der Code in kleinere Blöcke aufgeteilt und erklärt wird. Der ganze Code gehört in die *src/sidebar-panel.js*, und den ersten Abschnitt finden Sie in Listing 24.10.

```
import { registerPlugin } from '@wordpress/plugins';
import { PluginDocumentSettingPanel } from '@wordpress/editor';
import { useEntityProp } from '@wordpress/core-data';
import { TextControl } from '@wordpress/components';
import { __ } from '@wordpress/i18n';

registerPlugin( 'wh-talks-metadata-panel', {
 render: () => {
  if ( 'talk' !== wp.data.select( 'core/editor' ).getCurrentPostType() ) {
   return null;
  }
  const [ meta, setMeta ] = useEntityProp( 'postType', 'talk', 'meta' );

  const eventName = meta?.wh_talks_event_name,
   language = meta?.wh_talks_language,
   duration = meta?.wh_talks_duration,
   videoLink = meta?.wh_talks_video_link,
   slidesLink = meta?.wh_talks_slides_link;

  const updateMeta = ( metaKey, newValue ) => {
   setMeta( { ...meta, [ metaKey ]: newValue } );
  };
```

Listing 24.10 Anfang der »src/sidebar-panel.js«, in der das Panel registriert wird

1. Am Anfang der Datei werden alle Komponenten importiert, die im Verlauf der Datei verwendet werden.
2. Mit dem Aufruf von `registerPlugin()` wird ein Block-Editor-Plugin erstellt. Der erste Parameter ist der Name des Plugins, der zweite ein Objekt, in dem die Eigenschaften des Plugins festgelegt werden. In `render` wird die Darstellung des Plugins umgesetzt.

3. Innerhalb von `render` wird zunächst geprüft, ob der aktuell angezeigte Inhaltstyp `talk` entspricht, damit das Panel nicht auch für Beiträge und Seiten angezeigt wird. Im nächsten Schritt wird der Hook `useEntityProp()` genutzt, der unscheinbar aussieht, aber ziemlich mächtig ist, er erlaubt nämlich auf recht einfache Weise die Veränderung von Werten beispielsweise eines Beitragsobjekts. Wird der Hook aufgerufen, wird als erster Parameter die Art der Entity übergeben, für die eine Eigenschaft verändert werden soll (hier also ein Inhaltstyp), der zweite Parameter ist der Entity-Typ, nämlich `talk`, also der spezifische Inhaltstyp, und der dritte Parameter ist die Eigenschaft, die verändert werden soll. Und hier sind die Werte möglich, die Sie vorhin in der JSON-Antwort beim Aufruf des REST-Endpunkts für Vorträge gesehen haben, also zum Beispiel `meta`, um auf die Metadaten zugreifen zu können, die in der REST-API angezeigt werden. Als Ergebnis des Hook-Aufrufs bekommen Sie ein Array zurück, in dem der erste Eintrag der aktuelle Wert der Eigenschaft ist (in diesem Fall ein Objekt der Metadaten) und der zweite Eintrag eine Setter-Funktion, mit der Sie den Wert der Eigenschaft aktualisieren können. Diese Einträge machen Sie über die Konstanten `meta` und `setMeta` zugänglich.
4. Im nächsten Schritt werden die Werte aus dem `meta`-Objekt in einzelne Konstanten gespeichert. Durch das `?` vor dem Punkt bei dem Ansprechen einer Objekteigenschaft wird verhindert, dass ein Fehler geworfen wird, wenn ein Eintrag im `meta`-Objekt nicht vorhanden ist, stattdessen wird dann `undefined` als Wert zurückgegeben.
5. In `updateMeta` speichern Sie eine Funktion, die über den Aufruf von `setMeta()` ein Metadatum aktualisiert. Dafür werden der Funktion der Meta-Key und der neue Wert für diesen Schlüssel übergeben, und `setMeta()` erhält als Parameter ein Objekt, das

alle Elemente aus `meta` mit dem aktualisierten Wert für den übergebenen `metaKey` enthält (`{ ...meta, [ metaKey ]: newValue }` nutzt die sogenannte Spread-Syntax, *https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/Spread_syntax*, mit der in diesem Fall alle Elemente aus `meta` in dem Objekt gespeichert werden und der Wert für den Eintrag, der dem Wert von `metaKey` entspricht, aktualisiert wird. Die eckigen Klammern sind notwendig, damit der in `metaKey` gespeicherte Wert als Objektschlüssel genutzt wird und nicht der String `metaKey`).

Weiter geht es mit dem Code in Listing 24.11, den Sie einfach direkt hinter den Code aus Listing 24.10 einfügen.

```
return (
 <PluginDocumentSettingPanel
  name='wh-talks-meta-panel'
  title={ __( 'Talk meta data', 'wh-talks' ) }
  icon='megaphone'
 >
  <TextControl
   label={ __(
    'Name of event where you held the talk',
    'wh-talks'
   ) }
   value={ eventName }
   onChange={ ( newValue ) =>
    updateMeta( 'wh_talks_event_name', newValue )
   }
  />
  <TextControl
   label={ __( 'Language', 'wh-talks' ) }
   value={ language }
   onChange={ ( newValue ) =>
    updateMeta( 'wh_talks_language', newValue )
   }
  />
  <TextControl
   label={ __( 'Duration', 'wh-talks' ) }
   value={ duration }
   onChange={ ( newValue ) =>
    updateMeta( 'wh_talks_duration', newValue )
   }
  />
  <TextControl
   label={ __( 'Video link', 'wh-talks' ) }
```

```
      value={ videoLink }
      type='url'
      onChange={ ( newValue ) =>
       updateMeta( 'wh_talks_video_link', newValue )
      }
     />
     <TextControl
      label={ __( 'Slides link', 'wh-talks' ) }
      value={ slidesLink }
      type='url'
      onChange={ ( newValue ) =>
       updateMeta( 'wh_talks_slides_link', newValue )
      }
     />
    </PluginDocumentSettingPanel>
   );
  },
} );
```

Listing 24.11 Der zweite Teil der »src/sidebar-panel.js« kümmert sich um die Ausgabe der Felder, mit denen Benutzer*innen die Metawerte verändern können.

1. Innerhalb von `return` wird definiert, was die Benutzer*innen sehen.
2. Die `PluginDocumentSettingPanel`-Komponente sorgt dafür, dass ein neues Panel in der Seitenleiste erstellt wird. Als Attribute werden ein interner Bezeichner (`name`), der sichtbare Titel des Panels (`title`) sowie ein Icon (`icon`) übergeben, das neben dem Titel angezeigt wird (hier wird dasselbe Dashicon genutzt wie für das Custom-Post-Type-Icon). Um den Titel übersetzbar zu machen, wird eine der JavaScript-Übersetzungsfunktionen von WordPress genutzt, die derselben Syntax folgen wie die PHP-Übersetzungsfunktionen.
3. Alles innerhalb dieser Komponente wird als Inhalt des Panels angezeigt.
4. Mit `TextControl` wird jetzt für jeden der Metawerte ein Textfeld angezeigt, dessen Beschriftung in `label` angegeben wird. Der Wert (`value`) ist der Wert der passenden Konstanten, in der oben die Werte aus dem `meta`-Objekt gespeichert wurden, und bei

einer Änderung des Feldwerts (`onChange`) wird die `updateMeta()`-Funktion aufgerufen, der der entsprechende Meta-Key und der neue Wert übergeben werden.

5. Am Ende werden alle Tags und Klammern wieder geschlossen.

Nachdem Sie die Datei gespeichert haben und das Script kompiliert wurde (wenn Sie `npm run start` parallel laufen haben, müssen Sie dazu nur die Datei abspeichern, wenn nicht, führen Sie den Befehl auf der Kommandozeile aus, sodass die weiteren Änderungen automatisch kompiliert werden), sollten Sie in der Einzelansicht eines Talk-Eintrags rechts ein neues Panel in der Seitenleiste sehen mit Textfeldern für die Metawerte (siehe Abbildung 24.5). Wenn Sie Daten in die Textfelder eingeben und den Talk-Post aktualisieren oder speichern, sollten die Werte ebenfalls gespeichert werden und nach dem Neuladen des Editors immer noch vorhanden sein.

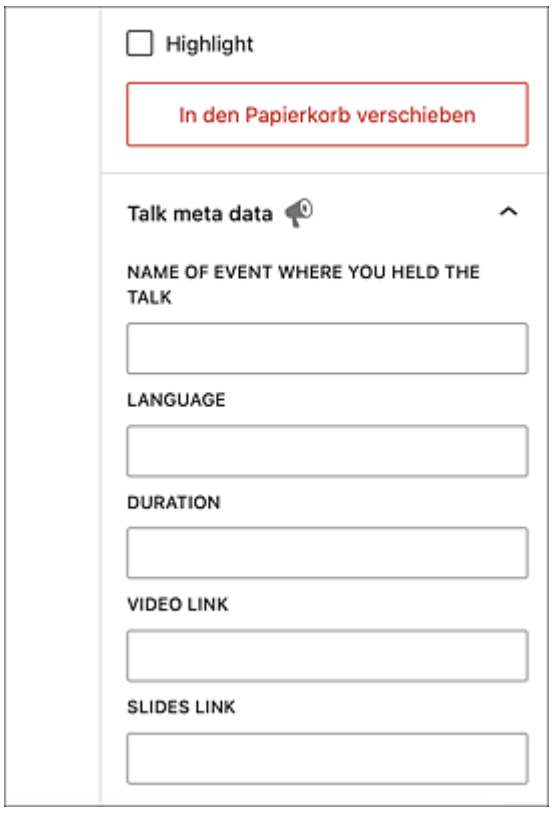


Abbildung 24.5 Das neue Panel in der Seitenleiste eines Talk-Eintrags für die Änderung der Metadaten

Hinweis: JSX, eine Syntaxerweiterung für JavaScript

Vielleicht haben Sie sich bei dem Code für das Panel in der Seitenleiste über die HTML-Tag-ähnliche Syntax gewundert, mit der die Eingabefelder und das Panel angegeben werden konnten (`<TextControl/>`). Dabei handelt es sich um *JSX*, eine Syntaxerweiterung für JavaScript (*https://reactjs.org/docs/introducing-jsx.html*), die übersichtlicheren und verständlicheren Code ermöglicht. Ohne JSX würde eine `TextControl`-Komponente wie folgt ausgegeben:

```
React.createElement(
 'TextControl',
 {
  label: __( 'Duration', 'wh-talks' ),
  value: duration,
  onChange: ( newValue ) => updateMeta( 'wh_talks_duration', newValue )
 }
);
```

Durch die Kompilierung des Codes mit `wp-scripts` wird das JSX im Hintergrund in normales JavaScript umgewandelt.

Panel-Script nicht im Widgets-Screen einbinden

Das Script für die Erstellung des Panels soll nicht im Widgets-Screen geladen werden, weil WordPress sonst eine PHP-Notice wegen einer Script-Abhängigkeit werfen würde und weil es dort sowieso nicht notwendig ist. Dafür fügen Sie den Code aus Listing 24.12 in die *inc/blocks.php* ein.

```
function remove_editor_modifications_script_from_widgets_screen() {
 global $block_editor_context;
 $context_name = $block_editor_context->name ?? null;
 if ( 'core/edit-widgets' !== $context_name ) {
```

```
  return;
 }
 wp_deregister_script( 'wh-talks-editor-modifications-editor-script' );
}
add_action( 'enqueue_block_editor_assets', __NAMESPACE__ .
'\\remove_editor_modifications_script_from_widgets_screen' );
```

Listing 24.12 Code zum Entfernen des Scripts mit den Editormodifikationen im Widgets-Screen

In der Funktion `remove_editor_modifications_script_from_widgets_screen()` wird der Editor-Kontext überprüft, und für alle Kontexte außer dem Widgets-Screen (`core/edit-widgets`) wird die Funktion frühzeitig beendet. Wenn der Widgets-Screen angezeigt wird, wird mit `wp_deregister_script()` das Panel-Script entfernt. `wh-talks-editor-modifications-editor-script` ist dabei das automatisch von WordPress generierte Handle des Scripts, zusammengesetzt aus dem Block-Namen (`wh-talks/editor-modifications`) und der Art des Scripts (`editorScript`).

24.5 Editor-Blöcke erstellen

Als Nächstes werden die beiden Blöcke zur Ausgabe der Metadaten erstellt. Wie schon kurz erwähnt, soll der eine Block einfach eine Liste mit den Metadaten ausgeben, die ausgefüllt wurden, wobei als Beschriftung die Labels aus der `get_string_metas()`-Funktion genutzt werden. Der zweite Block ermöglicht die Ausgabe eines einzelnen Metawerts mit einem beliebigen Label. Sowohl die Auswahl des Meta-Keys, zu dem der Wert angezeigt werden soll, als auch die Festlegung des Labels passieren dabei über Einstellungen in der Block-Seitenleiste.

24.5.1 Array mit Infos zu den Metadaten im Editor verfügbar machen

Bevor Sie mit der Programmierung der Blöcke starten, wird eine kleine Vorbereitung getroffen: Für beide Blöcke werden die Namen der Metawerte benötigt sowie die Beschriftungen, die den Benutzer*innen in der Liste aller Metawerte oder bei der Option zur Auswahl des Meta-Keys angezeigt werden. Um diesen Code nicht an mehreren Stellen zu definieren und dann auch an mehreren Stellen pflegen zu müssen, greifen Sie hier wieder auf das Array aus `get_string_metas()` zurück und fügen die Werte des Arrays als Inline Script ein, sodass es in den Block-Scripts zur Verfügung steht. Schreiben Sie dazu den Code aus Listing 24.13 in die *inc/blocks.php*.

```
function enqueue_editor_assets() {
 wp_localize_script(
  'wh-talks-meta-list-editor-script',
  'whTalksObject',
  [
   'metas' => get_string_metas(),
  ]
```

```
 );
}
```

Listing 24.13 Mit »wp_localize_script()« wird ein Inline Script vor dem Script ausgegeben, dessen Handle als erster Parameter angegeben wird.

In der Funktion `enqueue_editor_assets()` wird mit `wp_localize_script()` im HTML ein Inline Script eingebunden, und zwar vor dem Script, dessen Handle als erster Parameter übergeben wird. Durch die Nutzung der *block.json* zur Konfiguration des Blocks und die Angabe des Dateipfads in `editorScript` wird dieses Handle von WordPress automatisch generiert und besteht aus dem Block-Namen (`wh-talks/meta-list`, den werden Sie gleich erstellen), in dem Slashs durch einen Bindestrich ersetzt werden, sowie einem Suffix, das die Art des Scripts markiert (für `editorScript` ist das Suffix beispielsweise `-editor-script`). Daraus ergibt sich das Handle `wh-talks-meta-list-editor-script`. Der zweite Parameter ist der Name der JavaScript-Variablen, in der ein Objekt mit den Daten des Arrays gespeichert wird, das als dritter Parameter angegeben wird.



Abbildung 24.6 »wp_localize_script()« fügt ein Inline Script vor dem Script ein, dessen Handle als erster Parameter übergeben wird.

Im Quellcode sieht das dann aus wie in Abbildung 24.6 bzw. im folgenden Listing.

```
var whTalksObject = {"metas":[{"key":"wh_talks_event_name","label":"Event"},
{"key":"wh_talks_language","label":"Language"},
{"key":"wh_talks_duration","label":"Duration"},
{"key":"wh_talks_video_link","label":"Video"},
{"key":"wh_talks_slides_link","label":"Slides"}]};
```

24.5.2 Block mit Liste der Metawerte erstellen

Jetzt sind Sie bereit zur Erstellung des ersten Blocks. Legen Sie dazu im *src*-Verzeichnis einen *blocks*-Ordner an und darin einen Ordner *meta-list*. Hier erstellen Sie zwei neue Dateien: eine *block.json* und eine *index.js*, sodass die Ordnerstruktur aussieht wie in Abbildung 24.7.

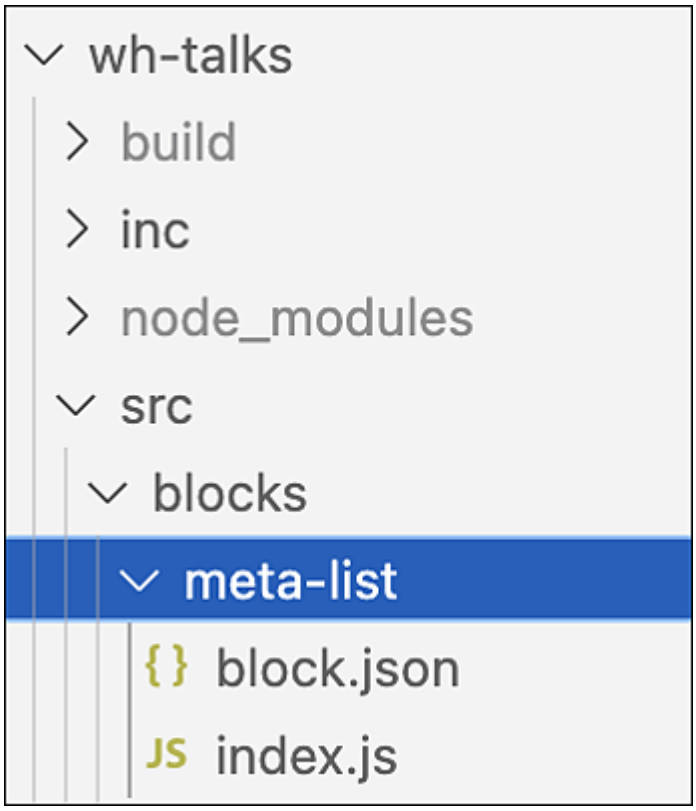


Abbildung 24.7 Die Verzeichnisstruktur für den ersten Block

Registrierung des Blocks in der blocks.php

Damit ein Block im Block-Editor ausgewählt werden kann, muss er registriert werden. Das geschieht mit der bereits bekannten Funktion `register_block_type()`, erweitern Sie dafür die Funktion `register_blocks()` in der *inc/blocks.php* um den Code aus Listing 24.14.

```
register_block_type(
 __DIR__ . '/../build/blocks/meta-list',
 [
  'render_callback' => __NAMESPACE__ . '\\render_meta_list_block',
 ]
);
```

Listing 24.14 Die Registrierung eines Blocks wird mit der »register_block_type()«-Funktion vorgenommen.

An `register_block_type()` wird als erster Parameter der Pfad zum Verzeichnis übergeben, in dem die *block.json*-Datei für den Block liegen wird (WordPress sucht in dem Verzeichnis automatisch nach einer *block.json*). Der zweite Parameter ist ein Array mit weiteren Argumenten, hier wird nur ein `render_callback` angegeben, also eine Funktion, die für die Anzeige des Blocks im Frontend aufgerufen wird und das Markup zurückgeben muss (seit WordPress 6.1 können Sie alternativ auch in der *block.json* über `render` einen Pfad zu einer Datei übergeben, die sich um die Ausgabe kümmert).

Diese Funktion erstellen Sie auch schon, damit WordPress keinen Fehler wirft, geben aber erst mal nur einen Platzhalterwert zurück (siehe Listing 24.15).

```
function render_meta_list_block( $attributes, $block_content ) {
 return 'Meta-List-Block';
}
```

Listing 24.15 Die Funktion zur Ausgabe im Frontend gibt zunächst nur einen Dummy-Inhalt zum Testen zurück.

Block-Konfiguration in der block.json

Sobald die Registrierung des Blocks vorgenommen ist, geht es mit der Konfiguration weiter. Wurde diese früher mit in dem JavaScript-Code des Blocks vorgenommen, ist seit WordPress 5.8 der empfohlene Weg die Konfiguration über eine *block.json*-Datei. Der

große Vorteil gegenüber der Konfiguration in JavaScript ist, dass so auch in PHP auf die Konfiguration des Blocks zugegriffen werden kann, was beispielsweise eine Optimierung beim Laden von CSS und Scripts für den Block im Frontend ermöglicht. Der Inhalt für die *block.json* ist in Listing 24.16 abgedruckt.

```
{
    "$schema": "https://schemas.wp.org/wp/6.7/block.json",
    "apiVersion": 3,
    "name": "wh-talks/meta-list",
    "version": "1.0.0",
    "title": "Talk metadata list",
    "icon": "megaphone",
    "description": "Display the talk metadata as list",
    "supports": {
        "html": false,
        "align": true
    },
    "textdomain": "wh-talks",
    "editorScript": "file:./index.js"
}
```

Listing 24.16 Die Konfigurationsdatei »block.json« für den Block mit der Liste der Metawerte eines Vortrags

1. Wie auch für die *theme.json* gibt es für die *block.json* ein JSON-Schema, das in *$schema* angegeben wird, sodass Editoren mit Unterstützung für JSON-Schema eine zur Datei passende Autovervollständigung anbieten.
2. Mit `apiVersion` wird die Version der Block-API angegeben, die für den Block genutzt werden soll. Version 3 wurde in WordPress 6.3 eingeführt, und die Neuerung dabei ist, wie vorhin schon angerissen, dass WordPress den Editor als iFrame lädt, wenn alle registrierten Blöcke mindestens die API-Version 3 nutzen.
3. `name` ist der Bezeichner des Blocks, der unter anderem für die HTML-Kommentare genutzt wird, um Blöcke im Editor zu

kennzeichnen. Er sollte aus einem Namespace und dem Namen des Blocks bestehen.

4. In `version` kann optional eine Version für den Block angegeben werden, die WordPress als Query-String für die Assets des Blocks nutzt. Ist es nicht angegeben, wird die Versionsnummer von WordPress verwendet.
5. Mit `title` geben Sie den Namen des Blocks an, der im Editor sichtbar ist und über den der Block im Inserter gefunden werden kann.
6. Über `icon` geben Sie das Block-Icon an, es können die Dashicons von WordPress genutzt werden.
7. `description` ist die für Benutzer*innen sichtbare Beschreibung des Blocks.
8. `supports` ermöglicht die Aktivierung oder Deaktivierung bestimmter Block-Editor-Funktionen, wie beispielsweise des HTML-Modus oder der Anzeige von Optionen zur Ausrichtung des Blocks.
9. Für `textdomain` geben Sie die Text-Domain des Plugins an.
10. `editorScript` enthält einen Pfad zu der Datei, die für die Block-Darstellung im Editor gebraucht wird.

Neben diesen Angaben sind einige weitere möglich, zum Beispiel das Angeben von Stylesheets oder Frontend-Scripts. Alle Möglichkeiten finden Sie auf der Seite zu Block-Metadaten im Block-Editor-Handbuch (*https://developer.wordpress.org/block-editor/reference-guides/block-api/block-metadata/*).

Block-Script erstellen

Nachdem die Konfiguration abgeschlossen ist, können Sie jetzt an das Block-Script gehen, die *index.js* im *src/blocks/meta-list*-Verzeichnis (siehe Listing 24.17).

```
/* global whTalksObject */
import { registerBlockType } from '@wordpress/blocks';
import { useBlockProps } from '@wordpress/block-editor';
import { useEntityProp } from '@wordpress/core-data';
import { __ } from '@wordpress/i18n';

registerBlockType( 'wh-talks/meta-list', {
 edit: () => {
  const [ meta ] = useEntityProp( 'postType', 'talk', 'meta' );

  let hasValues = false;

  return (
   <ul { ...useBlockProps() }>
    { whTalksObject.metas.map( ( metaInfo, i ) => {
     let metaValue = meta?.[ metaInfo.key ];
     if ( ! metaValue ) {
      return false;
     }

     if ( metaInfo.key.endsWith( '_link' ) ) {
      try {
       const url = new URL( metaValue );
       metaValue = (
        <a href='#' onClick={ ( e ) => e.preventDefault() }>
         { url.host }
        </a>
       );
      } catch {
       metaValue = <a>{ __(
        'This does not seem to be a valid URL',
        'wh-talks'
       ) }</a>;
      }
     }
     hasValues = true;
     return (
      <li key={ i }>
       <span className='wh-talks-meta-label'>
        { metaInfo.label }:
       </span>
       { ' ' }
       { metaValue }
      </li>
     );
    } ) }
    { ! hasValues && <li>{ __( 'Lists all talk meta data.', 'wh-talks' ) }
</li> }
```

```
    </ul>
   );
  }
} );
```

Listing 24.17 Die »src/blocks/meta-list/index.js« stellt den Block im Editor dar.

1. Zu Anfang wird in einem Kommentar angegeben, dass `whTalksObject` eine globale Variable ist, damit Editoren sich nicht über die vermeintliche nicht deklarierte Variable beschweren.
2. Danach werden, wie in der Datei für das Panel in der Seitenleiste, die benötigten Komponenten importiert.
3. `registerBlockType()` kümmert sich um die Registrierung des Blocks im JavaScript-Kontext. Da Sie vorhin bereits die serverseitige Registrierung in PHP vorgenommen und dort den Pfad zur *block.json* angegeben haben, reicht es hier aus, wenn der erste Parameter der `name`-Eigenschaft aus der *block.json* entspricht, damit WordPress die Verbindung herstellen kann und die Konfiguration für diesen Block verwendet wird. Der zweite Parameter, der an die Funktion übergeben wird, ist ein Objekt mit dem Code, der festlegt, was im Block-Editor angezeigt (`edit`).
4. Innerhalb der `edit`-Funktion wird zunächst über den bereits aus dem Panel-Code bekannten `useEntityProp()`-Hook das Objekt mit den Metadaten des Posts in `meta` gespeichert (Metadaten werden in dem Block nur angezeigt, deshalb braucht es die Setter-Funktion hier nicht, die von dem Hook auch zurückgegeben wird). Anschließend wird eine Variable erstellt, die als Flag dient und speichert, ob mindestens einer der anzuzeigenden Metadaten einen Wert enthält.

5. Im `return`-Statement wird das Markup für den Block zusammengebaut. Als Wrapper-Element des Blocks soll eine ungeordnete Liste fungieren (`ul`), die durch den Aufruf des `useBlockProps()`-Hooks markiert wird, damit der Block-Editor weiß, welches Element der Wrapper ist (diese Möglichkeit gibt es erst seit der API-Version 2, vorher wurde automatisch ein Wrapper-Element um den Block gelegt, wodurch das Markup im Editor deutlich unterschiedlich von dem im Frontend war).
6. Innerhalb des `ul`-Elements wird mit der `map()`-Methode das Array von Metainfos durchlaufen, das Sie mit `wp_localize_script()` in den Editor eingebunden haben. An die Callback-Funktion wird der Wert des aktuellen Array-Eintrags übergeben sowie der Array-Index.
7. In der Callback-Funktion wird als Erstes geprüft, ob zu dem Meta-Key (`metaInfo. key`) ein Eintrag in dem `meta`-Objekt vorliegt (das ist nur für Meta-Keys der Fall, die mit `register_post_meta()` registriert wurden). Ist das nicht der Fall, wird `false` zurückgegeben, was bei `map()` bedeutet, dass nichts ausgegeben und der nächste Durchlauf gestartet wird.
8. Die Werte aus den beiden URL-Feldern sollen so dargestellt werden, dass der Host der URL (also die Domain mit eventuellen Subdomains, aber ohne Protokoll und Pfad) sichtbar ist und auf die eingegebene URL verlinkt. Dafür wird geprüft, ob der Meta-Key auf `_link` endet, ob es sich also um eins der beiden Metadaten handelt. Ist das der Fall, wird versucht, mit dem Wert ein URL-Objekt zu erstellen. Der `try/catch`-Block ist notwendig, weil JavaScript einen Fehler wirft, sollte es sich nicht um eine URL handeln. Wenn eine gültige URL vorliegt, wird ein Linkelement mit dem Host als Linktext erstellt. Im Editor soll der Link nicht richtig verlinken,

damit die Benutzer*innen nicht versehentlich beim Auswählen des Blocks einen Link öffnen, also geben Sie als `href` ein `#`-Symbol an und verhindern im `onClick`-Event, dass beim Klicken auf den Link etwas passiert. Wenn der Metawert keine URL war (jetzt geht es im `catch`-Block weiter), wird als `metaValue` ein Linkelement ohne `href` gespeichert mit dem Hinweis als Linktext, dass es sich nicht um eine gültige URL handelt.

9. Nach der Prüfung auf Links und deren Verarbeitung wird `hasValues` auf `true` gesetzt, und ein Listenelement mit dem Label aus dem `metaInfo`-Objekt und dem Metawert wird zurückgegeben.
10. Nach der `map()`-Methode wird geprüft, ob keine Werte ausgegeben wurden (`! hasValues`), und in einem Listeneintrag die Funktion des Blocks beschrieben.

Wenn Sie nun in die Bearbeitungsansicht eines Vortrags wechseln, Metainformationen über das Panel in der Seitenleiste eingeben und den Block TALK METADATA LIST einbauen, sollte Ihnen eine Liste der Metawerte angezeigt werden (siehe Abbildung 24.8).

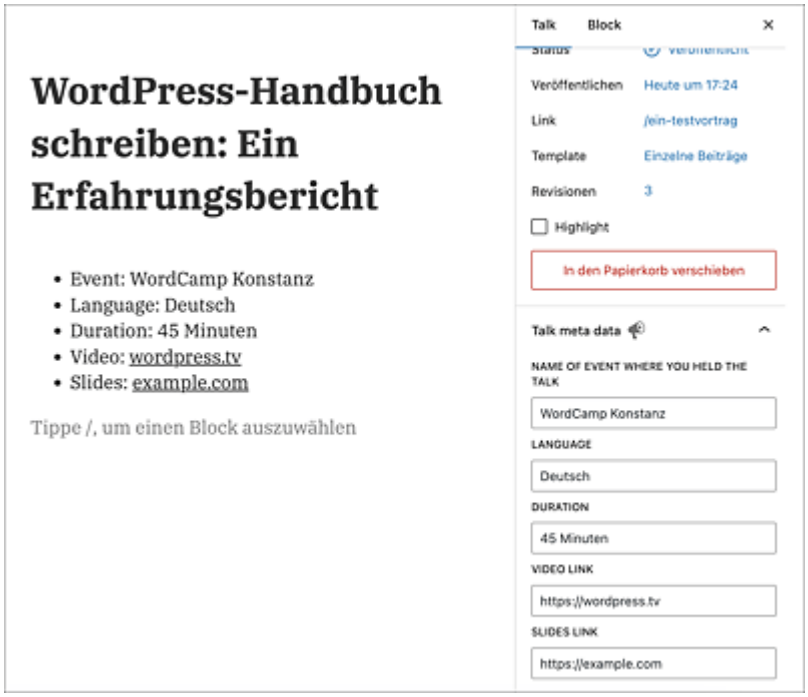


Abbildung 24.8 Der Block zur Darstellung der Vortrags-Metawerte in Aktion

Im Backend sieht also eigentlich alles gut aus für diesen ersten Block, nur wenn Sie ins Frontend wechseln, sehen Sie dort statt der Liste von Metadaten den String Meta-List-Block. Als Nächstes steht also die Frontend-Ausgabe an.

Frontend-Ausgabe des Blocks

Oft speichern Blöcke direkt Inhalte in die Datenbank oder haben zumindest Attribute, in denen Einstellungen des Blocks gespeichert werden, die dann als Teil des Block-HTML-Kommentars in der Datenbank landen. Der Block zur Auflistung der Metadaten hat weder Optionen noch direkt selbst einen Inhalt, da er nur Metawerte anzeigt, die unabhängig vom Inhalt existieren. Deshalb gibt es im JavaScript keine extra Funktion zur Speicherung des Blocks in der Datenbank, sodass lediglich der Block-Kommentar gespeichert wird.

Tipp: Dynamische statt statische Blöcke erstellen

Bei der Frontend-Ausgabe für Blöcke gibt es grob zwei Möglichkeiten: das komplette Markup direkt im Block-Editor generieren und in der Datenbank speichern oder kein/wenig Markup in der Datenbank speichern und das finale Markup in PHP zusammenbauen. Die statische Variante ist performanter als die dynamische, da für die Ausgabe nur das Markup aus der Datenbank geholt, und ausgegeben werden muss. Wenn Sie allerdings etwas komplexere Blöcke haben, bei denen es nicht unwahrscheinlich ist, dass Sie mal das Markup ändern möchten, das den Block im Frontend darstellt – und zwar auch für alle bereits eingesetzten Blöcke –, sind Sie mit dynamischen Blöcken besser bedient. Hierbei wird das Markup des Blocks nämlich in PHP beim Seitenaufruf erzeugt und kann daher für alle Vorkommen eines Blocks einfach angepasst werden.

Wenn Sie einen statischen Block nutzen, müssten Sie im Block-Script eine veraltete Version des Markups festlegen, damit WordPress schon eingebaute Blöcke mit dem alten Markup zuordnen und umwandeln kann. Um bestehende Blöcke in das neue Markup zu überführen, müssten Sie die Inhalte im Editor aufrufen und aktualisieren, die die Blöcke enthalten.

Jetzt wird die vorhin mit einer Dummy-Ausgabe angelegte Funktion `render_meta_list_block()` mit richtigem Inhalt gefüllt (siehe Listing 24.18).

```
function render_meta_list_block() {
 $markup = '';
 foreach ( get_string_metas() as $meta ) {
  $meta_key   = $meta['key'] ?? null;
  $meta_label = $meta['label'] ?? null;
  if ( empty( $meta_key ) || empty( $meta_label ) ) {
   continue;
  }
```

```
  $meta_value_markup = get_meta_value_markup( $meta_key );
  if ( empty( $meta_value_markup ) ) {
   continue;
  }

  $markup .= sprintf(
   '<li><span class="wh-talks-meta-label">%s:</span> %s</li>',
   $meta_label,
   $meta_value_markup
  );
 }

 if ( '' === $markup ) {
  return '';
 }

 $wrapper_attributes = get_block_wrapper_attributes();

 return "<ul {$wrapper_attributes}>$markup</ul>";
}
```

Listing 24.18 Die vollständige Funktion zur Ausgabe des Frontend-Markups für den Metadaten-Liste-Block

1. In der Funktion wird eine Variable mit einem leeren String angelegt. Anschließend wird das Array von Metainfos aus der `get_string_metas()`-Funktion durchlaufen, um hier in PHP eigentlich dasselbe Verhalten abzubilden wie im Editor mit JavaScript.
2. Innerhalb der Schleife werden die Werte der Array-Schlüssel `key` und `label` gespeichert, und es wird geprüft, ob für beides ein Wert vorliegt. Ist das nicht der Fall, wird der Schleifendurchlauf abgebrochen und mit dem nächsten weitergemacht.
3. Anschließend wird das Markup für den Meta-Key aus der Funktion `get_meta_value_markup()` besorgt, deren Erstellung gleich im Anschluss gezeigt wird.
4. Wenn der Markup-String aus der Funktion leer ist, wird der Schleifendurchlauf abgebrochen, andernfalls wird ein

Listenelement mit dem Wert und der Beschriftung erstellt und an den Wert von `$markup` angehängt.

5. Nach der Schleife wird geprüft, ob `$markup` leer ist, um dann einen leeren String zurückzugeben, andernfalls wird über die Funktion `get_block_wrapper_attributes()` ein String mit den Block-Wrapper-Attributen erzeugt, im Fall dieses Blocks ist das nur das `class`-Attribut mit dem Block-Klassennamen und gegebenenfalls im Editor eingetragenen zusätzlichen Klassen. Danach wird der String in ein `ul`-Element eingebaut und zwischen das öffnende und schließende `ul` das in der Schleife generierte Markup eingefügt. Das Gesamt-Markup wird dann über `return` zurückgegeben.

Jetzt fehlt nur noch die Funktion `get_meta_value_markup()`, deren Code Sie im folgenden Listing sehen.

```
function get_meta_value_markup( $meta_key ) {
 $meta_value = get_post_meta( get_the_ID(), $meta_key, true ) ?? null;
 if ( empty( $meta_value ) ) {
  return '';
 }

 if ( strrpos( $meta_key, '_link' ) !== strlen( $meta_key ) - strlen(
'_link' ) ) {
  return esc_html( $meta_value );
 }

 if ( 1 !== preg_match( '/^(http:|https:)?\/\//', $meta_value ) ) {
  return '';
 }

 $url  = esc_url( $meta_value );
 $host = parse_url( $url, PHP_URL_HOST );
 return sprintf(
  '<a href="%s">%s</a>',
  $url,
  $host
 );
}
```

Listing 24.19 Die Funktion gibt das Markup für den übergebenen Meta-Key zurück.

1. Der Funktion wird der Meta-Key (`$meta_key`) übergeben, zu dem sie den Wert oder das Markup (im Fall eines Links) zurückgeben soll.
2. Mit `get_post_meta()` wird am Anfang der Funktion der entsprechende Metawert zum Key für den aktuellen Post ermittelt. Als ersten Parameter erwartet die Funktion die Post-ID (`get_the_ID()` gibt diese zurück), als zweiten Parameter optional einen Meta-Key (sonst würden alle Metadaten zu dem Post zurückgegeben) und als dritten Parameter optional die Angabe, ob ein einzelner Wert zurückgegeben werden soll.
3. Wenn kein Metawert für den Key vorhanden ist, gibt die Funktion einen leeren String zurück.
4. Im nächsten Schritt wird geprüft, ob kein Link-Meta-Key angefragt wurde, um in dem Fall den von eventuell vorhandenem HTML befreiten Metawert zurückzugeben, da nichts weiter an dem String geändert werden muss.
5. Handelt es sich um einen Link, wird überprüft, ob der String mit `http://`, `https://` oder `//` anfängt, also zumindest oberflächlich die Voraussetzungen für eine URL erfüllt, und falls das nicht zutrifft, wird ein leerer String zurückgegeben.
6. Fängt der String wie eine URL an, wird er durch `esc_url()` geschickt, um alles zu entfernen, was in URLs nichts zu suchen hat, und anschließend wird mit der `parse_url()`-Funktion der Hostname ermittelt.
7. Abschließend wird mit `sprintf()` das Link-Markup gebaut und das Ergebnis zurückgegeben.

Wenn Sie jetzt einen Vortrags-Post mit dem Block und eingetragenen Metadaten im Frontend öffnen, sollten Sie etwas sehen wie in Abbildung 24.9.



Abbildung 24.9 Der Metadaten-Liste-Block im Frontend

24.5.3 Block zur Anzeige eines einzelnen Metawerts erstellen

Der zweite Block soll es ermöglichen, einen einzelnen Metawert anzuzeigen. Dafür wird es in der Seitenleiste des Blocks zwei Optionen geben, eine zur Auswahl des Meta-Keys, für den der Wert angezeigt werden soll, und eine für eine Beschriftung, die vor dem Wert angezeigt wird.

Registrierung des Blocks

Wie auch beim ersten Block steht am Anfang die Registrierung in der *blocks.php* an. Dazu erweitern Sie die Funktion `register_blocks()` um den Code aus Listing 24.20.

```
register_block_type(
 __DIR__ . '/../build/blocks/single-meta',
 [
  'render_callback' => __NAMESPACE__ . '\\render_single_meta_block',
```

```
 ]
);
```

Listing 24.20 Am Anfang der Block-Erstellung steht wie beim ersten Block die Registrierung in PHP.

Und ebenfalls wie beim ersten Block erstellen Sie als Nächstes eine Dummy-Callback-Funktion zur Darstellung des Blocks im Frontend (siehe Listing 24.21).

```
function render_single_meta_block( $attributes, $block_content ) {
 return 'Single meta block';
}
```

Listing 24.21 Erst mal tut es wieder eine Dummy-Funktion für die Ausgabe im Frontend.

Block-Konfiguration

Erstellen Sie im *src/blocks*-Verzeichnis einen Ordner *single-meta* und darin eine *block.json*. Diese Datei füllen Sie mit dem Code aus Listing 24.22.

```
{
    "$schema": "https://schemas.wp.org/wp/6.7/block.json",
    "apiVersion": 3,
    "name": "wh-talks/single-meta",
    "version": "1.0.0",
    "title": "Talk meta value",
    "icon": "megaphone",
    "description": "Display single talk meta value",
    "attributes": {
        "metaKey": {
            "type": "string",
            "default": "wh_talks_event_name"
        },
        "label": {
            "type": "string",
            "default": ""
        }
    },
    "supports": {
        "html": false,
        "align": true
    },
    "textdomain": "wh-talks",
```

```
    "editorScript": "file:./index.js"
}
```

Listing 24.22 Bei der Konfiguration des Blocks gibt es etwas Neues: Attribute.

Der Großteil der Datei kommt Ihnen von dem ersten Block stark bekannt vor, aber es gibt eine Neuerung: In `attributes` werden Attribute des Blocks angegeben, in denen Blöcke beispielsweise Werte von Einstellungen speichern oder aber auch Inhalte, die in HTML-Elementen vorliegen. Der Block bekommt die beiden Attribute `metaKey` und `label`, beide vom Typ `string`, und das `metaKey`-Attribut mit dem ersten Meta-Key aus dem Array in `get_string_metas()` als Standardwert. In `metaKey` wird der Wert der Option gespeichert, um den Meta-Key auszuwählen, für den der Metawert angezeigt werden soll, in `label` wird die Beschriftung aus einem Textfeld gespeichert.

Das Block-Script

Wie das Script für das Panel ist auch dieses Script wieder etwas länger und wird deshalb in zwei Abschnitte geteilt. Legen Sie eine *index.js* im *single-meta*-Ordner an und fügen Sie den Anfang des Block-Scripts aus Listing 24.23 ein.

```
/* global whTalksObject */
import { registerBlockType } from '@wordpress/blocks';
import { useBlockProps, InspectorControls } from '@wordpress/block-editor';
import { useEntityProp } from '@wordpress/core-data';
import { SelectControl, TextControl, PanelBody } from '@wordpress/components';
import { __ } from '@wordpress/i18n';

registerBlockType( 'wh-talks/single-meta', {
 edit: ( { attributes, setAttributes } ) => {
  const { metaKey, label } = attributes;
  const [ meta ] = useEntityProp( 'postType', 'talk', 'meta' );

  const currentMetaInfoEntry = whTalksObject.metas.find( ( metaInfo ) => {
   return metaInfo.key === metaKey;
  } );

  const selectOptions = whTalksObject.metas.map( ( metaInfo ) => {
```

```
 return {
  value: metaInfo.key,
  label: metaInfo.label,
 };
} );

let value = meta?.[ metaKey ];
if ( value && metaKey.endsWith( '_link' ) ) {
 try {
  const url = new URL( value );
  value = (
   <a href="#" onClick={ ( e ) => e.preventDefault() }>
    { url.host }
   </a>
  );
 } catch {
  value = __( 'No valid URL', 'wh-talks' );
 }
}
if ( ! value ) {
 /* translators: placeholder when no meta value exists. s = label for meta key */
 value = __( '»%s« meta value', 'wh-talks' ).replace(
  '%s',
  currentMetaInfoEntry.label
 );
}

let labelMarkup = '';
if ( label ) {
 labelMarkup = (
  <span className="wh-talks-meta-label">{ label }</span>
 );
}
```

Listing 24.23 Anfang des Block-Scripts

1. Die ersten Zeilen sind denen das anderen Block-Scripts ähnlich: Es wird auf die globale Variable hingewiesen, und die Komponenten werden importiert. Anschließend wird die `registerBlockType()`-Funktion aufgerufen mit dem Namen des Plugins und dem Objekt, das `edit` enthält.
2. An die `edit`-Funktion werden dieses Mal Elemente des Funktionsattributs übergeben, und zwar `attributes` und `setAttributes`, beides Elemente des Objekts, das als Attribut an die Funktion übergeben wird (mit der Syntax `{ attributes,`

`setAttributes }` werden die Elemente `attributes` und `setAttributes` aus einem Objekt angesprochen und direkt zugänglich gemacht). In `attributes` sind die Attribute des Blocks gespeichert, die in der *block.json* definiert wurden, mit `setAttributes` können sie verändert werden.

3. In der Funktion werden zunächst die Werte der beiden Attribute in die Konstanten `metaKey` und `label` gespeichert und, wie aus dem anderen Block bekannt, die Metawerte in `meta` bereitgestellt.
4. In `currentMetaInfoEntry` soll das Array mit `label` und `key` aus der `get_string_metas()`-Funktion gespeichert werden, für den der Block gerade den Wert anzeigt. Dafür wird auf dem Array in `whTalksObject.metas` die `find()`-Methode ausgeführt, bei der jedes Element eines Arrays an eine Funktion übergeben wird und der erste Wert zurückgegeben und damit in `currentMetaInfoEntry` gespeichert wird, für den die Funktion `true` zurückgibt. Um den Eintrag des aktuell gewählten Meta-Keys zu finden, wird in der Funktion geprüft, ob der `key`-Eintrag des gerade durchlaufenen Array-Elements dem Wert des `metaKey`-Block-Attributs entspricht.
5. Als Nächstes wird in `selectOptions` das Objekt gespeichert, das später an das `options`-Attribut der `SelectControl` übergeben wird, um die Auswahlmöglichkeiten darzustellen. Dafür wird die `map()`-Methode verwendet, die auf ein Array angewendet werden kann und für jeden Eintrag eine Funktion aufruft und aus den Rückgabewerten der Funktion ein neues Array erstellt, das dann in `selectOptions` gespeichert wird. Für das `options`-Attribut wird ein Array mit Objekten benötigt, wobei jedes Objekt eine Auswahlmöglichkeit darstellt und aus einem `value`- und einem `label`-Element bestehen muss. Daher wird von `map()`

für jeden Array-Eintrag ein Objekt mit den beiden Keys zurückgegeben, wobei `value` den Meta-Key enthält und `label` die Beschriftung, die für Benutzer*innen sichtbar ist.

6. Anschließend wird der String zusammengebaut, der in dem Block angezeigt wird. Dafür wird zuerst in `value` der Wert für den Meta-Key aus dem `meta`-Objekt gespeichert. Danach wird wieder geprüft, ob es sich um einen Meta-Key für einen Link handelt. Falls ja, folgt wie in dem anderen Block der `try/catch`-Block, in dem versucht wird, das Linkelement zu erzeugen. Ist keine gültige URL angegeben, wird nur ein String ausgegeben, der auf die ungültige URL hinweist.
7. Danach wird getestet, ob `value` etwas enthält. Falls nicht, wird ein beschreibender Text ausgegeben, damit der Block irgendeinen Inhalt enthält, sonst könnte er im Editor nach dem Einfügen nur noch umständlich wieder ausgewählt werden und wäre quasi unsichtbar. Außerdem wird dieser String angezeigt, wenn der Block im Site-Editor verwendet wird.
8. Im nächsten Schritt wird ein leerer String in `labelMarkup` gespeichert und im Anschluss, wenn das `label`-Attribut nicht leer ist, mit einem `span`-Element mit dem Inhalt des `label`-Attributs überschrieben.

Weiter geht es mit dem Code in Listing 24.24, der direkt an den aus Listing 24.23 anschließt.

```
return (
 <>
  <InspectorControls>
   <PanelBody title={ __( 'Talk meta settings', 'wh-talks' ) }>
    <SelectControl
     label={ __( 'Meta value to display', 'wh-talks' ) }
     value={ metaKey }
     options={ selectOptions }
     onChange={ ( metaKey ) =>
      setAttributes( { metaKey } )
     }
```

```
      />
      <TextControl
       label={ __( 'Label', 'wh-talks' ) }
       value={ label }
       onChange={ ( label ) =>
        setAttributes( { label } )
       }
      />
     </PanelBody>
    </InspectorControls>
    <p { ...useBlockProps() }>
     { labelMarkup }
     { value }
    </p>
   </>
  );
 }
} );
```

Listing 24.24 Ende des Scripts zur Registrierung des Blocks

1. Innerhalb des `return`-Statements folgt die Ausgabe des Markups, das den Benutzer*innen angezeigt wird. Die `InspectorControls`-Komponente gibt ihren Inhalt innerhalb der Block-Seitenleiste aus. Mit `PanelBody` wird ein neuer Abschnitt mit einem Titel in der Seitenleiste erstellt, und darin werden ein Select-Element sowie ein Textfeld ausgegeben. Der `SelectControl` wird eine Beschriftung (`label`), der Wert des `metaKey`-Attributs als ausgewählter Wert (`value`) und das weiter oben zusammengesetzte `selectOptions`-Array als die verfügbaren Auswahlmöglichkeiten (`options`) übergeben. Wird ein anderer Wert ausgewählt (`onChange`), wird der neu gewählte Meta-Key (`metaKey`) mit der `setAttributes()`-Funktion im `metaKey`-Attribut gespeichert. Da die Variable, die den neuen Meta-Key enthält, genauso heißt wie das Attribut, müssen in `setAttributes()` nicht explizit Key und Wert übergeben werden (`setAttributes( { metaKey: newValue } )`), stattdessen setzt JavaScript automatisch den Wert für den Schlüssel, der dem Variablennamen entspricht. Für `TextControl` werden ebenfalls

`label`, `value` und `onChange` angegeben, nur `options` wird hier nicht gebraucht.

2. Nachdem `PanelBody` und `InspectorControls` geschlossen wurden, folgt die Ausgabe, die im Block angezeigt wird. Das `p`-Element wird mit `{ ...useBlockProps() }` als Block-Wrapper markiert, und innerhalb des Absatzes werden die Werte von `labelMarkup` und `value` ausgegeben.

Öffnen Sie nun den Editor für einen Vortrag mit Metawerten und fügen Sie den Block Talk meta value ein. Wenn Sie einen Veranstaltungsnamen eingegeben haben, sollte dieser direkt angezeigt werden, und in der Seitenleiste sollten die Optionen zum Ändern des anzuzeigenden Metadatums sowie des Labels vorhanden sein (siehe Abbildung 24.10).



Abbildung 24.10 Nachdem Sie den »Talk meta value«-Block eingefügt haben, finden Sie rechts in der Seitenleiste die Optionen.

Die Frontend-Ausgabe

Jetzt fehlt noch die Ausgabe im Frontend, damit die Inhalte wie gewünscht sichtbar werden. Dafür ersetzen Sie die bisher wenig sinnvolle Funktion `render_single_meta_block()` in der *blocks.php* mit dem Code aus Listing 24.25.

```
function render_single_meta_block( $attributes, $block_content ) {
 $meta_key = $attributes['metaKey'] ?? null;
```

```
 if ( empty( $meta_key ) ) {
  return '';
 }
 $meta_value_markup = get_meta_value_markup( $meta_key );
 if ( empty( $meta_value_markup ) ) {
  return '';
 }

 $wrapper_attributes = get_block_wrapper_attributes();

 $label = $attributes['label'] ?? null;
 if ( empty( $label ) ) {
  return "<p {$wrapper_attributes }>$meta_value_markup</p>";
}

 return "<p {$wrapper_attributes}><span class='wh-talks-meta-label'>$label
</span>$meta_value_markup</p>";
}
```

Listing 24.25 Die Ausgabe des Blocks für einen einzelnen Metawert

1. In der Funktion wird zunächst überprüft, ob ein `metaKey`-Attribut vorhanden ist, ansonsten wird ein leerer String zurückgegeben.
2. Ist ein Meta-Key gegeben, folgt der Aufruf der bereits besprochenen `get_meta_value_markup()`-Funktion, deren Ergebnis in `$meta_value_markup` gespeichert wird.
3. Kommt ein leerer String von der Funktion zurück, wird der Funktionsdurchlauf wieder abgebrochen und ein leerer String zurückgegeben.
4. Gibt es einen Wert für den Meta-Key, werden zunächst wie in der Render-Funktion des anderen Blocks wieder die Wrapper-Attribute geholt und danach wird überprüft, ob dem Meta-Wert ein Label vorangestellt werden muss. Ist kein Label gegeben, wird der Wert von `$meta_value_markup` in ein `p`-Element gepackt und zurückgegeben. Ist ein Label vorhanden, wird vor dem Metawert noch das Label eingefügt.

Wechseln Sie jetzt ins Frontend des Vortrags, sollten Sie etwas angezeigt bekommen, das Abbildung 24.11 ähnelt.



Abbildung 24.11 Der neue Block im Frontend

Nur die Ausgabe für bestimmte Meta-Keys erlauben

Wenn wir uns jetzt noch ein bisschen mit dem Thema Sicherheit beschäftigen, fällt auf, dass bei diesem Block im Frontend ein Metawert anhand eines dynamisch übergebenen Meta-Keys ausgegeben wird. Dabei sollte nicht davon ausgegangen werden, dass es nur einer der Meta-Keys sein kann, die im Editor als Auswahlmöglichkeiten für die Select-Option gesetzt sind – theoretisch könnte jeder Meta-Key ausgegeben werden, weshalb die Gefahr besteht, dass Metadaten zu einem Vortrag ausgegeben werden, die nicht angezeigt werden sollen.

Um das zu beheben, wird die Funktion `get_meta_value_markup()` um einen entsprechenden Test erweitert (siehe Listing 24.26).

```
function get_meta_value_markup( $meta_key ) {
$allowed_meta = false;

 foreach ( get_string_metas() as $meta ) {
  if ( $meta_key !== $meta['key'] ) {
   continue;
  }
```

```
 $allowed_meta = true;
 break;
}

if ( ! $allowed_meta ) {
 return '';
}

$meta_value = get_post_meta( get_the_ID(), $meta_key, true ) ?? null;
```

Listing 24.26 Zur Sicherheit wird geprüft, ob der übergebene Meta-Key einer derjenigen ist, die ausgegeben werden sollen.

Der neue Code ist fett hervorgehoben.

1. Die `$allowed_meta`-Flag-Variable wird während der Prüfung genutzt, um zu kennzeichnen, ob der Meta-Key erlaubt ist oder nicht.
2. Zur Prüfung wird das Meta-Info-Array aus `get_string_metas()` durchlaufen. Darin wird geprüft, ob der an die Funktion übergebene `$meta_key` dem `key`-Wert des gerade in der Schleife aktuellen Unter-Arrays des Meta-Info-Arrays entspricht. Ist das nicht der Fall, wird der aktuelle Schleifendurchlauf beendet.
3. Wenn die Keys übereinstimmen, wird die `$allowed_meta`-Flag auf `true` gesetzt und die Schleife beendet.
4. Nach der Schleife wird getestet, ob die Flag noch auf `false` steht, also kein erlaubter Meta-Key übergeben wurde, und in dem Fall wird ein leerer String zurückgegeben.

Der Test könnte auch direkt in die Render-Funktion für den Single-Meta-Block eingebaut werden, aber so greift der Check auch, wenn andere Blöcke auf die Funktion zugreifen. Falls weitere Blöcke erstellt werden, die andere Dinge mit den Metadaten anstellen, als diesen String anzufragen, sollte die Prüfung in eine komplett eigene Funktion ausgelagert werden, damit sie unabhängig von der Ausgabe verwendet werden kann.

24.5.4 Die Blöcke nur für den talk-Inhaltstyp erlauben

Aktuell ist es noch so, dass die beiden neuen Blöcke in jeden Inhalt eingebaut werden könnten, seien es Seiten, Beiträge oder auch Widget-Bereiche. Erlaubt sein sollen die Blöcke aber nur in der Bearbeitungsansicht von Einträgen des `talks`-Inhaltstyps und im Website-Editor. Um diese Einschränkung umzusetzen, können Sie den Filter `allowed_block_types_all` einsetzen, der die erlaubten Blöcke in einem bestimmten Editorkontext filtert. Bevor Sie die Funktion schreiben, fügen Sie die folgende Zeile oben in der *inc/blocks.php* nach der Zeile `namespace WH\Talks;` ein:

```
use WP_Block_Type_Registry;
```

Das `use`-Statement sorgt dafür, dass die Klasse aus dem globalen Namespace in den `WH\Talks`-Namespace importiert wird. Ansonsten würde ein Aufruf von `WP_Block_Type_Registry` in Listing 24.27 zu einem fatalen Fehler und Programmabbruch führen. Für nicht im Namespace existierende Funktionen wird automatisch im globalen Namespace geschaut, für Klassen nicht. Eine Alternative zur Nutzung von `use` wäre, beim Aufruf der Klasse ihren Namespace mit anzugeben, im Fall des globalen Namespace wäre das nur ein Backslash, also `\WP_Block_Type_Registry`.

Nachdem das erledigt ist, kann es mit der Funktion weitergehen, die die erlaubten Blöcke filtert. Der entsprechende Code ist in Listing 24.27 zu sehen und gehört ebenfalls in die *inc/blocks.php*.

```
function filter_allowed_blocks(
 $allowed_block_types,
 $block_editor_context
) {
 if (
  'talk' === get_post_type()
  || 'core/edit-site' === $block_editor_context->name
 ) {
  return $allowed_block_types;
 }
```

```
 if ( false === $allowed_block_types ) {
  return $allowed_block_types;
 }

 $blocks_to_remove = [
  'wh-talks/meta-list',
  'wh-talks/single-meta',
 ];
 if ( is_array( $allowed_block_types ) ) {
  foreach ( $allowed_block_types as $key => $block_type ) {
   if ( ! in_array( $block_type, $blocks_to_remove, true ) ) {
    continue;
   }

   unset( $allowed_block_types[ $key ] );
  }
  return $allowed_block_types;
 }

 $tmp = WP_Block_Type_Registry::get_instance()->get_all_registered();

 $allowed_block_types = [];
 foreach ( $tmp as $block_name => $block_object ) {
  if ( in_array( $block_name, $blocks_to_remove, true ) ) {
   continue;
  }

  $allowed_block_types[] = $block_name;
 }

 return $allowed_block_types;
}
add_filter(
 'allowed_block_types_all',
 __NAMESPACE__ . '\\filter_allowed_blocks',
 10,
 2
);
```

Listing 24.27 Die Funktion zur Einschränkung der neuen Blöcke auf »talk«-Posts und auf den Website-Editor

1. Die Funktion erhält als ersten Parameter die erlaubten Block-Typen. Das ist entweder ein boolescher Wert (`true`, wenn alle Blöcke erlaubt sind, `false`, wenn keine erlaubt sind) oder ein Array mit den Slugs der erlaubten Blöcke. Der zweite Parameter ist der Kontext, in dem der Block-Editor angezeigt wird, für den die erlaubten Blöcke gefiltert werden.

2. Innerhalb der Funktion wird geprüft, ob der Block-Editor entweder gerade einen Eintrag des talk-Inhaltstyps zeigt (`'talk' === get_post_type()`) oder ob der Site-Editor angezeigt wird (`$block_editor_context->name === 'core/edit-site'`). In diesen Fällen wird einfach der Wert von `$allowed_block_types` zurückgegeben, ohne die erlaubten Blöcke anzupassen.
3. Danach wird geprüft, ob `$allowed_block_types` den Wert `false` hat, also aus irgendeinem Grund gar keine Blöcke erlaubt sein sollen. Auch in diesem Fall kann die Funktion früh beendet und der Wert von `$allowed_block_types` direkt zurückgegeben werden.
4. Im nächsten Schritt wird ein Array mit den Block-Slugs angelegt, die entfernt werden sollen.
5. Danach wird geprüft, ob `$allowed_block_types` ein Array ist. Das kann der Fall sein, wenn sich bereits ein anderes Plugin oder ein Theme in den Filter gehängt und eine konkrete Liste erlaubter Blöcke zurückgegeben hat.
6. Wenn es ein Array ist, werden die Einträge aus `$allowed_block_types` durchlaufen, wobei neben dem aktuellen Array-Wert (`$block_type`) auch der aktuelle Index (`$key`) nutzbar gemacht wird, sodass der aktuelle Eintrag einfach aus dem Array entfernt werden kann, wenn es nötig ist.
7. Innerhalb der `foreach()`-Schleife wird mit `! in_array( $block_type, $blocks_to_remove, true )` geprüft, ob der aktuelle Slug aus `$block_type` nicht in dem `$blocks_to_remove`-Array vorkommt. Wenn das zutrifft, wird der aktuelle Schleifendurchlauf abgebrochen, und es wird mit dem nächsten Element weitergemacht. Ist es einer der gesuchten Blöcke, wird der Eintrag über `unset( $allowed_block_types[ $key ] )` aus dem Array entfernt.

8. Nach der Schleife wird das veränderte Array zurückgegeben.
9. Wenn wir bis zu diesem Punkt in der Funktion kommen, ist `$allowed_block_types` weder `false` noch ein Array, es sind also erst mal alle Blöcke erlaubt. Jetzt muss ein Array mit allen registrierten Blöcken erstellt werden, das nur die aus `$blocks_to_remove` nicht enthält. Dafür wird `WP_Block_Type_Registry::get_instance()->get_all_registered()` genutzt, das ein Array aller registrierten Block-Objekte zurückgibt, wobei der Key zu den Block-Objekten der Name des Blocks bzw. der Slug ist (also der Wert, auf den Sie prüfen möchten).
10. Nach dem Speichern eines leeren Arrays in `$allowed_block_types` wird in einer `foreach()`-Schleife das Array mit den Block-Objekten durchlaufen. In der Schleife wird geprüft, ob der Array-Key (`$block_name`) zum aktuellen Block-Objekt in `$blocks_to_remove` vorkommt, um dann zum nächsten Schleifendurchlauf zu wechseln. Ist der Slug kein Teil des Arrays, wird er dem `$allowed_block_types`-Array hinzugefügt.
11. Am Ende der Funktion wird `$allowed_block_types` zurückgegeben, das jetzt ein Array mit den Slugs aller registrierten Blöcke außer denen aus `$blocks_to_remove` enthält.
12. Abschließend wird die Funktion `filter_allowed_blocks()` an den `allowed_block_types_all`-Filter gehängt, wobei an `add_filter()` hier auch der dritte (die Priorität, mit der der Filter ausgeführt wird – `10` ist der Standardwert; Filter mit höherer Priorität werden nach Filtern mit niedrigerer Priorität ausgeführt) und vierte Parameter (die Anzahl der Filterparameter, die an die Filterfunktion übergeben werden

sollen) genutzt werden müssen, damit die Filterfunktion auch Zugriff auf den zweiten Parameter, den Block-Kontext, hat.

Wenn Sie jetzt in einen Vortrags-Post einen der beiden Blöcke einfügen möchten, sollte das weiter funktionieren, in einem Beitrag oder einer Seite sollte es nicht möglich sein (siehe Abbildung 24.12 und Abbildung 24.13).

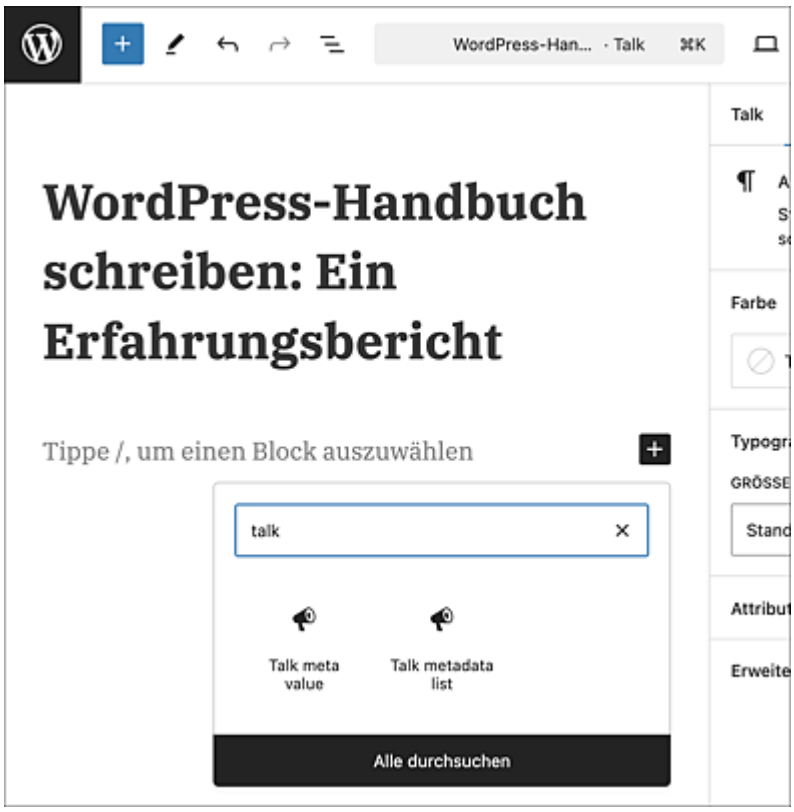


Abbildung 24.12 In der Bearbeitungsansicht eines Vortrags sind die beiden Blöcke weiter verfügbar.

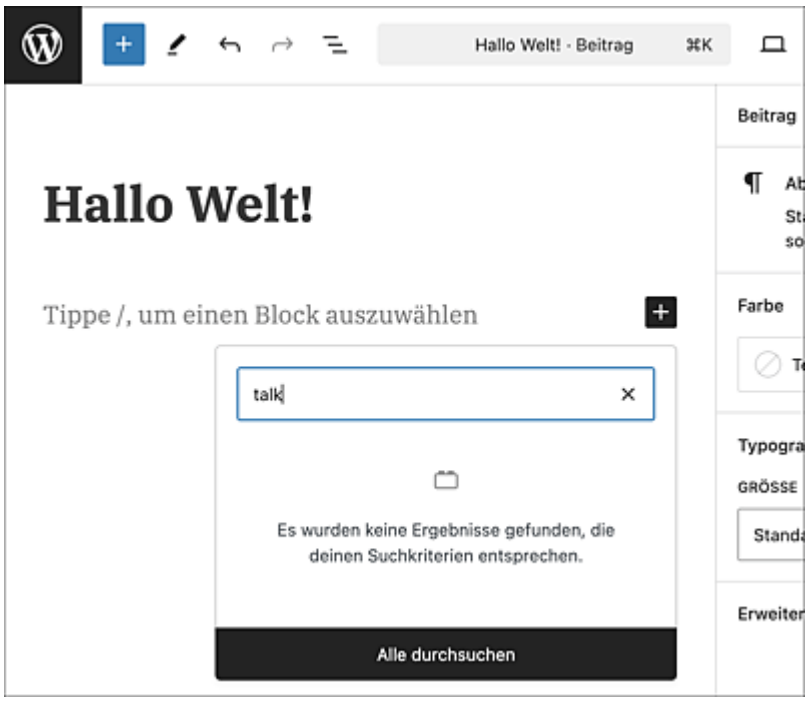


Abbildung 24.13 In einen Beitrag können die Blöcke nicht mehr eingefügt werden.

24.6 Eine Checkbox zur Hervorhebung eines Vortrags ins Veröffentlichungs-Panel einfügen

Bei der Erstellung des Sidebar-Panels in Abschnitt 1.4 haben Sie bereits das SlotFill `PluginDocumentSettingPanel` kennengelernt, mit dem Panels in der Seitenleiste eingefügt werden können. Ein anderes SlotFill ermöglicht die Erweiterung des Panels in der Editor-Sidebar, in der der Veröffentlichungsstatus, das Template und andere Infos stehen. Dort soll eine Checkbox platziert werden, mit der ein Metawert verändert wird, der im Theme genutzt werden kann, um die entsprechenden Einträge hervorzuheben (siehe Abbildung 24.14).



Abbildung 24.14 So soll das Ergebnis am Ende aussehen: eine Checkbox, mit der ein Vortrag als Highlight markiert werden kann.

24.6.1 Script erstellen

Erstellen Sie dazu die Datei *src/highlight-talk.js* und importieren Sie sie in der *src/index.js*, wie im nachfolgenden Listing zu sehen.

```
import './sidebar-panel';
import './highlight-talk';
```

Jetzt können Sie die *hightlight-talk.js* mit dem Code aus Listing 24.28 füllen, der die Checkbox einfügt.

```
import { registerPlugin } from '@wordpress/plugins';
import { PluginPostStatusInfo } from '@wordpress/edit-post';
import { useEntityProp } from '@wordpress/core-data';
import { CheckboxControl } from '@wordpress/components';
import { __ } from '@wordpress/i18n';

registerPlugin( 'wh-hide-page-title-checkbox-post-status-info', {
 render: () => {
  if ( 'talk' !== wp.data.select( 'core/editor' ).getCurrentPostType() ) {
   return null;
  }
  const [ meta, setMeta ] = useEntityProp( 'postType', 'talk', 'meta' );
  const updateMeta = ( newValue ) => {
   setMeta( { ...meta, wh_talks_is_highlight: newValue } );
  };
  const checked = meta?.wh_talks_is_highlight;
  return (
   <PluginPostStatusInfo>
    <CheckboxControl
     label={ __( 'Highlight', 'wh-talks' ) }
     checked={ checked }
     onChange={ () => {
      updateMeta( ! checked );
     } }
    />
   </PluginPostStatusInfo>
  );
 },
} );
```

Listing 24.28 Der Code für die »highlight-talk.js«, mit dem die Checkbox erstellt wird

1. Wie bereits gewohnt, werden am Anfang der Datei die Komponenten importiert, die verwendet werden.
2. Es wird wieder die `registerPlugin()`-Funktion genutzt, die schon für das Panel in der Seitenleiste zum Einsatz gekommen ist. Nach dem Parameter mit dem Plugin-Bezeichner wird sich im `render`-Attribut des Objekts, das als zweiter Parameter übergeben wird, um die Ausgabe des Plugins gekümmert.

3. Zunächst wird in der Funktion mit `'talk' !== wp.data.select( 'core/editor' ).getCurrentPostType()` geprüft, ob ein anderer Inhaltstyp als `talk` angezeigt wird, und dann wird `null` zurückgegeben, damit die Checkbox nicht für andere Inhaltstypen angezeigt wird.
4. Der Teil mit `useEntityProp()` wurde in Abschnitt 24.4.2, »Das Panel erstellen«, genauer erläutert, in Kürze: Der Hook kann genutzt werden, um den Wert eines Attributs einer Entity (zum Beispiel ein Vortrags-Post) auszulesen (`meta`) und auch zu verändern (`setMeta()`). `updateMeta` enthält eine Funktion, die die Metadaten des Inhalts aktualisiert, sodass für `wh_talks_is_highlight` der Wert der Checkbox gespeichert ist (auch diese Syntax wird in Abschnitt 24.4.2 etwas genauer erklärt).
5. Danach wird in `checked` der aktuelle Metawert gespeichert.
6. In das `return`-Statement kommt, wie aus den anderen Beispielen bekannt, die Ausgabe, die im Editor angezeigt wird. Die Komponente `PluginPostStatusInfo` sorgt dafür, dass der zwischen dem öffnenden und dem schließenden Tag eingefügte Inhalt in dem SlotFill des Panels mit den Veröffentlichungsinformationen angezeigt wird. Anders als dem SlotFill für das komplett neue Panel werden diesem keine Attribute übergeben.
7. Als einziges Element soll eine Checkbox angezeigt werden. Dafür kommt die `CheckboxControl`-Komponente zum Einsatz, der Sie die Attribute `label` (die sichtbare Beschriftung), `checked` (ob die Checkbox angehakt ist oder nicht) und `onChange` (wird ausgeführt, wenn sich der Status der Checkbox ändert) übergeben. An `checked` wird der Metawert übergeben, und die `onChange`-Funktion ruft `updateMeta` auf, damit der Metawert auf

das Gegenteil des aktuellen Werts geändert wird (`! checked` als Parameter bedeutet, dass, wenn die Checkbox beim Anklicken angehakt ist, `false` an `updateMeta()` übergeben wird und andersherum).

Wenn Sie den Code mit einem der beiden npm-Scripts kompilieren, sollten Sie im Editor jetzt bereits die Checkbox sehen können, wie in Abbildung 24.14 gezeigt. Allerdings können Sie den Status bisher nicht verändern, da der Meta-Key noch nicht registriert wurde, in dem der Wert gespeichert werden soll.

24.6.2 Meta-Key registrieren

Wie bei dem Sidebar-Panel, das sich um die Metawerte kümmert, die von den Blöcken angezeigt werden, muss auch für die Checkbox explizit der Meta-Key registriert werden und in der REST-API sichtbar sein, damit der Metawert vom Block-Editor aus verändert werden kann.

Erweitern Sie deshalb die `register_meta()`-Funktion in *inc/cpt.php* um den Code aus Listing 24.29.

```
register_post_meta(
 'talk',
 'wh_talks_is_highlight',
 [
  'single'       => true,
  'show_in_rest' => true,
  'type'         => 'boolean',
 ]
);
```

Listing 24.29 Die Post-Meta für den Checkbox-Wert muss registriert werden, bevor die Checkbox im Editor richtig funktioniert.

Die Funktion `register_post_meta()` wurde für die anderen Metadaten auch verwendet, nach dem Inhaltstyp kommt der

Parameter mit dem Namen `wh_talks_is_highlight`, den Sie eben im JavaScript-Teil schon gesehen haben, und der dritte Parameter gibt an, dass der Meta-Key einen einzelnen Wert enthält, in der REST-API sichtbar sein soll und vom Typ boolean ist (`true` und `false` bzw. in der Datenbank `1` und `0`).

Nun sollten Sie die Checkbox verwenden können, und eine angehakte Checkbox sollte nach dem Speichern und einem Neuladen des Editors noch angehakt sein.

24.6.3 Highlight-Klassen zum Styling im Frontend ausgeben

Jetzt kann zwar ein Vortrag als Highlight markiert werden, im Frontend wird bisher allerdings nichts ausgegeben, was das Theme nutzen könnte, um Einträge mit dem Highlight-Metawert auch wirklich optisch hervorzuheben. Um das zu ändern, werden Sie im nächsten Schritt die Ergebnisse der Funktionen `post_class()` und `body_class()` verändern, die beide in Kapitel 23 eingesetzt wurden, um Klassen mit Informationen über den dargestellten Inhalt auszugeben.

`post_class()` wird dabei für das Element genutzt, das den Beitrag umschließt (oft ein `article`-Element), und wird ebenfalls bei Beitragsübersichten eingesetzt, sodass auch dort Artikel anders dargestellt werden können.

`body_class()` gibt einen String mit den Klassen für das `body`-Element aus, ist zum Zweck der Hervorhebung einzelner Posts also nur sinnvoll für die Einzelansicht.

Wenn es keine Block-Themes geben würde, wäre die Veränderung des `post_class()`-Ergebnisses vielleicht schon ausreichend, bei

Block-Themes wird diese Funktion aber nur bei der Nutzung des Abfrage-Loop-Blocks aufgerufen, also bei Übersichtsseiten, nicht in der Einzelansicht. Daher wird im Folgenden die Ausgabe für beide Funktionen angepasst, der Code dafür steht in Listing 24.30.

```
function add_highlight_post_class( $classes, $class, $post_id ) {
 if ( 'talk' !== get_post_type() ) {
  return $classes;
 }

 $is_highlight = (bool) get_post_meta( $post_id, 'wh_talks_is_highlight', true );
 if ( $is_highlight ) {
  $classes[] = 'highlight-talk';
 }

 return $classes;
}
add_filter( 'post_class', __NAMESPACE__ . '\\add_highlight_post_class', 10, 3 );

function add_highlight_body_class( $classes ) {
 if ( 'talk' !== get_post_type() || ! is_single() ) {
  return $classes;
 }

 $is_highlight = (bool) get_post_meta( get_the_ID(), 'wh_talks_is_highlight', true
);
 if ( $is_highlight ) {
  $classes[] = 'highlight-talk';
 }

 return $classes;
}
add_filter( 'body_class', __NAMESPACE__ . '\\add_highlight_body_class' );
```

Listing 24.30 Hier wird das Ergebnis der Funktionen »post_class()« und »body_class()« angepasst, sodass für hervorgehobene Vorträge eine Klasse eingefügt wird.

1. Die `add_highlight_post_class()`-Funktion wird an den `post_class`-Filter gehängt und verändert die Ausgabe von `post_class()`.
2. Die Funktion bekommt drei Parameter übergeben: `$classes` ist ein Array mit allen Klassen, `$class` ein Array von Klassen, die zusätzlich angegeben wurden, und `$post_id` ist die ID des Posts.

3. Innerhalb von `add_highlight_post_class()` wird zunächst geprüft, ob es sich beim Post um einen `talk` handelt, ansonsten werden die Klassen unverändert zurückgegeben.
4. Im nächsten Schritt wird mit `get_post_meta()` der Wert des Meta-Keys `wh_talks_is_highlight` ermittelt, wofür drei Parameter übergeben werden: die Post-ID, der Meta-Key und die Angabe, dass nur ein einzelner Wert zurückgegeben werden soll. Falls der Vortrag als Highlight markiert wurde, wird dem `$classes`-Array die Klasse `highlight-talk` hinzugefügt. Am Ende wird das Array mit den Klassen zurückgegeben.
5. Die zweite Funktion, `add_highlight_body_class()`, wird an den `body_class`-Filter gehängt und verändert die Ausgabe der `body_class()`-Funktion. Dieser Filter bietet keinen Parameter mit der Post-ID an, daher wird nur der erste benutzt: ein Array mit den ganzen Klassen.
6. Hier wird zunächst geprüft, ob überhaupt ein Inhalt angezeigt wird, bei dem eine Anpassung der Klassen nötig sein könnte. Im Fall von `body_class` wird nicht nur auf den Inhaltstyp `talk` getestet, sondern auch darauf, ob eine Einzelansicht angezeigt wird (`is_single()`). Sollte etwas anderes als eine Einzelansicht eines `talk`-Posts angezeigt werden, wird das Klassen-Array direkt zurückgegeben.
7. Danach wird wie bei der Funktion für die Post-Klassen der Metawert ermittelt, nur dieses Mal mit `get_the_ID()` als erstem Wert, um die Post-ID zu übergeben.
8. Im Nachgang wird ebenfalls, falls es ein Highlight-Vortrag ist, die Klasse `highlight-talk` hinzugefügt, und das Ergebnis wird zurückgegeben.

Wenn Sie einen als Highlight markierten Vortrag nun im Frontend anschauen und sich in den DevTools Ihres Browsers (Rechtsklick auf die Seite und Untersuchen) den Quelltext anschauen, sollte das `body`-Element unter anderem die Klasse `highlight-talk` haben, und wenn Sie ein Classic Theme einsetzen, wird die Klasse bei dem Wrapper-Element des Inhalts ebenfalls angezeigt (siehe Abbildung 24.15).

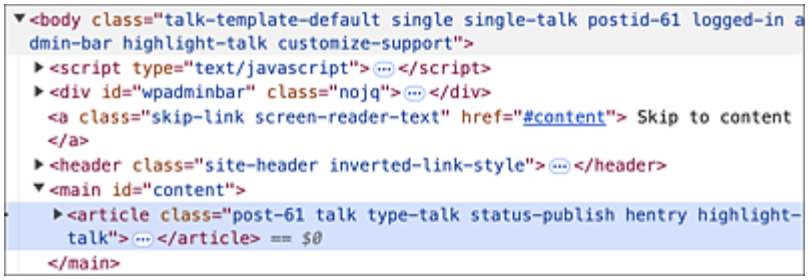


Abbildung 24.15 Ganz am Ende der Klassen im »class«-Attribut des »body«-Elements können Sie die Klasse »highlight-talk« sehen.

Auf der Übersichtsseite der Vorträge (*example.com/talks/*) sollte die Klasse bei dem Wrapper-Element des Vortrags-Teasers auftauchen, unabhängig davon, ob Sie ein Block- oder Classic Theme nutzen (siehe Abbildung 24.16). Das `body`-Element sollte in dieser Ansicht keine `highlight-talk`-Klasse erhalten haben.

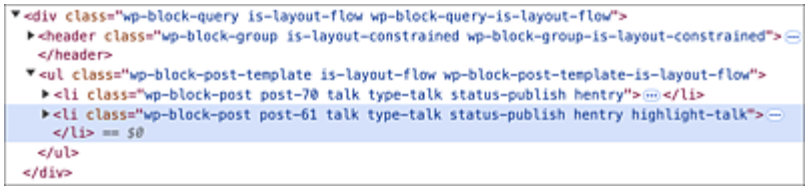


Abbildung 24.16 In der Übersicht der Talks wird im Abfrage-Loop-Block für das Wrapper-Element des hervorgehobenen Beitrags am Ende die Klasse eingefügt.

Damit kann ein Theme hervorgehobene Beiträge per CSS ansprechen und gestalten.

24.7 Block-Template für den Inhaltstyp erstellen

Bei der Registrierung eines Inhaltstyps kann optional ein Array von Blöcken angegeben werden, die beim Anlegen eines neuen Eintrags dieses Inhaltstyps in den Editor eingefügt werden. So wäre es zum Beispiel vorstellbar, dass direkt der Block zur Auflistung der gesetzten Metawerte eingefügt wird.

Verändern Sie dafür den Code der `register_cpt()`-Funktion in der *inc/cpt.php* mit den hervorgehobenen Stellen aus Listing 24.31.

```
$cpt_template = (array) apply_filters(
 'wh_talks_cpt_block_template',
 [
  [ 'core/paragraph' ],
  [ 'core/more' ],
  [ 'wh-talks/meta-list' ],
 ]
);

$args = [
 'labels'       => $labels,
 'supports'     => [
  'title',
  'editor',
  'custom-fields',
  'excerpt',
  'thumbnail',
  'revisions',
 ],
 'hierarchical' => false,
 'public'       => true,
 'show_in_rest' => true,
 'template'     => $cpt_template,
 'has_archive'  => true,
 'rest_base'    => 'talks',
 'rewrite'      => [ /* translators: slug for permalink of CPT */
  'slug' => __( 'talks', 'wh-talks' ),
 ],
 'menu_icon'    => 'dashicons-megaphone',
];
```

Listing 24.31 Erweiterung der Argumente für den Inhaltstyp um ein Block-Template, das mit »apply_filters()« aber filterbar gemacht wird

1. In `$cpt_template` wird ein Array gespeichert, das die Blöcke für das Template enthält. Ein Template besteht aus einem Array mit Arrays, von denen jedes als ersten Wert den Slug eines Blocks enthält und optional als zweiten Wert ein Array, mit dem Block-Attribute festgelegt werden können. Mit dem dritten Wert in einem Block-Array kann ein Array weiterer Block-Arrays angegeben werden, die dann als Kind-Blöcke des Blocks verwendet werden, zum Beispiel bei einem Gruppen-Block.
2. Durch die Nutzung von `apply_filters()` ist das Array über den Filter `wh_talks_cpt_block_template` veränderbar, und zwar genau so, wie Sie gerade den `post_class`-Filter verwendet haben, um Post-Klassen anzupassen. Der erste Parameter für `apply_filters()` ist der Name des Filters, der zweite der Wert, der gefiltert werden kann und von `apply_filters()` zurückgegeben wird. Mit weiteren Parametern können Sie zusätzliche Daten übergeben, die vielleicht für Funktionen nützlich sind, die den Filter verwenden, wie beispielsweise der Block-Kontext beim Filter `allowed_block_types_all`.
3. `$cpt_template` wird im `$args`-Array, den Argumenten für den `register_post_type()`-Aufruf, als Wert für `template` angegeben.

Erstellen Sie nun einen neuen Vortrag, sollten ein Absatz-Block, der Mehr-Block und der Block zur Anzeige der Metadatenliste eingefügt sein.

24.8 Übersetzung

Um Übersetzungen für ein Plugin anzuzeigen, müssen sie geladen werden. Da in dem Plugin auch Strings innerhalb von JavaScript-Code übersetzt werden müssen, ist hier ein bisschen mehr Arbeit notwendig, der Code ist in Listing 24.32 abgedruckt.

```
function load_translations() {
 load_plugin_textdomain(
  'wh-talks',
  false,
  dirname( plugin_basename( __FILE__ ) ) . '/languages'
 );
 wp_set_script_translations(
  'wh-talks-meta-list-editor-script',
  'wh-talks',
  plugin_dir_path( __FILE__ ) . '/languages/'
 );
 wp_set_script_translations(
  'wh-talks-single-meta-editor-script',
  'wh-talks',
  plugin_dir_path( __FILE__ ) . '/languages/'
 );
 wp_set_script_translations(
  'wh-talks-editor-modifications-editor-script',
  'wh-talks',
  plugin_dir_path( __FILE__ ) . '/languages/'
 );
}
add_action( 'init', __NAMESPACE__ . '\\load_translations', 11 );
```

Listing 24.32 Wegen der JavaScript-Übersetzungen ist das Laden der Übersetzungen etwas aufwendiger.

1. In der `load_translations()`-Funktion werden mit `load_plugin_textdomain()` die Übersetzungen für übersetzbare Strings in PHP geladen. Der erste Parameter ist die Text-Domain, der zweite ist veraltet, und als dritter Parameter wird der Pfad zum Ordner mit den Übersetzungen angegeben (wenn Sie ein Plugin im WordPress.org-Verzeichnis hochladen, reicht der erste Parameter aus, da die Übersetzungen dann über

translate.wordpress.org gepflegt und nicht direkt im Plugin-Verzeichnis mit ausgeliefert werden).

2. Anschließend folgen drei Aufrufe der Funktion `wp_set_script_translations()`. Diese Funktion muss für jede JavaScript-Datei aufgerufen werden, die in WordPress eingebunden wird und übersetzbare Strings enthält. Im Fall dieses Plugins sind es drei: je eine für die beiden Blöcke und eine dritte für die Erweiterungen am Editor (*src/index.js* bzw. nach dem Bauen *build/index.js*).
3. Der erste Parameter für die Datei ist das Handle der JavaScript-Datei, für die die Übersetzung geladen werden soll. Da die Scripts nicht von Ihnen selbst eingebunden wurden, sondern das durch die Aufrufe von `register_block_type()` mit Angabe der *block.json*-Datei automatisch von WordPress übernommen wird, werden auch automatisch Handles generiert. Das Format dafür ist der Block-Name, in dem `/` durch einen - ersetzt wird, und angehängt die Art des Scripts, wobei das `editorScript` aus der *block.json* beispielsweise zu `editor-script` wird. So wird für das `editorScript` des `wh-talks/single-meta` das Handle `wh-talks-single-meta-editor-script` generiert.
4. Der zweite Parameter ist die Text-Domain, für die die Übersetzungen geladen werden sollen. Der dritte Parameter ist der Pfad zum Ordner mit den Sprachen – dabei ist es wichtig, dass Sie hier einen abschließenden Slash anfügen!
5. Ebenfalls wichtig ist in dem Zusammenhang mit Script-Übersetzungen, dass der `init`-Hook mit einer höheren Priorität als `10` ausgeführt wird, damit die Funktion hier nach der Registrierung der Script-Übersetzungen abläuft, die WordPress automatisch für Scripts aus *block.json*-Dateien vornimmt. Dabei wird im Core kein Pfad zu den Übersetzungen angegeben.

WordPress versucht deshalb, die Übersetzungen von woanders zu laden, Sie überschreiben mit den Aufrufen in `load_translations()` die Aufrufe aus dem Core.

Jetzt müssen Sie noch die Übersetzungen anlegen. Um die speziellen Übersetzungsdateien für die Strings aus dem JavaScript-Code zu erstellen, sollten Sie am besten mit WP-CLI arbeiten. Wie das funktioniert, wird am Ende von Abschnitt 22.2.3, »Übersetzungen (Internationalisierung)«, erläutert. Im Kurzdurchlauf:

1. Erstellen Sie mit dem WP-CLI-Befehl `wp i18n make-pot` eine POT-Datei Ihrer Plugin-Strings (enthält auch die JavaScript-Strings). Schließen Sie dabei die JS-Dateien des `src`-Ordners aus, es sollen für die Übersetzungen nur die Dateien analysiert werden, die später eingebunden werden.
2. Erstellen Sie mit einem Tool Ihrer Wahl aus der POT-Datei eine Übersetzung in einer PO-Datei.
3. Generieren Sie mit WP-CLI über `wp i18n make-json` JSON-Übersetzungsdateien für die Strings der JavaScript-Dateien. Dabei wird für jede JavaScript-Datei eine Übersetzungsdatei mit den entsprechenden Strings erstellt. Geben Sie dabei die Flag `--no-purge` an, um die Strings nicht aus der PO-Datei zu löschen.

24.9 Inhalte beim Deinstallieren löschen

Wenn ein Plugin gelöscht wird, bleiben alle Inhalte standardmäßig erhalten, die mit dem Plugin erstellt wurden, genauso wie alle Metadaten, Optionen etc. Da Sie die WordPress-Installation der Nutzer*innen Ihres Plugins natürlich aufgeräumt hinterlassen möchten, erstellen Sie als Nächstes eine Funktion, die beim Deinstallieren des Plugins die Daten entfernt. Fügen Sie dazu den Code aus Listing 24.33 in der *wh-talks. php* vor den `require_once`-Statements ein.

```
register_uninstall_hook(
 __FILE__,
 __NAMESPACE__ . '\\uninstall'
);

function uninstall() {
 $post_ids = get_posts(
  [
   'post_type'   => 'talk',
   'post_status' => 'any',
   'numberposts' => -1,
   'fields'      => 'ids',
  ]
 );

 foreach ( $post_ids as $post_id ) {
  wp_delete_post( $post_id, true );
 }
}
```

Listing 24.33 Bei der Deinstallation werden mit der »uninstall()«-Funktion die Inhalte des Plugins entfernt.

1. Der `register_uninstall_hook()` gibt eine Funktion an, die beim Deinstallieren des Plugins ausgeführt wird. Im Gegensatz zu den Hooks zur Aktivierung und Deaktivierung ist hier keine anonyme Funktion als zweiter Parameter möglich.

2. In der `uninstall()`-Funktion werden mit `get_posts()` die IDs aller Posts vom Inhaltstyp `talk` ermittelt. Dafür wird im Argument-Array der `post_type` angegeben, der `post_status` auf `any` gesetzt, damit beispielsweise auch Entwürfe und Inhalte im Papierkorb berücksichtigt werden, und es sollen alle Elemente ausgelesen werden (`-1`). Mit der Angabe `'fields' => 'ids'` erreichen Sie, dass statt eines Arrays mit Post-Objekten ein Array nur mit den IDs zurückgegeben wird. Das ist schneller und in diesem Fall alles, was Sie an Informationen über die Einträge benötigen.
3. Anschließend wird das Array mit den IDs in einer Schleife durchlaufen, und für jedes Element wird die Funktion `wp_delete_post()` ausgeführt, der als erster Parameter die ID übergeben wird und als zweiter die Angabe, dass die Elemente komplett gelöscht werden sollen, statt sie in den Papierkorb zu legen. Bei dem Aufruf werden auch die Metadaten der Einträge gelöscht.

24.10 Plugin veröffentlichen

Wie bei den Themes gibt es auch bei Plugins generell zwei Arten, wie Sie das Plugin veröffentlichen können: kostenlos und kostenpflichtig.

24.10.1 Kostenlos

Sie können Ihr Plugin in das offizielle WordPress.org-Plugin-Verzeichnis einreichen (*https://wordpress.org/plugins/developers/*). Neben dem Vorteil der Auffindbarkeit und Installationsmöglichkeit direkt über das WordPress-Backend bekommt jedes Plugin eine eigene Seite mit Bewertungssystem, Support-Forum und mehr. Sie müssen Übersetzungen nicht selbst mitliefern, sondern können das Plugin über translate.wordpress.org übersetzen, und auch um den Update-Mechanismus kümmert sich WordPress in Verbindung mit WordPress.org selbst.

Um ein Plugin einreichen zu können, gibt es einige Richtlinien, die Sie einhalten müssen. Detailliert finden Sie die unter *https://developer.wordpress.org/plugins/wordpress-org/detailed-plugin-guidelines/*, hier folgt eine Zusammenfassung:

1. Alles im Plugin-Ordner muss kompatibel mit der *GNU General Public License* sein.
2. Plugin-Entwickler*innen sind verantwortlich für die Inhalte und Aktionen ihrer Plugins.
3. Die Version im WordPress.org-Verzeichnis muss stabil und die aktuelle Version sein. Den Code von anderswo zu verteilen und

ihn im Verzeichnis nicht auf dem aktuellen Stand zu halten, kann dazu führen, dass ein Plugin entfernt wird.

4. Der Quellcode des Plugins muss öffentlich in nicht minifizierter Version zugänglich sein – entweder als Teil des Plugins oder über einen Link in der Readme zum GitHub-Repo oder ähnlich.
5. Plugins dürfen keine Funktionen enthalten, die nur mit Bezahlung nutzbar sind.
6. Plugins als Interface für einen externen Service sind erlaubt, auch für Bezahldienste.
7. Plugins dürfen ohne Zustimmung der Nutzer*innen keine externen Requests machen. Eine Ausnahme bilden Plugins, die Funktionen von *Software as a Service* anbieten.
8. Plugins dürfen keinen ausführbaren Code über Drittanbietersysteme senden.
9. Es darf nichts Unerlaubtes, Unredliches oder moralisch Verwerfliches getan werden.
10. Es dürfen keine externen Links öffentlich angezeigt werden, ohne um Einverständnis zu fragen.
11. Das Backend darf nicht gehijackt werden.
12. Es darf kein Spam für SEO oder Ähnliches in der Readme enthalten sein.
13. Es müssen Bibliotheken genutzt werden, die der WordPress-Core mitbringt, statt eigene Versionen mitzuliefern.
14. Es dürfen nur Release-Commits in das Plugin-Verzeichnis gemacht werden, nicht alle Entwicklungs-Commits.
15. Die Plugin-Versionsnummer muss bei jeder neuen Version erhöht werden.

16. Bei Einreichung muss ein komplett funktionierendes Plugin vorliegen.
17. Plugins müssen Markenrechte, Copyrights und Projektnamen respektieren.

Wenn Sie diese Richtlinien alle erfüllen, sollten Sie für Ihr Plugin noch eine *readme. txt*-Datei anlegen, die im obersten Verzeichnis Ihres Plugins liegt und Informationen zum Plugin enthält, welche im Plugin-Verzeichnis angezeigt werden. Sie können hier zum Beispiel definieren, mit welcher WordPress-Version das Plugin getestet wurde und welche PHP-Version mindestens notwendig ist. Außerdem können Sie eine Beschreibung einfügen, häufig gestellte Fragen beantworten, Screenshots zeigen und Versionsänderungen in einem Changelog notieren. Die möglichen Optionen mit englischer Beschreibung finden Sie in einer Beispiel-*readme.txt*-Datei unter *https://wordpress.org/plugins/readme.txt*.

Nachdem das erledigt ist, können Sie das Plugin in das Verzeichnis einreichen. Nachdem Sie es hochgeladen haben, wird es manuell überprüft und bei positiver Prüfung freigeschaltet. Wenn etwas zu beanstanden ist, bekommen Sie eine Mail.

24.10.2 Kostenpflichtig

Wie bei Themes können Sie auch bei Plugins, die Sie kostenpflichtig anbieten möchten, den Weg über einen Marktplatz gehen oder eine eigene Lösung präsentieren – die Vor- und Nachteile sind die gleichen, die auch im Themes-Abschnitt beschrieben wurden. Sie können in der Regel denselben Marktplatz verwenden, auf dem Themes verkauft werden, zum Beispiel Creative Market. Auch dort gibt es Richtlinien, die Plugins erfüllen müssen. Wie der genaue Ablauf und die genauen Regeln sind, unterscheidet sich von

Marktplatz zu Marktplatz. Schauen Sie sich einfach einmal um, das Vorgehen ist auf den Seiten dokumentiert.

24.11 Weitere nützliche Funktionen für Plugins

Nachdem Sie in den vorherigen Abschnitten die Programmierung eines kleinen Plugins durchlaufen haben, das aber schon einige Bereiche abdeckt, sind natürlich bei Weitem nicht alle Funktionen und Möglichkeiten behandelt worden, die WordPress für die Nutzung in Plugins bereitstellt. Deshalb werden Sie in den folgenden Abschnitten eine Sammlung weiterer wichtiger und nützlicher Funktionen finden, die Sie vielleicht in Ihrem Plugin gebrauchen können.

Dabei handelt es sich nicht um per Copy-and-paste funktionsfähige Plugins, sondern eher um Codeschnipsel, die Sie in Ihr bestehendes Plugin übernehmen können.

24.11.1 Die Möglichkeiten der block.json im Detail

Sie haben die *block.json*-Datei und einige ihrer Konfigurationsmöglichkeiten bereits bei der Erstellung der Blöcke kennengelernt. Hier werden alle Einträge durchgegangen, die in WordPress 6.7 in der *block.json* unterstützt werden. Listing 24.34 zeigt eine Beispieldatei, in der bis auf `viewScriptModule` alle Attribute vorhanden sind (auf `viewScriptModule` wird danach noch kurz eingegangen). Bei den Werten sind längst nicht alle Möglichkeiten ausgeschöpft, in der Beschreibung wird aber auch auf die weiteren Optionen eingegangen.

```
{
  "$schema": "https://schemas.wp.org/wp/6.7/block.json",
  "apiVersion": 3,
  "name": "wh-talks/test",
  "title": "Test block",
```

```
"category": "text",
"parent": ["core/group"],
"ancestor": ["core/query"],
"allowedBlocks": ["core/paragraph", "core/image"]
"icon": "megaphone",
"description": "Block description",
"keywords": ["keywords", "find"],
"version": "1.0.0",
"textdomain": "wh-talks",
"attributes": {
  "testAttribute": {
    "type": "string",
    "default": ""
  },
  "highlight": {
    "type": boolean,
    "default": false
  }
},
"providesContext": {
  "wh-talks/testAttribute": "testAttribute"
},
"usesContext": ["wh-talks/otherAttribute"],
"supports": {
  "anchor": true,
  "align": ["wide", "center"]
},
"selectors": {
  "root": "wp-block-wh-talks-test"
},
"styles": [
  {
    "name": "special-style",
    "label": "Special Style"
  }
],
"example": {
  "attributes": {
    "testAttribute": "Example value"
  }
},
"blockHooks": {
  "wh-talks/meta-list": "after"
},
"editorScript": "file:./index.js",
"script": "file:./general.js",
"viewScript": "file:./frontend.js",
"editorStyle": "file:./editor.css",
"style": "file:./style.css",
"viewStyle": "file:./frontend.css",
"variations": [
  {
    "name": "highlight",
    "title": "Highlight",
```

```
      "attributes": {
        "highlight": true
      },
      "scope": [ "block" ],
      "isActive": [ "highlight" ]
    }
  ]
}
```

Listing 24.34 Fast alle Eigenschaften, die Sie in der »block.json« einstellen können

- In `$schema` ist die URL zu einem JSON-Schema gespeichert, das bei unterstützenden Editoren dafür sorgt, dass die Autovervollständigung angepasst wird. Im Beispiel ist die URL zur JSON-Datei von WordPress 6.6 eingefügt. Die immer neueste Datei finden Sie unter *https://schemas.wp.org/trunk/block.json*.
- `apiVersion` gibt die Block-API-Version an, mit der der Block entwickelt ist.
- Mit `name` geben Sie den internen Bezeichner des Blocks an.
- `title` ist der für Benutzer*innen sichtbare Blockname.
- `category` legt fest, in welcher Kategorie der Block im Inserter angezeigt werden soll. Der WordPress-Core bringt die Kategorien `text`, `media`, `design`, `widgets`, `theme` und `embed` mit.
- Mit `parent` können Sie angeben, dass ein Block nur als direkter Kind-Block eines anderen eingefügt werden kann, im Beispiel nur innerhalb eines Gruppen-Blocks.
- `ancestor` bietet die Möglichkeit, zu definieren, dass einer der übergeordneten Blöcke ein bestimmter anderer Block sein muss, in der Beispiel-JSON der Abfrage-Loop-Block.
- `allowedBlocks` ist ein Array von den Blöcken, die innerhalb des Blocks verwendet werden dürfen, wenn der Block einen Bereich zum Einfügen weiterer Blöcke enthält, wie der Spalten-Block.

- `icon` setzt das Block-Icon, kann ein Slug der WordPress-Dashicons sein.
- `description` enthält eine genauere Beschreibung des Blocks, die in der Block-Seitenleiste unter dem Titel angezeigt wird.
- Standardmäßig wird der Block über den Block-Namen gefunden, Sie können mit `keywords` aber ein Array zusätzlicher Begriffe angeben, für die er ebenfalls gefunden werden soll.
- Die `version` gibt dem Block eine eigene Versionsnummer, die WordPress als Query-Parameter für die URLs der Scripts und Stylesheets des Blocks nutzt, um sicherzustellen, dass keine veralteten Dateien aus dem Cache ausgeliefert werden.
- Mit `textdomain` geben Sie die Text-Domain des Plugins an, damit die Block-Metadaten übersetzt werden können.
- `attributes` beinhaltet die Attribute des Blocks. Die Werte für Attribute können entweder als Teil des Block-HTML-Kommentars gespeichert werden oder im HTML-Markup des Blocks.
- `providesContext` erlaubt es, die Werte von einem oder mehreren Attributen eines Blocks den Blöcken zugänglich zu machen, die in dem Block verschachtelt sind. Dafür wird als Key der Bezeichner des Kontexts gesetzt, den ein Kind-Block nutzt, um den Kontext auszulesen, und als Wert das Attribut, das zugänglich gemacht werden soll.
- `usesContext` wird von Kind-Blöcken genutzt, um auf den Kontext eines übergeordneten Blocks zuzugreifen. Dieser Kontext ist sowohl im JavaScript zugänglich als auch in PHP-Funktionen zum dynamischen Rendern eines Blocks (mehr Informationen gibt es im Block-Editor-Handbuch unter *https://developer.wordpress.org/block-editor/reference-guides/block-api/block-context/*).

- Mit `supports` können Sie Block-Optionen unterstützen, die vom WordPress-Core bereitgestellt werden. Der Eintrag erwartet ein Objekt mit weiteren Attributen:
  - Mit `anchor` wird im Beispielcode das Feld für den HTML-Anker aktiviert, mit dem Sie einen Wert für das `id`-Attribut des Block-Elements angeben können, und mit `align` die weite und zentrierte Ausrichtungsoption. Neben `wide` und `center` sind hier noch die Angaben `full`, `left` und `right` möglich oder statt des Arrays ein true, um alles zu unterstützen.
  - Das ebenfalls mögliche Attribut `alignWide` hat in meinen Tests keine praktische Auswirkung gehabt.
  - Über `"ariaLabel": true` können Sie aktivieren, dass dem Block im Markup mit der `ariaLabel`-Eigenschaft eine Beschriftung übergeben wird, die im Frontend dann als `aria-label` ausgegeben wird.
  - Mit `className: false` könnten Sie deaktivieren, dass WordPress automatisch einen Block-Klassennamen für das Block-Wrapper-Element setzt. Das wird beispielsweise vom Überschriften- und Absatz-Blocks genutzt.
  - Das `color`-Attribut erwartet wiederum ein Objekt, in dem Sie angeben können, ob Optionen für Hintergrundfarbe (`background`), Verläufe (`gradients`), Link- (`link`) und Textfarbe (`text`) sowie Farbe von Überschriften (`heading`) und Buttons (`button`) aktiv sein sollen. Mit `enableContrastChecker` können Sie darüber hinaus steuern, ob der Check für Farbkontraste angezeigt werden soll (standardmäßig aktiv).
  - `customClassName` ermöglicht die Deaktivierung des Felds zum Festlegen eigener Klassennamen.

- `dimensions` ist ein Objekt mit den Attributen `aspectRatio` und `minHeight` und ermöglicht damit die Steuerung des Seitenverhältnisses und der Mindesthöhe im Editor.
- Mit `filter` können Sie Unterstützung für Duotone-Filter aktivieren.
- `background` ermöglicht die Anzeige der Steuerelemente für das Hintergrundbild (`background.backgroundImage`) und die Größe des Hintergrundbilds (`background. backgroundSize`).
- `html` gibt an, ob für den Block in den HTML-Modus gewechselt werden kann oder nicht.
- Wenn `inserter` auf `false` steht, erscheint der Block nicht im Block-Inserter und kann nur programmatisch eingefügt werden.
- Standardmäßig kann ein Block im Editor umbenannt werden. Wenn Sie `renaming` auf `false` setzen, ist das nicht mehr möglich.
- `layout` hat nur Auswirkungen, wenn ein Block andere Blöcke enthalten kann. Sie können damit steuern, ob die Layout-Optionen angezeigt werden sollen, etwa für die Ausrichtung der inneren Blöcke und die Breite. Wenn Sie `layout` einfach auf `true` setzen, haben Sie die Möglichkeit, die Breite und Ausrichtung der inneren Blöcke zu steuern, Sie können aber nicht ändern, ob die inneren Blöcke beispielsweise nebeneinander angeordnet werden sollen.
  Dafür müssen Sie ein Objekt als Wert übergeben, dem Sie mit `layout.default` wiederum ein Objekt übergeben, in dem die Informationen für das Standard-Layout angegeben werden. `layout.default.type` ist dabei der Standard-Layout-Typ, hier haben Sie die Wahl aus `constrained` (der Flow der Elemente ist ganz normal untereinander, aber die Breite der inneren

Elemente verwendet automatisch die des Inhalts), `flex` (der Wrapper-Block wird zu einem Flex-Container, sodass die inneren Blöcke nebeneinander oder untereinander angeordnet werden können) und `grid`, was Ihnen ermöglicht, die inneren Blöcke als Grid anzuzeigen.
Neben dem `type` können Sie hier auch noch die Standardwerte für die weiteren Layoutoptionen festlegen, möglich sind hier die Steuerung der Inhaltsbreite der inneren Blöcke (`contentSize`) sowie die weite Breite (`wideSize`), die Ausrichtung der Elemente innerhalb des Wrappers (`justifyContent`), ob die Elemente horizontal oder vertikal fließen sollen (`orientation`) und was passieren soll, wenn nicht genug Platz ist (`flexWrap`), außerdem die vertikale Ausrichtung `verticalAlignment`, die Mindestbreite von Spalten (`minimumColumnWidth`) sowie die Spaltenanzahl (`columnCount`).
Innerhalb von `layout` haben Sie neben `default` noch die Möglichkeit, die folgenden Eigenschaften zu steuern: `layout.allowSwitching`, um anzugeben, ob der Layout-Typ im Editor geändert werden darf, `layout.allowEditing`, um festzulegen, ob überhaupt Layout-Optionen angezeigt werden, und `layout.allow Inheriting` für die Anzeige der Option der Inhaltsbreite der inneren Blöcke. Darüber hinaus gibt es `layout.allowSizingOnChildren`, um bei der Flex-Option bei den Kind-Blöcken Optionen für die Größe anzuzeigen, `layout.allowVertical Alignment` für die Anzeige der vertikalen Layout-Option bei Flex-Layouts und `layout.allowJustification` sowie `layout.allowOrientation` für die entsprechenden Ausrichtungsoptionen, wenn als Layout Flex genutzt wird. Als letzte Option gibt es noch `layout.allowCustomContentAndWideSize`, das die Sichtbarkeit der Felder für Inhaltsbreite und weite Breite steuert, wenn `constrained` als Layout-Typ ausgewählt ist.

    - Mit `multiple` können Sie bestimmen, dass ein Block nur einmal pro Seite oder Beitrag eingesetzt werden können soll.
    - Durch die Angabe von `reusable: false` wird bei einem Block die Option zur Umwandlung in einen wiederverwendbaren Block nicht angezeigt.
    - Seit WordPress 6 gibt es die Möglichkeit, Blöcke im Editor für gewisse Operationen zu sperren. So kann verhindert werden, dass Blöcke bewegt und/oder entfernt werden können. Mit `lock: false` verstecken Sie die Option im Editor, sodass ein Block, der beispielsweise als Teil einer Block-Vorlage gesperrt ist, nicht einfach im Editor entsperrt werden kann.
    - `position.sticky` ermöglicht die Aktivierung der Option, einen Block mit `position: sticky` auszustatten.
    - Mit `spacing` können Sie festlegen, ob und, wenn ja, welche Optionen für die Festlegung von Abständen der Block unterstützen soll. Möglich sind Innen- (`padding`) und Außenabstände (`margin`).
    - `shadow` erlaubt es, die Box-Shadow-Option für Blöcke zu aktivieren.
    - `typography` erlaubt die Aktivierung der Optionen für Schriftgröße (`fontSize`), Zeilenhöhe (`lineHeight`) und Ausrichtung (`textAlign`).
    - `interactivity` ermöglicht es anzugeben, ob der Block Interactivity-API-Funktionen nutzt. Auf diese API und die möglichen Werte wird in einem späteren Abschnitt genauer eingegangen.
- `selectors` erlaubt es, die Selektoren zu verändern, auf die die Styles des Blocks angewendet werden. So können Sie

beispielsweise über `selectors.color` festlegen, auf welchen Selektor die Farbeinstellungen eines Blocks wirken sollen. Sie können auch noch kleinteiliger gehen und beispielsweise über `selectors.color.text` nur den Selektor für die Textfarbe ändern, die Hintergrundfarbe wird dann auf den Root-Selektor wirken, entweder den von WordPress festgelegten oder den von Ihnen über `selectors.root` angegebenen.
Neben `selectors.root` und `selectors.color` ist noch `selectors.border`, `selectors. dimensions`, `selectors.spacing` und `selectors.typography` möglich.

- `styles` beinhaltet ein Array von Objekten, in denen zusätzliche Block-Stile definiert sind. Für jeden Stil müssen ein interner Bezeichner (`name`) und eine Beschriftung (`label`) angegeben werden. Mit `isDefault` kann ein Stil als Standardstil festgelegt werden.
- `example` erlaubt es, Beispielwerte für Attribute anzugeben, die dann in der Block-Vorschau genutzt werden.
- `blockHooks` gibt Ihnen die Möglichkeit, einen Block automatisch relativ zu einem anderen Block einzufügen. Dafür geben Sie ein Objekt an, wobei die Namen der Eigenschaften dem Namen des Blocks entsprechen müssen, und der Wert einer der vier Möglichkeiten `before`, `after`, `firstChild` und `lastChild`.
- Mit `editorScript` können Sie das Script angeben, das den Block-Code enthält, entweder ein Handle eines registrierten Scripts oder einen Pfad zu einer Datei.
- `script` erlaubt es, ein Script anzugeben, das sowohl im Editor als auch im Frontend eingebunden wird.
- Mit `viewScript` geben Sie ein oder mehrere Scripts an, die nur im Frontend eingebunden werden.

- `viewScriptModule` ist in diesem Beispiel nicht angegeben, da es nicht zusammen mit `viewScript` genutzt werden kann. `viewScriptModule` gibt an, welches Script-Modul eingebunden werden soll, interessant aktuell besonders im Zusammenhang mit der Interactivity API.
- `editorStyle` bindet ein oder mehrere Stylesheets im Editor ein.
- `style` erlaubt Ihnen die Einbindung eines oder mehrerer Stylesheets im Editor und im Frontend.
- Mit `viewStyle` können Sie Stylesheets angeben, die nur im Frontend ausgegeben werden.
- `variations` ermöglicht es, Block-Variationen anzugeben, bei denen ein oder mehrere Attribut-Werte geändert werden.
- Mit `render` können Sie eine PHP-Datei angeben, die für das Rendering des Blocks genutzt wird.

Die Handbuchseite zur block.json finden Sie unter *https://developer.wordpress.org/block-editor/reference-guides/block-api/block-metadata/*.

24.11.2 Block Bindings API: Metadaten und andere Quellen mit Core-Blöcken verknüpfen

Seit WordPress 6.5 gibt es die sogenannte *Block Bindings API*, die es ermöglicht, bestehende Blöcke mit Datenquellen wie Post-Meta zu verknüpfen. Das klingt auf den ersten Blick vielleicht recht unspektakulär, kann Ihnen aber sehr viel Arbeit ersparen, denn es kann die Entwicklung eigener Blöcke zumindest in einigen Fällen überflüssig machen, auch wenn die API noch am Anfang steht und stetig weiterentwickelt wird.

Erinnern Sie sich an das Plugin für die Verwaltung von Vorträgen, das in diesem Kapitel entwickelt wurde. Dabei wurden Vortragsdaten über Post-Meta gespeichert und über Felder in der Dokument-Seitenleiste gepflegt. Die Blöcke zur Anzeige der Daten waren relativ einfach, eine Liste für alle Beiträge und einer für einen einzelnen Wert. Letzteren Block könnten wir uns schon jetzt fast sparen, indem wir einen Core-Absatz-Block nehmen und ihm als Daten-Quelle beispielsweise die Post-Meta-Angabe für die Vortragssprache geben. Dieser Absatz-Block würde dann immer den aktuellen Post-Meta-Wert anzeigen, und seit WordPress 6.6 könnte der Wert auch direkt über den Absatz verändert werden. Was noch nicht untergebracht werden könnte, wäre das Präfix, aber zur praktischen Veranschaulichung werden wir jetzt trotzdem einmal dieses Beispiel umsetzen. Zuerst konnte die API nur über die Code-Ansicht genutzt werden, seit WordPress 6.7 gibt es auch ein User-Interface. Wir werden beide Optionen durchgehen.

Vorher kurz zur größten Einschränkung der API: Sie wird nur für vier Core-Blöcke und die angegebenen Attribute unterstützt: Bild (`url`, `alt` und `title`), Absatz (`content`), Überschrift (`content`) und Button (`url`, `text`, `linkTarget` und `rel`). Eine Erweiterung auf weitere Core-Blöcke und auch Custom Blocks ist natürlich bereits geplant.

Post-Meta mit einem Absatz-Block verbinden

Jetzt aber zum Praxisbeispiel. Wir möchten den Namen des Events in einem Absatz-Block ausgeben. Zunächst gehen wir den Weg über die Code-Ansicht, danach über das User-Interface.

1. **Block Bindings API über die Code-Ansicht nutzen**

   Um den Absatz-Block über die Code-Ansicht mit dem Meta-Wert zu verbinden, ist nichts weiter nötig als das Block-Markup

aus Listing 24.35, das Sie in einen Talk-Eintrag einfügen müssen (am besten geben Sie vorher schon einen Namen für das Event an).

```
<!-- wp:paragraph {
    "metadata":{
        "bindings":{
            "content":{
                "source":"core/post-meta",
                "args":{
                    "key":"wh_talks_event_name"
                }
            }
        }
    }
} -->
<p></p>
<!-- /wp:paragraph -->
```

Listing 24.35 Markup eines Absatz-Blocks, der über die Block Bindings API mit dem »wh_talks_event_name«-Metawert verbunden ist

Im Block-Markup spielt sich die Nutzung der Block Bindings API im öffnenden Block-Kommentar ab. Innerhalb von `metadata.bindings.content.source` geben wir an, dass als Quelle die Post-Metadaten verwendet werden sollen, und über `args.key` den Metakey, zu dem der Wert ausgegeben werden soll.

Wenn Sie jetzt aus der Quellcode-Ansicht in die visuelle Ansicht wechseln, sehen Sie einen Absatzblock, der den aktuellen Wert des Event-Namens anzeigt (in Abbildung 24.17 zu sehen). Wie kurz angerissen können Sie den Wert über den Absatzblock auch verändern und werden im Feld in der Dokument-Sidebar und, falls sie eingefügt sind, in den Custom Blocks sehen, wie sich der Wert live verändert.



Abbildung 24.17 Der Wert des Absatz-Blocks spiegelt den Meta-Wert wider, und Sie können den Metawert sogar über den Absatzblock bearbeiten.

2. **Block Bindings API über das User-Interface nutzen**

   Um die API über das User-Interface zu nutzen, fügen Sie einen Absatz-Block in den Editor ein, wählen in der Block-Seitenleiste über das Plus-Symbol bei ATTRIBUTE den Eintrag CONTENT aus und danach über den neuen Button CONTENT im Attribute-Panel den Eintrag WH_TALKS_EVENT_NAME (siehe Abbildung 24.18).

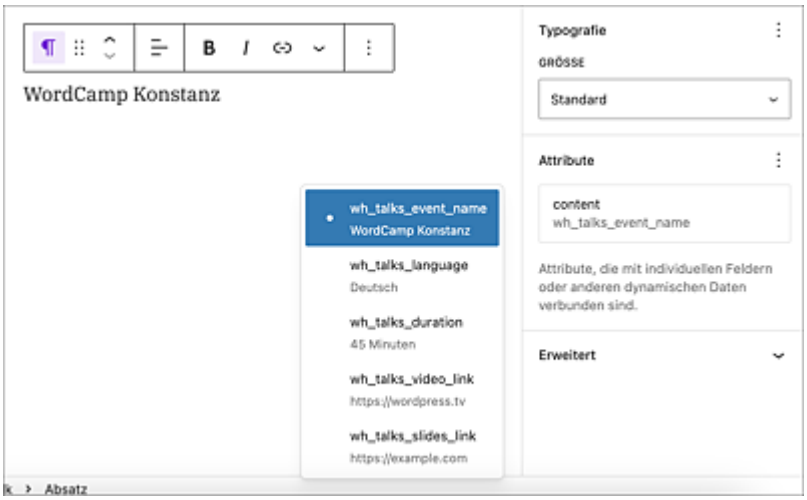


Abbildung 24.18 Seit WordPress 6.7 kann die Block Bindings API über ein User-Interface genutzt werden.

Für die Nutzung durch Anwender*innen würde es sich natürlich anbieten, die Block-Bindings in Block-Vorlagen oder Templates einzubinden.

Seit WordPress 6.6 können auch Blöcke mit Block-Binding bearbeitet werden, wenn sie in einem Abfrage-Loop-Block

auftauchen, Sie können also aus einem Eintrag heraus Metawerte eines anderen Eintrags verändern.

Sie können auch eigene Quellen für Daten registrieren. Wie das funktioniert, können Sie in einem Blogpost von Justin Tadlock nachlesen (*https://developer.wordpress.org/news/2024/03/06/introducing-block-bindings-part-2-working-with-custom-binding-sources/*).

24.11.3 Interactivity API: Interaktive Frontend-Erfahrung umsetzen

In WordPress 6.5 wurde die *Interactivity API* eingeführt, die WordPress-Entwickler*innen nutzen können, um interaktive Elemente im Frontend umzusetzen. Angelehnt ist die API an Frameworks wie Alpine.js, und sie bringt unterschiedliche Direktiven mit, die Sie im Block-Markup nutzen können, um z. B. auf einen Klick zu reagieren.

Da mir kein sinnvoller Einsatzzweck im Bereich des hier gezeigten Plugins eingefallen ist, werde ich Ihnen die API anhand des offiziellen Beispiel-Plugins näher bringen. Wechseln Sie dafür in das *wp-content/plugins*-Verzeichnis, und führen Sie den folgenden Befehl aus:

```
npx @wordpress/create-block@latest wh-interactive-block --template
@wordpress/create-block-interactive-template
```

Nach Ausführung werden Sie, falls es noch nicht installiert ist, gefragt, ob das Paket `@wordpress/create-block` installiert werden darf. Bestätigen Sie das, und warten Sie ab, bis alle Abhängigkeiten installiert sind. Anschließend sehen Sie die Ordner- und Dateistruktur aus Abbildung 24.19.

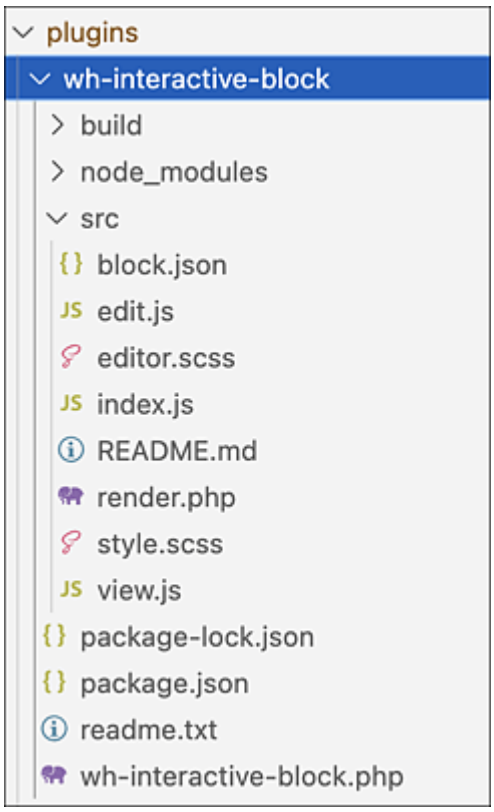


Abbildung 24.19 Nach Ausführen des Befehls haben Sie ein kleines fertiges Plugin im Ordner »wh-interactive-block«.

Das Plugin wurde auch direkt gebaut, das heißt, Sie können es einfach in WordPress aktivieren und den Block »Wh Interactive Block« in eine Seite oder in einen Beitrag einfügen. Sie sollten dann im Backend und Frontend den Block sehen, wie in Abbildung 24.20 und Abbildung 24.21.

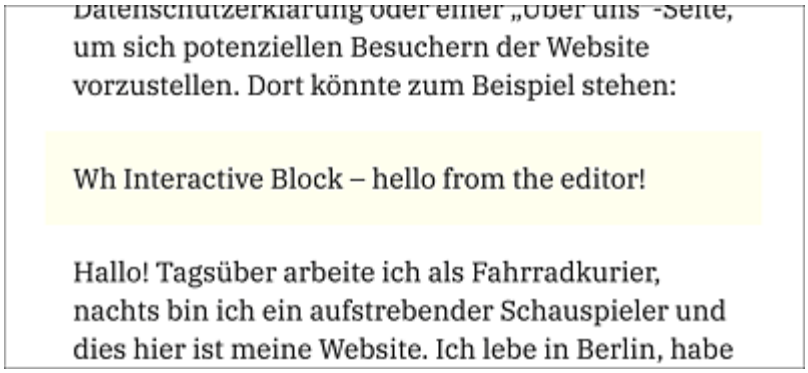


Abbildung 24.20 Im Backend ist der Block nur ein Platzhalter.



Abbildung 24.21 Im Frontend kommt der interessante, der interaktive Teil.

Im Backend hat der Block keine Funktion, der spannende Teil ist das Frontend, auf den sich die Interactivity API beschränkt – sie ist nicht dafür gedacht, im Editor genutzt zu werden. Im Frontend besteht der Block aus einem Button, mit dem Sie einen Absatz anzeigen und verbergen können, und einem weiteren Button zum Umschalten des Blocks in ein dunkles Design.

Die generelle Entwicklung eines Blocks ändert sich mit Nutzung der Interactivity API nicht, deshalb werden wir uns die Registrierung des Blocks und den Code für das Backend nicht anschauen, sondern uns auf die Teile beschränken, die aus Interactivity-API-Sicht interessant sind.

Vorbereitungen für die Interactivity API

Wenn Sie die Interactivity API nutzen wollen, müssen Sie das NPM-Paket `@wordpress/interactivity` installieren und bei den Befehlen `wp-scripts start` und `wp-scripts build` das Flag `--experimental-modules` ergänzen. Die entsprechenden Zeilen in der *package.json* sehen wie folgt aus:

```
"dependencies": {
  "@wordpress/interactivity": "^6.11.0"
},
"scripts": {
  "build": "wp-scripts build --experimental-modules",
```

```
  "start": "wp-scripts start --experimental-modules"
}
```

Außerdem müssen Sie in der *block.json* `supports.interactivity` auf `true` setzen und das View-Script über `viewScriptModule` statt `viewScript` einbinden. Die entscheidenden Teile der *src/block.json* sehen so aus:

```
{
  [...]
  "supports": {
    "interactivity": true
  }
  [...]
  "viewScriptModule": "file:./view.js"
}
```

Die API im Einsatz

Das Markup für das Frontend wird über die *src/render.php* erstellt, die über die *block.json* mit der Zeile `"render": "file:./render.php"` eingebunden wird. In dieser Datei sehen wir den Einsatz der Direktiven von der Interactivity API – das Markup ist in Listing 24.36 abgebildet.

```
<?php
$unique_id = wp_unique_id( 'p-' );

wp_interactivity_state(
    'create-block',
    array(
        'isDark'    => false,
        'darkText'  => esc_html__( 'Switch to Light', 'wh-interactive-block' ),
        'lightText' => esc_html__( 'Switch to Dark', 'wh-interactive-block' ),
        'themeText'    => esc_html__( 'Switch to Dark', 'wh-interactive-block' ),
    )
);
?>

<div
    <?php echo get_block_wrapper_attributes(); ?>
    data-wp-interactive="create-block"
    <?php echo wp_interactivity_data_wp_context( array( 'isOpen' => false ) ); ?>
    data-wp-watch="callbacks.logIsOpen"
    data-wp-class--dark-theme="state.isDark"
```

```
>
    <button
        data-wp-on--click="actions.toggleTheme"
        data-wp-text="state.themeText"
    ></button>

    <button
        data-wp-on--click="actions.toggleOpen"
        data-wp-bind--aria-expanded="context.isOpen"
        aria-controls="<?php echo esc_attr( $unique_id ); ?>"
    >
        <?php esc_html_e( 'Toggle', 'wh-interactive-block' ); ?>
    </button>

    <p
        id="<?php echo esc_attr( $unique_id ); ?>"
        data-wp-bind--hidden="!context.isOpen"
    >
        <?php
            esc_html_e( 'Wh Interactive Block - hello from an interactive block!',
'wh-interactive-block' );
        ?>
    </p>
</div>
```

Listing 24.36 Code für das Frontend-Markup des Beispiel-Blocks für die Nutzung der Interactivity API

1. Am Anfang wird eine eindeutige ID generiert, die später gebraucht wird.
2. Anschließend wird mit `wp_interactivity_state()` ein globaler State hinzugefügt. Ein State bezeichnet bei reaktiven JavaScript-Frameworks eine zentrale Stelle, an der Zustände gespeichert werden können (ist der Dark-Mode an oder nicht), und diese zentrale Stelle nutzen alle Blöcke in unserem Fall, um zum Beispiel festzustellen, ob das dunkle oder helle Design angezeigt werden muss.

   In dem State werden Daten zum Dark-Mode gespeichert, unter anderem eine Flag, ob der Dark-Mode aktiv ist, und Beschriftungen für den Button.

3. Innerhalb des öffnenden `div`-Elements wird mit `get_block_wrapper_attributes()` ein String der Block-Attribute eingefügt, die die Einstellungen im Editor widerspiegeln.
4. Mit `data-wp-interactive` wird die API für das `div` und dessen Kindelemente »aktiviert«.
5. Über `wp_interactivity_data_wp_context()` wird ein `data-wp-context`-Attribut generiert, das wie folgt aussieht: `data-wp-context='{"isOpen":false}'`. Dieses Attribut erzeugt einen lokalen Kontext für das `div` und die Kindelemente mit dem Attribut `isOpen`, der als einzige Quelle für das Lesen und Bearbeiten des `isOpen`-Zustands genutzt wird, um überall im Block auf denselben Wert zugreifen zu können.
6. `data-wp-watch` ermöglicht das Ausführen einer Callback-Funktion sowohl beim Erstellen des Nodes, also in diesem Fall beim Aufrufen der Seite mit dem Block, als auch bei jeder Änderung des Kontexts, also der `isOpen`-Eigenschaft (wo der Callback definiert wird, erfahren Sie gleich).
7. `data-wp-class--dark-theme="state.isDark"` gibt an, dass dem `div` die Klasse `dark-theme` zugewiesen wird, wenn der Wert `isDark` aus dem globalen State `true` ist.
8. Auch den `button`-Elementen werden Direktiven mitgegeben. Mit `data-wp-on--click` wird auf das `click`-Event gelauscht und die `toggleTheme`- beziehungsweise `toggleOpen`-Aktion ausgeführt (wo die definiert werden, sehen wir uns gleich an).
9. `data-wp-text="state.themeText"` sorgt dafür, dass die Beschriftung des Theme-Toggle-Buttons dem Wert von `themeText` aus dem State entspricht.
10. `data-wp-bind--aria-expanded="context.isOpen"` setzt den Wert des `aria-expanded`-Attributs immer auf den aktuellen `isOpen`-

Wert.

11. Ein ganz normales HTML-Attribut ist `aria-controls`, und hier kommt auch die anfangs erstellte ID zum Einsatz: Hiermit wird angegeben, dass der Toggle-Open-Button den weiter unten kommenden Absatz steuert, der die zufällige ID als Wert für sein `id`-Attribut bekommt.
12. Daneben wird beim Absatz mit `data-wp-bind--hidden` das `hidden`-Attribut gesetzt, wenn `context.isOpen` nicht `false` ist.

Kommen wir nun zum JavaScript-Code, der für die Frontend-Funktion des Blocks selbst geschrieben werden muss, also nicht von der Interactivity API kommt:

```
import { store, getContext } from '@wordpress/interactivity';

const { state } = store( 'create-block', {
    state: {
        get themeText() {
            return state.isDark ? state.darkText : state.lightText;
        },
    },
    actions: {
        toggleOpen() {
            const context = getContext();
            context.isOpen = ! context.isOpen;
        },
        toggleTheme() {
            state.isDark = ! state.isDark;
        },
    },
    callbacks: {
        logIsOpen: () => {
            const { isOpen } = getContext();
            // Log the value of `isOpen` each time it changes.
            console.log( `Is open: ${ isOpen }` );
        },
    },
} );
```

Listing 24.37 Das ist der gesamte Code für die Frontend-Funktion des Blocks.

1. Am Anfang werden die Funktionen `store` und `getContext` aus dem `@wordpress/interactivity`-Paket importiert.

2. Mit der `store`-Funktion wird ein Store mit dem als erster Parameter übergebenen Namespace erstellt. Dieser Namespace muss dem Wert entsprechen, der im HTML-Markup an `data-wp-interactive` und als erster Parameter an `wp_interactivity_state()` übergeben wurde.
3. In dem Store wird ein Getter für `themeText` erstellt, sodass das Ergebnis von `state.themeText` abhängig vom aktuellen `isDark`-Wert ist.
4. Anschließend werden die beiden Actions `toggleOpen()` und `toggleTheme()` definiert, die von den Buttons im HTML getriggert werden. In `toggleOpen()` wird über `getContext()` der Kontext des Blocks geholt und der `isOpen`-Wert auf den gegenteiligen Wert gesetzt, in `toggleTheme()` wird der `isDark`-Wert im State umgekehrt.
5. Außerdem wird im Store der `logIsOpen`-Callback angelegt, der den aktuellen Wert von `isOpen` in die Browser-Konsole schreibt.

Auch wenn es sich hierbei nur um ein einfaches Beispiel handelt, sollte es doch eine erste gute Übersicht über die Möglichkeiten der Interactivity API geben.

24.11.4 Eigene Block-Editor-Sidebar erstellen

Bei der Umsetzung des Plugins für die Vortragsverwaltung haben Sie bereits die Möglichkeiten kennengelernt, die Dokument-Seitenleiste des Block-Editors zu erweitern sowie Optionen für Blöcke in der Block-Seitenleiste zu erstellen. Wenn Sie ein Plugin mit vielen Optionen innerhalb des Block-Editors haben, kann es sinnvoll sein, dafür eine eigene Editor-Sidebar zu erstellen. Das ist wieder über ein SlotFill möglich, und den entsprechenden Code finden Sie in Listing 24.38. Diesen könnten Sie so zum Beispiel in eine weitere

JavaScript-Datei des *wh-talks*-Plugins schreiben, die Sie in die *src/index.js* importieren, wie schon die beiden Dateien für das Seitenleisten-Panel und die Checkbox.

```
import { registerPlugin } from '@wordpress/plugins';
import { PluginSidebar } from '@wordpress/editor';
import { useEntityProp } from '@wordpress/core-data';
import { CheckboxControl, PanelBody } from '@wordpress/components';
import { __ } from '@wordpress/i18n';

registerPlugin( 'wh-talks-sidebar', {
 render: () => {
  if ( 'talk' !== wp.data.select( 'core/editor' ).getCurrentPostType() ) {
   return null;
  }
  const [ meta, setMeta ] = useEntityProp( 'postType', 'talk', 'meta' );
  const updateMeta = ( newValue ) => {
   setMeta( { ...meta, wh_talks_is_highlight: newValue } );
  };
  const checked = meta?.wh_talks_is_highlight;
  return (
   <PluginSidebar
    name='wh-talks-sidebar'
    title={ __( 'Talk sidebar', 'wh-talks' ) }
    icon='megaphone'
   >
    <PanelBody>
     <CheckboxControl
      label={ __( 'Highlight', 'wh-talks' ) }
      checked={ checked }
      onChange={ () => {
       updateMeta( ! checked );
      } }
     />
    </PanelBody>
   </PluginSidebar>
  );
 },
} );
```

Listing 24.38 Registrierung einer eigenen Seitenleiste für den »talk«-Inhaltstyp

1. Es beginnt mit dem Import aller benötigten Komponenten.
2. Mit `registerPlugin()` wird das Block-Editor-Plugin `wh-talks-sidebar` registriert, in dessen `render`-Methode zunächst sichergestellt wird, dass die nachfolgende Ausgabe nur für Elemente des `talk`-Inhaltstyps geschieht.

3. Anschließend wird das Objekt mit den aktuellen Werten der über die REST-API sichtbaren Metadaten in `meta` gespeichert und eine Funktion zum Verändern dieses Meta-Objekts in `setMeta` (mehr zu `useEntityProp()` finden Sie in Abschnitt 24.4.2, »Das Panel erstellen«).
4. Danach wird der Wert des `wh_talks_is_highlight`-Meta-Keys gespeichert, um gleich den Status der Checkbox anzugeben.
5. Mit dem `return`-Statement wird der `PluginSidebar`-SlotFill zurückgegeben, der dafür sorgt, dass eine neue Sidebar im Editor erstellt wird. Der Komponente wird ein interner Bezeichner für die Sidebar übergeben (`name`), ein sichtbarer Name (`title`) sowie das Megafon-Icon der WordPress-Dashicons.
6. In der Seitenleiste wird die gleiche Checkbox ausgegeben wie im Talks-Plugin in dem Status-und-Sichtbarkeit-Panel.

Das Ergebnis im Editor sieht aus wie das in Abbildung 24.22.

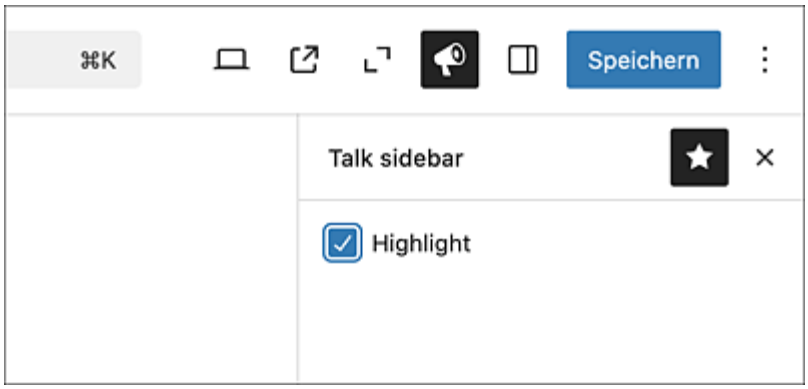


Abbildung 24.22 Eigene Seitenleiste in der Bearbeitungsansicht eines Vortrags

24.11.5 Optionen in der Dokument-Sidebar deaktivieren

Wenn Sie den Eintrag eines Inhaltstyps bearbeiten, finden Sie in der Block-Editor-Sidebar unterschiedliche Optionen, etwa zum

Festlegen des Beitragsbildes oder – bei Beiträgen – der Auswahl von Schlagwörtern und Kategorien (wie Sie hier ein eigenes Panel einbauen können, erfahren Sie in Abschnitt 24.4).

Wenn nun nicht alle dieser Optionen benötigt werden, zum Beispiel die für das Beitragsbild in der Bearbeitungsansicht von Seiten, können Sie einzelne Optionen deaktivieren. Um die Option für das Beitragsbild bei Seiten zu entfernen, können Sie den Code aus Listing 24.39 verwenden. Der kann beispielsweise in die `src/index.js` des Talks-Plugins eingefügt werden, oder Sie lagern den Code in eine eigene Datei aus (oder ein eigenes Plugin, da die Veränderung mit den Vorträgen eigentlich nichts zu tun hat).

```
wp.domReady( () => {
 let actionDone = false;
 wp.data.subscribe( () => {
  if ( actionDone ) {
   return;
  }
  const postType = wp.data
   .select( 'core/editor' )
   .getCurrentPostType();

  if ( postType === null ) {
   return;
  }
  if ( 'page' === postType ) {
   wp.data
    .dispatch( 'core/editor' )
    .removeEditorPanel( 'featured-image' );
  }
  actionDone = true;
 } );
} );
```

Listing 24.39 Code zur Deaktivierung der Beitragsbild-Option für Seiten

1. Mit `wp.domReady()` wird gewartet, bis das DOM geladen ist (das *Document Object Model*, eine Repräsentation der Webseite als Knoten und Objekte, sodass beispielsweise JavaScript damit interagieren kann – mehr Informationen dazu gibt es beim

MDN unter *https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model*).

2. Wenn das DOM vollständig geladen ist, wird eine `actionDone`-Flag gesetzt und mit `wp.data.subscribe()` an alle Änderungen gehängt, die im Editor ab jetzt passieren. Sobald eine Änderung passiert, wird die Callback-Funktion ausgeführt. Dieses Vorgehen ist notwendig, da die Information, um welchen Inhaltstyp es sich handelt, asynchron bereitgestellt wird und noch nicht vorliegt, wenn das DOM geladen ist.
3. Innerhalb der Funktion wird geprüft, ob die geplante Aktion durchgeführt worden ist, also `actionDone` auf `true` steht, damit der Funktionsdurchlauf direkt früh abgebrochen werden kann. Das ist wichtig, damit so wenige Aktionen wie möglich bei jeder Editoränderung ausgeführt werden, um die Performance nicht negativ zu beeinflussen.
4. Wenn die Aktion noch nicht abgeschlossen ist, wird über `wp.data.select( 'core/editor' ).getCurrentPostType()` der Inhaltstyp des aktuell angezeigten Posts ermittelt. Das ist der Aufruf, der direkt nach `wp.domReady()` noch `null` zurückliefert, weshalb die Nutzung von `wp.data.subscribe()` notwendig ist.
5. Wenn das Ergebnis des Aufrufs `null` ist, wird die Funktion beendet.
6. Liegt der Inhaltstyp `page` vor, wird mit dem Aufruf von `wp.data.dispatch( 'core/editor' ).removeEditorPanel( 'featured-image' )` die Beitragsbild-Option entfernt.
7. In jedem Fall wird am Ende die `actionDone`-Flag auf `true` gesetzt, auch wenn ein anderer Inhaltstyp zurückgegeben wird, damit bei der nächsten Editoränderung der `subscribe`-Callback direkt früh beendet werden kann.

Wenn Sie ein Theme verwenden, das Beitragsbilder unterstützt, und eine Bearbeitungsansicht einer Seite aufrufen, ohne den Code einzufügen, sollten Sie in der Seitenleiste ein Ergebnis wie das in Abbildung 24.23 sehen. Nach dem Einfügen des Codes ist das Panel dann entfernt (siehe Abbildung 24.24).

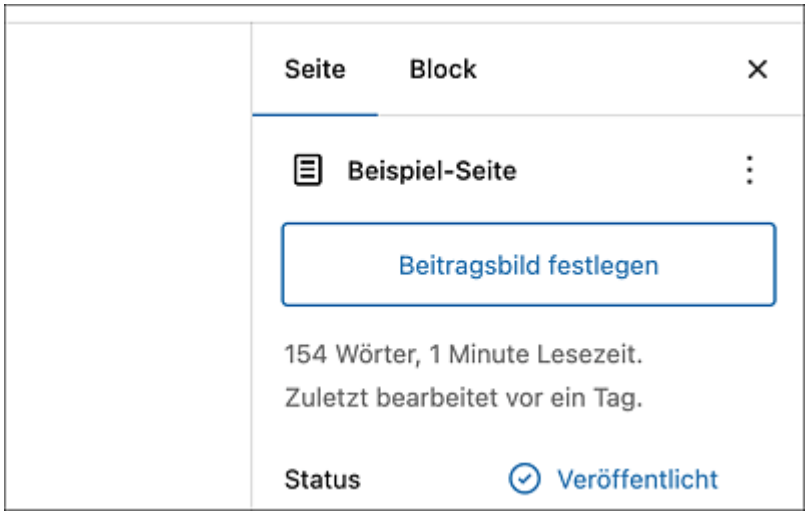


Abbildung 24.23 Die Editor-Sidebar für Seiten vor der Anpassung mit der Option für das Beitragsbild

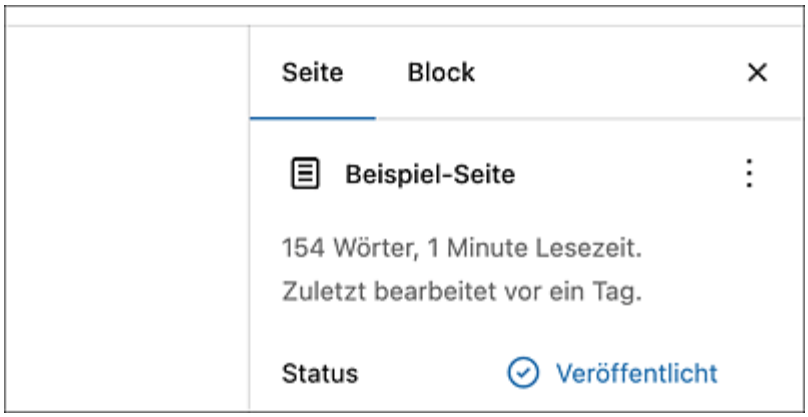


Abbildung 24.24 Nach der Anpassung wird die Beitragsbild-Option nicht mehr angezeigt.

Natürlich können Sie auf dieselbe Art und Weise auch andere Optionen deaktivieren. Im GitHub-Repository des Block-Editors gibt es ein Verzeichnis der Editor-Komponenten (*https://github.com/WordPress/gutenberg/tree/wp/6.7/packages/edit*

or/src/components). Da können Sie nach `PANEL_NAME` suchen und finden dann beispielsweise in der Datei *post-discussion/panel.js* die Zeile `const PANEL_NAME = 'discussion-panel';`). Die Diskussionseinstellungen können Sie mit dem folgenden Aufruf entfernen:

```
wp.data
  .dispatch( 'core/editor' )
  .removeEditorPanel( 'discussion-panel' );
```

24.11.6 Bestimmte Blöcke deaktivieren

Als Nutzer*in können Sie im Block-Editor über das Drei-Punkte-Menü oben rechts mit VOREINSTELLUNGEN • BLOCK-MANAGER einstellen, welche Blöcke im Block-Inserter angezeigt werden sollen. Das betrifft aber nur Ihren WordPress-Account in dem gerade genutzten Browser, ist also keine Lösung, wenn Sie beispielsweise bestimmte Blöcke für alle deaktivieren möchten.

Um Blöcke für alle Nutzer*innen zu deaktivieren, gibt es unterschiedliche Lösungen, entweder in JavaScript mit einer Liste erlaubter oder verbotener Blöcke oder in PHP. Wenn die JavaScript-Funktion eingesetzt wird, werden die Blöcke so deaktiviert, dass bei bereits eingefügten Blöcken der Hinweis angezeigt wird, dass dieser Block nicht unterstützt wird. Die PHP-Variante verbirgt die Blöcke nur im Inserter, sodass sie nicht eingefügt werden können, bereits eingefügte Blöcke aber ohne Hinweis bestehen bleiben.

In dem folgenden Beispiel werden Sie den Weg über PHP gehen und eine leicht veränderte Funktion nutzen, die in Abschnitt 24.5.4 genutzt wurde, um Blöcke nur in bestimmten Inhaltstypen und dem Website-Editor zu erlauben. In Listing 24.40 ist der Code abgedruckt, der den Pullquote- und den Vers-Block deaktiviert.

```
function filter_allowed_blocks( $allowed_block_types ) {
 if ( false === $allowed_block_types ) {
  return $allowed_block_types;
 }

 $blocks_to_remove = [
  'core/pullquote',
  'core/verse',
 ];
 if ( is_array( $allowed_block_types ) ) {
  foreach ( $allowed_block_types as $key => $block_type ) {
   if ( ! in_array( $block_type, $blocks_to_remove, true ) ) {
    continue;
   }

   unset( $allowed_block_types[ $key ] );
  }
  return $allowed_block_types;
 }

 $tmp = WP_Block_Type_Registry::get_instance()->get_all_registered();

 $allowed_block_types = [];
 foreach ( $tmp as $block_name => $block_object ) {
  if ( in_array( $block_name, $blocks_to_remove, true ) ) {
   continue;
  }

  $allowed_block_types[] = $block_name;
 }

 return $allowed_block_types;
}
add_filter(
 'allowed_block_types_all',
 __NAMESPACE__ . '\\filter_allowed_blocks', 10
);
```

Listing 24.40 Deaktivierung des Pullquote- und des Vers-Blocks für alle Benutzer*innen

Innerhalb der Funktion wird geprüft, ob gar keine Blöcke erlaubt sind, und in dem Fall wird `$allowed_block_types` direkt zurückgegeben. Anschließend wird ein Array der zu entfernenden Blöcke definiert und überprüft, ob `$allowed_block_types` ein Array ist, also nur bestimmte Blöcke erlaubt sind. In diesem Fall wird das Array durchlaufen und nach den Blöcken aus `$blocks_to_remove` durchsucht. Wenn einer davon vorkommt, wird der entsprechende

Eintrag aus dem Array entfernt, und nach dem Durchlauf aller Blöcke des Arrays wird das veränderte Array zurückgegeben.

Ist `$allowed_block_types` kein Array und nicht `false`, muss ein Array aller registrierten Blöcke erstellt werden, abzüglich der aus `$blocks_to_remove`. Dafür wird mit `WP_Block_Type_Registry::get_instance()->get_all_registered()` das Array aller Block-Objekte geholt, die in einer Schleife durchlaufen werden, in der ein Array mit den Block-Slugs zusammengebaut wird. Blöcke aus dem Array zu entfernender Blöcke werden übersprungen.

24.11.7 Konfiguration bestehender Blöcke anpassen

Dank der *block.json*-Datei, die seit WordPress 5.8 der empfohlene Weg zur Konfiguration eines Blocks ist und von allen Core-Blöcken genutzt wird (und von Drittanbieter-Blöcken genutzt werden sollte, die auf dem aktuellen Stand sind), kann die Konfiguration eines Blocks recht einfach über PHP angepasst werden. Dabei lassen sich alle Werte verändern, die in der *block.json* gesetzt werden können (*https://developer. wordpress.org/block-editor/reference-guides/block-api/block-metadata/*), das heißt, Sie können zum Beispiel für einen Block die Unterstützung zusätzlicher vom Core bereitgestellter Optionen aktivieren oder, wenn Sie möchten, theoretisch sogar das Editor-Script austauschen.

Der entscheidende Filter dafür ist `block_type_metadata`, mit dem vor der Registrierung die Metadaten jedes Blocks verändert werden können, der mit einer *block.json* konfiguriert wurde.

Der Block-Editor erlaubt nach und nach immer mehr Änderungen an einzelnen Style-Eigenschaften, manchmal möchten Sie aber vielleicht bei Blöcken noch zusätzliche Optionen hinzufügen oder

entfernen, so wird in Listing 24.41 die Text-Dekoration-Option beim Überschriften-Block entfernt.

```
add_filter(
 'block_type_metadata',
 function( $metadata ) {
  $block_name = $metadata['name'] ?? null;
  if ( 'core/heading' !== $block_name ) {
   return $metadata;
  }

  $metadata['supports']['typography']['__experimentalTextDecoration'] = false;
  return $metadata;
 },
 10,
 2
);
```

Listing 24.41 Code zum Entfernen der Option zur Änderung der Text-Dekoration für den Überschriften-Block

1. Wie bereits erwähnt, kommt der Filter `block_type_metadata` zum Einsatz, der an die Callback-Funktion ein Array mit den Metadaten eines Blocks aus dessen *block.json* übergibt.
2. Um nur den Überschriften-Block anzusprechen, testen Sie zunächst, ob der `name`-Eintrag aus dem Array etwas anderes enthält als `core/heading`, um die Metadaten für diese Fälle unverändert zurückzugeben.
3. Handelt es sich dabei um den Überschriften-Block, entfernen Sie mit `$metadata ['supports']['typography'] ['__experimentalTextDecoration'] = false` die Unterstützung für die Text-Dekoration. Wie der Name `__experimentalTextDecoration` vermuten lässt, ist die Implementierung bisher nicht final, das heißt, es könnte sich da noch etwas ändern, und es wird sich auf jeden Fall in Zukunft der Schlüssel ändern, mit dem die Funktion in der *block.json* angesprochen werden kann (vermutlich auf `textDecoration`). Sie werden den Code dann also noch mal anpassen müssen. Wie die

Array-Keys für solche experimentellen Funktionen heißen, können Sie dem Code von Core-Blöcken entnehmen, für diese Option zum Beispiel dem Navigations-Block (*https://github.com/WordPress/gutenberg/blob/wp/6.7/packages/block-library/src/heading/block.json#L72*). Die offiziellen Optionen finden Sie auf der anfangs verlinkten Handbuchseite.

24.11.8 Shortcode erstellen

Ein *Shortcode* ist ein Platzhalter, der in Plugins oder im WordPress-Core registriert und dann von Nutzer*innen im Inhalt eingefügt werden kann. Ein typisches Beispiel dafür ist `[gallery]` zur Darstellung der zu einem Beitrag oder einer Seite hochgeladenen Bilder als Galerie. Bei der Ausgabe des Inhalts auf der Website wird der Shortcode nach den Vorgaben des Plugins, in dem er registriert wurde, in HTML umgewandelt.

Bis zur Einführung des Block-Editors waren Shortcodes die einzige Möglichkeit im WordPress-Core, mit der komplexere Inhaltselemente ohne direktes Einfügen von HTML-Code im Editor erzeugt werden konnten. Viele Page-Builder haben im Hintergrund auf Shortcodes gesetzt (und tun es immer noch).

Seit der Block-Editor Einzug gehalten hat, fristen Shortcodes eher ein Schattendasein, was generell auch gut ist, da Blöcke einfacher einzubauen und zu konfigurieren sind als Shortcodes. Blöcke können Benutzer*innen eine deutlich besser nutzbare Möglichkeit geben, komplexere Inhaltselemente zu erzeugen, da nicht mehr ein Code in der richtigen Art und Weise mit den notwendigen Attributen eingefügt werden muss, sondern beispielsweise Felder eines Blocks ausgefüllt werden können.

Aber nicht alle können oder wollen einen Block erstellen, um ein Inhaltselement umzusetzen. Deshalb lernen Sie in diesem Abschnitt, wie Shortcodes erzeugt werden können.

Selbstschließenden Shortcode erstellen

Als erstes Beispiel erstellen Sie einen Shortcode, der nur aus einem einzelnen Tag besteht und einen Teaser für eine Seite oder einen Beitrag darstellt, wobei der Titel verlinkt wird. Dazu muss mindestens das `id`-Attribut mit der ID des Inhalts übergeben werden, für den der Teaser ist. Als optionale Attribute sollen die Nutzer*innen die Möglichkeit haben, mit `title` und `content` einen angepassten Titel und einen Anreißer zu übergeben.

Beispielhaft soll die Anwendung so aussehen, wie in folgendem Listing dargestellt.

```
[wh_post_teaser id="17" title="Angepasster Titel" content="Der angepasste
Anreißertext"]
```

Listing 24.42 So könnte der Shortcode in eine Seite oder einen Beitrag eingefügt werden, um einen Teaser für den Inhalt mit der ID 17 zu erstellen.

Im Frontend kann das Ergebnis aussehen wie in Abbildung 24.25. Für den praktischen Einsatz bietet es sich an, den Teaser visuell hervorzuheben, damit er im übrigen Inhalt nicht untergeht.

Listing 24.43 zeigt den Code, der für die Erstellung des Shortcodes notwendig ist, Sie können ihn in eine beliebige Datei eines Plugins einfügen, solange diese in die Haupt-Plugin-Datei eingebunden wird.

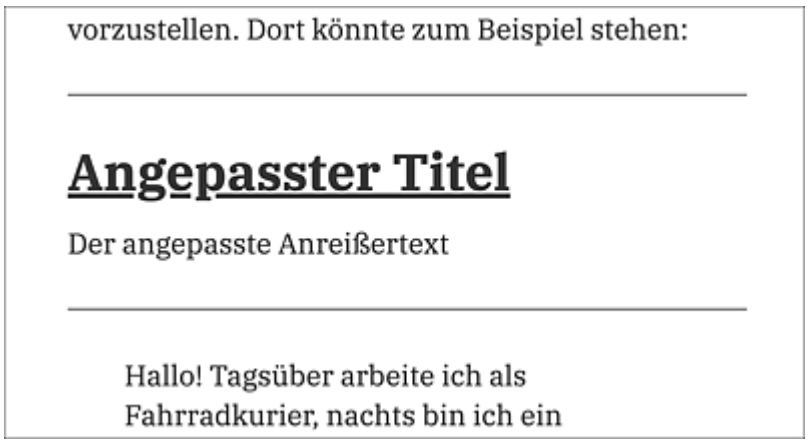


Abbildung 24.25 Ein Beitrag mit einem Post-Teaser für den Hallo-Welt-Beitrag, zur besseren Unterscheidung mit einem visuellen Trenner oben und unten

```
add_action( 'init', function() {
    add_shortcode( 'wh_post_teaser', function( $atts ) {
        $post_id = $atts['id'] ?? 0;
        if ( empty( $post_id ) ) {
            return '';
        }

        $post_to_show = get_post( $post_id );
        if ( $post_to_show === null ) {
            return '';
        }

        $default_heading_level = 2;

        $atts = shortcode_atts(
            array(
                'title' => $post_to_show->post_title,
                'content' => get_the_excerpt( $post_to_show ),
                'heading_level' => $default_heading_level
            ),
            $atts
        );

        if ( empty( $atts['title'] ) ) {
            return '';
        }

        $heading_level = $atts['heading_level'];
        if ( ! is_numeric( $heading_level ) ) {
            $heading_level = $default_heading_level;
        }

        return sprintf(
            '<div class="wh-post-teaser">
                <h%1$s class="wh-post-teaser-title"><a href="%2$s">%3$s</a>
                </h%1$s>
```

```
                <div class="wh-post-teaser-content">
                    %4$s
                </div>
            </div>',
            $heading_level,
            get_the_permalink( $post_to_show ),
            $atts['title'],
            $atts['content'] !== '' ? apply_filters( 'the_content',
            $atts['content'] ) : ''
        );
    } );
} );
```

Listing 24.43 Code für das Hinzufügen des Teaser-Shortcodes

1. Die anonyme Funktion zur Registrierung des Shortcodes wird an den `init`-Hook angehängt.
2. Um einen Shortcode zu erstellen, gibt es die Funktion `add_shortcode()`, die als ersten Parameter den Namen des Shortcodes erhält und als zweiten die Funktion, die sich um die Umwandlung des Shortcodes in HTML kümmert (Sie verwenden hier eine weitere anonyme Funktion). Dieser Funktion wird als erster Parameter ein Array `$atts` mit den Shortcode-Attributen übergeben und als zweiter und dritter Parameter (die hier nicht zum Einsatz kommen) der Inhalt, der zwischen dem öffnenden und schließenden Shortcode-Tag steht, sowie der Shortcode-Tag selbst.
3. Um den Teaser auf einen Inhalt zu verlinken, muss dem Shortcode ein `id`-Attribut mit der ID des Beitrags oder der Seite mitgegeben werden. Innerhalb des Shortcode-Callbacks wird deshalb zunächst geprüft, ob das `id`-Attribut vorhanden und nicht leer ist und ob über `get_post()` zu der ID ein Inhalt ermittelt werden kann. Wenn etwas davon nicht zutrifft oder nicht erfolgreich ist, wird ein leerer String als Ergebnis des Shortcodes zurückgegeben, da kein sinnvolles Markup erzeugt werden kann.

4. Über die Funktion `shortcode_atts()` wird ein Array von Standardwerten für die anderen möglichen Shortcode-Attribute angegeben, die Sie anbieten möchten, und mit den übergebenen Shortcode-Attributen zusammengeführt. Wenn also beispielsweise kein Titel über das `title`-Attribut angegeben wurde, wird der Titel des über die ID ermittelten Inhalts genutzt. Das Ergebnis wird in dem `$atts`-Array gespeichert, sodass die Array-Schlüssel `title`, `content` und `heading_level` auf jeden Fall vorhanden sind – entweder mit den von den Usern übergebenen Werten oder den Standardwerten.
5. Wenn ein leerer Titel übergeben wurde oder der angeteaserte Beitrag keinen Titel enthält, geben Sie erneut einen leeren String zurück, denn ohne Titel ergibt das Teaser-Element keinen Sinn.
6. Im Anschluss wird geprüft, ob der Wert von `$atts['heading_level']` nicht numerisch ist, um in diesem Fall den Standardwert 2 zu nutzen. Abschließend wird das Markup zusammengebaut und zurückgegeben.

Sie können den Shortcode jetzt, wie anfangs beispielhaft gezeigt, mit dem Shortcode-Block nutzen und überall einbauen.

Achtung: Shortcode-Callbacks dürfen das Ergebnis nicht direkt ausgeben

Wichtig ist, dass Shortcode-Callback-Funktionen das Ergebnis des Shortcodes mittels `return` zurückgeben und nicht direkt per `echo` oder ähnlich ausgeben.

Shortcode mit Start- und End-Tag erstellen

Der Shortcode, den Sie im vorigen Abschnitt erstellt haben, besteht nur aus einem Tag und Attributen. Das ist aber lediglich eine Möglichkeit, einen Shortcode zu erstellen. Die zweite Möglichkeit ist, dass der Shortcode einen Inhalt mit Start- und End-Tag umschließt. Als einfaches Beispiel werden Sie einen Shortcode umsetzen, bei dem das Start- und das End-Tag mit einem `mark`-Element ersetzt wird, sodass der Inhalt zwischen den Shortcode-Tags im Frontend visuell hervorgehoben ist. Wie die Anwendung des Shortcodes im Editor aussehen kann, ist nachfolgend zu sehen.

```
[wh_mark]Im Frontend hervorgehobener Text[/wh_mark]
```

Abbildung 24.26 zeigt, wie das Ergebnis des Shortcodes im Frontend mit dem Twenty-Twenty-Five-Theme aussieht.

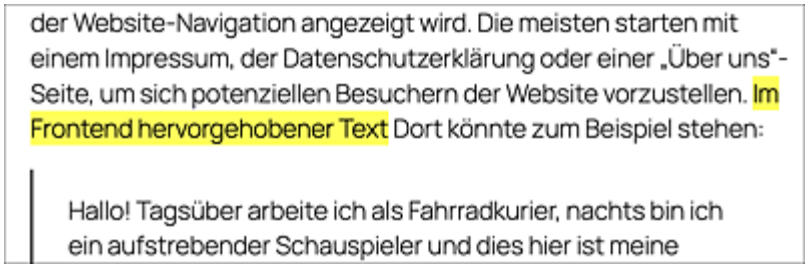


Abbildung 24.26 Nicht schön, aber deutlich hervorgehoben: das Ergebnis des Shortcodes im Frontend

Im Vergleich zum anderen Shortcode ist dieser hier sehr einfach, daher hält sich auch der dafür notwendige Code in Grenzen, der in Listing 24.44 abgebildet ist.

```
add_shortcode( 'wh_mark', function( $atts, $content ) {
    return sprintf(
        '<mark>%s</mark>',
        $content
    );
} );
```

Listing 24.44 Code für den Markierungs-Shortcode

Hier wird der zweite Parameter in der Callback-Funktion genutzt: In `$content` ist enthalten, was zwischen den beiden Shortcode-Tags steht. Im einfachsten Fall ist es nur Text, es kann aber natürlich auch HTML-Markup sein. Diesen Inhalt umschließen Sie mit einem `mark`-Element und geben ihn zurück.

Achtung: Vorsicht vor Shortcodes für umfangreiche Veränderungen

Beim Einsatz von Shortcodes gilt es eine Sache besonders zu beachten. Wenn Sie das Plugin deaktivieren, das einen oder mehrere Shortcodes hinzufügt, die Sie auf Ihrer Website eingesetzt haben, werden diese Shortcodes einfach im Frontend ausgegeben, also ohne die Umwandlung in das Markup.

Bei Anwendungsfällen, in denen der Shortcode nur ein paar Mal eingesetzt wird und vielleicht auch nur Kleinigkeiten anpasst, wie etwa im Markierungsbeispiel aus Listing 24.44, ist das nicht so wild, weil die Shortcodes relativ schnell händisch durch eine andere Lösung ersetzt oder entfernt werden können. Wenn Sie aber beispielsweise einen Page-Builder einsetzen, der die Layouts mit Shortcodes umsetzt, dann haben Sie je nach Umfang Ihrer Website viel Arbeit vor sich, um die Shortcodes zu entfernen und das Layout neu aufzubauen.

Streng genommen haben Sie das Problem mit Inhalten, die dann nicht mehr angezeigt werden, auch bei dynamischen Blöcken, die das Markup nicht direkt in der Datenbank speichern. Wenn der Block aber außer dem Block-Kommentar gar keinen Inhalt speichert, wie der Neueste-Beiträge-Block, wird immerhin dann direkt gar nichts ausgegeben statt nicht umgewandelten Codes.

24.11.9 Eigene Taxonomie erstellen

In Abschnitt 24.3 haben Sie bereits gelernt, wie ein Inhaltstyp erstellt werden kann. Um Elemente von Inhaltstypen zusammenzufassen oder zu gruppieren, gibt es *Taxonomien* bzw. die *Terme* einer Taxonomie (Kategorien in WordPress sind eine Taxonomie, die Allgemein-Kategorie ist ein Term).

Und wie einen Inhaltstyp können Sie auch eine eigene Taxonomie hinzufügen. In Listing 24.45 ist abgebildet, wie dem `talk`-Inhaltstyp aus Abschnitt 24.3 eine Taxonomie hinzugefügt werden kann, um die Zielgruppe von Vorträgen festzulegen.

```
add_action(
 'init',
 function() {
  $labels = [
   'name' => __( 'Target groups', 'wh-talks' ),
   'singular_name' => __( 'Target group', 'wh-talks' ),
   'search_items' => __( 'Search target groups', 'wh-talks' ),
   'all_items' => __( 'All target groups', 'wh-talks' ),
   'parent_item' => __( 'Parent target group', 'wh-talks' ),
   'edit_item' => __( 'Edit target group', 'wh-talks' ),
   'view_item' =>__( 'View target item', 'wh-talks' ),
   'update_item' => __( 'Update target group', 'wh-talks' ),
   'add_new_item' => __( 'Add new target group', 'wh-talks' ),
   'new_item_name' => __( 'New target group name', 'wh-talks' ),
   'not_found' => __( 'No target groups found', 'wh-talks' ),
   'no_terms' => __( 'No target groups', 'wh-talks' ),
   'filter_by_item' => __( 'Filter by target group', 'wh-talks' ),
   'item_link' => __( 'Target group link', 'wh-talks' ),
   'item_link_description' => __( 'A link to a target group', 'wh-talks' ),
  ];
  $args = [
   'labels'       => $labels,
   'public'       => true,
   'hierarchical' => true,
   'show_in_rest' => true,
   'rewrite'      => [ /* translators: slug for target group taxonomy rewrite */
    'slug' => __( 'target-group', 'wh-talks' ),
   ]
  ];
  register_taxonomy( 'target_group', 'talk', $args );
 }
);
```

Listing 24.45 Registrierung einer »target_group«-Taxonomie für die Vorträge

1. Innerhalb der Callback-Funktion, die an den `init`-Hook gehängt wird, wird zunächst ein Array mit einigen Beschriftungen erstellt, ähnlich wie bei der Registrierung des Inhaltstyps (alle möglichen Labels finden Sie in der Doku zur `get_taxonomy_labels()`-Funktion unter *https://developer.wordpress.org/reference/functions/get_taxonomy_labels/*).
2. Danach wird das Array mit den Argumenten für die Taxonomie erstellt. Es beinhaltet das Array mit den Beschriftungen (`labels`) und die Angabe, dass die Taxonomie öffentlich (`public`) sowie hierarchisch sein soll, das bedeutet, dass Sie Eltern-Kind-Beziehungen wie bei Kategorien erstellen können. Mit `show_in_rest` legen Sie fest, dass die Taxonomie in der REST-API sichtbar sein soll, und in `rewrite` definieren Sie einen übersetzbaren Slug, der für die Permalinks der Term-Archive genutzt wird.
3. Abschließend wird mit `register_taxonomy()` die Taxonomie `target_group` für den Inhaltstyp `talk` registriert mit den vorher festgelegten Argumenten als dritten Parameter.

Wenn Sie den Code mit in das Talks-Plugin eingebaut und aktiviert haben, sollten Sie im Backend links im Admin-Menü als Unterpunkt zu dem Inhaltstyp-Menüpunkt den neuen Menüpunkt für die Taxonomie angezeigt bekommen (siehe Abbildung 24.27).

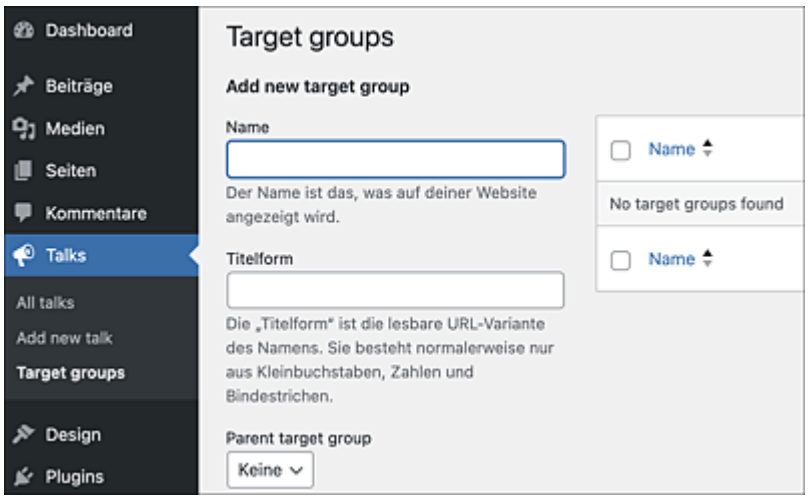


Abbildung 24.27 Die neue Taxonomie im Backend

24.11.10 Einstellungen im Backend einfügen

Einige Plugins kommen ganz ohne Einstellungen aus (beispielsweise das Talks-Plugin aus diesem Kapitel), viele bringen aber die Möglichkeit zur Konfiguration mit. Die Einstellungen sind dann entweder auf einer oder mehreren eigenen Seiten im Backend untergebracht oder auf einer bestehenden Seite integriert. In diesem Abschnitt werden Sie beide Varianten kennenlernen, und für beide nutzen Sie die *Settings API* von WordPress.

Die Nutzung der API bringt Ihnen die folgenden Vorteile:

- Ein zu den WordPress-Core-Einstellungen passendes Erscheinungsbild.
- Die Implementierung der Optionen ist zukunftssicher.
- WordPress kümmert sich um das sichere Abschicken und Speichern der Daten durch die Nutzung von Nonces (zufälligen Werten, die beim Verarbeiten eines Formulars geprüft werden, um sicherzustellen, dass die Daten von der Einstellungsseite abgeschickt wurden) und stellt die Möglichkeit bereit, einen

Sanitize-Callback zu definieren, um die Daten vor dem Speichern zu bereinigen.

Lernen Sie anhand der nächsten Codebeispiele, wie sich die API einsetzen lässt.

Option auf vorhandener Seite einfügen

Beginnen Sie mit der einfacheren Variante, nämlich eine Option zu einer vorhandenen Einstellungsseite von WordPress hinzuzufügen. Das bietet sich an, wenn Sie nur eine überschaubare Anzahl Optionen haben, die thematisch zu einer der bestehenden Einstellungsseiten passt.

In Listing 24.46 sehen Sie den Code, mit dem Sie eine Option zu der Seite Einstellungen • Schreiben hinzufügen können.

```
function settings_init() {
 $option_name = 'wh_talks_hide_blocks';
 register_setting(
  'writing',
  $option_name,
  [
   'type'              => 'boolean',
   'sanitize_callback' => function( $value ) {
    return 'on' === $value;
   }
  ]
 );

 add_settings_section(
  'wh_talks_settings_section',
  __( 'Talks settings', 'wh-talks' ),
  __NAMESPACE__ . '\\settings_section_callback',
  'writing'
 );

 add_settings_field(
  $option_name,
  __( 'Hide blocks', 'wh-talks' ),
  __NAMESPACE__ . '\\hide_field_callback',
  'writing',
  'wh_talks_settings_section',
  [
```

```
    'option_name' => $option_name
   ]
  );
}
add_action( 'admin_init', __NAMESPACE__ . '\\settings_init' );

function settings_section_callback() {
 $section_description = __( 'Talk-related settings', 'wh-talks' );
 echo "<p>$section_description</p>";
}

function hide_field_callback( $args ) {
 $current_value = (bool) get_option( 'wh_talks_hide_blocks', false );
 printf(
  '<input type="checkbox" name="%1$s" id="%1$s" %2$s>
  <label for="%1$s">%3$s</label>',
  $args['option_name'],
  checked( $current_value, true, false ),
  __(
   'Hide the blocks for displaying talk metadata in the editor',
   'wh-talks'
  )
 );
}
```

Listing 24.46 Hinzufügen einer Option zu den »Schreiben«-Einstellungen

1. In der Funktion `settings_init()`, die an den `admin_init`-Hook gehängt wird, rufen Sie verschiedene Funktionen der Settings-API auf, um die Einstellung zu registrieren. Am Anfang speichern Sie den Namen der Option in der Variablen `$option_name`, damit Sie ihn später flexibel wiederverwenden können.
2. Die Funktion `register_setting()` ist dafür zuständig, eine Option zu registrieren. Nur so registrierte Optionen können mit der Settings-API abgespeichert werden. Der erste Parameter ist der Name einer Gruppe von Optionen, standardmäßig gibt es die Gruppen `general`, `discussion`, `media`, `reading`, `writing` und `options`. Der zweite Parameter ist der Name der Option, wie sie in der `wp_options`-Tabelle in der Datenbank gespeichert werden soll. Der dritte Parameter ist ein optionales Array, in dem im Beispiel der Datentyp der Option auf `boolean` gesetzt wird und

mit dem `sanitize_callback` der übergebene Wert aus dem Formular vor dem Speichern in der Datenbank bereinigt wird (die Option wird durch eine Checkbox dargestellt, und wenn die angehakt ist, wird als Wert `on` übergeben).

3. Mit der Funktion `add_settings_section()` können Sie einen neuen Bereich für Optionen auf einer Einstellungsseite erstellen. Der erste Parameter ist die ID des Bereichs, über die ihm später Felder zugewiesen werden können. Als zweiten Parameter übergeben Sie die Überschrift des Bereichs und als dritten eine Callback-Funktion, deren Ausgabe unter der Überschrift und über dem ersten Feld des Bereichs ausgegeben wird (die Callback-Funktion wird gleich besprochen). Der vierte Parameter enthält den Slug der Einstellungsseite, auf der der Bereich angezeigt werden soll. Den Slug können Sie unter anderem herausfinden, indem Sie die Seite mit den Einstellungen öffnen, auf der Sie die Option einfügen möchten, und sich die URL anschauen, zum Beispiel *example.com/wp-admin/options-writing.php*. Der Teil nach options- ohne die Dateiendung ist der Slug, der angegeben werden muss.

4. Über `add_settings_field()` fügen Sie schließlich das Feld hinzu, mit dem die User die Option nutzen können. Der erste Parameter ist die ID des Felds, der zweite die Beschriftung, die in den Einstellungen in der linken Spalte angezeigt wird (siehe Abbildung 24.28), der dritte ist eine Callback-Funktion, die das Feld anzeigt, der vierte Parameter gibt die Einstellungsseite an, der fünfte den Bereich auf der Seite und der sechste ein Array an zusätzlichen Daten. Dieses Array wird an den Callback übergeben, der das Feld anzeigt, und kann mit beliebigen Einträgen gefüllt werden. Von WordPress verarbeitet werden die Array-Keys `label_for` und `class`. Wenn `label_for` angegeben ist, wird die Beschriftung des Felds in der linken Spalte in ein `label`-

Element gepackt mit dem Wert, der für `label_for` angegeben wurde, als `for`-Attribut. Mit `class` können Sie der Tabellenzeile der Option eine Klasse geben.

5. Die Funktion `settings_section_callback()` wurde als Callback an `add_settings_section()` übergeben und wird genutzt, um einen einleitenden Absatz über den Feldern anzuzeigen (siehe Abbildung 24.28). Sie können die Funktion auch einfach leer lassen, wenn Sie nichts ausgeben möchten.

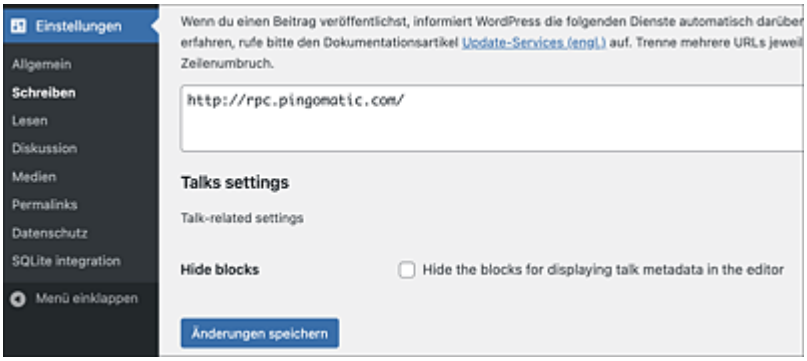


Abbildung 24.28 Die neue Option auf der bestehenden Einstellungsseite

6. In `hide_field_callback()`, der Callback-Funktion von `add_settings_field()`, wird das Feld-Markup ausgegeben. Wie beschrieben, erhält die Funktion als Parameter das Array, das als letzter Parameter an `add_settings_field()` übergeben wurde.

   In der Funktion wird zuerst der aktuelle Wert der Option ermittelt, als Standardwert soll `false` genutzt werden, wenn die Option nicht gesetzt ist (durch den (`bool`)-Type-Cast erreichen Sie, dass die in der Datenbank für `true` und `false` gespeicherten Werte `1` und `0` wieder als `true` und `false` vorliegen). Danach wird mittels `printf()` eine Checkbox ausgegeben, die als Wert für das `name`-Attribut den Namen der Option bekommen muss, damit das Speichern des Werts funktioniert. Den Optionsnamen nutzen Sie der Einfachheit halber auch für das `id`-Attribut und

entsprechend das `for`-Attribut des `label`-Elements, um die Verknüpfung zwischen Beschriftung und Feld vorzunehmen. Mit `checked()` geben Sie für das `input`-Element das `checked`-Attribut aus, wenn die Option `true` ist.

Wenn Sie diesen Code zum Beispiel in das Talks-Plugin einbauen und auf der Seite EINSTELLUNGEN • SCHREIBEN ganz nach unten scrollen, sollten Sie die Option sehen, wie in Abbildung 24.28 gezeigt.

Auf den Wert der Option können Sie jetzt an einer beliebigen Stelle mit `(bool) get_option( 'wh_talks_hide_blocks', false )` zugreifen.

Eigene Seite

Wenn Sie viele Optionen anzeigen müssen oder Ihre Optionen zu keiner der vorhandenen Einstellungsseiten passt, können Sie eine eigene Seite erstellen. Im folgenden Beispiel werden Sie die Seite als Unterpunkt zu dem Menüpunkt EINSTELLUNGEN einfügen, Sie können aber auch andere Menüpunkte als Oberpunkte wählen oder, sollte es angebracht sein, einen eigenen neuen Oberpunkt erstellen (das sollten Sie nur tun, wenn durchaus häufiger auf die Einstellungen zugegriffen werden muss und nicht nur einmal bei der Einrichtung, damit die obere Menüebene nicht unnötig unübersichtlich wird).

Den gesamten Code für die Einstellungsseite finden Sie in Listing 24.47.

```
function add_options_page() {
 add_submenu_page(
  'options-general.php',
  __( 'Talks options', 'wh-talks' ),
  __( 'Talks options', 'wh-talks' ),
  'manage_options',
  'wh_talks',
  __NAMESPACE__ . '\\render_options_page'
 );
}
add_action( 'admin_menu', __NAMESPACE__ . '\\add_options_page' );
```

```php
function render_options_page() {
 if ( ! current_user_can( 'manage_options' ) ) {
  return;
 }
 ?>
 <div class="wrap">
  <h1><?php echo esc_html( get_admin_page_title() ); ?></h1>
  <form action="options.php" method="post">
   <?php
   settings_fields( 'wh_talks' );
   do_settings_sections( 'wh_talks' );
   submit_button( 'Save Settings' );
   ?>
  </form>
 </div>
 <?php
}

function settings_init() {
 $option_name = 'wh_talks_hide_blocks';
 register_setting(
  'wh_talks',
  $option_name,
  [
   'type'              => 'boolean',
   'sanitize_callback' => function( $value ) {
    return 'on' === $value;
   }
  ]
 );

 add_settings_section(
  'wh_talks_settings_section',
  __( 'Block visibility', 'wh-talks' ),
  __NAMESPACE__ . '\\settings_section_callback',
  'wh_talks'
 );

 add_settings_field(
  $option_name,
  __( 'Hide blocks', 'wh-talks' ),
  __NAMESPACE__ . '\\hide_field_callback',
  'wh_talks',
  'wh_talks_settings_section',
  [
   'option_name' => $option_name,
   'label_for' => true,
  ]
 );
}
add_action( 'admin_init', __NAMESPACE__ . '\\settings_init' );

function settings_section_callback() {
```

```
}

function hide_field_callback( $args ) {
 $current_value = (bool) get_option( 'wh_talks_hide_blocks', false );
 printf(
  '<input type="checkbox" name="%1$s" id="%1$s" %2$s>
  <label for="%1$s">%3$s</label>',
  $args['option_name'],
  checked( $current_value, true, false ),
  __(
   'Hide the blocks for displaying talk metadata in the editor',
   'wh-talks'
  )
 );
}
```

Listing 24.47 Der Code für die Erstellung einer eigenen Einstellungsseite

1. In der `add_options_page()`-Funktion, die an den Hook `admin_menu` angehängt wird, fügen Sie mit `add_submenu_page()` die Unterseite zum Menü hinzu. Dafür geben Sie als ersten Parameter den Dateinamen der Oberseite an, der zweite Parameter ist der Seitentitel, der dritte der Titel für den Menüeintrag, der vierte ist die Berechtigung (`manage_options`), die Benutzer*innen haben müssen, um auf die Seite zugreifen zu können. Als fünften Parameter übergeben Sie den Menü-Slug für die Seite und als letzten Parameter die Callback-Funktion, die sich um die Darstellung der Seite kümmert.
2. In der Callback-Funktion `render_options_page()` wird zunächst geprüft, ob jemand mit der notwendigen Berechtigung die Seite aufruft, ansonsten wird nichts ausgegeben. Ist die Berechtigung vorhanden, wird der Titel der Seite mit `get_admin_page_title()` als `h1` ausgegeben und innerhalb eines Formulars die Felder. Wichtig ist der Aufruf von `settings_fields()`, denn die Funktion fügt ein paar Felder ein, die dafür sorgen, dass das Speichern funktioniert. Unter anderem ist auch ein Nonce-Feld dabei, das bei der Verarbeitung der Formulardaten nach dem Speichern überprüft wird. Der übergebene Parameter muss

dem Wert entsprechen, der als erster Parameter für `register_setting()` genutzt wird. Mit der Funktion `do_settings_sections()` geben Sie die mit `add_settings_section()` registrierten Bereiche für die Seite `wh_talks` aus und mit `submit_button()` einen Button zum Speichern des Formulars.

3. Es folgen die bereits aus dem anderen Beispiel bekannten Funktionen `settings_init()`, `settings_section_callback()` und `hide_field_callback()`. In `settings_init()` wurden die Seite und die Optionsgruppe `writing` jeweils durch `wh_talks` ersetzt, und der Titel des Settings-Bereichs wurde angepasst. In `settings_section_callback()` wird keine Einleitung mehr ausgegeben, der Rest ist gleich geblieben.

Wenn Sie den Code in einem Plugin einfügen, müssten Sie im Menü den Punkt Einstellungen • Talks options und nach Anklicken die Seite sehen, die in Abbildung 24.29 gezeigt wird.

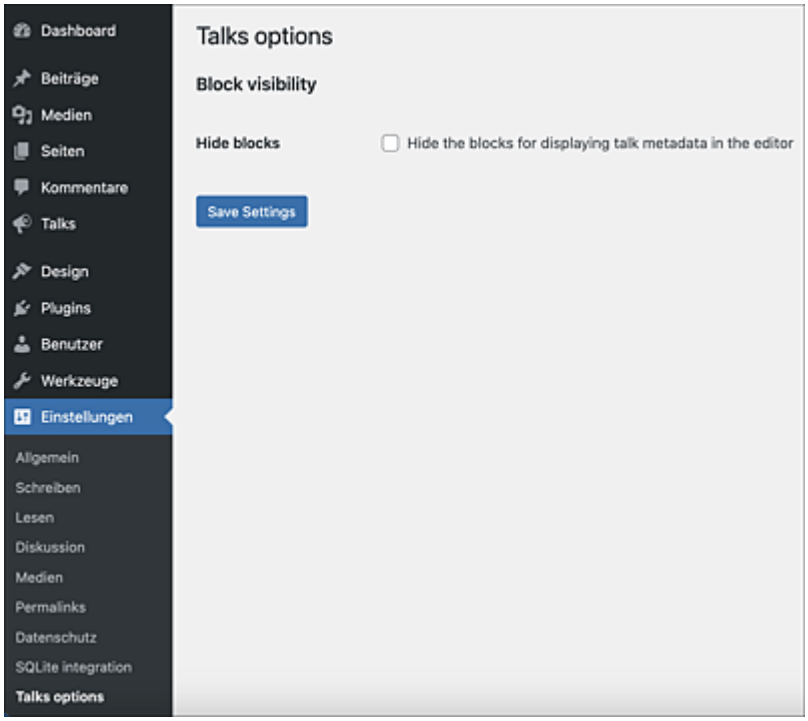


Abbildung 24.29 Eine eigene Seite bietet sich an, wenn ein paar mehr Optionen angezeigt werden sollen als in diesem Beispiel.

Den Wert der Option könnten Sie jetzt in der Funktion aus Abschnitt 24.11.6 nutzen, um die beiden Blöcke des Plugins zu deaktivieren, wenn die Option gesetzt ist.

REST-API erweitern

Die WordPress-*REST-API* besteht aus unterschiedlichen Endpunkten, die Informationen in Form von einer JSON-Antwort ausgeben, ein einfach zu verarbeitendes und auch recht lesbares Datenformat, das Sie bereits beim Inhalt von *theme.json* und *block.json* gesehen haben. Der Block-Editor nutzt die API eigentlich für alle Funktionen, die es im Editor gibt, sei es zum Anzeigen der Inhalte des Beitrags, zum Abspeichern, zum Holen von Metadaten, zum Ändern eines Menüs mit dem Navigations-Block etc.

Für Sie bietet die REST-API Möglichkeiten zur Erweiterung, sodass Sie beispielsweise zusätzliche Informationen zu den Einträgen des `talk`-Inhaltstyps aus dem in diesem Kapitel entwickelten Plugin bereitstellen könnten oder direkt einen Markup-String, den Sie für die Darstellung eines Blocks benötigen.

In diesem Abschnitt werden Sie lernen, wie Sie ein bestehendes Objekt verändern können, das von REST-Endpunkten zurückgegeben wird, wie Sie den zusätzlichen Eintrag für ein Objekt nur ausgeben, wenn ein bestimmter Parameter gesetzt ist, und wie Sie einen eigenen REST-Endpunkt erstellen können.

Vorhandenes Objekt verändern

Als Erstes beginnen Sie damit, dem Objekt der Elemente des `talk`-Inhaltstyps ein zusätzliches Feld hinzuzufügen. Dieses zusätzliche Feld werden Sie dann in der REST-API immer angezeigt bekommen, wenn von einem Endpunkt ein oder mehrere Objekte für den Inhaltstyp zurückgegeben werden, beispielsweise bei der Übersicht unter *example.com/wp-json/wp/v2/talks*. Den Code dazu sehen Sie in Listing 24.48.

```
add_action( 'rest_api_init', function () {
 register_rest_field(
  'talk',
  'wh_talks_teaser',
  [
   'get_callback' => function( $post_obj ) {
    $permalink      = get_the_permalink( $post_obj['id'] );
    $title          = get_the_title( $post_obj['id'] );
    $event_name     = get_post_meta(
     $post_obj['id'],
     'wh_talks_event_name',
     true
    );
    $heading_string = ! empty( $event_name )
     ? "$event_name: $title"
     : $title;

    $post_thumbnail_markup =
     has_post_thumbnail( $post_obj['id'] )
     ? sprintf(
      '<figure class="teaser-image">%s</figure>',
      get_the_post_thumbnail(
       $post_obj['id'],
       'medium-large'
      )
     ) : '';

    return sprintf(
     '<article class="talk-teaser">
     <h2 class="teaser-title"><a href="%s">%s</a></h2>
     %s
     </article>',
     $permalink,
     $heading_string,
     $post_thumbnail_markup
    );
   },
  ]
 );
});
```

Listing 24.48 Registrierung eines zusätzlichen Felds für den REST-Endpunkt der »talk«-Einträge

1. Um die REST-API zu verändern, wird eine anonyme Funktion an den `rest_api_init`-Hook gehängt.
2. In der Funktion wird direkt `register_rest_field()` aufgerufen. Die Funktion erwartet als ersten Parameter den Objekttyp, zu dem das Feld hinzugefügt werden soll, im Beispiel ist es der `talk`-Inhaltstyp. Der zweite Parameter ist der Name des Felds, dem Sie ein Präfix mitgeben, um Namenskollisionen zu vermeiden. Der dritte Parameter ist ein Array mit zusätzlichen Argumenten, im Beispiel wird nur der Array-Key `get_callback` benötigt, in dem die Ausgabe des Felds geregelt wird, wenn es mit einem GET-Request angefragt wird.
3. Der Callback-Funktion von `get_callback` wird das Objekt des Posts übergeben, für den das Feld angezeigt wird. Anhand dessen können Sie die Daten für den richtigen Vortrag zurückgeben.
4. Im Callback werden zunächst verschiedene Daten in Variablen abgespeichert:
   - Der Permalink, an den Sie mit der `get_the_permalink()`-Funktion kommen, der die ID des Posts übergeben wird.
   - Der Titel des Posts, ermittelt mit `get_the_title()`, wobei als Attribut erneut die ID des Posts genutzt wird.
   - Der Name der Veranstaltung, bei der der Vortrag gehalten wurde. Der ist im Meta-Key `wh_talks_event_name` gespeichert, dessen Wert Sie mit `get_post_meta()` auslesen, indem als erster Parameter wieder die Post-ID übergeben wird, als zweiter der Meta-Key und als dritter die Angabe, dass ein einzelner Wert zurückgegeben werden soll.

- In `$heading_string` wird der Wert gespeichert, der als Titel des Teasers angezeigt werden soll. Wenn ein Veranstaltungsname für den Vortrag vergeben wurde, soll dieser dem Namen des Vortrags vorangestellt werden. Falls kein Veranstaltungsname vorhanden ist, wird nur der Name des Vortrags gespeichert.

5. Am Ende der Callback-Funktion wird mittels `sprintf()` das Markup für den Teaser zusammengesetzt und mit `return` zurückgegeben.

Wenn Sie nun den Code beispielsweise in die *inc/cpt.php* des `wh-talks`-Plugins einfügen und mit aktiviertem Plugin bei Ihrer WordPress-Site den Pfad */wp-json/wp/v2/talks* aufrufen, sollten Sie in den zurückgegebenen talk-Objekten den Eintrag `wh_talks_teaser` sehen (vorausgesetzt natürlich, Sie haben mindestens einen Vortrag angelegt). Wie das in Firefox aussehen kann, sehen Sie in Abbildung 24.30, wenn Sie einen anderen Browser nutzen, kann es sein, dass die Ausgabe weniger formatiert ist.

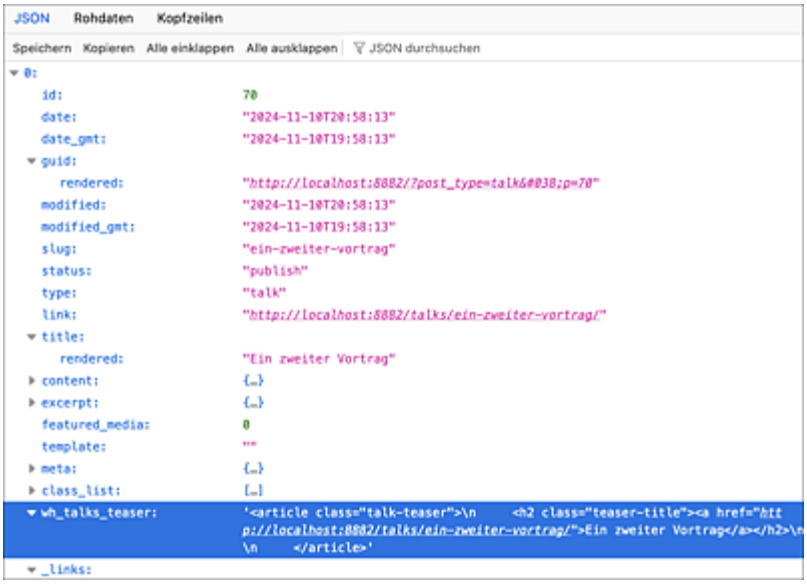


Abbildung 24.30 Das veränderte Objekt eines Vortrags in der REST-API

Feld nur für bestimmten Parameter ausgeben

Wenn Sie Felder zu bestehenden Antwortobjekten hinzufügen, sind diese in den Antworten auf alle Anfragen enthalten, die nicht explizit die Felder angeben, die zurückgegeben werden sollen. Dadurch werden unnötig Daten übertragen und Anfragen an die Datenbank gemacht, was negative Auswirkungen auf die Zeit der Anfrage hat.

Die WP-REST-API stellt das Attribut `_fields` bereit, mit dem die Ausgabe auf bestimmte Felder eingeschränkt werden kann. Da das Markup für einen Teaser eines Vortrags nur in sehr bestimmten Fällen benötigt wird, sollte es standardmäßig nicht mit ausgegeben werden, sondern nur, wenn in `_fields` der Wert `wh_talks_teaser` vorhanden ist.

In Listing 24.49 sehen Sie den angepassten Code der `add_rest_field()`-Funktion, die veränderten Stellen sind hervorgehoben.

```
add_action( 'rest_api_init', function () {
 register_rest_field(
  'talk',
  'wh_talks_teaser',
  [
   'get_callback' => function( $post_obj, $attr, $request ) {
     $params = $request->get_query_params();
     if ( ! isset( $params['_fields'] ) ) {
       return null;
     }

     $fields = wp_parse_list( $params['_fields'] );
     if ( ! in_array( 'wh_talks_teaser', $fields ) ) {
       return null;
     }

     $permalink      = get_the_permalink( $post_obj['id'] );
     $title          = get_the_title( $post_obj['id'] );
     $event_name     = get_post_meta(
       $post_obj['id'],
       'wh_talks_event_name',
       true
     );
```

```
      $heading_string = ! empty( $event_name )
        ? "$event_name: $title"
        : $title;

      $post_thumbnail_markup =
       has_post_thumbnail( $post_obj['id'] )
       ? sprintf(
         '<figure class="teaser-image">%s</figure>',
         get_the_post_thumbnail(
           $post_obj['id'],
           'medium-large'
         )
       ) : '';

     return sprintf(
      '<article class="talk-teaser">
      <h2 class="teaser-title"><a href="%s">%s</a></h2>
      %s
      </article>',
      $permalink,
      $heading_string,
      $post_thumbnail_markup
     );
    },
   ]
  );
});
```

Listing 24.49 Die angepasste »add_rest_field()«-Funktion sorgt dafür, dass das neue Feld nur ausgegeben wird, wenn »_fields« den Wert »wh_talks_teaser« enthält.

Bei der Callback-Funktion für `get_callback` werden nun neben dem Parameter mit dem Post-Objekt noch der zweite und dritte verwendet. Im zweiten (`$attr`) ist der Name des Felds gespeichert, im dritten, auf den Sie es für den Anwendungsfall abgesehen haben, das Request-Objekt.

Innerhalb des Callbacks holen Sie sich mit `$request->get_query_params()` ein Array mit den Query-Parametern der Anfrage-URL. Wenn der `_fields`-Parameter nicht mit dabei ist, geben Sie `null` zurück.

Im nächsten Schritt wandeln Sie die Liste der angegebenen Felder aus `$params['_fields']` mit der `wp_parse_list()`-Funktion in ein Array um und prüfen, ob `wh_talks_teaser` in dem Array enthalten

ist. Wenn das nicht der Fall ist, geben Sie wieder `null` zurück. Ist das Feld mit angegeben, wird der Code ausgeführt, um den Wert für das Feld zu erzeugen. In Abbildung 24.31 sehen Sie das Ergebnis der Anpassung, die entsprechende Anfrage-URL ist */wp-json/wp/v2/talks?_fields=wh_talks_teaser*. Wenn Sie die URL für alle Felder aufrufen, ist das hinzugefügte Feld leer, wie in Abbildung 24.32 zu sehen.



Abbildung 24.31 Die Antwort mit dem Objekt, das auf das »wh_talks_teaser«- Feld eingeschränkt wurde

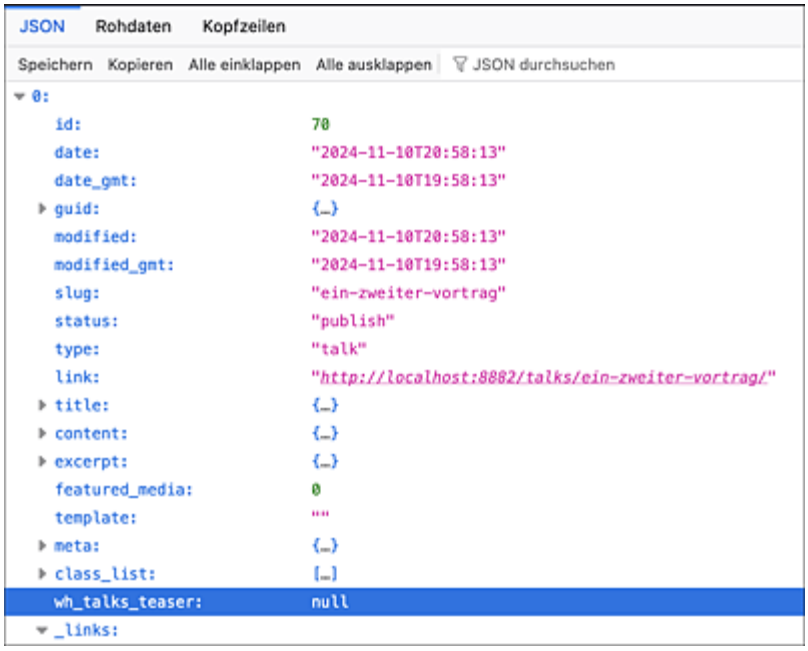


Abbildung 24.32 Wenn Sie den Endpunkt ohne Einschränkung auf das Feld aufrufen, ist das hinzugefügte Feld leer.

Eigenen Endpunkt erstellen

Benötigen Sie einen eigenen REST-Endpunkt, weil Sie keinen bestehenden erweitern möchten oder weil Sie zum Beispiel Endpunkte hinzufügen, mit denen durch einen POST-Request eine Aktion ausgeführt wird, ist auch das relativ einfach umzusetzen.

Sie werden beispielhaft einen Endpunkt umsetzen, mit dem Sie sich den String zu einem der registrierten Metadaten eines Vortrags ausgeben lassen können. Bei den Metawerten zu Veranstaltungsname, Vortragssprache und -dauer ist das einfach der Metawert, bei den Link-Metas ist es Link-Markup. Neben der Registrierung des REST-Endpunkts ist noch eine Anpassung einer Funktion aus der *inc/cpt.php* des *wh-talks*-Plugins notwendig, die direkt im Anschluss folgt.

In Listing 24.50 sehen Sie den Code zum Hinzufügen eines eigenen Endpunkts.

```
function register_rest_endpoint() {
 register_rest_route(
  'wh-talks/v1',
  '/talks/(?P<id>[\d]+)/meta-string/(?P<meta_key>[\w]+)/',
  [
   'methods'  => \WP_REST_Server::READABLE,
   'permission_callback' => '__return_true',
   'callback' => function( $request ) {
    $talk_id = $request['id'] ?? 0;
    $meta_key = $request['meta_key'] ?? '';
    if ( '' === $meta_key || empty( $talk_id ) ) {
     return new \WP_REST_Response(
      [
       'message' => 'Required parameter missing',
      ],
      400
     );
    }
    $allowed_meta_keys = [
     'wh_talks_event_name',
     'wh_talks_duration',
     'wh_talks_language',
     'wh_talks_video_link',
     'wh_talks_slides_link',
    ];

    if (
```

```
      ! in_array(
       $meta_key,
       $allowed_meta_keys,
       true
      )
     ) {
      return new \WP_REST_Response(
       [
        'message' => 'Invalid param',
       ],
       403
      );
     }

     if ( 'talk' !== get_post_type( $talk_id ) ) {
      return new \WP_REST_Response(
       [
        'message' => 'Invalid param',
       ],
       403
      );
     }

     if ( 'publish' !== get_post_status( $talk_id ) ) {
      return new \WP_REST_Response(
       [
        'message' => 'Invalid param',
       ],
       403
      );
     }

     return new \WP_REST_Response(
      [
       'body' => get_meta_value_markup(
        $meta_key,
        $talk_id
       )
      ]
     );
    },
   ]
  );
}
add_action( 'rest_api_init', __NAMESPACE__ . '\\register_rest_endpoint' );
```

Listing 24.50 Der Code fügt einen neuen Endpunkt zur REST-API hinzu.

1. Innerhalb der `register_rest_endpoint()`-Funktion rufen Sie `register_rest_route()` auf, um den neuen Endpunkt einzufügen. Der erste Parameter für die Funktion ist der

Namespace für die Route. Der Core hat den Namespace `wp/v2`, und es ist eine gute Idee, ebenfalls eine Versionierung mit einzubauen, damit Sie für Änderungen nicht direkt den bestehenden Endpunkt anpassen müssen, sondern einen neuen erstellen, der dann eine höhere Versionsnummer hat. So kann Code, der von der Struktur des alten Endpunkts ausgeht, einfach weiter funktionieren. Deshalb legen Sie den Namespace `wh-talks/v1` fest.
Der zweite Parameter ist die Route, also der Pfad, der nach dem Namespace eingefügt wird. Der kann entweder statisch sein oder dynamische Teile beinhalten, wie in diesem Beispiel. `/talks/(?P<id>[\d]+)/meta-string/(?P<meta_key>[\w]+)/` beinhaltet zwei Teile mit einem regulären Ausdruck. `(?P<id>[\d]+)` bedeutet, dass an dieser Stelle eine beliebige Zahl erwartet wird, `(?P<meta_key>[\w]+)` steht für eine beliebige Kette von Wortzeichen, das sind Zahlen und Buchstaben sowie der Unterstrich. Durch die Angabe der Bezeichner in den spitzen Klammern legen Sie fest, wie auf die Werte zugegriffen werden kann, die von den regulären Ausdrücken gefunden wurden.
Als dritten Parameter erwartet `register_rest_route()` ein Array mit zusätzlichen Argumenten. Angegeben werden die von der Route unterstützten HTTP-Methoden (in diesem Fall nur `GET`), der Callback, der prüft, ob die Anfrage an die Route die notwendigen Berechtigungen hat (mit `__return_true` wird eine Funktion von WordPress angegeben, die immer `true` zurückgibt, da keine besonderen Berechtigungen nötig sind), und die Callback-Funktion, die beim Aufrufen der Route ausgeführt wird.

2. An die Callback-Funktion wird mit `$request` das Request-Objekt übergeben, das in der Funktion genutzt wird, um die Daten für

`id` und `meta_key` zu ermitteln, die das Ergebnis der regulären Ausdrücke sind.

3. Es folgt ein Test, der ermittelt, ob beide Parameter zumindest auf den ersten Blick sinnvolle Werte enthalten, andernfalls wird eine Fehlermeldung ausgegeben.
4. In dem `$allowed_meta_keys`-Array sind die Meta-Keys gespeichert, die ausgegeben werden dürfen. Bei der nachfolgenden Prüfung wird getestet, ob der Wert von `$meta_key` in diesem Array enthalten ist, andernfalls wird erneut eine Fehlermeldung zurückgegeben. Durch diese Prüfung verhindern Sie, dass andere Metawerte des Posts ausgelesen werden können.
5. Als Nächstes wird überprüft, ob der Post, dessen ID übergeben wurde, nicht den richtigen Inhaltstyp hat `( 'talk' !== get_post_type( $talk_id ))`, um in dem Fall erneut einen Fehler zu werfen.
6. Wenn wir jetzt immer noch dabei sind, liegt schon mal eine ID eines Vortrags-Posts vor, und es wurde ein erlaubter Meta-Key übergeben. Jetzt muss noch geprüft werden, ob der Post öffentlich ist, sonst könnten die Metawerte auch zu Entwürfen oder geschützten Posts abgefragt werden. Dafür testen Sie, ob der Post-Status nicht öffentlich ist `( 'publish' !== get_post_status( $talk_id ) )`, um für den Fall wieder einen Fehler auszugeben.
7. Wenn keine der Prüfungen zutreffend war, wird am Ende eine REST-Antwort mit dem Metawert-String zurückgegeben. Dafür wird die Funktion `get_meta_value_markup()` aufgerufen, die in Abschnitt 24.5.2, »Block mit Liste der Metawerte erstellen«, in der *inc/cpt.php* erstellt wurde. Diese Funktion wird gleich im nächsten Schritt ein bisschen angepasst.

8. Die `register_rest_endpoint()`-Funktion wird an den `rest_api_init`-Hook angehängt.

Wie beschrieben, muss jetzt noch die Funktion `get_meta_value_markup()` angepasst werden. Das Problem: Bisher wurde die Funktion immer im Kontext der Post-Loop aufgerufen, weshalb die ID des Inhalts nicht übergeben werden musste, für den ein Meta-Key zurückgegeben werden sollte: `get_the_ID()` hat die ID innerhalb der Funktion bereitgestellt.

In der REST-API sind wir in diesem Kontext nicht mehr und müssen daher die ID als Parameter übergeben. In Listing 24.51 ist der Code der angepassten Funktion abgedruckt, die geänderten Stellen sind hervorgehoben.

```
function get_meta_value_markup( $meta_key, $post_id = null ) {
 $allowed_meta = false;

 foreach ( get_string_metas() as $meta ) {
  if ( $meta_key !== $meta['key'] ) {
   continue;
  }

  $allowed_meta = true;
  break;
 }

 if ( ! $allowed_meta ) {
  return '';
 }

 $post_id = $post_id ?? get_the_ID();

 $meta_value = get_post_meta( $post_id, $meta_key, true ) ?? null;
 if ( empty( $meta_value ) ) {
  return '';
 }

 if ( strrpos( $meta_key, '_link' ) !== strlen( $meta_key ) - strlen(
'_link' ) ) {
  return esc_html( $meta_value );
 }

 if ( 1 !== preg_match( '/^(http:|https:)?\/\//', $meta_value ) ) {
  return '';
 }
```

```
 $url  = esc_url( $meta_value );
 $host = parse_url( $url, PHP_URL_HOST );
 return sprintf(
  '<a href="%s">%s</a>',
  $url,
  $host
 );
}
```

Listing 24.51 Die angepasste »get_meta_value_markup()«-Funktion

Als zweiten Funktionsparameter fügen Sie `$post_id` hinzu und geben als Standardwert `null` an, sodass nur optional ein Wert für diesen Parameter übergeben werden muss. In der Funktion wird, bevor der Metawert ermittelt wird, in `$post_id` entweder der eigene Wert gespeichert oder, wenn `$post_id` den Wert `null` hat, das Ergebnis von `get_the_ID()`, das bisher in der Funktion genutzt wurde. Dem `get_post_meta()`-Aufruf wird dann für das erste Attribut der Wert von `$post_id` übergeben.

Durch diese Anpassung kann die Funktion im neuen Kontext genutzt werden, funktioniert aber weiterhin auch für die bereits vorhandenen Funktionsaufrufe, ohne dass diese angepasst werden müssen. In Abbildung 24.33 sehen Sie die Ausgabe des REST-Endpunkts für die URL */wp-json/wh-talks/v1/talks/183/meta-string/wh_talks_video_link* (die ID müssen Sie entsprechend anpassen, sodass sie der ID eines veröffentlichten Vortrags entspricht).



Abbildung 24.33 Die Ausgabe eines gültigen Aufrufs des neuen REST-Endpunkts für die Ausgabe des Videolinks zu einem Vortrag

A Glossar

Accessibility Accessibility oder Barrierefreiheit bezeichnet das Handwerk, das sich – im Fall der Web-Accessibility – damit beschäftigt, Websites möglichst gut für alle Menschen zugänglich zu machen. Dabei geht es sowohl um Designaspekte wie ausreichenden Farbkontrast oder Schriften, die nicht zu klein sind, als auch um die technische Umsetzung, etwa dass Inhalte mit den dafür vorgesehenen HTML-Elementen ausgezeichnet sind und dass eine Website mit unterschiedlichsten Bedienkonzepten (egal ob Maus, Tastatur, Screenreader, …) genutzt werden kann.

Adminbar Bereich sam oberen Rand des Browserfensters, in dem Sie nützliche Links finden, beispielsweise zum Erstellen neuer Inhalte oder um zur Seite der verfügbaren Aktualisierungen zu springen. Wird je nach Einstellung in Ihrem Profil auch im Frontend angezeigt.

Admin-Menü Das Menü auf der linken Seite im Backend, das bis auf Vollbild-Ansichten des Block-Editors überall im Backend angezeigt wird.

Affiliate Ein System, bei dem Sie bei sich Werbung veröffentlichen (im einfachsten Fall einen Link, sonst Bilder oder animierte Werbung) und Provision erhalten, wenn jemand beispielsweise auf den Link geklickt und einen Kauf bei Ihrem Werbepartner getätigt hat.

Backend Der für die Öffentlichkeit nicht sichtbare Bereich einer WordPress-Installation, in dem Sie die Inhalte verwalten, Einstellungen anpassen, Design-Änderungen vornehmen et cetera.

Block Die Bestandteile, aus denen Sie im Block-Editor Ihre Inhalte zusammensetzen. So gibt es beispielsweise einen Absatz-Block, einen Bild-Block und einen Listen-Block, aber auch einen Spalten-Block, sodass komplexere Inhaltslayouts möglich sind.

Block-Editor Ein komplett neues Editor-Konzept, das in WordPress 5 eingefügt wurde und den bis dahin eingesetzten Editor für Inhalte von Beiträgen und Seiten ersetzt hat. Inzwischen ist der Block-Editor auch in weiteren Bereichen des Backends aktiv, wie bei der Bearbeitung von Widget-Bereichen bei Classic Themes und als Website-Editor zur Bearbeitung der gesamten Website, wenn ein Block-Theme genutzt wird.

Block-Inserter Das Interface, in dem Sie einen Block zum Einfügen in den Editor auswählen können.

Block-Seitenleiste, Dokumenten-Seitenleiste Die rechte Seitenleiste im Block-Editor ist in zwei Bereiche aufgeteilt. Der eine, die Dokumenten-Seitenleiste, beherbergt Einstellungen, die die Seite oder den Beitrag als Ganzes betreffen. Der andere Bereich ist die Block-Seitenleiste mit Einstellungen des aktuell ausgewählten Blocks.

Block Bindings API Diese API ermöglicht es, Blöcke mit Daten unterschiedlicher Quellen zu »füttern«. So ist es standardmäßig möglich, Post-Meta-Werte als Quelle anzugeben und beispielsweise in einem Absatz-Block den Wert eines Post-Meta-Eintrags anzuzeigen. Daneben lassen sich auch weitere Quellen registrieren.

Block-Theme Eine neue Art Themes, die in WordPress 5.9 eingeführt wurden. Die Templates bestehen hier nur noch aus dem Markup von Blöcken, was die Bearbeitung der gesamten Website in dem Website-Editor ermöglicht.

Blog Eine meist chronologische Liste von Beiträgen. Ein Blog-System, als das WordPress gestartet ist, ist damit ein System zur Erstellung und Verwaltung eines Blogs.

Blogroll Eine Blogroll ist eine Liste von Blogs, die ein*e Autor*in empfiehlt. Während der Hochzeit des Bloggens waren Blogrolls oft in der Seitenleiste eines Blogs anzutreffen. Eine Blogroll dient dazu, Leser*innen thematisch ähnliche oder interessante Webseiten und Blogs vorzustellen. Mit der Zeit hat die Nutzung von Blogrolls abgenommen.

Brute-Force-Attacke Bei einer Brute-Force-Attacke versuchen Angreifer*innen, durch gezieltes oder ungezieltes Raten von Zugangsdaten Zugriff auf einen Account oder ein Gerät zu erlangen. Durch den Einsatz leistungsstarker Computer oder vieler Geräte können diese Angriffe recht effektiv durchgeführt werden.

Buchhaltungssoftware Anwendung, die Unternehmen und Einzelpersonen bei der Verwaltung ihrer Transaktionen und Buchhaltungsprozesse unterstützt. Sie unterstützt das Erfassen von Einnahmen und Ausgaben und das Erstellen von Rechnungen.

Canonical URL Die URL, unter der ein Inhalt bei Suchmaschinen indexiert werden soll. Durch die Angabe einer Canonical URL können Sie verhindern, dass Inhalte doppelt indexiert werden, wenn sie unter mehreren URLs erreichbar sind.

Child-Theme Ein Theme, das die Eigenschaften eines anderen Themes (das Parent-Theme) erbt und anpassen kann. Wenn Sie also beispielsweise ein Classic Theme einsetzen möchten, bei dem Sie das Template für Beiträge verändern möchten, können Sie ein Child-Theme für das Theme anlegen und darin nur dieses Template überschreiben, alles andere wird vom Parent-Theme übernommen.

Classic Theme Die klassische Art Themes, mit PHP-Dateien als Templates. Diese Themes funktionieren weiterhin, mit einem Classic Theme haben Sie allerdings keinen Zugriff auf den Website-Editor.

CLI Das Kommandozeilen-Interface (Command Line Interface) bezeichnet ein Interface, durch das Sie mit Textbefehlen Aktionen etwa lokal auf Ihrem System oder auf einem Webserver ausführen können. Unter macOS ist das Standardprogramm dafür das Terminal, unter Windows gibt es cmd. Zunächst ist die Interaktion mit einem System über diese Art Interface meist abschreckend, da es nicht so komfortabel wie ein schickes User-Interface ist, aber wenn Sie sich ein bisschen eingefunden haben, werden Sie feststellen, dass viele Aktionen schneller über das CLI ausgeführt werden können als über das »normale« User-Interface.

Content-Management-System (CMS) Ein System zur Verwaltung von Inhalten, die beispielsweise aus Seiten, Beiträgen und Medien (Bildern, Audiodateien et cetera) bestehen können. Viele Content-Management-Systeme, wie auch WordPress, bringen die notwendigen Funktionen mit, um neben der Inhaltsverwaltung auch die für alle sichtbare Website zu erstellen. Andere Content-Management-Systeme sind nur für die Verwaltung der Inhalte gedacht, und die Website ist von dem CMS entkoppelt (Headless CMS).

Core Siehe *WordPress-Core*.

Crawler Ein Bot (Programm) einer Suchmaschine, der Webseiten besucht und den Links folgt, um den Index für Suchmaschinen zu aktualisieren.

Cross- und Upsell Verkaufstechniken, die darauf abzielen, den Umsatz zu steigern, indem zusätzliche (Crosssell) oder teurere Produkte (Upsell) verkauft werden.

Custom Post-Type (CPT) WordPress bringt die Möglichkeit mit, eigene Inhaltstypen (Post-Types) zu erstellen, die wie Seiten und Beiträge funktionieren. Diese Inhaltstypen werden Custom Post-Type genannt.

Debuggen Die Suche nach der Ursache eines Problems in einer Software sowie dessen Behebung.

Dezentrales Netzwerk Dezentrale Netzwerke sind Plattformen für soziale Interaktionen, die auf einem verteilten Netzwerk von unabhängigen Servern basieren anstatt auf einer zentralen Instanz. Diese Struktur ermöglicht es verschiedenen Servern, miteinander zu kommunizieren und Daten auszutauschen, wodurch Benutzer*innen über verschiedene Instanzen hinweg interagieren können. Nutzer*innen haben mehr Kontrolle über ihre Daten und können auswählen, welchem Server sie beitreten.

Document Object Model (DOM) Das Document Object Model stellt die Struktur einer Webseite dar und ermöglicht es beispielsweise JavaScript, auf die Struktur zuzugreifen und sie zu verändern.

E-Commerce Siehe *Online-Shop*.

Erweiterungen Im Kontext komplexer WordPress-Plugins wie WooCommerce hat es sich etabliert, auf ein Kern-Plugin zu setzen, das durch zusätzliche Plugins erweitert werden kann. Diese zusätzlichen Plugins werden häufig Erweiterungen (engl. Extensions) genannt.

Footer Die Fußzeile einer Website wird auch oft als Footer bezeichnet. Er enthält oft allgemeine Informationen wie Copyright-Hinweise und Links zu Impressum oder Datenschutzerklärung.

Frontend Der für alle sichtbare Bereich der Website.

Full-Site-Editing Bezeichnet die Möglichkeit, beim Einsatz eines Block-Themes über den Website-Editor die komplette Website im WordPress-Backend zu bearbeiten, ohne Code schreiben zu müssen.

Gettext-Funktion Eine Funktion, um eine Zeichenkette übersetzbar zu machen. Gettext ist eine Programmbibliothek mit Werkzeugen zur Internationalisierung von Software.

Google Analytics Google Analytics ist ein kostenloses Tool von Google, das den Betreiber*innen von Websites hilft, detaillierte Daten über die Besucher*innen ihrer Webseite zu sammeln und zu analysieren. Die Verwendung von Google Analytics in der EU ist durch die DSGVO nur unter bestimmten Voraussetzungen erlaubt.

Gravatar Gravatar ist ein Kofferwort aus »Globally Recognized Avatar«, in etwa: global bekannter Avatar. Hierbei handelt es sich um einen Online-Dienst der Firma Automattic, der E-Mail-Adressen mit Avatar-Bildern verknüpft, die dann zum Beispiel in Kommentar-Bereichen in WordPress genutzt werden können.

Header Der Header ist der obere Bereich einer Webseite. Er erscheint typischerweise auf jeder Seite einer Website. Oft enthält er wichtige Elemente wie das Logo, die Hauptnavigation und manchmal Suchfunktionen oder Kontaktinformationen.

Hooks (Actions und Filter) Hooks sind ein wichtiges Konzept in der Entwicklung für WordPress. Über Action- und Filter-Hooks können Sie sich an bestimmten Stellen in den Code von WordPress einhängen und Aktionen ausführen oder Daten verändern.

Hoster Siehe *Webhosting*.

Hyperscaler Sammelbegriff für Cloud-Anbieter, die IT-Infrastruktur in großem Umfang bereitstellen und betreiben können. Bekannte Beispiele sind Amazon Web Services (AWS), Microsofts Azure und die Google Cloud Platform.

Hypertext Transfer Protocol Secure Hypertext Transfer Protocol Secure – kurz HTTPS – ist eine erweiterte Version des HTTP-Protokolls. Es dient zur sicheren Kommunikation über das Internet und verschlüsselt die Datenübertragung zwischen Client und Server mittels TLS (Transport Layer Security), um Abhör- und Manipulationsrisiken zu minimieren.

Inhaltstyp WordPress bringt unterschiedliche Inhaltstypen (Post-Types) mit, deren Daten in derselben Datenbanktabelle gespeichert werden. Neben Beiträgen und Seiten sind unter anderem auch Medien und Block-Templates ein Inhaltstyp. Sie können eigene Inhaltstypen registrieren, sogenannte Custom Post-Types.

Interactivity API Eine Frontend-API, die es ermöglicht, über sogenannte Direktiven (Attribute im HTML) Interaktivität zu Blöcken hinzuzufügen.

Internationalisierung Der Vorgang, um die Voraussetzungen zu schaffen, dass Software übersetzt werden kann.

JSX Eine Erweiterung für JavaScript, mit der Sie HTML-ähnliche Syntax schreiben können, um HTML-Markup zu generieren.

Kanonische URL Siehe *Canonical URL*.

Knowledge Graph Der Knowledge Graph von Google sorgt dafür, dass bei einigen Suchanfragen in einem separaten Bereich (in der Desktop-Ansicht rechts von der Suchergebnisliste) Informationen etwa zu einer Person angezeigt werden, nach der gesucht wurde.

Kompromittierung Unsicherer Status einer Website, auf der Schadcode eingeschleust wurde. Nach einer Kompromittierung muss analysiert werden, was genau wo verändert wurde, um die Website durch Beheben der Kompromittierung wieder in einen sicheren Zustand zu bringen.

Let’s Encrypt Let's Encrypt ist eine gemeinnützige Zertifizierungsstelle, die kostenlose SSL/TLS-Zertifikate bereitstellt, um HTTPS-Verschlüsselung für Webseiten zu ermöglichen.

Lightbox Ein gängiges Konzept, in dem Medien, meist Bilder, nach einem Klick vergrößert in einem Overlay dargestellt werden.

Load Balancing Load Balancing ist eine Technik zur Verteilung von eingehendem Datenverkehr oder von Rechenlast auf mehrere Server. Damit können Performance,

Verfügbarkeit und Skalierbarkeit von Systemen optimiert werden. Für kleine und mittelgroße Websites ist Load Balancing nicht unbedingt erforderlich.

Malware Von Angreifer*innen eingeschleuster Code, der entweder versteckt oder offensichtlich unerwünschte und eventuell für Besucher*innen gefährliche Aktionen ausführt.

Mastodon Mastodon ist ein dezentraler Mikroblogging-Dienst, der es Benutzer*innen ermöglicht, Nachrichten, Bilder und Videos in kurzen Beiträgen zu teilen. Im Gegensatz zu zentralisierten sozialen Netzwerken basiert Mastodon auf einem Netzwerk unabhängiger Server, die von Privatpersonen oder Organisationen betrieben werden. Diese Server, auch Instanzen genannt, können untereinander kommunizieren und Nachrichten austauschen.

Mediathek Eine Ansicht, die Sie über das Admin-Menü unter dem Punkt Medien aufrufen können. Hier sehen Sie alle Dateien, die zu Ihrer WordPress-Installation hochgeladen wurden, und Sie können in die Bearbeitungsansicht der einzelnen Medien wechseln, um beispielsweise für Bilder einen Alternativtext anzugeben.

Metadaten Siehe *Post-Meta*.

Namespace Namespaces in PHP helfen dabei, Probleme mit gleich benannten Klassen, Funktionen, Konstanten oder Ähnlichem zu verhindern, indem ein Namensraum beispielsweise für die Klasse in einer Datei geschaffen wird, sodass sie nicht mehr im globalen Namensraum liegt.

npm Der Package-Manager von Node.js.

Node.js Eine Laufzeitumgebung für JavaScript, die die Ausführung von JavaScript außerhalb des Browsers ermöglicht, beispielsweise auf einem Server, wie PHP.

Nonce Eine zufällige Zeichenkette, die als verstecktes Feld in Formulare eingefügt wird, sodass WordPress vor der Verarbeitung des abgeschickten Formulars sicherstellen kann, dass die Formulardaten von diesem Formular abgeschickt wurden und nicht woanders herkommen.

Online-Shop Website mit allen funktionellen und verwaltungstechnischen Eigenschaften eines Ladengeschäfts, aber einer virtuellen Verkaufsfläche und halb automatischen Online-Bezahl- und Abrechnungssystemen. Die Herausforderungen an solch eine Website sind die übersichtliche Produktpräsentation, die Vermittlung von Vertrauen in den Shop, die unkomplizierte Abrechnung und die Anbindungen an die Warenwirtschaft und Online-Bezahlsysteme.

Password Policies Password Policies legen bestimmte Regeln fest, die auf einer Website oder bei einem Service für die Zusammensetzung von Passwörtern gelten. Sie können zum Beispiel eine bestimmte Mindestlänge, eine Anzahl von Ziffern und Sonderzeichen oder andere Aspekte umfassen. Richtig eingesetzt sind Password Policies effektiv darin, die Sicherheit einer Website oder eines Dienstes zu erhöhen.

Payment-Gateways Schnittstelle zu einem Zahlungsdienstleister, über den einzelne Buchungen abgerechnet werden können.

Redaktionsplan Ein Redaktionsplan ist ein Dokument, das das Planen, Erstellen und Veröffentlichen von Inhalten für verschiedene Medien koordiniert. Er hilft Redaktionen und Einzelpersonen, Veröffentlichungen zu koordinieren, Deadlines einzuhalten und konsistente Inhalte zu veröffentlichen.

Permalink Der Link zu einem Inhalt, beispielsweise einem Beitrag oder einer Seite.

PHP Die Programmiersprache, in der große Teile von WordPress geschrieben sind.

Plugin Plugins ermöglichen es, den Funktionsumfang von WordPress zu erweitern. Es gibt beispielsweise Plugins zur Erstellung von Online-Shops oder zur Erkennung von Spam-Kommentaren (zu Online-Shops gibt es in diesem Buch ein eigenes Kapitel).

Post-Type Siehe *Inhaltstyp*.

Post-Meta WordPress-Inhalte wie Beiträge und Seiten können mit Metadaten angereichert werden, um neben dem Inhalt zusätzliche Informationen zu speichern. Diese Informationen können von Themes oder Plugins genutzt werden.

Prüfsumme Eine Zeichenkette (ein Hash), die aus den Dateien und deren Inhalt eines Verzeichnisses erstellt wird. Anhand der Prüfsumme kann getestet werden, ob beispielsweise die lokal vorliegenden Dateien einer WordPress-Installation identisch mit denen sind, die auf WordPress.org heruntergeladen werden können.

QuickEdit Möglichkeit zur Bearbeitung von Informationen eines Beitrags oder einer Seite direkt in der Übersichtsliste, ohne die Bearbeitungsansicht öffnen zu müssen. Die Bearbeitungsmöglichkeiten beschränken sich auf einige grundlegende Dinge wie den Titel, den Slug oder Kategorien/Tags.

React JavaScript-Framework, das die Grundlage des Block-Editors bildet.

REST-API Sammlung von Endpunkten, die eine Software (zum Beispiel WordPress) oder ein Dienst anbieten, um das Auslesen, Hinzufügen, Aktualisieren und Verändern von Daten zu ermöglichen, aber auch um andere Aktionen auszuführen.

Ein Formular-Plugin könnte beispielsweise einen REST-Endpunkt anbieten, an den die Daten des im Frontend abgeschickten Formulars gesendet werden. Der Endpunkt sorgt dann dafür, dass eine Mail mit den Daten verschickt wird und/oder die Formulardaten gespeichert werden (nachdem die Daten validiert und von möglichem Schadcode bereinigt wurden).

Revision Ein gespeicherter Zwischenstand des Inhalts einer Seite oder eines Beitrags (oder auch von Einstellungen im Website-Editor). Sie können den aktuellen Stand mit früheren Revisionen vergleichen und bei Bedarf eine Revision wiederherstellen.

Rewrite-Rules In der Datenbank gespeicherte Regeln, anhand derer WordPress entscheidet, welcher Inhalt zu einer aufgerufenen URL ausgegeben wird.

robots-Meta-Tag Mit dem `robots`-Meta-Tag können Sie angeben, wie eine Suchmaschine mit einer Webseite umgehen soll. Sie können unter anderem festlegen, ob eine Seite indexiert und ob den Links auf der Website gefolgt werden soll.

robots.txt Eine Datei im obersten Verzeichnis Ihrer Website, mit der Sie beispielsweise Suchmaschinen komplett verbieten können, eine Website zu indexieren, oder bestimmte Crawler ausschließen, wie den Google-Bilder-Bot.

SaaS (Software-as-a-Service) SaaS ist ein auf Abonnements basierendes Softwarebereitstellungsmodell. Anwendungen werden dabei über das Internet bereitgestellt. Nutzer*innen greifen auf die Anwendungen dabei in der Regel zu, ohne sie lokal installieren zu müssen.

Schadcode Siehe *Malware*.

Schema.org Ein Standard zur Anreicherung einer Webseite mit strukturierten Daten, um beispielsweise anzugeben, um was für eine Art Seite es sich handelt. So können etwa eine Produktseite in einem Online-Shop gekennzeichnet und der Preis sowie die durchschnittliche Sternebewertung des Produkts direkt in den Suchergebnissen angezeigt werden.

Security by Obscurity Als Security by Obscurity bezeichnet man ein Sicherheitskonzept, bei dem Systeme dadurch geschützt werden, dass ihre Funktionsweise oder Details ihrer Funktion geheim gehalten werden. In der Praxis bietet diese Strategie keinen ausreichenden Schutz vor Angriffen und sollte daher nicht eingesetzt werden.

Semantic Versioning Beim Semantic Versioning spiegelt die Änderung der Versionsnummer die Art der Änderungen des Software-Updates wider. Bei einem 2.0.0-Release gibt es Änderungen an der öffentlichen API der Software, die nicht

rückwärtskompatibel sind, eine 2.1.0-Version bringt neue Funktionen mit, ist mit der Vorversion aber kompatibel, und 2.1.1 beinhaltet lediglich Bugfixes.

Settings API Funktionen von WordPress, die beim Erstellen von Einstellungen und Einstellungsseiten im WordPress-Backend helfen.

Shortcode Vor der Einführung des Block-Editors waren Shortcodes die einzige Möglichkeit, komplexere Inhaltselemente im Editor einzufügen, ohne HTML schreiben zu müssen. Shortcodes sind durch die eckigen Klammern gekennzeichnet, ein Beispiel ist `[gallery]`.

SQLite Ein dateibasiertes Datenbanksystem, das über ein Plugin auch mit WordPress genutzt werden kann (standardmäßig arbeitet WordPress mit den Datenbanksystemen MySQL oder MariaDB).

Staging Zusatzfeature einiger Hosting-Angebote. Testumgebung, die eine Kopie einer Website darstellen kann, an der Aktualisierungen oder Änderungen getestet werden können, ohne die eigentliche produktive Website zu beeinflussen.

Suchmaschinenoptimierung Bezeichnet verschiedenste Vorgehensweisen, um einen Inhalt für einen oder mehrere Suchbegriffe möglichst weit oben in den Ergebnissen der Suchmaschinen zu platzieren. Teilt sich auf in Onpage-Optimierung (die Optimierung der Seite selbst) und Offpage-Optimierung (alle Maßnahmen abseits der Seite, wie Social-Media-Aktivitäten, Verlinkungen von anderen Websites et cetera).

Taxonomie Mit einer Taxonomie können Inhalte gruppiert werden. Kategorien und Schlagwörter sind die Standard-Taxonomien in WordPress.

Template Themes, die für die Darstellung einer WordPress-Website verantwortlich sind, bestehen aus Templates. Dabei gibt es unterschiedliche Arten von Templates, zum Beispiel eins zum Anzeigen der Einzelansicht eines Beitrags und eins für die Startseite. WordPress sucht anhand der sogenannten Template-Hierarchie das passendste Template für den gerade aufgerufenen Inhalt.

Template-Hierarchie Das System, nach dem WordPress entscheidet, welches Theme-Template für eine bestimmte Art aufgerufene Seite geladen werden soll.

Term Begriffe einer Taxonomie, zum Beispiel die Kategorie »Allgemein«.

Theme Themes heißen in WordPress die Designs, die Sie installieren können, um das Aussehen Ihrer Website zu verändern.

VG Wort Die VG Wort (Verwertungsgesellschaft Wort) ist eine deutsche Verwertungsgesellschaft, die Urheberrechte für Autor*innen und Verlage wahrnimmt und Tantiemen für die Nutzung von Textwerken ausschüttet. Sie sorgt dafür, dass Autor*innen für Kopien und Nutzungen ihrer Texte, beispielsweise in Medien oder durch Vervielfältigung, vergütet werden.

Warenwirtschaftssystem Software zur Verwaltung von Warenbestellungen, Lieferungen und Verkäufen. Ermöglicht (teil)automatisierte Steuerung und Überblick über die Warenlogistik.

Webhoster Ein Anbieter, der Ihnen Dinge wie einen Webserver und Datenbanken zur Verfügung stellt, sodass Sie relativ niedrigschwellig Ihre Website online stellen können.

Webserver Software, die sich um die Auslieferung Ihrer Website kümmert. Die bekanntesten sind Apache und nginx.

Web Accessibility Initiative (WAI) Initiative des W3C (World Wide Web Consortium), das Standards im Bereich der Barrierefreiheit entwickelt und Materialien bereitstellt, um bei der Umsetzung zu unterstützen.

Web Content Accessibility Guidelines (WCAG) Von der WAI entwickelte Standards für Barrierefreiheit im Web.

Wildcard (Domain) Platzhalter, der genutzt werden kann, um alle Subdomains einer Domain auf eine bestimmte IP-Adresse oder einen bestimmten Server zu verweisen, ohne die einzelnen Subdomains konfigurieren zu müssen.

WordPress-Core Bezeichnung für die WordPress-Software. Wenn Sie beispielsweise irgendwo davon lesen, dass Sie die Dateien des Core nicht anpassen sollten, weil Änderungen bei Updates verloren gehen, sind damit die WordPress-Dateien in *wp-admin* und *wp-includes* gemeint.

WordPress.com Anders als das Open-Source-System WordPress, das Sie von WordPress.org herunterladen können, ist WordPress.com ein Dienst, der WordPress-Instanzen bereitstellt. Sie müssen sich also nicht selbst um das Hosting kümmern, sind aber auch eingeschränkt in den Funktionen.

WordPress-Meetups Lokale Treffen von WordPress-Interessierten. In vielen Städten treffen sich Meetup-Gruppen in einem monatlichen Rhythmus, um über aktuelle WordPress-Themen zu diskutieren, gemeinsam Probleme zu lösen oder Vorträge anzuhören. Alle deutschsprachigen Meetups finden sich auf *https://wpmeetups.de*.

WordPress.org Der Anlaufpunkt für das Open-Source-System WordPress. Hier finden Sie so ziemlich alles zum WordPress-Projekt, unter anderem Download-Möglichkeiten der Software, Support-Foren, das Plugin- und das Theme-Verzeichnis, Dokumentation und Möglichkeiten, sich beim WordPress-Projekt einzubringen.

WP-Admin Siehe *Backend*.

XML-RPC-Schnittstelle Die XML-RPC-Schnittstelle von WordPress ermöglicht es Programmen, mit WordPress-Websites zu kommunizieren. Trotz der Einführung der neueren REST-API wird die alte XML-RPC-API weiterhin von manchen Diensten und Apps genutzt. Sie gilt jedoch als Schwachpunkt für Brute-Force- und DDoS-Attacken und sollte daher nach Möglichkeit deaktiviert werden.

Zwei-Faktor-Authentifizierung Die Zwei-Faktor-Authentifizierung, oft auch 2FA, ist eine Sicherheitsmethode, bei der Benutzer*innen zwei unterschiedliche Nachweise ihrer Identität erbringen müssen. Um Zugriff auf ein System oder eine Website zu erhalten, ist eine Kombination aus Wissen (z. B. Passwort) und Besitz (z. B. Smartphone für einen Code) notwendig. Zwei-Faktor-Authentifizierung erhöht die Sicherheit erheblich, da Angreifer*innen beide Faktoren überwinden müssten, um Zugriff zu erlangen.

Stichwortverzeichnis

↓A ↓B ↓C ↓D ↓E ↓F ↓G ↓H ↓I ↓J ↓K ↓L ↓M ↓N ↓O ↓P ↓Q
↓R ↓S ↓T ↓U ↓V ↓W ↓X ↓Y ↓Z

@wordpress/scripts [→ 24.4 Panel in der Editor-Seitenleiste zur Bearbeitung der Metadaten]

404 [→ 23.1 Block-Theme erstellen]

A ⇑

Abfrage-Loop-Block [→ 10.3 Block-Theme anpassen: den Website-Editor kennenlernen]

Abo [→ 12.2 WooCommerce installieren und einrichten]

Abos [→ 12.4 Zusätzliche Erweiterungen für Ihren Shop]

Accessibility [→ A Glossar]

Account [→ 7.1 Rollen verstehen]

Action-Hook → siehe [Hooks]

Activity Streams [→ 15.2 Das Fediverse]

ActivityPub [→ 15.2 Das Fediverse] [→ 15.2 Das Fediverse] [→ 15.2 Das Fediverse] [→ 15.2 Das Fediverse] [→ 15.3 IndieWeb oder Fediverse?]

Adminbar [→ 5.1 Das Backend kennenlernen] [→ 7.4 Benutzerprofil bearbeiten] [→ A Glossar]

Admin-Bereich → siehe [Backend]

Admin-E-Mail-Adresse [→ 5.2 Einstellungen]

Administrationsbereich → siehe [Backend]

Admin-Menü [→ 5.1 Das Backend kennenlernen] [→ A Glossar]

- *Beiträge* [→ 5.1 Das Backend kennenlernen]
- *Benutzer* [→ 5.1 Das Backend kennenlernen]
- *Dashboard* [→ 5.1 Das Backend kennenlernen]
- *Design* [→ 5.1 Das Backend kennenlernen]
- *Kommentare* [→ 5.1 Das Backend kennenlernen]
- *Medien* [→ 5.1 Das Backend kennenlernen]
- *Plugins* [→ 5.1 Das Backend kennenlernen]
- *Seiten* [→ 5.1 Das Backend kennenlernen]
- *Werkzeuge* [→ 5.1 Das Backend kennenlernen]

Affiliate [→ 13.1 Zielgruppe und Plan] [→ 13.1 Zielgruppe und Plan]

Agentur [→ 9.1 Was ist WordPress Multisite?]

Akismet [→ 8.4 Spamschutz mit Antispam Bee] [→ 11.4 Plugins installieren]

Aktive Installationen [→ 11.3 Gute Plugins erkennen]

Aktualisierung [→ 11.5 Plugin aktualisieren]

Aktualisierungen [→ 5.1 Das Backend kennenlernen] [→ 18.1 Login-Sicherheit]

Algorithmen [→ 13.1 Zielgruppe und Plan] [→ 15.1 Das IndieWeb]

Alle Blöcke kopieren [→ 10.3 Block-Theme anpassen: den Website-Editor kennenlernen]

Allgemeine Einstellungen [→ 5.2 Einstellungen]

All-in-One-Plugins [→ 11.3 Gute Plugins erkennen]

Alternativtext [→ 19.2 PageSpeed Insights – Missstände verstehen und bei Bedarf abarbeiten]

für Bilder [→ 6.7 Inhalte planen]

schreiben [→ 6.7 Inhalte planen]

Angebotspreis [→ 12.2 WooCommerce installieren und einrichten]

Antispam Bee [→ 2.1 WordPress Playground – eine ganze WordPress-Installation lokal im Browser] [→ 19.5 Profi-Cache mit WP Rocket]

Anwendungspasswörter [→ 7.4 Benutzerprofil bearbeiten]

Apple Pay [→ 12.2 WooCommerce installieren und einrichten]

Apps [→ 1.2 Eine kurze Geschichte von WordPress]

Arbeitsspeicher [→ 4.1 Welches Hosting darf's denn sein?]

Archiv-Seiten [→ 13.3 Struktur und Navigation]

Artikelnummer [→ 12.2 WooCommerce installieren und einrichten]

Auth-Code [→ 17.2 Eine Website migrieren]

Authenticator-App [→ 18.1 Login-Sicherheit]

Automatische Updates [→ 11.5 Plugin aktualisieren]

deaktivieren [→ 16.1 WordPress, Plugins und Themes aktualisieren]

Automattic [→ 12.2 WooCommerce installieren und einrichten]

Auto-Updates [→ 1.2 Eine kurze Geschichte von WordPress] [→ 9.3 Bedienung des Netzwerks und Unterschiede zum Standard]

Avada [→ 10.5 Neues Theme suchen und installieren]

Avatar [→ 8.3 Datenschutz-Bedenken und -Lösungen] [→ 10.3 Block-Theme anpassen: den Website-Editor kennenlernen]

Avatare [→ 5.2 Einstellungen]

AVIF [→ 6.5 Medien organisieren] [→ 19.6 Bilder optimieren] [→ 19.6 Bilder optimieren]

B ⇑

b2/cafelog [→ 1.2 Eine kurze Geschichte von WordPress]

Backend [→ 1.3 Warum WordPress?] [→ 1.4 Anleitung zum Handbuch] [→ 2.1 WordPress Playground – eine ganze WordPress-Installation lokal im Browser] [→ 5.1 Das Backend kennenlernen] [→ 10.3 Block-Theme anpassen: den Website-Editor kennenlernen] [→ A Glossar]

URL [→ 5.1 Das Backend kennenlernen]

Backup [→ 1.4 Anleitung zum Handbuch] [→ 4.1 Welches Hosting darf's denn sein?] [→ 9.2 Die Multisite-Einrichtung]

[→ 16.1 WordPress, Plugins und Themes aktualisieren] [→ 18.1 Login-Sicherheit] [→ 18.1 Login-Sicherheit]

Aufbewahrung [→ 16.3 Backups planen und durchführen]

in Dropbox [→ 16.3 Backups planen und durchführen]

planen [→ 16.3 Backups planen und durchführen]

wiederherstellen [→ 16.3 Backups planen und durchführen]

Banküberweisung [→ 12.2 WooCommerce installieren und einrichten]

Barrierefreiheit [→ 6.7 Inhalte planen] [→ 10.6 Barrierefreiheit von Themes testen] [→ 19.2 PageSpeed Insights – Missstände verstehen und bei Bedarf abarbeiten] [→ 19.2 PageSpeed Insights – Missstände verstehen und bei Bedarf abarbeiten] [→ A Glossar]

Screen-Reader [→ 19.2 PageSpeed Insights – Missstände verstehen und bei Bedarf abarbeiten]

Tastaturbedienbarkeit [→ 10.6 Barrierefreiheit von Themes testen]

Tastaturnavigation [→ 19.2 PageSpeed Insights – Missstände verstehen und bei Bedarf abarbeiten]

Barrierefreiheitsrichtlinien [→ 6.7 Inhalte planen]

Basic Authentication [→ 18.3 WordPress abhärten]

BBCode [→ 8.4 Spamschutz mit Antispam Bee]

Befehlspalette [→ 10.3 Block-Theme anpassen: den Website-Editor kennenlernen]

Beispiel-Websites [→ 1.4 Anleitung zum Handbuch]

Beitrag erstellen → siehe [Inhalte erstellen]

Beitrag oben halten [→ 6.1 Beiträge schreiben/Seiten anlegen]

Beitrag speichern [→ 6.1 Beiträge schreiben/Seiten anlegen]

Beiträge [→ 5.1 Das Backend kennenlernen]

Beiträge pro Seite [→ 5.2 Einstellungen]

Beiträge verwalten [→ 6.2 Beiträge/Seiten verwalten]

Beitragsbild [→ 6.1 Beiträge schreiben/Seiten anlegen]

Beitragsformat [→ 5.2 Einstellungen] [→ 15.2 Das Fediverse]

Beitragsseite [→ 10.4 Classic Theme anpassen]

Beitrags-Template ändern [→ 6.1 Beiträge schreiben/Seiten anlegen]

Beitragsübersicht [→ 13.2 Design und Layout]

Benachrichtigung [→ 9.3 Bedienung des Netzwerks und Unterschiede zum Standard]

Benutzer [→ 5.1 Das Backend kennenlernen] [→ 9.3 Bedienung des Netzwerks und Unterschiede zum Standard] [→ 9.3 Bedienung des Netzwerks und Unterschiede zum Standard]

- *bearbeiten* [→ 7.4 Benutzerprofil bearbeiten]
- *deaktivieren* [→ 7.5 Benutzer*in deaktivieren oder löschen]
- *Filterung nach Rolle* [→ 7.3 Benutzer*innen verwalten: sortieren, filtern, suchen]
- *löschen* [→ 7.5 Benutzer*in deaktivieren oder löschen]
- *neues Passwort* [→ 7.4 Benutzerprofil bearbeiten]

Selbstregistrierung [→ 7.2 Benutzer*innen hinzufügen]

Benutzername [→ 18.1 Login-Sicherheit]

Benutzerrolle [→ 5.2 Einstellungen] [→ 9.3 Bedienung des Netzwerks und Unterschiede zum Standard] [→ 14.3 Funktionalitäten und Plugins]

Benutzerverwaltung [→ 9.3 Bedienung des Netzwerks und Unterschiede zum Standard]

Berechtigung [→ 7.1 Rollen verstehen] [→ 9.3 Bedienung des Netzwerks und Unterschiede zum Standard]

Bestenliste [→ 13.4 Funktionen und Plugins]

Best-Practices beim Einsatz von Übersetzungsfunktionen [→ 22.2 WordPress-Konzepte]

Beta-Versionen [→ 16.1 WordPress, Plugins und Themes aktualisieren]

Bewertungen [→ 11.3 Gute Plugins erkennen]

Bild einfügen [→ 6.1 Beiträge schreiben/Seiten anlegen] [→ 6.1 Beiträge schreiben/Seiten anlegen]

Bilddateien [→ 19.7 Optimieren für den Klimaschutz]

Bilder optimieren [→ 6.5 Medien organisieren] [→ 19.2 PageSpeed Insights – Missstände verstehen und bei Bedarf abarbeiten] [→ 19.6 Bilder optimieren] [→ 19.6 Bilder optimieren]

JPEGmini [→ 6.5 Medien organisieren]

TinyPNG [→ 6.5 Medien organisieren]

Bilder verkleinern [→ 19.6 Bilder optimieren]

Bildformat [→ 6.5 Medien organisieren]

Bildgröße beschränkt [→ 6.5 Medien organisieren]

Bildgrößen [→ 5.2 Einstellungen]

Bitnami [→ 2.1 WordPress Playground – eine ganze WordPress-Installation lokal im Browser]

Block

- *einfügen* [→ 6.1 Beiträge schreiben/Seiten anlegen]
- *gruppieren* [→ 6.1 Beiträge schreiben/Seiten anlegen]
- *kopieren* [→ 6.1 Beiträge schreiben/Seiten anlegen]
- *sperren* [→ 6.1 Beiträge schreiben/Seiten anlegen]
- *umwandeln* [→ 6.1 Beiträge schreiben/Seiten anlegen]
- *verschieben* [→ 6.1 Beiträge schreiben/Seiten anlegen] [→ 6.1 Beiträge schreiben/Seiten anlegen] [→ 6.1 Beiträge schreiben/Seiten anlegen]

Block Bindings API [→ 6.4 Individuelle Felder nutzen] [→ 24.11 Weitere nützliche Funktionen für Plugins] [→ A Glossar]

block.json [→ 24.4 Panel in der Editor-Seitenleiste zur Bearbeitung der Metadaten] [→ 24.5 Editor-Blöcke erstellen]

Block-Breadcrumb [→ 6.1 Beiträge schreiben/Seiten anlegen]

Blöcke [→ 1.2 Eine kurze Geschichte von WordPress]

- *Abfrage-Loop* [→ 6.1 Beiträge schreiben/Seiten anlegen]
- *Absatz* [→ 6.1 Beiträge schreiben/Seiten anlegen]
- *Abstandshalter* [→ 6.1 Beiträge schreiben/Seiten anlegen]
- *An-/Abmelden* [→ 6.1 Beiträge schreiben/Seiten anlegen]

Änderungsdatum [→ 6.1 Beiträge schreiben/Seiten anlegen]

Archive [→ 6.1 Beiträge schreiben/Seiten anlegen]

Archiv-Titel [→ 6.1 Beiträge schreiben/Seiten anlegen]

Audio [→ 6.1 Beiträge schreiben/Seiten anlegen]

Autor [→ 6.1 Beiträge schreiben/Seiten anlegen]

Avatar [→ 6.1 Beiträge schreiben/Seiten anlegen]

Begriffsbeschreibung [→ 6.1 Beiträge schreiben/Seiten anlegen]

Beitragsbild [→ 6.1 Beiträge schreiben/Seiten anlegen]

Beitrags-Template [→ 6.1 Beiträge schreiben/Seiten anlegen]

Bild [→ 6.1 Beiträge schreiben/Seiten anlegen] [→ 6.1 Beiträge schreiben/Seiten anlegen]

Biografie des Autors [→ 6.1 Beiträge schreiben/Seiten anlegen]

Buttons [→ 6.1 Beiträge schreiben/Seiten anlegen]

Classic [→ 6.1 Beiträge schreiben/Seiten anlegen]

Cover [→ 6.1 Beiträge schreiben/Seiten anlegen]

Datei [→ 6.1 Beiträge schreiben/Seiten anlegen]

Datum [→ 6.1 Beiträge schreiben/Seiten anlegen]

Details [→ 6.1 Beiträge schreiben/Seiten anlegen]

Einbettungen [→ 6.1 Beiträge schreiben/Seiten anlegen]

Formular für Kommentare [→ 6.1 Beiträge schreiben/Seiten anlegen]

Galerie [→ 6.1 Beiträge schreiben/Seiten anlegen]

Gruppe [→ 6.1 Beiträge schreiben/Seiten anlegen]

Individuelles HTML [→ 6.1 Beiträge schreiben/Seiten anlegen]

Kalender [→ 6.1 Beiträge schreiben/Seiten anlegen]

Kategorien [→ 6.1 Beiträge schreiben/Seiten anlegen]

Kategorien-Liste [→ 6.1 Beiträge schreiben/Seiten anlegen]

Kommentare [→ 6.1 Beiträge schreiben/Seiten anlegen]

Liste [→ 6.1 Beiträge schreiben/Seiten anlegen]

Medien und Text [→ 6.1 Beiträge schreiben/Seiten anlegen]

Mehr [→ 6.1 Beiträge schreiben/Seiten anlegen]

Nächster Beitrag [→ 6.1 Beiträge schreiben/Seiten anlegen]

Name des Autors [→ 6.1 Beiträge schreiben/Seiten anlegen]

Navigation [→ 6.1 Beiträge schreiben/Seiten anlegen]

Neueste Beiträge [→ 6.1 Beiträge schreiben/Seiten anlegen]

Neueste Kommentare [→ 6.1 Beiträge schreiben/Seiten anlegen]

Pullquote [→ 6.1 Beiträge schreiben/Seiten anlegen]

Raster [→ 6.1 Beiträge schreiben/Seiten anlegen]

RSS [→ 6.1 Beiträge schreiben/Seiten anlegen]

Schlagwörter [→ 6.1 Beiträge schreiben/Seiten anlegen]

Schlagwörter-Wolke [→ 6.1 Beiträge schreiben/Seiten anlegen]

Seitenliste [→ 6.1 Beiträge schreiben/Seiten anlegen]

Seitenumbruch [→ 6.1 Beiträge schreiben/Seiten anlegen]

Shortcode [→ 6.1 Beiträge schreiben/Seiten anlegen]

Social Icons [→ 6.1 Beiträge schreiben/Seiten anlegen]

Spalten [→ 6.1 Beiträge schreiben/Seiten anlegen]

Stapel [→ 6.1 Beiträge schreiben/Seiten anlegen]

Suchen [→ 6.1 Beiträge schreiben/Seiten anlegen]

Tabelle [→ 6.1 Beiträge schreiben/Seiten anlegen]

Textauszug [→ 6.1 Beiträge schreiben/Seiten anlegen]

Titel [→ 6.1 Beiträge schreiben/Seiten anlegen]

Titel der Suchergebnisse [→ 6.1 Beiträge schreiben/Seiten anlegen]

Trenner [→ 6.1 Beiträge schreiben/Seiten anlegen]

Überschrift [→ 6.1 Beiträge schreiben/Seiten anlegen]

Vers [→ 6.1 Beiträge schreiben/Seiten anlegen]

Video [→ 6.1 Beiträge schreiben/Seiten anlegen]

Vorformatiert [→ 6.1 Beiträge schreiben/Seiten anlegen]

Vorheriger Beitrag [→ 6.1 Beiträge schreiben/Seiten anlegen]

Website-Logo [→ 6.1 Beiträge schreiben/Seiten anlegen]

Website-Titel [→ 6.1 Beiträge schreiben/Seiten anlegen]

Website-Untertitel [→ 6.1 Beiträge schreiben/Seiten anlegen]

Weiterlesen [→ 6.1 Beiträge schreiben/Seiten anlegen]

Zeile [→ 6.1 Beiträge schreiben/Seiten anlegen]

Zitat [→ 6.1 Beiträge schreiben/Seiten anlegen]

Blöcke in WordPress 6.7 [→ 6.1 Beiträge schreiben/Seiten anlegen]

Blöcke sperren [→ 23.4 Weitere nützliche Funktionen für Themes]

Block-Editor [→ 1.2 Eine kurze Geschichte von WordPress] [→ 6.1 Beiträge schreiben/Seiten anlegen] [→ 10.1 Classic Themes und Block-Themes: Was ist das?] [→ A Glossar]

- *Ansichtsmodus* [→ 6.1 Beiträge schreiben/Seiten anlegen]
- *Dokumentenansicht* [→ 6.1 Beiträge schreiben/Seiten anlegen] [→ 23.1 Block-Theme erstellen]
- *individuelle Felder* [→ 6.4 Individuelle Felder nutzen]
- *Seitenleiste* [→ 6.1 Beiträge schreiben/Seiten anlegen]
- *Tipps zur Bedienung* [→ 6.1 Beiträge schreiben/Seiten anlegen]
- *Toolbar* [→ 10.3 Block-Theme anpassen: den Website-Editor kennenlernen]
- *Werkzeugleiste* [→ 6.1 Beiträge schreiben/Seiten anlegen]

Block-Editor:Code-Ansicht [→ 6.4 Individuelle Felder nutzen]

Block-Entwicklung, Block-Funktionen aktivieren/deaktivieren [→ 24.11 Weitere nützliche Funktionen für Plugins]

Block-HTML bearbeiten [→ 6.1 Beiträge schreiben/Seiten anlegen]

Block-Inserter [→ 6.1 Beiträge schreiben/Seiten anlegen] [→ A Glossar]

Block-Seitenleiste [→ 6.1 Beiträge schreiben/Seiten anlegen] [→ 10.3 Block-Theme anpassen: den Website-Editor

kennenlernen]

Block-Stile

kopieren [→ 6.1 Beiträge schreiben/Seiten anlegen]

registrieren [→ 23.4 Weitere nützliche Funktionen für Themes]

Block-Stil-Variation erstellen [→ 23.4 Weitere nützliche Funktionen für Themes]

Block-Theme [→ 8.1 Kommentar-Einstellungen] [→ 10.3 Block-Theme anpassen: den Website-Editor kennenlernen] [→ A Glossar]

erstellen [→ 23.1 Block-Theme erstellen]

exportieren [→ 23.1 Block-Theme erstellen]

Stile [→ 10.3 Block-Theme anpassen: den Website-Editor kennenlernen]

Block-Theme-Entwicklung

404-Template [→ 23.1 Block-Theme erstellen]

Archiv-Template [→ 23.1 Block-Theme erstellen]

Beitragsliste [→ 23.1 Block-Theme erstellen]

Blöcke und Templates sperren [→ 23.4 Weitere nützliche Funktionen für Themes]

Block-Stil-Variation erstellen [→ 23.4 Weitere nützliche Funktionen für Themes]

Block-Typ-Vorlage [→ 23.1 Block-Theme erstellen]

Block-Vorlage in Seitenvorlage umwandeln [→ 23.1 Block-Theme erstellen]

Einzelansicht Beiträge [→ 23.1 Block-Theme erstellen]

Footer [→ 23.1 Block-Theme erstellen]

Grundgerüst [→ 23.1 Block-Theme erstellen]

Grundlagen [→ 23.1 Block-Theme erstellen]

Header [→ 23.1 Block-Theme erstellen]

Index-Template [→ 23.1 Block-Theme erstellen]

Meta-Angaben in der style.css [→ 23.1 Block-Theme erstellen]

Mindestanforderungen [→ 23.1 Block-Theme erstellen]

Referenzen entfernen [→ 23.1 Block-Theme erstellen]

Seiten-Template [→ 23.1 Block-Theme erstellen]

Seitenvorlagen erstellen [→ 23.1 Block-Theme erstellen]

Stilvariationen erstellen [→ 23.1 Block-Theme erstellen]

Stylesheet einbinden [→ 23.1 Block-Theme erstellen]

Such-Template [→ 23.1 Block-Theme erstellen]

Template kopieren [→ 23.1 Block-Theme erstellen]

Template-Vorlagen [→ 23.1 Block-Theme erstellen]

theme.json [→ 23.1 Block-Theme erstellen]

Theme-Dateien exportieren [→ 23.1 Block-Theme erstellen]

Übersetzung [→ 23.1 Block-Theme erstellen]

Vorlage erstellen [→ 23.1 Block-Theme erstellen]

Block-Themes [→ 1.2 Eine kurze Geschichte von WordPress]
[→ 10.1 Classic Themes und Block-Themes: Was ist das?]
[→ 10.5 Neues Theme suchen und installieren]

Templates [→ 10.3 Block-Theme anpassen: den Website-Editor kennenlernen]

Block-Themes und Classic Themes – Unterschiede [→ 23.1 Block-Theme erstellen]

Block-Vorlage [→ 5.1 Das Backend kennenlernen] [→ 6.1 Beiträge schreiben/Seiten anlegen] [→ 10.3 Block-Theme anpassen: den Website-Editor kennenlernen]

erstellen [→ 6.1 Beiträge schreiben/Seiten anlegen] [→ 10.3 Block-Theme anpassen: den Website-Editor kennenlernen]

Block-Werkzeugleiste [→ 6.1 Beiträge schreiben/Seiten anlegen]

Blog [→ 1.1 Das richtige Handbuch für Sie] [→ 13.1 Zielgruppe und Plan] [→ 13.1 Zielgruppe und Plan] [→ 15.2 Das Fediverse]

Blog-Farmen [→ 9.1 Was ist WordPress Multisite?]

Blogger [→ 17.2 Eine Website migrieren]

blogname [→ 9.3 Bedienung des Netzwerks und Unterschiede zum Standard]

Blog-Netzwerk → siehe [Multisite]

Blogs [→ 13.2 Design und Layout]

Blog-System [→ 1.1 Das richtige Handbuch für Sie]

Blueprint → siehe [Playground-Blueprint]

Blueprint-Builder [→ 2.3 Playground mit Blueprints vorkonfigurieren]

Bluesky [→ 15.1 Das IndieWeb]

Bookmarklet [→ 15.2 Das Fediverse]

Bots [→ 19.7 Optimieren für den Klimaschutz]

Brotli [→ 19.2 PageSpeed Insights – Missstände verstehen und bei Bedarf abarbeiten]

Browser-Caching [→ 19.7 Optimieren für den Klimaschutz]

Brute-Force-Attacken [→ 18.1 Login-Sicherheit] [→ 18.1 Login-Sicherheit] [→ 18.3 WordPress abhärten]

Buchhaltungssoftware [→ 12.1 Die Vor- und Nachteile von WooCommerce]

Buchtipps [→ 1.5 Nach der letzten Seite]

BuddyPress [→ 15.1 Das IndieWeb]

Buffer [→ 13.5 Blogging-Tipps]

Bundle [→ 12.2 WooCommerce installieren und einrichten] [→ 12.4 Zusätzliche Erweiterungen für Ihren Shop]

C ⇑

Cache-Generierung [→ 19.4 Einfaches Caching-Plugin ohne Schnickschnack – »Cachify«]

Cache-Invalidierung bei Dateiänderung [→ 23.1 Block-Theme erstellen]

Cachify [→ 19.2 PageSpeed Insights – Missstände verstehen und bei Bedarf abarbeiten]

Caching [→ 19.2 PageSpeed Insights – Missstände verstehen und bei Bedarf abarbeiten]

Browser-Cache [→ 19.2 PageSpeed Insights – Missstände verstehen und bei Bedarf abarbeiten]

Caching-Plugins [→ 13.4 Funktionen und Plugins] [→ 19.1 Grundlagen zur Optimierung]

Canonical URL [→ A Glossar]

CDN [→ 19.3 Optimierungs-Plugin »Autoptimize«] [→ 19.5 Profi-Cache mit WP Rocket]

Checksums prüfen [→ 21.1 Gehackt, was nun?]

Child-Theme [→ 8.1 Kommentar-Einstellungen] [→ 8.3 Datenschutz-Bedenken und -Lösungen] [→ 10.1 Classic Themes und Block-Themes: Was ist das?] [→ 23.2 Classic Theme anpassen und erstellen] [→ A Glossar]

- *erstellen* [→ 23.2 Classic Theme anpassen und erstellen]
- *Parent-Theme-Änderungen übernehmen* [→ 23.2 Classic Theme anpassen und erstellen]
- *Template-Datei anpassen* [→ 23.2 Classic Theme anpassen und erstellen]
- *Template-Datei überschreiben* [→ 23.2 Classic Theme anpassen und erstellen]

Child-Theme mit Parent-Theme vergleichen [→ 23.2 Classic Theme anpassen und erstellen]

Chrome [→ 18.2 Die Wichtigkeit von SSL/TLS]

CI [→ 17.1 Testen und entwickeln mit Entwicklungs-, Staging- und Live-Umgebungen]

Classic Editor [→ 6.8 Classic Editor statt Block-Editor nutzen]

Classic Theme [→ 5.1 Das Backend kennenlernen] [→ 10.1 Classic Themes und Block-Themes: Was ist das?] [→ 23.2 Classic Theme anpassen und erstellen] [→ A Glossar]

anpassen [→ 5.1 Das Backend kennenlernen] [→ 10.4 Classic Theme anpassen] [→ 23.2 Classic Theme anpassen und erstellen]

erstellen [→ 23.2 Classic Theme anpassen und erstellen]

umsetzen [→ 23.2 Classic Theme anpassen und erstellen]

ClassicPress [→ 1.3 Warum WordPress?]

Classic-Theme-Entwicklung

Archiv-Titel ausgeben [→ 23.2 Classic Theme anpassen und erstellen]

Archivübersicht [→ 23.2 Classic Theme anpassen und erstellen]

Auf Kommentare prüfen [→ 23.2 Classic Theme anpassen und erstellen]

Auf Suchergebnisseite prüfen [→ 23.2 Classic Theme anpassen und erstellen]

Barrierearmer Weiterlesen-Link [→ 23.2 Classic Theme anpassen und erstellen]

Beitragsauszug ausgeben [→ 23.2 Classic Theme anpassen und erstellen]

Beitragsbild ausgeben [→ 23.2 Classic Theme anpassen und erstellen]

Beitragsdatum ausgeben [→ 23.2 Classic Theme anpassen und erstellen]

Beitragsinhalt ausgeben [→ 23.2 Classic Theme anpassen und erstellen]

Beitragsliste anzeigen [→ 23.2 Classic Theme anpassen und erstellen]

Beitragstitel ausgeben [→ 23.2 Classic Theme anpassen und erstellen]

Block-basierte Template-Teile [→ 23.2 Classic Theme anpassen und erstellen]

Editor-Stylesheet einbinden [→ 23.2 Classic Theme anpassen und erstellen]

Eigenes Template-Tag erstellen [→ 23.2 Classic Theme anpassen und erstellen]

Einzelansicht von Beiträgen und anderen Nicht-Seiten [→ 23.2 Classic Theme anpassen und erstellen]

Footer [→ 23.2 Classic Theme anpassen und erstellen]

Grundgerüst erstellen [→ 23.2 Classic Theme anpassen und erstellen]

Grundlagen [→ 23.2 Classic Theme anpassen und erstellen]

Header [→ 23.2 Classic Theme anpassen und erstellen]

HTML5-Markup aktivieren [→ 23.2 Classic Theme anpassen und erstellen]

index.php [→ 23.2 Classic Theme anpassen und erstellen]

Inhalt nicht gefunden, 404 [→ 23.2 Classic Theme anpassen und erstellen]

Kategorienliste ausgeben [→ 23.2 Classic Theme anpassen und erstellen]

Kommentaranzahl ausgeben [→ 23.2 Classic Theme anpassen und erstellen]

Kommentare anzeigen [→ 23.2 Classic Theme anpassen und erstellen]

Kommentare auslesen [→ 23.2 Classic Theme anpassen und erstellen]

Kommentare und andere Reaktionen getrennt ausgeben [→ 23.2 Classic Theme anpassen und erstellen]

Kommentarformular ausgeben [→ 23.2 Classic Theme anpassen und erstellen]

Kommentar-Navigation ausgeben [→ 23.2 Classic Theme anpassen und erstellen]

Kommentar-Template einbinden [→ 23.2 Classic Theme anpassen und erstellen]

Logo ausgeben [→ 23.2 Classic Theme anpassen und erstellen]

Loop [→ 23.2 Classic Theme anpassen und erstellen]

Menüposition registrieren [→ 23.2 Classic Theme anpassen und erstellen]

Metadaten [→ 23.2 Classic Theme anpassen und erstellen]

Mindestanforderungen [→ 23.2 Classic Theme anpassen und erstellen]

Navigation anzeigen [→ 23.2 Classic Theme anpassen und erstellen]

Paginierung [→ 23.2 Classic Theme anpassen und erstellen]

Permalink ausgeben [→ 23.2 Classic Theme anpassen und erstellen]

prüfen, ob Kommentare erlaubt sind [→ 23.2 Classic Theme anpassen und erstellen]

prüfen, ob Widgets vorhanden sind [→ 23.2 Classic Theme anpassen und erstellen]

Seiten-Template [→ 23.2 Classic Theme anpassen und erstellen]

singular.php [→ 23.2 Classic Theme anpassen und erstellen]

Sprungmarke vom Weiterlesen-Link entfernen [→ 23.2 Classic Theme anpassen und erstellen]

style.css [→ 23.2 Classic Theme anpassen und erstellen]

Stylesheet einbinden [→ 23.2 Classic Theme anpassen und erstellen]

Suchanfrage ausgeben [→ 23.2 Classic Theme anpassen und erstellen]

Suchergebnisse [→ 23.2 Classic Theme anpassen und erstellen]

Suchformular ausgeben [→ 23.2 Classic Theme anpassen und erstellen]

Template-Teil einbinden [→ 23.2 Classic Theme anpassen und erstellen]

Template-Teile auslagern (Header, Footer, Widget-Bereich) [→ 23.2 Classic Theme anpassen und erstellen]

Template-Teile einbinden [→ 23.2 Classic Theme anpassen und erstellen]

Template-Teile für Blogübersicht [→ 23.2 Classic Theme anpassen und erstellen]

Theme-Aufbau [→ 23.2 Classic Theme anpassen und erstellen]

Theme-Support für Beitragsbilder [→ 23.2 Classic Theme anpassen und erstellen]

Theme-Support für Editor-Styles [→ 23.2 Classic Theme anpassen und erstellen]

Theme-Support für Titel, Logo und Feed-Links [→ 23.2 Classic Theme anpassen und erstellen]

Titel ausgeben [→ 23.2 Classic Theme anpassen und erstellen]

Übersetzung [→ 23.2 Classic Theme anpassen und erstellen]

Untertitel ausgeben [→ 23.2 Classic Theme anpassen und erstellen]

Widget-Bereich ausgeben [→ 23.2 Classic Theme anpassen und erstellen]

Widget-Bereich registrieren [→ 23.2 Classic Theme anpassen und erstellen]

»Pluggable« Funktionen [→ 23.2 Classic Theme anpassen und erstellen]

CLI [→ 1.2 Eine kurze Geschichte von WordPress] [→ 2.2 wp-now: Playground für Entwickler*innen]

Cloudflare [→ 19.5 Profi-Cache mit WP Rocket]

Cloud-Hosting [→ 4.1 Welches Hosting darf's denn sein?]

CO2-Ausstoß [→ 19.7 Optimieren für den Klimaschutz] [→ 19.7 Optimieren für den Klimaschutz]

CO2-Emissionen [→ 19.7 Optimieren für den Klimaschutz]

Code-Editor [→ 22.3 Entwicklungs- und Testumgebung einrichten]

Coding-Standards [→ 22.1 Coding-Standards]

Accessibility [→ 22.1 Coding-Standards]

CSS [→ 22.1 Coding-Standards]

HTML [→ 22.1 Coding-Standards]

PHP [→ 22.1 Coding-Standards]

Community [→ 12.1 Die Vor- und Nachteile von WooCommerce] [→ 12.1 Die Vor- und Nachteile von WooCommerce]

Community-Plattform [→ 14.1 Aufgaben einer Vereins-Website]

Composer [→ 3.3 Lando-Installation testen]

Consent-Banner [→ 13.4 Funktionen und Plugins]

Container [→ 3.1 Installation von Docker]

Content-Management-System [→ 1.1 Das richtige Handbuch für Sie] [→ A Glossar]

Cookies [→ 13.4 Funktionen und Plugins]

Core Web Vitals [→ 19.2 PageSpeed Insights – Missstände verstehen und bei Bedarf abarbeiten] [→ 19.2 PageSpeed Insights – Missstände verstehen und bei Bedarf abarbeiten]

Core-Block-Stile entfernen [→ 23.4 Weitere nützliche Funktionen für Themes]

Cronjob [→ 16.1 WordPress, Plugins und Themes aktualisieren]

Cross-Posting [→ 13.5 Blogging-Tipps]

Crossposting [→ 15.1 Das IndieWeb]

Crosssell [→ 12.2 WooCommerce installieren und einrichten]

CSS [→ 1.2 Eine kurze Geschichte von WordPress]

CSS anpassen (Classic Theme) [→ 5.1 Das Backend kennenlernen]

CSS minifizieren [→ 19.2 PageSpeed Insights – Missstände verstehen und bei Bedarf abarbeiten] [→ 19.2 PageSpeed Insights – Missstände verstehen und bei Bedarf abarbeiten] [→ 19.5 Profi-Cache mit WP Rocket]

CSS-Code optimieren [→ 19.3 Optimierungs-Plugin »Autoptimize«]

CSS-Dateien zusammenfügen [→ 19.3 Optimierungs-Plugin »Autoptimize«]

Custom Fields → siehe [Individuelle Felder]

Custom Post-Type → siehe [Inhaltstyp]

Custom Templates (theme.json) [→ 23.1 Block-Theme erstellen]

Customizer [→ 1.2 Eine kurze Geschichte von WordPress] [→ 5.1 Das Backend kennenlernen] [→ 10.1 Classic Themes und Block-Themes: Was ist das?] [→ 10.3 Block-Theme anpassen: den Website-Editor kennenlernen] [→ 10.4 Classic Theme anpassen] [→ 10.4 Classic Theme anpassen] [→ 10.4 Classic Theme anpassen] [→ 23.2 Classic Theme anpassen und erstellen]

D ⇑

Dark-Mode [→ 10.4 Classic Theme anpassen]

Dashboard [→ 5.1 Das Backend kennenlernen] [→ 9.3 Bedienung des Netzwerks und Unterschiede zum Standard] [→ 13.4 Funktionen und Plugins]

anpassen [→ 5.1 Das Backend kennenlernen]

Dashboard-Widget [→ 5.1 Das Backend kennenlernen] [→ 9.3 Bedienung des Netzwerks und Unterschiede zum Standard]

Datei-Editor [→ 18.3 WordPress abhärten]

Dateien auf Unterschiede vergleichen [→ 21.1 Gehackt, was nun?]

Dateien hochladen [→ 5.1 Das Backend kennenlernen]

Dateigröße [→ 19.6 Bilder optimieren]

Dateisuche [→ 6.5 Medien organisieren]

Dateisystem [→ 9.1 Was ist WordPress Multisite?]

Dateitypen [→ 9.3 Bedienung des Netzwerks und Unterschiede zum Standard] [→ 9.3 Bedienung des Netzwerks und Unterschiede zum Standard]

Daten importieren [→ 17.2 Eine Website migrieren]

Datenbank [→ 9.1 Was ist WordPress Multisite?]

aktualisieren [→ 9.3 Bedienung des Netzwerks und Unterschiede zum Standard] [→ 9.3 Bedienung des Netzwerks und Unterschiede zum Standard]

migrieren [→ 17.2 Eine Website migrieren]

Datenbankabfragen einsehen [→ 22.3 Entwicklungs- und Testumgebung einrichten]

Datenbank-Backup [→ 16.3 Backups planen und durchführen]

Datenbank-Tabellen [→ 9.3 Bedienung des Netzwerks und Unterschiede zum Standard]

Datenbank-Updates → siehe [Datenbank aktualisieren]

Datenbankverbindung [→ 5.3 Einstellungstipps für Fortgeschrittene]

Datenhoheit [→ 1.3 Warum WordPress?]

Datenschutz [→ 12.2 WooCommerce installieren und einrichten] [→ 13.4 Funktionen und Plugins]

Datenschutzeinstellungen [→ 5.2 Einstellungen]

Datenschutzerklärung [→ 6.9 Impressum und Datenschutzerklärung nicht vergessen]

Datenvolumen [→ 19.2 PageSpeed Insights – Missstände verstehen und bei Bedarf abarbeiten]

DDoS [→ 18.3 WordPress abhärten]

Deaktivieren [→ 11.6 Plugins deaktivieren und löschen]

Debuggen [→ 22.3 Entwicklungs- und Testumgebung einrichten]

Debug-Mode aktivieren [→ 22.3 Entwicklungs- und Testumgebung einrichten]

Deployment [→ 17.1 Testen und entwickeln mit Entwicklungs-, Staging- und Live-Umgebungen] [→ 17.1 Testen und entwickeln mit Entwicklungs-, Staging- und Live-Umgebungen]

Design [→ 10.1 Classic Themes und Block-Themes: Was ist das?]

dewp.space [→ 15.2 Das Fediverse]

Dezentrales Netzwerk [→ A Glossar]

DHL [→ 12.2 WooCommerce installieren und einrichten]

Digitales Produkt [→ 12.2 WooCommerce installieren und einrichten]

Diskussion [→ 8.1 Kommentar-Einstellungen]

DNS-Prefetch [→ 19.5 Profi-Cache mit WP Rocket]

DNS-Rebind-Schutz [→ 3.3 Lando-Installation testen]

Docker [→ 3.1 Installation von Docker]

Installation [→ 3.1 Installation von Docker]

OrbStack [→ 3.1 Installation von Docker]

Docker Desktop [→ 3.1 Installation von Docker]

Docker-Image [→ 3.1 Installation von Docker]

Dokumentation [→ 1.5 Nach der letzten Seite]

Dokumentenansicht [→ 6.1 Beiträge schreiben/Seiten anlegen]

DOM [→ 24.11 Weitere nützliche Funktionen für Plugins]

Domain [→ 9.1 Was ist WordPress Multisite?]

anpassen [→ 17.2 Eine Website migrieren]

Domain-Mapping [→ 9.2 Die Multisite-Einrichtung]

Double-Opt-in [→ 7.2 Benutzer*innen hinzufügen] [→ 12.2 WooCommerce installieren und einrichten]

DPD [→ 12.2 WooCommerce installieren und einrichten]

Drei-Punkte-Menü

Block [→ 6.1 Beiträge schreiben/Seiten anlegen]

Block-Editor [→ 6.1 Beiträge schreiben/Seiten anlegen]

DSGVO [→ 4.1 Welches Hosting darf's denn sein?] [→ 5.1 Das Backend kennenlernen] [→ 8.3 Datenschutz-Bedenken und -Lösungen] [→ 12.2 WooCommerce installieren und einrichten]

Daten anonymisieren [→ 8.3 Datenschutz-Bedenken und -Lösungen]

Daten löschen [→ 8.3 Datenschutz-Bedenken und -Lösungen]

Duoton [→ 10.3 Block-Theme anpassen: den Website-Editor kennenlernen]

Dynamische Blöcke [→ 24.5 Editor-Blöcke erstellen]

E ⇑

E-Commerce [→ 12.1 Die Vor- und Nachteile von WooCommerce]

Editor [→ 6.1 Beiträge schreiben/Seiten anlegen]

Shortcuts [→ 10.3 Block-Theme anpassen: den Website-Editor kennenlernen]

Einrichtung [→ 9.2 Die Multisite-Einrichtung]

Einstellungen [→ 5.2 Einstellungen]

Einstellungen für Fortgeschrittene [→ 5.3 Einstellungstipps für Fortgeschrittene]

E-Mail-Domains limitieren [→ 9.3 Bedienung des Netzwerks und Unterschiede zum Standard]

E-Mail-Newsletter [→ 14.1 Aufgaben einer Vereins-Website]

Embed Privacy [→ 13.4 Funktionen und Plugins]

Emojis entfernen [→ 19.3 Optimierungs-Plugin »Autoptimize«]

Engagement [→ 15.1 Das IndieWeb]

Entwicklungsgrundlagen [→ 22.1 Coding-Standards]

Entwicklungs-Konzepte [→ 22.2 WordPress-Konzepte]

Hooks [→ 22.2 WordPress-Konzepte]

Entwicklungsumgebung [→ 17.1 Testen und entwickeln mit Entwicklungs-, Staging- und Live-Umgebungen]

einrichten [→ 22.3 Entwicklungs- und Testumgebung einrichten]

Entwurf speichern [→ 6.1 Beiträge schreiben/Seiten anlegen]

Erneuerbare Energien [→ 19.7 Optimieren für den Klimaschutz]

Errata [→ 1.4 Anleitung zum Handbuch]

Erweiterbarkeit [→ 11.1 Plugins testen – sicher]

Erweiterungen → siehe [Plugins]

EU [→ 12.2 WooCommerce installieren und einrichten] [→ 12.2 WooCommerce installieren und einrichten] [→ 12.2 WooCommerce installieren und einrichten]

Excerpt [→ 6.1 Beiträge schreiben/Seiten anlegen]

Export [→ 5.1 Das Backend kennenlernen]

F ⇑

Facebook [→ 15.2 Das Fediverse]

Facepile [→ 15.1 Das IndieWeb]

Fallback-Template index.php [→ 23.2 Classic Theme anpassen und erstellen]

Farbfehlsichtigkeiten [→ 10.6 Barrierefreiheit von Themes testen]

Farbkontraste [→ 10.6 Barrierefreiheit von Themes testen] [→ 10.6 Barrierefreiheit von Themes testen] [→ 19.2 PageSpeed Insights – Missstände verstehen und bei Bedarf abarbeiten]

Farbpalette [→ 10.3 Block-Theme anpassen: den Website-Editor kennenlernen]

Farbverlauf [→ 10.3 Block-Theme anpassen: den Website-Editor kennenlernen]

Favicon → siehe [Website-Icon]

Fediverse [→ 13.2 Design und Layout] [→ 13.5 Blogging-Tipps] [→ 15.1 Das IndieWeb] [→ 15.1 Das IndieWeb] [→ 15.2 Das Fediverse] [→ 15.2 Das Fediverse] [→ 15.3 IndieWeb oder Fediverse?]

Fehlerseite erstellen (Block-Theme) [→ 23.1 Block-Theme erstellen]

Filter-Hook → siehe [Hooks]

Fluid Typography [→ 23.1 Block-Theme erstellen]

Fokusreihenfolge [→ 10.6 Barrierefreiheit von Themes testen]

Folgen [→ 15.2 Das Fediverse]

Follower [→ 15.2 Das Fediverse]

Footer [→ 13.2 Design und Layout]

Forced Updates [→ 16.1 WordPress, Plugins und Themes aktualisieren]

Fork [→ 1.2 Eine kurze Geschichte von WordPress]

Freemium [→ 12.2 WooCommerce installieren und einrichten]

Frontend [→ 5.1 Das Backend kennenlernen] [→ 10.3 Block-Theme anpassen: den Website-Editor kennenlernen] [→ A Glossar]

Full-Site-Editing → siehe [Website-Editor]

functions.php [→ 23.1 Block-Theme erstellen]

G ⇑

Gehackt

- *aufräumen* [→ 21.1 Gehackt, was nun?]
- *reparieren oder neu aufbauen* [→ 21.1 Gehackt, was nun?]
- *Sofortmaßnahmen* [→ 21.1 Gehackt, was nun?]

Germanized für WooCommerce [→ 12.2 WooCommerce installieren und einrichten]

Geschäftssysteme [→ 12.1 Die Vor- und Nachteile von WooCommerce]

Geschichte von WordPress [→ 1.2 Eine kurze Geschichte von WordPress]

Geschlossener Bereich [→ 14.2 Design, Struktur und Navigation]

Gestaltung [→ 13.1 Zielgruppe und Plan]

Gettext [→ A Glossar]

Gettext-Funktion [→ 22.2 WordPress-Konzepte]

Git [→ 17.1 Testen und entwickeln mit Entwicklungs-, Staging- und Live-Umgebungen]

GitHub [→ 11.3 Gute Plugins erkennen] [→ 15.1 Das IndieWeb] [→ 17.1 Testen und entwickeln mit Entwicklungs-, Staging- und Live-Umgebungen]

GitLab [→ 17.1 Testen und entwickeln mit Entwicklungs-, Staging- und Live-Umgebungen]

Google [→ 19.1 Grundlagen zur Optimierung]

Google Analytics [→ 13.4 Funktionen und Plugins]

Google Fonts [→ 19.3 Optimierungs-Plugin »Autoptimize«]

Google Fonts lokal einbinden [→ 23.1 Block-Theme erstellen]

Google Pay [→ 12.2 WooCommerce installieren und einrichten]

google-webfonts-helper [→ 23.1 Block-Theme erstellen]

Gravatar [→ 5.2 Einstellungen] [→ 7.4 Benutzerprofil bearbeiten] [→ 8.3 Datenschutz-Bedenken und -Lösungen] [→ 8.4 Spamschutz mit Antispam Bee] [→ 13.4 Funktionen und Plugins] [→ A Glossar]

Datenschutz [→ 5.2 Einstellungen]

Größere WordPress-Updates [→ 16.1 WordPress, Plugins und Themes aktualisieren]

Gruppe-Block [→ 10.3 Block-Theme anpassen: den Website-Editor kennenlernen]

Gruppen [→ 14.3 Funktionalitäten und Plugins]

Gutenberg → siehe [Block-Editor]

Gzip [→ 19.2 PageSpeed Insights – Missstände verstehen und bei Bedarf abarbeiten]

H ⇑

Hacker-Angriffe [→ 18.1 Login-Sicherheit]

Hacking [→ 9.3 Bedienung des Netzwerks und Unterschiede zum Standard]

Hashtag [→ 15.2 Das Fediverse]

Header-Bilder [→ 13.2 Design und Layout]

Heartbeat-API [→ 19.5 Profi-Cache mit WP Rocket]

Hello Dolly [→ 11.4 Plugins installieren]

Hintergrundbild [→ 10.3 Block-Theme anpassen: den Website-Editor kennenlernen] [→ 10.4 Classic Theme anpassen]

Hintergrundbild setzen (Classic Theme) [→ 5.1 Das Backend kennenlernen]

Hintergrundupdates [→ 16.1 WordPress, Plugins und Themes aktualisieren]

Homebrew Website Club [→ 15.1 Das IndieWeb]

Homepage [→ 10.4 Classic Theme anpassen]

Homepage-Einstellungen [→ 5.2 Einstellungen]

Honeypot [→ 8.4 Spamschutz mit Antispam Bee]

Hooks [→ 22.2 WordPress-Konzepte] [→ 22.2 WordPress-Konzepte] [→ 22.2 WordPress-Konzepte] [→ A Glossar]

Hoster wechseln [→ 17.1 Testen und entwickeln mit Entwicklungs-, Staging- und Live-Umgebungen]

Hosting [→ 1.3 Warum WordPress?] [→ 9.1 Was ist WordPress Multisite?]

optimieren [→ 19.7 Optimieren für den Klimaschutz] [→ 19.7 Optimieren für den Klimaschutz]

Hosting-Anbieter [→ 19.7 Optimieren für den Klimaschutz]

Hosting-Paket [→ 4.1 Welches Hosting darf's denn sein?]

htaccess [→ 9.2 Die Multisite-Einrichtung]

HTML [→ 1.2 Eine kurze Geschichte von WordPress]

HTML-Validierung [→ 22.1 Coding-Standards]

htpasswd [→ 17.1 Testen und entwickeln mit Entwicklungs-, Staging- und Live-Umgebungen]

HTTP/2 [→ 19.3 Optimierungs-Plugin »Autoptimize«] [→ 19.5 Profi-Cache mit WP Rocket]

HTTPS [→ 18.2 Die Wichtigkeit von SSL/TLS] [→ 18.2 Die Wichtigkeit von SSL/TLS] [→ 19.2 PageSpeed Insights – Missstände verstehen und bei Bedarf abarbeiten]

Hyperscaler [→ 4.1 Welches Hosting darf’s denn sein?]

⇑

iFrames [→ 19.2 PageSpeed Insights – Missstände verstehen und bei Bedarf abarbeiten]

Import [→ 2.3 Playground mit Blueprints vorkonfigurieren] [→ 5.1 Das Backend kennenlernen] [→ 6.1 Beiträge schreiben/Seiten anlegen]

Blogger [→ 5.1 Das Backend kennenlernen]

LiveJournal [→ 5.1 Das Backend kennenlernen]

Tumblr [→ 5.1 Das Backend kennenlernen]

WordPress.com [→ 5.1 Das Backend kennenlernen]

Impressum [→ 6.9 Impressum und Datenschutzerklärung nicht vergessen] [→ 12.2 WooCommerce installieren und einrichten]

In Vorlage umwandeln [→ 10.3 Block-Theme anpassen: den Website-Editor kennenlernen]

Indexierung in Suchmaschinen [→ 5.2 Einstellungen]

IndieAuth [→ 15.1 Das IndieWeb]

IndieWeb [→ 1.1 Das richtige Handbuch für Sie] [→ 8.2 Kommentare freigeben] [→ 8.5 Webmentions] [→ 15.1 Das IndieWeb] [→ 15.1 Das IndieWeb] [→ 15.1 Das IndieWeb] [→ 15.1 Das IndieWeb] [→ 15.3 IndieWeb oder Fediverse?]

Individuelle Felder [→ 6.4 Individuelle Felder nutzen]

Individuelles Feld anlegen [→ 6.4 Individuelle Felder nutzen]

Individuelles Feld nutzen [→ 6.4 Individuelle Felder nutzen]

Individuelles Template [→ 10.3 Block-Theme anpassen: den Website-Editor kennenlernen]

Inhalt nur für Screenreader ausgeben [→ 23.2 Classic Theme anpassen und erstellen]

Inhalt speichern [→ 6.1 Beiträge schreiben/Seiten anlegen]

Inhalte [→ 13.1 Zielgruppe und Plan]

Inhalte erstellen [→ 6.1 Beiträge schreiben/Seiten anlegen]

Inhalte veröffentlichen [→ 6.1 Beiträge schreiben/Seiten anlegen]

- *Sichtbarkeit* [→ 6.1 Beiträge schreiben/Seiten anlegen]
- *Status* [→ 6.1 Beiträge schreiben/Seiten anlegen]

Inhalte verwalten [→ 6.2 Beiträge/Seiten verwalten]

Inhaltseditor [→ 6.1 Beiträge schreiben/Seiten anlegen]

Inhaltsplanung [→ 6.7 Inhalte planen]

Inhaltsstruktur [→ 19.7 Optimieren für den Klimaschutz]

Inhaltsstruktur planen [→ 6.7 Inhalte planen]

Inhalts-Template ändern [→ 6.1 Beiträge schreiben/Seiten anlegen]

Inhaltstyp [→ 5.1 Das Backend kennenlernen] [→ 9.2 Die Multisite-Einrichtung] [→ 14.1 Aufgaben einer Vereins-Website] [→ 15.1 Das IndieWeb] [→ 24.1 Installation von Node.js] [→ 24.3 Inhaltstyp erstellen] [→ 24.3 Inhaltstyp erstellen] [→ A Glossar]

erstellen [→ 24.3 Inhaltstyp erstellen]

Installation [→ 1.4 Anleitung zum Handbuch]

lokal [→ 2.1 WordPress Playground – eine ganze WordPress-Installation lokal im Browser]

Interactivity API [→ 24.11 Weitere nützliche Funktionen für Plugins] [→ A Glossar]

Internationalisierung [→ 22.2 WordPress-Konzepte] [→ A Glossar]

Intranet [→ 13.1 Zielgruppe und Plan] [→ 18.3 WordPress abhärten]

IP-Adresse [→ 8.3 Datenschutz-Bedenken und -Lösungen] [→ 18.3 WordPress abhärten]

J ⇑

JavaScript [→ 13.4 Funktionen und Plugins]

minifizieren [→ 19.5 Profi-Cache mit WP Rocket]

verzögern [→ 19.3 Optimierungs-Plugin »Autoptimize«]

JavaScript-Dateien zusammenfügen [→ 19.3 Optimierungs-Plugin »Autoptimize«]

Jetpack [→ 12.2 WooCommerce installieren und einrichten]

Jimdo [→ 1.3 Warum WordPress?]

JPEG XL [→ 19.6 Bilder optimieren]

JSON [→ 18.3 WordPress abhärten]

JSX [→ 24.4 Panel in der Editor-Seitenleiste zur Bearbeitung der Metadaten]

K ⇑

Kategorien [→ 5.1 Das Backend kennenlernen] [→ 5.2 Einstellungen]

verwalten [→ 6.2 Beiträge/Seiten verwalten]

zuweisen [→ 6.1 Beiträge schreiben/Seiten anlegen]

Kleine WordPress-Updates [→ 16.1 WordPress, Plugins und Themes aktualisieren]

Kleinunternehmerregelung [→ 12.2 WooCommerce installieren und einrichten]

Klimaschutz [→ 19.1 Grundlagen zur Optimierung] [→ 19.7 Optimieren für den Klimaschutz]

Knowledge Graph [→ 20.2 Viele SEO-Maßnahmen in einem Plugin mit »Yoast SEO«]

Kommentare [→ 1.1 Das richtige Handbuch für Sie] [→ 5.1 Das Backend kennenlernen] [→ 5.1 Das Backend kennenlernen] [→ 8.1 Kommentar-Einstellungen] [→ 15.1 Das IndieWeb]

Benachrichtigungen [→ 5.2 Einstellungen]

bereinigen [→ 19.5 Profi-Cache mit WP Rocket]

Datenschutz [→ 8.3 Datenschutz-Bedenken und -Lösungen]

Einstellungen [→ 5.2 Einstellungen] [→ 6.1 Beiträge schreiben/Seiten anlegen]

erlauben [→ 5.2 Einstellungen]

freigeben [→ 5.2 Einstellungen] [→ 8.2 Kommentare freigeben]

Funktion [→ 13.2 Design und Layout]

Kommentartypen [→ 8.2 Kommentare freigeben]

Moderation [→ 5.2 Einstellungen]

Spamschutz [→ 8.4 Spamschutz mit Antispam Bee]

Sperrliste [→ 5.2 Einstellungen]

Kompatibilität [→ 11.5 Plugin aktualisieren]

Kontaktformular [→ 11.2 Von der Anforderung zum Plugin] [→ 14.3 Funktionalitäten und Plugins]

Kostenpflichtige Plugins [→ 11.1 Plugins testen – sicher] [→ 11.3 Gute Plugins erkennen] [→ 11.5 Plugin aktualisieren]

Kreditkarte [→ 12.2 WooCommerce installieren und einrichten]

Kritisches CSS [→ 19.2 PageSpeed Insights – Missstände verstehen und bei Bedarf abarbeiten] [→ 19.3 Optimierungs-Plugin »Autoptimize«]

Kubrick-Theme [→ 1.2 Eine kurze Geschichte von WordPress]

L ⇑

Ladengeschäft [→ 12.1 Die Vor- und Nachteile von WooCommerce]

Ladezeit [→ 19.1 Grundlagen zur Optimierung]

Lagerbestand [→ 12.2 WooCommerce installieren und einrichten]

Lando [→ 3.1 Installation von Docker]

Problem mit lokalen Domains [→ 3.3 Lando-Installation testen]

Largest Contentful Paint [→ 19.2 PageSpeed Insights – Missstände verstehen und bei Bedarf abarbeiten] [→ 19.5 Profi-Cache mit WP Rocket]

Lastschrift [→ 12.2 WooCommerce installieren und einrichten]

Layout [→ 10.3 Block-Theme anpassen: den Website-Editor kennenlernen]

Layoutverschiebungen [→ 19.2 PageSpeed Insights – Missstände verstehen und bei Bedarf abarbeiten]

Lazy Loading [→ 19.2 PageSpeed Insights – Missstände verstehen und bei Bedarf abarbeiten] [→ 19.5 Profi-Cache mit WP Rocket] [→ 19.5 Profi-Cache mit WP Rocket] [→ 19.7 Optimieren für den Klimaschutz]

Lesbarkeit [→ 10.6 Barrierefreiheit von Themes testen]

Let's Encrypt [→ 18.2 Die Wichtigkeit von SSL/TLS]

Lightbox [→ 6.1 Beiträge schreiben/Seiten anlegen]

Likes [→ 15.1 Das IndieWeb] [→ 15.2 Das Fediverse]

Link anpassen [→ 6.1 Beiträge schreiben/Seiten anlegen]

Little, Mike [→ 1.2 Eine kurze Geschichte von WordPress]

LiveJournal [→ 17.2 Eine Website migrieren]

Live-Umgebungen [→ 17.1 Testen und entwickeln mit Entwicklungs-, Staging- und Live-Umgebungen]

Live-Vorschau [→ 10.2 Installierte Themes vorab ansehen und aktivieren] [→ 10.4 Classic Theme anpassen] [→ 10.5 Neues Theme suchen und installieren] [→ 11.3 Gute Plugins erkennen]

Live-Website [→ 17.1 Testen und entwickeln mit Entwicklungs-, Staging- und Live-Umgebungen]

Lizenz-Bestimmungen [→ 11.3 Gute Plugins erkennen]

Load Balancing [→ 4.1 Welches Hosting darf's denn sein?]

localhost [→ 3.3 Lando-Installation testen]

Login [→ 18.1 Login-Sicherheit]

Lokale Testumgebung [→ 11.4 Plugins installieren]

Loop [→ 23.2 Classic Theme anpassen und erstellen]

Lossless Kompression [→ 19.6 Bilder optimieren]

M ⇑

MailPoet [→ 12.2 WooCommerce installieren und einrichten]

Major Updates [→ 16.1 WordPress, Plugins und Themes aktualisieren]

Managed Server [→ 4.1 Welches Hosting darf's denn sein?]

Managed WordPress Hosting [→ 4.1 Welches Hosting darf's denn sein?] [→ 4.1 Welches Hosting darf's denn sein?]

Manuelle Aktualisierungen [→ 11.5 Plugin aktualisieren]

MariaDB [→ 1.3 Warum WordPress?] [→ 4.1 Welches Hosting darf's denn sein?]

Marketing-E-Mails [→ 12.2 WooCommerce installieren und einrichten]

Marktplatz [→ 12.4 Zusätzliche Erweiterungen für Ihren Shop]

Mastodon [→ 8.5 Webmentions] [→ 15.1 Das IndieWeb] [→ 15.2 Das Fediverse] [→ 15.2 Das Fediverse] [→ A Glossar]

Mediathek [→ 5.1 Das Backend kennenlernen] [→ 6.5 Medien organisieren] [→ A Glossar]

- *Listen- und Rasteransicht* [→ 6.5 Medien organisieren]
- *Mehrfachauswahl* [→ 6.5 Medien organisieren]
- *Suche* [→ 6.5 Medien organisieren]

Medien einfügen [→ 6.1 Beiträge schreiben/Seiten anlegen]

Medieneinstellungen [→ 5.2 Einstellungen]

Medienverwaltung [→ 6.5 Medien organisieren] [→ 9.1 Was ist WordPress Multisite?]

Meetups [→ 12.1 Die Vor- und Nachteile von WooCommerce]

Mehrere Blöcke auswählen [→ 6.1 Beiträge schreiben/Seiten anlegen]

Mehrfachbearbeitung von Beiträgen/Seiten [→ 6.2 Beiträge/Seiten verwalten]

Mehrsprachigkeit [→ 9.1 Was ist WordPress Multisite?]

- *MultilingualPress* [→ 9.1 Was ist WordPress Multisite?]
- *Multisite Language Switcher* [→ 9.1 Was ist WordPress Multisite?]

Menüeinträge hinzufügen [→ 10.4 Classic Theme anpassen]

Menü-Positionen [→ 5.1 Das Backend kennenlernen] [→ 10.4 Classic Theme anpassen] [→ 10.4 Classic Theme anpassen]

Menüs

bearbeiten [→ 10.4 Classic Theme anpassen]

Classic Theme [→ 5.1 Das Backend kennenlernen]

Meta-Box [→ 12.2 WooCommerce installieren und einrichten]

Metadaten [→ 6.4 Individuelle Felder nutzen]

Metadaten eines Inhalts ausgeben [→ 6.4 Individuelle Felder nutzen]

Microblog [→ 15.1 Das IndieWeb]

Microformats [→ 15.1 Das IndieWeb] [→ 15.1 Das IndieWeb]

Micropub [→ 15.1 Das IndieWeb]

Migration [→ 17.1 Testen und entwickeln mit Entwicklungs-, Staging- und Live-Umgebungen]

Monetarisierung [→ 13.1 Zielgruppe und Plan]

Movable Type [→ 17.2 Eine Website migrieren]

Mullenweg, Matt [→ 1.2 Eine kurze Geschichte von WordPress] [→ 9.1 Was ist WordPress Multisite?]

Multipurpose-Themes [→ 10.5 Neues Theme suchen und installieren]

Multisite [→ 1.2 Eine kurze Geschichte von WordPress] [→ 2.1 WordPress Playground – eine ganze WordPress-Installation lokal im Browser] [→ 7.1 Rollen verstehen] [→ 9.1 Was ist WordPress Multisite?] [→ 9.1 Was ist WordPress Multisite?] [→ 9.1 Was ist WordPress Multisite?] [→ 9.1 Was ist WordPress

Multisite?] [→ 9.2 Die Multisite-Einrichtung] [→ 9.3 Bedienung des Netzwerks und Unterschiede zum Standard] [→ 9.3 Bedienung des Netzwerks und Unterschiede zum Standard] [→ 9.3 Bedienung des Netzwerks und Unterschiede zum Standard]

MySQL [→ 1.3 Warum WordPress?] [→ 4.1 Welches Hosting darf's denn sein?]

N ⇑

Nachhaltigkeit [→ 19.7 Optimieren für den Klimaschutz]

Nachnahme [→ 12.2 WooCommerce installieren und einrichten]

Namespace [→ 24.3 Inhaltstyp erstellen] [→ 24.5 Editor-Blöcke erstellen]

Navigations-Block [→ 10.3 Block-Theme anpassen: den Website-Editor kennenlernen]

Navigationsmenü erstellen [→ 10.4 Classic Theme anpassen]

Netzwerkadministrator [→ 9.2 Die Multisite-Einrichtung] [→ 9.3 Bedienung des Netzwerks und Unterschiede zum Standard]

Netzwerkanalyse [→ 19.1 Grundlagen zur Optimierung]

Netzwerk-Einrichtung [→ 9.2 Die Multisite-Einrichtung]

Netzwerk-Einstellungen [→ 9.3 Bedienung des Netzwerks und Unterschiede zum Standard]

Netzwerknutzlasten [→ 19.2 PageSpeed Insights – Missstände verstehen und bei Bedarf abarbeiten]

Netzwerkverwaltung [→ 9.3 Bedienung des Netzwerks und Unterschiede zum Standard]

Netzwerkweit

aktivieren [→ 9.3 Bedienung des Netzwerks und Unterschiede zum Standard] [→ 9.3 Bedienung des Netzwerks und Unterschiede zum Standard]

freischalten [→ 9.3 Bedienung des Netzwerks und Unterschiede zum Standard]

Neue Registrierung erlauben [→ 9.3 Bedienung des Netzwerks und Unterschiede zum Standard]

Neue Seite erstellen [→ 10.3 Block-Theme anpassen: den Website-Editor kennenlernen]

Neues Menü erstellen [→ 10.4 Classic Theme anpassen]

Neues Template [→ 10.3 Block-Theme anpassen: den Website-Editor kennenlernen]

Neues Theme suchen [→ 10.5 Neues Theme suchen und installieren]

Newsletter [→ 13.1 Zielgruppe und Plan]

versenden [→ 14.3 Funktionalitäten und Plugins]

Nightly-Version [→ 16.1 WordPress, Plugins und Themes aktualisieren]

Node.js [→ 2.2 wp-now: Playground für Entwickler*innen] [→ 24.1 Installation von Node.js] [→ A Glossar]

installieren [→ 24.1 Installation von Node.js]

Nonce [→ A Glossar]

Notfallmaßnahmen [→ 21.1 Gehackt, was nun?]

npm [→ 24.1 Installation von Node.js]

npx [→ 2.2 wp-now: Playground für Entwickler*innen]

Nulled Plugins [→ 11.3 Gute Plugins erkennen]

O ⇑

OAuth [→ 15.1 Das IndieWeb]

oEmbed [→ 20.1 Schlank mit allen wichtigen Funktionen – Plugin »The SEO Framework«]

Öffentliche Vorschau [→ 13.4 Funktionen und Plugins]

One-Click-Installer [→ 4.2 Installation mit einem One-Click-Installer]

Online-Community [→ 14.1 Aufgaben einer Vereins-Website]

Online-Präsenz [→ 13.1 Zielgruppe und Plan]

Online-Shop [→ 1.1 Das richtige Handbuch für Sie] [→ 4.1 Welches Hosting darf's denn sein?] [→ 11.3 Gute Plugins erkennen] [→ 18.2 Die Wichtigkeit von SSL/TLS]

Open Source [→ 1.3 Warum WordPress?]

OpenGraph [→ 15.1 Das IndieWeb]

Open-Graph-Metadaten [→ 20.1 Schlank mit allen wichtigen Funktionen – Plugin »The SEO Framework«]

OpenID [→ 15.1 Das IndieWeb]

Open-Source-Lizenz [→ 1.2 Eine kurze Geschichte von WordPress]

Openverse [→ 6.1 Beiträge schreiben/Seiten anlegen]

options.php [→ 5.3 Einstellungstipps für Fortgeschrittene]

OrbStack [→ 3.1 Installation von Docker]

P ⇑

package.json [→ 24.4 Panel in der Editor-Seitenleiste zur Bearbeitung der Metadaten]

Page-Builder [→ 10.5 Neues Theme suchen und installieren] [→ 24.11 Weitere nützliche Funktionen für Plugins]

PageSpeed [→ 10.5 Neues Theme suchen und installieren] [→ 19.2 PageSpeed Insights – Missstände verstehen und bei Bedarf abarbeiten]

PageSpeed Insights [→ 19.2 PageSpeed Insights – Missstände verstehen und bei Bedarf abarbeiten]

Papierkorb [→ 6.2 Beiträge/Seiten verwalten] [→ 8.2 Kommentare freigeben]

Parent-Theme [→ 23.2 Classic Theme anpassen und erstellen]

Password Policies [→ 18.1 Login-Sicherheit] [→ 18.1 Login-Sicherheit] [→ A Glossar]

Passwort [→ 7.3 Benutzer*innen verwalten: sortieren, filtern, suchen]

wiederherstellen [→ 21.2 Die häufigsten Standardproblemlösungen]

Passwortgeneratoren [→ 18.1 Login-Sicherheit]

Passwortmanager [→ 7.4 Benutzerprofil bearbeiten] [→ 18.1 Login-Sicherheit] [→ 18.1 Login-Sicherheit] [→ 18.1 Login-Sicherheit]

Passwortregeln → siehe [Password Policies]

Passwortschutz [→ 6.1 Beiträge schreiben/Seiten anlegen] [→ 17.1 Testen und entwickeln mit Entwicklungs-, Staging- und Live-Umgebungen]

Passwort-Sicherheit [→ 18.1 Login-Sicherheit]

Patch Updates [→ 16.1 WordPress, Plugins und Themes aktualisieren]

Pattern → siehe [Block-Vorlage]

patterns-Verzeichnis [→ 23.1 Block-Theme erstellen]

Payment-Gateways [→ 12.2 WooCommerce installieren und einrichten] [→ 12.2 WooCommerce installieren und einrichten] [→ 12.2 WooCommerce installieren und einrichten]

PayPal [→ 12.4 Zusätzliche Erweiterungen für Ihren Shop]

Performance [→ 11.3 Gute Plugins erkennen]

- *Cumulative Layout Shift* [→ 19.2 PageSpeed Insights – Missstände verstehen und bei Bedarf abarbeiten]
- *First Contentful Paint* [→ 19.2 PageSpeed Insights – Missstände verstehen und bei Bedarf abarbeiten]
- *Interaction to Next Paint* [→ 19.2 PageSpeed Insights – Missstände verstehen und bei Bedarf abarbeiten]
- *Largest Contentful Paint* [→ 19.2 PageSpeed Insights – Missstände verstehen und bei Bedarf abarbeiten]

Speed Index [→ 19.2 PageSpeed Insights – Missstände verstehen und bei Bedarf abarbeiten]

Total Blocking Time [→ 19.2 PageSpeed Insights – Missstände verstehen und bei Bedarf abarbeiten]

Weiterleitungen vermeiden [→ 19.2 PageSpeed Insights – Missstände verstehen und bei Bedarf abarbeiten]

Performance-Optimierung [→ 1.4 Anleitung zum Handbuch]

Performance-Probleme [→ 9.1 Was ist WordPress Multisite?]

Permalinks [→ 5.2 Einstellungen] [→ 6.1 Beiträge schreiben/Seiten anlegen] [→ 17.1 Testen und entwickeln mit Entwicklungs-, Staging- und Live-Umgebungen] [→ A Glossar]

Personenbezogene Daten exportieren/löschen [→ 5.1 Das Backend kennenlernen] [→ 8.3 Datenschutz-Bedenken und -Lösungen]

Perspektive [→ 13.1 Zielgruppe und Plan]

PESOS [→ 15.1 Das IndieWeb]

PHP [→ 1.2 Eine kurze Geschichte von WordPress] [→ 4.1 Welches Hosting darf's denn sein?]

PHP 8.1 [→ 16.1 WordPress, Plugins und Themes aktualisieren]

PHP 8.2 [→ 16.1 WordPress, Plugins und Themes aktualisieren]

PHP 8.3 [→ 16.1 WordPress, Plugins und Themes aktualisieren]

PHP_CodeSniffer [→ 22.1 Coding-Standards]

phpMyAdmin [→ 17.2 Eine Website migrieren]

PHP-Version [→ 4.1 Welches Hosting darf's denn sein?] [→ 11.1 Plugins testen – sicher] [→ 17.1 Testen und entwickeln mit

Entwicklungs-, Staging- und Live-Umgebungen]

Physisches Produkt [→ 12.2 WooCommerce installieren und einrichten] [→ 12.2 WooCommerce installieren und einrichten]

Pingbacks [→ 5.2 Einstellungen] [→ 8.2 Kommentare freigeben] [→ 13.2 Design und Layout] [→ 15.1 Das IndieWeb] [→ 15.1 Das IndieWeb] [→ 15.1 Das IndieWeb] [→ 15.1 Das IndieWeb]

Pingdom [→ 10.5 Neues Theme suchen und installieren]

Ping-o-Matic [→ 5.2 Einstellungen]

Pixelfed [→ 15.2 Das Fediverse]

Playground [→ 2.1 WordPress Playground – eine ganze WordPress-Installation lokal im Browser] [→ 11.3 Gute Plugins erkennen]

- *Blueprint anwenden* [→ 2.3 Playground mit Blueprints vorkonfigurieren]
- *Blueprint erstellen* [→ 2.3 Playground mit Blueprints vorkonfigurieren]
- *Blueprint-Optionen* [→ 2.3 Playground mit Blueprints vorkonfigurieren]
- *Export* [→ 2.1 WordPress Playground – eine ganze WordPress-Installation lokal im Browser]
- *Import* [→ 2.1 WordPress Playground – eine ganze WordPress-Installation lokal im Browser]
- *Plugin vorinstallieren* [→ 2.1 WordPress Playground – eine ganze WordPress-Installation lokal im Browser]

Speichermöglichkeiten [→ 2.1 WordPress Playground – eine ganze WordPress-Installation lokal im Browser]

Studio [→ 2.4 Komfortablere Playground-Nutzung mit »Studio« von WordPress.com]

Theme vorinstallieren [→ 2.1 WordPress Playground – eine ganze WordPress-Installation lokal im Browser]

Playground für Entwickler*innen → siehe [wp-now]

Playground-Blueprint [→ 2.3 Playground mit Blueprints vorkonfigurieren]

*Benutzer*in anmelden* [→ 2.3 Playground mit Blueprints vorkonfigurieren] [→ 2.3 Playground mit Blueprints vorkonfigurieren]

Import [→ 2.3 Playground mit Blueprints vorkonfigurieren]

Konstanten setzen [→ 2.3 Playground mit Blueprints vorkonfigurieren] [→ 2.3 Playground mit Blueprints vorkonfigurieren]

Metadaten [→ 2.3 Playground mit Blueprints vorkonfigurieren]

Multisite aktivieren [→ 2.3 Playground mit Blueprints vorkonfigurieren]

Netzwerkzugriff aktivieren [→ 2.3 Playground mit Blueprints vorkonfigurieren]

Optionen setzen [→ 2.3 Playground mit Blueprints vorkonfigurieren] [→ 2.3 Playground mit Blueprints vorkonfigurieren]

PHP-Erweiterungen [→ 2.3 Playground mit Blueprints vorkonfigurieren]

PHP-Version angeben [→ 2.3 Playground mit Blueprints vorkonfigurieren]

Plugin aktivieren [→ 2.3 Playground mit Blueprints vorkonfigurieren]

Plugin installieren [→ 2.3 Playground mit Blueprints vorkonfigurieren]

Plugins vorinstallieren [→ 2.3 Playground mit Blueprints vorkonfigurieren]

Schritte definieren [→ 2.3 Playground mit Blueprints vorkonfigurieren]

Theme aktivieren [→ 2.3 Playground mit Blueprints vorkonfigurieren]

Theme installieren [→ 2.3 Playground mit Blueprints vorkonfigurieren]

Website-URL ändern [→ 2.3 Playground mit Blueprints vorkonfigurieren]

WordPress-Version angeben [→ 2.3 Playground mit Blueprints vorkonfigurieren]

WP-CLI installieren [→ 2.3 Playground mit Blueprints vorkonfigurieren]

Plugin [→ A Glossar]

Plugin aktualisieren [→ 11.5 Plugin aktualisieren]

Plugin installieren [→ 5.1 Das Backend kennenlernen]

Plugin veröffentlichen [→ 24.10 Plugin veröffentlichen]

kostenlos [→ 24.10 Plugin veröffentlichen]

kostenpflichtig [→ 24.10 Plugin veröffentlichen]

WordPress.org-Richtlinien [→ 24.10 Plugin veröffentlichen]

Plugin-Auswahl [→ 11.3 Gute Plugins erkennen] [→ 11.3 Gute Plugins erkennen]

Plugin-Datei-Editor deaktivieren [→ 5.3 Einstellungstipps für Fortgeschrittene]

Plugin-Dateien bearbeiten [→ 5.1 Das Backend kennenlernen]

Plugin-Entwickler [→ 11.1 Plugins testen – sicher]

Plugin-Entwicklung

Activation-Hook [→ 24.3 Inhaltstyp erstellen]

Admin-Menü-Unterpunkt erstellen [→ 24.11 Weitere nützliche Funktionen für Plugins]

Alle registrierten Blöcke abfragen [→ 24.5 Editor-Blöcke erstellen]

Auf Einzelansicht prüfen [→ 24.6 Eine Checkbox zur Hervorhebung eines Vortrags ins Veröffentlichungs-Panel einfügen]

Beitragsbild-Option deaktivieren [→ 24.11 Weitere nützliche Funktionen für Plugins]

Berechtigungen prüfen [→ 24.11 Weitere nützliche Funktionen für Plugins]

Bereich auf Einstellungsseite erstellen [→ 24.11 Weitere nützliche Funktionen für Plugins]

Bereiche einer Einstellungsseite ausgeben [→ 24.11 Weitere nützliche Funktionen für Plugins]

Beschriftungen für den Inhaltstyp definieren [→ 24.3 Inhaltstyp erstellen]

Bestehenden REST-Endpunkt verändern [→ 24.11 Weitere nützliche Funktionen für Plugins]

Bestimmte Blöcke deaktivieren [→ 24.11 Weitere nützliche Funktionen für Plugins]

Block automatisch relativ zu einem anderen Block einfügen [→ 24.11 Weitere nützliche Funktionen für Plugins]

Block erstellen [→ 24.5 Editor-Blöcke erstellen]

Block nur als Kind eines anderen Blocks erlauben [→ 24.11 Weitere nützliche Funktionen für Plugins]

Block registrieren [→ 24.4 Panel in der Editor-Seitenleiste zur Bearbeitung der Metadaten] [→ 24.5 Editor-Blöcke erstellen] [→ 24.5 Editor-Blöcke erstellen]

block.json-Möglichkeiten im Detail [→ 24.11 Weitere nützliche Funktionen für Plugins]

Block-Editor-Änderungen überwachen [→ 24.11 Weitere nützliche Funktionen für Plugins]

Block-Editor-Kontext prüfen [→ 24.5 Editor-Blöcke erstellen]

Block-Editor-Kontext testen [→ 24.4 Panel in der Editor-Seitenleiste zur Bearbeitung der Metadaten]

Block-Editor-Plugin registrieren [→ 24.4 Panel in der Editor-Seitenleiste zur Bearbeitung der Metadaten] [→ 24.11 Weitere nützliche Funktionen für Plugins]

Block-Editor-Script angeben [→ 24.5 Editor-Blöcke erstellen] [→ 24.11 Weitere nützliche Funktionen für Plugins]

Block-Editor-Sidebar erstellen [→ 24.11 Weitere nützliche Funktionen für Plugins]

Block-Editor-Stylesheet laden [→ 24.11 Weitere nützliche Funktionen für Plugins]

Block-Frontend-Script laden [→ 24.11 Weitere nützliche Funktionen für Plugins]

Block-Frontend-Style laden [→ 24.11 Weitere nützliche Funktionen für Plugins]

Block-Icon festlegen [→ 24.11 Weitere nützliche Funktionen für Plugins]

Block-Kategorie festlegen [→ 24.11 Weitere nützliche Funktionen für Plugins]

Block-Konfiguration in der block.json [→ 24.5 Editor-Blöcke erstellen] [→ 24.5 Editor-Blöcke erstellen]

Block-Render-Callback [→ 24.5 Editor-Blöcke erstellen]

Block-Render-Funktion über block.json angeben [→ 24.11 Weitere nützliche Funktionen für Plugins]

Block-Script erstellen [→ 24.5 Editor-Blöcke erstellen] [→ 24.5 Editor-Blöcke erstellen]

Block-Script für Frontend und Backend laden [→ 24.11 Weitere nützliche Funktionen für Plugins]

Block-Seitenleiste erstellen [→ 24.5 Editor-Blöcke erstellen]

Block-Style für Frontend und Backend [→ 24.11 Weitere nützliche Funktionen für Plugins]

Block-Template erstellen [→ 24.7 Block-Template für den Inhaltstyp erstellen]

Block-Variationen erstellen [→ 24.11 Weitere nützliche Funktionen für Plugins]

Block-Wrapper-Attribute ausgeben [→ 24.5 Editor-Blöcke erstellen] [→ 24.5 Editor-Blöcke erstellen]

body-Klassen verändern [→ 24.6 Eine Checkbox zur Hervorhebung eines Vortrags ins Veröffentlichungs-Panel einfügen]

Coding-Standards prüfen [→ 24.4 Panel in der Editor-Seitenleiste zur Bearbeitung der Metadaten]

Datei einbinden [→ 24.3 Inhaltstyp erstellen]

Daten aus PHP im Block-Editor verfügbar machen [→ 24.5 Editor-Blöcke erstellen]

Deactivation-Hook [→ 24.3 Inhaltstyp erstellen]

Diskussionseinstellungen im Block-Editor entfernen [→ 24.11 Weitere nützliche Funktionen für Plugins]

Dokument-Sidebar-Optionen deaktivieren [→ 24.11 Weitere nützliche Funktionen für Plugins]

Dynamische statt statische Blöcke [→ 24.5 Editor-Blöcke erstellen]

Edit-Funktion für Block erstellen [→ 24.5 Editor-Blöcke erstellen]

Editor-Script einbinden [→ 24.4 Panel in der Editor-Seitenleiste zur Bearbeitung der Metadaten]

Editor-Seitenleiste erweitern [→ 24.4 Panel in der Editor-Seitenleiste zur Bearbeitung der Metadaten]

Eigene Einstellungsseite erstellen [→ 24.11 Weitere nützliche Funktionen für Plugins]

Eigenen REST-Endpunkt erstellen [→ 24.11 Weitere nützliche Funktionen für Plugins]

Einstellung im Backend einfügen [→ 24.11 Weitere nützliche Funktionen für Plugins]

Einstellungsfeld erstellen [→ 24.11 Weitere nützliche Funktionen für Plugins]

Einstellungsseite erweitern [→ 24.11 Weitere nützliche Funktionen für Plugins]

Erlaubte Kindblöcke festlegen [→ 24.11 Weitere nützliche Funktionen für Plugins]

Felder im Block-Editor erstellen [→ 24.4 Panel in der Editor-Seitenleiste zur Bearbeitung der Metadaten]

Filter erstellen [→ 24.7 Block-Template für den Inhaltstyp erstellen]

Frontend-Ausgabe eines Blocks erstellen [→ 24.5 Editor-Blöcke erstellen] [→ 24.5 Editor-Blöcke erstellen]

Grundgerüst [→ 24.2 Das Grundgerüst des Plugins]

Inhalt im Veröffentlichungs-Panel des Block-Editors einfügen [→ 24.6 Eine Checkbox zur Hervorhebung eines Vortrags ins Veröffentlichungs-Panel einfügen]

Inhalt löschen [→ 24.9 Inhalte beim Deinstallieren löschen]

Inhalte bei Deinstallation löschen [→ 24.9 Inhalte beim Deinstallieren löschen]

Inhaltstyp erstellen [→ 24.3 Inhaltstyp erstellen]

Inhaltstyp im Block-Editor prüfen [→ 24.4 Panel in der Editor-Seitenleiste zur Bearbeitung der Metadaten] [→ 24.6 Eine Checkbox zur Hervorhebung eines Vortrags ins Veröffentlichungs-Panel einfügen] [→ 24.11 Weitere nützliche Funktionen für Plugins]

Inhaltstyp in PHP überprüfen [→ 24.5 Editor-Blöcke erstellen]

Inhaltstyp konfigurieren [→ 24.3 Inhaltstyp erstellen]

Interactivity API nutzen [→ 24.11 Weitere nützliche Funktionen für Plugins]

JavaScript-Übersetzungsfunktion [→ 24.4 Panel in der Editor-Seitenleiste zur Bearbeitung der Metadaten]

Konfiguration von Blöcken anpassen [→ 24.11 Weitere nützliche Funktionen für Plugins]

Markup für Einstellungsfeld [→ 24.11 Weitere nützliche Funktionen für Plugins]

Mindestanforderungen [→ 24.2 Das Grundgerüst des Plugins]

Namespace angeben [→ 24.3 Inhaltstyp erstellen]

Optionen registrieren [→ 24.11 Weitere nützliche Funktionen für Plugins]

Optionsseite erstellen [→ 24.11 Weitere nützliche Funktionen für Plugins]

Permalink ermitteln [→ 24.11 Weitere nützliche Funktionen für Plugins]

Permalink-Struktur aktualisieren [→ 24.3 Inhaltstyp erstellen]

Plugin veröffentlichen [→ 24.10 Plugin veröffentlichen]

Plugin-Metadaten [→ 24.2 Das Grundgerüst des Plugins]

Postdaten auslesen [→ 24.11 Weitere nützliche Funktionen für Plugins]

post-Klassen verändern [→ 24.6 Eine Checkbox zur Hervorhebung eines Vortrags ins Veröffentlichungs-Panel

einfügen]

Post-Meta im Block-Editor verändern [→ 24.4 Panel in der Editor-Seitenleiste zur Bearbeitung der Metadaten] [→ 24.6 Eine Checkbox zur Hervorhebung eines Vortrags ins Veröffentlichungs-Panel einfügen] [→ 24.11 Weitere nützliche Funktionen für Plugins]

Post-Meta in PHP ermitteln [→ 24.5 Editor-Blöcke erstellen] [→ 24.6 Eine Checkbox zur Hervorhebung eines Vortrags ins Veröffentlichungs-Panel einfügen] [→ 24.11 Weitere nützliche Funktionen für Plugins]

Post-Meta mit Blöcken verbinden [→ 24.11 Weitere nützliche Funktionen für Plugins]

Post-Meta registrieren [→ 24.3 Inhaltstyp erstellen] [→ 24.6 Eine Checkbox zur Hervorhebung eines Vortrags ins Veröffentlichungs-Panel einfügen]

REST-API erweitern [→ 24.11 Weitere nützliche Funktionen für Plugins]

REST-Feld nur für bestimmten Parameter ausgeben [→ 24.11 Weitere nützliche Funktionen für Plugins]

Sanitization [→ 24.11 Weitere nützliche Funktionen für Plugins]

Script für SlotFill erstellen [→ 24.6 Eine Checkbox zur Hervorhebung eines Vortrags ins Veröffentlichungs-Panel einfügen]

Scripte nur in bestimmten Editor-Kontexten laden [→ 24.4 Panel in der Editor-Seitenleiste zur Bearbeitung der Metadaten]

Seitenleisten-Panel erstellen [→ 24.4 Panel in der Editor-Seitenleiste zur Bearbeitung der Metadaten]

selbstschließender Shortcode [→ 24.11 Weitere nützliche Funktionen für Plugins]

Shortcode erstellen [→ 24.11 Weitere nützliche Funktionen für Plugins]

Shortcode mit Start- und End-Tag [→ 24.11 Weitere nützliche Funktionen für Plugins]

Shortcode registrieren [→ 24.11 Weitere nützliche Funktionen für Plugins]

Sicherheit [→ 24.5 Editor-Blöcke erstellen]

Standard-Shortcode-Attributwerte setzen [→ 24.11 Weitere nützliche Funktionen für Plugins]

Taxonomie erstellen [→ 24.11 Weitere nützliche Funktionen für Plugins]

Taxonomie in REST-API sichtbar machen [→ 24.11 Weitere nützliche Funktionen für Plugins]

Text-Domain laden [→ 24.8 Übersetzung]

Titel ermitteln [→ 24.11 Weitere nützliche Funktionen für Plugins]

Übersetzung [→ 24.8 Übersetzung]

Übersetzungen für JavaScript-Dateien laden [→ 24.8 Übersetzung]

Übersetzungen laden [→ 22.2 WordPress-Konzepte]

Uninstall-Hook [→ 24.9 Inhalte beim Deinstallieren löschen]

Verfügbare Blöcke filtern [→ 24.5 Editor-Blöcke erstellen]

Vorbereitung auf Entwicklung für den Block-Editor [→ 24.4 Panel in der Editor-Seitenleiste zur Bearbeitung der Metadaten]

Wert einer Option ermitteln [→ 24.11 Weitere nützliche Funktionen für Plugins]

Plugin-Installation [→ 11.4 Plugins installieren]

Plugin-Marktplätze [→ 11.3 Gute Plugins erkennen]

Plugin-Programmierung [→ 1.1 Das richtige Handbuch für Sie]

Plugins [→ 1.2 Eine kurze Geschichte von WordPress] [→ 5.1 Das Backend kennenlernen] [→ 11.1 Plugins testen – sicher] [→ 11.3 Gute Plugins erkennen] [→ 12.1 Die Vor- und Nachteile von WooCommerce]

Aged Content Message [→ 13.4 Funktionen und Plugins]

Antispam Bee [→ 8.4 Spamschutz mit Antispam Bee]

Autoptimize [→ 19.3 Optimierungs-Plugin »Autoptimize«]

Avatar Privacy [→ 5.2 Einstellungen] [→ 13.4 Funktionen und Plugins]

BackWPup [→ 16.3 Backups planen und durchführen] [→ 17.2 Eine Website migrieren]

Better Search Replace [→ 17.2 Eine Website migrieren]

BuddyPress [→ 14.3 Funktionalitäten und Plugins]

Cachify [→ 19.4 Einfaches Caching-Plugin ohne Schnickschnack – »Cachify«]

Composite Products for WooCommerce [→ 12.4 Zusätzliche Erweiterungen für Ihren Shop]

Disable REST API [→ 18.3 WordPress abhärten]

Duplicator [→ 17.2 Eine Website migrieren]

Embed Privacy [→ 13.4 Funktionen und Plugins]

Enable Mastodon Apps [→ 15.2 Das Fediverse]

Form Block [→ 14.3 Funktionalitäten und Plugins]

Friends [→ 15.2 Das Fediverse]

Germanized für WooCommerce [→ 12.2 WooCommerce installieren und einrichten]

Imagify [→ 19.6 Bilder optimieren]

IndieAuth [→ 15.1 Das IndieWeb]

IndieBlocks [→ 15.1 Das IndieWeb]

IndieWeb [→ 15.1 Das IndieWeb]

Mailjet [→ 14.3 Funktionalitäten und Plugins]

Micropub [→ 15.1 Das IndieWeb]

NodeInfo [→ 15.2 Das Fediverse]

Password Protected [→ 17.1 Testen und entwickeln mit Entwicklungs-, Staging- und Live-Umgebungen]

PayPal Payments for WooCommerce [→ 12.4 Zusätzliche Erweiterungen für Ihren Shop]

Product Bundles for WooCommerce [→ 12.4 Zusätzliche Erweiterungen für Ihren Shop]

Product Vendors for WooCommerce [→ 12.4 Zusätzliche Erweiterungen für Ihren Shop]

Protect Login [→ 18.1 Login-Sicherheit]

Public Post Review [→ 13.4 Funktionen und Plugins]

Query Monitor [→ 22.3 Entwicklungs- und Testumgebung einrichten] [→ 23.2 Classic Theme anpassen und erstellen]

Real Cookie Banner [→ 13.4 Funktionen und Plugins]

Regenerate Thumbnails [→ 23.4 Weitere nützliche Funktionen für Themes]

Safe Redirect Manager [→ 6.1 Beiträge schreiben/Seiten anlegen]

Slim Maintenance Mode [→ 16.2 Wartungsmodus aktivieren – Plugin »Slim Maintenance Mode«]

Statify [→ 13.4 Funktionen und Plugins]

The SEO Framework [→ 20.1 Schlank mit allen wichtigen Funktionen – Plugin »The SEO Framework«]

User Role Editor [→ 14.3 Funktionalitäten und Plugins]

WebFinger [→ 15.2 Das Fediverse]

Webmention [→ 15.1 Das IndieWeb]

WooCommerce [→ 12.2 WooCommerce installieren und einrichten]

WooCommerce Bookings [→ 12.4 Zusätzliche Erweiterungen für Ihren Shop]

WooCommerce Subscriptions [→ 12.4 Zusätzliche Erweiterungen für Ihren Shop]

WordPress Beta Tester [→ 16.1 WordPress, Plugins und Themes aktualisieren]

Worthy [→ 13.4 Funktionen und Plugins]

WP Downgrade [→ 16.1 WordPress, Plugins und Themes aktualisieren]

WP Rocket [→ 19.5 Profi-Cache mit WP Rocket]

WP Staging [→ 17.2 Eine Website migrieren]

Yoast SEO [→ 20.2 Viele SEO-Maßnahmen in einem Plugin mit »Yoast SEO«]

Plugins aktualisieren [→ 11.5 Plugin aktualisieren]

Plugins deaktivieren [→ 11.6 Plugins deaktivieren und löschen]

Plugins löschen [→ 11.6 Plugins deaktivieren und löschen]

plugins-Ordner verschieben [→ 5.3 Einstellungstipps für Fortgeschrittene]

Plugin-Übersicht [→ 11.5 Plugin aktualisieren]

Plugin-Update [→ 5.1 Das Backend kennenlernen]

Plugin-Updates [→ 16.1 WordPress, Plugins und Themes aktualisieren]

Plugin-Verwaltung [→ 9.3 Bedienung des Netzwerks und Unterschiede zum Standard]

Plugin-Verzeichnis [→ 11.3 Gute Plugins erkennen]

Poedit [→ 22.2 WordPress-Konzepte]

Polyfills [→ 19.2 PageSpeed Insights – Missstände verstehen und bei Bedarf abarbeiten]

POSSE [→ 15.1 Das IndieWeb] [→ 15.1 Das IndieWeb]

Postfächer [→ 4.1 Welches Hosting darf's denn sein?]

Post-Meta [→ 6.4 Individuelle Felder nutzen] [→ A Glossar]

registrieren [→ 24.3 Inhaltstyp erstellen]

Post-Type → siehe [Inhaltstyp]

POT-Datei erstellen [→ 24.8 Übersetzung]

Private Nachrichten [→ 14.3 Funktionalitäten und Plugins]

Produktbeschreibung [→ 12.2 WooCommerce installieren und einrichten]

Produktbild [→ 12.2 WooCommerce installieren und einrichten]

Produktdaten [→ 12.2 WooCommerce installieren und einrichten]

Produkt-Editor [→ 12.2 WooCommerce installieren und einrichten]

Produkteigenschaften [→ 12.2 WooCommerce installieren und einrichten]

Produkttypen [→ 12.2 WooCommerce installieren und einrichten]

Produktübersicht [→ 12.2 WooCommerce installieren und einrichten]

Profile [→ 14.3 Funktionalitäten und Plugins]

Programmierhilfen finden [→ 22.3 Entwicklungs- und Testumgebung einrichten]

Programmierung [→ 1.1 Das richtige Handbuch für Sie]

Programmschnittstellen [→ 12.1 Die Vor- und Nachteile von WooCommerce]

Protokoll [→ 11.5 Plugin aktualisieren]

Provisionen [→ 12.1 Die Vor- und Nachteile von WooCommerce]

Publikum [→ 13.5 Blogging-Tipps]

Q ⇑

Qualitätskontrolle [→ 11.3 Gute Plugins erkennen]

QuickEdit [→ 6.2 Beiträge/Seiten verwalten] [→ A Glossar]

R ⇑

Rankingfaktor [→ 19.1 Grundlagen zur Optimierung]

Raster [→ 10.3 Block-Theme anpassen: den Website-Editor kennenlernen]

React [→ A Glossar]

Rechenleistung [→ 4.1 Welches Hosting darf's denn sein?]

Rechnung [→ 12.2 WooCommerce installieren und einrichten]

Rechtliche Aspekte nicht vergessen [→ 6.9 Impressum und Datenschutzerklärung nicht vergessen]

Rechtssicherheit [→ 12.2 WooCommerce installieren und einrichten]

Redaktionsplan [→ 13.5 Blogging-Tipps]

reddit [→ 15.1 Das IndieWeb]

Release-Candidate [→ 16.1 WordPress, Plugins und Themes aktualisieren]

Rendering-blockierende Ressourcen [→ 19.2 PageSpeed Insights – Missstände verstehen und bei Bedarf abarbeiten]

Reposts [→ 15.1 Das IndieWeb]

Requests [→ 19.1 Grundlagen zur Optimierung]

Responsive Design [→ 10.4 Classic Theme anpassen]

Responsive Images [→ 19.2 PageSpeed Insights – Missstände verstehen und bei Bedarf abarbeiten] [→ 19.6 Bilder

optimieren]

REST-API [→ 1.2 Eine kurze Geschichte von WordPress] [→ 7.4 Benutzerprofil bearbeiten] [→ 18.3 WordPress abhärten] [→ 18.3 WordPress abhärten] [→ 24.3 Inhaltstyp erstellen] [→ 24.11 Weitere nützliche Funktionen für Plugins] [→ A Glossar]

REST-Endpunkt eines Inhaltstyps aufrufen [→ 24.3 Inhaltstyp erstellen]

Revision [→ A Glossar]

wiederherstellen [→ 6.3 Mit Revisionen arbeiten]

Revisionen [→ 5.3 Einstellungstipps für Fortgeschrittene] [→ 6.1 Beiträge schreiben/Seiten anlegen] [→ 6.3 Mit Revisionen arbeiten] [→ 10.3 Block-Theme anpassen: den Website-Editor kennenlernen]

Anzahl einstellen [→ 5.3 Einstellungstipps für Fortgeschrittene]

RewriteRule [→ 18.2 Die Wichtigkeit von SSL/TLS]

Rewrite-Rules [→ A Glossar]

robots.txt [→ A Glossar]

Robots-Meta-Tag [→ 20.1 Schlank mit allen wichtigen Funktionen – Plugin »The SEO Framework«] [→ A Glossar]

Rollen [→ 7.1 Rollen verstehen]

Abonnent [→ 7.1 Rollen verstehen]

Administrator [→ 7.1 Rollen verstehen]

Autor [→ 7.1 Rollen verstehen]

Redakteur [→ 7.1 Rollen verstehen]

Super-Administrator [→ 7.1 Rollen verstehen]

Root Server [→ 4.1 Welches Hosting darf's denn sein?]

RSS [→ 15.1 Das IndieWeb] [→ 15.2 Das Fediverse]

RSS-Feeds [→ 5.2 Einstellungen] [→ 5.2 Einstellungen]

S ⇑

SaaS [→ 9.1 Was ist WordPress Multisite?] [→ 12.1 Die Vor- und Nachteile von WooCommerce]

Same-Day-Delivery [→ 12.2 WooCommerce installieren und einrichten]

Schadcode [→ 9.1 Was ist WordPress Multisite?] [→ 11.1 Plugins testen – sicher]

ausfindig machen [→ 21.1 Gehackt, was nun?]

Schatten [→ 10.3 Block-Theme anpassen: den Website-Editor kennenlernen]

Scheckzahlung [→ 12.2 WooCommerce installieren und einrichten]

Schema.org [→ 15.1 Das IndieWeb] [→ 20.1 Schlank mit allen wichtigen Funktionen – Plugin »The SEO Framework«] [→ A Glossar]

Schlagwörter [→ 5.1 Das Backend kennenlernen]

verwalten [→ 6.2 Beiträge/Seiten verwalten]

zuweisen [→ 6.1 Beiträge schreiben/Seiten anlegen]

Schnittstellen [→ 11.1 Plugins testen – sicher]

Schriften [→ 19.5 Profi-Cache mit WP Rocket]

Schriften → siehe [Typografie]

Schriften-Bibliothek [→ 10.3 Block-Theme anpassen: den Website-Editor kennenlernen]

Schriftgröße [→ 19.2 PageSpeed Insights – Missstände verstehen und bei Bedarf abarbeiten]

Scripts-API [→ 19.2 PageSpeed Insights – Missstände verstehen und bei Bedarf abarbeiten]

Secure Sockets Layer → siehe [SSL]

Security by Obscurity [→ 18.4 No-Gos der WordPress-Sicherheit] [→ A Glossar]

Seite erstellen → siehe [Inhalte erstellen]

Seite speichern [→ 6.1 Beiträge schreiben/Seiten anlegen]

Seiten [→ 5.1 Das Backend kennenlernen]

verwalten [→ 6.2 Beiträge/Seiten verwalten]

Seitenleiste → siehe [Sidebar]

Seitenstruktur [→ 14.2 Design, Struktur und Navigation]

Seiten-Template ändern [→ 6.1 Beiträge schreiben/Seiten anlegen]

Semantic Versioning [→ 22.1 Coding-Standards] [→ 24.4 Panel in der Editor-Seitenleiste zur Bearbeitung der Metadaten] [→ A Glossar]

SEO [→ 1.1 Das richtige Handbuch für Sie] [→ 13.5 Blogging-Tipps] [→ 19.2 PageSpeed Insights – Missstände verstehen und bei Bedarf abarbeiten] [→ 19.2 PageSpeed Insights –

Missstände verstehen und bei Bedarf abarbeiten] [→ 20.1 Schlank mit allen wichtigen Funktionen – Plugin »The SEO Framework«] [→ A Glossar]

Server [→ 4.1 Welches Hosting darf's denn sein?] [→ 9.2 Die Multisite-Einrichtung] [→ 11.5 Plugin aktualisieren] [→ 18.1 Login-Sicherheit] [→ 18.2 Die Wichtigkeit von SSL/TLS]

Server-Adresse [→ 9.2 Die Multisite-Einrichtung]

Serverfehler [→ 21.2 Die häufigsten Standardproblemlösungen]

Serverpfad [→ 18.3 WordPress abhärten]

Server-Voraussetzungen [→ 1.3 Warum WordPress?]

Settings API [→ A Glossar]

Shared Webhosting [→ 4.1 Welches Hosting darf's denn sein?] [→ 4.1 Welches Hosting darf's denn sein?]

Shared-Hosting [→ 15.2 Das Fediverse]

Shop-Details [→ 12.2 WooCommerce installieren und einrichten]

Shortcode [→ 6.1 Beiträge schreiben/Seiten anlegen] [→ 24.11 Weitere nützliche Funktionen für Plugins] [→ A Glossar]

Sicherheit [→ 18.1 Login-Sicherheit]

Sicherheitskopie → siehe [Backup]

Sicherheitslücke [→ 9.1 Was ist WordPress Multisite?] [→ 16.1 WordPress, Plugins und Themes aktualisieren] [→ 18.1 Login-Sicherheit] [→ 18.1 Login-Sicherheit]

Sicherheitsmaßnahmen [→ 18.1 Login-Sicherheit]

Sicherheitsprobleme [→ 4.1 Welches Hosting darf’s denn sein?]

Sidebar [→ 10.3 Block-Theme anpassen: den Website-Editor kennenlernen] [→ 10.3 Block-Theme anpassen: den Website-Editor kennenlernen]

Site-Editor → siehe [Website-Editor]

Site-ID [→ 9.3 Bedienung des Netzwerks und Unterschiede zum Standard]

Sitemap

- *bei Bing einreichen* [→ 20.3 Sitemap bei Google und Bing einreichen]
- *bei Google einreichen* [→ 20.3 Sitemap bei Google und Bing einreichen]
- *zur Inhaltsplanung* [→ 6.7 Inhalte planen]

siteurl [→ 9.3 Bedienung des Netzwerks und Unterschiede zum Standard]

Skalierbarkeit [→ 12.1 Die Vor- und Nachteile von WooCommerce]

Slider [→ 19.2 PageSpeed Insights – Missstände verstehen und bei Bedarf abarbeiten]

SlotFill [→ 24.6 Eine Checkbox zur Hervorhebung eines Vortrags ins Veröffentlichungs-Panel einfügen] [→ 24.11 Weitere nützliche Funktionen für Plugins]

Smartphones [→ 14.2 Design, Struktur und Navigation]

SMS [→ 18.1 Login-Sicherheit]

Social Media [→ 8.5 Webmentions] [→ 13.1 Zielgruppe und Plan] [→ 13.3 Struktur und Navigation] [→ 13.5 Blogging-Tipps]

Spam [→ 1.1 Das richtige Handbuch für Sie] [→ 8.1 Kommentar-Einstellungen] [→ 8.2 Kommentare freigeben] [→ 9.3 Bedienung des Netzwerks und Unterschiede zum Standard] [→ 13.4 Funktionen und Plugins]

Speicherplatz [→ 9.3 Bedienung des Netzwerks und Unterschiede zum Standard]

Sponsoring [→ 13.1 Zielgruppe und Plan]

Spotify [→ 13.4 Funktionen und Plugins]

Sprache [→ 5.2 Einstellungen] [→ 7.4 Benutzerprofil bearbeiten]

Sprunglink [→ 23.2 Classic Theme anpassen und erstellen]

SQL-Injection-Angriffe [→ 18.4 No-Gos der WordPress-Sicherheit]

SQLite [→ 2.2 wp-now: Playground für Entwickler*innen]

Squarespace [→ 1.3 Warum WordPress?]

SSH-Zugang [→ 17.2 Eine Website migrieren]

SSL [→ 4.1 Welches Hosting darf’s denn sein?] [→ 4.1 Welches Hosting darf’s denn sein?] [→ 4.2 Installation mit einem One-Click-Installer] [→ 9.2 Die Multisite-Einrichtung] [→ 18.2 Die Wichtigkeit von SSL/TLS] [→ 18.2 Die Wichtigkeit von SSL/TLS] [→ 18.2 Die Wichtigkeit von SSL/TLS]

SSL erzwingen [→ 18.2 Die Wichtigkeit von SSL/TLS]

SSL im Backend erzwingen [→ 5.3 Einstellungstipps für Fortgeschrittene]

Staging [→ 11.1 Plugins testen – sicher] [→ 16.1 WordPress, Plugins und Themes aktualisieren] [→ A Glossar]

Staging-Umgebung [→ 17.1 Testen und entwickeln mit Entwicklungs-, Staging- und Live-Umgebungen]

Standardproblemlösungen [→ 21.2 Die häufigsten Standardproblemlösungen]

- *404-Fehler – Seite nicht gefunden* [→ 21.2 Die häufigsten Standardproblemlösungen]
- *Allowed Memory exhausted* [→ 21.2 Die häufigsten Standardproblemlösungen]
- *Änderungen nicht sichtbar* [→ 21.2 Die häufigsten Standardproblemlösungen]
- *Datenbankproblem* [→ 21.2 Die häufigsten Standardproblemlösungen]
- *kein Zugriff auf die Website* [→ 21.2 Die häufigsten Standardproblemlösungen]
- *keine Suchmaschinen-Indexierung* [→ 21.2 Die häufigsten Standardproblemlösungen]
- *Passwort wiederherstellen* [→ 21.2 Die häufigsten Standardproblemlösungen]
- *Serverfehler* [→ 21.2 Die häufigsten Standardproblemlösungen]
- *unsichere Verbindung* [→ 21.2 Die häufigsten Standardproblemlösungen]

Standardsprache [→ 9.3 Bedienung des Netzwerks und Unterschiede zum Standard]

Standard-Theme [→ 13.2 Design und Layout]

State [→ 24.11 Weitere nützliche Funktionen für Plugins]

Steuersätze [→ 12.2 WooCommerce installieren und einrichten]

Stilbuch [→ 10.3 Block-Theme anpassen: den Website-Editor kennenlernen] [→ 23.1 Block-Theme erstellen]

Stile [→ 10.3 Block-Theme anpassen: den Website-Editor kennenlernen]

Stile-Seitenleiste [→ 10.3 Block-Theme anpassen: den Website-Editor kennenlernen] [→ 23.1 Block-Theme erstellen]

Stile-Sidebar [→ 10.3 Block-Theme anpassen: den Website-Editor kennenlernen]

Storefront [→ 12.3 Letzter Schliff und Testkauf]

Stromverbrauch [→ 19.7 Optimieren für den Klimaschutz]

style.css [→ 23.1 Block-Theme erstellen]

style.css Child-Theme [→ 23.2 Classic Theme anpassen und erstellen]

style.css einbinden [→ 23.2 Classic Theme anpassen und erstellen]

Stylesheet einbinden [→ 23.1 Block-Theme erstellen]

Subdomain [→ 9.2 Die Multisite-Einrichtung] [→ 9.2 Die Multisite-Einrichtung] [→ 9.3 Bedienung des Netzwerks und Unterschiede zum Standard]

Suchmaschinenoptimierung → siehe [SEO]

Suchmaschinen-Sichtbarkeit [→ 5.2 Einstellungen]

Super-Administrator [→ 9.3 Bedienung des Netzwerks und Unterschiede zum Standard]

Support [→ 11.3 Gute Plugins erkennen]

Support-Forum [→ 1.5 Nach der letzten Seite] [→ 11.3 Gute Plugins erkennen] [→ 11.3 Gute Plugins erkennen]

SVG [→ 9.3 Bedienung des Netzwerks und Unterschiede zum Standard]

Synchronisierte Vorlage [→ 10.3 Block-Theme anpassen: den Website-Editor kennenlernen]

Syntaxhervorhebung [→ 7.4 Benutzerprofil bearbeiten]

T ⇑

Tabelle [→ 10.3 Block-Theme anpassen: den Website-Editor kennenlernen]

Tags → siehe [Schlagwörter]

Tastaturkürzel [→ 8.2 Kommentare freigeben]

Taxonomie [→ A Glossar]

Template [→ 10.3 Block-Theme anpassen: den Website-Editor kennenlernen] [→ 10.3 Block-Theme anpassen: den Website-Editor kennenlernen] [→ A Glossar]

bearbeiten [→ 10.3 Block-Theme anpassen: den Website-Editor kennenlernen]

hinzufügen [→ 10.3 Block-Theme anpassen: den Website-Editor kennenlernen]

Template-Blöcke kopieren [→ 23.1 Block-Theme erstellen]

Template-Datei überschreiben [→ 23.2 Classic Theme anpassen und erstellen]

Template-Dateien identifizieren [→ 23.2 Classic Theme anpassen und erstellen]

Template-Editor [→ 10.3 Block-Theme anpassen: den Website-Editor kennenlernen] [→ 10.3 Block-Theme anpassen: den Website-Editor kennenlernen]

Template-Hierarchie [→ 23.2 Classic Theme anpassen und erstellen] [→ A Glossar]

Archiv-Seiten [→ 23.2 Classic Theme anpassen und erstellen]

Einzelansicht von Seiten und Beiträgen [→ 23.2 Classic Theme anpassen und erstellen]

Sonderseiten [→ 23.2 Classic Theme anpassen und erstellen]

Startseiten-Templates [→ 23.2 Classic Theme anpassen und erstellen]

Suchseite [→ 23.2 Classic Theme anpassen und erstellen]

Template-Tags [→ 22.2 WordPress-Konzepte]

Template-Teil [→ 10.3 Block-Theme anpassen: den Website-Editor kennenlernen] [→ 10.3 Block-Theme anpassen: den Website-Editor kennenlernen] [→ 10.3 Block-Theme anpassen: den Website-Editor kennenlernen] [→ 10.3 Block-Theme anpassen: den Website-Editor kennenlernen]

angeben (theme.json) [→ 23.1 Block-Theme erstellen]

bearbeiten [→ 10.3 Block-Theme anpassen: den Website-Editor kennenlernen]

einfügen [→ 23.1 Block-Theme erstellen]

ersetzen [→ 23.1 Block-Theme erstellen]

erstellen [→ 23.1 Block-Theme erstellen]

hinzufügen [→ 10.3 Block-Theme anpassen: den Website-Editor kennenlernen]

loslösen [→ 23.1 Block-Theme erstellen]

Term [→ A Glossar]

Terminal [→ 17.2 Eine Website migrieren]

Testen [→ 9.2 Die Multisite-Einrichtung]

Textauszug [→ 6.1 Beiträge schreiben/Seiten anlegen] [→ 10.4 Classic Theme anpassen]

Text-Domain [→ 22.2 WordPress-Konzepte]

The SEO Framework [→ 20.1 Schlank mit allen wichtigen Funktionen – Plugin »The SEO Framework«]

allgemeine Einstellungen [→ 20.1 Schlank mit allen wichtigen Funktionen – Plugin »The SEO Framework«]

Description-Einstellungen [→ 20.1 Schlank mit allen wichtigen Funktionen – Plugin »The SEO Framework«]

Einstellungen im Editor [→ 20.1 Schlank mit allen wichtigen Funktionen – Plugin »The SEO Framework«]

Feed-Einstellungen [→ 20.1 Schlank mit allen wichtigen Funktionen – Plugin »The SEO Framework«]

Indexierung von Inhalten steuern [→ 20.1 Schlank mit allen wichtigen Funktionen – Plugin »The SEO Framework«]

Robots-Meta-Einstellungen [→ 20.1 Schlank mit allen wichtigen Funktionen – Plugin »The SEO Framework«]

Schema.org-Einstellungen [→ 20.1 Schlank mit allen wichtigen Funktionen – Plugin »The SEO Framework«]

Sitemap-Einstellungen [→ 20.1 Schlank mit allen wichtigen Funktionen – Plugin »The SEO Framework«]

Social-Einstellungen [→ 20.1 Schlank mit allen wichtigen Funktionen – Plugin »The SEO Framework«]

Startseiten-Einstellungen [→ 20.1 Schlank mit allen wichtigen Funktionen – Plugin »The SEO Framework«]

Titel-Einstellungen [→ 20.1 Schlank mit allen wichtigen Funktionen – Plugin »The SEO Framework«]

Webmaster-Meta-Einstellungen [→ 20.1 Schlank mit allen wichtigen Funktionen – Plugin »The SEO Framework«]

Theme [→ A Glossar]

hochladen [→ 10.5 Neues Theme suchen und installieren]

installieren [→ 5.1 Das Backend kennenlernen] [→ 10.5 Neues Theme suchen und installieren]

löschen [→ 5.1 Das Backend kennenlernen]

Vorschau [→ 5.1 Das Backend kennenlernen]

wechseln [→ 5.1 Das Backend kennenlernen]

Theme veröffentlichen [→ 23.3 Theme veröffentlichen]

im WordPress.org-Verzeichnis [→ 23.3 Theme veröffentlichen]

kostenlos [→ 23.3 Theme veröffentlichen]

kostenpflichtig [→ 23.3 Theme veröffentlichen]

theme.json [→ 23.1 Block-Theme erstellen]

Blockabstandsoptionen steuern [→ 23.1 Block-Theme erstellen]

Block-Vorlagen angeben [→ 23.1 Block-Theme erstellen]

CSS-Variablen [→ 23.1 Block-Theme erstellen]

customTemplates [→ 23.1 Block-Theme erstellen]

eigene Schriftgrößen deaktivieren [→ 23.1 Block-Theme erstellen]

Einstellungen auf Block-Ebene überschreiben [→ 23.1 Block-Theme erstellen]

Farbeinstellungen anpassen [→ 23.1 Block-Theme erstellen]

Farbpalette definieren [→ 23.1 Block-Theme erstellen]

Fluid Typography [→ 23.1 Block-Theme erstellen]

Hintergrundbild-Optionen steuern [→ 23.1 Block-Theme erstellen]

individuelle Zeilenabstände deaktivieren [→ 23.1 Block-Theme erstellen]

Layoutbreite definieren [→ 23.1 Block-Theme erstellen]

Layout-Einstellungsmöglichkeiten [→ 23.1 Block-Theme erstellen]

Lightbox-Funktion steuern [→ 23.1 Block-Theme erstellen]

Option zur Sticky-Positionierung steuern [→ 23.1 Block-Theme erstellen]

patterns [→ 23.1 Block-Theme erstellen]

Rahmen-Einstellungen ändern [→ 23.1 Block-Theme erstellen]

Schatten-Option steuern [→ 23.1 Block-Theme erstellen]

Schrift einbinden [→ 23.1 Block-Theme erstellen]

Schriftgrößen definieren [→ 23.1 Block-Theme erstellen]

Schriftstile anwenden [→ 23.1 Block-Theme erstellen]

Seitenverhältnis-Optionen ändern [→ 23.1 Block-Theme erstellen]

Standard-Schriftgrößen deaktivieren [→ 23.1 Block-Theme erstellen]

Standard-Styles setzen [→ 23.1 Block-Theme erstellen]

templateParts [→ 23.1 Block-Theme erstellen]

Templates angeben [→ 23.1 Block-Theme erstellen]

Template-Teile angeben [→ 23.1 Block-Theme erstellen]

Typografie-Optionen [→ 23.1 Block-Theme erstellen]

Zeilenhöhen definieren [→ 23.1 Block-Theme erstellen]

Theme-Auswahl [→ 12.2 WooCommerce installieren und einrichten]

Theme-Datei-Editor deaktivieren [→ 5.3 Einstellungstipps für Fortgeschrittene]

Theme-Dateien bearbeiten (Classic Theme) [→ 5.1 Das Backend kennenlernen]

Theme-Entwicklung [→ 23.1 Block-Theme erstellen]

Bildgrößen registrieren [→ 23.4 Weitere nützliche Funktionen für Themes]

Block-Stile registrieren [→ 23.4 Weitere nützliche Funktionen für Themes]

Core-Block-Stile entfernen [→ 23.4 Weitere nützliche Funktionen für Themes]

Option zum Block-Sperren ausblenden [→ 23.4 Weitere nützliche Funktionen für Themes]

Theme-Metadaten [→ 23.1 Block-Theme erstellen] [→ 23.2 Classic Theme anpassen und erstellen] [→ 23.2 Classic Theme anpassen und erstellen]

Themenideen [→ 13.1 Zielgruppe und Plan]

Theme-Programmierung [→ 1.1 Das richtige Handbuch für Sie]

Themes [→ 1.2 Eine kurze Geschichte von WordPress] [→ 9.3 Bedienung des Netzwerks und Unterschiede zum Standard] [→ 10.1 Classic Themes und Block-Themes: Was ist das?] [→ 11.1 Plugins testen – sicher] [→ 13.1 Zielgruppe und Plan]

Premium-Themes [→ 10.2 Installierte Themes vorab ansehen und aktivieren]

Standard-Themes [→ 10.2 Installierte Themes vorab ansehen und aktivieren]

Theme-Update [→ 5.1 Das Backend kennenlernen] [→ 16.1 WordPress, Plugins und Themes aktualisieren]

Theme-Verwaltung [→ 9.3 Bedienung des Netzwerks und Unterschiede zum Standard] [→ 9.3 Bedienung des Netzwerks und Unterschiede zum Standard]

Theme-Verzeichnis [→ 10.5 Neues Theme suchen und installieren]

Threads [→ 13.5 Blogging-Tipps]

TinyMCE [→ 6.8 Classic Editor statt Block-Editor nutzen]

Tipps zum Inhalte schreiben [→ 6.7 Inhalte planen]

Titel vergeben [→ 6.1 Beiträge schreiben/Seiten anlegen]

TLS → siehe [SSL]

Touch-Geräte [→ 19.2 PageSpeed Insights – Missstände verstehen und bei Bedarf abarbeiten]

Trackbacks [→ 5.2 Einstellungen] [→ 8.2 Kommentare freigeben] [→ 15.1 Das IndieWeb]

Tracking [→ 13.4 Funktionen und Plugins]

Transients bereinigen [→ 19.5 Profi-Cache mit WP Rocket]

Transport Layer Security → siehe [SSL]

Tumblr [→ 15.1 Das IndieWeb] [→ 17.2 Eine Website migrieren]

Twitter Cards [→ 15.1 Das IndieWeb]

Two-Factor Authentication → siehe [Zwei-Faktor-Authentifizierung]

Typografie [→ 10.3 Block-Theme anpassen: den Website-Editor kennenlernen] [→ 10.3 Block-Theme anpassen: den Website-Editor kennenlernen]

U ⇑

Überschreiben eines Custom Themes durch W.org-Theme verhindern [→ 23.2 Classic Theme anpassen und erstellen]

Überschriften-Hierarchie [→ 19.2 PageSpeed Insights – Missstände verstehen und bei Bedarf abarbeiten]

Übersetzung [→ 11.5 Plugin aktualisieren] [→ 22.2 WordPress-Konzepte] [→ 23.1 Block-Theme erstellen]

anlegen [→ 22.2 WordPress-Konzepte]

für JavaScript-Strings [→ 22.2 WordPress-Konzepte]

mit Poedit anlegen [→ 22.2 WordPress-Konzepte]

Übersetzung laden (Theme) [→ 23.1 Block-Theme erstellen] [→ 23.2 Classic Theme anpassen und erstellen]

Übersetzungsfunktionen [→ 22.2 WordPress-Konzepte]

Übersetzungstemplate (POT) mit WP-CLI generieren [→ 22.2 WordPress-Konzepte]

Umsatzsteuer [→ 12.2 WooCommerce installieren und einrichten]

Unbenutzte Plugins [→ 11.6 Plugins deaktivieren und löschen]

Unternehmens-Website [→ 1.1 Das richtige Handbuch für Sie]

Unterverzeichnisse [→ 9.2 Die Multisite-Einrichtung] [→ 9.2 Die Multisite-Einrichtung]

Update-Benachrichtigungen [→ 16.1 WordPress, Plugins und Themes aktualisieren]

Updates [→ 11.3 Gute Plugins erkennen] [→ 16.1 WordPress, Plugins und Themes aktualisieren]

Update-Verhalten [→ 16.1 WordPress, Plugins und Themes aktualisieren]

Upload [→ 11.5 Plugin aktualisieren]

Upload-Dialog [→ 11.4 Plugins installieren]

uploads-Ordner verschieben [→ 5.3 Einstellungstipps für Fortgeschrittene]

Upload-Verzeichnis [→ 9.1 Was ist WordPress Multisite?] [→ 9.1 Was ist WordPress Multisite?]

Upsell [→ 12.2 WooCommerce installieren und einrichten]

URL-Struktur → siehe [Permalinks]

use-Statement [→ 24.5 Editor-Blöcke erstellen]

V ⇑

Verein [→ 14.1 Aufgaben einer Vereins-Website]

Vereins-Website [→ 1.1 Das richtige Handbuch für Sie]

Verkleinern-Funktion [→ 6.1 Beiträge schreiben/Seiten anlegen]

Verlinkung [→ 13.3 Struktur und Navigation]

Vermieten [→ 12.4 Zusätzliche Erweiterungen für Ihren Shop]

Veröffentlichung per E-Mail [→ 5.2 Einstellungen]

Veröffentlichung zu einem späteren Zeitpunkt planen [→ 6.1 Beiträge schreiben/Seiten anlegen]

Versand [→ 12.2 WooCommerce installieren und einrichten] [→ 12.2 WooCommerce installieren und einrichten]

Versandart [→ 12.2 WooCommerce installieren und einrichten]

Versandklassen [→ 12.2 WooCommerce installieren und einrichten] [→ 12.2 WooCommerce installieren und

einrichten]

Versandzonen [→ 12.2 WooCommerce installieren und einrichten]

Versionierung [→ 22.1 Coding-Standards]

Verzeichnisstruktur [→ 9.3 Bedienung des Netzwerks und Unterschiede zum Standard]

VG Wort [→ 13.4 Funktionen und Plugins]

Videoformate [→ 19.2 PageSpeed Insights – Missstände verstehen und bei Bedarf abarbeiten]

Videos [→ 19.7 Optimieren für den Klimaschutz]

Videos hosten [→ 6.5 Medien organisieren]

Video-Tutorials [→ 1.5 Nach der letzten Seite]

Viewport [→ 10.3 Block-Theme anpassen: den Website-Editor kennenlernen] [→ 10.4 Classic Theme anpassen] [→ 19.6 Bilder optimieren]

Viewport-Meta-Element [→ 23.2 Classic Theme anpassen und erstellen]

Visueller Editor [→ 7.4 Benutzerprofil bearbeiten]

Vorhersehbarkeit [→ 13.1 Zielgruppe und Plan]

Vorlage [→ 5.1 Das Backend kennenlernen] [→ 6.1 Beiträge schreiben/Seiten anlegen] [→ 10.3 Block-Theme anpassen: den Website-Editor kennenlernen] [→ 10.3 Block-Theme anpassen: den Website-Editor kennenlernen] [→ 10.3 Block-Theme anpassen: den Website-Editor kennenlernen] [→ 10.3 Block-

Theme anpassen: den Website-Editor kennenlernen] [→ 23.1 Block-Theme erstellen]

erstellen [→ 6.1 Beiträge schreiben/Seiten anlegen] [→ 23.1 Block-Theme erstellen]

sperren [→ 10.3 Block-Theme anpassen: den Website-Editor kennenlernen]

Vorschau [→ 6.1 Beiträge schreiben/Seiten anlegen]

Vorteile von WordPress [→ 1.3 Warum WordPress?]

VPN [→ 18.3 WordPress abhärten]

VPS [→ 4.1 Welches Hosting darf's denn sein?]

W ⇑

WAI [→ 6.7 Inhalte planen]

Walled Gardens [→ 15.1 Das IndieWeb]

Warenkorb [→ 12.2 WooCommerce installieren und einrichten] [→ 12.2 WooCommerce installieren und einrichten] [→ 13.4 Funktionen und Plugins]

Warenwirtschaft [→ 12.1 Die Vor- und Nachteile von WooCommerce] [→ 12.2 WooCommerce installieren und einrichten]

Warenwirtschaftssysteme [→ 12.1 Die Vor- und Nachteile von WooCommerce]

Wartung [→ 1.4 Anleitung zum Handbuch]

Wartungsmodus aktivieren [→ 16.2 Wartungsmodus aktivieren – Plugin »Slim Maintenance Mode«]

WCAG → siehe [Web Content Accessibility Guidelines]

Web 2.0 [→ 15.1 Das IndieWeb]

Web Accessibility Initiative (WAI) [→ A Glossar]

Web Content Accessibility Guidelines [→ 6.7 Inhalte planen] [→ 10.6 Barrierefreiheit von Themes testen] [→ 10.6 Barrierefreiheit von Themes testen] [→ 22.1 Coding-Standards] [→ 22.1 Coding-Standards] [→ A Glossar]

WebFinger [→ 15.2 Das Fediverse] [→ 15.2 Das Fediverse] [→ 15.2 Das Fediverse]

Webfonts [→ 19.2 PageSpeed Insights – Missstände verstehen und bei Bedarf abarbeiten] [→ 23.1 Block-Theme erstellen]

Webhoster [→ 16.1 WordPress, Plugins und Themes aktualisieren] [→ A Glossar]

Weblog → siehe [Blog]

Webmaster-Tools [→ 20.1 Schlank mit allen wichtigen Funktionen – Plugin »The SEO Framework«]

Webmention [→ 8.2 Kommentare freigeben] [→ 8.5 Webmentions] [→ 15.1 Das IndieWeb] [→ 15.1 Das IndieWeb] [→ 15.1 Das IndieWeb] [→ 15.1 Das IndieWeb] [→ 15.2 Das Fediverse]

WebP [→ 6.5 Medien organisieren] [→ 19.6 Bilder optimieren] [→ 19.6 Bilder optimieren]

Webserver [→ 2.1 WordPress Playground – eine ganze WordPress-Installation lokal im Browser] [→ 19.1 Grundlagen zur Optimierung] [→ A Glossar]

Webshop → siehe [Online-Shop]

Website migrieren [→ 17.2 Eine Website migrieren]

Website-Adresse (URL) [→ 5.2 Einstellungen] [→ 9.3 Bedienung des Netzwerks und Unterschiede zum Standard]

Website-Editor [→ 1.2 Eine kurze Geschichte von WordPress] [→ 10.1 Classic Themes und Block-Themes: Was ist das?] [→ 10.3 Block-Theme anpassen: den Website-Editor kennenlernen] [→ 10.3 Block-Theme anpassen: den Website-Editor kennenlernen] [→ 10.3 Block-Theme anpassen: den Website-Editor kennenlernen] [→ 10.3 Block-Theme anpassen: den Website-Editor kennenlernen] [→ 10.3 Block-Theme anpassen: den Website-Editor kennenlernen] [→ 10.3 Block-Theme anpassen: den Website-Editor kennenlernen] [→ A Glossar]

Website-Editor (Block-Theme) [→ 5.1 Das Backend kennenlernen]

Website-Icon [→ 5.2 Einstellungen] [→ 5.2 Einstellungen] [→ 10.4 Classic Theme anpassen] [→ 10.4 Classic Theme anpassen]

Website-Informationen [→ 10.4 Classic Theme anpassen]

Website-Sprache [→ 5.2 Einstellungen]

Website-Titel und -Untertitel [→ 5.2 Einstellungen]

Website-URL ändern [→ 5.3 Einstellungstipps für Fortgeschrittene]

Website-Zustand [→ 5.1 Das Backend kennenlernen] [→ 5.1 Das Backend kennenlernen]

Webspace [→ 4.1 Welches Hosting darf’s denn sein?] [→ 9.1 Was ist WordPress Multisite?]

Weiterleitungen [→ 6.1 Beiträge schreiben/Seiten anlegen] [→ 9.2 Die Multisite-Einrichtung]

Welche Template-Dateien wurden geladen [→ 23.2 Classic Theme anpassen und erstellen]

Werbeanzeige [→ 13.3 Struktur und Navigation]

Werkzeuge [→ 5.1 Das Backend kennenlernen]

Werkzeugleiste [→ 6.1 Beiträge schreiben/Seiten anlegen] [→ 7.4 Benutzerprofil bearbeiten]

Widgets [→ 1.2 Eine kurze Geschichte von WordPress] [→ 5.1 Das Backend kennenlernen] [→ 10.4 Classic Theme anpassen] [→ 10.4 Classic Theme anpassen] [→ 13.1 Zielgruppe und Plan] [→ 13.4 Funktionen und Plugins]

Widget-Bereiche füllen [→ 10.4 Classic Theme anpassen]

Widgets-Bereiche [→ 5.1 Das Backend kennenlernen]

Wildcard [→ 9.2 Die Multisite-Einrichtung]

Windows Subsystem for Linux → siehe [WSL]

Wix [→ 1.3 Warum WordPress?]

WooCommerce [→ 7.1 Rollen verstehen] [→ 12.1 Die Vor- und Nachteile von WooCommerce] [→ 12.1 Die Vor- und Nachteile von WooCommerce]

installieren [→ 12.2 WooCommerce installieren und einrichten]

WooCommerce Tax [→ 12.2 WooCommerce installieren und einrichten]

WooCommerce.com [→ 11.3 Gute Plugins erkennen]

WooCommerce-Shops [→ 12.2 WooCommerce installieren und einrichten]

WooPayments [→ 12.2 WooCommerce installieren und einrichten] [→ 12.2 WooCommerce installieren und einrichten] [→ 12.2 WooCommerce installieren und einrichten]

WordCamp [→ 5.1 Das Backend kennenlernen] [→ 15.1 Das IndieWeb]

WordPress eXtended RSS → siehe [WXR]

WordPress klonen [→ 17.2 Eine Website migrieren]

WordPress Playground → siehe [Playground]

WordPress.com [→ 1.3 Warum WordPress?] [→ 9.1 Was ist WordPress Multisite?] [→ 9.1 Was ist WordPress Multisite?] [→ 15.1 Das IndieWeb] [→ A Glossar]

WordPress.org [→ 11.1 Plugins testen – sicher] [→ 11.3 Gute Plugins erkennen] [→ 11.3 Gute Plugins erkennen] [→ A Glossar]

WordPress-APIs [→ 22.2 WordPress-Konzepte]

WordPress-App [→ 6.6 Die mobilen Apps] [→ 18.3 WordPress abhärten]

WordPress-Backend [→ 7.1 Rollen verstehen]

WordPress-Community [→ 5.1 Das Backend kennenlernen] [→ 11.1 Plugins testen – sicher] [→ 18.1 Login-Sicherheit]

WordPress-Core [→ 2.2 wp-now: Playground für Entwickler*innen] [→ A Glossar]

WordPress-Downgrade [→ 16.1 WordPress, Plugins und Themes aktualisieren]

WordPress-Export [→ 17.2 Eine Website migrieren]

WordPress-Hosting [→ 4.1 Welches Hosting darf's denn sein?]

WordPress-Konfigurationsdatei → siehe [wp-config.php]

WordPress-Meetup [→ 5.1 Das Backend kennenlernen] [→ 11.3 Gute Plugins erkennen] [→ 15.1 Das IndieWeb] [→ A Glossar]

WordPress-Ökosystem [→ 12.1 Die Vor- und Nachteile von WooCommerce]

WordPress-Update [→ 5.1 Das Backend kennenlernen]

WordPress-Updater [→ 11.5 Plugin aktualisieren]

WordPress-URL [→ 5.2 Einstellungen]

WordPress-Veranstaltungen [→ 5.1 Das Backend kennenlernen]

World Wide Web [→ 1.2 Eine kurze Geschichte von WordPress]

wp_enqueue_style() [→ 23.1 Block-Theme erstellen] [→ 23.2 Classic Theme anpassen und erstellen]

WP-CLI [→ 1.2 Eine kurze Geschichte von WordPress] [→ 4.1 Welches Hosting darf's denn sein?] [→ 17.2 Eine Website migrieren] [→ 21.1 Gehackt, was nun?] [→ 22.2 WordPress-Konzepte] [→ 24.8 Übersetzung]

auf geänderte Dateien prüfen [→ 21.1 Gehackt, was nun?]

wp-config.php [→ 5.3 Einstellungstipps für Fortgeschrittene] [→ 9.2 Die Multisite-Einrichtung]

wp-content [→ 2.2 wp-now: Playground für Entwickler*innen]

wp-content-Ordner verschieben [→ 5.3 Einstellungstipps für Fortgeschrittene]

WP-Cron [→ 16.1 WordPress, Plugins und Themes aktualisieren]

wp-now [→ 2.2 wp-now: Playground für Entwickler*innen]

- *Blueprint anwenden* [→ 2.3 Playground mit Blueprints vorkonfigurieren]
- *Datei-Speicherort* [→ 2.2 wp-now: Playground für Entwickler*innen]
- *Installation* [→ 2.2 wp-now: Playground für Entwickler*innen]
- *Installation anpassen* [→ 2.2 wp-now: Playground für Entwickler*innen]
- *PHP-Version festlegen* [→ 2.2 wp-now: Playground für Entwickler*innen]
- *start* [→ 2.2 wp-now: Playground für Entwickler*innen]
- *WordPress-Instanz erstellen* [→ 2.2 wp-now: Playground für Entwickler*innen]
- *WordPress-Version festlegen* [→ 2.2 wp-now: Playground für Entwickler*innen]

WSL [→ 3.2 Installation von Lando] [→ 3.2 Installation von Lando] [→ 3.2 Installation von Lando]

WXR [→ 2.3 Playground mit Blueprints vorkonfigurieren] [→ 2.3 Playground mit Blueprints vorkonfigurieren]

Xdebug [→ 22.3 Entwicklungs- und Testumgebung einrichten]

XML-Export [→ 16.3 Backups planen und durchführen]

XML-RPC [→ 18.3 WordPress abhärten]

xmlrpc.php [→ 21.2 Die häufigsten Standardproblemlösungen]

XML-RPC-Schnittstelle [→ A Glossar]

deaktivieren [→ 21.2 Die häufigsten Standardproblemlösungen]

Y ⇑

Yoast SEO [→ 20.2 Viele SEO-Maßnahmen in einem Plugin mit »Yoast SEO«]

Autoren-Archiv-Einstellungen [→ 20.2 Viele SEO-Maßnahmen in einem Plugin mit »Yoast SEO«]

Beitragseinstellungen [→ 20.2 Viele SEO-Maßnahmen in einem Plugin mit »Yoast SEO«]

Breadcrumbs [→ 20.2 Viele SEO-Maßnahmen in einem Plugin mit »Yoast SEO«]

Crawl-Optimierung [→ 20.2 Viele SEO-Maßnahmen in einem Plugin mit »Yoast SEO«]

Einstellungen [→ 20.2 Viele SEO-Maßnahmen in einem Plugin mit »Yoast SEO«]

Einstellungen für Beitragsformat-Archive [→ 20.2 Viele SEO-Maßnahmen in einem Plugin mit »Yoast SEO«]

Einstellungen für Datums-Archive [→ 20.2 Viele SEO-Maßnahmen in einem Plugin mit »Yoast SEO«]

Einstellungen für Suchergebnisse und Fehlerseiten [→ 20.2 Viele SEO-Maßnahmen in einem Plugin mit »Yoast SEO«]

Einstellungen im Editor [→ 20.2 Viele SEO-Maßnahmen in einem Plugin mit »Yoast SEO«]

Einstellungen importieren/exportieren [→ 20.2 Viele SEO-Maßnahmen in einem Plugin mit »Yoast SEO«]

Feed-Einstellungen [→ 20.2 Viele SEO-Maßnahmen in einem Plugin mit »Yoast SEO«]

Kategorien und Schlagwörter [→ 20.2 Viele SEO-Maßnahmen in einem Plugin mit »Yoast SEO«]

Konfigurationsassistent [→ 20.2 Viele SEO-Maßnahmen in einem Plugin mit »Yoast SEO«]

Massenbearbeitung [→ 20.2 Viele SEO-Maßnahmen in einem Plugin mit »Yoast SEO«]

Medienseiten-Einstellungen [→ 20.2 Viele SEO-Maßnahmen in einem Plugin mit »Yoast SEO«]

Seiten-Einstellungen [→ 20.2 Viele SEO-Maßnahmen in einem Plugin mit »Yoast SEO«]

Startseiten-Einstellungen [→ 20.2 Viele SEO-Maßnahmen in einem Plugin mit »Yoast SEO«]

Webmaster-Tool-Einstellungen [→ 20.2 Viele SEO-Maßnahmen in einem Plugin mit »Yoast SEO«]

Website-Informationen [→ 20.2 Viele SEO-Maßnahmen in einem Plugin mit »Yoast SEO«]

Werkzeuge [→ 20.2 Viele SEO-Maßnahmen in einem Plugin mit »Yoast SEO«]

YouTube [→ 13.1 Zielgruppe und Plan] [→ 13.4 Funktionen und Plugins] [→ 19.7 Optimieren für den Klimaschutz]

Z ⇑

Zählmarken [→ 13.4 Funktionen und Plugins]

Zahlungsmethoden [→ 12.2 WooCommerce installieren und einrichten]

Zeichensatz [→ 23.2 Classic Theme anpassen und erstellen]

Zeiteinstellungen [→ 5.2 Einstellungen]

Zielgruppe [→ 13.1 Zielgruppe und Plan] [→ 13.1 Zielgruppe und Plan]

Zugriffsrechte [→ 14.1 Aufgaben einer Vereins-Website] [→ 14.3 Funktionalitäten und Plugins]

Zusätzliche Bildgröße registrieren [→ 23.4 Weitere nützliche Funktionen für Themes]

Zusätzliches CSS [→ 10.4 Classic Theme anpassen] [→ 10.4 Classic Theme anpassen]

Zwei-Faktor-Authentifizierung [→ 18.1 Login-Sicherheit] [→ 18.1 Login-Sicherheit] [→ 18.1 Login-Sicherheit] [→ A Glossar]

Rechtliche Hinweise

Das vorliegende Werk ist in all seinen Teilen urheberrechtlich geschützt. Weitere Hinweise dazu finden Sie in den Allgemeinen Geschäftsbedingungen des Anbieters, bei dem Sie das Werk erworben haben.

Die automatisierte Analyse des Werkes, um daraus Informationen insbesondere über Muster, Trends und Korrelationen gemäß § 44b UrhG (»Text und Data Mining«) zu gewinnen, ist untersagt.

Markenschutz

Die in diesem Werk wiedergegebenen Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. können auch ohne besondere Kennzeichnung Marken sein und als solche den gesetzlichen Bestimmungen unterliegen.

Haftungsausschluss

Ungeachtet der Sorgfalt, die auf die Erstellung von Text, Abbildungen und Programmen verwendet wurde, können weder Verlag noch Autor*innen, Herausgeber*innen, Übersetzer*innen oder Anbieter für mögliche Fehler und deren Folgen eine juristische Verantwortung oder irgendeine Haftung übernehmen.

Über das Autorenteam



Florian Brinkmann ist Webentwickler und langjähriges Mitglied der deutschsprachigen WordPress-Community Er schreibt unter anderem für sein eigenes Blog (*florianbrinkmann.com/blog*), veröffentlicht Fachartikel auf KrautPress.de und sporadisch für t3n.



Simon Kraft arbeitet seit 2008 mit WordPress. Er ist tief in der WordPress-Szene verwurzelt. Im WP Letter, seinem wöchentlichen Newsletter, hält er die wichtigsten Themen der WordPress-Welt fest.

Dokumentenarchiv

Das Dokumentenarchiv umfasst alle Abbildungen und ggf. Tabellen und Fußnoten dieses E-Books im Überblick.

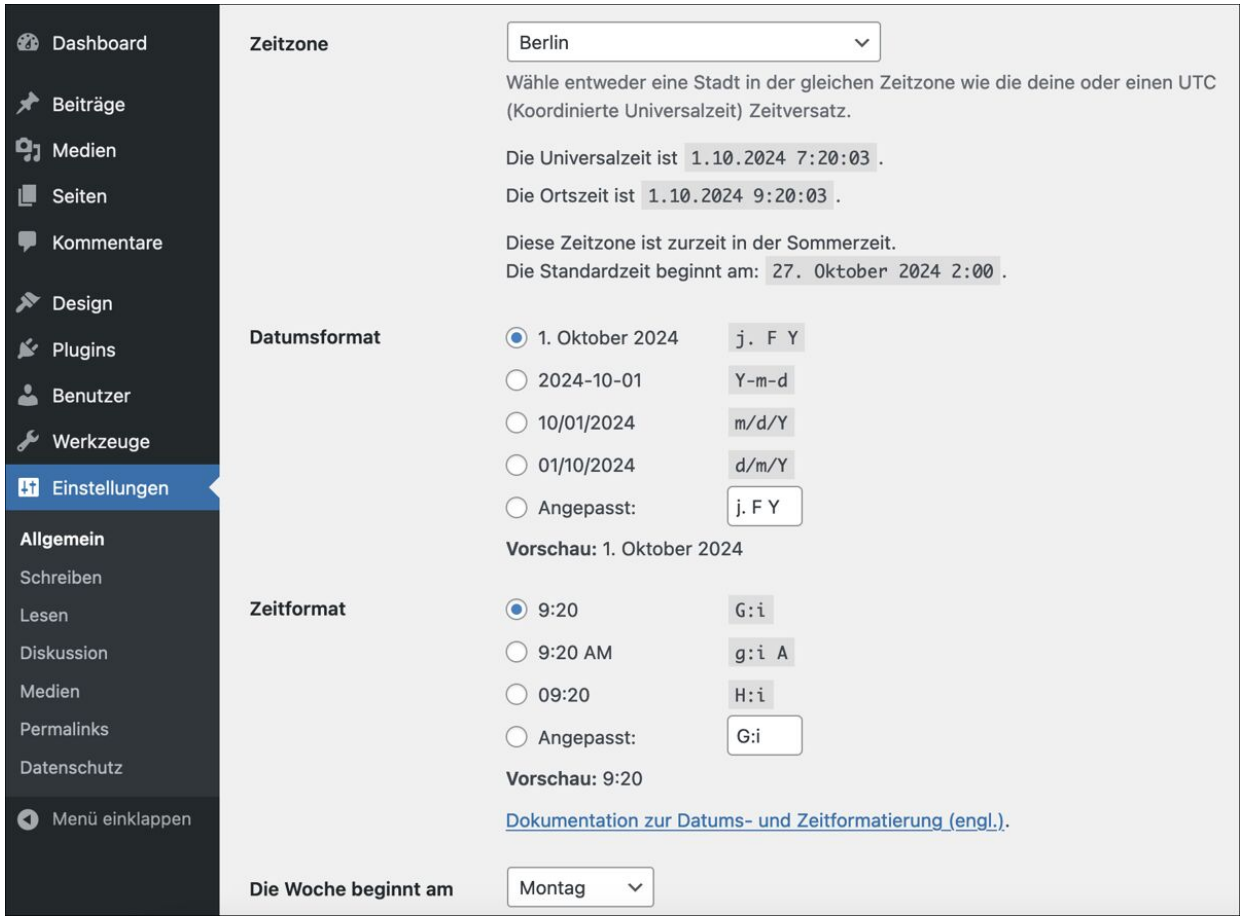


Abbildung 1.1 In einer Bildunterschrift werden die Interaktionselemente in Anführungszeichen gesetzt: »Einstellungen« • »Allgemein« • »Datumsformat«.

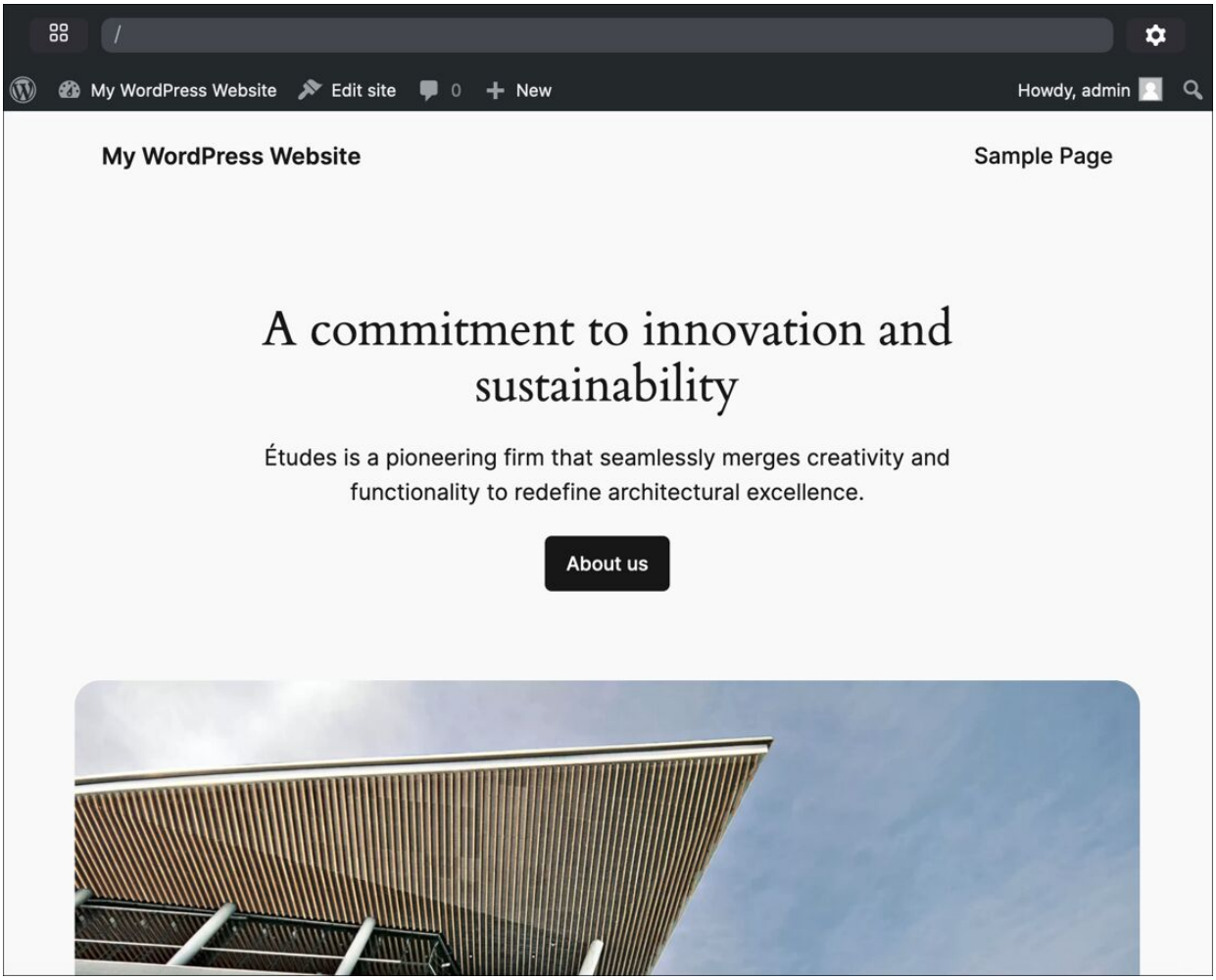


Abbildung 2.1 Wenn Sie »playground.wordpress.net« aufrufen, wird in Ihrem Browser eine frische WordPress-Installation erstellt.

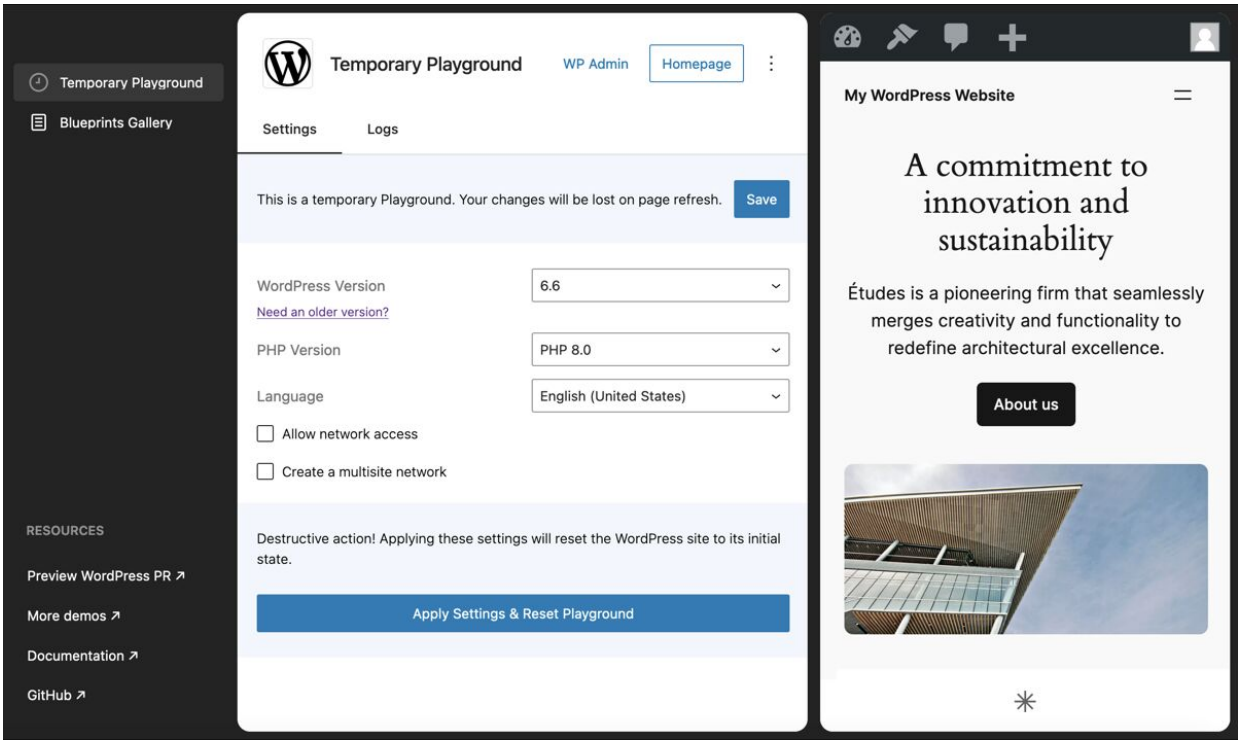


Abbildung 2.2 In dieser Ansicht des Playground haben Sie die Möglichkeit, links aus fertigen Playground-Blueprints zu wählen und in der Mitte die Einstellungen der Playground-Instanz zu ändern.

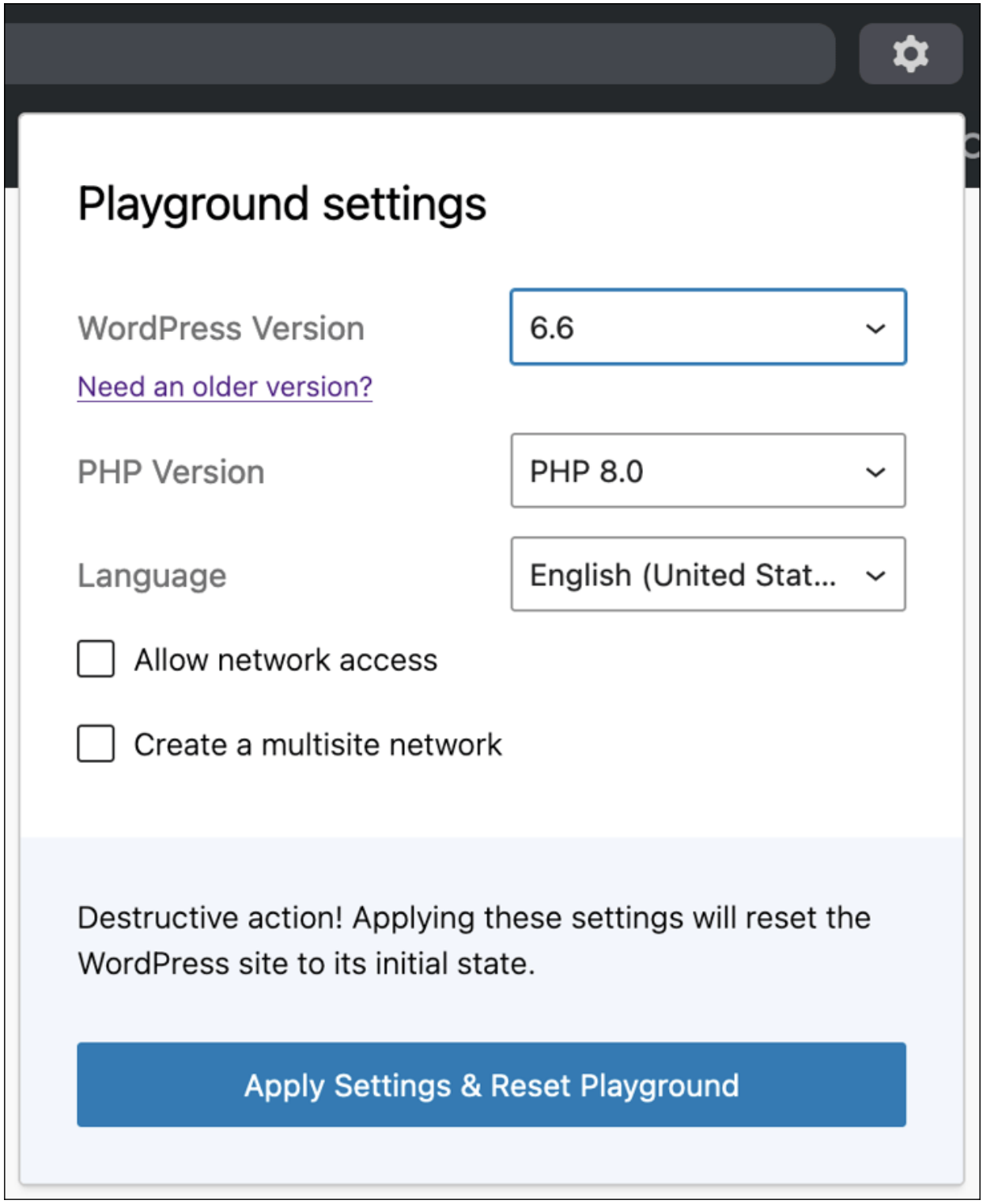

Playground settings
WordPress Version
6.6
Need an older version?
PHP Version
PHP 8.0
Language
English (United Stat...
Allow network access
Create a multisite network
Destructive action! Applying these settings will reset the WordPress site to its initial state.
Apply Settings & Reset Playground

Abbildung 2.3 Einstellungen wie PHP- und WordPress-Version können auch über das Zahnrad-Icon rechts neben der Adressleiste erreicht werden.

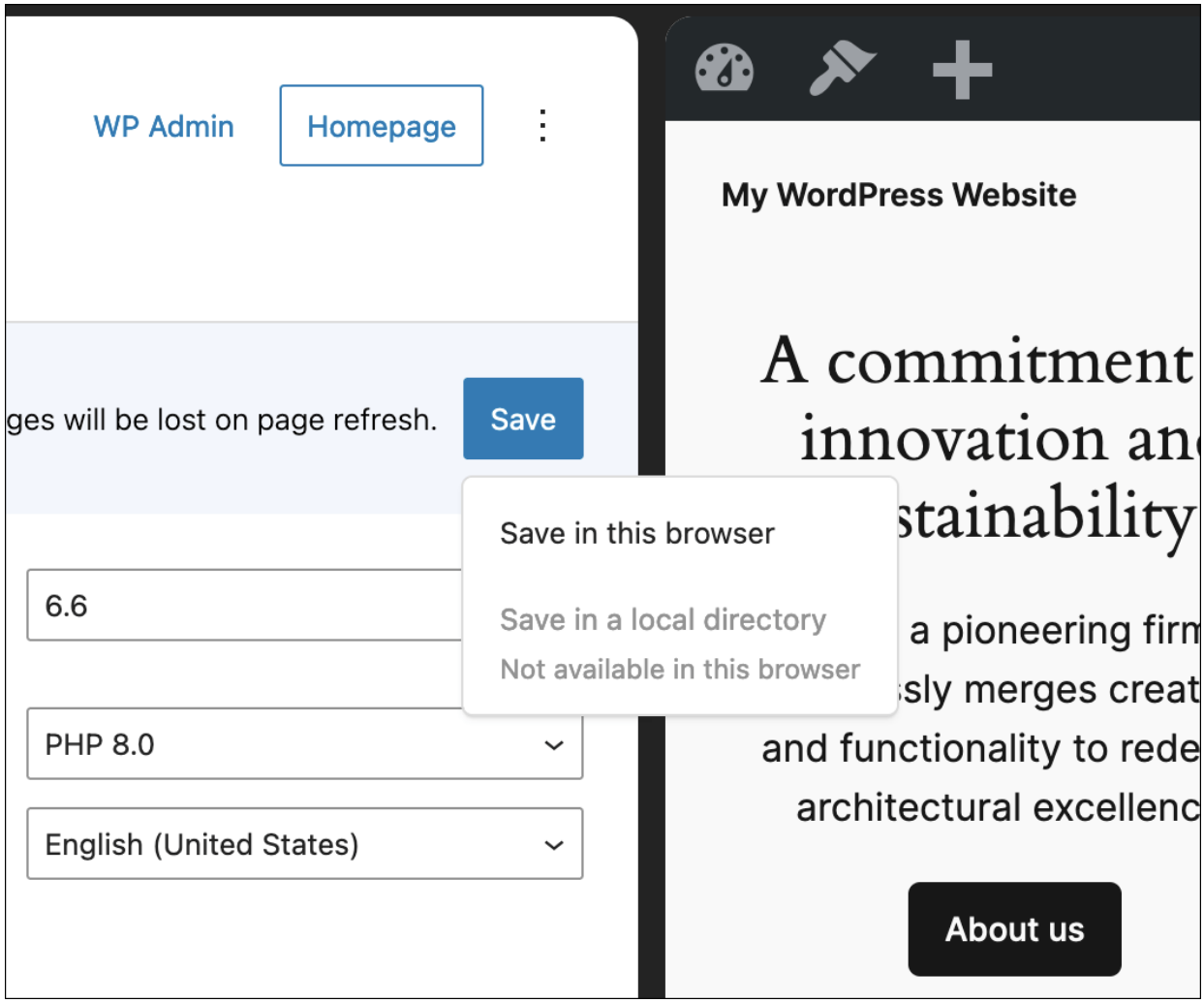


Abbildung 2.4 Sie können den Playground im Browser oder – in Browsern, die dies unterstützen – lokal speichern.

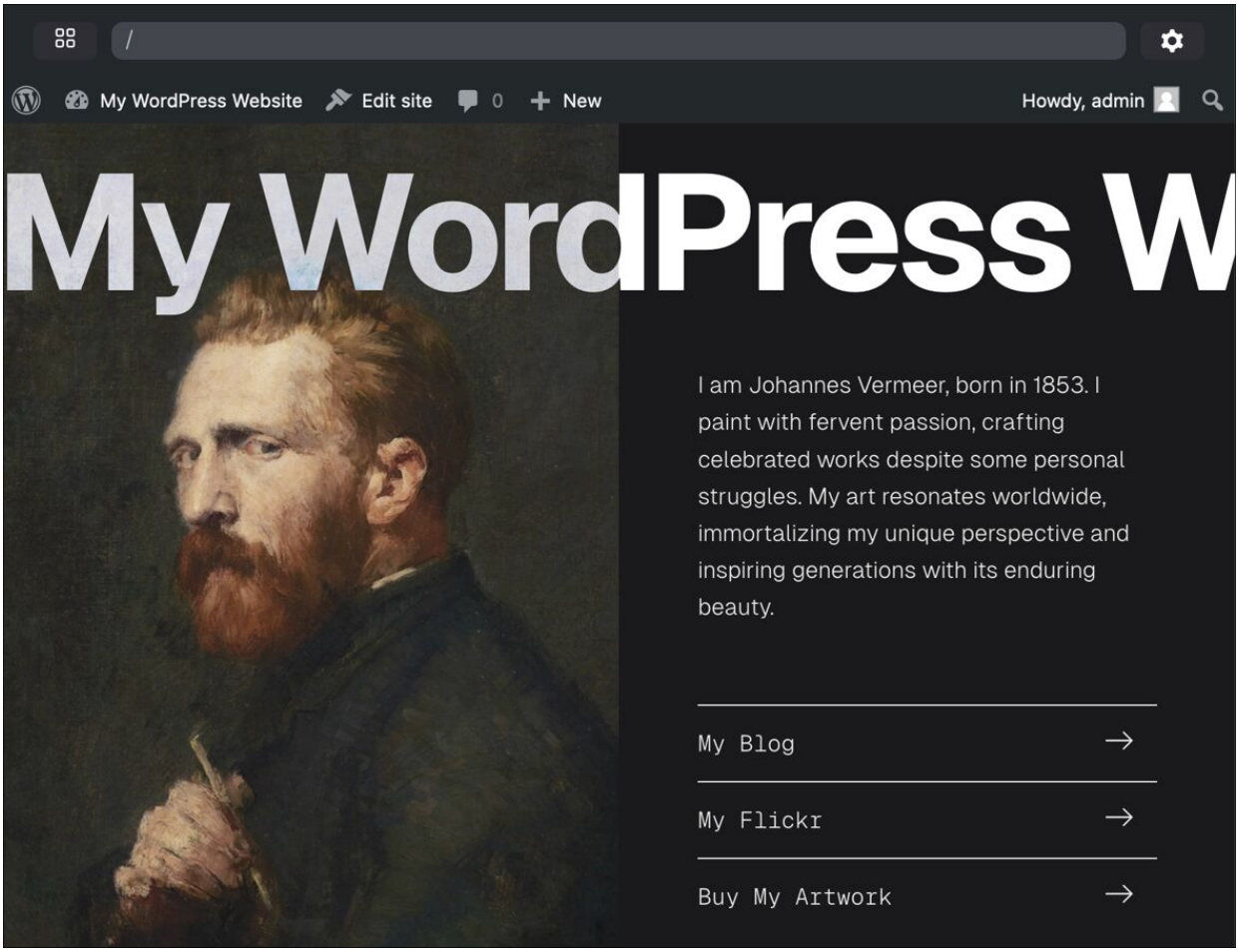


Abbildung 2.5 Das Theme ist nicht nur vorinstalliert, sondern auch direkt aktiv.

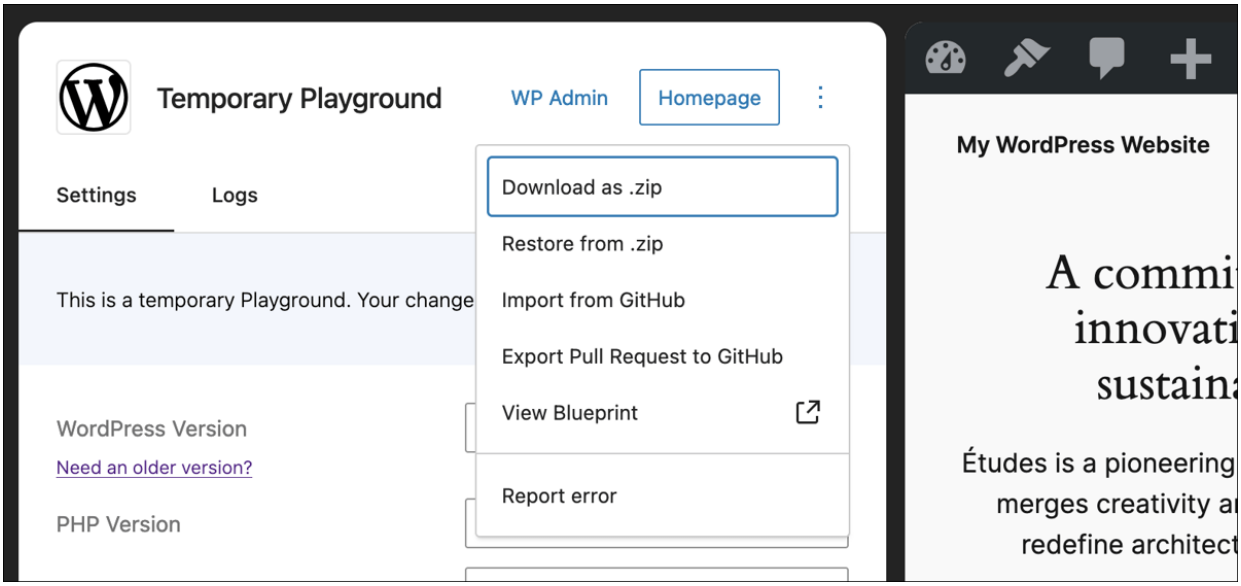


Abbildung 2.6 Obwohl die Instanz temporär in Ihrem Browser entsteht, können Sie sie als ZIP exportieren und später wieder importieren, um weiterzuarbeiten.

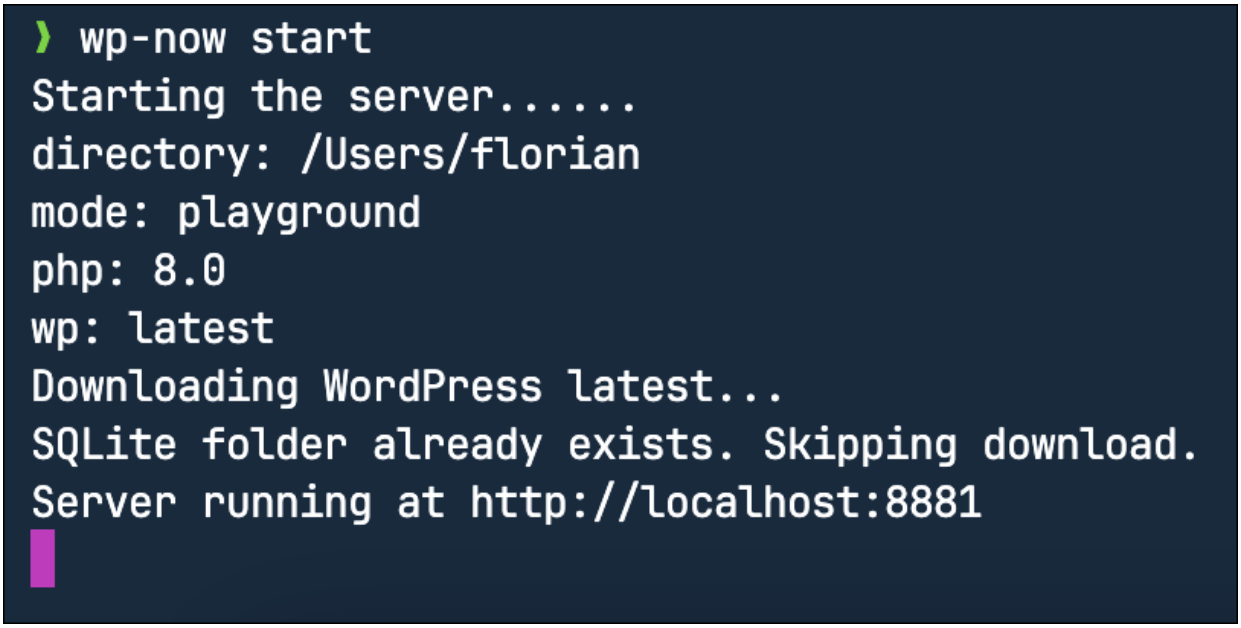


Abbildung 2.7 So sieht beispielhaft die Ausgabe von wp-now im Terminal aus.

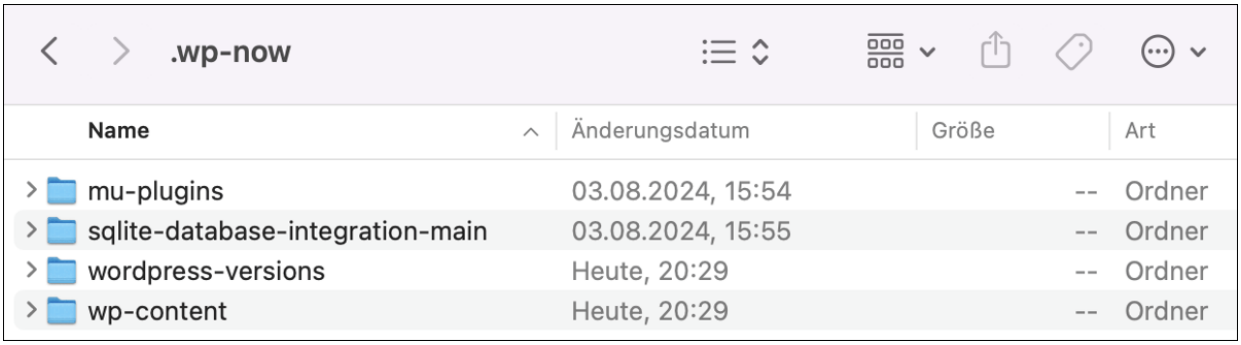


Abbildung 2.8 So in etwa sieht der Inhalt des ».wp-now«-Verzeichnisses nach Erstellen einer Instanz aus.

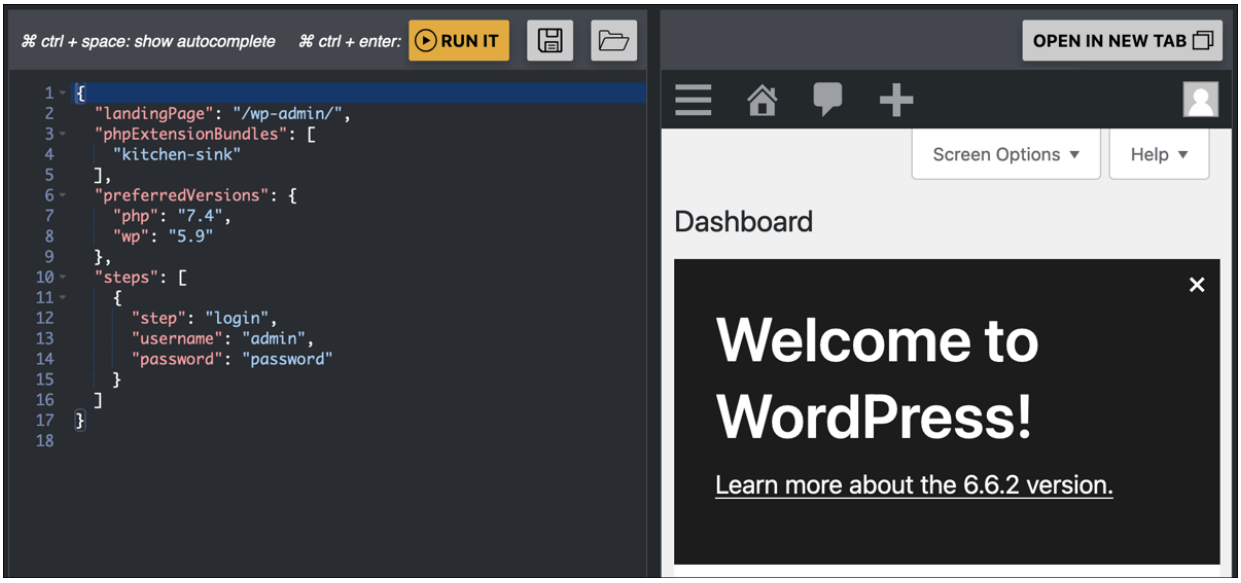


Abbildung 2.9 Der Blueprint-Builder ist ein nützliches Tool beim Erstellen von Blueprints.

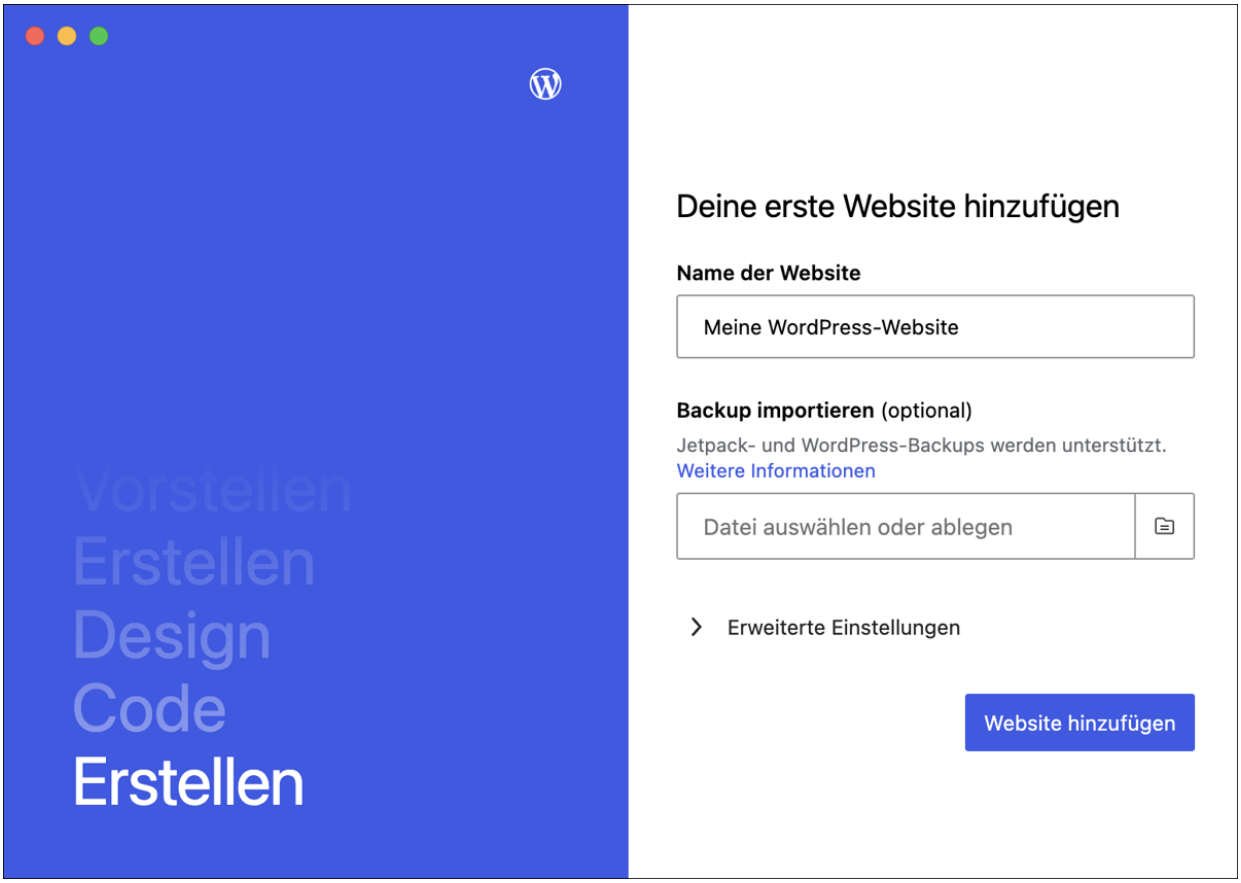


Abbildung 2.10 Mit »Studio« haben Sie ein hübsches Interface für Playground-Instanzen.

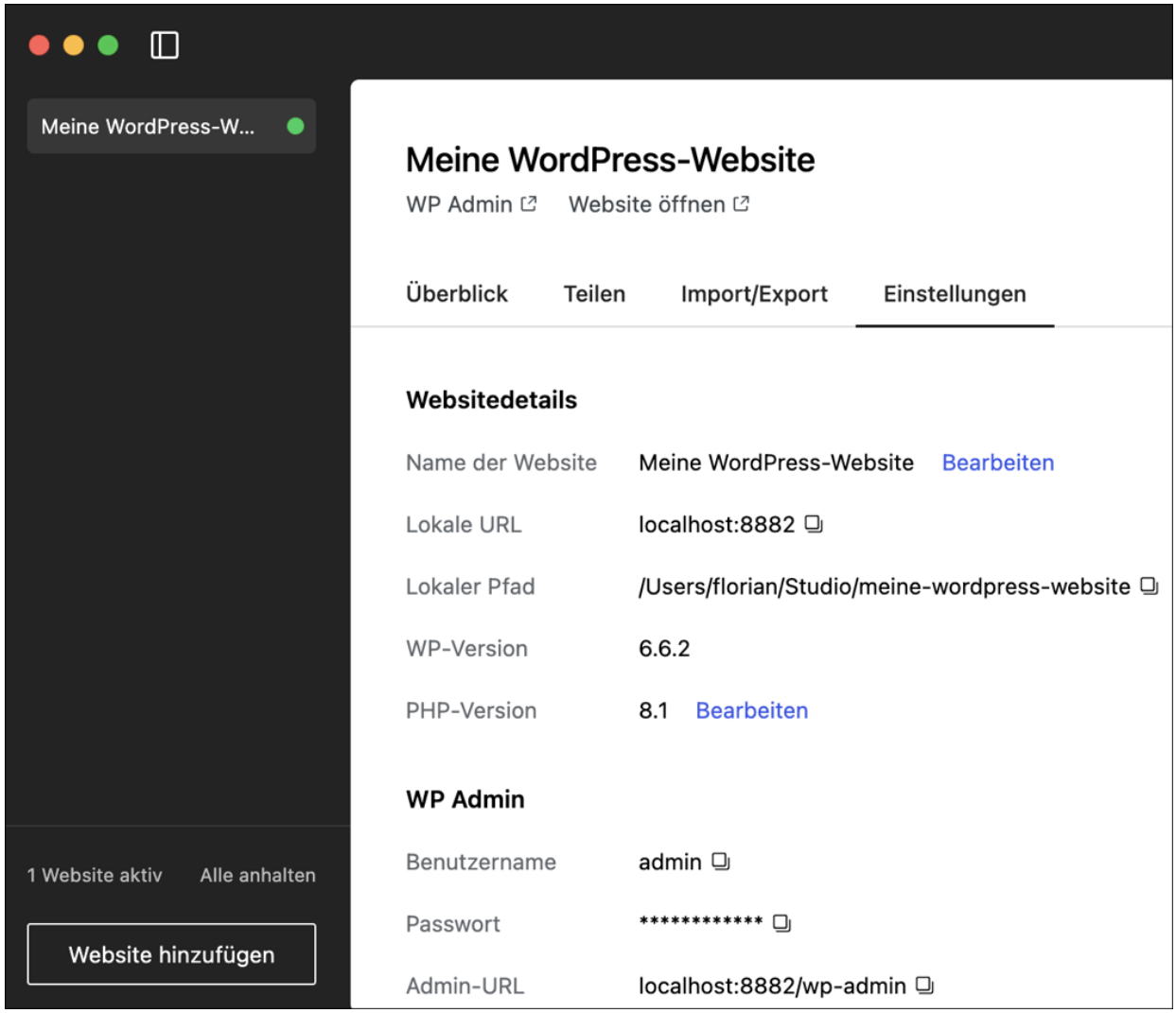


Abbildung 2.11 In den Einstellungen zu einer Site sehen Sie Informationen wie den Pfad zu den WordPress-Dateien.

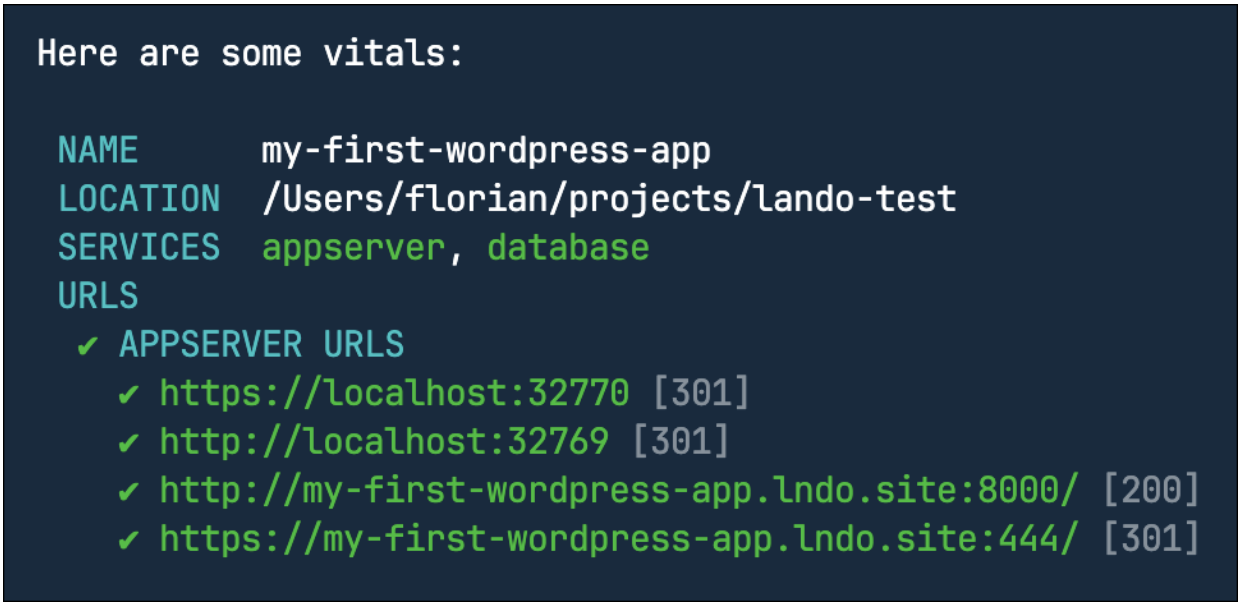


Abbildung 3.1 Ein paar Daten zur Erreichbarkeit der Lando-WordPress-Installation am Ende des »lando start«-Befehls

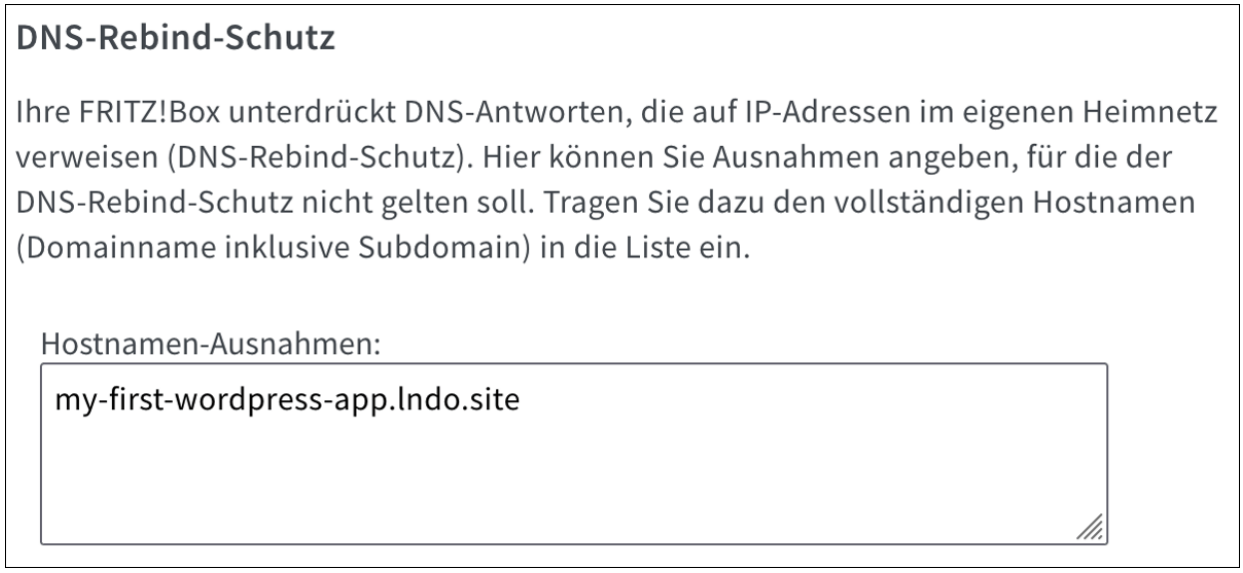


Abbildung 3.2 Die Angabe der Lando-Domain als Hostnamen-Ausnahme im »DNS-Rebind-Schutz« einer FRITZ!Box kann das Problem der Erreichbarkeit der lokalen Domain beheben.

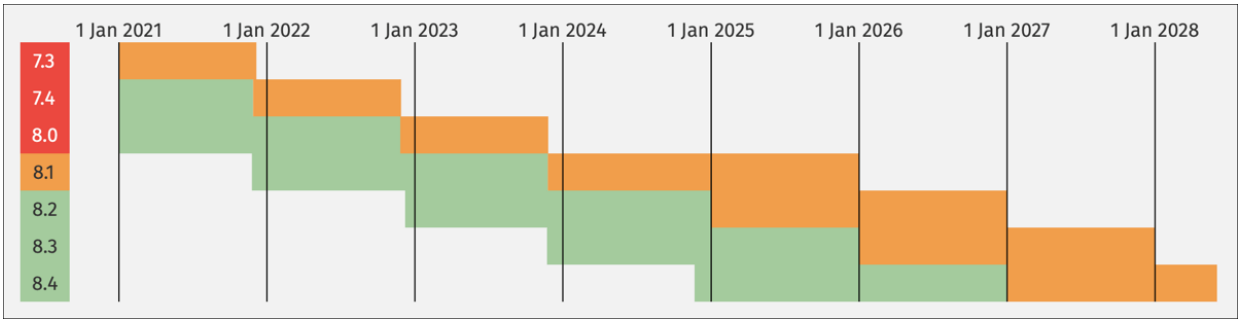


Abbildung 4.1 PHP erhält wie andere Programmiersprachen Software-Updates. Zwei Jahre nach Veröffentlichung wechselt eine PHP-Version in eine Phase, in der sie nur noch Sicherheitsupdates bekommt, nach weiteren zwei Jahren enden auch diese. Ein im November 2024 veröffentlichtes PHP 8.4 erreicht damit also Ende 2028 sein Ende und sollte spätestens dann nicht mehr eingesetzt werden.

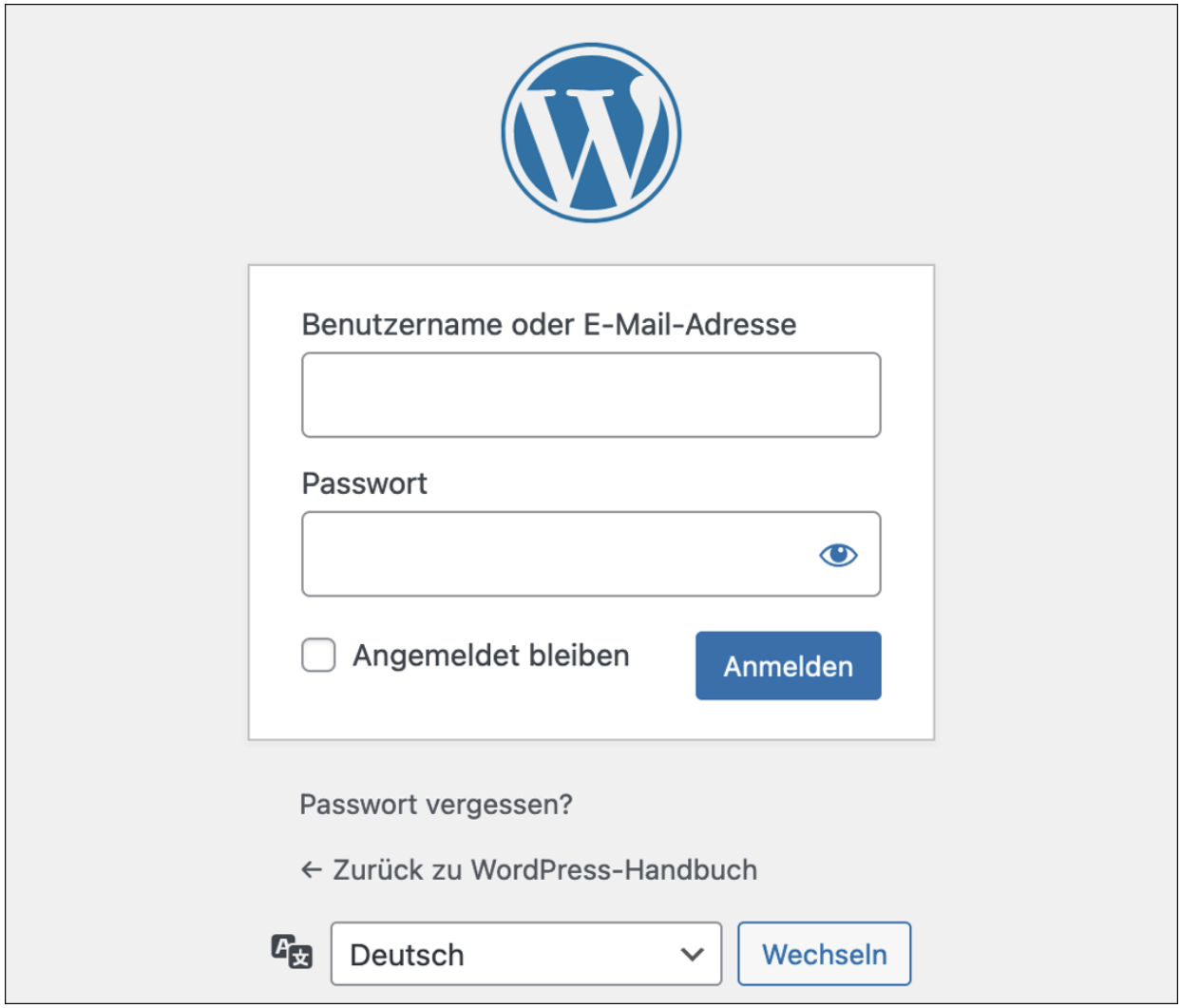


Abbildung 5.1 Loggen Sie sich mit Ihrer während der Installation vergebenen Benutzername-Passwort-Kombination im Backend ein.



Abbildung 5.2 Die Adminbar am oberen Rand des Browserfensters beinhaltet hilfreiche Links und sieht je nach installierten Plugins unterschiedlich aus.

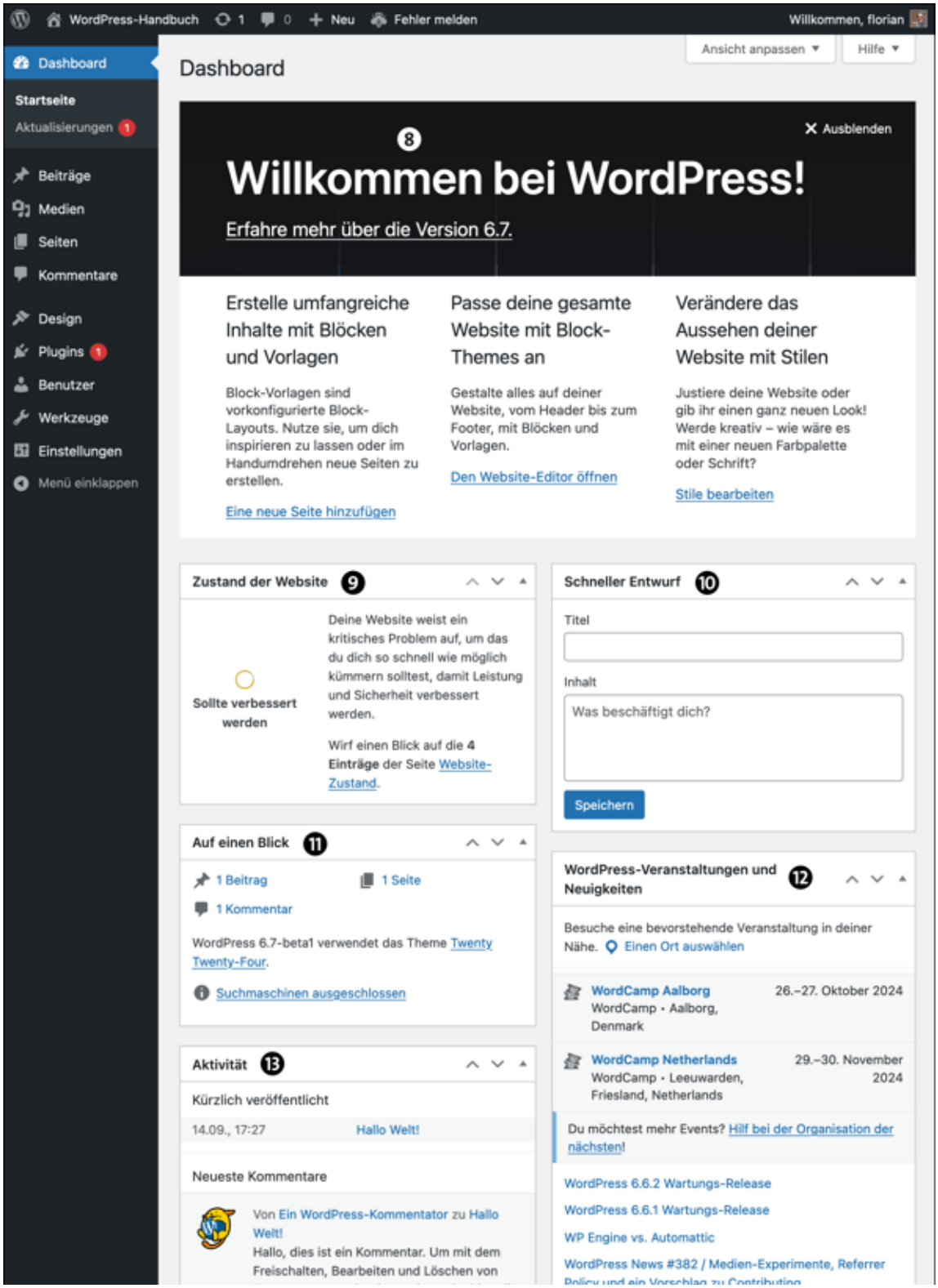

WordPress-Handbuch
1
0
Neu
Fehler melden
Willkommen, florian
Dashboard
Startseite
Aktualisierungen 1
Beiträge
Medien
Seiten
Kommentare
Design
Plugins 1
Benutzer
Werkzeuge
Einstellungen
Menü einklappen
Dashboard
Ansicht anpassen
Hilfe
8
Ausblenden
Willkommen bei WordPress!
Erfahre mehr über die Version 6.7.
Erstelle umfangreiche Inhalte mit Blöcken und Vorlagen
Block-Vorlagen sind vorkonfigurierte Block-Layouts. Nutze sie, um dich inspirieren zu lassen oder im Handumdrehen neue Seiten zu erstellen.
Eine neue Seite hinzufügen
Passe deine gesamte Website mit Block-Themes an
Gestalte alles auf deiner Website, vom Header bis zum Footer, mit Blöcken und Vorlagen.
Den Website-Editor öffnen
Verändere das Aussehen deiner Website mit Stilen
Justiere deine Website oder gib ihr einen ganz neuen Look! Werde kreativ – wie wäre es mit einer neuen Farbpalette oder Schrift?
Stile bearbeiten
Zustand der Website 9
Sollte verbessert werden
Deine Website weist ein kritisches Problem auf, um das du dich so schnell wie möglich kümmern solltest, damit Leistung und Sicherheit verbessert werden.
Wirf einen Blick auf die 4 Einträge der Seite Website-Zustand.
Schneller Entwurf 10
Titel
Inhalt
Was beschäftigt dich?
Speichern
Auf einen Blick 11
1 Beitrag
1 Seite
1 Kommentar
WordPress 6.7-beta1 verwendet das Theme Twenty Twenty-Four.
Suchmaschinen ausgeschlossen
WordPress-Veranstaltungen und Neuigkeiten 12
Besuche eine bevorstehende Veranstaltung in deiner Nähe.
Einen Ort auswählen
WordCamp Aalborg
WordCamp · Aalborg, Denmark
26.–27. Oktober 2024
WordCamp Netherlands
WordCamp · Leeuwarden, Friesland, Netherlands
29.–30. November 2024
Du möchtest mehr Events? Hilf bei der Organisation der nächsten!
WordPress 6.6.2 Wartungs-Release
WordPress 6.6.1 Wartungs-Release
WP Engine vs. Automattic
WordPress News #382 / Medien-Experimente, Referrer
Aktivität 13
Kürzlich veröffentlicht
14.09., 17:27
Hallo Welt!
Neueste Kommentare
Von Ein WordPress-Kommentator zu Hallo Welt!
Hallo, dies ist ein Kommentar. Um mit dem Freischalten, Bearbeiten und Löschen von



Abbildung 5.3 Das Dashboard ist die erste Seite, die Sie nach dem Einloggen ins Backend sehen.

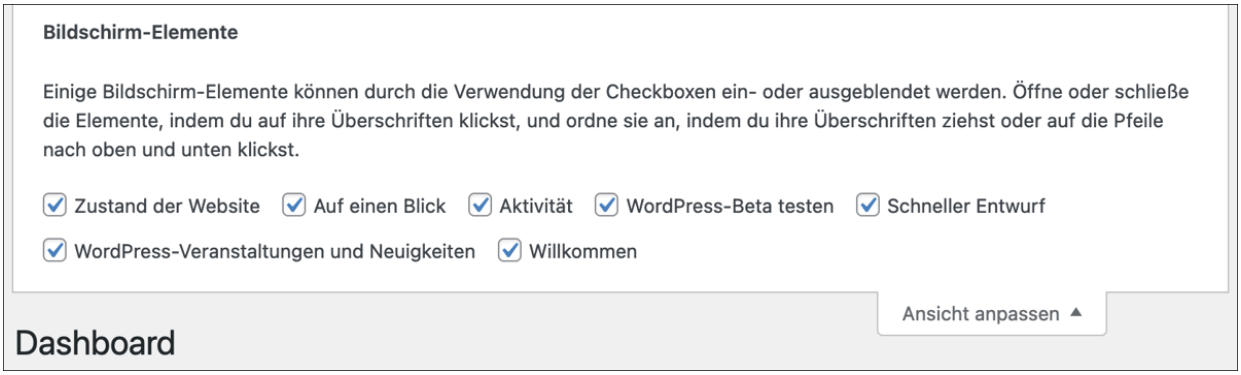
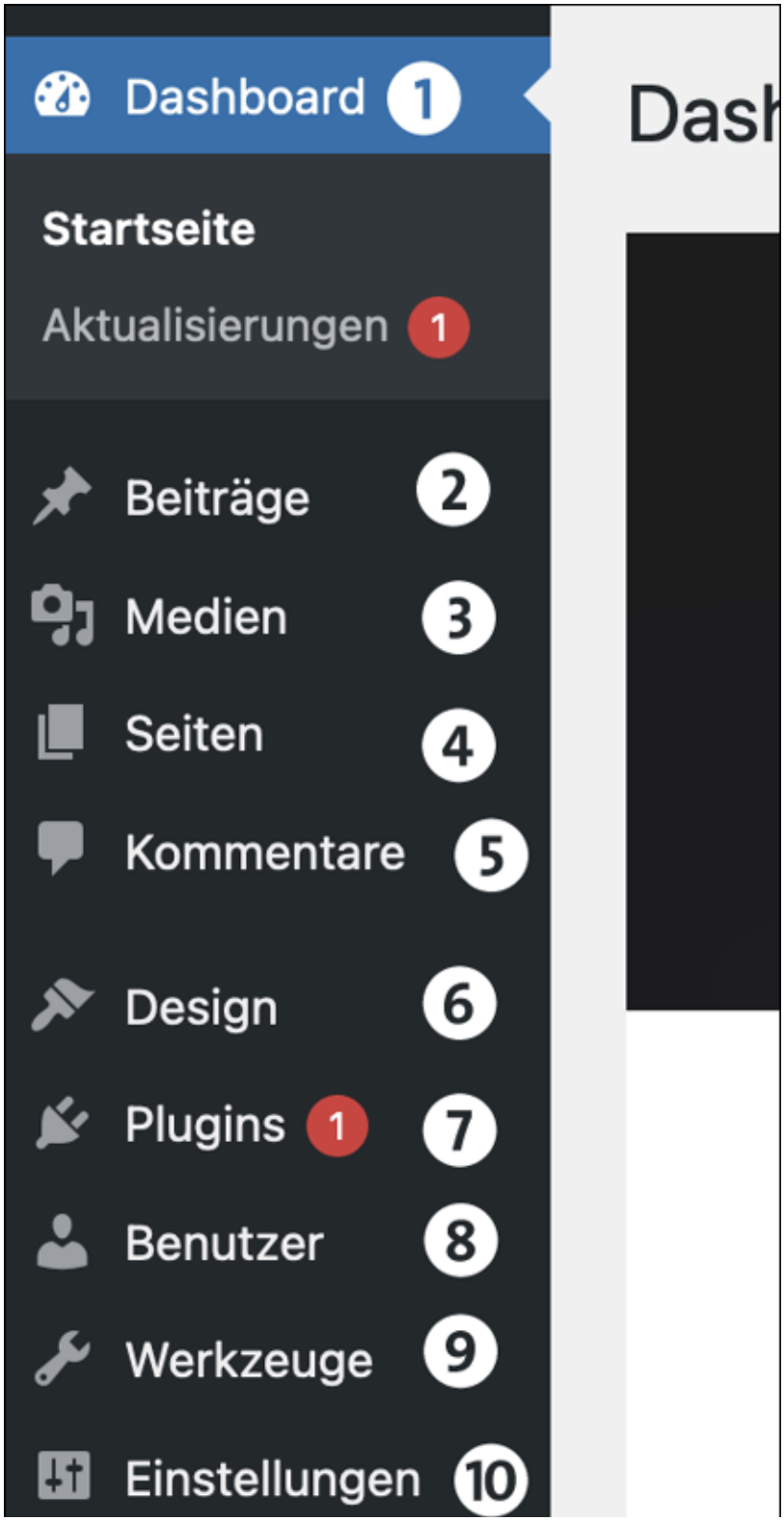


Abbildung 5.4 Die angezeigten Widgets im Dashboard lassen sich über den Button »Ansicht anpassen« ändern.


Dashboard 1
Startseite
Aktualisierungen 1
Beiträge 2
Medien 3
Seiten 4
Kommentare 5
Design 6
Plugins 1 7
Benutzer 8
Werkzeuge 9
Einstellungen 10
Dash

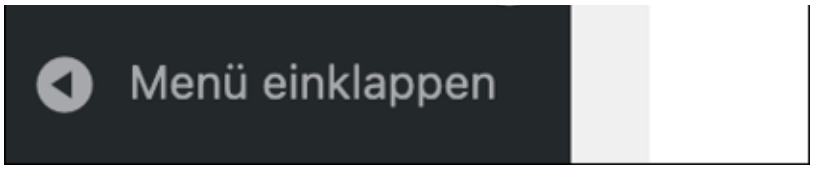


Abbildung 5.5 Über das Admin-Menü auf der linken Seite gelangen Sie zu allen Bereichen des Backends.

Hilfe ▼

WordPress-Aktualisierungen

Es kann einige Minuten dauern, bis die Aktualisierungen abgeschlossen sind. Wenn du nach 5 Minuten keine Rückmeldung erhalten hast oder wenn Fehler auftreten, lies bitte den Abschnitt Hilfe weiter oben.

Aktuelle Version: 6.7-beta1

Zuletzt überprüft am 6. Oktober 2024 um 16:21 CEST Uhr. Erneut überprüfen.

Diese Website wird automatisch mit jeder neuen Version von WordPress auf den aktuellsten Stand gebracht.
Auf automatische Aktualisierungen nur für Wartungs- und Sicherheits-Updates wechseln.

Du benutzt eine Entwicklungsversion von WordPress.

Version 6.7-beta1–en_US erneut installieren

Version 6.6.2–en_US erneut installieren

Deine Website wird während des Aktualisierungsprozesses in einen Wartungsmodus geschaltet. Sobald die Aktualisierung abgeschlossen ist, wird der Wartungsmodus deaktiviert.

Plugins (1)

Für die folgenden Plugins sind neue Versionen verfügbar. Markiere diejenigen, die du aktualisieren möchtest und klicke auf „Plugins aktualisieren“.

Plugins aktualisieren

☐ Alle auswählen

☐ **WordPress Beta Tester**
Du hast Version 3.5.5 installiert. Aktualisiere auf Version 3.5.6. Details der Version 3.5.6 anzeigen.
Kompatibilität mit WordPress 6.7: unbekannt

☐ Alle auswählen

Plugins aktualisieren

Abbildung 5.6 Über »Dashboard« • »Aktualisierungen« können Sie alle WordPress-Komponenten aktualisieren.

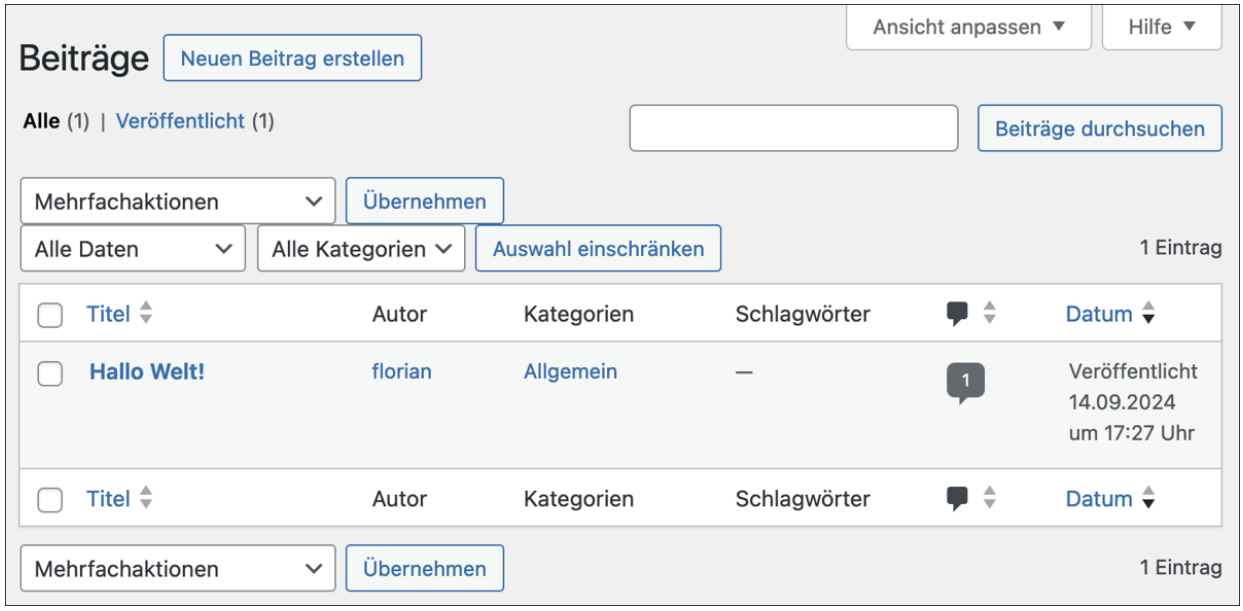


Abbildung 5.7 Die Liste mit allen Beiträgen Ihrer Website

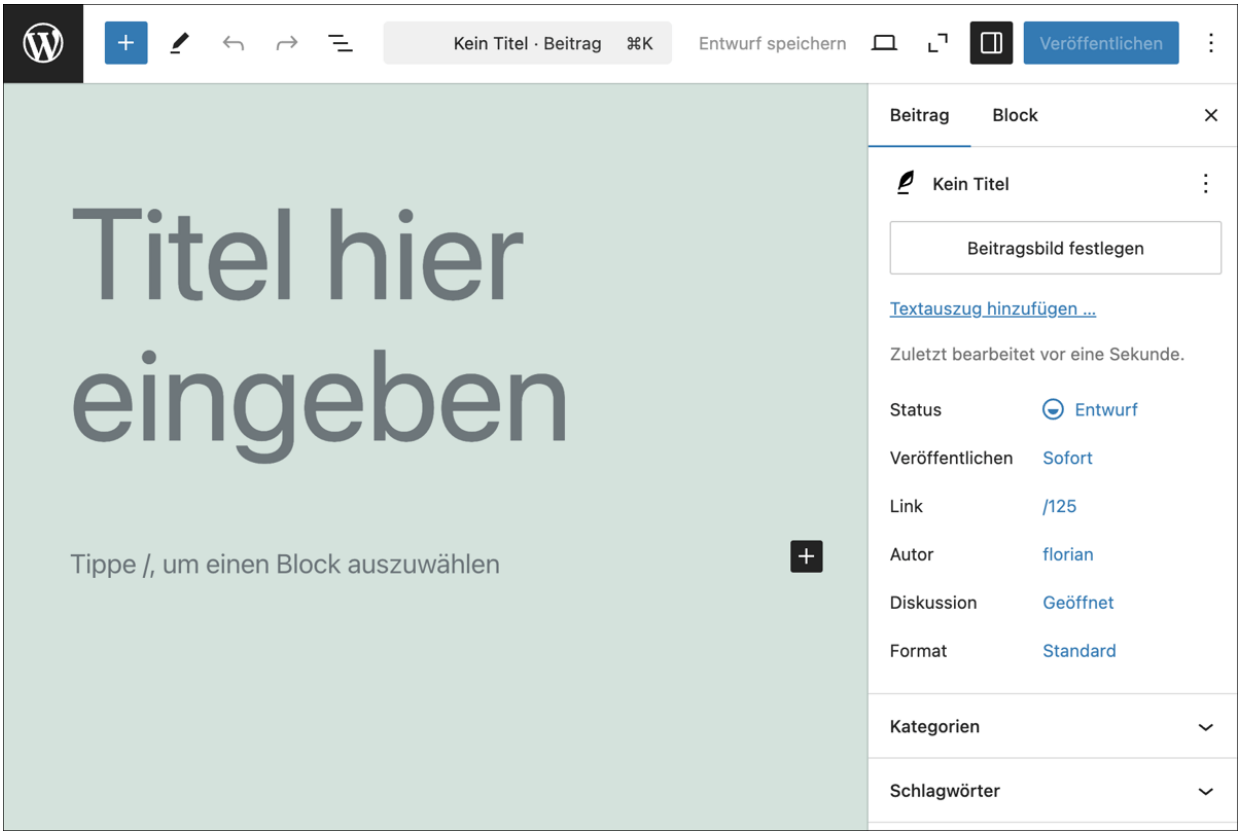


Abbildung 5.8 Der Block-Editor zum Anlegen eines neuen Beitrags.

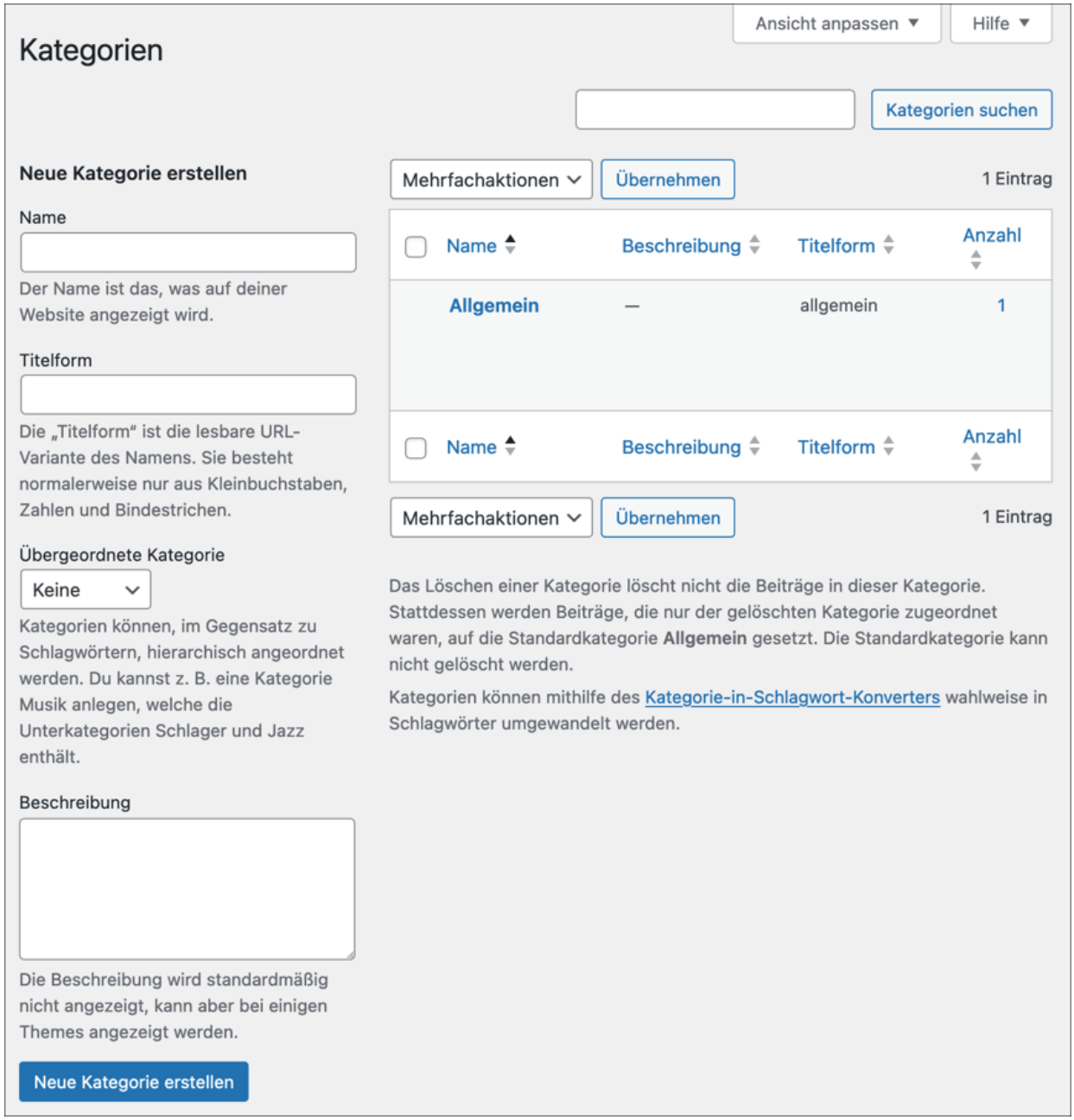


Abbildung 5.9 Über Kategorien können Sie Beiträge nach thematischer Gemeinsamkeit zusammenfassen.

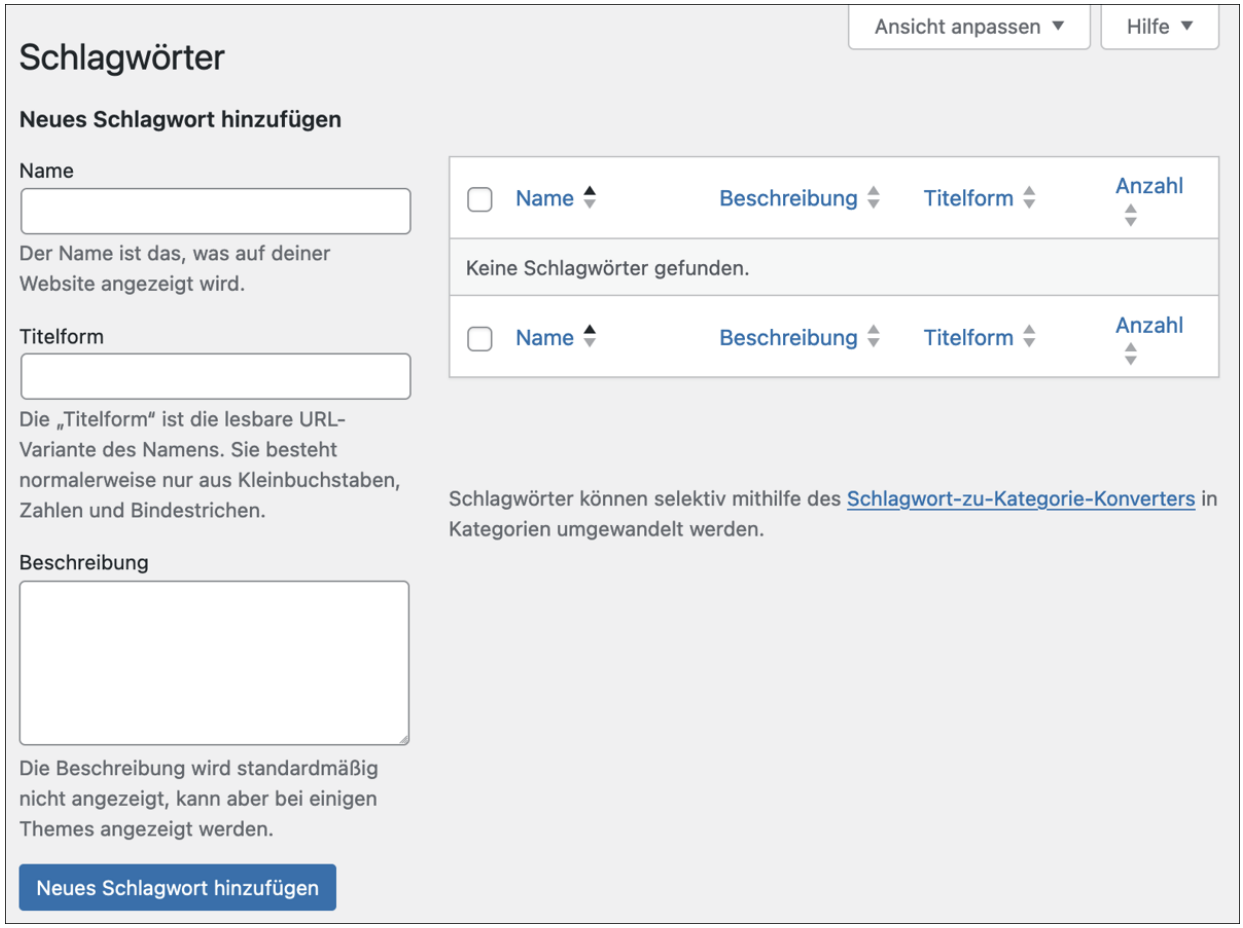


Abbildung 5.10 Mit Schlagwörtern können Sie Ihre Beiträge bei Bedarf feiner unterteilen.

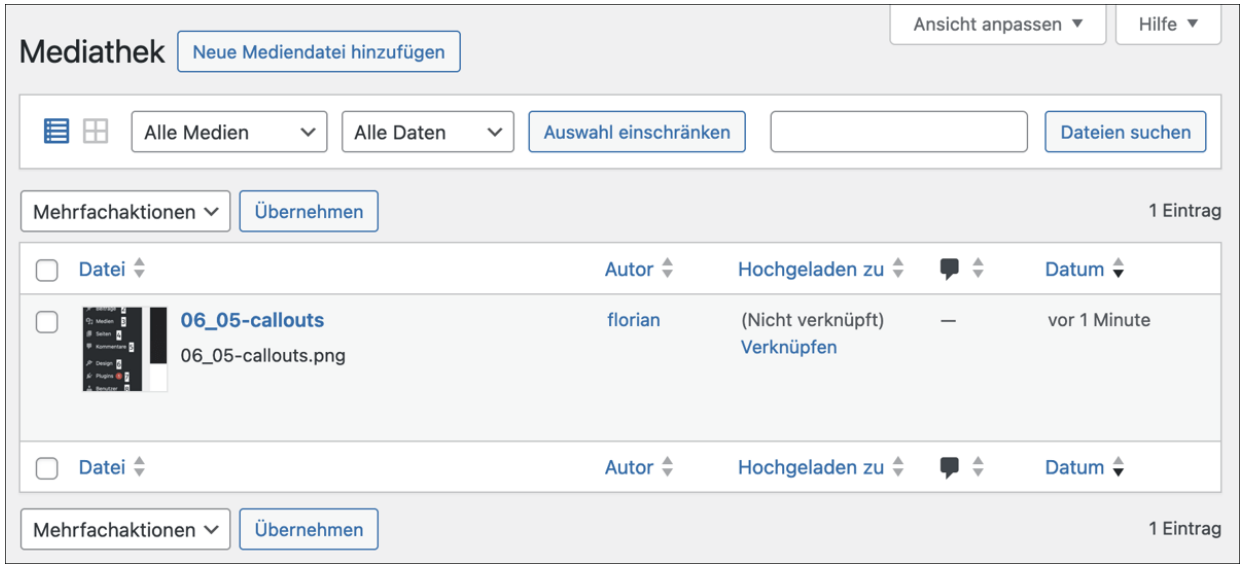


Abbildung 5.11 Die Mediathek zeigt Ihnen alle Medien an, die über WordPress zur Website hochgeladen wurden.

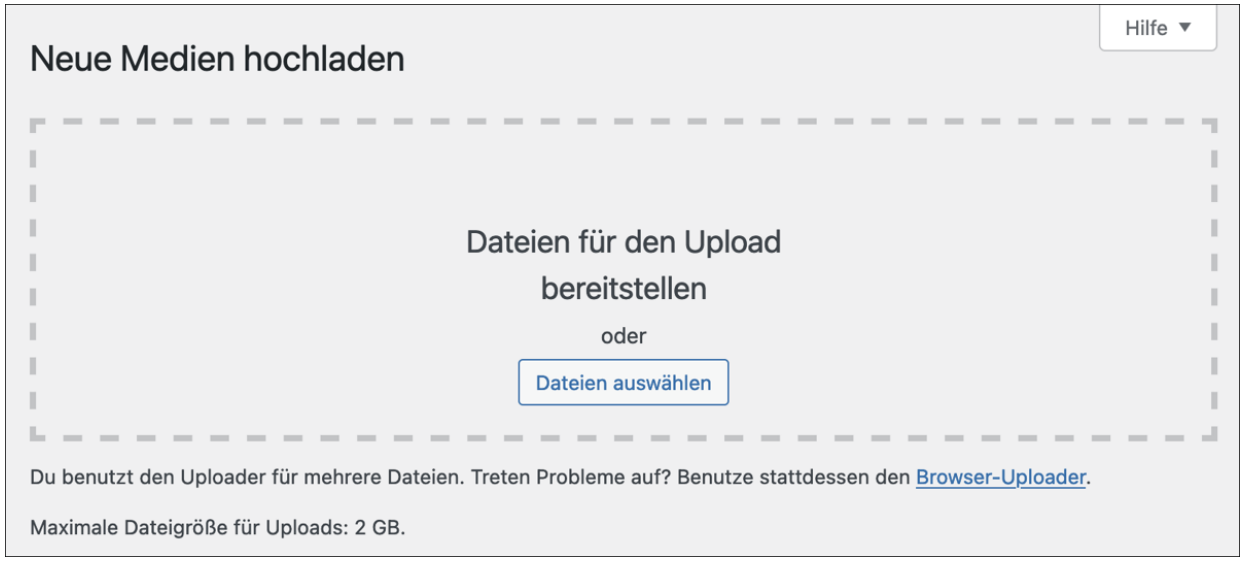


Abbildung 5.12 Neben den Möglichkeiten des Uploads per Drag-and-drop über die Mediathek an sich gibt es noch eine separate Seite zum Hochladen neuer Dateien.

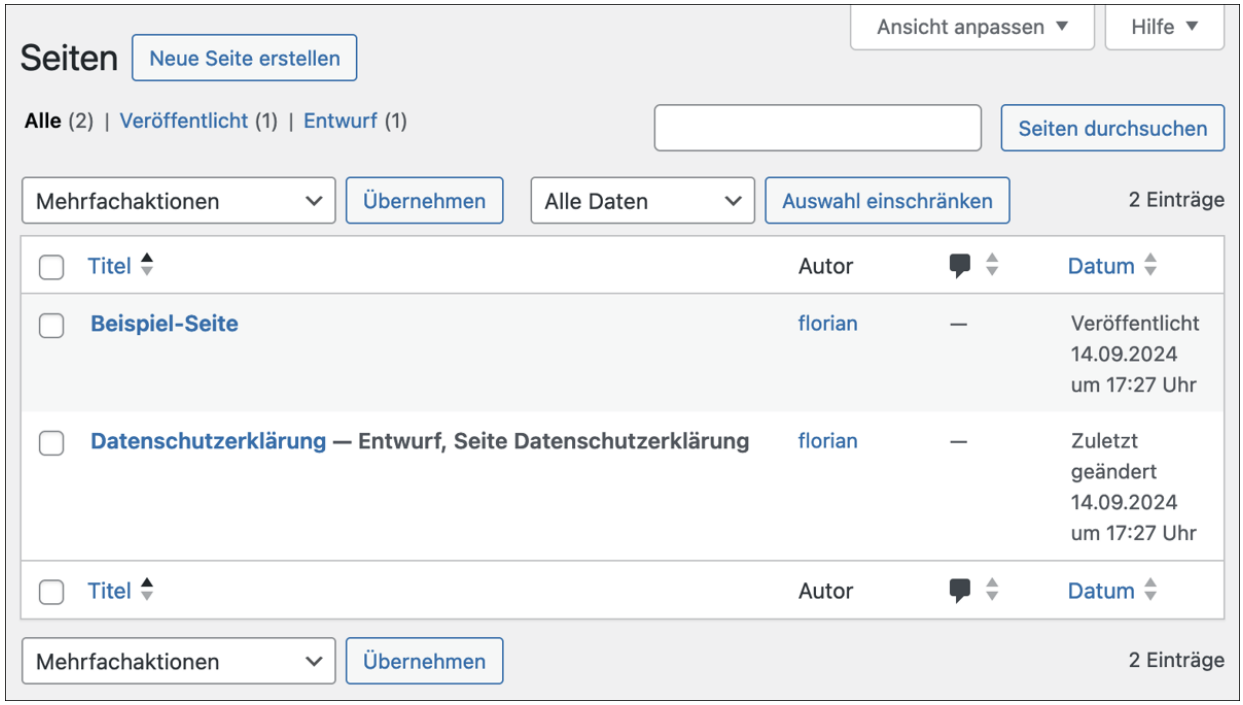


Abbildung 5.13 Auf einer frischen Installation finden Sie in der Liste aller Seiten eine Beispielseite und eine Vorlage für die Datenschutzerklärung, die Sie vor Live-Schaltung der Website unbedingt bearbeiten müssen.

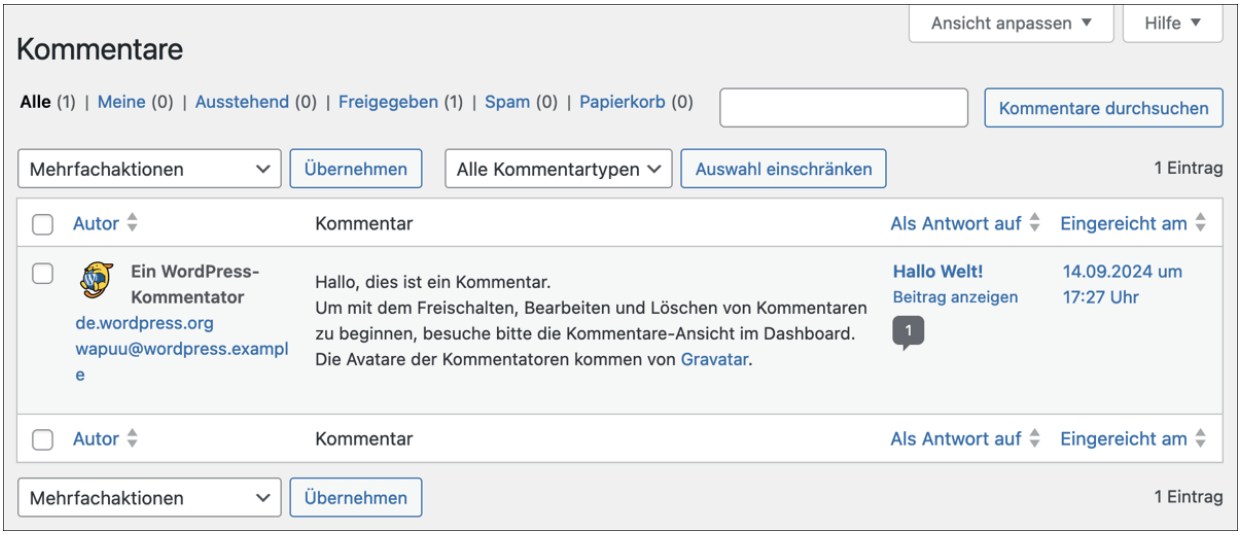


Abbildung 5.14 Hier finden Sie alle Kommentare, die auf Ihrer Website zu Inhalten geschrieben wurden. Wenn Sie ein Plugin wie das Webmention-Plugin installiert haben, finden Sie hier auch Reaktionen von anderen Sites und Netzwerken.

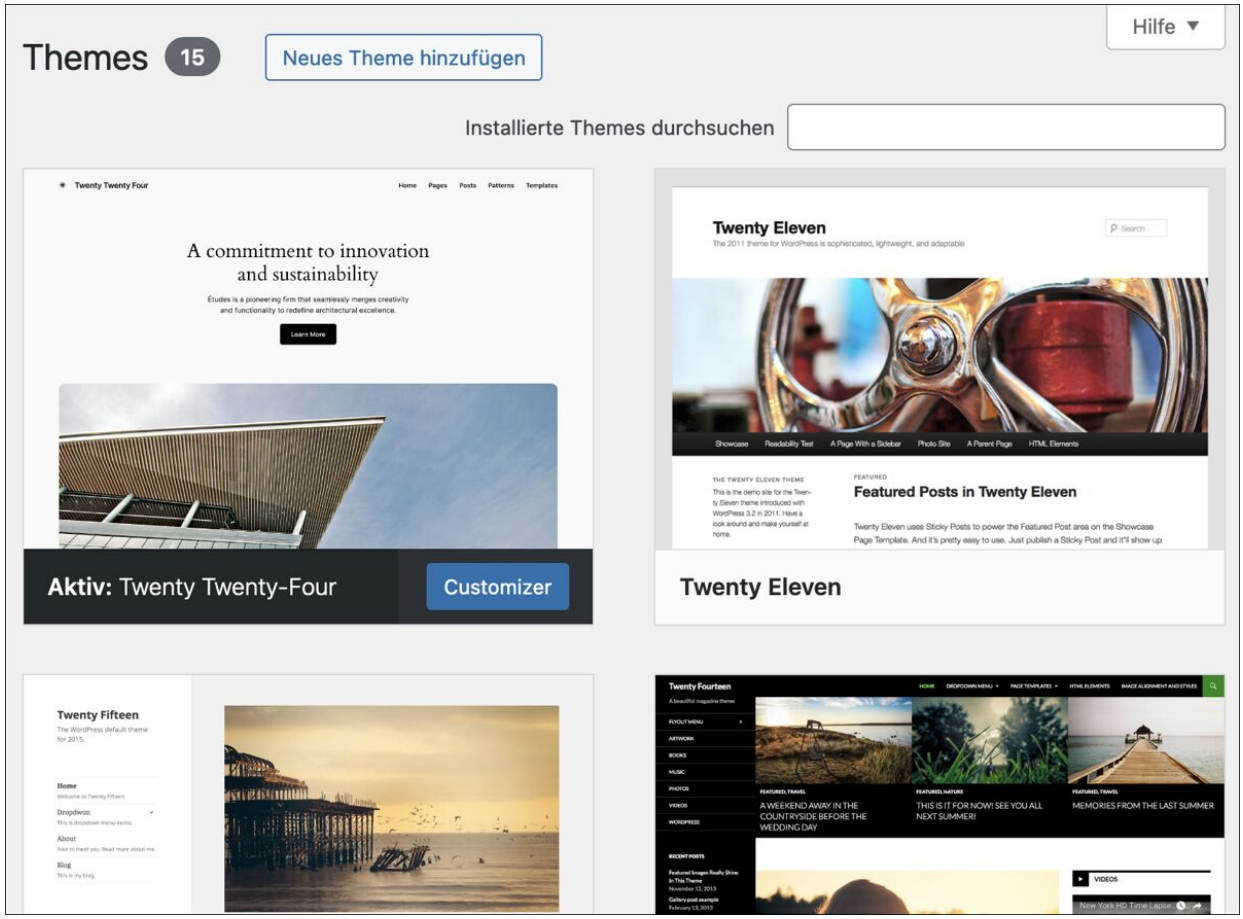


Abbildung 5.15 Mit Themes können Sie das Layout Ihrer Website wechseln.



Abbildung 5.16 Im Customizer können Sie Classic Themes mit einer Live-Vorschau anpassen.

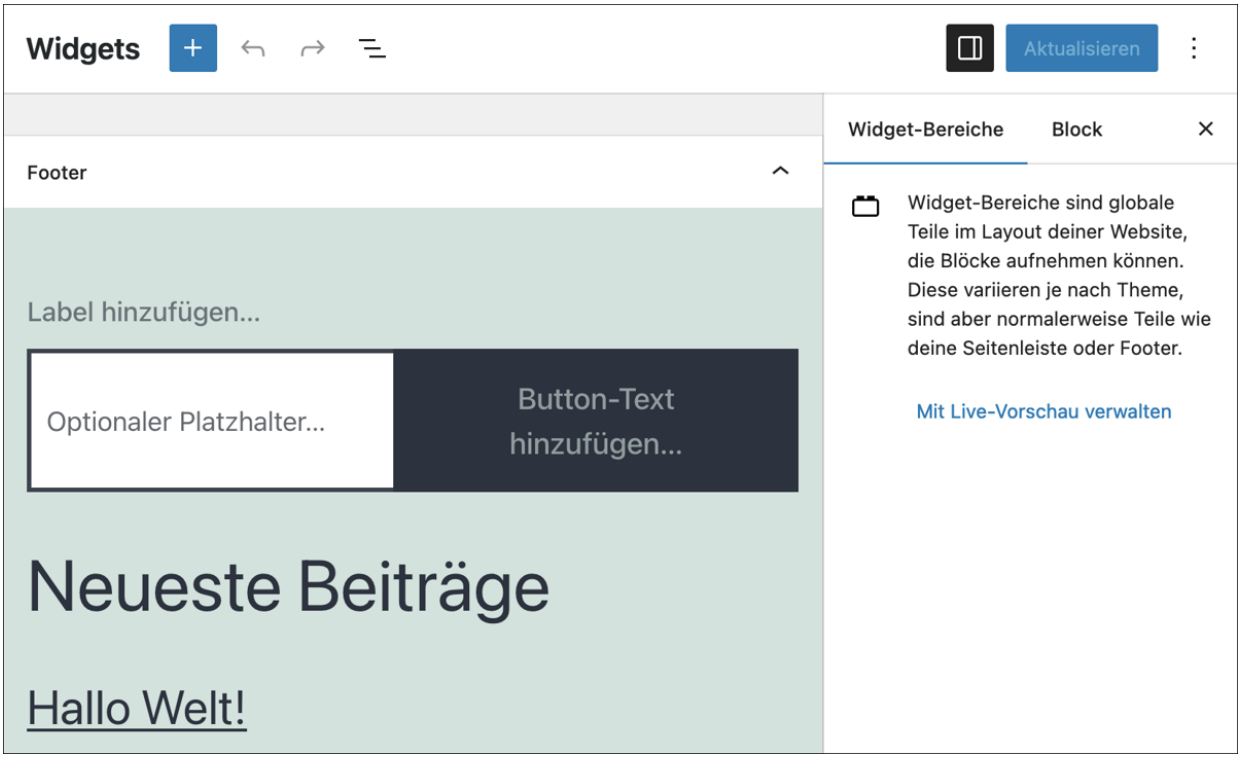


Abbildung 5.17 Seit WordPress 5.8 werden Widget-Bereiche auch mit dem Block-Editor bearbeitet, mit dem Sie seit WordPress 5 Inhalte für Seiten und Beiträge anlegen. Abhängig vom Theme gibt es hier mehrere Bereiche, in die Sie Blöcke einfügen können.

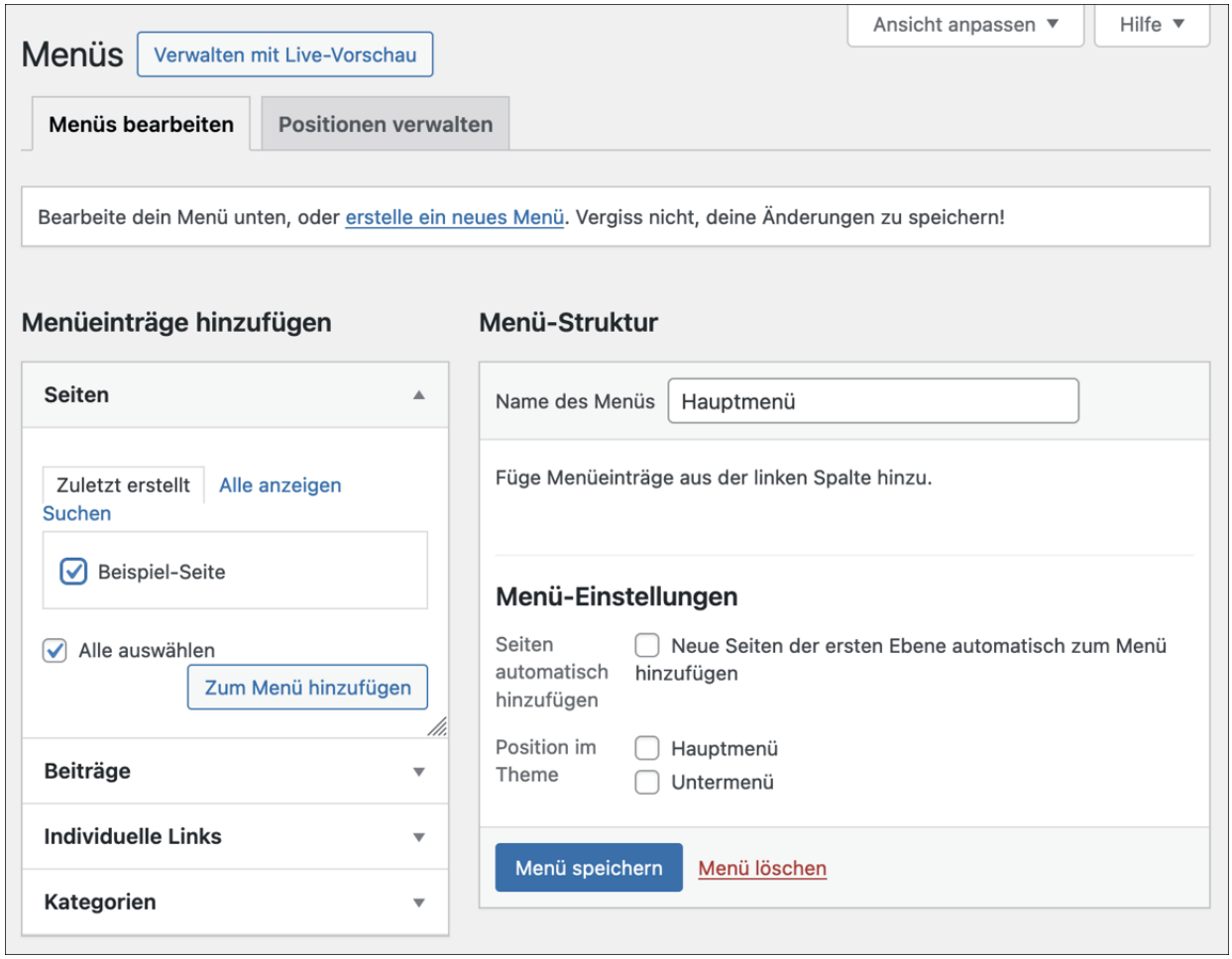


Abbildung 5.18 Bei Classic Themes werden Menüs über eine eigene Einstellungsseite angelegt und verwaltet.

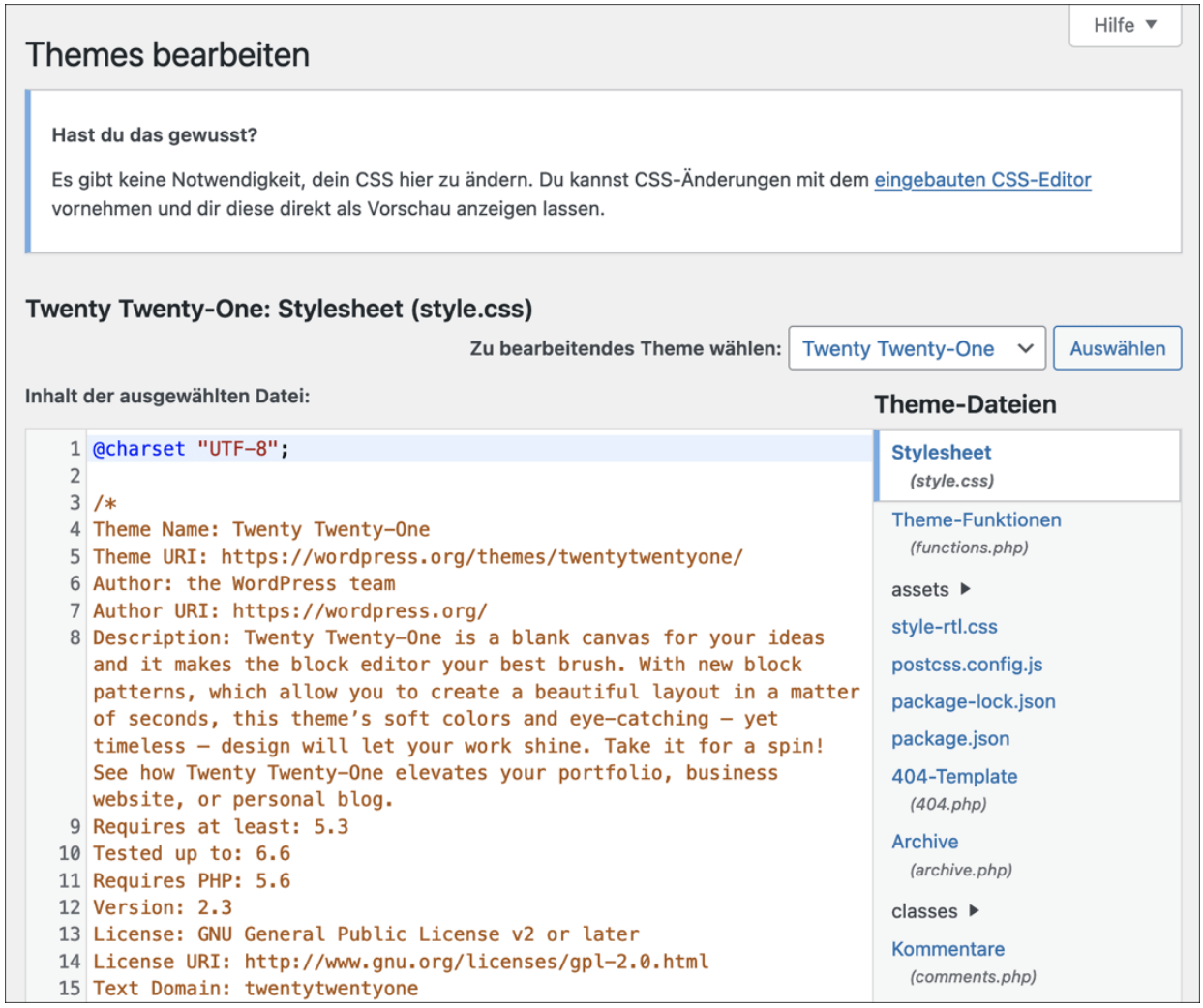


Abbildung 5.19 Für Notfälle kann der Theme-Datei-Editor mal ganz nützlich sein, aber er ist mit Vorsicht zu genießen (sonst führt er selbst zu einem Notfall).

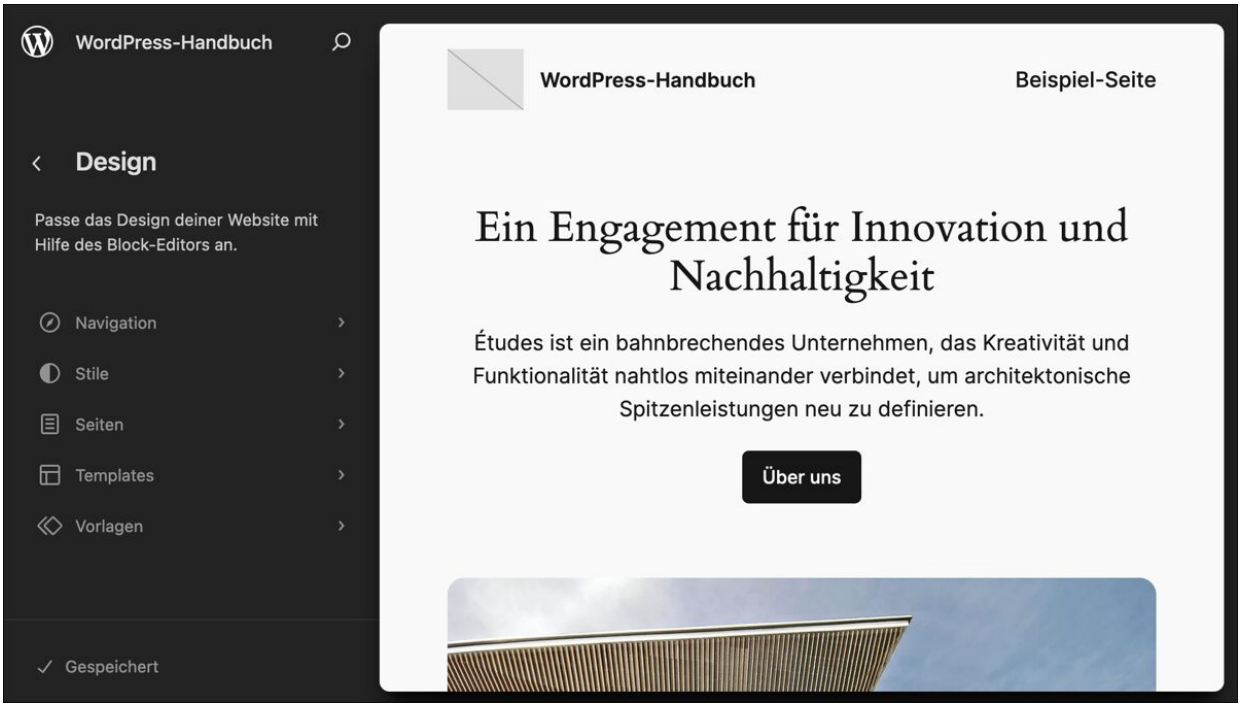


Abbildung 5.20 Im »Website-Editor« können Sie das gesamte Layout eines Block-Themes anpassen.

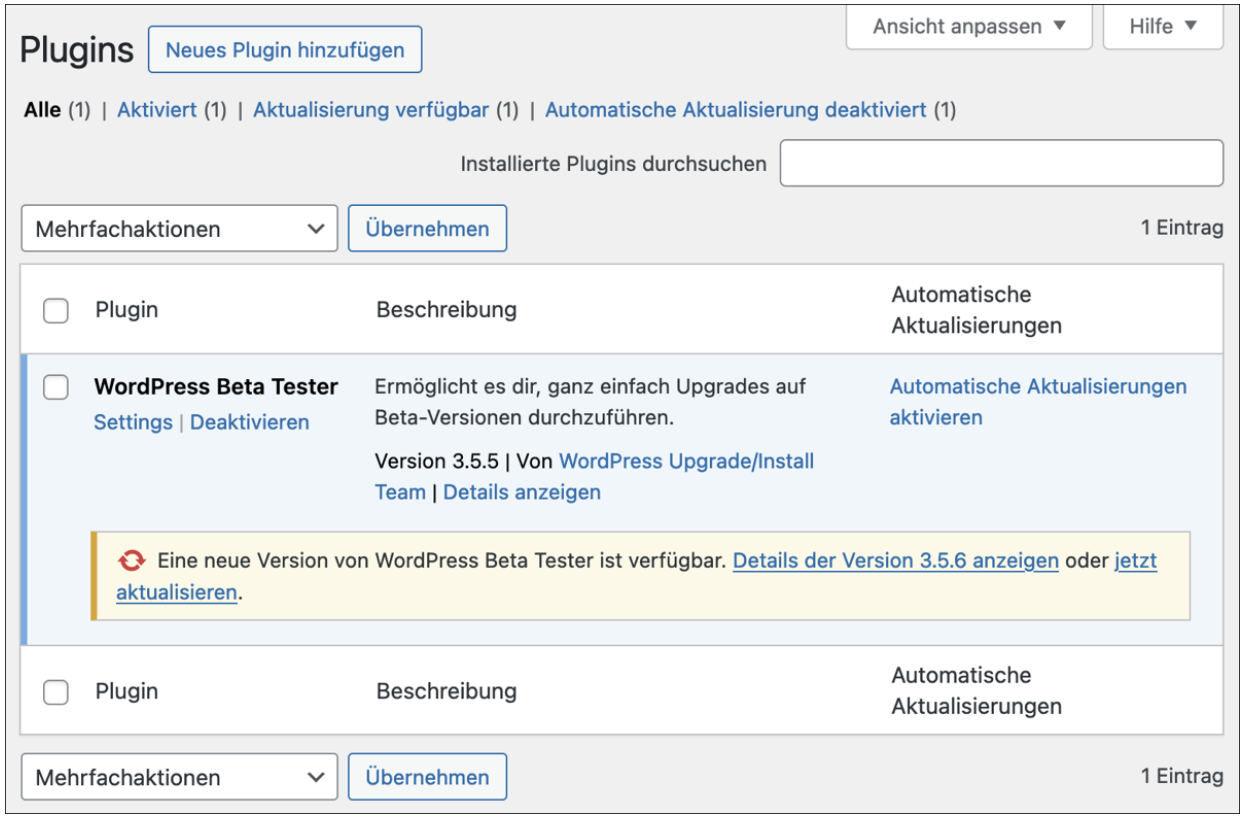


Abbildung 5.21 Über Plugins können Sie WordPress um weitere Funktionen erweitern. In dieser Liste sehen Sie alle installierten Plugins.

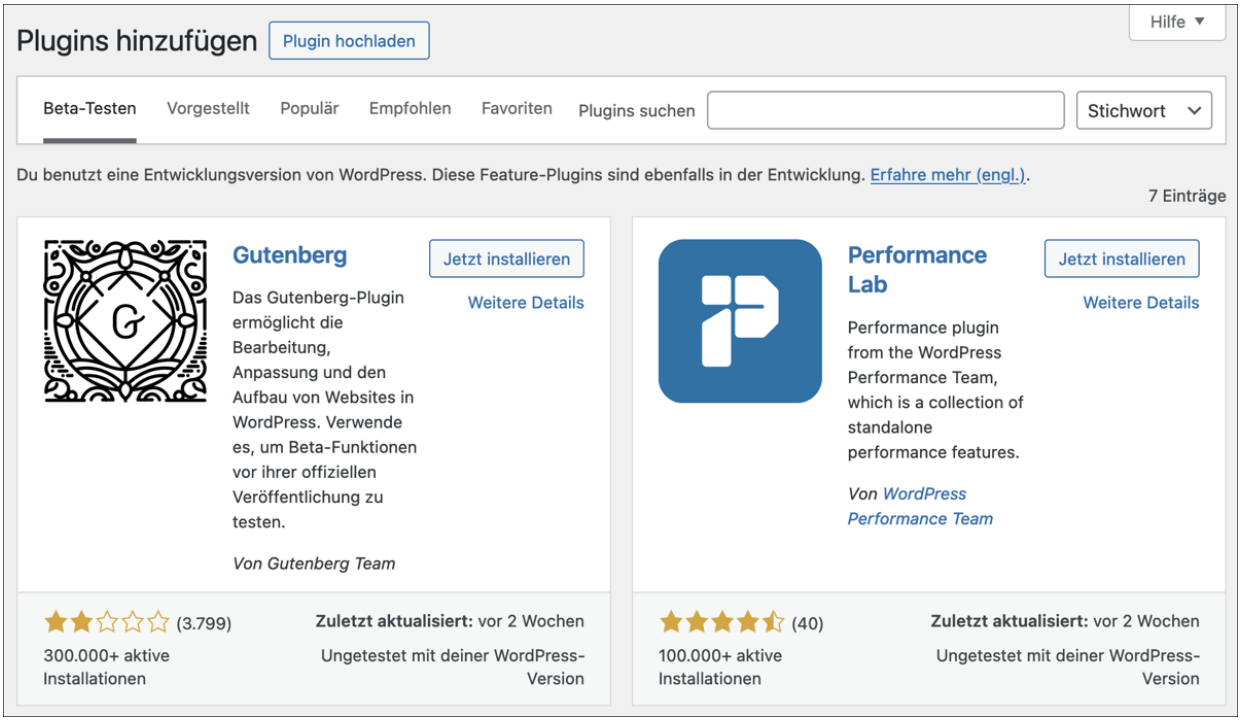


Abbildung 5.22 Eine der Stärken von WordPress ist die große Anzahl an Plugins, um den Funktionsumfang zu erweitern. Installieren Sie am besten aber nicht einfach blind irgendwelche Plugins, sondern stellen Sie sicher, dass ein Plugin wirklich das macht, was Sie brauchen, und dass es noch gepflegt wird.

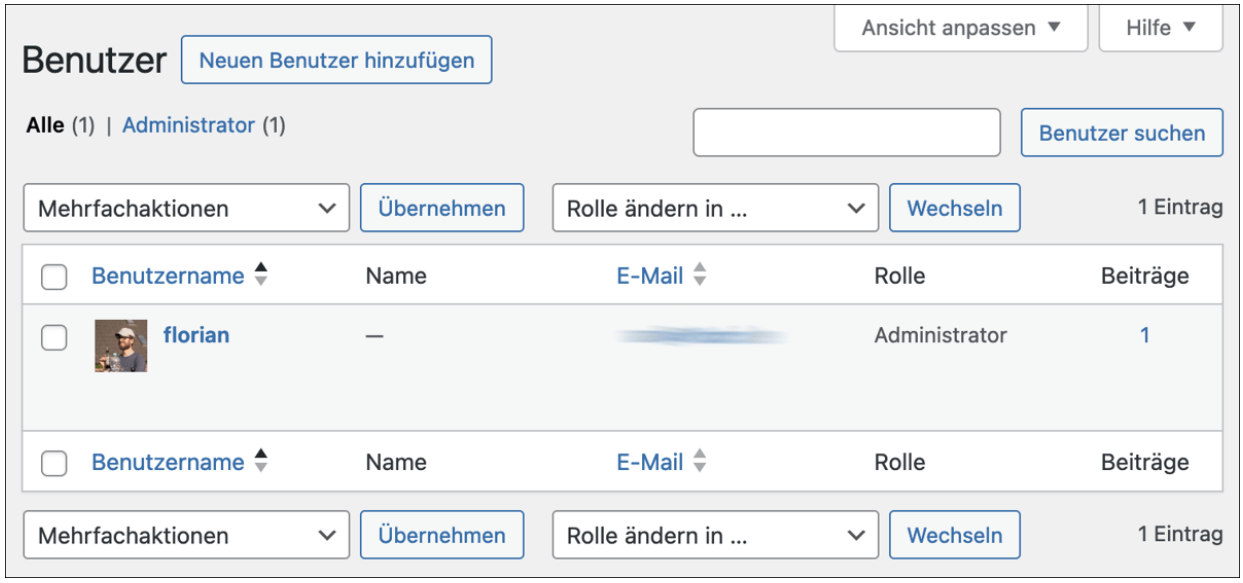


Abbildung 5.23 Auf einer frischen Installation wird in der Liste der registrierten Accounts nur Ihr eigener auftauchen, den Sie bei der Installation erstellt haben. Falls Sie den Dienst »Gravatar« nutzen und Ihre E-Mail-Adresse hinterlegt ist, die bei der Installation angegeben wurde, wird Ihr Profilbild anstelle eines generischen Avatars angezeigt.

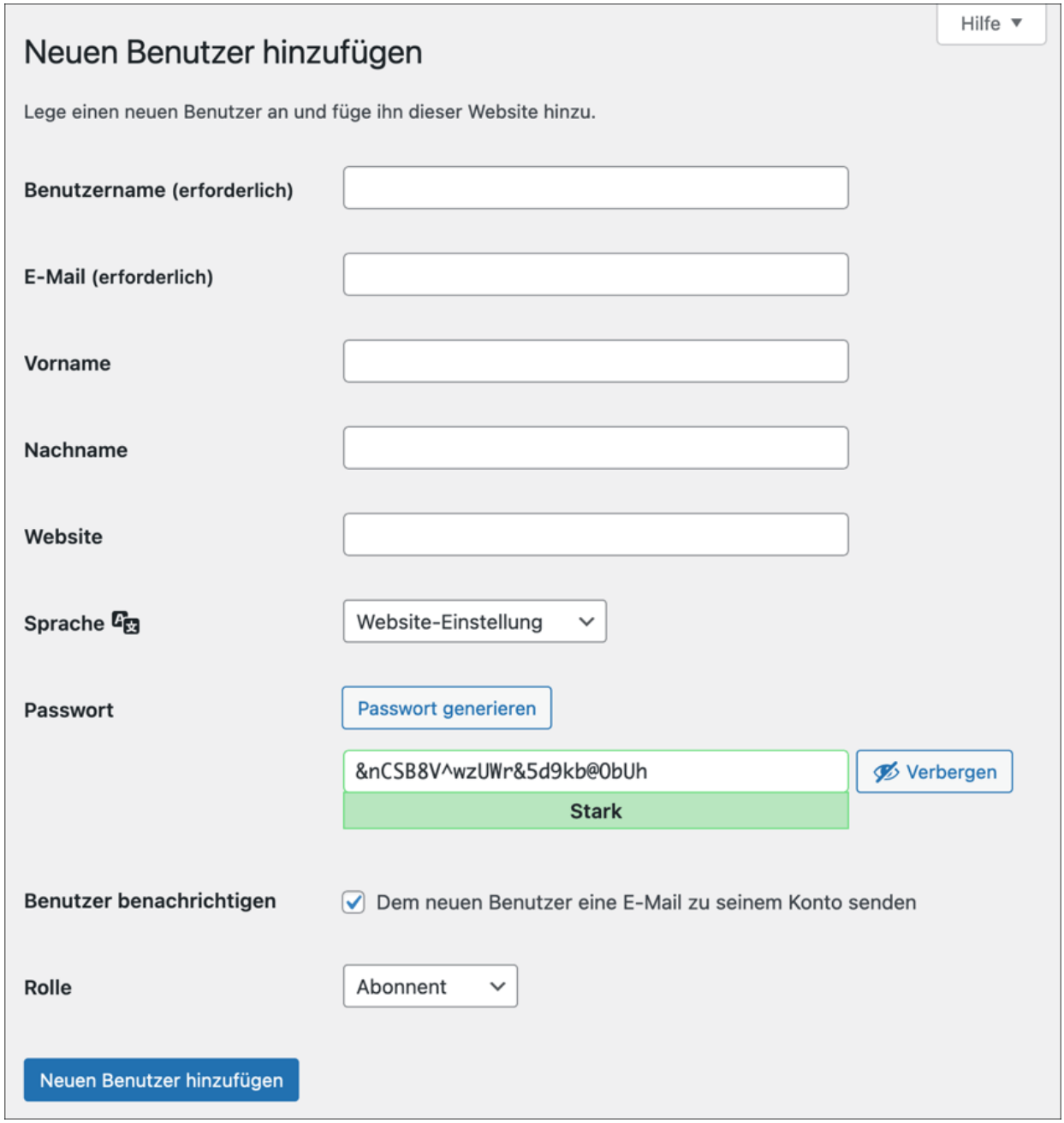
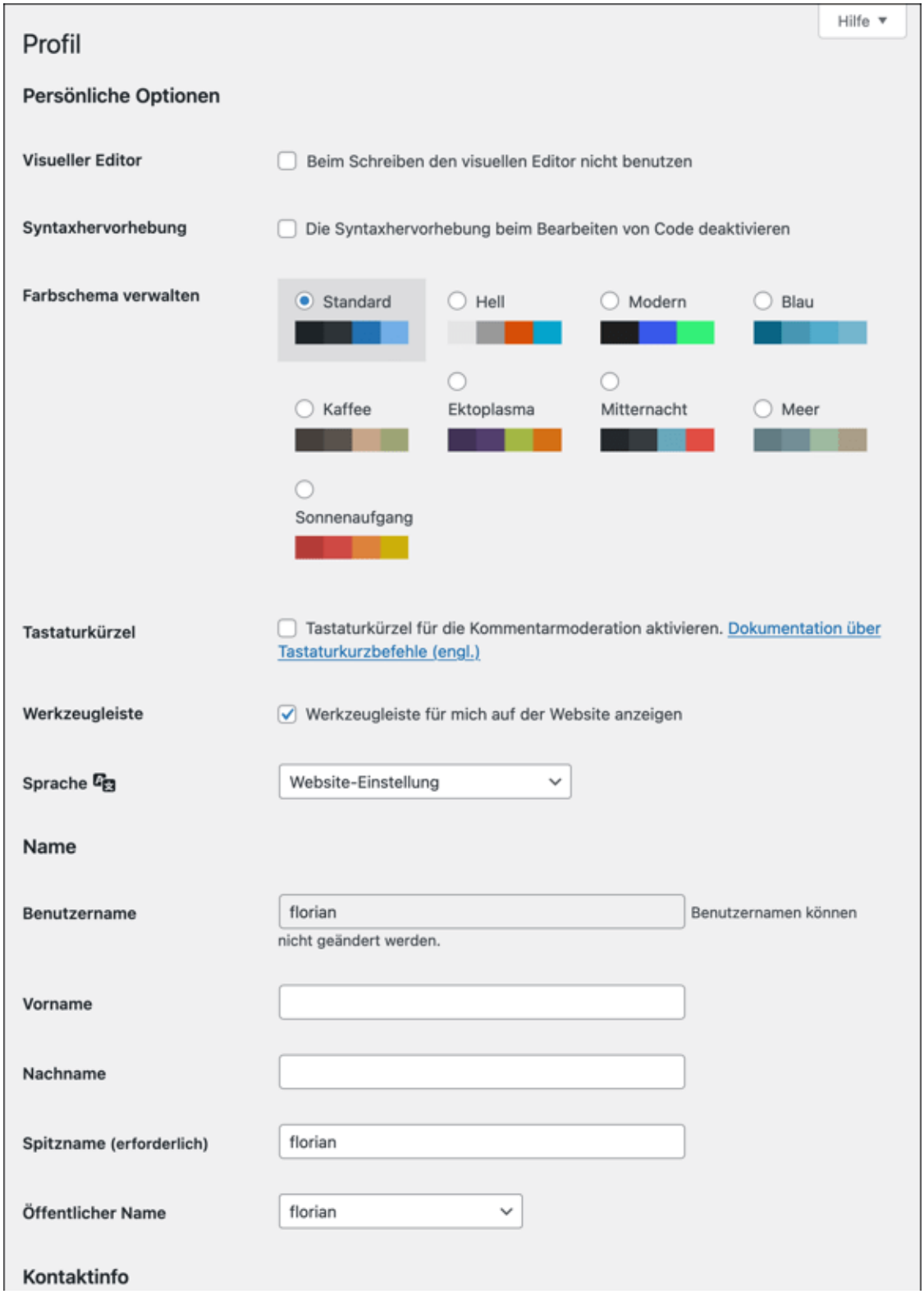


Abbildung 5.24 Wenn Sie mit anderen Personen an der Website arbeiten, können Sie unter »Benutzer« • »Neuen Benutzer hinzufügen« Accounts für diese Personen anlegen.


Hilfe
Profil
Persönliche Optionen
Visueller Editor
Beim Schreiben den visuellen Editor nicht benutzen
Syntaxhervorhebung
Die Syntaxhervorhebung beim Bearbeiten von Code deaktivieren
Farbschema verwalten
Standard
Hell
Modern
Blau
Kaffee
Ektoplasma
Mitternacht
Meer
Sonnenaufgang
Tastaturkürzel
Tastaturkürzel für die Kommentarmoderation aktivieren. Dokumentation über Tastaturkurzbefehle (engl.)
Werkzeugleiste
Werkzeugleiste für mich auf der Website anzeigen
Sprache
Website-Einstellung
Name
Benutzername
florian
Benutzernamen können nicht geändert werden.
Vorname
Nachname
Spitzname (erforderlich)
florian
Öffentlicher Name
florian
Kontaktinfo

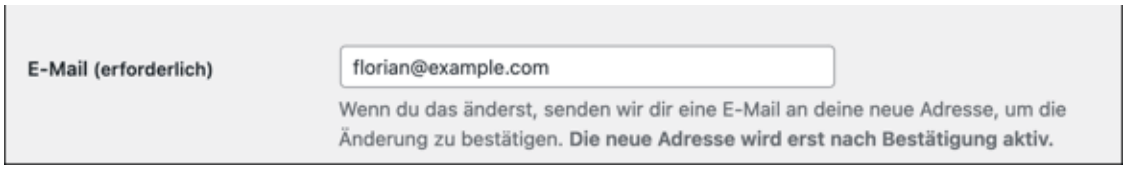


Abbildung 5.25 Unter »Benutzer« • »Profil« können Sie Ihr eigenes Profil bearbeiten.

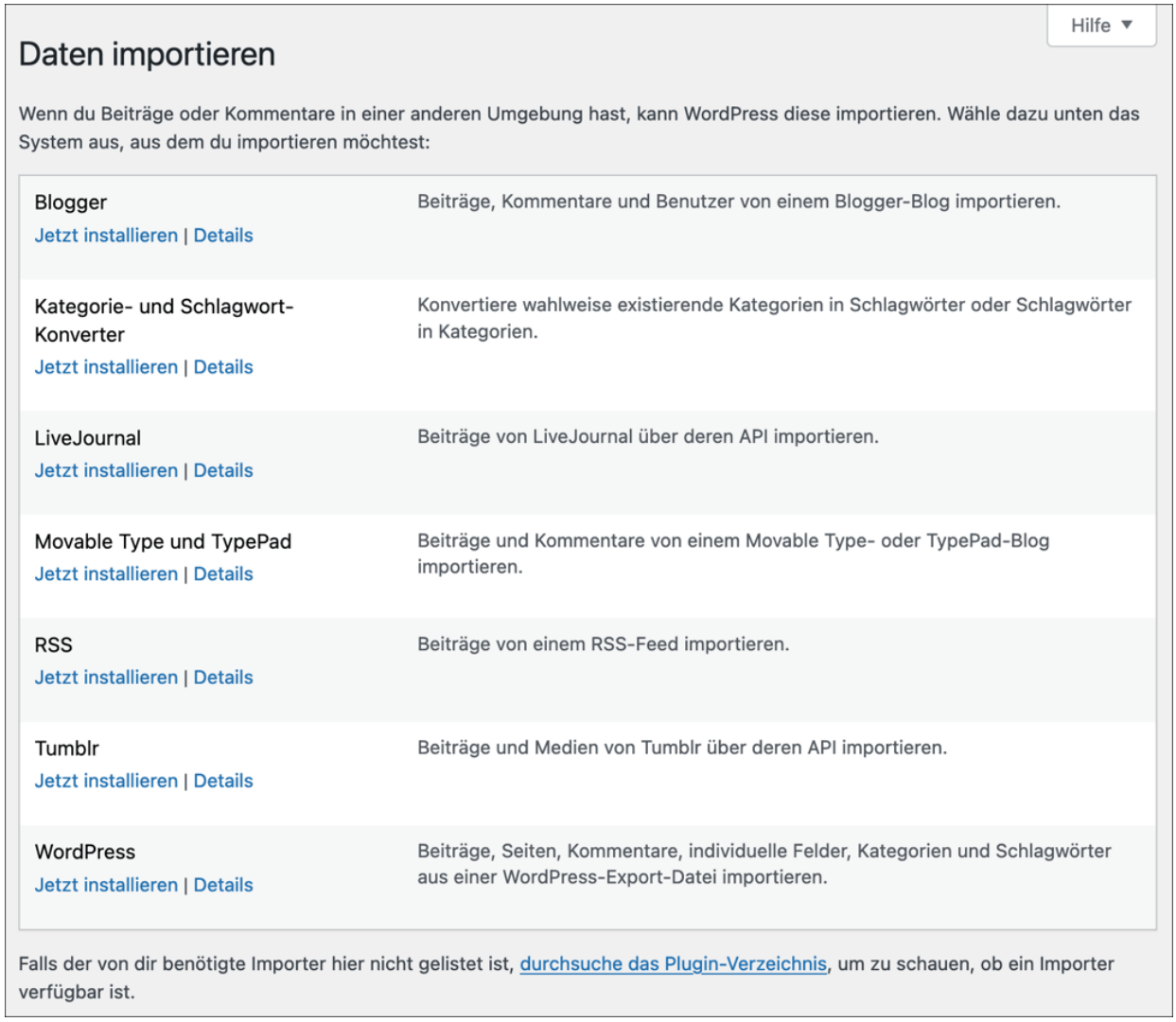


Abbildung 5.26 Zum Import von Inhalten anderer Plattformen gibt es ein paar offizielle Plugins.

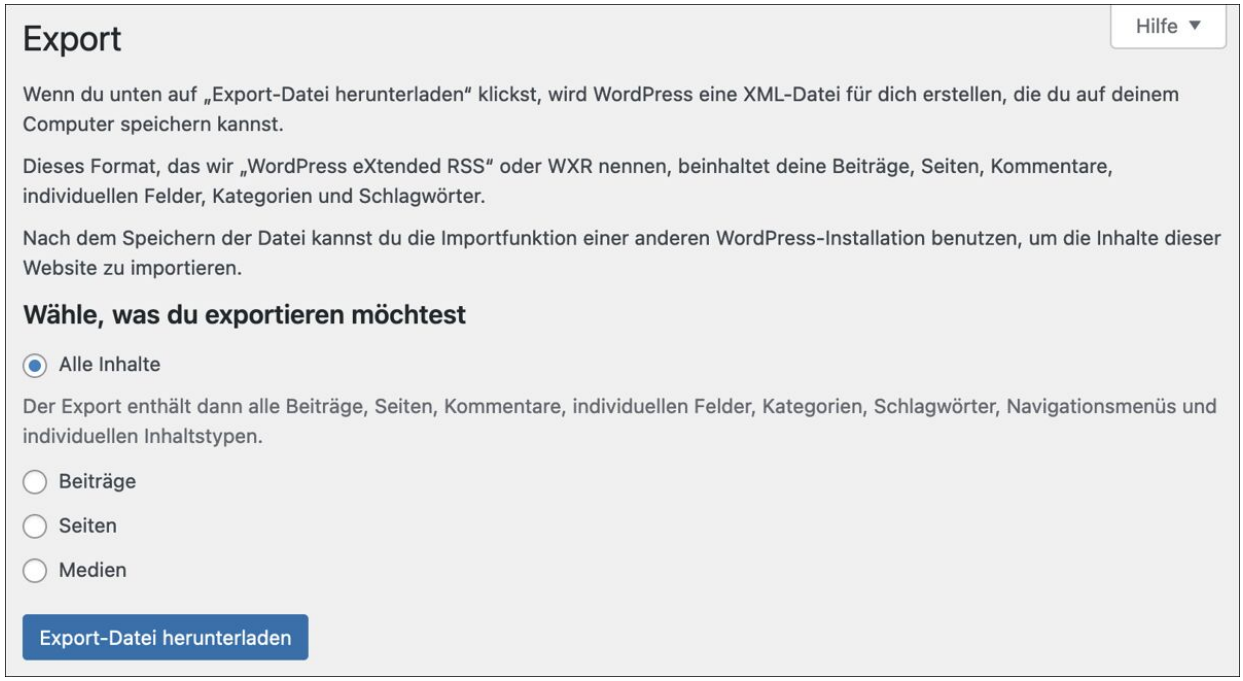


Abbildung 5.27 Hier können Sie Ihre Daten exportieren.

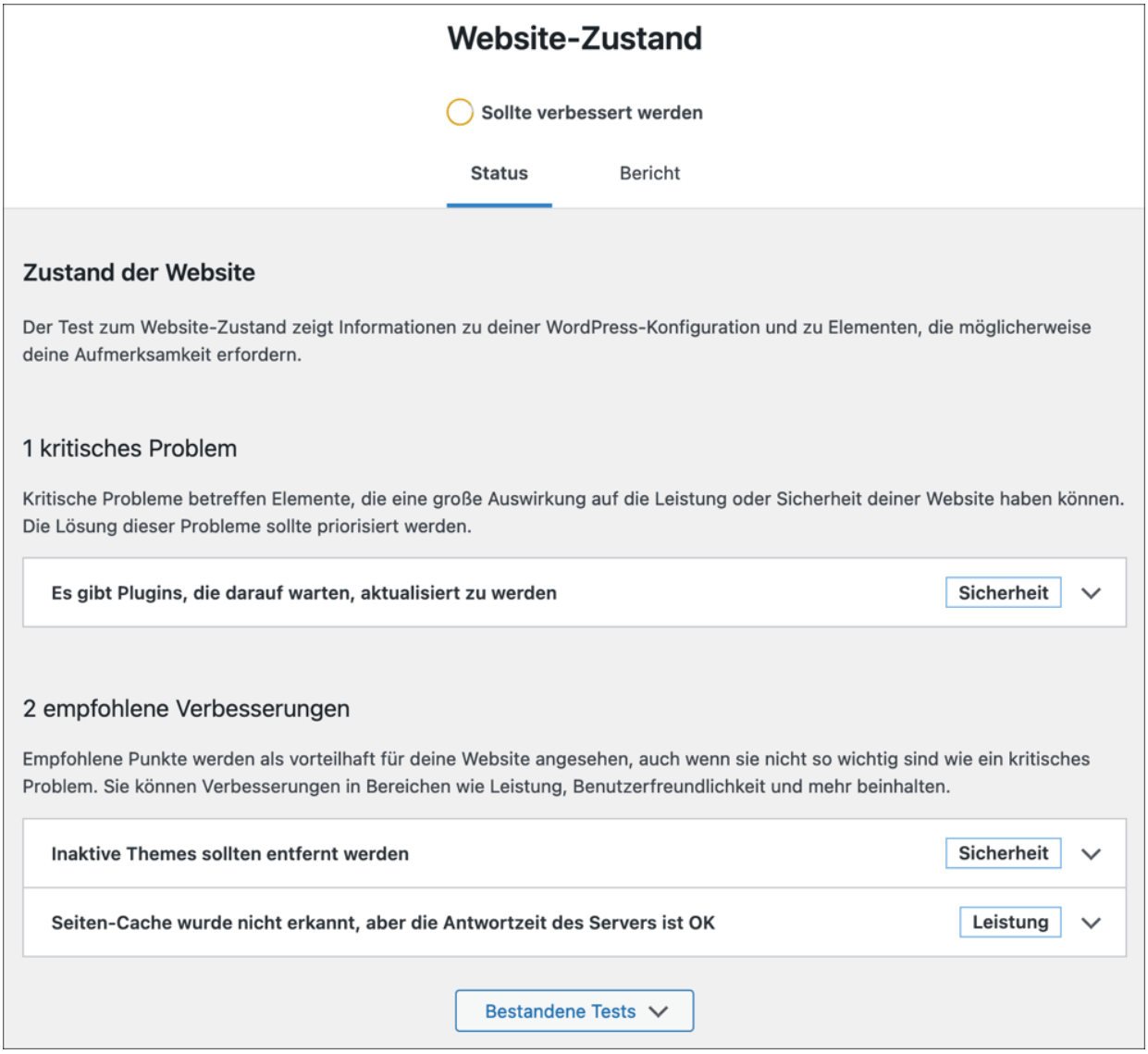
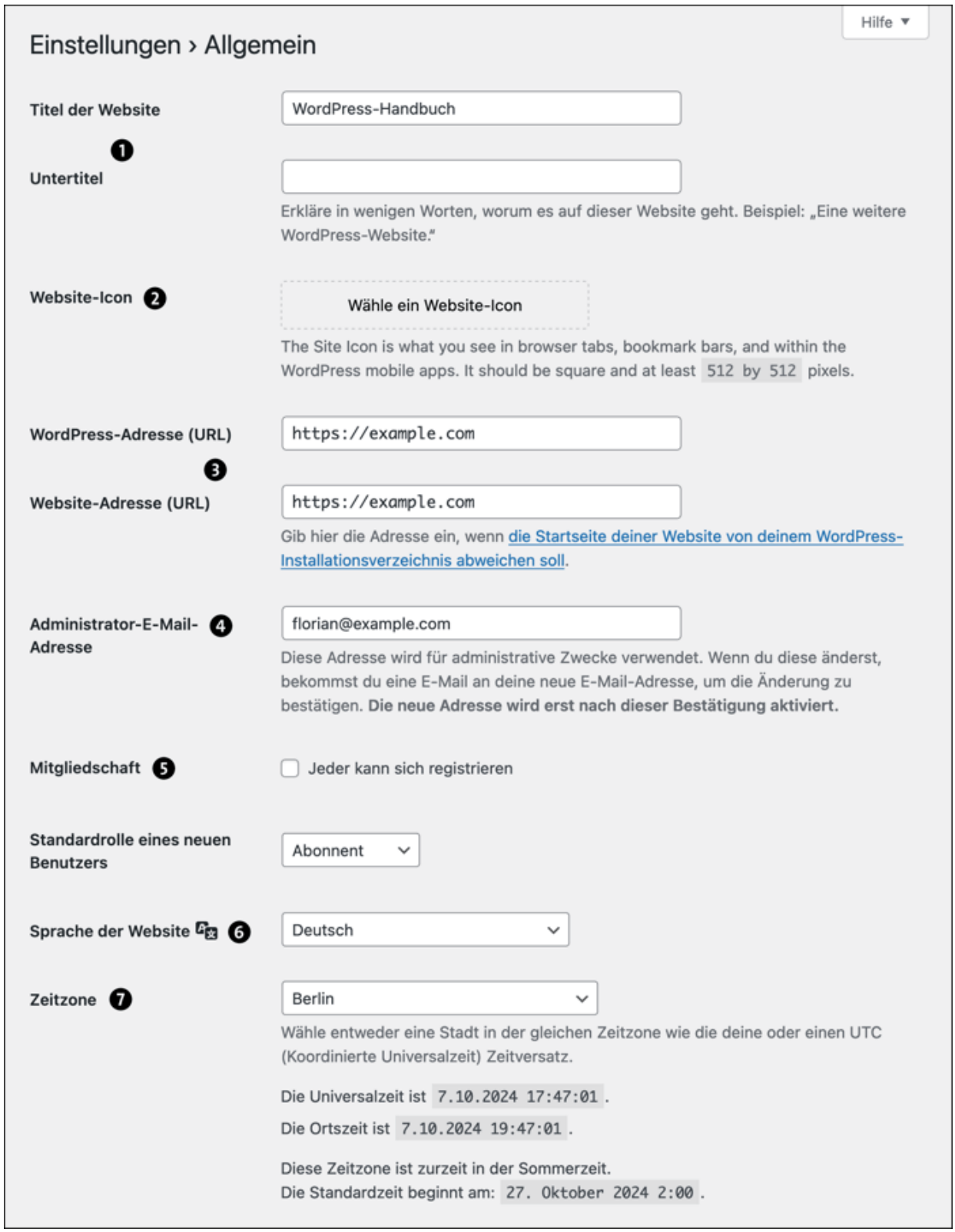


Abbildung 5.28 Das »Website-Zustand«-Werkzeug ist ein nützliches Helferlein, um Ihnen Hinweise zu geben, ob an Ihrer WordPress-Installation und/oder dem Webserver etwas verändert werden könnte.


Hilfe
Einstellungen › Allgemein
Titel der Website
WordPress-Handbuch
1
Untertitel
Erkläre in wenigen Worten, worum es auf dieser Website geht. Beispiel: „Eine weitere WordPress-Website."
Website-Icon 2
Wähle ein Website-Icon
The Site Icon is what you see in browser tabs, bookmark bars, and within the WordPress mobile apps. It should be square and at least 512 by 512 pixels.
WordPress-Adresse (URL)
https://example.com
3
Website-Adresse (URL)
https://example.com
Gib hier die Adresse ein, wenn die Startseite deiner Website von deinem WordPress-Installationsverzeichnis abweichen soll.
Administrator-E-Mail-Adresse 4
florian@example.com
Diese Adresse wird für administrative Zwecke verwendet. Wenn du diese änderst, bekommst du eine E-Mail an deine neue E-Mail-Adresse, um die Änderung zu bestätigen. Die neue Adresse wird erst nach dieser Bestätigung aktiviert.
Mitgliedschaft 5
Jeder kann sich registrieren
Standardrolle eines neuen Benutzers
Abonnent
Sprache der Website 6
Deutsch
Zeitzone 7
Berlin
Wähle entweder eine Stadt in der gleichen Zeitzone wie die deine oder einen UTC (Koordinierte Universalzeit) Zeitversatz.
Die Universalzeit ist 7.10.2024 17:47:01 .
Die Ortszeit ist 7.10.2024 19:47:01 .
Diese Zeitzone ist zurzeit in der Sommerzeit.
Die Standardzeit beginnt am: 27. Oktober 2024 2:00 .

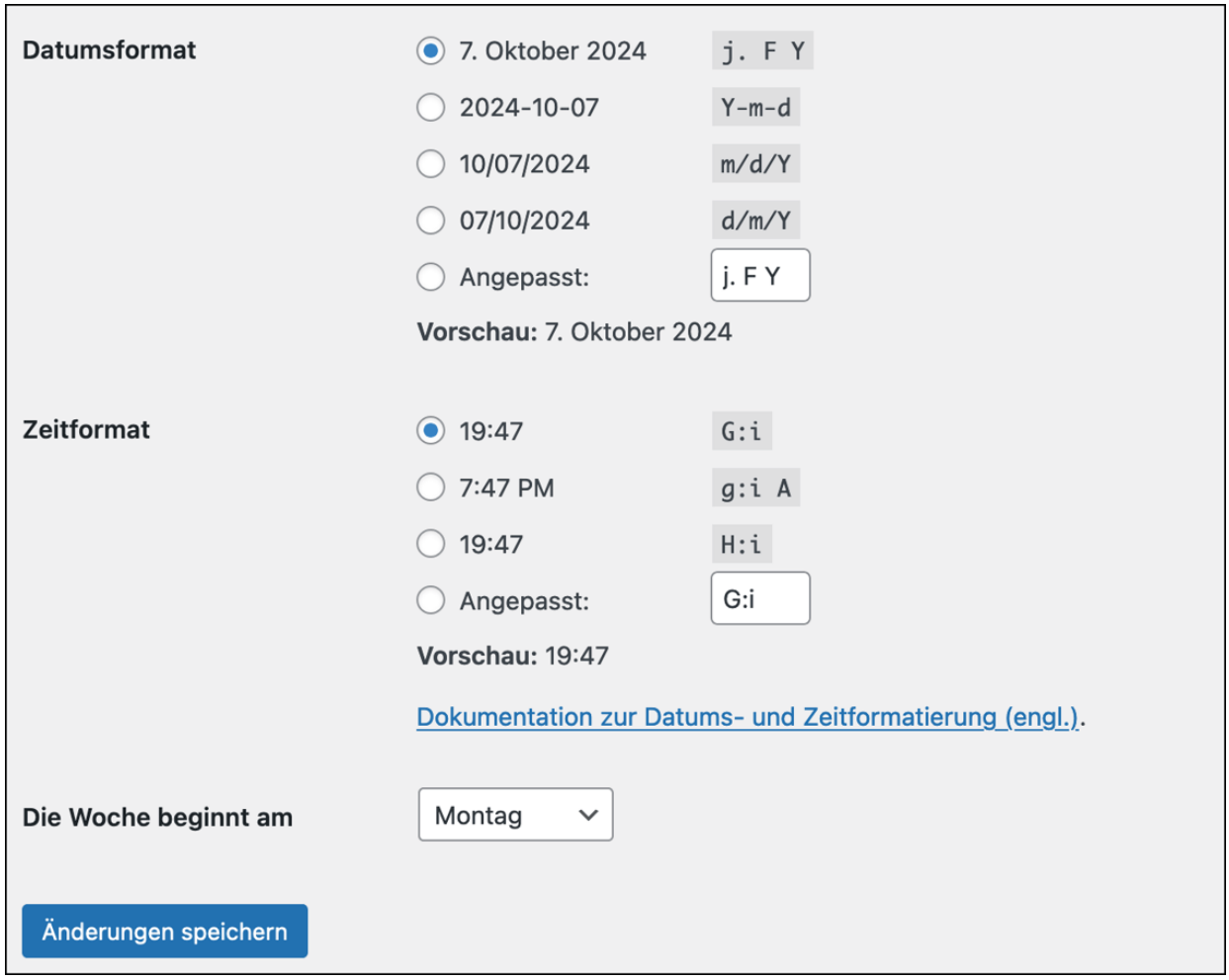


Abbildung 5.29 Auf der »Allgemein«-Einstellungsseite werden wichtige Einstellungen vorgenommen, etwa der Website-Titel.

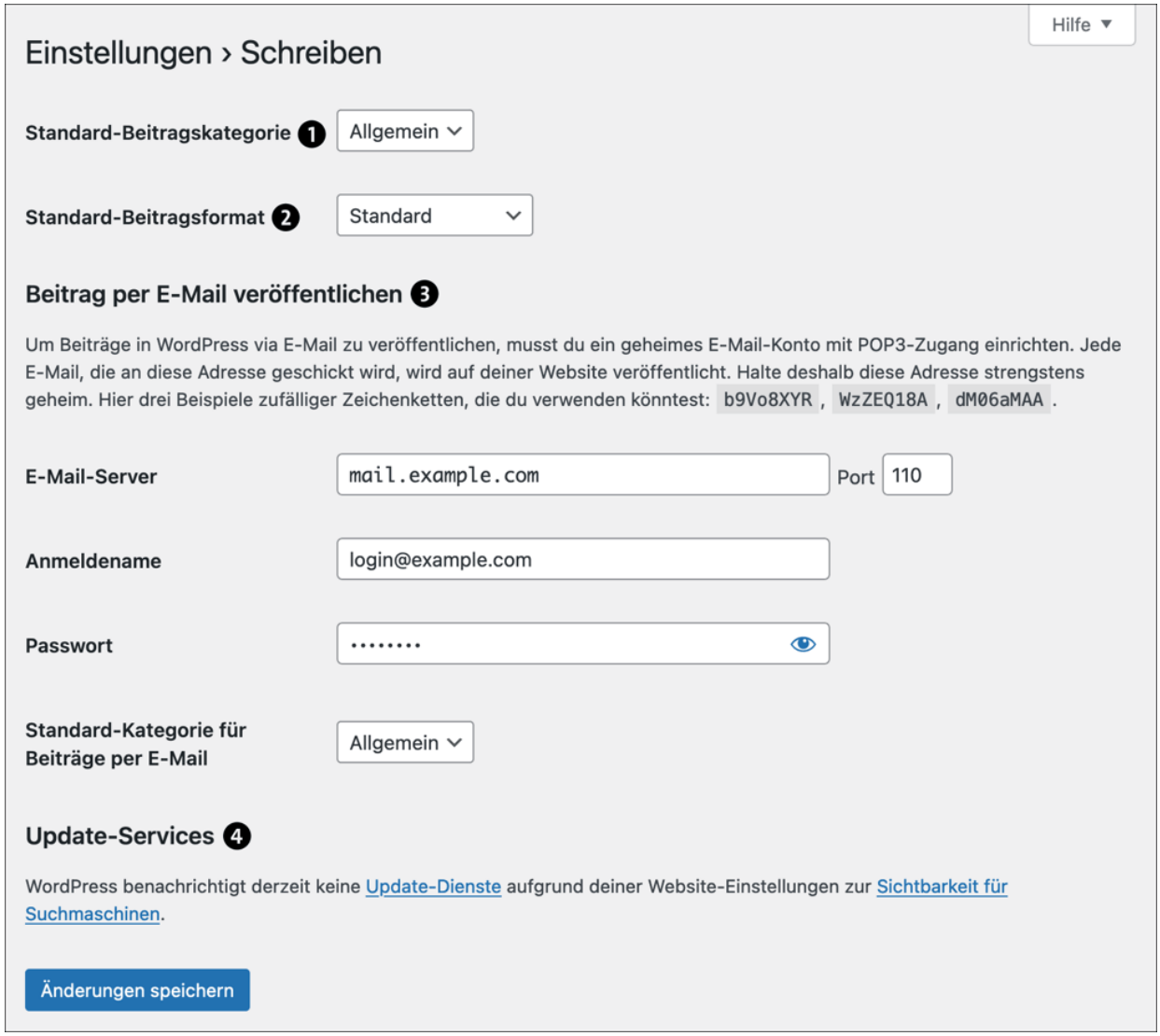


Abbildung 5.30 Die Einstellungsseite »Schreiben« wird meist vermutlich nur zur Kenntnis genommen, ohne Änderungen zu machen.

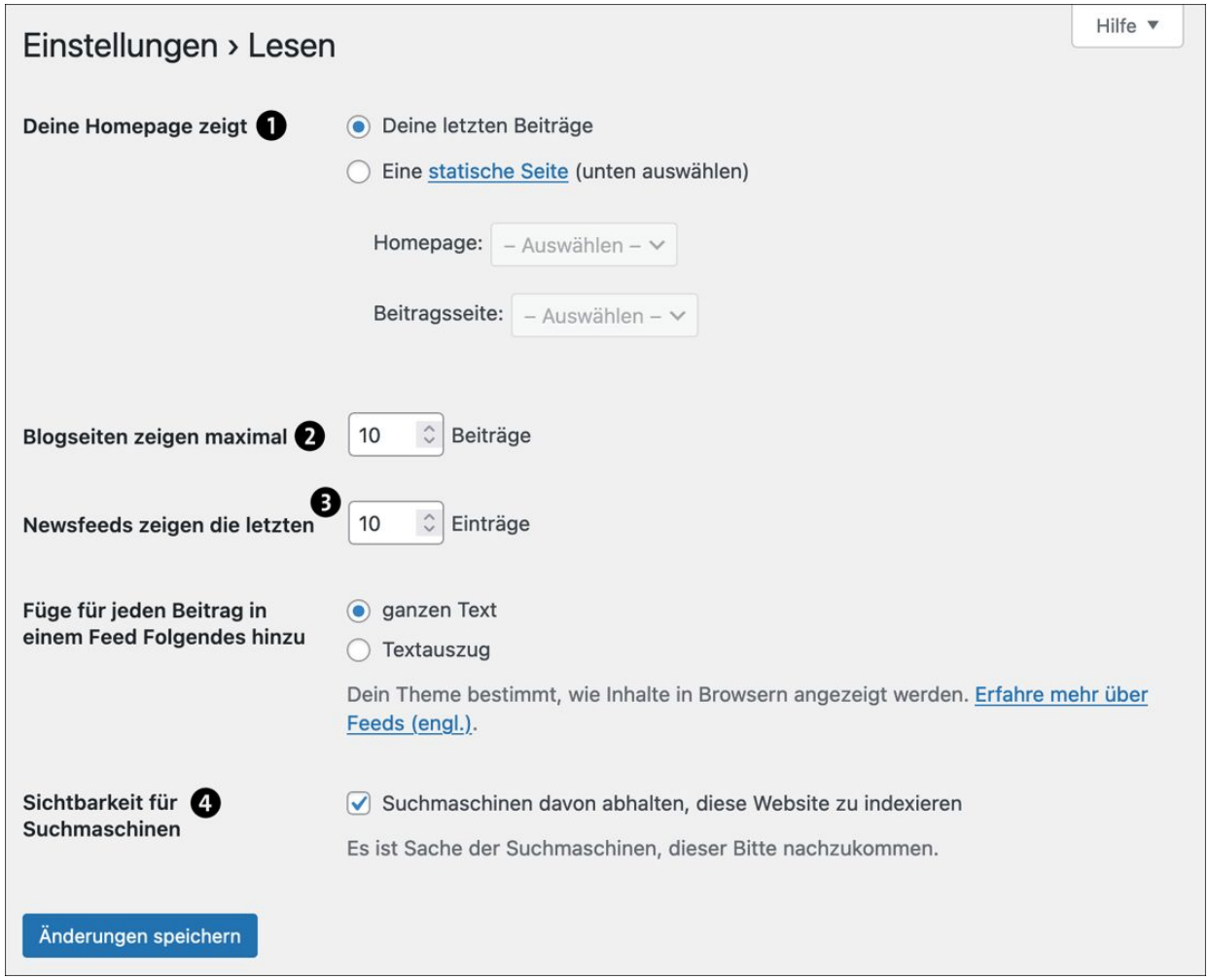
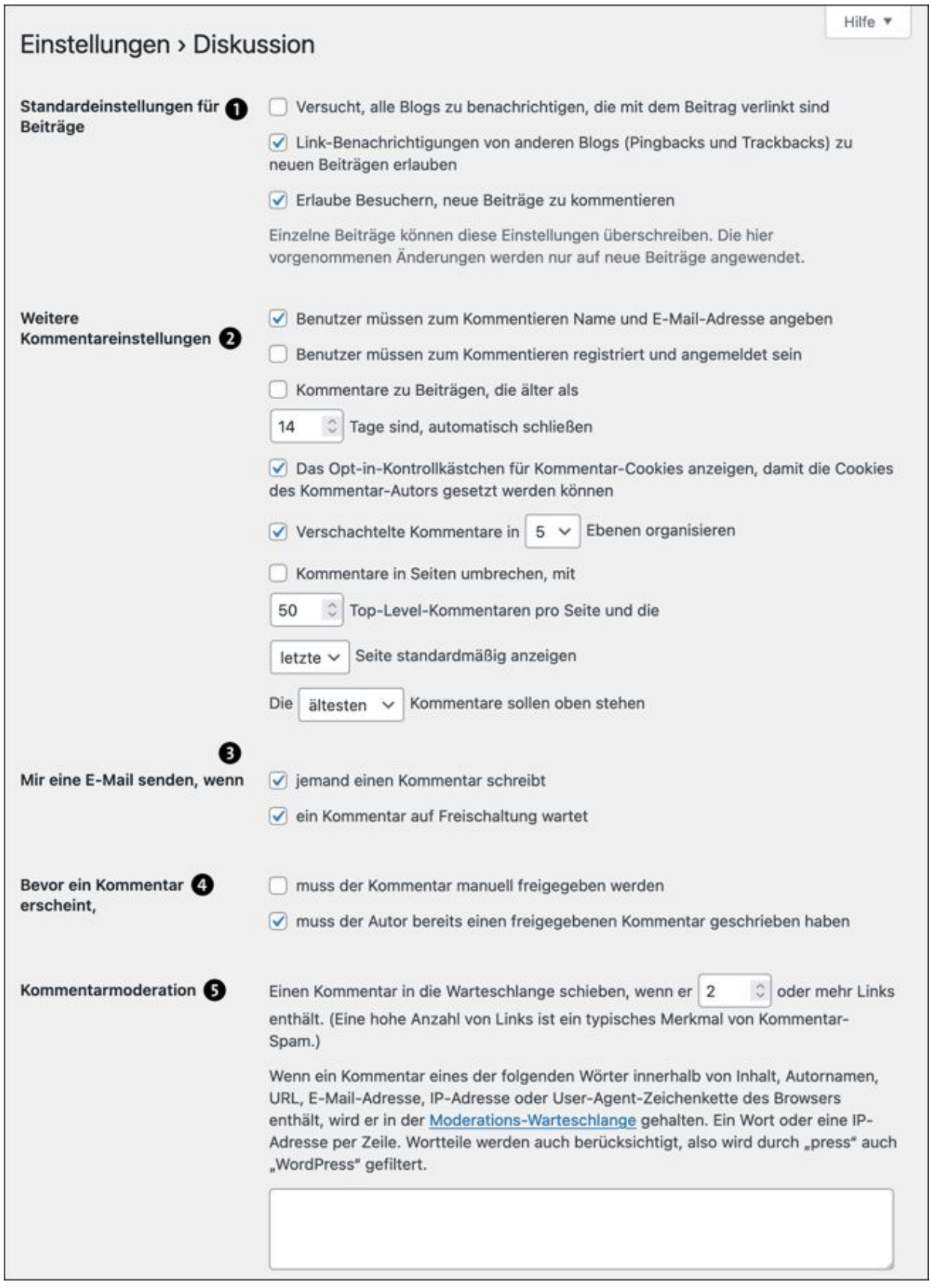


Abbildung 5.31 Bei den »Lesen«-Einstellungen können Sie unter anderem festlegen, ob die Startseite die letzten Beiträge oder eine statische Seite anzeigt.


Hilfe
Einstellungen › Diskussion
Standardeinstellungen für Beiträge 1
Versucht, alle Blogs zu benachrichtigen, die mit dem Beitrag verlinkt sind
Link-Benachrichtigungen von anderen Blogs (Pingbacks und Trackbacks) zu neuen Beiträgen erlauben
Erlaube Besuchern, neue Beiträge zu kommentieren
Einzelne Beiträge können diese Einstellungen überschreiben. Die hier vorgenommenen Änderungen werden nur auf neue Beiträge angewendet.
Weitere Kommentareinstellungen 2
Benutzer müssen zum Kommentieren Name und E-Mail-Adresse angeben
Benutzer müssen zum Kommentieren registriert und angemeldet sein
Kommentare zu Beiträgen, die älter als
14 Tage sind, automatisch schließen
Das Opt-in-Kontrollkästchen für Kommentar-Cookies anzeigen, damit die Cookies des Kommentar-Autors gesetzt werden können
Verschachtelte Kommentare in 5 Ebenen organisieren
Kommentare in Seiten umbrechen, mit
50 Top-Level-Kommentaren pro Seite und die
letzte Seite standardmäßig anzeigen
Die ältesten Kommentare sollen oben stehen
3
Mir eine E-Mail senden, wenn
jemand einen Kommentar schreibt
ein Kommentar auf Freischaltung wartet
Bevor ein Kommentar erscheint, 4
muss der Kommentar manuell freigegeben werden
muss der Autor bereits einen freigegebenen Kommentar geschrieben haben
Kommentarmoderation 5
Einen Kommentar in die Warteschlange schieben, wenn er 2 oder mehr Links enthält. (Eine hohe Anzahl von Links ist ein typisches Merkmal von Kommentar-Spam.)
Wenn ein Kommentar eines der folgenden Wörter innerhalb von Inhalt, Autornamen, URL, E-Mail-Adresse, IP-Adresse oder User-Agent-Zeichenkette des Browsers enthält, wird er in der Moderations-Warteschlange gehalten. Ein Wort oder eine IP-Adresse per Zeile. Wortteile werden auch berücksichtigt, also wird durch „press“ auch „WordPress“ gefiltert.

Kommentar-Sperrliste 6

Wenn ein Kommentar eines der folgenden Wörter innerhalb von Inhalt, Autornamen, URL, E-Mail-Adresse, IP-Adresse oder User-Agent-Zeichenkette des Browsers enthält, wird er in den Papierkorb verschoben. Ein Wort oder eine IP-Adresse per Zeile. Wortteile werden auch berücksichtigt, also wird durch „press" auch „WordPress" gefiltert.

Avatare 7

Ein Avatar ist ein Bild, das einem Benutzer über mehrere Websites hinweg zugeordnet ist. In diesem Bereich kannst du wählen, ob Avatare von Benutzern, die mit dieser Website interagieren, angezeigt werden sollen.

Avataranzeige

- [x] Avatare anzeigen

Avatare anzeigen mit der Einstufung bis einschließlich

- (•) G — Jugendfrei
- ( ) PG — Entspricht dem deutschen FSK12
- ( ) R — Entspricht in etwa dem deutschen FSK18
- ( ) X — Keine Jugendfreigabe

Standard-Avatar

Für Benutzer, die keinen eigenen, individuellen Avatar haben, kann wahlweise ein allgemeines Logo oder ein auf Basis der E-Mail-Adresse erzeugter Avatar angezeigt werden.

- (•) Geheimnisvolle Person
- ( ) Kein Avatar
- ( ) Gravatar-Logo
- ( ) Identicon (automatisch generiert)
- ( ) Wavatar (automatisch generiert)
- ( ) MonsterID (automatisch generiert)
- ( ) Retro (automatisch generiert)

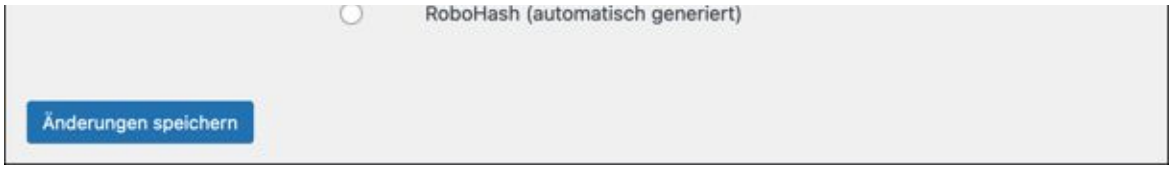


Abbildung 5.32 Auf der »Diskussion«-Einstellungsseite können Sie die Kommentarfunktion Ihrer Website konfigurieren.

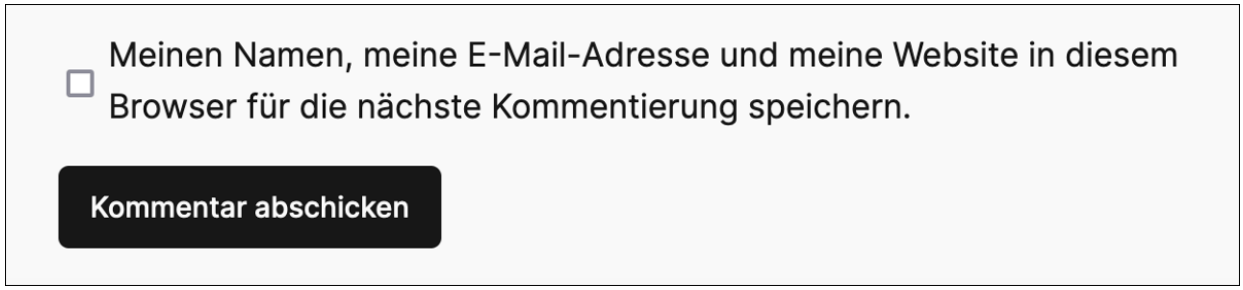


Abbildung 5.33 Wenn Personen oft Kommentare schreiben, kann es hilfreich sein, wenn Name, Mail-Adresse und Website automatisch vorausgefüllt werden.

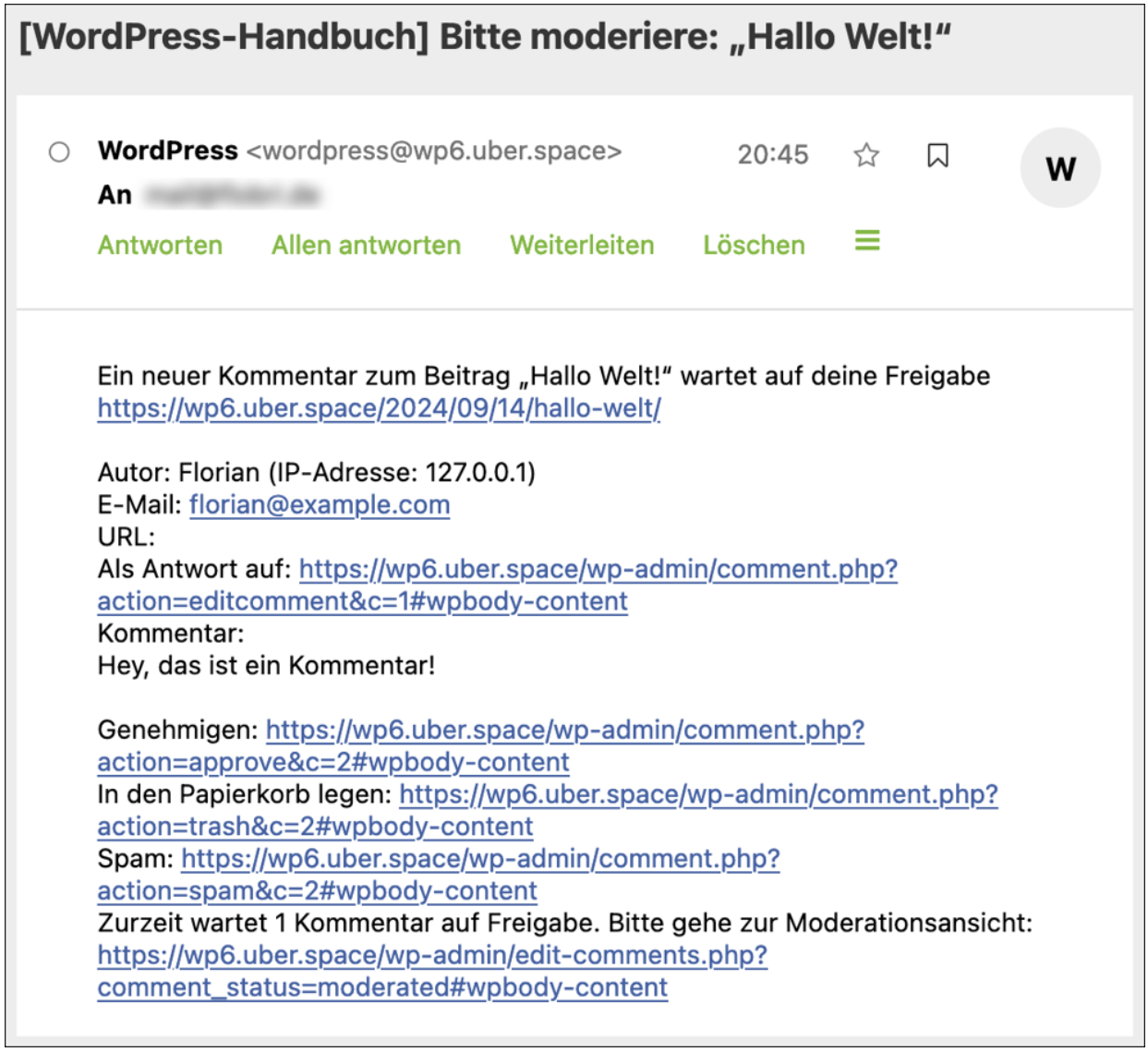


Abbildung 5.34 Wenn Sie E-Mail-Benachrichtigungen eingestellt haben, bekommen Sie bei einem neuen Kommentar eine solche E-Mail mit hilfreichen Links zum Markieren als Spam oder zum Löschen.

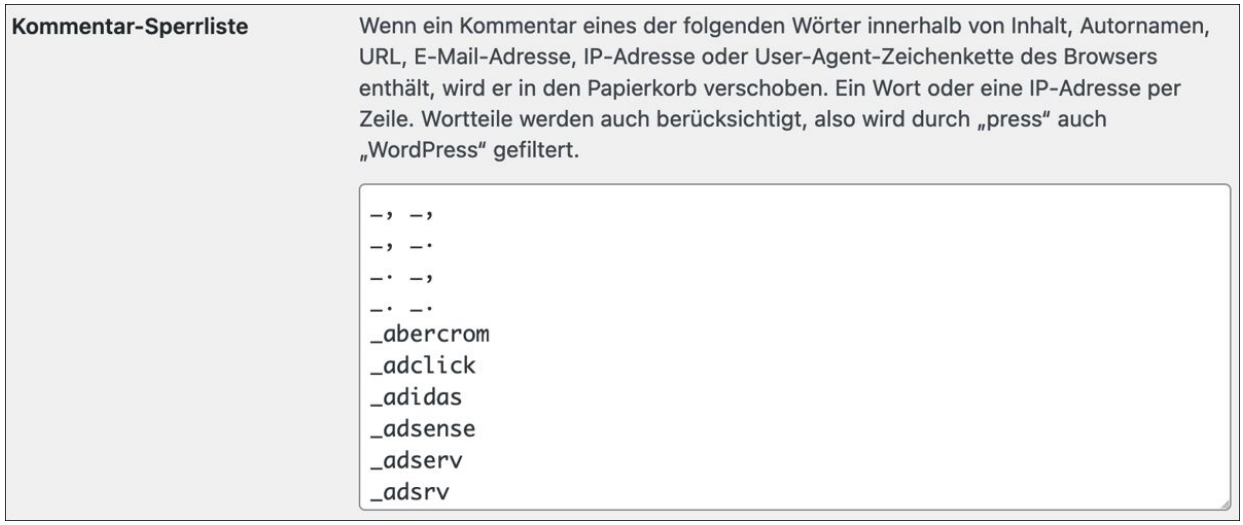


Abbildung 5.35 Die empfohlene Blocklist ist über 700 KB groß und beinhaltet entsprechend viele Begriffe und andere Zeichenketten aus aller Welt.

Avatare

Lade deine Lieblingsavatarbilder hoch – oder erstelle sie – und verbinde sie mit deiner E-Mail-Adresse.

Verknüpfe sekundäre E-Mail-Adressen mit dem Profil und lege deinen Avatar fest. Weitere Informationen.

E-Mail auswählen

florian@example.com (primär)

E-Mail-Adresse hinzufügen



Bild hochladen



Abbildung 5.36 Gravatar ist ein Dienst von Automattic, der Firma hinter WordPress.com. Daher nutzt der Dienst auch denselben Login wie WordPress.com, Sie müssen sich also für die Nutzung von Gravatar dort einen Account erstellen, falls noch nicht geschehen.

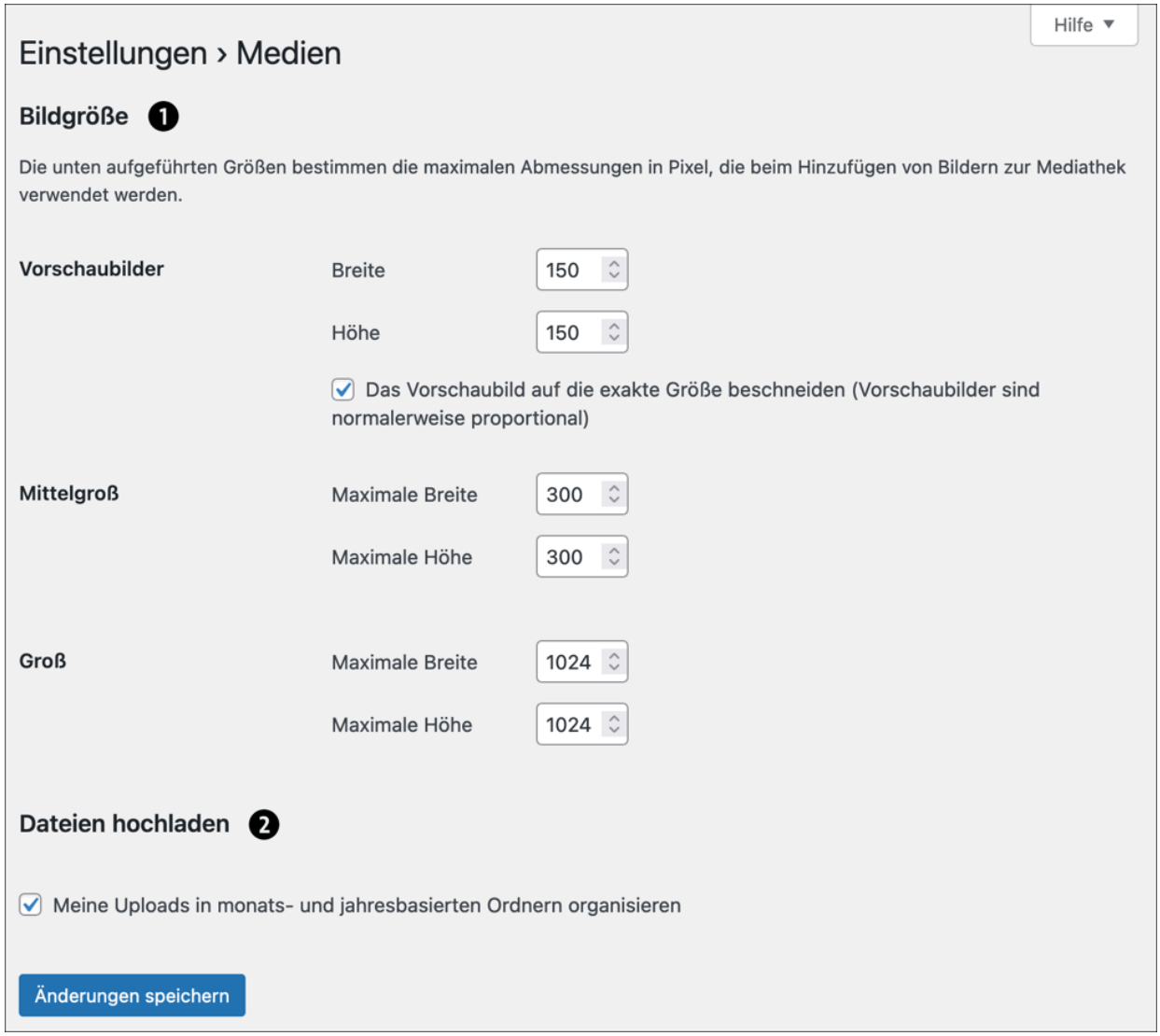


Abbildung 5.37 Auf der »Medien«-Einstellungsseite gibt es Optionen für die von WordPress generierten Vorschaubilder.

Hilfe ▾

Einstellungen › Permalinks

WordPress bietet die Möglichkeit, individuelle URL-Strukturen für deine Permalinks und Archive zu erstellen. Individuelle URL-Strukturen verbessern Aussehen, Benutzerfreundlichkeit und Vorwärtskompatibilität der Links. Eine Reihe an Platzhaltern ist verfügbar, hier sind einige Beispiele, um dir den Einstieg zu erleichtern.

Gebräuchliche Einstellungen

Wähle die Permalink-Struktur für deine Website. Die Verwendung des Schlagworts `%postname%` macht Links leicht verständlich und kann dazu beitragen, dass deine Beiträge in Suchmaschinen besser platziert werden.

Permalink-Struktur ❶

- ○ **Einfach**
  `https://wp6.uber.space/?p=123`
- ◉ **Tag und Name**
  `https://wp6.uber.space/2024/10/09/beispielbeitrag/`
- ○ **Monat und Name**
  `https://wp6.uber.space/2024/10/beispielbeitrag/`
- ○ **Numerisch**
  `https://wp6.uber.space/archive/123`
- ○ **Beitragsname**
  `https://wp6.uber.space/beispielbeitrag/`
- ○ **Individuelle Struktur**
  `https://wp6.uber.space`
  `/%year%/%monthnum%/%day%/%postname%/`

  Verfügbare Platzhalter:

  %year% %monthnum% %day% %hour% %minute% %second%
  %post_id% %postname% %category% %author%

Optional

Wenn du magst, kannst du hier individuelle Strukturen für deine Kategorien- und Schlagwörter-URLs anlegen. Zum Beispiel, wenn du `thema` als Kategorie-Basis verwendest, sehen die Links so aus: `https://wp6.uber.space/thema/allgemein/` . Falls das Feld leer bleibt, werden die Standardwerte verwendet.

Kategorie-Basis ❷

Schlagwort-Basis ❸

Änderungen speichern

Abbildung 5.38 Auf der Einstellungsseite »Permalinks« können Sie die URL-Struktur anpassen, die WordPress für Ihre Inhalte erzeugt.

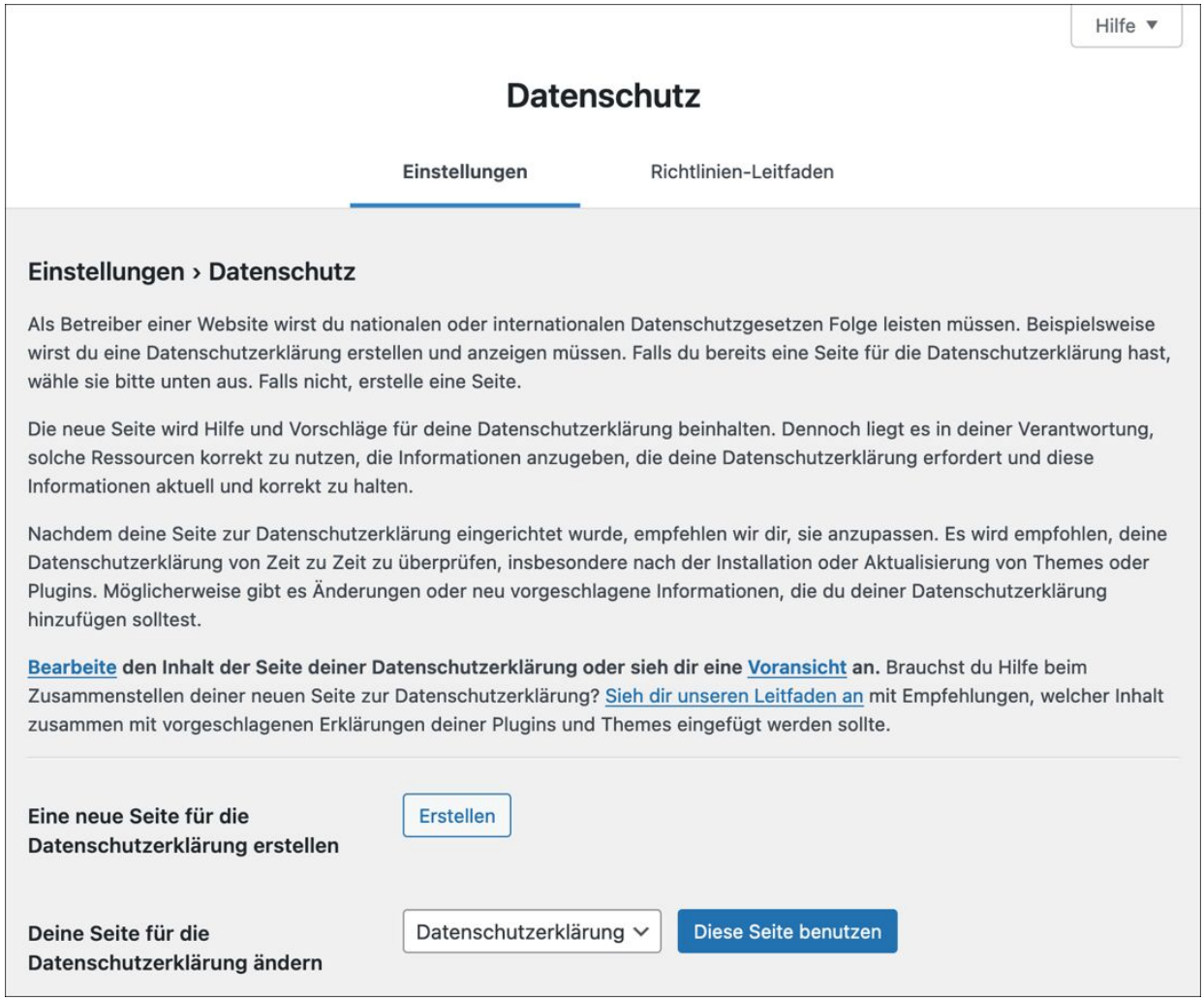


Abbildung 5.39 Im »Einstellungen«-Tab können Sie eine Seite für die Datenschutzerklärung erstellen und festlegen, welche Seite die Datenschutzerklärung enthält.

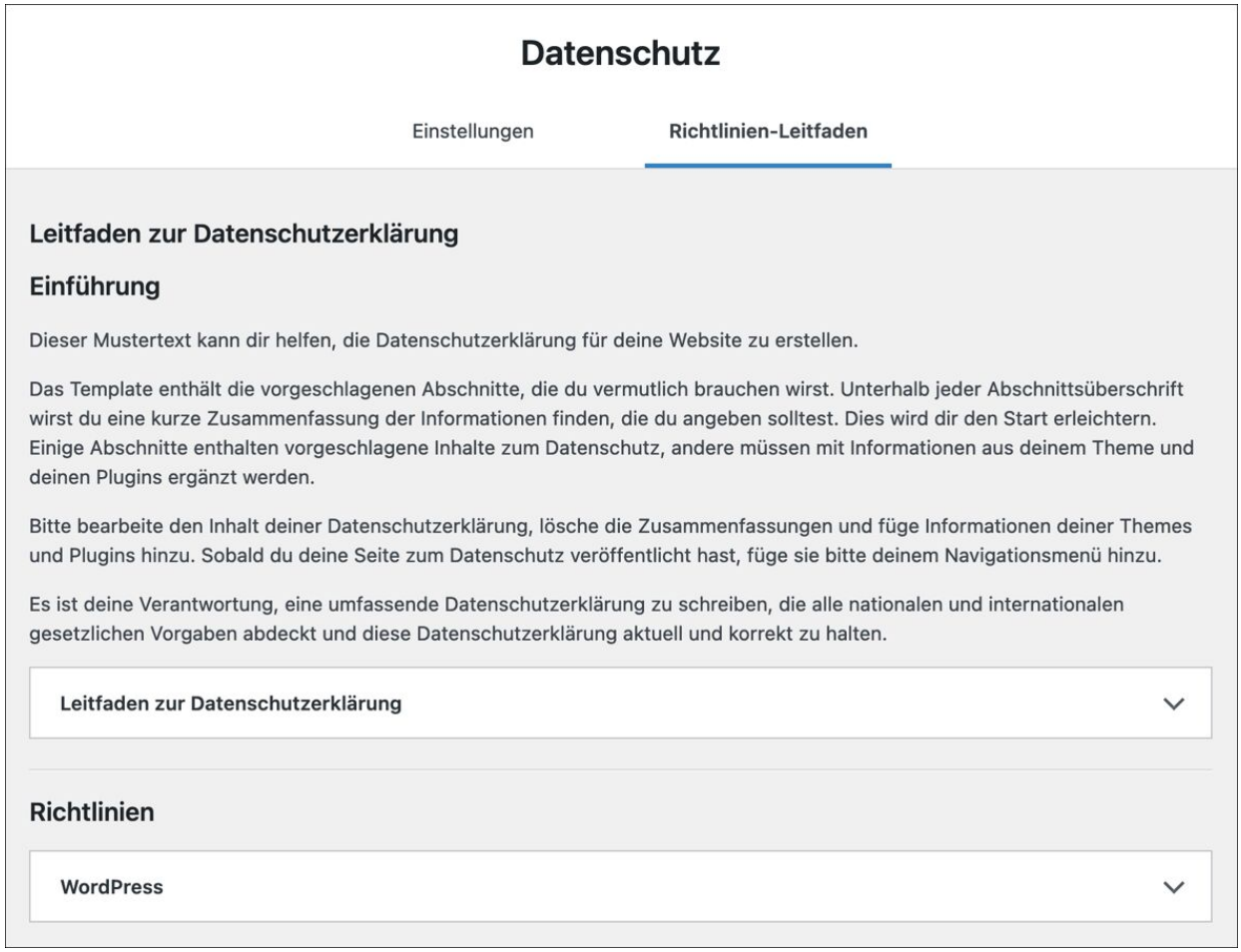


Abbildung 5.40 Im Tab »Richtlinien-Leitfaden« sehen Sie nützliche Tipps zum Anlegen einer Datenschutzerklärung. Die Richtlinien können von Plugins und Themes erweitert werden.

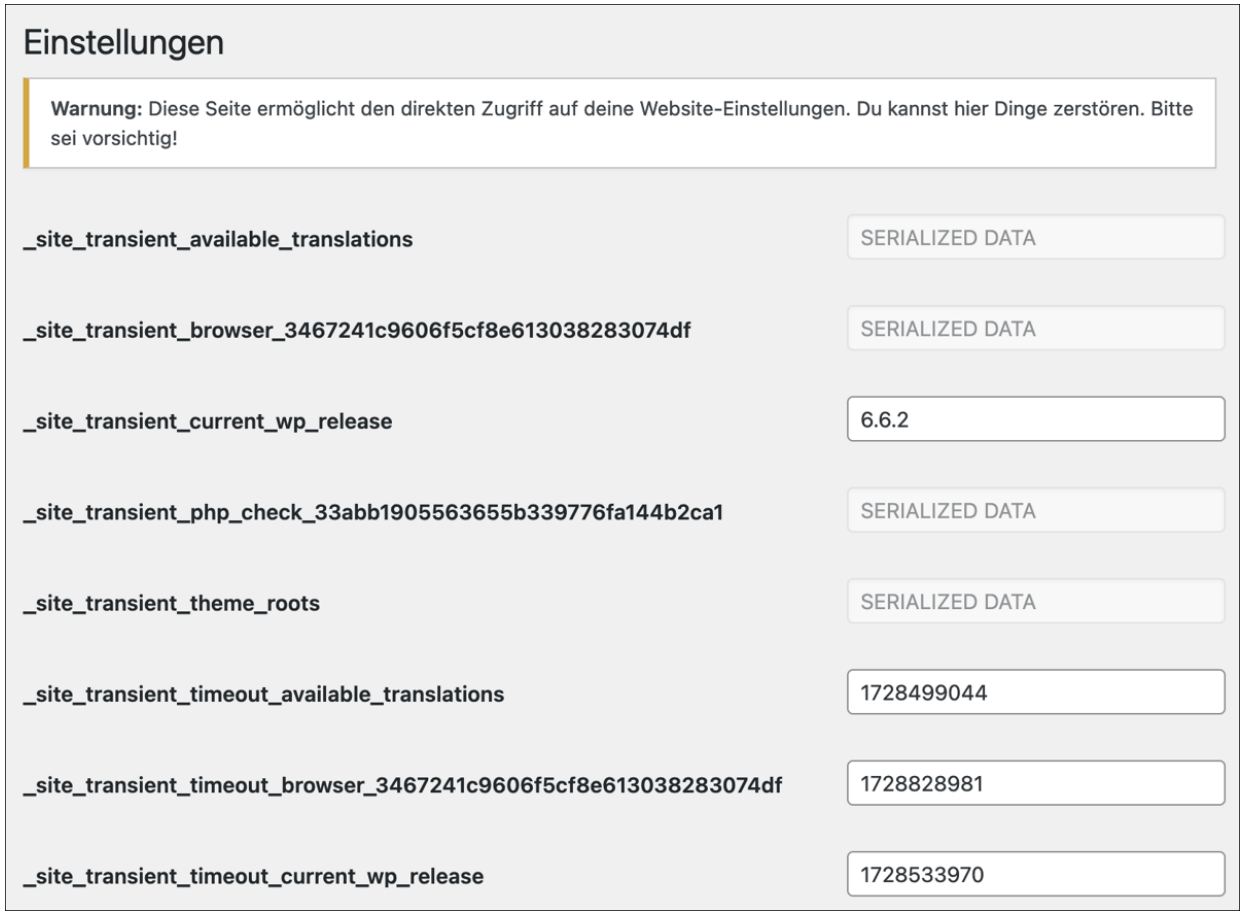


Abbildung 5.41 Unter »wp-admin/options.php« gibt es eine Seite, auf der alle Einstellungen aus der »wp_options«-Tabelle aufgelistet sind und viele direkt verändert werden können.

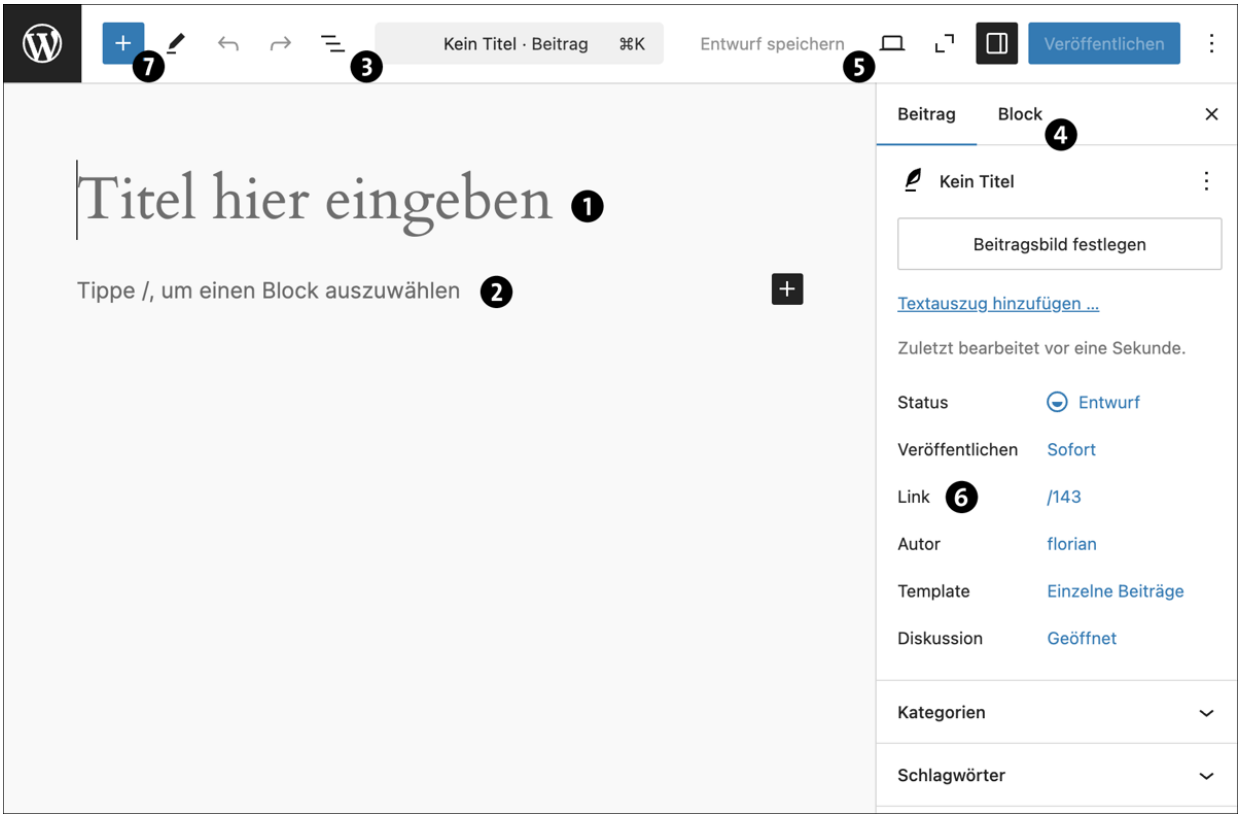


Abbildung 6.1 Im Editor haben Sie links einen großen Bereich für Ihre Inhalte, auf der rechten Seite eine Seitenleiste mit Einstellungsmöglichkeiten für den Beitrag/die Seite und Optionen für ausgewählte Blöcke.

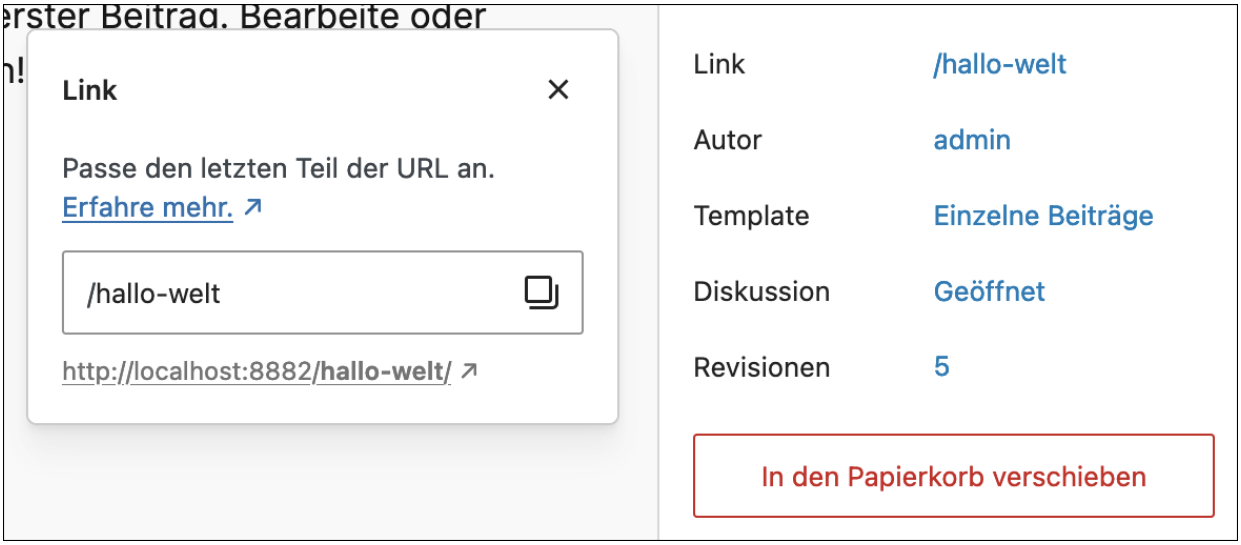


Abbildung 6.2 In der Seitenleiste finden Sie die Option, den »Link« eines Beitrags oder einer Seite anzupassen. Wenn Sie aber den Link eines schon veröffentlichten Inhalts anpassen, sollten Sie daran denken, eine Weiterleitung von der alten auf die neue URL einzurichten.

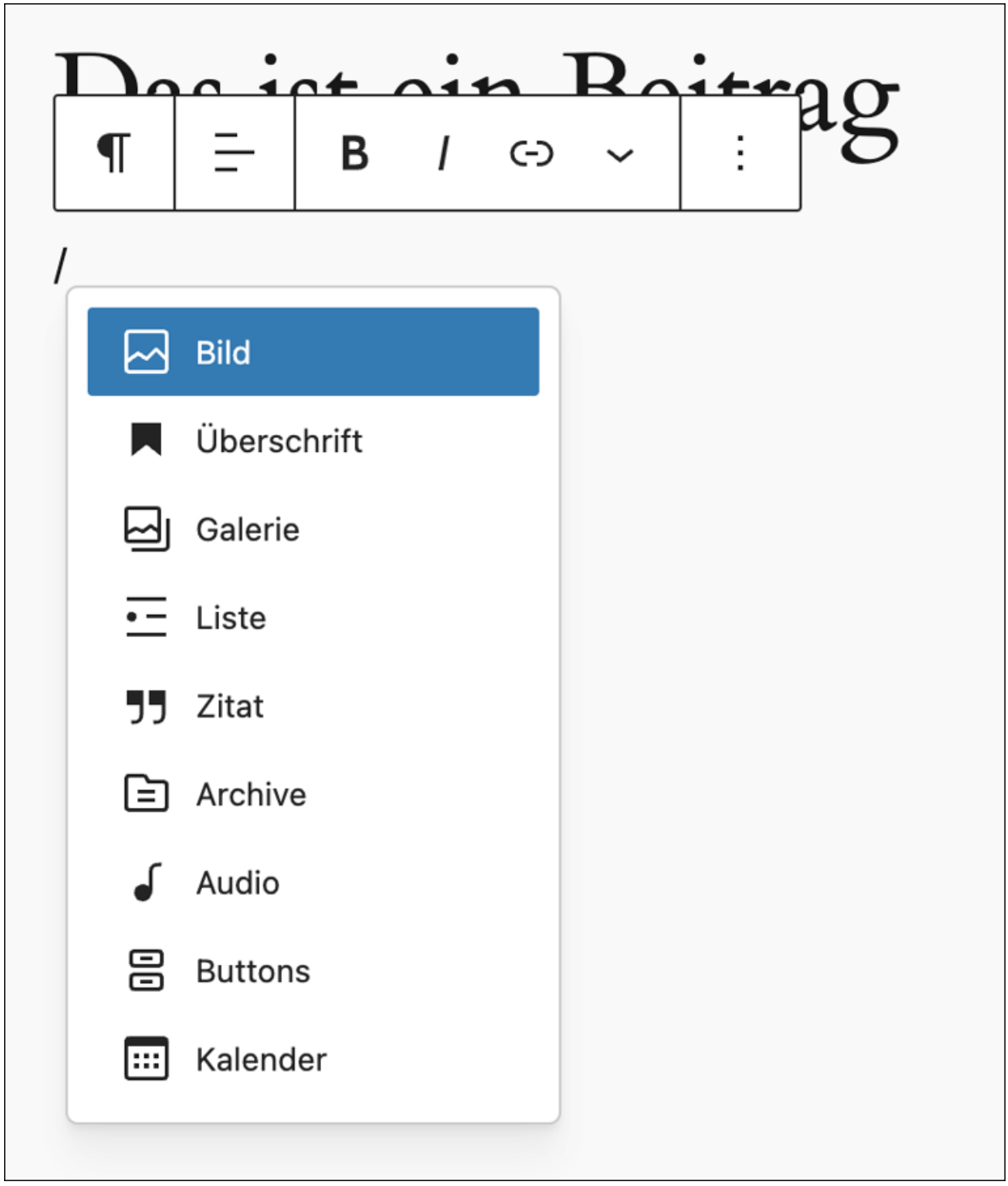

Bild
Überschrift
Galerie
Liste
Zitat
Archive
Audio
Buttons
Kalender

Abbildung 6.3 Der schnellste Weg zum neuen Block ist der »/«-Befehl, mit dem sich eine Auswahlliste an Blöcken öffnet, die Sie über Eingabe des Blocknamens einschränken können.

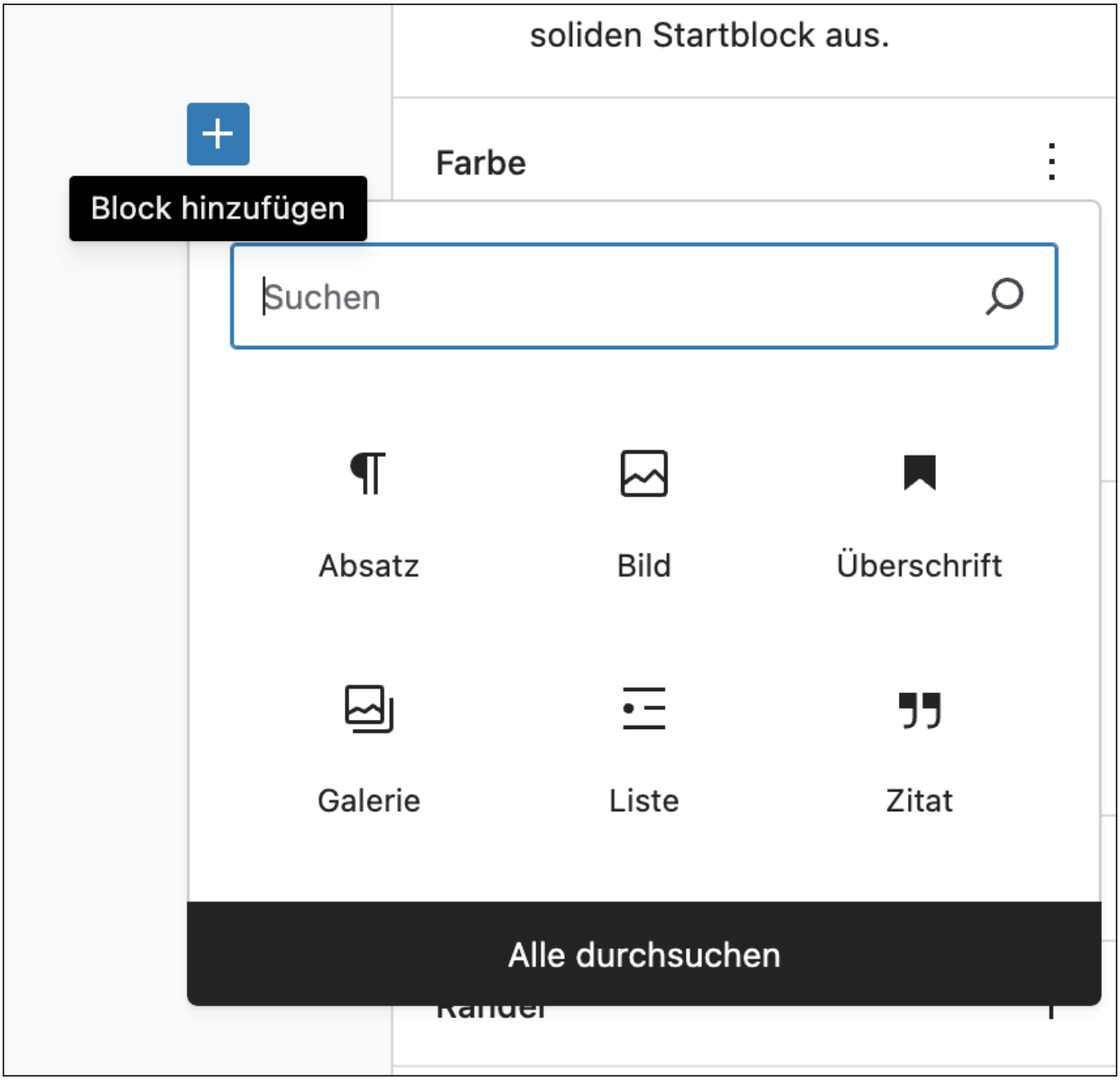


Abbildung 6.4 Über das Plus-Symbol auf der rechten Seite bei einem leeren Absatzblock gelangen Sie in eine kleine Variante des Block-Inserters, der im nächsten Punkt vorgestellt wird. Auch hier können Sie über die Suche die Liste der angezeigten Blöcke einschränken.

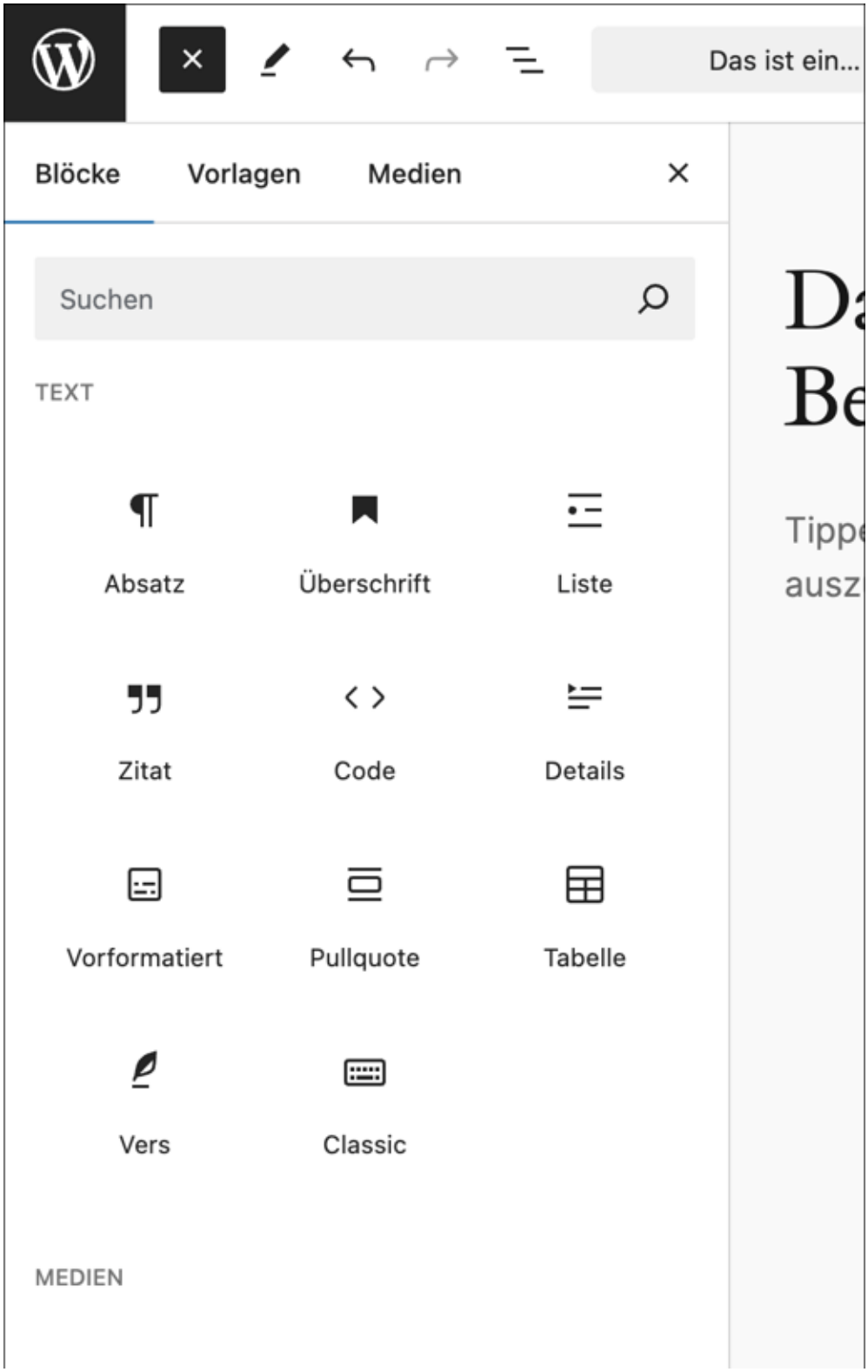

Das ist ein...
Blöcke
Vorlagen
Medien
Suchen
TEXT
Absatz
Überschrift
Liste
Zitat
Code
Details
Vorformatiert
Pullquote
Tabelle
Vers
Classic
MEDIEN



Abbildung 6.5 Die große Version des Block-Inserters erreichen Sie oben links über den Plus-Button oder den Button »Alle durchsuchen« in der kleinen Inserter-Version, die Sie über eins der Plus-Symbole in der Bearbeitungsansicht öffnen können.

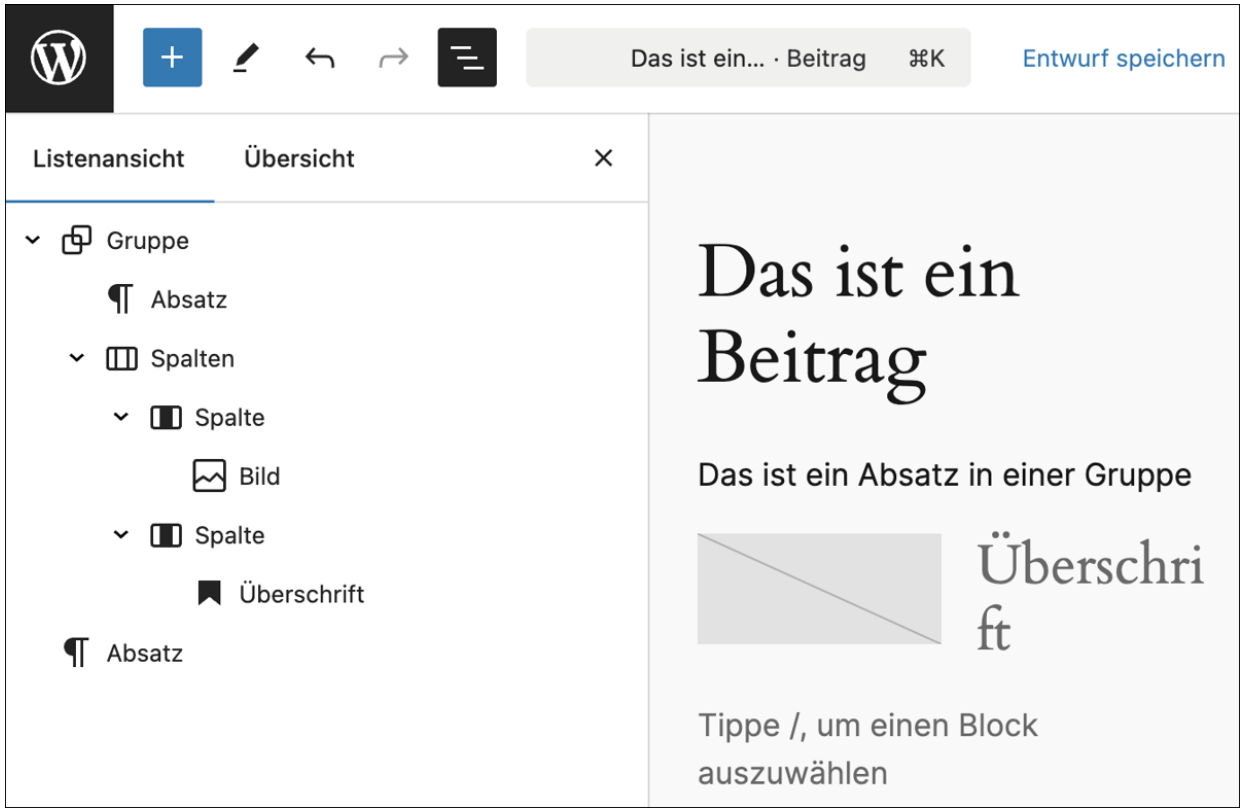


Abbildung 6.6 Mit der Dokumentenansicht haben Sie Zugriff auf eine Listenansicht der im Beitrag oder der Seite eingefügten Blöcke und können so gegebenenfalls einfacher den Überblick bei verschachtelten Block-Strukturen behalten.

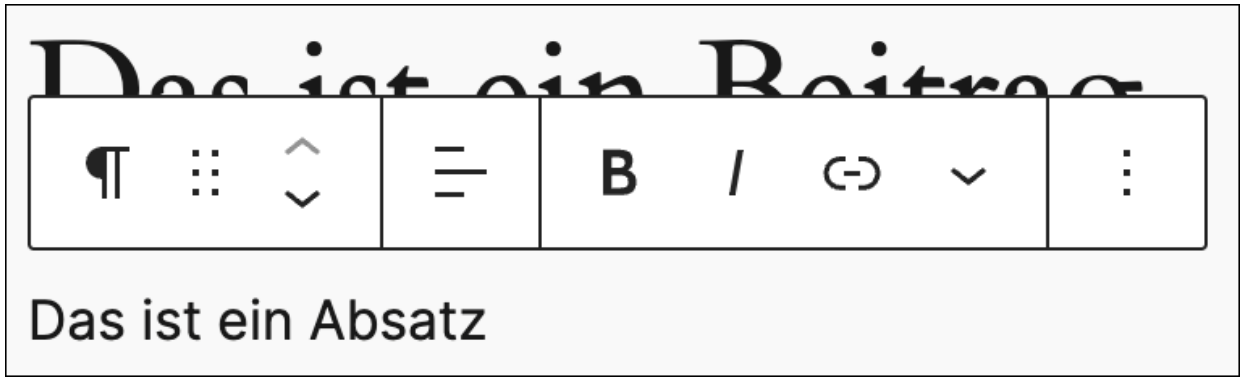
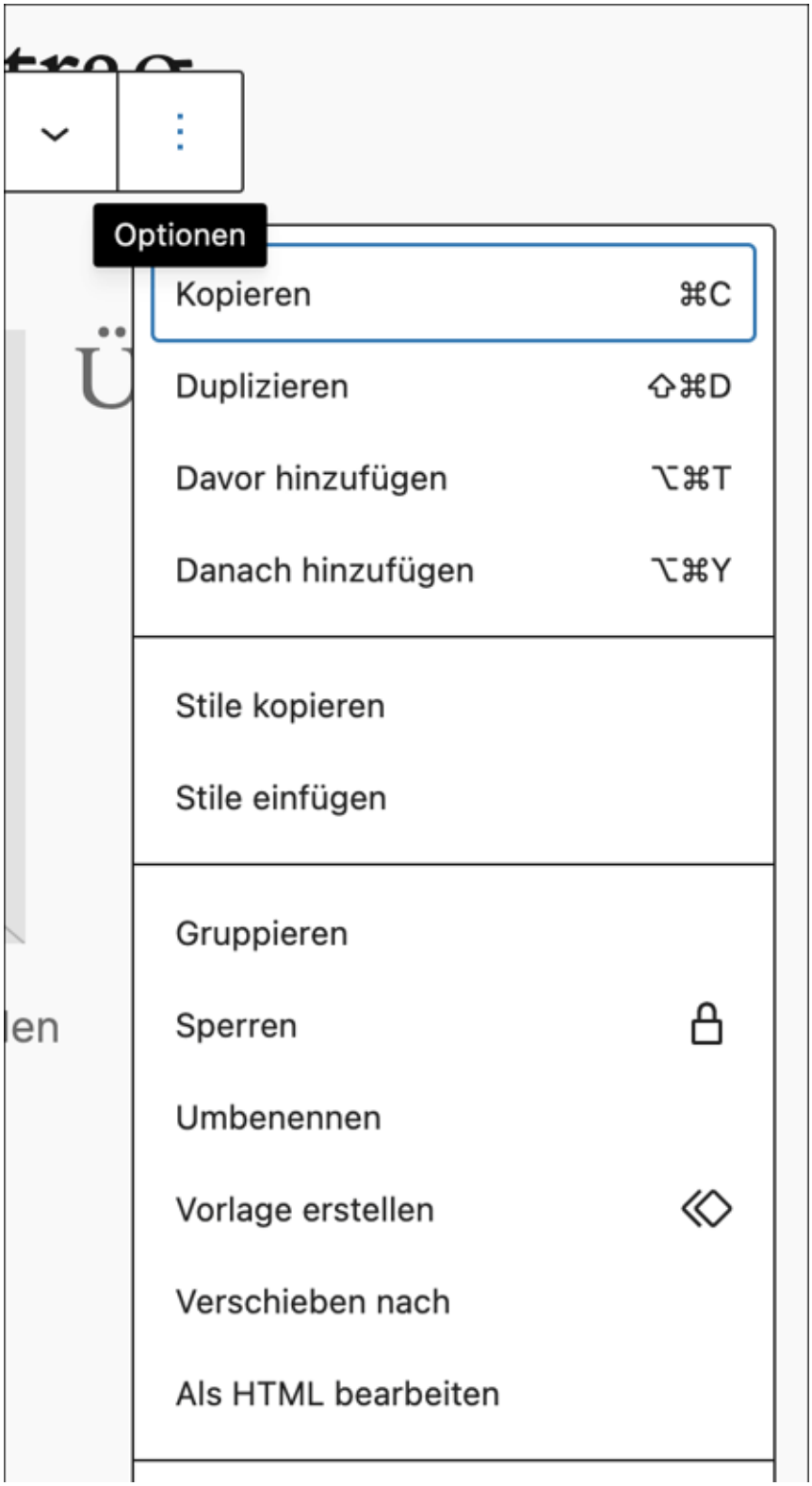


Abbildung 6.7 Die Block-Werkzeugleiste bietet je nach ausgewähltem Block unterschiedliche Optionen – beim Absatz-Block zum Beispiel zur Textausrichtung und zum Setzen einer Verlinkung.


Optionen
Kopieren ⌘C
Duplizieren ⇧⌘D
Davor hinzufügen ⌥⌘T
Danach hinzufügen ⌥⌘Y
Stile kopieren
Stile einfügen
Gruppieren
Sperren
Umbenennen
Vorlage erstellen
Verschieben nach
Als HTML bearbeiten

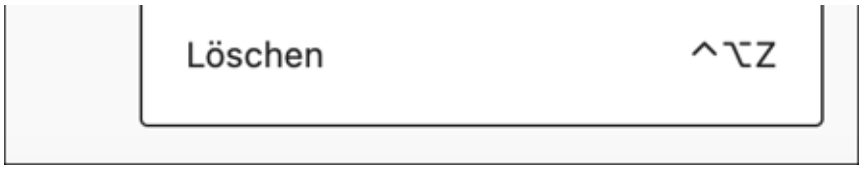


Abbildung 6.8 Über das Drei-Punkte-Menü in der Block-Toolbar können Sie unter anderem eine Vorlage aus dem oder den Blöcken erstellen, die gerade ausgewählt sind, und den Block verschieben.

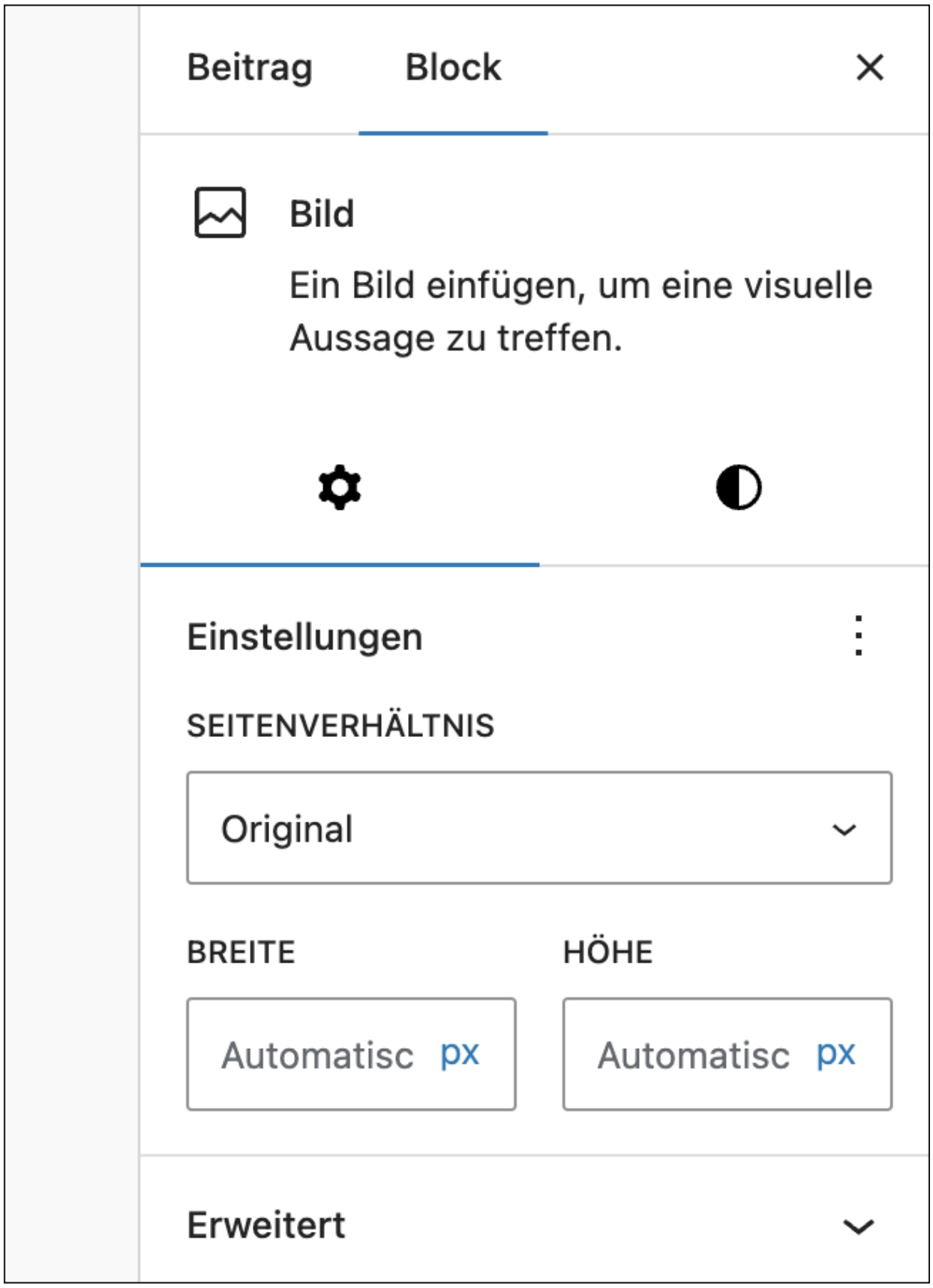

Beitrag
Block
Bild
Ein Bild einfügen, um eine visuelle Aussage zu treffen.
Einstellungen
SEITENVERHÄLTNIS
Original
BREITE
HÖHE
Automatisc px
Automatisc px
Erweitert

Abbildung 6.9 Für den ausgewählten Block können Sie in der Seitenleiste mehr Einstellungen vornehmen als direkt über die Block-Werkzeugleiste.

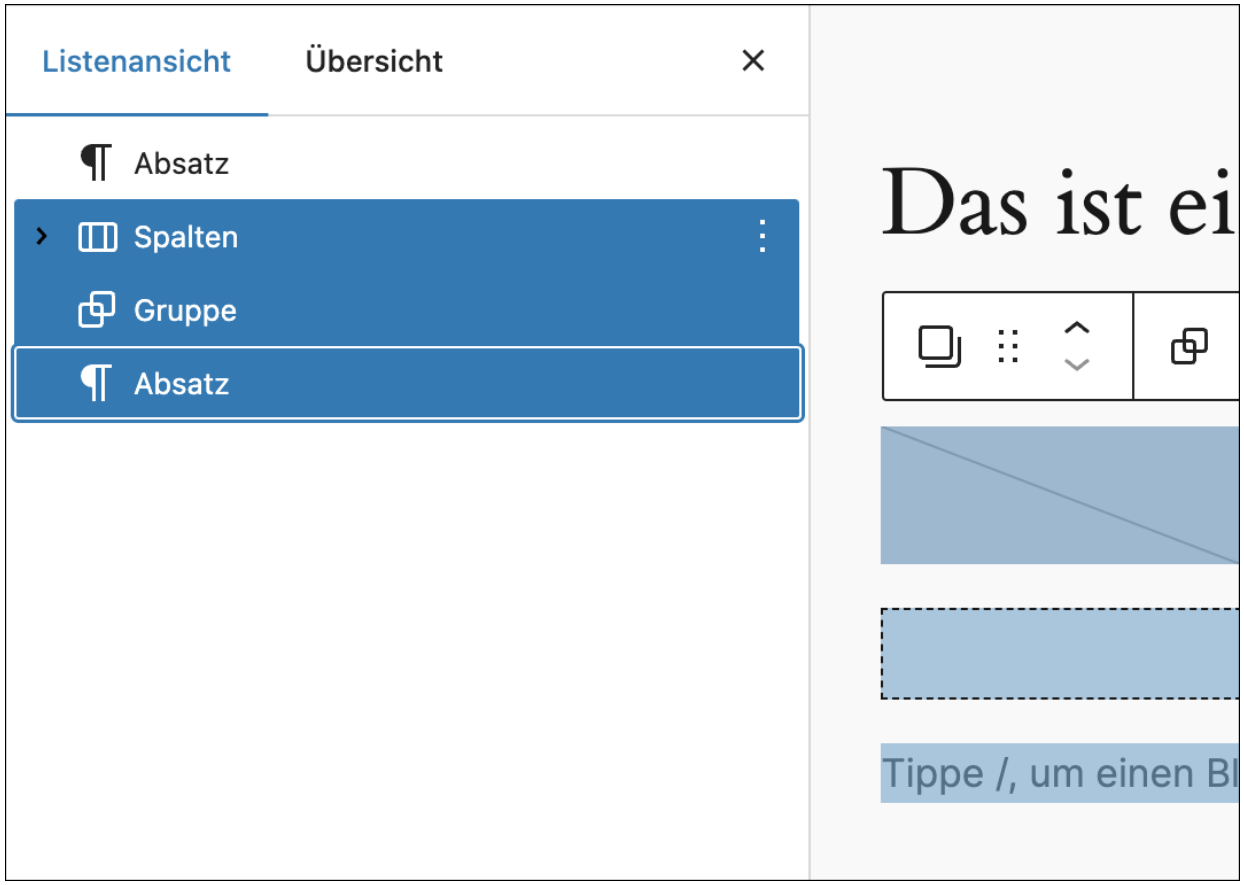


Abbildung 6.10 Die Auswahl mehrere Blöcke ist in der Dokumentenansicht meist einfacher als im Editor.

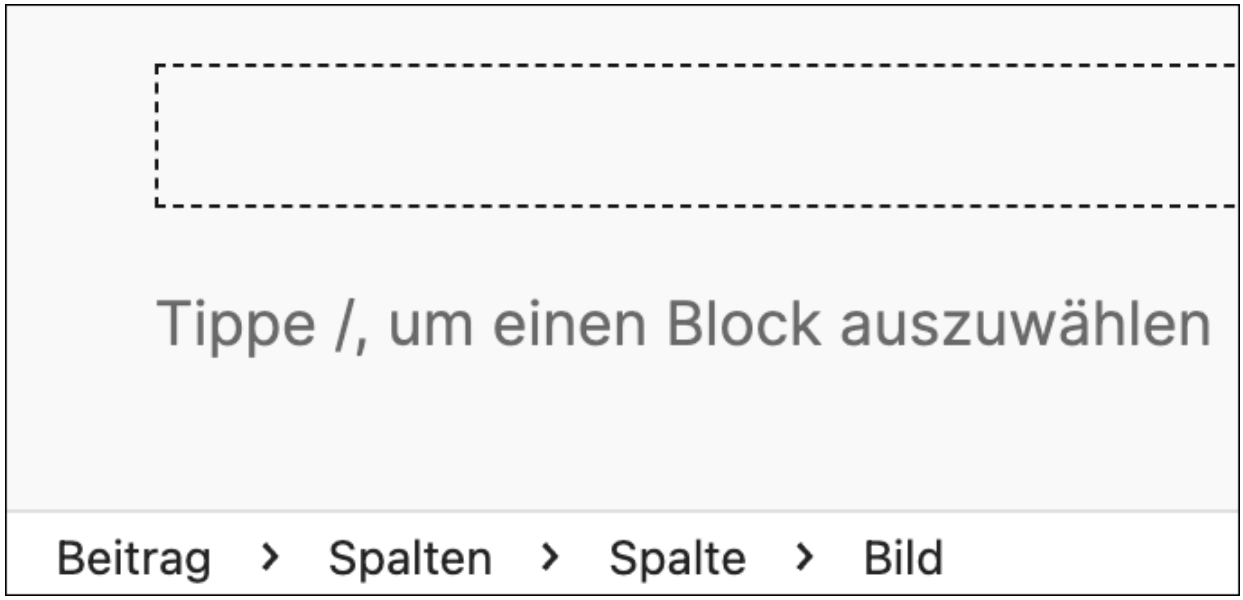
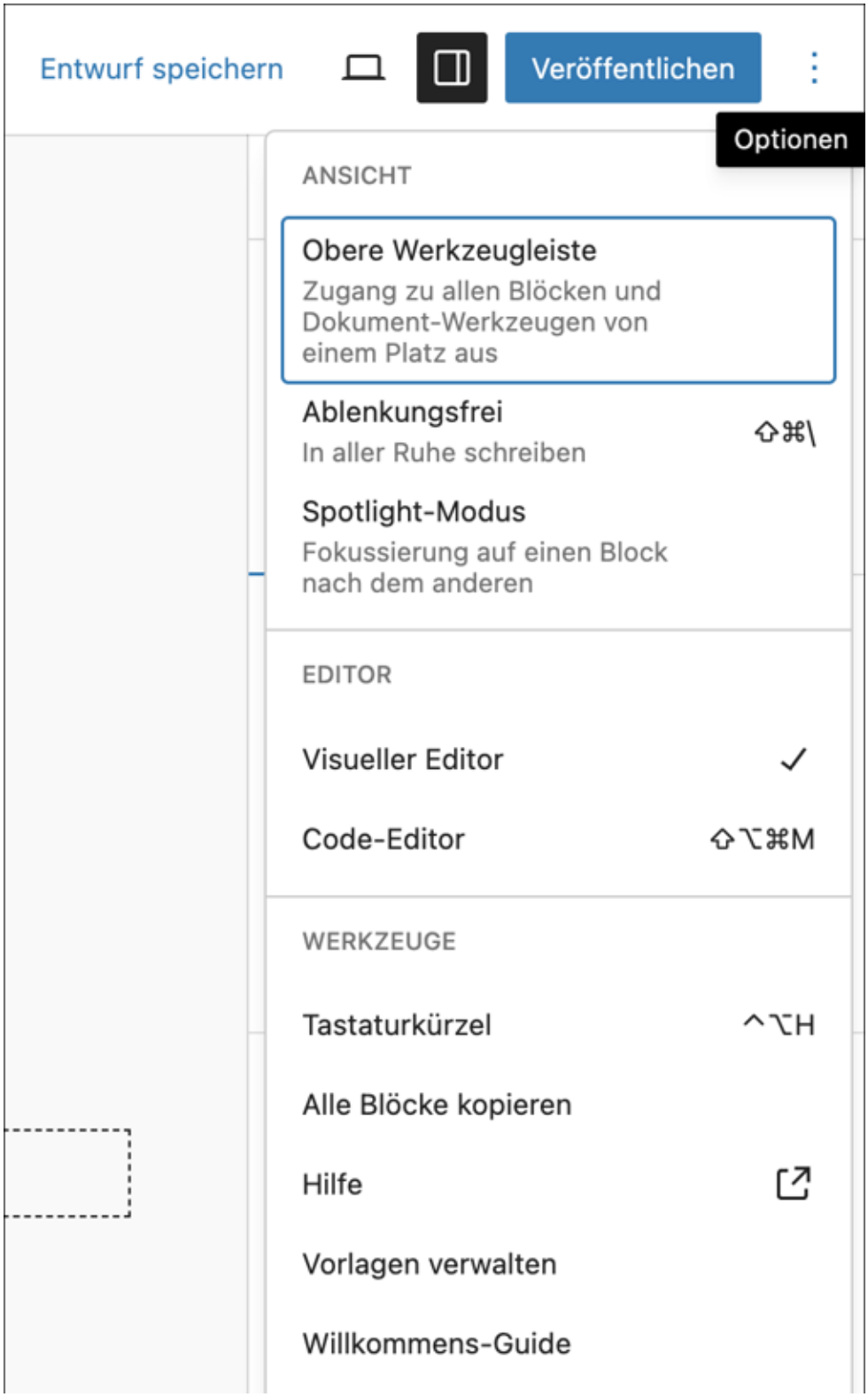


Abbildung 6.11 Diese kleine unscheinbare Breadcrumb ist besonders nützlich bei tief verschachtelten Block-Strukturen.


Entwurf speichern
Veröffentlichen
Optionen
ANSICHT
Obere Werkzeugleiste
Zugang zu allen Blöcken und Dokument-Werkzeugen von einem Platz aus
Ablenkungsfrei
In aller Ruhe schreiben
⇧⌘\
Spotlight-Modus
Fokussierung auf einen Block nach dem anderen
EDITOR
Visueller Editor
Code-Editor
⇧⌥⌘M
WERKZEUGE
Tastaturkürzel
^⌥H
Alle Blöcke kopieren
Hilfe
Vorlagen verwalten
Willkommens-Guide

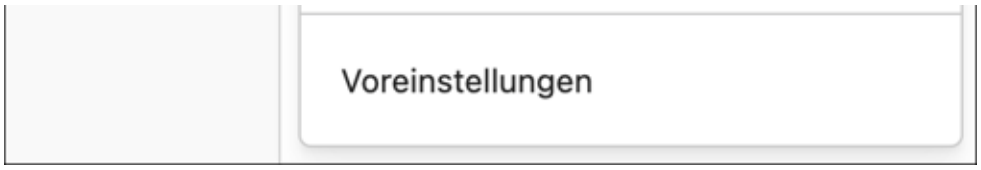


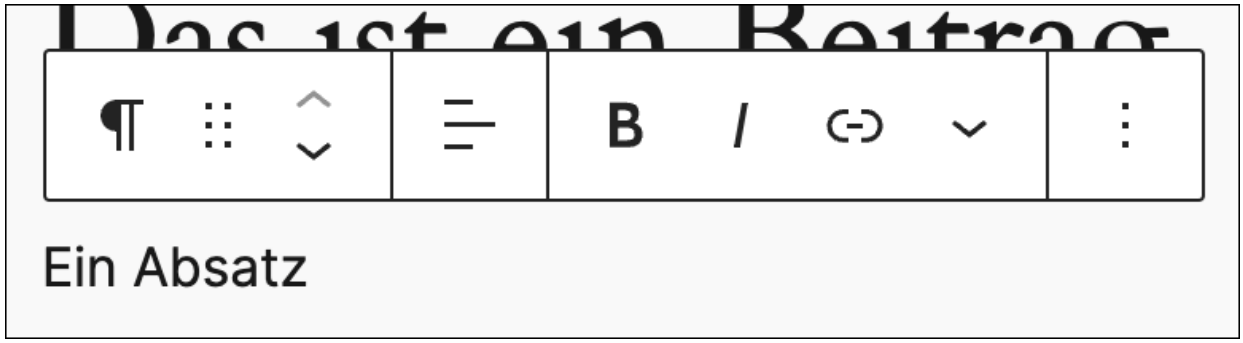
Abbildung 6.12 Im Editor-Drei-Punkte-Menü können Sie unter anderem den Editor-Modus ändern und die Voreinstellungen anpassen.

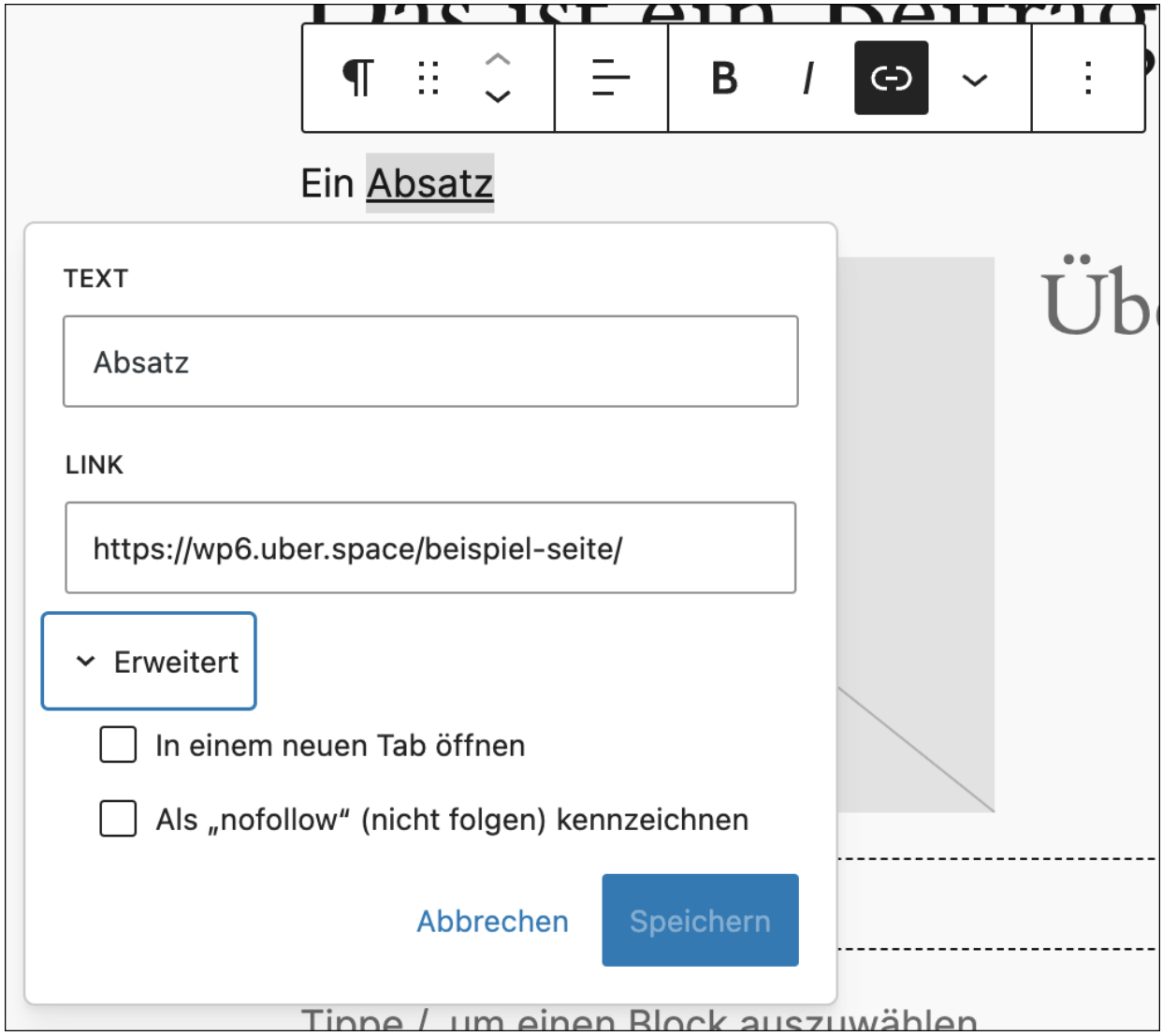


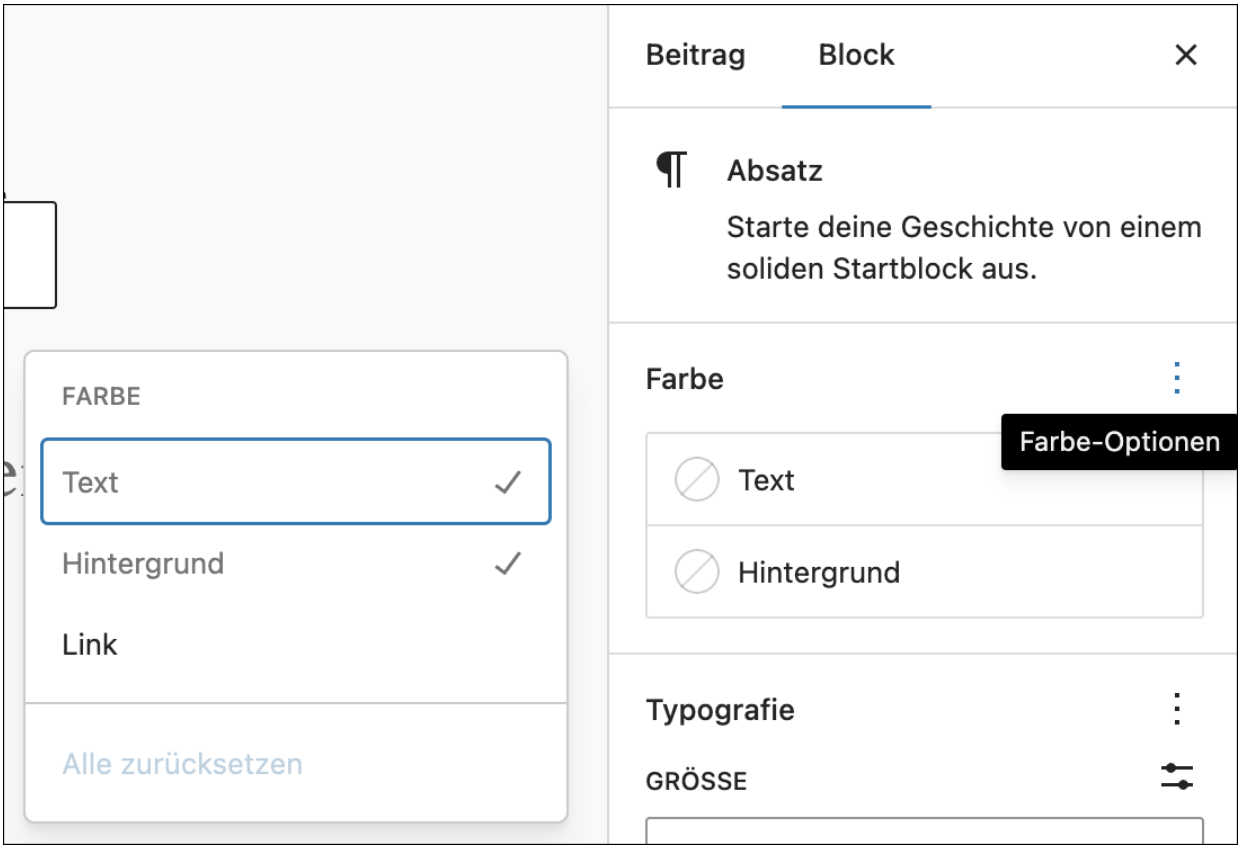
Abbildung 6.13 Wenn Sie die Option »Obere Werkzeugleiste« aktivieren, werden die Optionen aus der Block-Toolbar mit in der Editor-Toolbar angezeigt; die Blöcke haben dann selbst keine eigene Toolbar mehr.

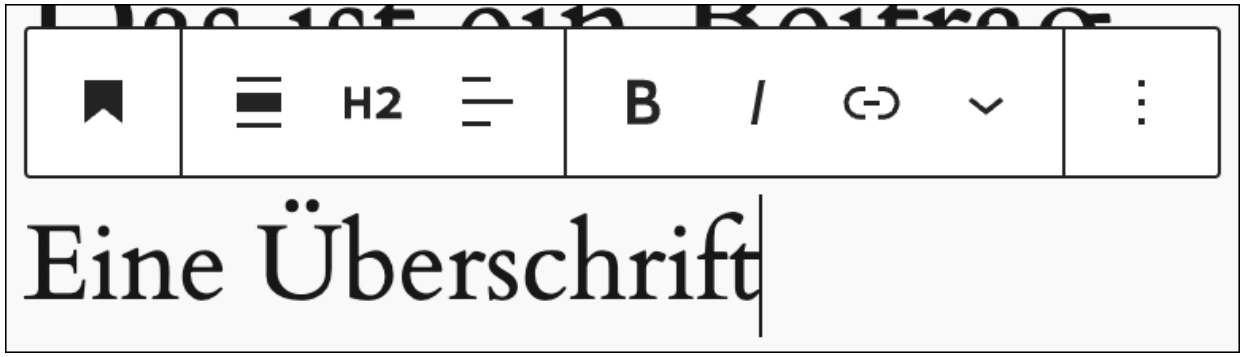


Abbildung 6.14 Im »Ablenkungsfrei«-Modus werden alle gegebenenfalls störenden Elemente des Editors ausgeblendet.

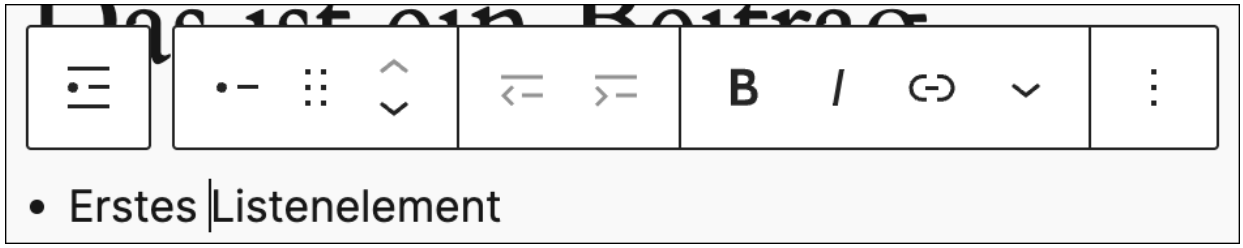

Das ist ein Beitrag
B
I
Ein Absatz

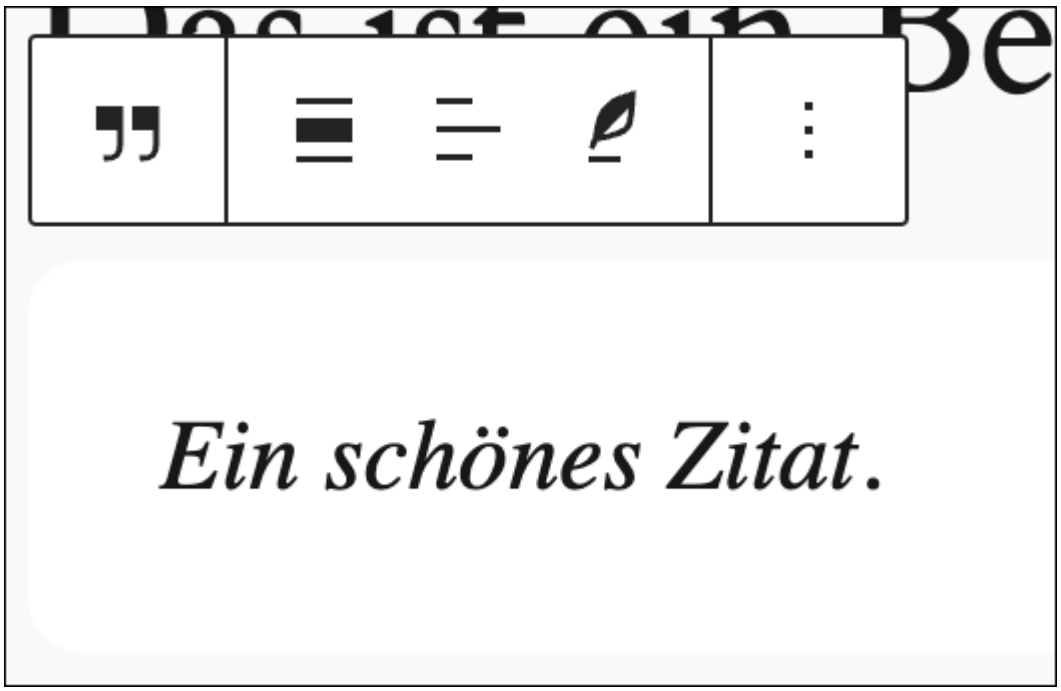

Ein Absatz
TEXT
Absatz
LINK
https://wp6.uber.space/beispiel-seite/
Erweitert
In einem neuen Tab öffnen
Als „nofollow“ (nicht folgen) kennzeichnen
Abbrechen
Speichern

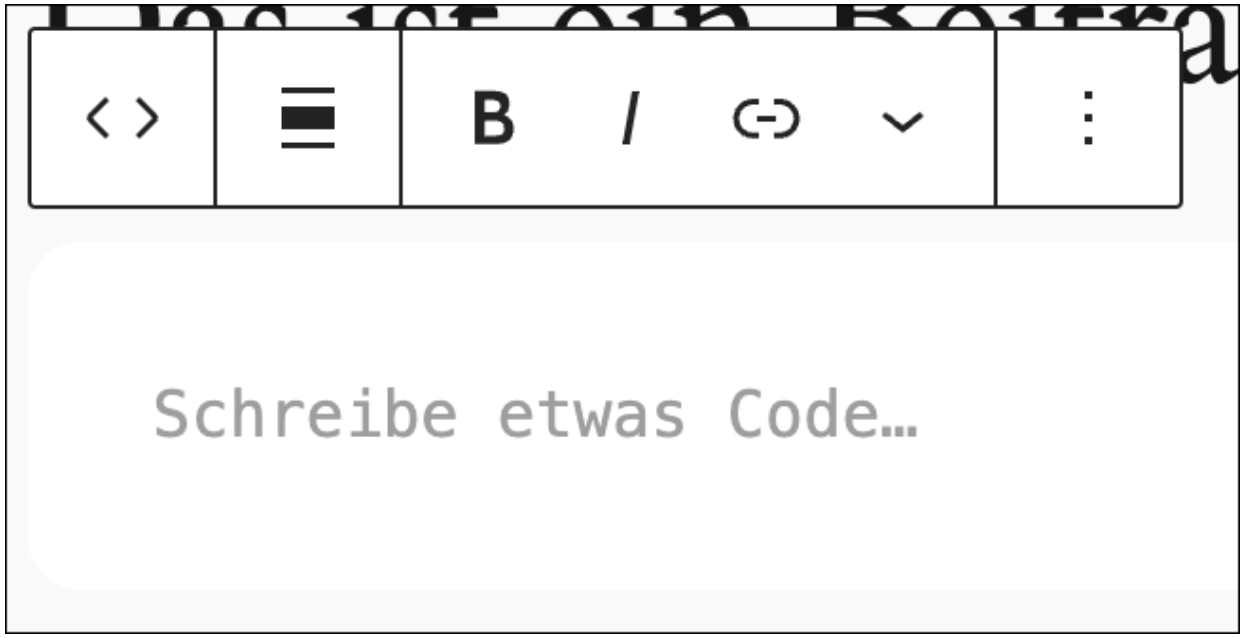

Beitrag
Block
Absatz
Starte deine Geschichte von einem soliden Startblock aus.
Farbe
Farbe-Optionen
Text
Hintergrund
Typografie
GRÖSSE
FARBE
Text
Hintergrund
Link
Alle zurücksetzen


H2
B
I
Eine Überschrift

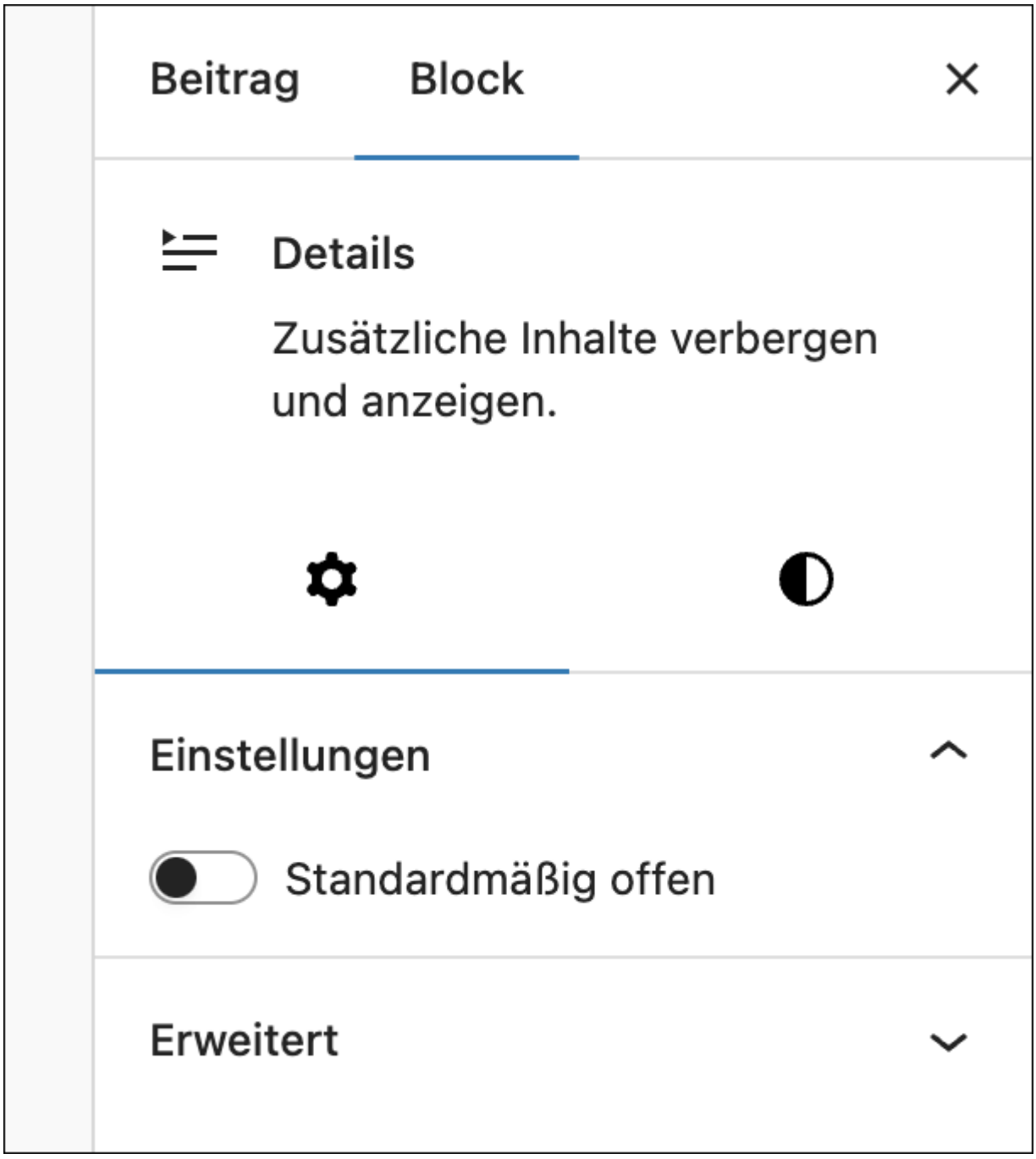

Erstes Listenelement
Zweites Listenelement

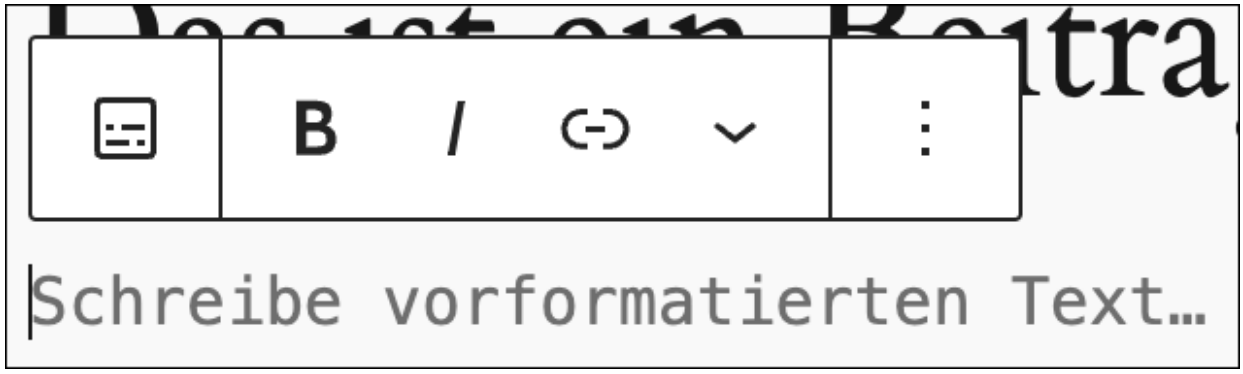

B
I
Erstes Listenelement

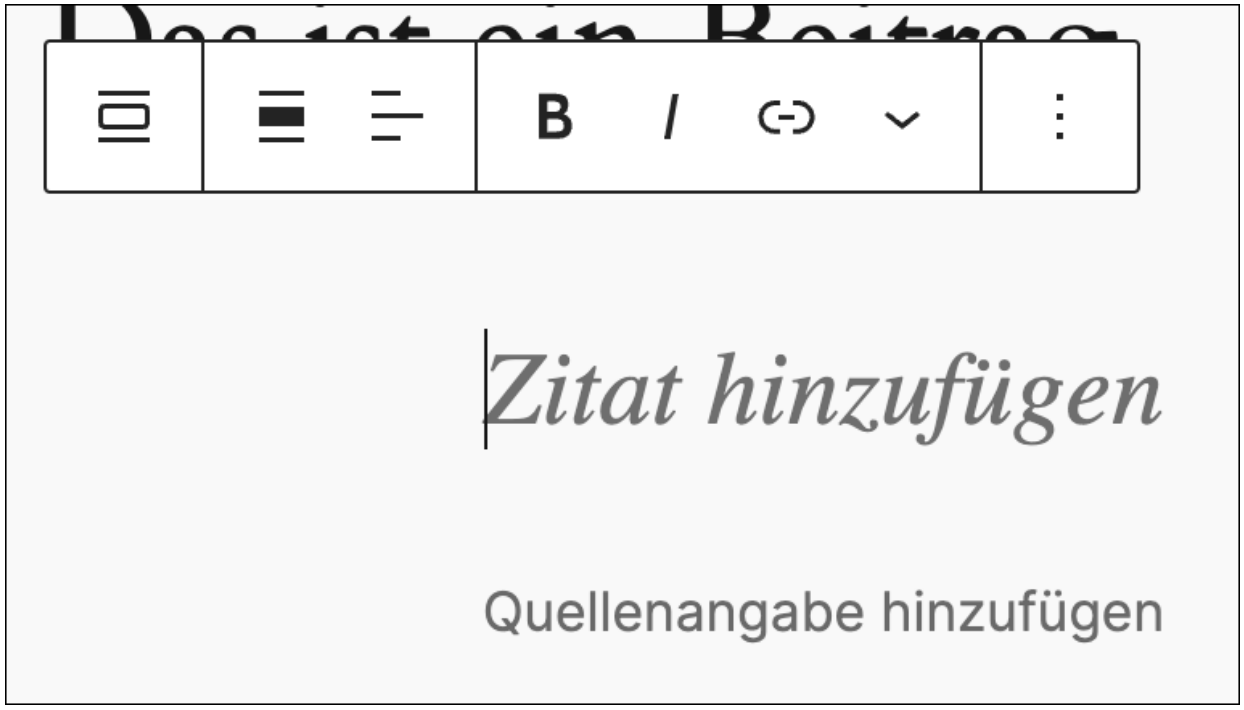

Ein schönes Zitat.

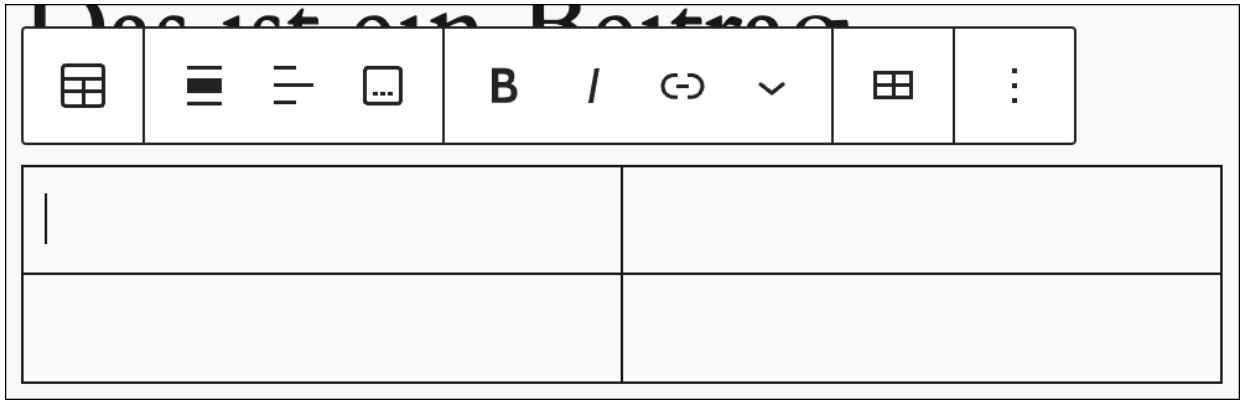

B
I
Schreibe etwas Code…

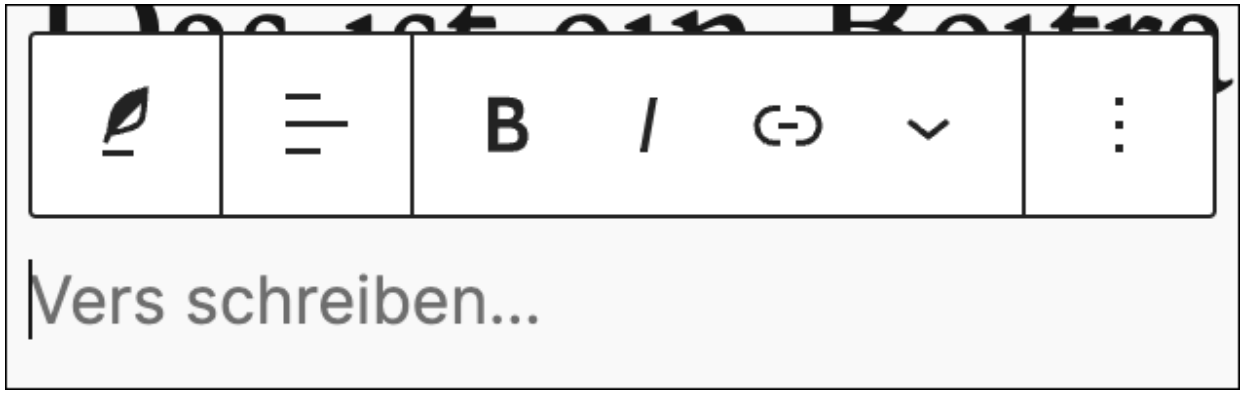

ein Beitrag
▼Eine Zusammenfassung schreiben …
Gib / ein, um einen verborgenen Block hinzuzufügen

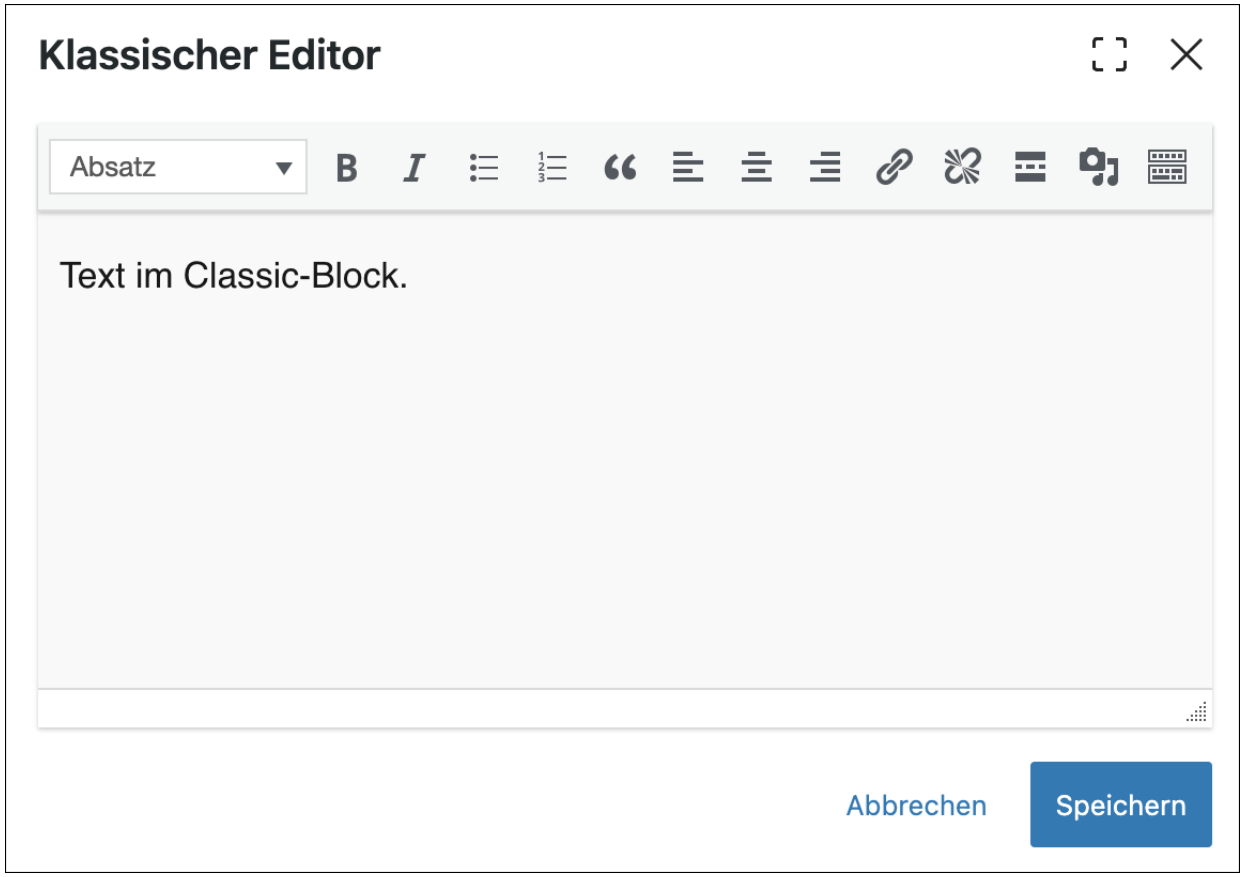

Beitrag
Block
Details
Zusätzliche Inhalte verbergen und anzeigen.
Einstellungen
Standardmäßig offen
Erweitert

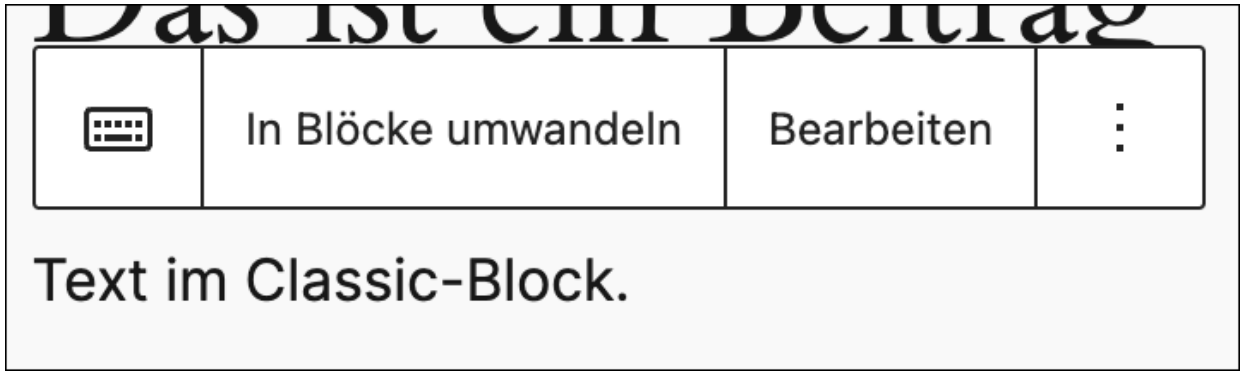

B
I
Schreibe vorformatierten Text…

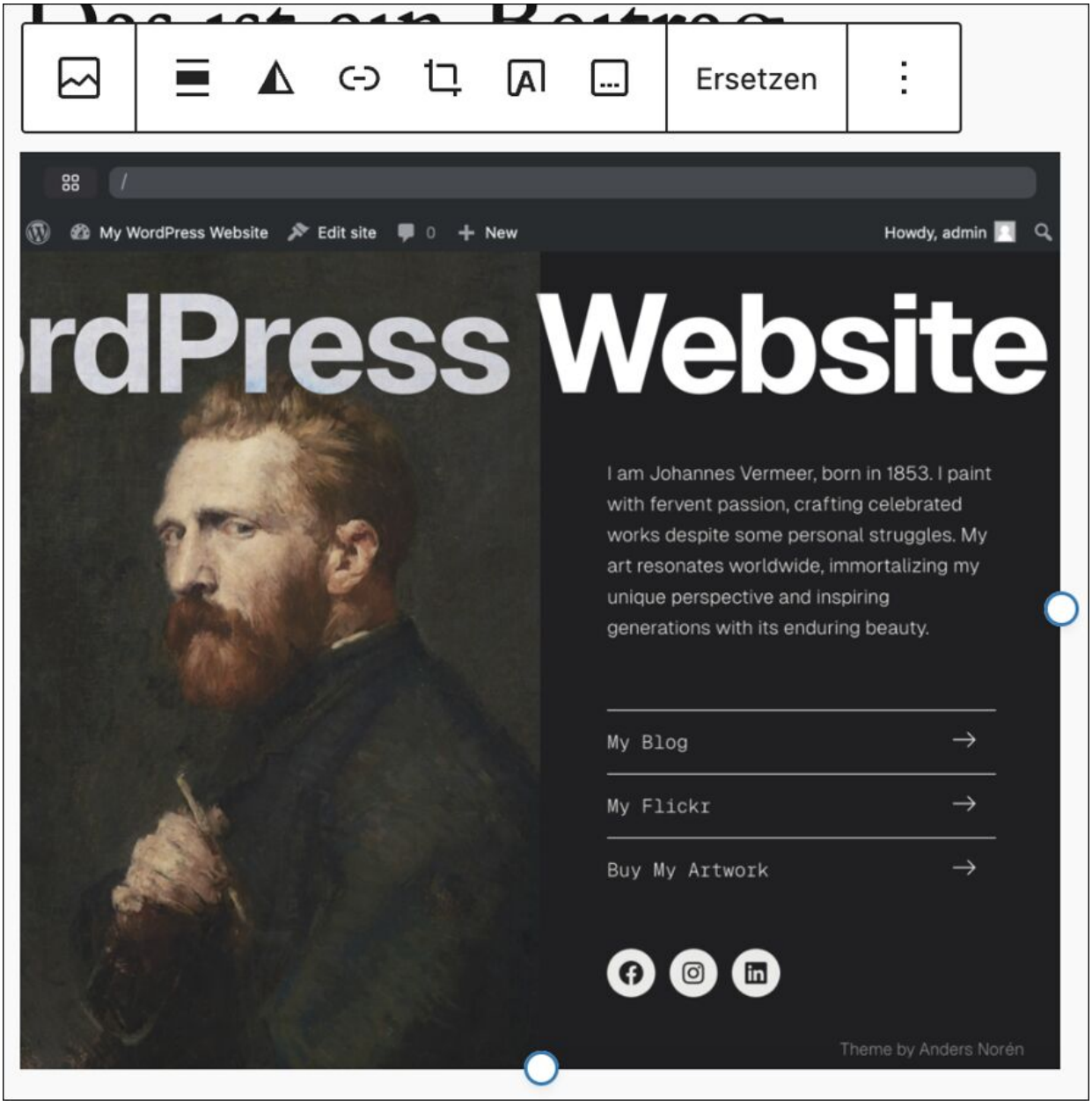

Zitat hinzufügen
Quellenangabe hinzufügen

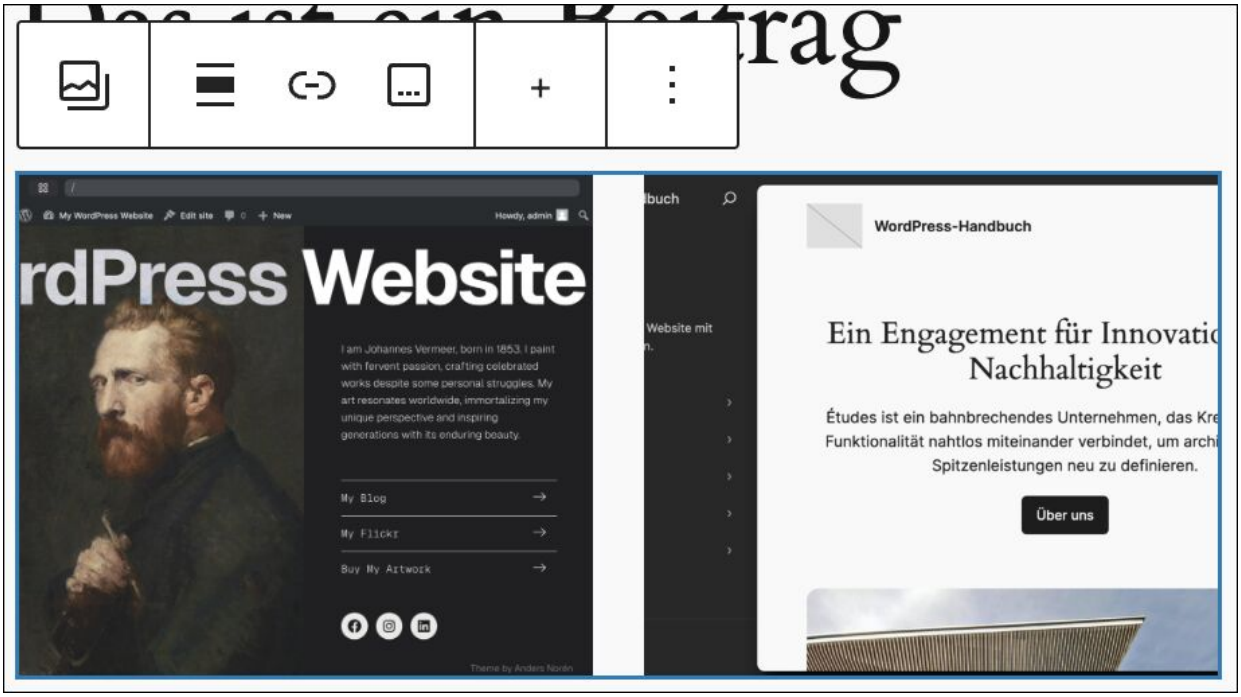


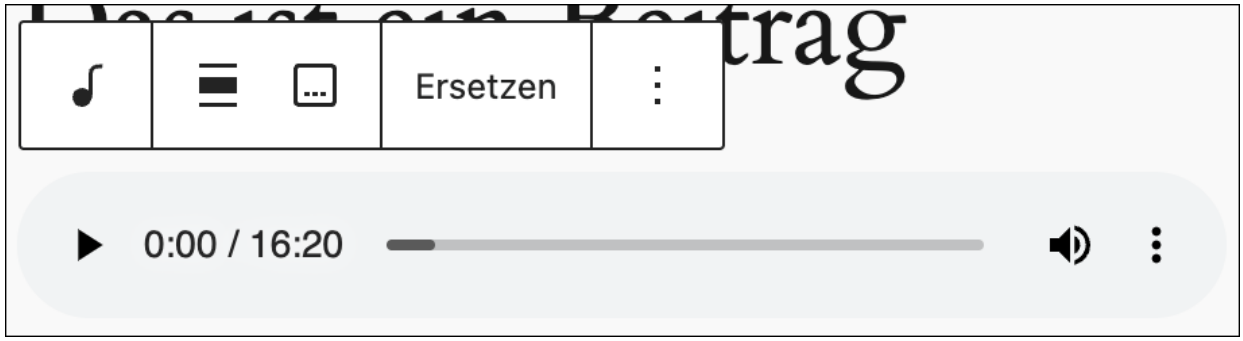

B
I
Vers schreiben...

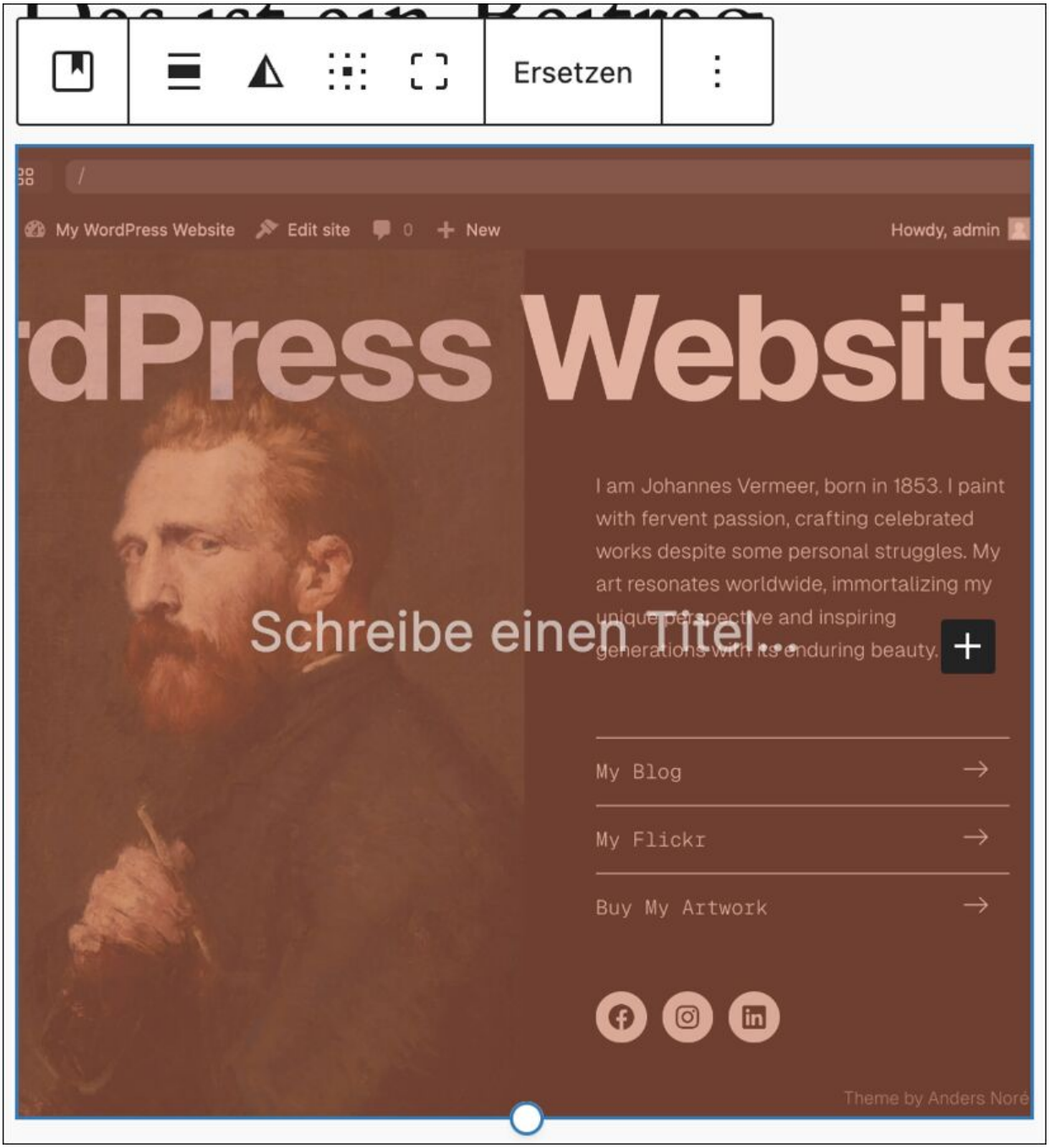

Klassischer Editor
Absatz
Text im Classic-Block.
Abbrechen
Speichern

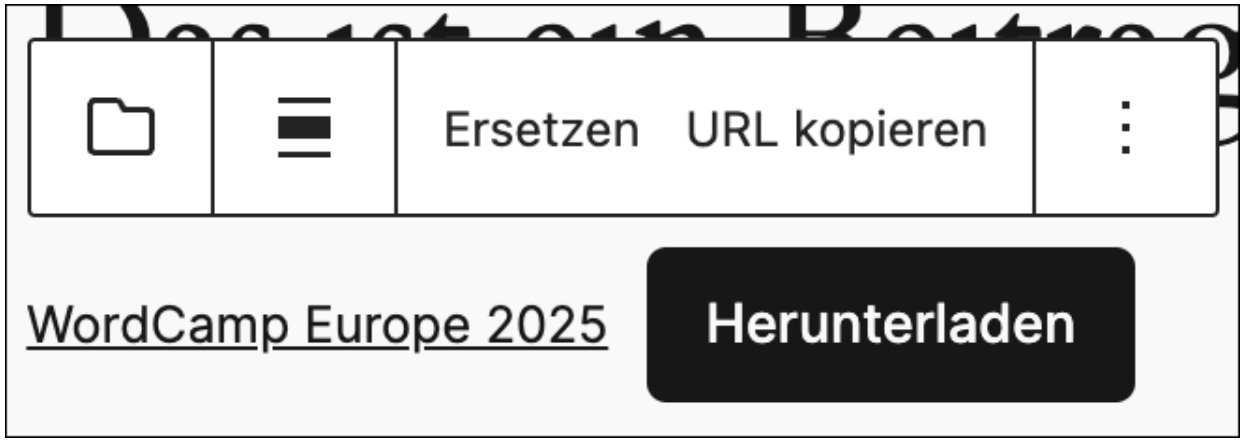

In Blöcke umwandeln
Bearbeiten
Text im Classic-Block.

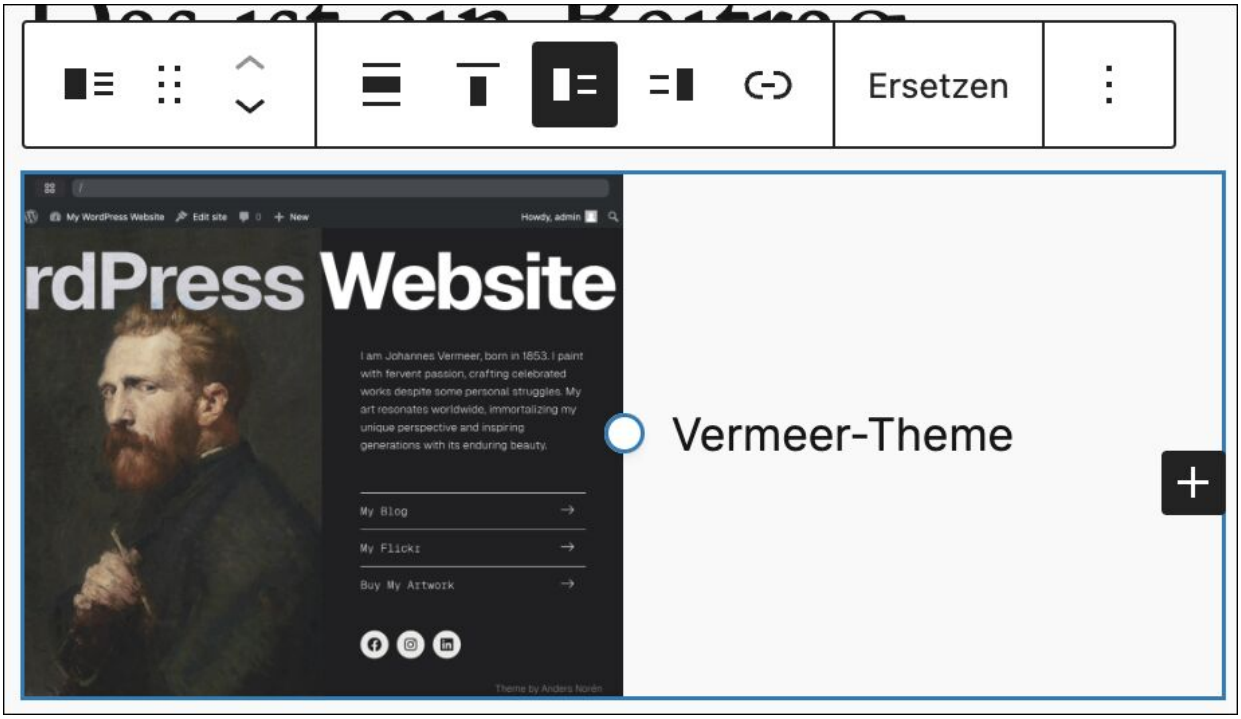

Ersetzen
/
My WordPress Website
Edit site
0
New
Howdy, admin
rdPress Website
I am Johannes Vermeer, born in 1853. I paint with fervent passion, crafting celebrated works despite some personal struggles. My art resonates worldwide, immortalizing my unique perspective and inspiring generations with its enduring beauty.
My Blog
My Flickr
Buy My Artwork
Theme by Anders Norén

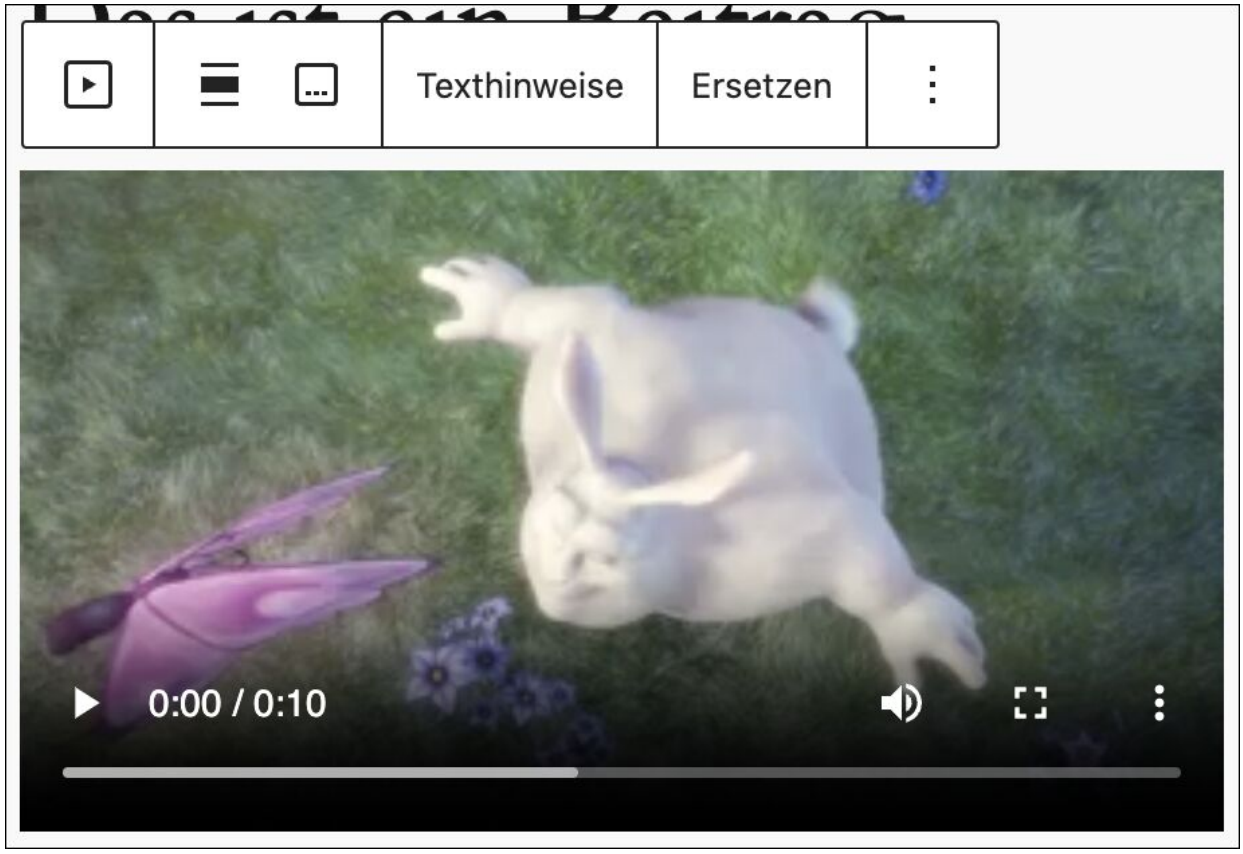

rag
My WordPress Website
Edit site
New
Howdy, admin
rdPress Website
I am Johannes Vermeer, born in 1853. I paint with fervent passion, crafting celebrated works despite some personal struggles. My art resonates worldwide, immortalizing my unique perspective and inspiring generations with its enduring beauty.
My Blog
My Flickr
Buy My Artwork
Theme by Anders Norén
buch
Website mit
WordPress-Handbuch
Ein Engagement für Innovatio
Nachhaltigkeit
Études ist ein bahnbrechendes Unternehmen, das Kre
Funktionalität nahtlos miteinander verbindet, um archi
Spitzenleistungen neu zu definieren.
Über uns

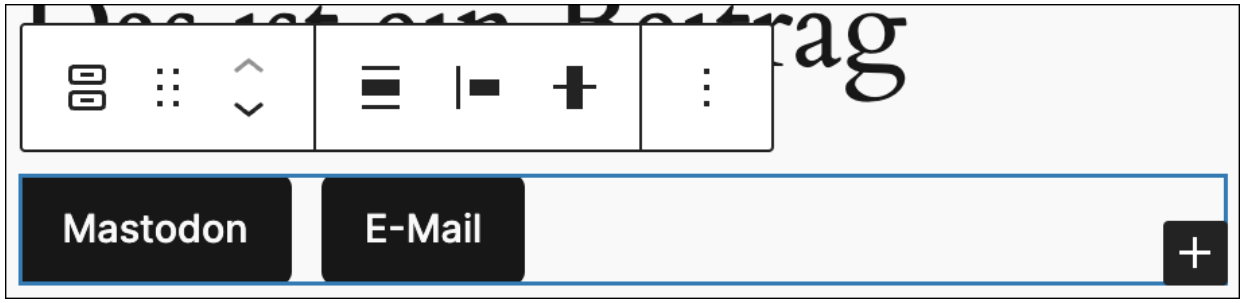

Ersetzen
0:00 / 16:20

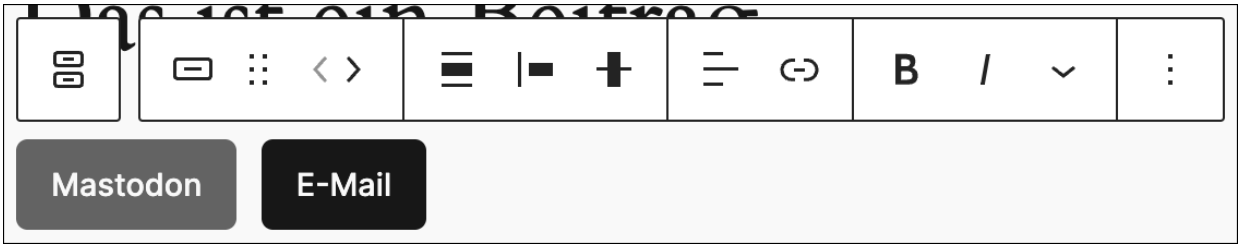

Ersetzen
My WordPress Website
Edit site
0
New
Howdy, admin
dPress Website
I am Johannes Vermeer, born in 1853. I paint with fervent passion, crafting celebrated works despite some personal struggles. My art resonates worldwide, immortalizing my unique perspective and inspiring generations with its enduring beauty.
Schreibe einen Titel…
My Blog
My Flickr
Buy My Artwork
Theme by Anders Norén


Ersetzen
URL kopieren
WordCamp Europe 2025
Herunterladen


Ersetzen
My WordPress Website
Edit site
New
Howdy, admin
rdPress Website
I am Johannes Vermeer, born in 1853. I paint with fervent passion, crafting celebrated works despite some personal struggles. My art resonates worldwide, immortalizing my unique perspective and inspiring generations with its enduring beauty.
My Blog
My Flickr
Buy My Artwork
Theme by Anders Norén
Vermeer-Theme


Texthinweise
Ersetzen
0:00 / 0:10

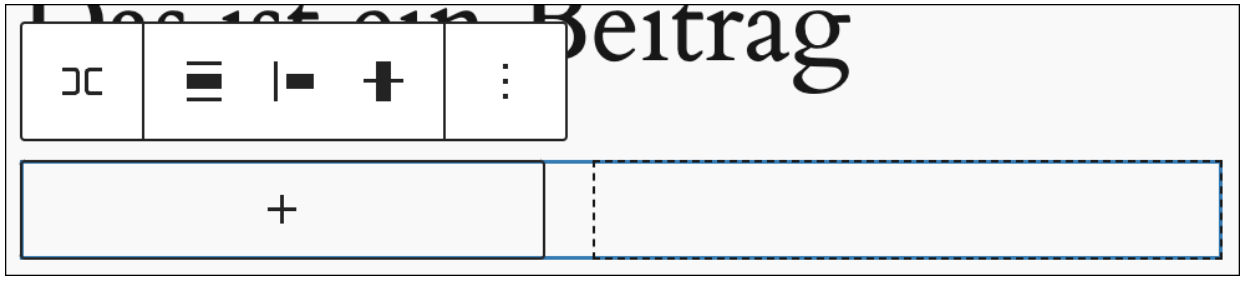

Mastodon
E-Mail

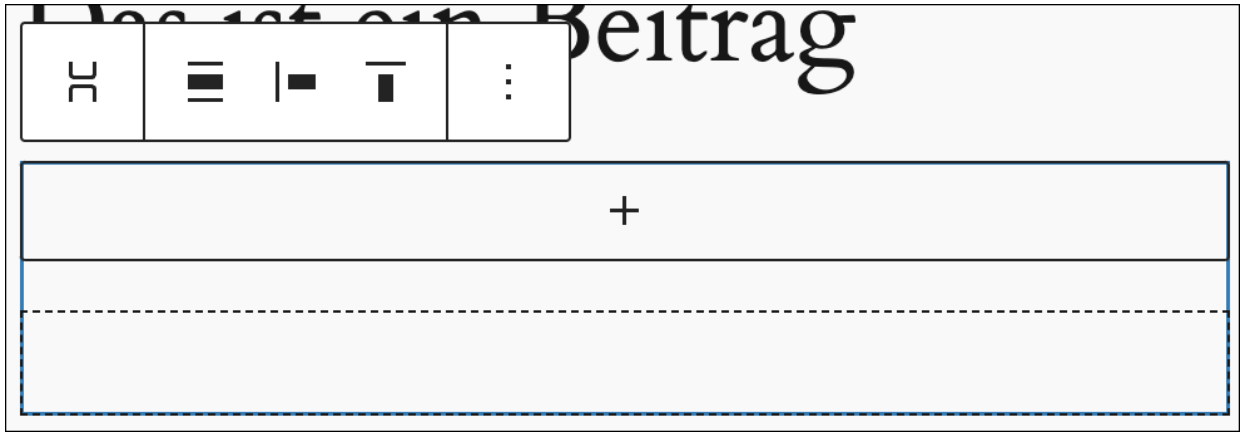

Mastodon
E-Mail

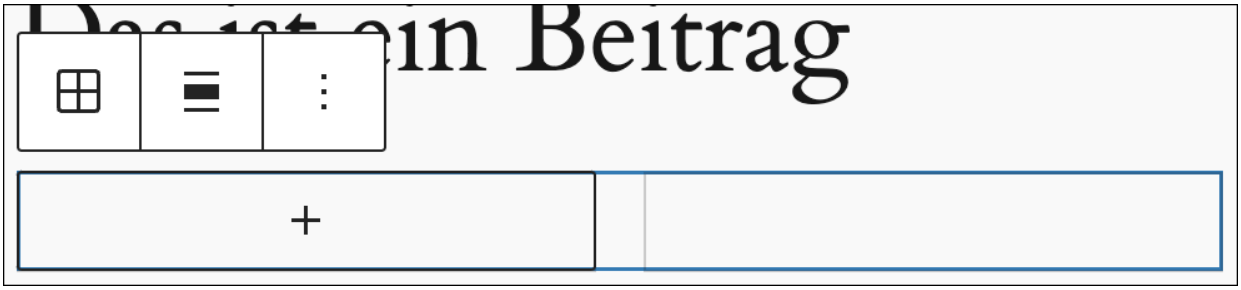

Beitrag
Text in der linken Spalte
Text in der rechten Spalte

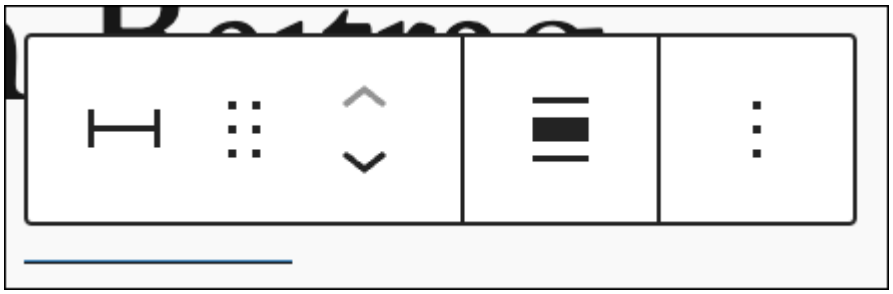

Text in der linken Spalte

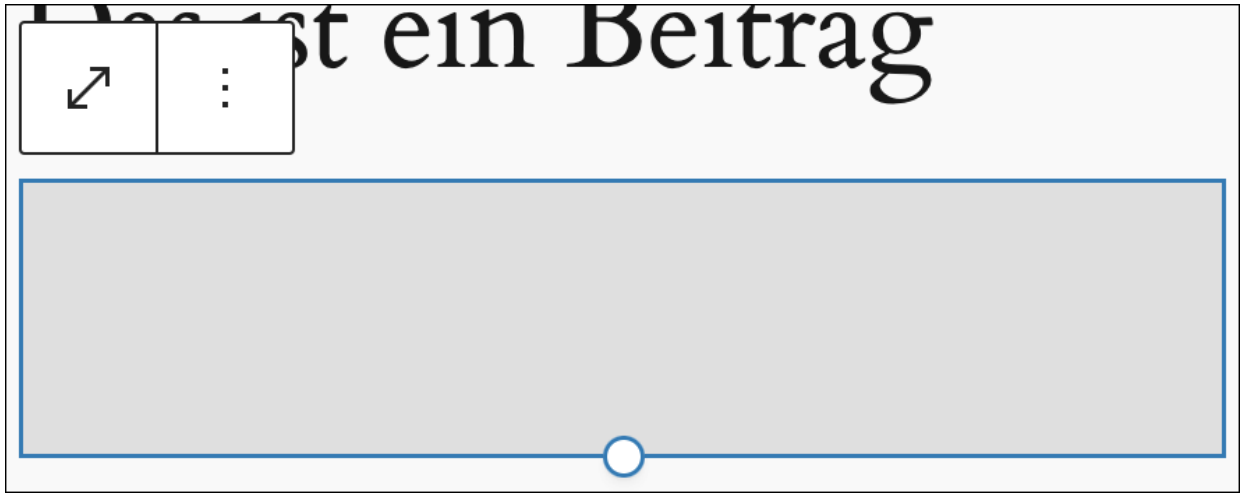

Beitrag
Absatz in einer Gruppe.

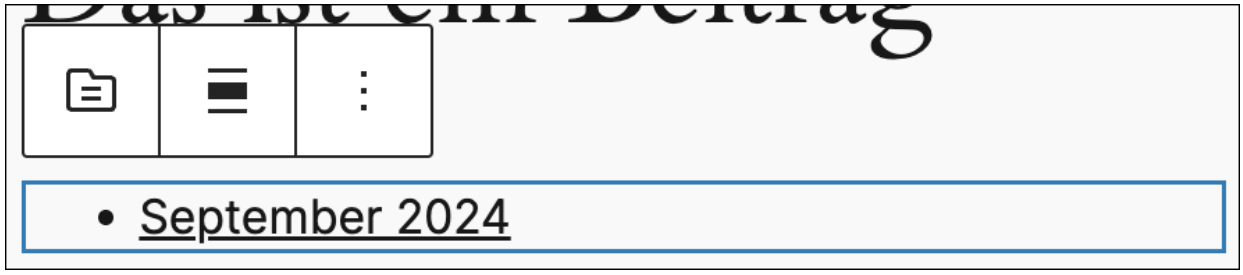

Beitrag


Beitrag

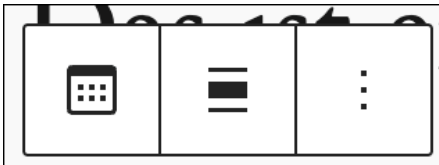

ein Beitrag
+




st ein Beitrag


Das ist ein Beitrag
September 2024

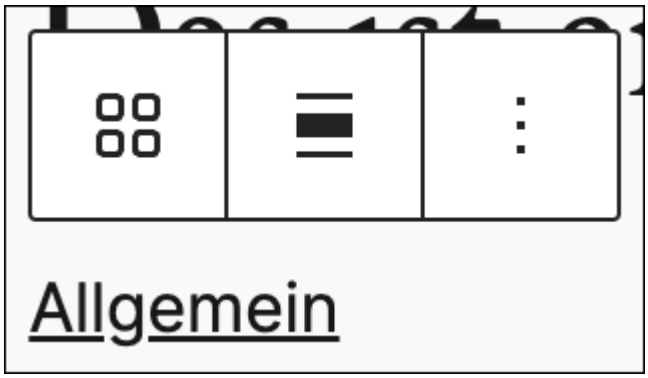
Das ist ein Beitrag



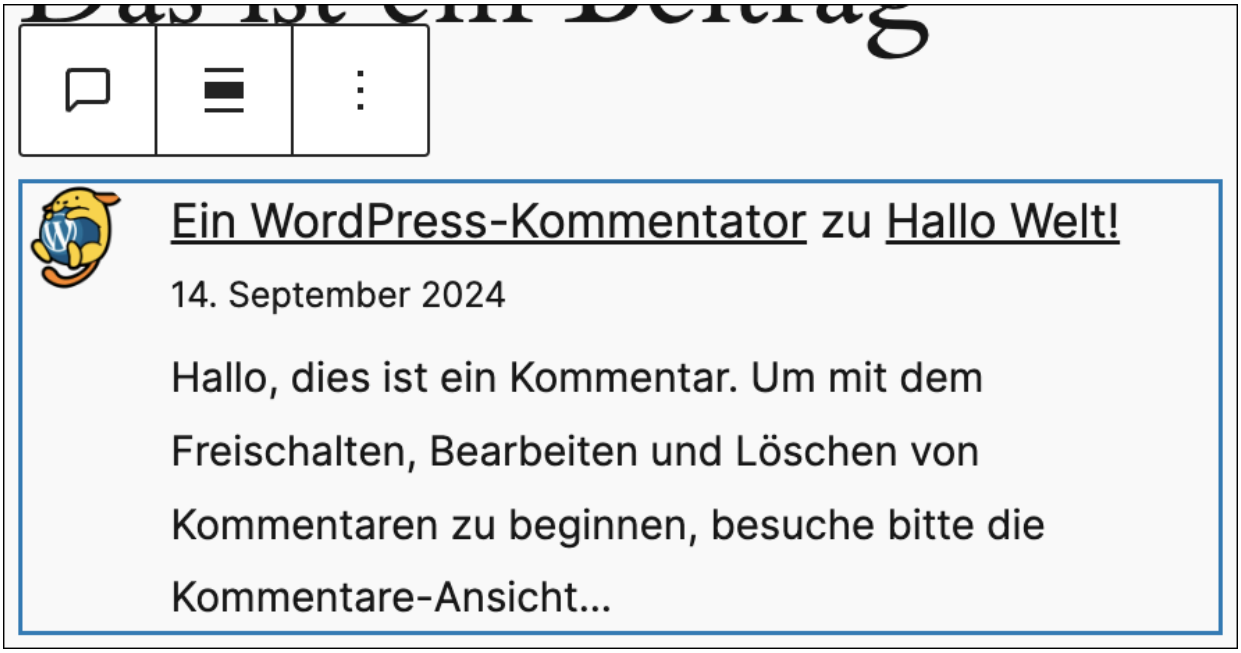
Oktober 2024

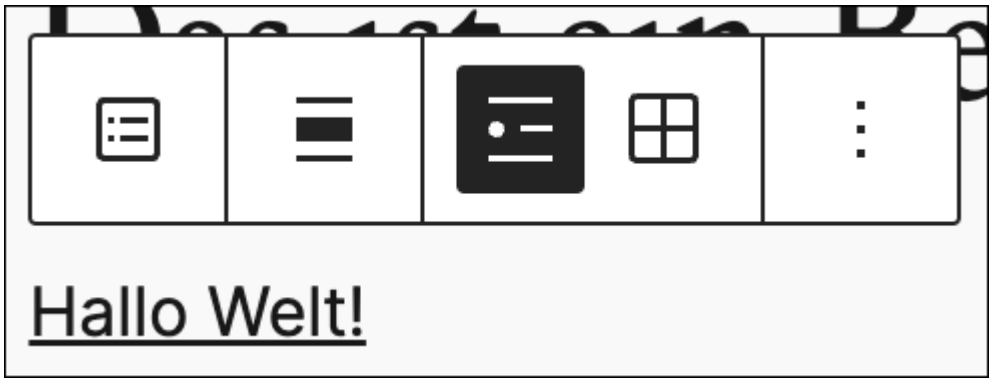
M	D	M	D	F	S	S
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

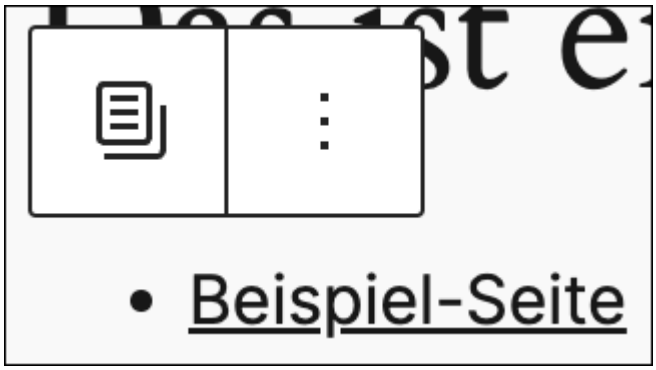
« Sep

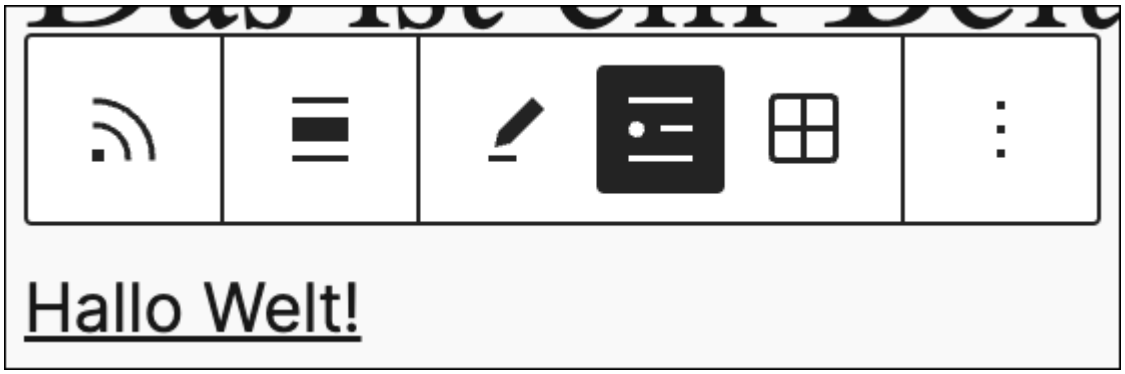

Allgemein

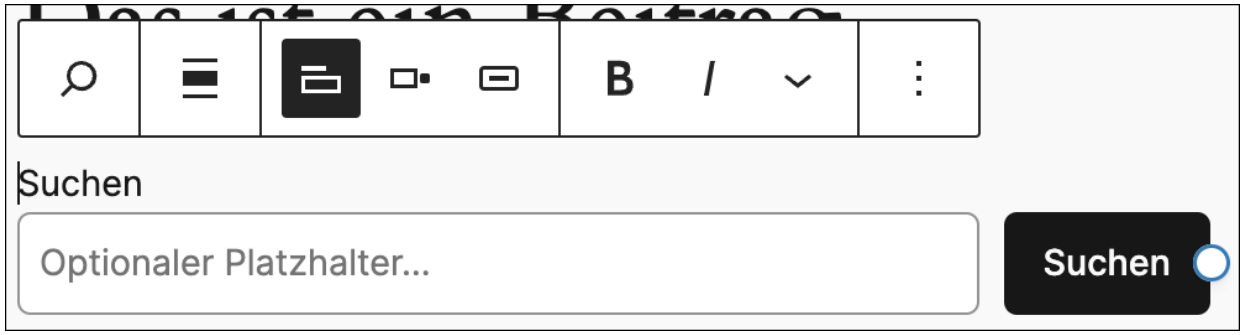

HTML
HTML
Vorschau
Schreibe etwas in HTML…

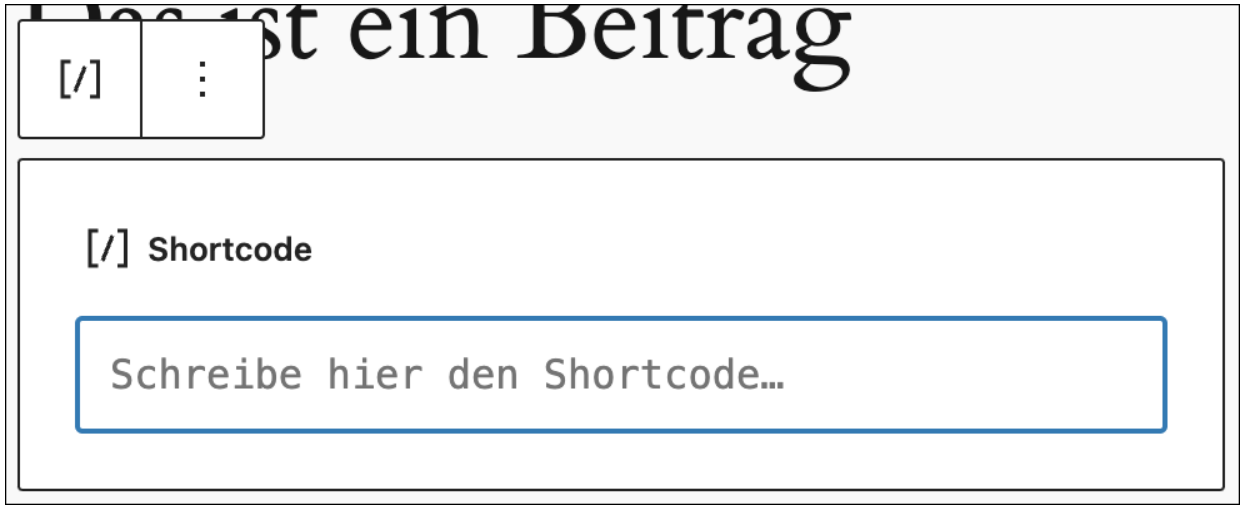

Ein WordPress-Kommentator zu Hallo Welt!
14. September 2024
Hallo, dies ist ein Kommentar. Um mit dem Freischalten, Bearbeiten und Löschen von Kommentaren zu beginnen, besuche bitte die Kommentare-Ansicht...

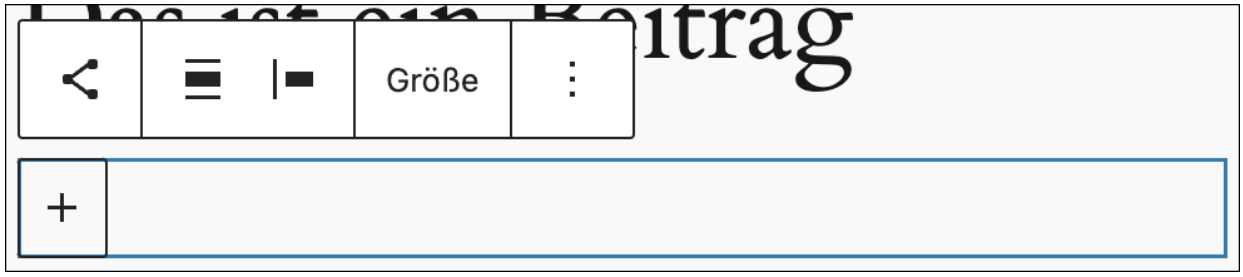

Hallo Welt!

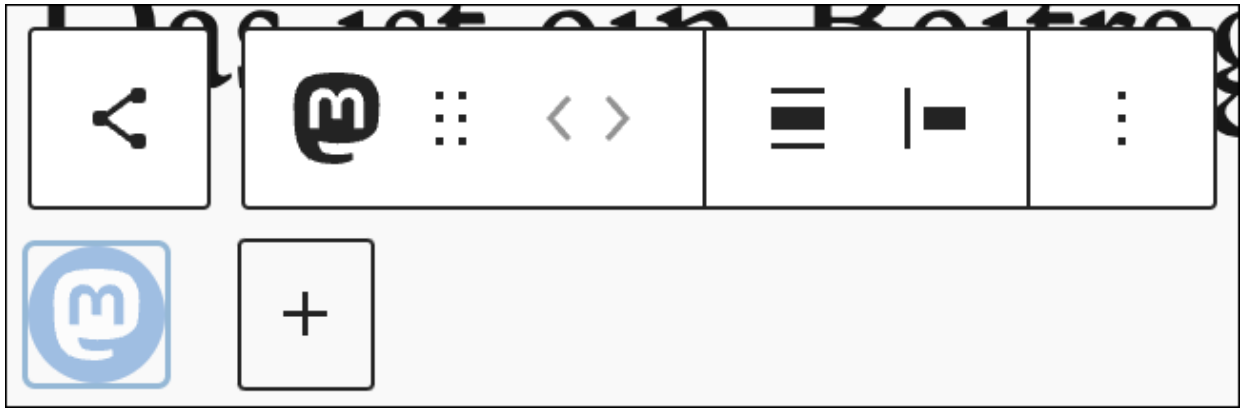

Beispiel-Seite

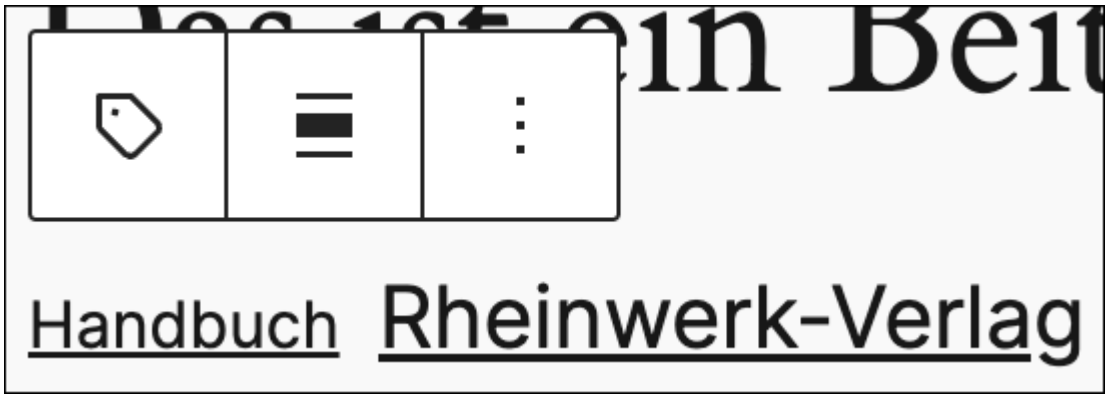

Hallo Welt!

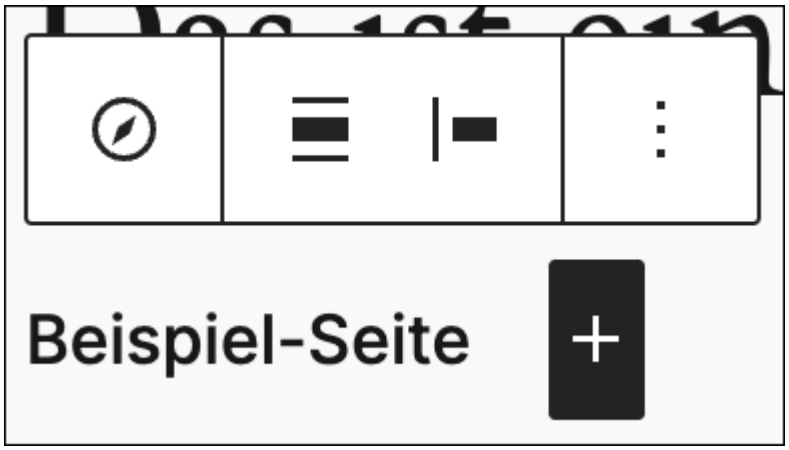

Suchen
Optionaler Platzhalter...
Suchen

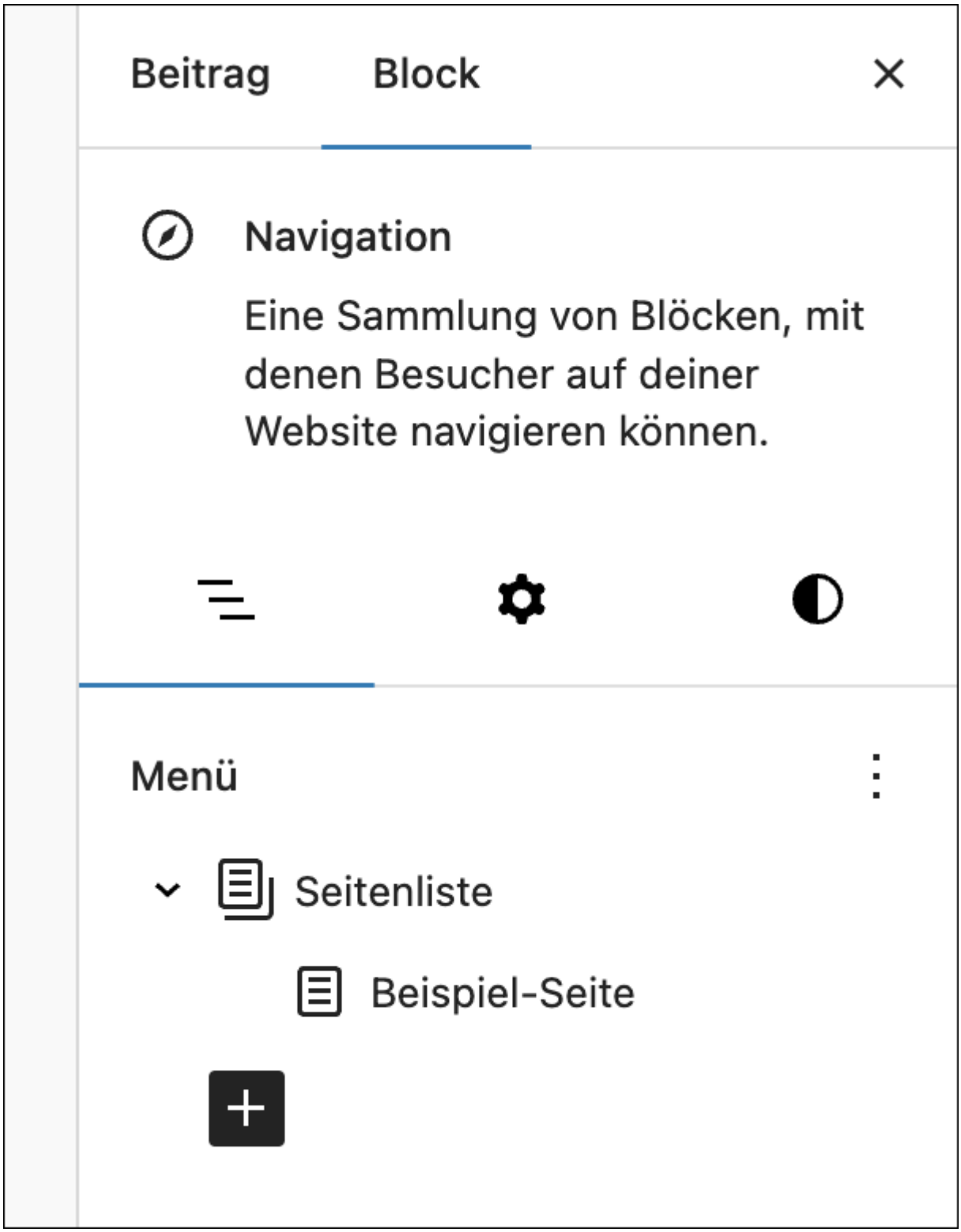

[/]
Shortcode
Schreibe hier den Shortcode…

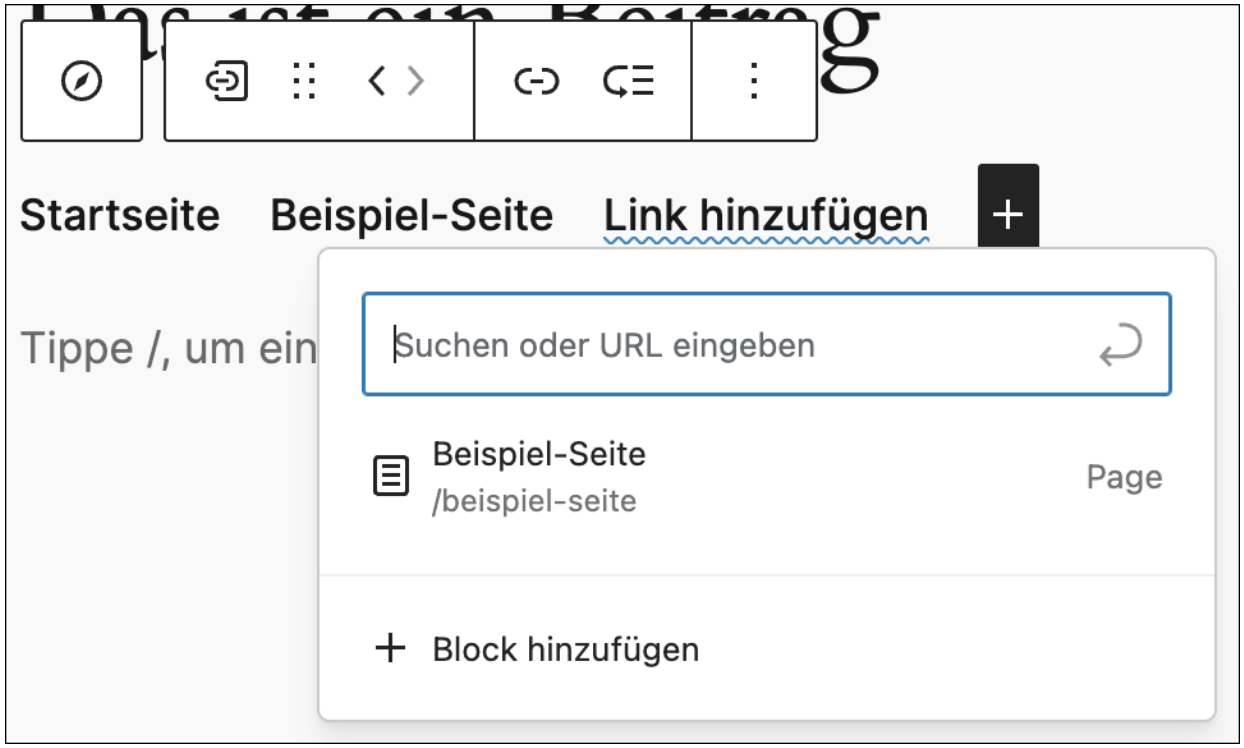

Größe

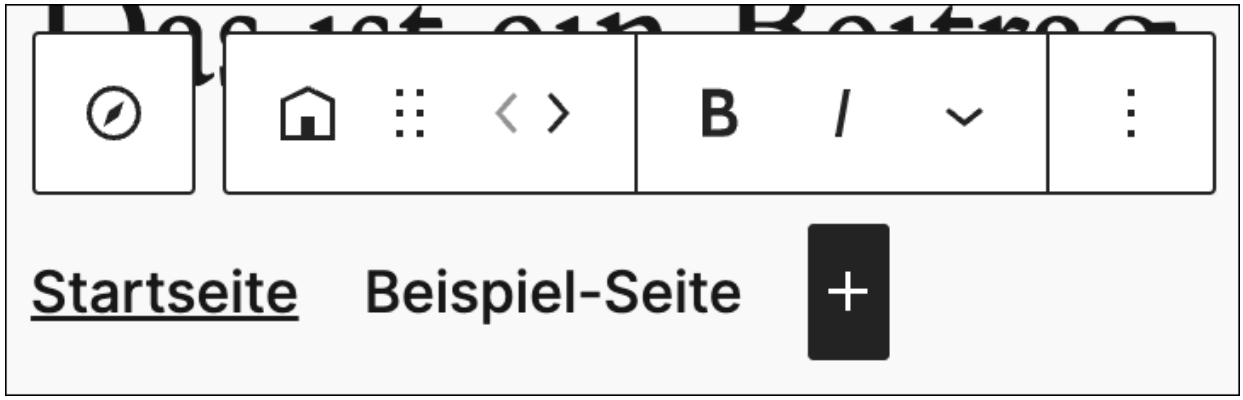


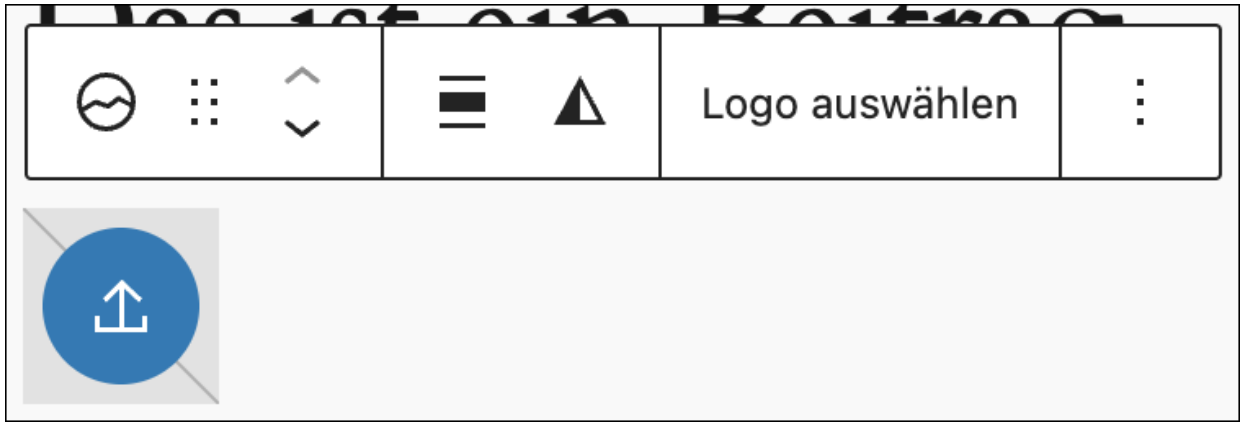

Handbuch
Rheinwerk-Verlag

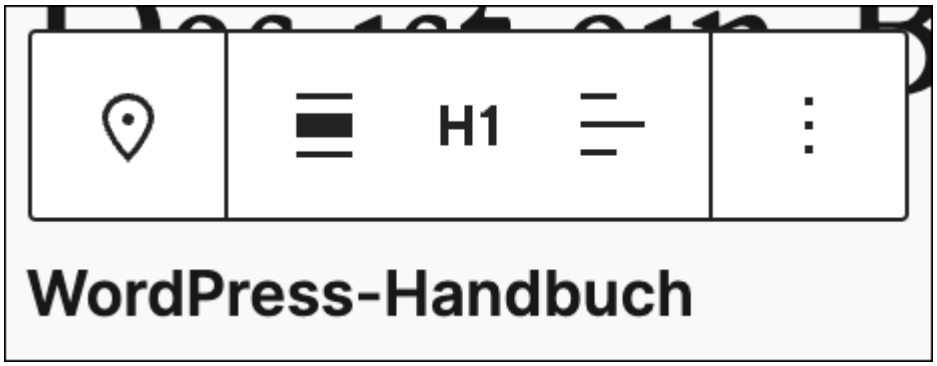

Beispiel-Seite

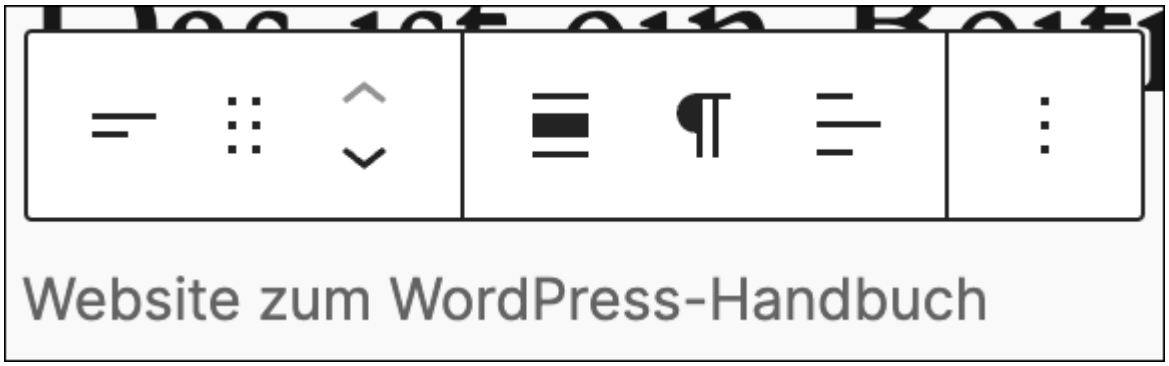

Beitrag
Block
Navigation
Eine Sammlung von Blöcken, mit denen Besucher auf deiner Website navigieren können.
Menü
Seitenliste
Beispiel-Seite


Startseite
Beispiel-Seite
Link hinzufügen
Tippe /, um ein
Suchen oder URL eingeben
Beispiel-Seite
/beispiel-seite
Page
Block hinzufügen

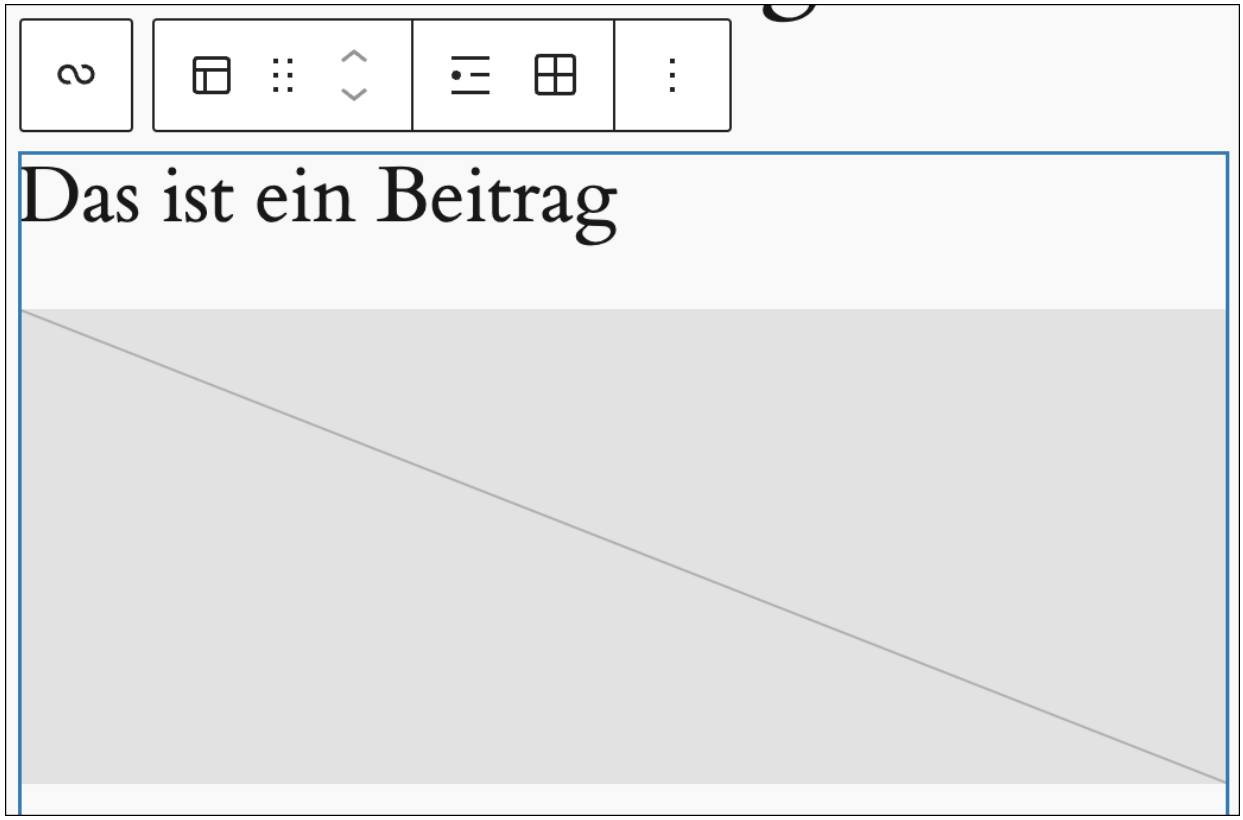

B
I
Startseite
Beispiel-Seite

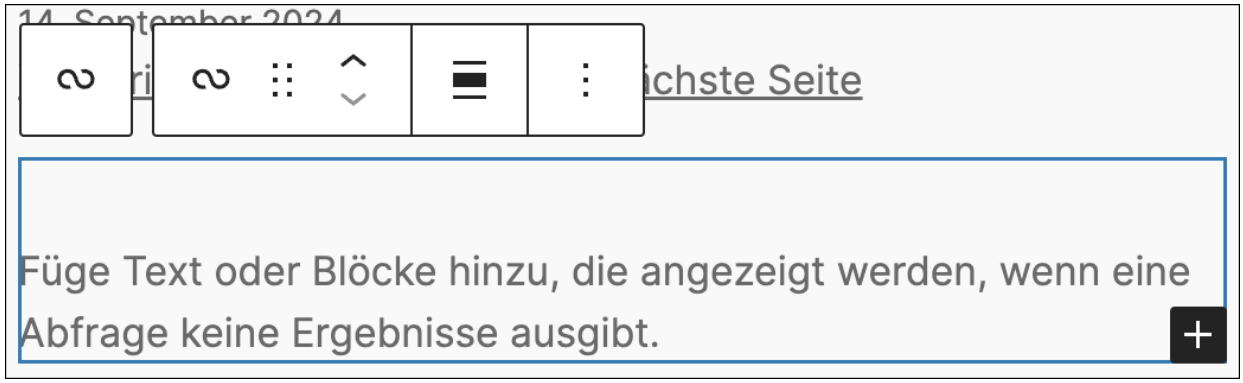

Logo auswählen

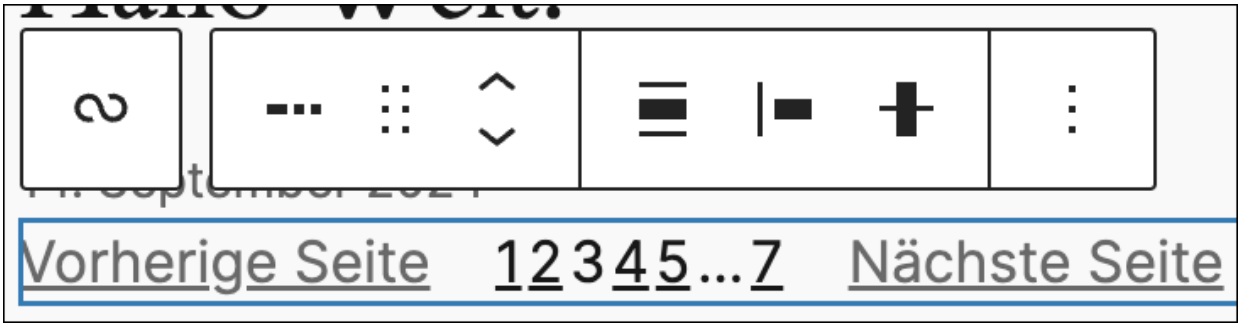

H1
WordPress-Handbuch

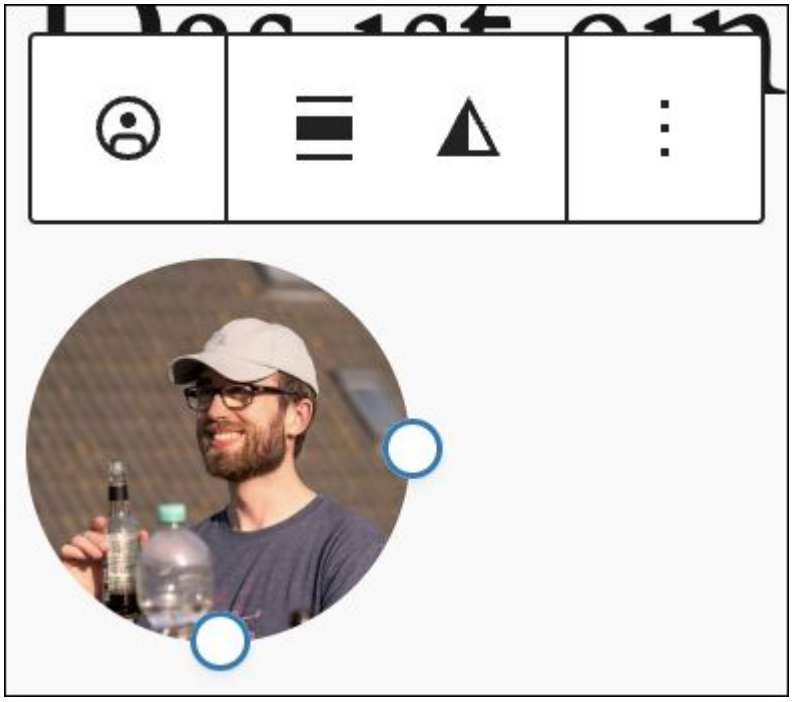

Website zum WordPress-Handbuch

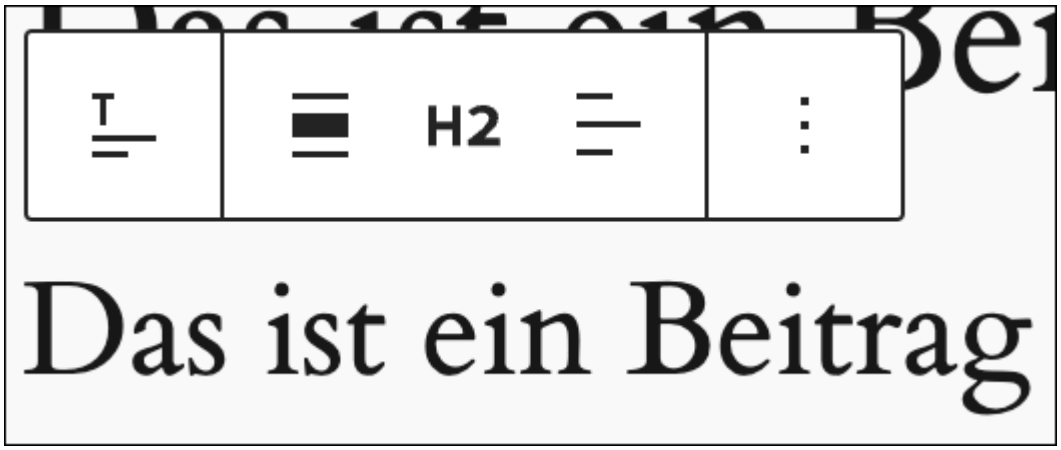

Ersetzen
Das ist ein Beitrag
Es wurde kein Textauszug gefunden
„Weiterlesen“-Linktext hinzufügen

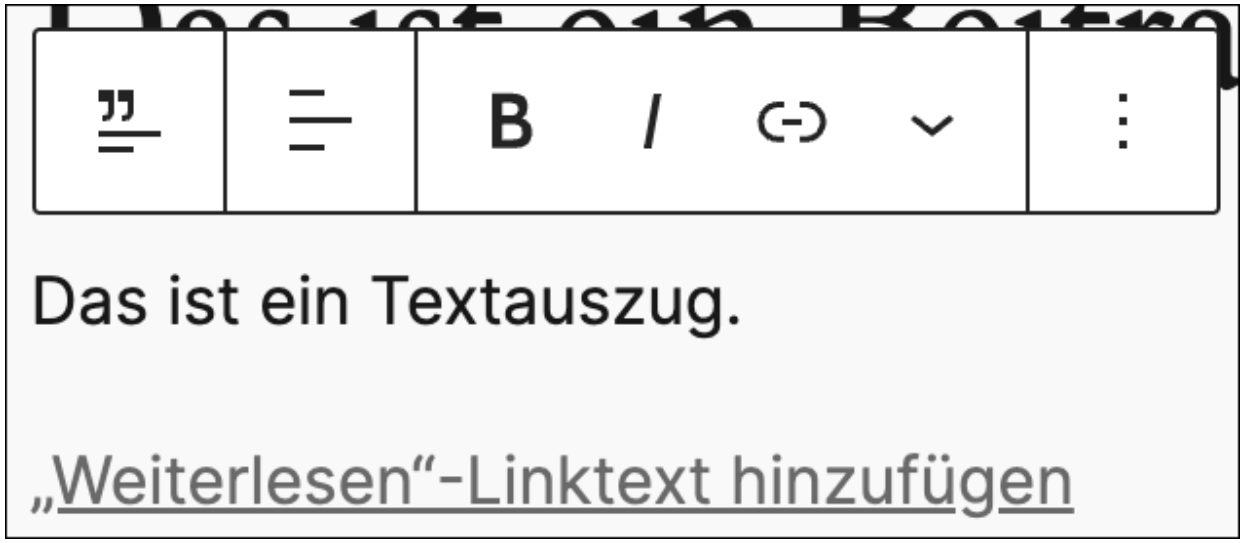

2
Das ist ein Beitrag

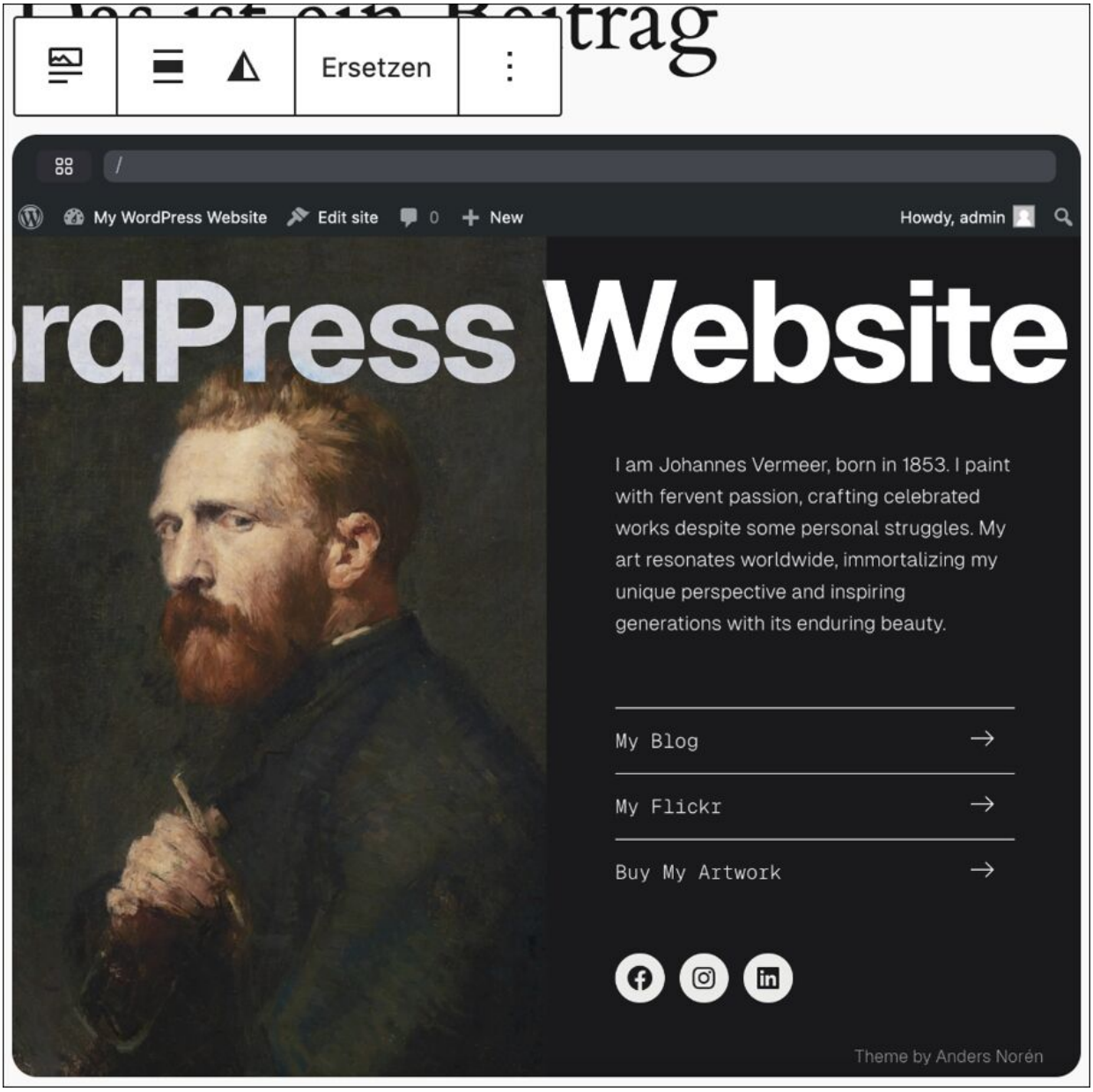

14. September 2024
ri
ächste Seite
Füge Text oder Blöcke hinzu, die angezeigt werden, wenn eine Abfrage keine Ergebnisse ausgibt.

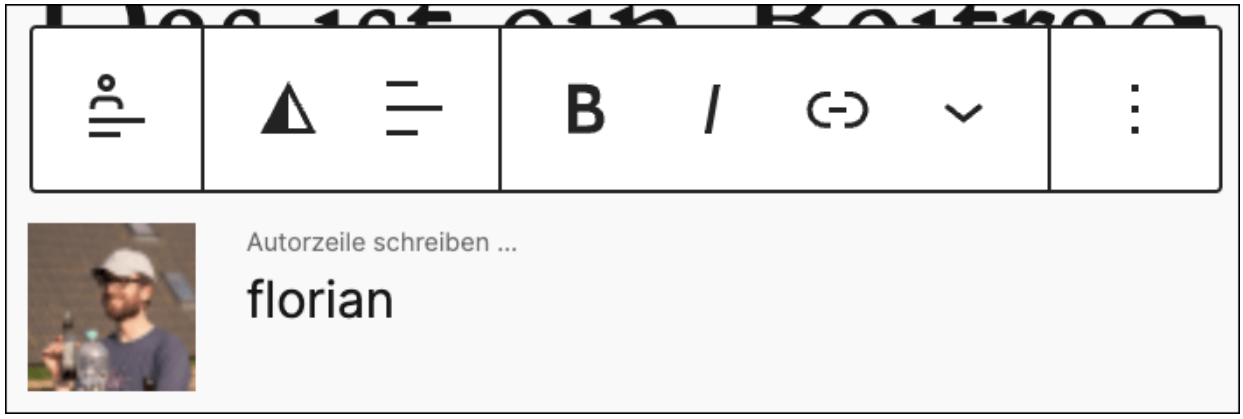

Vorherige Seite
1 2 3 4 5 … 7
Nächste Seite

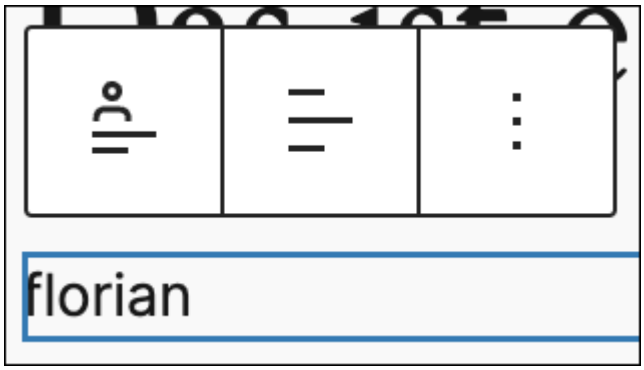


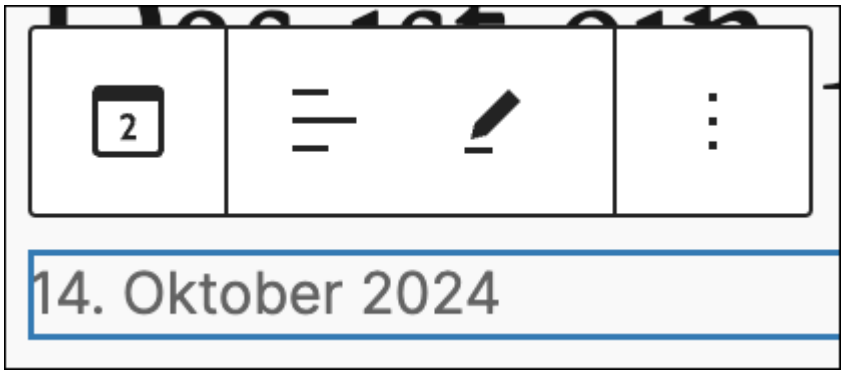

H2
Das ist ein Beitrag

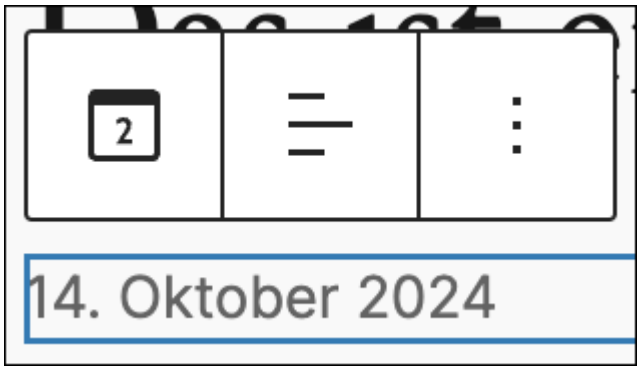

B
I
Das ist ein Textauszug.
„Weiterlesen“-Linktext hinzufügen

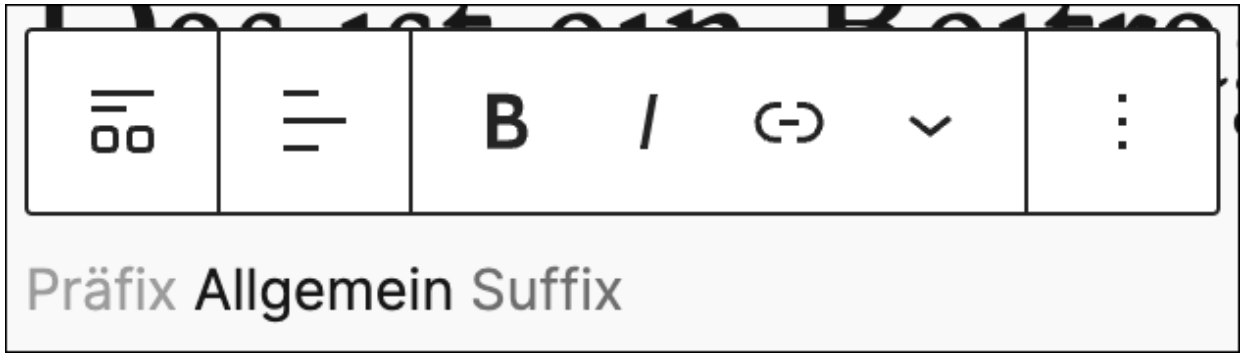

Ersetzen
/
My WordPress Website
Edit site
0
New
Howdy, admin
rdPress Website
I am Johannes Vermeer, born in 1853. I paint with fervent passion, crafting celebrated works despite some personal struggles. My art resonates worldwide, immortalizing my unique perspective and inspiring generations with its enduring beauty.
My Blog
My Flickr
Buy My Artwork
Theme by Anders Norén

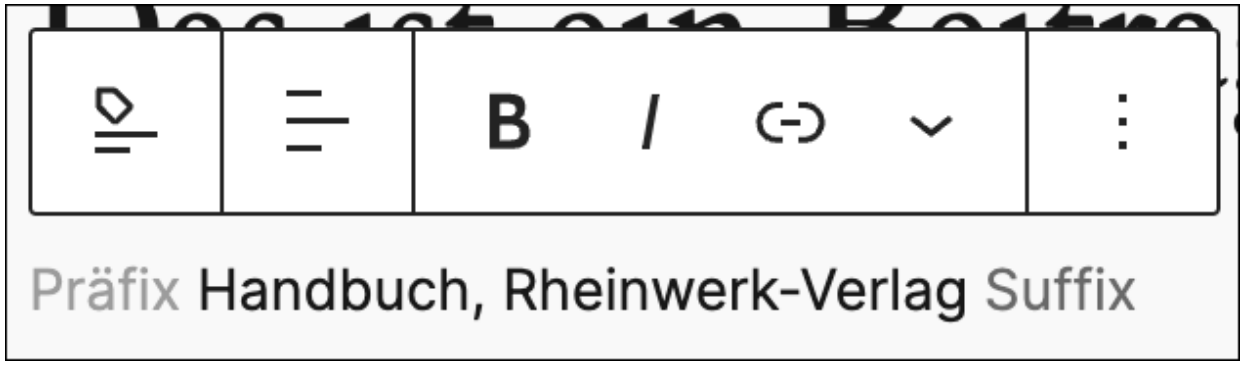

B
I
Autorzeile schreiben ...
florian

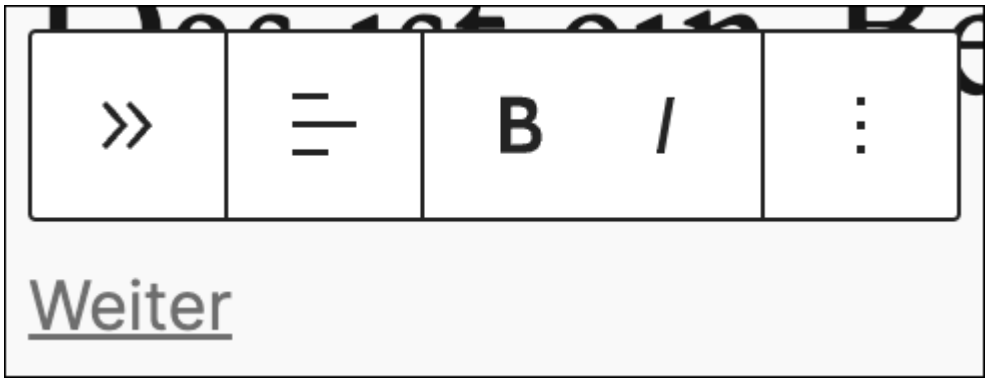

florian

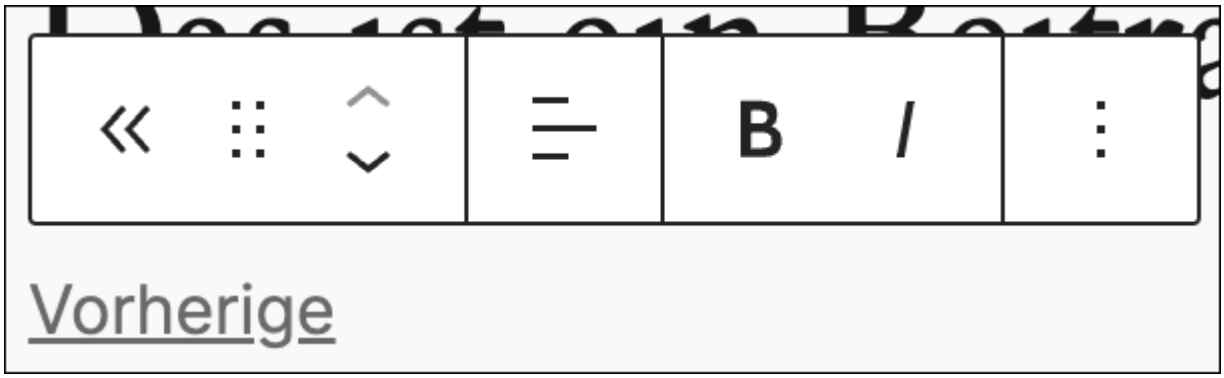

14. Oktober 2024

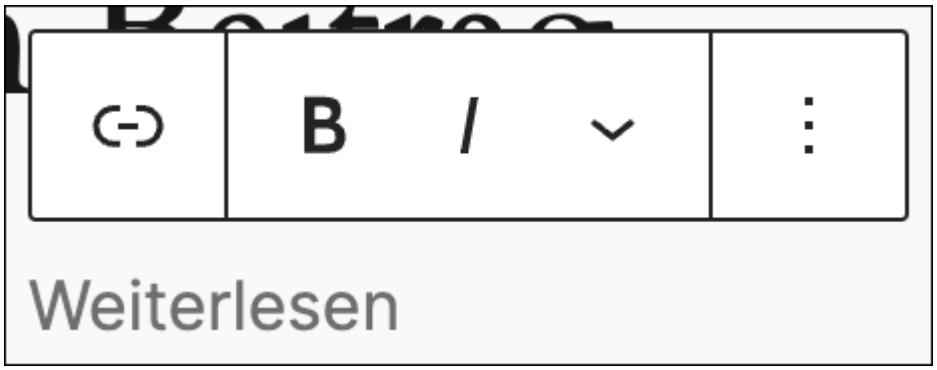

14. Oktober 2024


B
I
Präfix Allgemein Suffix

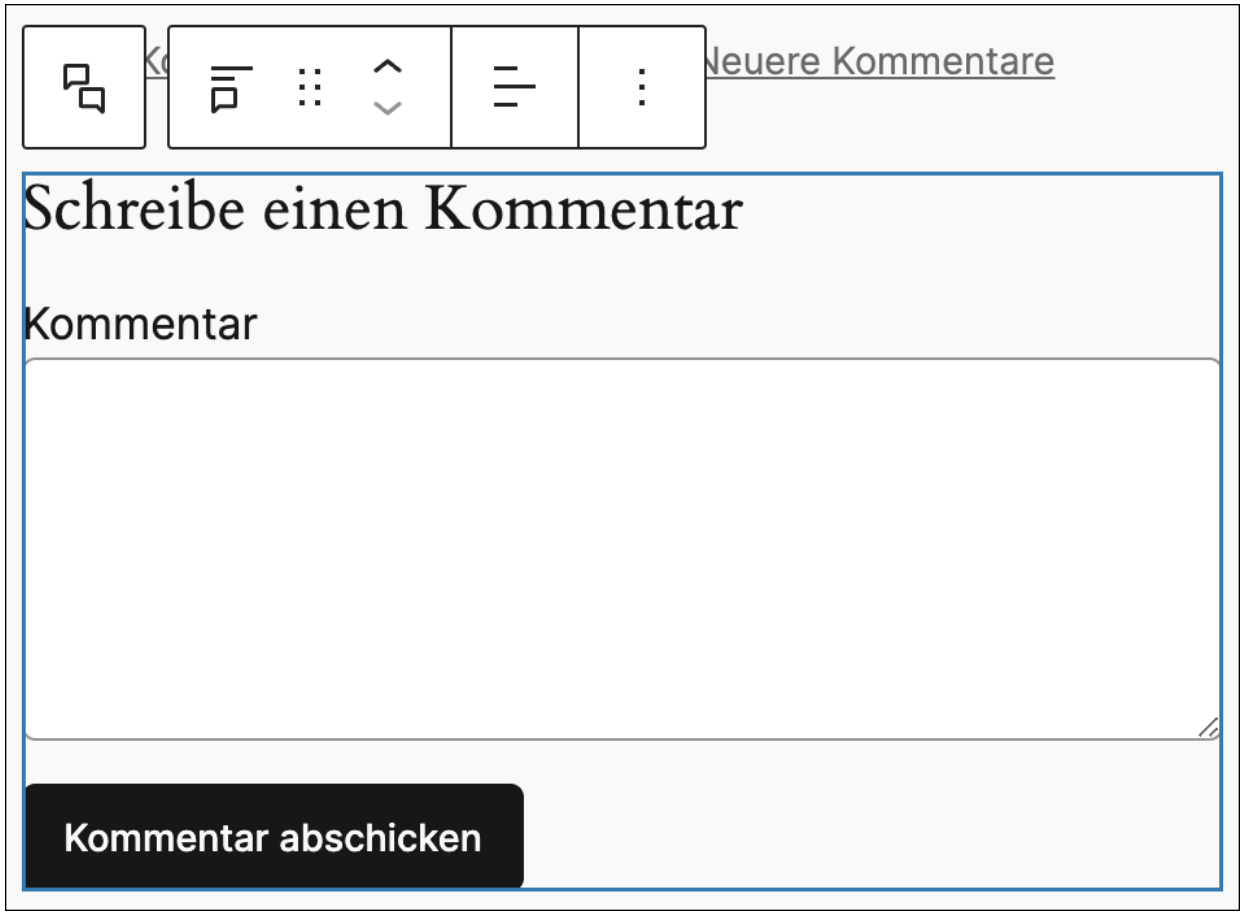

B
I
Präfix Handbuch, Rheinwerk-Verlag Suffix

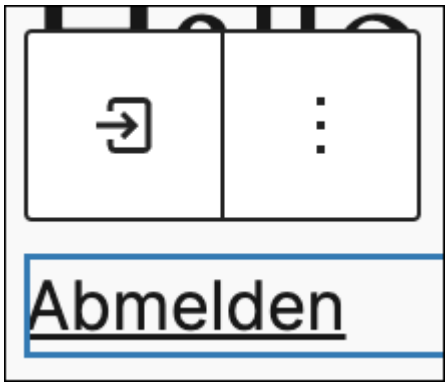

B
I
Weiter

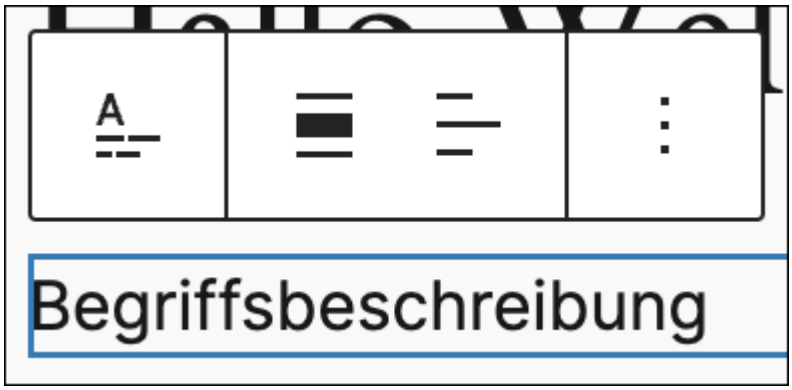

B
I
Vorherige

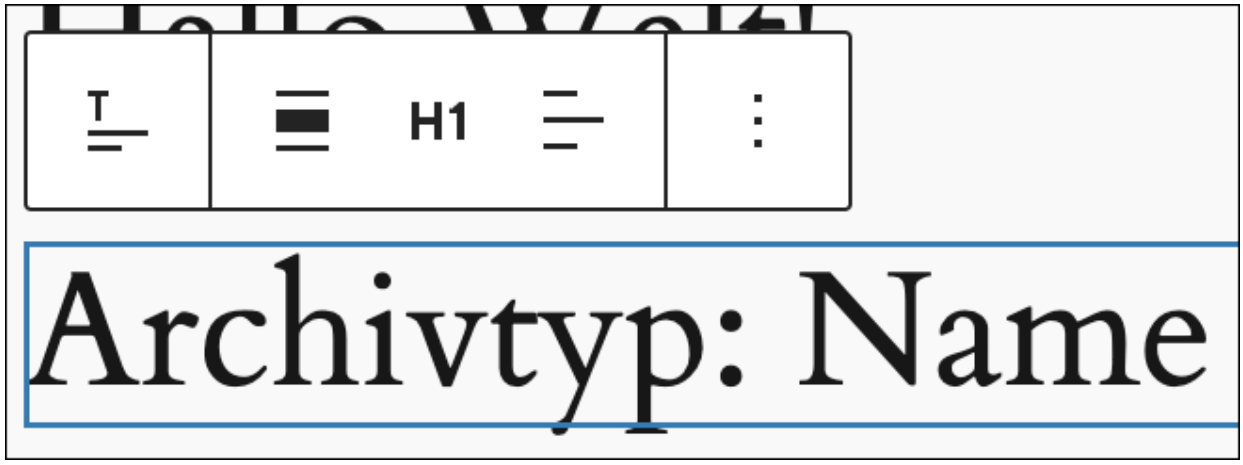

B
I
Weiterlesen


Eine Antwort zu "Hallo Welt!"
Ein WordPress-Kommentator
2024-09-14T17:27:24 Bearbeiten
Hallo, dies ist ein Kommentar.
Antworten
Kommentare-Seitennummerierungs-Block: Das Durchblättern der Kommentare ist in den Diskussionseinstellungen deaktiviert
Schreibe einen Kommentar
Kommentar
Kommentar abschicken

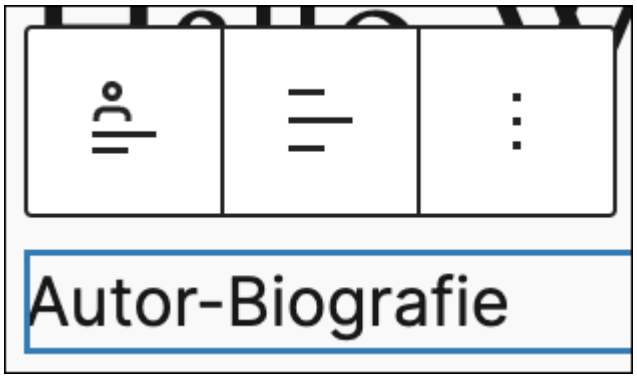

Neuere Kommentare
Schreibe einen Kommentar
Kommentar
Kommentar abschicken


Abmelden


Begriffsbeschreibung


H1
Archivtyp: Name


H1
Suchergebnisse für:
„Suchbegriff“


Autor-Biografie


Hallo Welt! · Beitrag
⌘K
Hallo Welt!
Das ist ein Beitrag
Es wurde kein Textauszug gefunden
„Weiterlesen“-Linktext hinzufügen
14. Oktober 2024
Hallo Welt!

Abbildung 6.15 Wenn die Verkleinern-Funktion aktiv ist, wird der Editor-Bereich verkleinert und nicht mehr über die gesamte verfügbare Breite angezeigt, und Sie können nur noch Blöcke der obersten Ebene auswählen. Wenn Sie den Block-Inserter öffnen, wird direkt der Tab mit den »Vorlagen« ausgewählt.

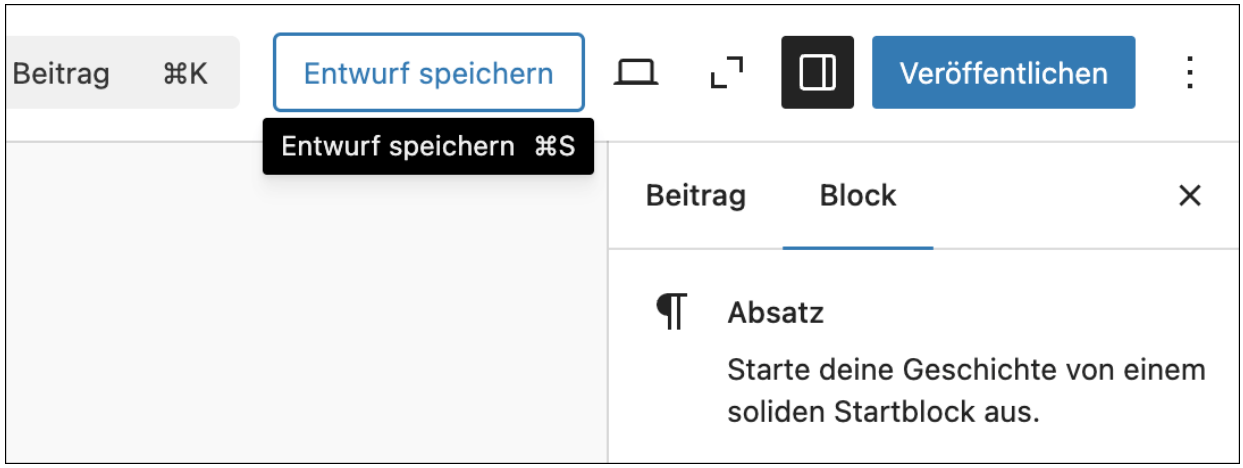


Abbildung 6.16 Sie können auch vor der Veröffentlichung jederzeit einen Zwischenstand mit Klick auf »Entwurf speichern« sichern.

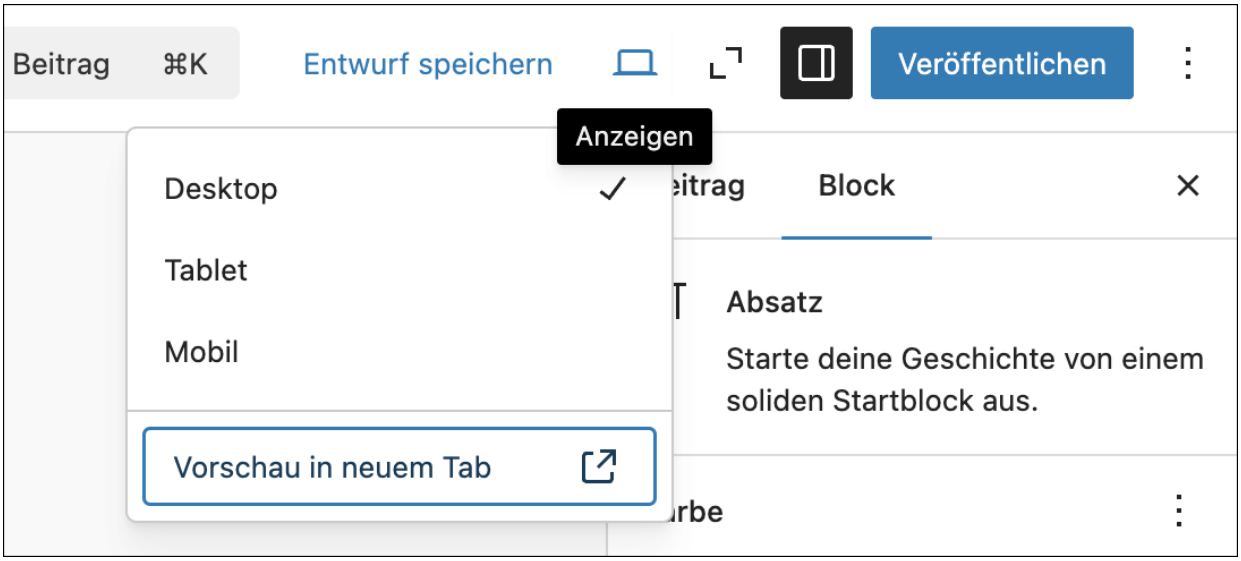


Abbildung 6.17 Sie können sich jederzeit vor der Veröffentlichung den aktuellen Stand eines Beitrags, einer Seite oder eines anderen Inhalts über die Vorschau im Frontend anschauen.



Abbildung 6.18 Sie können einfach ein Bild von Ihrem lokalen System auf den Bild-Block ziehen, um es in die Mediathek hochzuladen und im Block anzuzeigen.

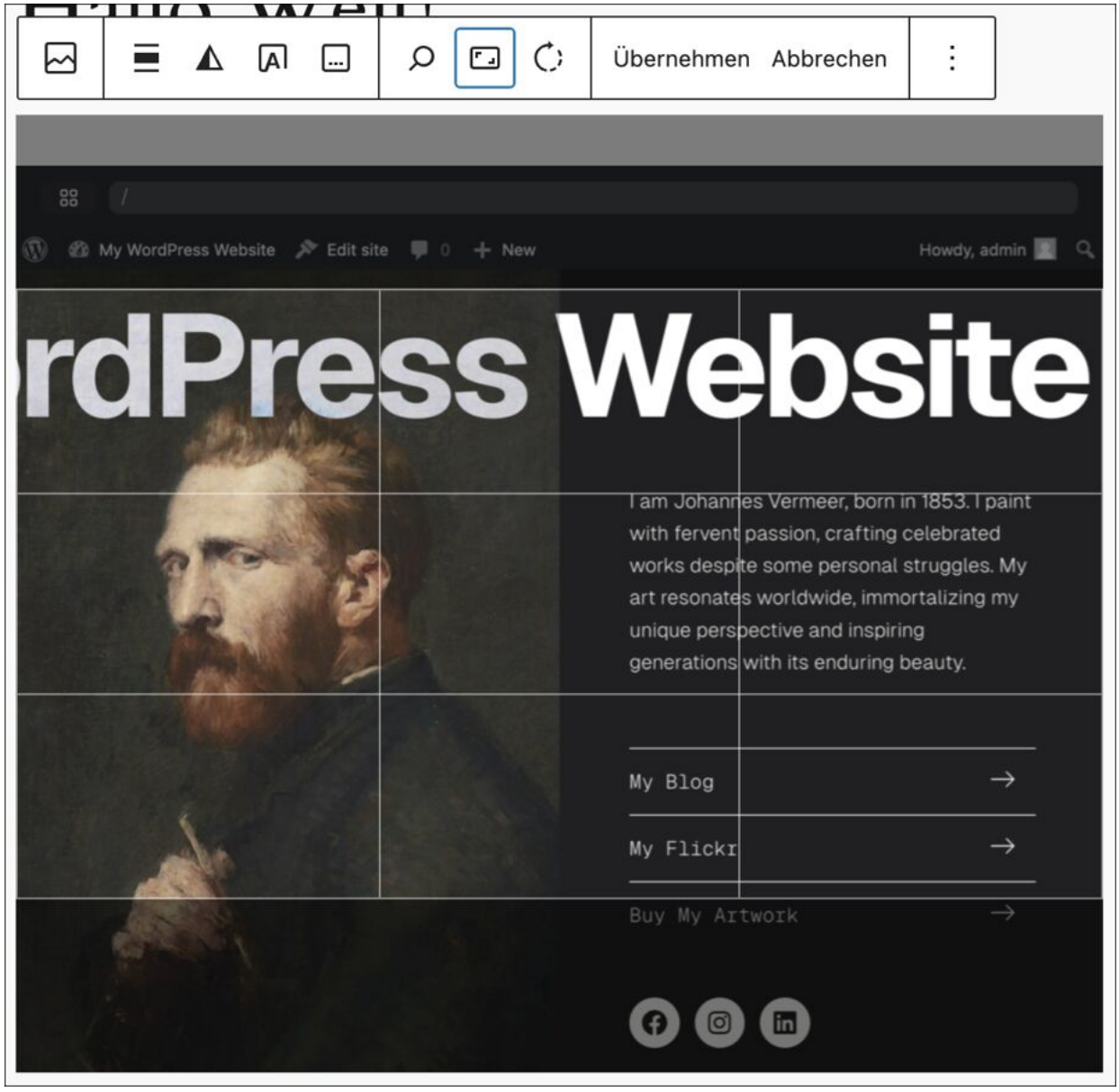


Abbildung 6.19 Beim Bild-Block bietet WordPress ein grundlegendes Werkzeug an, um das Bild zuzuschneiden und zu drehen.

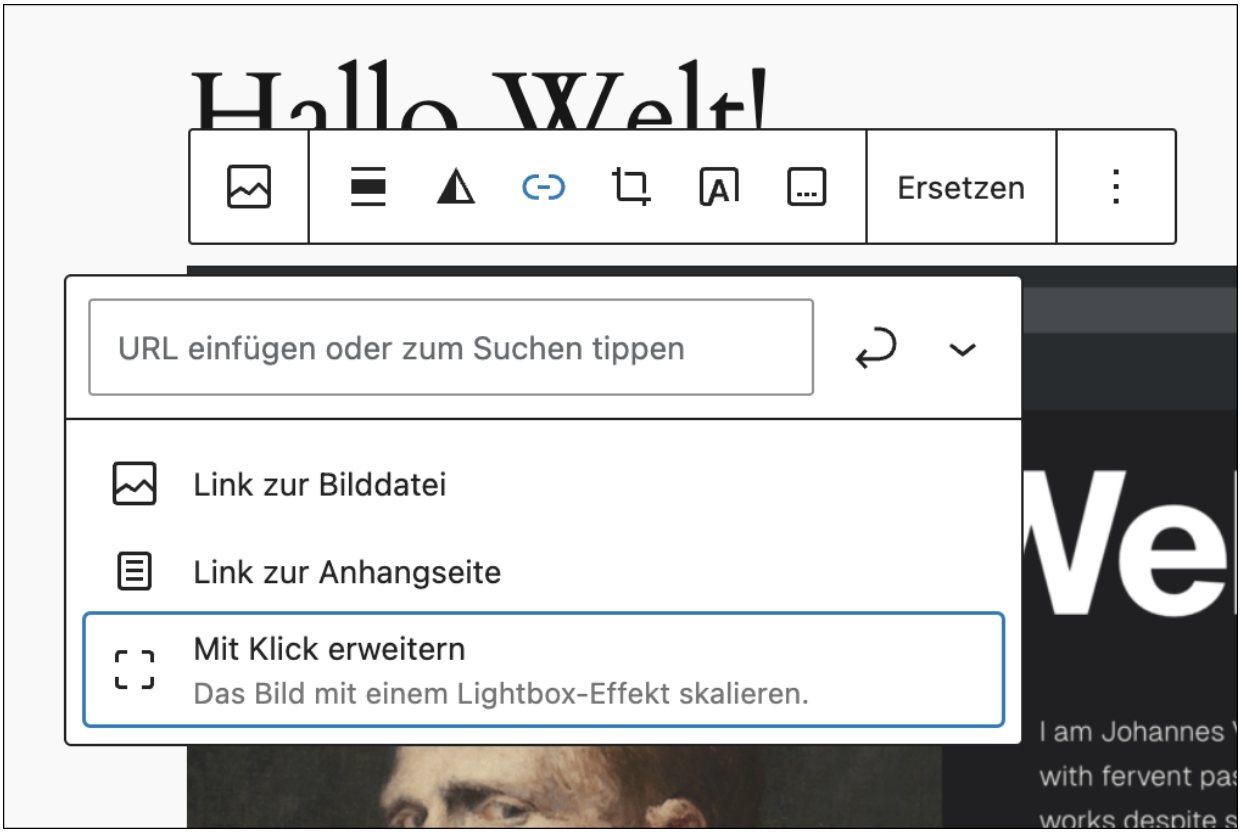


Abbildung 6.20 Sie können einstellen, dass das Bild bei Klick in einer Lightbox dargestellt wird.

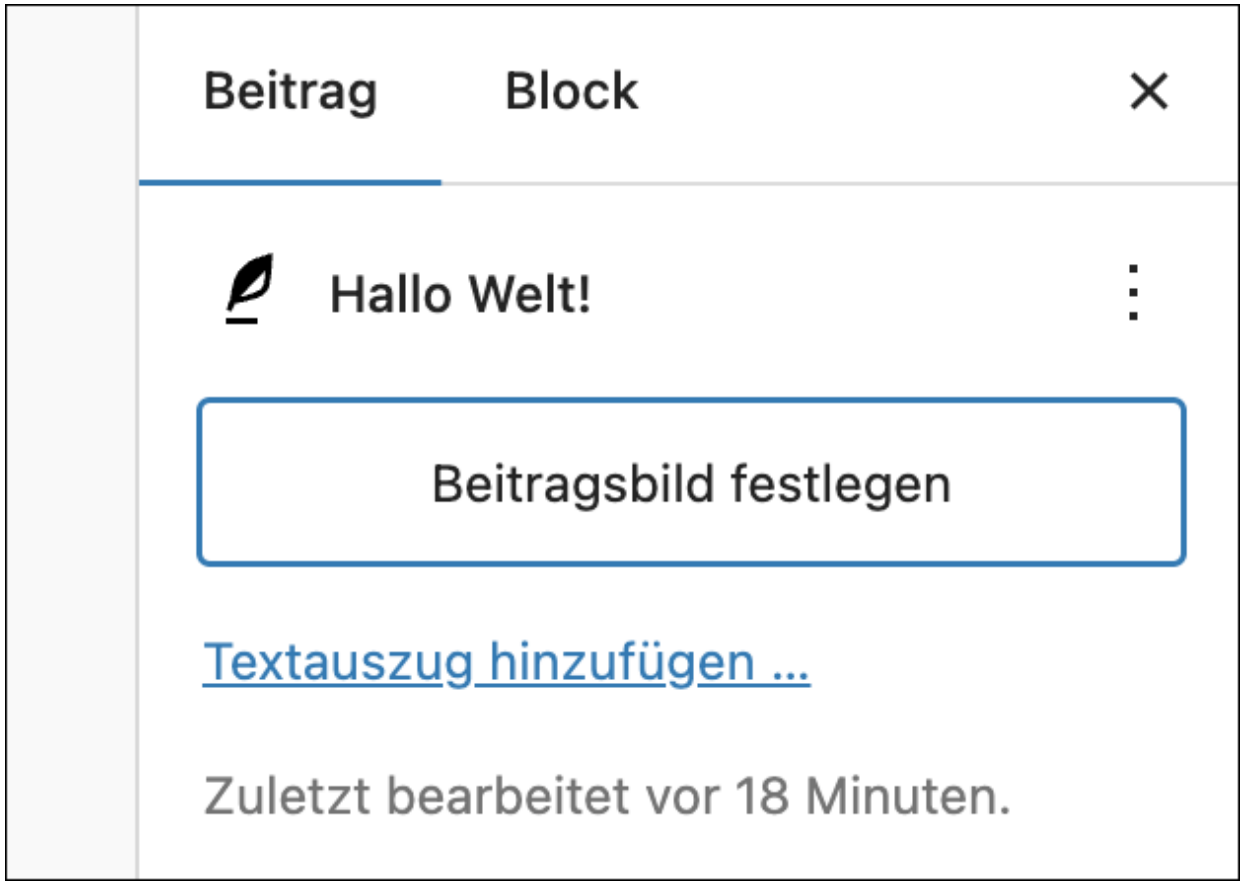


Abbildung 6.21 In der Dokumenten-Seitenleiste können Sie ein Beitragsbild festlegen, das vom Theme an unterschiedlichen Stellen genutzt werden kann, etwa in der Beitragsübersicht.

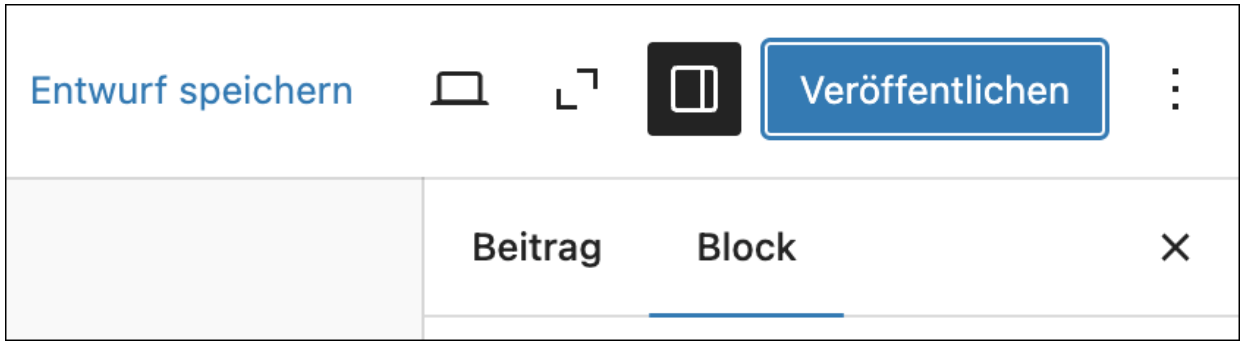
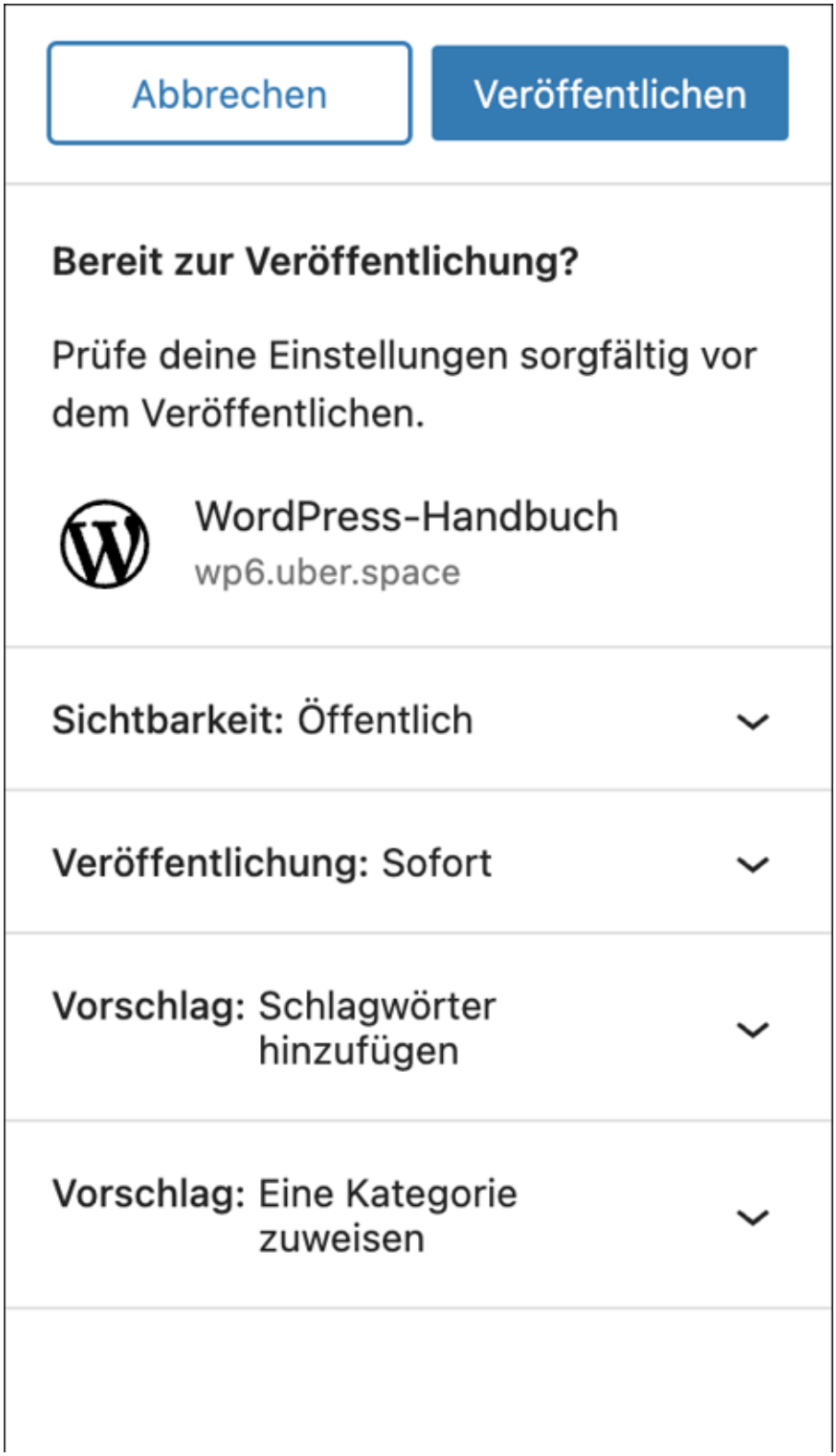


Abbildung 6.22 Schwer zu übersehen: der Button zur Veröffentlichung Ihres Beitrags oder Ihrer Seite


Abbrechen
Veröffentlichen
Bereit zur Veröffentlichung?
Prüfe deine Einstellungen sorgfältig vor dem Veröffentlichen.
WordPress-Handbuch
wp6.uber.space
Sichtbarkeit: Öffentlich
Veröffentlichung: Sofort
Vorschlag: Schlagwörter hinzufügen
Vorschlag: Eine Kategorie zuweisen

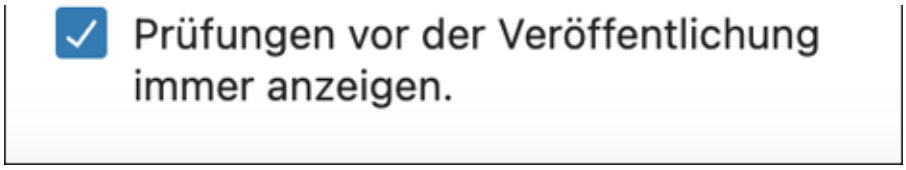


Abbildung 6.23 Vor der Veröffentlichung wird ein Dialog angezeigt, der für Sie zusammenfasst, wie der Inhalt veröffentlicht wird.

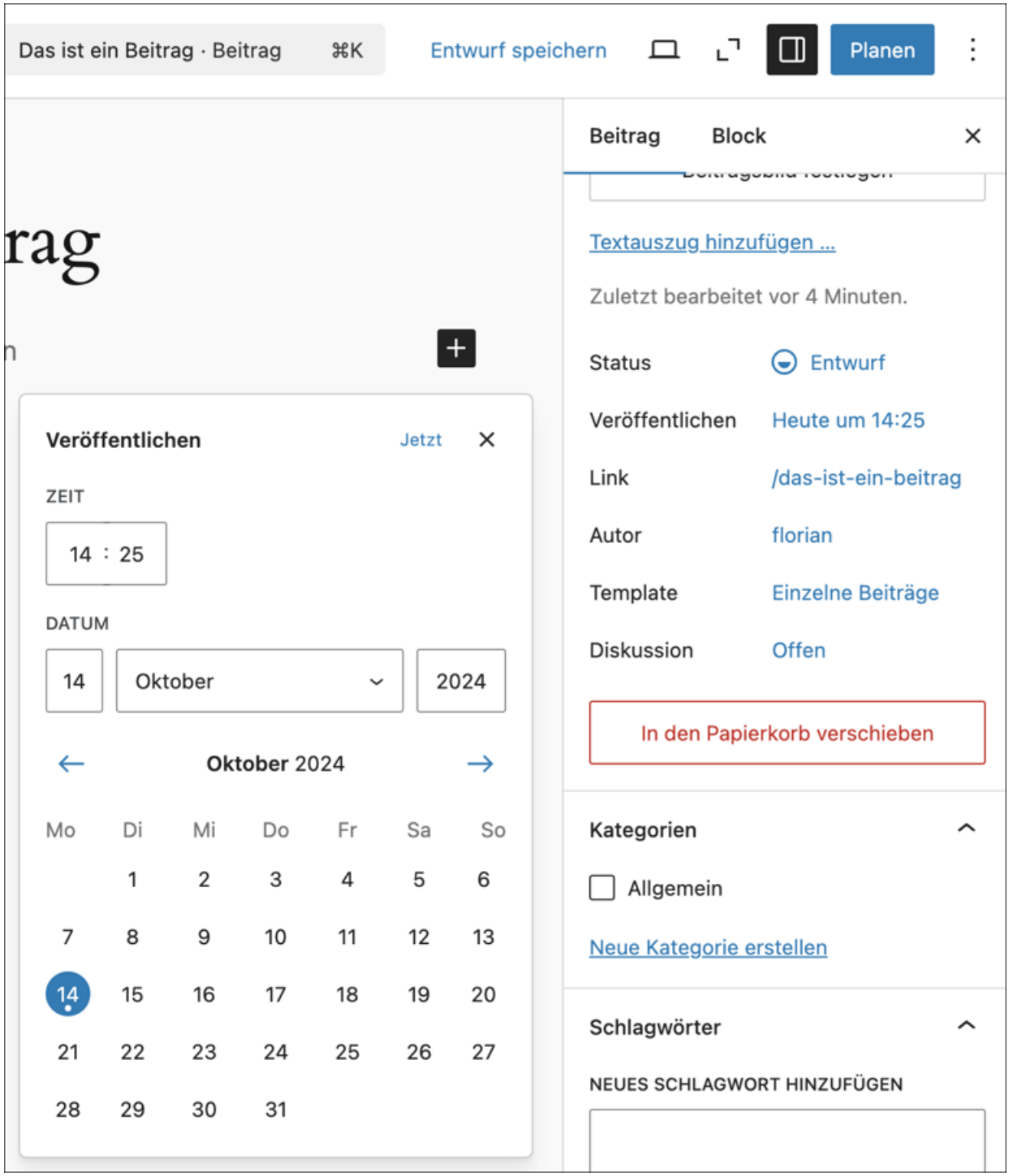

Das ist ein Beitrag · Beitrag ⌘K
Entwurf speichern
Planen
Beitrag
Block
rag
Veröffentlichen
Jetzt
ZEIT
14 : 25
DATUM
14
Oktober
2024
Oktober 2024
Mo Di Mi Do Fr Sa So
1 2 3 4 5 6
7 8 9 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
28 29 30 31
Textauszug hinzufügen …
Zuletzt bearbeitet vor 4 Minuten.
Status
Entwurf
Veröffentlichen
Heute um 14:25
Link
/das-ist-ein-beitrag
Autor
florian
Template
Einzelne Beiträge
Diskussion
Offen
In den Papierkorb verschieben
Kategorien
Allgemein
Neue Kategorie erstellen
Schlagwörter
NEUES SCHLAGWORT HINZUFÜGEN

Abbildung 6.24 Sie können einen Inhalt auch für die Veröffentlichung planen, sodass er zu einem bestimmten Zeitpunkt online geht. Rechts oben ist die Beschriftung des Buttons dann nicht mehr »Veröffentlichen«, sondern »Planen«.

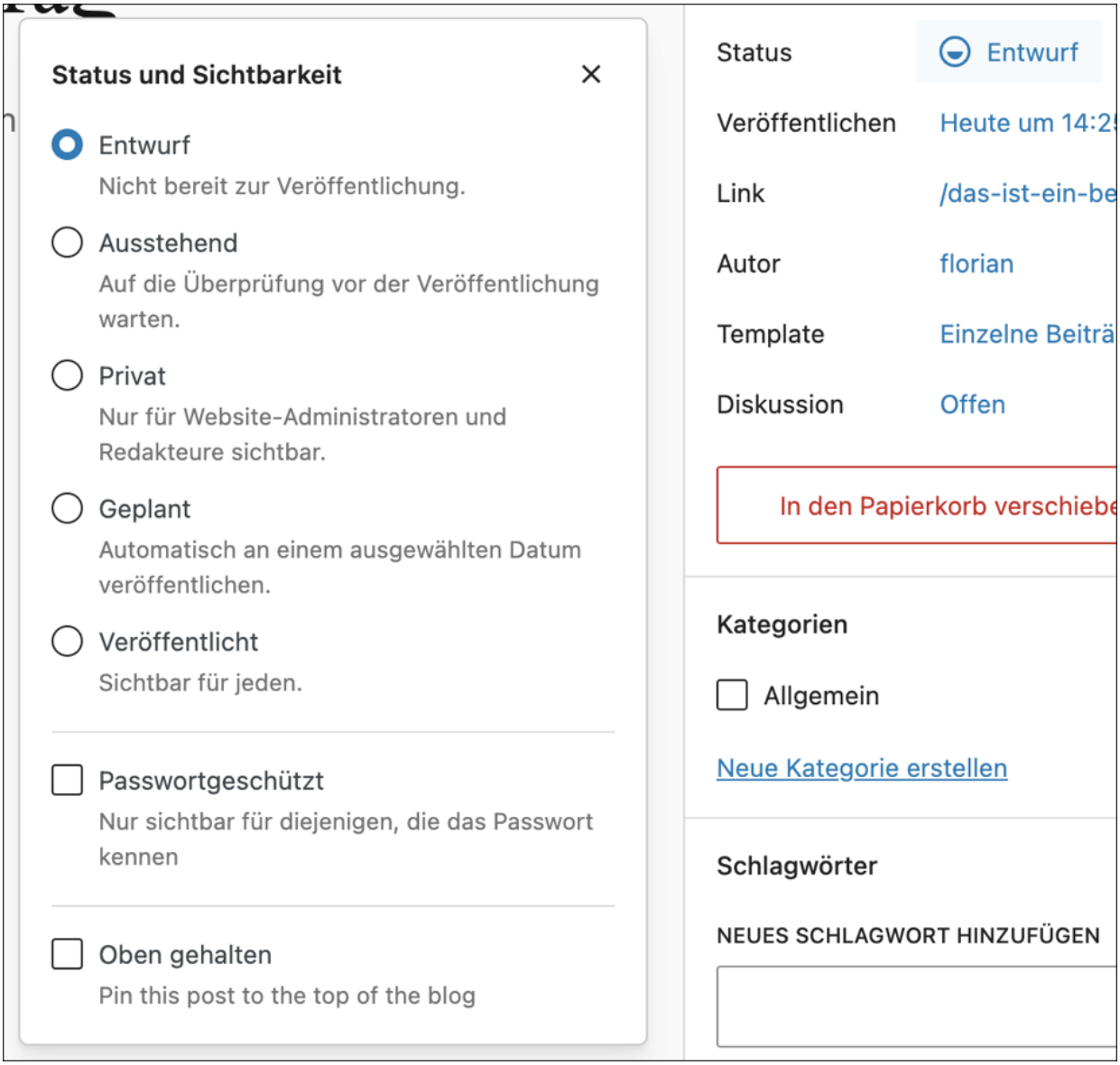


Abbildung 6.25 Beiträge und Seiten können unterschiedliche Status haben.



Abbildung 6.26 Ein privater Inhalt wird mit einem »Privat:«-Präfix vor dem Titel gekennzeichnet.



Abbildung 6.27 Bei einem passwortgeschützten Inhalt werden im Frontend nur der Titel und ein Passwortfeld angezeigt. Vor dem Titel wird ein »Geschützt:« eingefügt.

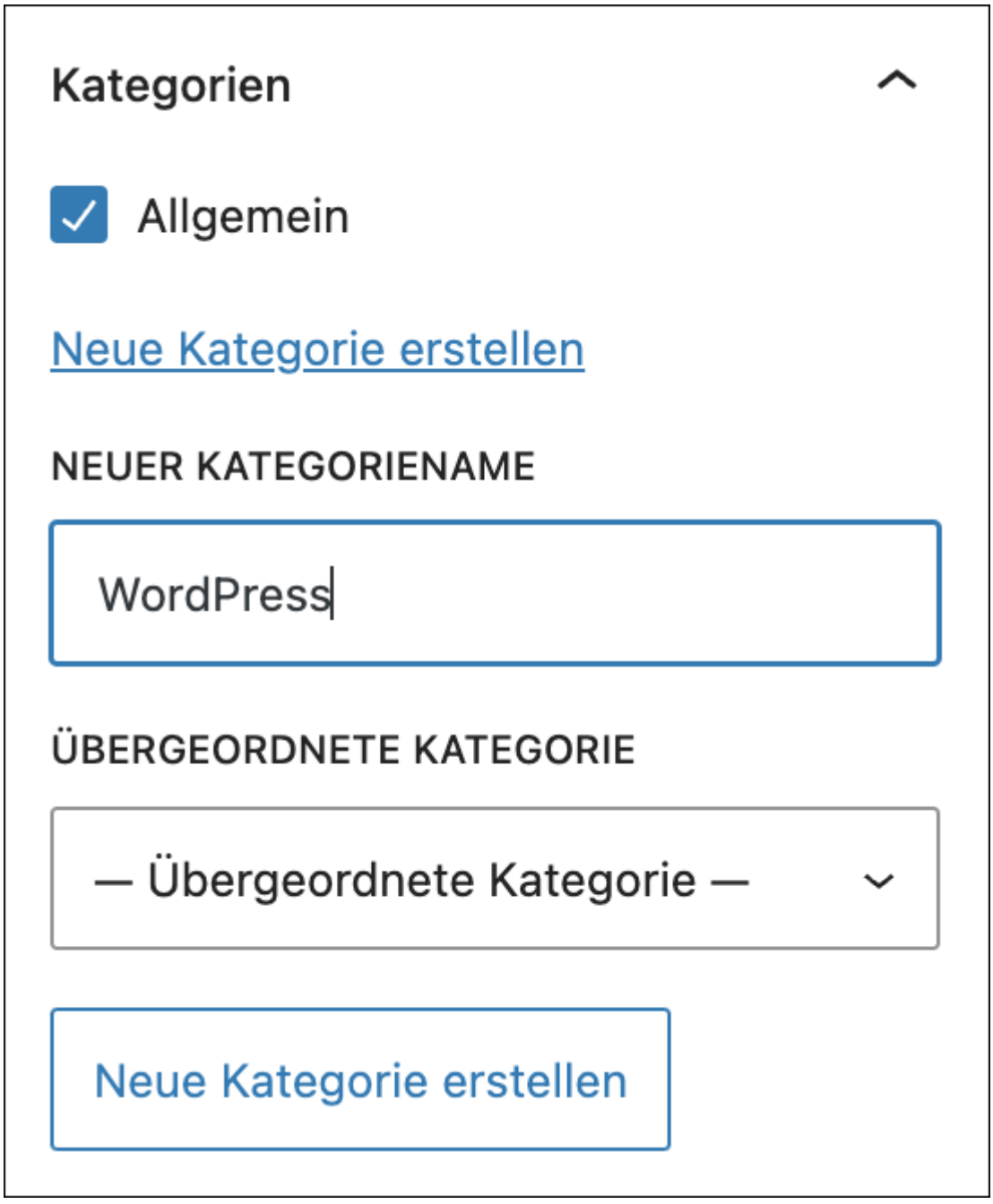


Abbildung 6.28 Kategorien können Sie nutzen, um Ihre Inhalte thematisch zu gruppieren. Sie können auch Unterkategorien anlegen.

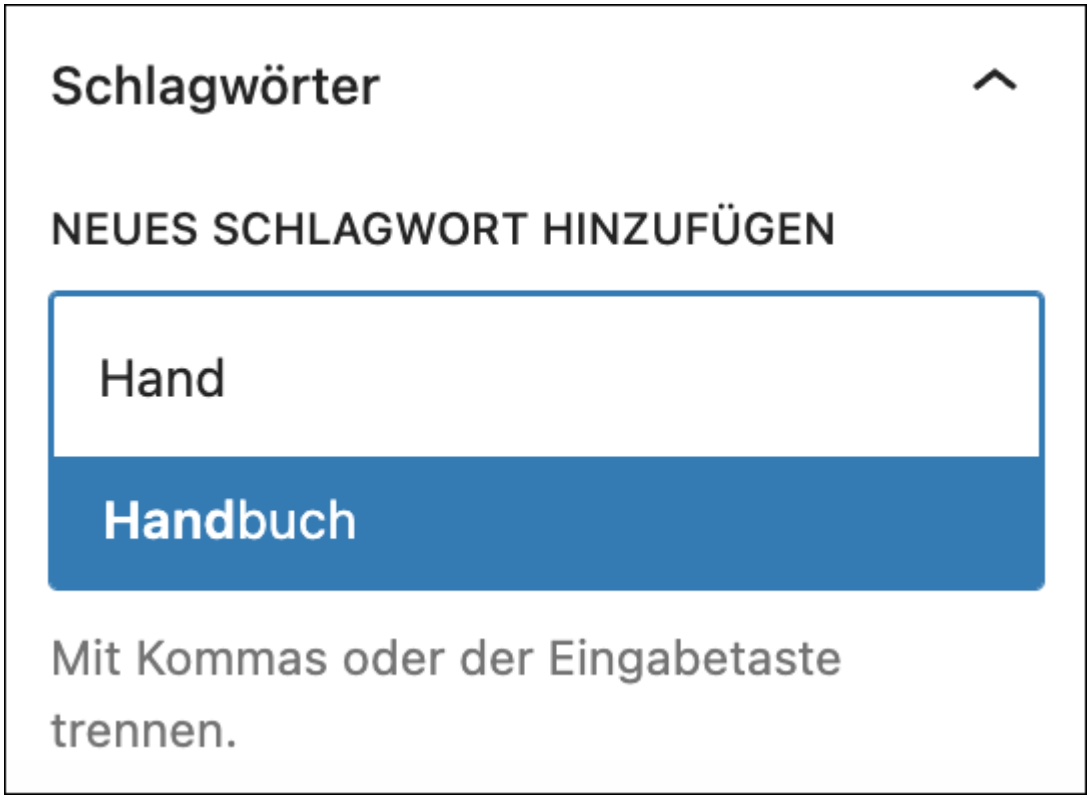


Abbildung 6.29 Schlagwörter geben Sie in ein Eingabefeld ein. Wenn bereits Schlagwörter existieren, wird Ihnen beim Tippen eine Autovervollständigung angezeigt.

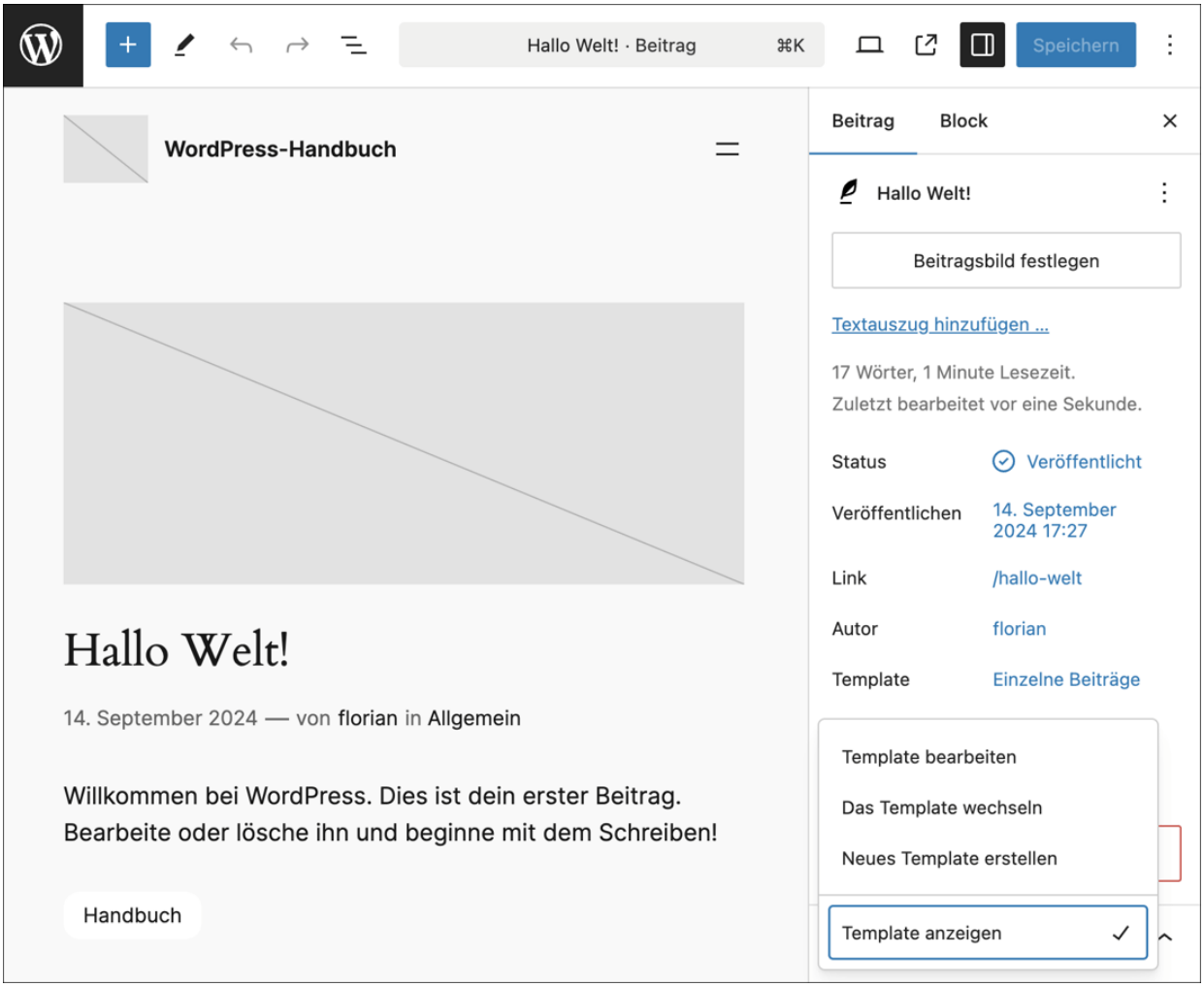


Abbildung 6.30 Wenn Sie »Template anzeigen« anklicken, wird der Inhalt so angezeigt, wie er im Template eingebunden ist. Sie können ihn in dieser Ansicht auch weiterbearbeiten.

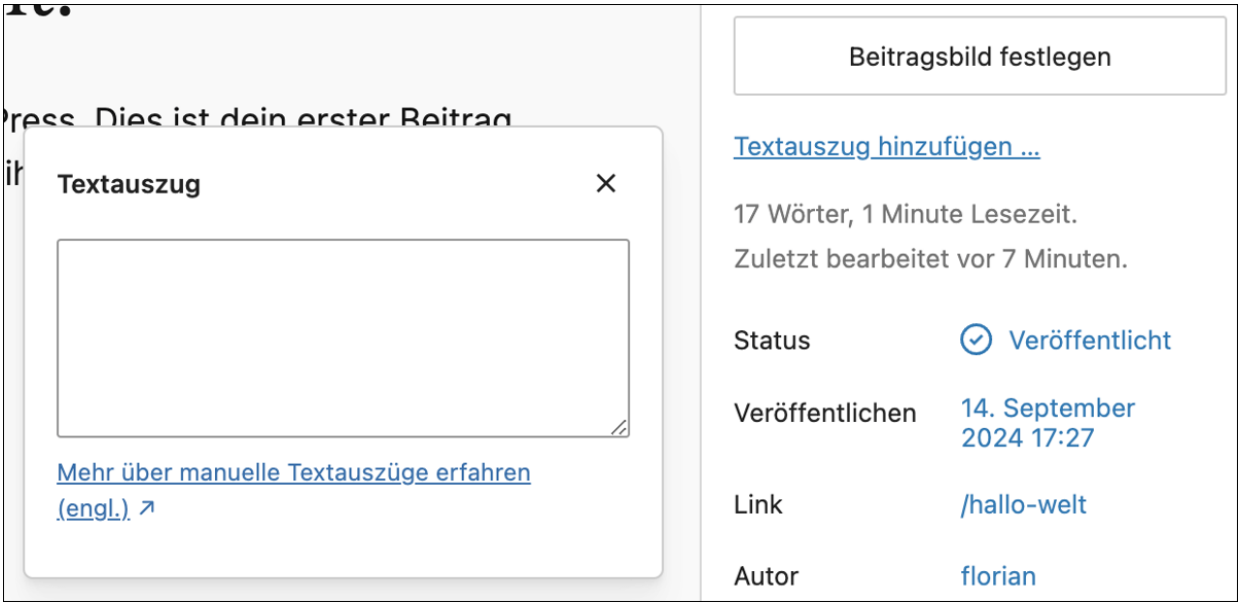


Abbildung 6.31 Der Textauszug ist meist ein kleiner Teaser-Text, der Lust auf den Beitrag machen soll.

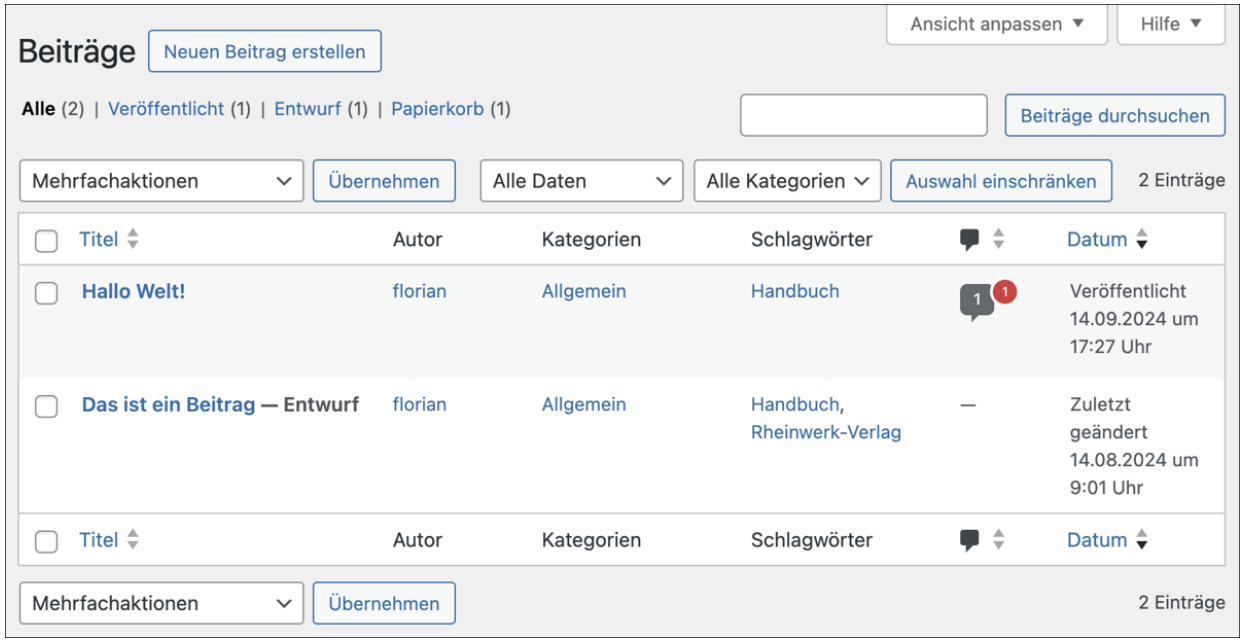


Abbildung 6.32 Beiträge und Seiten sind in WordPress sogenannte »Inhaltstypen«, die abgesehen von ein paar Ausnahmen in ähnlichen Übersichtslisten angezeigt werden.

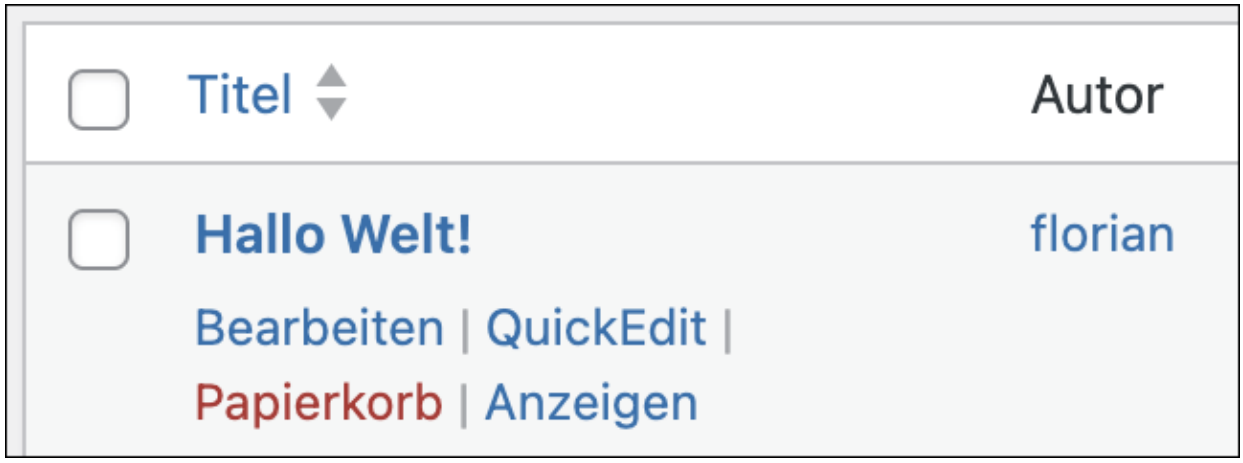


Abbildung 6.33 Wenn der Cursor über einem Eintrag in der Beitrags- oder Seitenliste platziert ist, erscheinen verschiedene Aktionslinks.

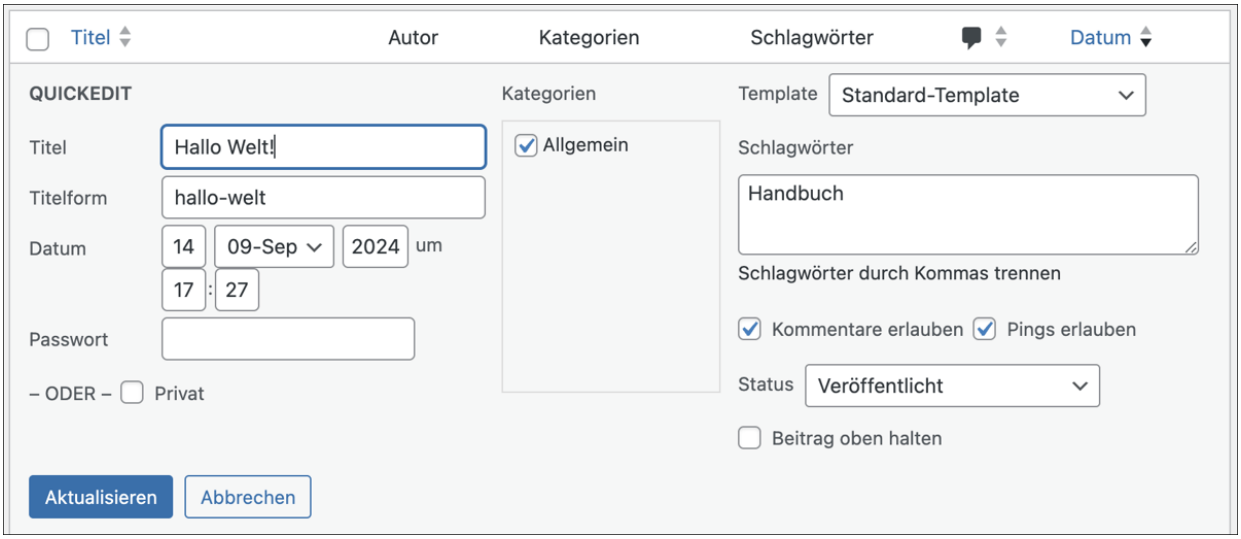


Abbildung 6.34 Über QuickEdit können Sie einige Aspekte eines Inhalts direkt in der Übersicht bearbeiten.

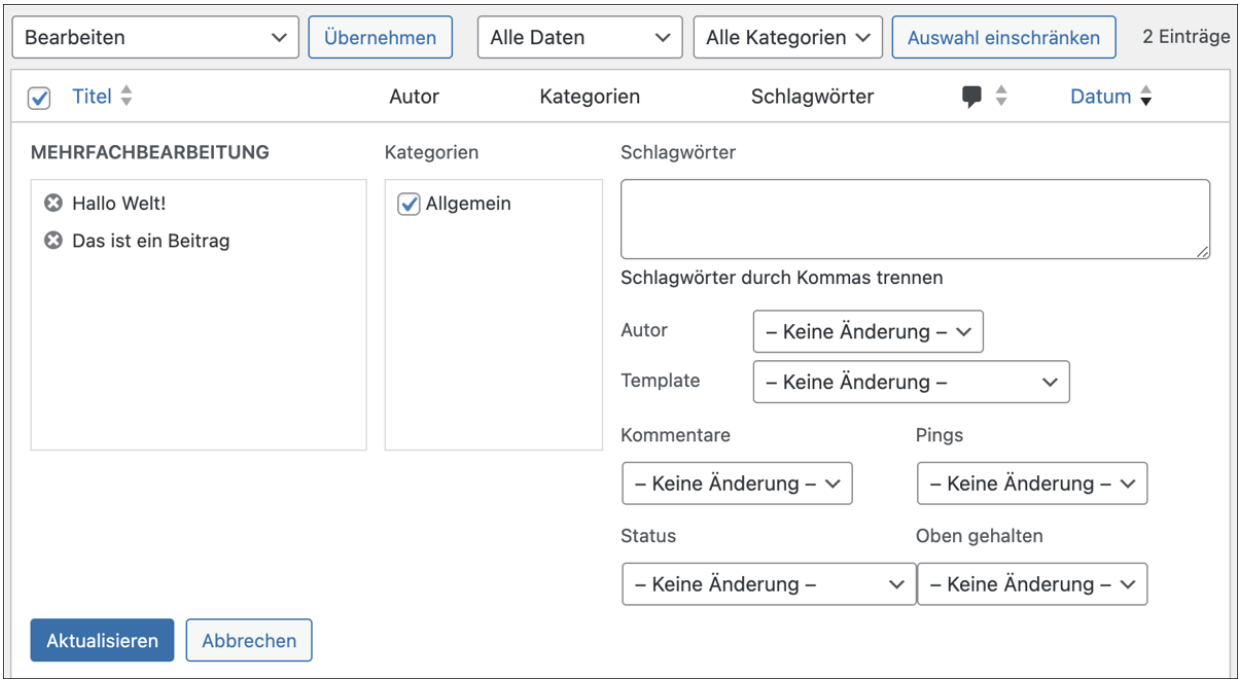


Abbildung 6.35 Sie können mit QuickEdit über die Mehrfachbearbeitung gleichzeitig mehrere Einträge bearbeiten.

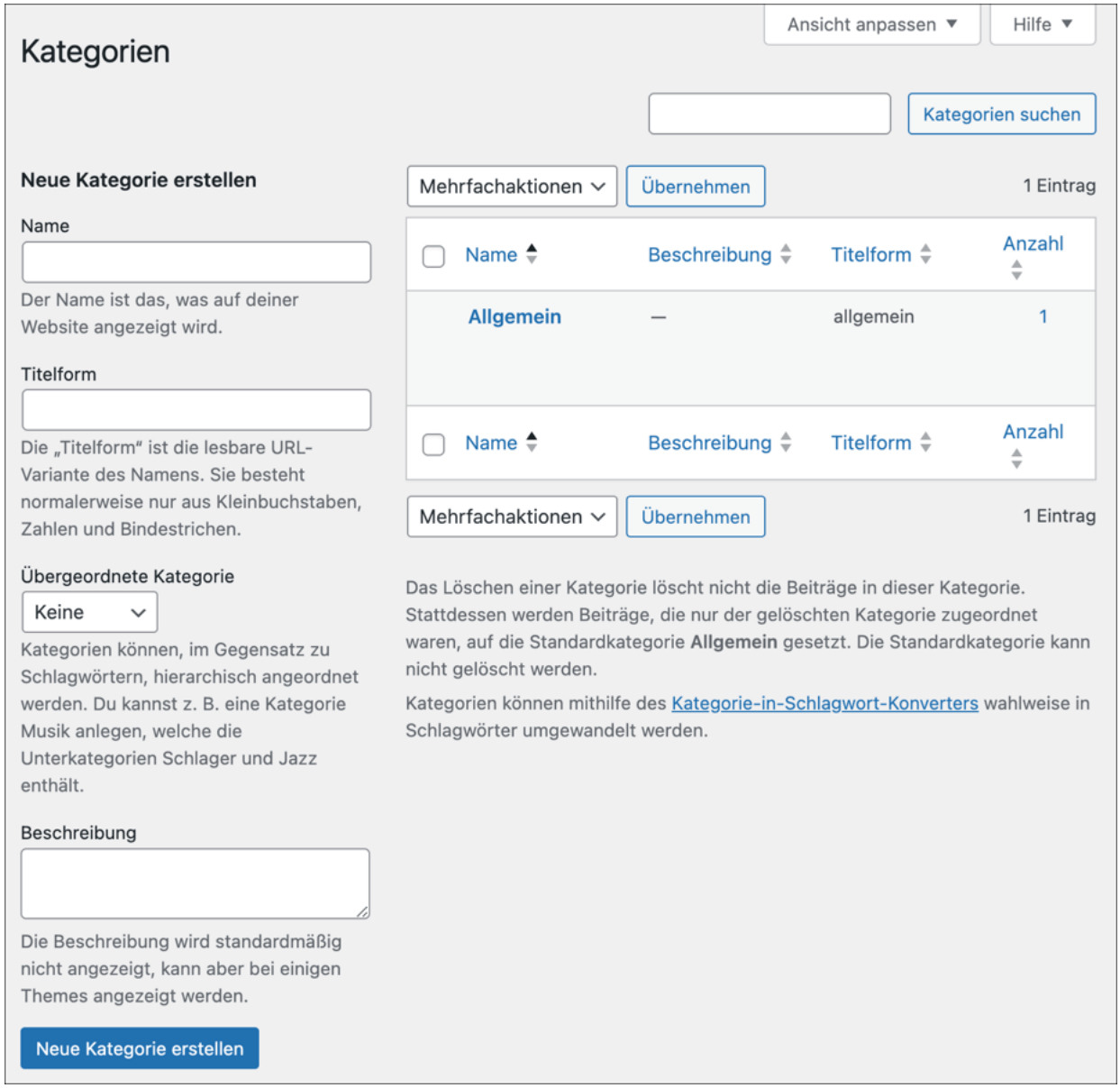


Abbildung 6.36 Unter »Beiträge« • »Kategorien« finden Sie eine Übersicht der Kategorien und können bestehende bearbeiten und neue Kategorien erstellen.

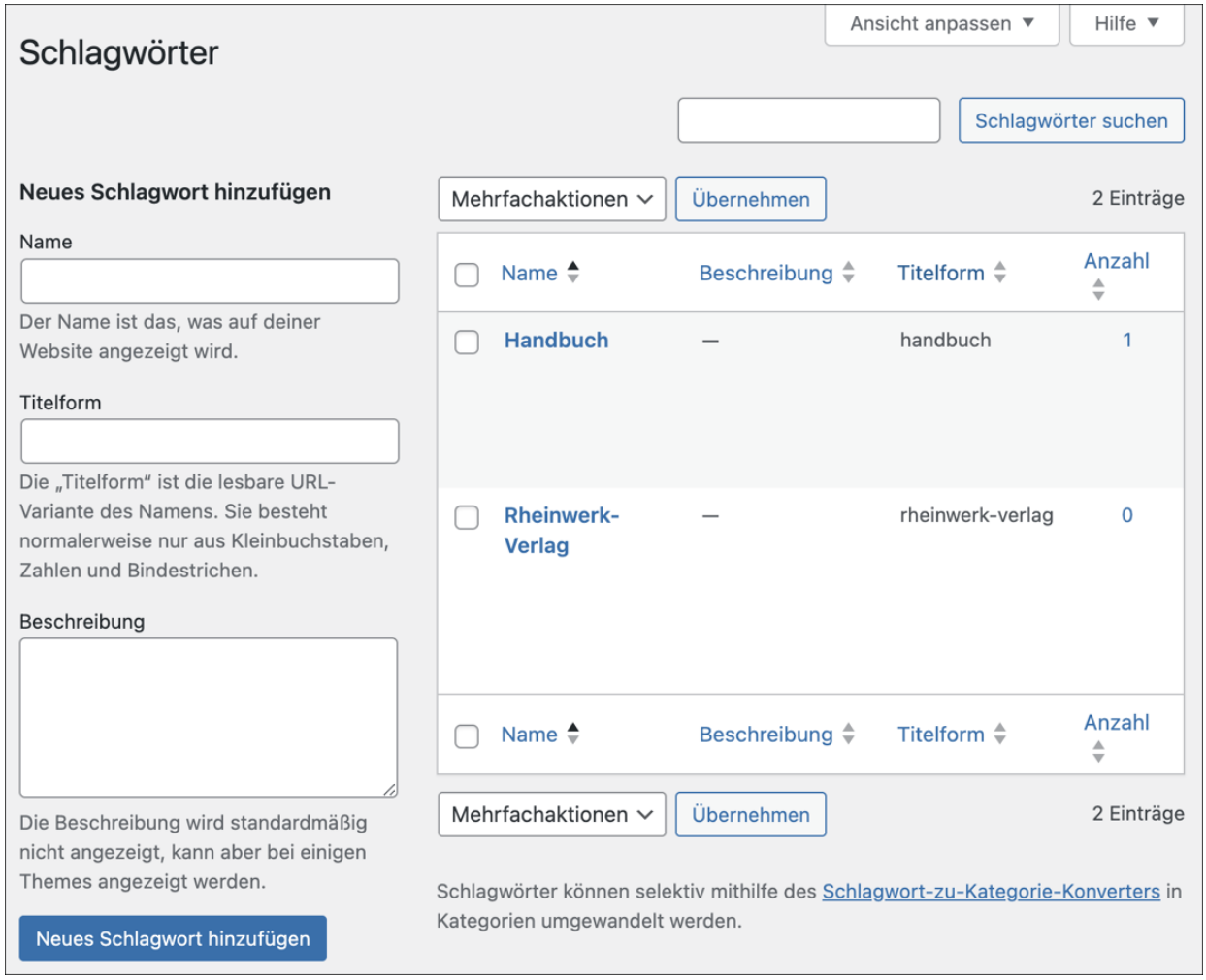


Abbildung 6.37 Die Schlagwörter lassen sich unter »Beiträge« • »Schlagwörter« anschauen und bearbeiten.

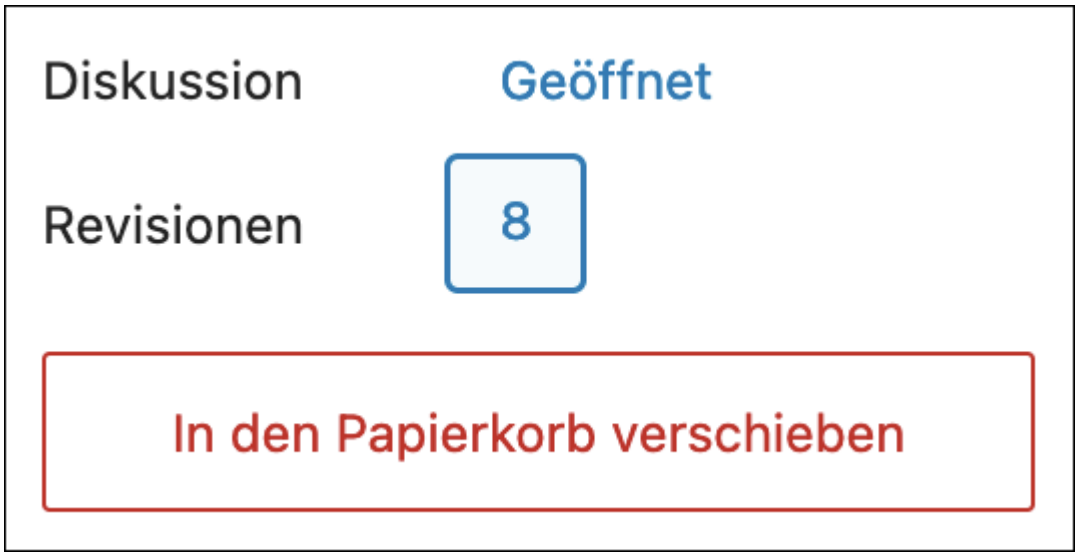


Abbildung 6.38 In der Dokumenten-Seitenleiste des Block-Editors finden Sie den Punkt »Revisionen«, über den Sie in die Revisionen-Ansicht gelangen.



Abbildung 6.39 In der Revisionsansicht können Sie unterschiedliche Stände eines Inhalts vergleichen und einen älteren Stand wiederherstellen.

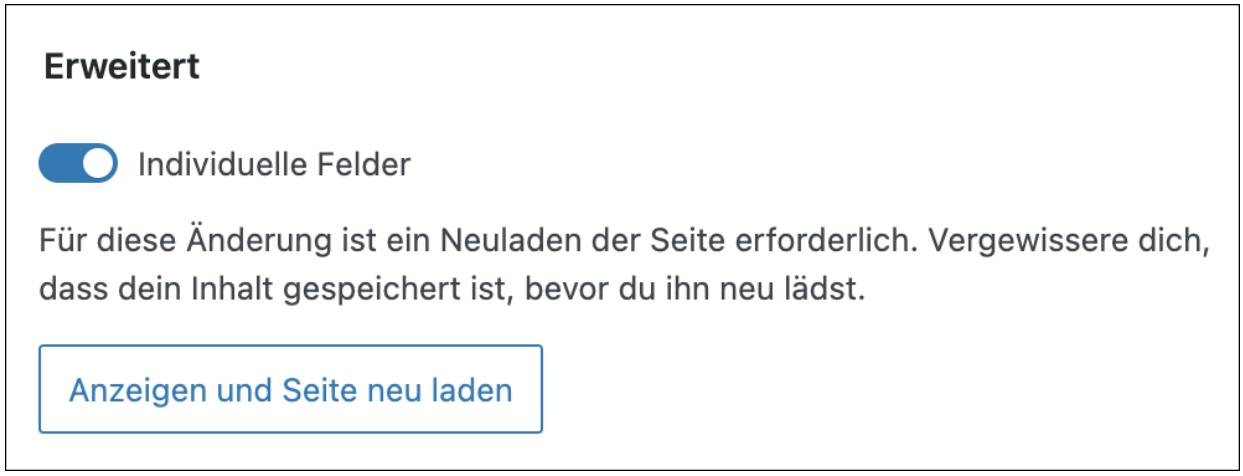


Abbildung 6.40 Die Funktion »Individuelle Felder« muss in den Block-Editor-Einstellungen aktiviert werden.

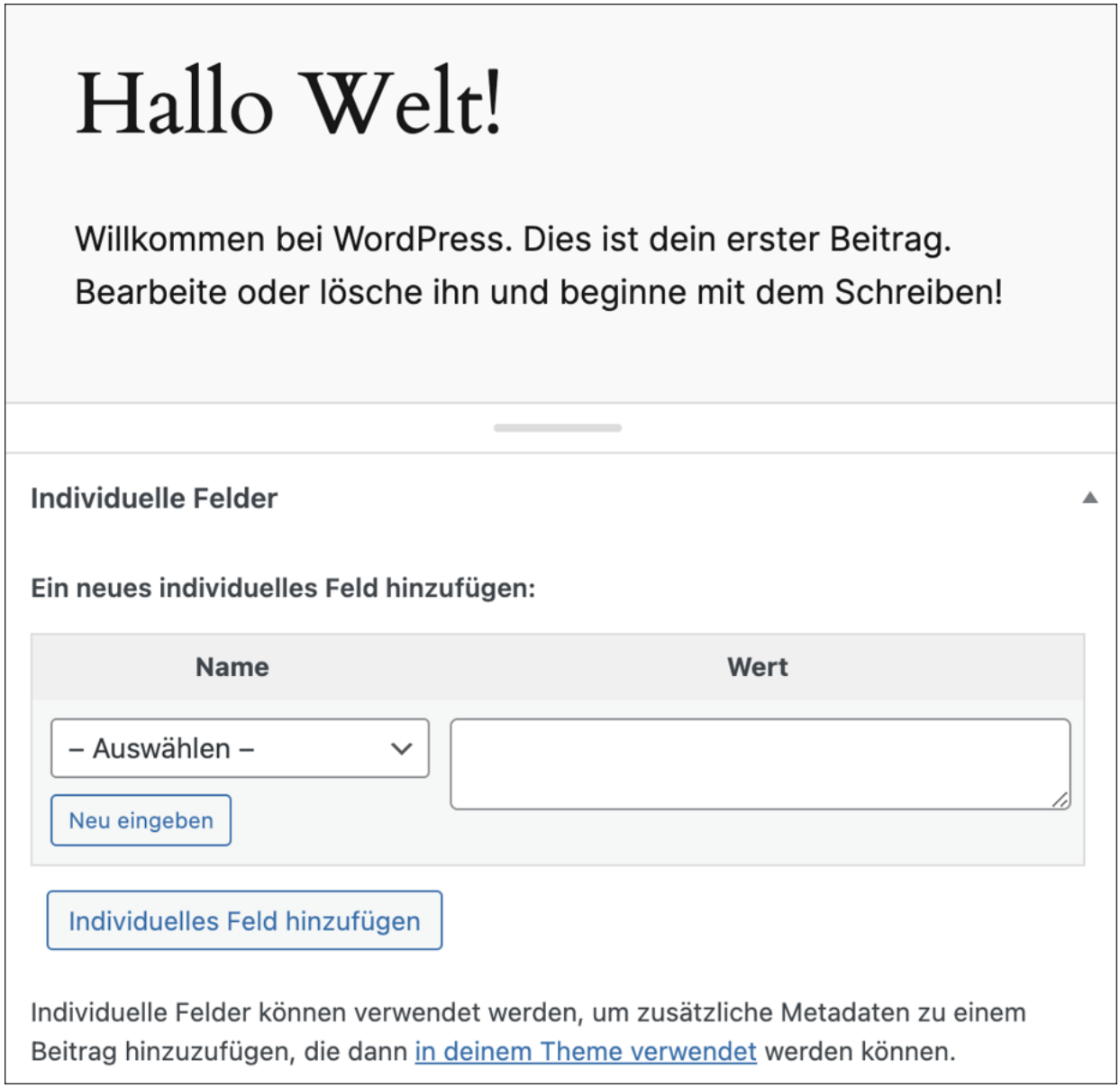


Abbildung 6.41 Nach der Aktivierung finden Sie das Panel für die individuellen Felder im Editor unter dem Bereich zur Inhaltsbearbeitung.



Abbildung 6.42 Wenn Sie einen Block auswählen, der die Block Bindings API verwendet, wird durch Nutzung eines lila Farbtons angezeigt, dass der Block die API nutzt.


ATTRIBUTE
content
Alle zurücksetzen
Attribute
Erweitert

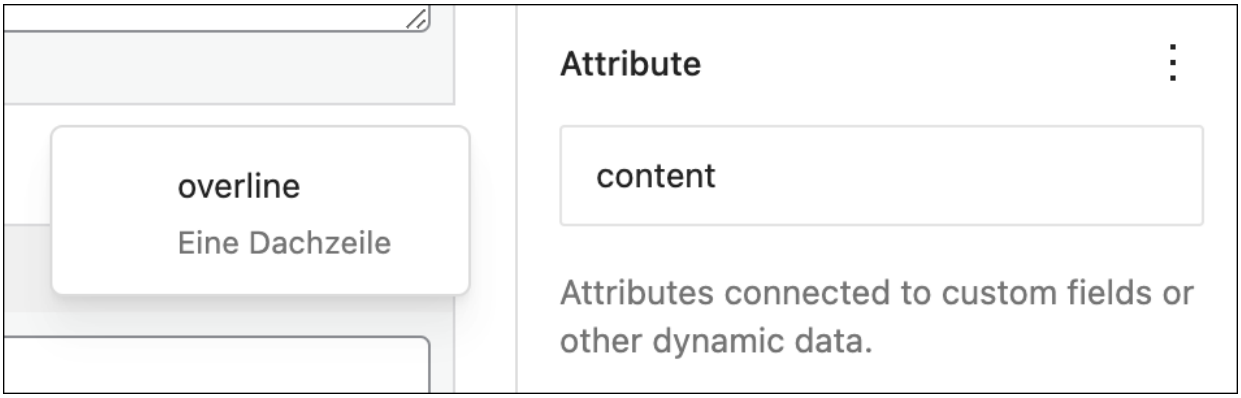


Abbildung 6.43 Das Festlegen der Verknüpfung zwischen einem Block und einem Postmetawert über das User-Interface funktioniert über das »Attribute«-Panel und ist ein zweischrittiger Prozess.

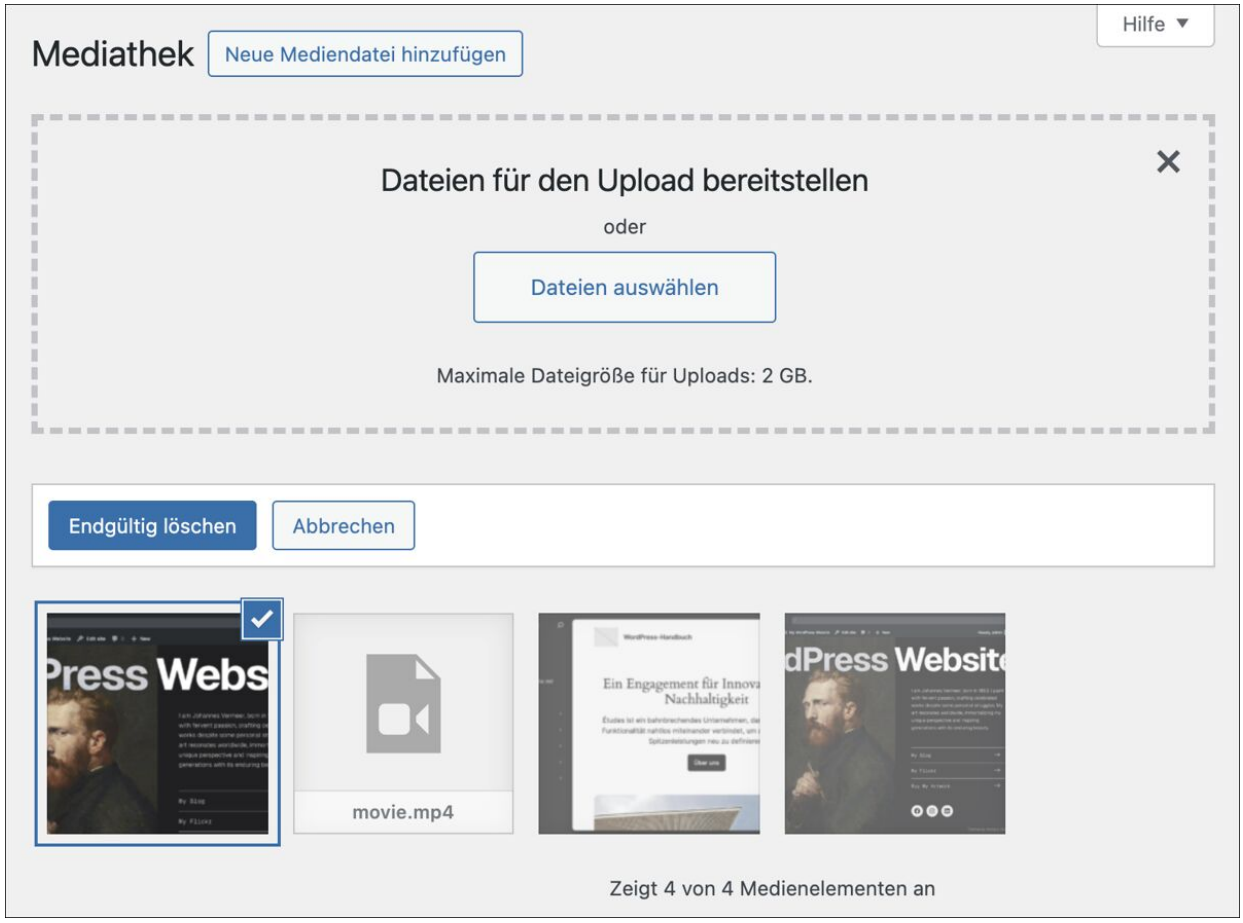


Abbildung 6.44 Die Mediathek zeigt Ihnen alle zu Ihrer Website hochgeladenen Dateien, egal ob der Upload direkt in der Mediathek erfolgt oder beispielsweise über den Bild-Block.



Abbildung 6.45 In der Listenansicht sehen Sie direkt in der Übersicht auch nähere Details zu den Dateien. Dafür sehen Sie auf einen Blick weniger Dateien.

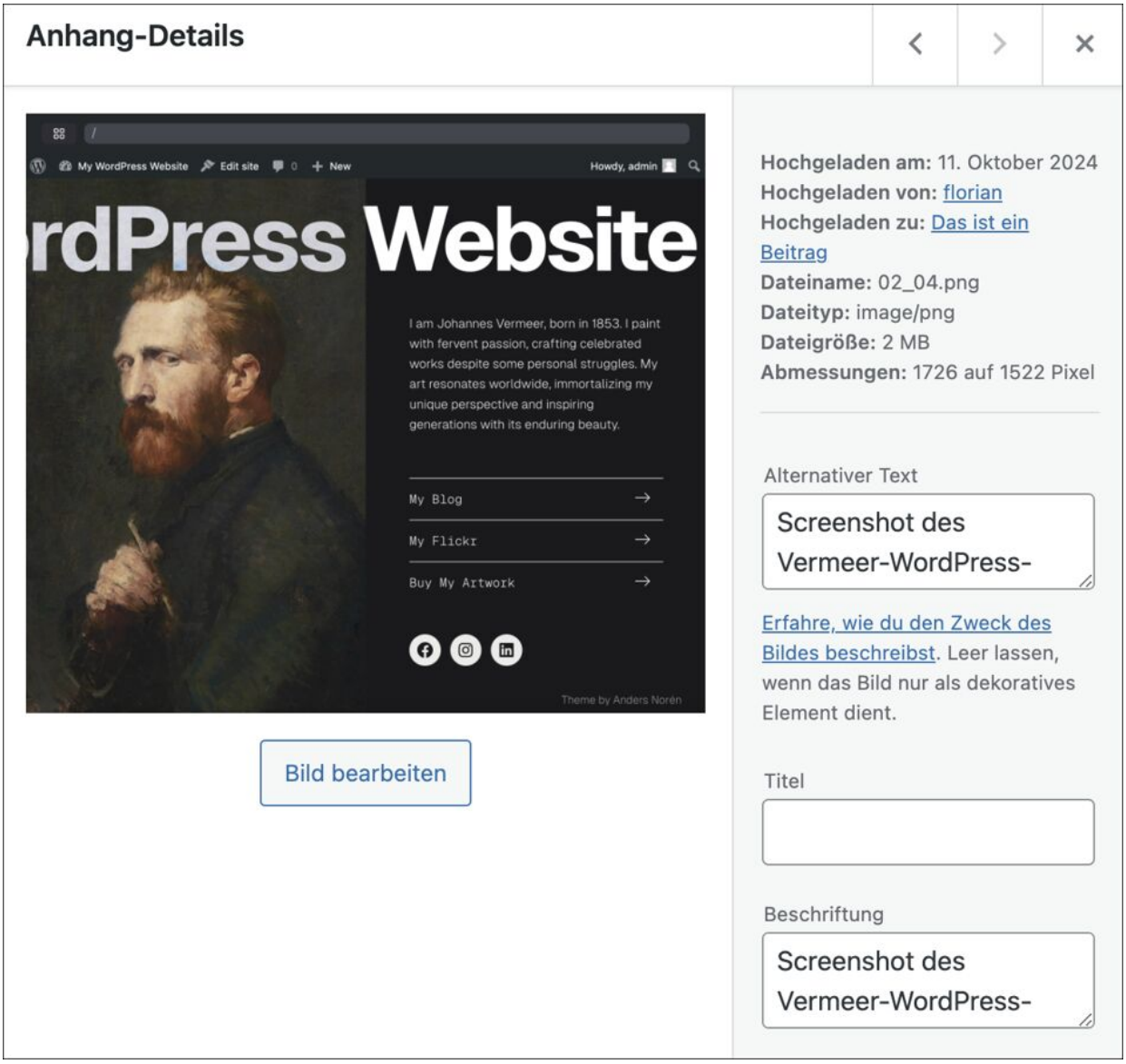
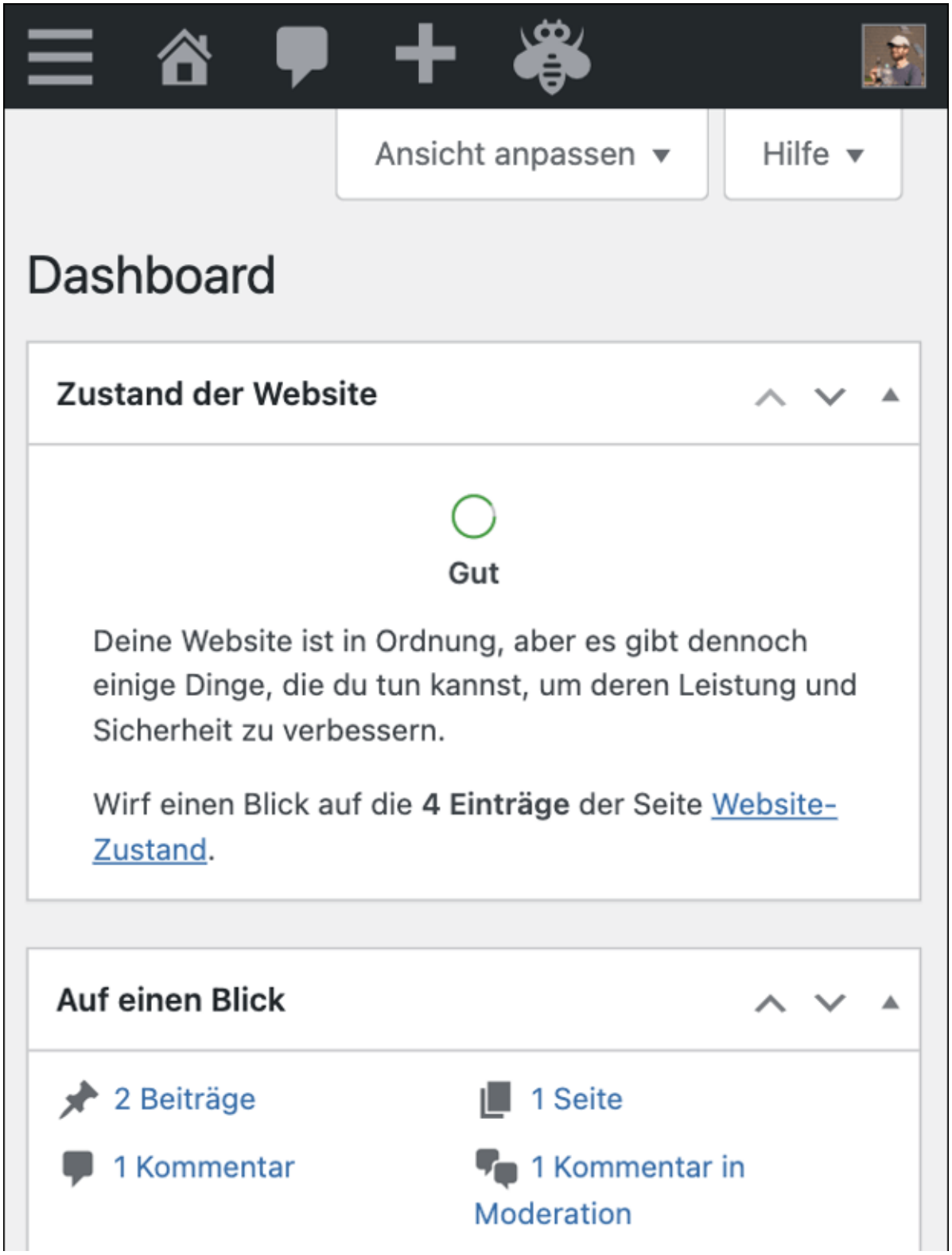


Abbildung 6.46 Die Pflege der Beschriftung ist für die Findbarkeit über die Suchfunktion nützlich.



Abbildung 6.47 Mit der Suche können Sie gezielt nach Dateien suchen. Wie gut das funktioniert, ist allerdings abhängig davon, wie gut die Dateien benannt beziehungsweise die Felder wie die Beschriftung gepflegt sind.


Ansicht anpassen
Hilfe
Dashboard
Zustand der Website
Gut
Deine Website ist in Ordnung, aber es gibt dennoch einige Dinge, die du tun kannst, um deren Leistung und Sicherheit zu verbessern.
Wirf einen Blick auf die 4 Einträge der Seite Website-Zustand.
Auf einen Blick
2 Beiträge
1 Seite
1 Kommentar
1 Kommentar in Moderation

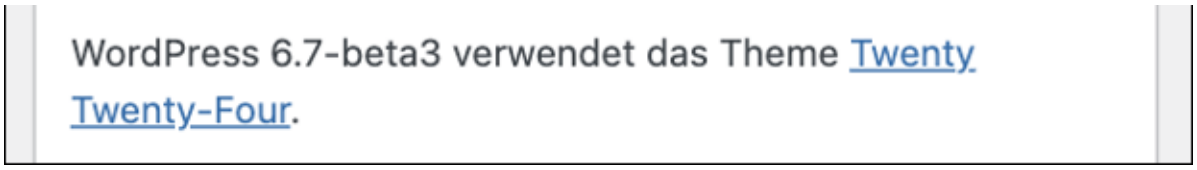


Abbildung 6.48 Da das WordPress-Backend responsive ist, lässt es sich generell auch mit Smartphones bedienen.

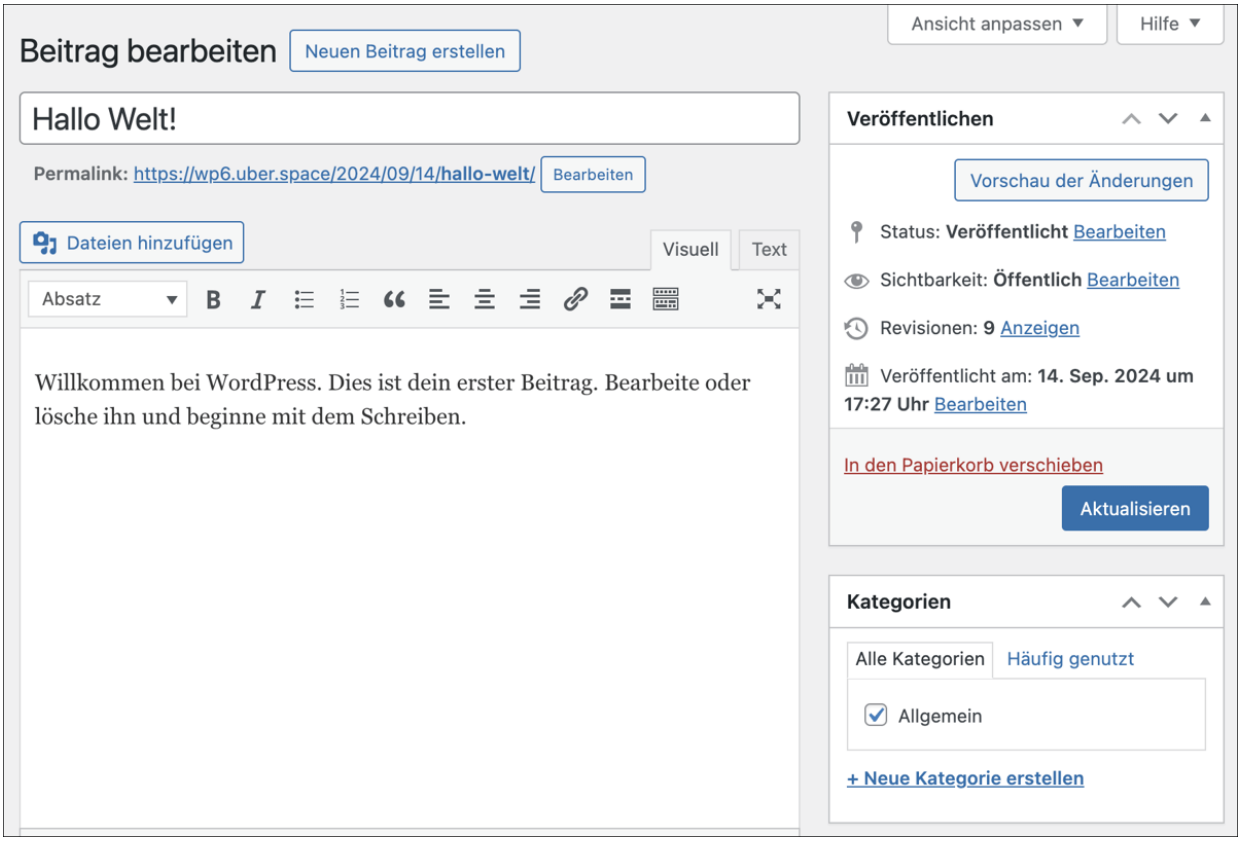


Abbildung 6.49 Mit dem »Classic Editor«-Plugin wird der Block-Editor gegen den TinyMCE-Editor ausgetauscht, der bis WordPress 5 standardmäßig genutzt wurde.

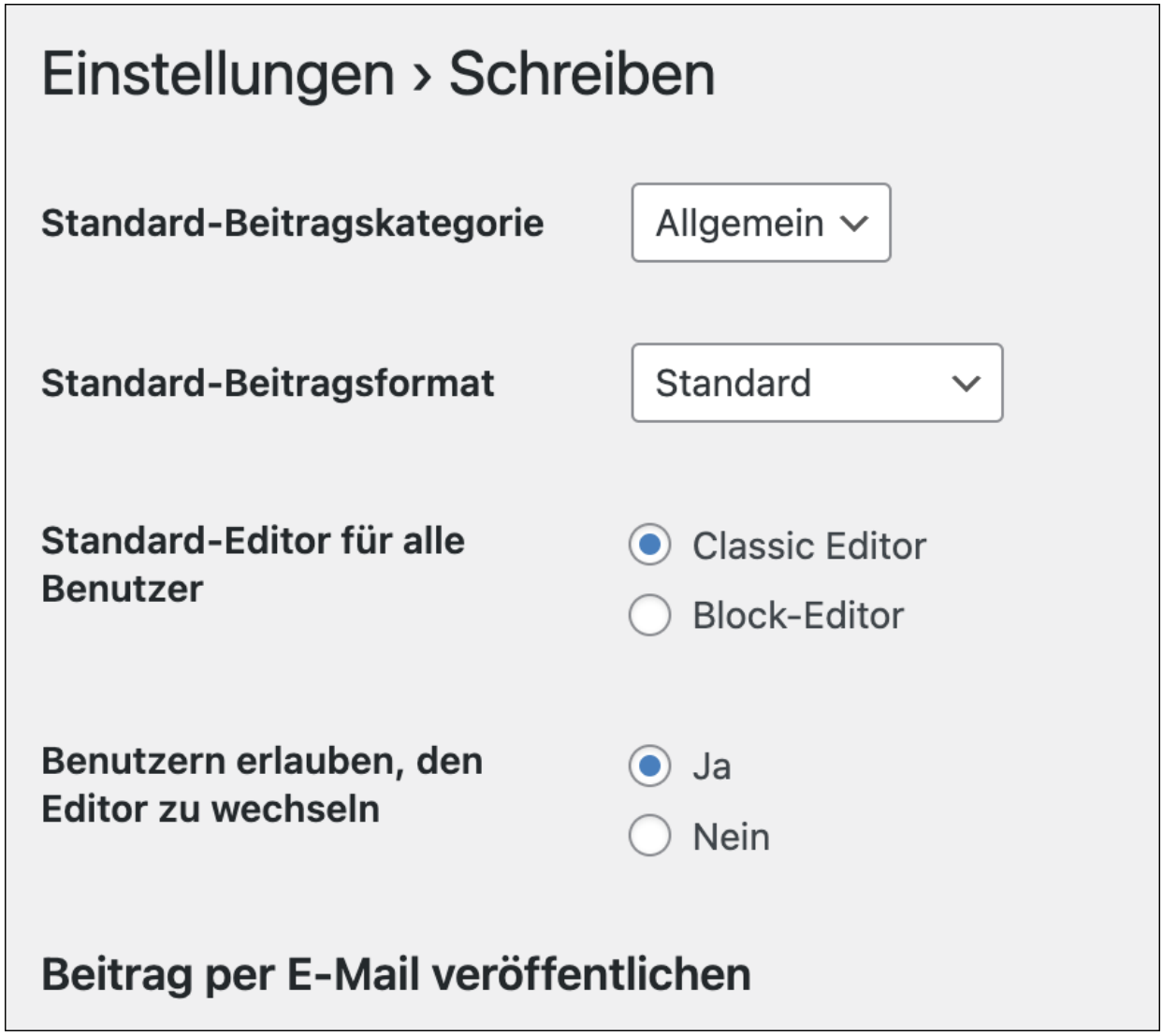


Abbildung 6.50 In den Einstellungen können Sie unter anderem anpassen, welcher Editor standardmäßig geöffnet wird.

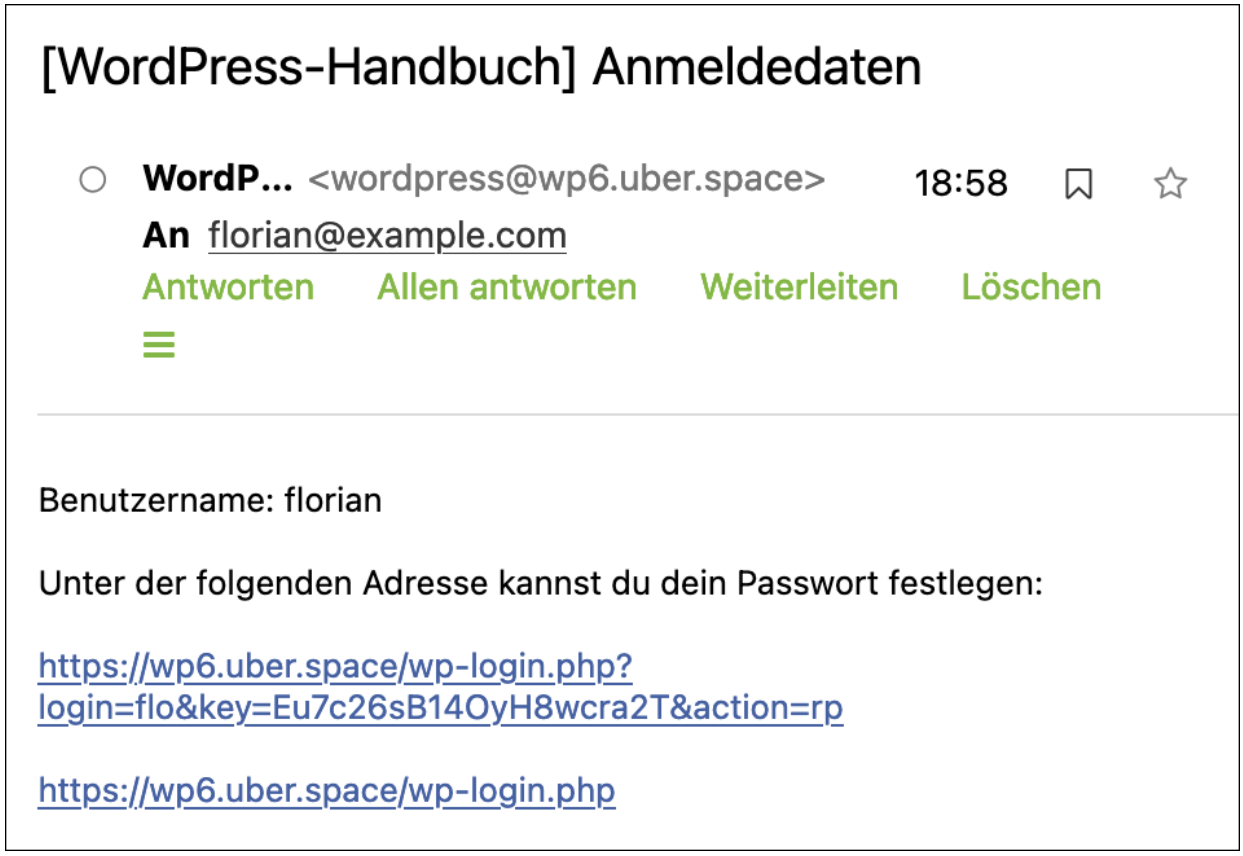


Abbildung 7.1 Die Bestätigung einer Anmeldung mit einem Link aus einer E-Mail wird »Double-Opt-in« genannt.

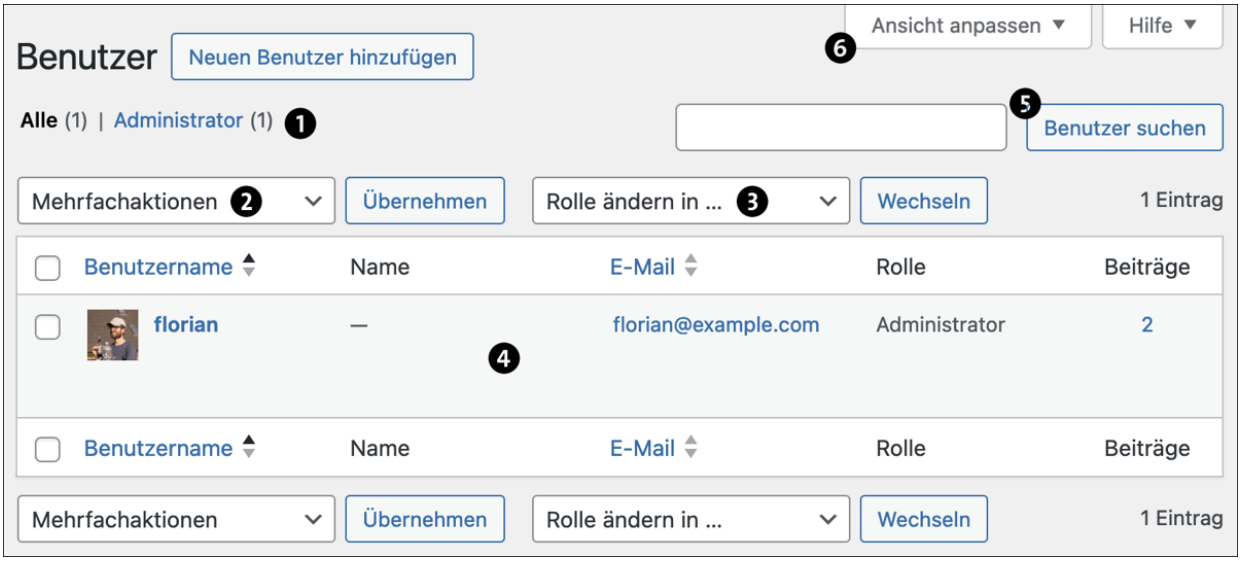


Abbildung 7.2 Von der Liste aller Benutzer*innen gelangen Sie in die Detailansicht zur Bearbeitung einzelner Accounts.


Hilfe
Profil
Persönliche Optionen
Visueller Editor 1
Beim Schreiben den visuellen Editor nicht benutzen
Syntaxhervorhebung 2
Die Syntaxhervorhebung beim Bearbeiten von Code deaktivieren
Farbschema verwalten 3
Standard
Hell
Modern
Blau
Kaffee
Ektoplasma
Mitternacht
Meer
Sonnenaufgang
Tastaturkürzel 4
Tastaturkürzel für die Kommentarmoderation aktivieren. Dokumentation über Tastaturkurzbefehle (engl.)
Werkzeugleiste 5
Werkzeugleiste für mich auf der Website anzeigen
Sprache 6
Website-Einstellung
Name 7
Benutzername
florian
Benutzernamen können nicht geändert werden.
Vorname
Nachname
Spitzname (erforderlich)
florian
Öffentlicher Name
florian
Kontaktinfo 8
E-Mail (erforderlich)
florian@example.com
Wenn du das änderst, senden wir dir eine E-Mail an deine neue Adresse, um die Änderung zu bestätigen. Die neue Adresse wird erst nach Bestätigung aktiv.

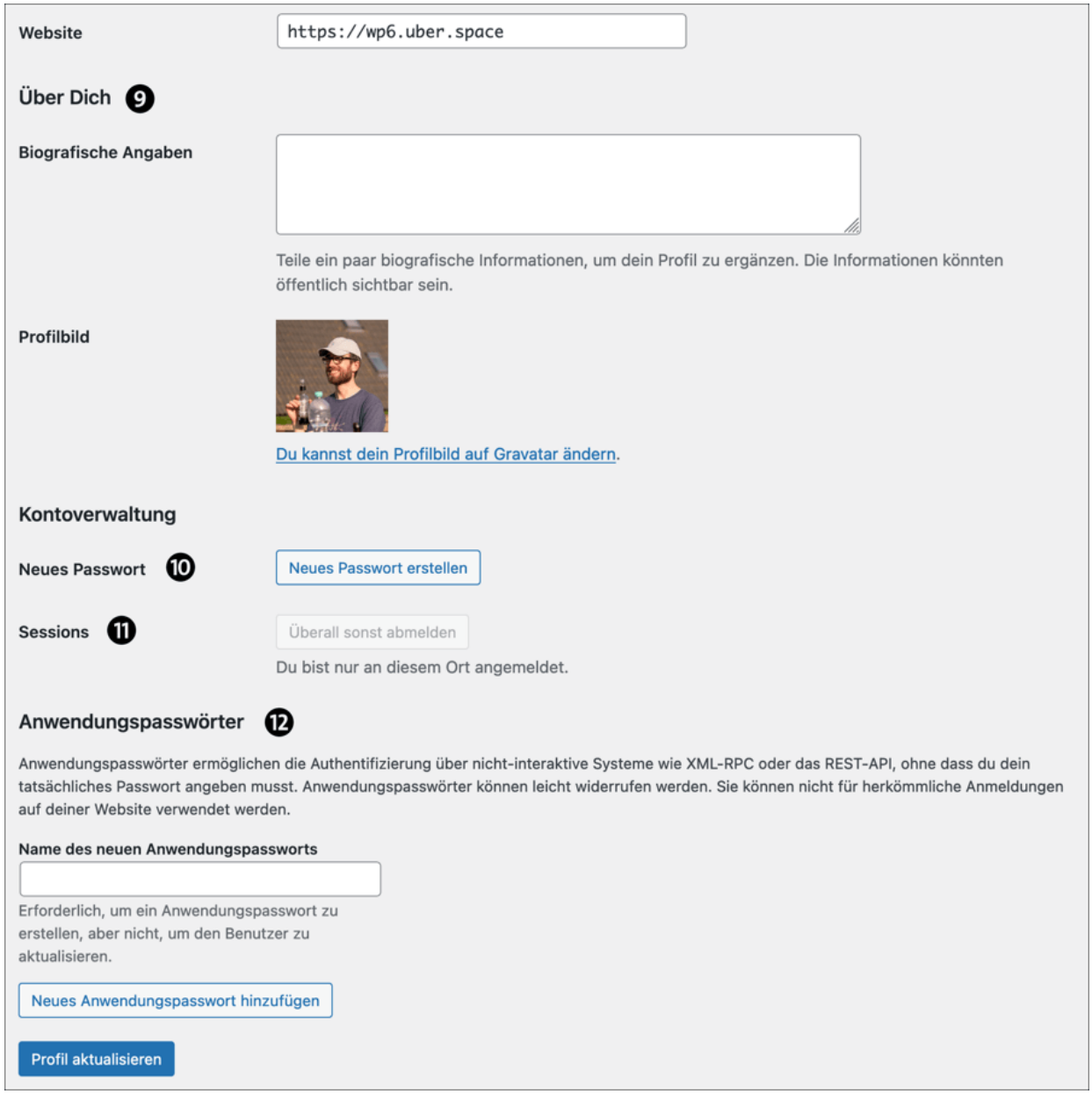


Abbildung 7.3 Die Bearbeitungsmöglichkeiten für ein Profil

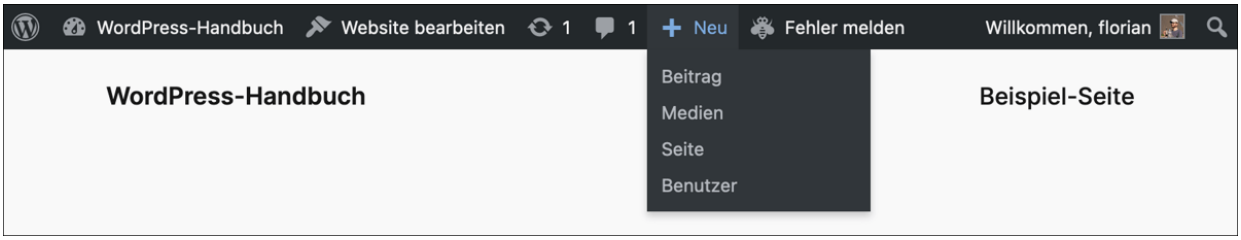


Abbildung 7.4 Die Adminbar bietet im Frontend ein paar nützliche Abkürzungen ins Backend, beispielsweise in die Bearbeitungsansicht des aktuellen Inhalts.

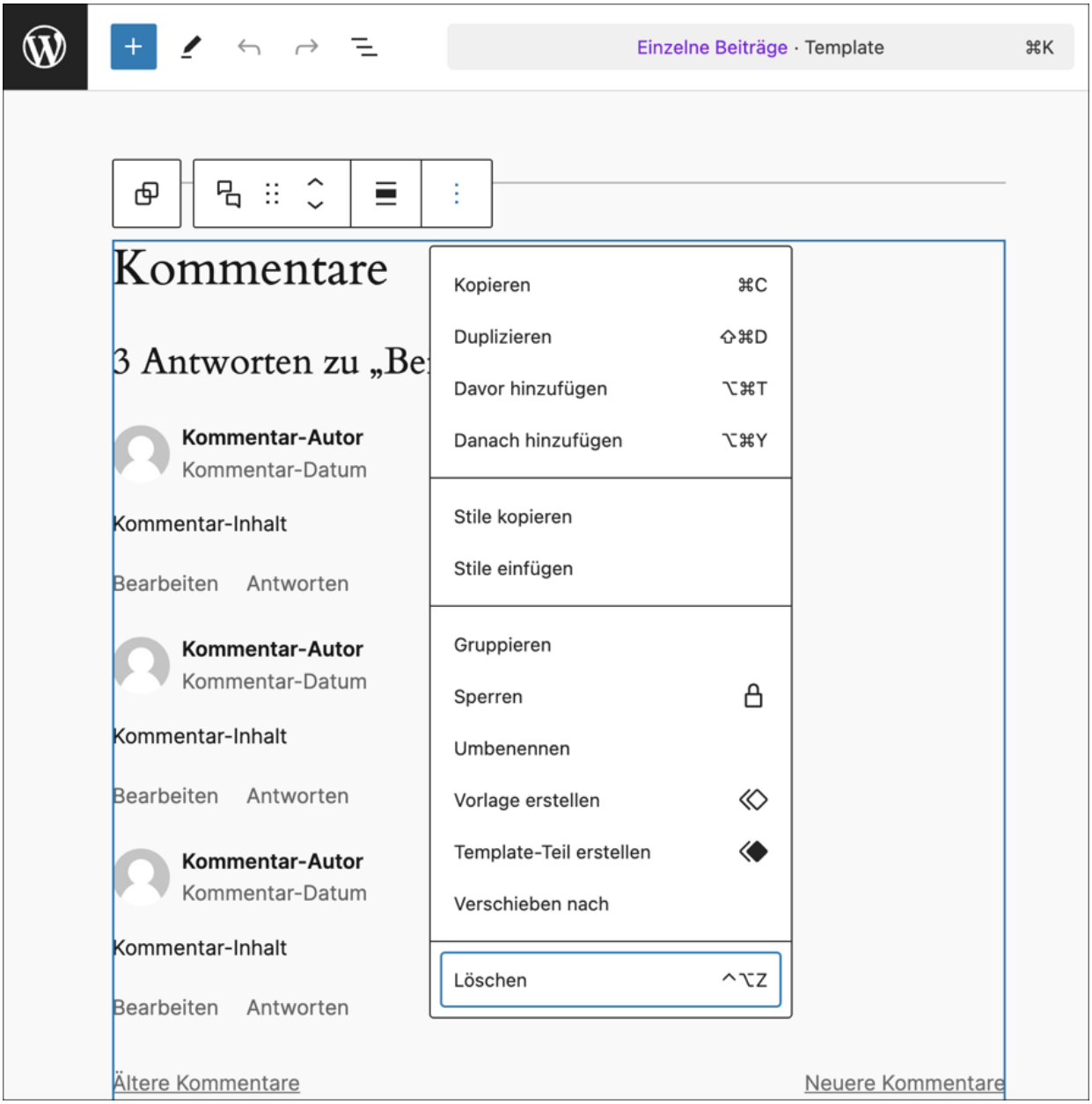


Abbildung 8.1 In einem Block-Theme können Sie den »Kommentare«-Block über den Website-Editor entfernen.

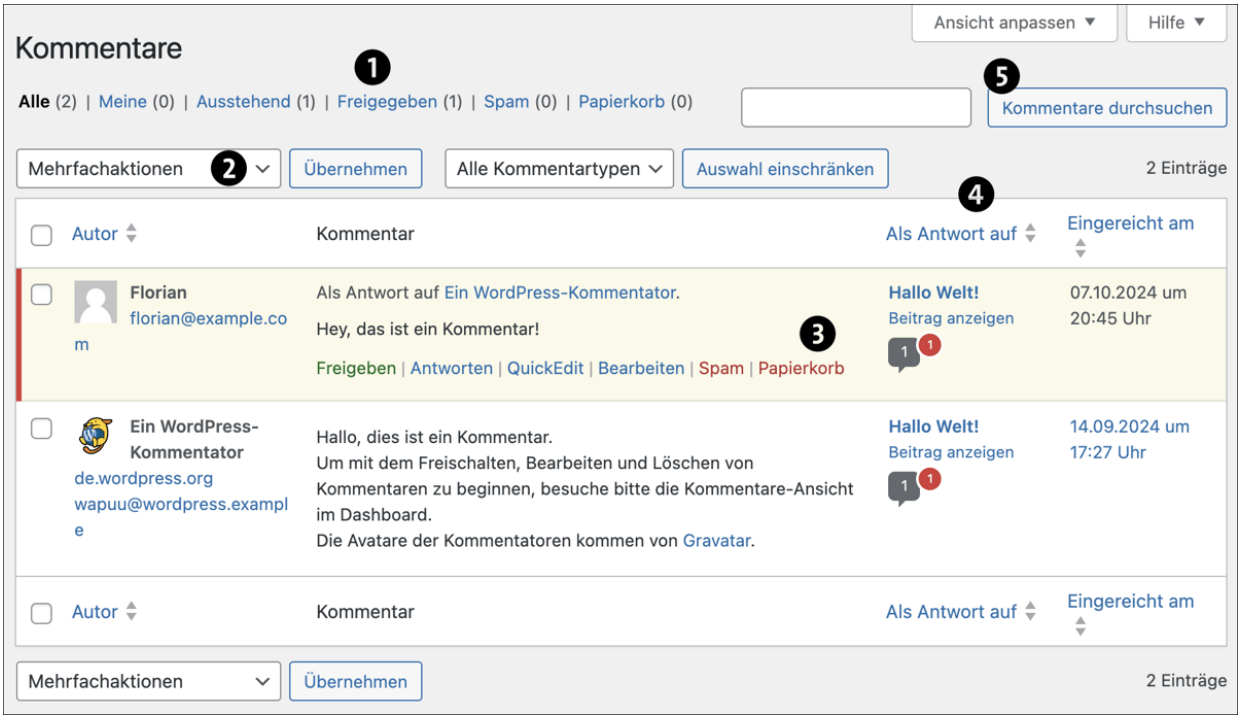


Abbildung 8.2 Die Liste der Kommentare ähnelt der für Seiten und Beiträge. Wenn ein Kommentar auf Freigabe wartet, wird er farblich hervorgehoben. Wenn Sie den Cursor über den Eintrag bewegen, erscheinen verschiedene Aktionslinks.

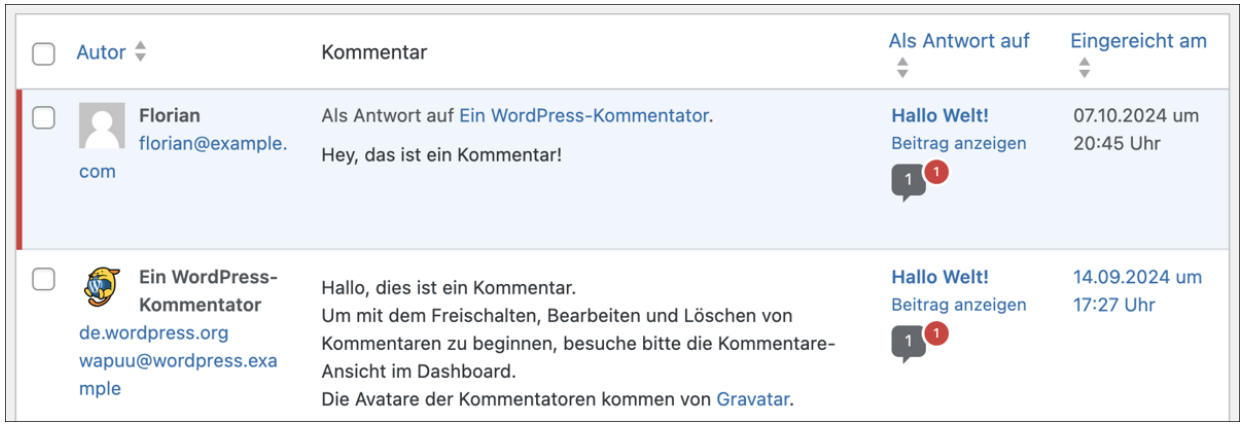


Abbildung 8.3 Der aktuell ausgewählte Kommentar wird mit einem blauen Hintergrund hervorgehoben.

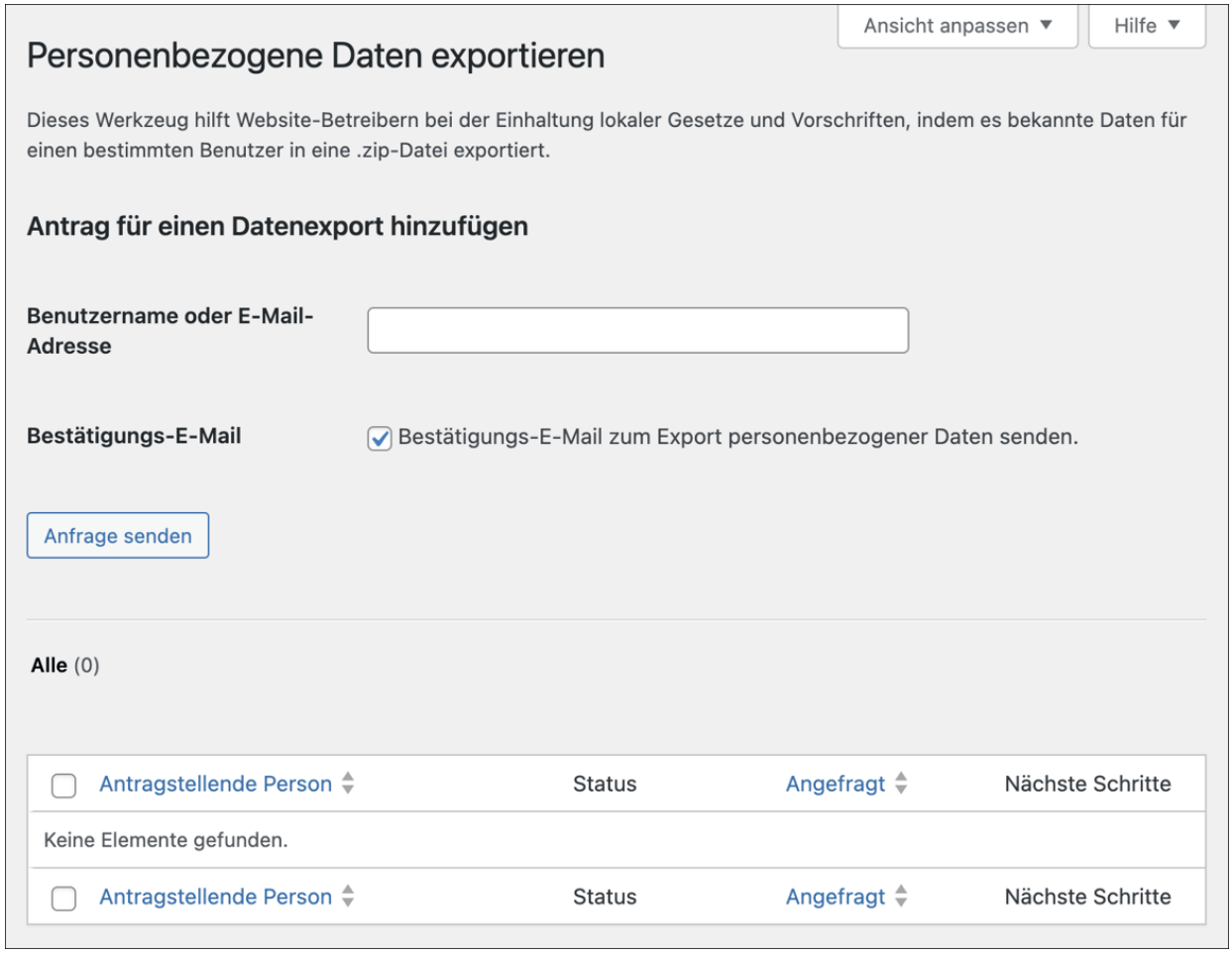


Abbildung 8.4 WordPress kommt mit einer Seite, auf der Sie einen Export von Daten einer Person durchführen können.

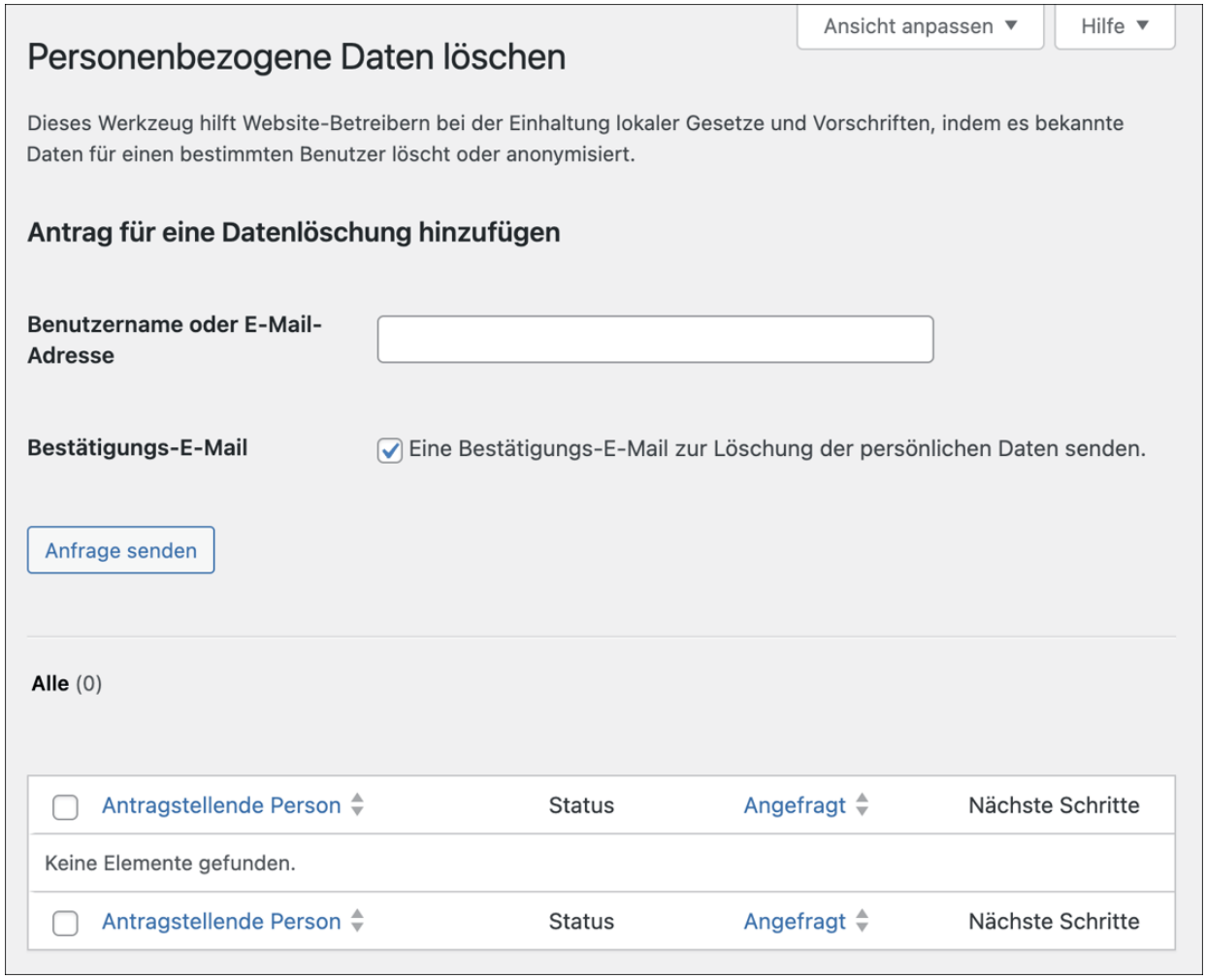


Abbildung 8.5 Auch für das Löschen beziehungsweise Anonymisieren von personenbezogenen Daten bietet WordPress ein Werkzeug.

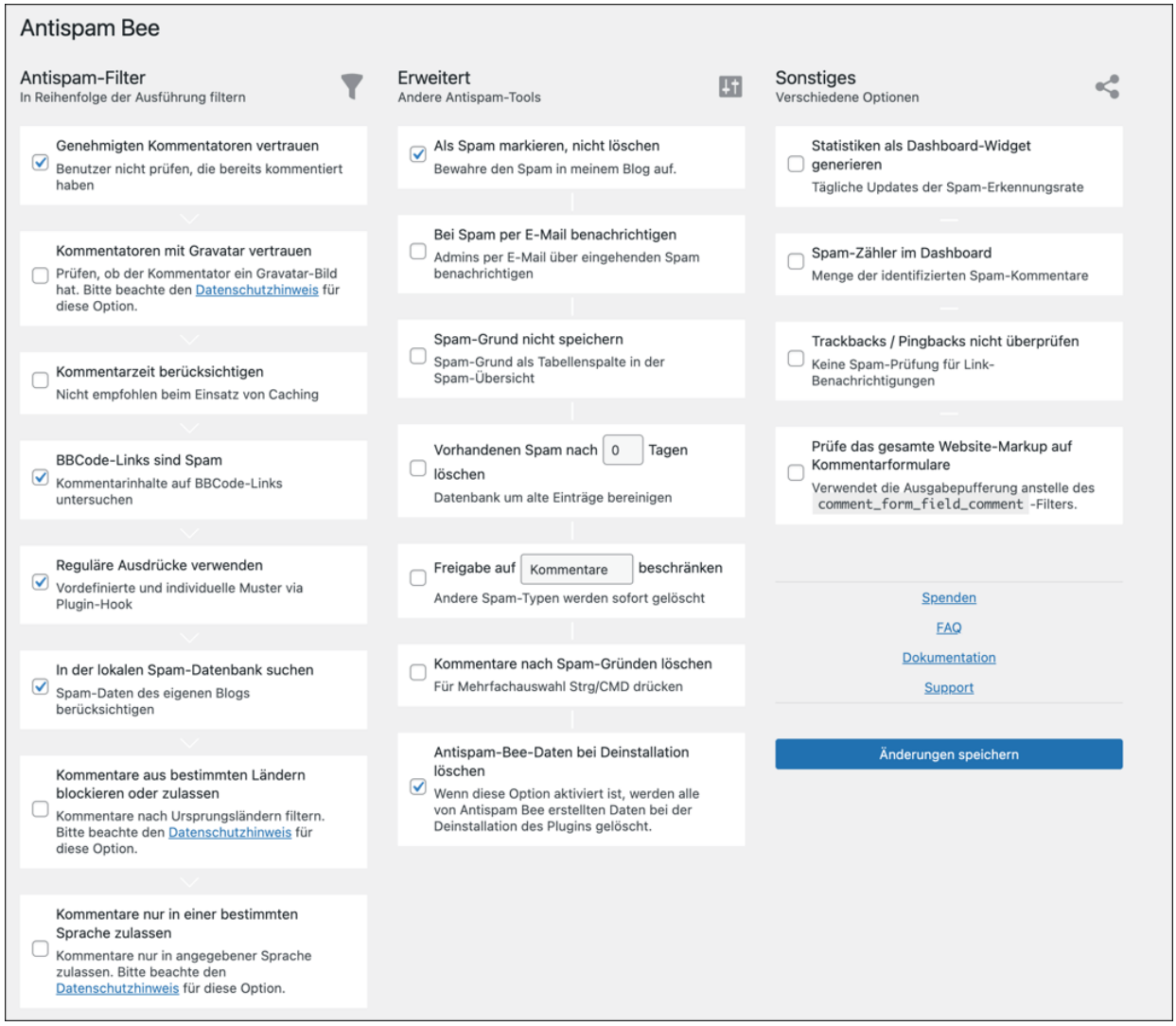


Abbildung 8.6 Die Einstellungen des Antispam-Plugins »Antispam Bee«

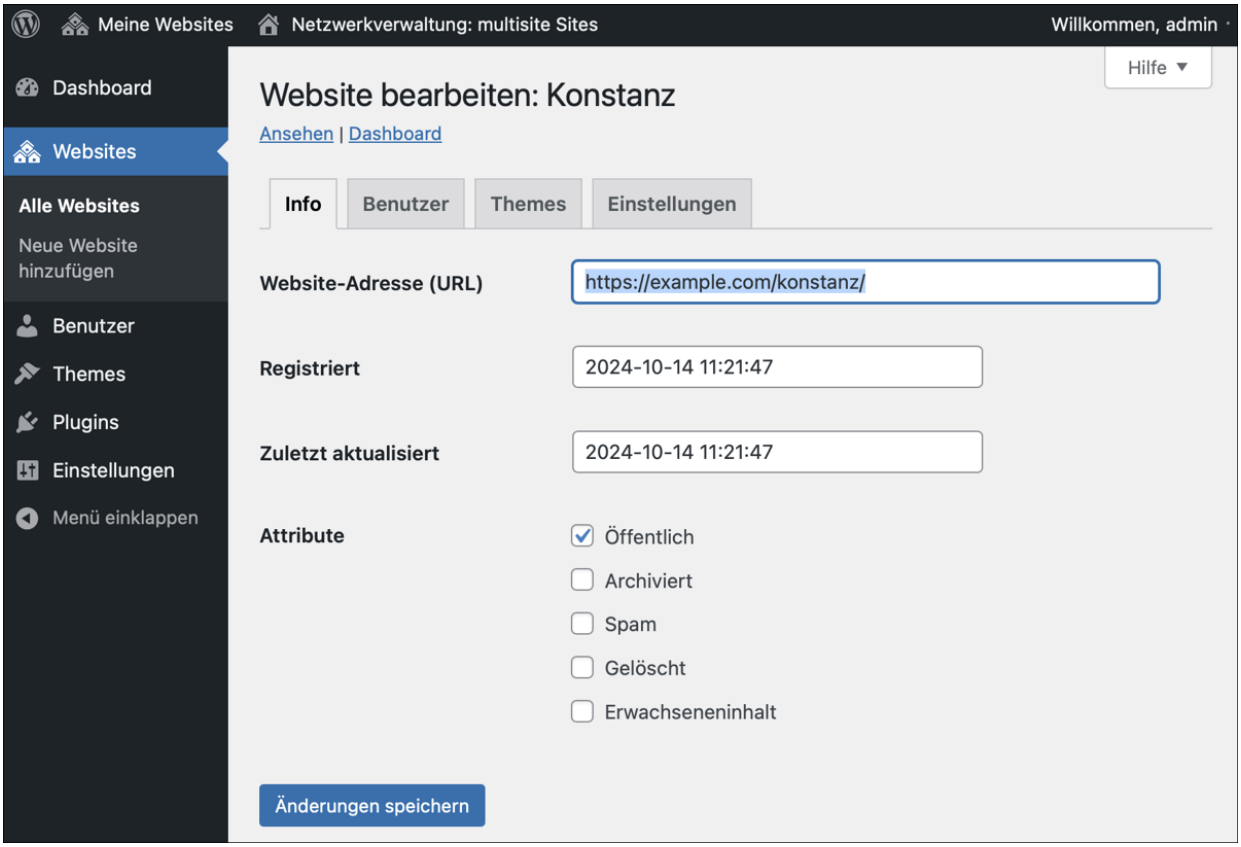


Abbildung 9.1 In den Detail-Einstellungen jeder Site Ihres Netzwerks können Sie nach der Installation die (Sub)domain anpassen (siehe Abschnitt 9.3).

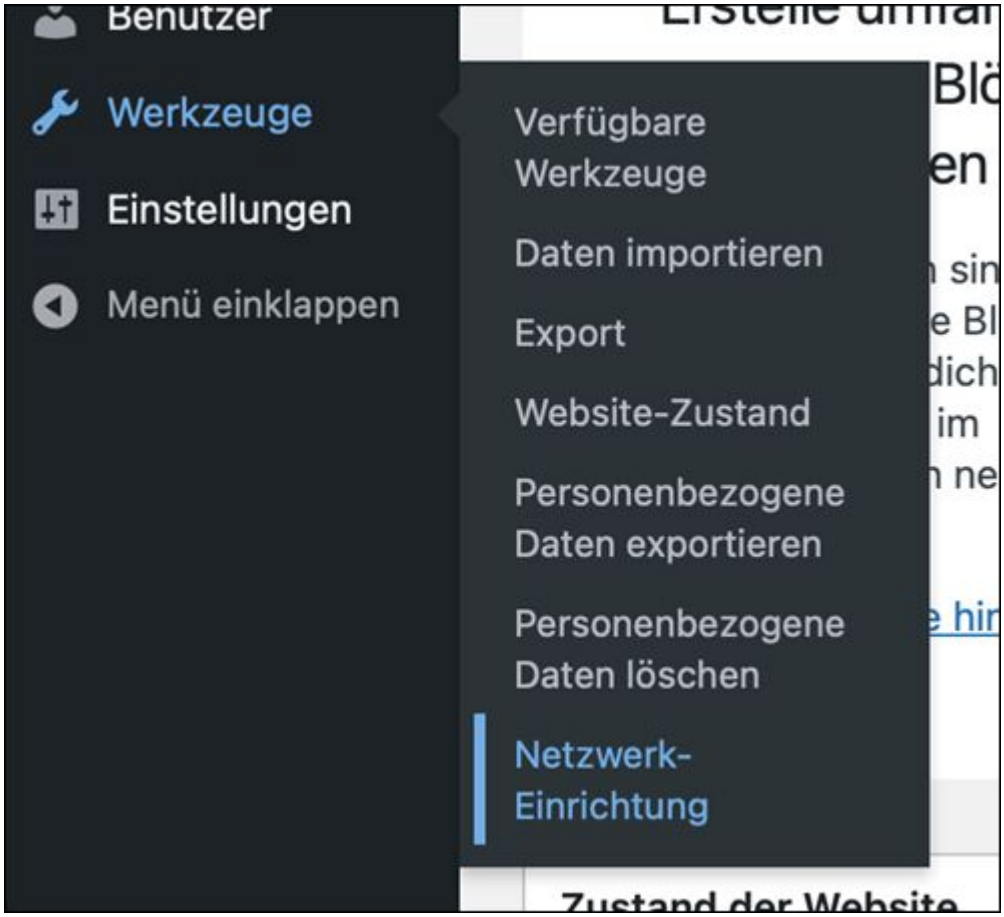


Abbildung 9.2 Wenn Ihre Anpassungen erfolgreich waren, finden Sie unter dem Menüpunkt **WERKZEUGE** einen neuen Eintrag **NETZWERK-EINRICHTUNG**.

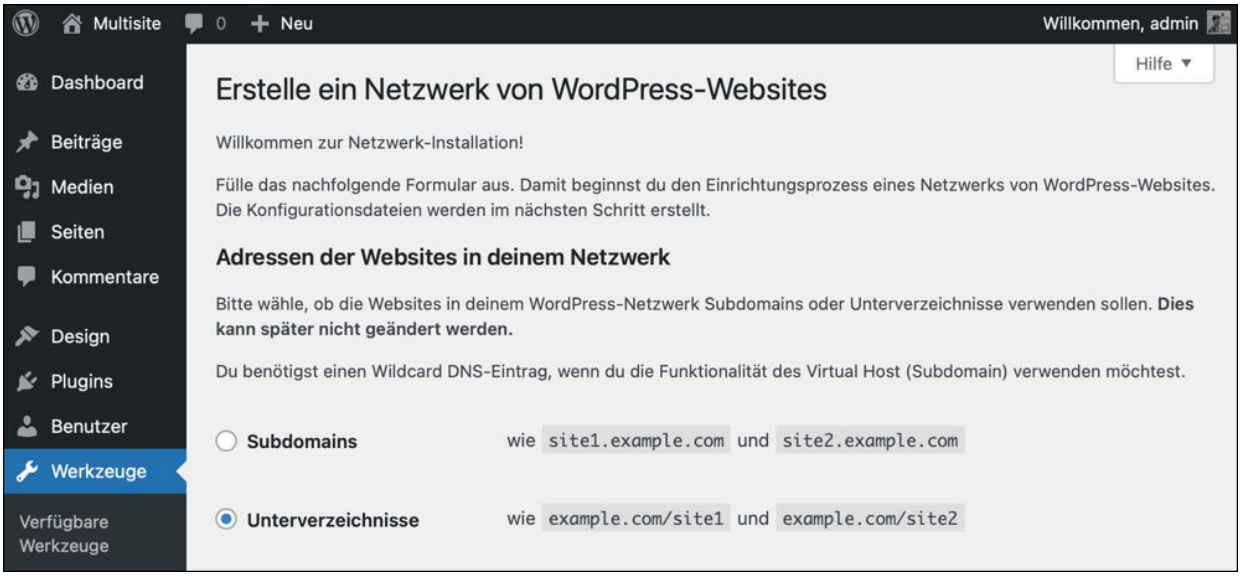


Abbildung 9.3 Hier können Sie Ihre Multisite einrichten und grundlegende Entscheidungen treffen.

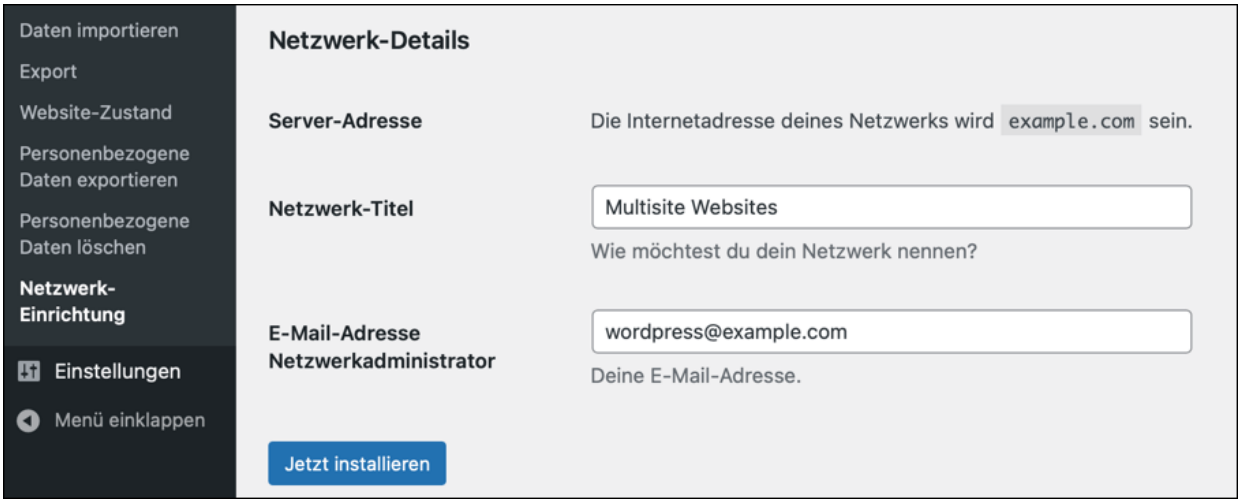


Abbildung 9.4 Zur Netzwerk-Einrichtung gehört auch die Angabe einiger Details, die von WordPress im laufenden Betrieb später verwendet werden.

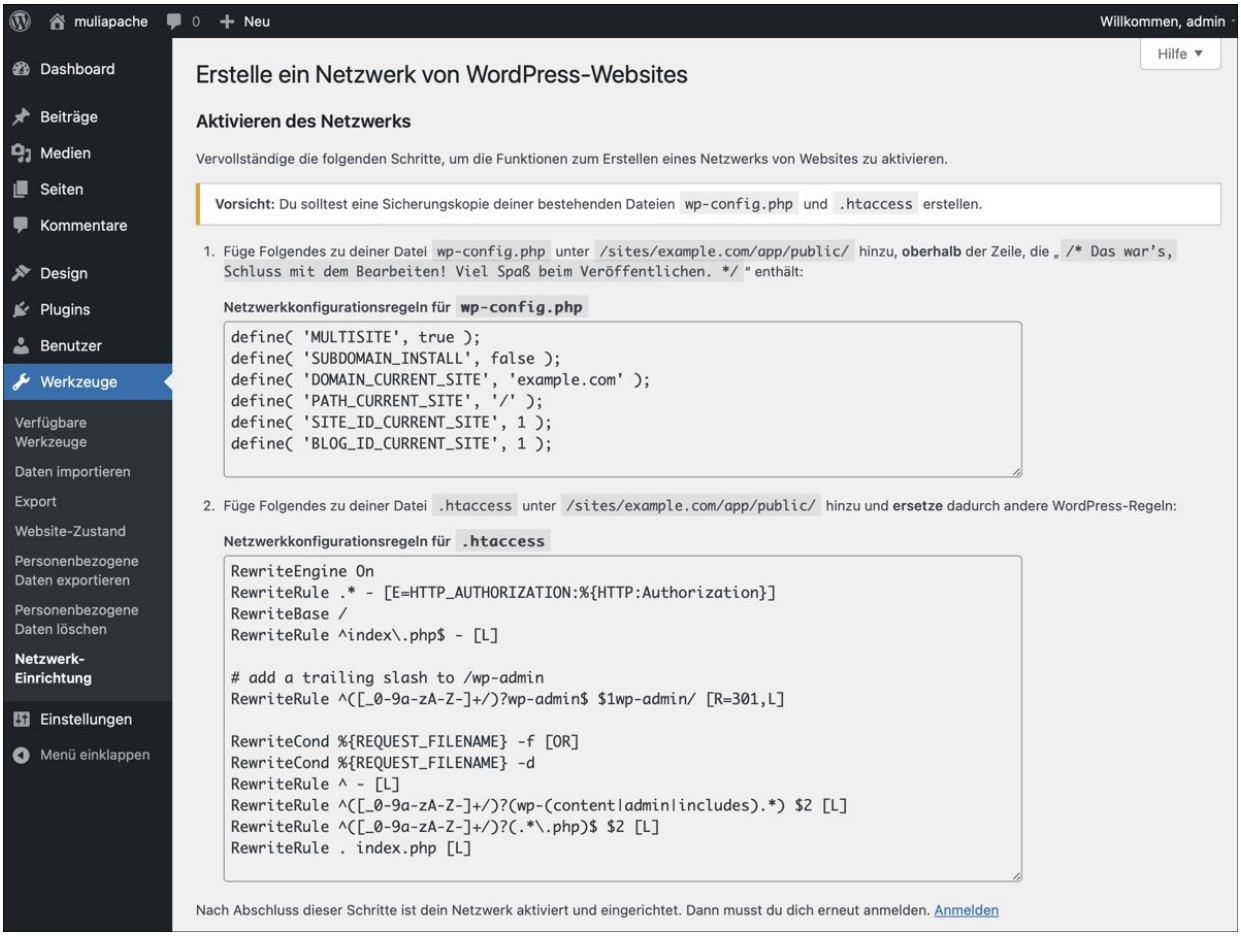


Abbildung 9.5 Auf der folgenden Seite präsentiert WordPress Ihnen zwei Code-Schnipsel, die Sie in verschiedene Dateien Ihrer WordPress-Installation einfügen können.

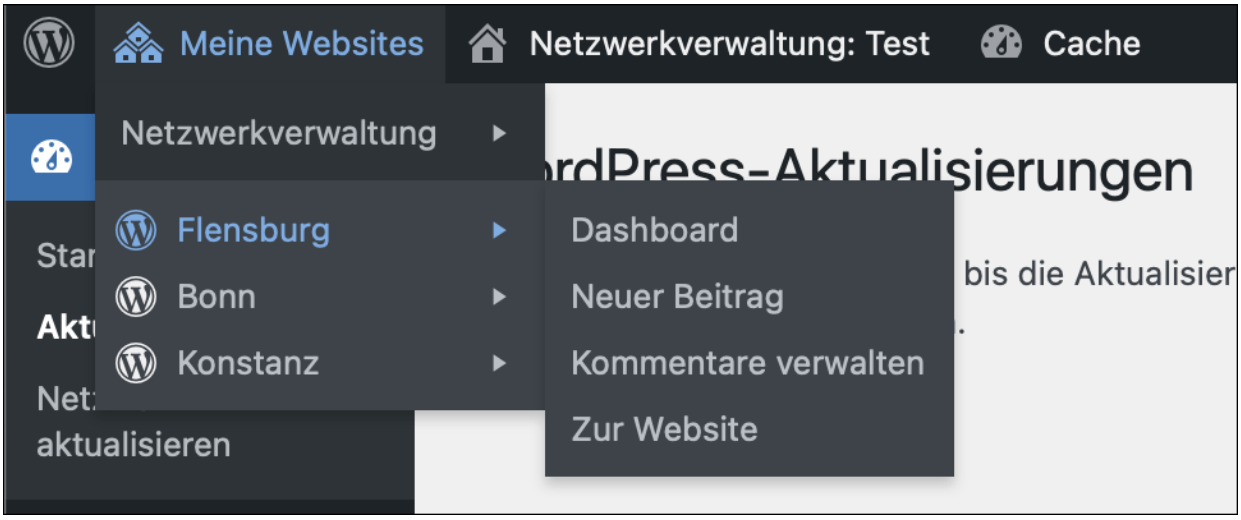


Abbildung 9.6 Als User mit Zugriff auf mehr als eine Site in einem Multisite-Netzwerk sehen Sie in der Werkzeugleiste oben links ein Element zum Wechsel zwischen den Sites.

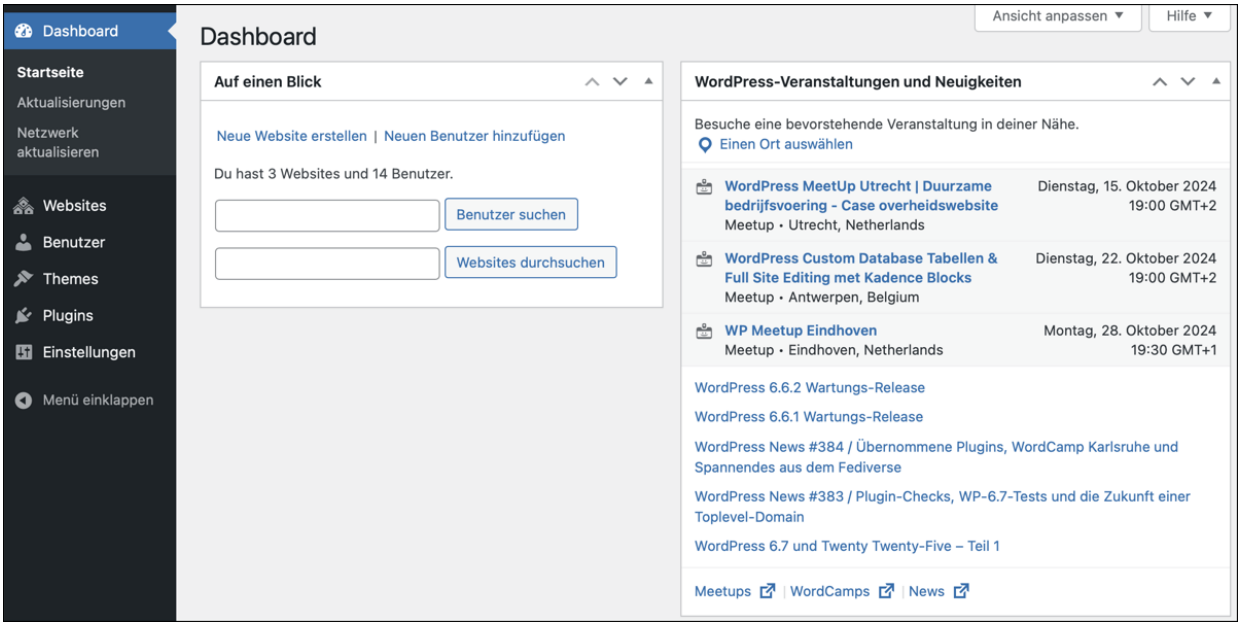


Abbildung 9.7 Das Dashboard der Netzwerkverwaltung ist stark an ein normales WordPress-Dashboard angelehnt.

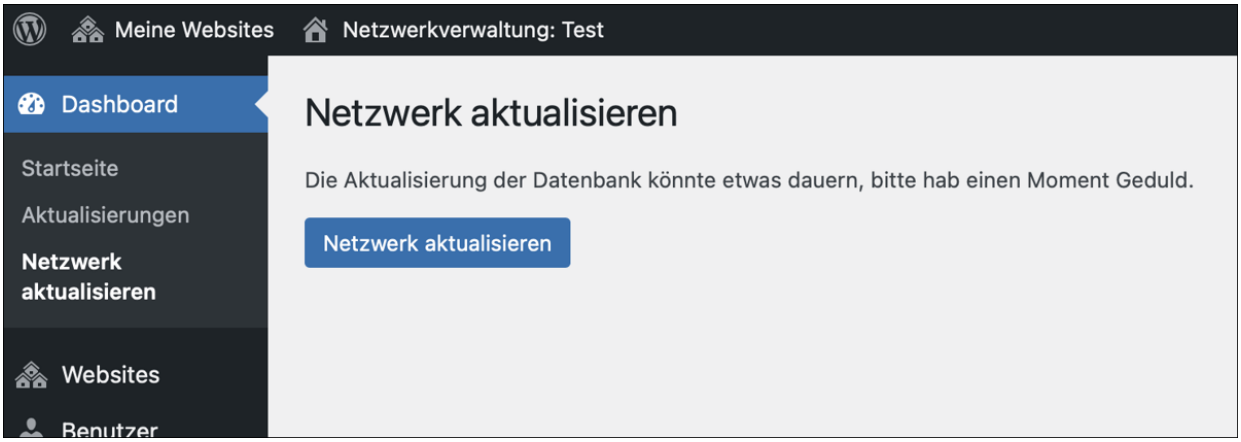


Abbildung 9.8 Auf der Seite »Netzwerk aktualisieren« können Sie sich nach größeren Updates um die Datenbank Ihrer Multisite kümmern.

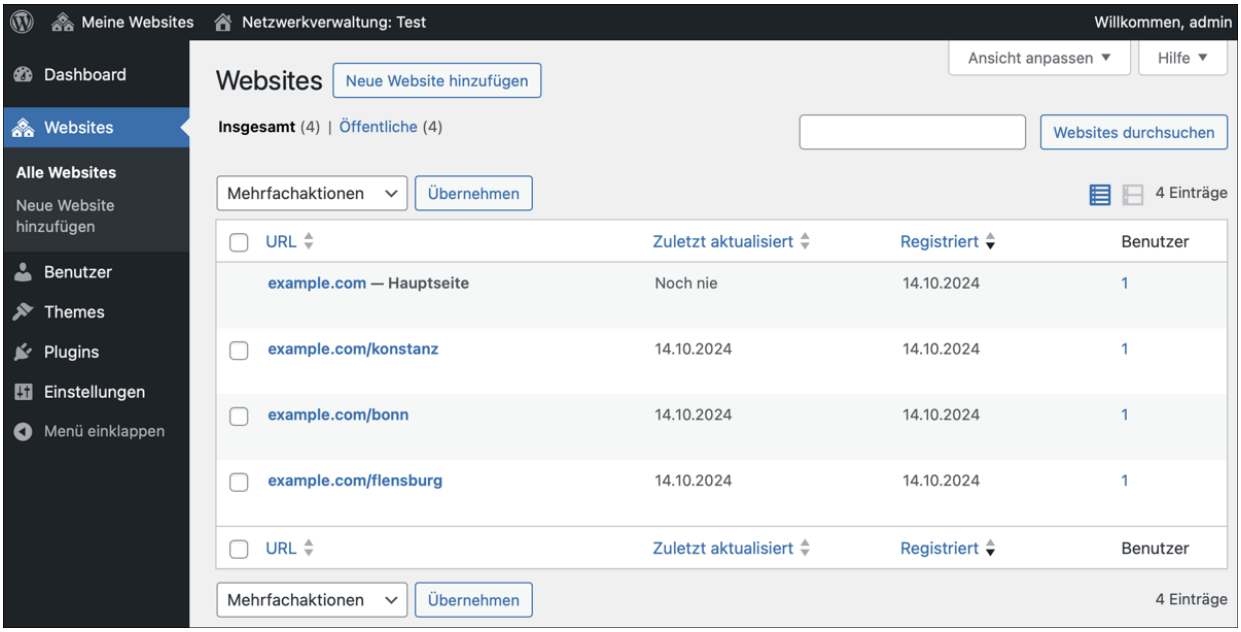


Abbildung 9.9 Unter dem Eintrag **WEBSITES** finden Sie alle Sites, die in Ihrem Multisite-Netzwerk angelegt sind, und können weitere hinzufügen.

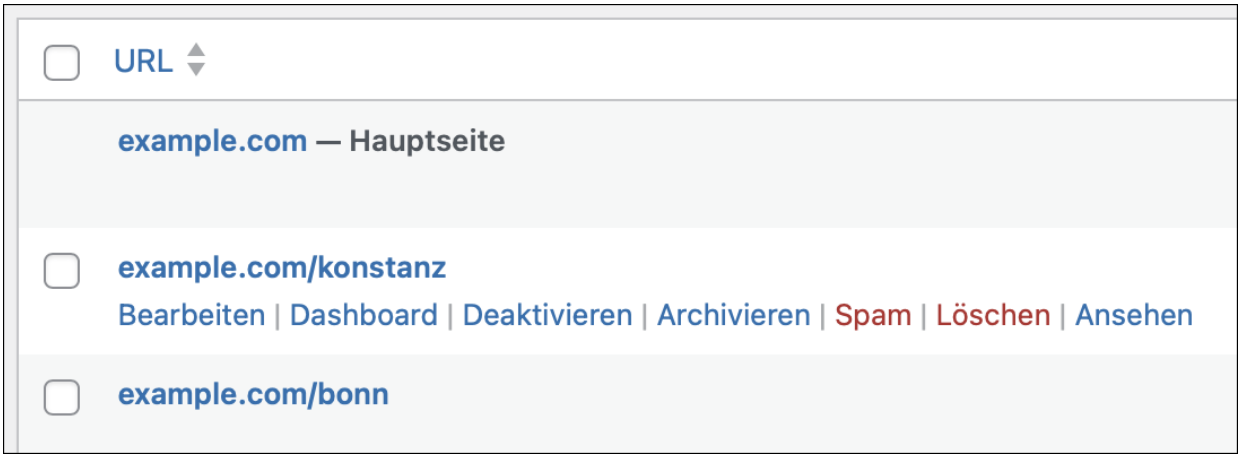


Abbildung 9.10 Wenn Sie Ihre Maus über Einträge in der Tabelle bewegen, zeigt WordPress Ihnen weitere Aktionen für die jeweilige Website in Ihrem Netzwerk an.

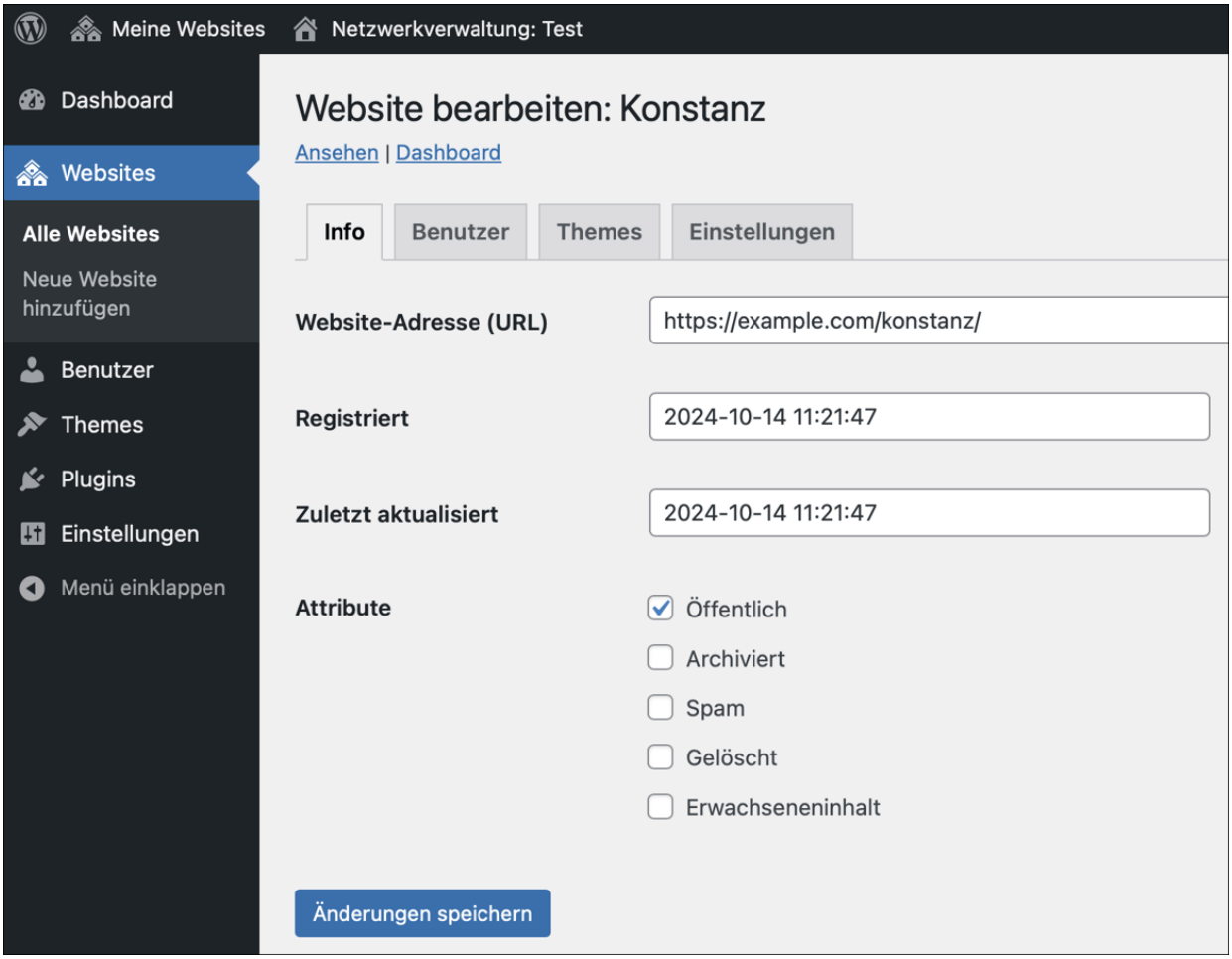


Abbildung 9.11 Die Bearbeiten-Ansicht einer Website kann von Plugins erweitert werden, ist aber immer aus verschiedenen Tabs zusammengesetzt.

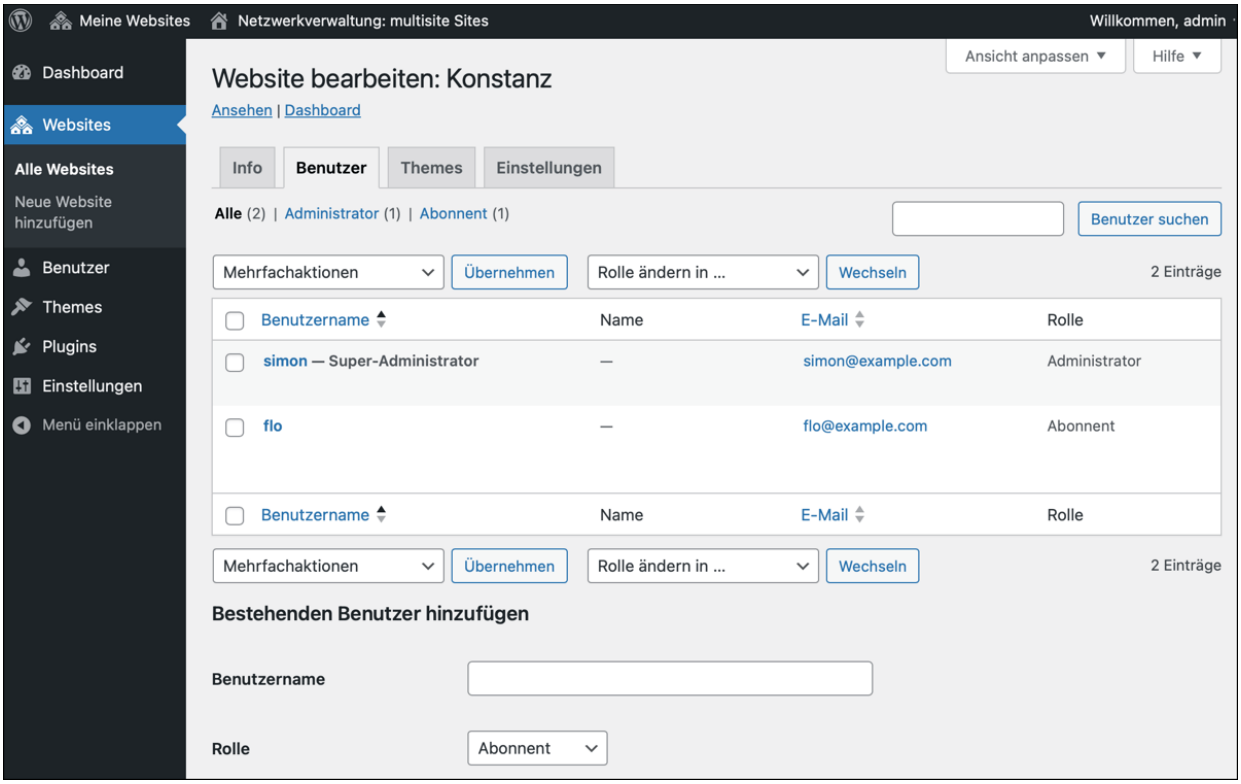


Abbildung 9.12 Der Benutzer-Tab gibt Ihnen eine Übersicht über alle Accounts mit Zugriff auf eine bestimmte Website in Ihrem Multisite-Netzwerk.

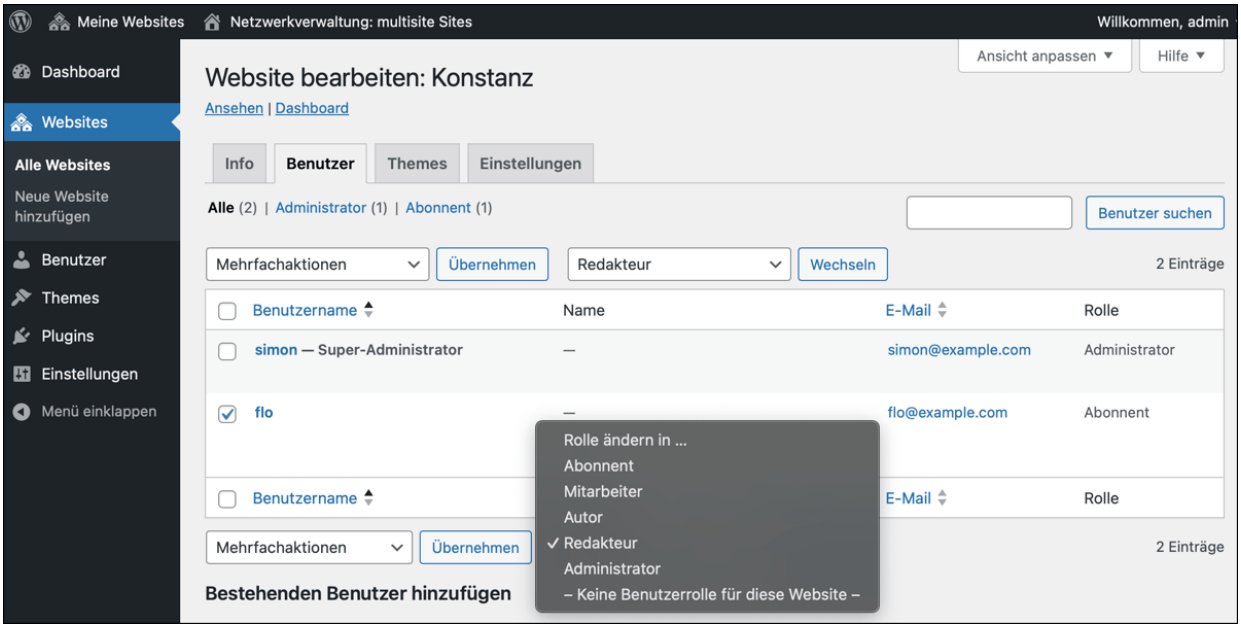


Abbildung 9.13 Direkt aus der Übersicht heraus können Sie die Rolle eines oder mehrerer Benutzer ändern.

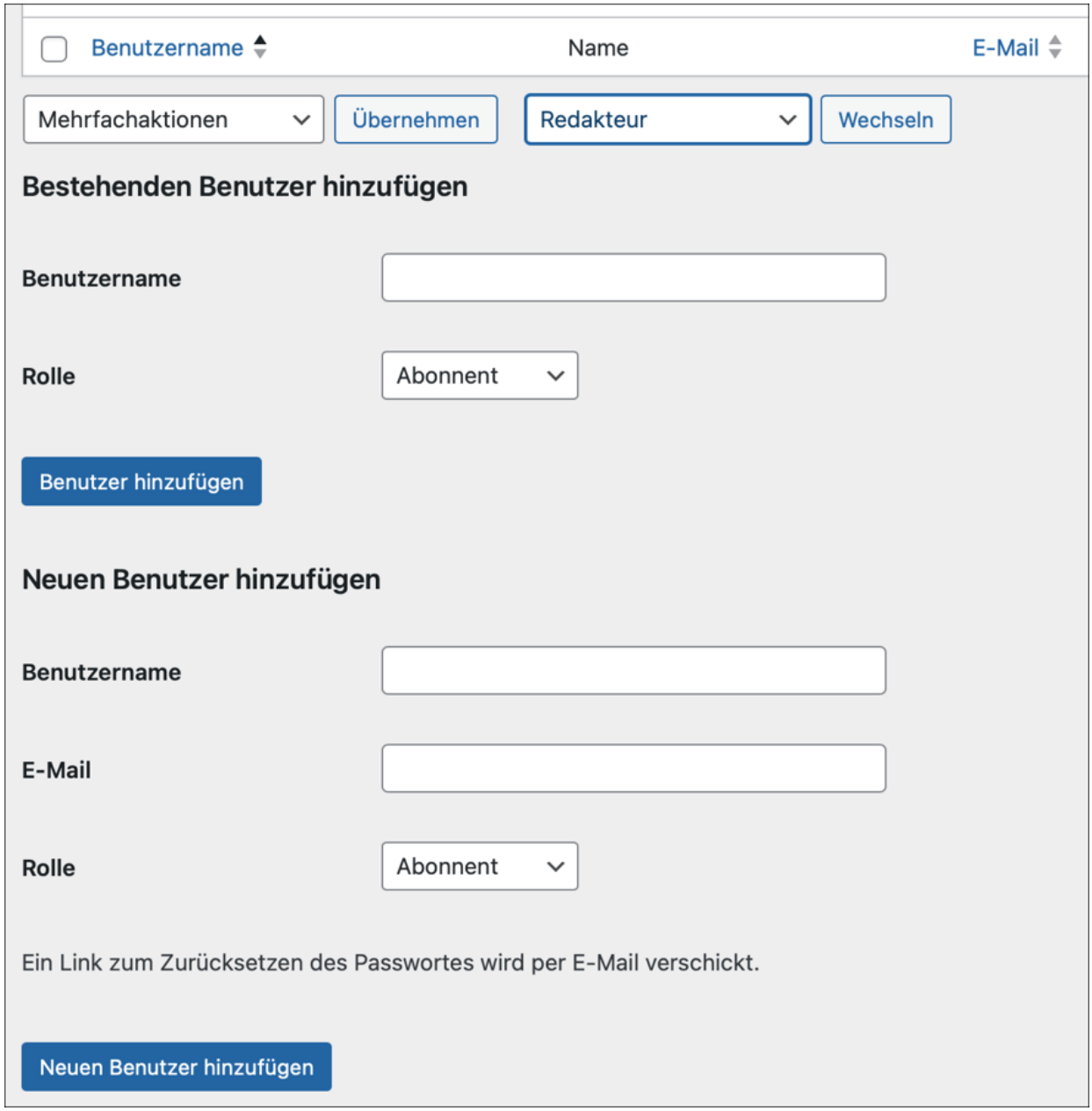


Abbildung 9.14 Unterhalb der Benutzer-Tabelle können Sie über die beiden Formulare neue User zu einer Website Ihrer Multisite hinzufügen.

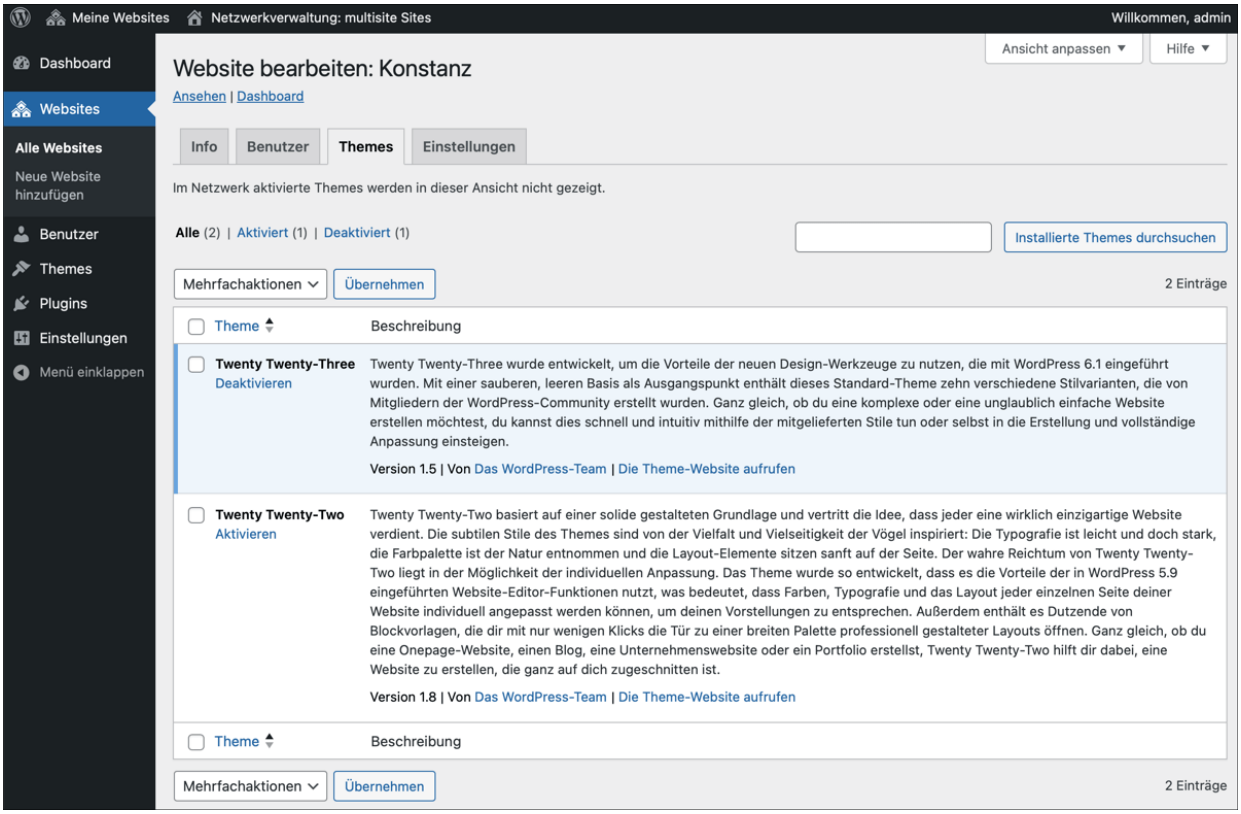


Abbildung 9.15 Im nächsten Tab können Sie auswählen, welche Themes auf dieser Website Ihres Netzwerks nutzbar sein sollen.

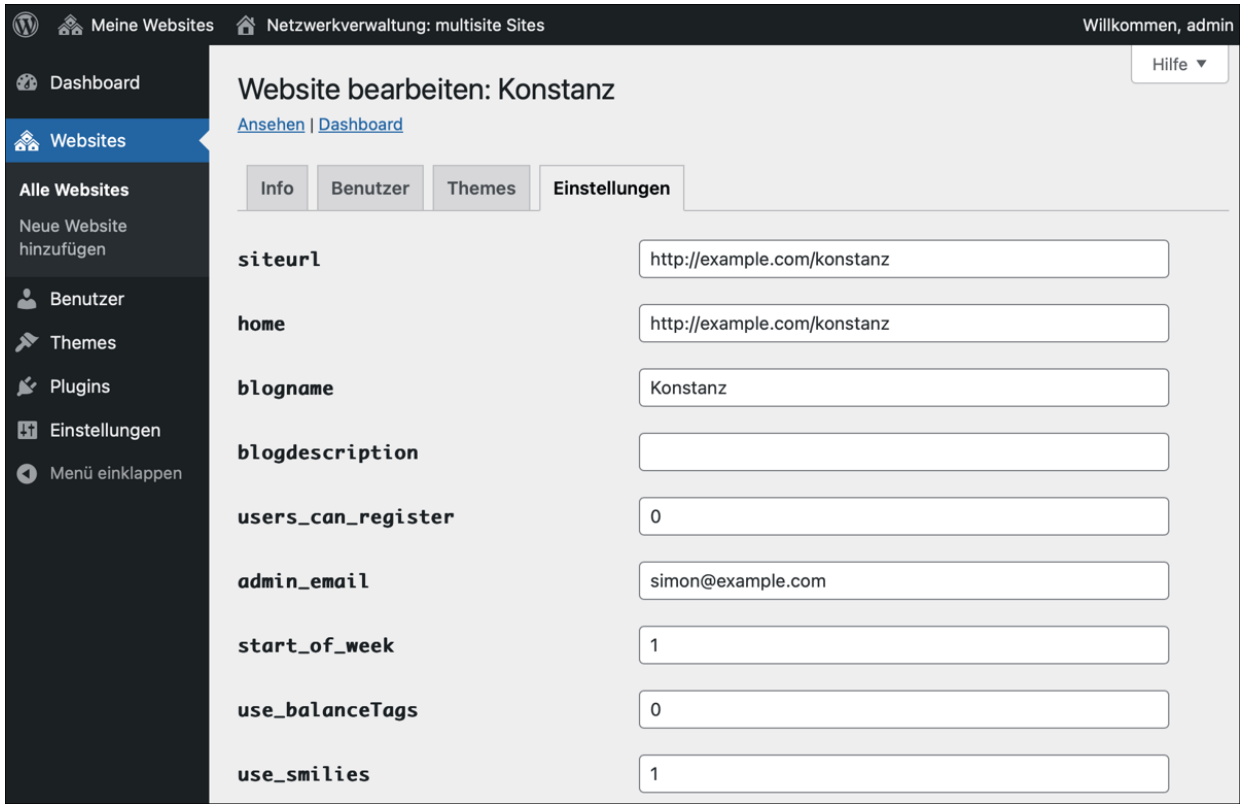


Abbildung 9.16 Im »Einstellungen«-Tab präsentiert sich Ihnen eine vollständige (und vollständig unformatierte) Liste aller Einstellungen einer Website.

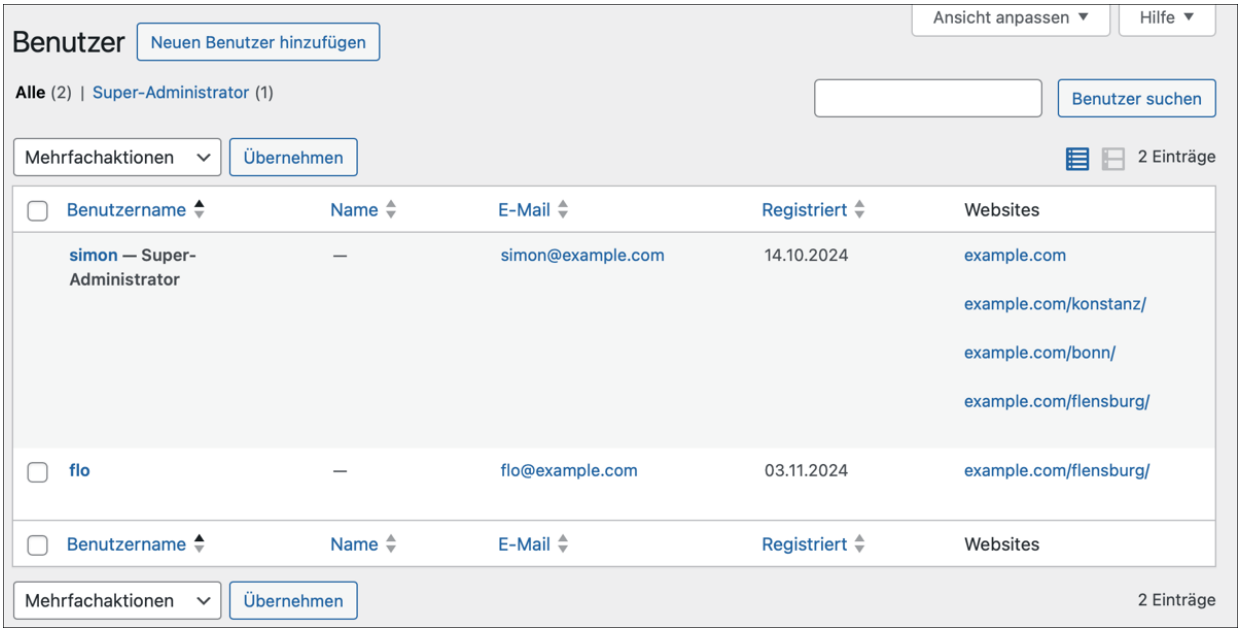


Abbildung 9.17 Auf dieser Seite erhalten Sie einen Überblick über alle Benutzer*innen auf Ihrer Website.

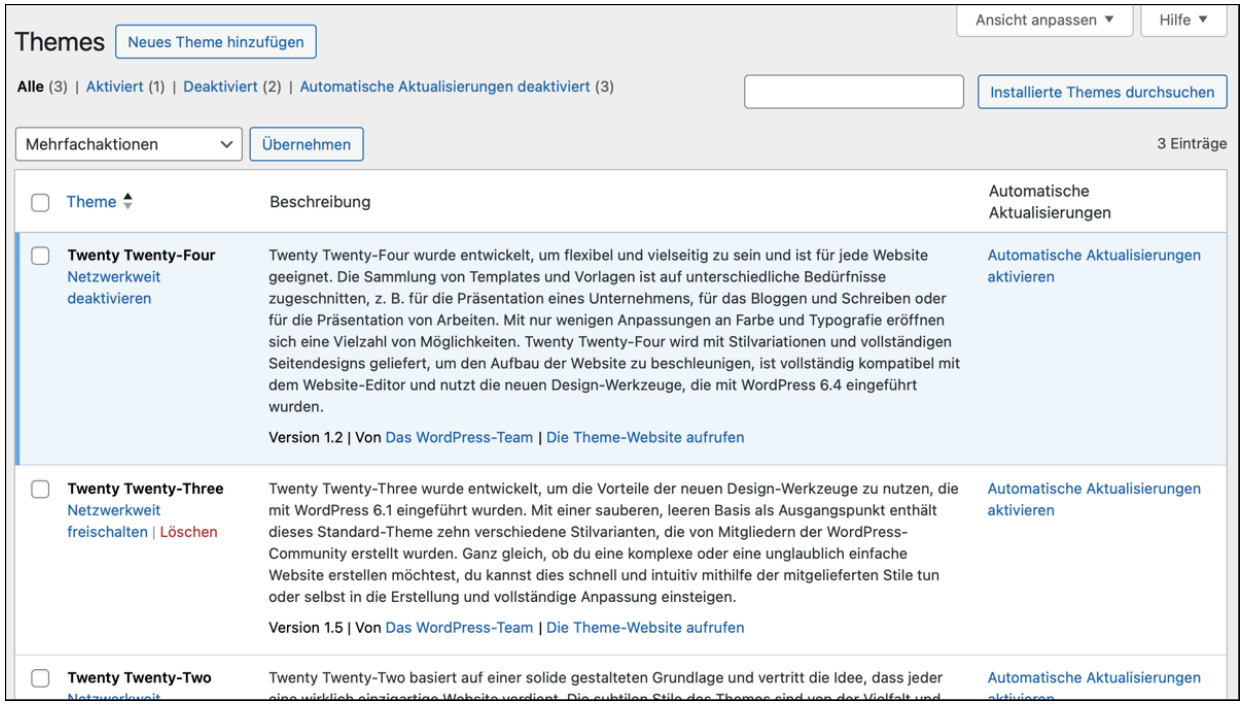


Abbildung 9.18 Als einfache Tabelle präsentieren sich Ihnen die Theme-Einstellungen Ihrer Multisite.

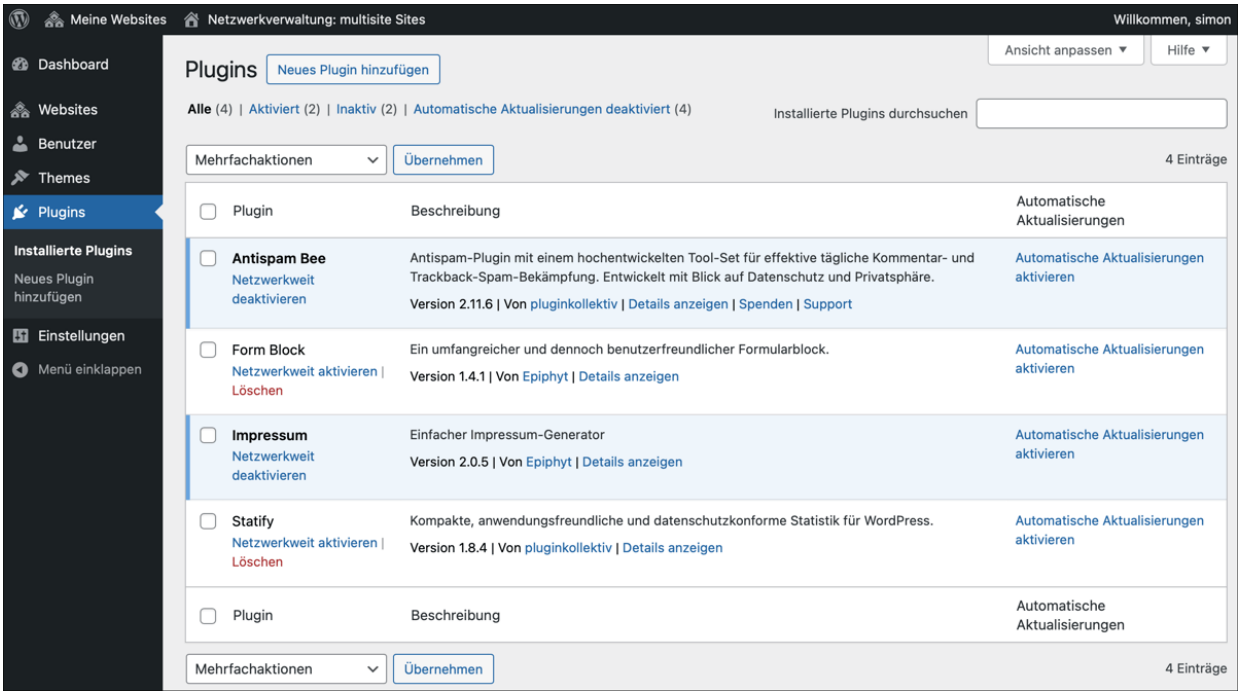


Abbildung 9.19 In der Plugin-Verwaltung erhalten Sie einen Überblick über die verschiedenen Plugins Ihrer Multisite und können weitere hinzufügen.

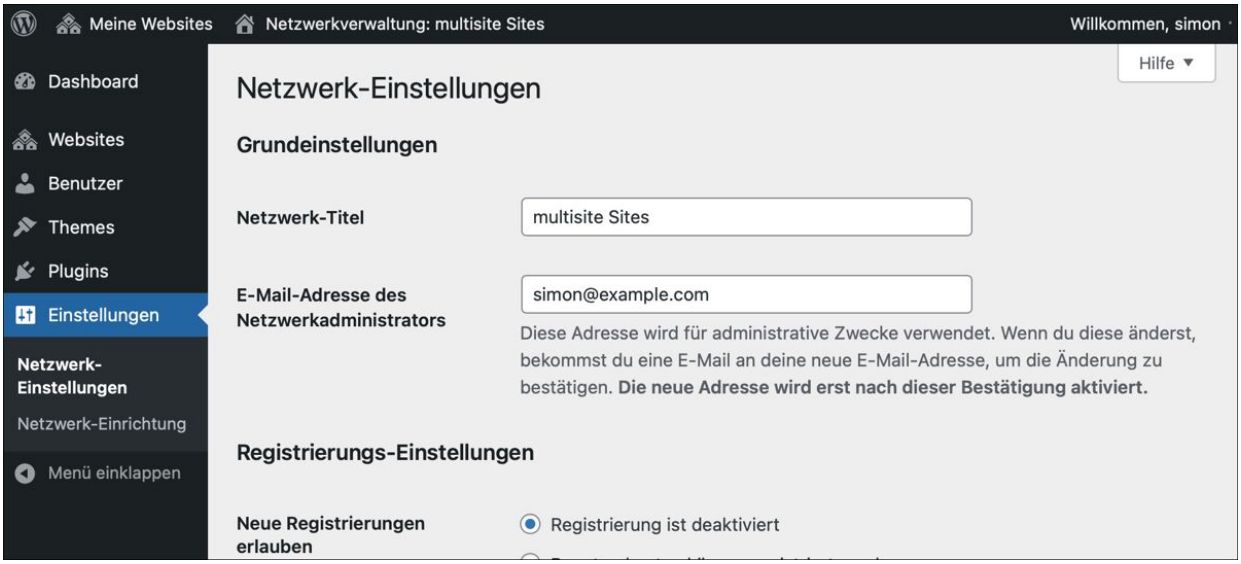


Abbildung 9.20 Der erste Abschnitt der Netzwerk-Einstellungen dreht sich um einige Grundeinstellungen.

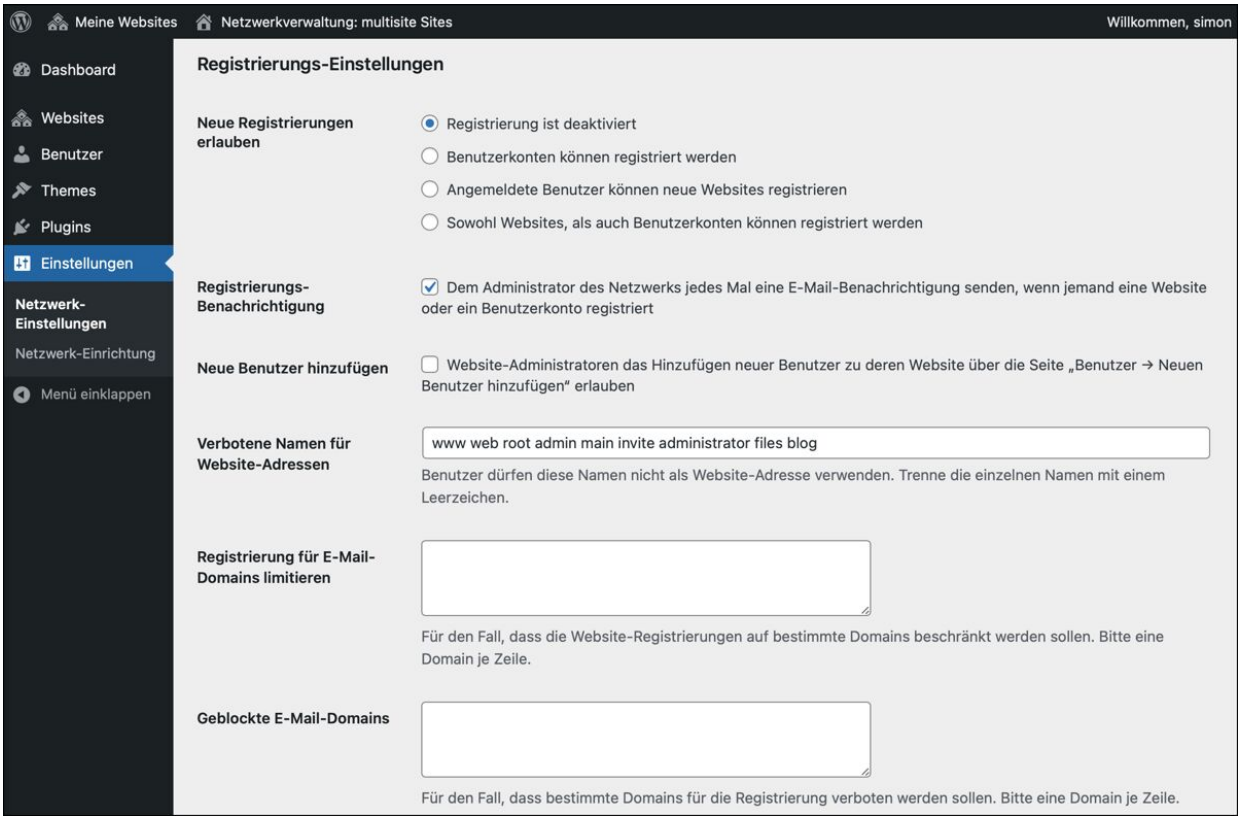


Abbildung 9.21 Im nächsten Abschnitt der Netzwerkeinstellungen können Sie verschiedene Vorgaben zum Verhalten Ihrer Multisite rund um das Anlegen von neuen User-Accounts und Sites machen.

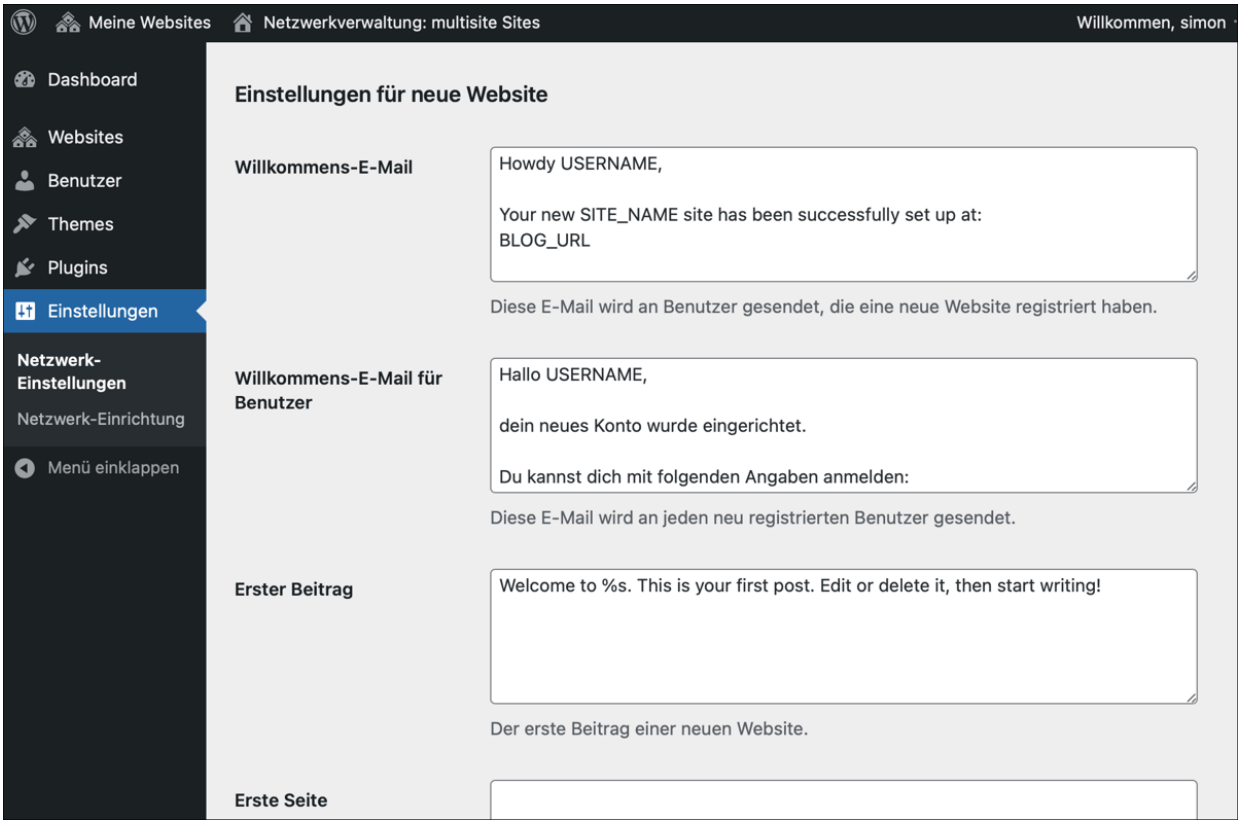


Abbildung 9.22 Der Abschnitt »Einstellungen für neue Websites« enthält allerlei Vorlagen, die beim Anlegen einer neuen Website in Ihrem Multisite-Netzwerk zum Einsatz kommen.

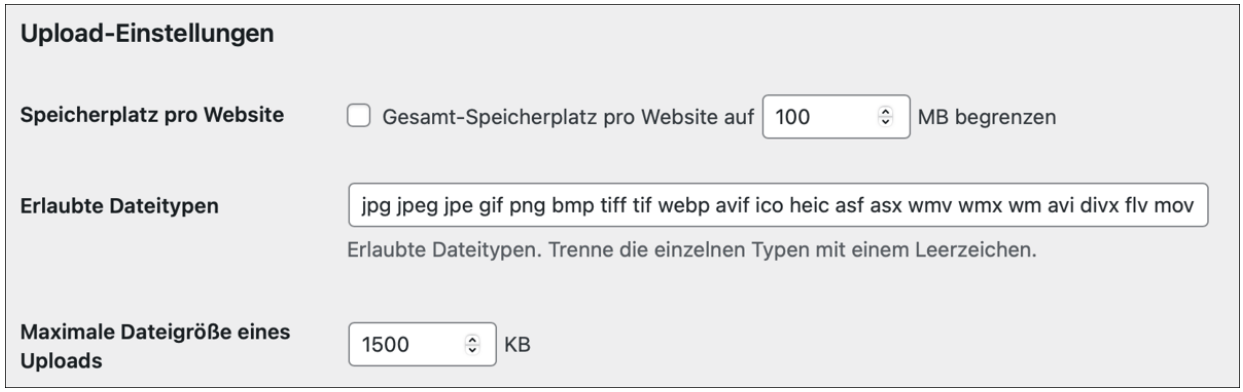


Abbildung 9.23 Die Upload-Einstellungen geben Ihnen Kontrolle darüber, wie viele und welche Dateien in Ihrer Multisite hochgeladen werden können.

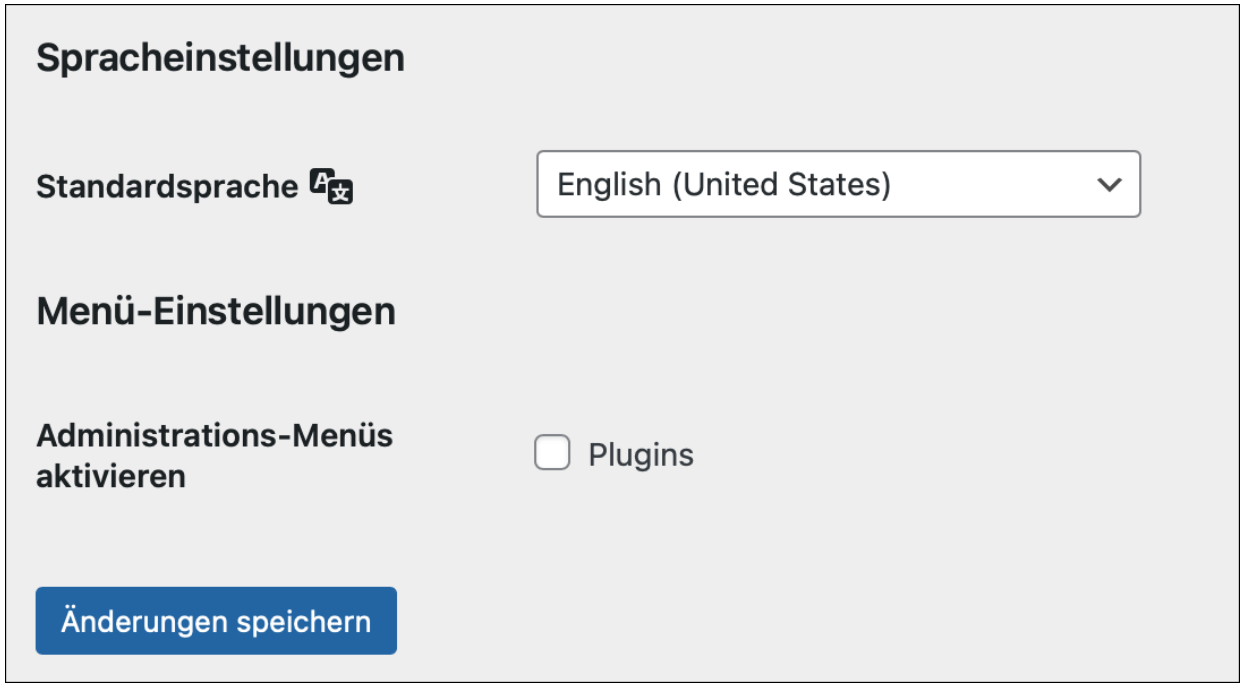


Abbildung 9.24 In den Netzwerk-Einstellungen können Sie außerdem festlegen, in welcher Sprache neue Websites in Ihrem Netzwerk angelegt werden sollen und ob Website-Admins Zugriff auf die Plugin-Verwaltung haben sollen.

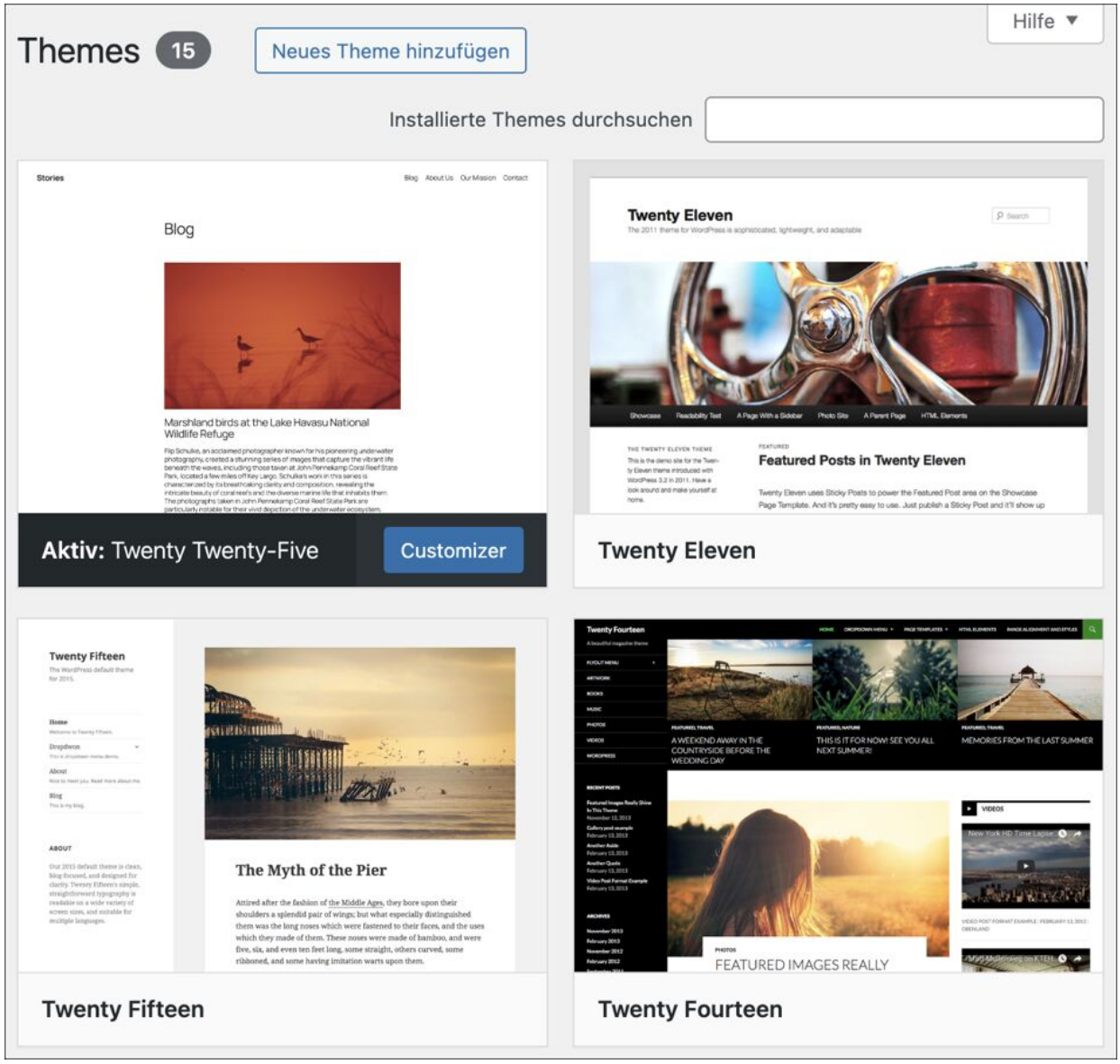


Abbildung 10.1 Eine frische WordPress-Installation kommt mit mehreren Standard-Themes von WordPress, von denen jedes Jahr ein neues veröffentlicht wird.

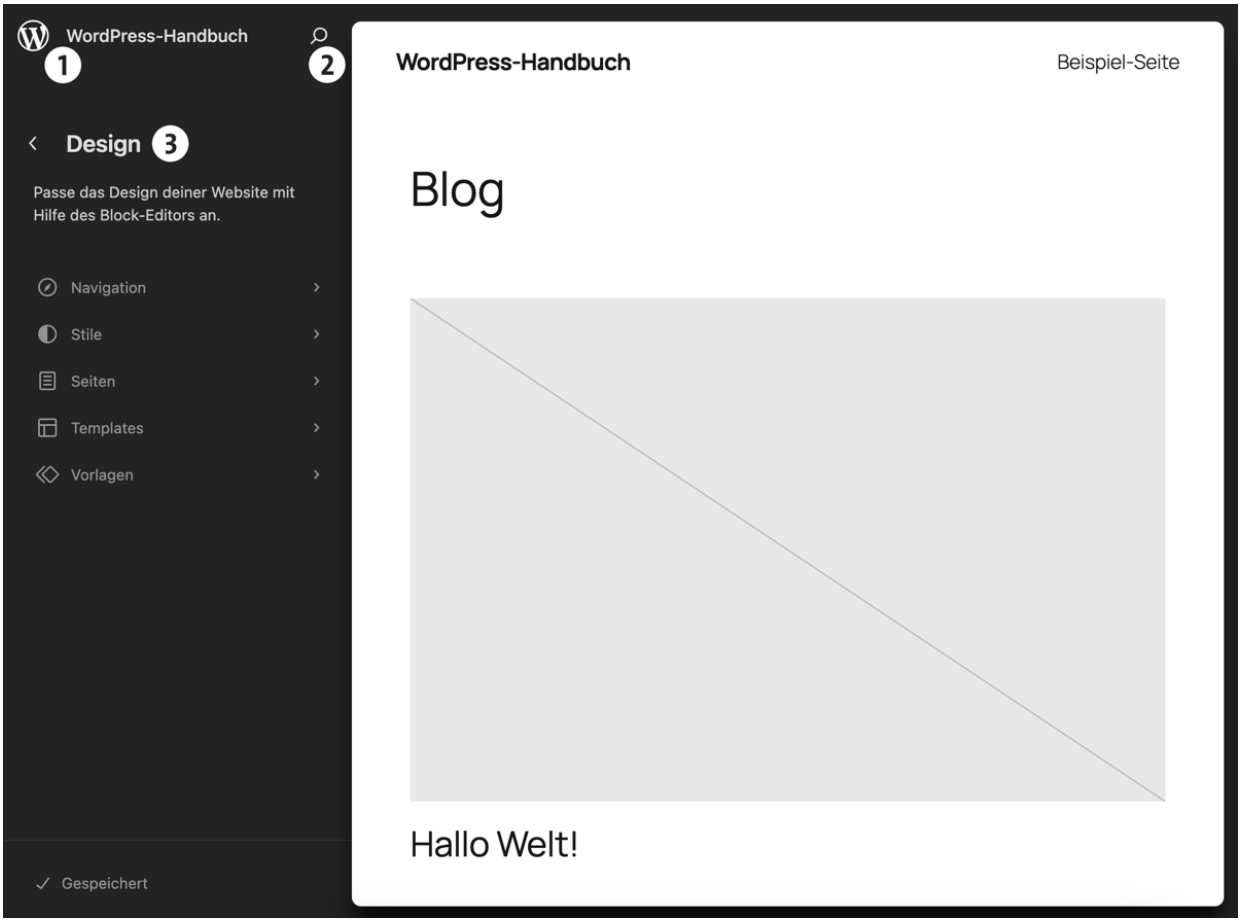


Abbildung 10.2 Der Website-Editor erlaubt die umfangreiche Bearbeitung von Websites mit Block-Themes direkt im Backend.

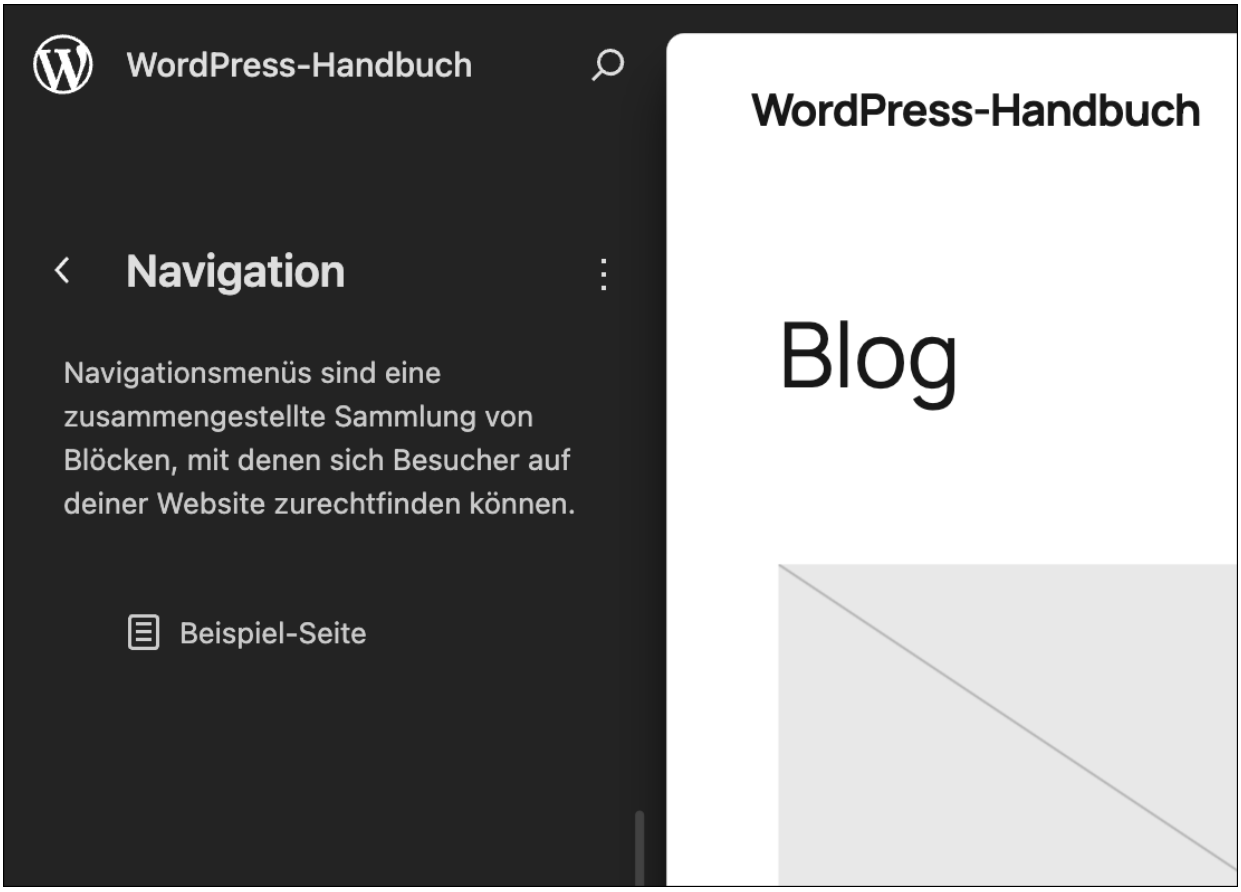


Abbildung 10.3 Über den »Navigation«-Menüpunkt in der Seitenleiste gelangen Sie zu einer Liste Ihrer Navigationsmenüs und können die Menüs bearbeiten.

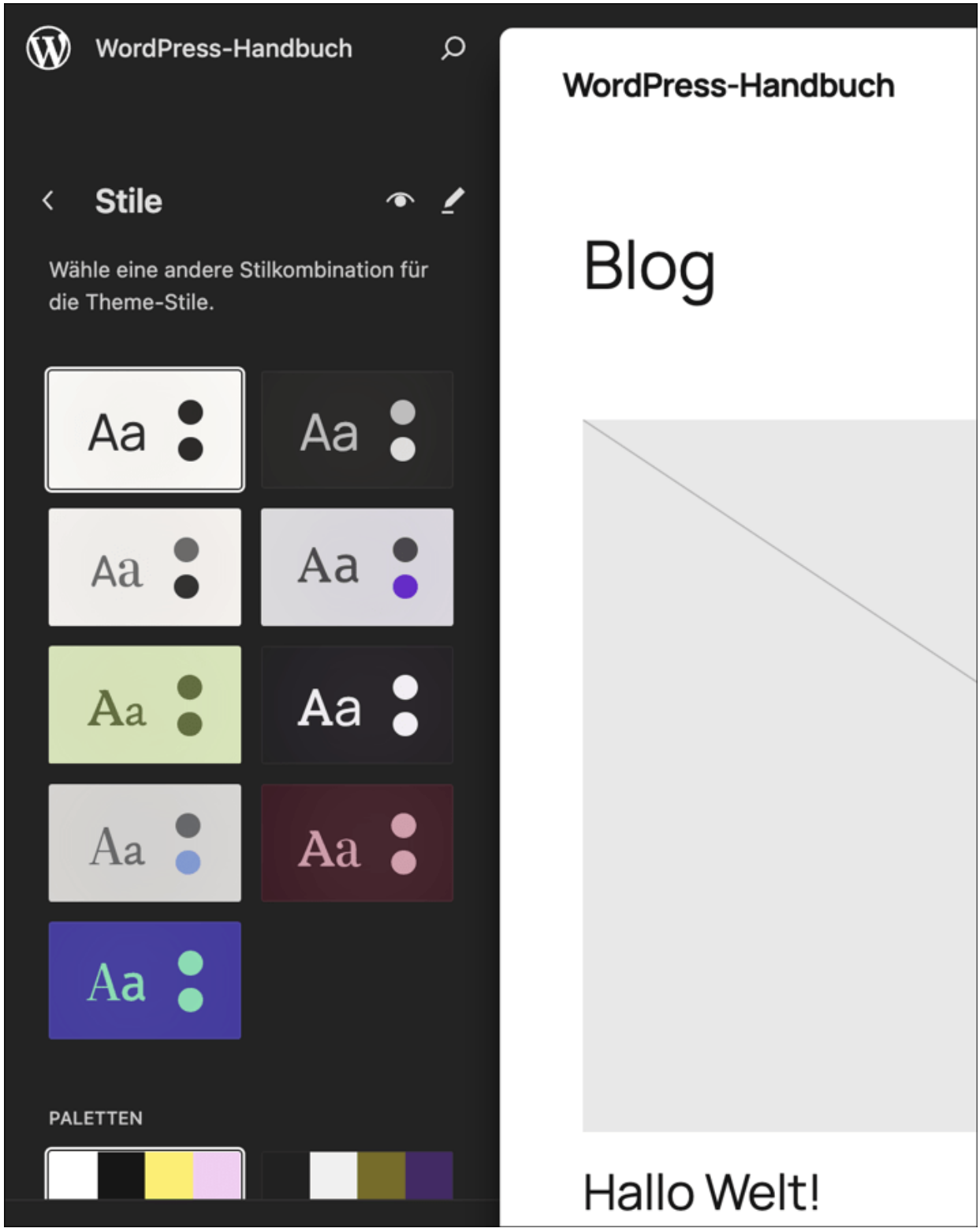

WordPress-Handbuch
Stile
Wähle eine andere Stilkombination für die Theme-Stile.
Aa
Aa
Aa
Aa
Aa
Aa
Aa
Aa
Aa
PALETTEN
WordPress-Handbuch
Blog
Hallo Welt!

Abbildung 10.4 Themes können unterschiedliche »Stile« mitbringen, die Variationen des normalen Designs sind und von Ihnen einfach aktiviert werden können.

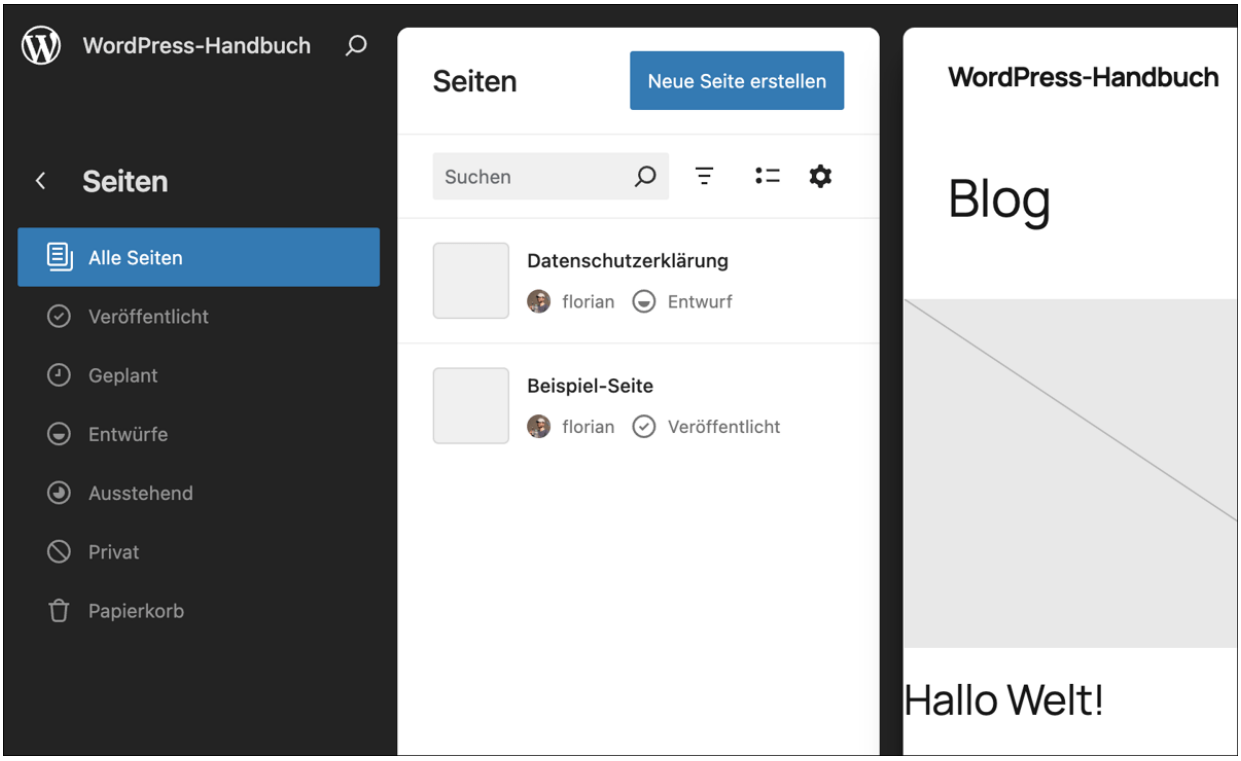


Abbildung 10.5 Im Website-Editor finden Sie auch eine Liste der Seiten Ihrer Installation, die Ihnen von den Funktionen bekannt vorkommen könnten.

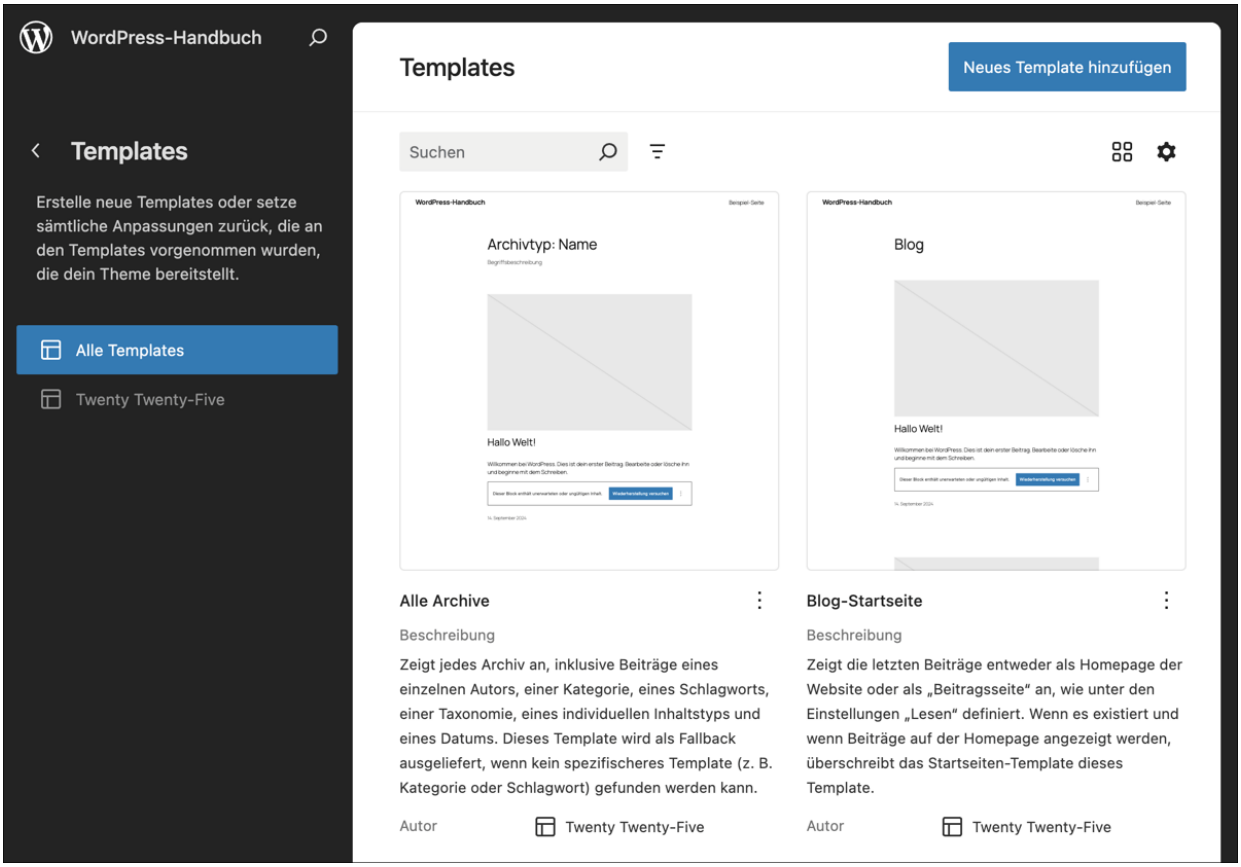


Abbildung 10.6 Auf der rechten Seite wird eine Liste aller Templates angezeigt.

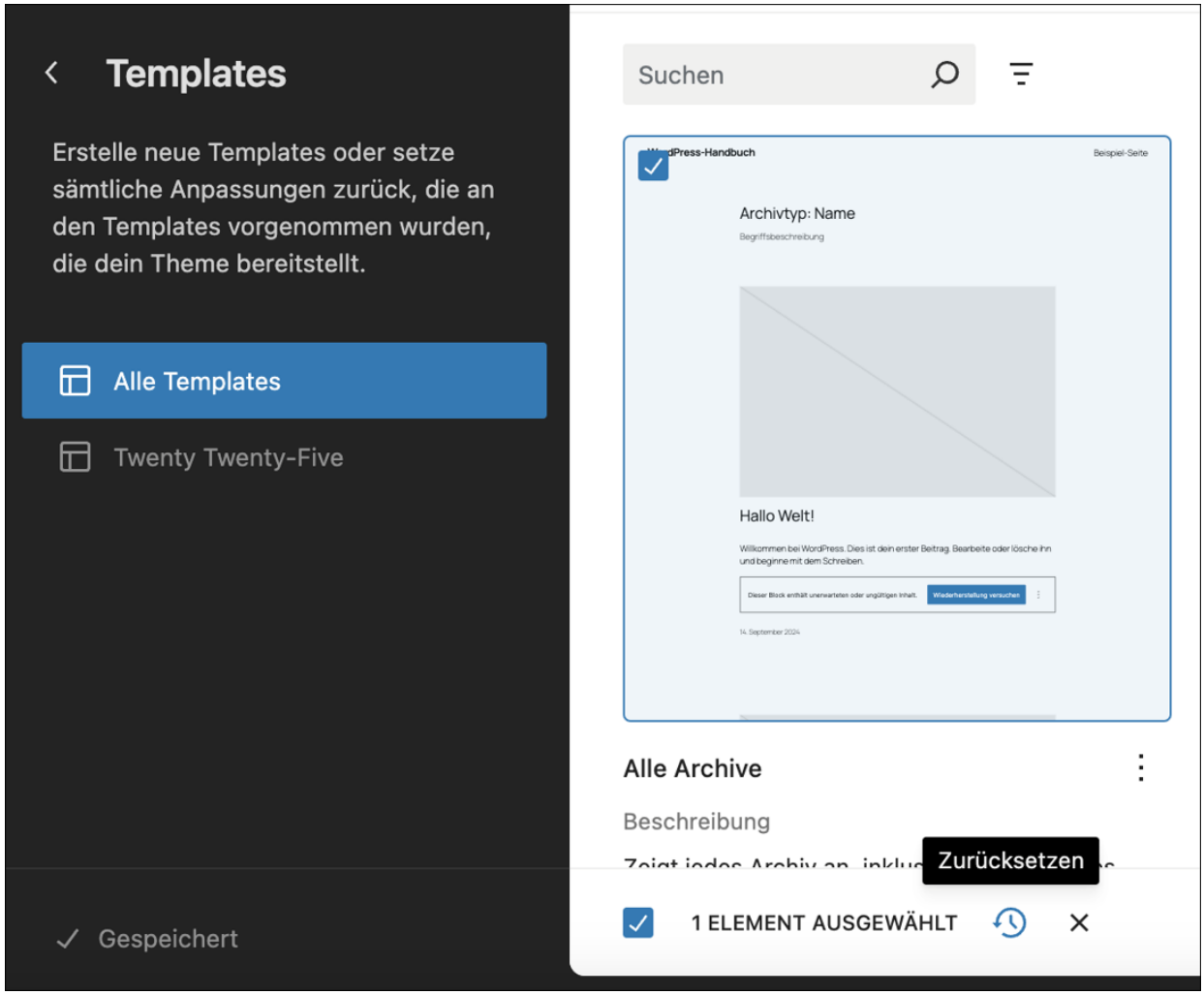


Abbildung 10.7 Veränderte Templates können wieder in den Ursprungszustand zurückgesetzt werden.

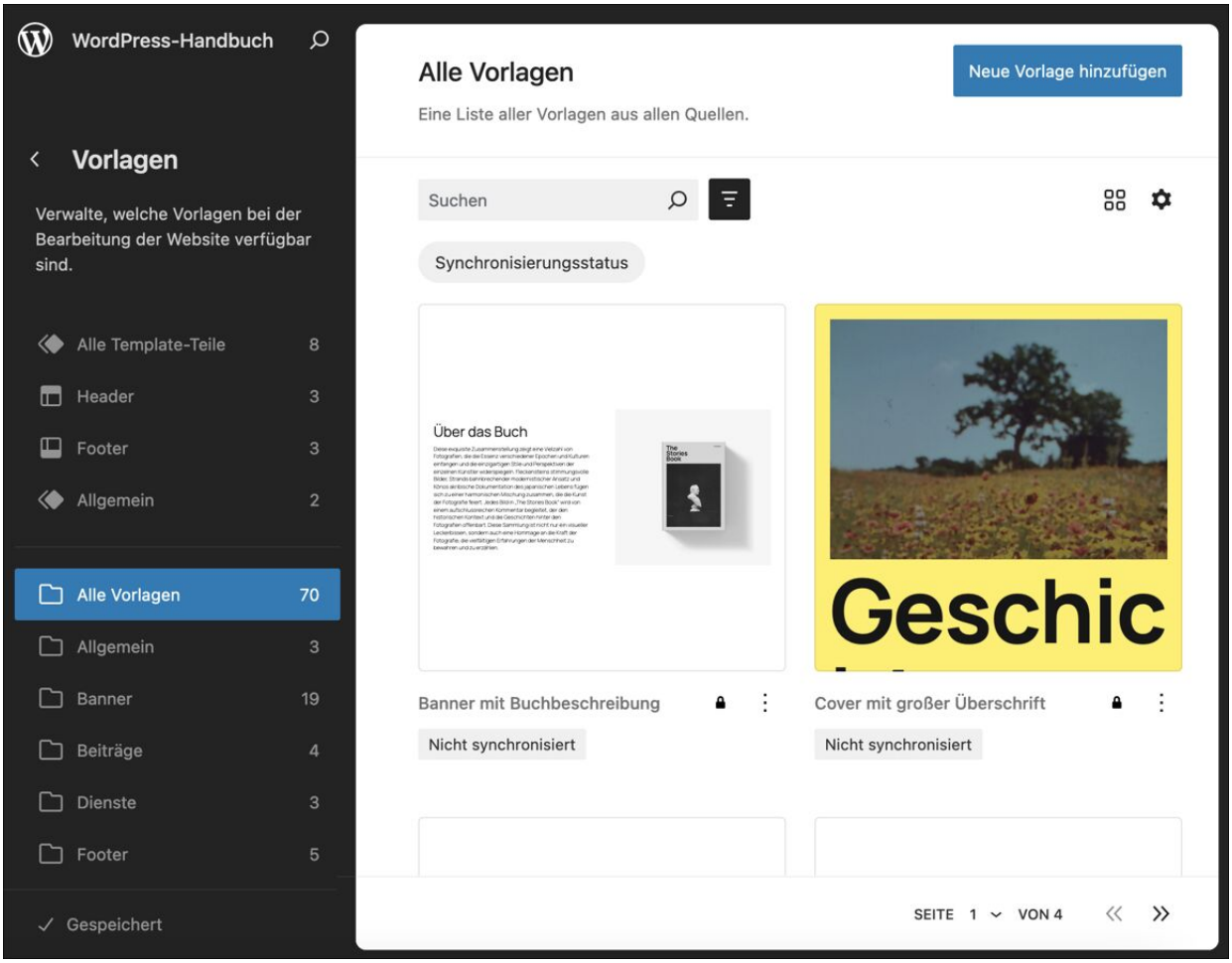


Abbildung 10.8 Unter dem Punkt »Vorlagen« finden Sie sowohl die Template-Teile, etwa den Header und Footer, als auch die Block-Vorlagen, die Block-Kombinationen zum einfachen Einfügen in den Editor bereitstellen.

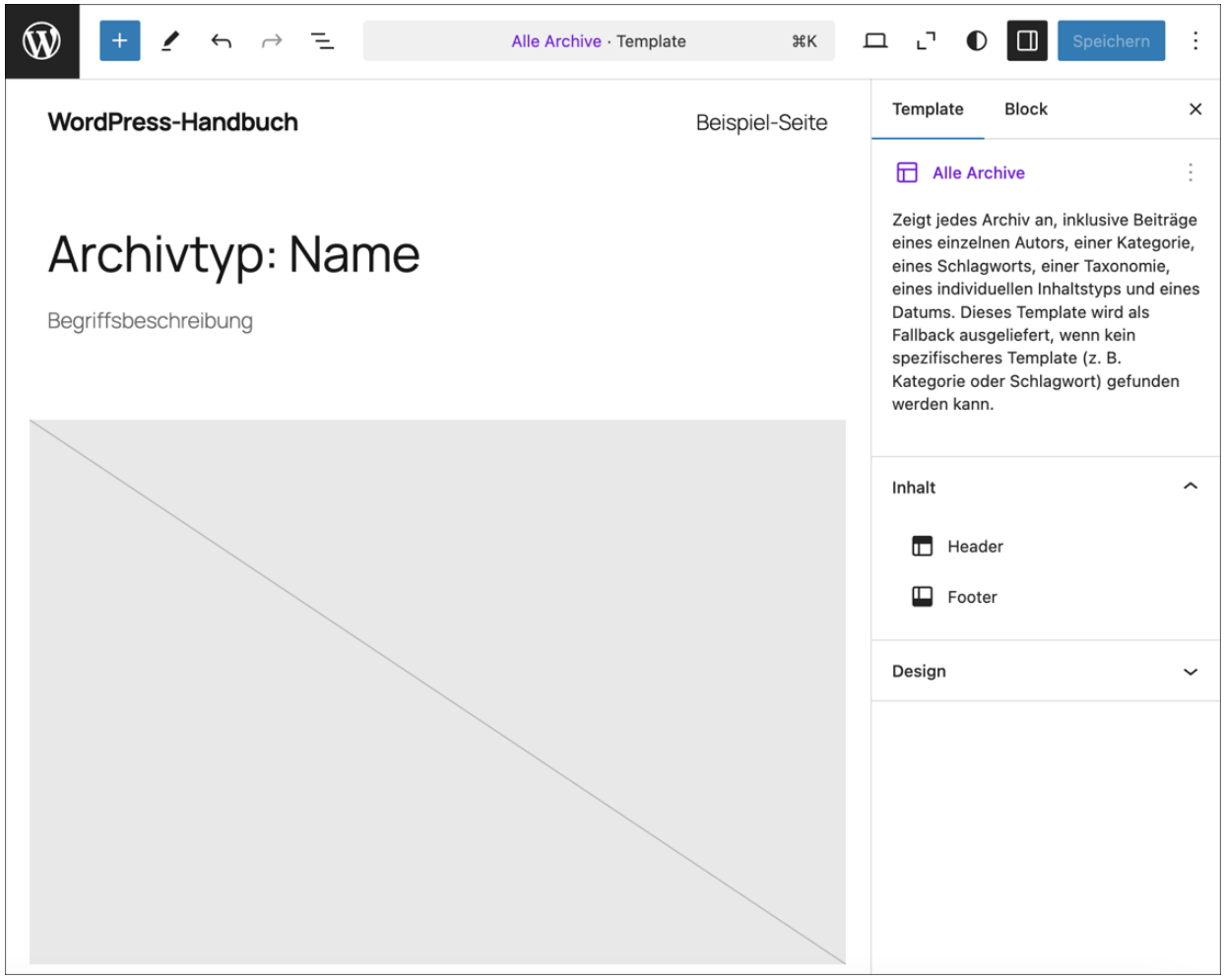


Abbildung 10.9 Diese Ansicht kommt Ihnen wahrscheinlich aus der Bearbeitungsansicht von Seiten und Beiträgen bekannt vor.

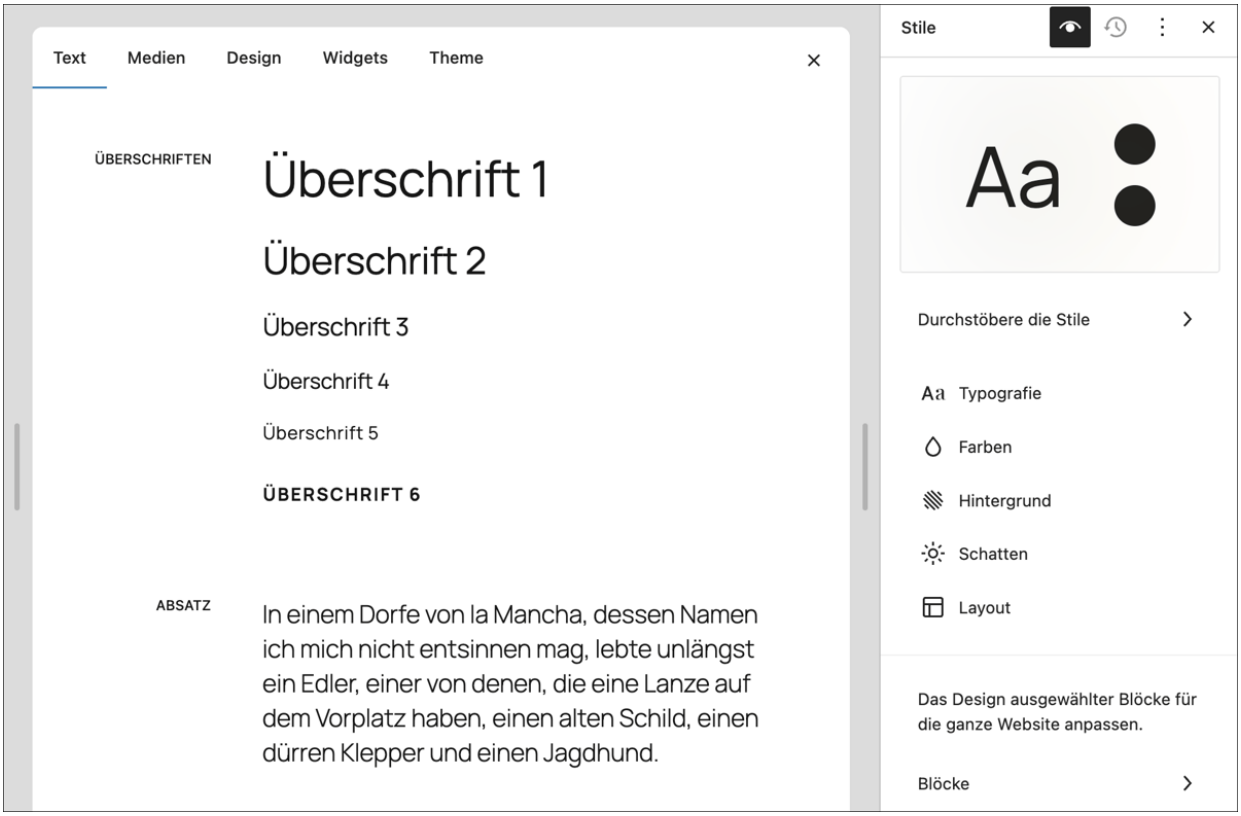
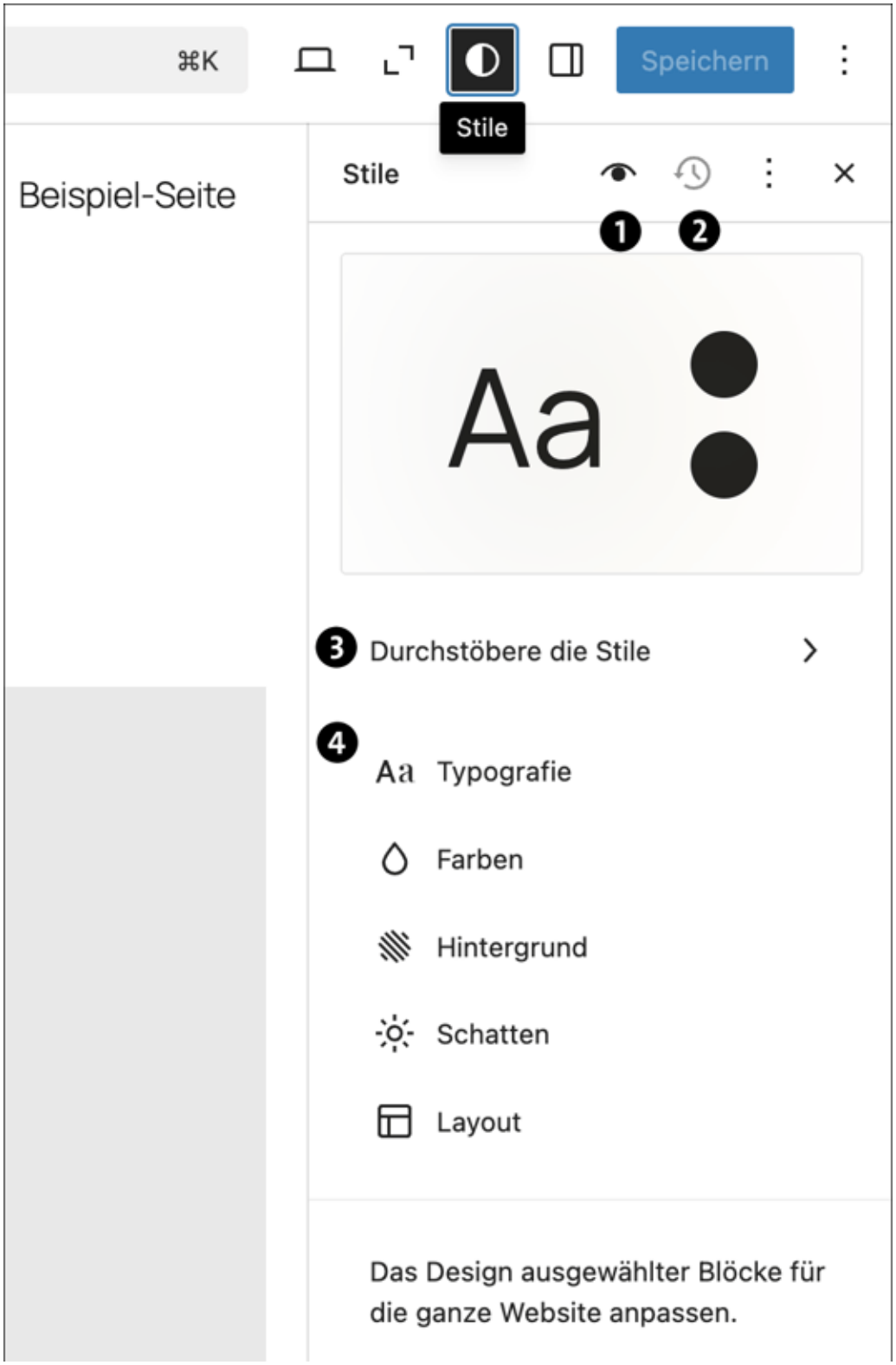


Abbildung 10.10 Das Stilbuch zeigt Ihnen alle Blöcke im aktuellen Design an.


⌘K
Speichern
Stile
Beispiel-Seite
Stile
1
2
Aa
3
Durchstöbere die Stile
4
Aa
Typografie
Farben
Hintergrund
Schatten
Layout
Das Design ausgewählter Blöcke für die ganze Website anpassen.

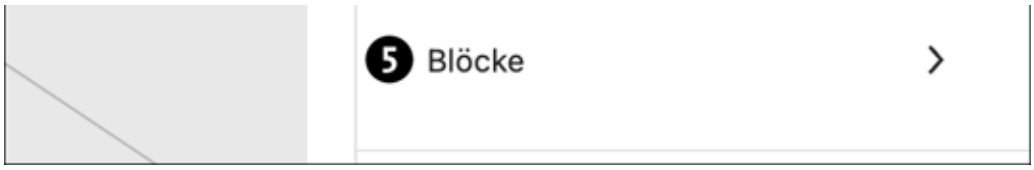


Abbildung 10.11 Die »Stile«-Seitenleiste erlaubt Ihnen die Anpassung des Designs vom Theme, wie beispielsweise Farben oder Schriften.

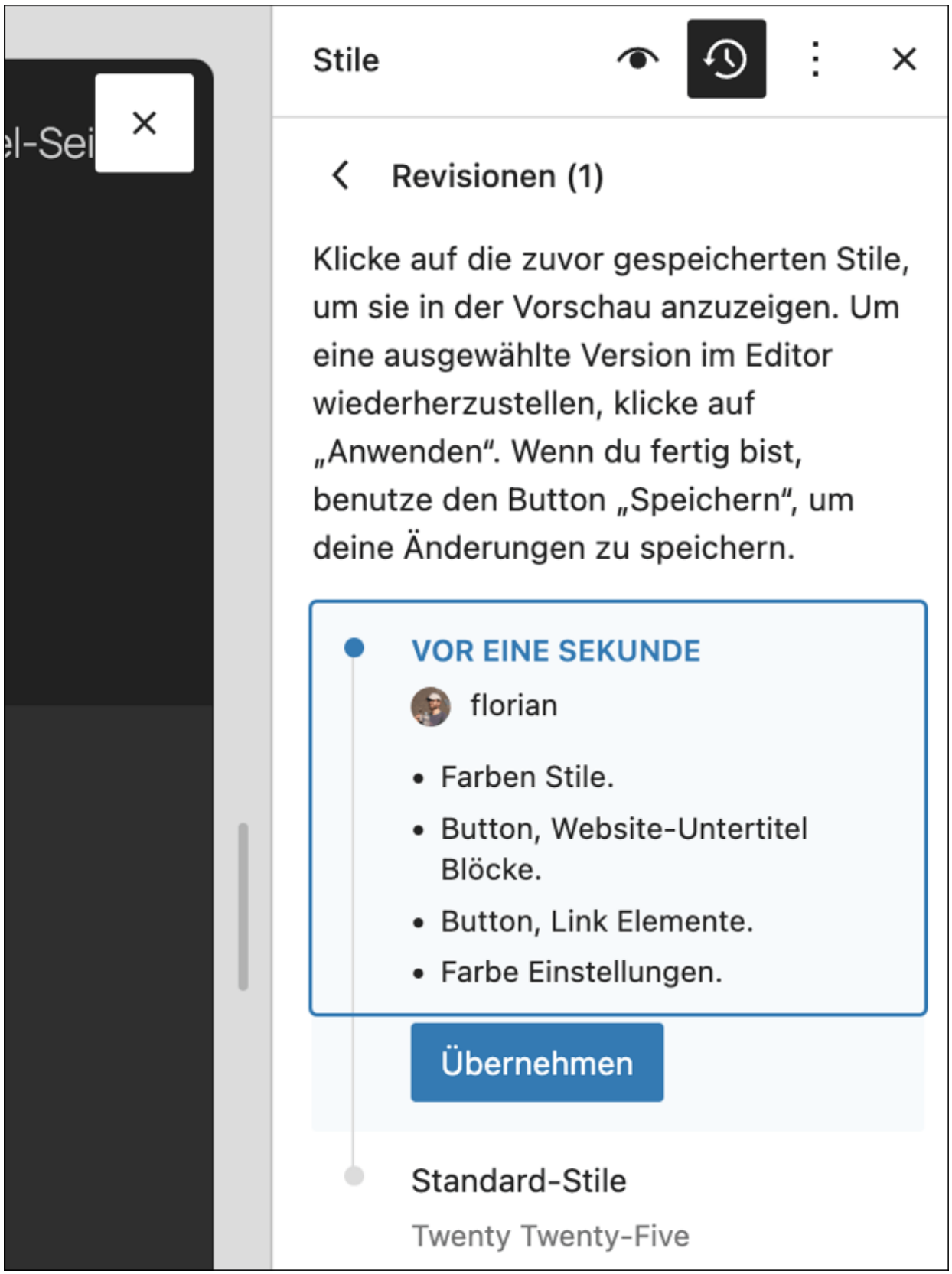

Stile
Revisionen (1)
Klicke auf die zuvor gespeicherten Stile, um sie in der Vorschau anzuzeigen. Um eine ausgewählte Version im Editor wiederherzustellen, klicke auf „Anwenden“. Wenn du fertig bist, benutze den Button „Speichern“, um deine Änderungen zu speichern.
VOR EINE SEKUNDE
florian
Farben Stile.
Button, Website-Untertitel Blöcke.
Button, Link Elemente.
Farbe Einstellungen.
Übernehmen
Standard-Stile
Twenty Twenty-Five

Abbildung 10.12 Nicht nur Beiträge und Seiten haben Revisionen, sondern auch die Anpassungen im Website-Editor.

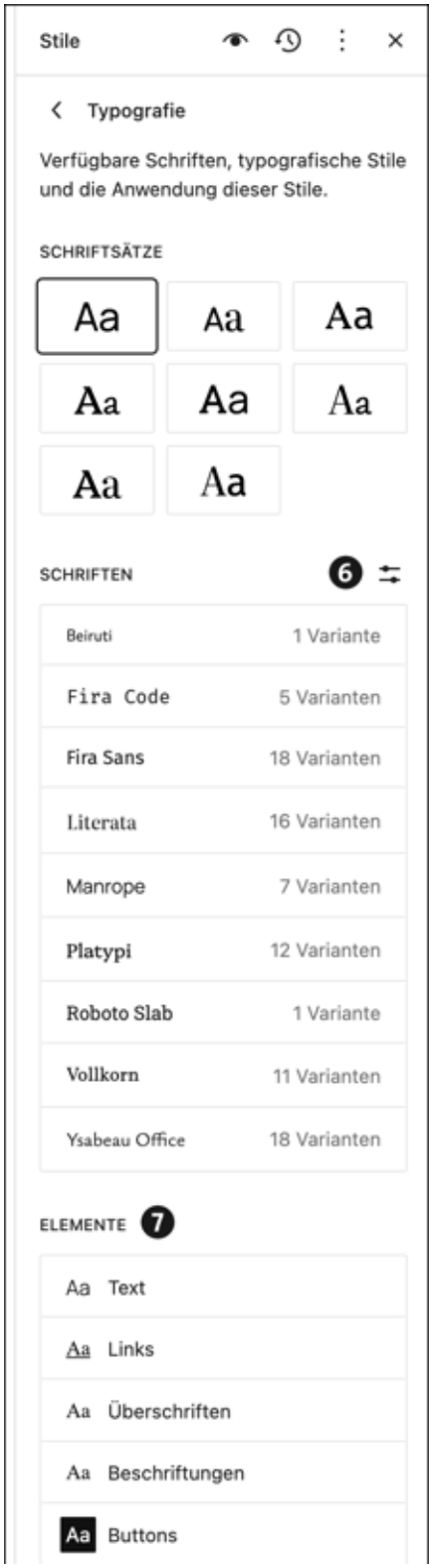

Stile
Typografie
Verfügbare Schriften, typografische Stile und die Anwendung dieser Stile.
SCHRIFTSÄTZE
Aa
Aa
Aa
Aa
Aa
Aa
Aa
Aa
SCHRIFTEN
6
Beiruti 1 Variante
Fira Code 5 Varianten
Fira Sans 18 Varianten
Literata 16 Varianten
Manrope 7 Varianten
Platypi 12 Varianten
Roboto Slab 1 Variante
Vollkorn 11 Varianten
Ysabeau Office 18 Varianten
ELEMENTE
7
Aa Text
Aa Links
Aa Überschriften
Aa Beschriftungen
Aa Buttons



Abbildung 10.13 Im Bereich »Typografie« der »Stile«-Seitenleiste können Sie die verfügbaren Schriften verwalten und Typografie-Optionen für bestimmte Elemente anpassen.

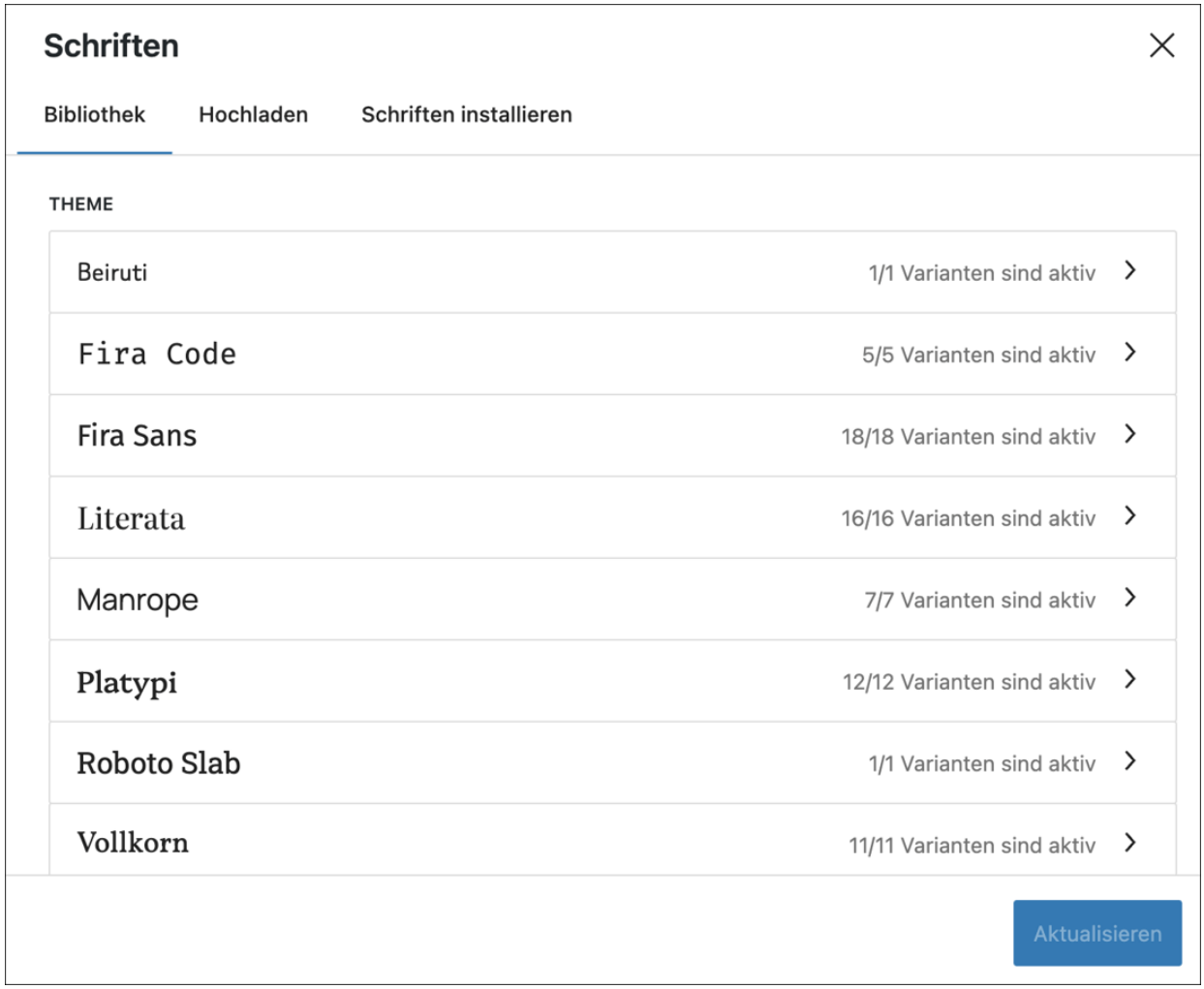
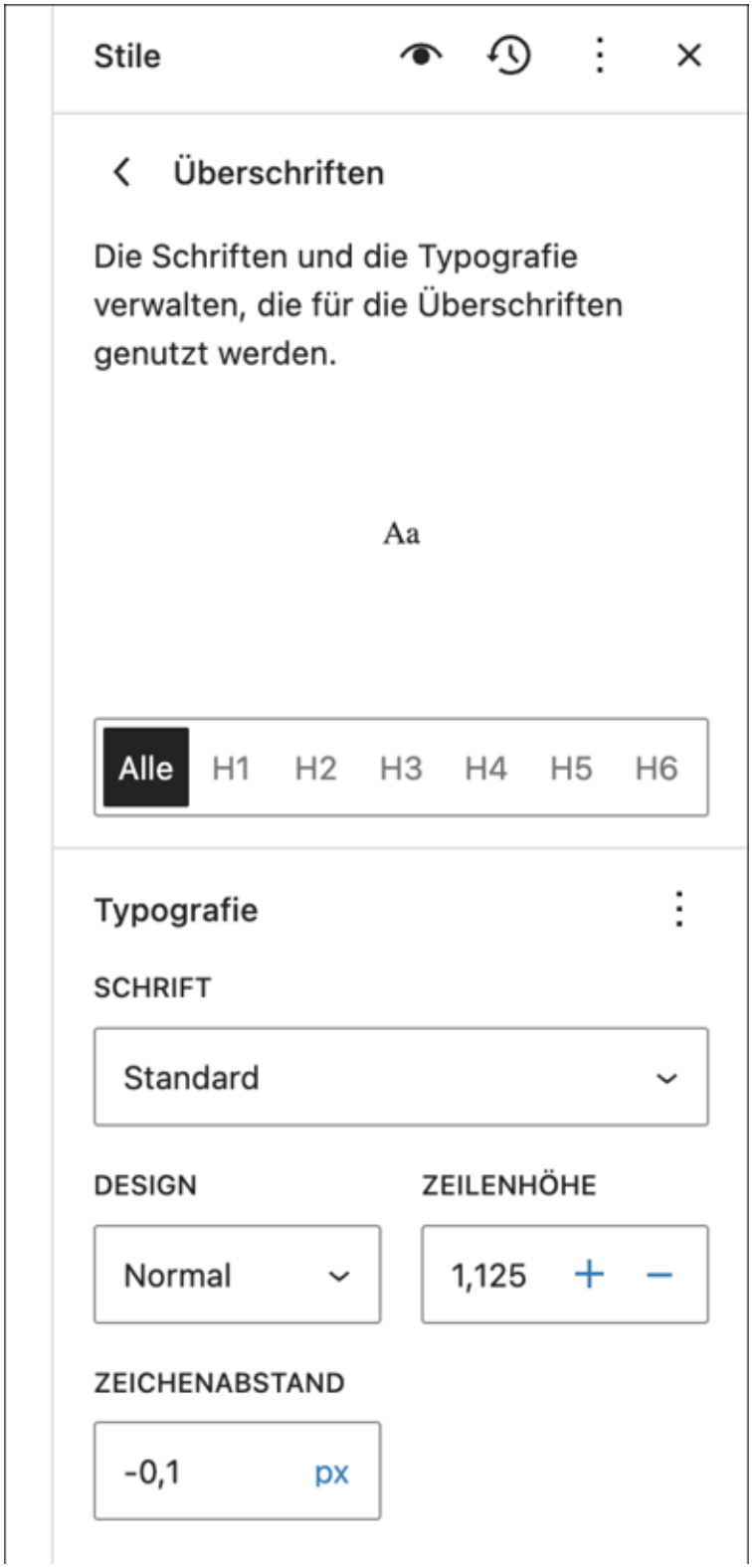


Abbildung 10.14 In der Schriften-Bibliothek sehen Sie, welche Schriften mit wie vielen Varianten installiert sind, und können Varianten deaktivieren oder aktivieren. Außerdem haben Sie über die Tabs »Hochladen« und »Schriften installieren« die Möglichkeit, zusätzliche Schriften hinzuzufügen.


Stile
Überschriften
Die Schriften und die Typografie verwalten, die für die Überschriften genutzt werden.
Aa
Alle
H1
H2
H3
H4
H5
H6
Typografie
SCHRIFT
Standard
DESIGN
Normal
ZEILENHÖHE
1,125
ZEICHENABSTAND
-0,1
px

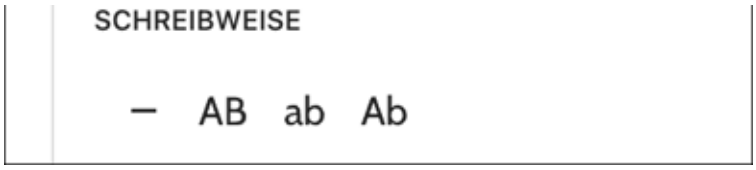


Abbildung 10.15 In den Typografie-Einstellungen können Sie für bestimmte Elemente wie beispielsweise Überschriften Dinge wie die Schriftart, Schriftgröße und Zeilenhöhe anpassen.

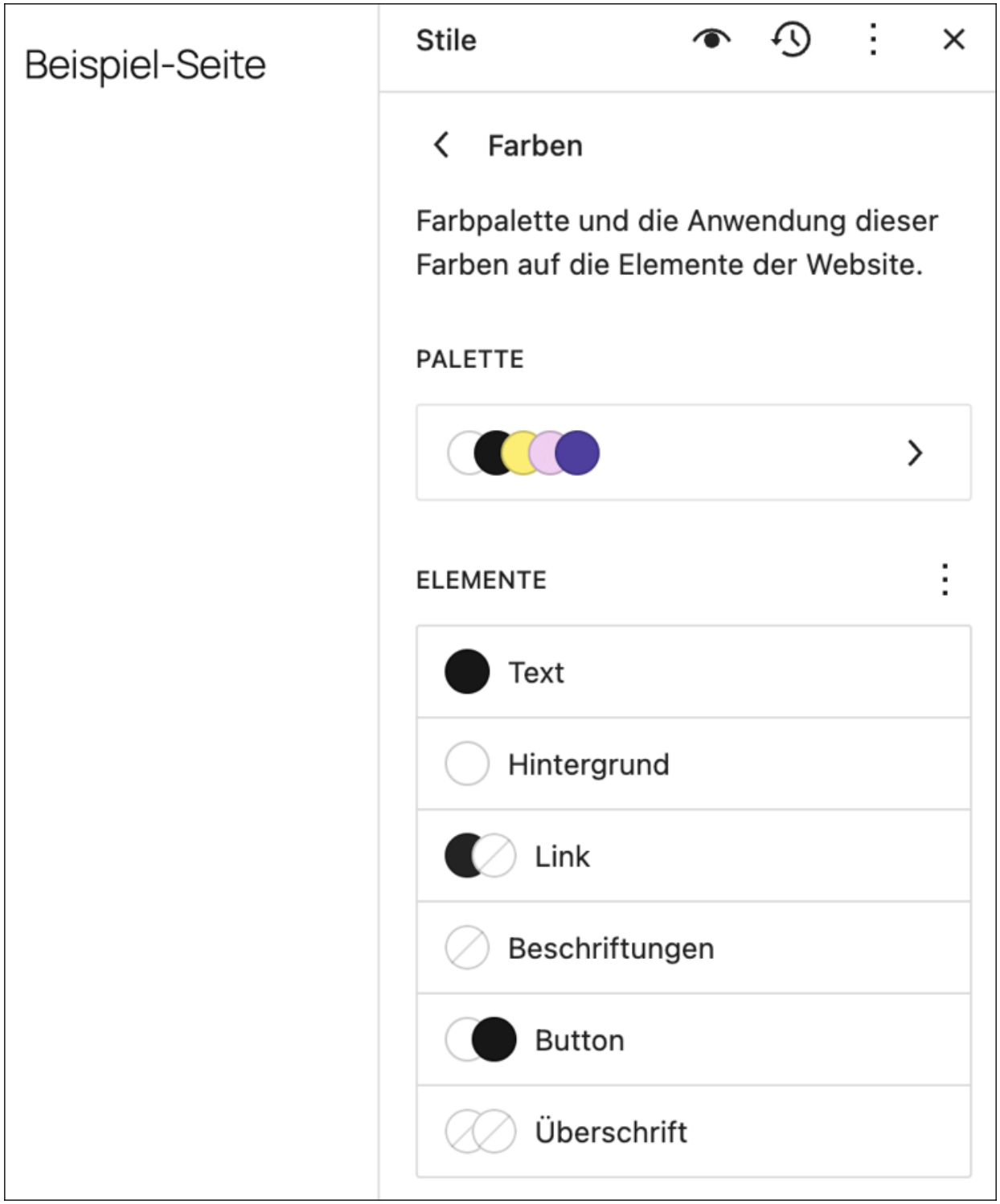
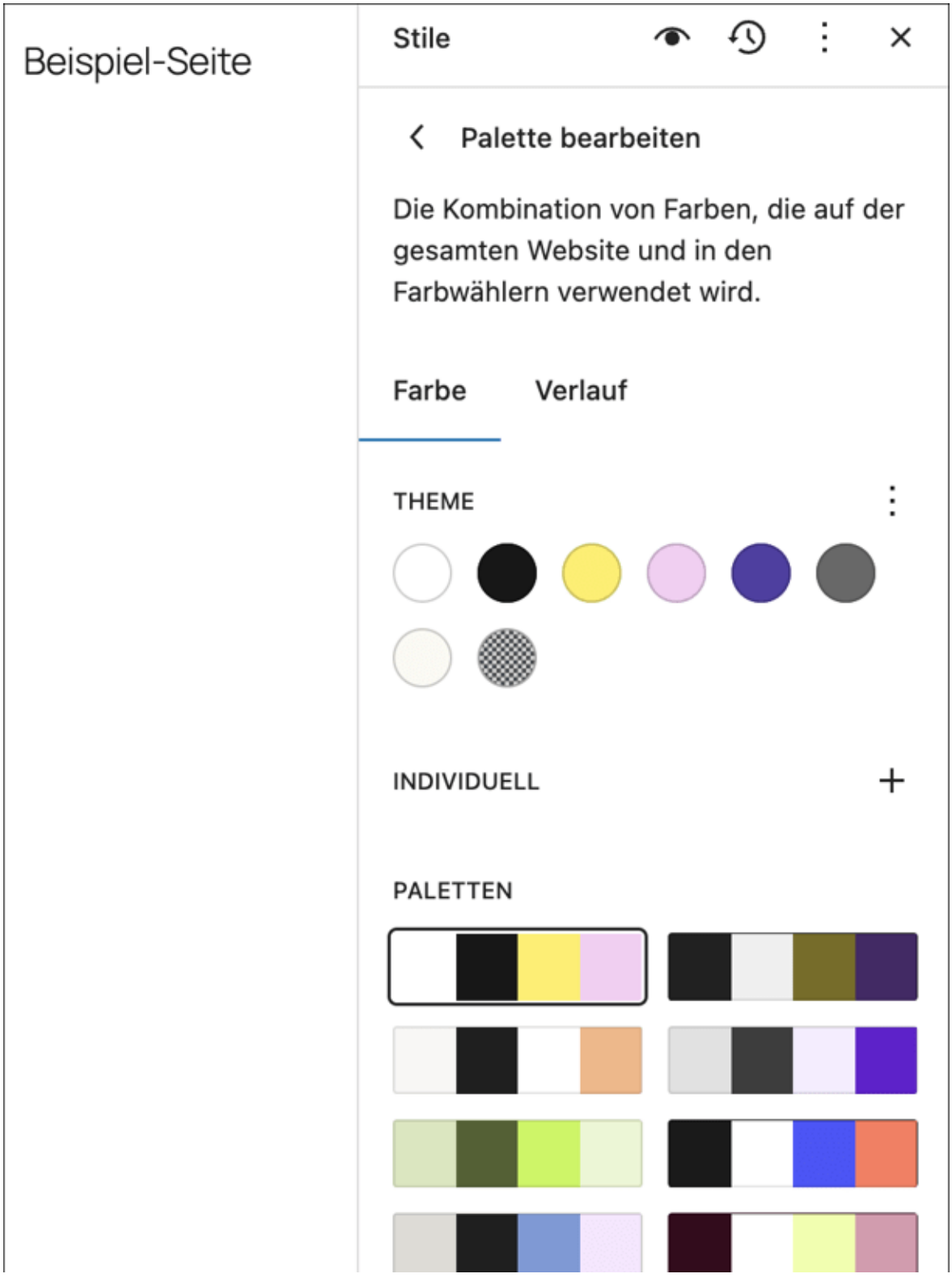

Beispiel-Seite
Stile
Farben
Farbpalette und die Anwendung dieser Farben auf die Elemente der Website.
PALETTE
ELEMENTE
Text
Hintergrund
Link
Beschriftungen
Button
Überschrift

Abbildung 10.16 Im »Farben«-Bereich können Sie die Farbpalette des Themes anpassen sowie Farbeinstellungen für verschiedene Elemente global setzen.


Beispiel-Seite
Stile
Palette bearbeiten
Die Kombination von Farben, die auf der gesamten Website und in den Farbwählern verwendet wird.
Farbe
Verlauf
THEME
INDIVIDUELL
PALETTEN

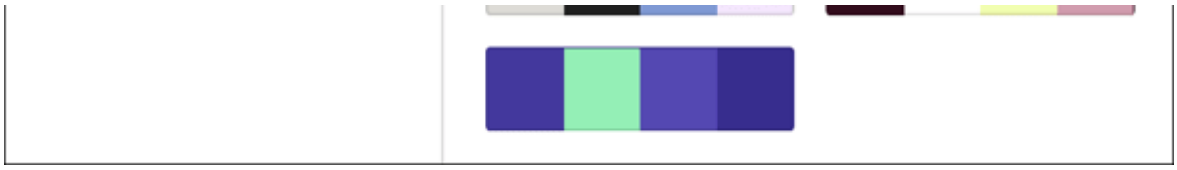


Abbildung 10.17 Im »Farbe«-Tab können Sie die Farbpaletten des Themes anpassen, im »Verlauf«-Tab können Sie die Verläufe und Duoton-Effekte verändern.

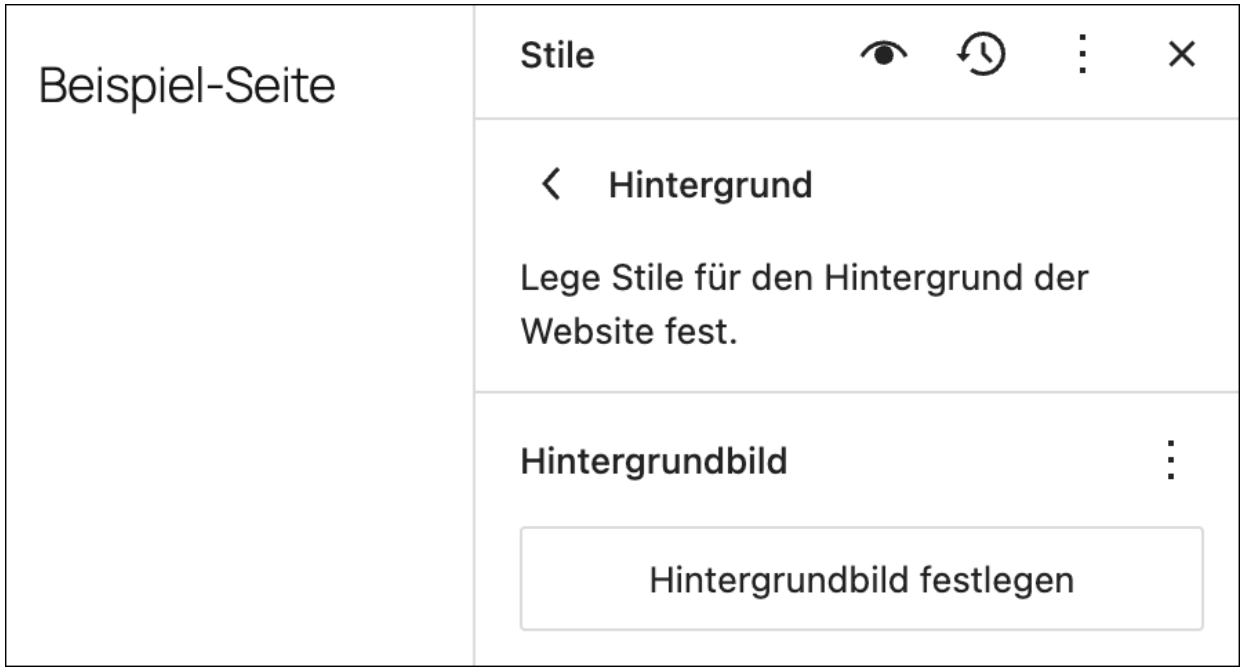


Abbildung 10.18 Unter dem Eintrag »Hintergrund« haben Sie die Möglichkeit, ein Hintergrundbild für die Website festzulegen.

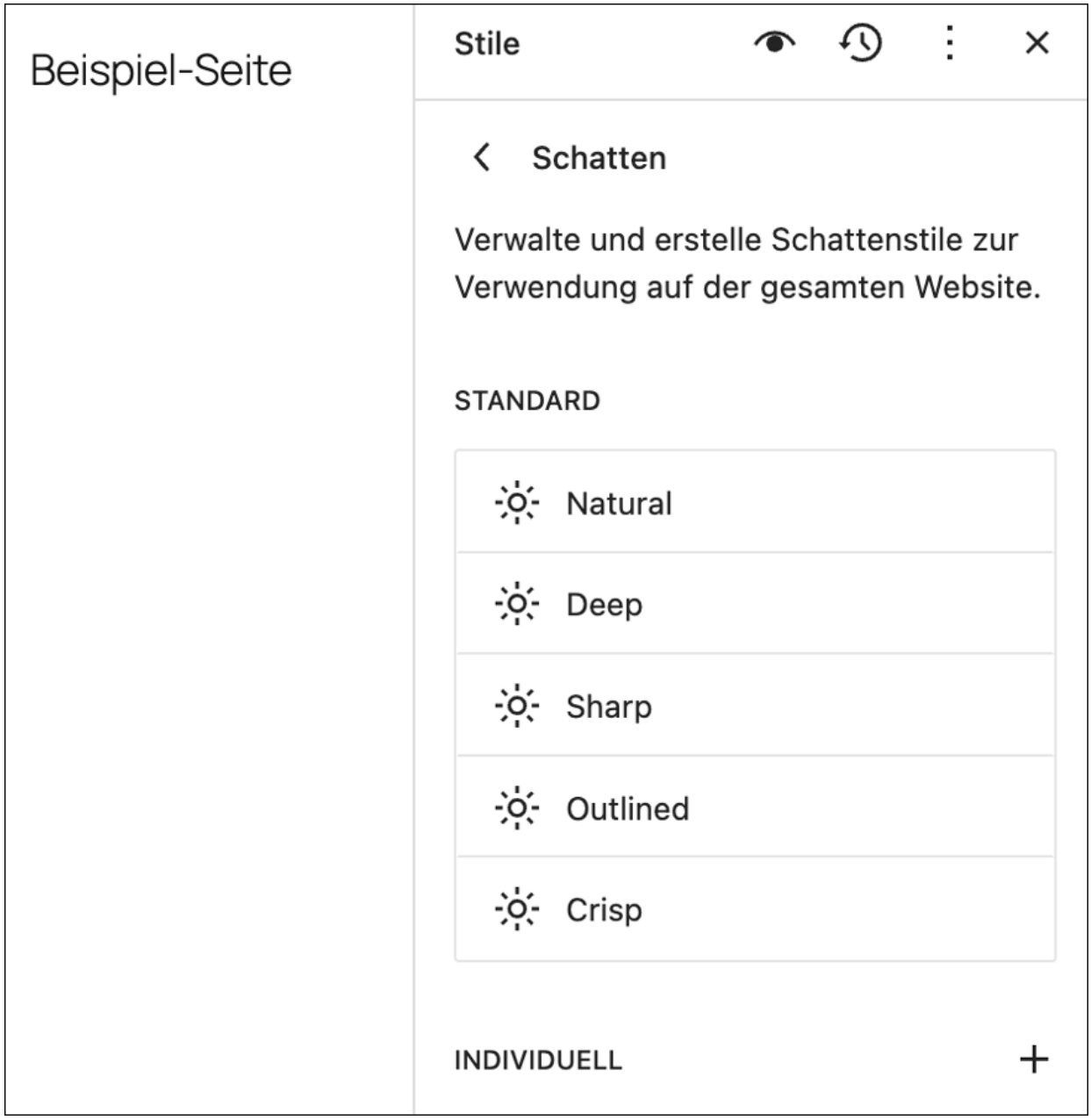


Abbildung 10.19 Blöcken wie dem Bild-Block können Sie einen Schatten-Effekt zuweisen. In der »Stile«-Seitenleiste können Sie die vorhandenen Schatten bearbeiten und individuelle Schatten hinzufügen.

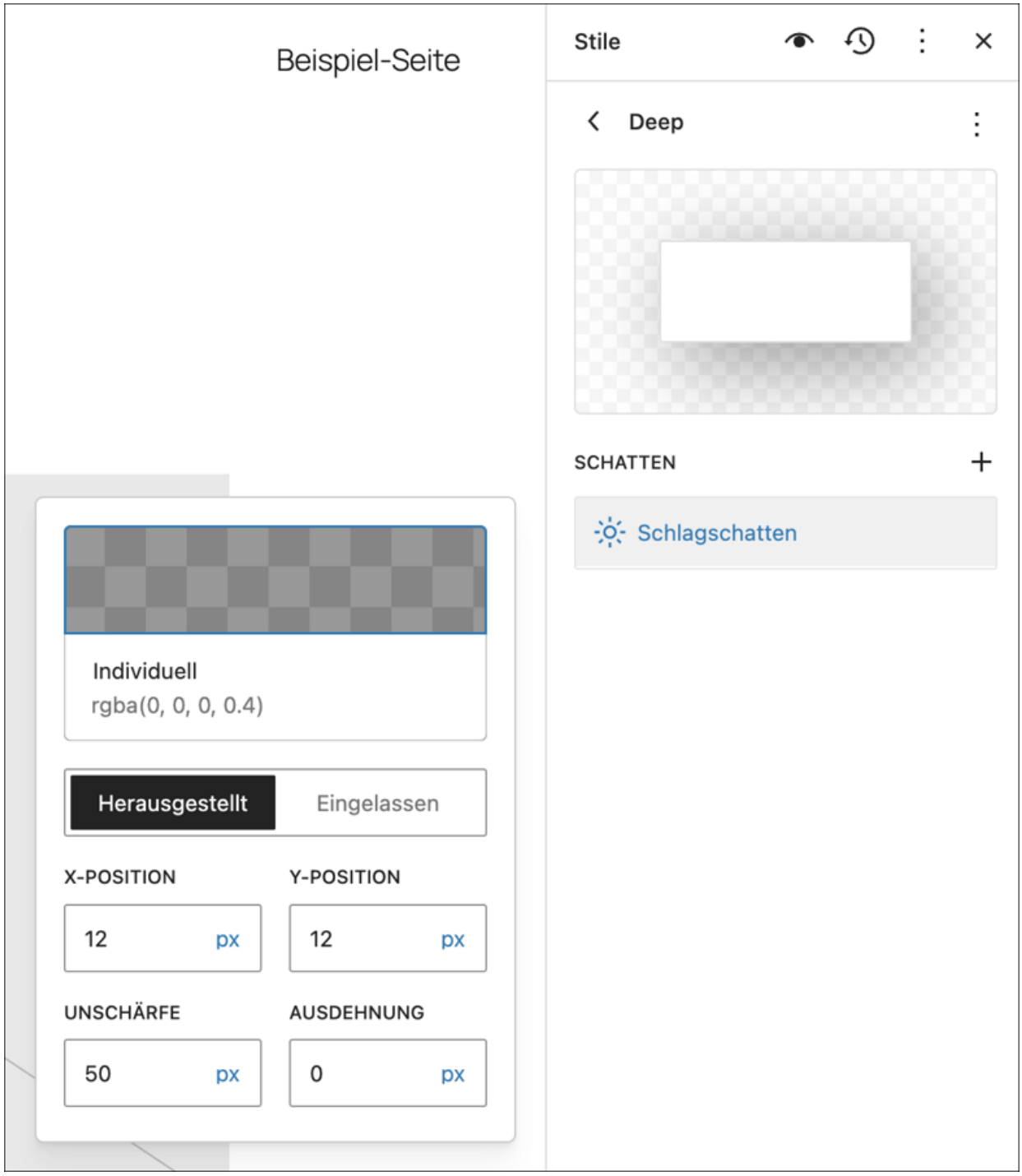

Beispiel-Seite
Stile
Deep
SCHATTEN
Schlagschatten
Individuell
rgba(0, 0, 0, 0.4)
Herausgestellt
Eingelassen
X-POSITION
Y-POSITION
12
px
12
px
UNSCHÄRFE
AUSDEHNUNG
50
px
0
px

Abbildung 10.20 In der Bearbeitungsansicht eines Schattens können Sie alle Werte verändern, die sich auch über CSS setzen lassen würden.

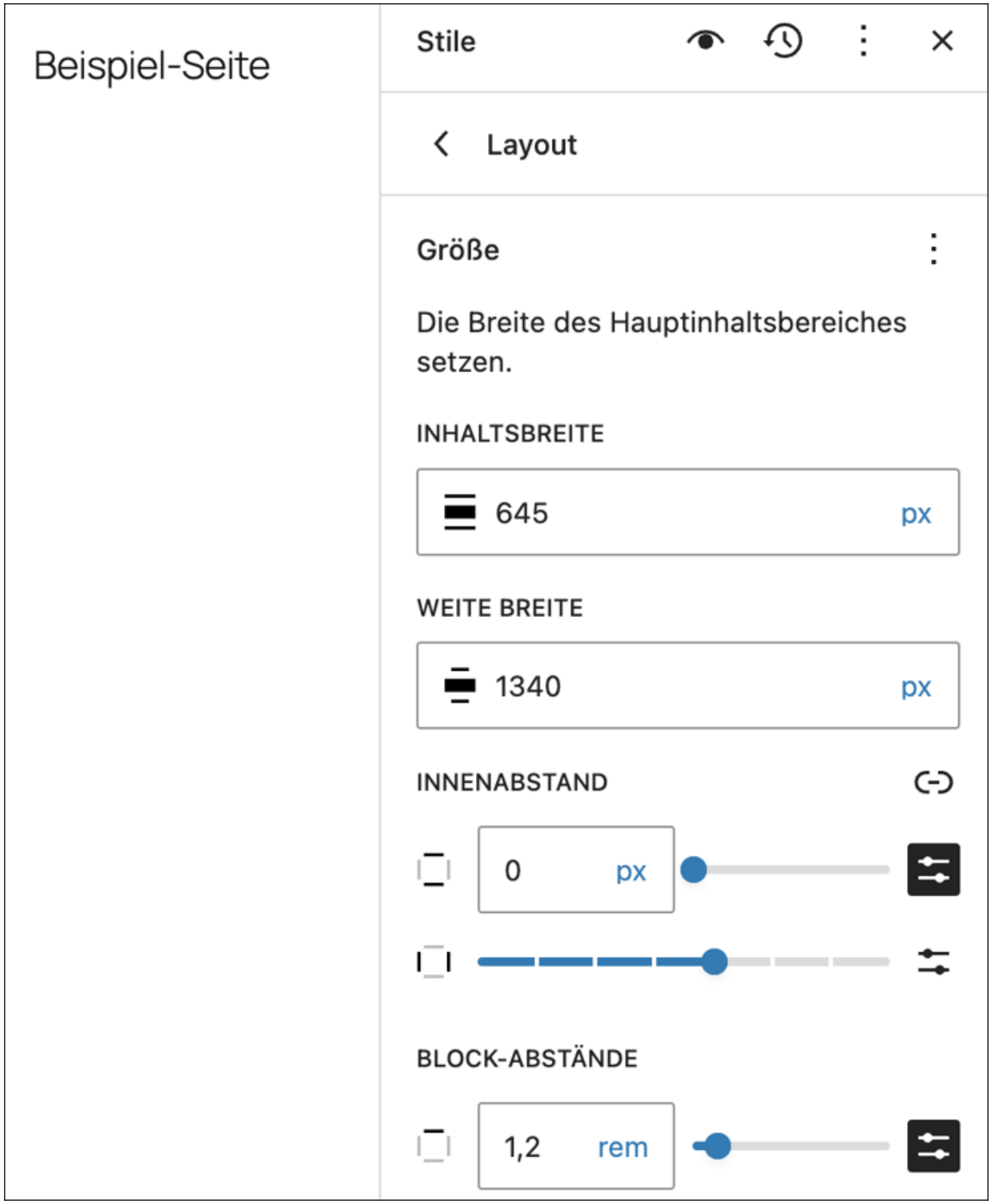

Beispiel-Seite
Stile
Layout
Größe
Die Breite des Hauptinhaltsbereiches setzen.
INHALTSBREITE
645
px
WEITE BREITE
1340
px
INNENABSTAND
0
px
BLOCK-ABSTÄNDE
1,2
rem

Abbildung 10.21 Im Bereich »Layout« lässt sich unter anderem die Inhaltsbreite anpassen.

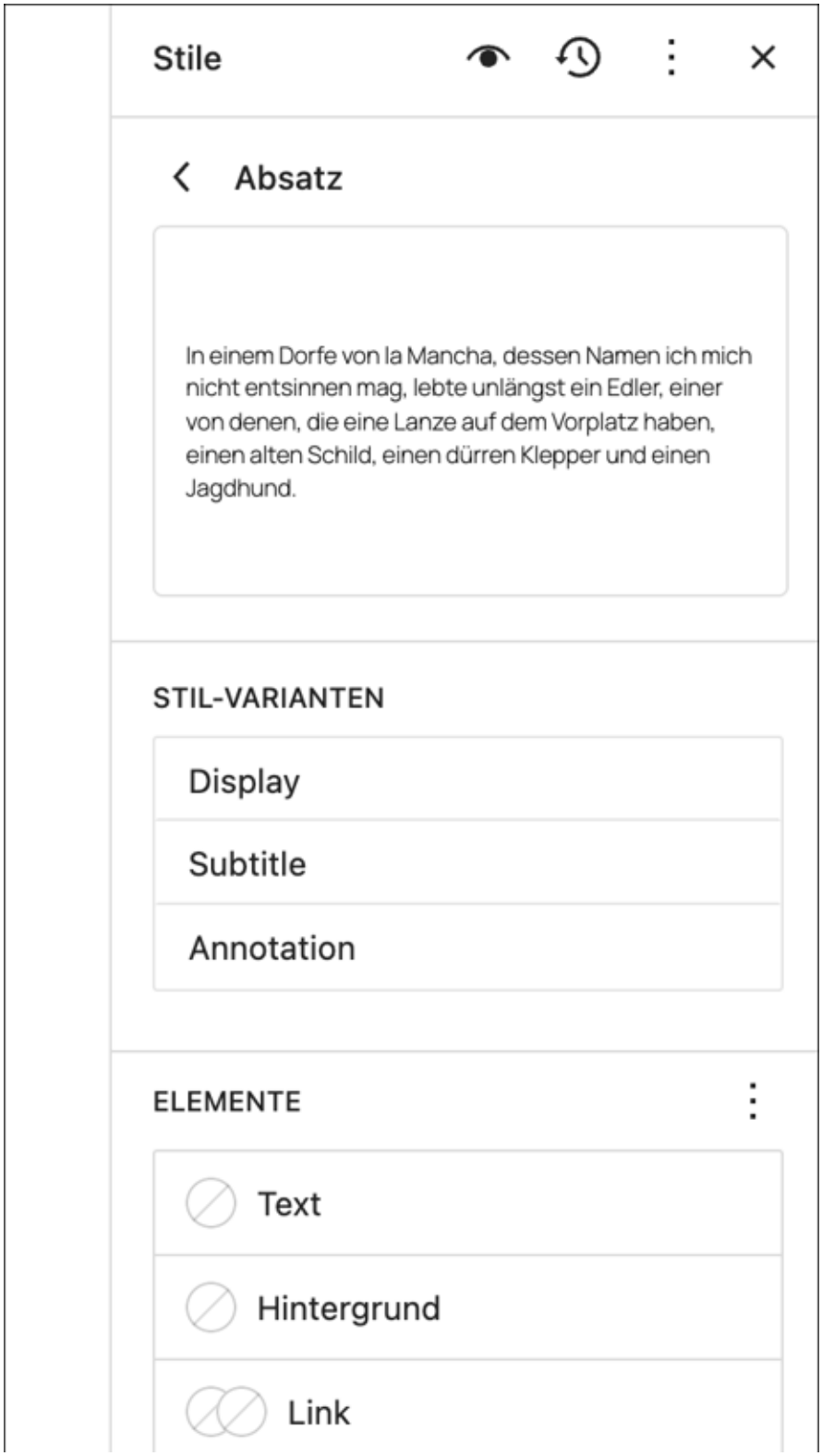

Stile
Absatz
In einem Dorfe von la Mancha, dessen Namen ich mich nicht entsinnen mag, lebte unlängst ein Edler, einer von denen, die eine Lanze auf dem Vorplatz haben, einen alten Schild, einen dürren Klepper und einen Jagdhund.
STIL-VARIANTEN
Display
Subtitle
Annotation
ELEMENTE
Text
Hintergrund
Link



Abbildung 10.22 In der »Blöcke«-Liste können Sie Stile für alle verfügbaren Blöcke anpassen. Beim Absatz-Block haben Sie beispielsweise Einstellungsmöglichkeiten für Farbe, Typografie und Abstände.



Abbildung 10.23 Das Interface zum Bearbeiten der Blöcke, aus denen Ihre Website besteht, gleicht dem des Editors für die Inhalte Ihrer Beiträge und Seiten.

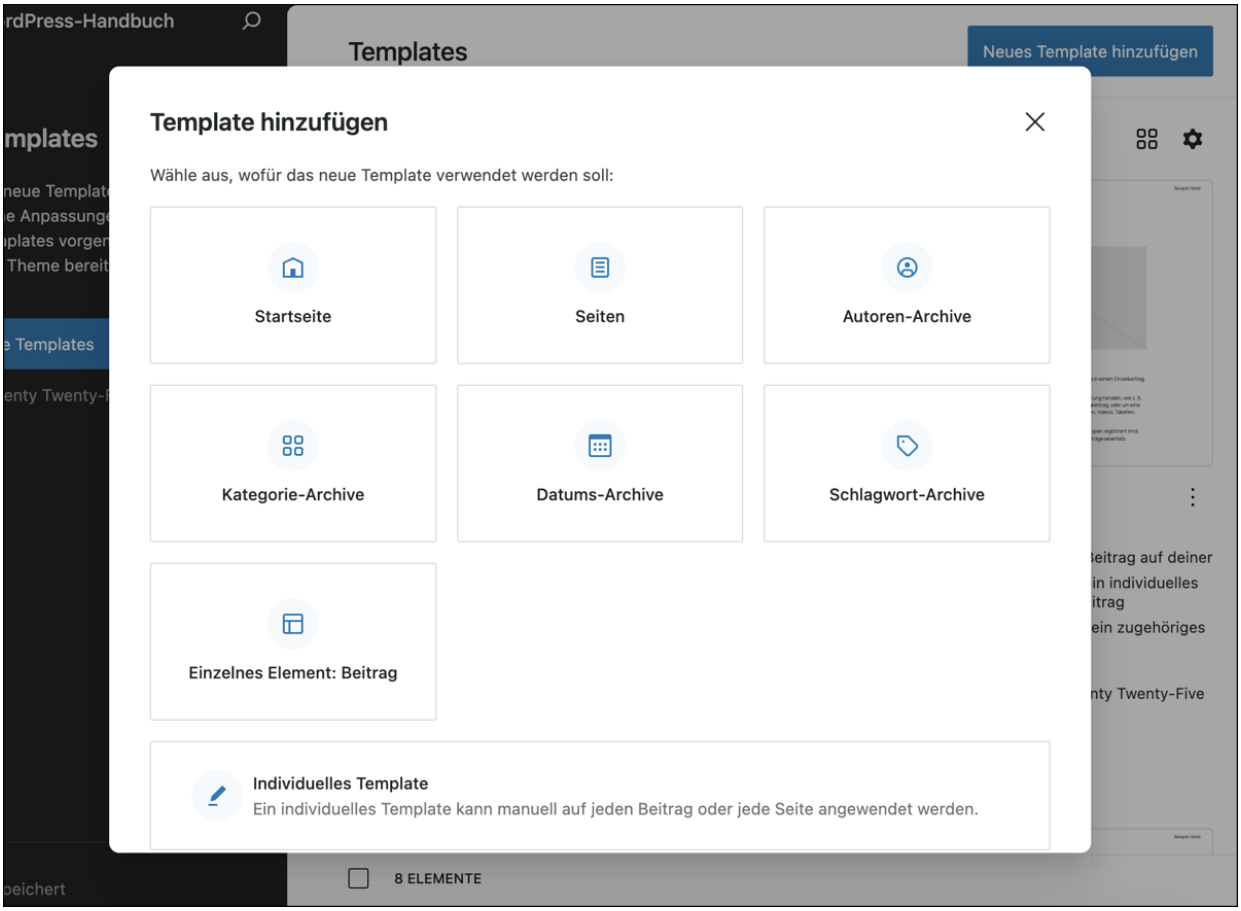


Abbildung 10.24 Wenn Sie sich in der Templates-Liste befinden, gibt es oben rechts einen Button zum Hinzufügen neuer Templates. Neben den Standard-WordPress-Templates, die im aktuellen Theme fehlen, können Sie auch ein »individuelles Template« erstellen, das Sie manuell für Beiträge und Seiten auswählen können.

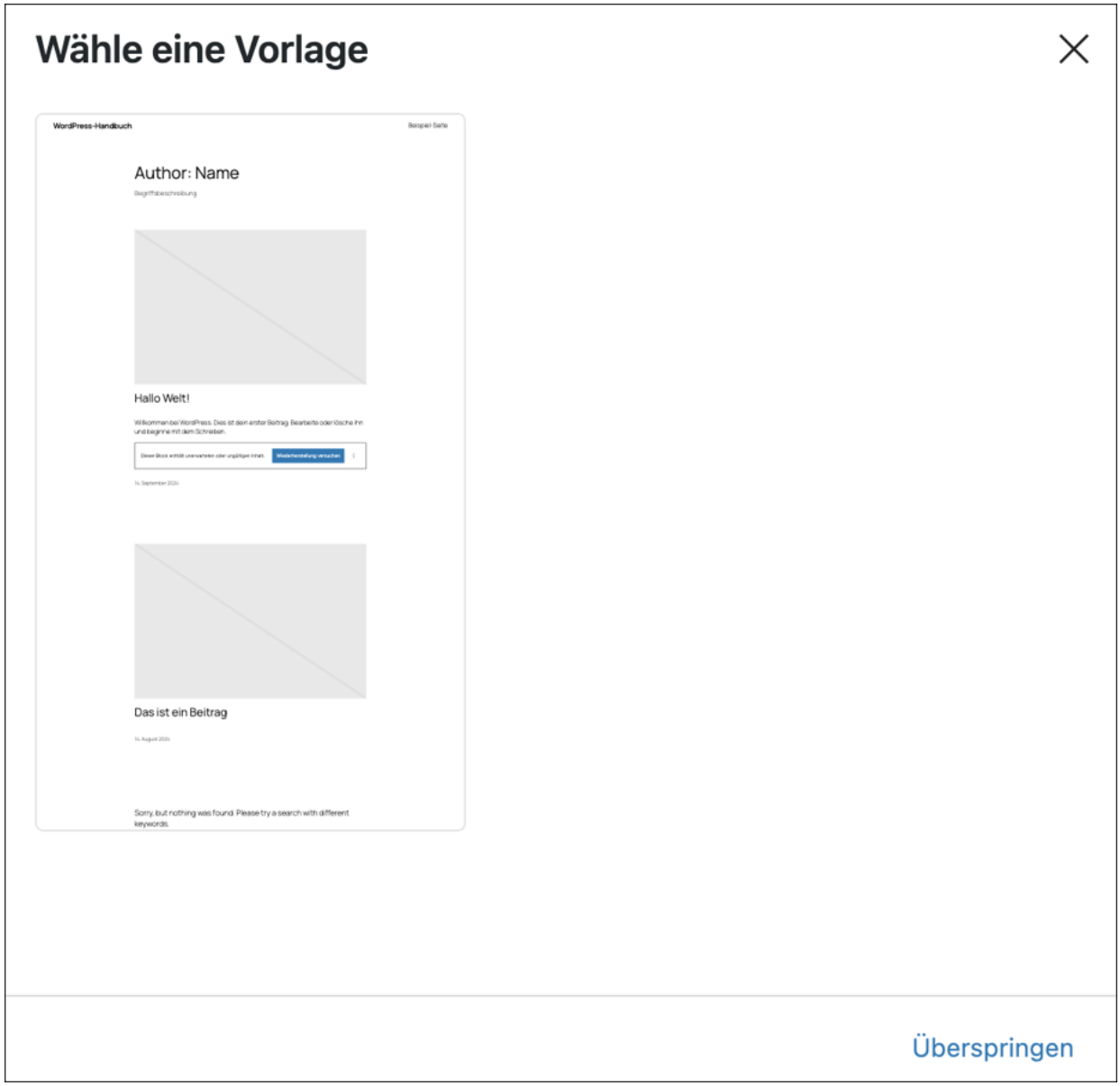


Abbildung 10.25 Bei der Erstellung eines Templates werden Ihnen gegebenenfalls Vorlagen angeboten, von denen Sie starten können.

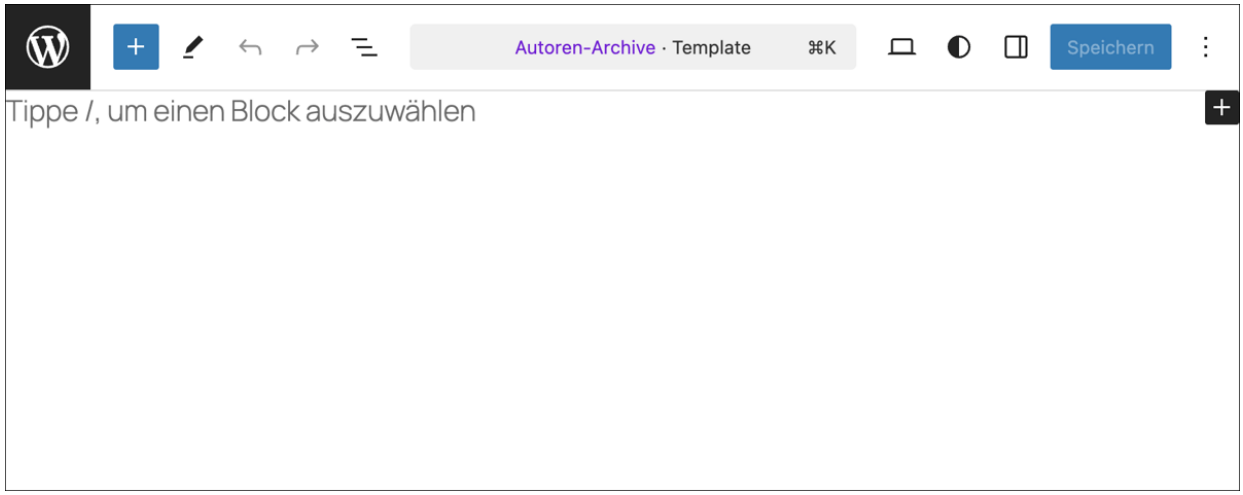
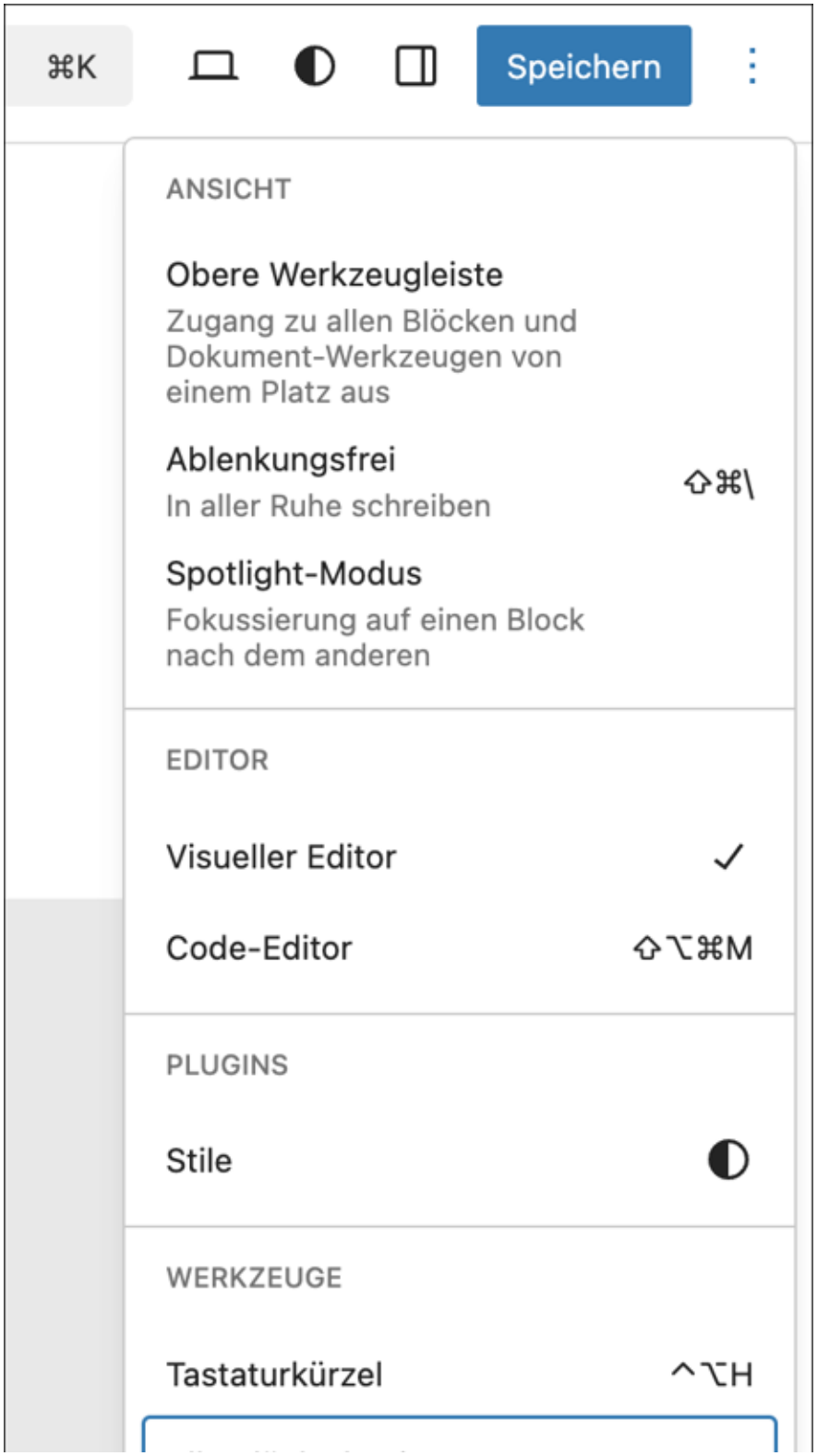


Abbildung 10.26 Bei einem neuen Template starten Sie von null, wenn Sie keine der Vorlagen auswählen. Aber auch wenn Sie keine Vorlage nutzen, können Sie sich per Copy-and-paste zumindest Dinge wie Header und Footer schnell aus anderen Templates rüberkopieren.


⌘K
Speichern
ANSICHT
Obere Werkzeugleiste
Zugang zu allen Blöcken und Dokument-Werkzeugen von einem Platz aus
Ablenkungsfrei
In aller Ruhe schreiben
⇧⌘\
Spotlight-Modus
Fokussierung auf einen Block nach dem anderen
EDITOR
Visueller Editor
Code-Editor
⇧⌥⌘M
PLUGINS
Stile
WERKZEUGE
Tastaturkürzel
^⌥H

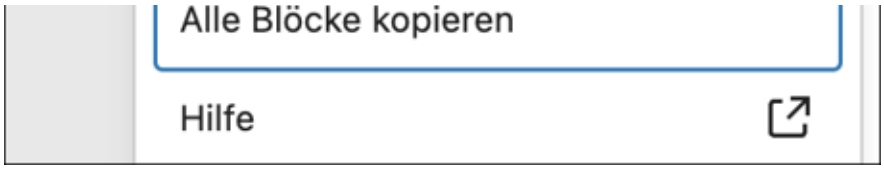


Abbildung 10.27 Die Blöcke aus einem bestehenden Template können gut als Ausgangslage für ein neues Template genutzt werden, um nicht bei null starten zu müssen, wenn es keine passende Vorlage gibt.



Abbildung 10.28 Das neue Template für Seiten von Autor*innen in Aktion

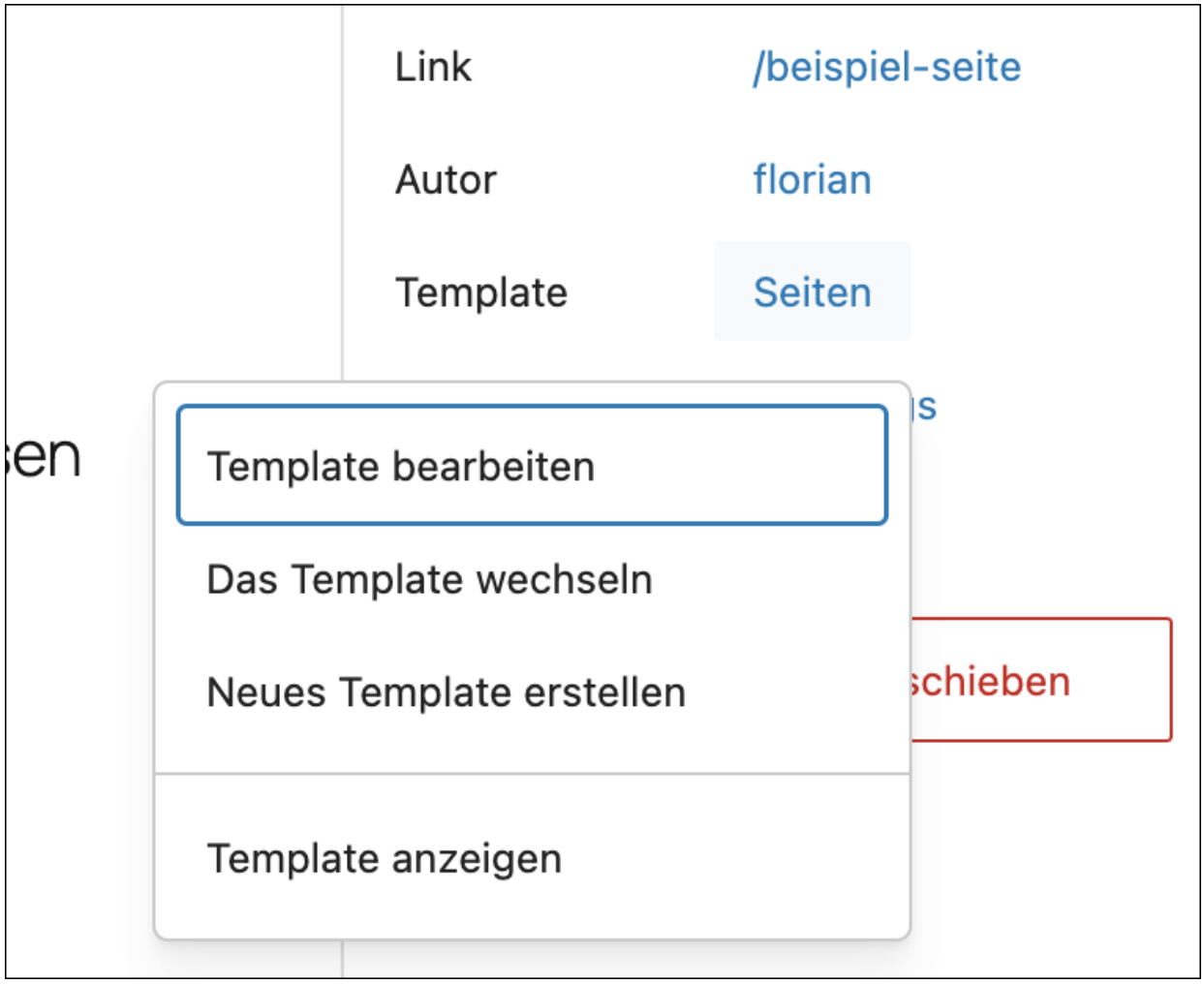


Abbildung 10.29 Durch einen Klick auf das aktuelle Template eines Beitrags oder einer Seite öffnet sich ein Overlay, das Ihnen einen Template-Wechsel ermöglicht oder Gelegenheit gibt, ein komplett neues Template zu erstellen.

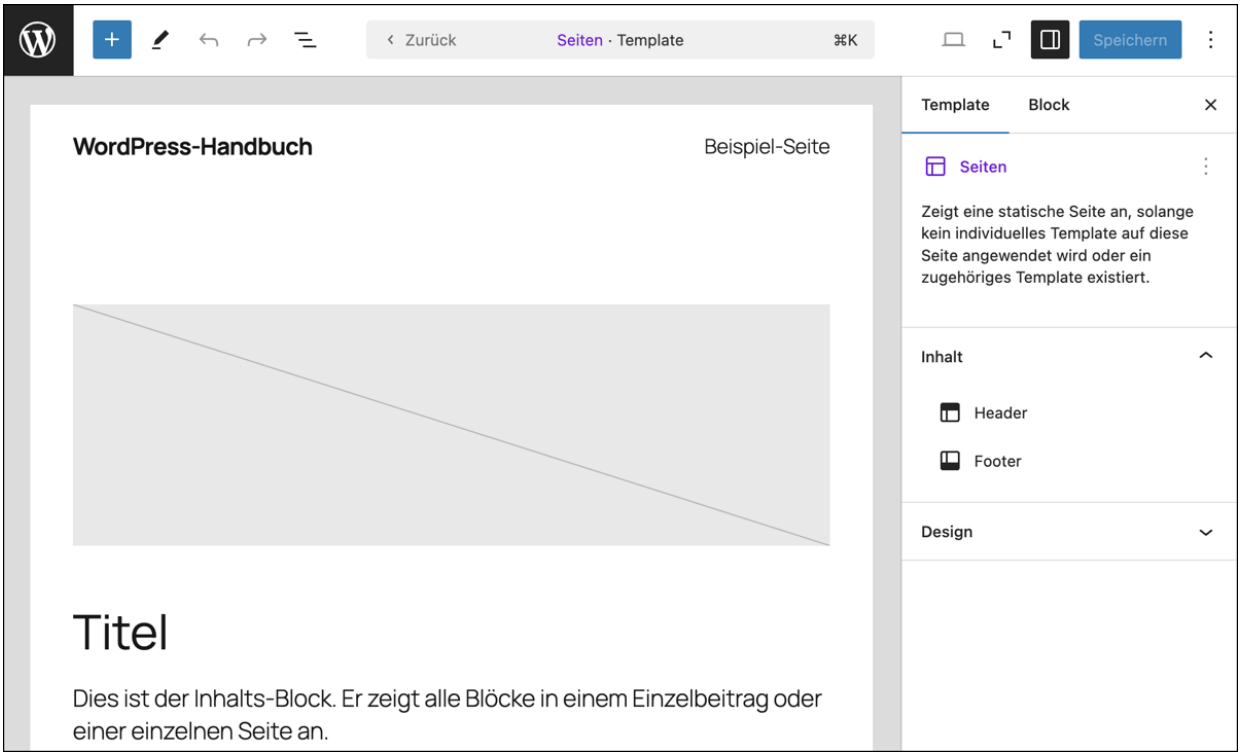


Abbildung 10.30 Mit dem Template-Editor können Sie Templates bearbeiten, die für Seiten und Beiträge zum Einsatz kommen. Er bietet nicht alle Funktionen, die der Site-Editor mitbringt, aber die Bearbeitungsansicht für die Blöcke des Templates ist gleich.

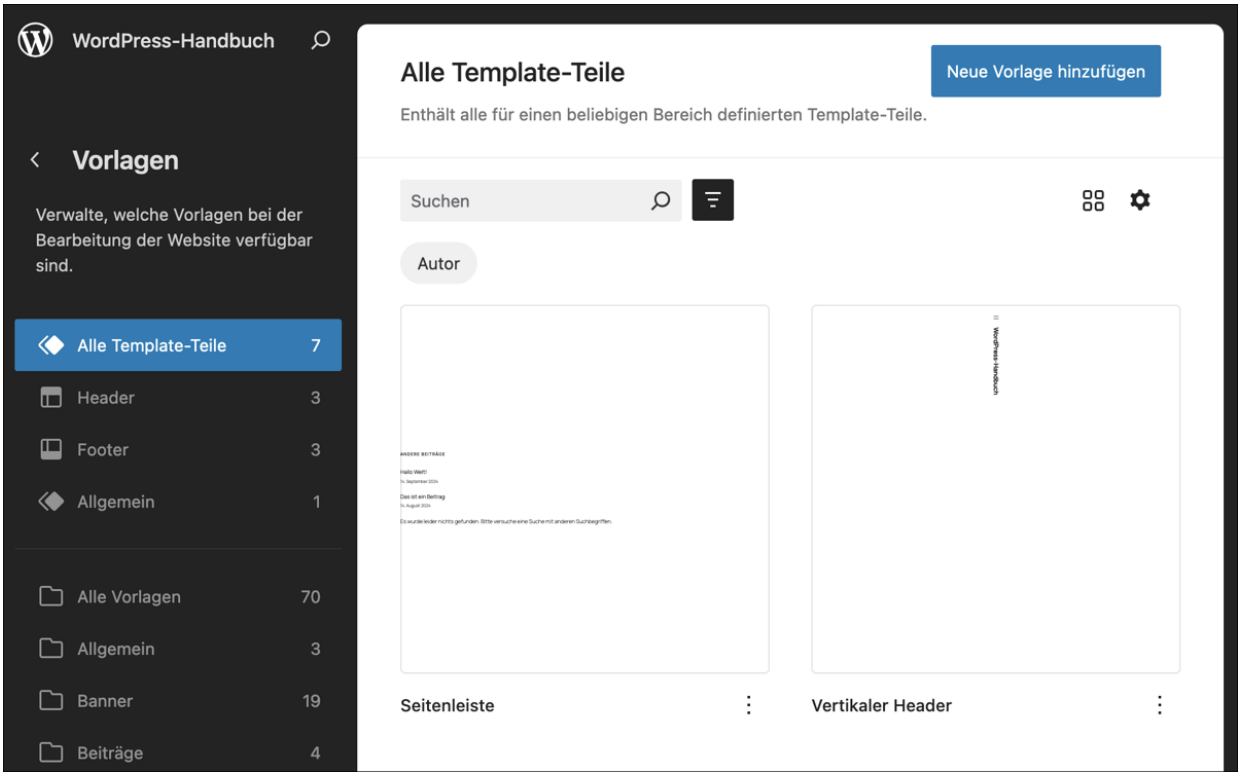


Abbildung 10.31 Über »Vorlagen« • »Alle Template-Teile« erreichen Sie eine Raster-Ansicht der Template-Teile, die vom Theme mitgebracht oder von Ihnen erstellt wurden.

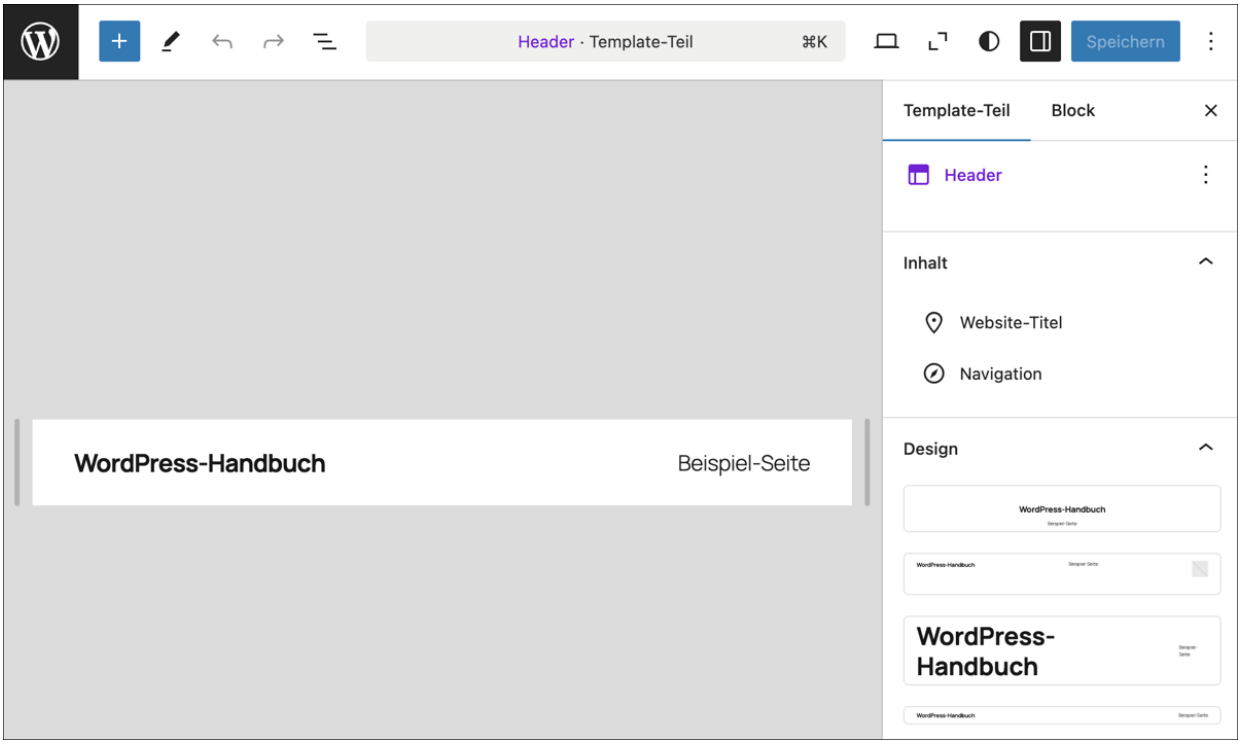


Abbildung 10.32 Für die Bearbeitung eines Templates-Teils öffnet sich wieder der Website-Editor, diesmal aber nur mit dem Template-Teil ohne den Rest einer kompletten Webseite drum herum.

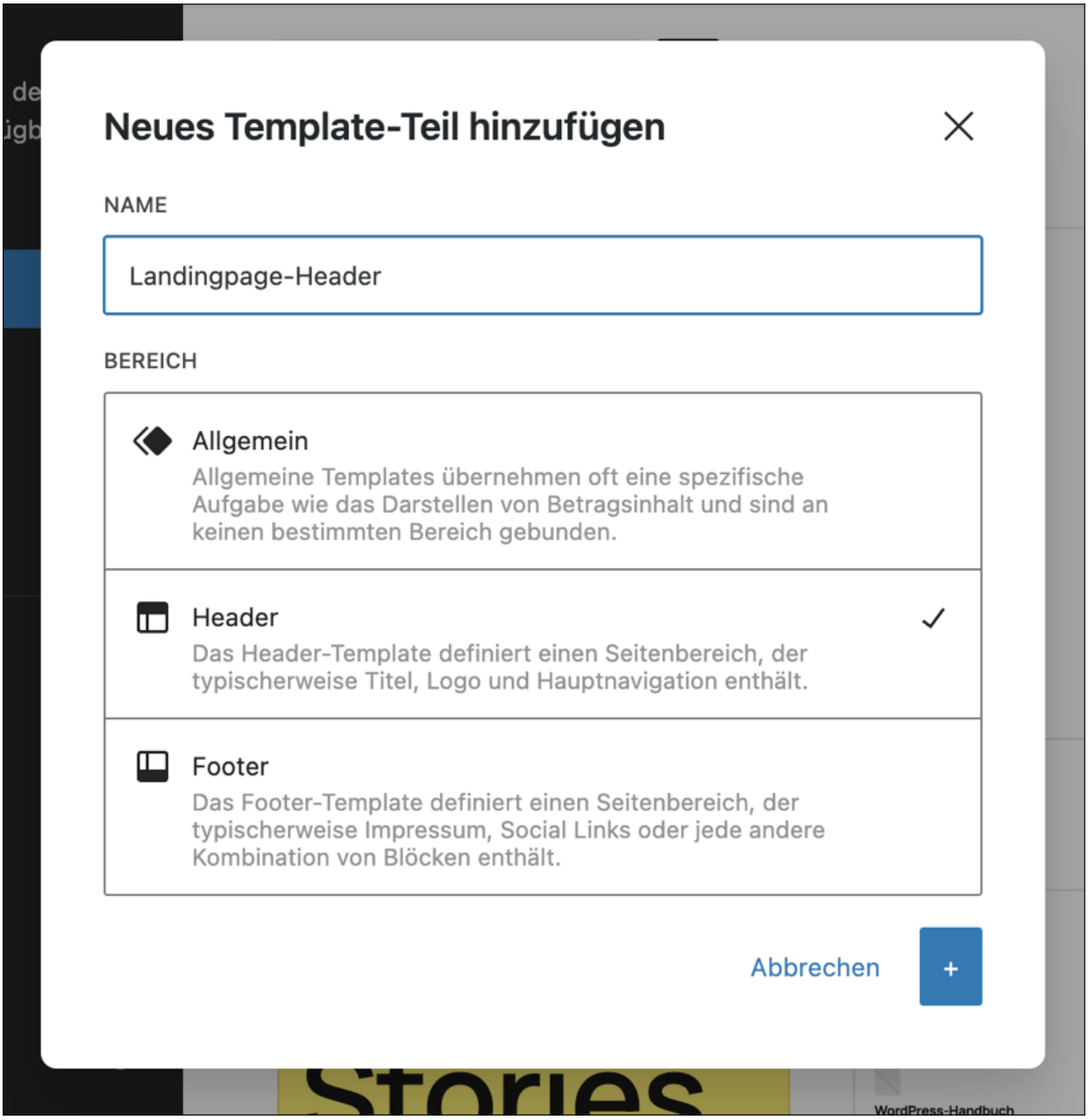


Abbildung 10.33 Beim Hinzufügen eines Template-Teils werden zuerst Name und Bereich festgelegt.

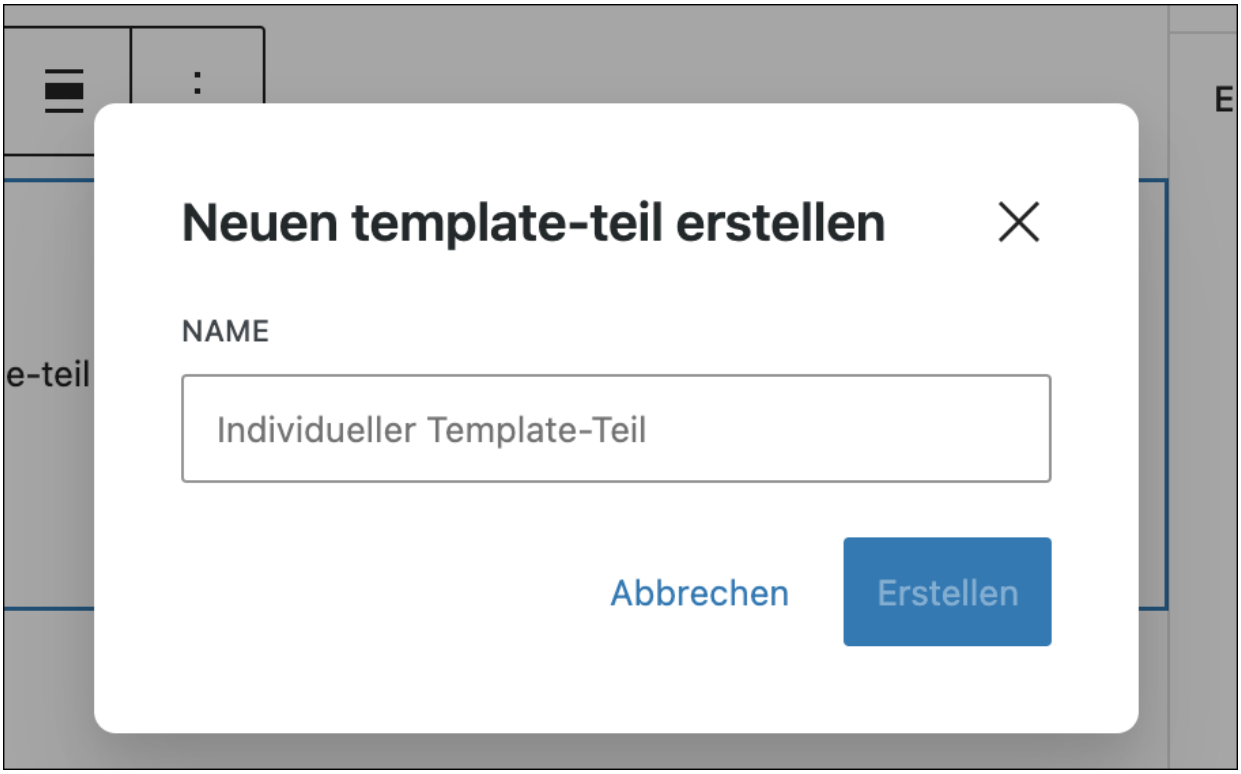


Abbildung 10.34 Template-Teile können auch direkt aus den Blöcken heraus erstellt werden, mit denen bestehende Template-Teile angezeigt werden.

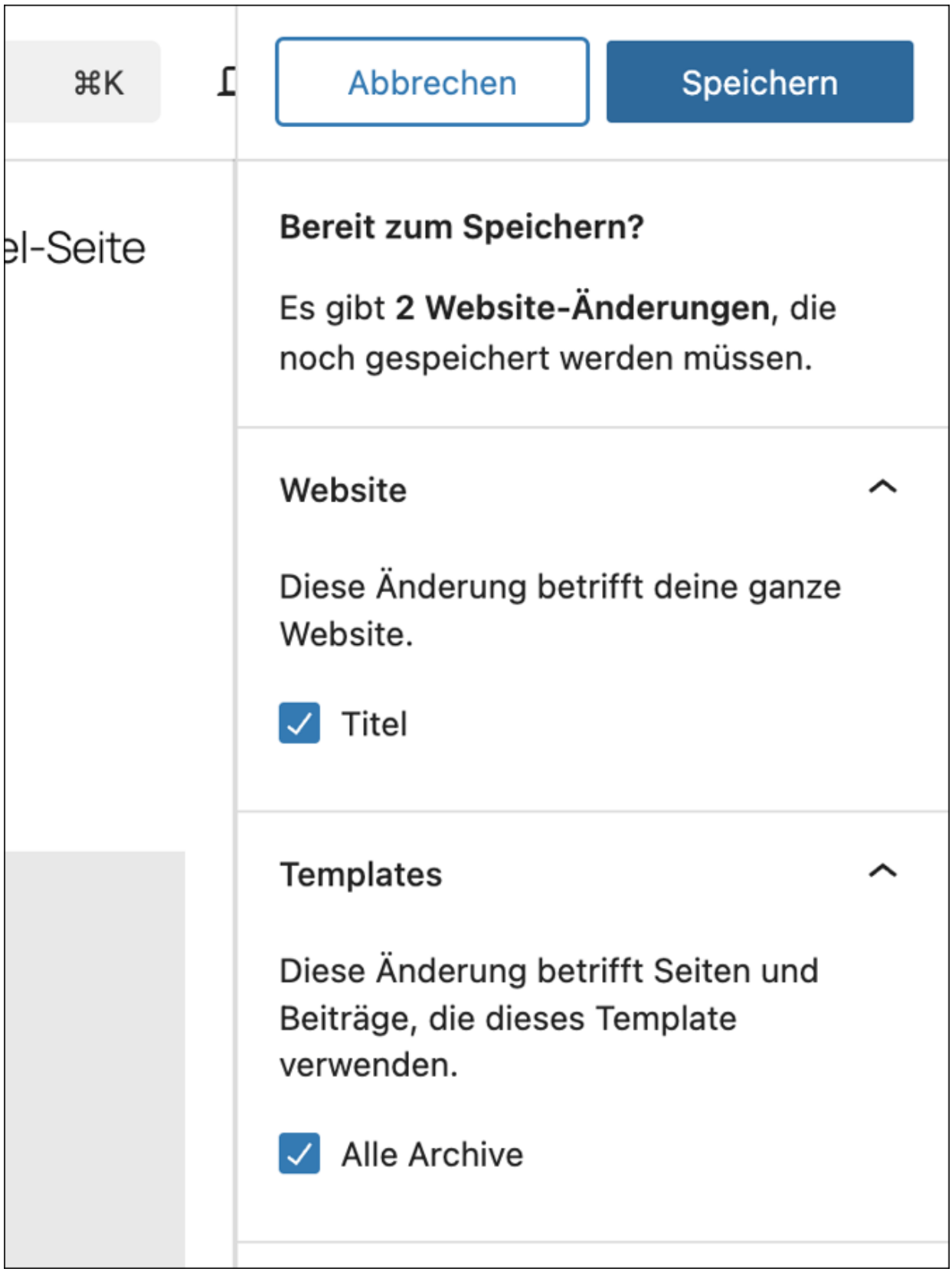

⌘K
Abbrechen
Speichern
el-Seite
Bereit zum Speichern?
Es gibt 2 Website-Änderungen, die noch gespeichert werden müssen.
Website
Diese Änderung betrifft deine ganze Website.
Titel
Templates
Diese Änderung betrifft Seiten und Beiträge, die dieses Template verwenden.
Alle Archive

Abbildung 10.35 Nach dem Klick auf den »Speichern«-Button im Website-Editor werden alle Komponenten aufgelistet, die Sie verändert haben. Hier haben Sie dann die Möglichkeit, Änderungen an einzelnen Komponenten doch nicht zu speichern.

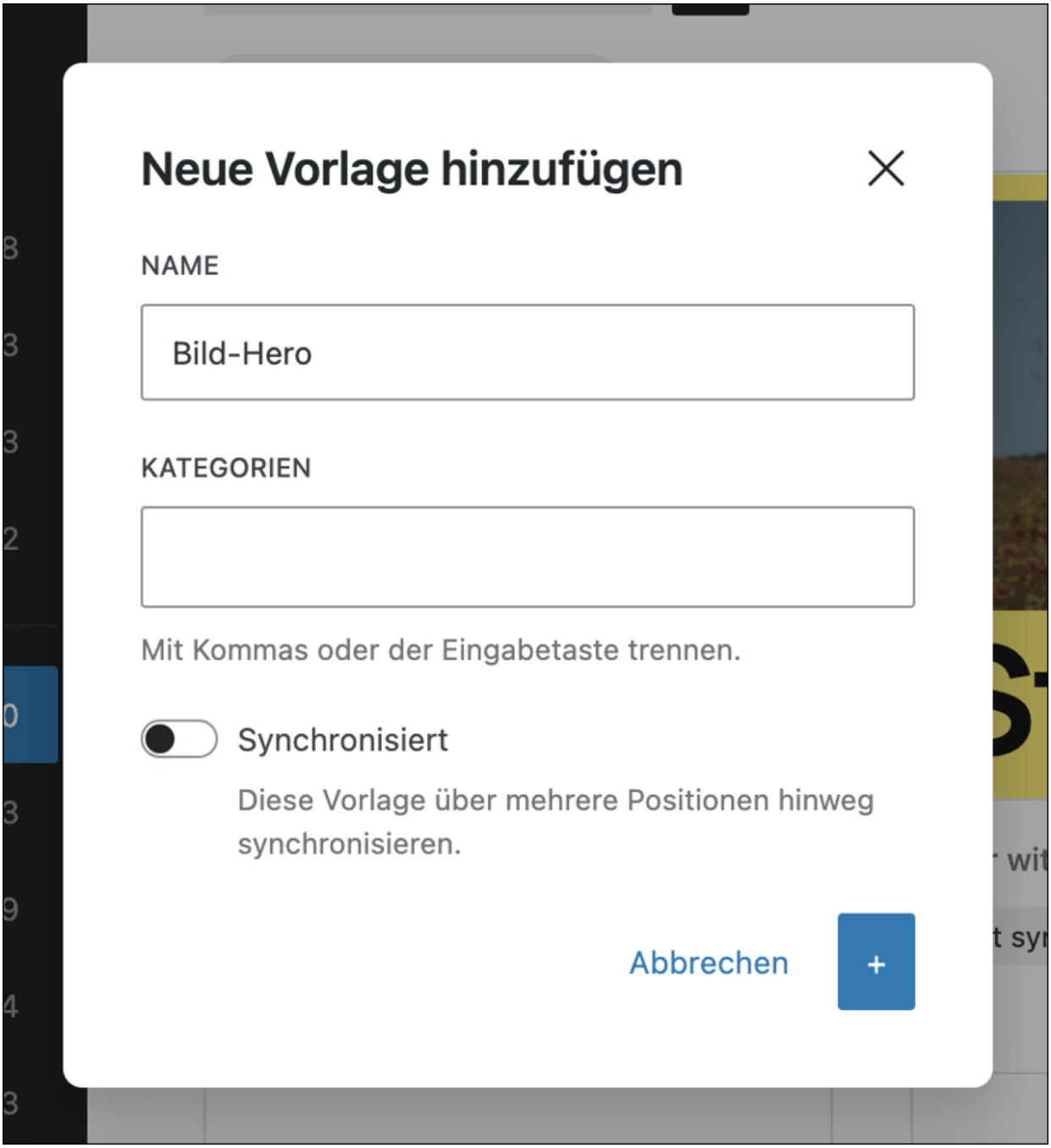

Neue Vorlage hinzufügen
NAME
Bild-Hero
KATEGORIEN
Mit Kommas oder der Eingabetaste trennen.
Synchronisiert
Diese Vorlage über mehrere Positionen hinweg synchronisieren.
Abbrechen
+

Abbildung 10.36 Eine Vorlage braucht einen Namen und optional eine Kategorie. Sie können außerdem festlegen, ob die Vorlage an allen Stellen, wo sie eingefügt wird, immer denselben Inhalt haben soll.

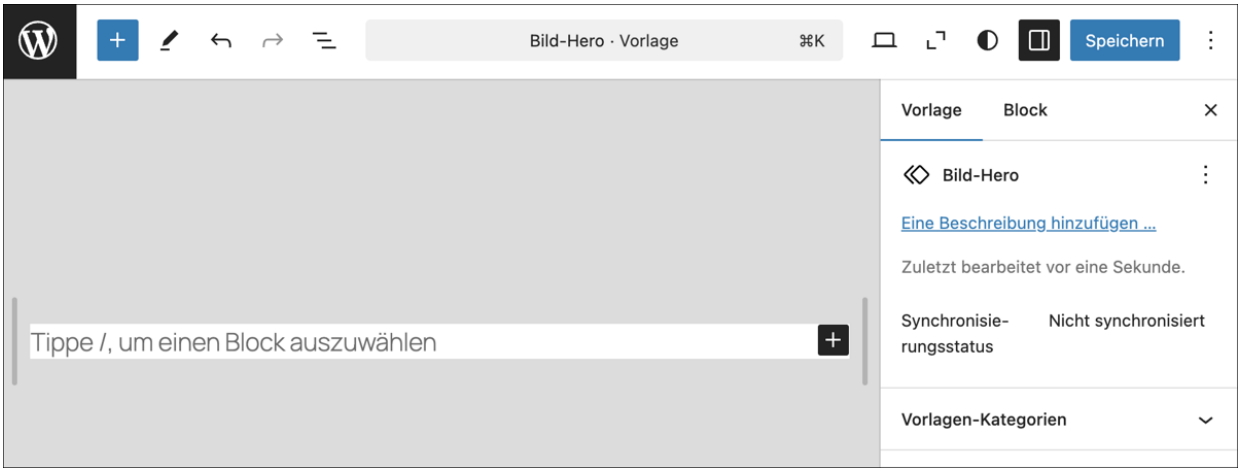


Abbildung 10.37 Der Editor für die Vorlage ähnelt dem für Template-Teile.

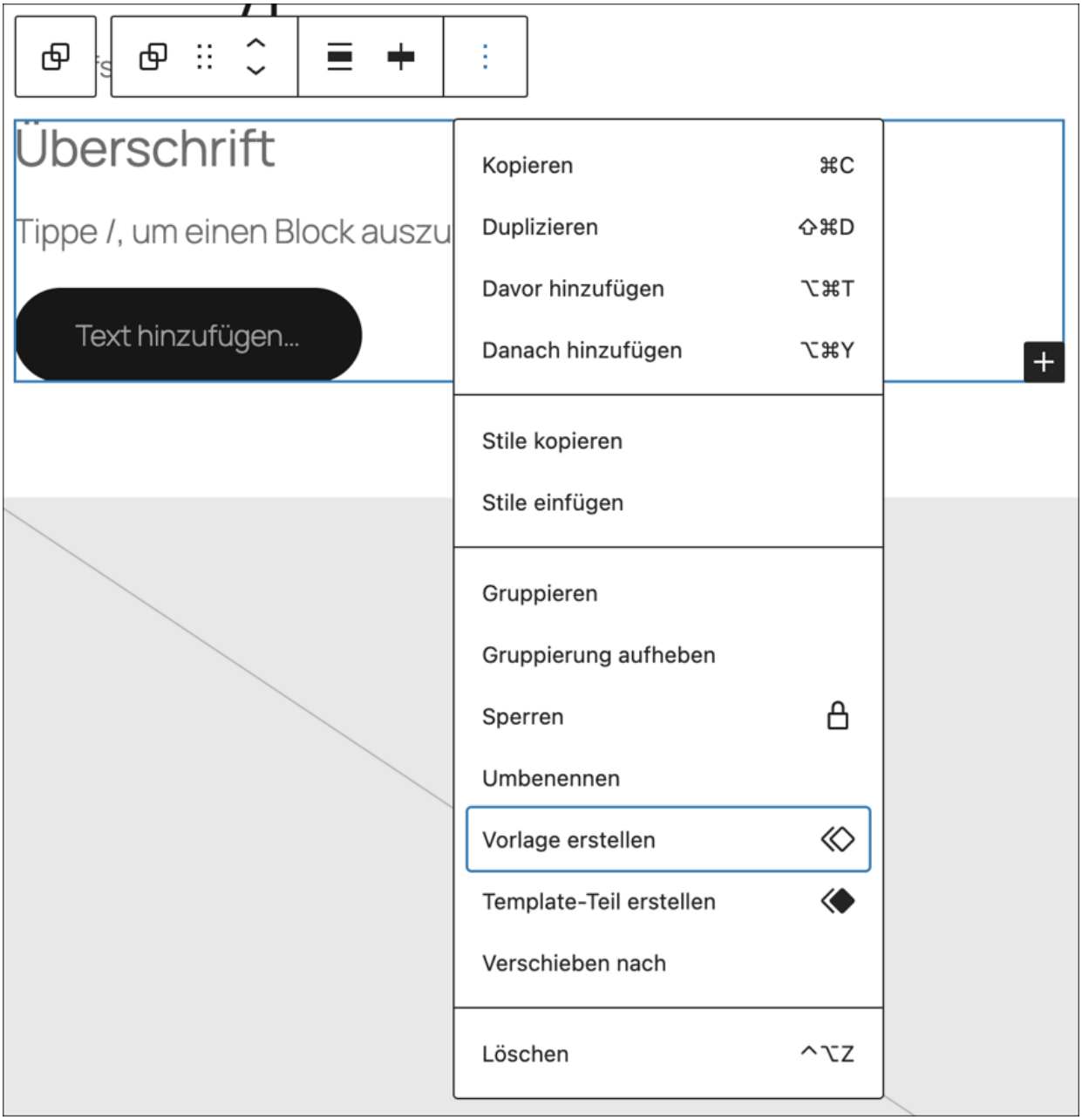


Abbildung 10.38 Sie können einfach direkt aus dem Editor heraus einen oder mehrere Blöcke in eine Vorlage umwandeln, sodass Sie den oder die Blöcke wiederverwenden können.

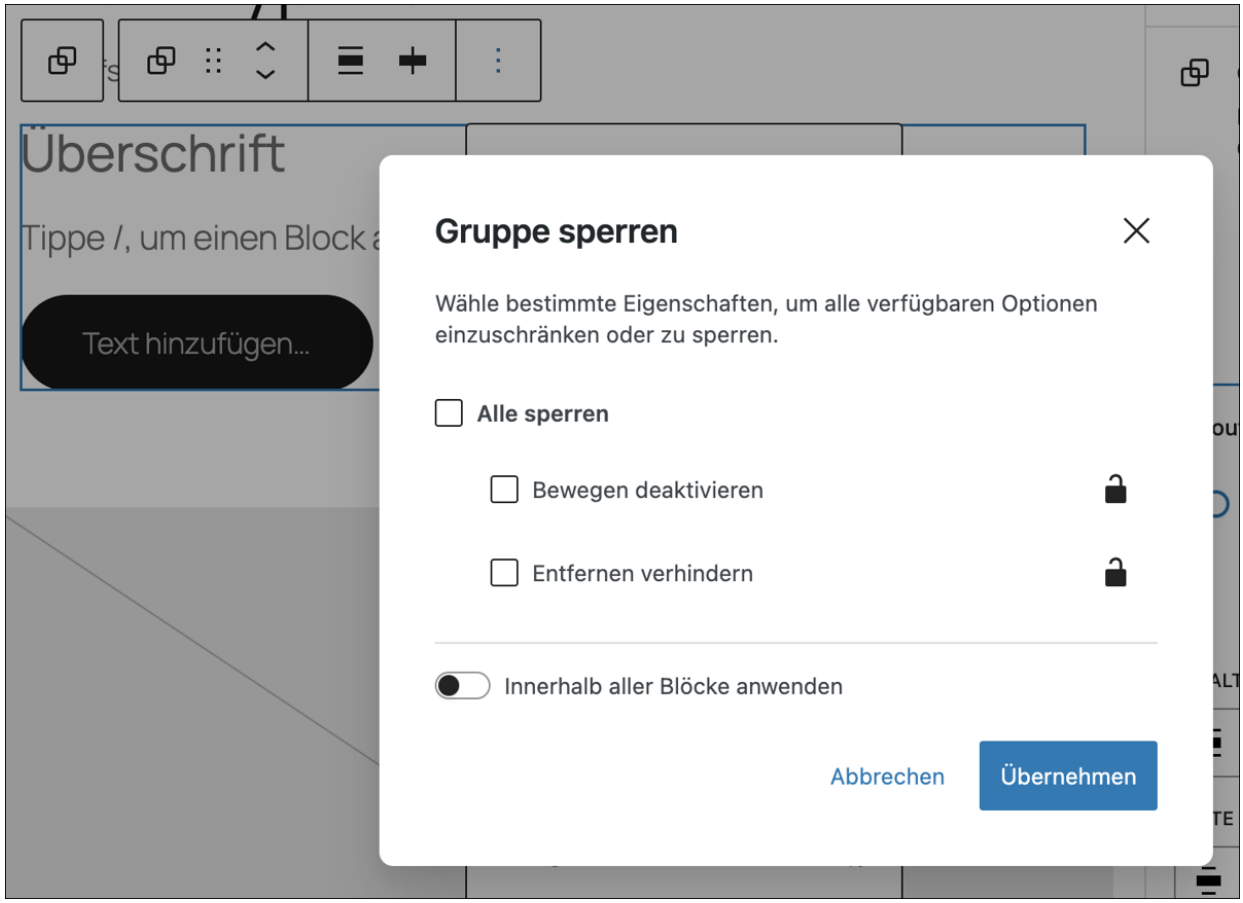


Abbildung 10.39 Durch das »Sperren« von Blöcken kann beispielsweise verhindert werden, dass die Reihenfolge von Blöcken verändert wird.

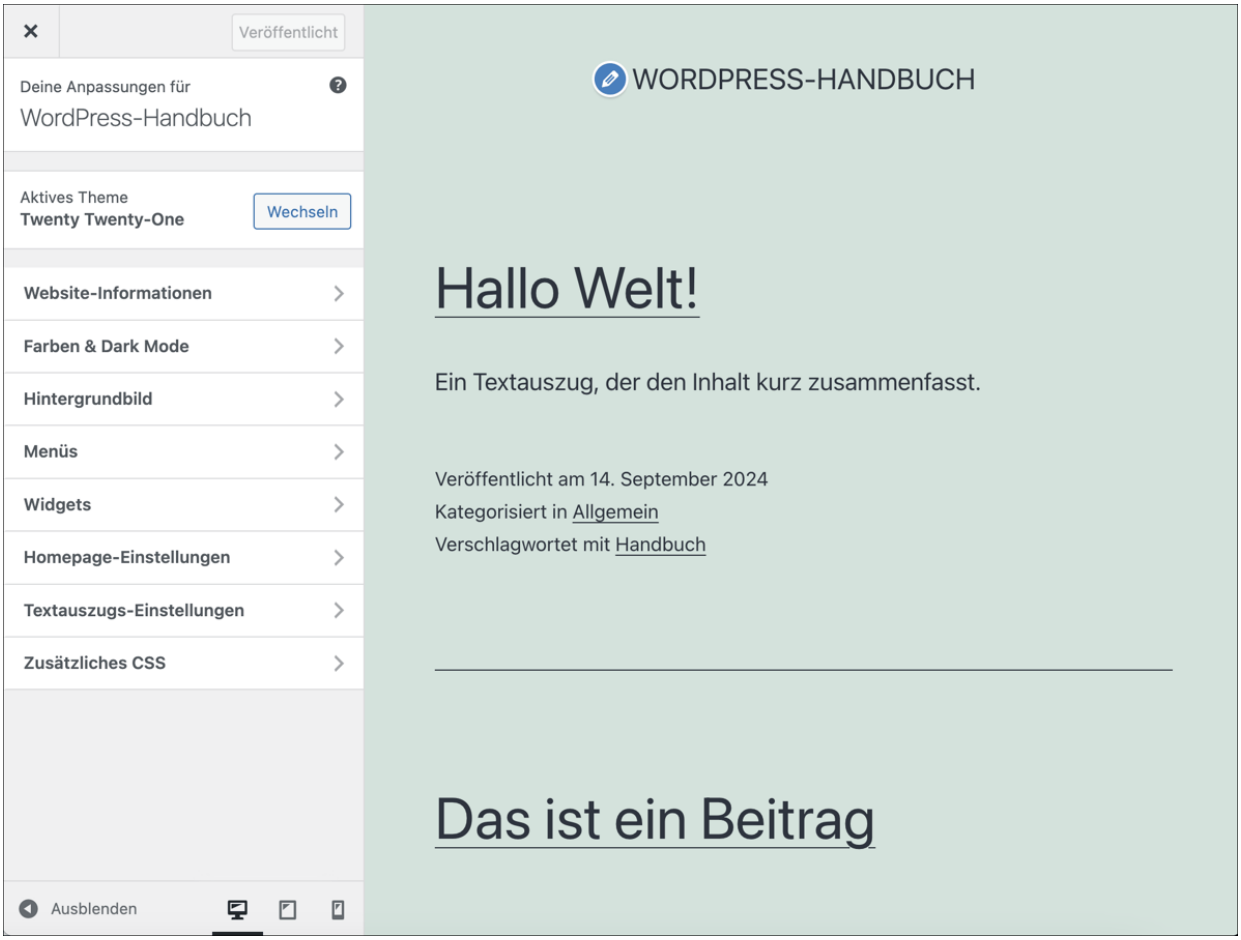


Abbildung 10.40 Der Customizer ermöglicht Ihnen die Anpassung von Classic Themes mit einer Live-Vorschau der Änderungen.

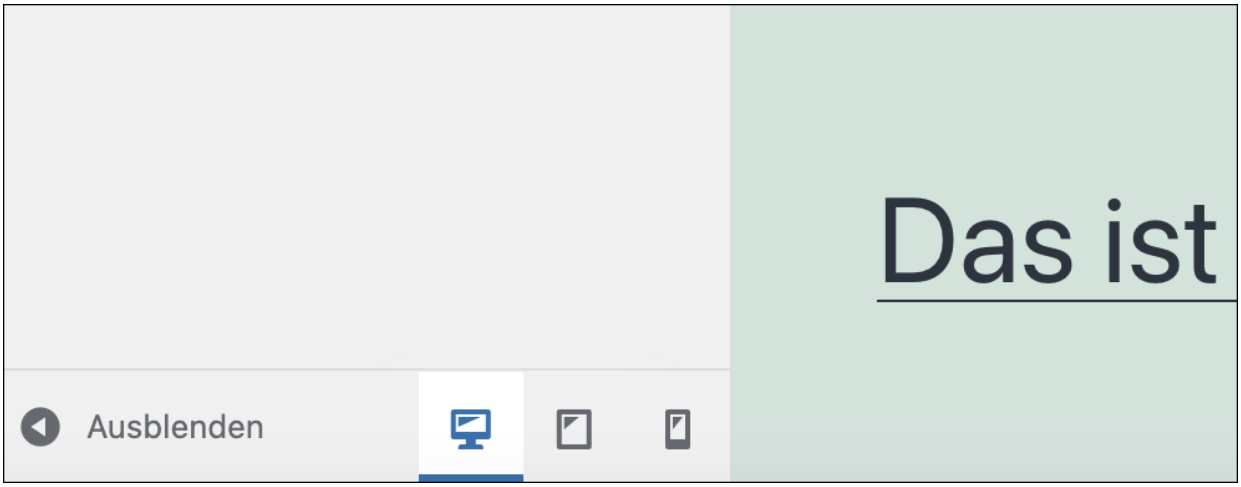


Abbildung 10.41 Am unteren Ende der Sidebar des Customizers finden Sie die Option, die Vorschau auf unterschiedliche Viewport-Größen zu verändern.

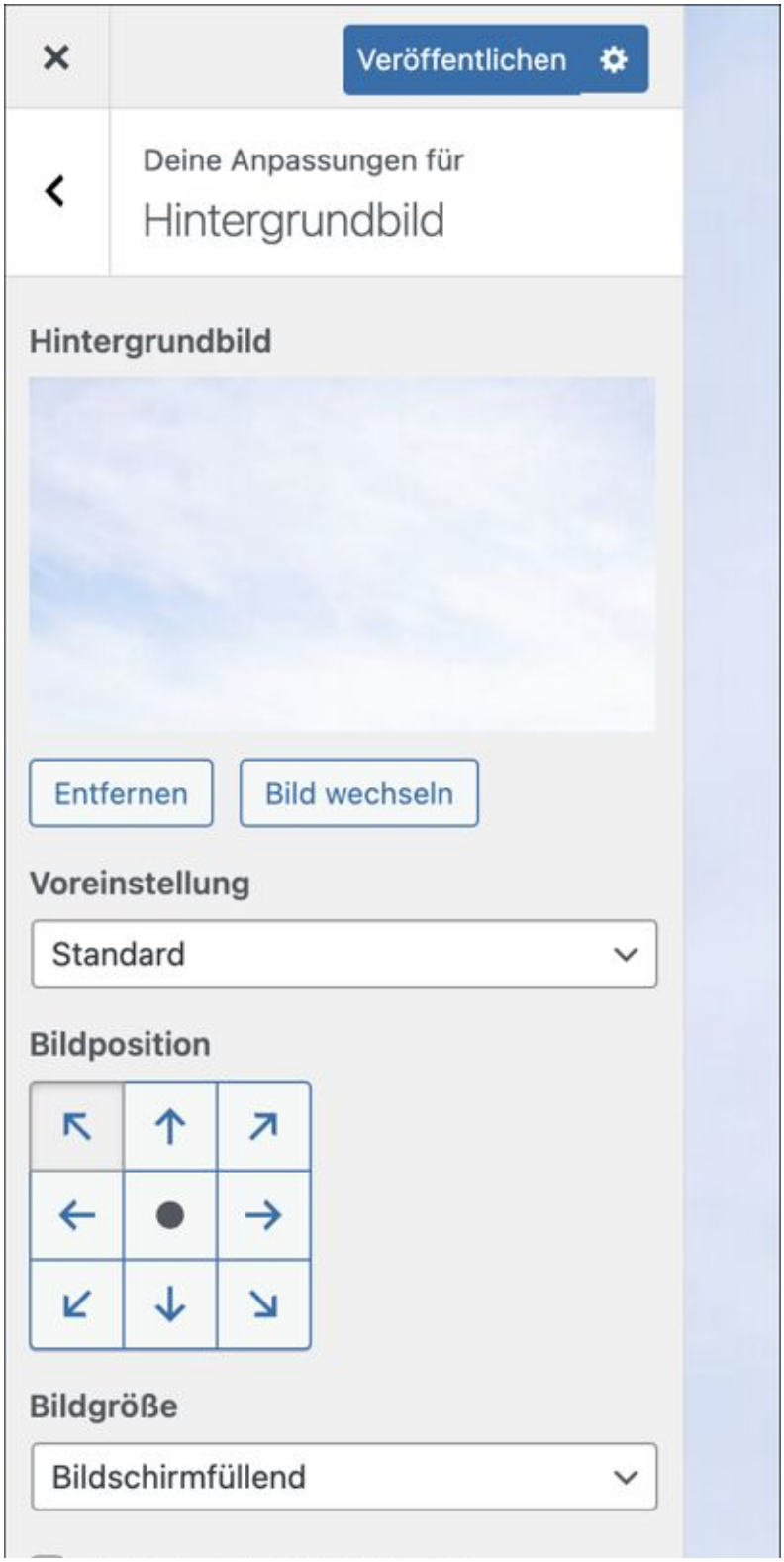

Veröffentlichen
Deine Anpassungen für
Hintergrundbild
Hintergrundbild
Entfernen
Bild wechseln
Voreinstellung
Standard
Bildposition
Bildgröße
Bildschirmfüllend



Abbildung 10.42 Wenn Sie ein Hintergrundbild festgelegt haben, erscheinen zusätzliche Optionen, um die Darstellung des Bilds zu steuern.

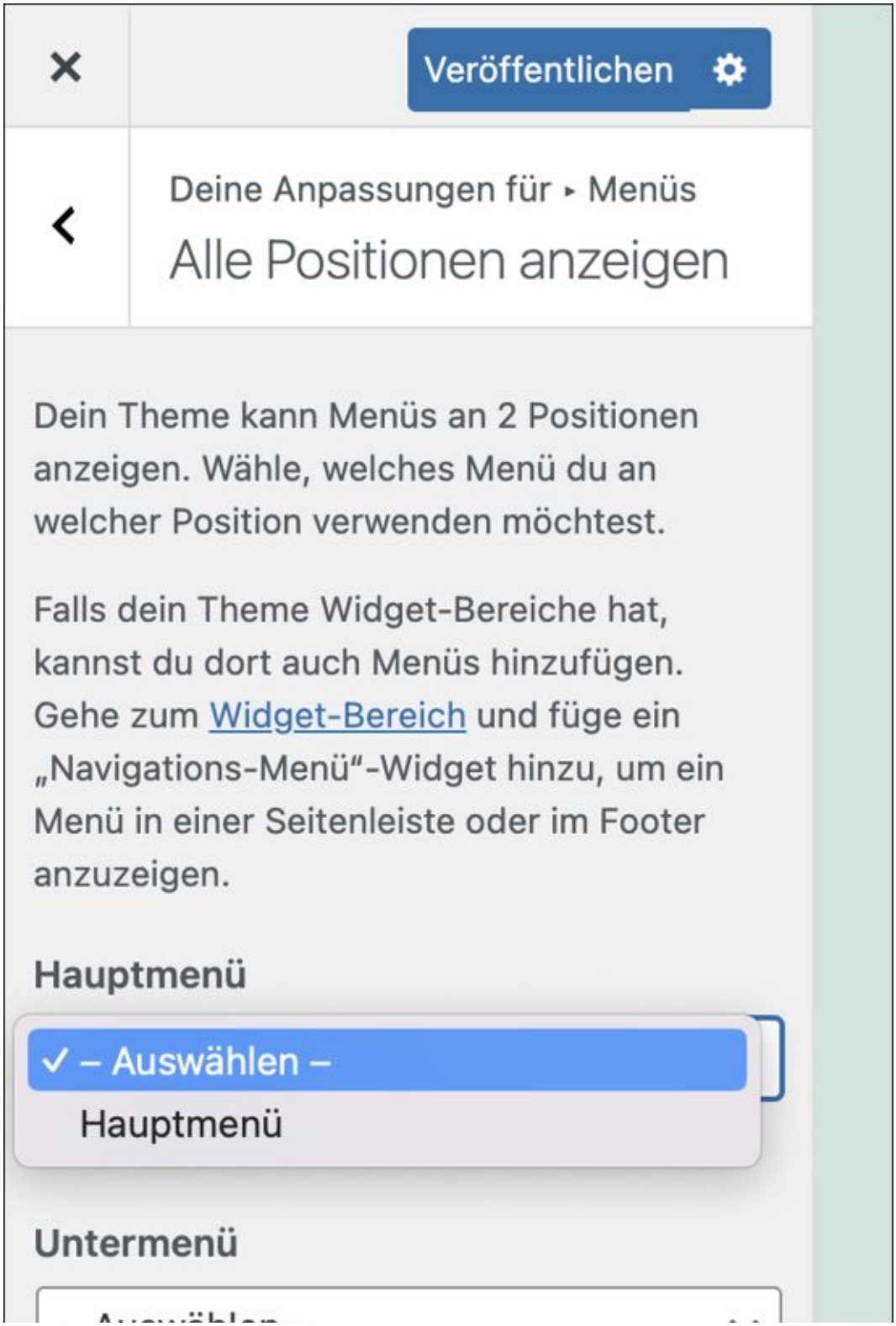

Veröffentlichen
Deine Anpassungen für ▸ Menüs
Alle Positionen anzeigen
Dein Theme kann Menüs an 2 Positionen anzeigen. Wähle, welches Menü du an welcher Position verwenden möchtest.
Falls dein Theme Widget-Bereiche hat, kannst du dort auch Menüs hinzufügen. Gehe zum Widget-Bereich und füge ein „Navigations-Menü“-Widget hinzu, um ein Menü in einer Seitenleiste oder im Footer anzuzeigen.
Hauptmenü
✓ – Auswählen –
Hauptmenü
Untermenü



Abbildung 10.43 Wenn Sie Menüs erstellt haben, können Sie diese einer Menüposition zuweisen und dadurch an der entsprechenden Stelle im Theme anzeigen.

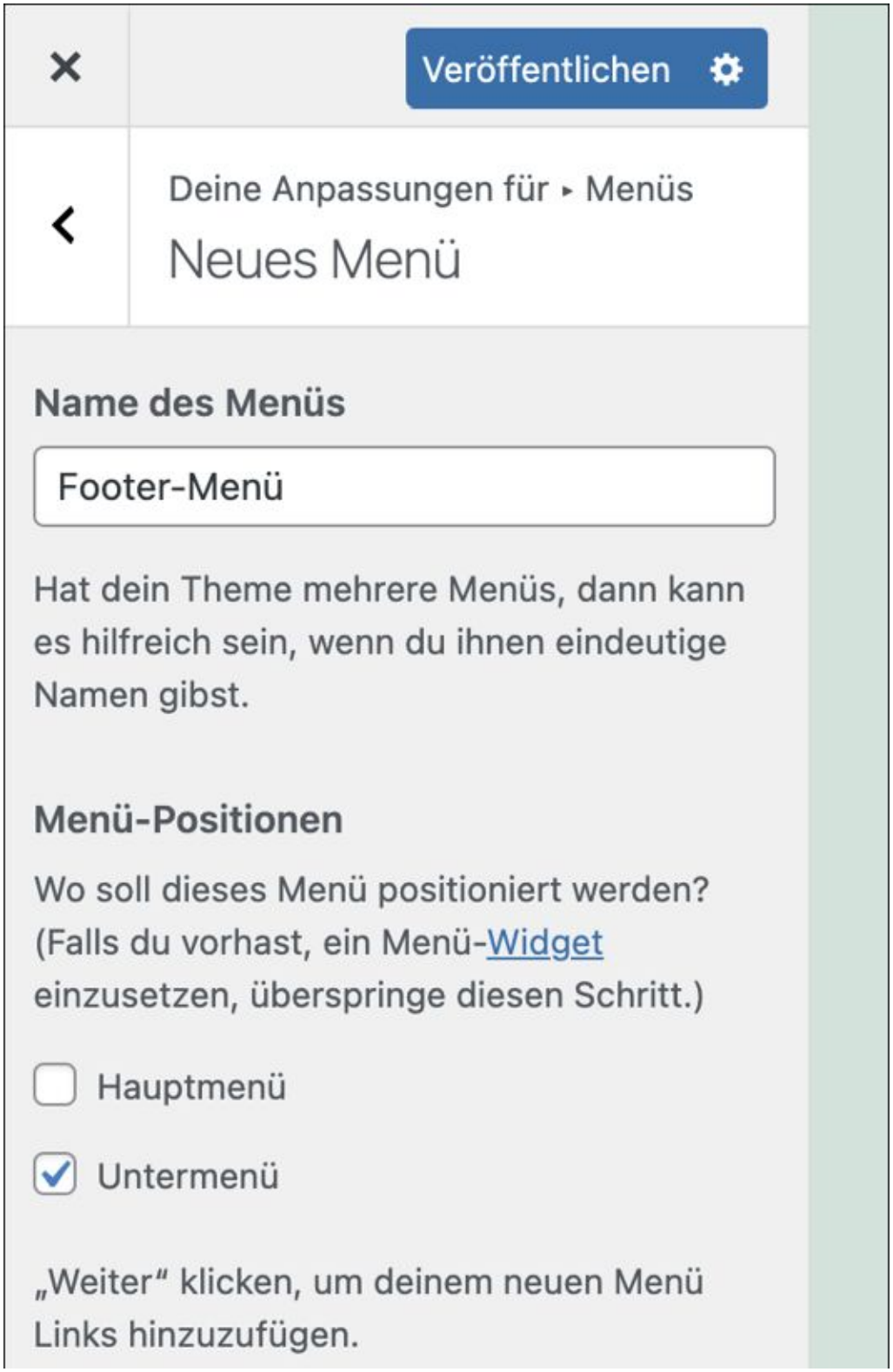

Veröffentlichen
Deine Anpassungen für ▸ Menüs
Neues Menü
Name des Menüs
Footer-Menü
Hat dein Theme mehrere Menüs, dann kann es hilfreich sein, wenn du ihnen eindeutige Namen gibst.
Menü-Positionen
Wo soll dieses Menü positioniert werden? (Falls du vorhast, ein Menü-Widget einzusetzen, überspringe diesen Schritt.)
Hauptmenü
Untermenü
„Weiter“ klicken, um deinem neuen Menü Links hinzuzufügen.



Abbildung 10.44 Im ersten Schritt für ein neues Menü im Customizer legen Sie einen Namen und optional die Menü-Position fest.

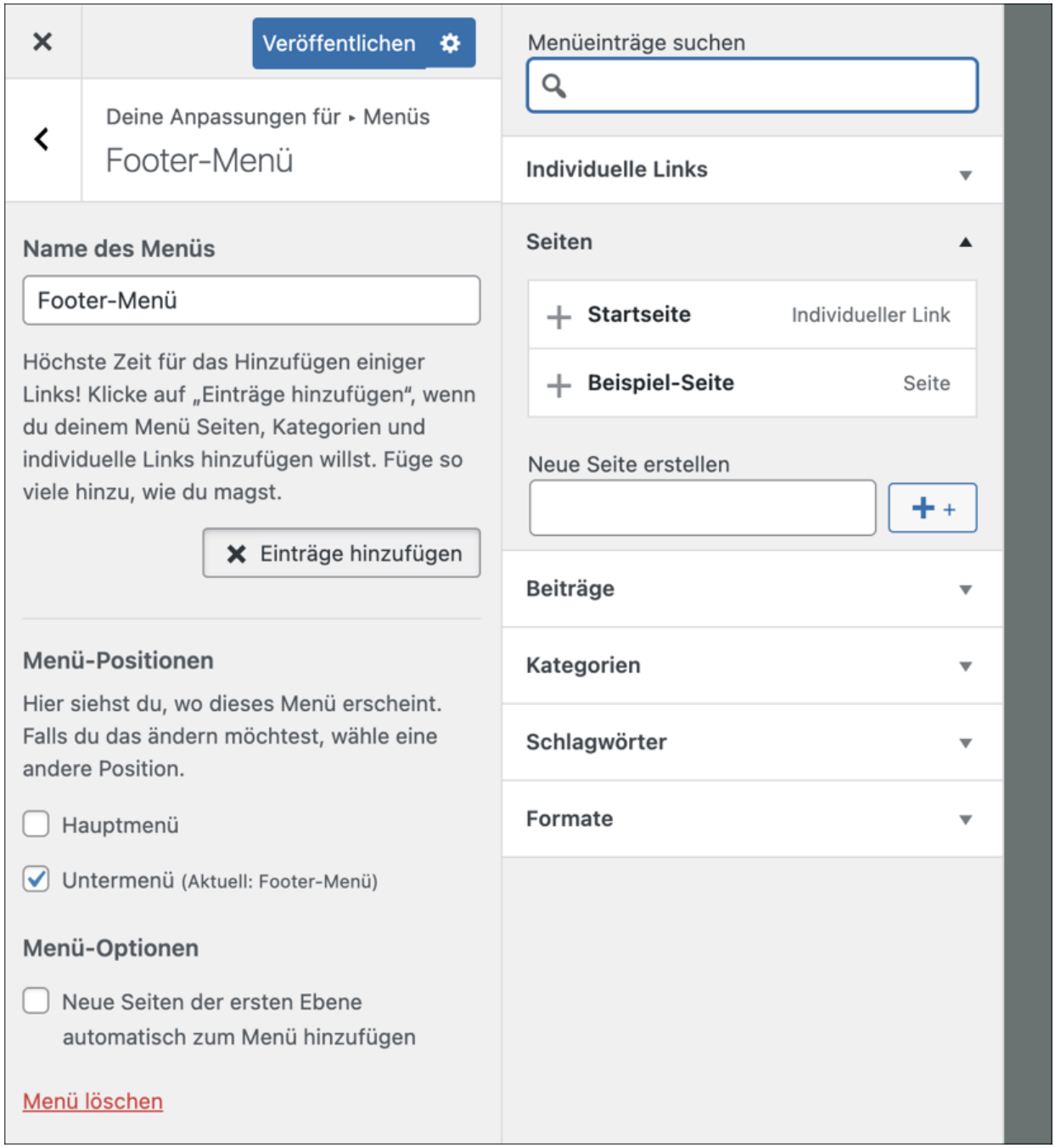
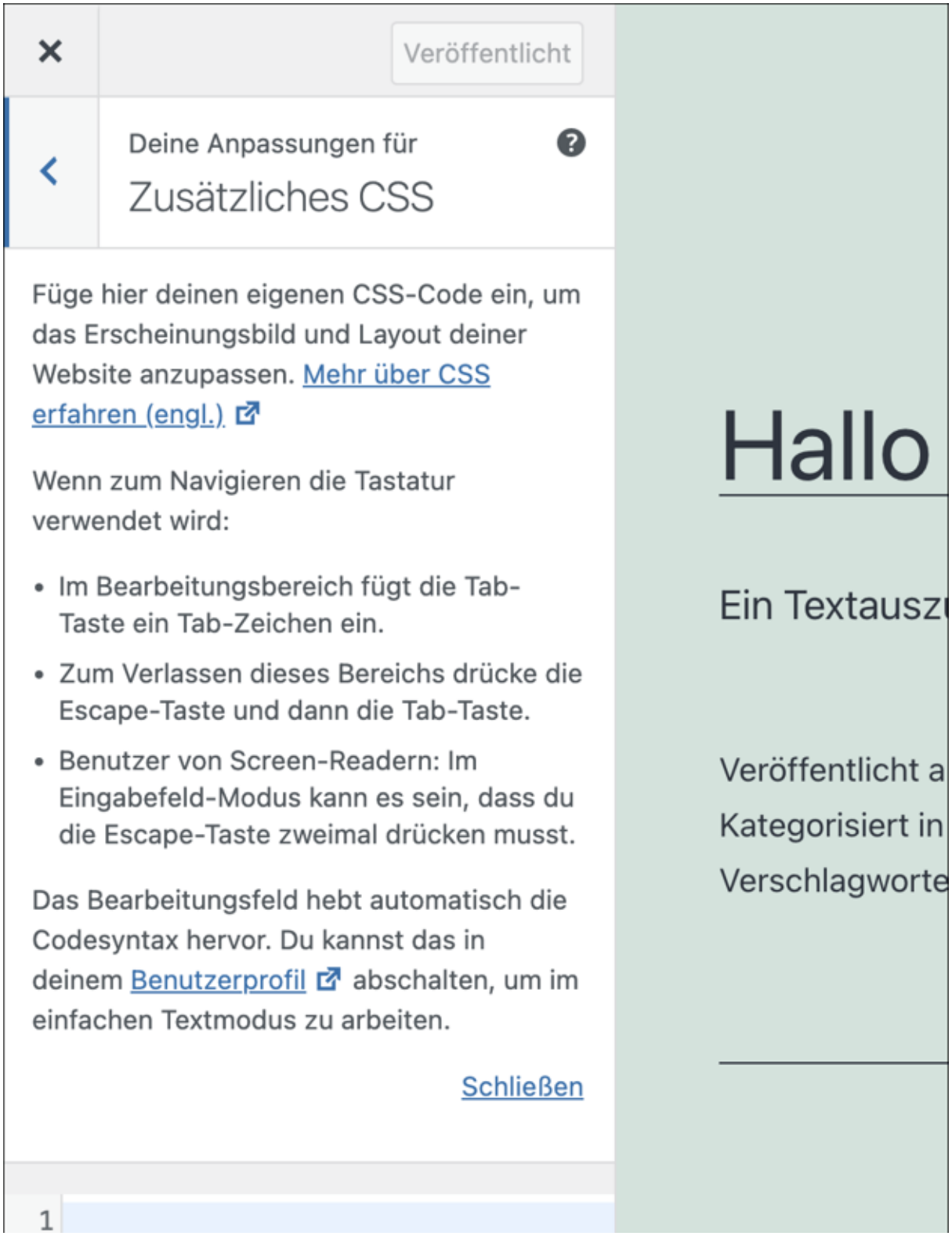


Abbildung 10.45 Im zweiten Schritt der Menü-Erstellung gelangen Sie in die Ansicht, in der Sie die Menüpunkte hinzufügen und verändern können.


Veröffentlicht
Deine Anpassungen für
Zusätzliches CSS
Füge hier deinen eigenen CSS-Code ein, um das Erscheinungsbild und Layout deiner Website anzupassen. Mehr über CSS erfahren (engl.)
Wenn zum Navigieren die Tastatur verwendet wird:
Im Bearbeitungsbereich fügt die Tab-Taste ein Tab-Zeichen ein.
Zum Verlassen dieses Bereichs drücke die Escape-Taste und dann die Tab-Taste.
Benutzer von Screen-Readern: Im Eingabefeld-Modus kann es sein, dass du die Escape-Taste zweimal drücken musst.
Das Bearbeitungsfeld hebt automatisch die Codesyntax hervor. Du kannst das in deinem Benutzerprofil abschalten, um im einfachen Textmodus zu arbeiten.
Schließen
1
Hallo
Ein Textausz
Veröffentlicht a
Kategorisiert in
Verschlagworte

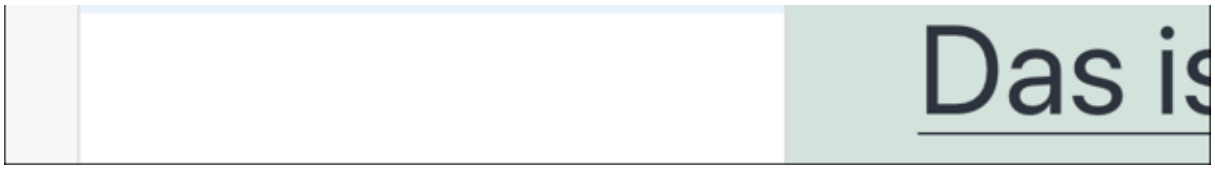


Abbildung 10.46 Sie können im Customizer das Aussehen des Themes über die verfügbaren Optionen hinaus anpassen, indem Sie eigenes CSS in das Feld »Zusätzliches CSS« einfügen.

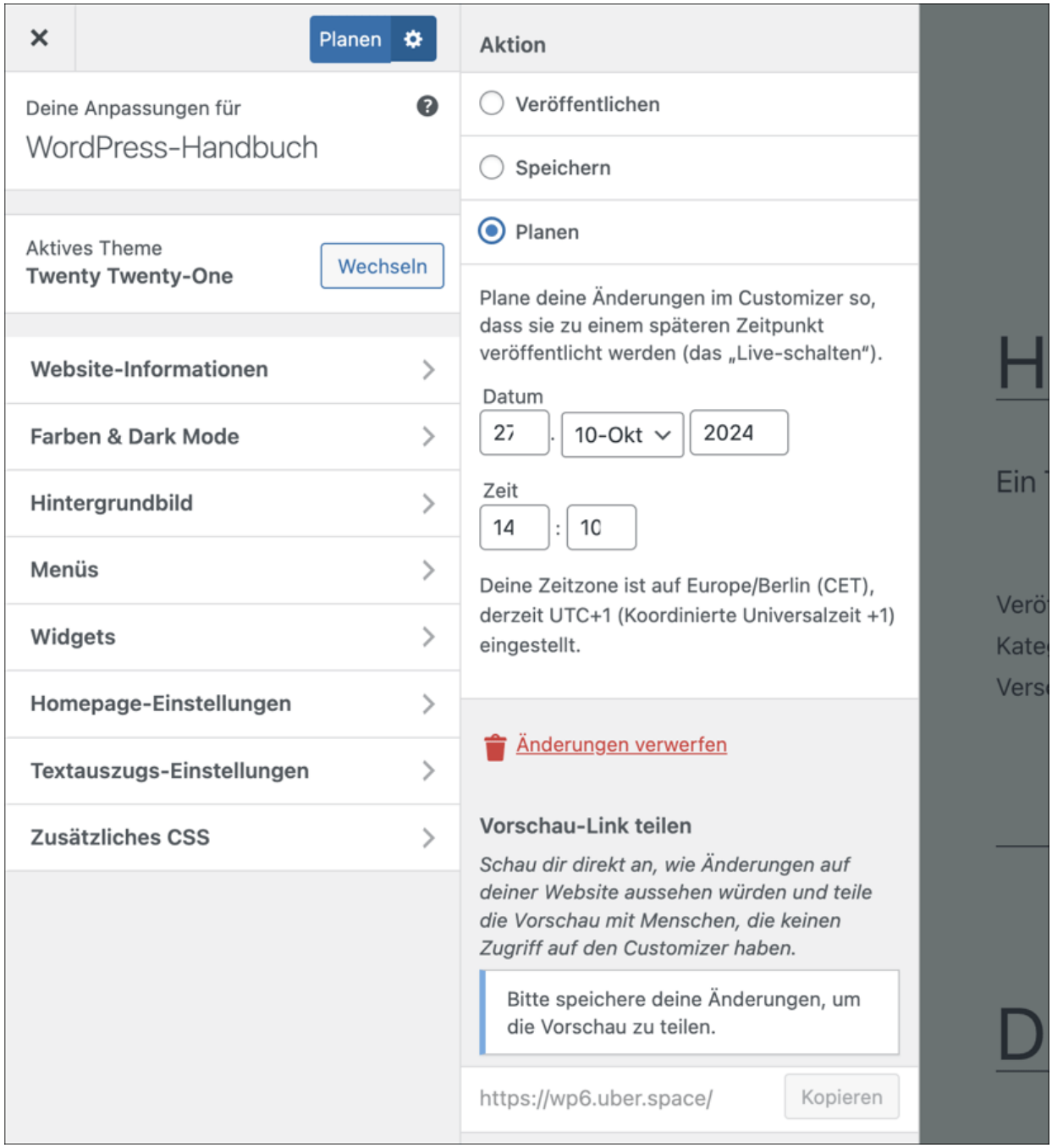

Planen
Deine Anpassungen für
WordPress-Handbuch
Aktives Theme
Twenty Twenty-One
Wechseln
Website-Informationen
Farben & Dark Mode
Hintergrundbild
Menüs
Widgets
Homepage-Einstellungen
Textauszugs-Einstellungen
Zusätzliches CSS
Aktion
Veröffentlichen
Speichern
Planen
Plane deine Änderungen im Customizer so, dass sie zu einem späteren Zeitpunkt veröffentlicht werden (das „Live-schalten“).
Datum
27
10-Okt
2024
Zeit
14
10
Deine Zeitzone ist auf Europe/Berlin (CET), derzeit UTC+1 (Koordinierte Universalzeit +1) eingestellt.
Änderungen verwerfen
Vorschau-Link teilen
Schau dir direkt an, wie Änderungen auf deiner Website aussehen würden und teile die Vorschau mit Menschen, die keinen Zugriff auf den Customizer haben.
Bitte speichere deine Änderungen, um die Vorschau zu teilen.
https://wp6.uber.space/
Kopieren

Abbildung 10.47 Sie können Ihre Änderungen im Customizer auch für die Zukunft planen, sodass sie automatisch zu einem späteren Zeitpunkt veröffentlicht werden.

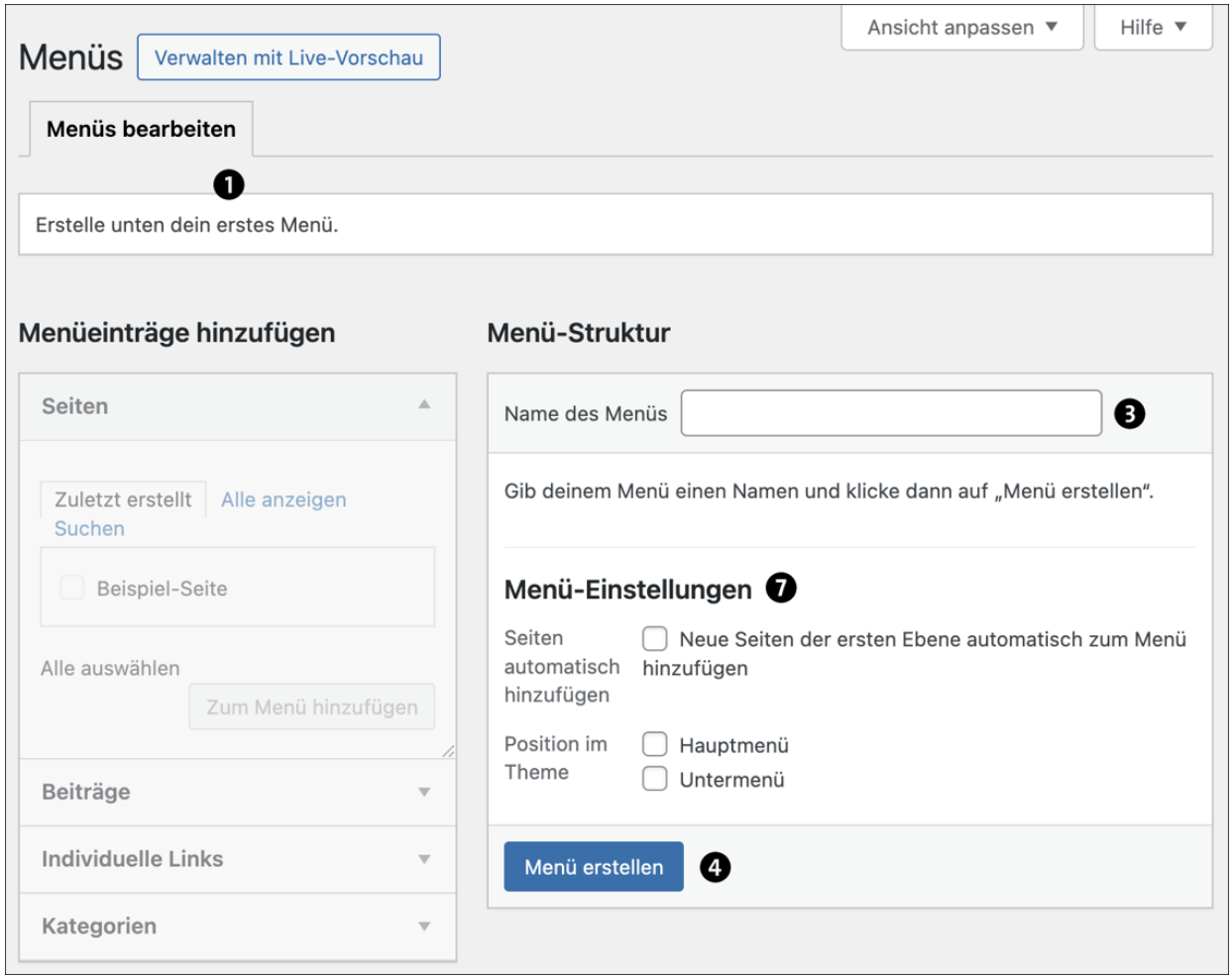


Abbildung 10.48 Unter »Design« • »Menüs« können Sie neue Menüs erstellen und bestehende Menüs bearbeiten.

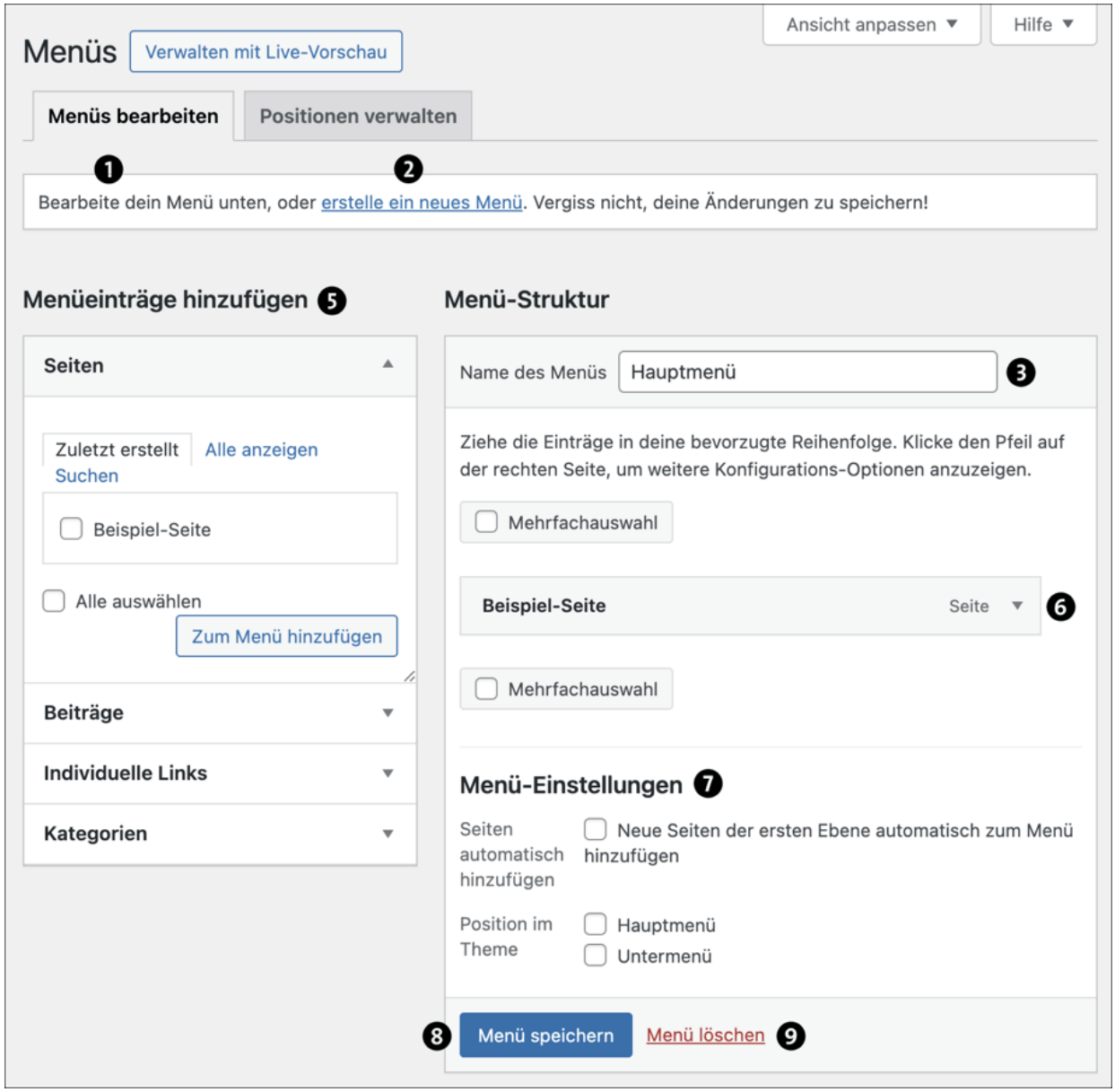


Abbildung 10.49 Diese Ansicht begrüßt Sie unter »Design« • »Menüs«, wenn Sie bereits ein Menü erstellt haben.



Abbildung 10.50 Per Drag-and-drop können Sie die Menüeinträge umsortieren und auch untereinander verschachteln.

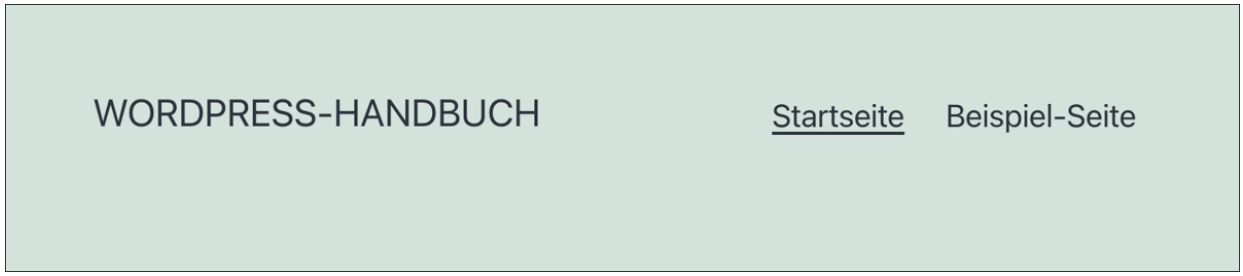


Abbildung 10.51 Das neue Menü als Hauptmenü im Theme Twenty Twenty-One.

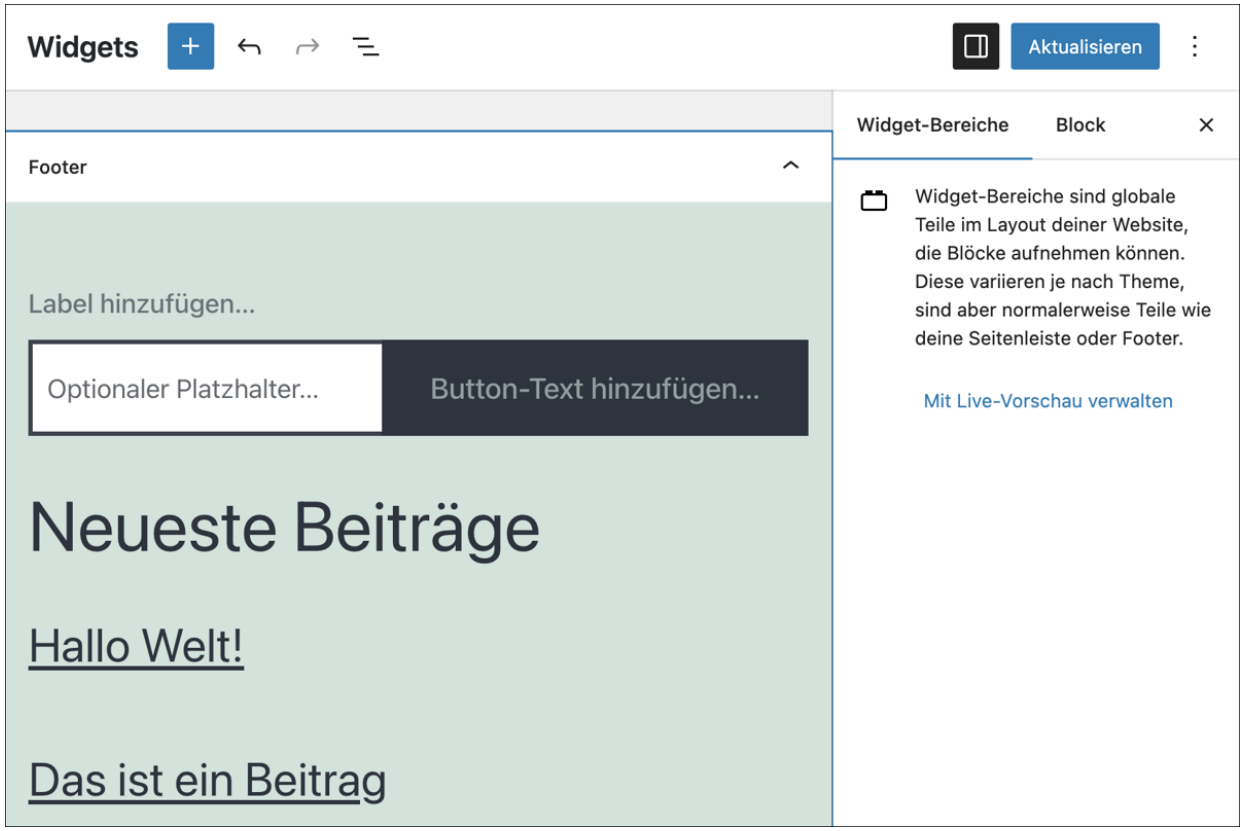


Abbildung 10.52 Unter »Design« • »Widgets« können Sie für Classic Themes die Widget-Bereiche füllen.

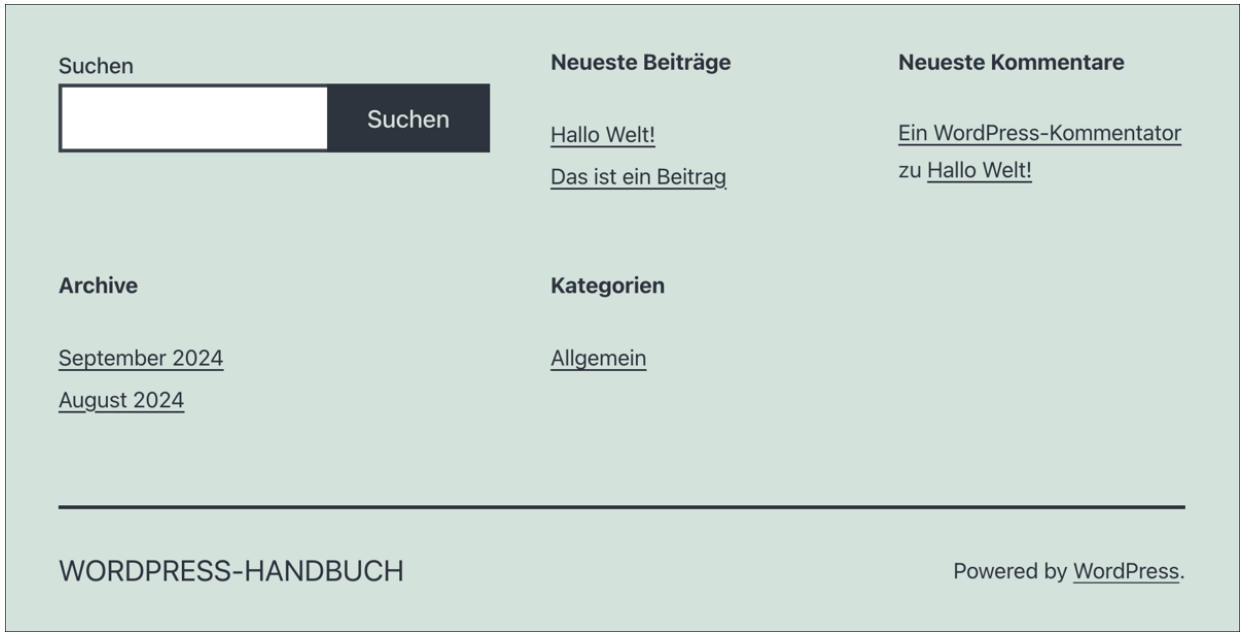


Abbildung 10.53 Der Widget-Bereich von Twenty Twenty-One im Footer

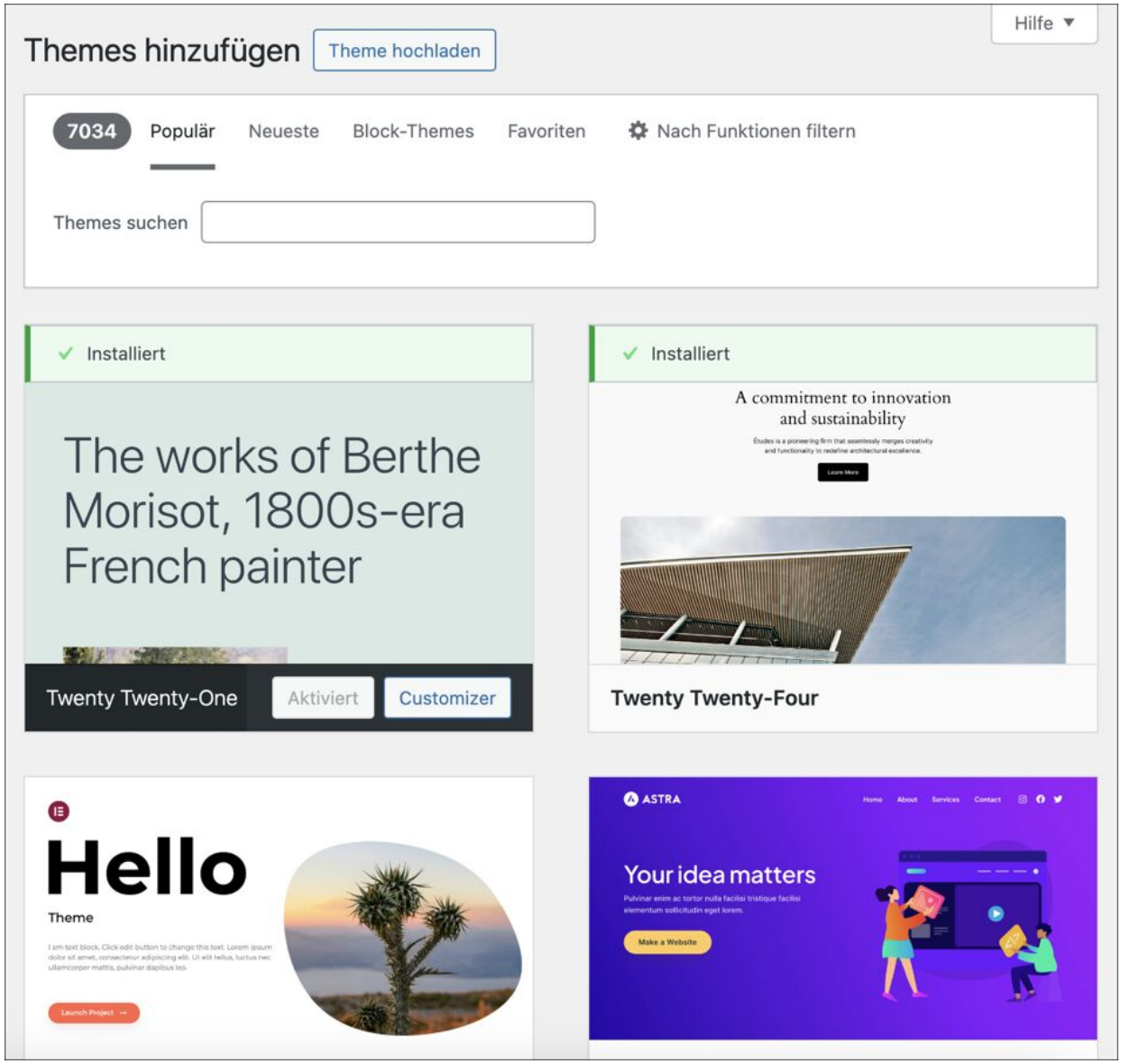


Abbildung 10.54 Die Themes aus dem WordPress.org-Theme-Verzeichnis können Sie direkt aus dem WordPress-Backend heraus durchsuchen. Dafür gehen Sie auf die Seite »Design« • »Themes« und klicken oben auf »Neues Theme hinzufügen«.

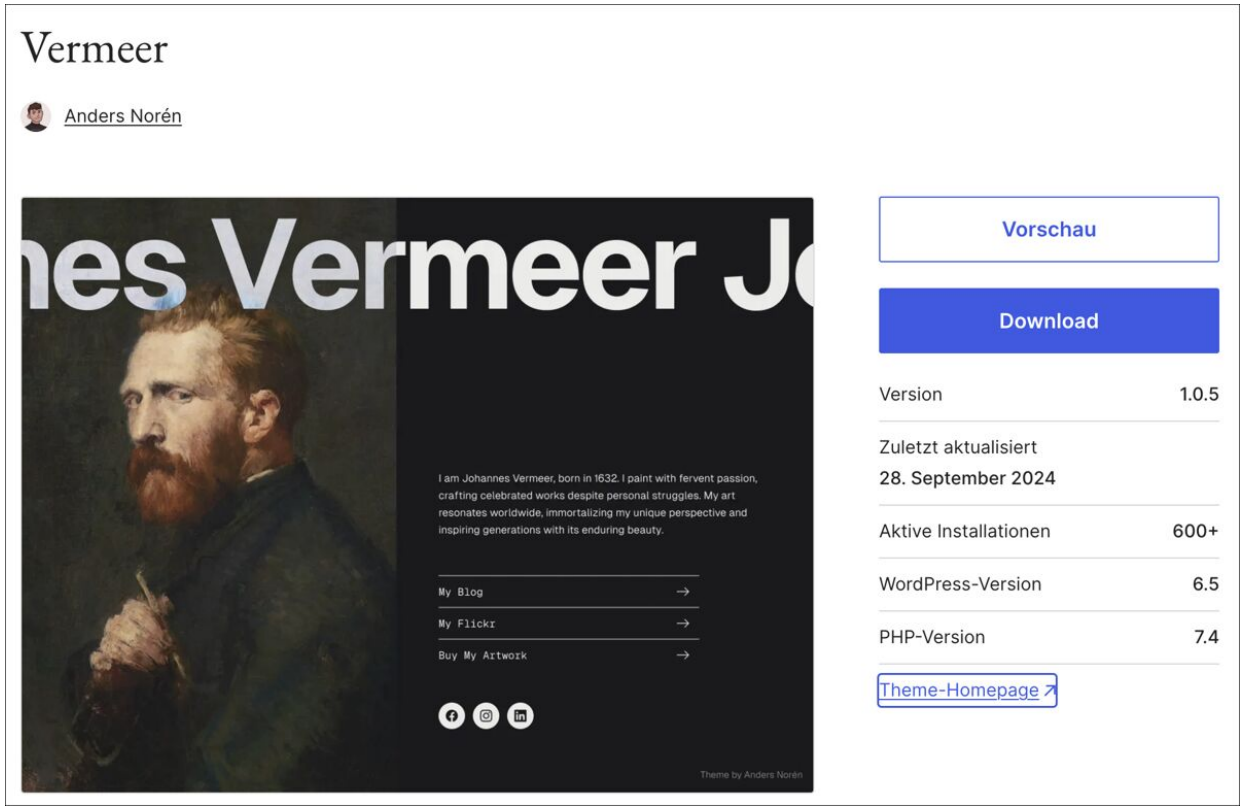


Abbildung 10.55 Über den Link »Theme-Homepage« gelangen Sie zu der Seite, die von den Theme-Autor*innen als Theme-Homepage hinterlegt wurde. Oft ist dort eine Demo-Site verlinkt, die eine bessere Vorschau bietet als die Vorschaufunktion des Theme-Verzeichnisses.

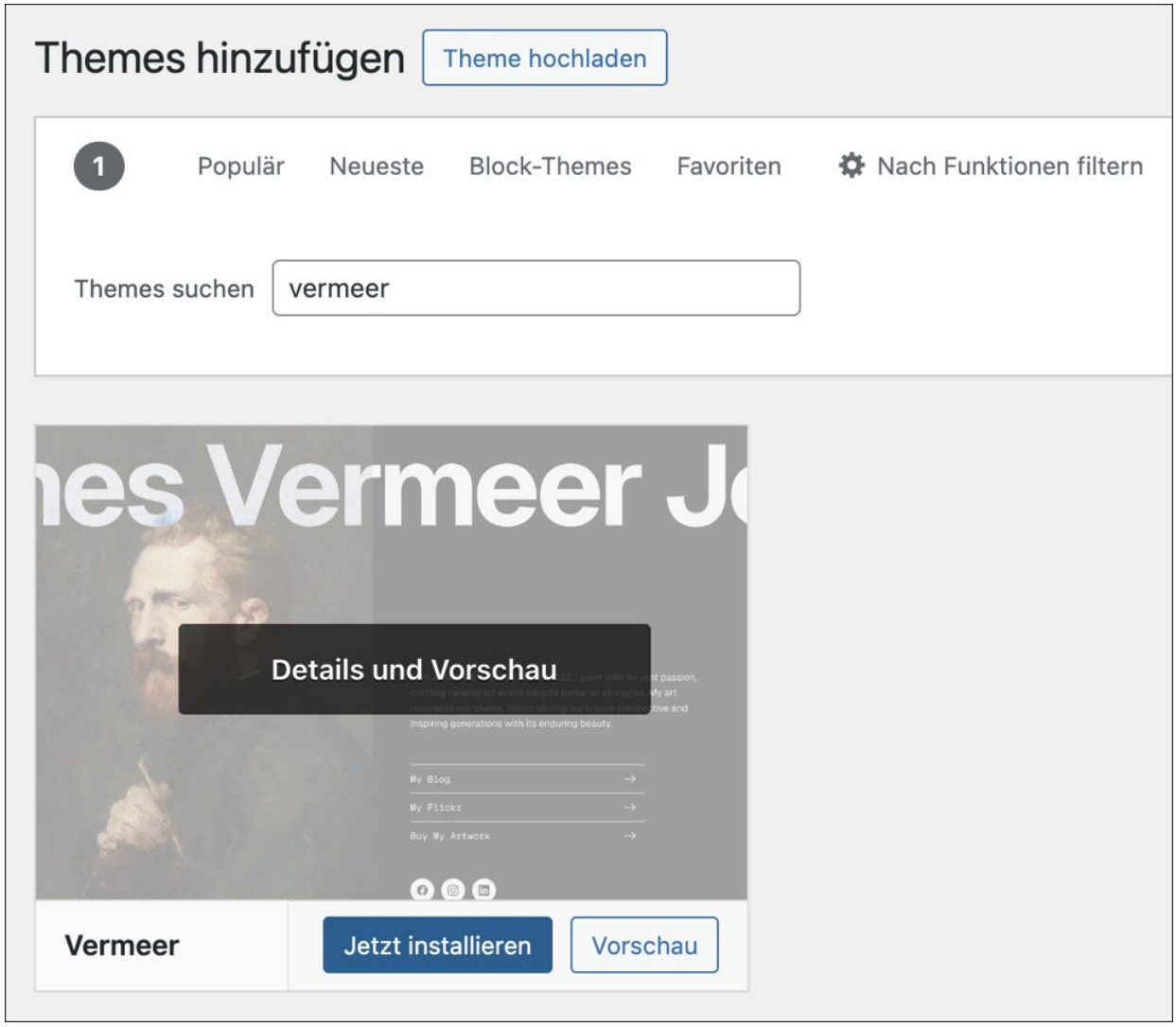


Abbildung 10.56 Ein Theme aus dem WordPress.org-Theme-Verzeichnis kann einfach direkt über das WordPress-Backend installiert werden.

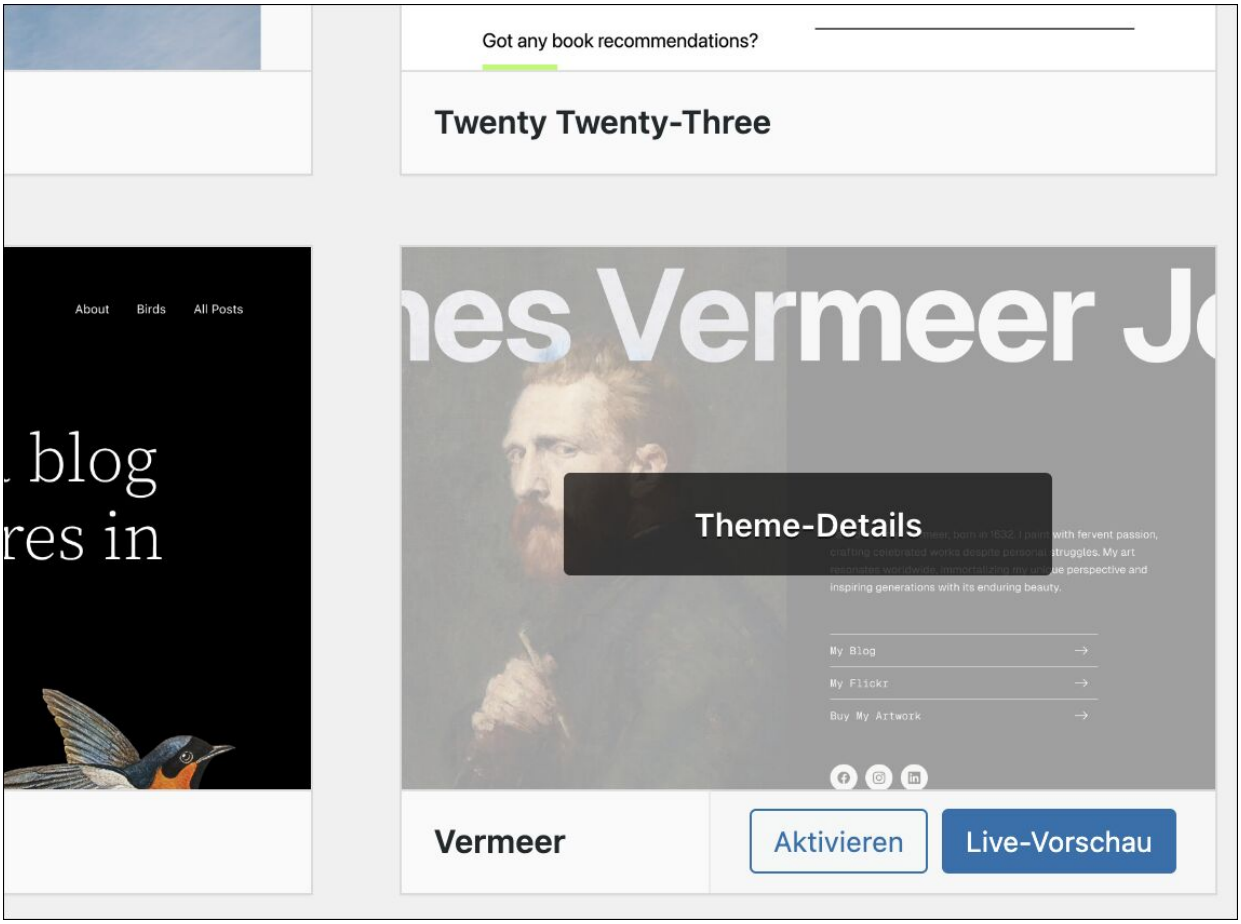


Abbildung 10.57 Nachdem das Theme installiert wurde, können Sie es direkt aktivieren oder erst mal in der Live-Vorschau anschauen und gegebenenfalls Änderungen vornehmen, bevor Sie es aktivieren.

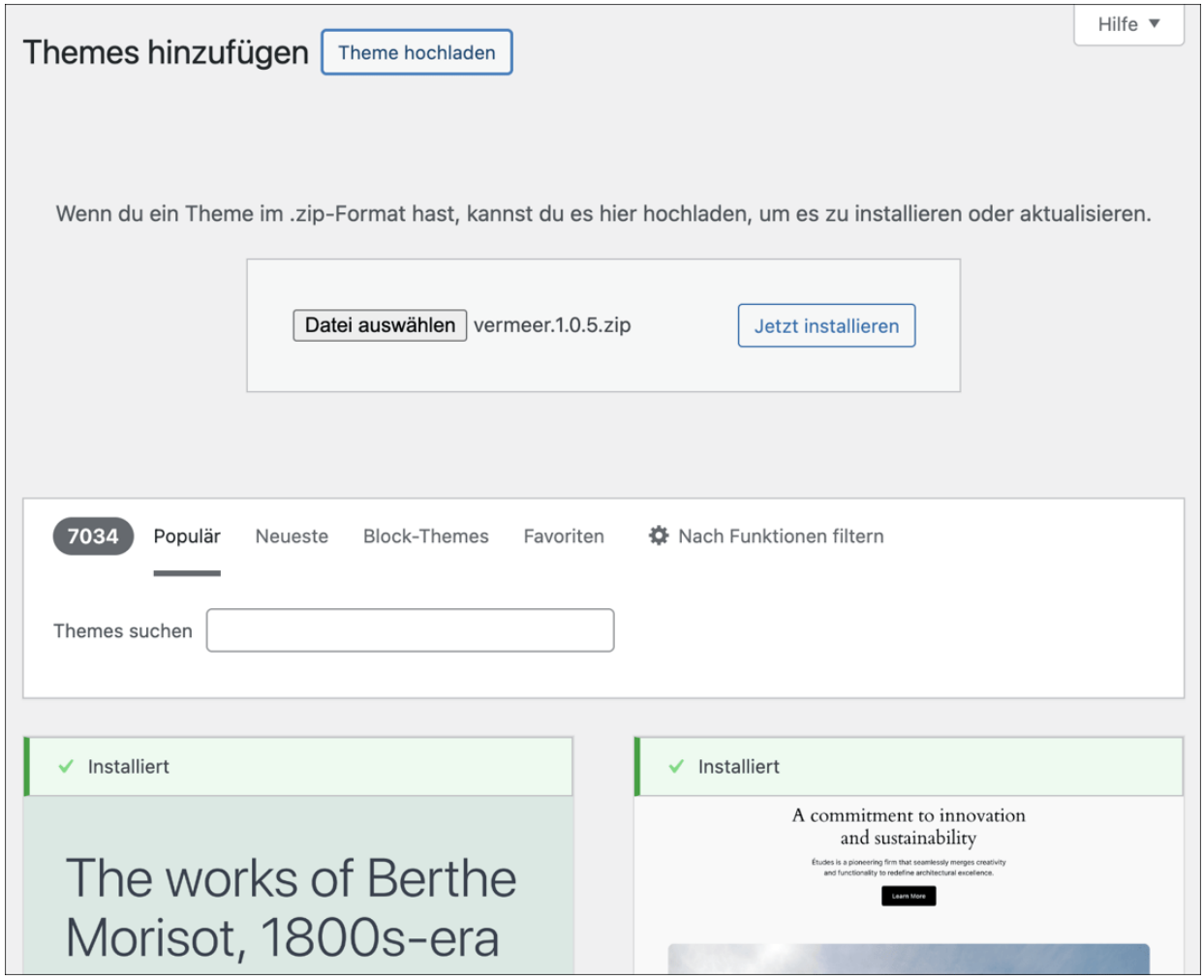


Abbildung 10.58 Ein Theme aus einer anderen Quelle als dem WordPress-Verzeichnis können Sie als ZIP hochladen.

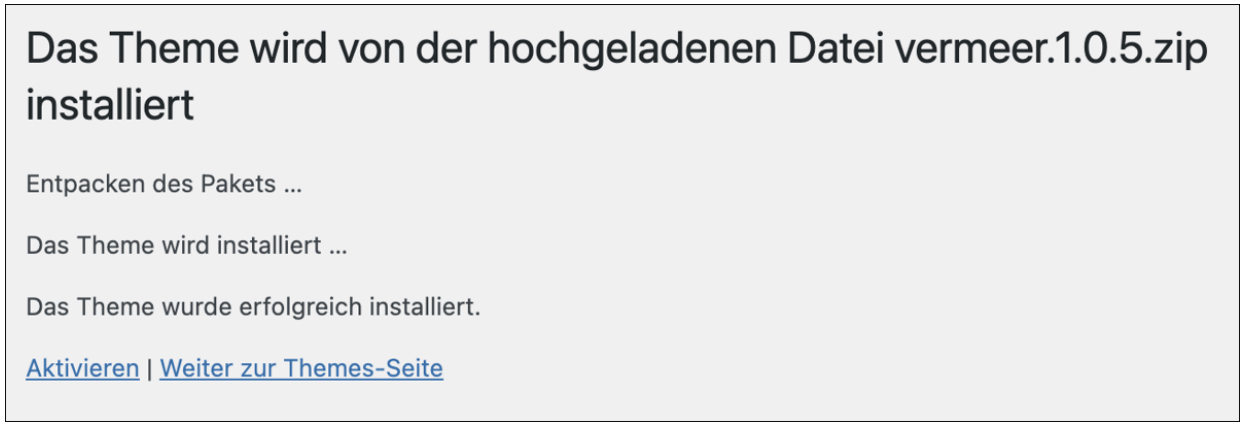


Abbildung 10.59 Bei der Installation durch Hochladen eines ZIP-Archivs werden Ihnen die durchgeführten Schritte angezeigt.

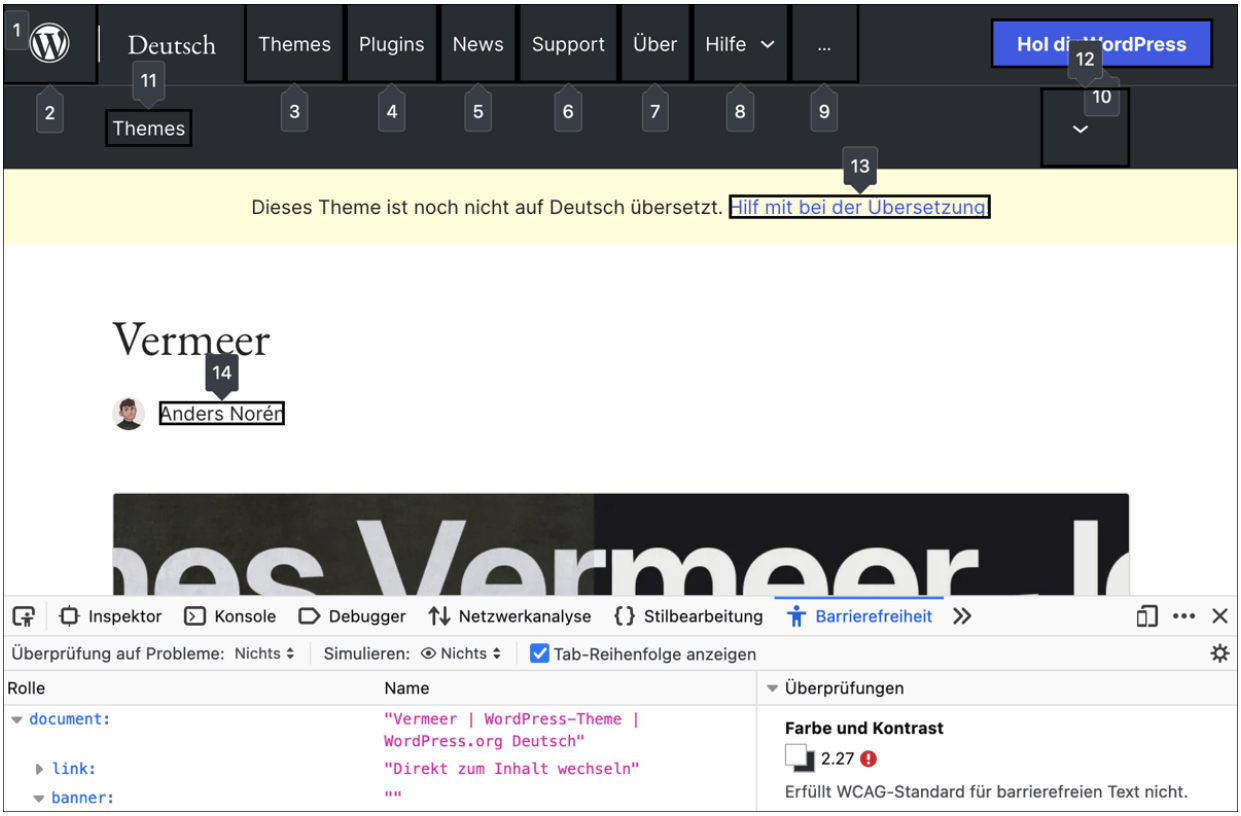


Abbildung 10.60 In Firefox kann mit den Webentwicklertools die Tab-Reihenfolge angezeigt werden, sodass einfach geprüft werden kann, ob sie sinnvoll ist.



Abbildung 10.61 Über das Barrierefreiheit-Werkzeug in den Firefox-Entwicklertools lassen sich Farbkontraste auf einer Website überprüfen.

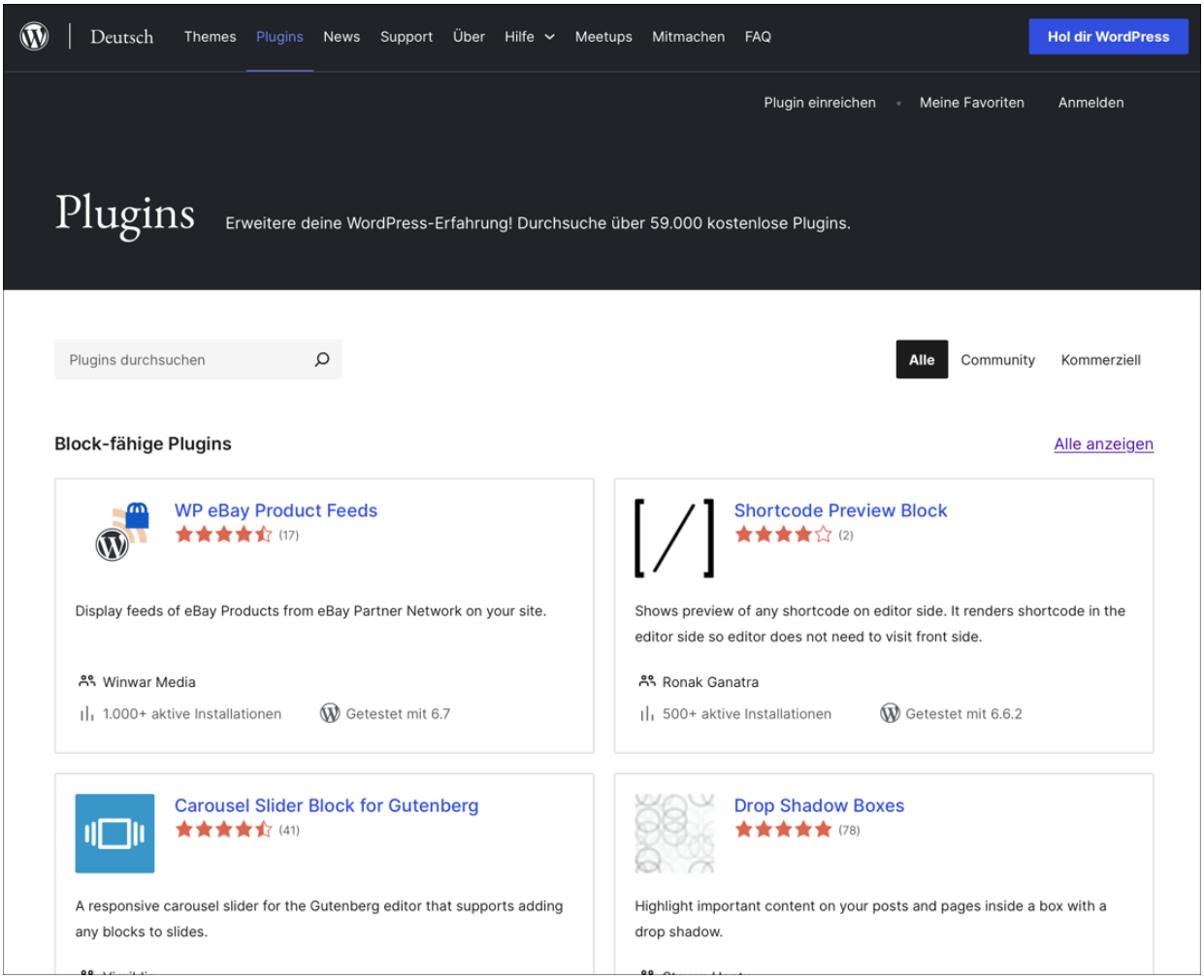


Abbildung 11.1 Im Plugin-Verzeichnis auf WordPress.org finden Sie knapp 60.000 kostenfreie Plugins.

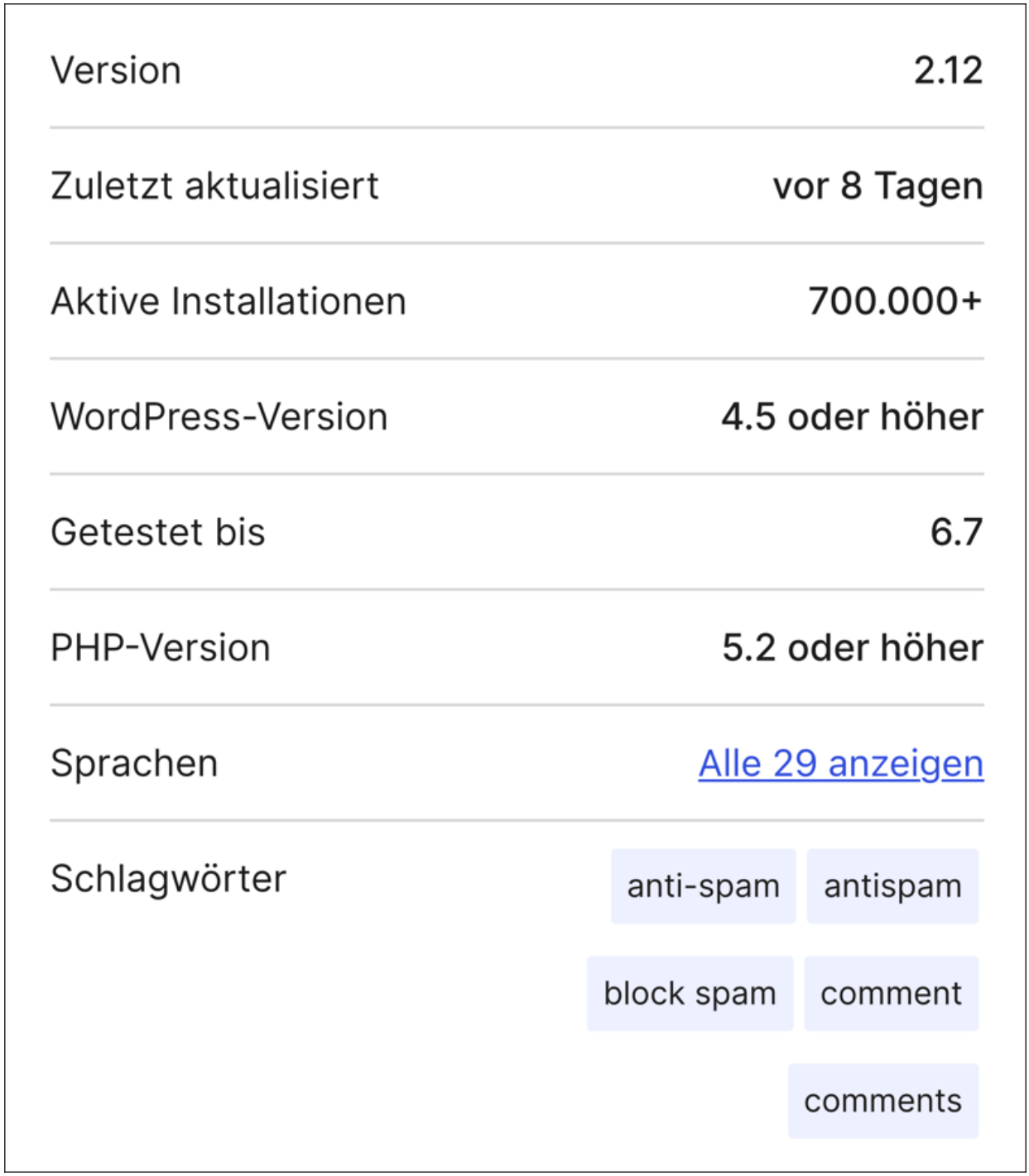


Abbildung 11.2 In der Einzelansicht auf WordPress.org finden Sie hilfreiche Informationen zu jedem Plugin.



Abbildung 11.3 Auf WordPress.org können Sie die Support-Anfragen im öffentlichen Forum jedes Plugins einsehen. Gelöste Anfragen werden mit einem grünen Haken markiert.

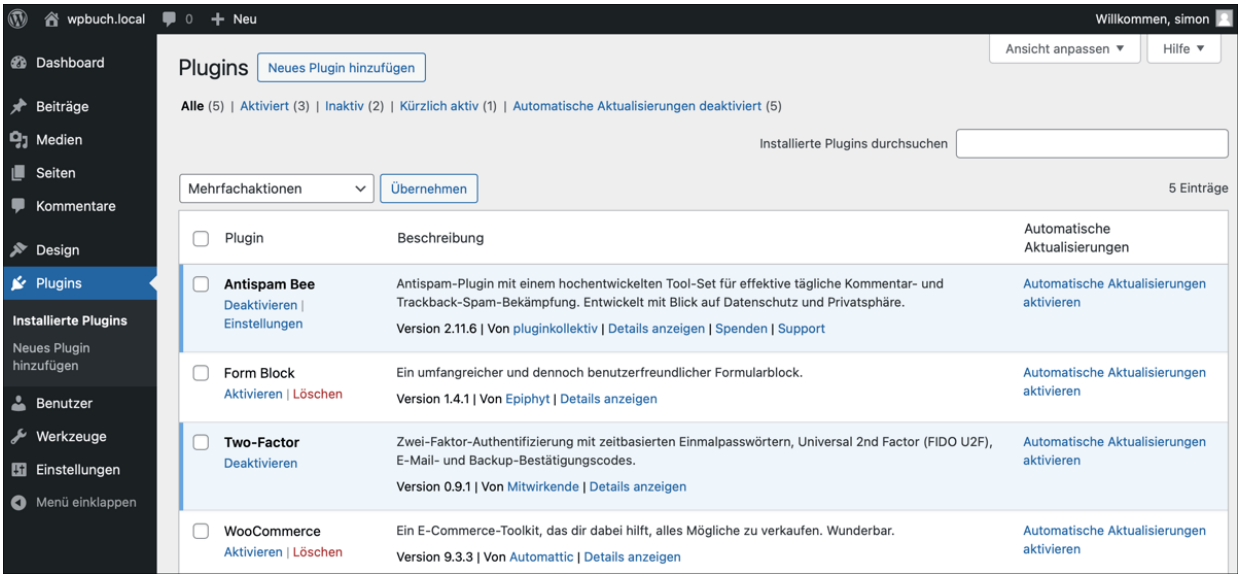


Abbildung 11.4 Auf der Seite »Plugins« erhalten Sie eine Übersicht über die Plugins, die auf Ihrer Website aktuell installiert sind.

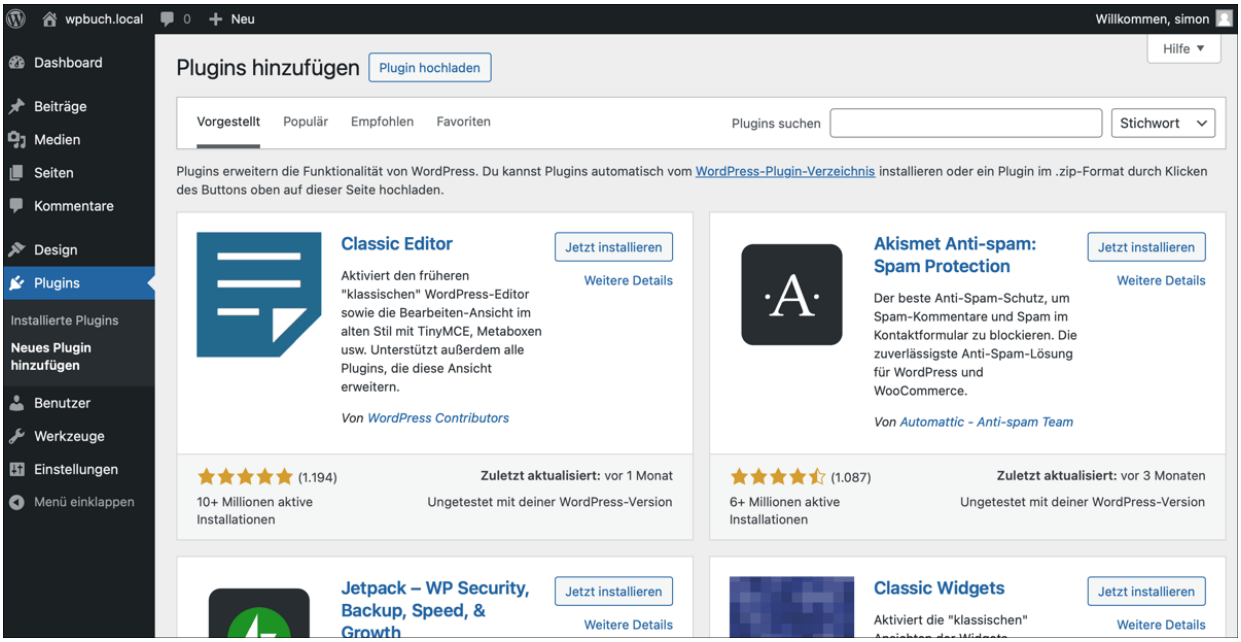


Abbildung 11.5 Unter »Neues Plugin hinzufügen« können Sie nach neuen Plugins suchen und diese installieren.

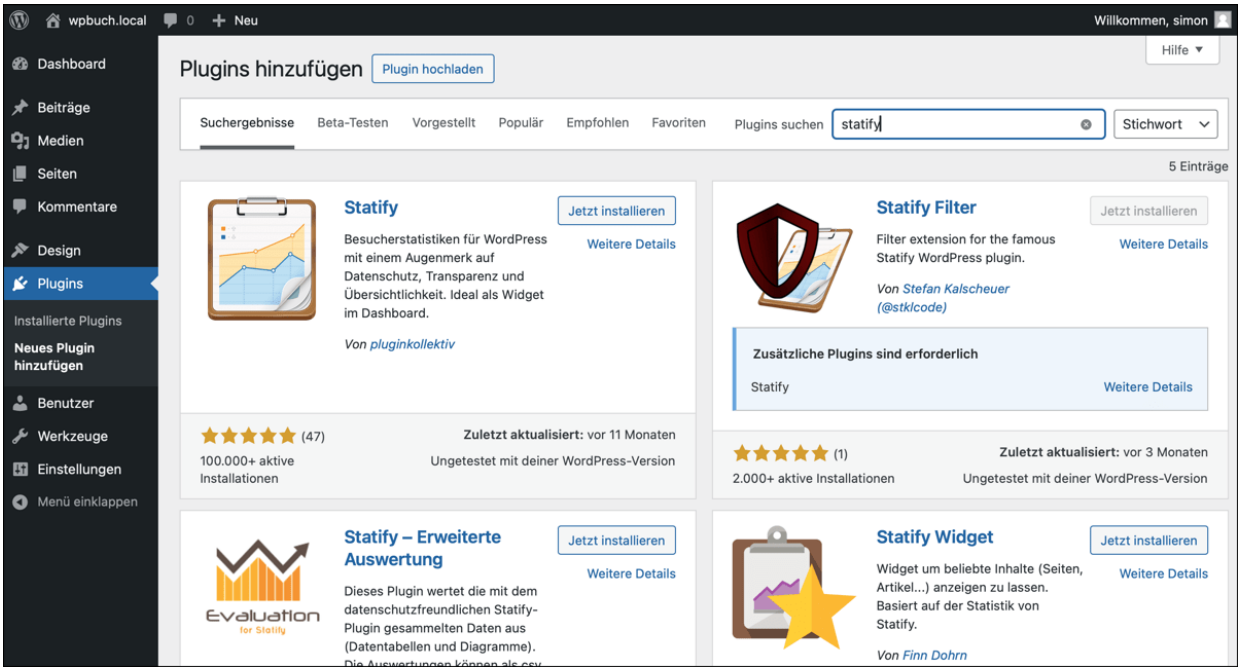


Abbildung 11.6 Über die Suche finden Sie neue Plugins von WordPress.org und können diese direkt installieren und aktivieren.

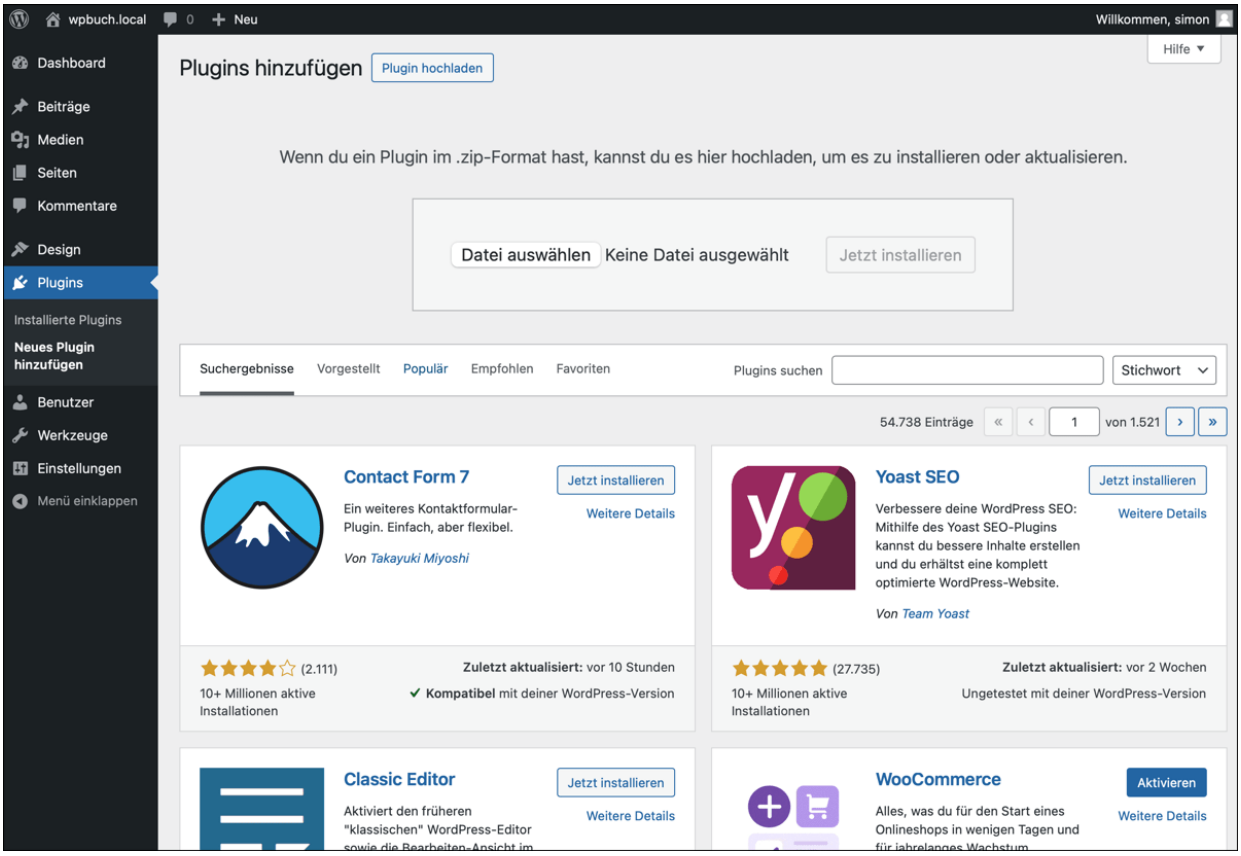


Abbildung 11.7 Im Upload-Dialog können Sie eine ZIP-Datei von Ihrem Computer auswählen und hochladen.

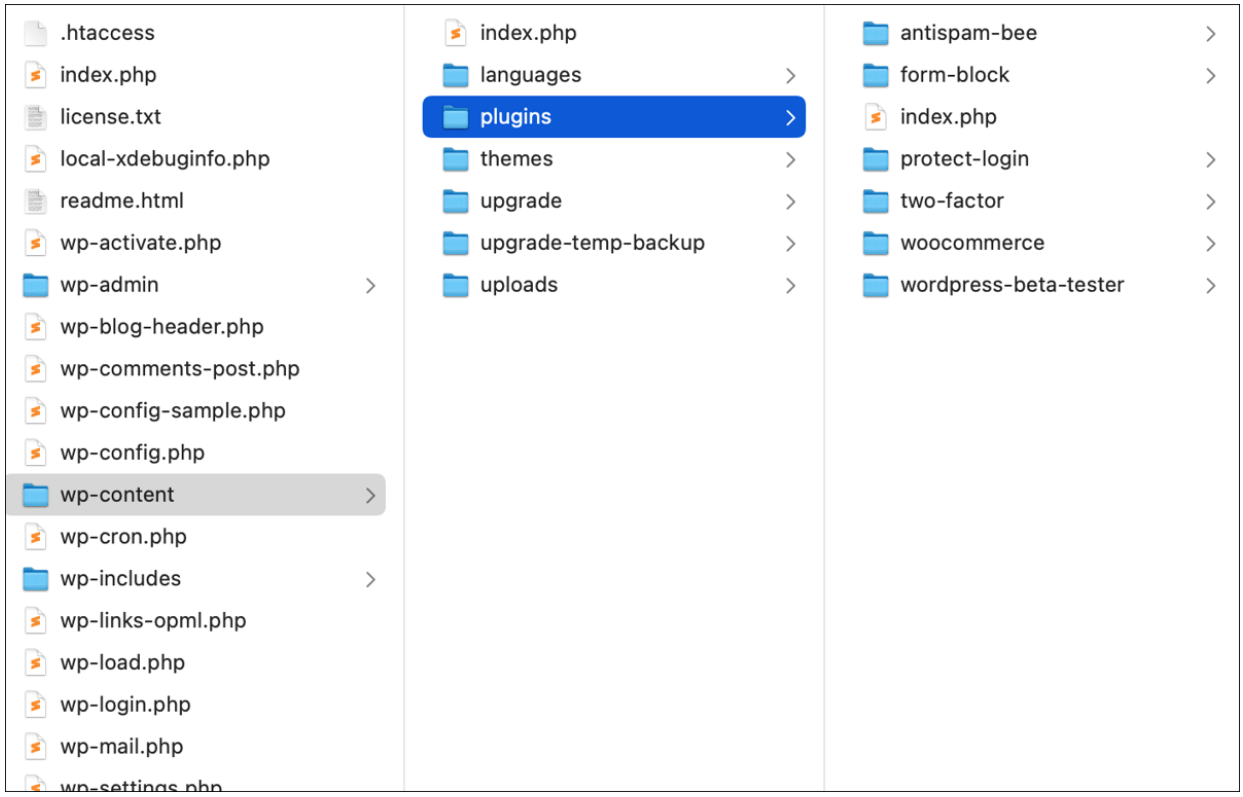


Abbildung 11.8 Im Verzeichnis `/wp-content/plugins/` finden Sie alle installierten Plugins in ihren jeweiligen Unterverzeichnissen.

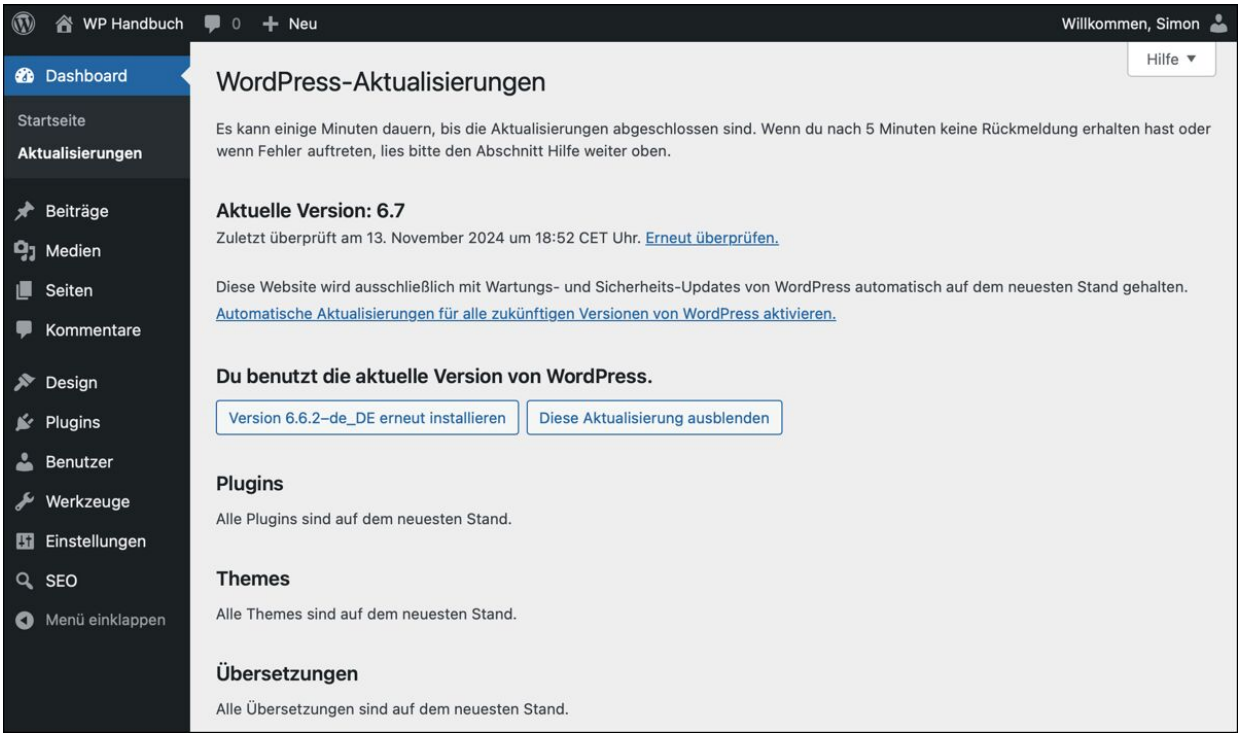


Abbildung 11.9 Unter »Dashboard • Aktualisierungen« erhalten Sie einen schnellen Überblick über alle verfügbaren Updates.

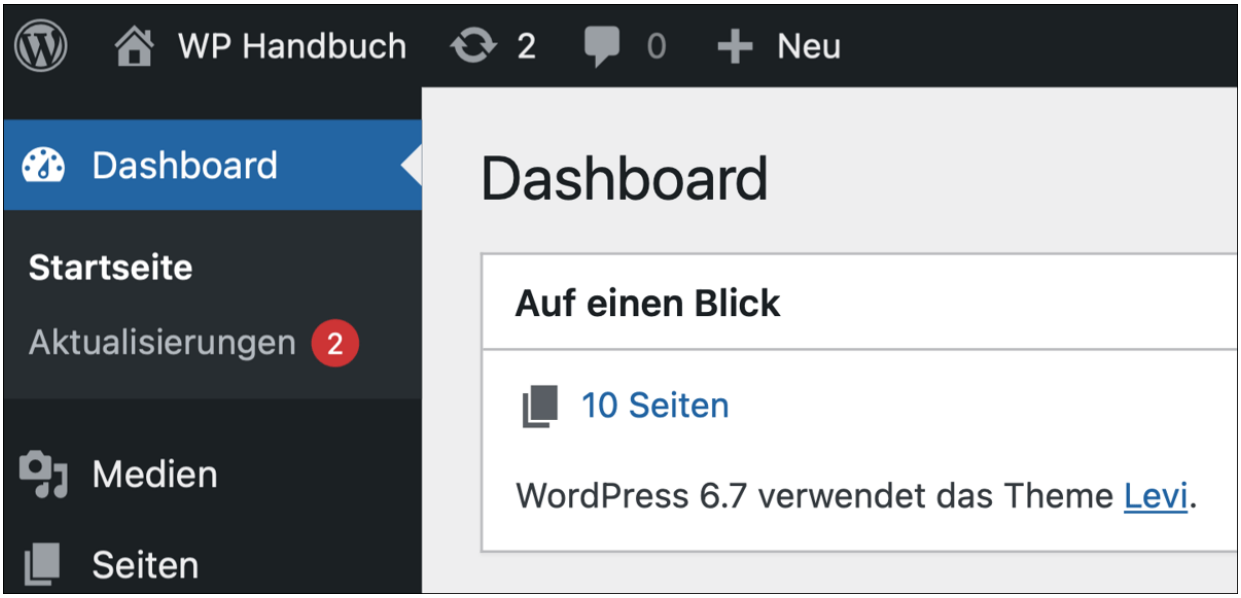


Abbildung 11.10 Wenn Updates verfügbar sind, zeigt WordPress im Admin-Menü und oben in der Werkzeugleiste eine farbig hinterlegte Zahl.



Abbildung 11.11 Im Bereich **PLUGINS** finden Sie die verfügbaren Updates Ihrer installierten Plugins.

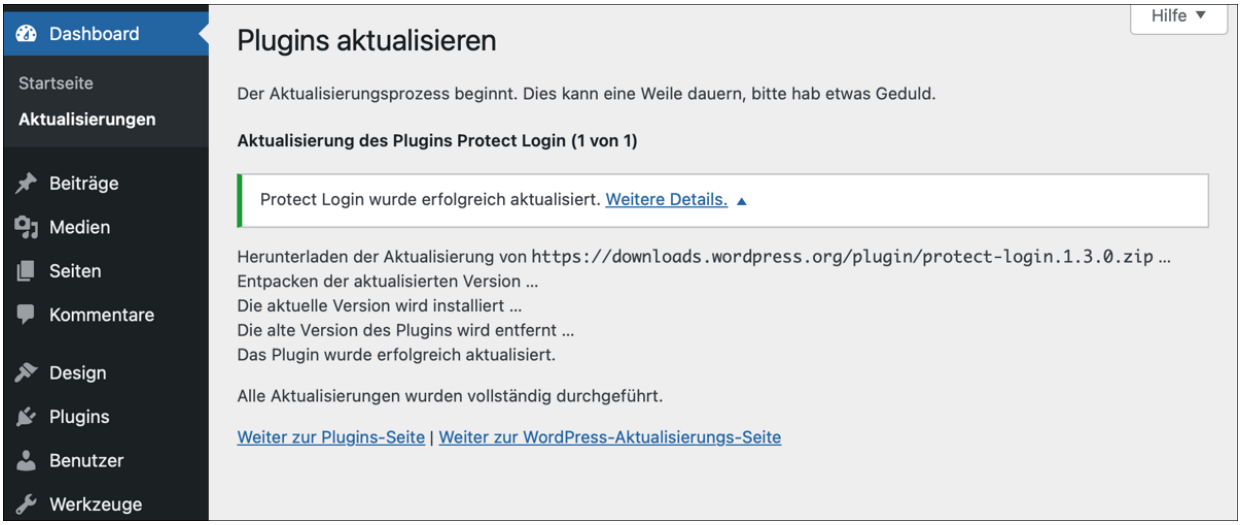


Abbildung 11.12 Im Update-Protokoll können Sie die Schritte nachvollziehen, die WordPress beim Aktualisieren durchgeführt hat.

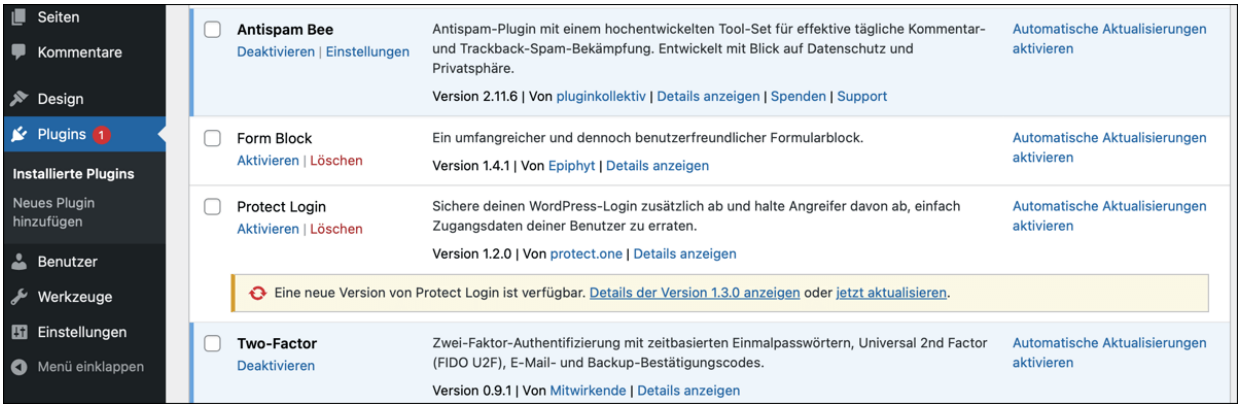


Abbildung 11.13 Auch aus der Plugin-Übersicht können Sie Plugin-Updates direkt ausführen.

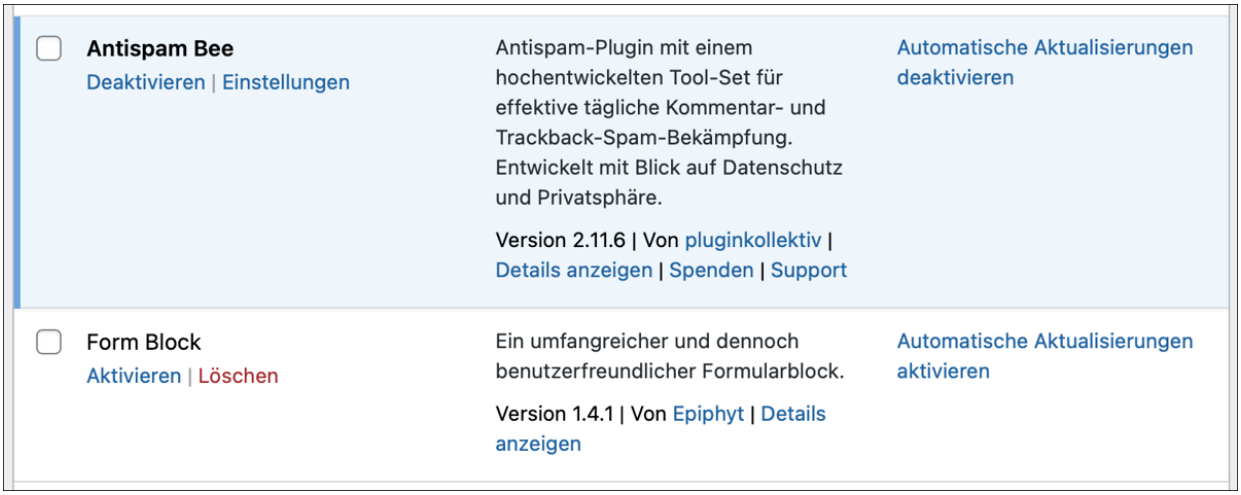


Abbildung 11.14 In der Plugin-Übersicht sehen Sie in einer gesonderten Spalte den Status der **AUTOMATISCHEN AKTUALISIERUNGEN** des jeweiligen Plugins.



Abbildung 11.15 Mit einem (S)FTP-Client können Sie auch manuell Plugin-Updates durchführen. Ob und welche Meldung Sie in diesem Fall angezeigt bekommen, hängt von Ihrem jeweiligen (S)FTP-Client ab.

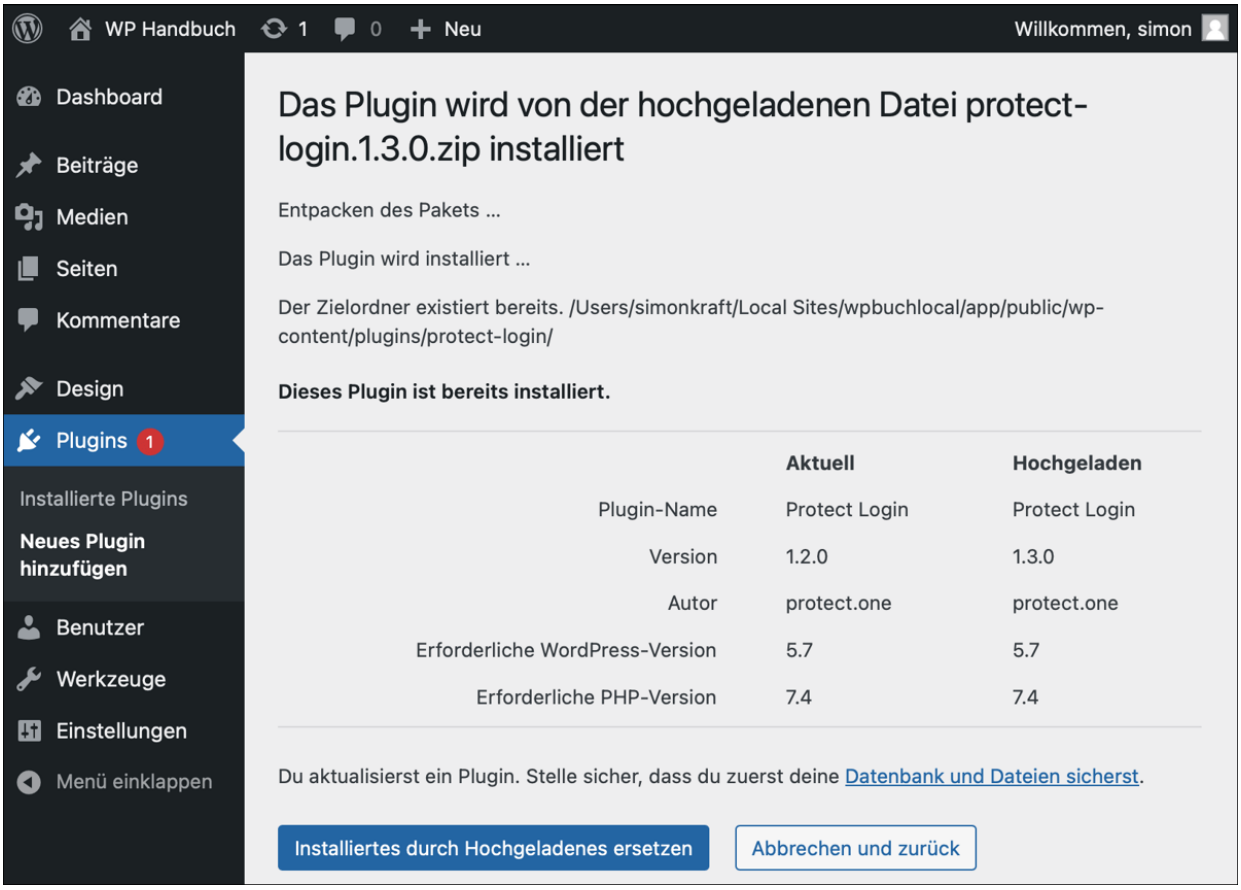


Abbildung 11.16 Wenn WordPress erkennt, dass Sie die neue Version eines bereits installierten Plugins hochgeladen haben, können Sie auch auf diesem Weg ein Update ausführen.



Abbildung 11.17 Nur inaktive Plugins können über das WordPress-Interface deinstalliert werden.

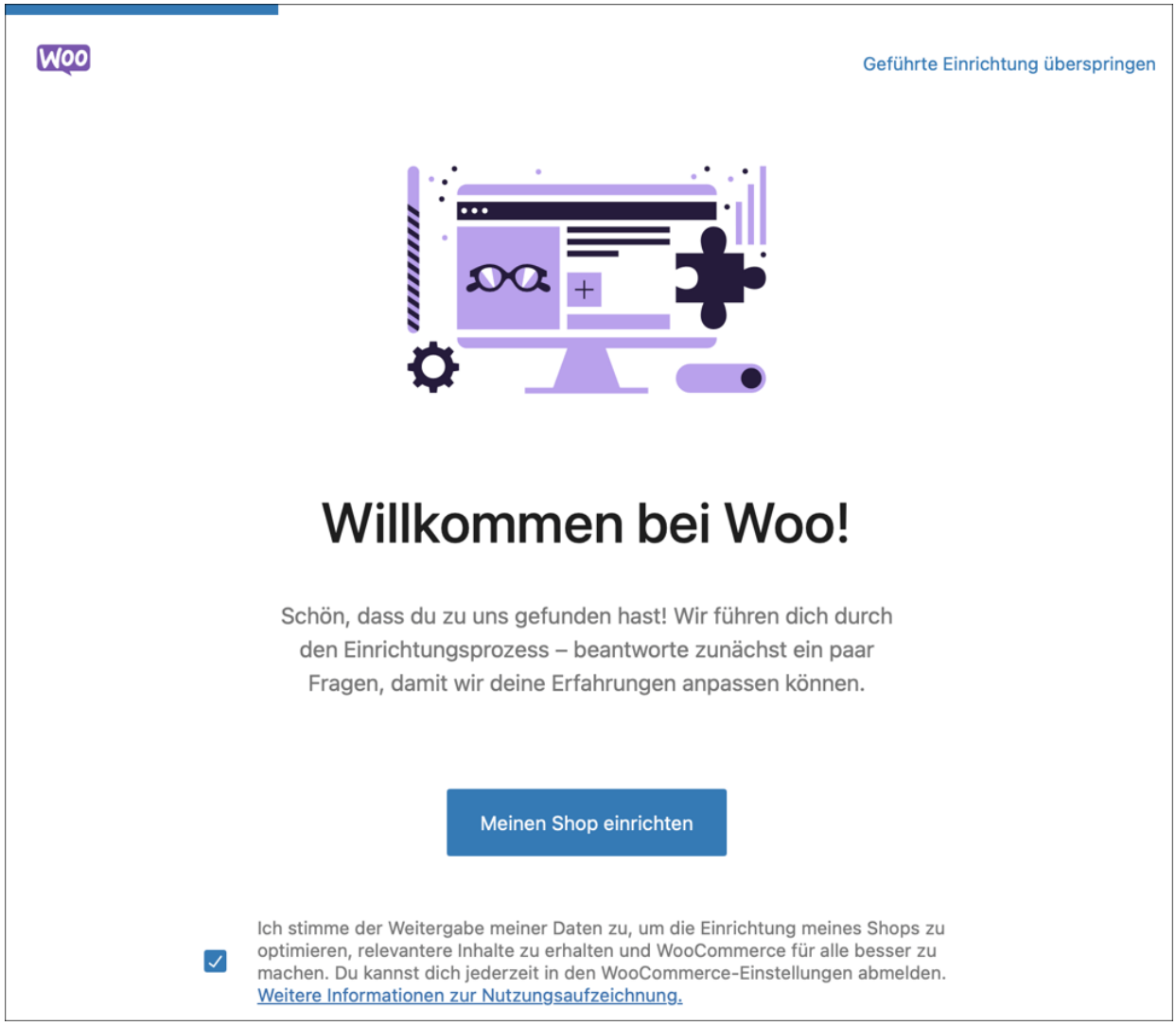


Abbildung 12.1 Nach der Installation von WooCommerce startet der Einrichtungsassistent automatisch.

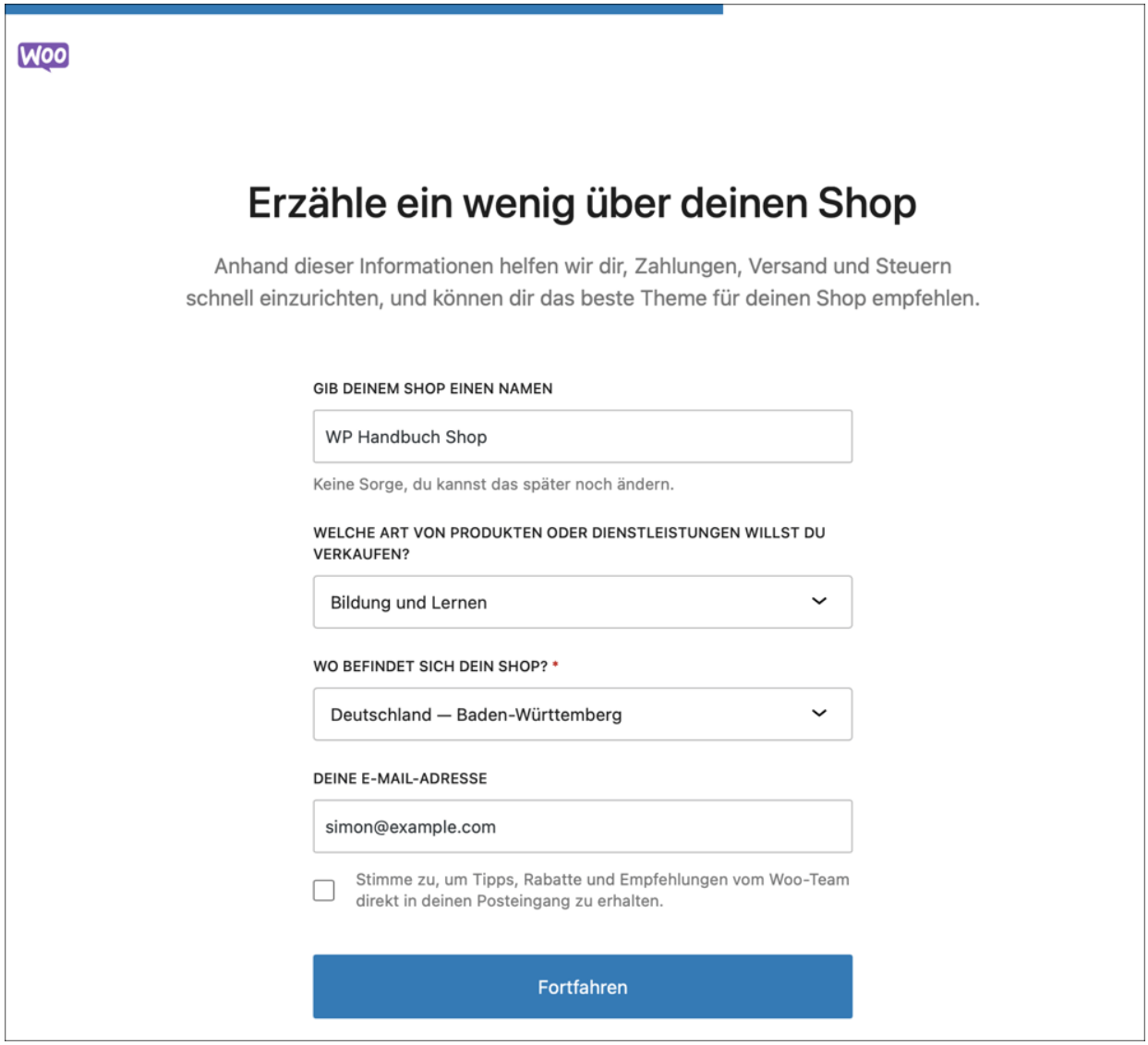


Abbildung 12.2 In welchen Bereich passt Ihr Shop am besten?

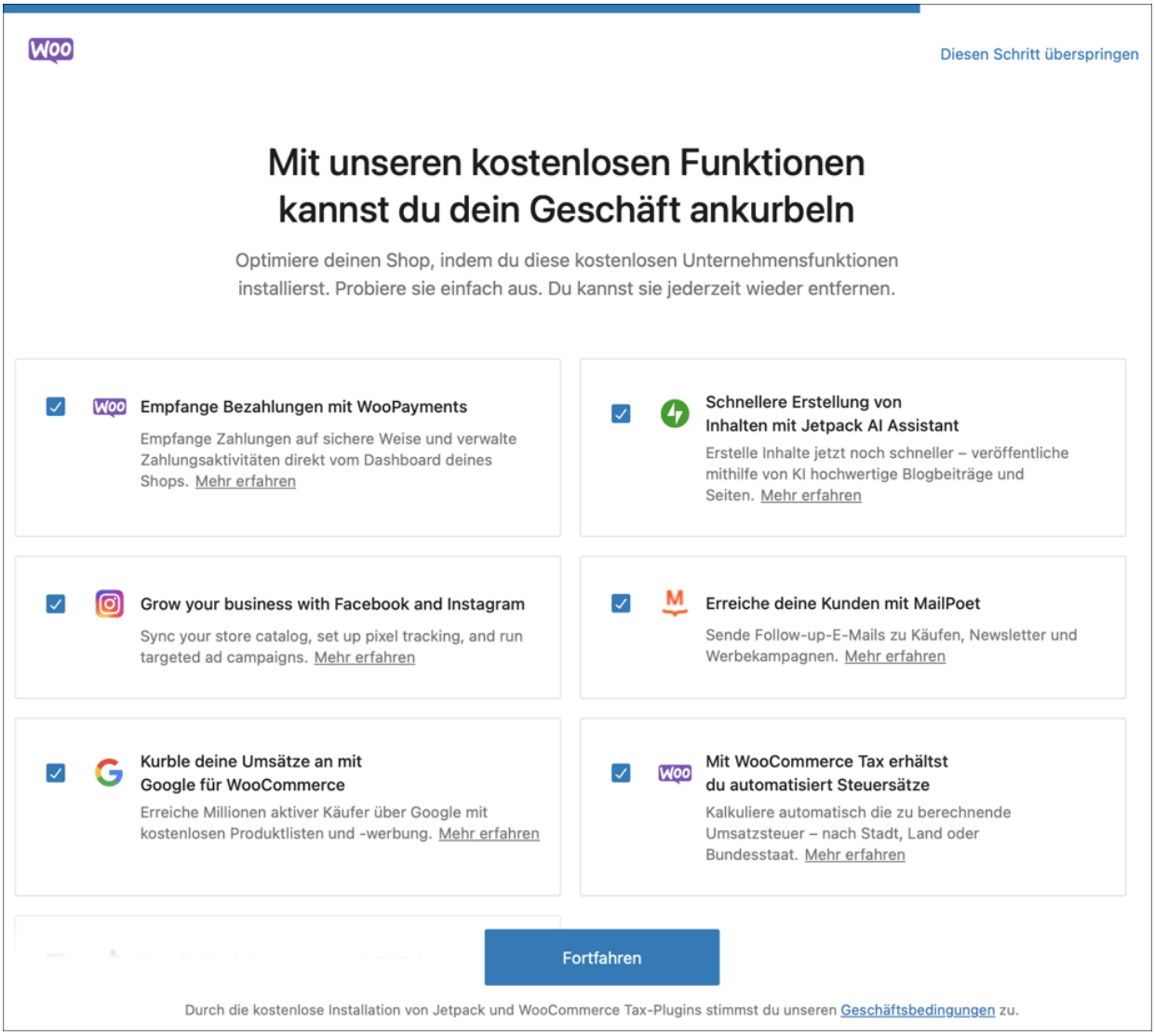


Abbildung 12.3 WooCommerce bietet Ihnen schon bei der Einrichtung eine Reihe kostenfreier Erweiterungen an. Prüfen Sie aber genau, welche davon Sie benutzen dürfen und benutzen wollen, bevor Sie fortfahren.

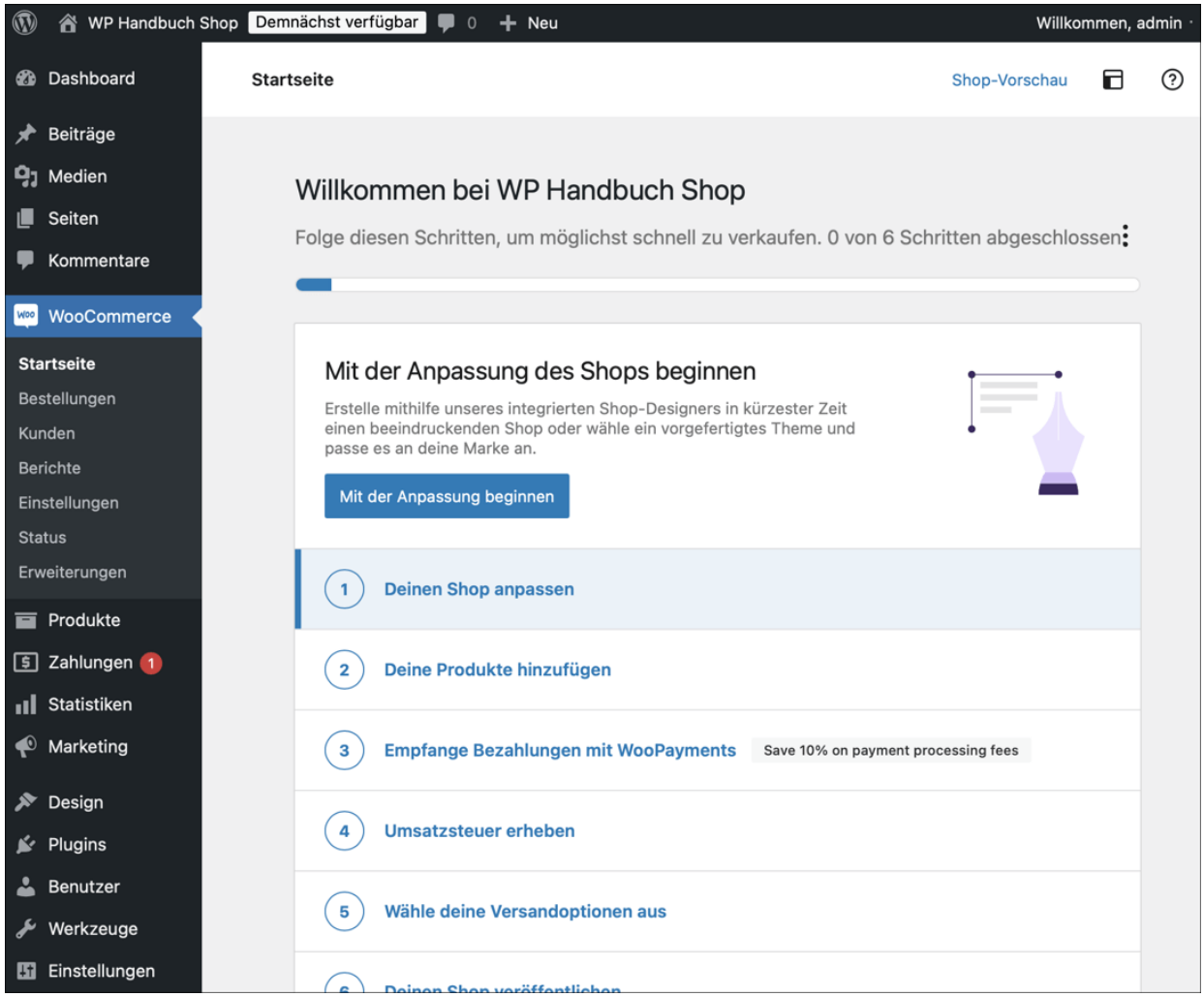


Abbildung 12.4 Auch nach dem Durchlaufen des Einrichtungsassistenten sind Sie mit der Einrichtung Ihres WooCommerce-Shops noch nicht fertig.

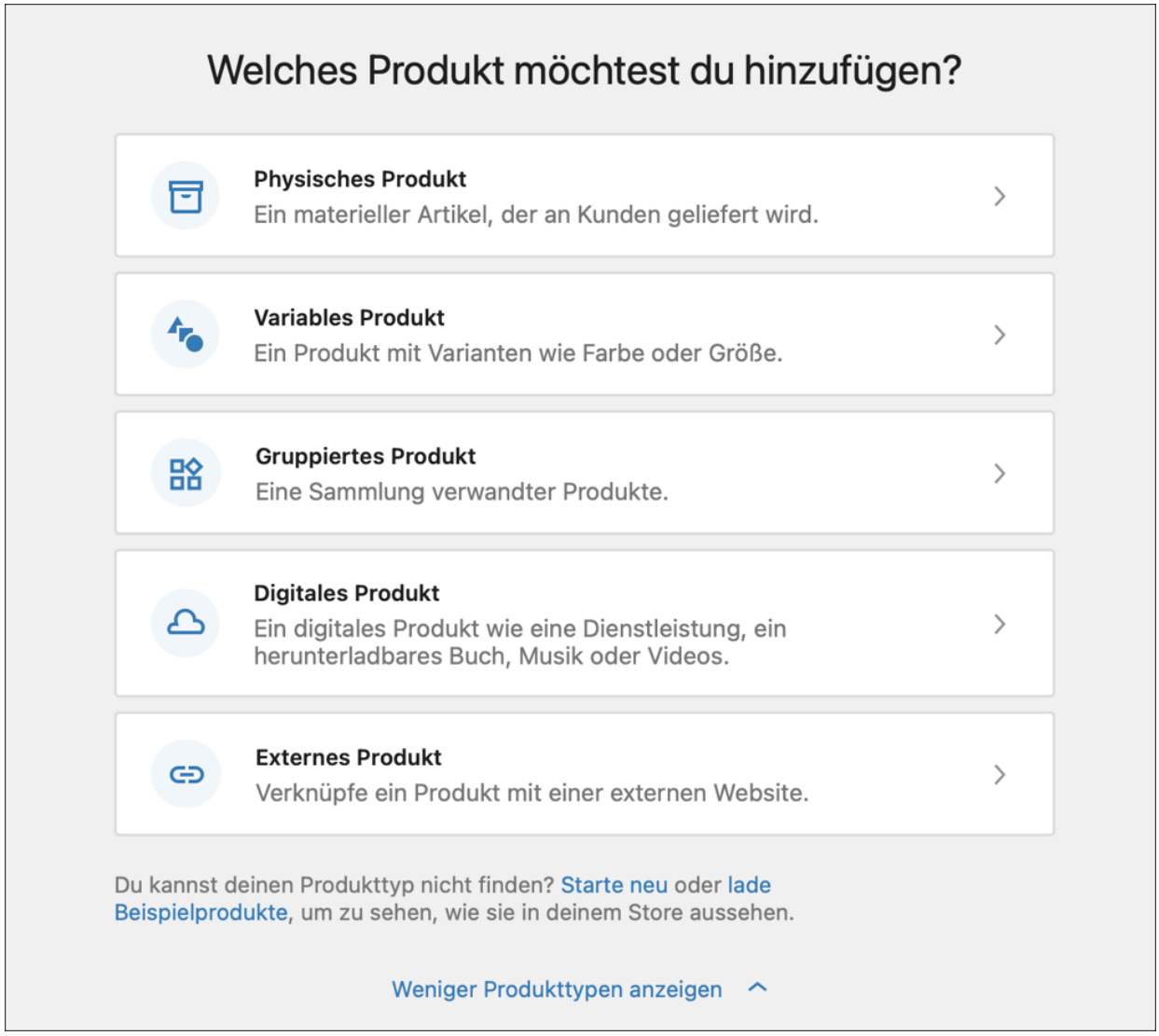


Abbildung 12.5 Sie haben die Wahl, welche Art von Produkt Sie hinzufügen möchten.

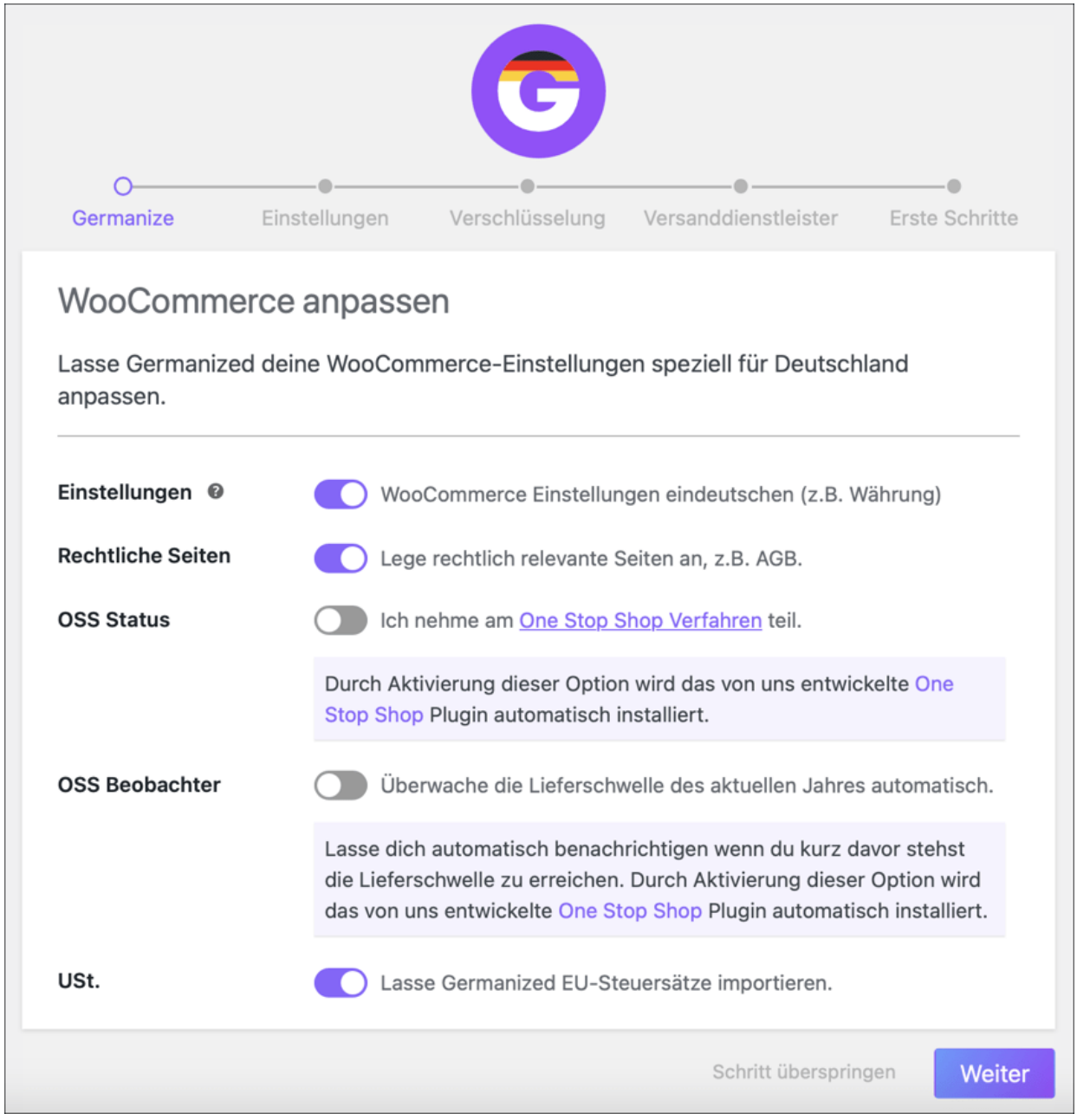


Abbildung 12.6 Auch »Germanized für WooCommerce« führt Sie mit einem Einrichtungsassistenten durch die wichtigsten Schritte.

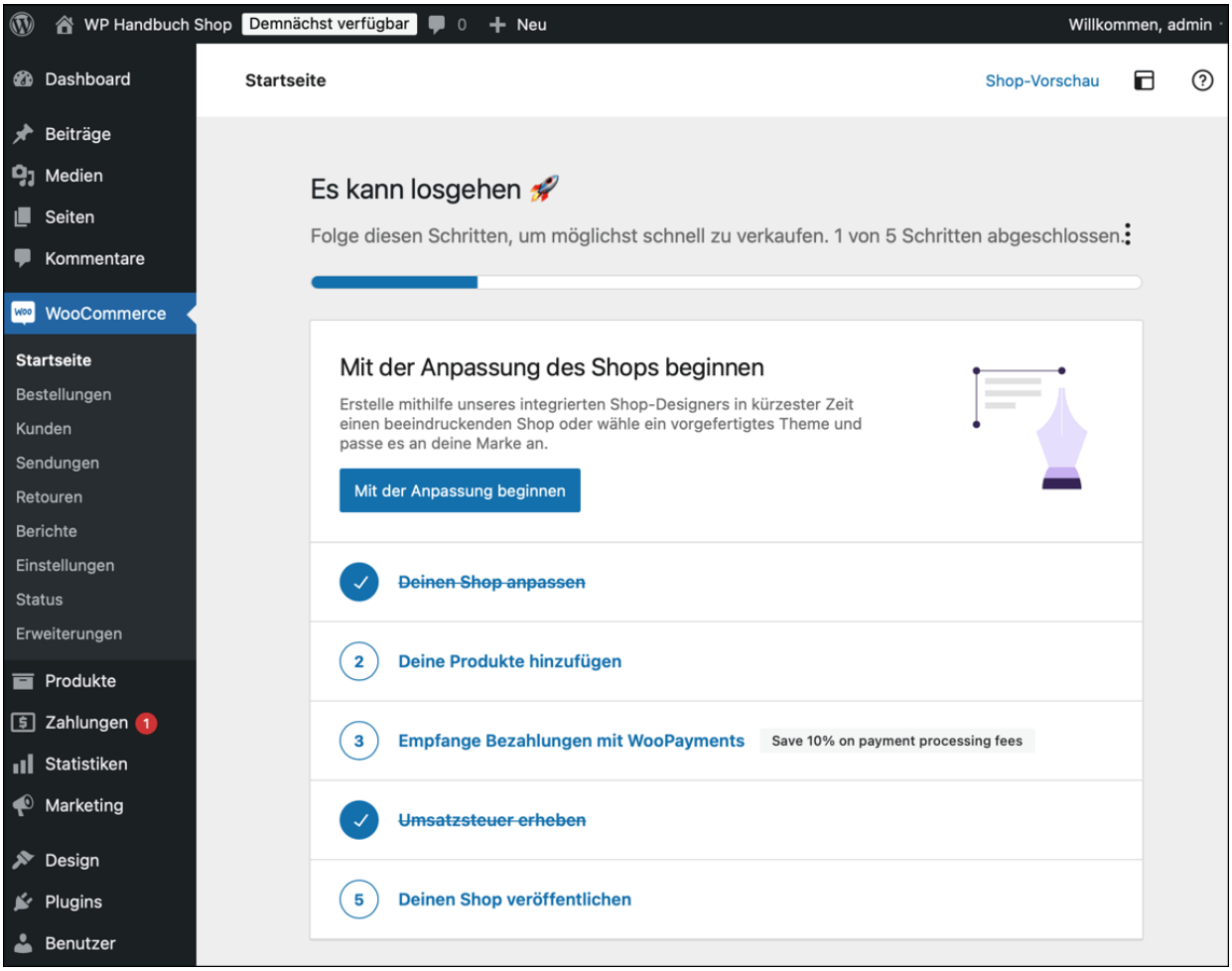


Abbildung 12.7 Die Stichpunkte, mit denen WooCommerce Sie zu leiten versucht, müssen nicht unbedingt in dieser Reihenfolge bearbeitet werden, sind aber eine gute Gedächtnisstütze.

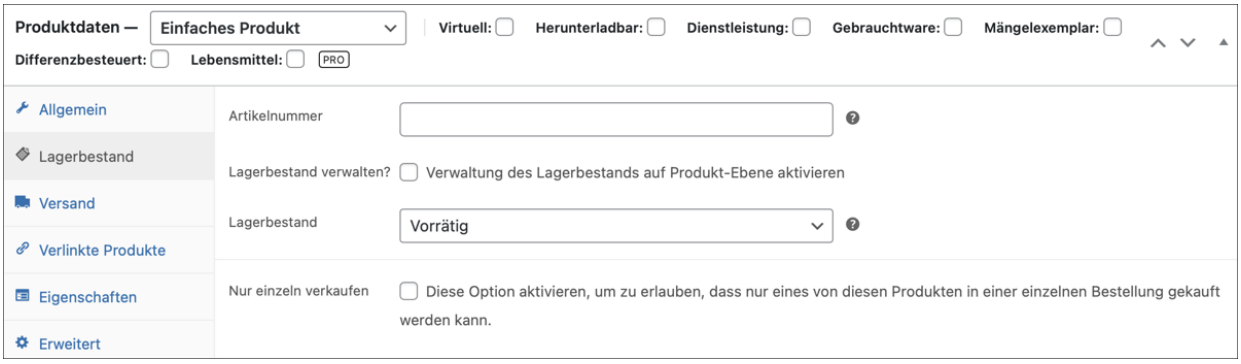


Abbildung 12.8 Die Meta-Box unterhalb des Produkt-Editors enthält die Kontrollelemente für alle wichtigen Metadaten rund um Ihr Produkt.

Name	Titelform	Sortieren nach	Begriffe
Marke	brand	Benutzerdefinierte Sortierung	– Begriffe konfigurieren
Color	color	Benutzerdefinierte Sortierung	Blue Begriffe konfigurieren

Abbildung 12.9 In der Übersicht zu den Produkteigenschaften finden Sie alle gespeicherten Begriffe und können sie bearbeiten.

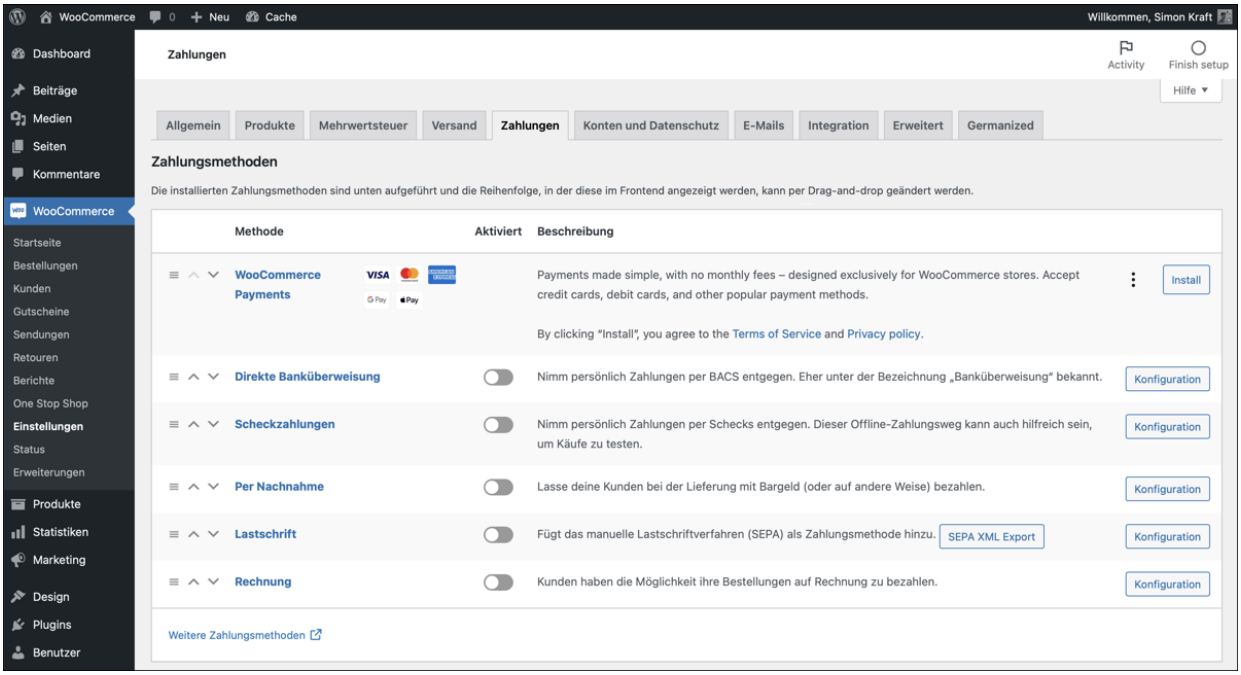


Abbildung 12.10 In einer kompakten Übersicht sehen Sie, welche Zahlungsmethoden in Ihrem Shop aktuell zur Verfügung stehen und welche tatsächlich aktiviert sind.

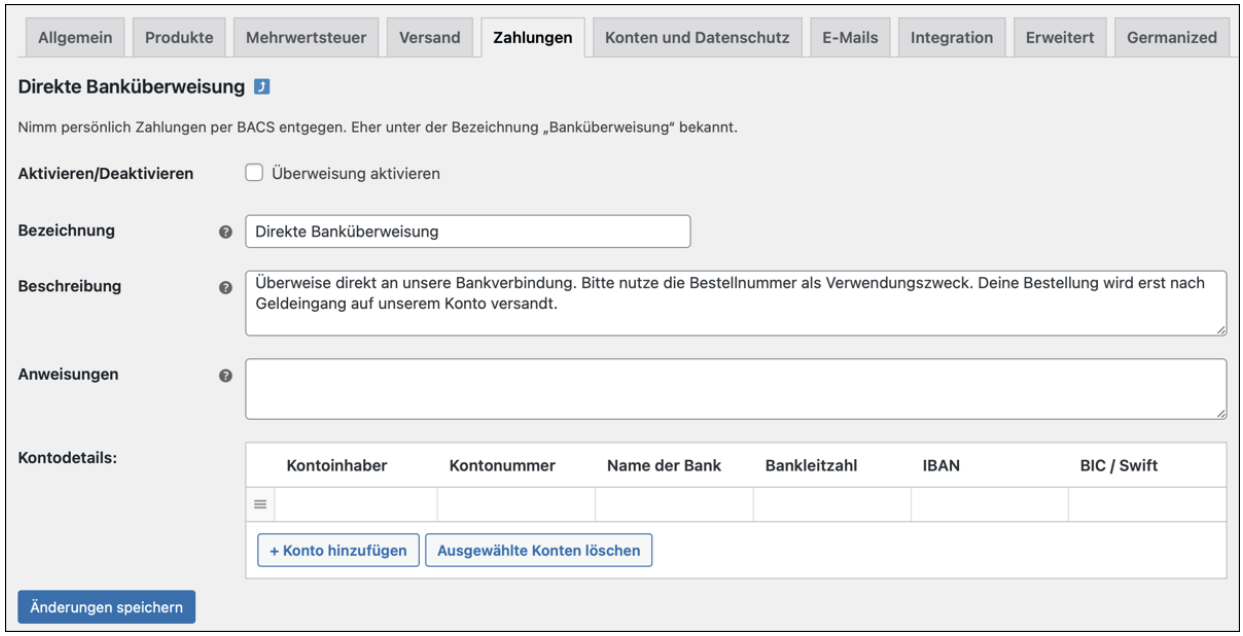


Abbildung 12.11 Konfigurieren können Sie das Überweisungs-Gateway direkt in WooCommerce, um die Abwicklung von eingehenden Zahlungen müssen Sie sich später aber selbst kümmern.

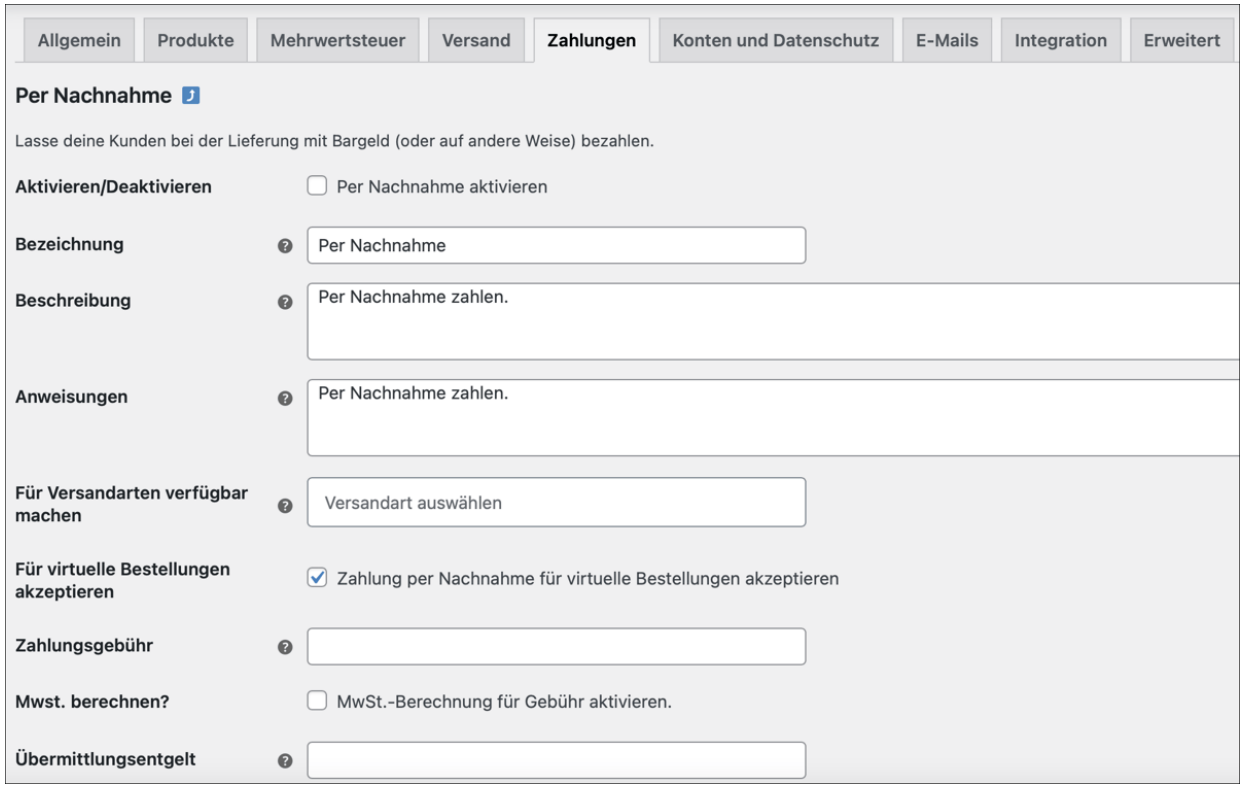


Abbildung 12.12 Vor allem im deutschsprachigen Raum ist das Zahlen per Nachnahme über den Versanddienstleister immer noch ein gängiger Weg.

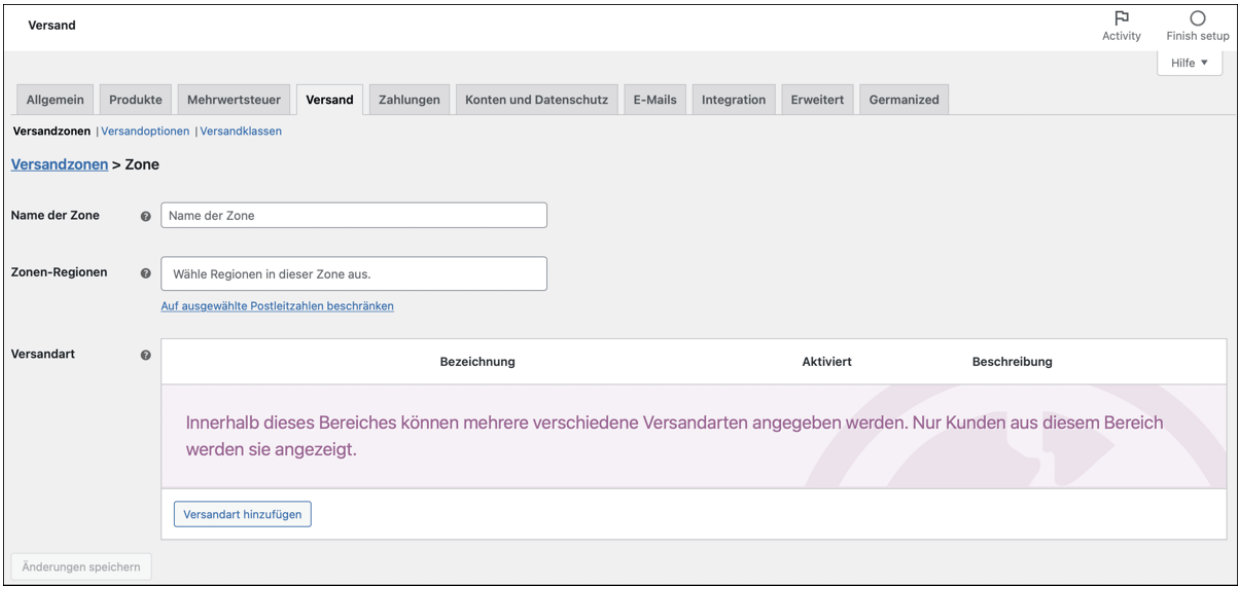


Abbildung 12.13 Als mehr oder weniger leere Seite präsentiert sich die Versandzonenverwaltung bei Ihrem ersten Besuch. Doch mit dieser unscheinbaren Funktion können Sie auch komplexere Versandszenarien abdecken.

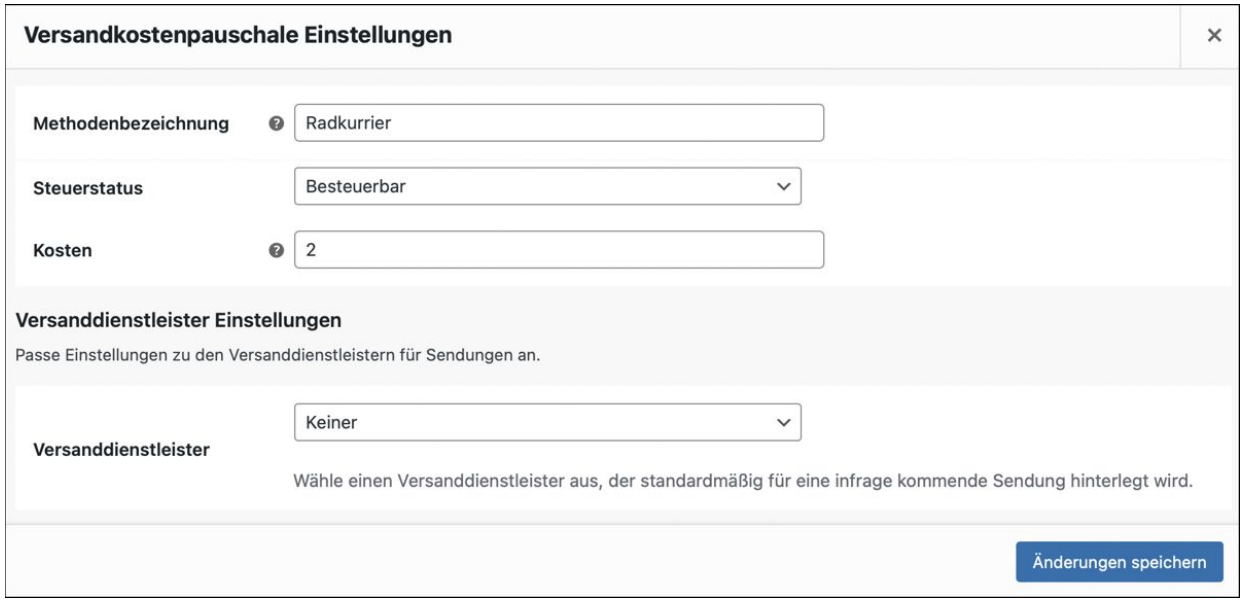


Abbildung 12.14 Mit einigen Anpassungen ist die pauschale Versandmethode erstaunlich flexibel einsetzbar.

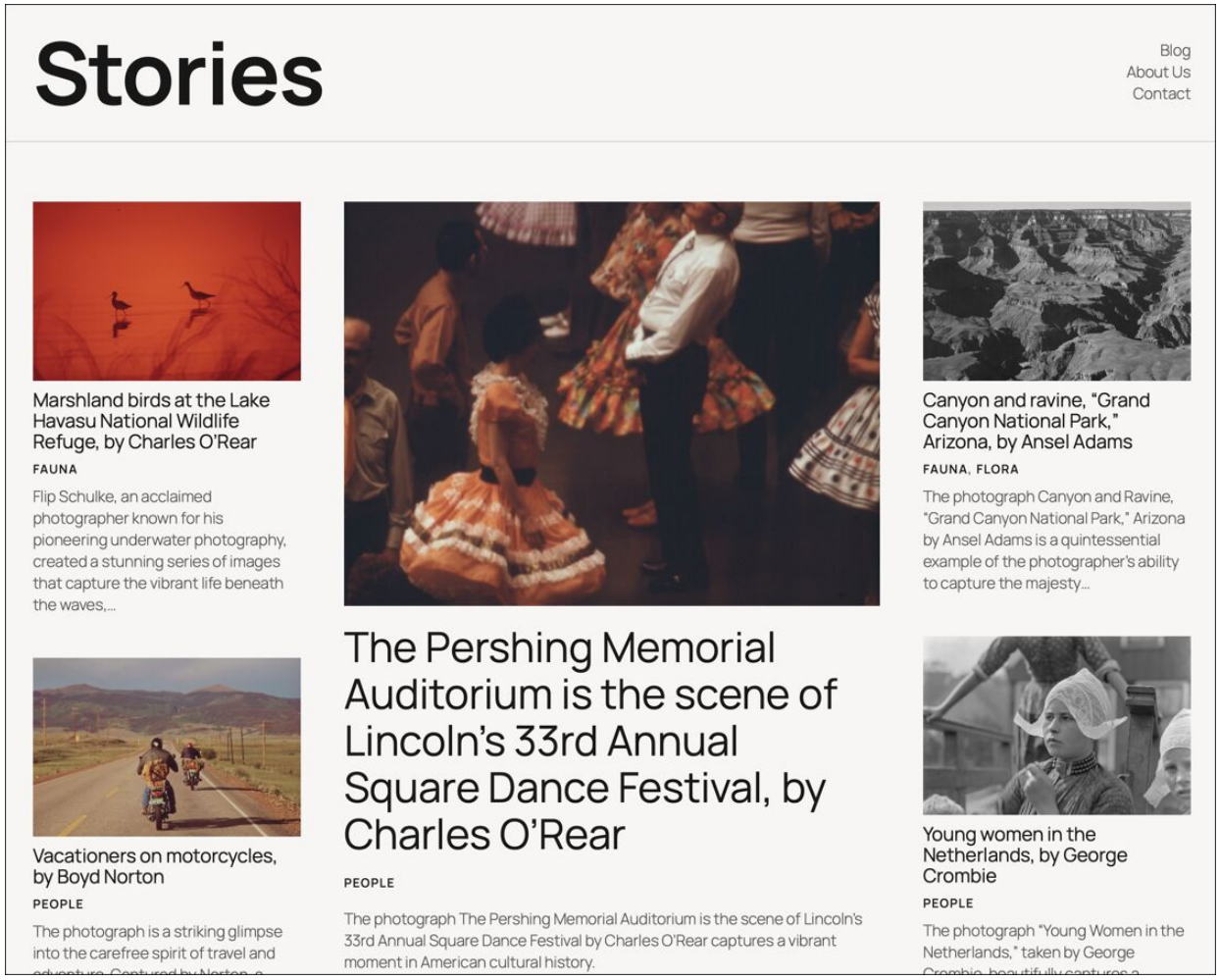


Abbildung 13.1 Das Standard-Theme »Twenty Twenty-Five« hat einen klaren Fokus auf Blogs.



Abbildung 13.2 Der Header-Bereich des Themes »Twenty Twenty-Five« Themes glänzt durch Schlichtheit.

Marshland birds at the Lake Havasu National Wildlife Refuge, by Charles O'Rear



Written by Fiona Presley in Fauna

Flip Schulke, an acclaimed photographer known for his pioneering underwater photography, created a stunning series of images that capture the vibrant life beneath the waves, including those taken at John Pennekamp Coral Reef State Park, located a few miles off Key Largo. Schulke's work in this series is characterized by its breathtaking clarity and composition, revealing the intricate beauty of coral reefs and the diverse marine life that inhabits them. The photographs taken in John Pennekamp Coral Reef State Park are particularly notable for their vivid depiction of the underwater ecosystem, showcasing colorful coral formations

Abbildung 13.3 Der Inhaltsbereich in »Twenty Twenty-Five« setzt für Blogbeiträge auf ein einspaltiges Design. Für komplexere Seiten sind aber auch aufwendigere Vorlagen enthalten.

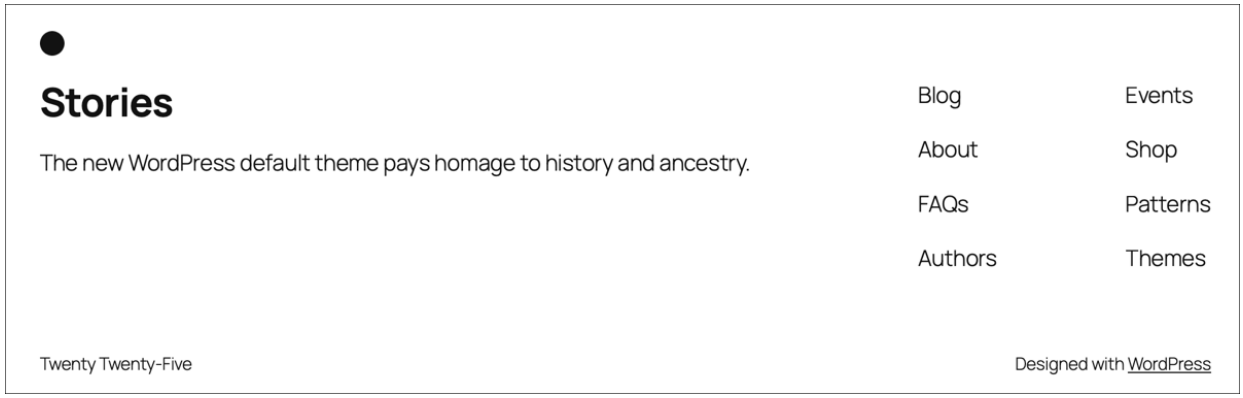


Abbildung 13.4 In der einfachsten Form enthält der Footer neben Namen und Beschreibung des Blogs einfach ein paar Links zur Navigation. Dank des Website-Editors können Sie Ihrer Fantasie hier aber auch freien Lauf lassen und andere Blöcke einfügen.

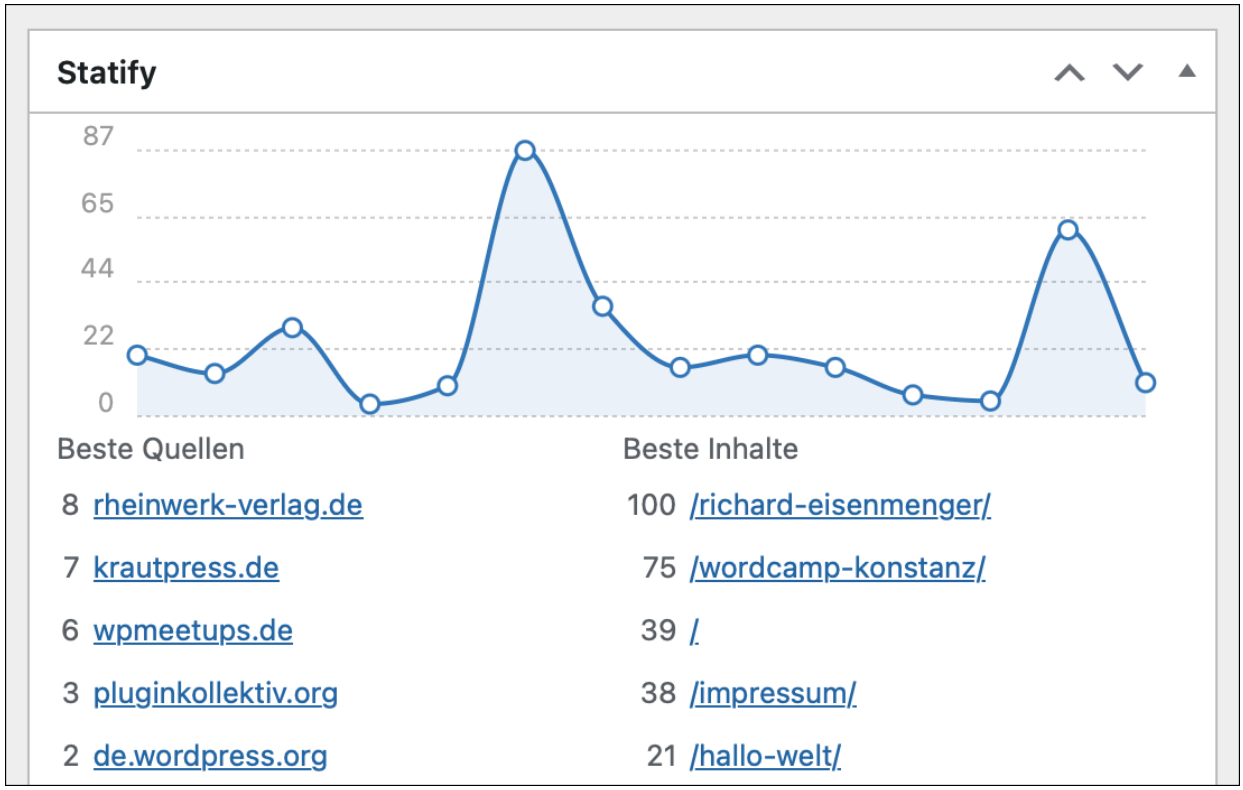


Abbildung 13.5 Statify zeigt die Statistik über die Zugriffszahlen als Dashboard-Widget an.

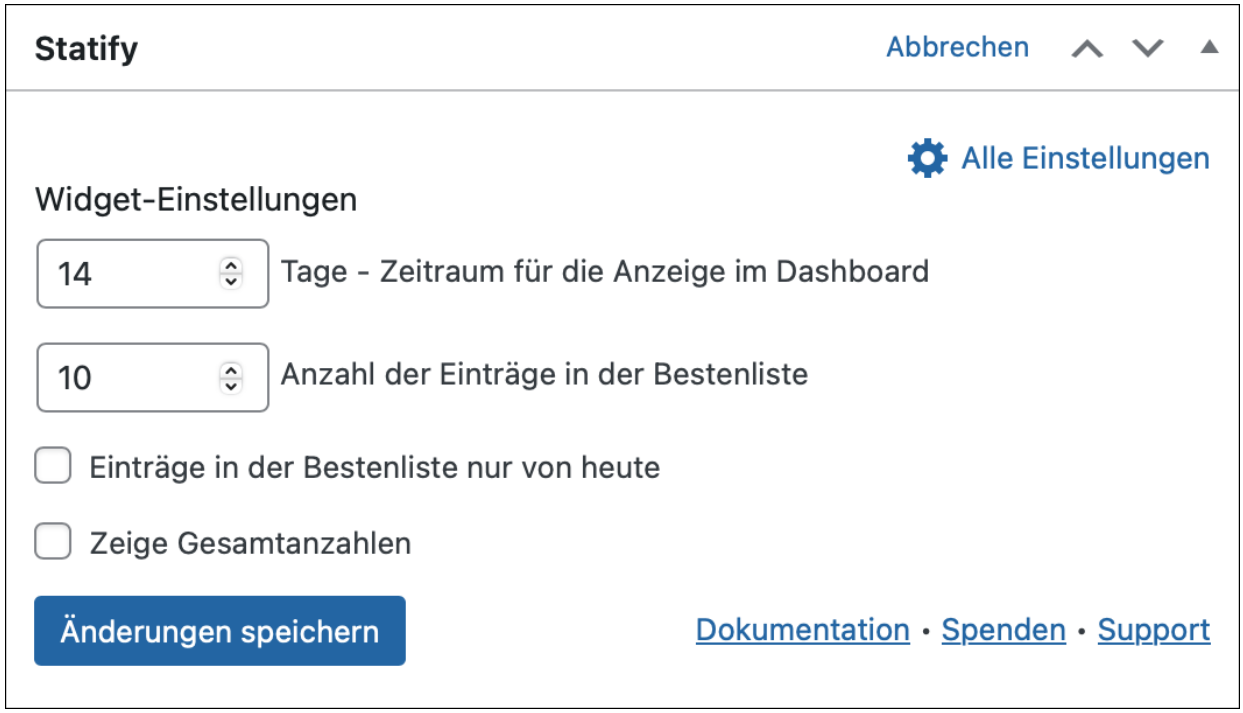


Abbildung 13.6 Sie können einige Einstellungen direkt im Widget verändern.

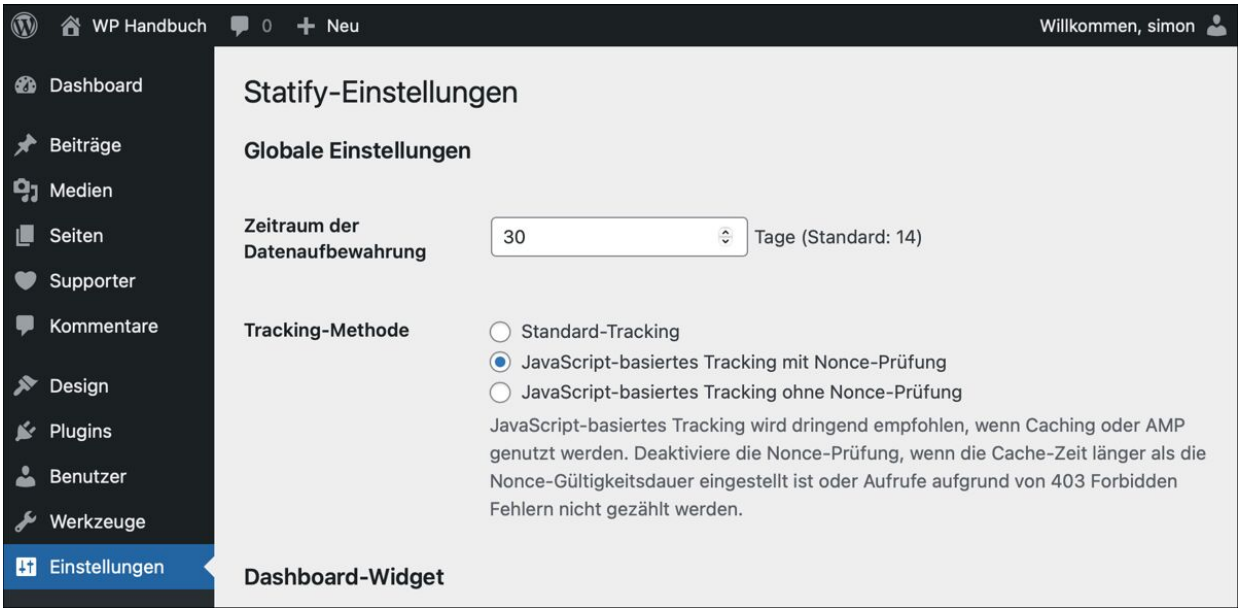


Abbildung 13.7 Die globalen Einstellungen auf der Statify-Einstellungsseite

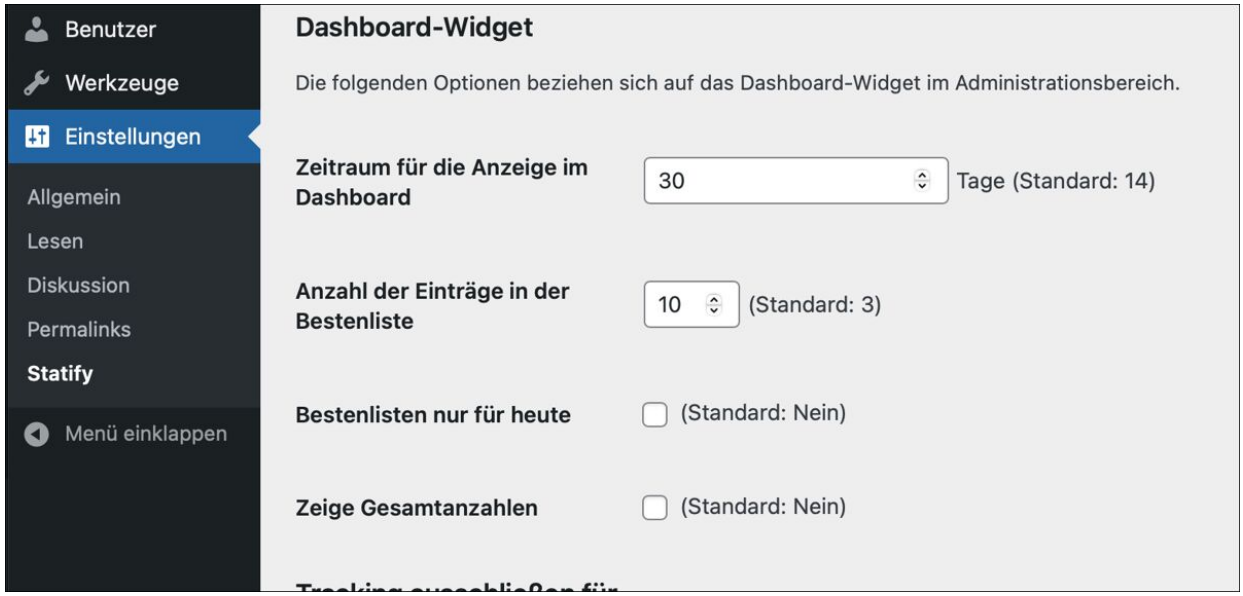


Abbildung 13.8 Die Einstellungen für das Dashboard-Widget

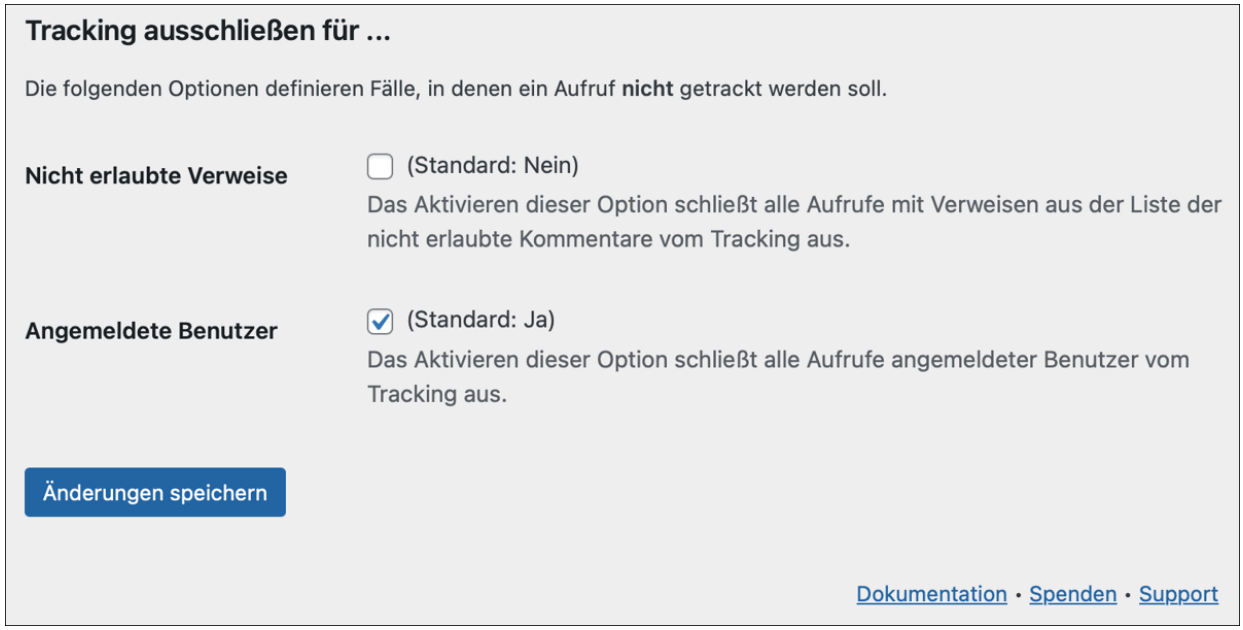


Abbildung 13.9 Optionen zum Ausschließen vom Tracking

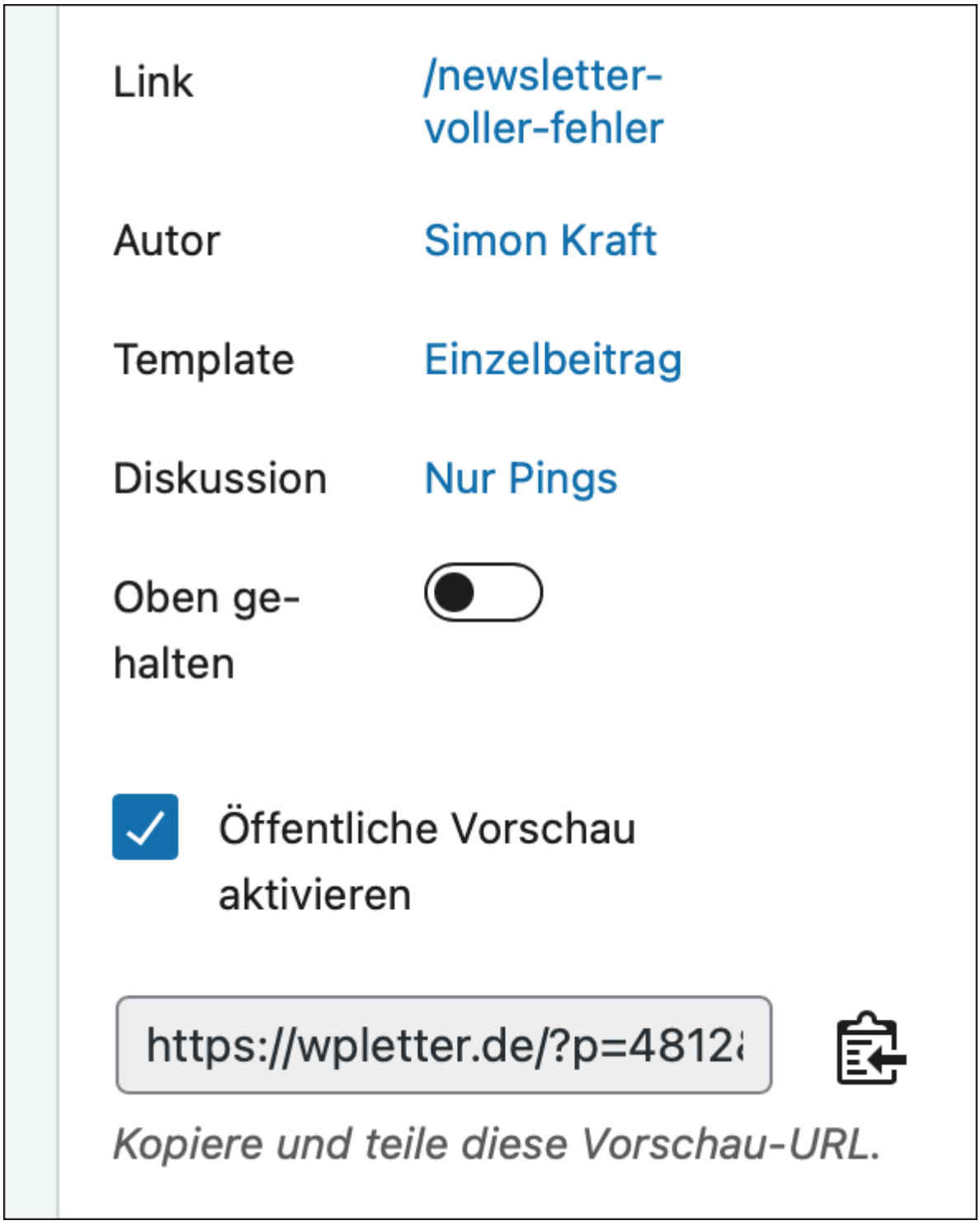

Link
/newsletter-voller-fehler
Autor
Simon Kraft
Template
Einzelbeitrag
Diskussion
Nur Pings
Oben gehalten
Öffentliche Vorschau aktivieren
https://wpletter.de/?p=4812
Kopiere und teile diese Vorschau-URL.

Abbildung 13.10 Das Plugin Public Post Review ermöglicht Ihnen das öffentliche Teilen einer Beitragsvorschau.

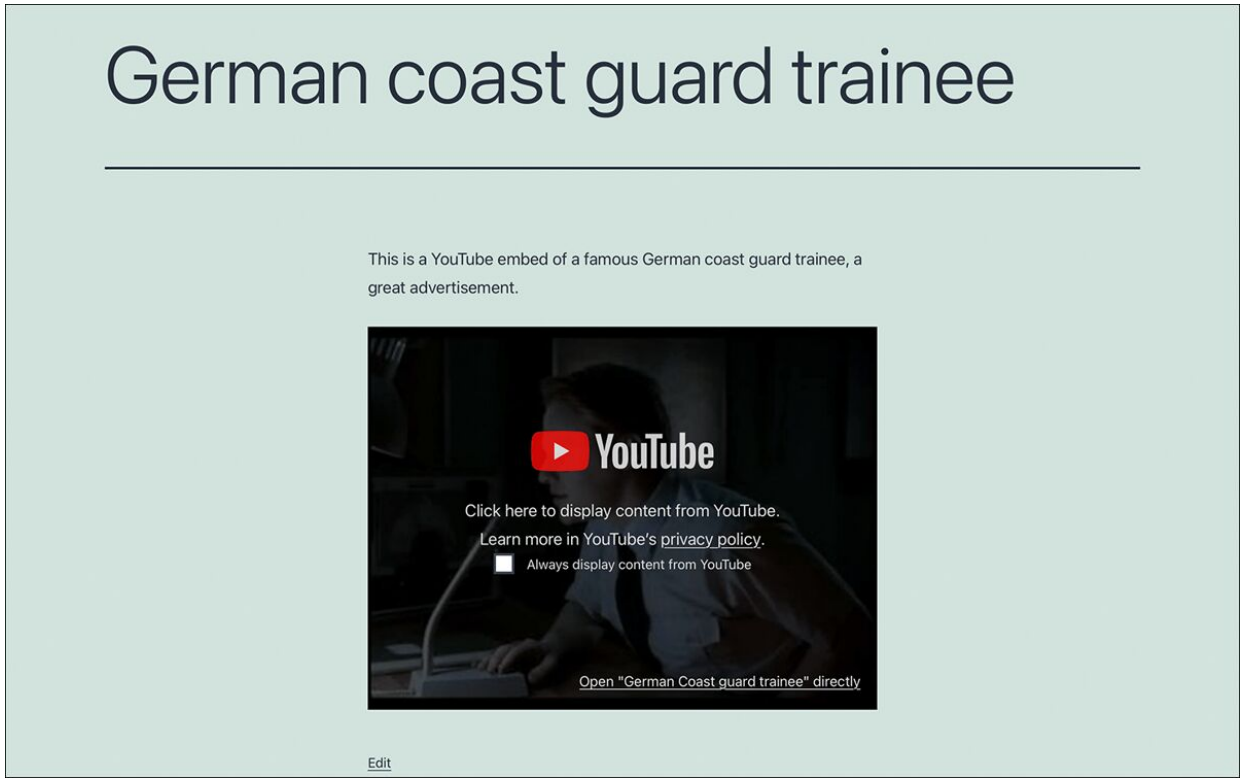


Abbildung 13.11 Embed Privacy verhindert, dass Einbettungen wie YouTube-Videos direkt geladen werden.

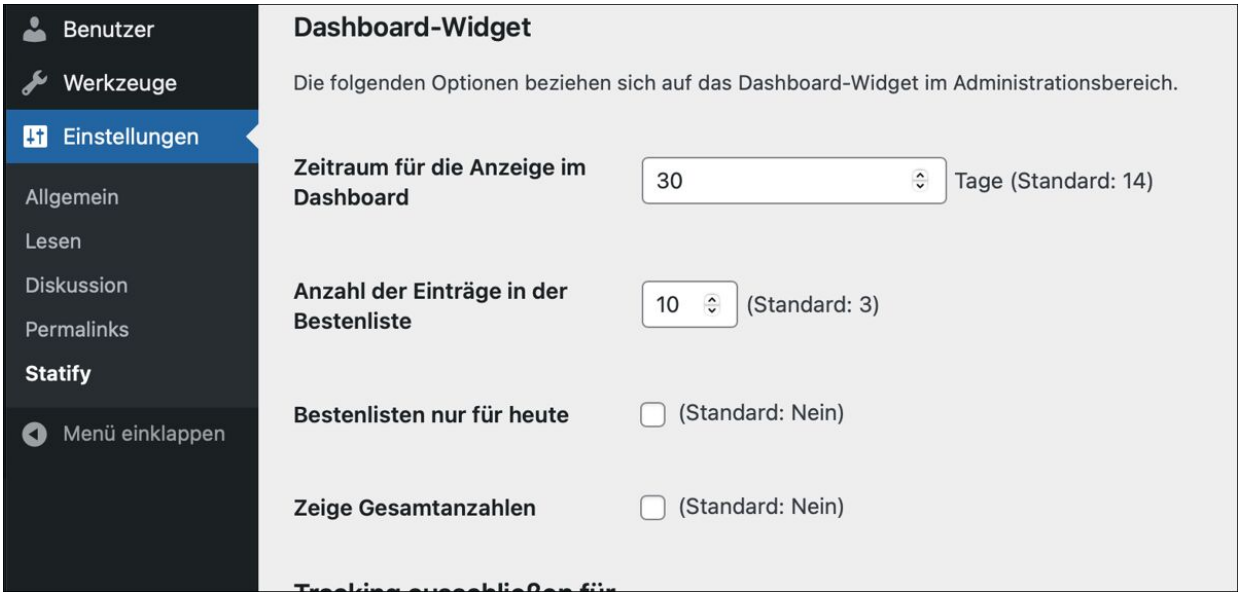


Abbildung 13.12 Mit dem Plugin Aged Content Message können Sie bei älteren Beiträgen automatisch eine Nachricht anzeigen lassen, dass der Inhalt gegebenenfalls nicht mehr aktuell ist.

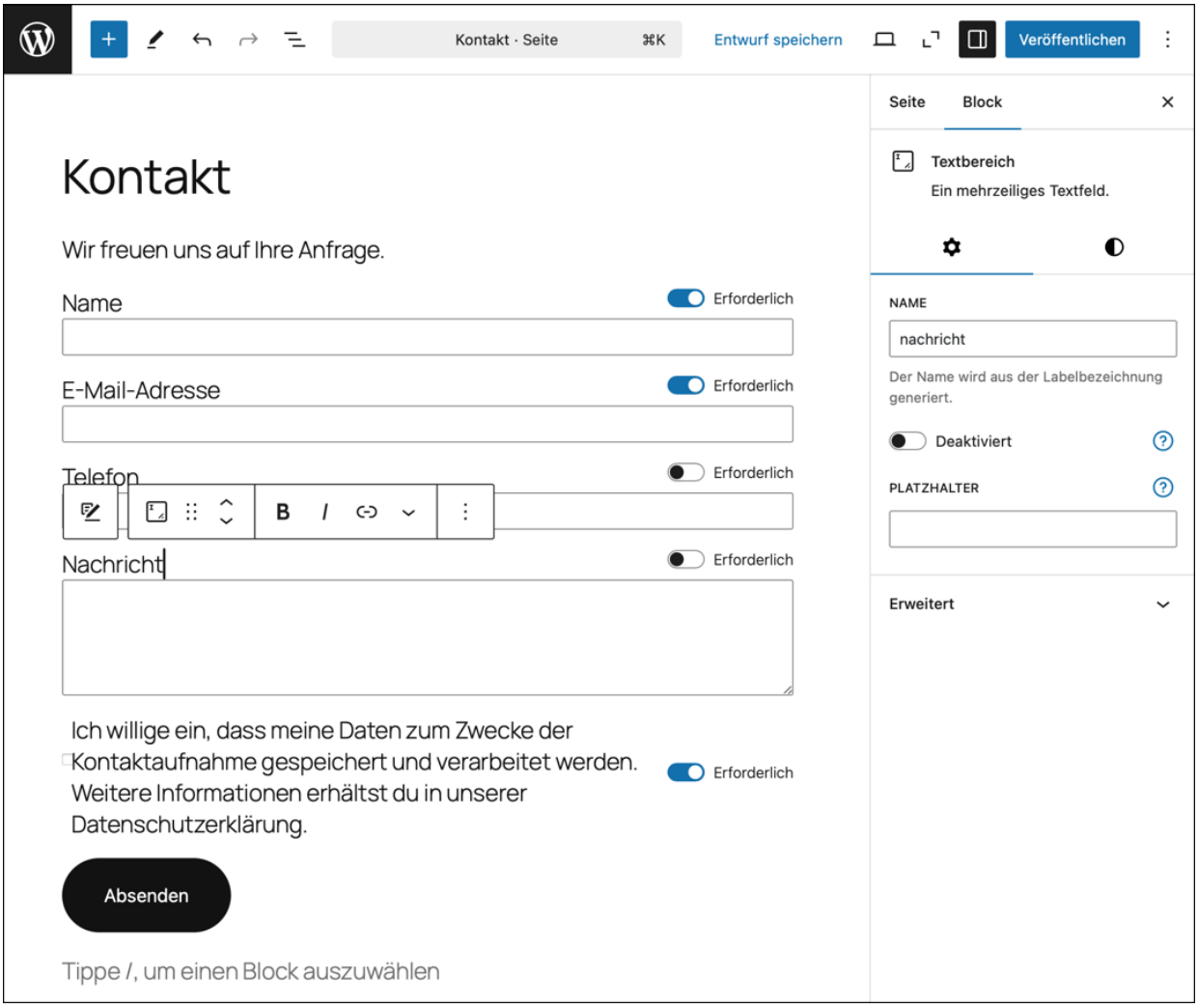


Abbildung 14.1 Das Form-Block-Plugin lässt Sie Formulare direkt im Block-Editor erstellen und anpassen.

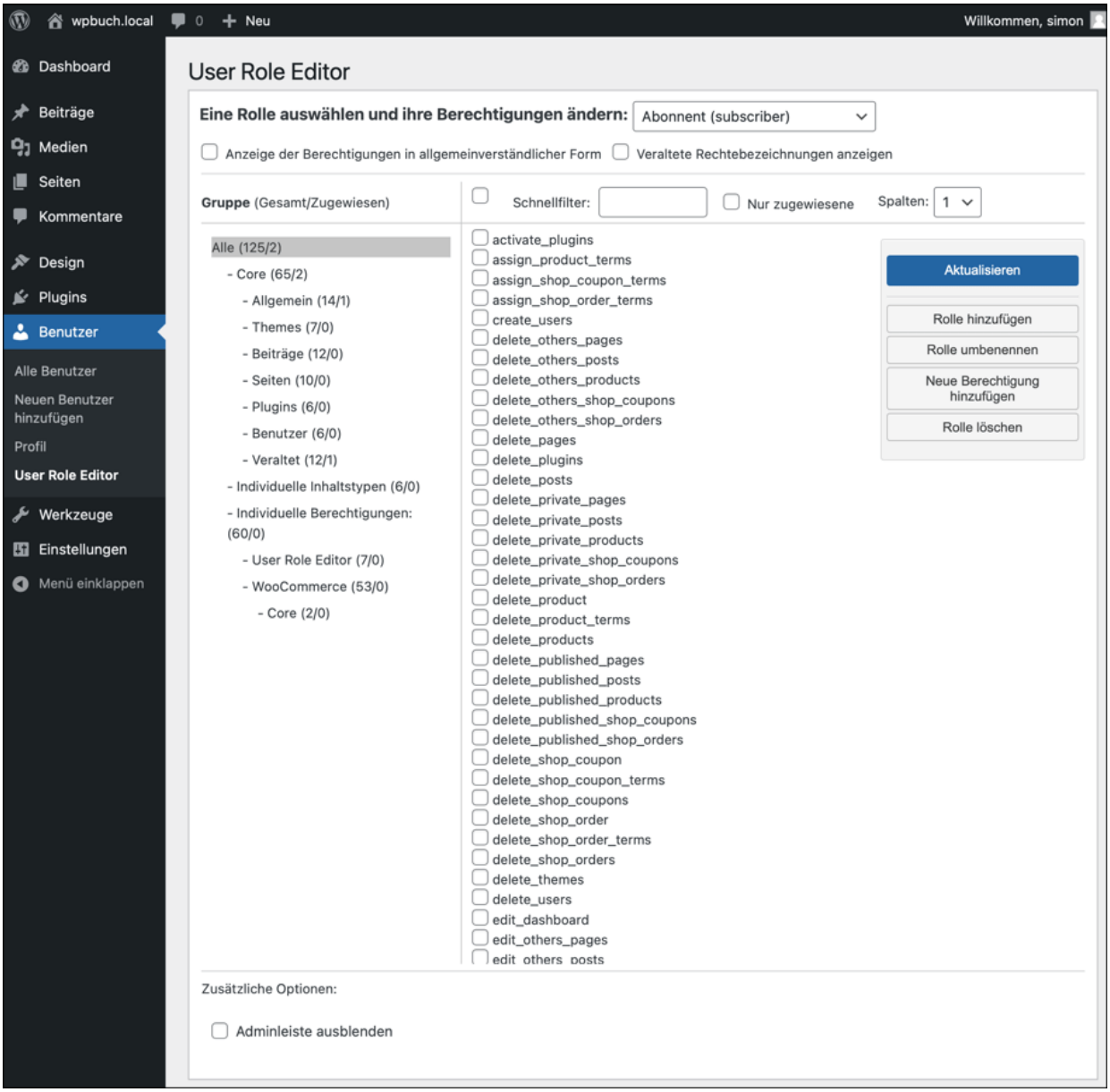


Abbildung 14.2 Mit dem Plugin User Role Editor können Sie bestehende Rollen bearbeiten und neue hinzufügen.



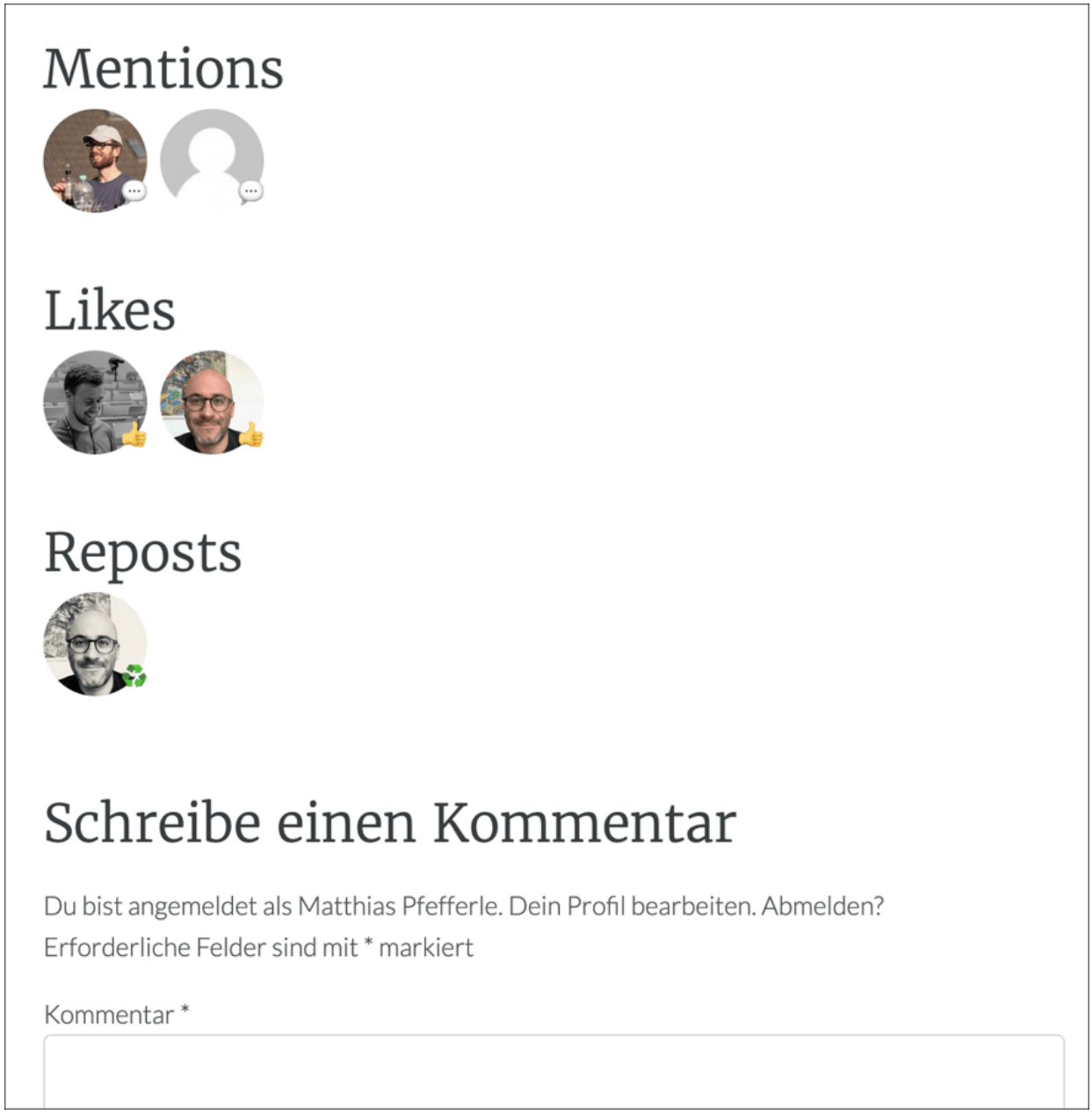


Abbildung 15.1 Beispiel für die Darstellung der unterschiedlichen Webmention-Typen



Abbildung 15.2 Die Brid.gy Startseite mit Anmeldemöglichkeiten für verschiedene soziale Netzwerke

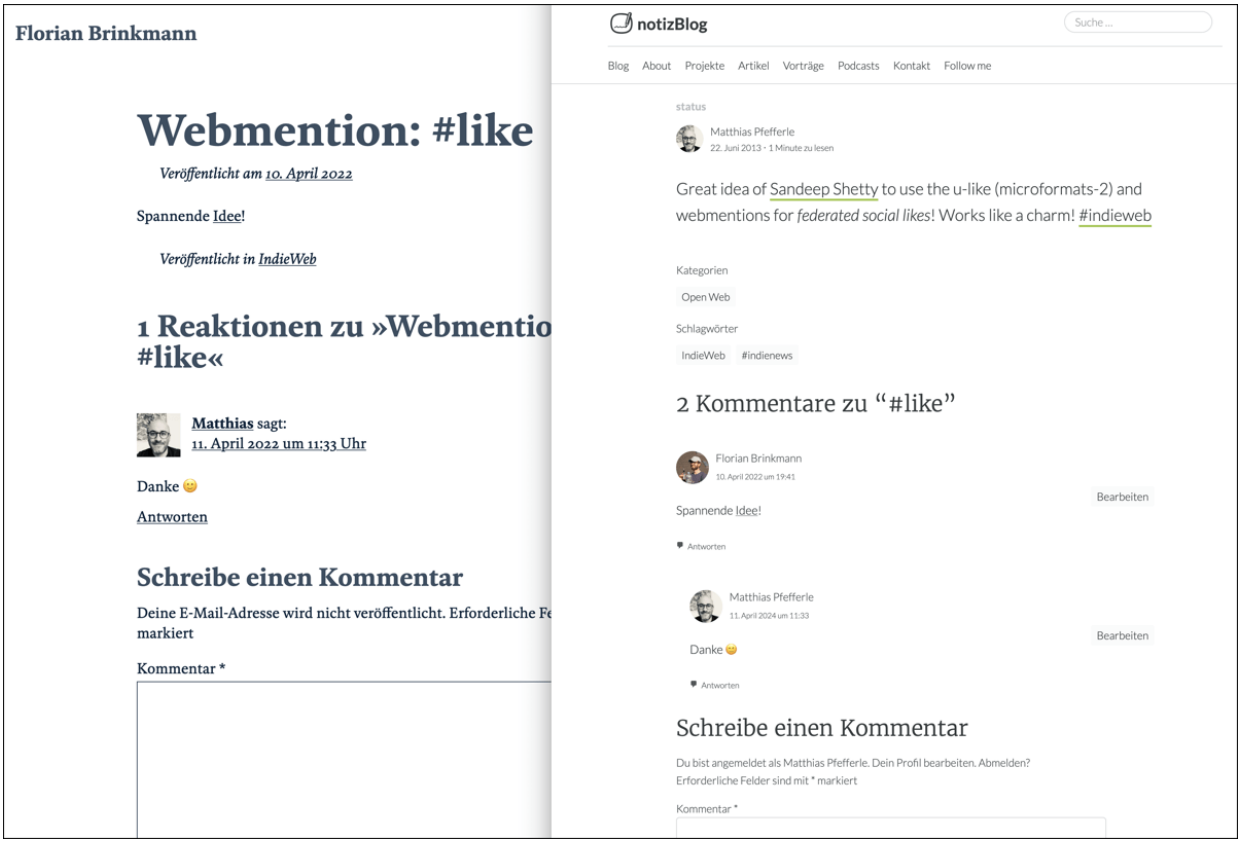


Abbildung 15.3 Ein Webmention-Beispiel für einen Kommentar-Thread

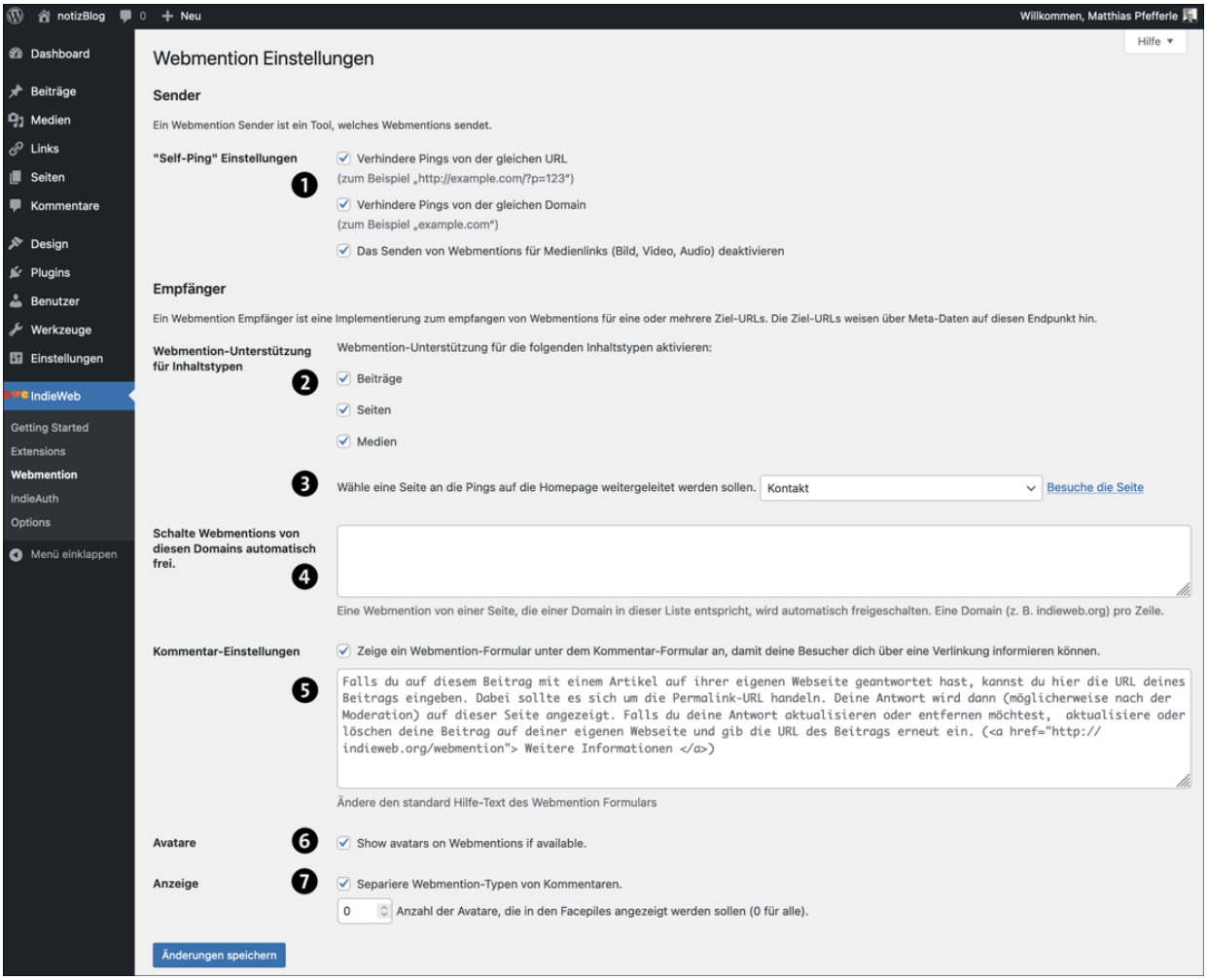


Abbildung 15.4 Die Einstellungen des Webmention-Plugins



Abbildung 15.5 »Facepile« Liste von »Like«-Webmentions

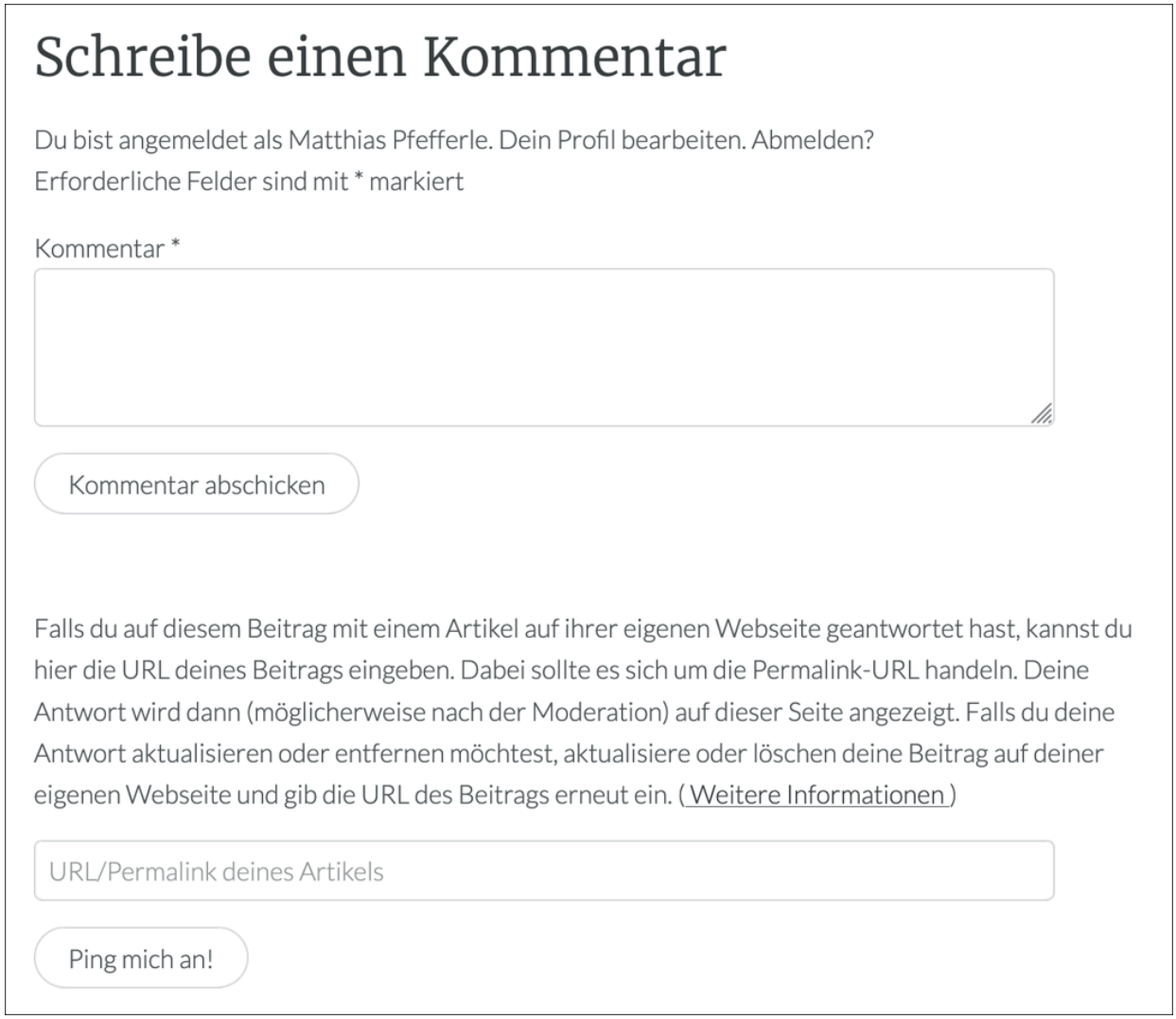


Abbildung 15.6 Das Webmention-Formular unter einem Kommentarfeld

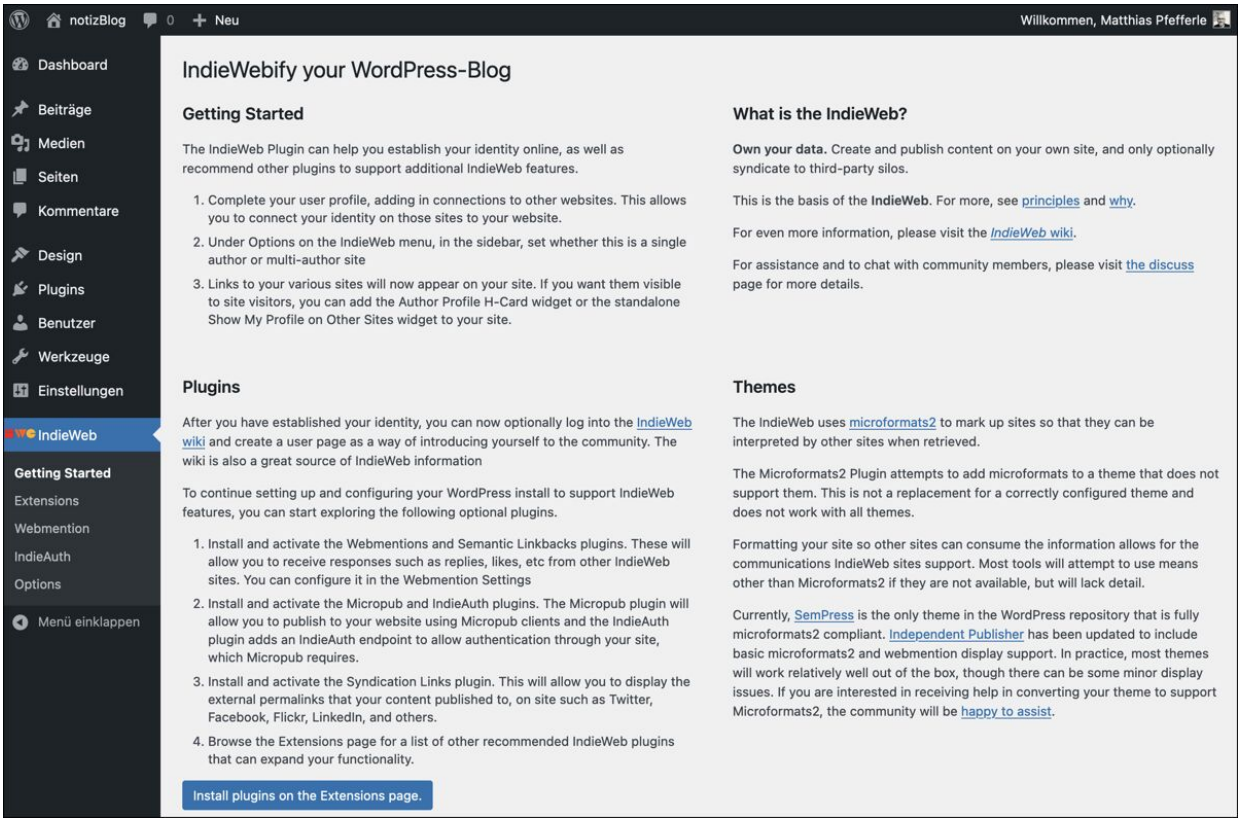


Abbildung 15.7 Die »Getting Started«-Seite des IndieWeb-Plugins

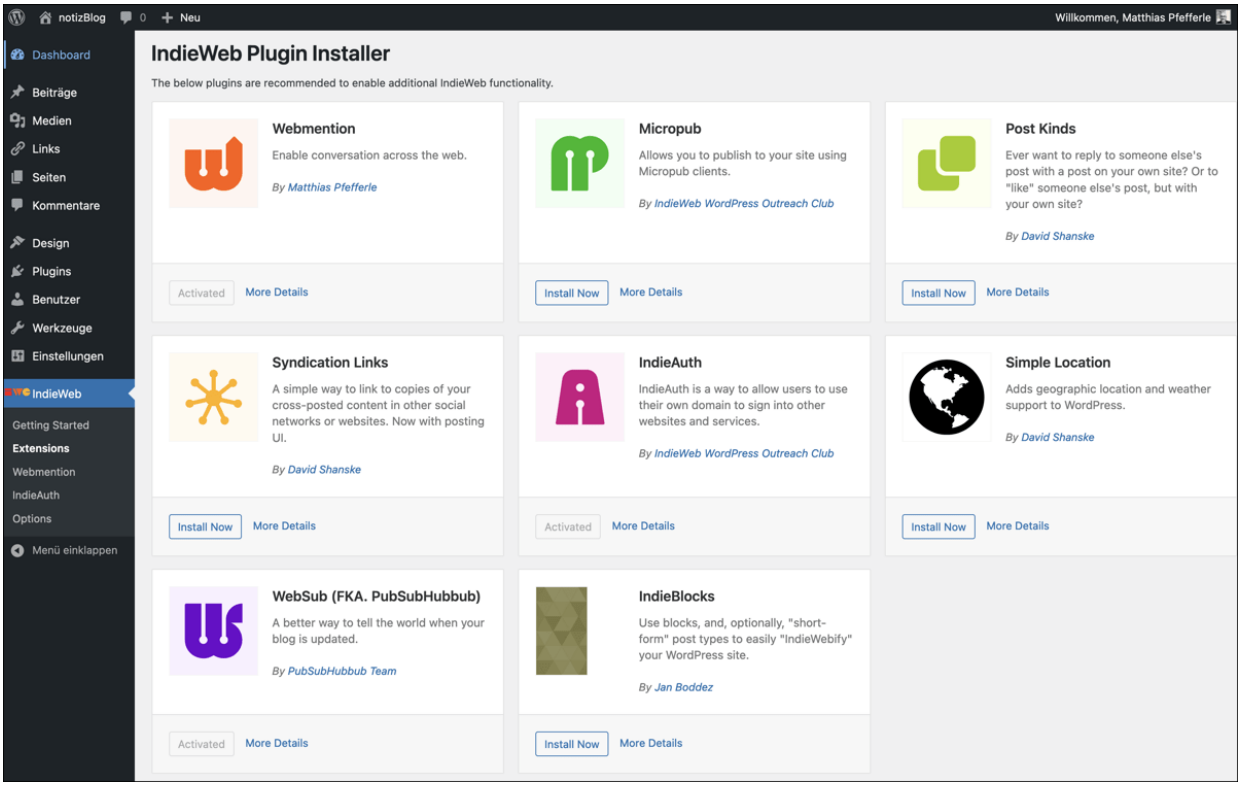


Abbildung 15.8 Die »Extensions«-Seite des IndieWeb-Plugins

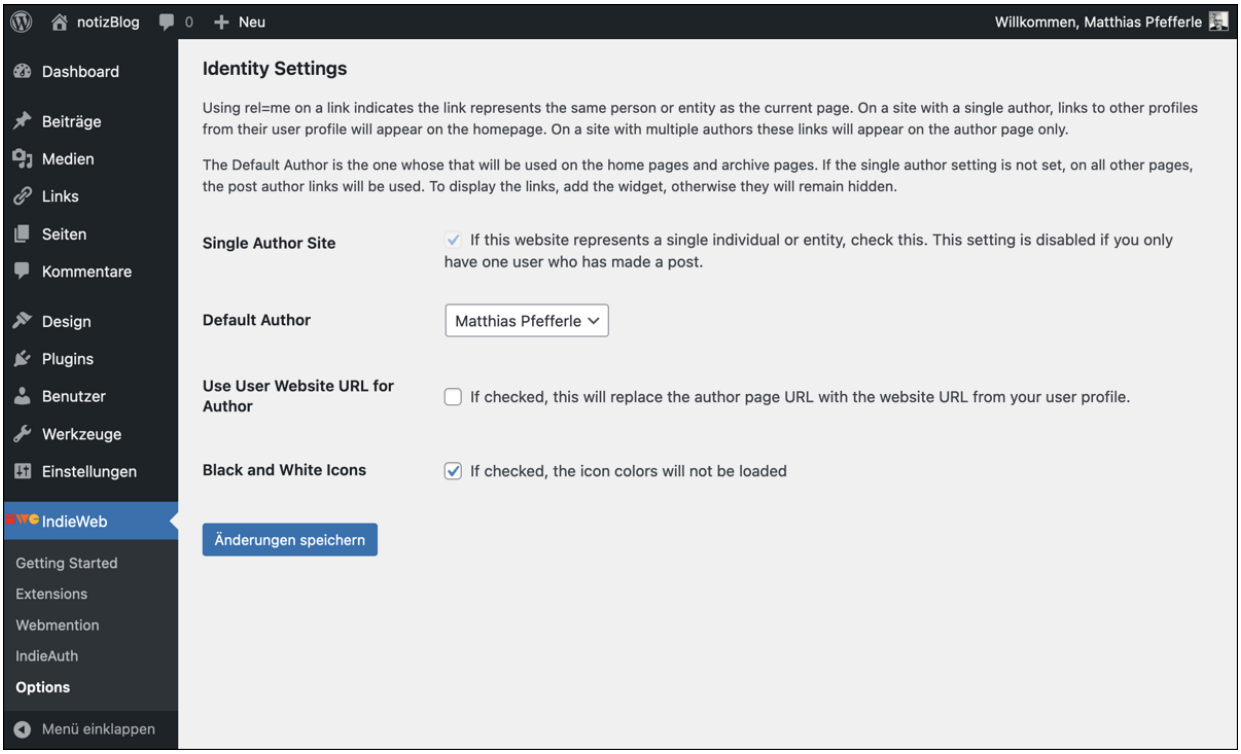


Abbildung 15.9 Die »Options«-Seite des IndieWeb-Plugins

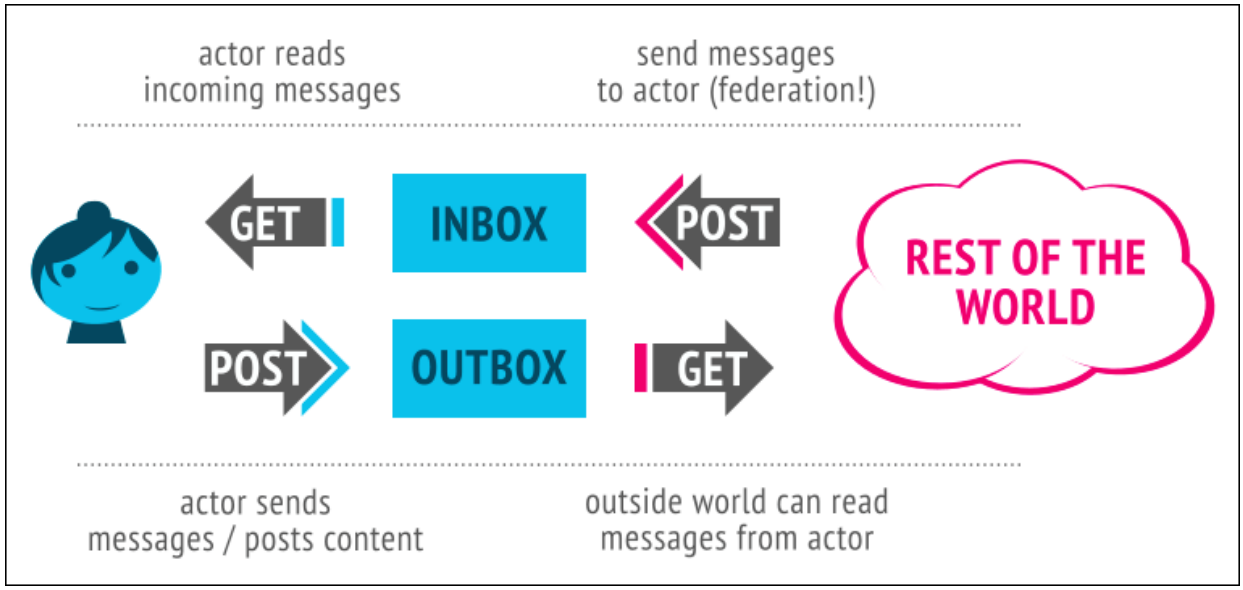
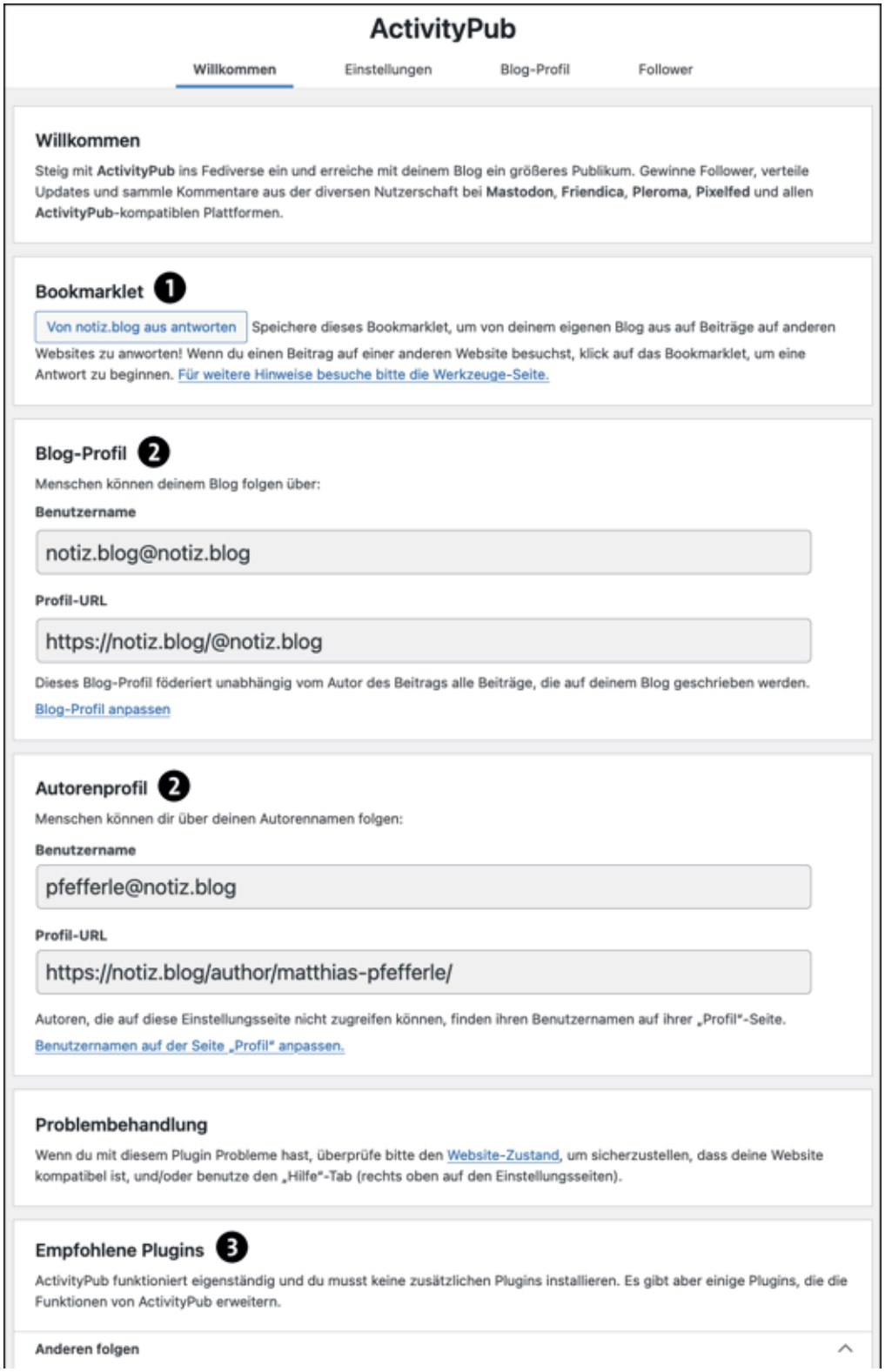


Abbildung 15.10 Schaubild zur Funktionsweise von ActivityPub


ActivityPub
Willkommen
Einstellungen
Blog-Profil
Follower
Willkommen
Steig mit ActivityPub ins Fediverse ein und erreiche mit deinem Blog ein größeres Publikum. Gewinne Follower, verteile Updates und sammle Kommentare aus der diversen Nutzerschaft bei Mastodon, Friendica, Pleroma, Pixelfed und allen ActivityPub-kompatiblen Plattformen.
Bookmarklet
1
Von notiz.blog aus antworten
Speichere dieses Bookmarklet, um von deinem eigenen Blog aus auf Beiträge auf anderen Websites zu anworten! Wenn du einen Beitrag auf einer anderen Website besuchst, klick auf das Bookmarklet, um eine Antwort zu beginnen. Für weitere Hinweise besuche bitte die Werkzeuge-Seite.
Blog-Profil
2
Menschen können deinem Blog folgen über:
Benutzername
notiz.blog@notiz.blog
Profil-URL
https://notiz.blog/@notiz.blog
Dieses Blog-Profil föderiert unabhängig vom Autor des Beitrags alle Beiträge, die auf deinem Blog geschrieben werden.
Blog-Profil anpassen
Autorenprofil
2
Menschen können dir über deinen Autorennamen folgen:
Benutzername
pfefferle@notiz.blog
Profil-URL
https://notiz.blog/author/matthias-pfefferle/
Autoren, die auf diese Einstellungsseite nicht zugreifen können, finden ihren Benutzernamen auf ihrer „Profil"-Seite.
Benutzernamen auf der Seite „Profil" anpassen.
Problembehandlung
Wenn du mit diesem Plugin Probleme hast, überprüfe bitte den Website-Zustand, um sicherzustellen, dass deine Website kompatibel ist, und/oder benutze den „Hilfe"-Tab (rechts oben auf den Einstellungsseiten).
Empfohlene Plugins
3
ActivityPub funktioniert eigenständig und du musst keine zusätzlichen Plugins installieren. Es gibt aber einige Plugins, die die Funktionen von ActivityPub erweitern.
Anderen folgen

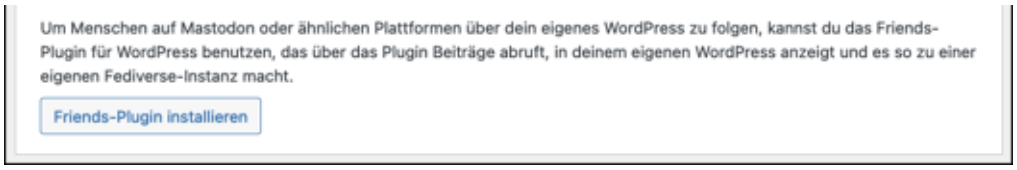
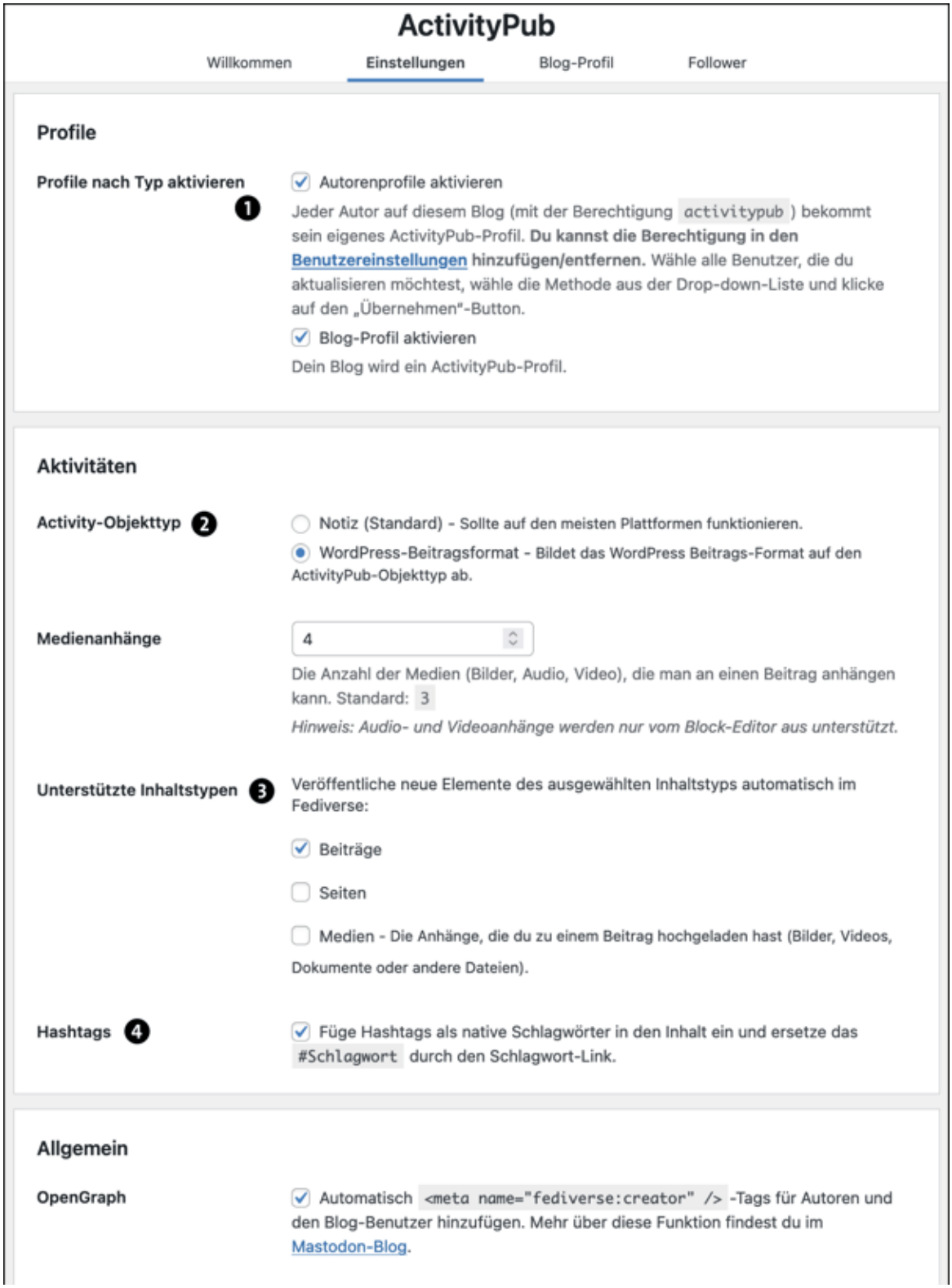


Abbildung 15.11 Die »Willkommen«-Seite des ActivityPub-Plugins


ActivityPub
Willkommen
Einstellungen
Blog-Profil
Follower
Profile
Profile nach Typ aktivieren
1
Autorenprofile aktivieren
Jeder Autor auf diesem Blog (mit der Berechtigung activitypub) bekommt sein eigenes ActivityPub-Profil. Du kannst die Berechtigung in den Benutzereinstellungen hinzufügen/entfernen. Wähle alle Benutzer, die du aktualisieren möchtest, wähle die Methode aus der Drop-down-Liste und klicke auf den „Übernehmen"-Button.
Blog-Profil aktivieren
Dein Blog wird ein ActivityPub-Profil.
Aktivitäten
Activity-Objekttyp
2
Notiz (Standard) - Sollte auf den meisten Plattformen funktionieren.
WordPress-Beitragsformat - Bildet das WordPress Beitrags-Format auf den ActivityPub-Objekttyp ab.
Medienanhänge
4
Die Anzahl der Medien (Bilder, Audio, Video), die man an einen Beitrag anhängen kann. Standard: 3
Hinweis: Audio- und Videoanhänge werden nur vom Block-Editor aus unterstützt.
Unterstützte Inhaltstypen
3
Veröffentliche neue Elemente des ausgewählten Inhaltstyps automatisch im Fediverse:
Beiträge
Seiten
Medien - Die Anhänge, die du zu einem Beitrag hochgeladen hast (Bilder, Videos, Dokumente oder andere Dateien).
Hashtags
4
Füge Hashtags als native Schlagwörter in den Inhalt ein und ersetze das #Schlagwort durch den Schlagwort-Link.
Allgemein
OpenGraph
Automatisch <meta name="fediverse:creator" /> -Tags für Autoren und den Blog-Benutzer hinzufügen. Mehr über diese Funktion findest du im Mastodon-Blog.

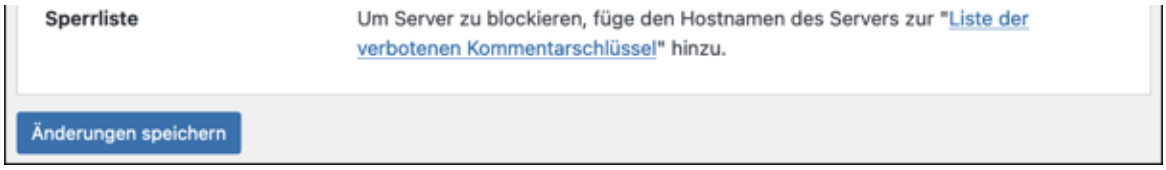


Abbildung 15.12 Die »Einstellungen«-Seite des ActivityPub-Plugins

ActivityPub

Willkommen | Einstellungen | **Blog-Profil** | Follower

Blog-Profil

Avatar verwalten



Das ActivityPub-Plugin nutzt das Website-Icon von WordPress als Avatar für das Blog-Profil. Du kannst das Website-Icon in den "Allgemeinen Einstellungen" von WordPress ändern.

Header-Bild verwalten



Header-Bild ändern | Header-Bild entfernen

Profil-ID ändern

notiz.blog @notiz.blog

Dieser Profilname föderiert unabhängig vom Autor des Beitrags alle Beiträge, die auf deinem Blog geschrieben werden.

Benutze bitte keinen vorhanden Autorennamen als Profil-ID für das Blog. Fediverse-Plattformen könnten Caching verwenden und damit verhindern, dass das Plugin funktioniert.

Beschreibung ändern

a weblog mainly about the open, portable, interoperable, small, social, synaptic, semantic, structured, distributed, (re-)decentralized, independent, microformatted and federated social web

Standardmäßig verwendet das ActivityPub-Plugin den WordPress-Untertitel als Beschreibung für das Blog-Profil.

Zusatzfelder

Deine Homepage, Social-Profile, Pronomen, Alter, was immer du willst.

Blog	notiz.blog	Bearbeiten



Abbildung 15.13 Die »Profil«-Seite des ActivityPub-Plugins

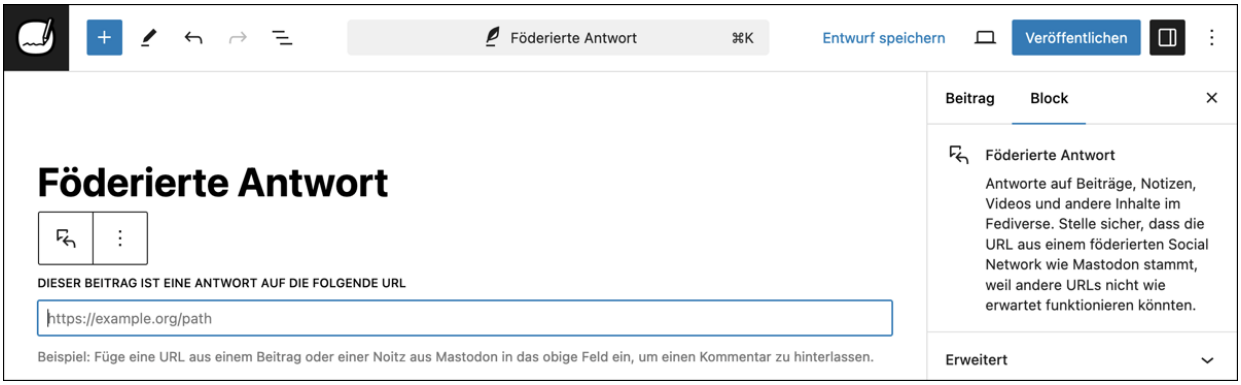


Abbildung 15.14 Der Block »Föderierte Antwort« im Block-Editor

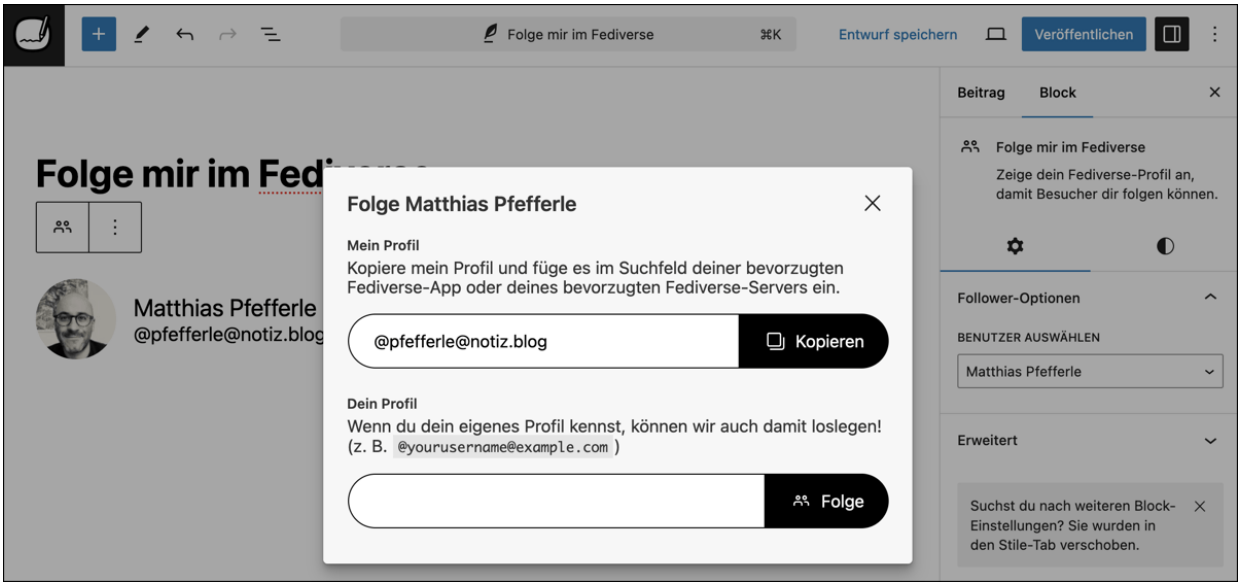


Abbildung 15.15 Der Block »Folge mir im Fediverse« mit Overlay im Block-Editor

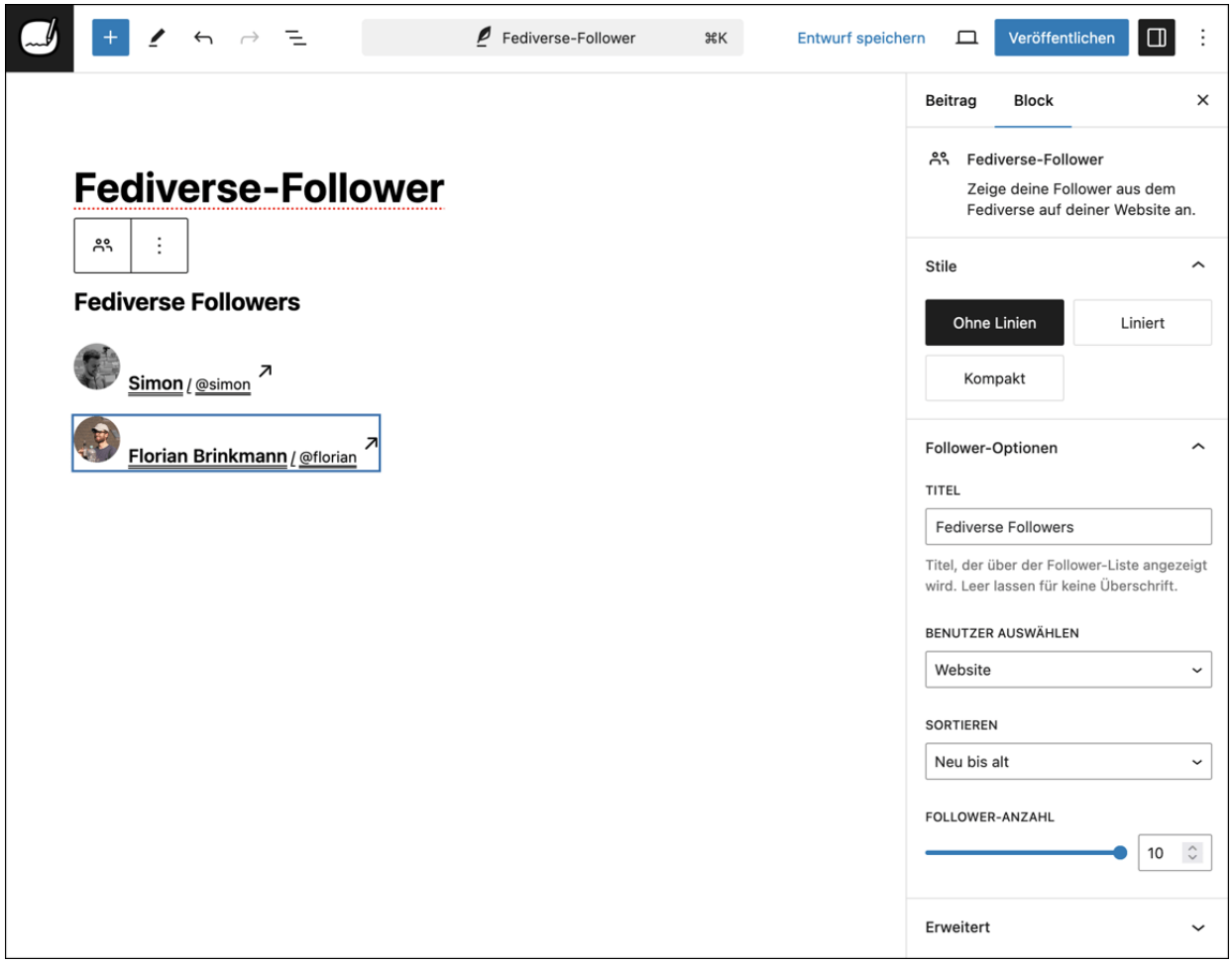


Abbildung 15.16 Der Block »Fediverse-Follower« im Block-Editor

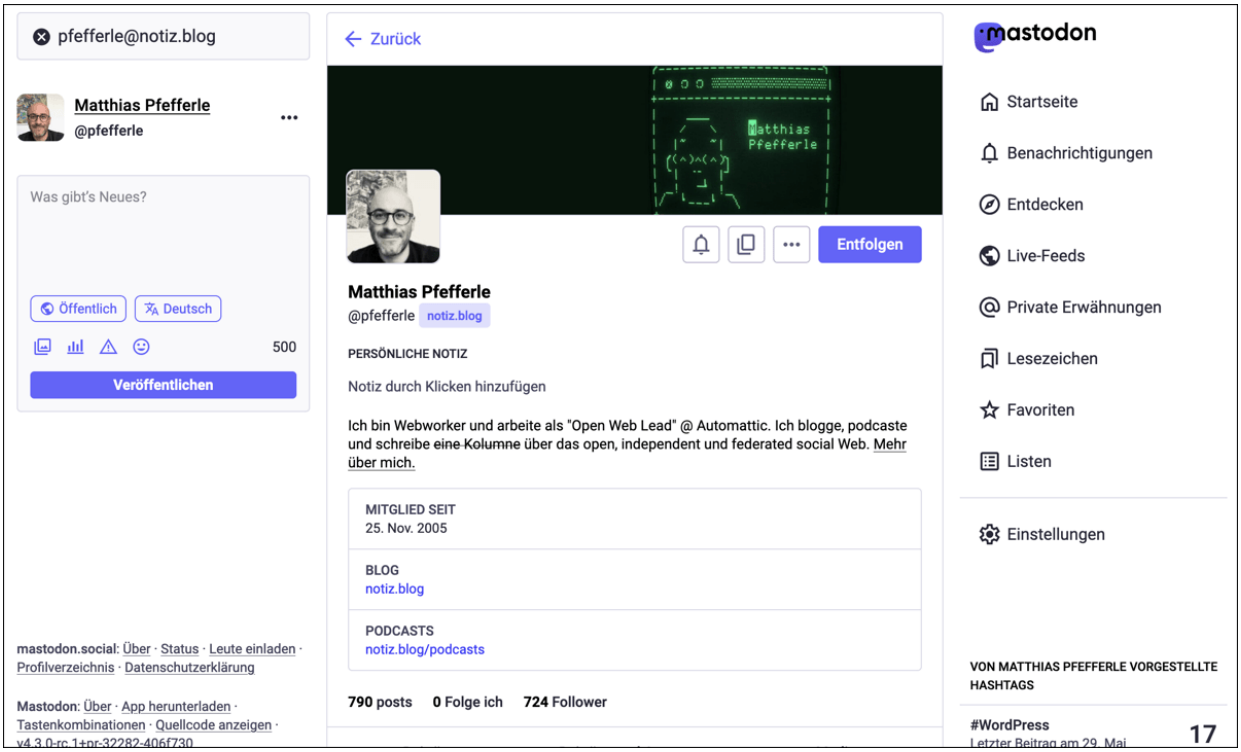


Abbildung 15.17 Ein Beispiel für die Anzeige eines Blog-Profils auf Mastodon

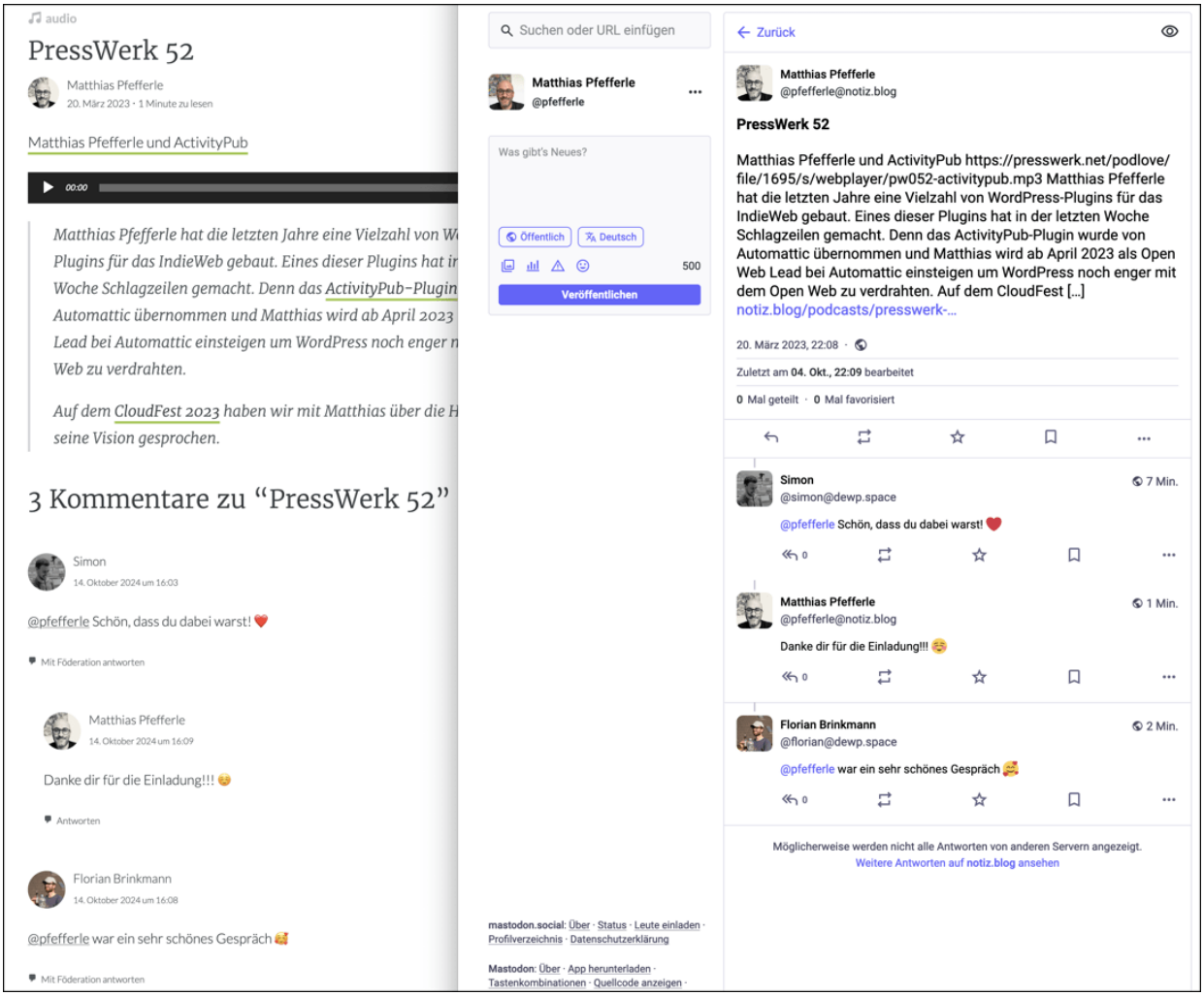


Abbildung 15.18 Ein Beispiel für föderierte Kommentare. Rechts die Reaktionen auf Mastodon und links die entsprechenden Kommentare auf der WordPress-Seite.



Abbildung 16.1 Die PHP-Version des Servers können Sie im »Website-Zustand«-Werkzeug einsehen.

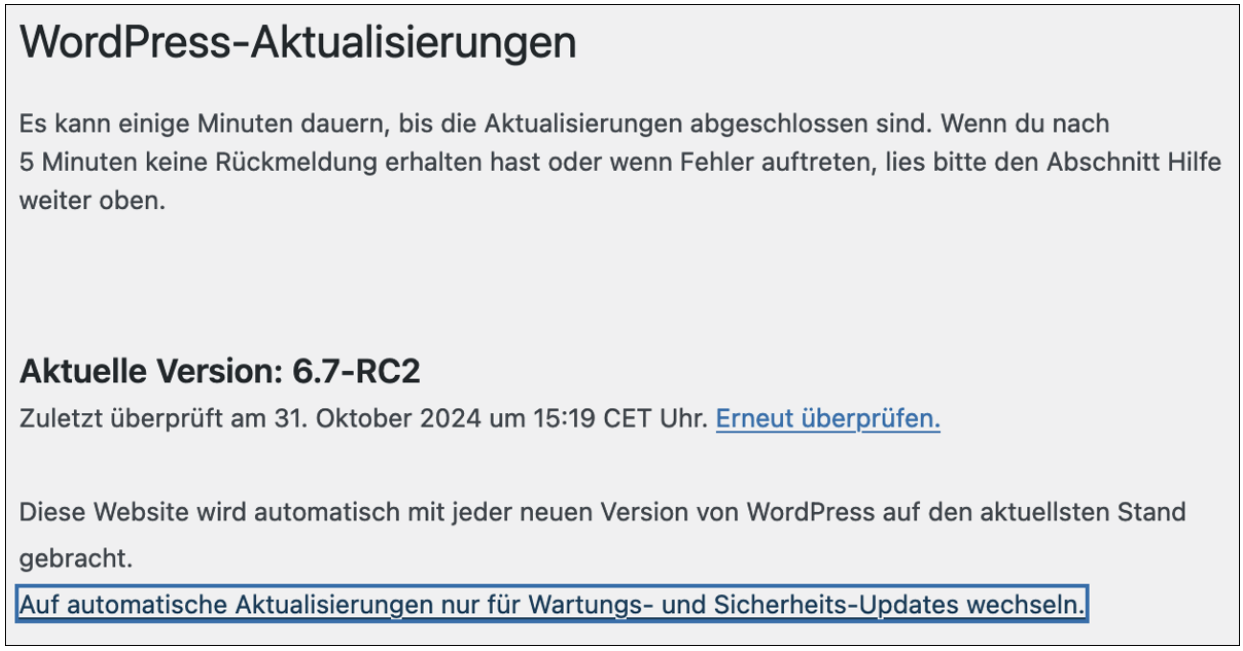


Abbildung 16.2 Über den hervorgehobenen Link am unteren Rand des Screenshots können die automatischen WordPress-Updates zwischen kleinen und allen Versionen umgestellt werden.

Hilfe ▼

WordPress-Beta testen

Bitte beachte: Sobald du deine Website auf eine dieser Beta-Versionen der Software umgestellt hast, wird es nicht immer möglich sein, ein Downgrade durchzuführen, da es sein kann, dass die Datenbankstruktur einer Hauptversion während der Entwicklung aktualisiert wird.

Bitte beachte: Für den von dir gewählten Beta-Stream sind keine Entwicklungsversionen verfügbar, sodass du reguläre Aktualisierungshinweise erhalten wirst.

WP-Beta-Tester-Einstellungen | Extra Einstellungen | Fehler melden

Core-Einstellungen

Es liegt in der Natur dieser Versionen, dass sie instabil sind und daher nicht genutzt werden sollten, wenn deine Daten wichtig sind. Fertige also bitte ein Backup deiner Datenbank an, bevor du auf eine Test-Version aktualisiert. Die beste Quelle für Informationen über die neuesten Beta-Versionen sind das Development-Blog (engl.) und das Beta-Forum (engl.).

Vielen Dank für deine Hilfe beim Testen von WordPress. Bitte melde alle Fehler, die du findest.

Standardmäßig benutzt deine WordPress-Installation den stabilen Aktualisierungskanal. Um den ursprünglichen Zustand wiederherzustellen, deaktiviere bitte das Plugin und installiere die WordPress-Version neu über die Seite Aktualisierungen im WordPress-Dashboard.

Aktuell ist diese Website für die Aktualisierung auf **version 6.6.3, 6.7-RC3, oder 6.7, whichever is released first** konfiguriert.

Bitte wähle den Aktualisierungs-Kanal, den du auf dieser Website verwenden möchtest:

Speichere nach dem Wechseln

○ **Point Release (Minorversions-Update)**	Das enthält den Entwicklungs-Code aus einem Zweig, der für eine 6.6.3-Punkt-Version in Vorbereitung ist. Dieser sollte bereits relativ stabil sein, ist allerdings hierüber verfügbar, bevor der Zweig bereit für die Veröffentlichung ist.
◉ **Bleeding Edge (Hauptversions-Update)**	Das ist der allerneueste Entwicklungs-Code aus dem 'trunk'-Zweig, der auch manchmal instabil sein kann. *Setze diesen nur ein, wenn du wirklich weißt, was du tust.*

Wähle eine der folgenden Stream-Optionen:

○ **Nightlies**	Die neuesten täglichen Updates.
◉ **Nur Beta/RC**	Dies gilt nur für die Beta-/RC-Veröffentlichungen des ausgewählten Kanals.
	Dies gilt nur für die Veröffentlichungen der Release-Kandidaten des ausgewählten

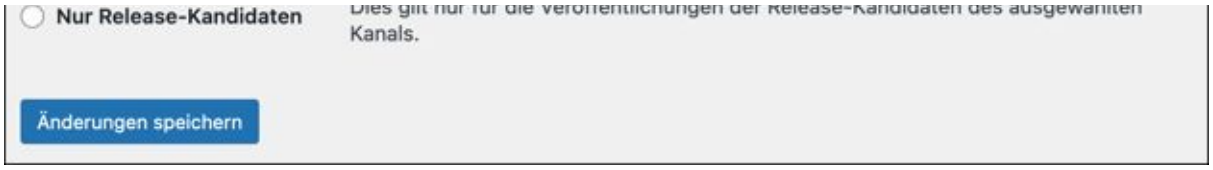


Abbildung 16.3 Das Plugin »WordPress Beta Tester« ermöglicht Ihnen recht einfach, Vorabversionen von WordPress zu testen.

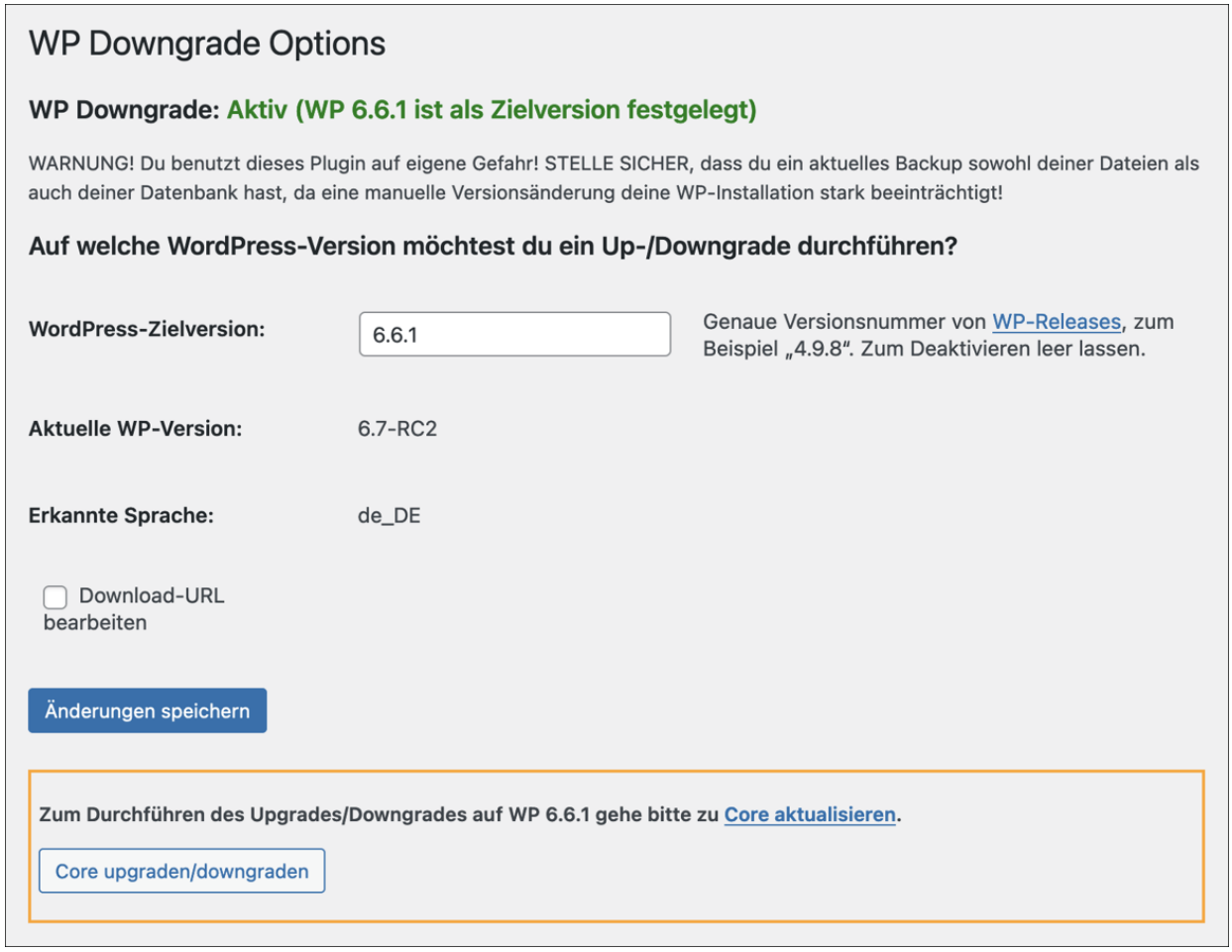


Abbildung 16.4 Mit dem Plugin »WP Downgrade« können Sie zu einer bestimmten WordPress-Version zurückwechseln.

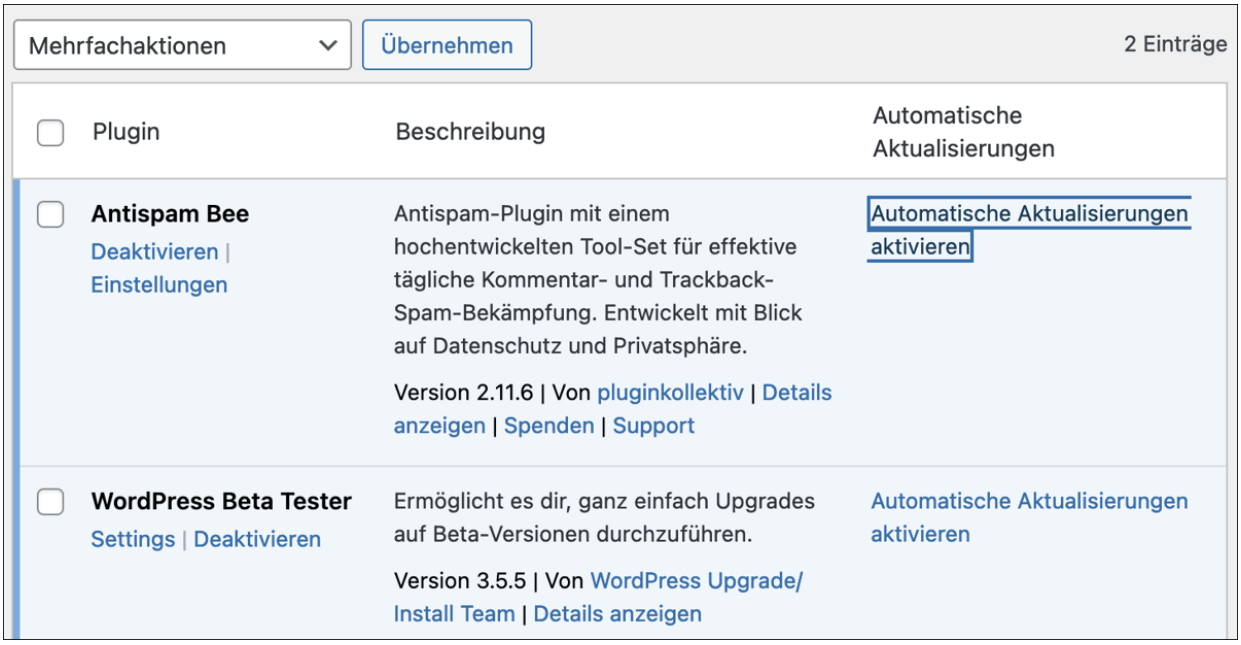


Abbildung 16.5 In der rechten Spalte der Plugin-Liste kann das Verhalten der automatischen Updates gesteuert werden.

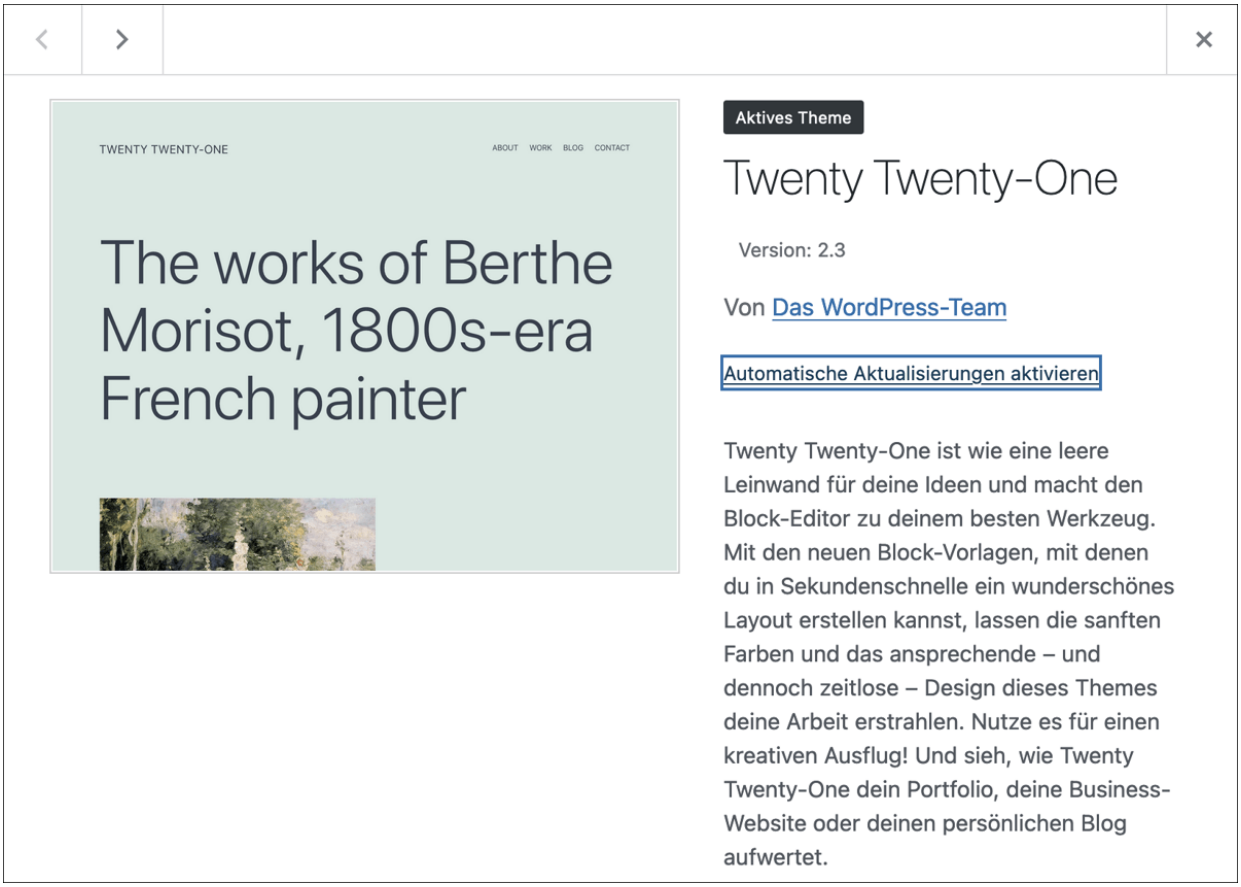


Abbildung 16.6 In dem Overlay mit Details zu einem Theme kann die automatische Aktualisierung für das Theme aktiviert oder deaktiviert werden.

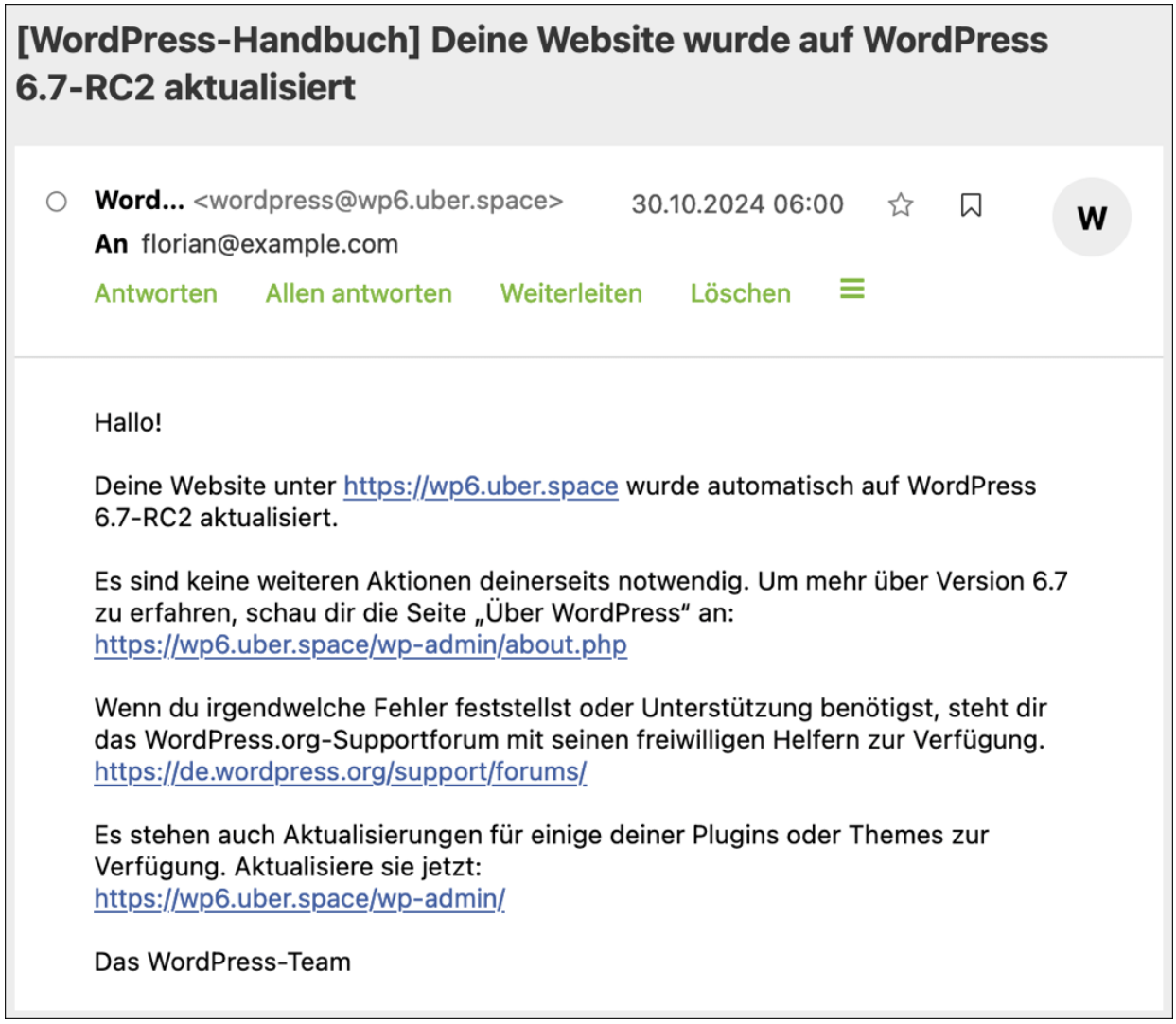


Abbildung 16.7 WordPress verschickt unter anderem E-Mail-Benachrichtigungen, wenn Updates durchgeführt wurden, damit Sie immer auf dem Laufenden sind.

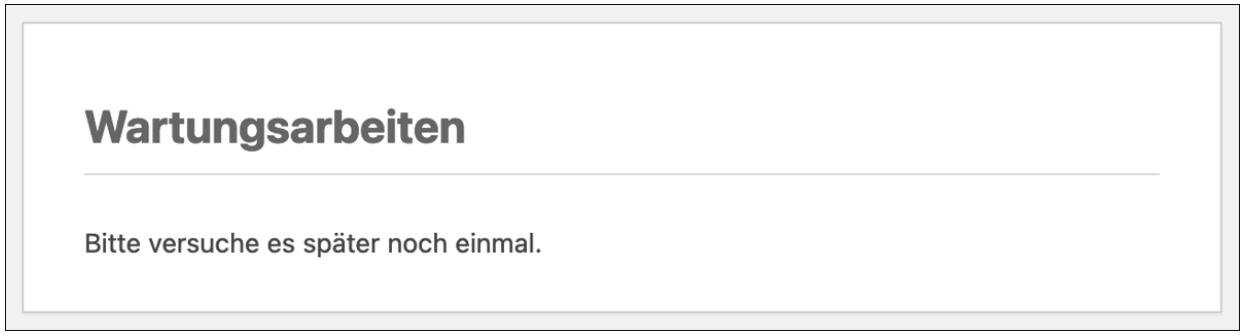


Abbildung 16.8 Das Plugin »Slim Maintenance Mode« bringt keine Einstellungsmöglichkeiten mit, sondern zeigt einfach eine kurze Info an, dass hier momentan nichts zu sehen ist.

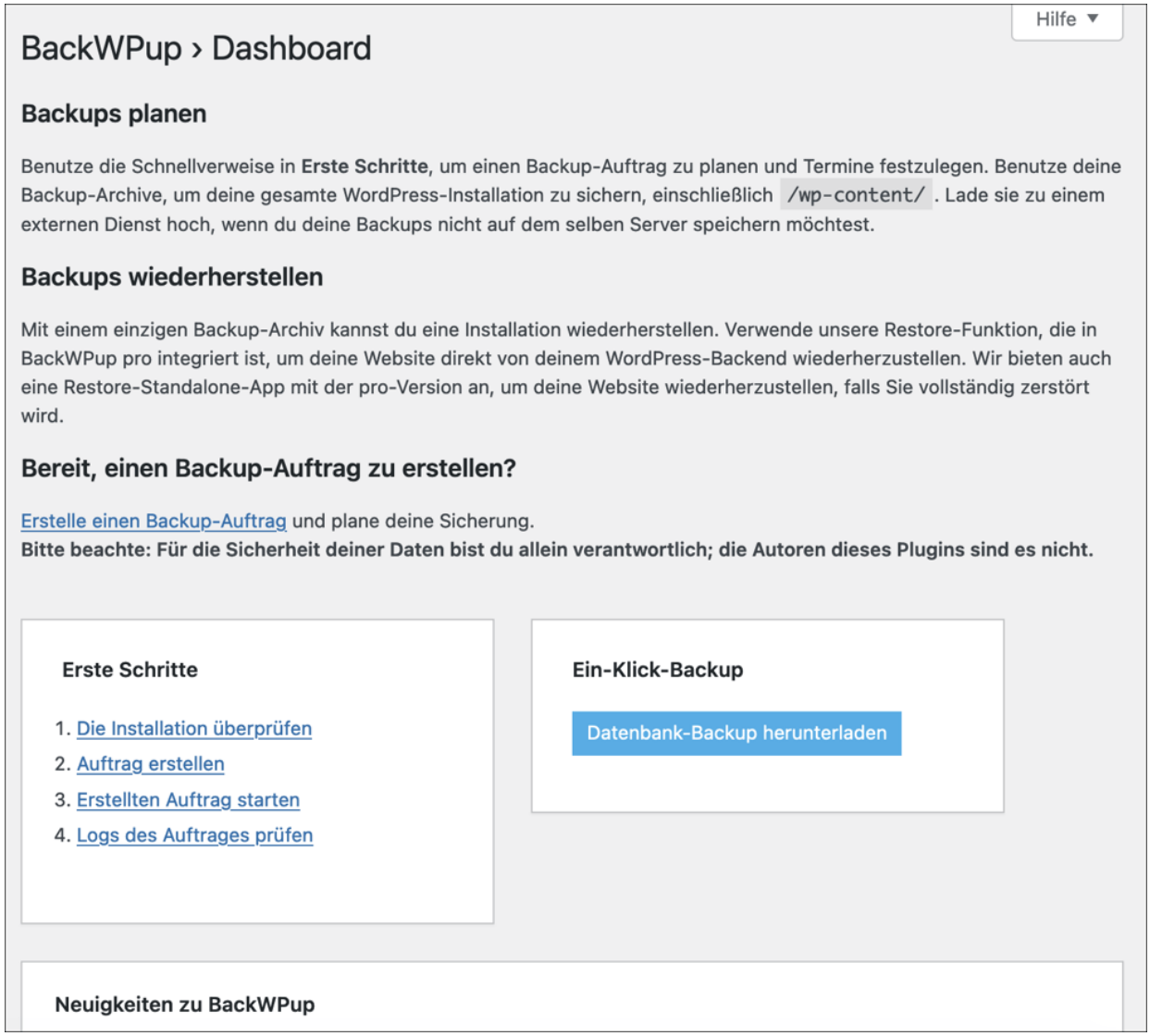


Abbildung 16.9 Funktionen bietet das Dashboard kaum, aber einen nützlichen Button, um direkt ein Datenbank-Backup herunterzuladen.

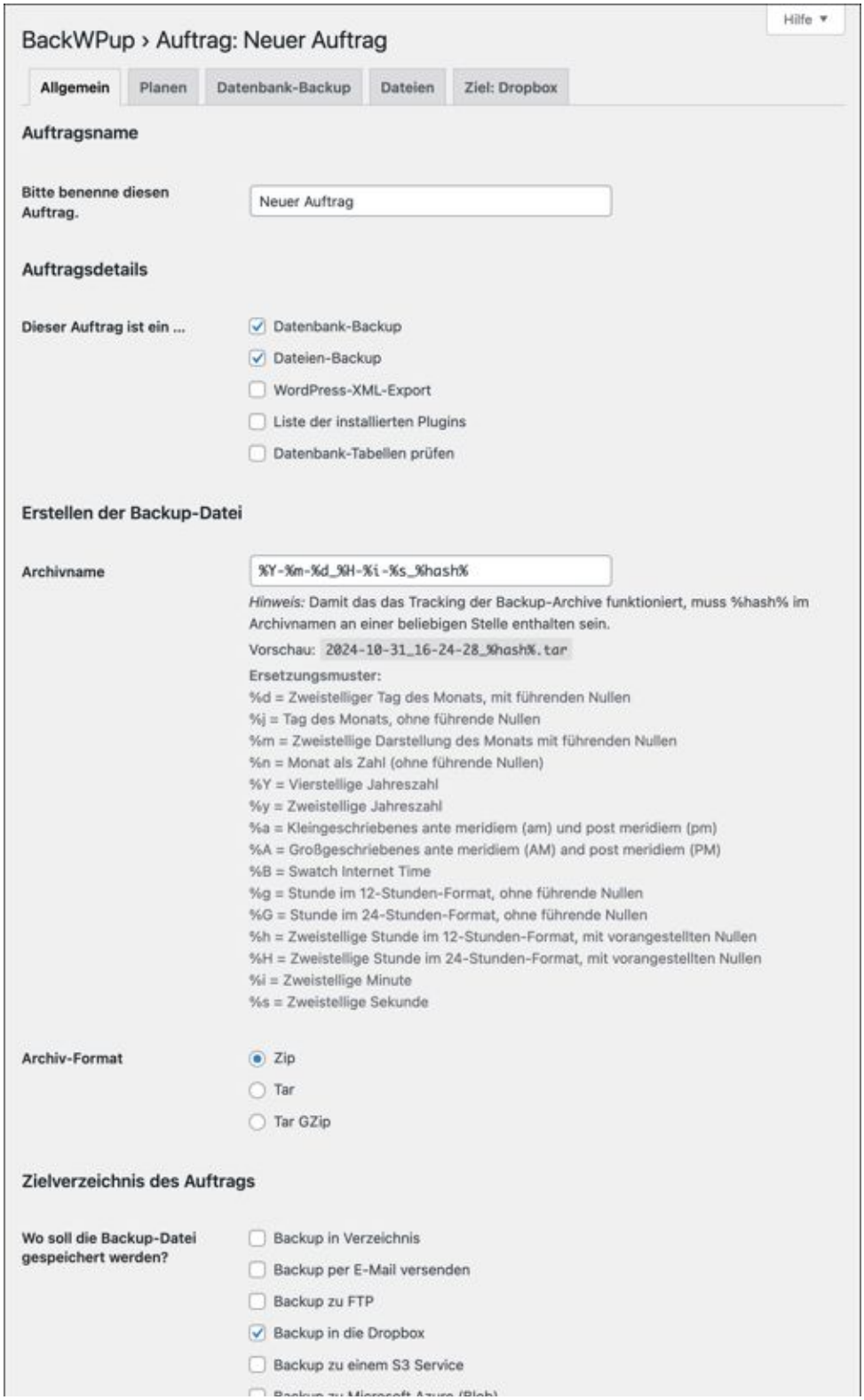

Hilfe
BackWPup › Auftrag: Neuer Auftrag
Allgemein
Planen
Datenbank-Backup
Dateien
Ziel: Dropbox
Auftragsname
Bitte benenne diesen Auftrag.
Neuer Auftrag
Auftragsdetails
Dieser Auftrag ist ein ...
Datenbank-Backup
Dateien-Backup
WordPress-XML-Export
Liste der installierten Plugins
Datenbank-Tabellen prüfen
Erstellen der Backup-Datei
Archivname
%Y-%m-%d_%H-%i-%s_%hash%
Hinweis: Damit das das Tracking der Backup-Archive funktioniert, muss %hash% im Archivnamen an einer beliebigen Stelle enthalten sein.
Vorschau: 2024-10-31_16-24-28_%hash%.tar
Ersetzungsmuster:
%d = Zweistelliger Tag des Monats, mit führenden Nullen
%j = Tag des Monats, ohne führende Nullen
%m = Zweistellige Darstellung des Monats mit führenden Nullen
%n = Monat als Zahl (ohne führende Nullen)
%Y = Vierstellige Jahreszahl
%y = Zweistellige Jahreszahl
%a = Kleingeschriebenes ante meridiem (am) und post meridiem (pm)
%A = Großgeschriebenes ante meridiem (AM) and post meridiem (PM)
%B = Swatch Internet Time
%g = Stunde im 12-Stunden-Format, ohne führende Nullen
%G = Stunde im 24-Stunden-Format, ohne führende Nullen
%h = Zweistellige Stunde im 12-Stunden-Format, mit vorangestellten Nullen
%H = Zweistellige Stunde im 24-Stunden-Format, mit vorangestellten Nullen
%i = Zweistellige Minute
%s = Zweistellige Sekunde
Archiv-Format
Zip
Tar
Tar GZip
Zielverzeichnis des Auftrags
Wo soll die Backup-Datei gespeichert werden?
Backup in Verzeichnis
Backup per E-Mail versenden
Backup zu FTP
Backup in die Dropbox
Backup zu einem S3 Service



Abbildung 16.10 Im Tab »Allgemein« legen Sie allgemeine Aspekte des Backup-Auftrags fest, wie etwa, wo das Backup gespeichert wird.

Hilfe ▼

BackWPup › Auftrag: Auftrag mit der ID 1

Allgemein | **Planen** | Datenbank-Backup | Dateien | Ziel: Dropbox

Auftragsplanung

Auftrag starten

- ○ nur manuell
- ◉ mit WordPress Cron
- ○ mit einem Link `https://wp6.uber.space/wp-cron.php?_nonce=ec161e45&backwpup_run=runext&jobid=1`

Kopiere den Link für einen externen Auftragsstart. Diese Option muss erst aktiviert werden, damit der Link funktioniert.

Auftrag per CLI starten

Benutze WP-CLI, um Aufträge von der Kommandozeile aus zu starten.

Ausführungszeit planen

Nächster Durchlauf: **So, 3 Nov 2024, 03:00**

Planungstyp

- ◉ Grundlegend
- ○ erweitert

Planer

Typ		Stunde	Minute
○ monatlich	am 1	3	0
◉ wöchentlich	Sonntag	3	0
○ täglich		3	0
○ stündlich			0

Änderungen speichern

Abbildung 16.11 Im »Planen«-Tab können Sie festlegen, ob das Backup automatisch ausgeführt werden soll.

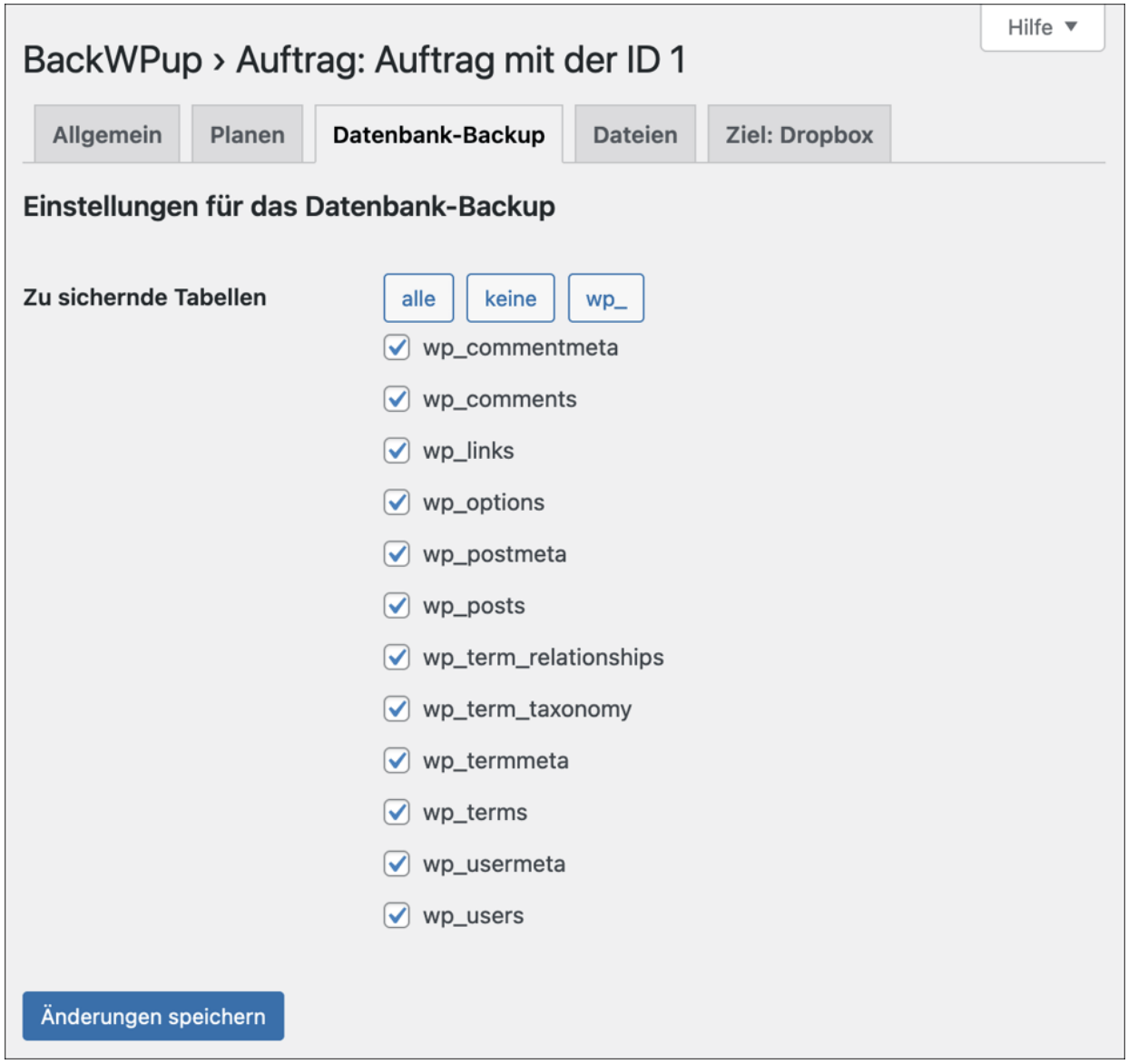


Abbildung 16.12 Hier können Sie einstellen, welche der Datenbank-Tabellen gesichert werden sollen.

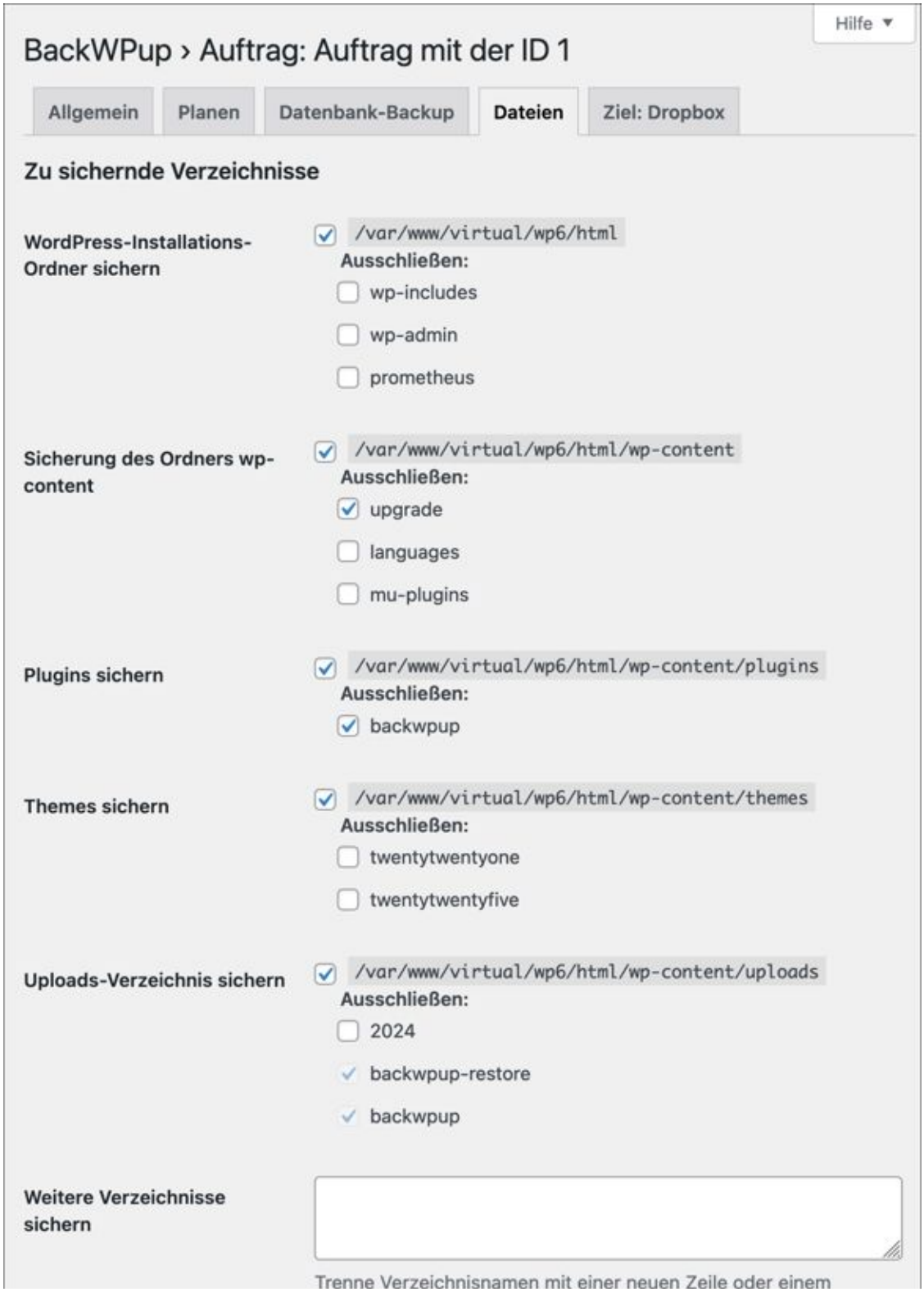

Hilfe
BackWPup › Auftrag: Auftrag mit der ID 1
Allgemein
Planen
Datenbank-Backup
Dateien
Ziel: Dropbox
Zu sichernde Verzeichnisse
WordPress-Installations-Ordner sichern
/var/www/virtual/wp6/html
Ausschließen:
wp-includes
wp-admin
prometheus
Sicherung des Ordners wp-content
/var/www/virtual/wp6/html/wp-content
Ausschließen:
upgrade
languages
mu-plugins
Plugins sichern
/var/www/virtual/wp6/html/wp-content/plugins
Ausschließen:
backwpup
Themes sichern
/var/www/virtual/wp6/html/wp-content/themes
Ausschließen:
twentytwentyone
twentytwentyfive
Uploads-Verzeichnis sichern
/var/www/virtual/wp6/html/wp-content/uploads
Ausschließen:
2024
backwpup-restore
backwpup
Weitere Verzeichnisse sichern
Trenne Verzeichnisnamen mit einer neuen Zeile oder einem



Abbildung 16.13 Im Tab »Dateien« können Sie anpassen, welche Verzeichnisse und Dateien im Dateien-Backup enthalten sein sollen.

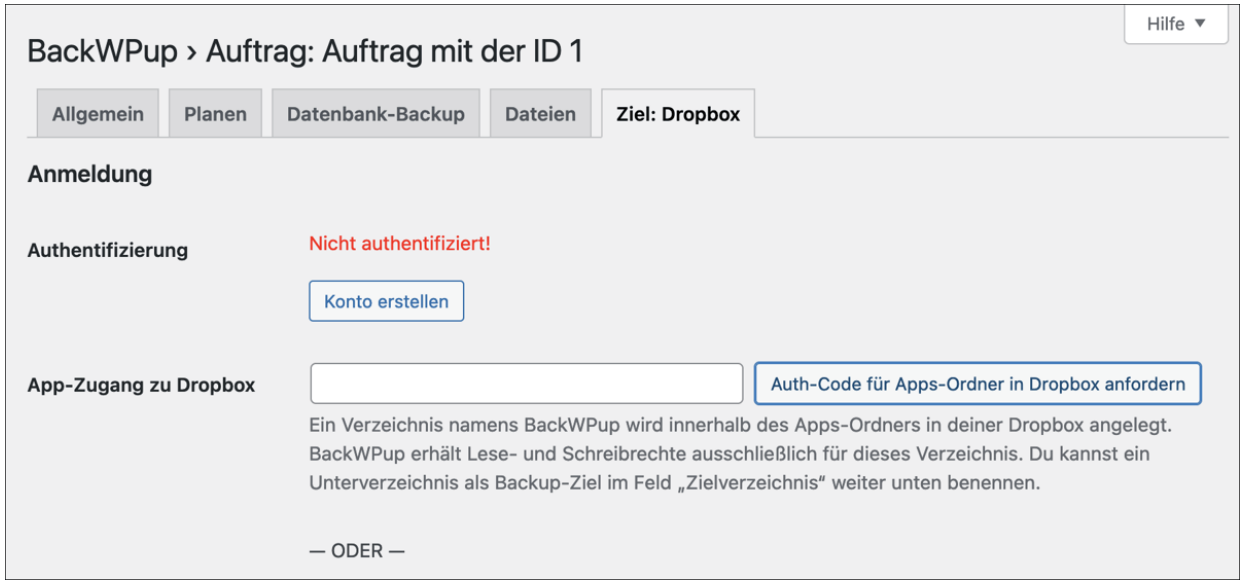


Abbildung 16.14 Im Tab zum Speichern in der Dropbox gibt es den Button »Auth-Code für Apps-Ordner in Dropbox anfordern«, mit dem Sie den Prozess starten können, um Ihrer Website Zugriff auf die Dropbox zu geben.

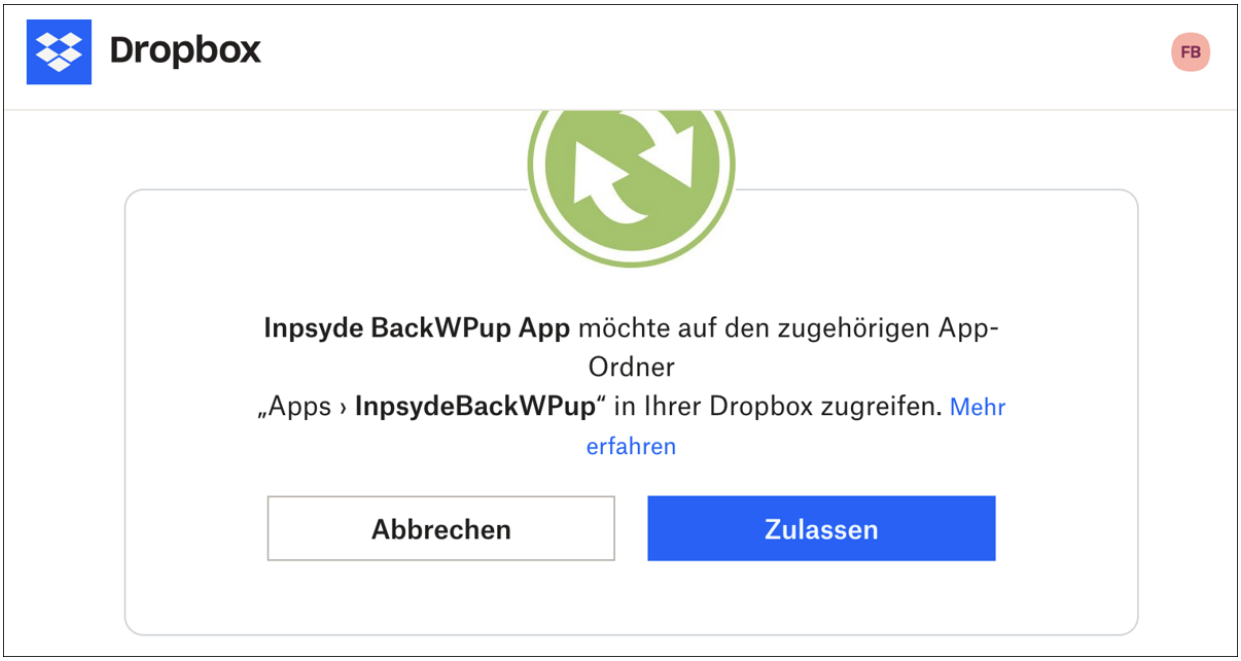


Abbildung 16.15 Sie müssen Ihrer Website Zugriff auf Ihre Dropbox geben, damit die Backups von Ihrer Website in der Dropbox gespeichert werden können.

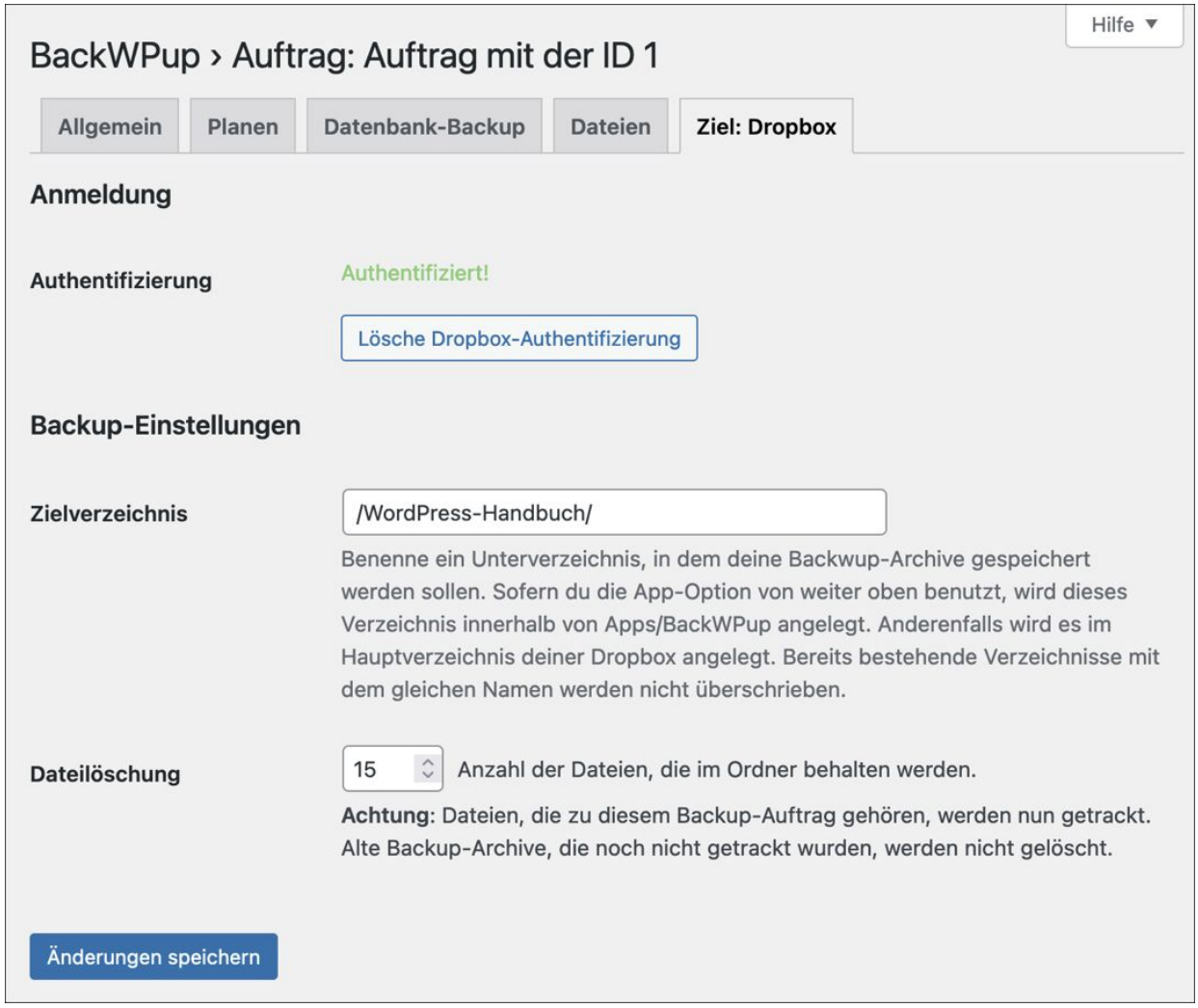


Abbildung 16.16 So oder so ähnlich sollte der Dropbox-Tab nach der erfolgreichen Verbindung mit Ihrem Dropbox-Konto aussehen.

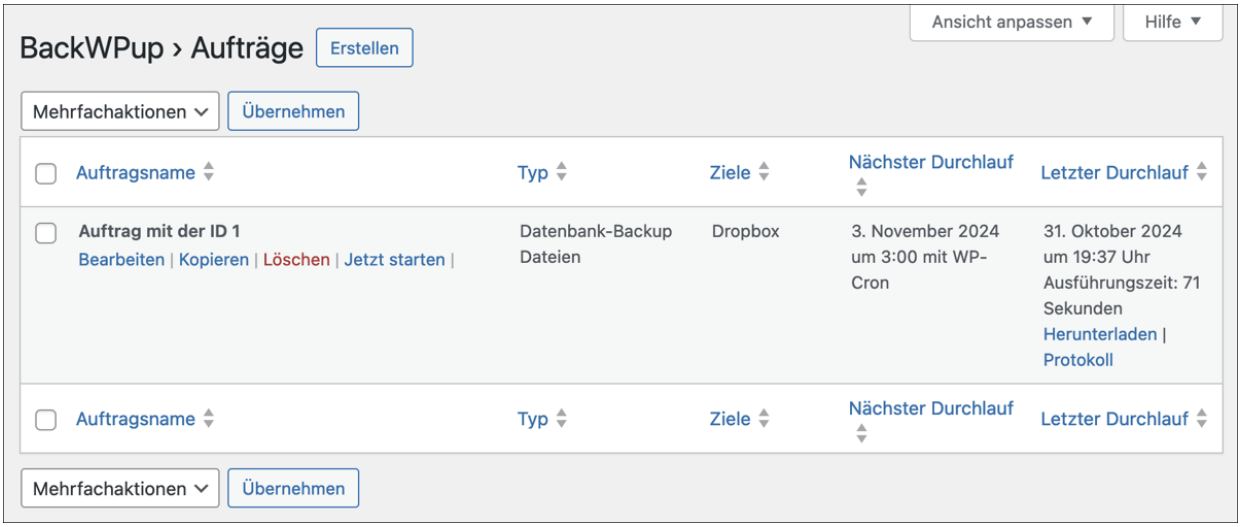


Abbildung 16.17 In der Übersicht der Aufträge können Sie unter anderem einen Auftrag direkt starten, auch wenn er regelmäßig automatisch ausgeführt wird.

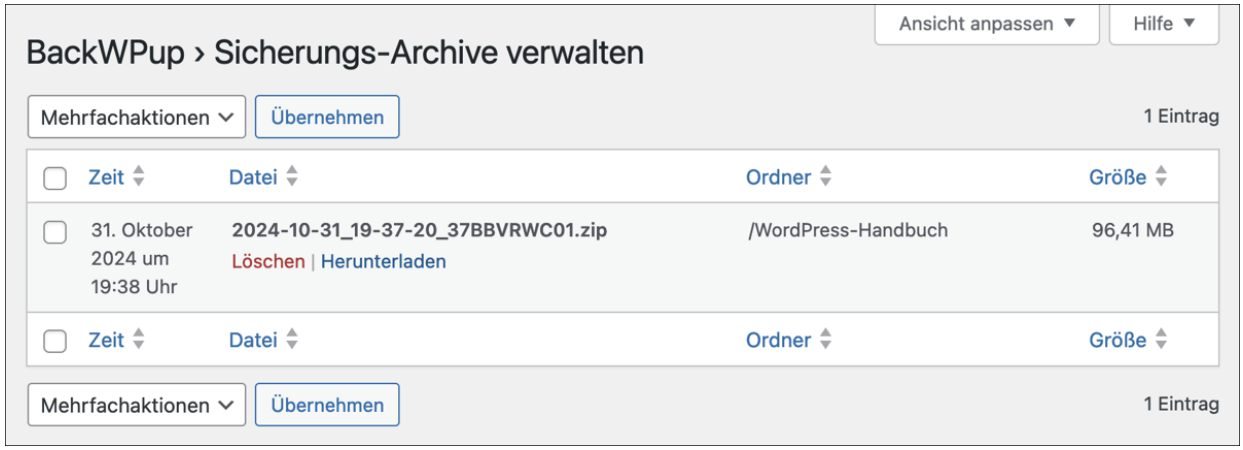


Abbildung 16.18 Unter »BackWPup« • »Backups« können Sie sich die gemachten Backups anschauen und Backups herunterladen.

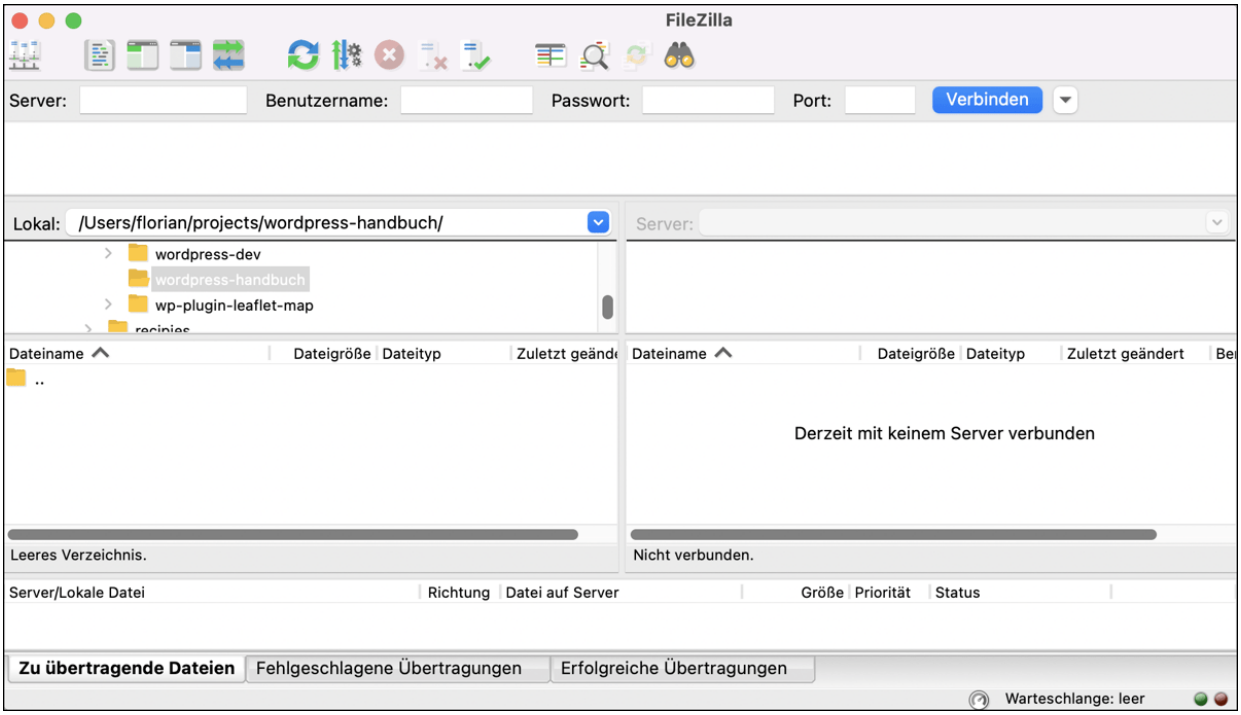


Abbildung 17.1 Auf der linken Seite sehen Sie in FileZilla Ihre lokalen Dateien und Ordner, rechts wird erst etwas angezeigt, wenn Sie sich mit einem Server verbunden haben.

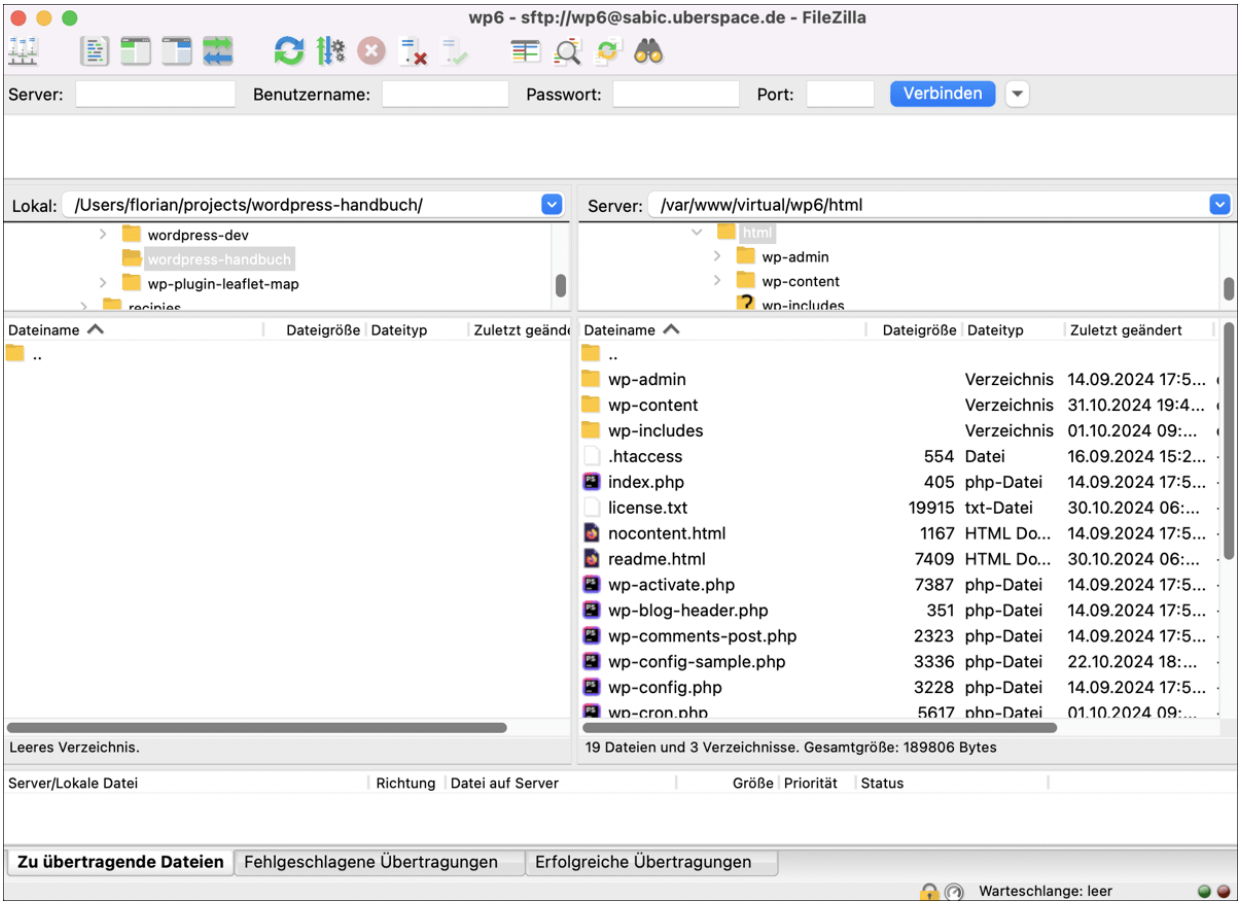


Abbildung 17.2 Nachdem Sie sich mit dem Server verbunden haben, wechseln Sie lokal zu einem leeren Verzeichnis und auf dem Server zum WordPress-Verzeichnis.

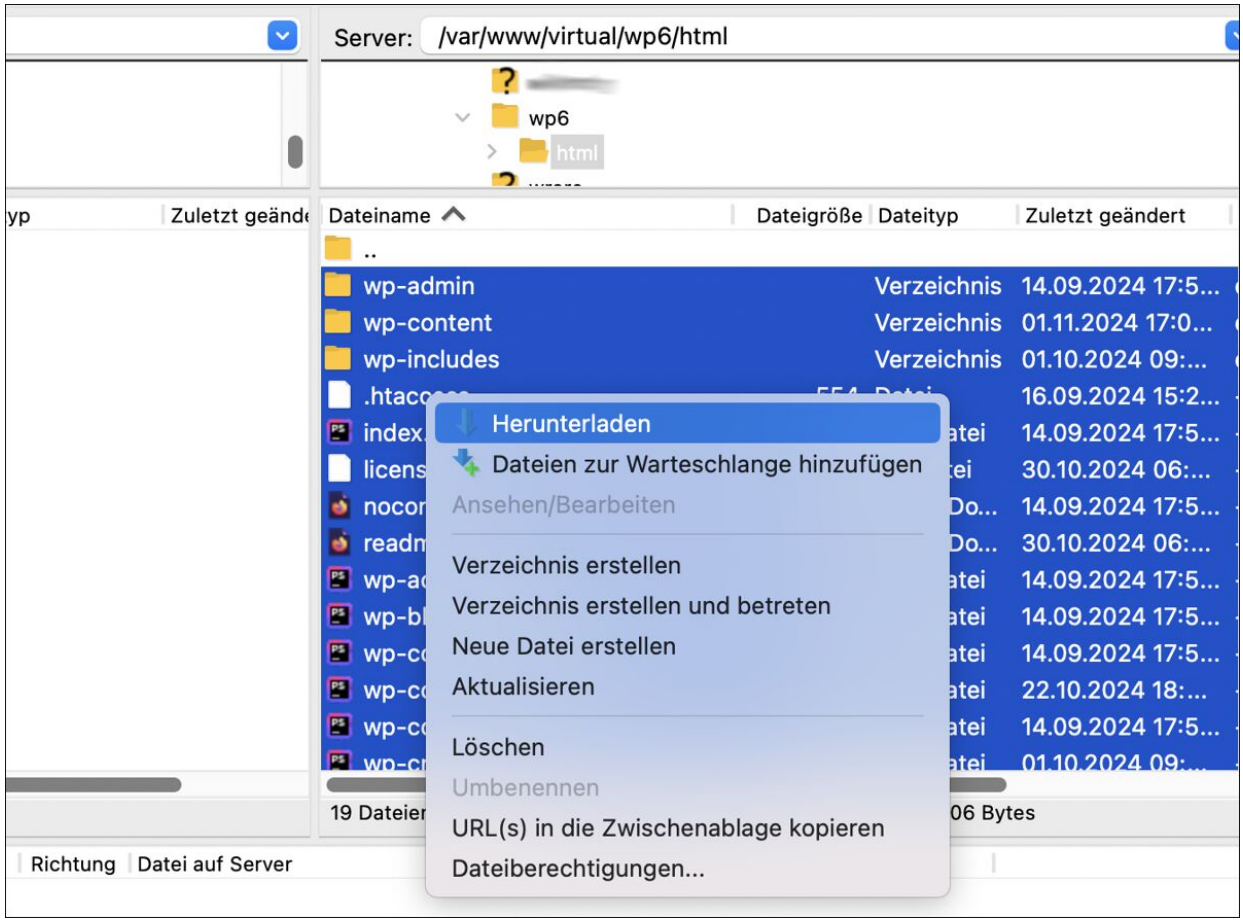


Abbildung 17.3 Über einen Rechtsklick können Sie die markierten Dateien vom Webserver auf Ihren lokalen Server herunterladen.

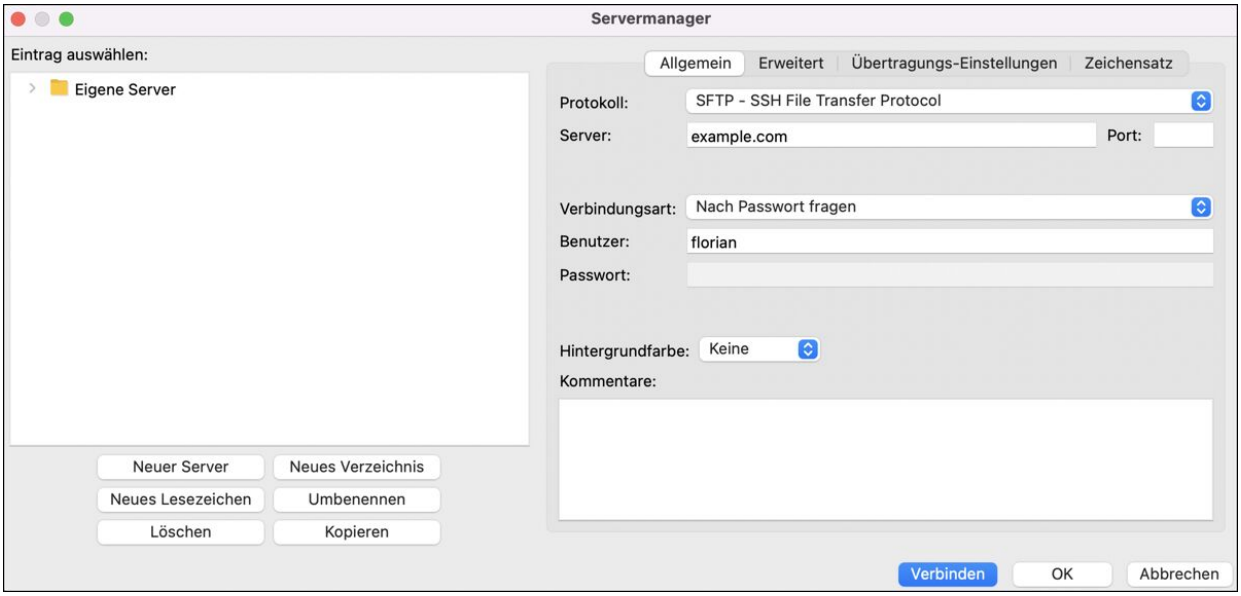


Abbildung 17.4 Im Servermanager von FileZilla können Sie Verbindungseinstellungen speichern.

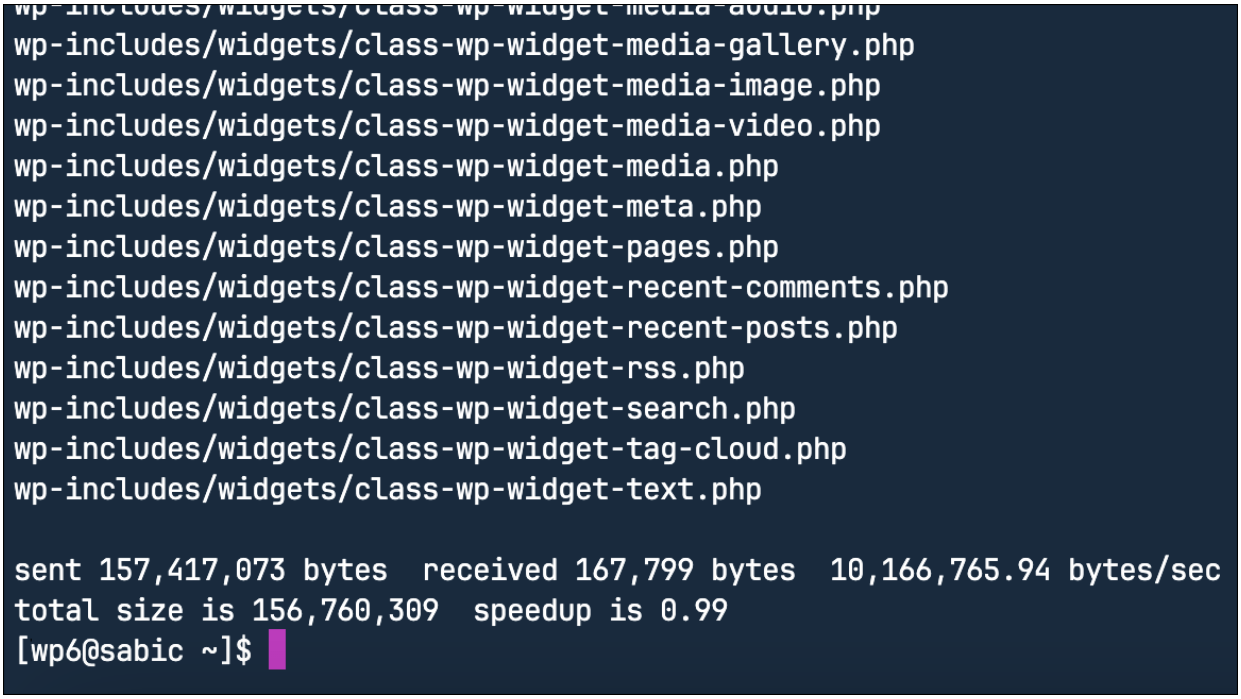


Abbildung 17.5 Mit »rsync« können Sie Dateien direkt zwischen zwei Servern übertragen, wenn Sie zu beiden einen SSH-Zugang haben.

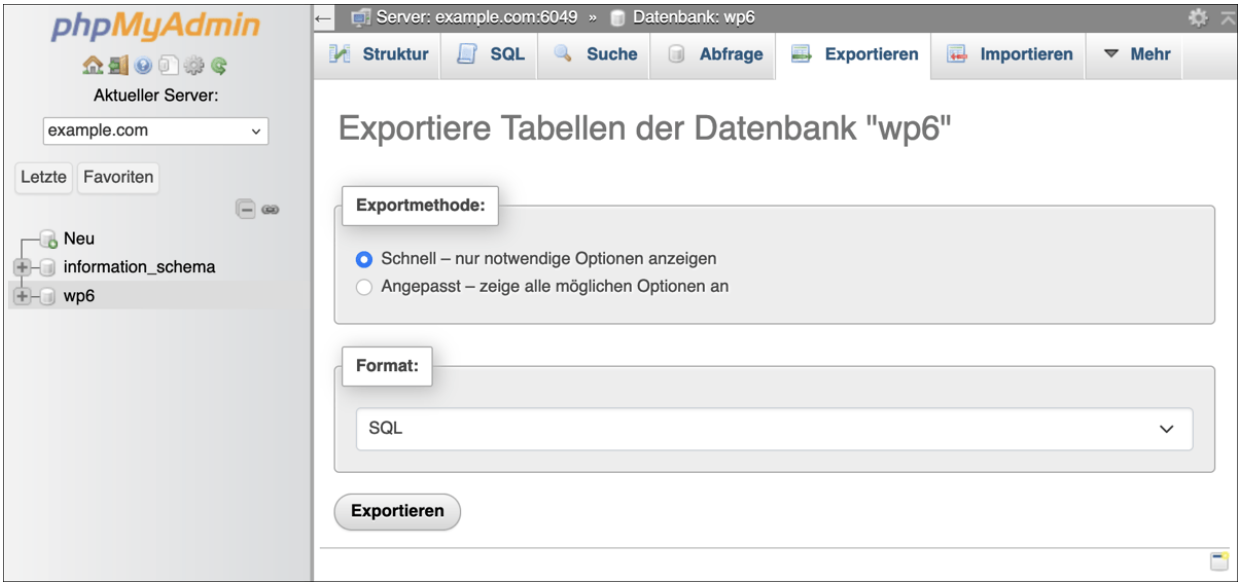


Abbildung 17.6 Die Standard-Einstellungen sollten für den Export meist passen, wie hier bei phpMyAdmin.

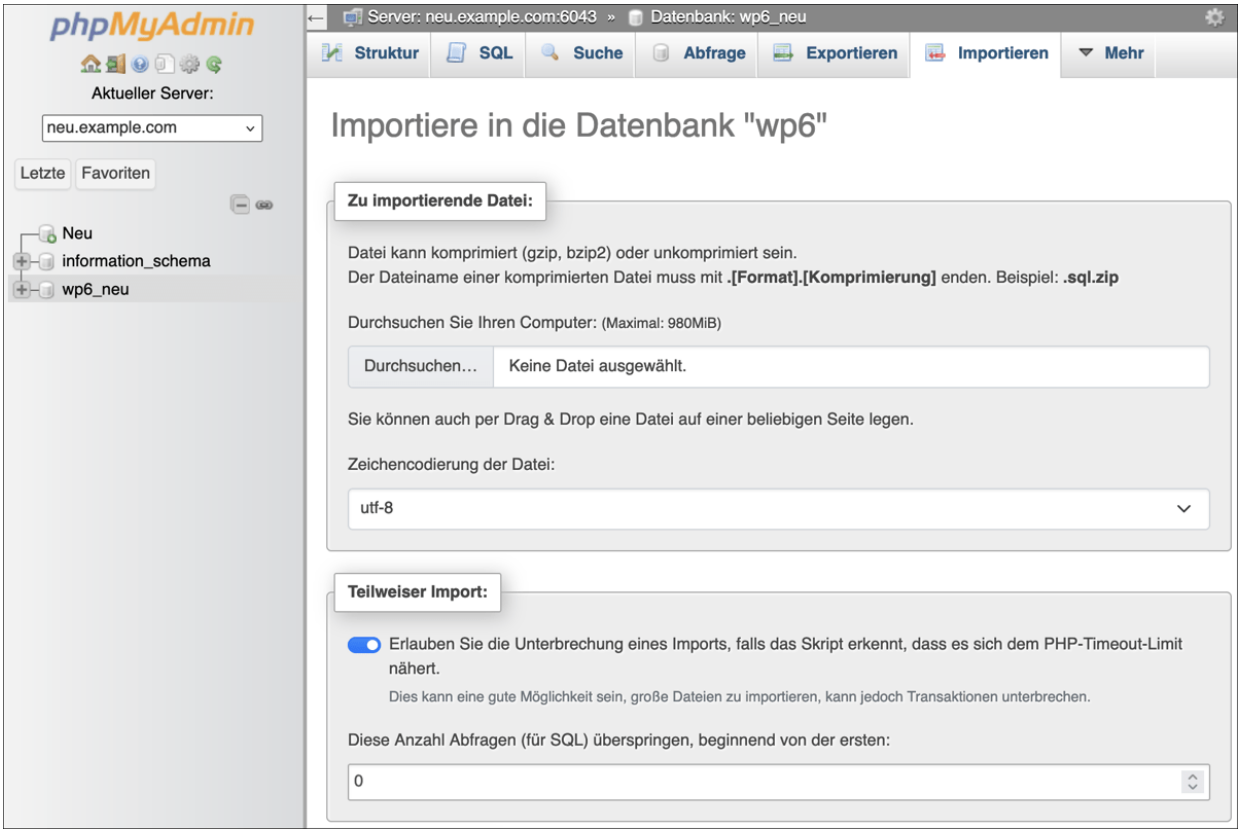


Abbildung 17.7 Im Import-Tab laden Sie die exportierte SQL-Datei hoch und importieren die Daten in eine leere Datenbank.

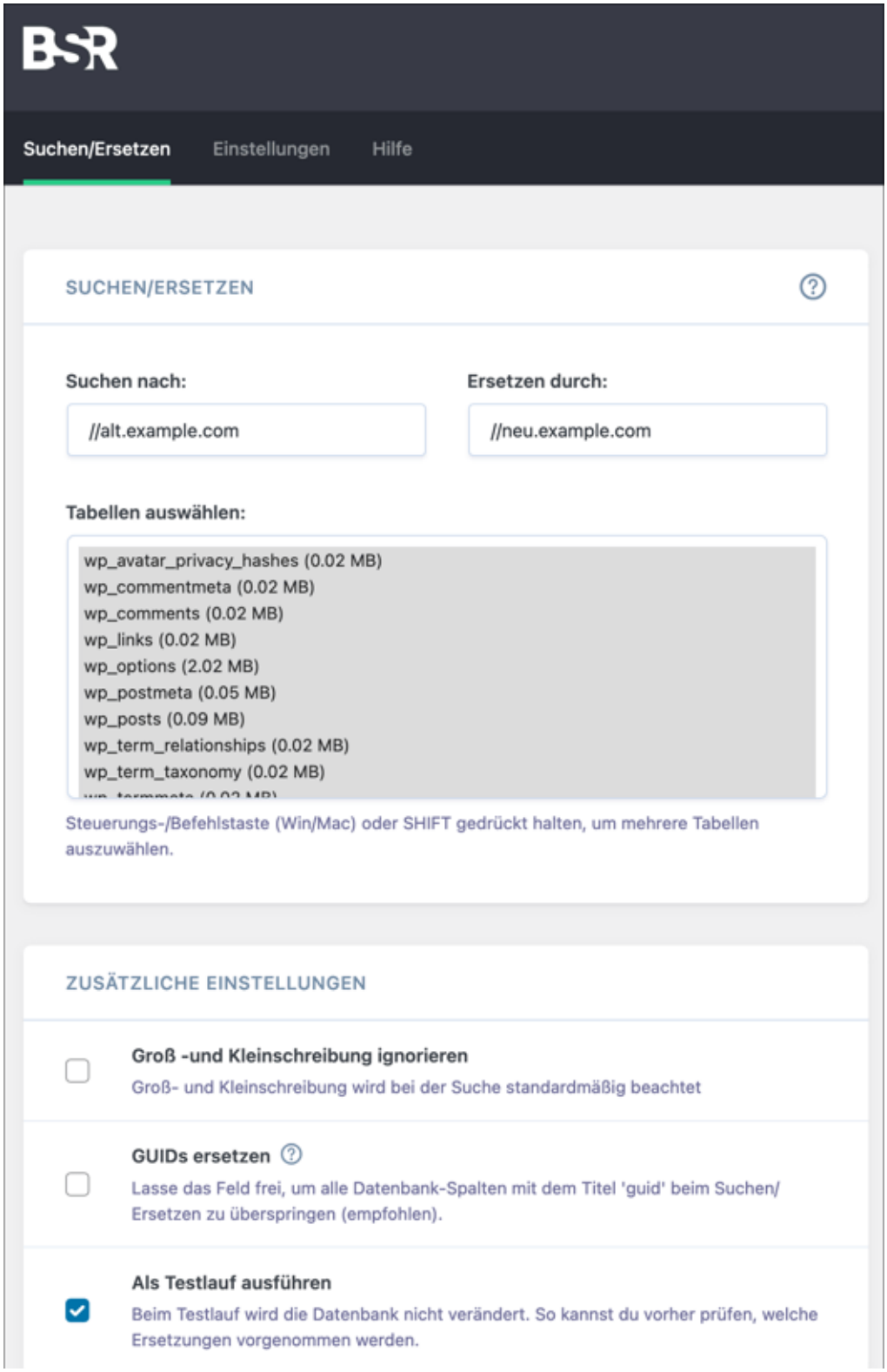

BSR
Suchen/Ersetzen
Einstellungen
Hilfe
SUCHEN/ERSETZEN
Suchen nach:
//alt.example.com
Ersetzen durch:
//neu.example.com
Tabellen auswählen:
wp_avatar_privacy_hashes (0.02 MB)
wp_commentmeta (0.02 MB)
wp_comments (0.02 MB)
wp_links (0.02 MB)
wp_options (2.02 MB)
wp_postmeta (0.05 MB)
wp_posts (0.09 MB)
wp_term_relationships (0.02 MB)
wp_term_taxonomy (0.02 MB)
Steuerungs-/Befehlstaste (Win/Mac) oder SHIFT gedrückt halten, um mehrere Tabellen auszuwählen.
ZUSÄTZLICHE EINSTELLUNGEN
Groß -und Kleinschreibung ignorieren
Groß- und Kleinschreibung wird bei der Suche standardmäßig beachtet
GUIDs ersetzen
Lasse das Feld frei, um alle Datenbank-Spalten mit dem Titel 'guid' beim Suchen/ Ersetzen zu überspringen (empfohlen).
Als Testlauf ausführen
Beim Testlauf wird die Datenbank nicht verändert. So kannst du vorher prüfen, welche Ersetzungen vorgenommen werden.



Abbildung 17.8 Mit »Better Search Replace« können Sie Werte in der Datenbank suchen und durch andere ersetzen, in unserem Fall die alte Domain durch die neue.

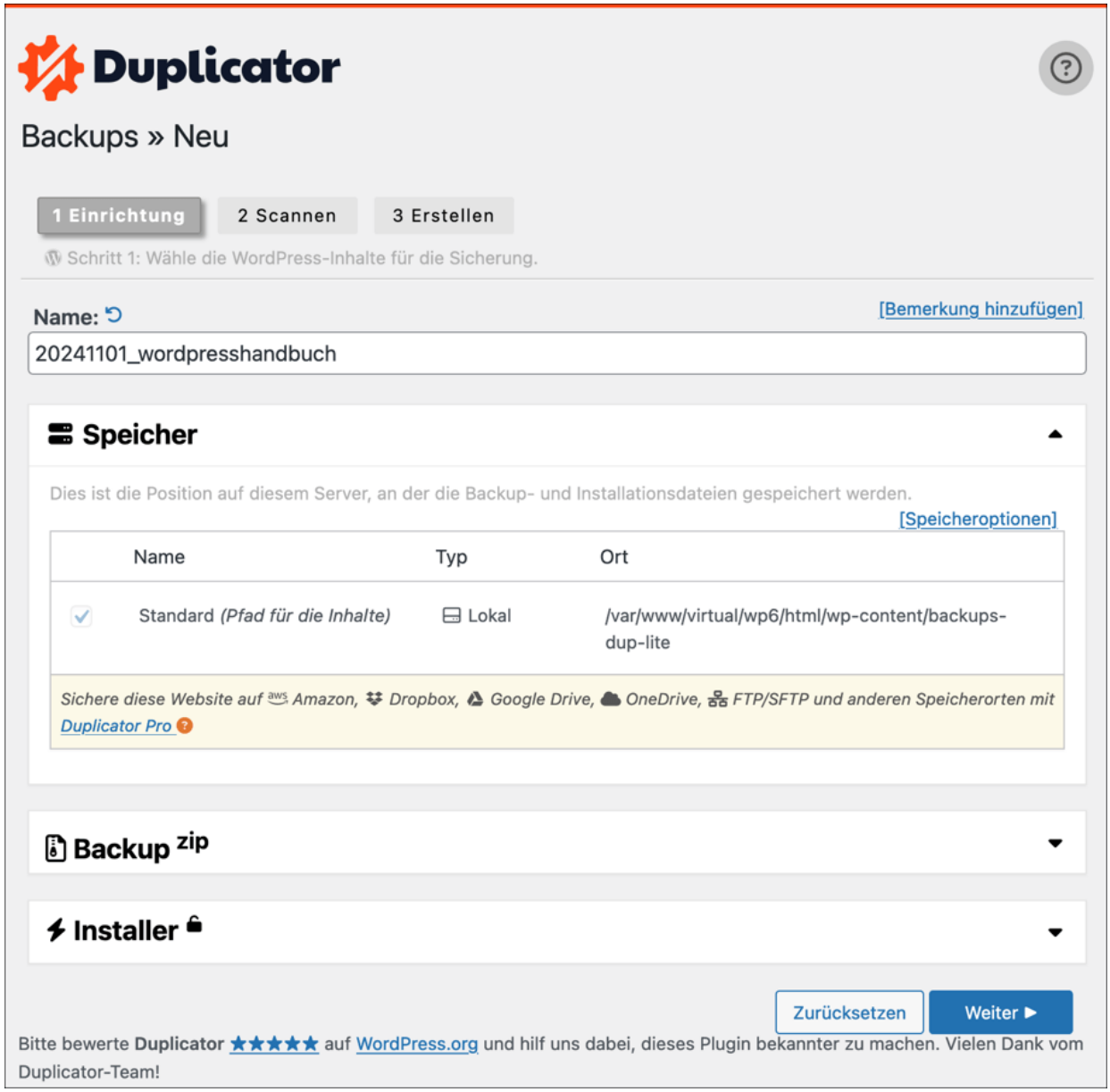


Abbildung 17.9 Duplicator gibt Ihnen die Möglichkeit, Einstellungen für die Erstellung eines Backups zu ändern, etwa Dateien oder Ordner auszuschließen. Hier können Sie bei den Standard-Einstellungen bleiben.

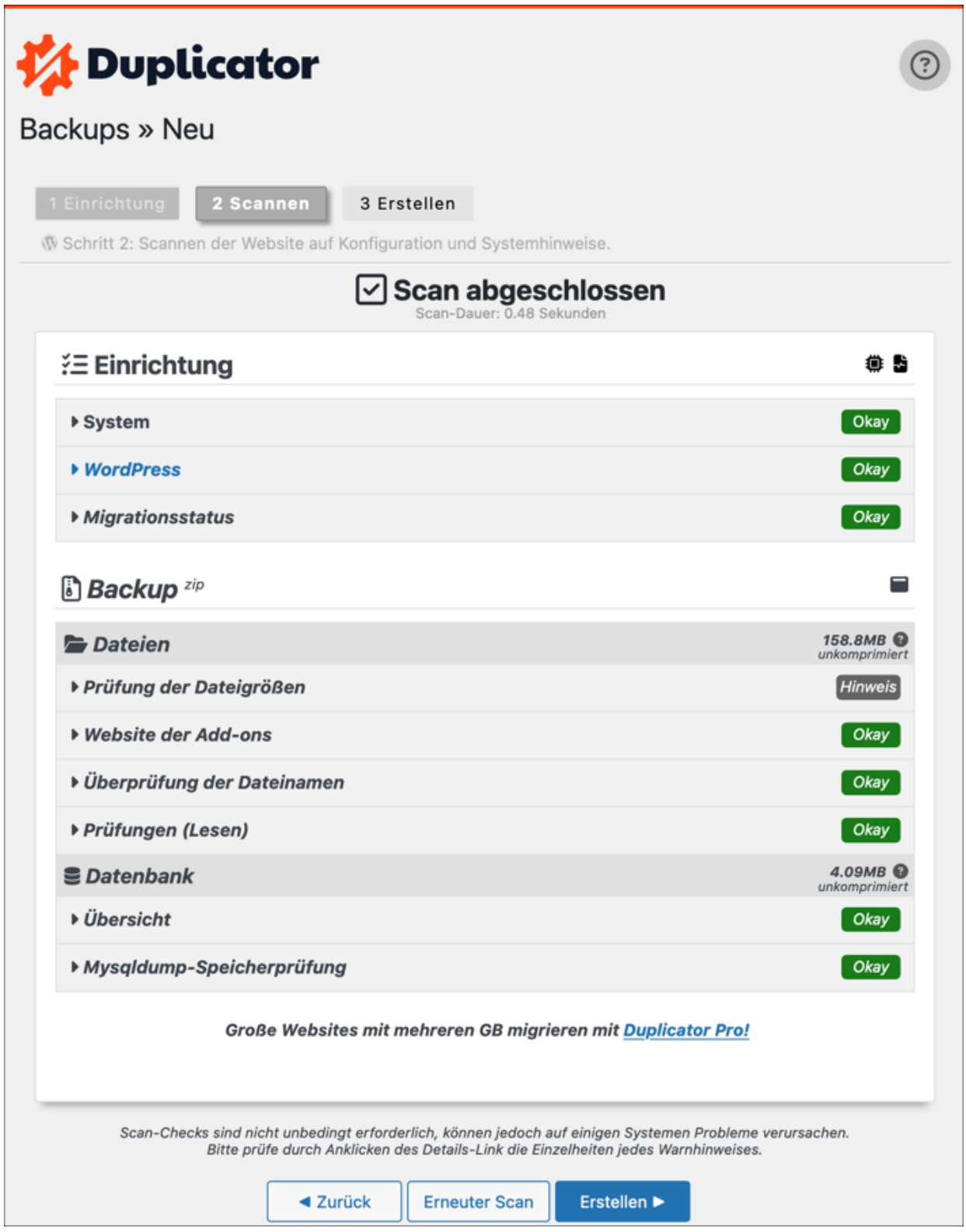

Duplicator
Backups » Neu
1 Einrichtung
2 Scannen
3 Erstellen
Schritt 2: Scannen der Website auf Konfiguration und Systemhinweise.
Scan abgeschlossen
Scan-Dauer: 0.48 Sekunden
Einrichtung
System
Okay
WordPress
Okay
Migrationsstatus
Okay
Backup zip
Dateien
158.8MB
unkomprimiert
Prüfung der Dateigrößen
Hinweis
Website der Add-ons
Okay
Überprüfung der Dateinamen
Okay
Prüfungen (Lesen)
Okay
Datenbank
4.09MB
unkomprimiert
Übersicht
Okay
Mysqldump-Speicherprüfung
Okay
Große Websites mit mehreren GB migrieren mit Duplicator Pro!
Scan-Checks sind nicht unbedingt erforderlich, können jedoch auf einigen Systemen Probleme verursachen.
Bitte prüfe durch Anklicken des Details-Link die Einzelheiten jedes Warnhinweises.
◄ Zurück
Erneuter Scan
Erstellen ►

Abbildung 17.10 Vor der Erstellung des Backups prüft Duplicator, ob es irgendwelche Probleme gibt, die einer problemlosen Erstellung im Weg stehen könnten.

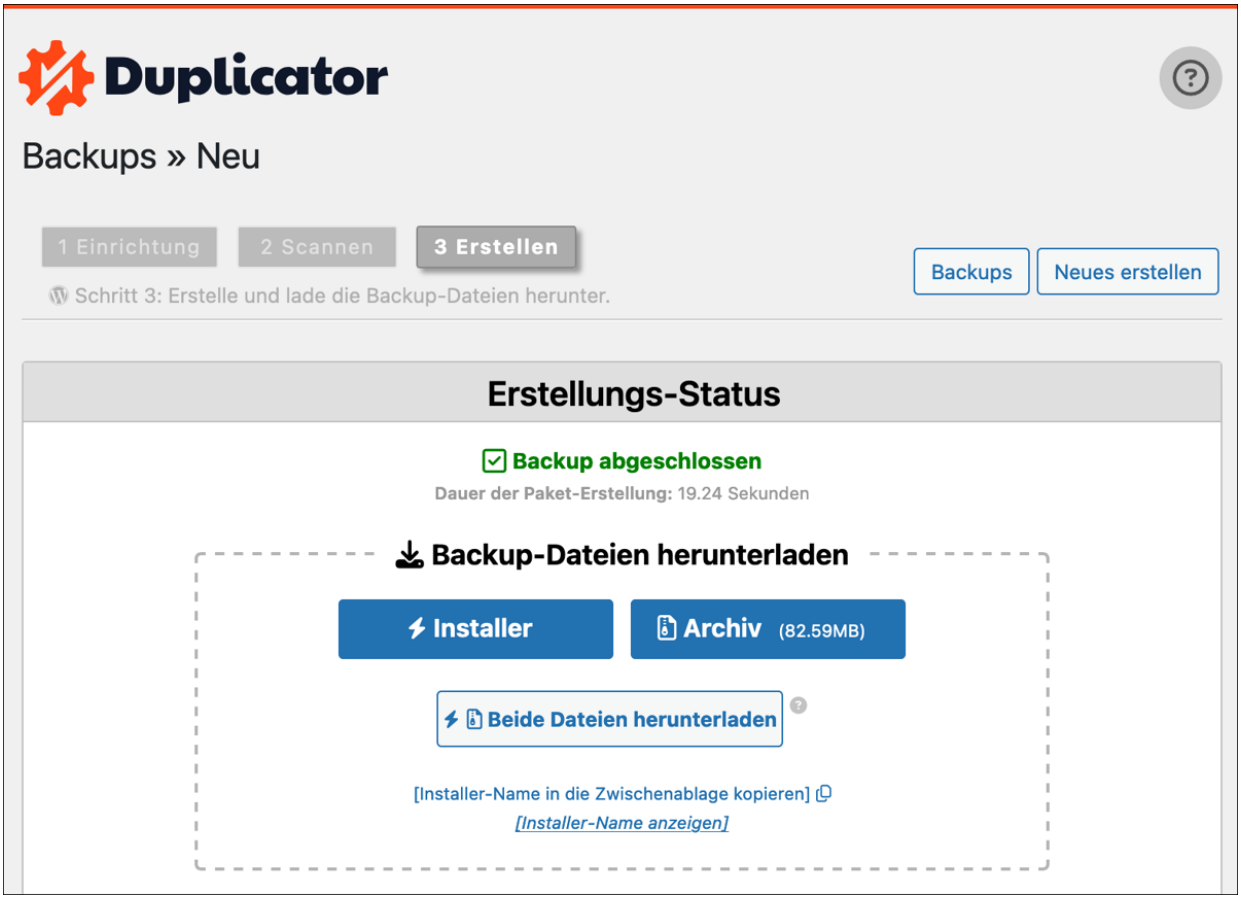


Abbildung 17.11 Laden Sie sich sowohl den »Installer« als auch das »Archiv« herunter.

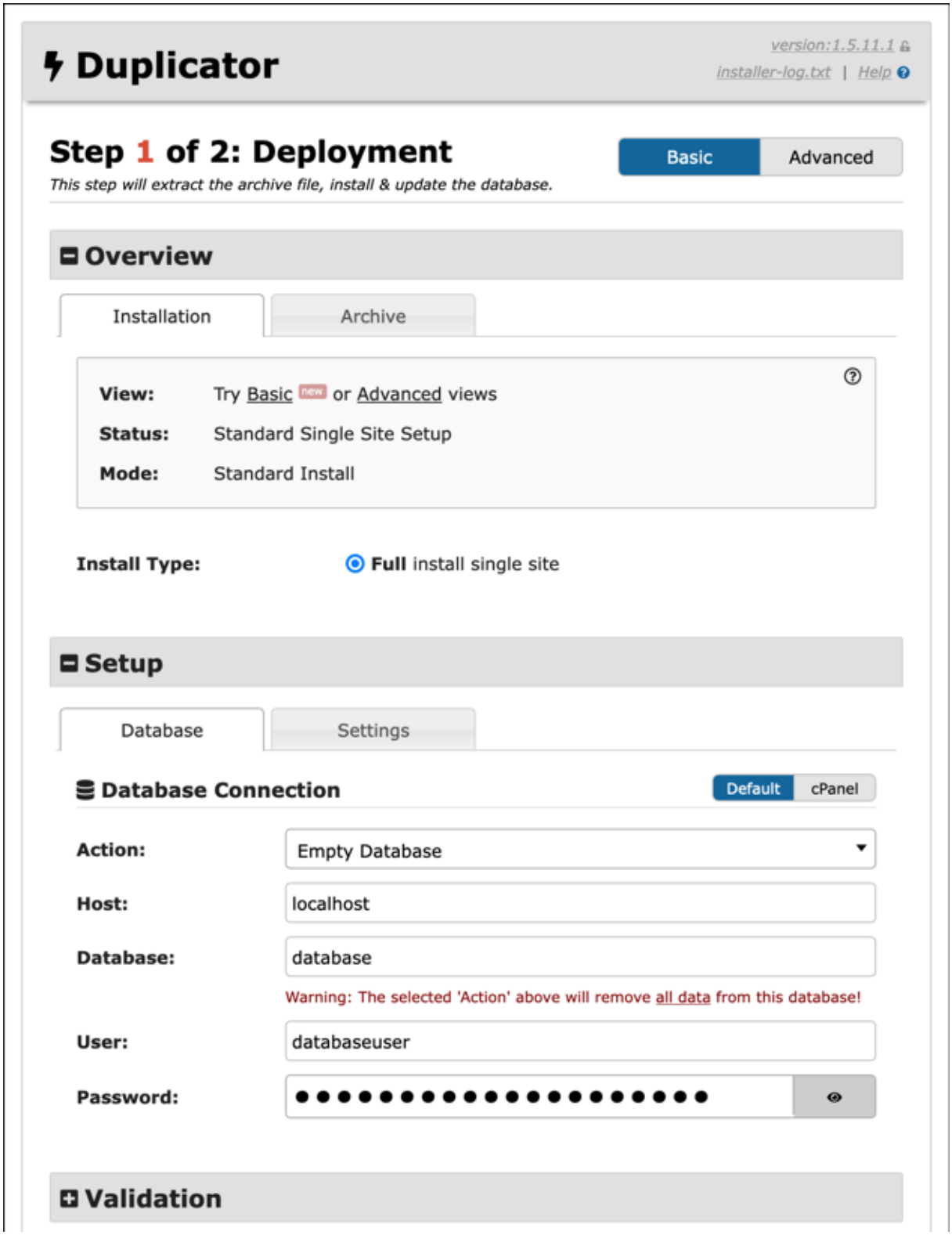

Duplicator
version:1.5.11.1
installer-log.txt | Help
Step 1 of 2: Deployment
This step will extract the archive file, install & update the database.
Basic
Advanced
Overview
Installation
Archive
View: Try Basic new or Advanced views
Status: Standard Single Site Setup
Mode: Standard Install
Install Type: Full install single site
Setup
Database
Settings
Database Connection
Default
cPanel
Action: Empty Database
Host: localhost
Database: database
Warning: The selected 'Action' above will remove all data from this database!
User: databaseuser
Password:
Validation



Abbildung 17.12 Im Installer von Duplicator geben Sie die neuen Datenbank-Daten an.

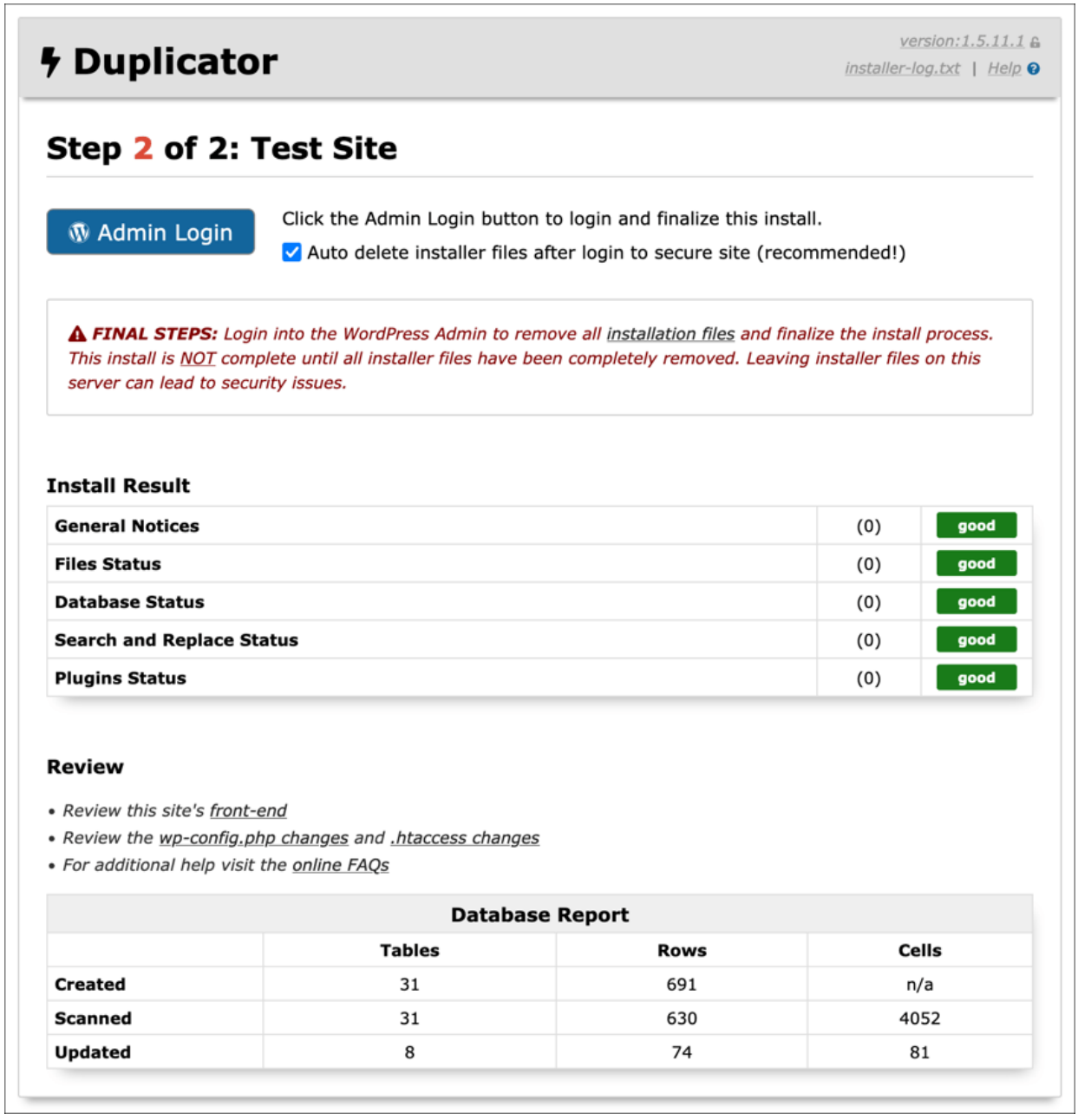


Abbildung 17.13 Und das war es auch schon: Die Website ist wiederhergestellt, und Sie können ins Backend wechseln.

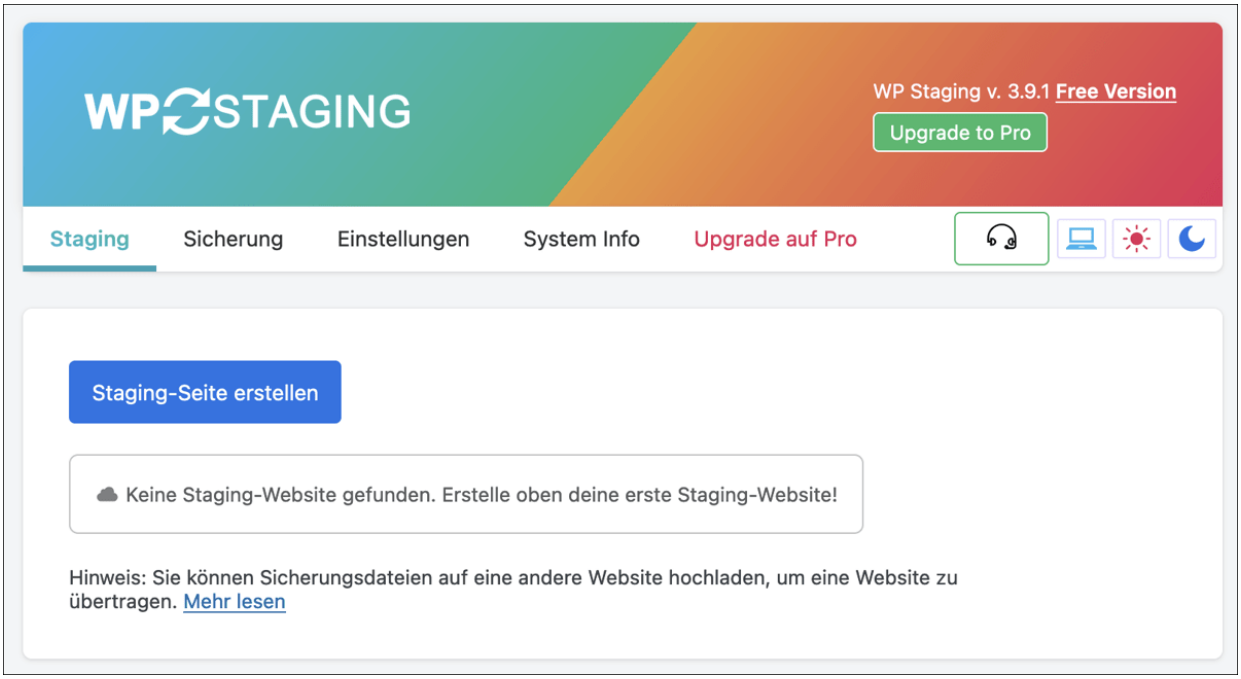


Abbildung 17.14 Mit dem Plugin »WP Staging« können Sie einfach eine Kopie Ihrer Website in einem Unterverzeichnis auf demselben Server erstellen.

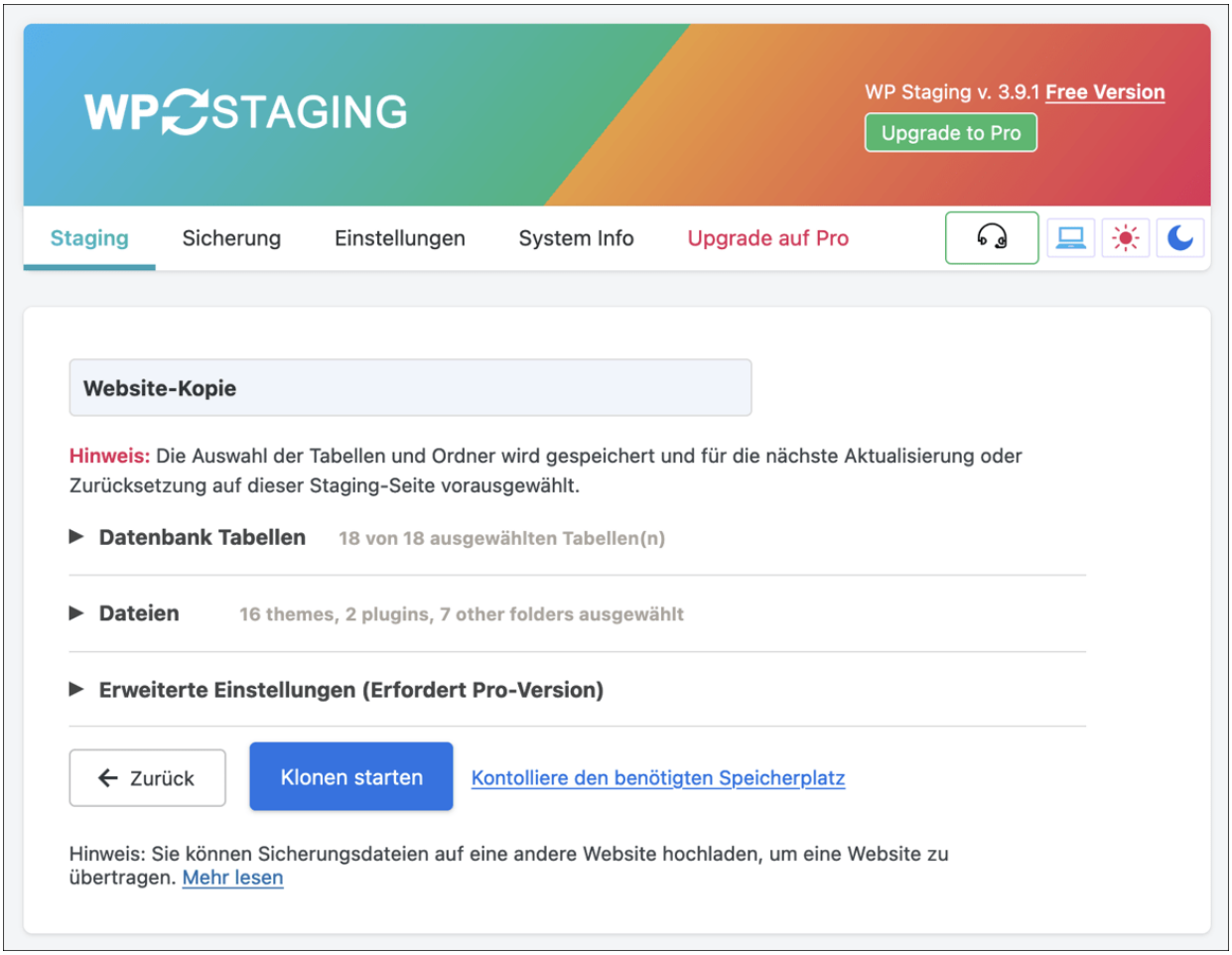


Abbildung 17.15 Sie können einen Namen für die Kopie angeben und danach direkt auf »Klonen starten« klicken, um die Kopie zu starten.

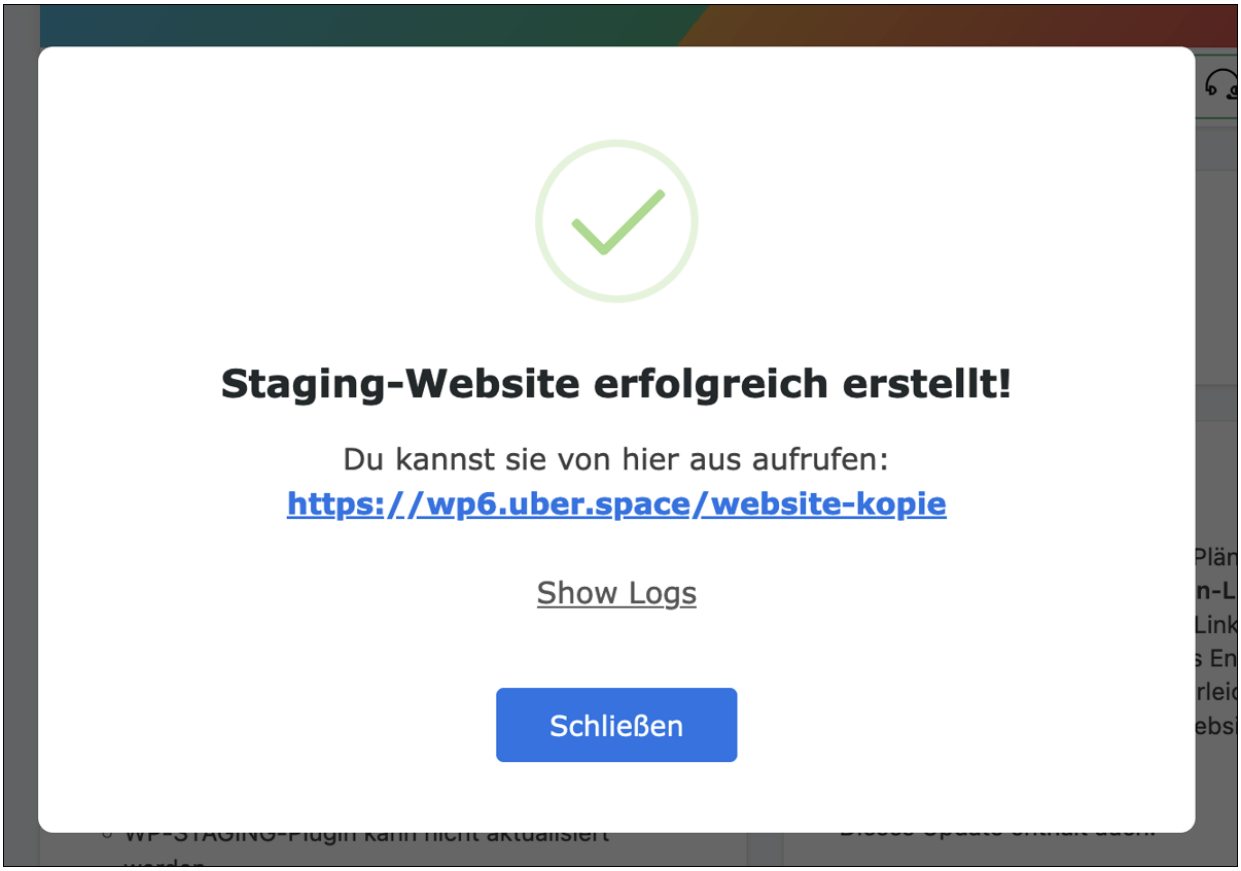


Abbildung 17.16 Nach erfolgreichem Erstellen der Kopie bekommen Sie in einem Modal den Link zur Kopie angezeigt.

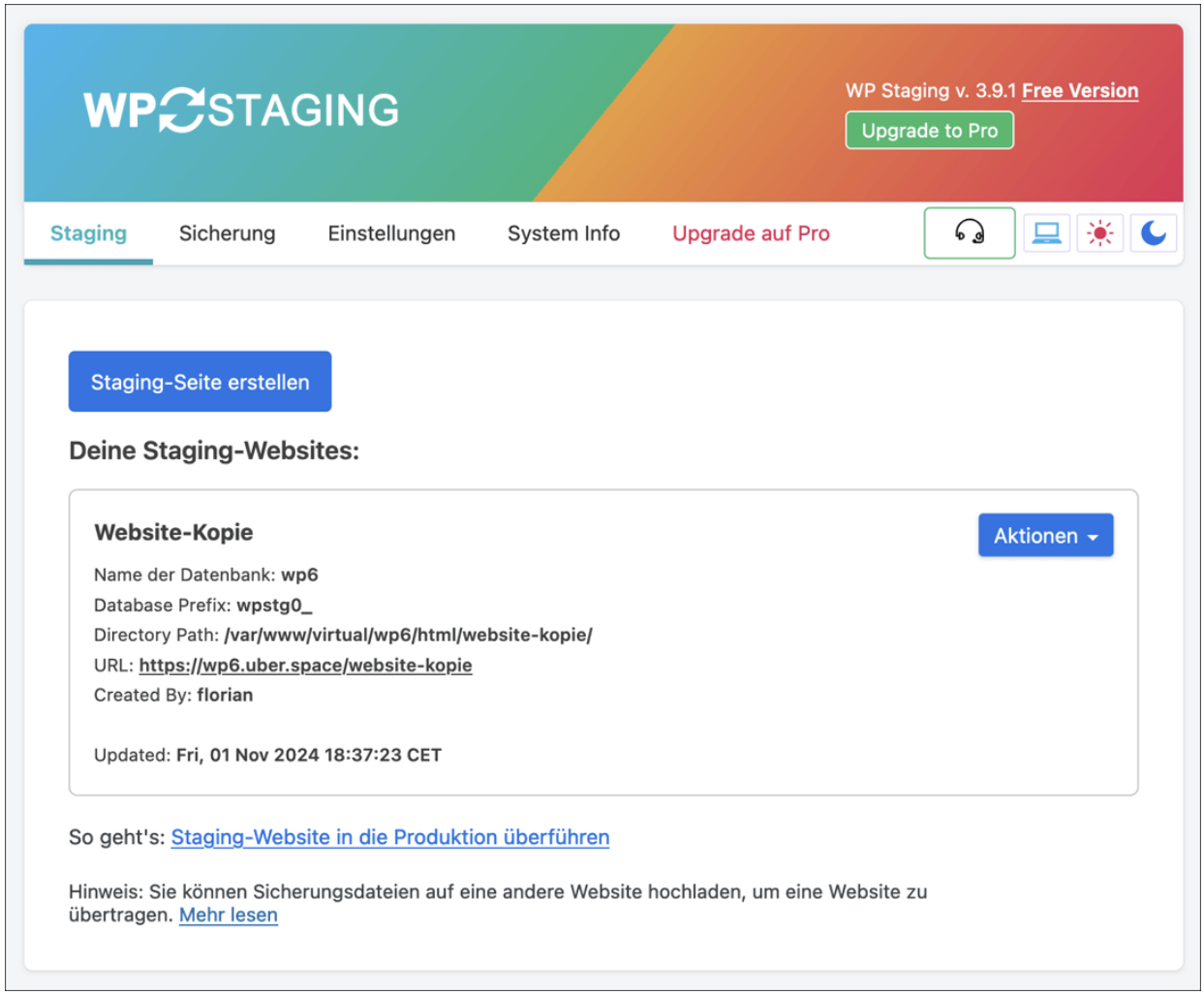


Abbildung 17.17 Sie können die Website-Kopie über den »Aktionen«-Toggle löschen oder auf den aktuellen Stand der Live-Site aktualisieren.

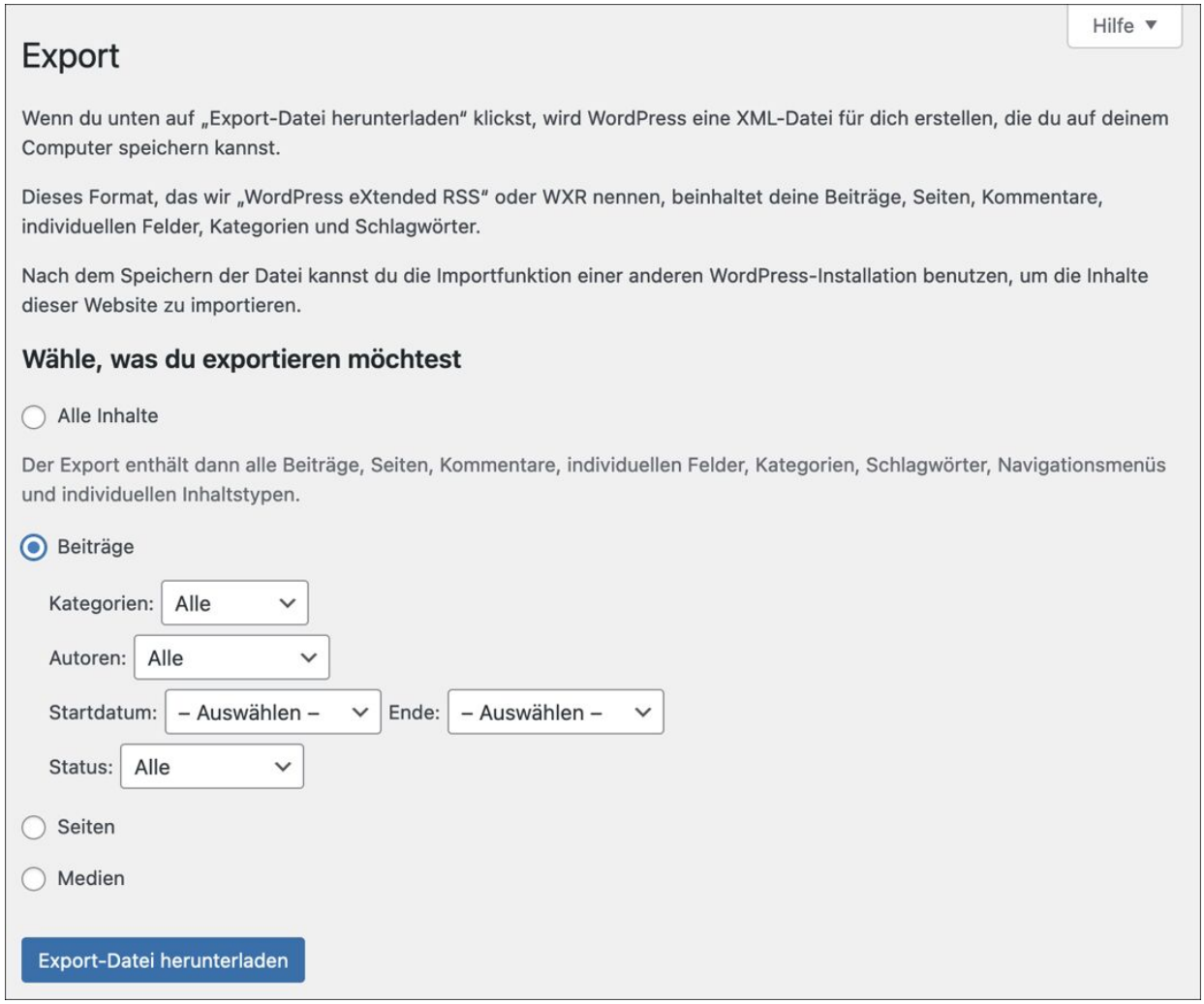


Abbildung 17.18 WordPress bietet Ihnen die Möglichkeit, Ihre Daten oder nur Teile davon als XML zu exportieren.

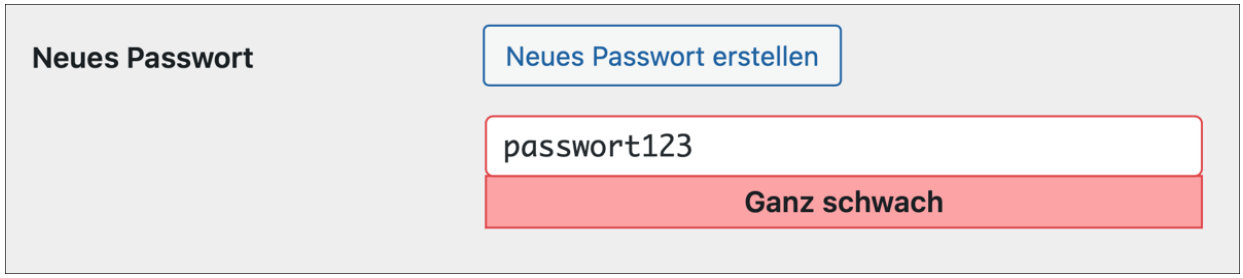


Abbildung 18.1 WordPress gibt Ihnen eine direkte Rückmeldung dazu, wie sicher ein Passwort ist, das Sie verwenden möchten.

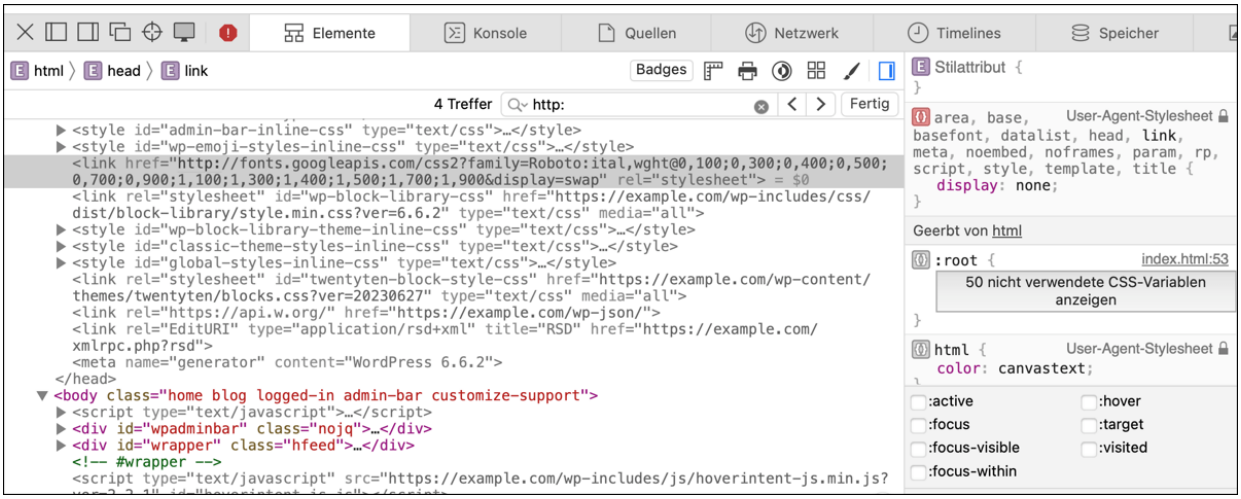


Abbildung 18.2 Zur Identifizierung der problematischen Elemente werfen Sie einen Blick in den Quelltext und suchen nach »http:«.

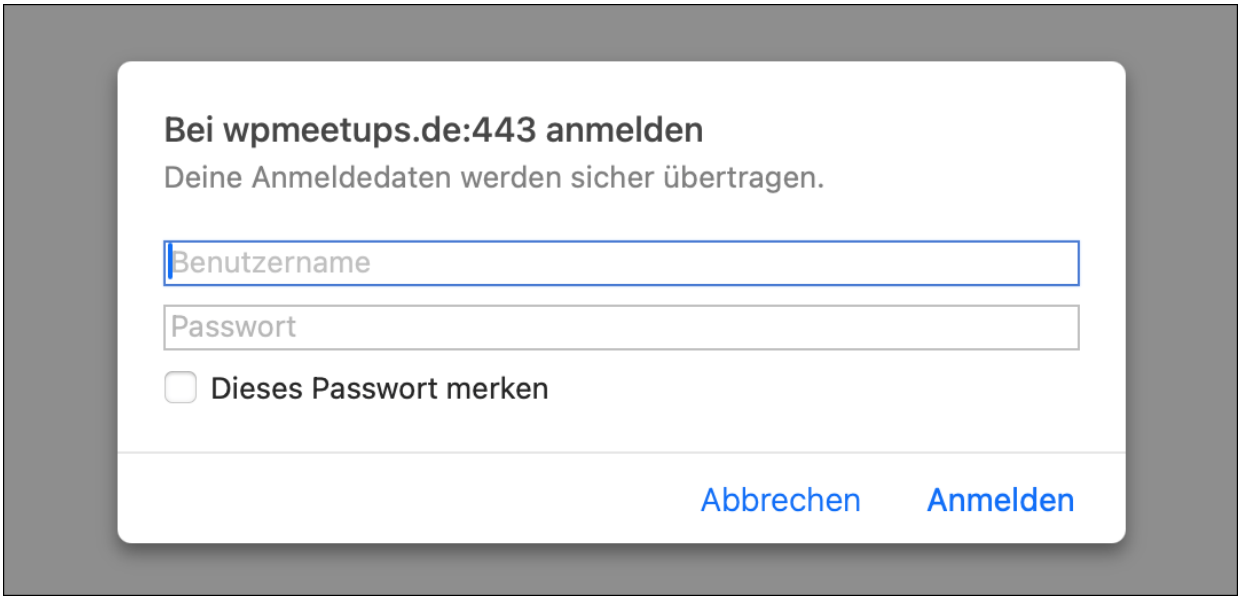


Abbildung 18.3 Über die »Basic Authentication« des Apache Webservers setzen Sie eine weitere Passwortsperre noch vor den Login-Mechanismus von WordPress.

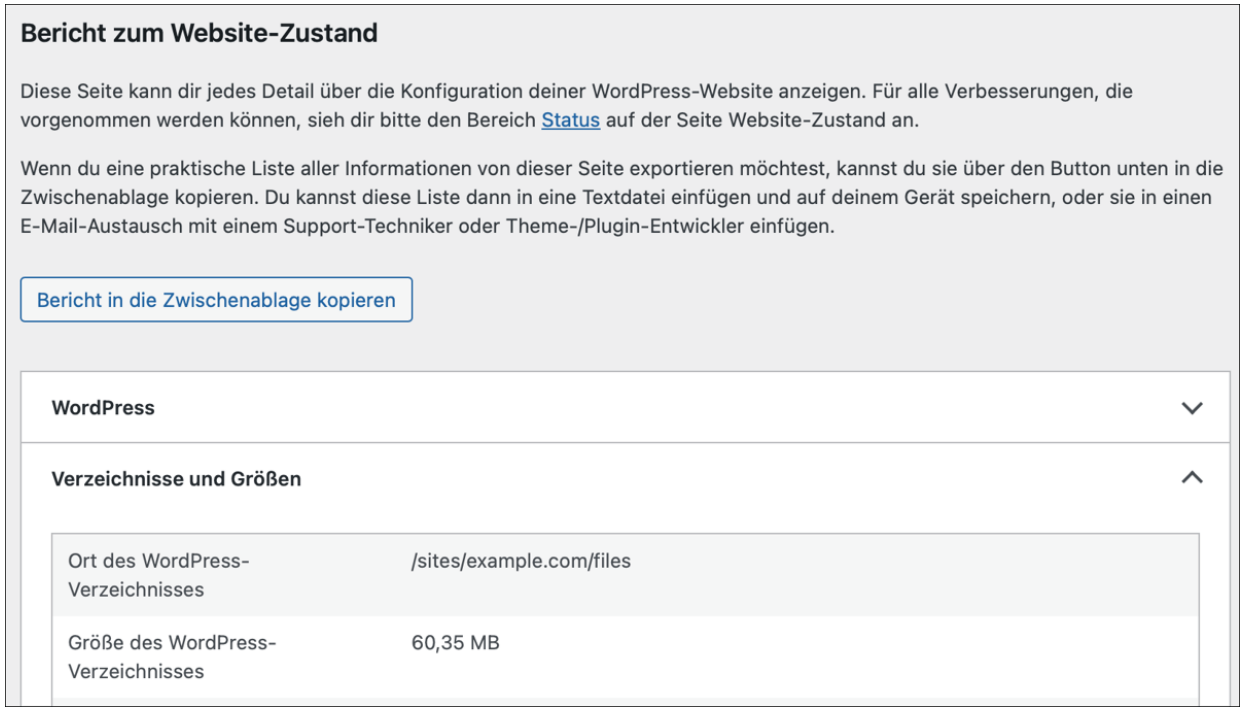


Abbildung 18.4 Im Tool »Website-Zustand« lässt sich der Serverpfad zur WordPress-Installation einfach herausfinden.

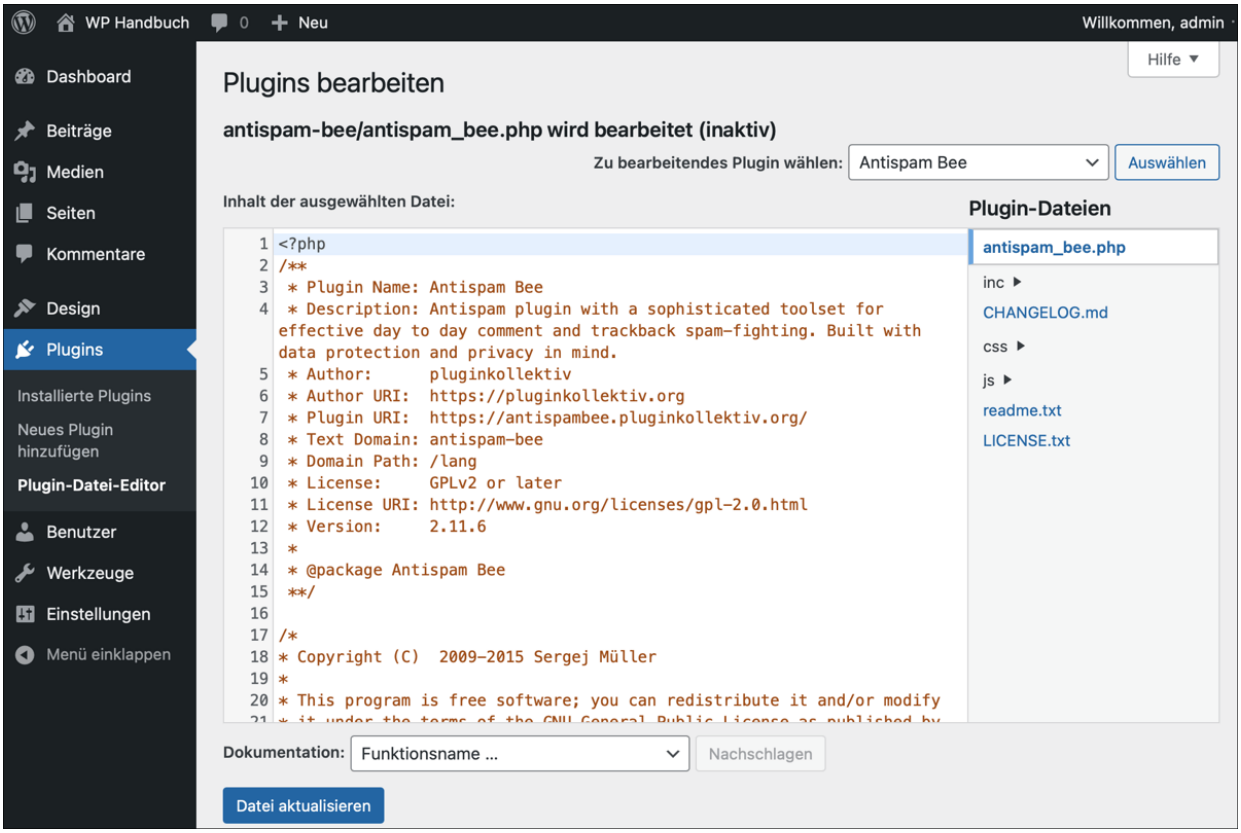


Abbildung 18.5 Im Datei-Editor können Sie Plugin- und Theme-Dateien bearbeiten und damit Probleme verursachen.

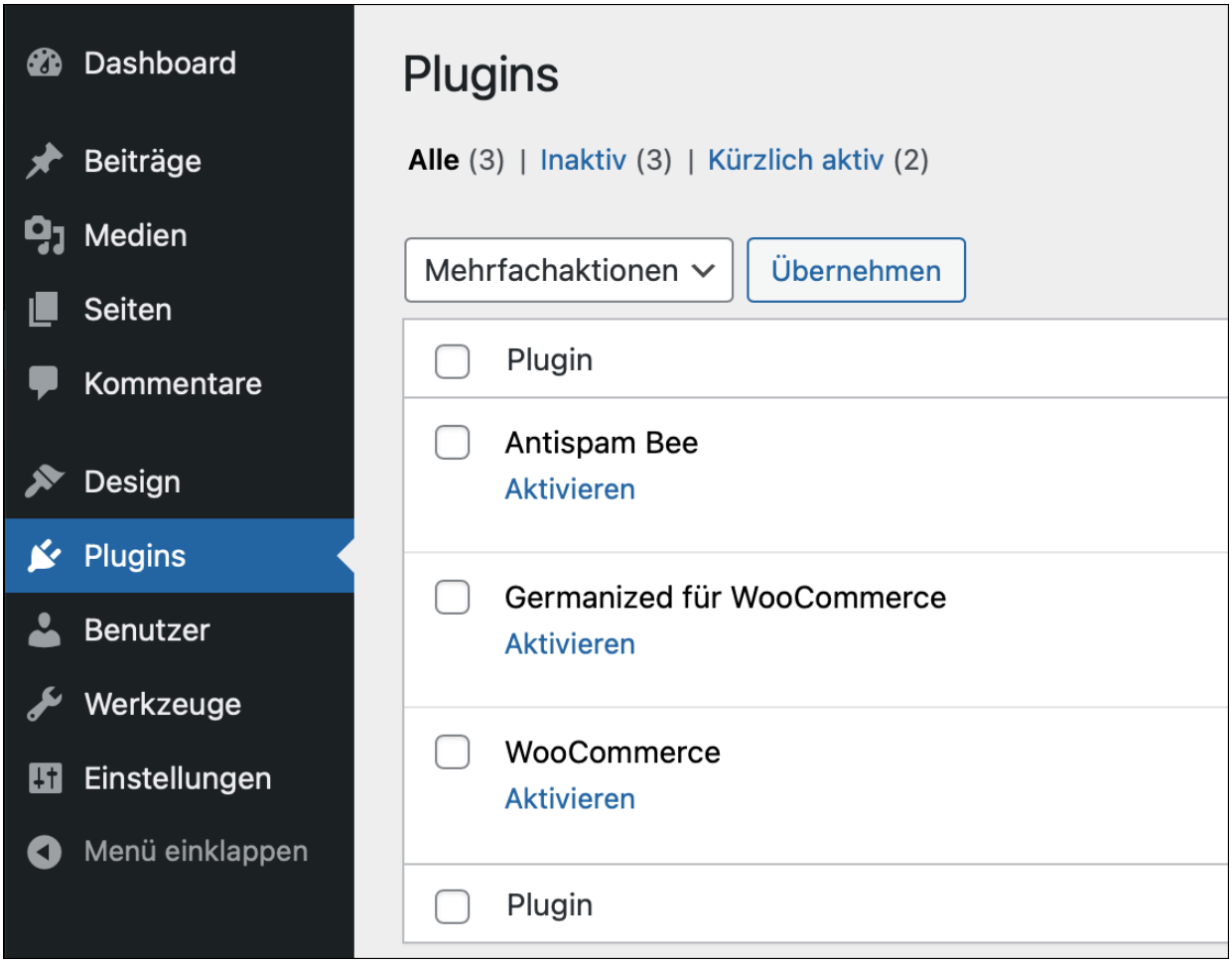


Abbildung 18.6 Wenn Sie das Installieren von Plugins und Themes deaktivieren, wird auch das Löschen installierter Plugins und Themes unmöglich.

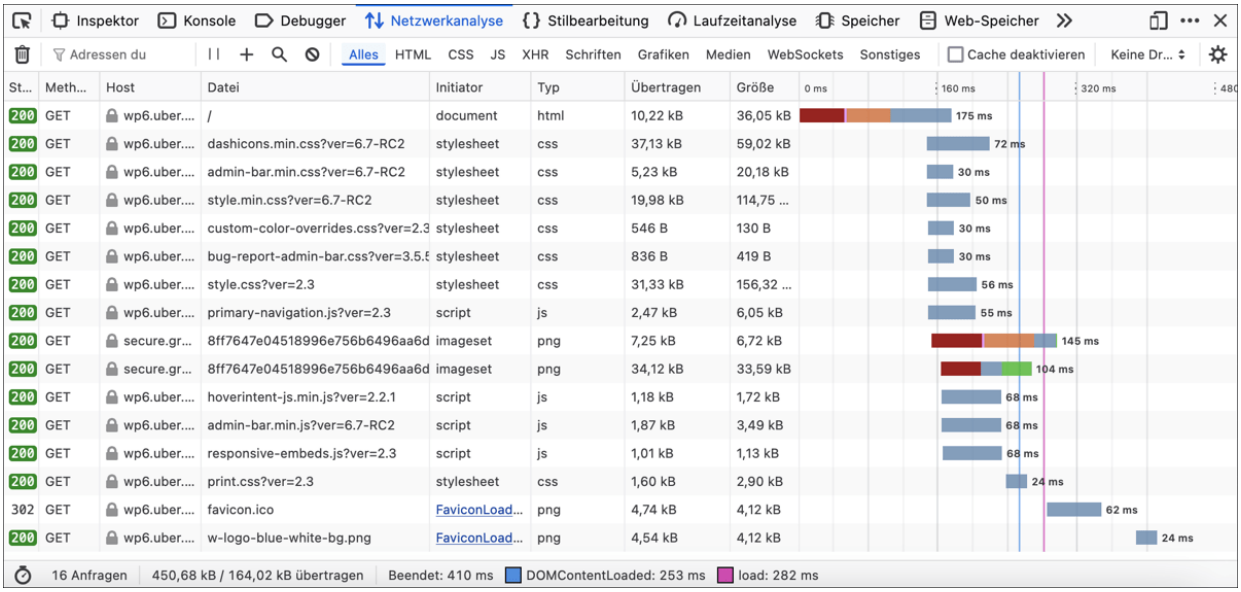


Abbildung 19.1 Sie können über den Netzwerk-Tab (Benennung je nach Browser unterschiedlich) einen Einblick in das Ladeverhalten Ihrer Website erhalten.

PageSpeed Insights

Link kopieren Dokumentation

Datum des Berichts: 02.11.2024, 10:33:35

https://wp6.uber.space/ Analysieren

Mobil Computer

So sieht die Leistung auf der Nutzerseite aus

Keine Daten

Leistungsprobleme diagnostizieren

Leistung	Barrierefreiheit	Best Practices	SEO
100	100	100	54

100

Leistung

Die Werte sind geschätzt und können variieren. Die Leistungsbewertung wird direkt aus diesen Messwerten berechnet. Siehe Rechner.

▲ 0–49 ■ 50–89 ● 90–100



MESSWERTE Ansicht maximieren

First Contentful Paint 1,2 s	Largest Contentful Paint 1,2 s
Total Blocking Time 0 ms	Cumulative Layout Shift 0
Speed Index 1,4 s	

Abbildung 19.2 Der erste Teil der Performanceanalyse mit PageSpeed Insights zeigt die Messwerte für die Bewertung der »Leistung«.

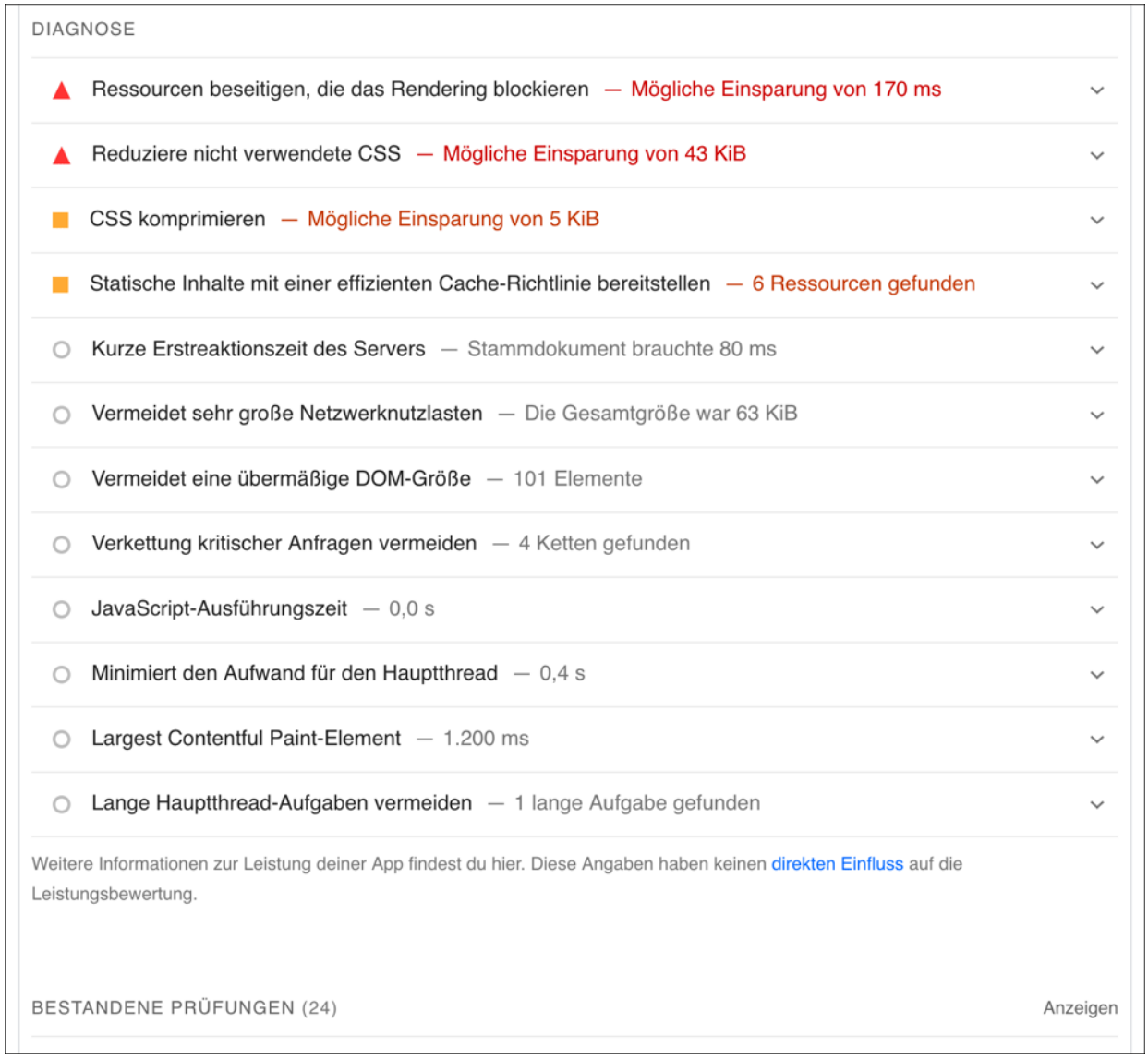


Abbildung 19.3 Unter »Diagnose« werden Punkte aufgelistet, bei denen Sie aus Sicht von PageSpeed Insights tätig werden könnten.

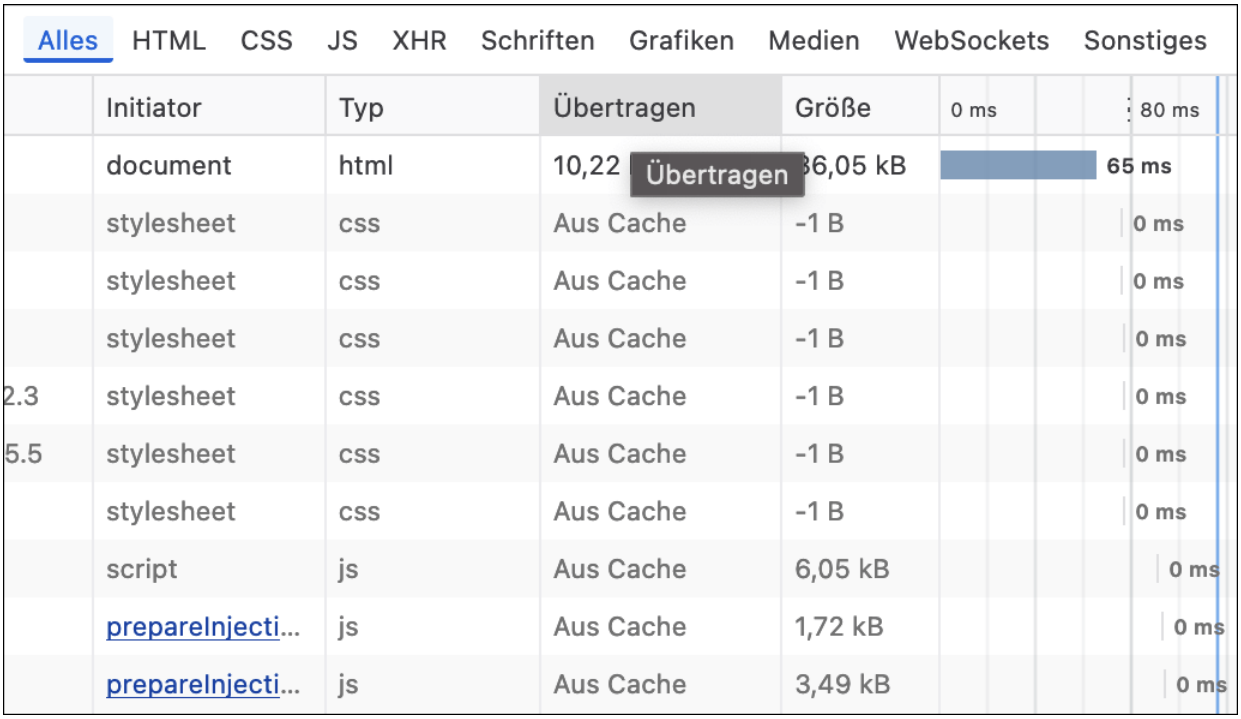


Abbildung 19.4 In Firefox können Sie in der Spalte »Übertragen« sehen, ob eine Anfrage aus dem Cache beantwortet wurde.

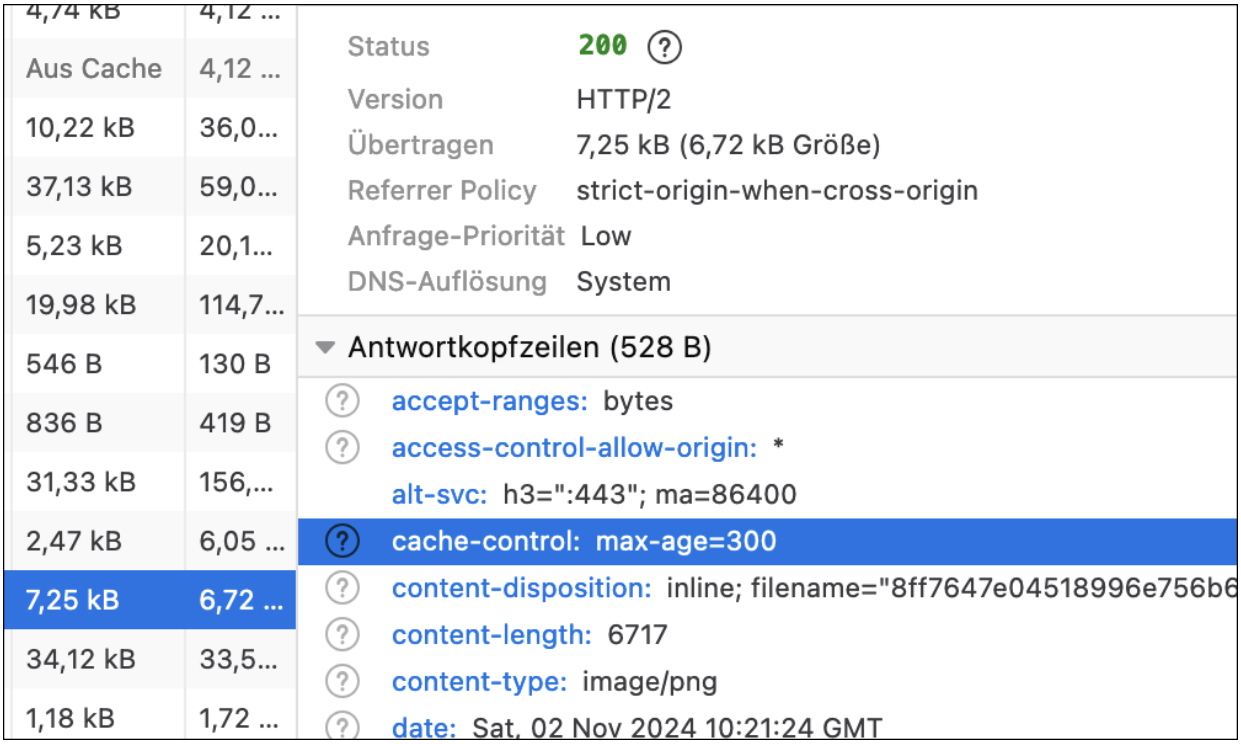


Abbildung 19.5 Anhand des Response-Headers »Cache Control« können Sie erkennen, ob vom Server eine Cache-Policy für eine bestimmte Ressource angegeben wurde. Das ist der Teil, auf den PageSpeed Insights prüft.

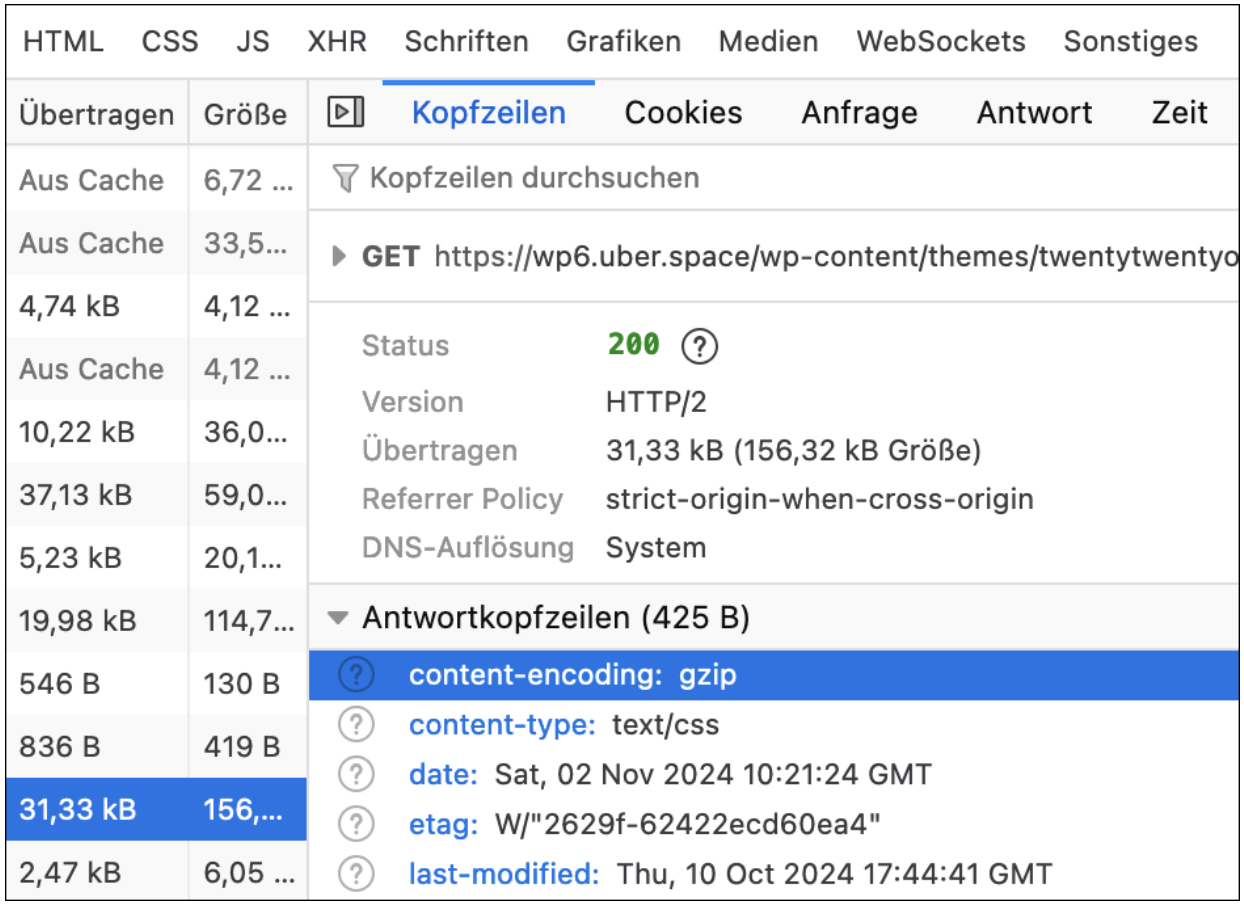


Abbildung 19.6 Der Antwort-Header »Accept-Encoding« gibt an, ob die Antwort komprimiert ist.

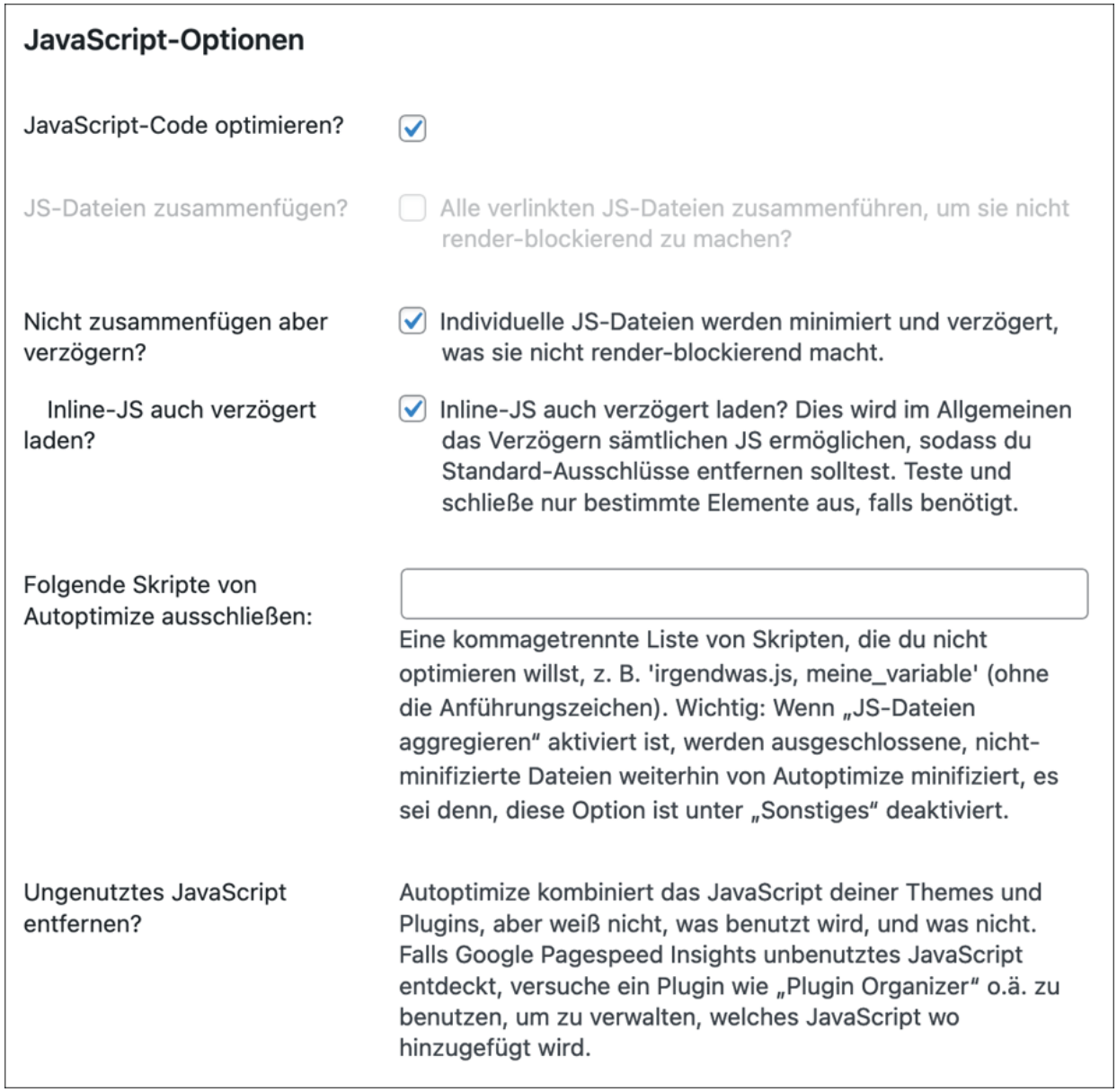


Abbildung 19.7 Die Optimierungsmöglichkeiten für JavaScript in Autoptimize

CSS-Optionen

CSS-Code optimieren?	☑
CSS-Dateien zusammenfügen?	☐ Alle verlinkten CSS-Dateien zusammenfügen? Falls diese Option nicht gewählt ist, werden individuelle CSS-Dateien an ihrem Ort verbleiben, aber minimiert werden.
Auch Inline-CSS zusammenfügen?	☐ Wähle diese Option, wenn Autoptimize CSS auch innerhalb des HTML hinzufügen soll.
Data: URIs für Bilder generieren (Inline Images)?	☐ Diese Option aktivieren, um kleine Hintergrundbilder direkt ins CSS einzufügen, und nicht als separate Downloads.
Render-blockierendes CSS entfernen?	☐ Den „above the fold" CSS-Code als Inline-CSS nach dem Laden der Seite einfügen. Bitte die FAQ lesen für weitere Informationen. Du kannst Regeln für verschiedene Seitentypen manuell erstellen oder dies auf dem Tab CriticalCSS vollautomatisch erledigen lassen.
Gesamten CSS-Code Inline einfügen?	☐ CSS komplett inline zu schreiben ist ein einfacher Weg, um zu verhindern, dass CSS den Bildaufbau verlangsamt, es kann aber generell nicht empfohlen werden, da es die HTML-Datei signifikant vergrößert. Zusätzlich könnte es Meta-Tags im Quelltext der Seite nach unten schieben, sodass Facebook und WhatsApp sie nicht mehr finden können, was die Vorschaubilder beim Teilen verschwinden lässt.
Folgende CSS-Dateien von Autoptimize ausschließen:	Eine durch Komma getrennte Liste von CSS-Dateien, die von der Optimierung ausgeschlossen werden soll. Wichtig: Ausgeschlossene nicht-minimierte Dateien werden noch immer von Autoptimize minimiert, es sei denn, diese Option wurde unter „Weitere Optionen" deaktiviert.
Nicht verwendetes CSS entfernen?	If Google Pagespeed Insights detects unused CSS, consider using the premium Rapidload service to **reduce your site's CSS size to up to 90%**, resulting in a slimmer, faster site!

Abbildung 19.8 Die Optimierungsmöglichkeiten für CSS in Autoptimize

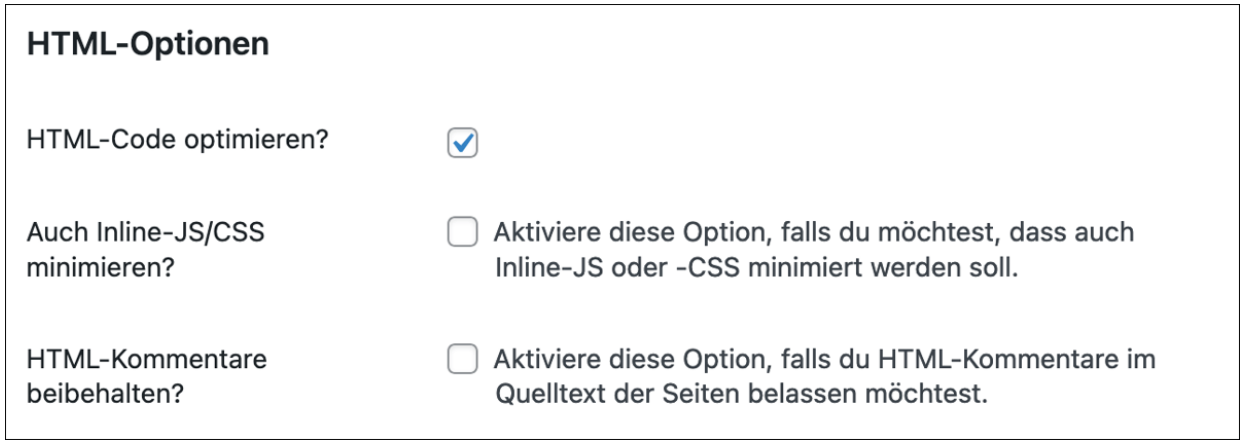


Abbildung 19.9 Auch das HTML kann minifiziert werden.

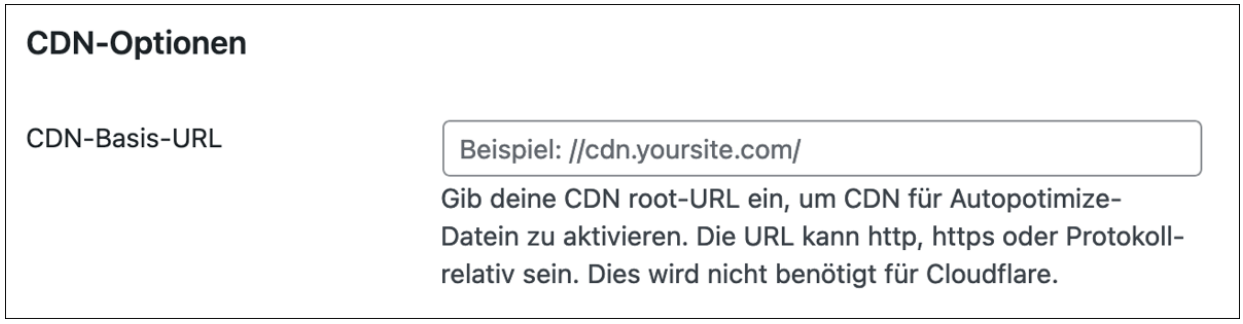


Abbildung 19.10 Wenn Sie ein CDN nutzen, können Sie auch die Autoptimize-Dateien darüber einbinden.

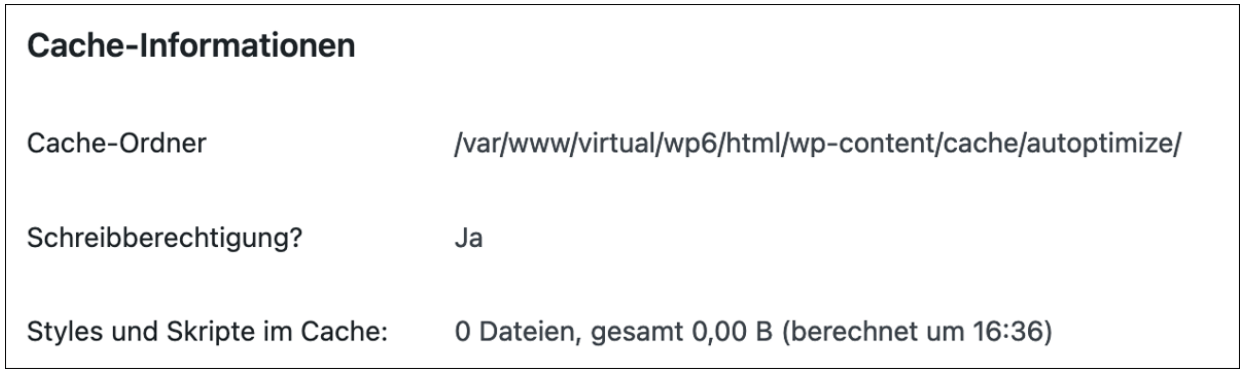


Abbildung 19.11 Die Cache-Informationen zeigen Ihnen ein paar interessante Informationen zum Autoptimize-Cache an.

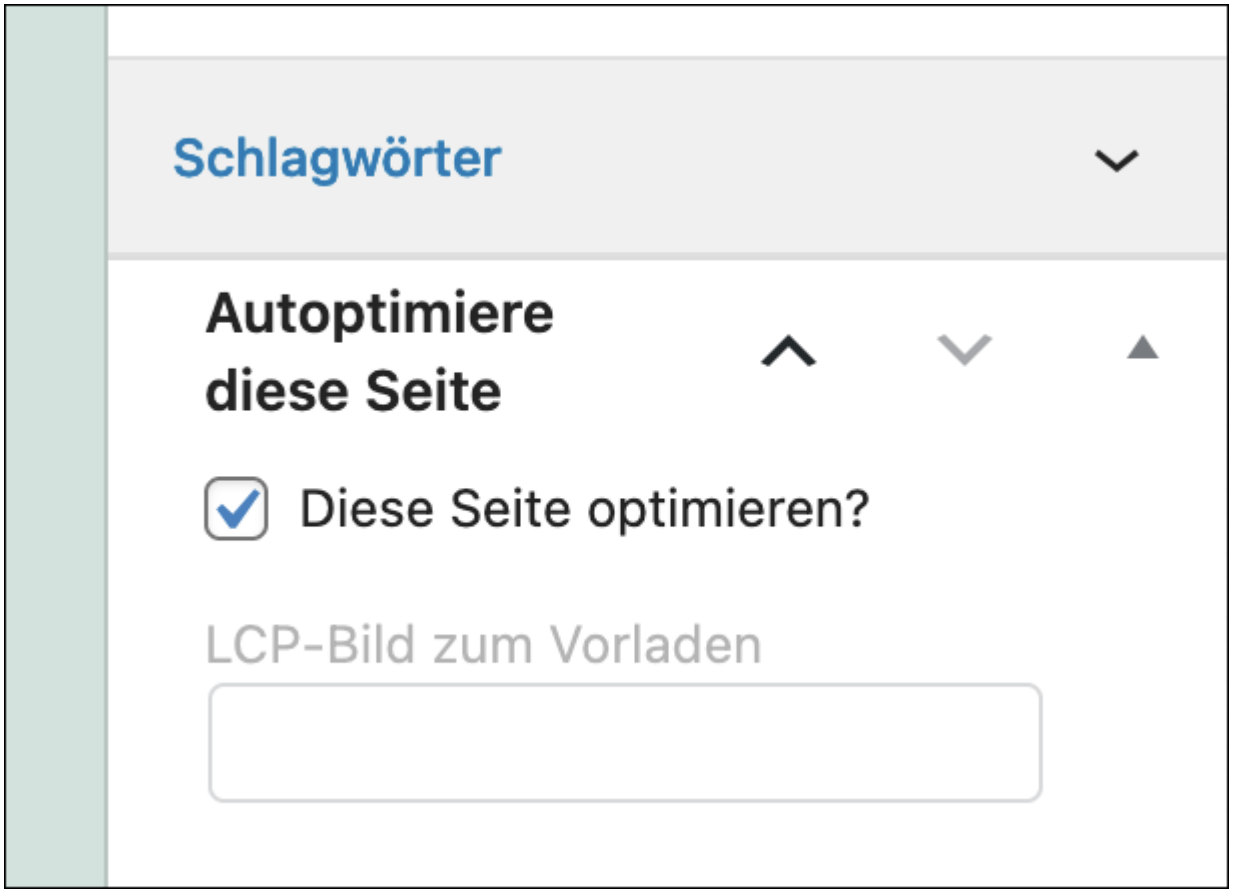


Abbildung 19.12 Sie können die Autoptimize-Einstellungen für einzelne Seiten oder Beiträge anpassen.



Abbildung 19.13 In der Box »Weitere Optionen« sind unterschiedlichste Optionen zusammengefasst, die in der Regel unangetastet bleiben können.

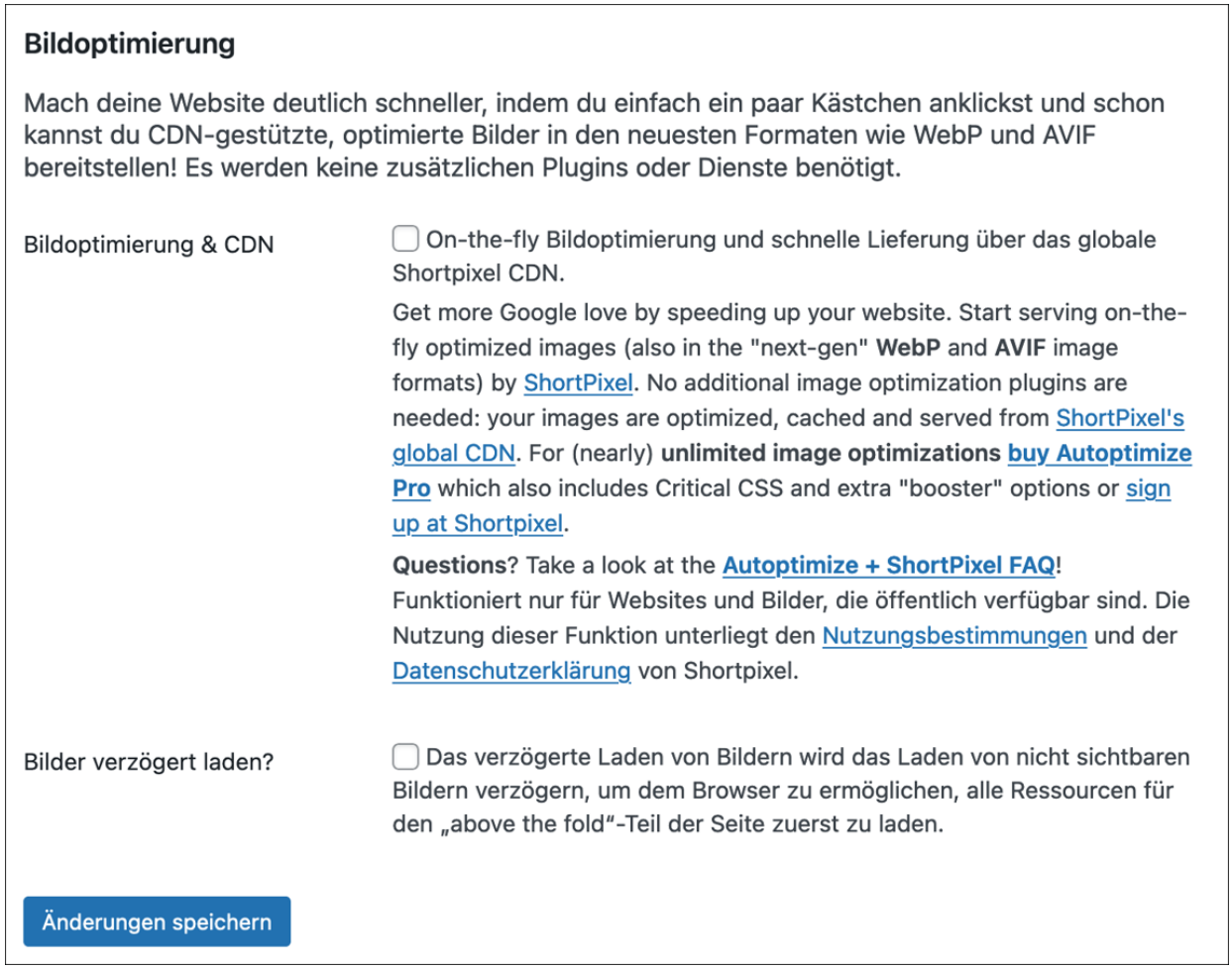


Abbildung 19.14 Den Tab mit den Bilder-Optionen können Sie wahrscheinlich einfach überspringen.

Regeln

Wie du die CriticalCSS-Regeln von Autoptimize nutzt
Klicke auf den seitlichen Pfeil, um Anweisungen ein-/auszublenden

Neue Regel hinzufügen | Bearbeite Standard-Regel CSS | CSS zu allen Regeln hinzufügen | Alle Regeln entfernen

Improve performance with critical CSS



Autoptimize's Critical CSS speeds up your site by improving Google PageSpeed Insights performance indicators like **"first contentful paint"** and "largest contentful paint". Having the **correct rules for each page** is key to avoid "cumulative layout shift".

Autoptimize Pro fully automates critical CSS rule creation, but also provides image optimization, CDN and extra powerful "booster" options all in one handy subscription to make your site even faster! See https://autoptimize.com/pro/ for more info and maybe a discount code?

Alternatively with criticalcss.com's 3EUR (4USD)/ month subscription you can generate the right critical CSS for any given page and create manual rules yourself. You can also **automate the entire process** by opting for the "WordPress" plan (3EUR or 4USD subscription + 8EUR or 10USD per domain); simply sign up, copy/ paste the API key in the field below and Autoptimize will automatically start creating rules!

If you have any questions or need support, head on over to our support forum and we'll help you get up and running in no time!

API-Schlüssel: Keine

Please enter a valid criticalcss.com API key to start.

Dein API-Schlüssel

Bitte gib hier deinen API-Schlüssel für criticalcss.com ein.

Gib deinen criticalcss.com API-Schlüssel oben ein. Der Schlüssel wird jedes Mal neu validiert, wenn ein neuer Auftrag an ihn gesendet wird.
Um deinen API-Schlüssel zu erhalten, gehe zu criticalcss.com >Konto > API-Schlüssel.
Anfragen zur Erstellung eines kritischen CSS über die API kosten £5 pro Domain und Monat.
Noch nicht sicher? Mit der 30 Tage Geld-zurück-Garantie hast du nichts zu verlieren!

Abbildung 19.15 Auch den Tab für kritisches CSS können Sie in der Regel links liegen lassen. Diese Optimierung ist nicht gerade einfach und kann sich auch negativ auf die Website-Performance auswirken.

Weitere Auto-Optimierungen

Die folgenden Einstellungen können die Leistung deiner Website noch weiter steigern.

Google Fonts

- (•) So beibehalten
- ( ) Google Fonts entfernen
- ( ) Im head kombinieren und verlinken (Fonts laden schnell, aber blockieren das Rendern), beinhaltet `display:swap` .
- ( ) Im head kombinieren und verlinken und verzögert laden (Fonts laden spät, aber blockieren nicht das Rendern), beinhaltet `display:swap` .

Emojis entfernen

[ ] Entfernt WordPress' internes Emoji-Inline-CSS, Inline JavaScript und eine weitere nicht optimierte JavaScript-Datei.

Abfragezeichenfolgen von statischen Ressourcen entfernen

[ ] Removing query strings (or more specifically the `ver` parameter) will not improve load time, but might improve performance scores.

WordPress-Block-CSS entfernen

[ ] WordPress fügt Block-CSS und globale Stile hinzu, um die einfache Gestaltung von blockbasierten Websites zu verbessern, die jedoch eine erhebliche Menge an CSS und SVG hinzufügen können. Wenn du sicher bist, dass deine Seite ohne Block-CSS und "globale Stile" auskommt, kannst du sie hier deaktivieren.

Zu Dritt-Domains vorverbinden (fortgeschrittene Benutzer)

Add 3rd party domains you want the browser to preconnect to, separated by comma's. Make sure to include the correct protocol (HTTP or HTTPS).

Spezifische Anfragen im Voraus laden (für Fortgeschrittene)

Komma-separierte Liste mit vollständigen URLs von im Voraus zu ladenden Ressourcen. Sparsam einsetzen!

Async Javascript-files (advanced users)

Comma-separated list of local or 3rd party JS-files that should loaded with the `async` flag. JS-files from your own site will be automatically excluded if added here. Die Konfiguration von asynchronem JavaScript ist einfacher und flexibler mit dem Plugin "Async Javascript". Hier klicken, um es zu installieren und zu aktivieren.

Optimiere YouTube-Videos

WP YouTube Lyte ermöglicht dir, deine Videos durch das Einfügen von responsiven „Lite YouTube Embeds“ verzögert zu laden. Hier klicken, um es zu installieren und zu aktivieren.

Änderungen speichern

Abbildung 19.16 Im Tab »Extras« finden Sie unterschiedlichste Einstellungsmöglichkeiten, die die Performance verbessern können.

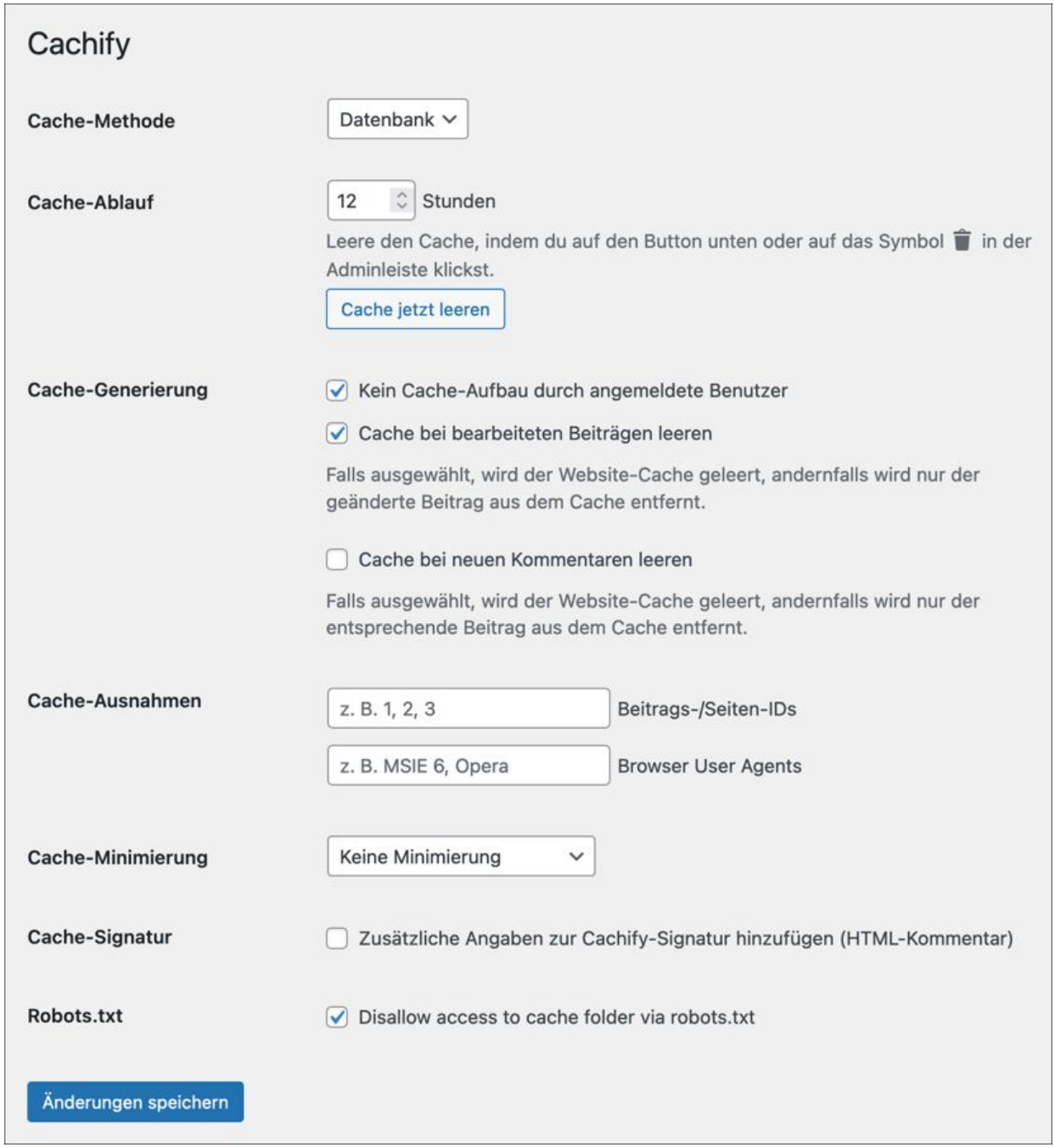


Abbildung 19.17 Die Cachify-Einstellungen sind verglichen mit anderen Caching-Plugins sehr übersichtlich.



Abbildung 19.18 Wenn die Cache-Methode auf »Festplatte« gestellt ist, müssen Sie die Webserver-Einstellung anpassen, um alle Vorteile davon nutzen zu können.

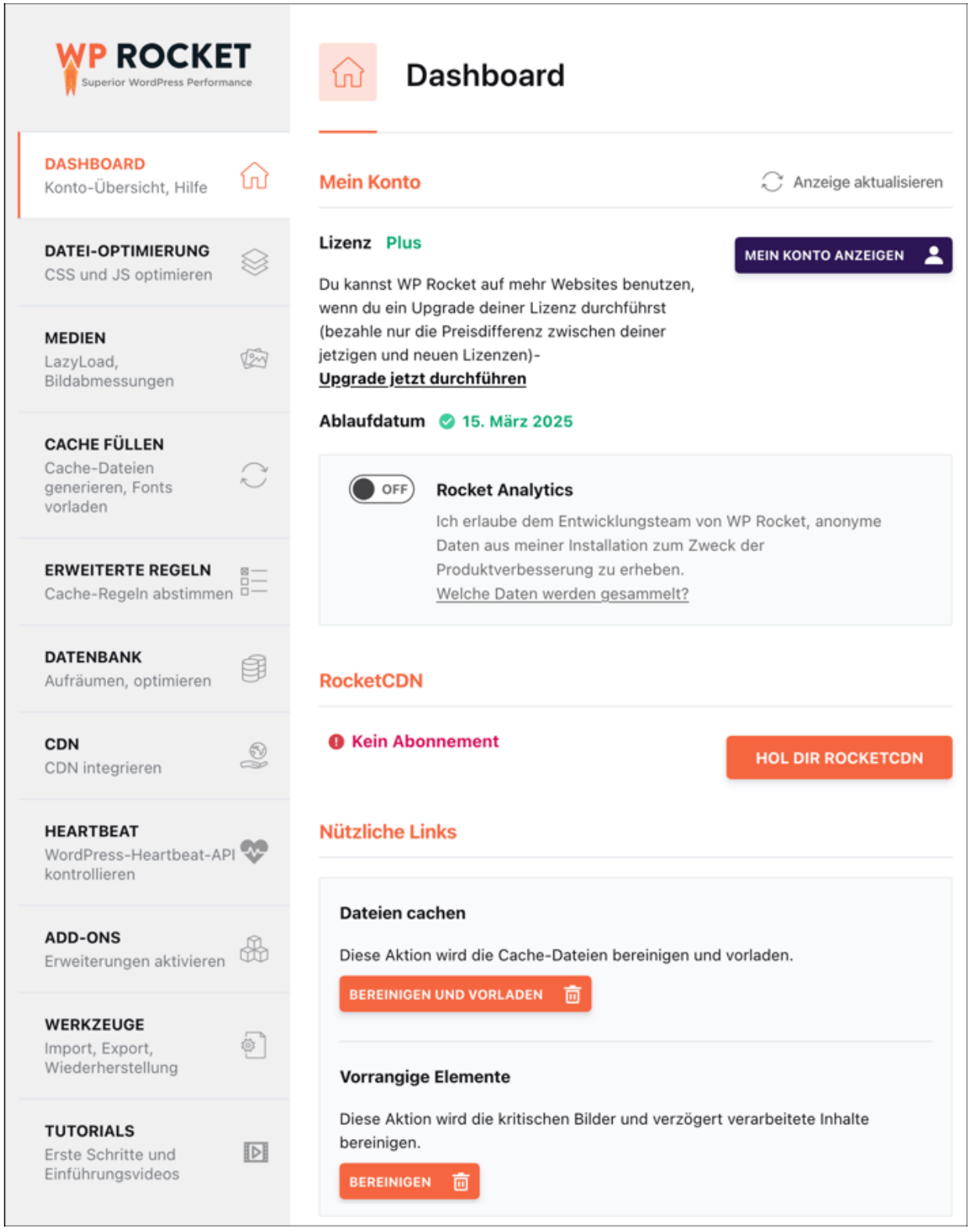

WP ROCKET
Superior WordPress Performance
DASHBOARD
Konto-Übersicht, Hilfe
DATEI-OPTIMIERUNG
CSS und JS optimieren
MEDIEN
LazyLoad, Bildabmessungen
CACHE FÜLLEN
Cache-Dateien generieren, Fonts vorladen
ERWEITERTE REGELN
Cache-Regeln abstimmen
DATENBANK
Aufräumen, optimieren
CDN
CDN integrieren
HEARTBEAT
WordPress-Heartbeat-API kontrollieren
ADD-ONS
Erweiterungen aktivieren
WERKZEUGE
Import, Export, Wiederherstellung
TUTORIALS
Erste Schritte und Einführungsvideos
Dashboard
Mein Konto
Anzeige aktualisieren
Lizenz Plus
MEIN KONTO ANZEIGEN
Du kannst WP Rocket auf mehr Websites benutzen, wenn du ein Upgrade deiner Lizenz durchführst (bezahle nur die Preisdifferenz zwischen deiner jetzigen und neuen Lizenzen)-
Upgrade jetzt durchführen
Ablaufdatum 15. März 2025
OFF
Rocket Analytics
Ich erlaube dem Entwicklungsteam von WP Rocket, anonyme Daten aus meiner Installation zum Zweck der Produktverbesserung zu erheben.
Welche Daten werden gesammelt?
RocketCDN
Kein Abonnement
HOL DIR ROCKETCDN
Nützliche Links
Dateien cachen
Diese Aktion wird die Cache-Dateien bereinigen und vorladen.
BEREINIGEN UND VORLADEN
Vorrangige Elemente
Diese Aktion wird die kritischen Bilder und verzögert verarbeitete Inhalte bereinigen.
BEREINIGEN

Abbildung 19.19 Das Dashboard zeigt Informationen zu Ihrer WP-Rocket-Lizenz und bietet rechts Buttons zum Löschen vom Cache.

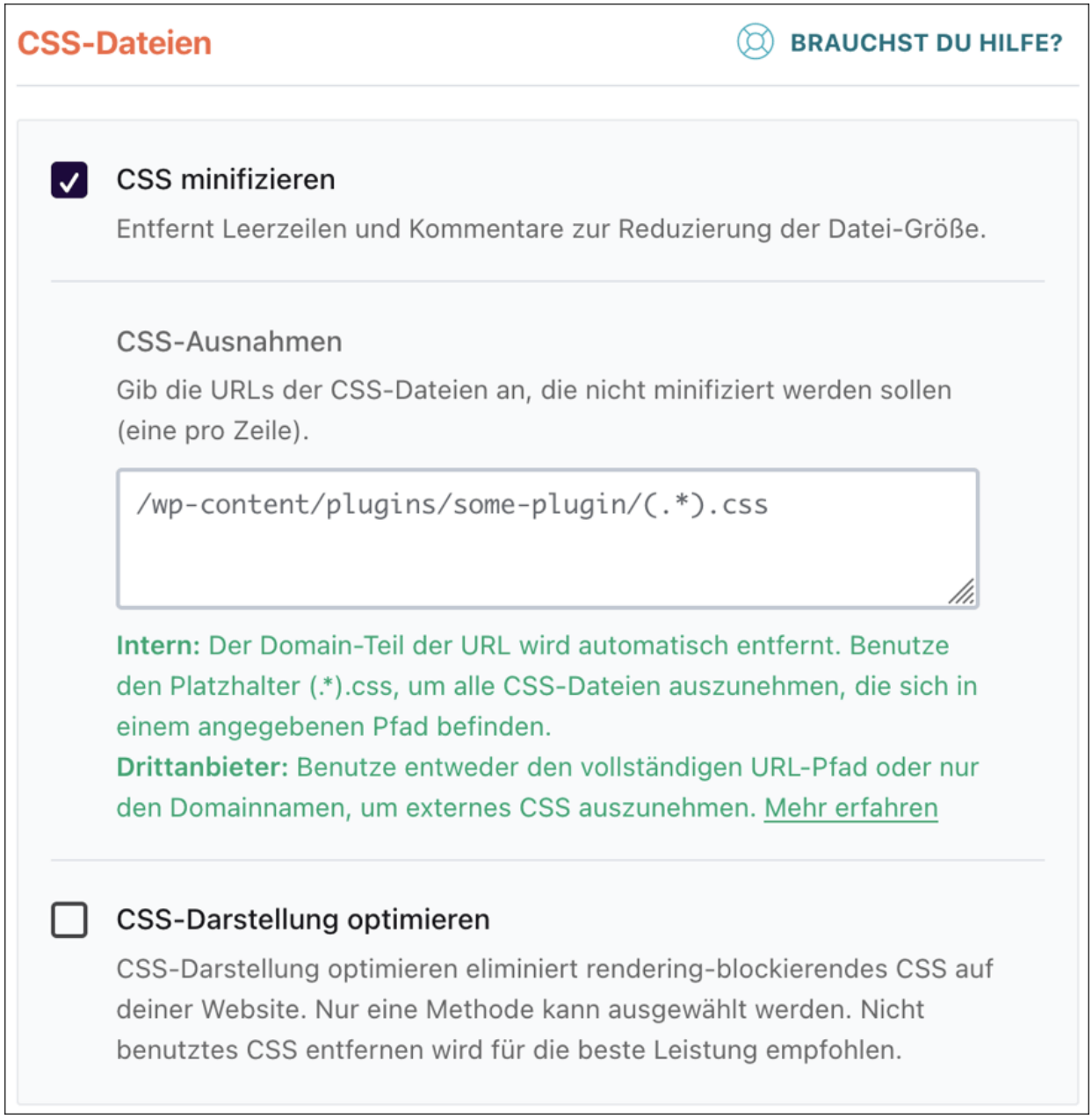


Abbildung 19.20 Die Optionen von WP Rocket für CSS-Dateien

JavaScript-Dateien

BRAUCHST DU HILFE?

- [x] **JavaScript minifizieren**

  Entfernt Leerzeilen und Kommentare zur Reduzierung der Datei-Größe.

- [ ] **JavaScript zusammenfassen *(„JavaScript minifizieren" erforderlich zur Aktivierung)***

  JavaScript zusammenfassen kombiniert internes, externes und inline JS, um HTTP-Anfragen zu verringern. nicht empfohlen, falls deine Website mit HTTP/2 läuft. Mehr erfahren

  JavaScript-Ausnahmen

  URLs von JavaScript-Dateien, die von der Minifizierung und Zusammenführung ausgenommen werden sollen (eine pro Zeile).

  ```
  /wp-content/themes/some-theme/(.*).js
  ```

  Intern: Der Domain-Teil der URL wird automatisch entfernt. Benutze den Platzhalter (.*).css, um alle JS-Dateien auszunehmen, die sich in einem angegebenen Pfad befinden.
  Drittanbieter: Benutze entweder den vollständigen URL-Pfad oder nur den Domainnamen, um externes JS auszunehmen. Mehr erfahren

- [ ] **JavaScript verzögert laden**

  Entfernt rendering-blockierendes JavaScript, kann die Ladezeit verbessern. Mehr erfahren

- [ ] **JavaScript-Ausführung verzögern**

  Verbessert die Leistung durch das Verzögern des Ladens von JavaScript-Dateien bis zur ersten Benutzerinteraktion (z. B: Scrollen, Klicken). Mehr erfahren

Abbildung 19.21 Die Möglichkeiten zur Optimierung von JavaScript durch WP Rocket

LazyLoad

BRAUCHST DU HILFE?

Es kann die tatsächliche und empfundene Ladezeit verbessern, weil Bilder, iframes und Videos erst geladen werden, wenn sie in den Viewport kommen (oder in die Nähe). Das verringert auch die Anzahl der HTTP-Anfragen. Mehr erfahren

- [ ] Für Bilder aktivieren
- [ ] Für CSS-Hintergrundbilder aktivieren
- [ ] Für iframes und Video aktivieren

Bilder- oder iframes-Ausnahmen

Gib Schlüsselwörter (z. B. Bild-Dateiname, CSS-Dateiname, CSS-Klasse, Domain) des auszuschließenden Bildes oder iframe-Code an (eins pro Zeile). Weitere Informationen

```
example-image.jpg
slider-image
background-image-style.css
```

Bildabmessungen

BRAUCHST DU HILFE?

Fügt Bildern fehlende width und height Attribute hinzu. Hilft, Layout-Sprünge zu verhindern und verbessert die Leseerfahrung für deine Besucher. Mehr erfahren

- [ ] Fehlende Bildabmessungen hinzufügen

Abbildung 19.22 Die Optionen für Medien in WP Rocket beinhalten Lazy Loading und Nachrüsten von »width« und »height« für Bilder.

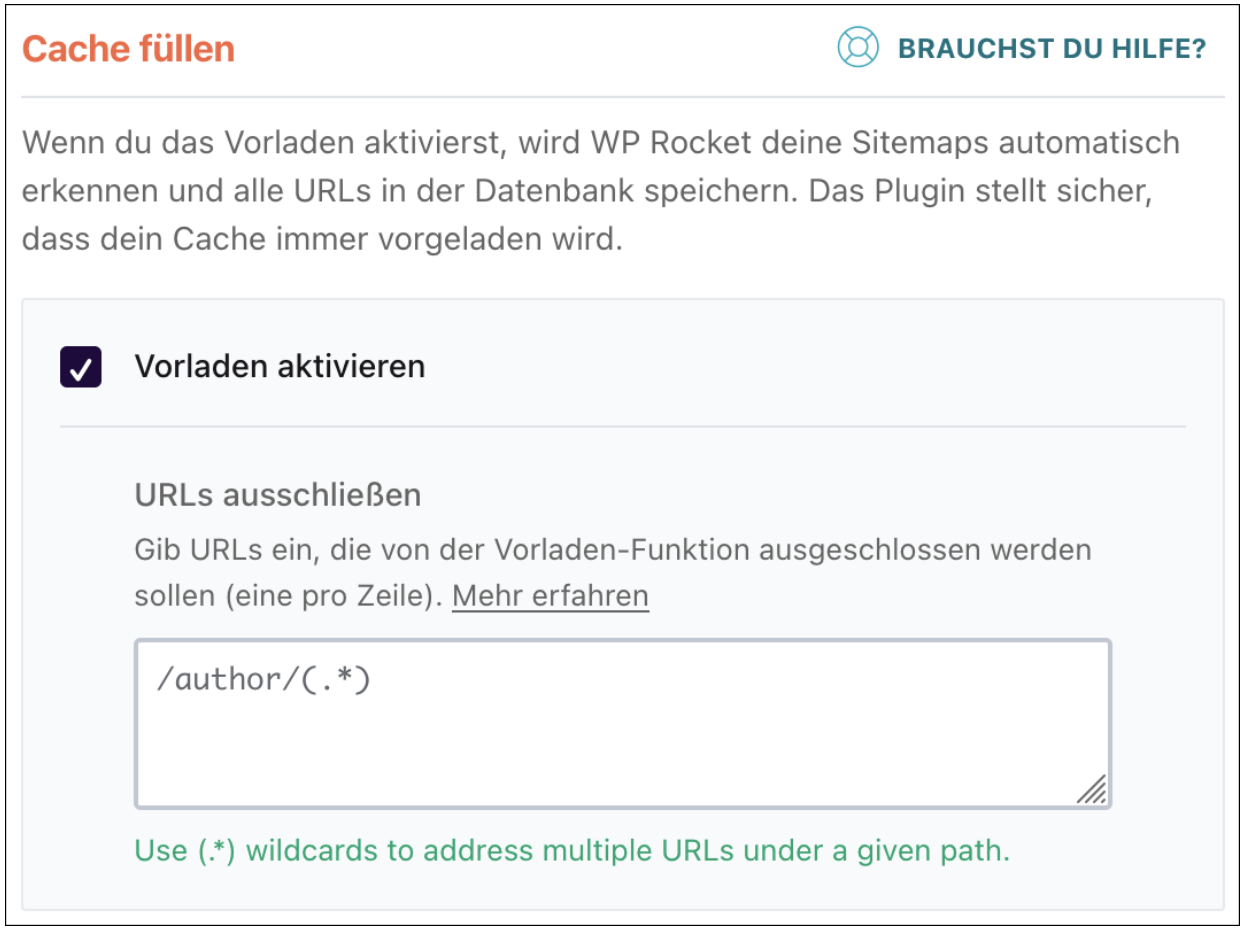


Abbildung 19.23 WP Rocket ermöglicht das Vorladen des Caches, sodass Besucher*innen auch nach dem Leeren des Caches eine schon gecachte Version aufrufen.

Links vorladen

BRAUCHST DU HILFE?

Links vorladen optimiert die wahrgenommene Ladezeit, indem eine verlinkte Seite bereits geladen wird, sobald sich der Mauszeiger eines Benutzers über einem Link befindet. Mehr erfahren

☐ Vorladen von Links aktivieren

DNS-Prefetch

BRAUCHST DU HILFE?

Kann das Laden externer Dateien beschleunigen, besonders über eine mobilen Netzwerkverbindung.

Vorzeitig aufzulösende URLs

Vorzeitig aufzulösende externe Domains (kein `http:` , eine pro Zeile)

```
//example.com
```

Fonts vorladen

BRAUCHST DU HILFE?

Optimiert die Leistung, indem es Browsern hilft, Fonts in CSS-Dateien zu entdecken. Mehr erfahren

Vorzuladende Fonts

URLs von Font-Dateien, die vorgeladen werden sollen (eine pro Zeile). Fonts müssen auf deiner eigenen Domain gehostet sein, oder auf der Domain, die du im CDN-Tab angegeben hast.

```
/wp-content/themes/your-theme/assets/fonts/font-file.woff
```

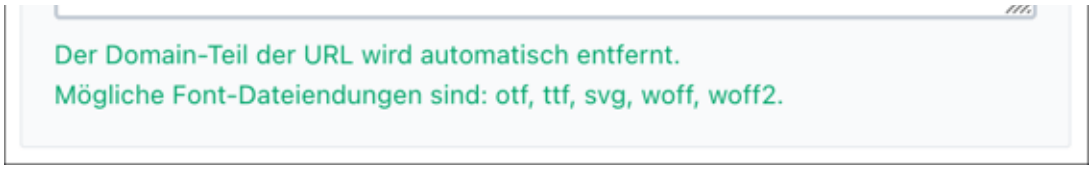


Abbildung 19.24 Die weiteren Optionen im Tab »Cache füllen« sind beispielsweise das Vorladen von Inhalten anderer Seiten, wenn sich der Cursor über einem Link zu dem Inhalt befindet.

Cache-Dauer

BRAUCHST DU HILFE?

Cache-Dateien, die älter als die angegebene Lebensdauer sind, werden gelöscht.
Aktiviere Cache füllen, um den Cache nach dem Ablauf der Lebensdauer automatisch neu zu generieren.

Zeitangabe, nach welcher der globale Cache geleert wird (0 = unbegrenzt)

Reduziere auf 10 Stunden oder weniger, falls du Probleme bemerkst, die periodisch aufzutreten scheinen. Warum?

10 Stunden

Nie cachen (URLs)

BRAUCHST DU HILFE?

Sensible Seiten, wie zum Beispiel eigene Login-/Logout-Seiten, sollten vom Cache ausgeschlossen werden.

URLs von Seiten oder Beiträgen, die niemals gecacht werden sollen (eine pro Zeile)

```
/example/(.*)
```

Der Domain-Teil der URL wird automatisch entfernt.
Nutze den Platzhalter (.*), um alle URLs eines bestimmten Pfades anzusprechen.

Nie cachen (Cookies)

BRAUCHST DU HILFE?

Gib vollständige oder Teile von Cookie-IDs an, die das Caching einer Seite verhindern sollen, sobald sie im Browser des Besuchers gesetzt sind (eine pro Zeile)

```
wpscan
```

Nie cachen (User Agents)

BRAUCHST DU HILFE?



Abbildung 19.25 In den »Erweiterten Regeln« können Sie Sonderregeln fürs Caching einstellen.

Beiträge/Seiten bereinigen

BRAUCHST DU HILFE?

Revisionen, Entwürfe und Papierkorb werden endgültig gelöscht. Nicht aktivieren, wenn du Revisionen und Entwürfe noch benötigst!

- [ ] **Revisionen**
  1.258 Revisionen in deiner Datenbank.
- [ ] **Automatisch erstellte Entwürfe**
  1 Entwurf in deiner Datenbank.
- [ ] **Gelöschte Einträge (Papierkorb)**
  0 gelöschte Beiträge in deiner Datenbank.

Kommentare bereinigen

Spam und Kommentare im Papierkorb werden endgültig gelöscht.

- [ ] **Spam-Kommentare**
  43.402 Spam-Kommentare in deiner Datenbank.
- [ ] **Gelöschte Kommentare (Papierkorb)**
  0 gelöschte Kommentare in deiner Datenbank.

Transients bereinigen

Transients sind zeitlich begrenzte Optionen; sie können problemlos gelöscht werden. Sobald deine Plugins sie benötigen, werden sie automatisch erneut generiert.



Abbildung 19.26 WP Rocket bietet unterschiedliche Aktionen an, um die Größe der Datenbank zu verringern. Hier sollten Sie aber nicht einfach alles anhaken und durchführen, da Dinge wie Revisionen und Spam-Kommentare hilfreich sein können.

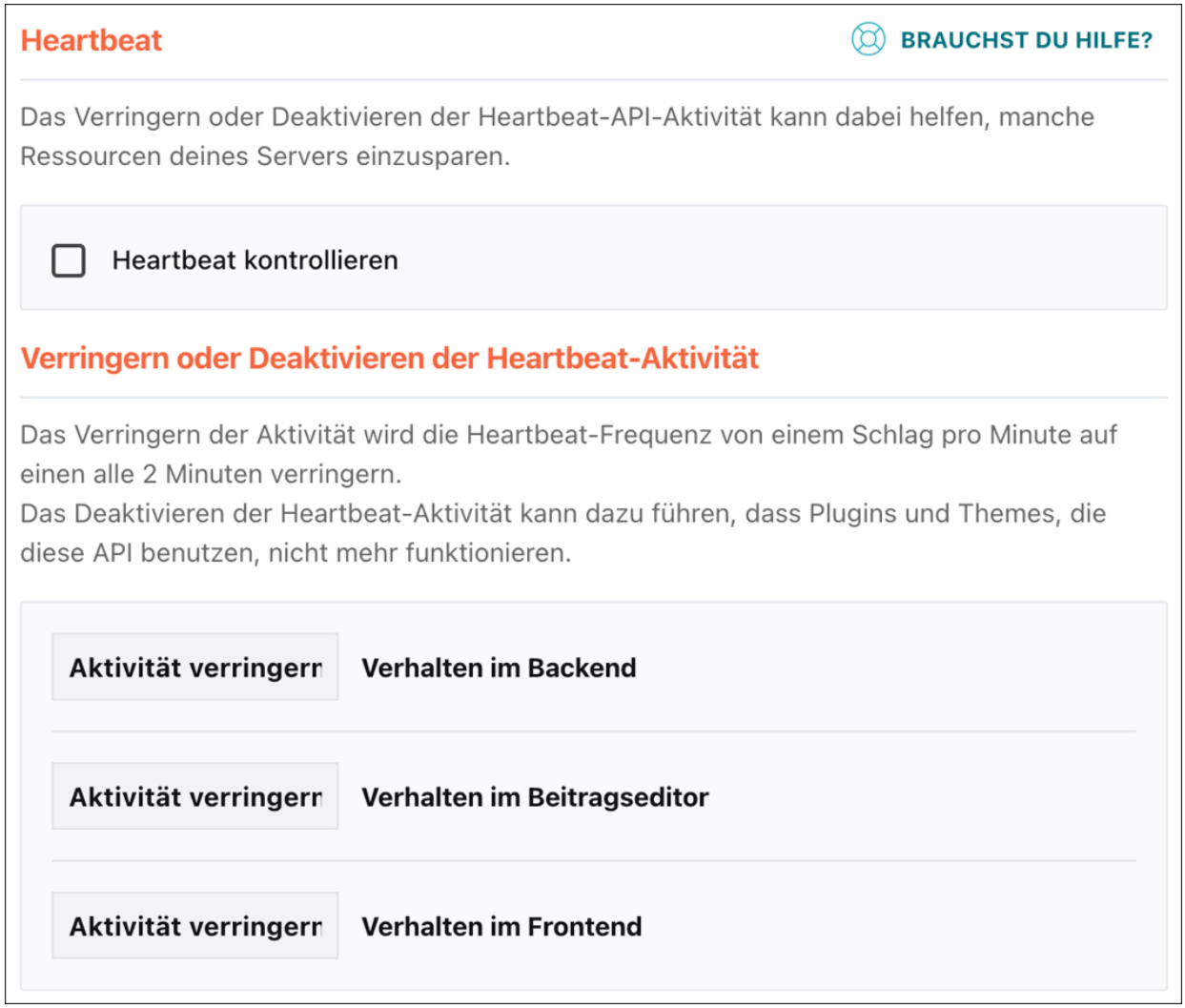


Abbildung 19.27 WP Rocket kann das Intervall der Heartbeat-API verringern, um die Serverlast zu reduzieren.

One-click Rocket Add-ons

One-Click Add-ons erweitern den Funktionsumfang ohne weitere Konfiguration. Aktiviere sie einfach, indem du den Schalter auf „An“ umlegst.

Personalisierter Cache **Add-on Status** OFF



Wenn du für jeden angemeldeten WordPress-Benutzer einen eigenen Satz von Cache-Dateien erstellen möchtest, musst du dieses Add-on aktivieren.

Personalisierter Cache ist nützlich, wenn du spezielle oder eingeschränkte Inhalte nur für bestimmte Benutzer anbietest.

Mehr erfahren

Varnish **Add-on Status** OFF



Du musst diese Einstellung aktivieren, wenn Varnish auf deinem Server aktiv ist.

Der Varnish-Cache wird automatisch jedes Mal geleert, wenn der Cache von WP Rocket geleert wird, damit er immer aktuell ist.

Mehr erfahren

WebP-Kompatibilität **Add-on Status** OFF

webp

Browserkompatibilität für WebP-Bilder verbessern

Wir haben kein kompatibles WebP-Plugin entdeckt!
Falls du nicht bereits WebP-Bilder auf deiner Website hast, ziehe den Einsatz von Imagify oder eines anderen unterstützten Plugins in Betracht. Mehr erfahren
Falls du kein WebP einsetzt, aktiviere diese Option nicht.

Rocket Add-ons

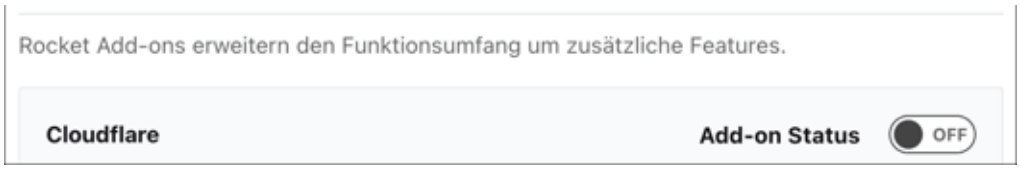


Abbildung 19.28 Im Tab »Add-ons« finden Sie ein paar Add-ons, die Sie ohne zusätzliche Konfiguration direkt aktivieren können, und ein paar, die konfiguriert werden müssen.

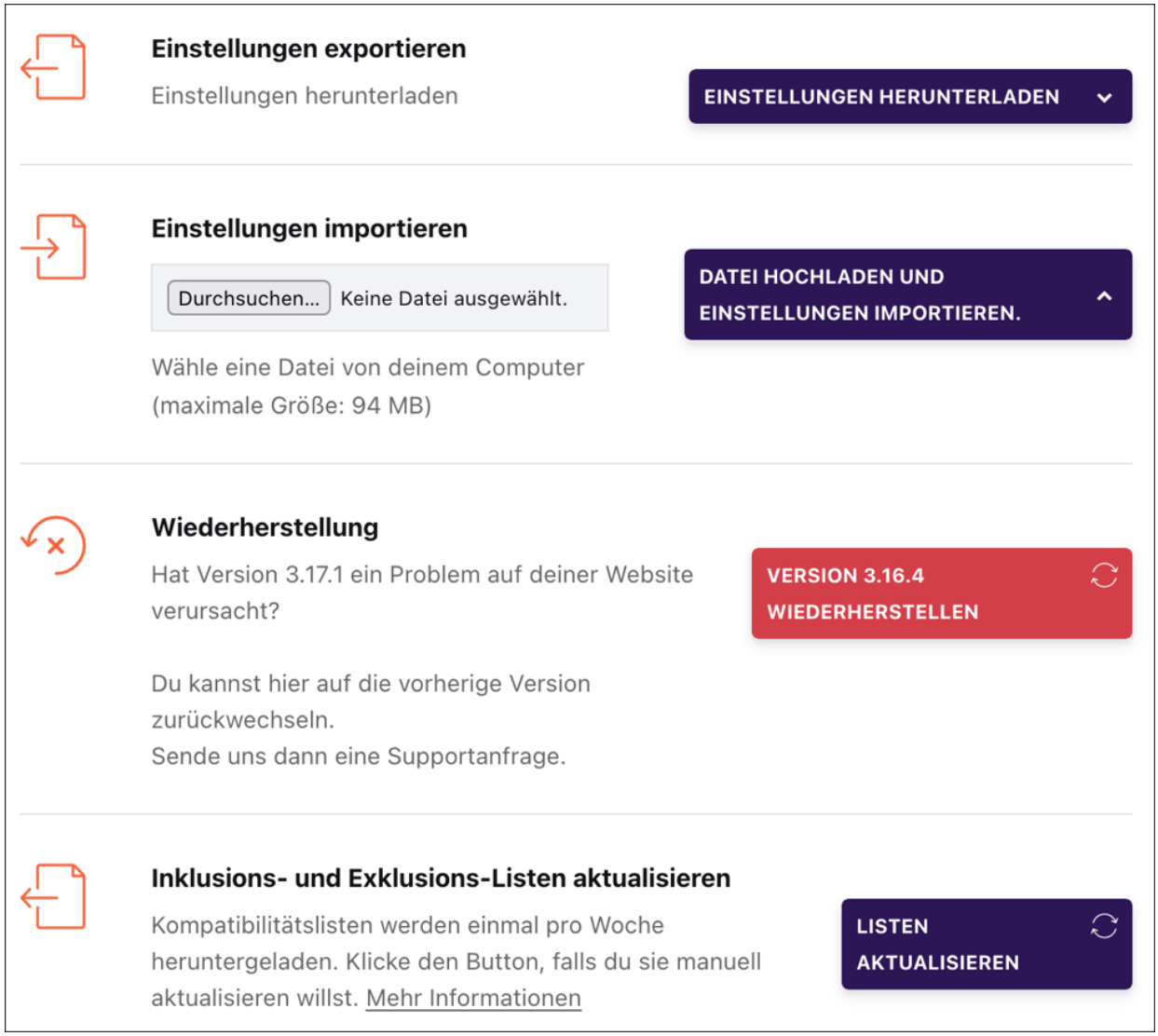


Abbildung 19.29 Sie können Ihre Einstellungen von WP Rocket exportieren, um sie auf einer anderen Instanz wieder zu importieren.

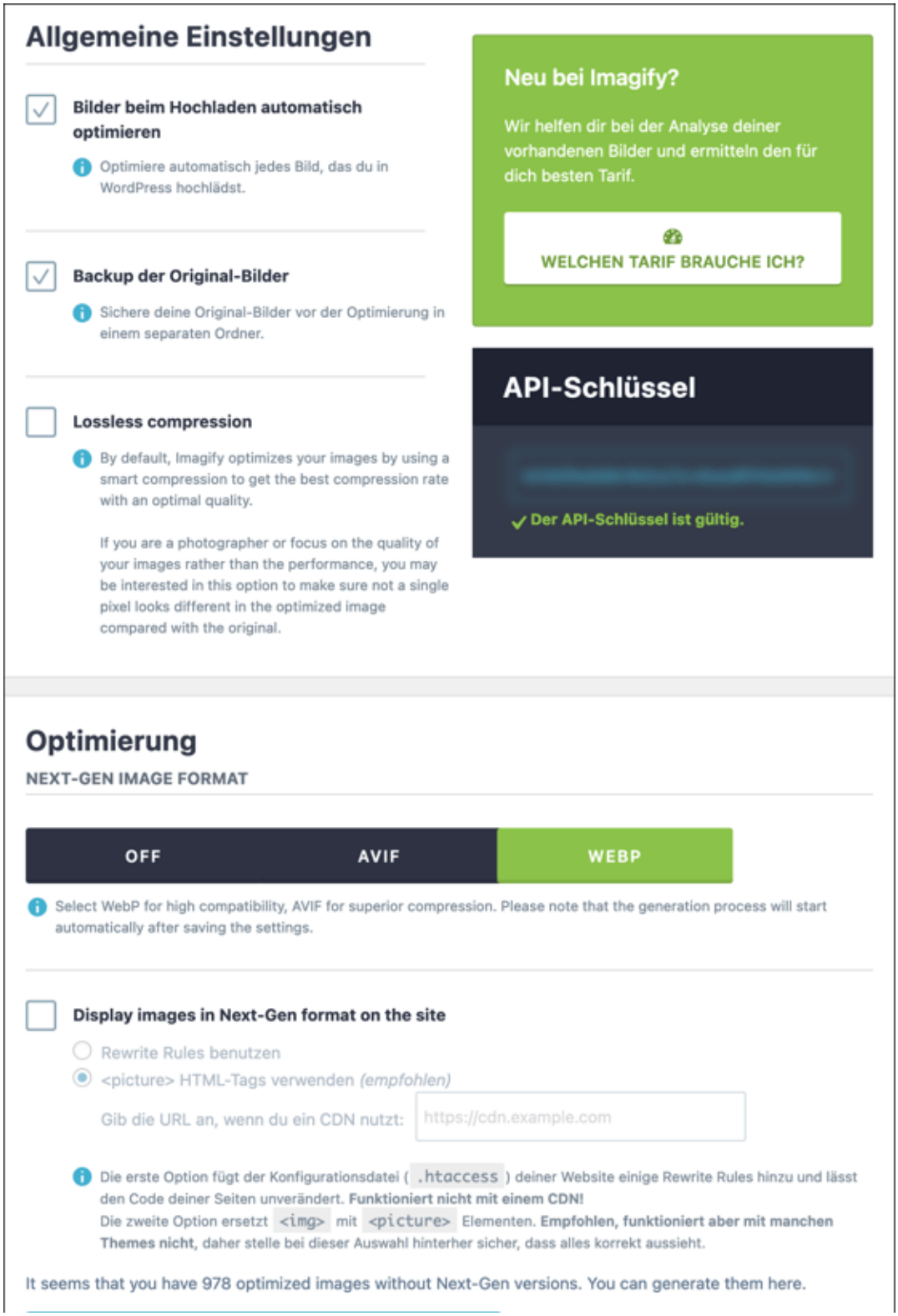
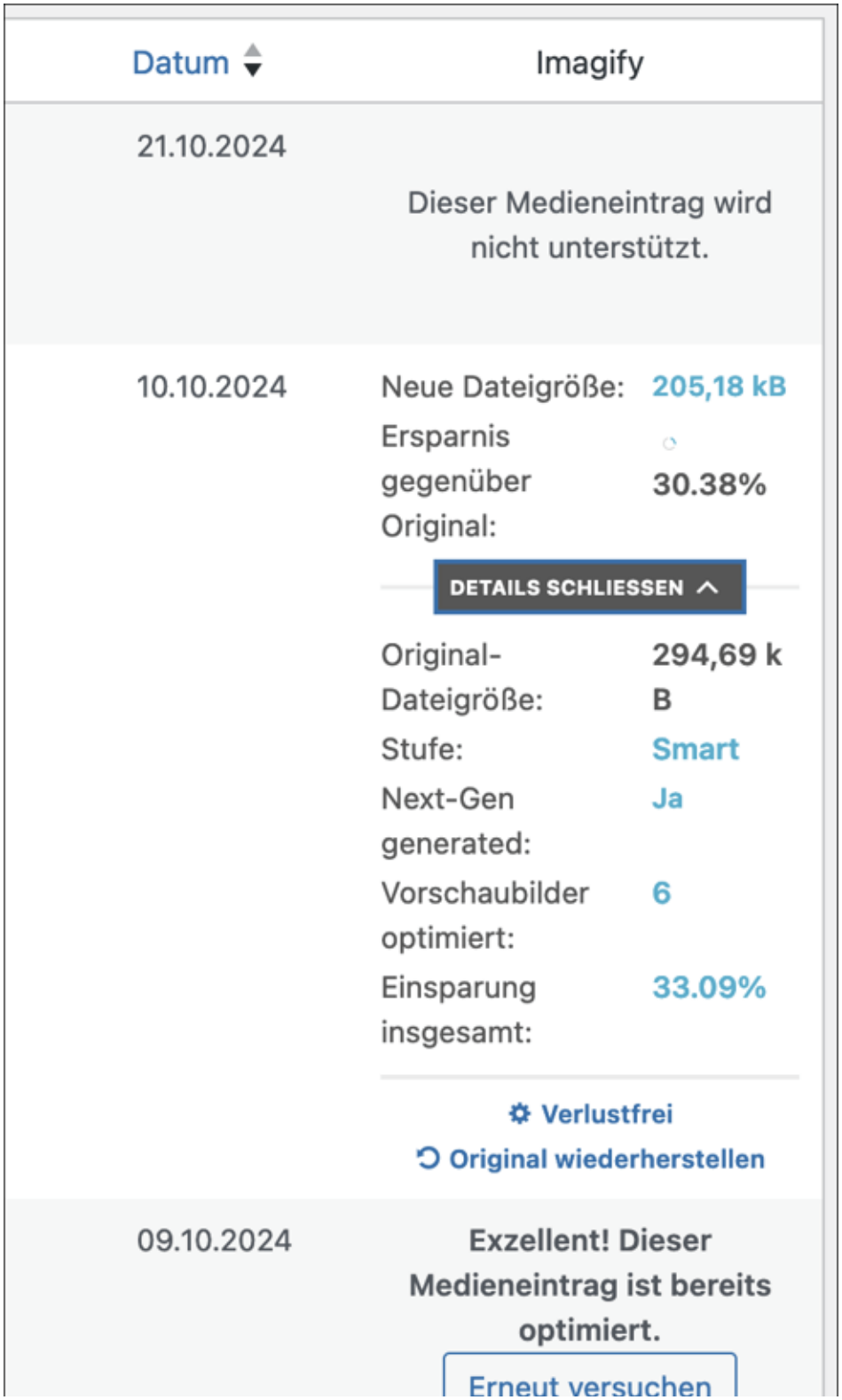

Allgemeine Einstellungen
Bilder beim Hochladen automatisch optimieren
Optimiere automatisch jedes Bild, das du in WordPress hochlädst.
Backup der Original-Bilder
Sichere deine Original-Bilder vor der Optimierung in einem separaten Ordner.
Lossless compression
By default, Imagify optimizes your images by using a smart compression to get the best compression rate with an optimal quality.
If you are a photographer or focus on the quality of your images rather than the performance, you may be interested in this option to make sure not a single pixel looks different in the optimized image compared with the original.
Neu bei Imagify?
Wir helfen dir bei der Analyse deiner vorhandenen Bilder und ermitteln den für dich besten Tarif.
WELCHEN TARIF BRAUCHE ICH?
API-Schlüssel
Der API-Schlüssel ist gültig.
Optimierung
NEXT-GEN IMAGE FORMAT
OFF
AVIF
WEBP
Select WebP for high compatibility, AVIF for superior compression. Please note that the generation process will start automatically after saving the settings.
Display images in Next-Gen format on the site
Rewrite Rules benutzen
<picture> HTML-Tags verwenden (empfohlen)
Gib die URL an, wenn du ein CDN nutzt:
https://cdn.example.com
Die erste Option fügt der Konfigurationsdatei (.htaccess) deiner Website einige Rewrite Rules hinzu und lässt den Code deiner Seiten unverändert. Funktioniert nicht mit einem CDN!
Die zweite Option ersetzt <img> mit <picture> Elementen. Empfohlen, funktioniert aber mit manchen Themes nicht, daher stelle bei dieser Auswahl hinterher sicher, dass alles korrekt aussieht.
It seems that you have 978 optimized images without Next-Gen versions. You can generate them here.



Abbildung 19.30 Die Einstellungen von Imagify finden sich alle auf einer Seite wieder.


Datum
Imagify
21.10.2024
Dieser Medieneintrag wird nicht unterstützt.
10.10.2024
Neue Dateigröße: 205,18 kB
Ersparnis gegenüber Original: 30.38%
DETAILS SCHLIESSEN
Original-Dateigröße: 294,69 kB
Stufe: Smart
Next-Gen generated: Ja
Vorschaubilder optimiert: 6
Einsparung insgesamt: 33.09%
Verlustfrei
Original wiederherstellen
09.10.2024
Exzellent! Dieser Medieneintrag ist bereits optimiert.
Erneut versuchen

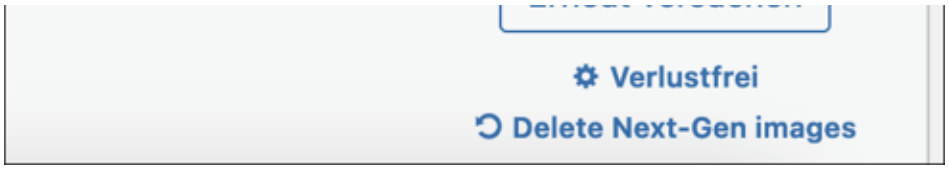


Abbildung 19.31 In der Listenansicht der Mediathek gibt es eine Spalte »Imagify«, in der Sie die Originalversion eines Bilds wiederherstellen oder nicht optimierte Bilder optimieren können.

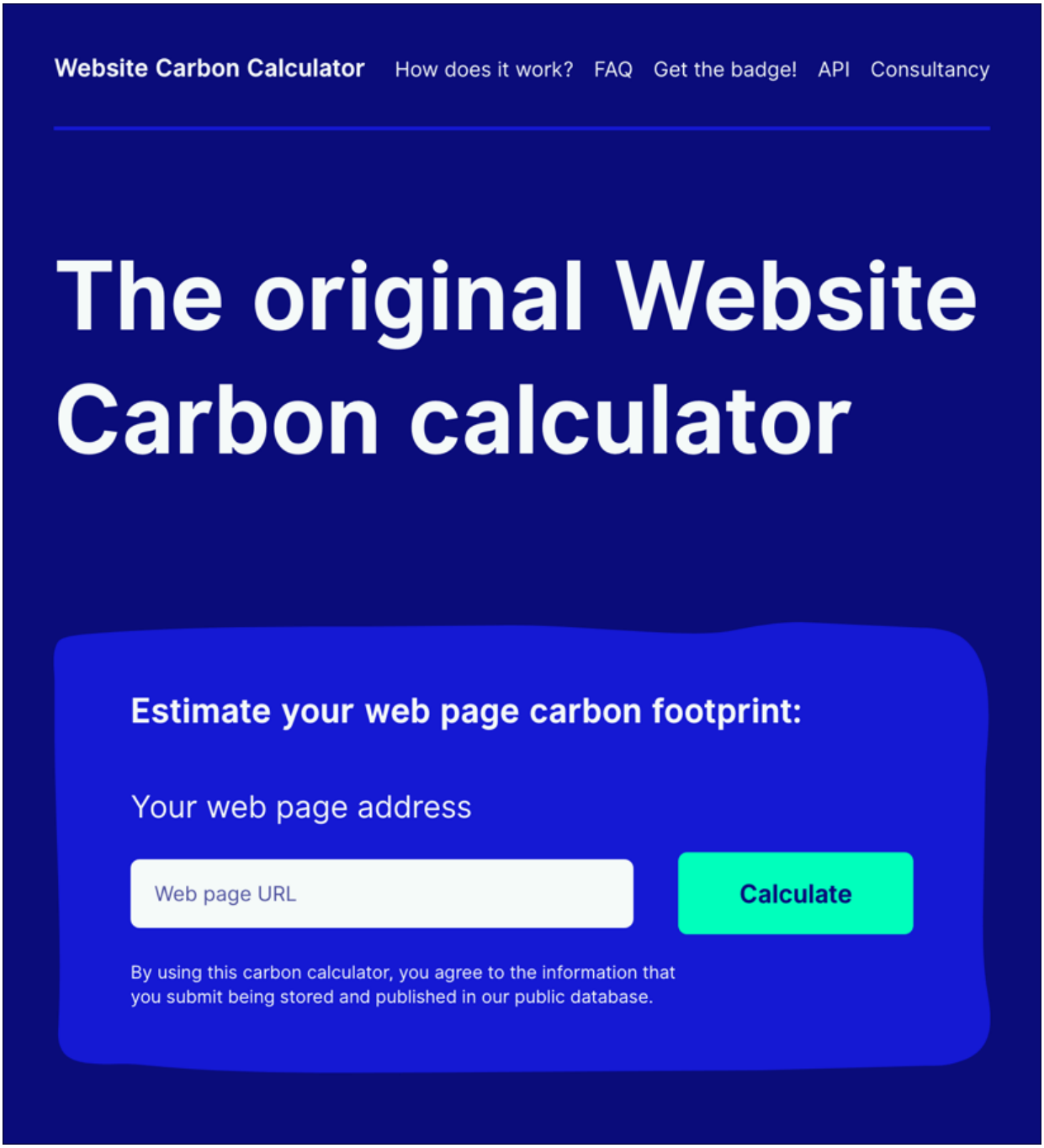


Abbildung 19.32 Mit dem Website Carbon Calculator können Sie den CO_2-Ausstoß Ihrer Website schätzen.

Website carbon results for: presswerk.net

A+

Hurrah! This web page achieves a carbon rating of A+

This is cleaner than 95% of all web pages globally



Learn about our rating system

This page was last tested on 3 Nov, 2024.

Copy URL



Only 0.05g of CO2 is produced every time someone visits this web page.

How do we calculate this?



This web page appears to be running on sustainable energy

How do we find this out?

Abbildung 19.33 Nach erfolgtem Test erhalten Sie drei wichtige Werte: eine Art Energie-Label, einen CO_2-Ausstoß pro Seitenaufruf und Daten zur Nachhaltigkeit des verwendeten Stroms Ihres Hosting-Anbieters.

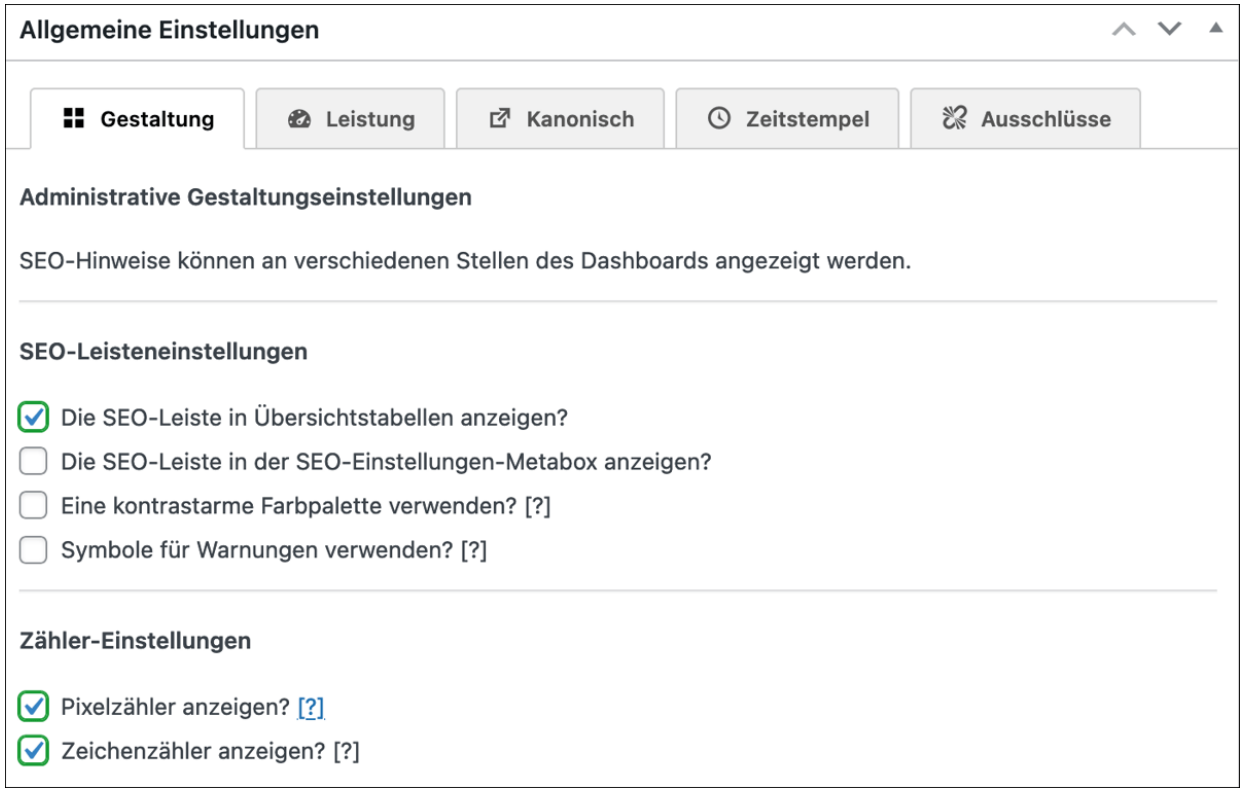


Abbildung 20.1 Im Tab »Gestaltung« der Box für die »Allgemeinen Einstellungen« können Sie unter anderem festlegen, ob in der Übersicht von Seiten und Beiträgen eine Spalte mit Hinweisen zum SEO-Stand für die Inhalte angezeigt werden soll.

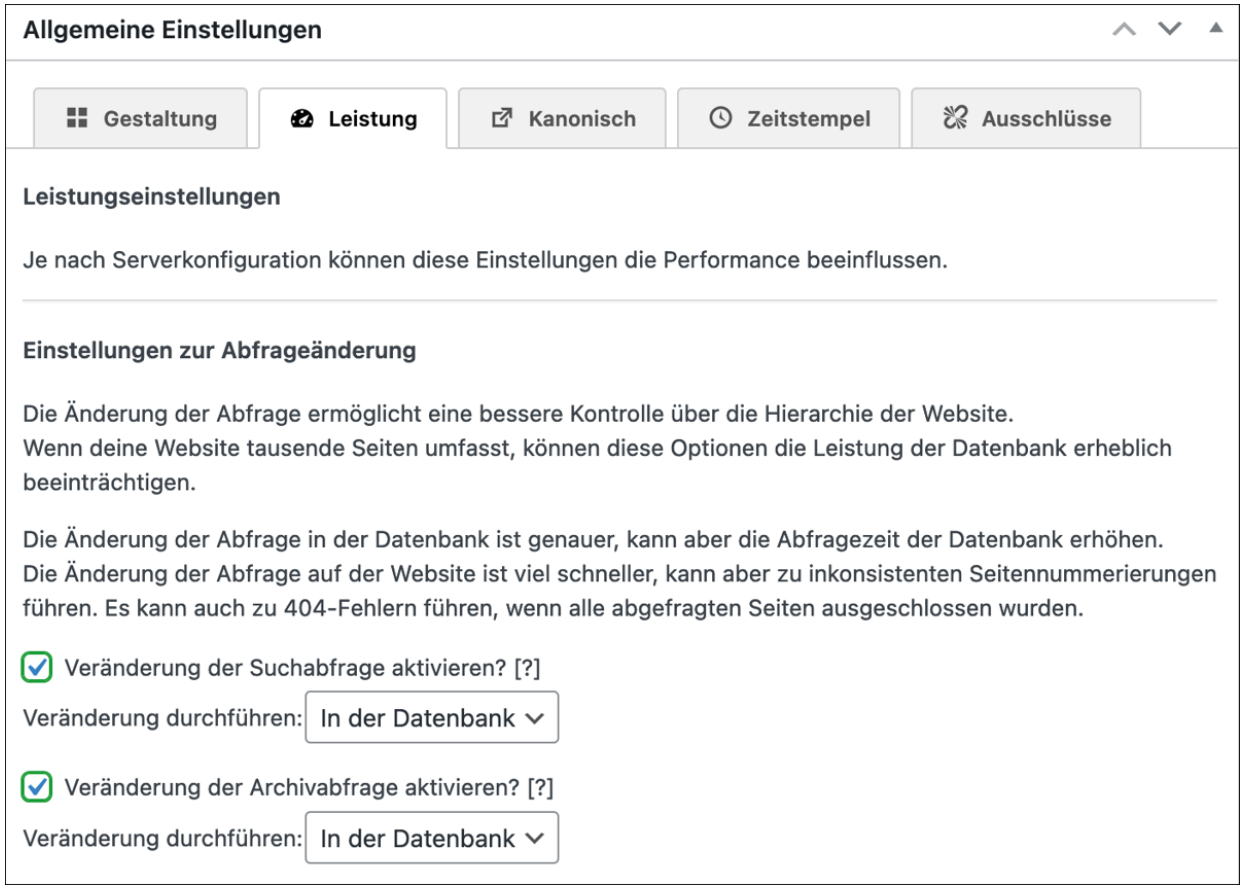


Abbildung 20.2 Sie können Seiten von der Suchfunktion auf der Website ausschließen und Inhalte aus den Archiv-Listen entfernen.

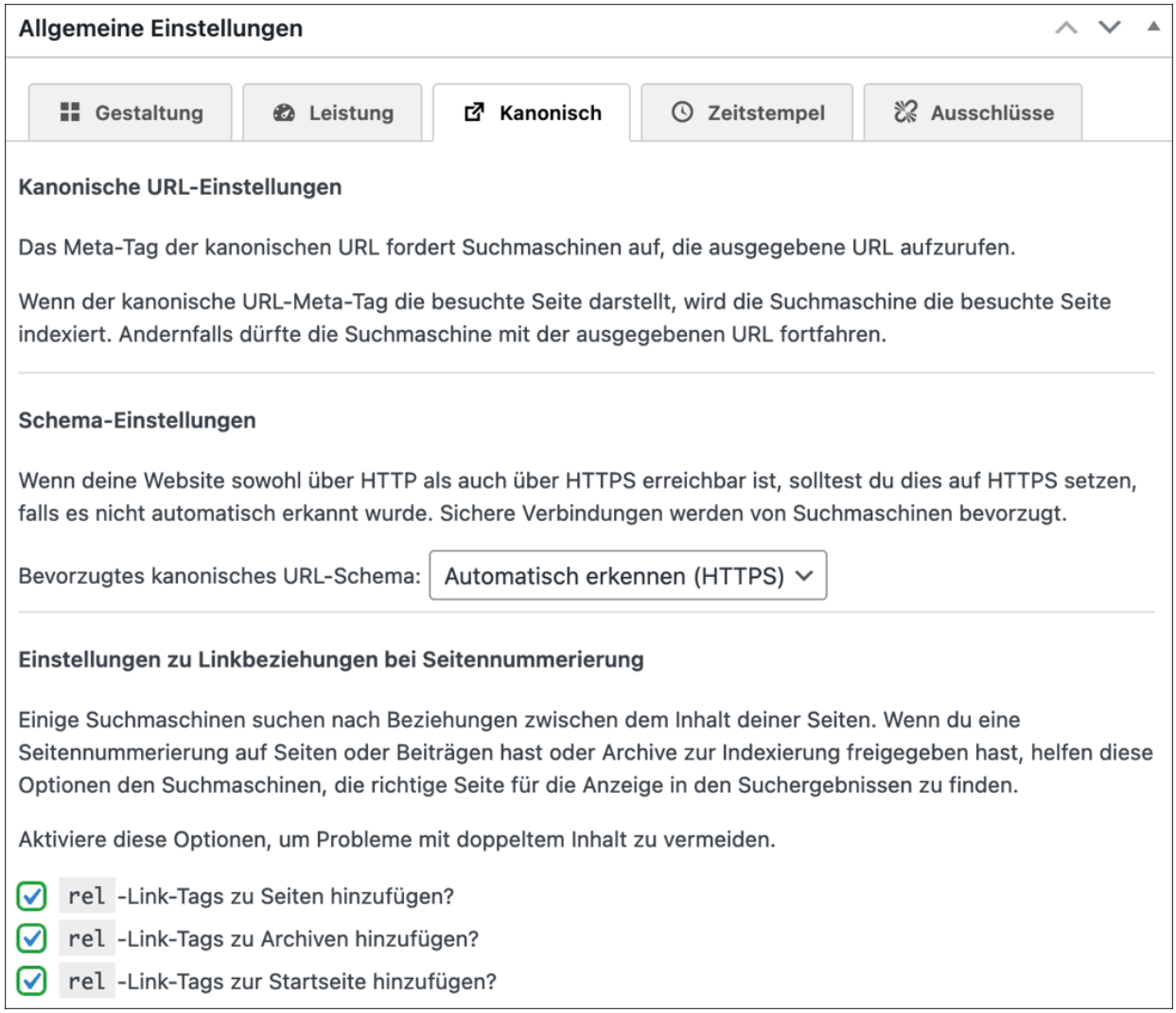


Abbildung 20.3 Die kanonische oder »canonical« URL ist eine Angabe für Suchmaschinen, welche URL für die Indexierung eines Inhalts verwendet werden soll, wenn ein Inhalt unter mehreren URLs verfügbar ist.

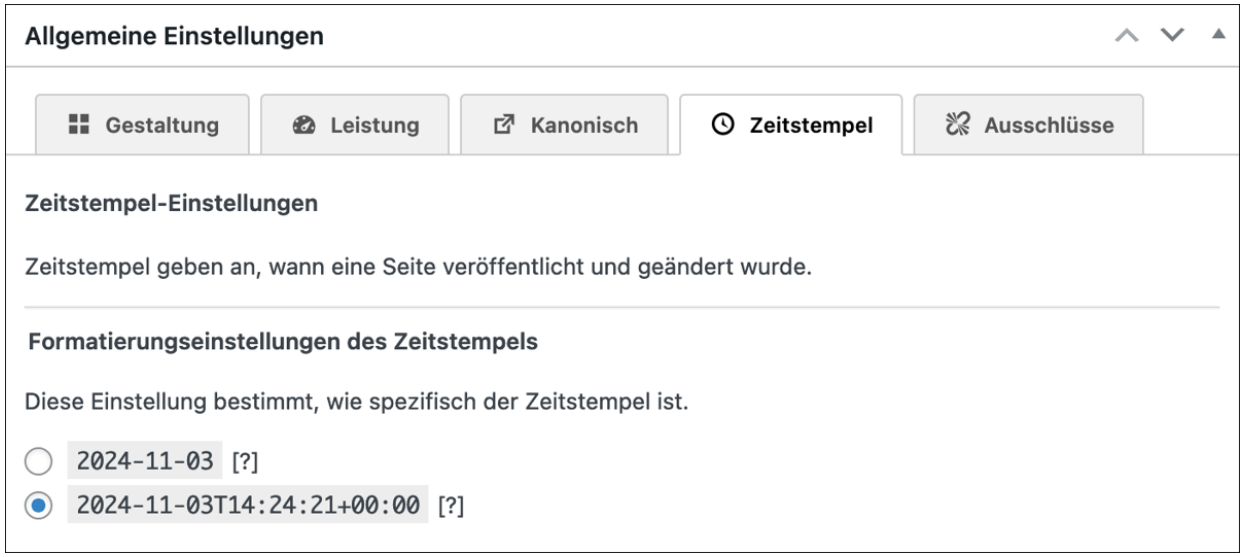


Abbildung 20.4 Hier können Sie das Datumsformat anpassen, das im Quelltext für das Veröffentlichungs- und Änderungsdatum eines Inhalts angezeigt wird.

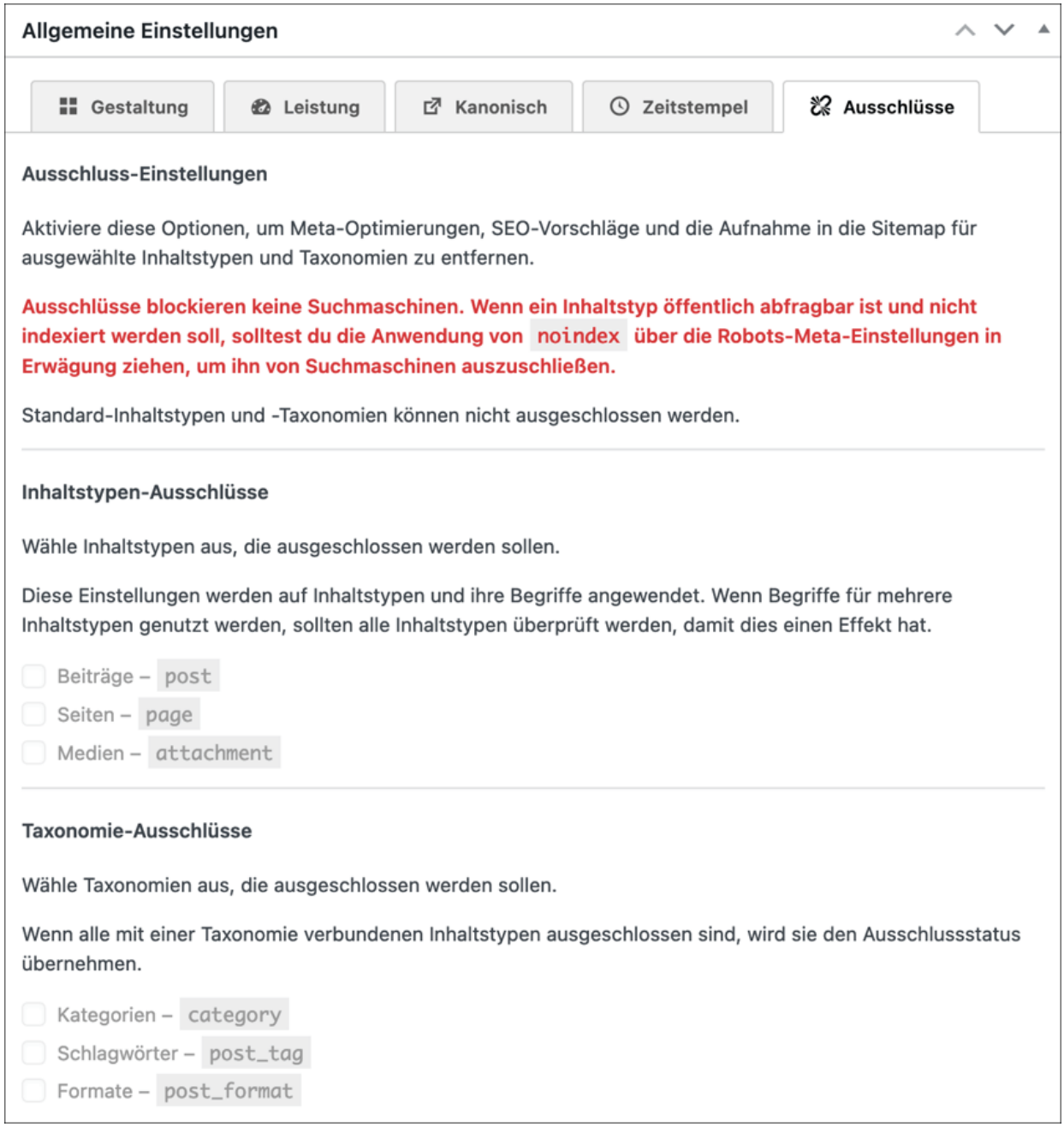


Abbildung 20.5 Sie können die Optimierung durch »The SEO Framework« für Inhaltstypen deaktivieren, die nicht standardmäßig von WordPress mitgebracht werden.

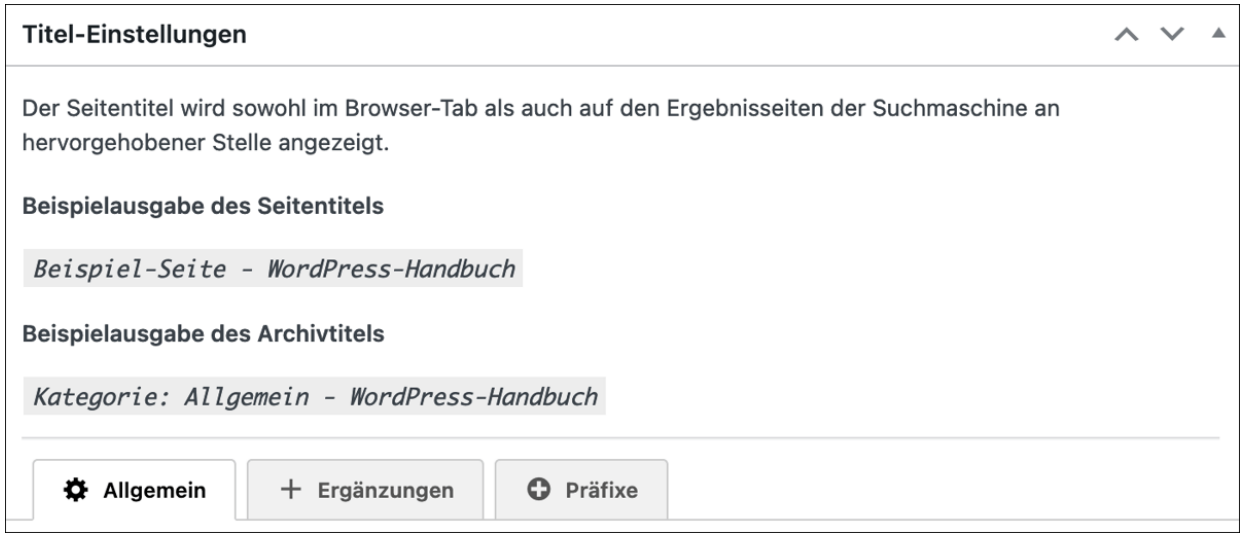


Abbildung 20.6 Oben in den »Titel-Einstellungen« wird eine Vorschau angezeigt.

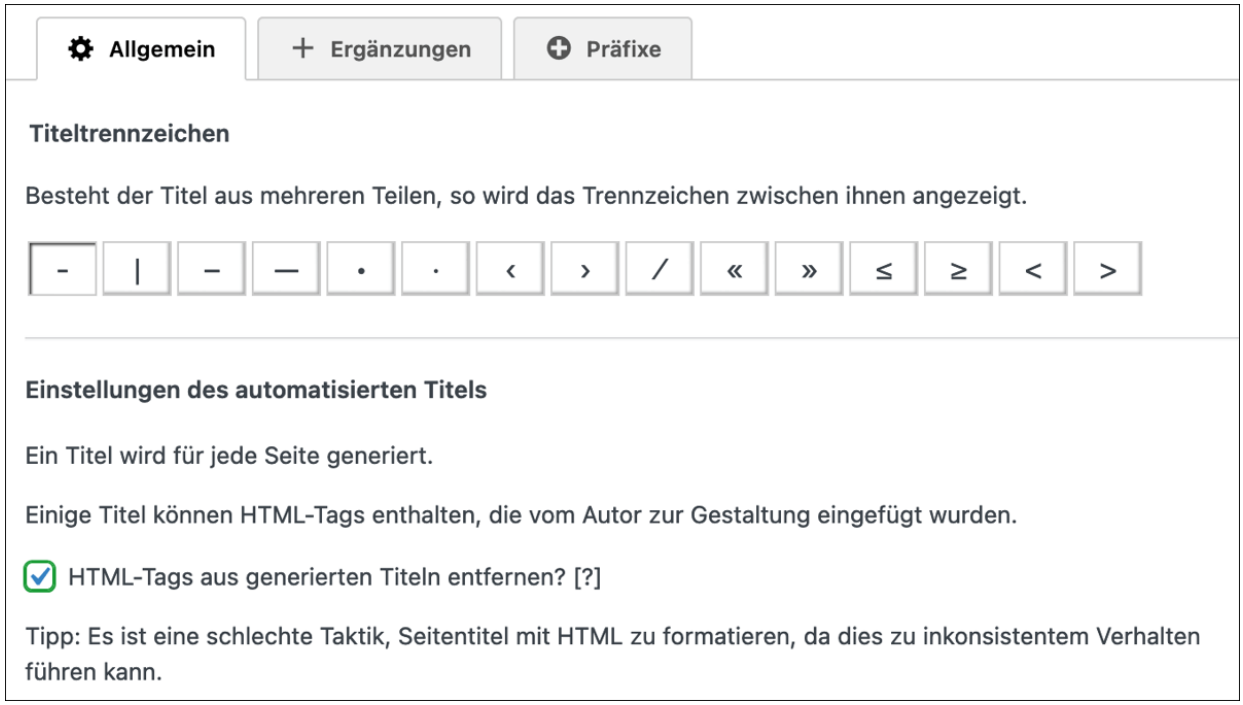


Abbildung 20.7 Hier können Sie das Trennzeichen für den Titel festlegen und steuern, ob HTML aus dem Titel entfernt werden soll.

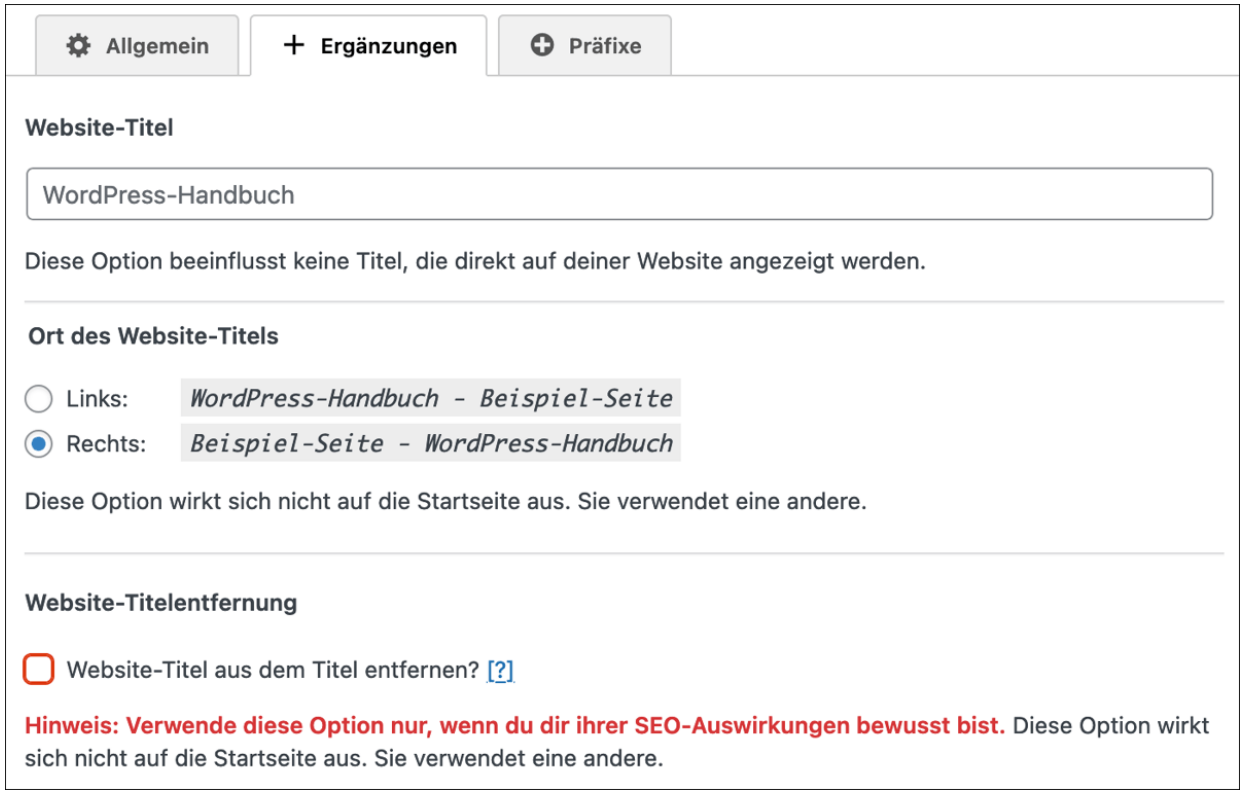


Abbildung 20.8 Im »Ergänzungen«-Tab können Sie einstellen, ob zusätzlich zum Titel beispielsweise einer Seite noch eine Ergänzung angezeigt werden soll.

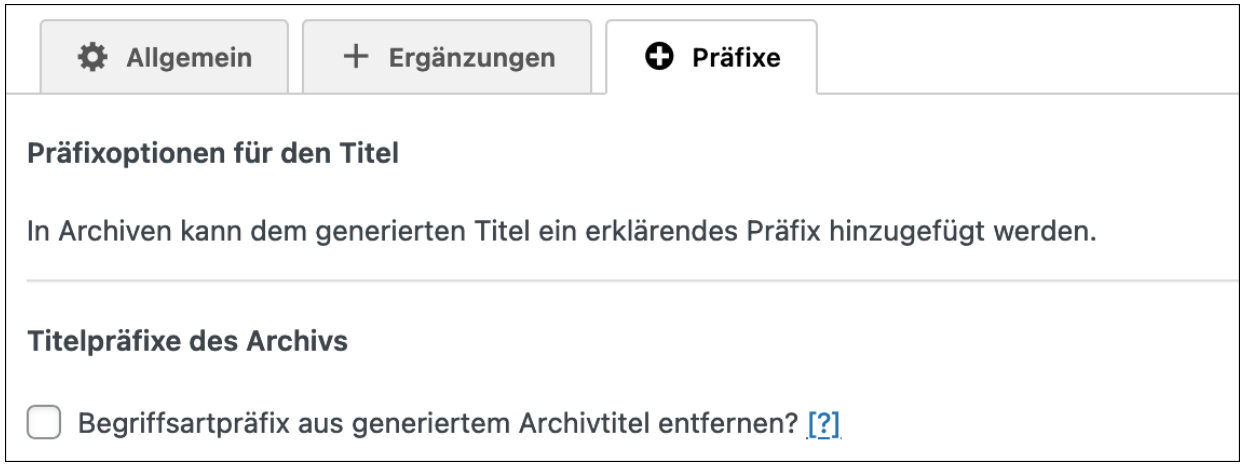


Abbildung 20.9 Hier finden Sie eine Einstellungsmöglichkeit für Titel von Archiv-Seiten.

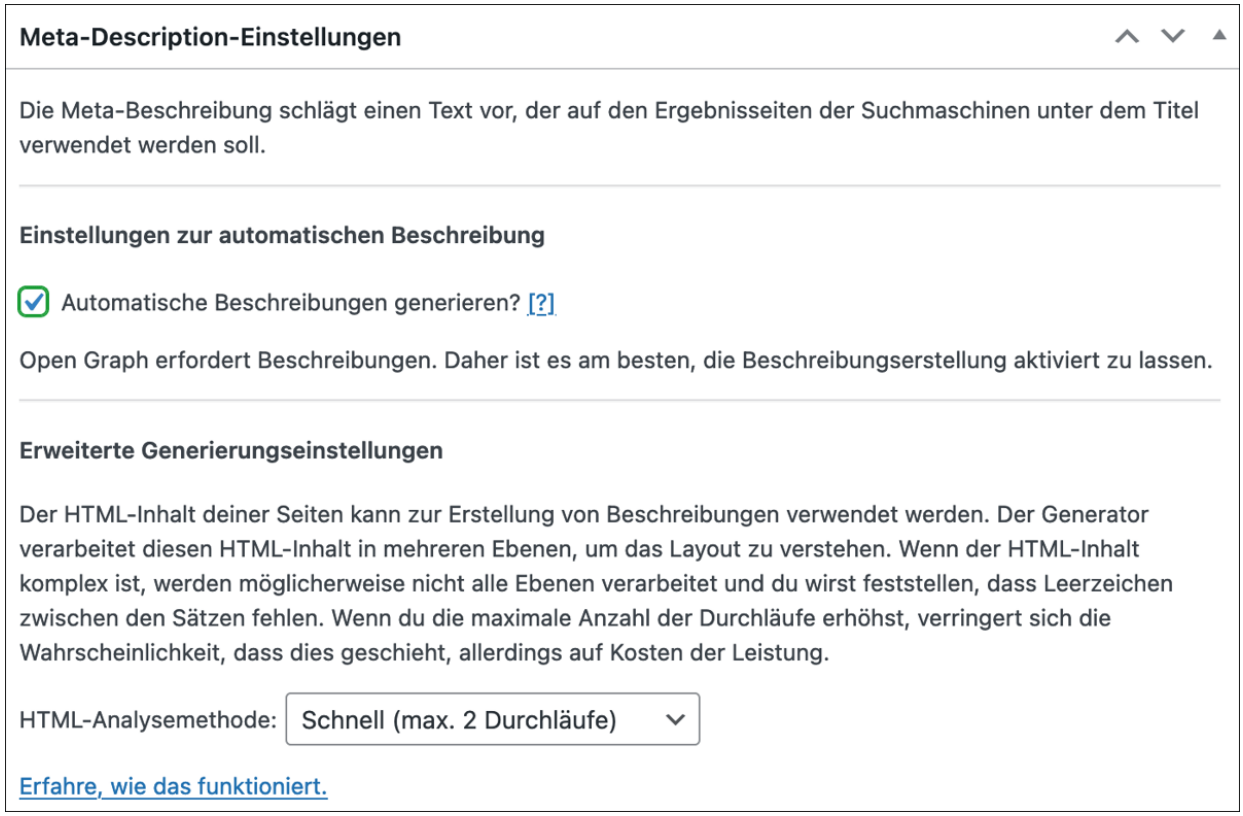


Abbildung 20.10 The SEO Framework bietet eine Option zur automatischen Generierung von Meta-Beschreibungen.

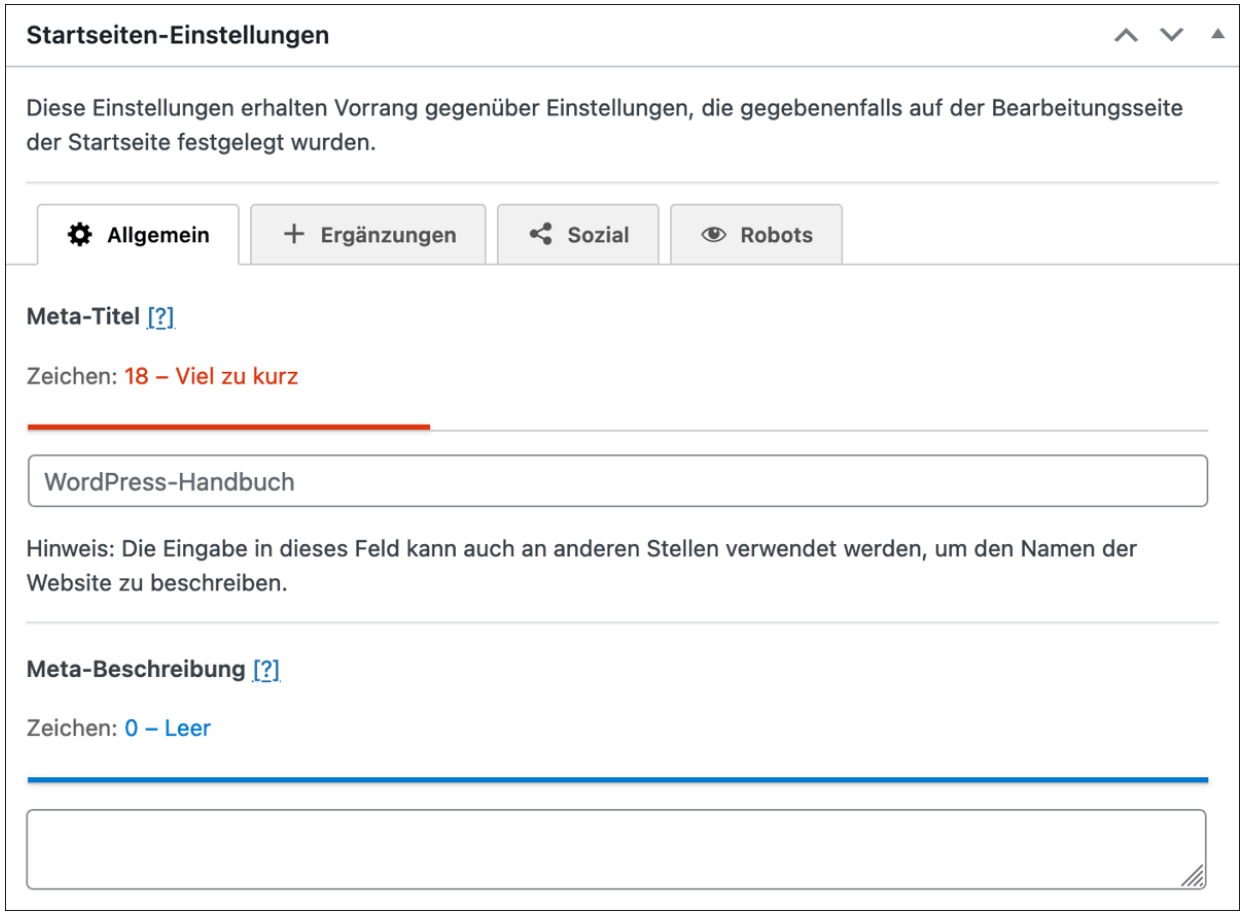


Abbildung 20.11 Hier können Sie Titel und Beschreibung der Startseite setzen.

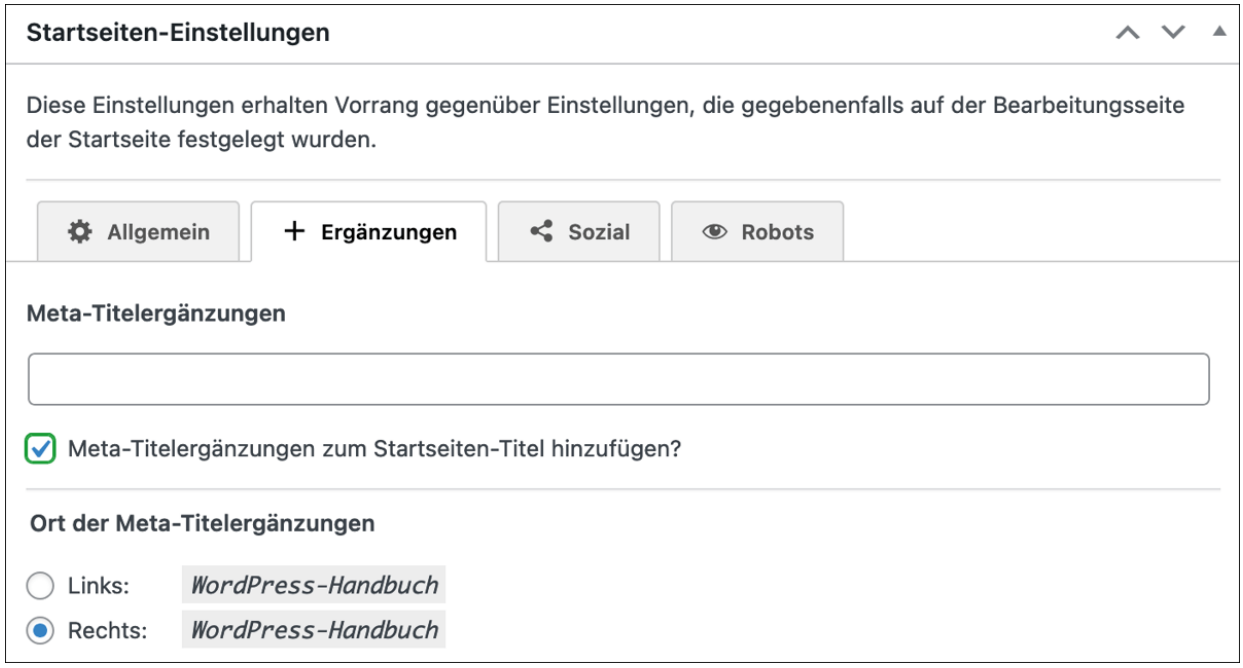
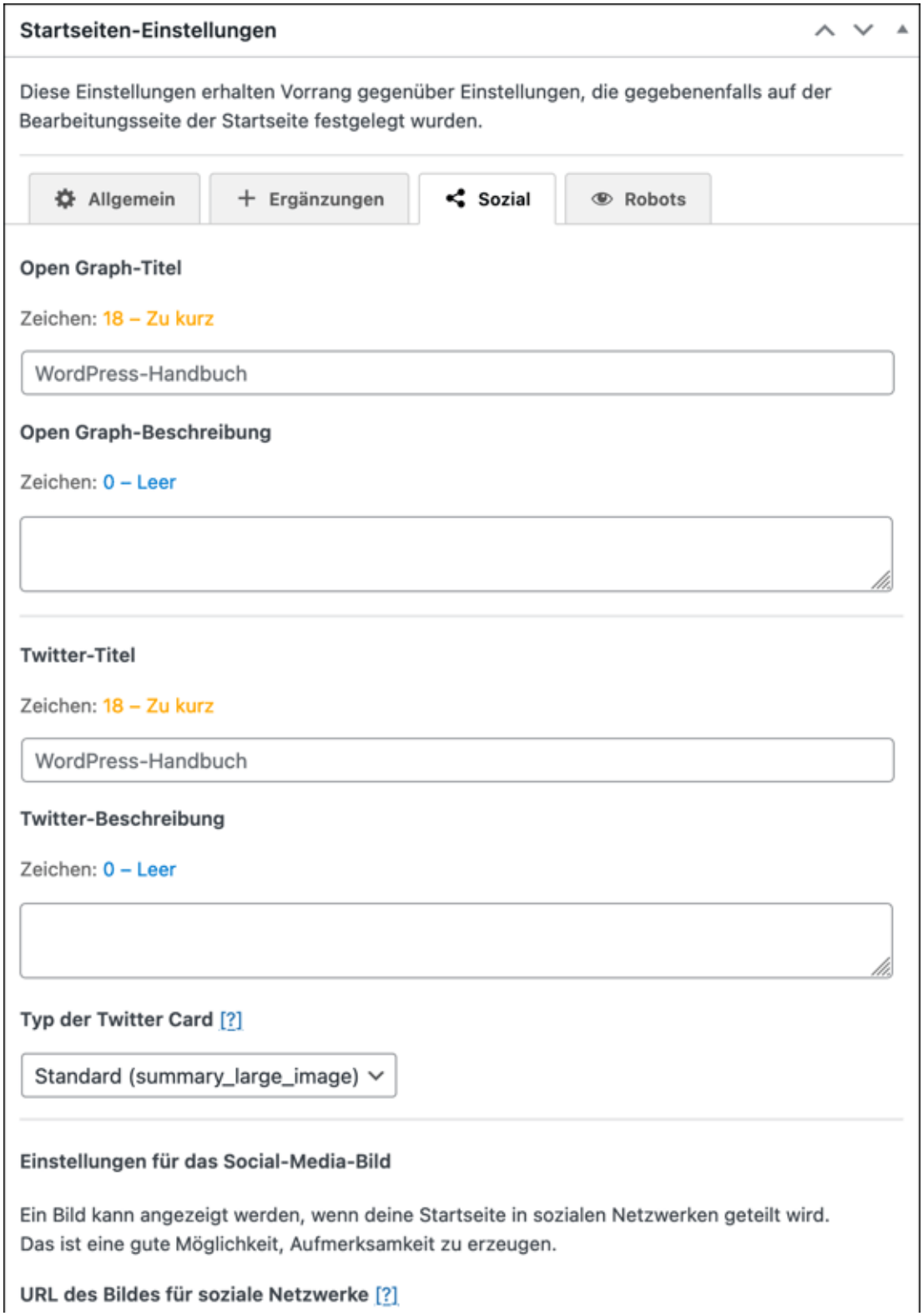


Abbildung 20.12 Einstellungen für Ergänzungen im Titel der Startseite


Startseiten-Einstellungen
Diese Einstellungen erhalten Vorrang gegenüber Einstellungen, die gegebenenfalls auf der Bearbeitungsseite der Startseite festgelegt wurden.
Allgemein
Ergänzungen
Sozial
Robots
Open Graph-Titel
Zeichen: 18 – Zu kurz
WordPress-Handbuch
Open Graph-Beschreibung
Zeichen: 0 – Leer
Twitter-Titel
Zeichen: 18 – Zu kurz
WordPress-Handbuch
Twitter-Beschreibung
Zeichen: 0 – Leer
Typ der Twitter Card [?]
Standard (summary_large_image)
Einstellungen für das Social-Media-Bild
Ein Bild kann angezeigt werden, wenn deine Startseite in sozialen Netzwerken geteilt wird.
Das ist eine gute Möglichkeit, Aufmerksamkeit zu erzeugen.
URL des Bildes für soziale Netzwerke [?]

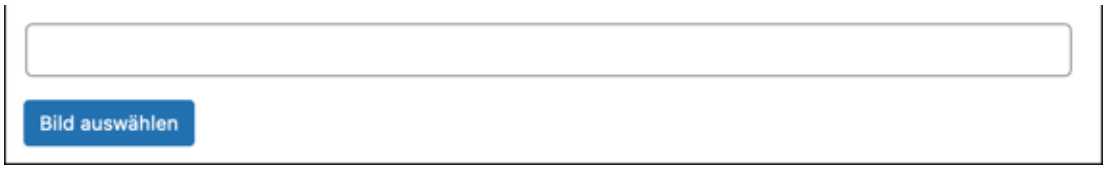


Abbildung 20.13 Hier können Sie die Werte für die Meta-Angaben für soziale Netzwerke anpassen.

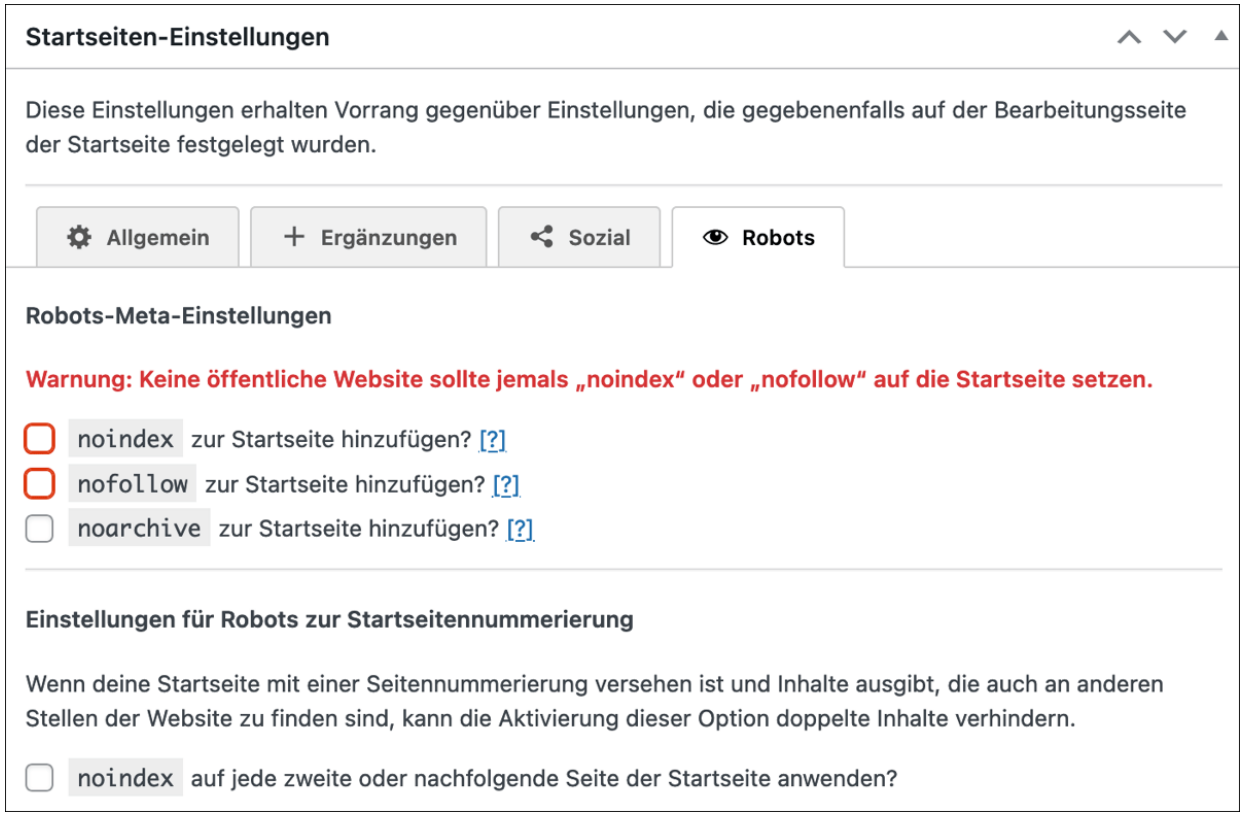


Abbildung 20.14 Hier finden Sie ein paar potenziell gefährliche Einstellungen, was die Sichtbarkeit Ihrer Website angeht.

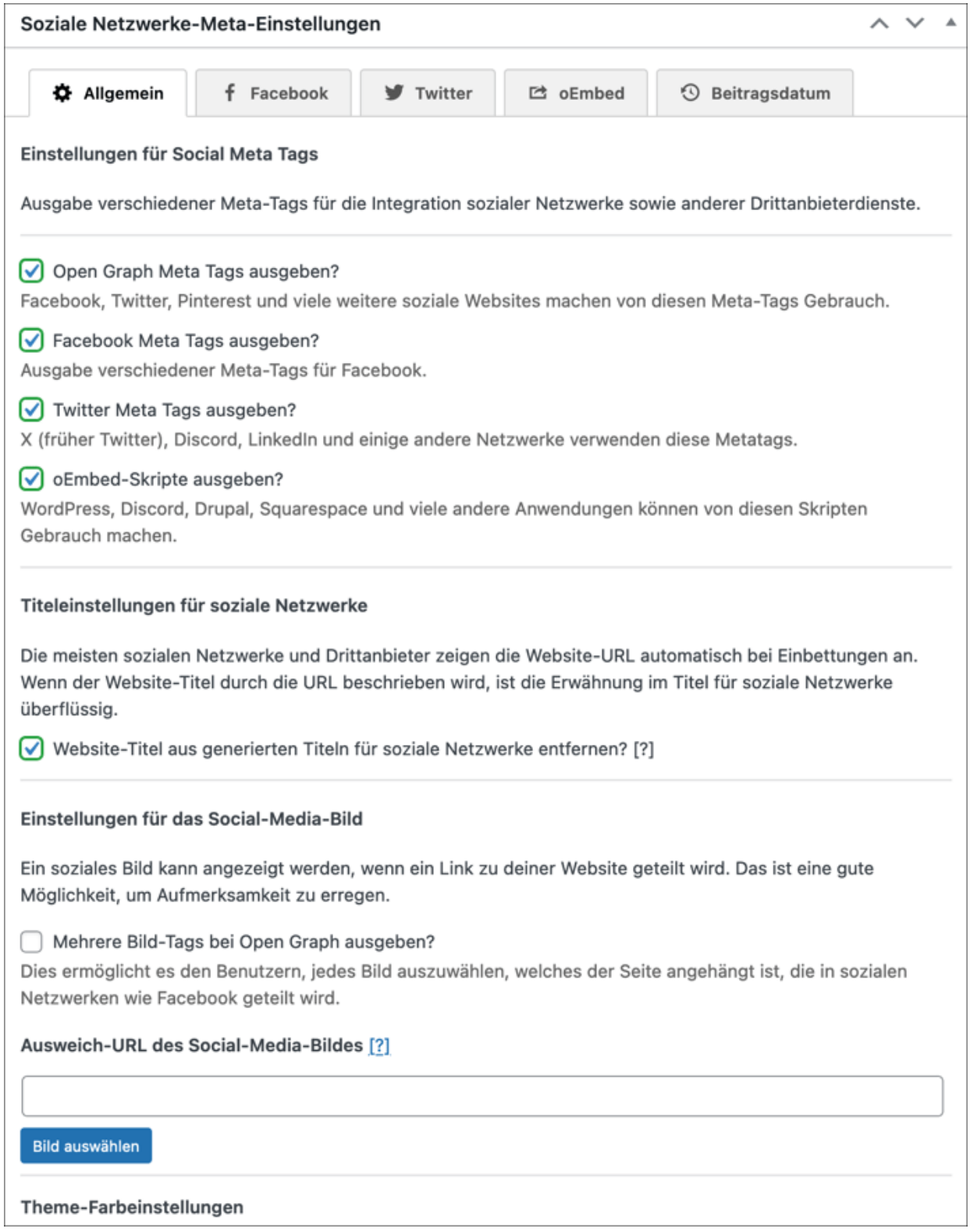

Soziale Netzwerke-Meta-Einstellungen
Allgemein
Facebook
Twitter
oEmbed
Beitragsdatum
Einstellungen für Social Meta Tags
Ausgabe verschiedener Meta-Tags für die Integration sozialer Netzwerke sowie anderer Drittanbieterdienste.
Open Graph Meta Tags ausgeben?
Facebook, Twitter, Pinterest und viele weitere soziale Websites machen von diesen Meta-Tags Gebrauch.
Facebook Meta Tags ausgeben?
Ausgabe verschiedener Meta-Tags für Facebook.
Twitter Meta Tags ausgeben?
X (früher Twitter), Discord, LinkedIn und einige andere Netzwerke verwenden diese Metatags.
oEmbed-Skripte ausgeben?
WordPress, Discord, Drupal, Squarespace und viele andere Anwendungen können von diesen Skripten Gebrauch machen.
Titeleinstellungen für soziale Netzwerke
Die meisten sozialen Netzwerke und Drittanbieter zeigen die Website-URL automatisch bei Einbettungen an. Wenn der Website-Titel durch die URL beschrieben wird, ist die Erwähnung im Titel für soziale Netzwerke überflüssig.
Website-Titel aus generierten Titeln für soziale Netzwerke entfernen? [?]
Einstellungen für das Social-Media-Bild
Ein soziales Bild kann angezeigt werden, wenn ein Link zu deiner Website geteilt wird. Das ist eine gute Möglichkeit, um Aufmerksamkeit zu erregen.
Mehrere Bild-Tags bei Open Graph ausgeben?
Dies ermöglicht es den Benutzern, jedes Bild auszuwählen, welches der Seite angehängt ist, die in sozialen Netzwerken wie Facebook geteilt wird.
Ausweich-URL des Social-Media-Bildes [?]
Bild auswählen
Theme-Farbeinstellungen

Abbildung 20.15 In den allgemeinen Einstellungen zu den Meta-Einstellungen für soziale Netzwerke können Sie unter anderem die Ausgabe verschiedener Meta-Tag-Arten für unterschiedliche Netzwerke aktivieren.



Abbildung 20.16 Hier können Sie Facebook-spezifische Informationen zu Ihrer Website hinterlegen.

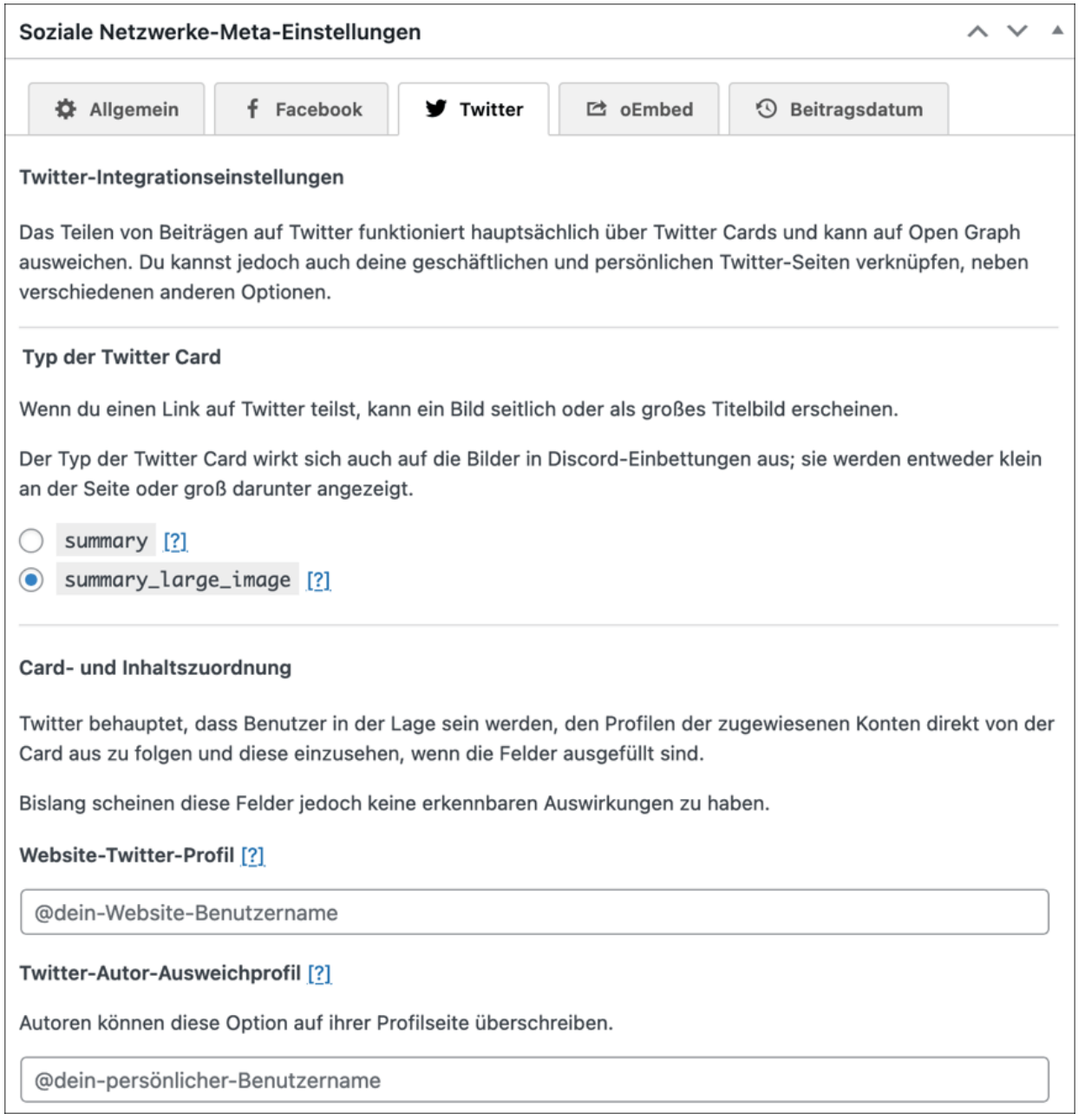


Abbildung 20.17 Für Twitter können Sie unter anderem den Twitter-Card-Typ definieren.

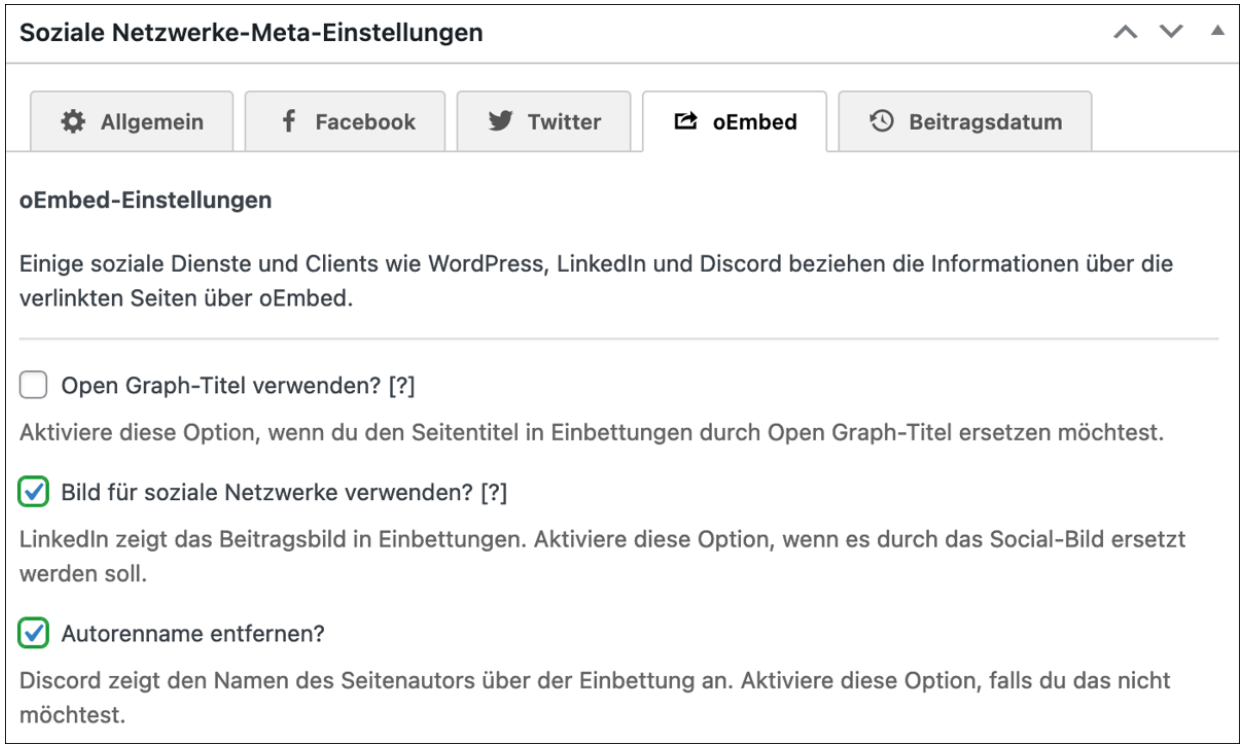


Abbildung 20.18 In den oEmbed-Einstellungen können Sie ein paar Eigenschaften verändern, die das Embed betreffen, wenn beispielsweise auf einer WordPress-Site ein Inhalt einer anderen WordPress-Site mittels eines oEmbeds als Vorschau angezeigt wird.

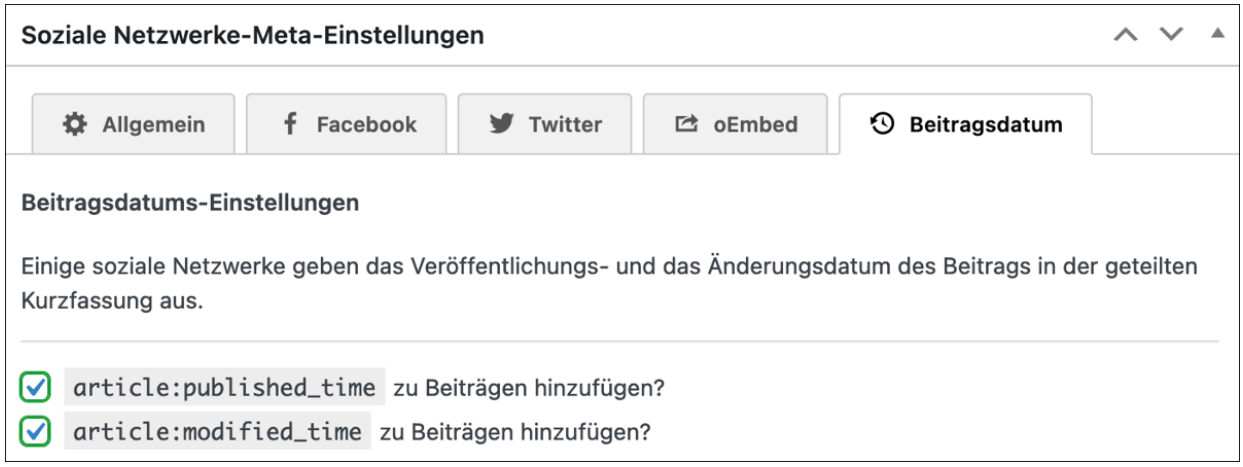
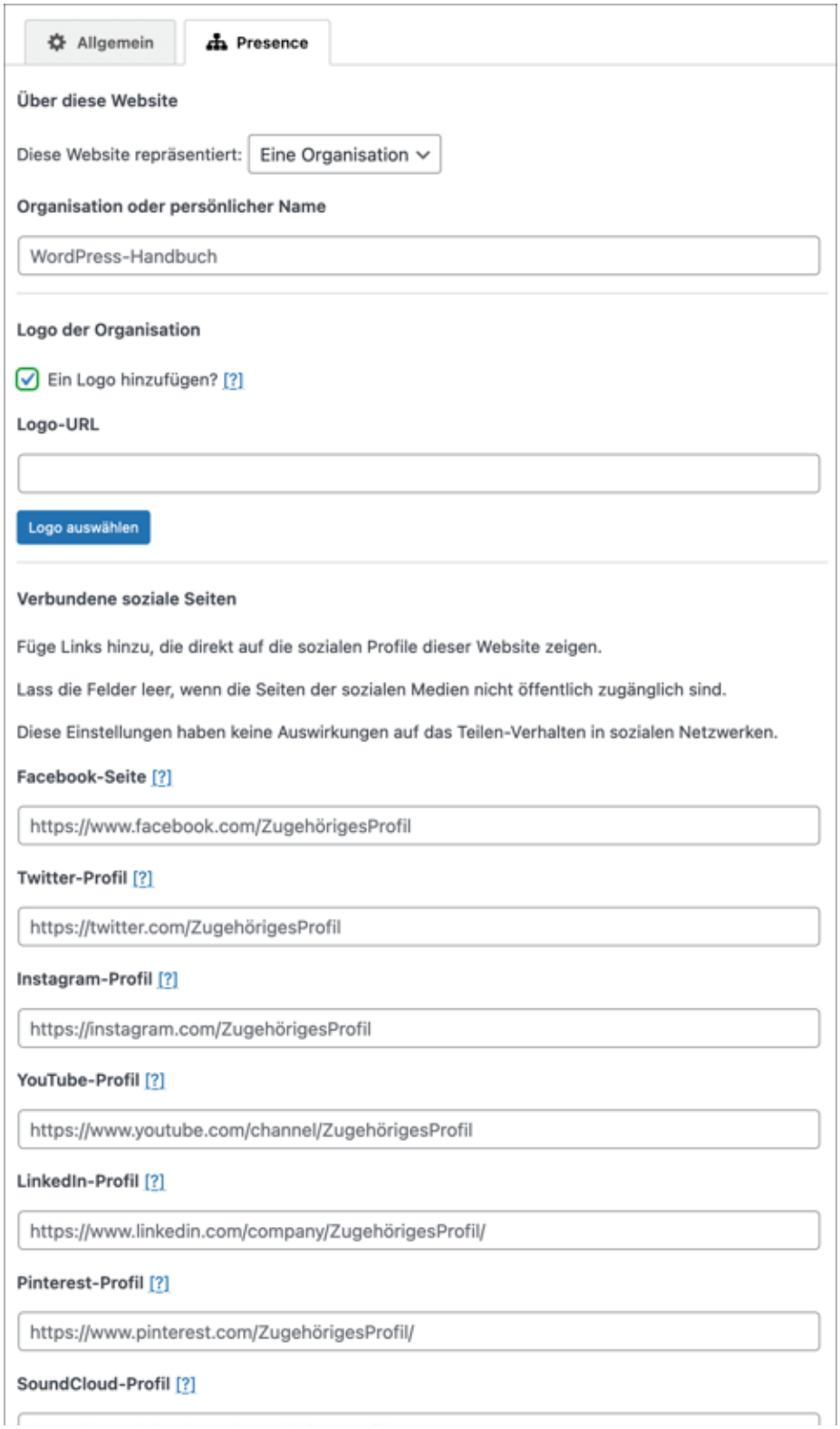


Abbildung 20.19 Hier steuern Sie, ob das Veröffentlichungs- und Änderungsdatum über entsprechende Tags ausgegeben wird.



Abbildung 20.20 Hier können Sie unter anderem angeben, ob überhaupt Schema.org-Markup für strukturierte Daten ausgegeben werden soll.


Allgemein
Presence
Über diese Website
Diese Website repräsentiert:
Eine Organisation
Organisation oder persönlicher Name
WordPress-Handbuch
Logo der Organisation
Ein Logo hinzufügen? [?]
Logo-URL
Logo auswählen
Verbundene soziale Seiten
Füge Links hinzu, die direkt auf die sozialen Profile dieser Website zeigen.
Lass die Felder leer, wenn die Seiten der sozialen Medien nicht öffentlich zugänglich sind.
Diese Einstellungen haben keine Auswirkungen auf das Teilen-Verhalten in sozialen Netzwerken.
Facebook-Seite [?]
https://www.facebook.com/ZugehörigesProfil
Twitter-Profil [?]
https://twitter.com/ZugehörigesProfil
Instagram-Profil [?]
https://instagram.com/ZugehörigesProfil
YouTube-Profil [?]
https://www.youtube.com/channel/ZugehörigesProfil
LinkedIn-Profil [?]
https://www.linkedin.com/company/ZugehörigesProfil/
Pinterest-Profil [?]
https://www.pinterest.com/ZugehörigesProfil/
SoundCloud-Profil [?]

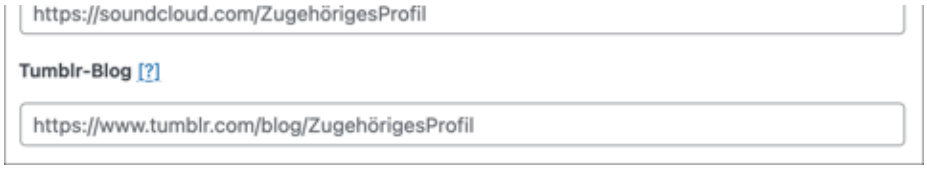


Abbildung 20.21 Im »Presence«-Tab können Sie angeben, ob die Site eine Person oder Organisation repräsentiert.

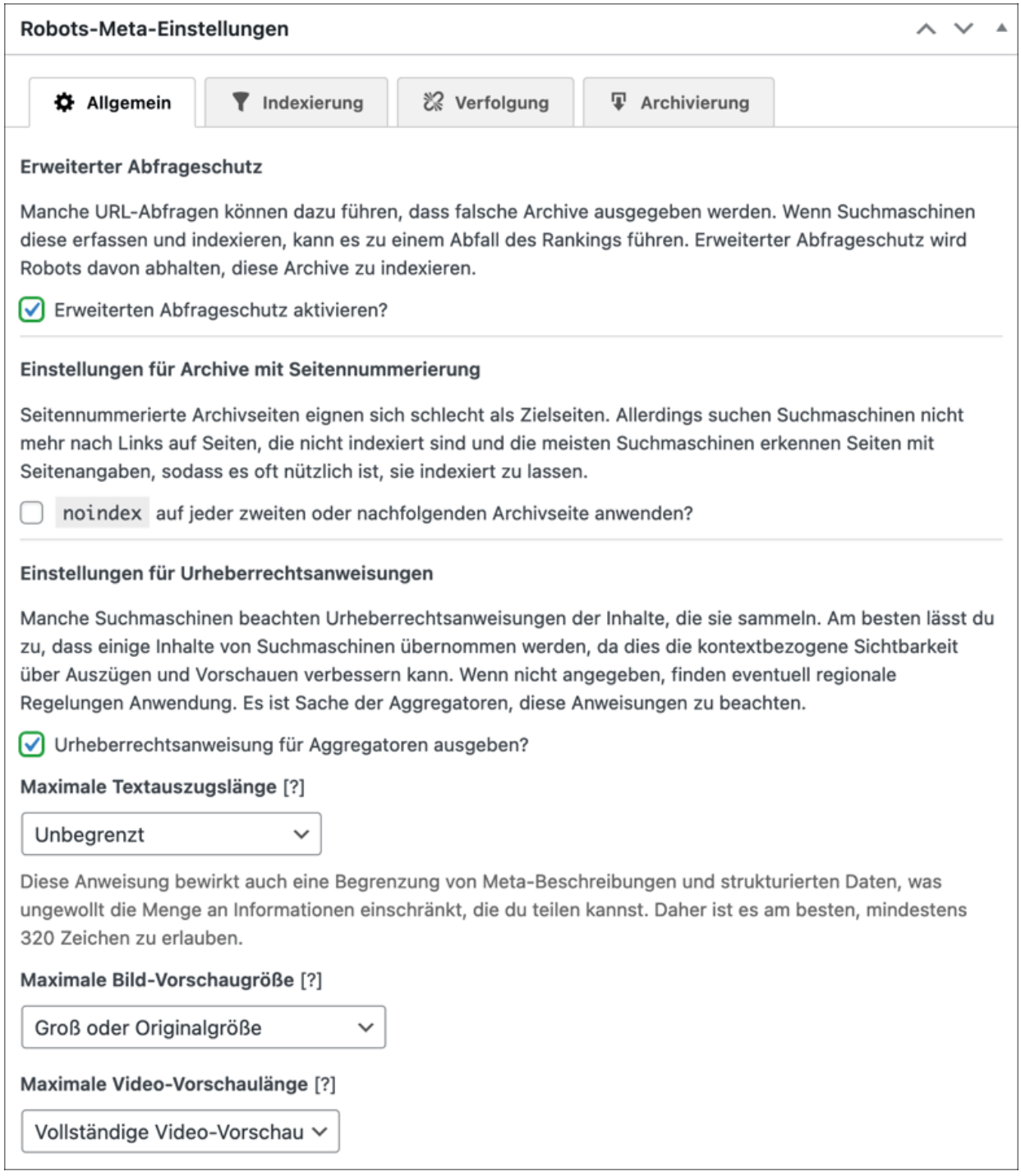

Robots-Meta-Einstellungen
Allgemein
Indexierung
Verfolgung
Archivierung
Erweiterter Abfrageschutz
Manche URL-Abfragen können dazu führen, dass falsche Archive ausgegeben werden. Wenn Suchmaschinen diese erfassen und indexieren, kann es zu einem Abfall des Rankings führen. Erweiterter Abfrageschutz wird Robots davon abhalten, diese Archive zu indexieren.
Erweiterten Abfrageschutz aktivieren?
Einstellungen für Archive mit Seitennummerierung
Seitennummerierte Archivseiten eignen sich schlecht als Zielseiten. Allerdings suchen Suchmaschinen nicht mehr nach Links auf Seiten, die nicht indexiert sind und die meisten Suchmaschinen erkennen Seiten mit Seitenangaben, sodass es oft nützlich ist, sie indexiert zu lassen.
noindex auf jeder zweiten oder nachfolgenden Archivseite anwenden?
Einstellungen für Urheberrechtsanweisungen
Manche Suchmaschinen beachten Urheberrechtsanweisungen der Inhalte, die sie sammeln. Am besten lässt du zu, dass einige Inhalte von Suchmaschinen übernommen werden, da dies die kontextbezogene Sichtbarkeit über Auszügen und Vorschauen verbessern kann. Wenn nicht angegeben, finden eventuell regionale Regelungen Anwendung. Es ist Sache der Aggregatoren, diese Anweisungen zu beachten.
Urheberrechtsanweisung für Aggregatoren ausgeben?
Maximale Textauszugslänge [?]
Unbegrenzt
Diese Anweisung bewirkt auch eine Begrenzung von Meta-Beschreibungen und strukturierten Daten, was ungewollt die Menge an Informationen einschränkt, die du teilen kannst. Daher ist es am besten, mindestens 320 Zeichen zu erlauben.
Maximale Bild-Vorschaugröße [?]
Groß oder Originalgröße
Maximale Video-Vorschaulänge [?]
Vollständige Video-Vorschau

Abbildung 20.22 Die allgemeinen Einstellungen für die Robots-Metadaten betreffen beispielsweise das Setzen von »noindex« für paginierte Archive.

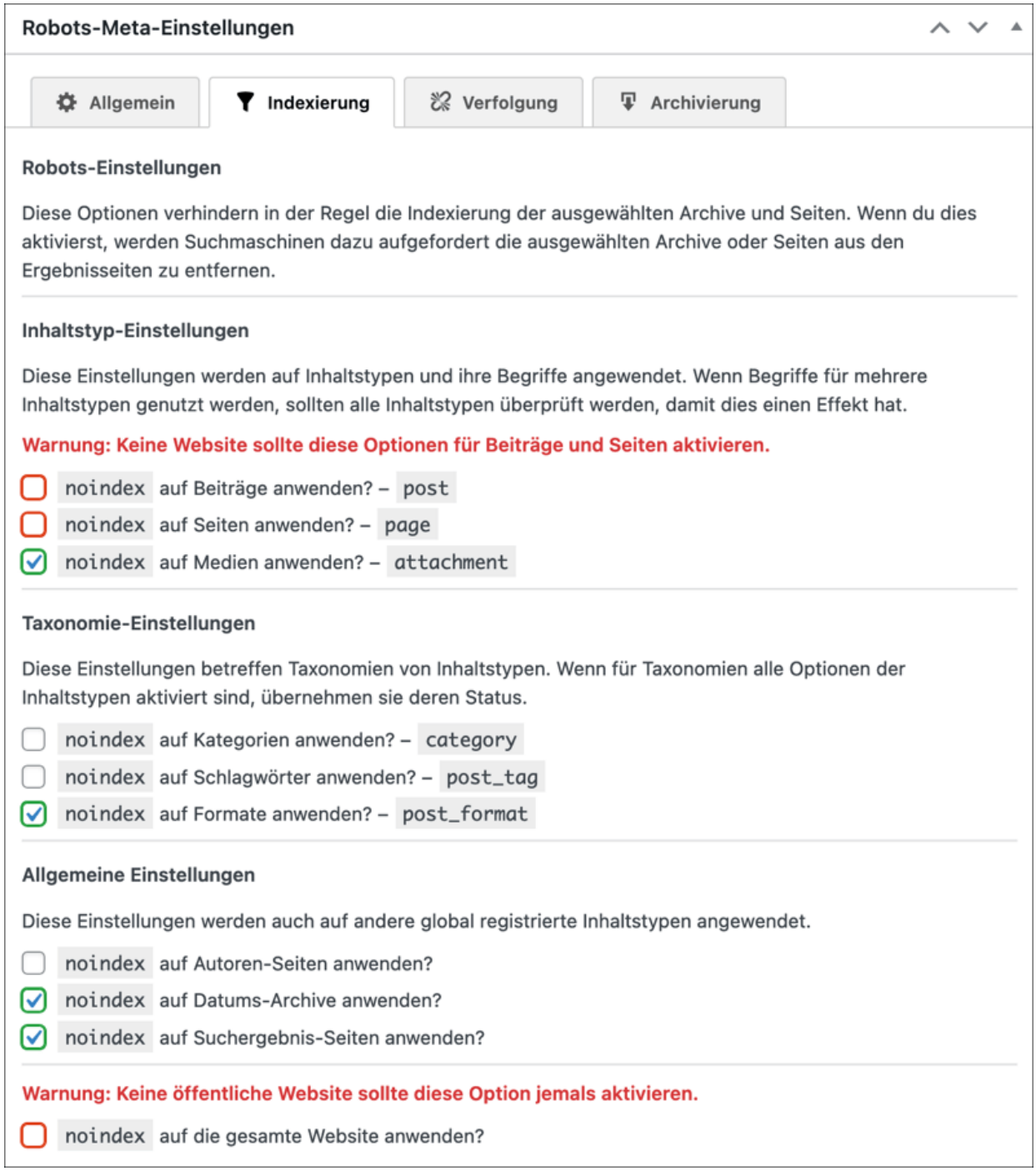

Robots-Meta-Einstellungen
Allgemein
Indexierung
Verfolgung
Archivierung
Robots-Einstellungen
Diese Optionen verhindern in der Regel die Indexierung der ausgewählten Archive und Seiten. Wenn du dies aktivierst, werden Suchmaschinen dazu aufgefordert die ausgewählten Archive oder Seiten aus den Ergebnisseiten zu entfernen.
Inhaltstyp-Einstellungen
Diese Einstellungen werden auf Inhaltstypen und ihre Begriffe angewendet. Wenn Begriffe für mehrere Inhaltstypen genutzt werden, sollten alle Inhaltstypen überprüft werden, damit dies einen Effekt hat.
Warnung: Keine Website sollte diese Optionen für Beiträge und Seiten aktivieren.
noindex auf Beiträge anwenden? – post
noindex auf Seiten anwenden? – page
noindex auf Medien anwenden? – attachment
Taxonomie-Einstellungen
Diese Einstellungen betreffen Taxonomien von Inhaltstypen. Wenn für Taxonomien alle Optionen der Inhaltstypen aktiviert sind, übernehmen sie deren Status.
noindex auf Kategorien anwenden? – category
noindex auf Schlagwörter anwenden? – post_tag
noindex auf Formate anwenden? – post_format
Allgemeine Einstellungen
Diese Einstellungen werden auch auf andere global registrierte Inhaltstypen angewendet.
noindex auf Autoren-Seiten anwenden?
noindex auf Datums-Archive anwenden?
noindex auf Suchergebnis-Seiten anwenden?
Warnung: Keine öffentliche Website sollte diese Option jemals aktivieren.
noindex auf die gesamte Website anwenden?

Abbildung 20.23 Hier können Sie je Inhaltsart und Taxonomie einstellen, ob diese generell nicht indexiert werden sollen.

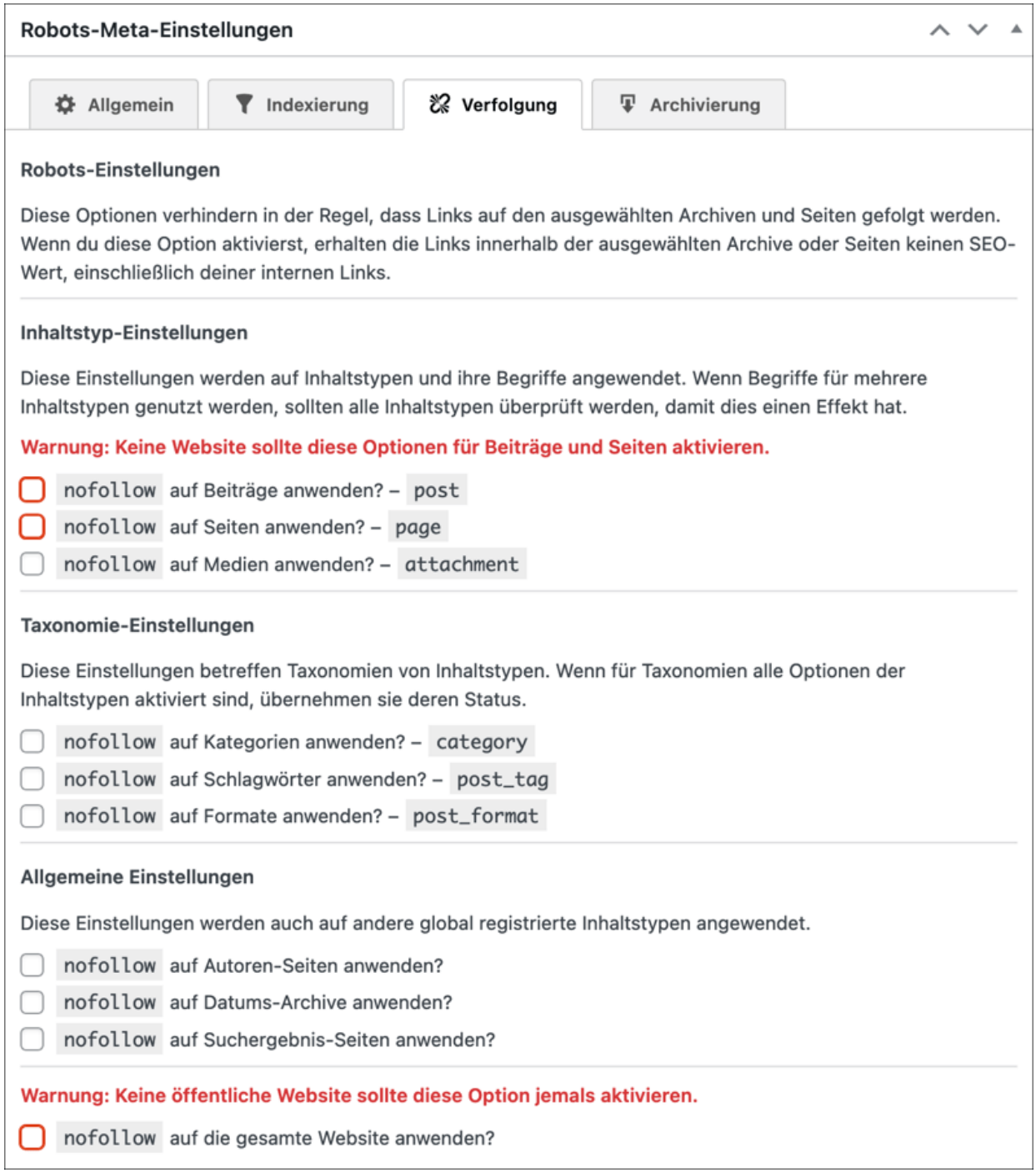

Robots-Meta-Einstellungen
Allgemein
Indexierung
Verfolgung
Archivierung
Robots-Einstellungen
Diese Optionen verhindern in der Regel, dass Links auf den ausgewählten Archiven und Seiten gefolgt werden. Wenn du diese Option aktivierst, erhalten die Links innerhalb der ausgewählten Archive oder Seiten keinen SEO-Wert, einschließlich deiner internen Links.
Inhaltstyp-Einstellungen
Diese Einstellungen werden auf Inhaltstypen und ihre Begriffe angewendet. Wenn Begriffe für mehrere Inhaltstypen genutzt werden, sollten alle Inhaltstypen überprüft werden, damit dies einen Effekt hat.
Warnung: Keine Website sollte diese Optionen für Beiträge und Seiten aktivieren.
nofollow auf Beiträge anwenden? – post
nofollow auf Seiten anwenden? – page
nofollow auf Medien anwenden? – attachment
Taxonomie-Einstellungen
Diese Einstellungen betreffen Taxonomien von Inhaltstypen. Wenn für Taxonomien alle Optionen der Inhaltstypen aktiviert sind, übernehmen sie deren Status.
nofollow auf Kategorien anwenden? – category
nofollow auf Schlagwörter anwenden? – post_tag
nofollow auf Formate anwenden? – post_format
Allgemeine Einstellungen
Diese Einstellungen werden auch auf andere global registrierte Inhaltstypen angewendet.
nofollow auf Autoren-Seiten anwenden?
nofollow auf Datums-Archive anwenden?
nofollow auf Suchergebnis-Seiten anwenden?
Warnung: Keine öffentliche Website sollte diese Option jemals aktivieren.
nofollow auf die gesamte Website anwenden?

Abbildung 20.24 Im »Verfolgung«-Tab finden Sie die entsprechenden Einstellungen von der »Noindex«-Seite für den »nofollow«-Wert.

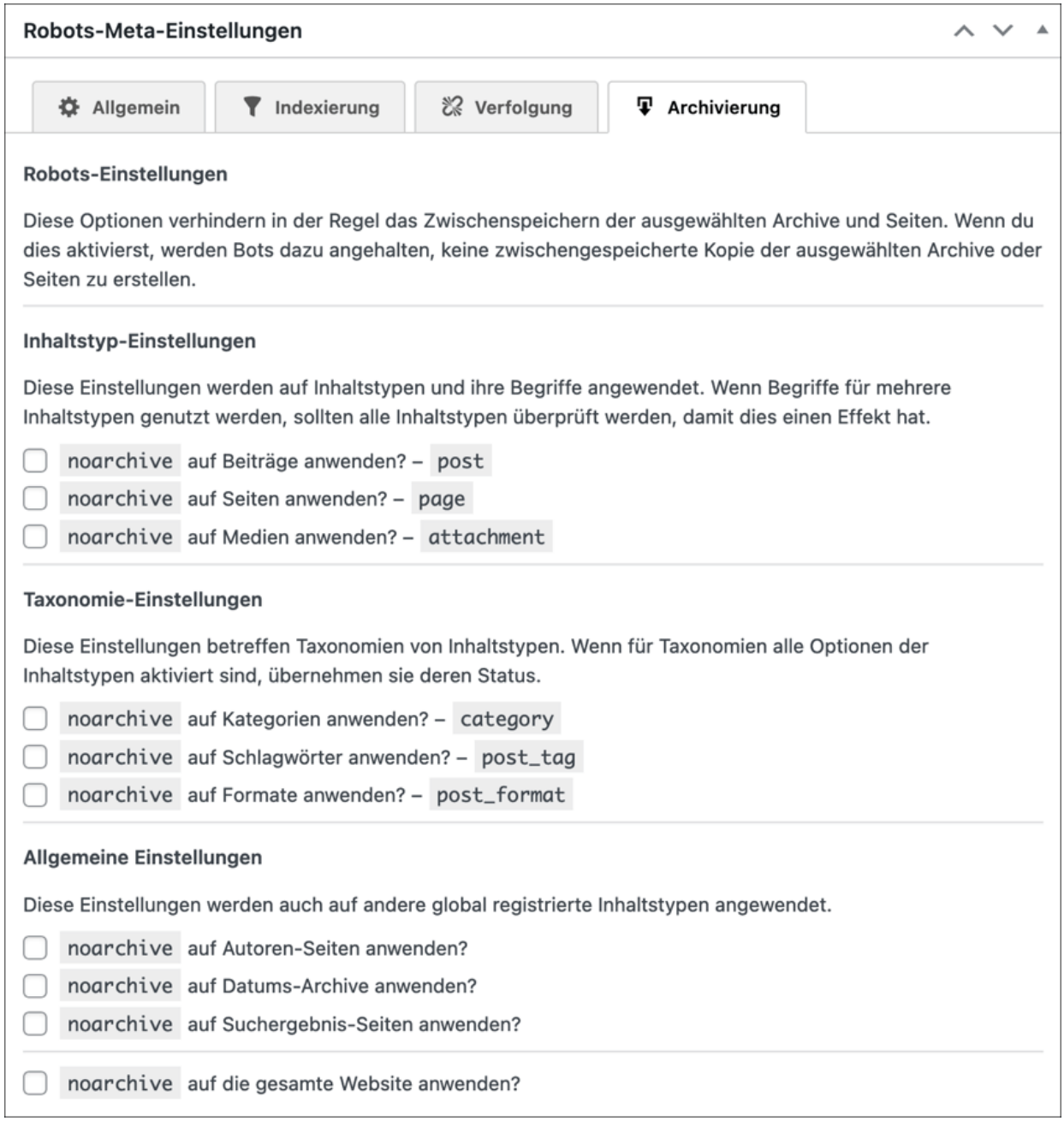


Abbildung 20.25 Hier können Sie die Einstellungen für den Wert »noarchive« anpassen.

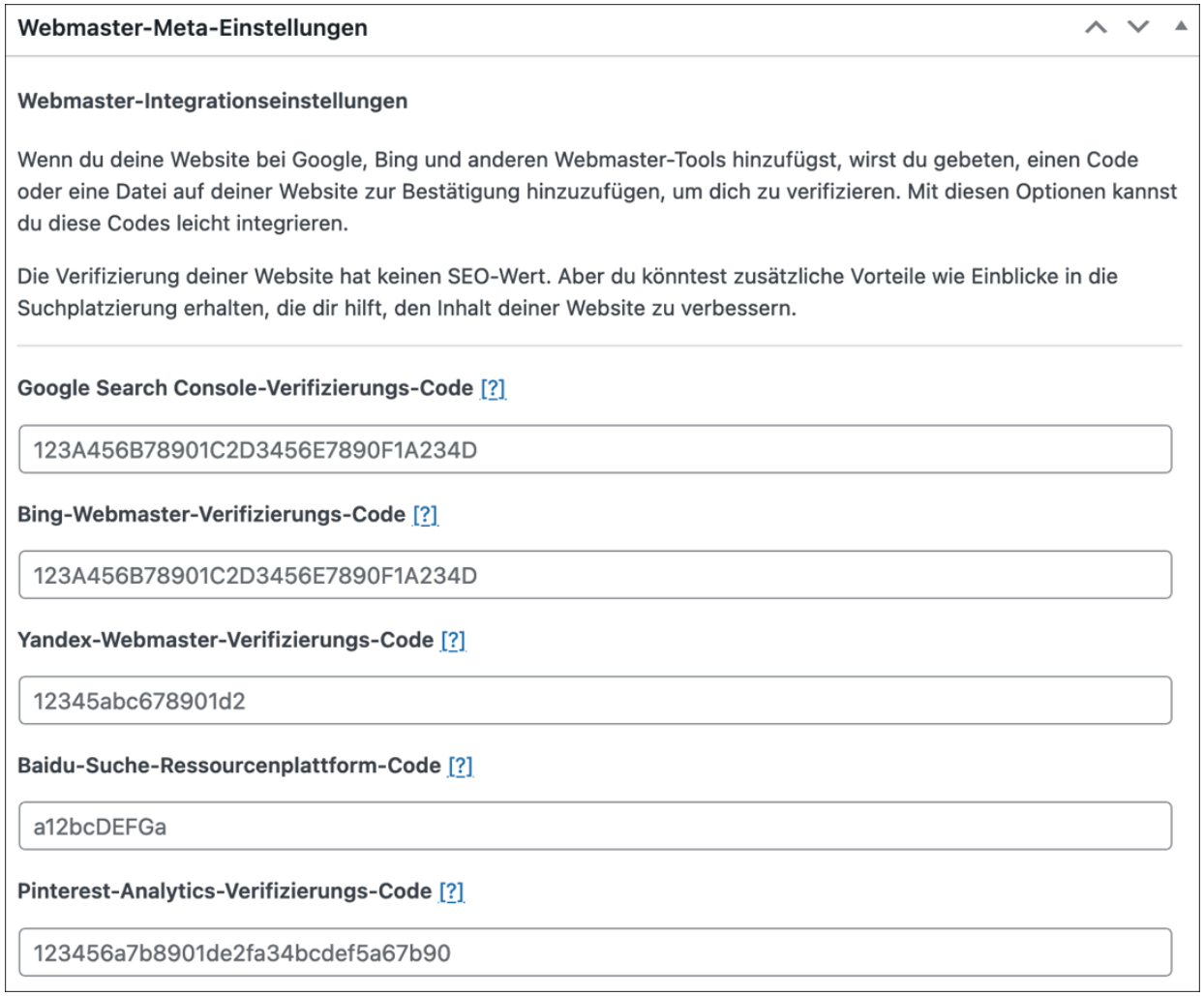


Abbildung 20.26 Wenn Sie sich bei einem oder mehreren der Webmaster-Tools von Suchmaschinen angemeldet haben, können Sie hier die Bestätigungscodes eingeben.

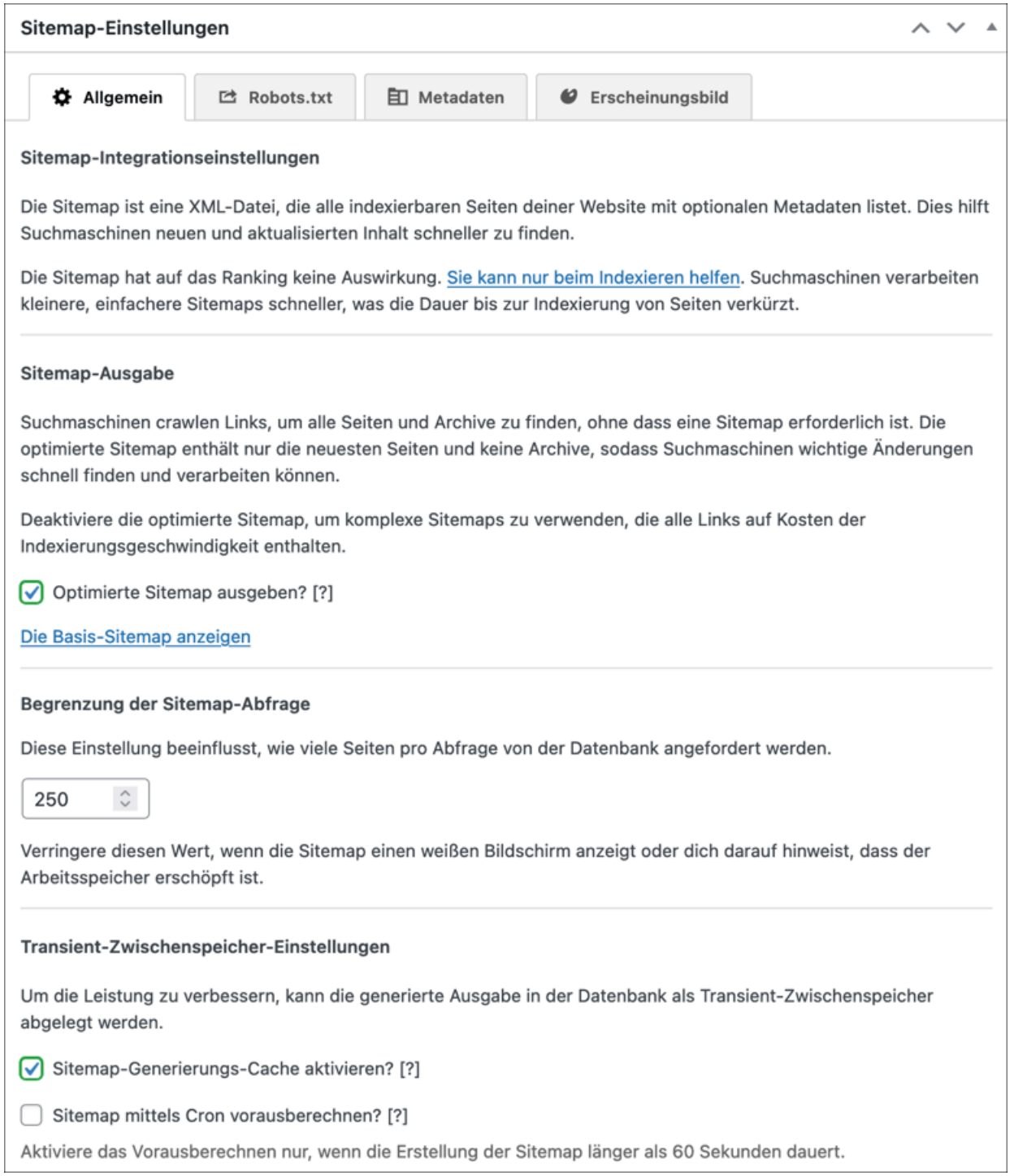

Sitemap-Einstellungen
Allgemein
Robots.txt
Metadaten
Erscheinungsbild
Sitemap-Integrationseinstellungen
Die Sitemap ist eine XML-Datei, die alle indexierbaren Seiten deiner Website mit optionalen Metadaten listet. Dies hilft Suchmaschinen neuen und aktualisierten Inhalt schneller zu finden.
Die Sitemap hat auf das Ranking keine Auswirkung. Sie kann nur beim Indexieren helfen. Suchmaschinen verarbeiten kleinere, einfachere Sitemaps schneller, was die Dauer bis zur Indexierung von Seiten verkürzt.
Sitemap-Ausgabe
Suchmaschinen crawlen Links, um alle Seiten und Archive zu finden, ohne dass eine Sitemap erforderlich ist. Die optimierte Sitemap enthält nur die neuesten Seiten und keine Archive, sodass Suchmaschinen wichtige Änderungen schnell finden und verarbeiten können.
Deaktiviere die optimierte Sitemap, um komplexe Sitemaps zu verwenden, die alle Links auf Kosten der Indexierungsgeschwindigkeit enthalten.
Optimierte Sitemap ausgeben? [?]
Die Basis-Sitemap anzeigen
Begrenzung der Sitemap-Abfrage
Diese Einstellung beeinflusst, wie viele Seiten pro Abfrage von der Datenbank angefordert werden.
250
Verringere diesen Wert, wenn die Sitemap einen weißen Bildschirm anzeigt oder dich darauf hinweist, dass der Arbeitsspeicher erschöpft ist.
Transient-Zwischenspeicher-Einstellungen
Um die Leistung zu verbessern, kann die generierte Ausgabe in der Datenbank als Transient-Zwischenspeicher abgelegt werden.
Sitemap-Generierungs-Cache aktivieren? [?]
Sitemap mittels Cron vorausberechnen? [?]
Aktiviere das Vorausberechnen nur, wenn die Erstellung der Sitemap länger als 60 Sekunden dauert.

Abbildung 20.27 Hier können Sie allgemeine Einstellungen für die Sitemap vornehmen.

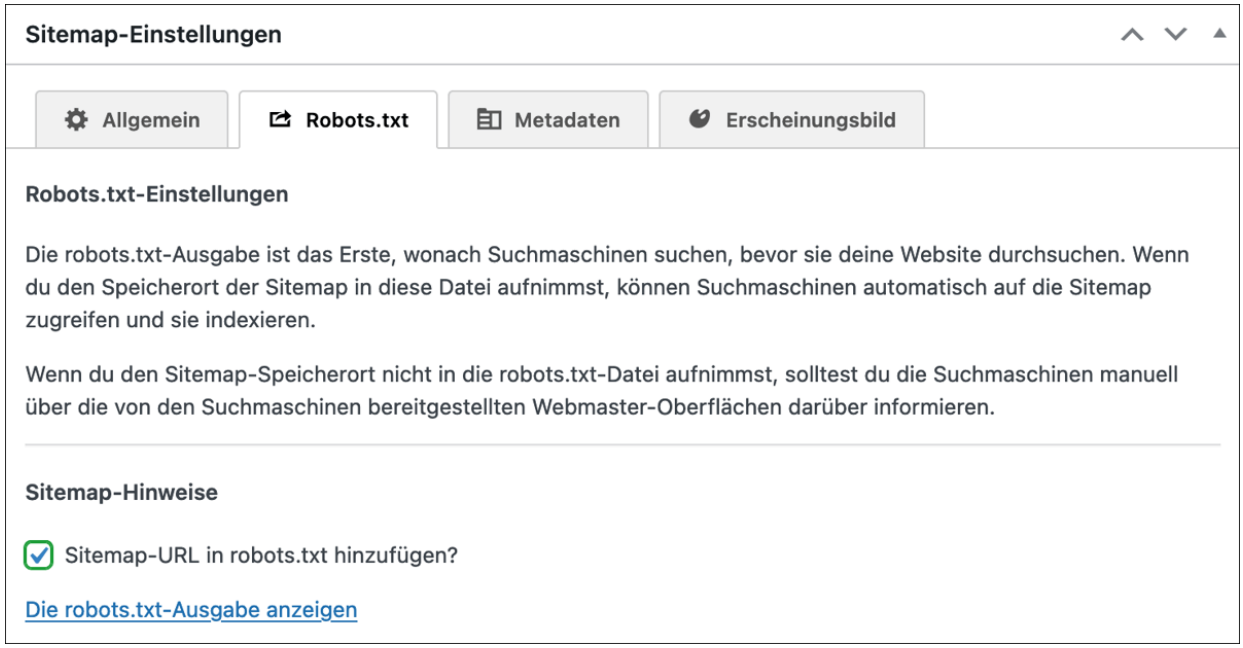


Abbildung 20.28 Die Sitemap kann in der robots.txt verlinkt werden.



Abbildung 20.29 Sie können optional den Zeitpunkt der letzten Aktualisierung eines Inhalts in die Sitemap aufnehmen.

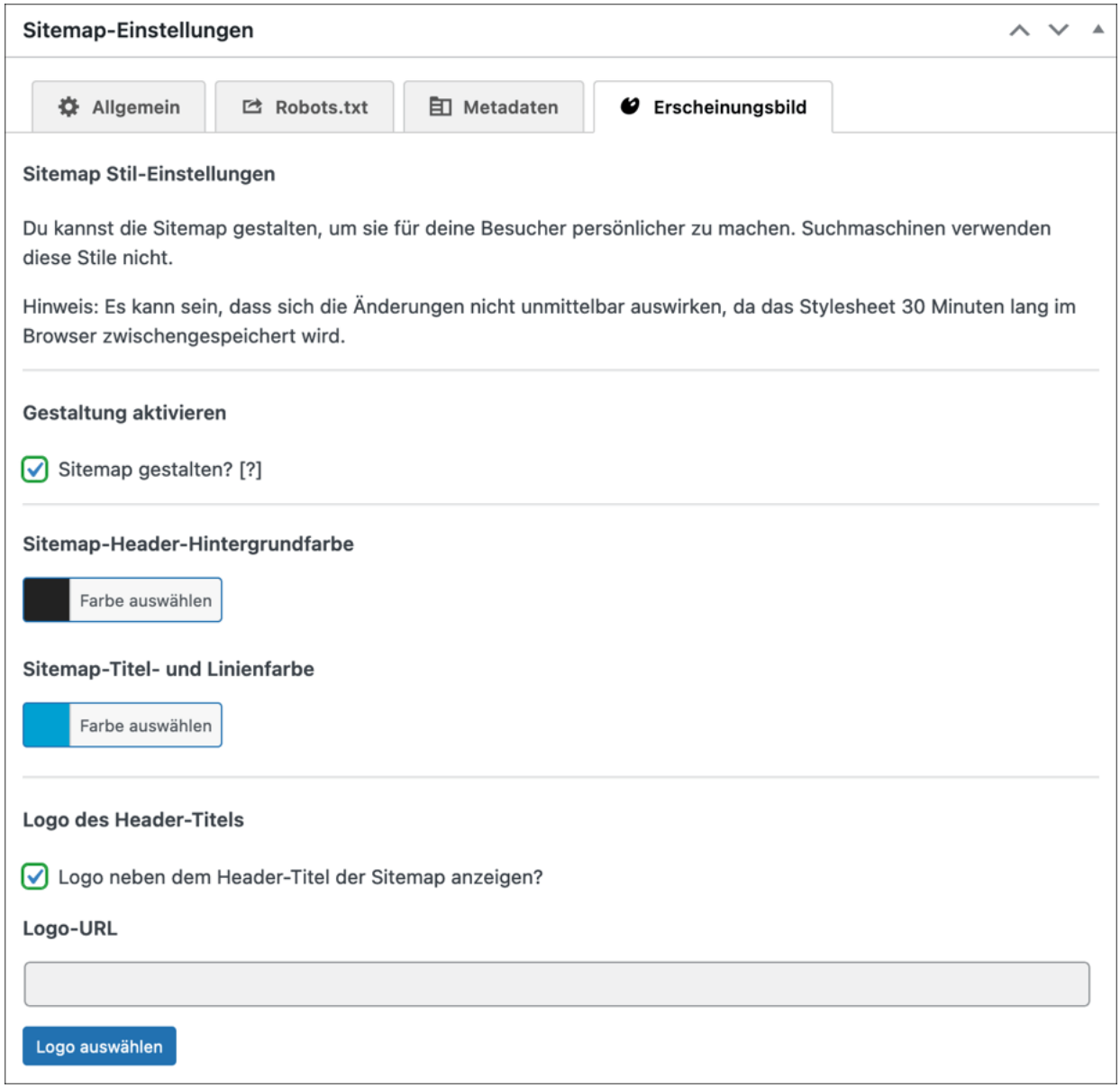


Abbildung 20.30 Hier können Sie die Sitemap ein bisschen stylen, falls sich mal menschliche Besucher*innen darauf verirren sollten.

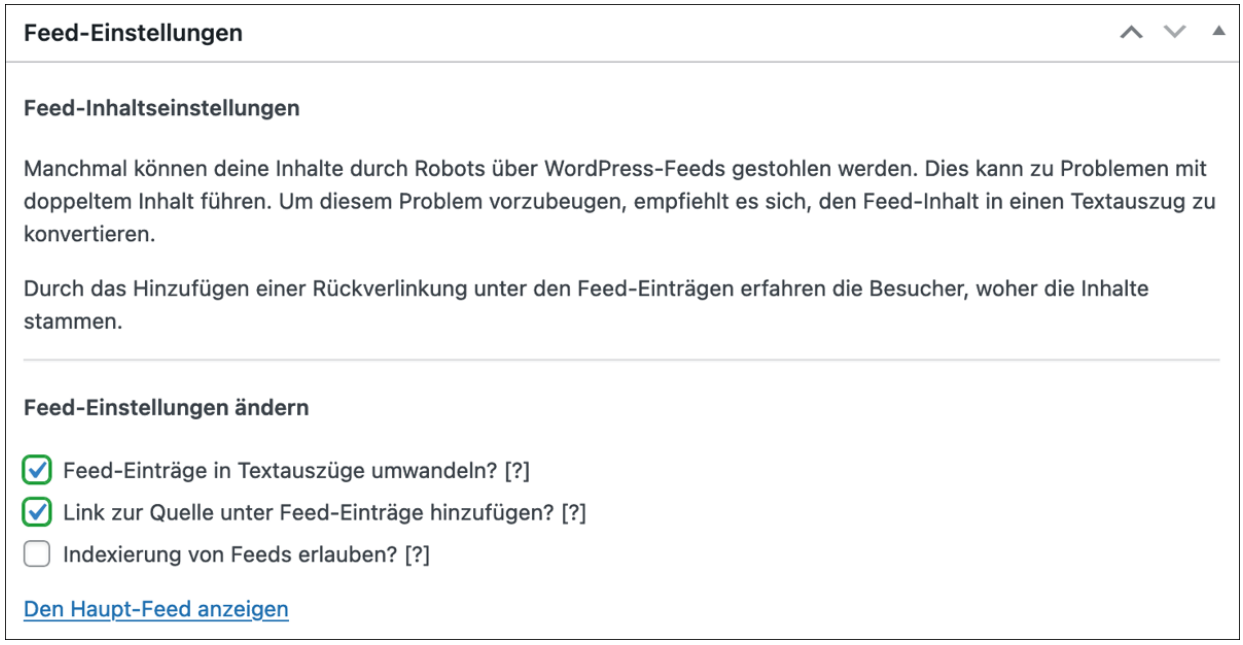


Abbildung 20.31 In den Feed-Einstellungen haben Sie Optionen für den Inhalt der RSS-Feeds, die WordPress für Ihre Website generiert.

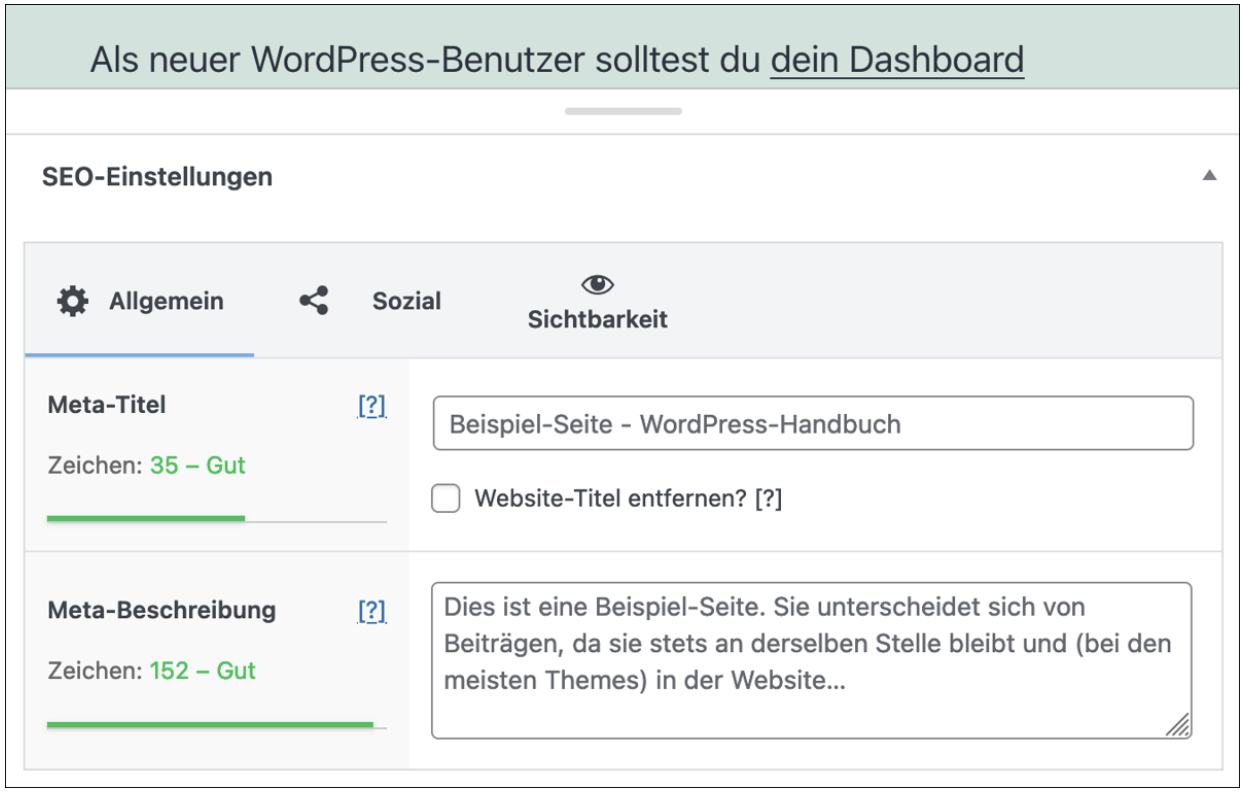


Abbildung 20.32 Unter dem Inhaltsbereich im Editor können Sie verschiedene SEO-Einstellungen auf Beitrags- beziehungsweise Seitenebene anpassen.

Allgemein - Yoast SEO

Dashboard | **Erstmalige Konfiguration**

Erzähl' uns was über deine Website, damit wir deine Seite ranken können!

Lass uns deine Website für die Suchmaschinen in Topform bringen. Befolge einfach diese 5 Schritte, damit Google versteht, worum es auf deiner Website geht.

OPTIMIERUNG DER SEO-DATEN

Beginnen wir mit der SEO-Datenoptimierung. Das heißt, wir scannen deine Website und erstellen eine Datenbank mit optimierten SEO-Daten. Dabei werden keine Inhalte oder Einstellungen auf deiner Website verändert und du musst nichts tun, sondern nur auf Start klicken!



Hinweis: Wenn du sehr viele Inhalte hast, kann diese Optimierung einen Moment dauern. Aber vertrau uns, das ist es wert!

Optimierung der SEO-Daten beginnen

Diese Funktion enthält und ersetzt den Text-Linkzähler und die interne Verlinkungsanalyse

Fortsetzen

DARSTELLUNG DER WEBSITE

SOCIAL-MEDIA-PROFILE

PERSÖNLICHE EINSTELLUNGEN

KONFIGURATION ABSCHLIESSEN

Abbildung 20.33 Wenn Sie Yoast SEO aktivieren, wird ein Konfigurationsassistent gestartet.

DARSTELLUNG DER WEBSITE

Hilf uns hier bitte! Ist deine Website über ein Unternehmen oder eine Einzelperson? Wenn du diesen Schritt ausführst, kann Google deine Website noch besser verstehen. Bonus: Du erhöhst deine Chancen darauf, gute Ergebnisse zu erzielen!

Repräsentiert deine Website eine Firma oder Person?

Organisation

Name der Website

WordPress-Handbuch

Organisations-Name

WordPress-Handbuch

Organisations-Logo



Bild auswählen

Speichern und fortfahren

Zurück

Abbildung 20.34 Im zweiten Schritt des Assistenten geben Sie Informationen zur Website an.

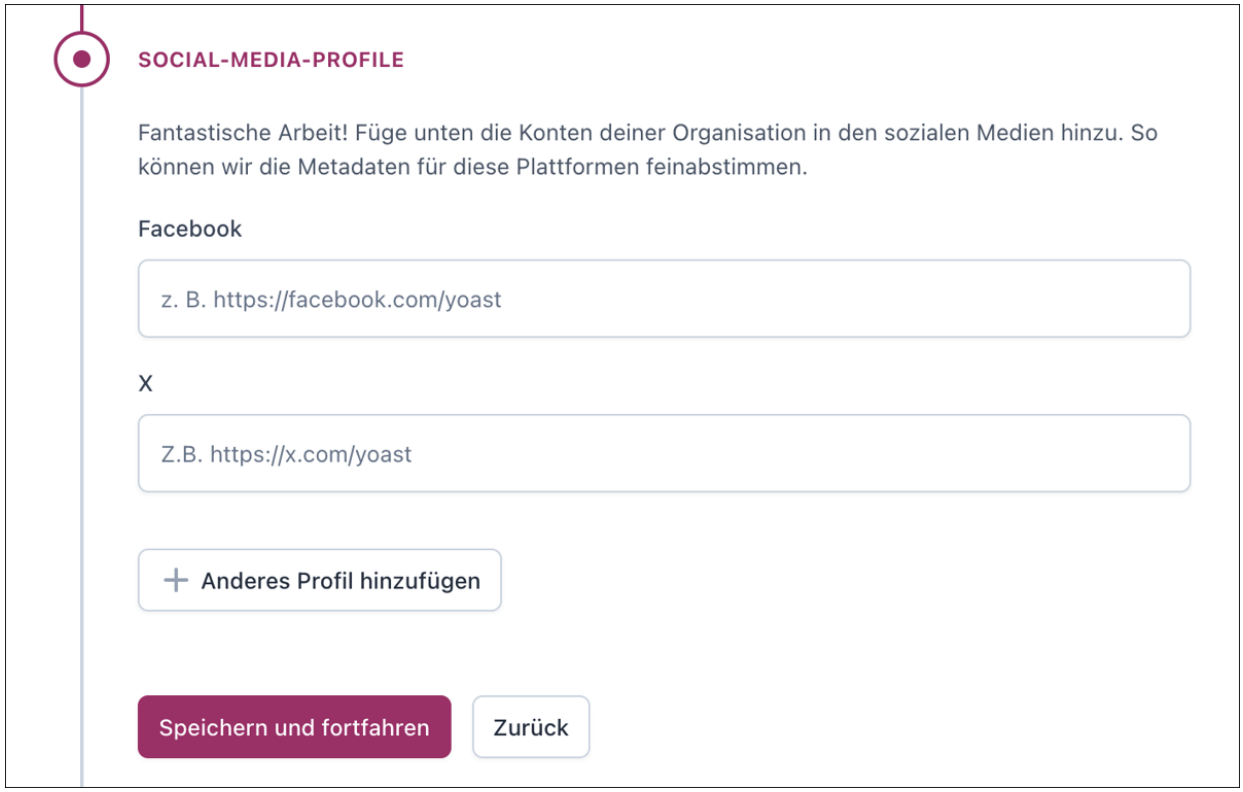


Abbildung 20.35 Im Einführungsdialog können Sie Social-Media-Profile angeben.

Website-Funktionen

Steuere, welche Funktionen du verwenden möchtest.

Schreiben



SEO-Analyse

Die SEO-Analyse bietet Vorschläge zur Verbesserung der Auffindbarkeit deiner Texte und stellt sicher, dass deine Inhalte den Best Practices entsprechen.

Weitere Informationen →

Funktion aktivieren



Lesbarkeits-Analyse

Die Lesbarkeitsanalyse bietet Vorschläge zur Verbesserung der Struktur und des Stils deines Textes.

Weitere Informationen →

Funktion aktivieren



Inklusive Sprachanalyse

Die Analyse der inklusiven Sprache bietet Vorschläge, um inklusivere Texte zu schreiben, damit mehr Menschen deine Inhalte verstehen können.

Weitere Informationen →

Funktion aktivieren



Yoast AI

Die KI-Funktionen helfen dir, bessere Inhalte zu erstellen, indem sie dir Optimierungsvorschläge machen, die du nach Belieben anwenden kannst.

Weitere Informationen →

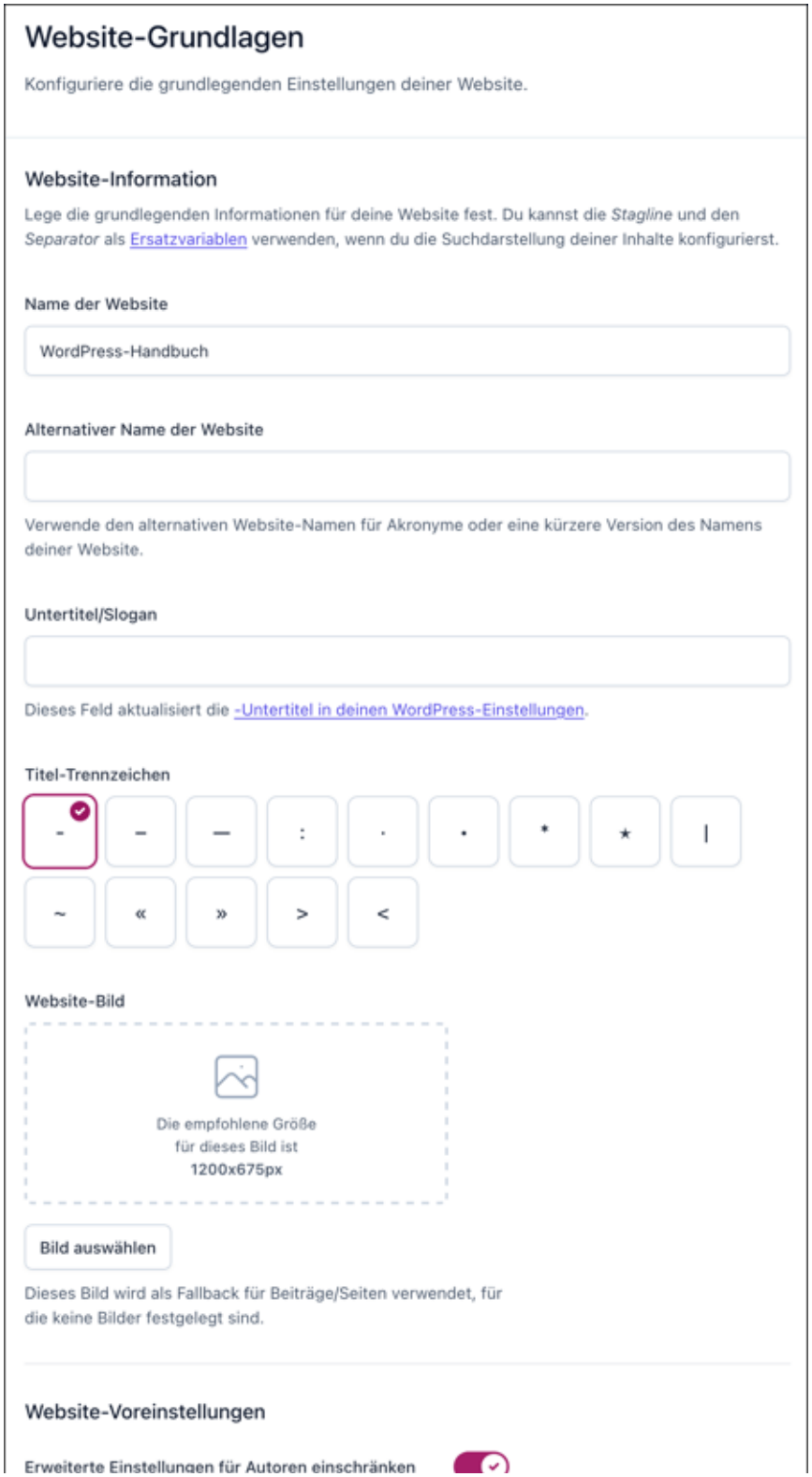
Entsperren mit Premium



Einblicke



Abbildung 20.36 Hier können Sie unterschiedlichste Funktionen von Yoast aktivieren oder deaktivieren.


Website-Grundlagen
Konfiguriere die grundlegenden Einstellungen deiner Website.
Website-Information
Lege die grundlegenden Informationen für deine Website fest. Du kannst die Stagline und den Separator als Ersatzvariablen verwenden, wenn du die Suchdarstellung deiner Inhalte konfigurierst.
Name der Website
WordPress-Handbuch
Alternativer Name der Website
Verwende den alternativen Website-Namen für Akronyme oder eine kürzere Version des Namens deiner Website.
Untertitel/Slogan
Dieses Feld aktualisiert die -Untertitel in deinen WordPress-Einstellungen.
Titel-Trennzeichen
- – — : · • * ★ |
~ « » > <
Website-Bild
Die empfohlene Größe
für dieses Bild ist
1200x675px
Bild auswählen
Dieses Bild wird als Fallback für Beiträge/Seiten verwendet, für die keine Bilder festgelegt sind.
Website-Voreinstellungen
Erweiterte Einstellungen für Autoren einschränken

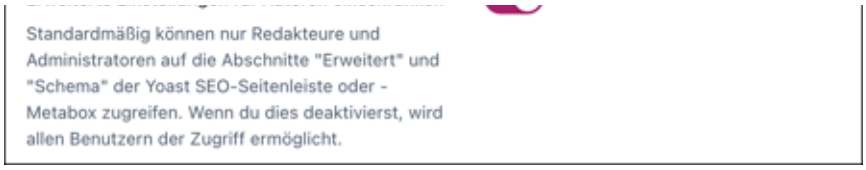


Abbildung 20.37 Auf der Einstellungsseite »Website-Grundlagen« können Sie unter anderem das Trennzeichen für den Titel ändern.

Website-Verbindungen

Verifiziere deine Website mit verschiedenen Werkzeugen. Dies wird deiner Homepage einen Meta-Tag zur Verifizierung hinzufügen. Folge dem Link in der Beschreibung, um zu einer Anleitung zu kommen, wie du deine Website für jede Plattform überprüfst.

Baidu

Verifizierungscode hinzufügen

Hole dir deinen Baidu-Verifizierungscode in den Baidu Webmaster Tools.

Bing

Verifizierungscode hinzufügen

Erhalte deinen Verifizierungscode in den Bing Webmaster Tools.

Google

Verifizierungscode hinzufügen

Hole dir deinen Google-Verifizierungscode in der Google Search Console.

Pinterest

Verifizierungscode hinzufügen

Bestätige deine Website bei Pinterest.

Yandex

Verifizierungscode hinzufügen

Hole dir deinen Yandex-Verifizierungscode in den Yandex Webmaster Tools.

Abbildung 20.38 Unter »Website-Verbindungen« können Sie Codes für die Webmaster-Tools unterschiedlicher Suchmaschinen eintragen.

Homepage

Wähle aus wie deine Startseite in Suchmaschinen und in sozialen Netzwerken dargestellt werden soll. Beachte, dass Nutzer diese Darstellung wahrscheinlich sehen werden, wenn sie nach deiner Marke suchen.

Darstellung in der Suche

Wähle, wie deine Homepage in den Suchergebnissen aussehen soll.

SEO-Titel

Variable eingeben

Titel der Website | Seite | Trennzeichen | Untertitel/Slogan

Meta-Beschreibung

Variable eingeben

Social-Media-Auftritt

Wählen, wie deine Homepage in den sozialen Medien aussehen soll.

Bild für Social-Media



Die empfohlene Größe
für dieses Bild ist
1200x675px

Bild auswählen

Titel für Social-Media

Abbildung 20.39 Hier nehmen Sie Einstellungen für die Startseite vor.

Beiträge

Wähle, wie deine beiträge in den Ergebnissen von Suchmaschinen und auf Social Media aussehen sollen.

Darstellung in der Suche

Lege fest, wie deine beiträge standardmäßig in den Suchergebnissen aussehen sollen. Du kannst die Einstellungen für einzelne beiträge jederzeit in der Seitenleiste oder Metabox von Yoast SEO anpassen.

Zeige beiträge in den Suchergebnissen



Deaktivieren bedeutet, dass beiträge nicht von Suchmaschinen indiziert wird und von XML-Sitemaps ausgeschlossen wird. Erfahre mehr über die Einstellungen der Suchergebnisse.

SEO-Titel

Variable eingeben

Titel | Seite | Trennzeichen | Titel der Website

Meta-Beschreibung

Variable eingeben

Social-Media-Auftritt

Lefe fest, wie deine beiträge standardmäßig in sozialen Medien aussehen sollen. Du kannst die Einstellungen für einzelne beiträge jederzeit in der Seitenleiste oder Metabox von Yoast SEO anpassen.

Bild für Social-Media



Abbildung 20.40 Hier können Sie Einstellungen für die Darstellung der Beiträge in den Suchergebnissen vornehmen.

Kategorien

Wähle, wie deine kategorien in den Ergebnissen von Suchmaschinen und auf Social Media aussehen sollen.
Diese Taxonomie wird für `Beiträge` verwendet.

Darstellung in der Suche

Lege fest, wie deine kategorien standardmäßig in den Suchergebnissen aussehen sollen. Du kannst die Einstellungen für einzelne kategorien jederzeit in der Yoast SEO-Metabox anpassen.

Zeige kategorien in den Suchergebnissen



Wenn du dies deaktivierst, werden Archivseiten für kategorien nicht von Suchmaschinen indiziert und von XML-Sitemaps ausgeschlossen. Erfahre mehr über die Einstellungen der Suchergebnisse.

SEO-Titel

Variable eingeben

Begriffstitel Archive Seite Trennzeichen Titel der Website

Meta-Beschreibung

Variable eingeben

Social-Media-Auftritt

Lege fest, wie deine kategorien standardmäßig in sozialen Medien aussehen sollen. Du kannst die Einstellungen für einzelne kategorien jederzeit in der Yoast SEO-Metabox anpassen.

Bild für Social-Media



Abbildung 20.41 Auf der Einstellungsseite für Kategorien steuern Sie die Darstellung von Kategorien in Suchergebnissen.

Crawl-Optimierung

Mach deine Website effizienter und umweltfreundlicher, indem du verhinderst, dass Suchmaschinen Dinge crawlen, die sie nicht brauchen, und indem du nicht genutzte WordPress-Funktionen entfernst. Erfahre mehr über Crawl-Einstellungen und wie sie deine Seite verbessern können.

Unerwünschte Metadaten entfernen

WordPress fügt eine Menge Links und Inhalte in die `<head>` und HTTP-Header deiner Website ein. In der Regel kannst du diese sicher deaktivieren, was helfen kann, Bytes, Strom und Bäume einzusparen.

Shortlinks entfernen

Entferne Links zu den WordPress-internen "Shortlink"-URLs für deine Beiträge. z. B.

```
<link rel="shortlink" href="https://www.example.com/?p=1" />
```

REST-API-Links entfernen

Entfernt Links zu den REST-API-Endpunkten deiner Website. z. B.

```
<link rel="https://api.w.org/" href="https://www.example.com/wp-json/" />
```

RSD/WLW-Links entfernen

Links entfernen, die von externen Systemen für die Veröffentlichung von Inhalten in deinem Blog verwendet werden. z. B.

```
<link rel="EditURI" type="application/rsd+xml" title="RSD" href="https://v
```

und

```
<link rel="wlwmanifest" type="application/wlwmanifest+xml" href="https://v
```

oEmbed-Links entfernen

Abbildung 20.42 Hier können Sie Funktionen deaktivieren, die Sie nicht nutzen.

Breadcrumbs

Konfiguriere das Aussehen und Verhalten deiner Breadcrumbs.

Aussehen der Breadcrumbs

Wähle das allgemeine Aussehen deiner Breadcrumbs.

Trenner zwischen den Breadcrumbs

»

Anker-Text für die Homepage

Startseite

Praefix für den Breadcrumb-Pfad.

Präfix hinzufügen

Präfix für Archiv-Breadcrumbs

Archive für

Präfix für Suchseiten-Breadcrumbs

Du suchtest nach

Breadcrumb für Fehlerseite 404

Abbildung 20.43 Breadcrumbs helfen bei der Navigation innerhalb einer Website-Struktur und können Suchmaschinen bei der Erkennung der Struktur unterstützen.

Autor-Archiv

Wenn du einen Blog mit nur einem Autor betreibst, ist das autorenarchiv (z.B. `https://wp6.uber.space/author/example/`) genau dasselbe wie deine Homepage. Das nennt man ein Problem mit doppeltem Inhalt. Wenn dies bei deiner Website der Fall ist, kannst du es entweder deaktivieren (dann wird es auf die Homepage umgeleitet) oder verhindern, dass es in den Suchergebnissen angezeigt wird.

autor-archiv aktivieren

Wenn du dies deaktivierst, wird autorenarchiv auf die Startseite deiner Webseite umgeleitet.

Darstellung in der Suche

Wähle, wie deine autor-archiv in Suchmaschinen aussehen sollen.

Zeige autor-archiv in den Suchergebnissen

Deaktivieren bedeutet, dass autor-archiv nicht von Suchmaschinen indiziert wird und von XML-Sitemaps ausgeschlossen wird. Erfahre mehr über die Einstellungen der Suchergebnisse.

autor-archiv ohne Beiträge in den Suchergebnissen anzeigen

Wenn du das deaktivierst, werden autor-archiv ohne Beiträge nicht von Suchmaschinen indiziert und von XML-Sitemaps ausgeschlossen.

SEO-Titel

Variable eingeben

Name , Autor bei Titel der Website Seite

Meta-Beschreibung



Abbildung 20.44 Hier können Sie Einstellungen für die Autor-Archive vornehmen.

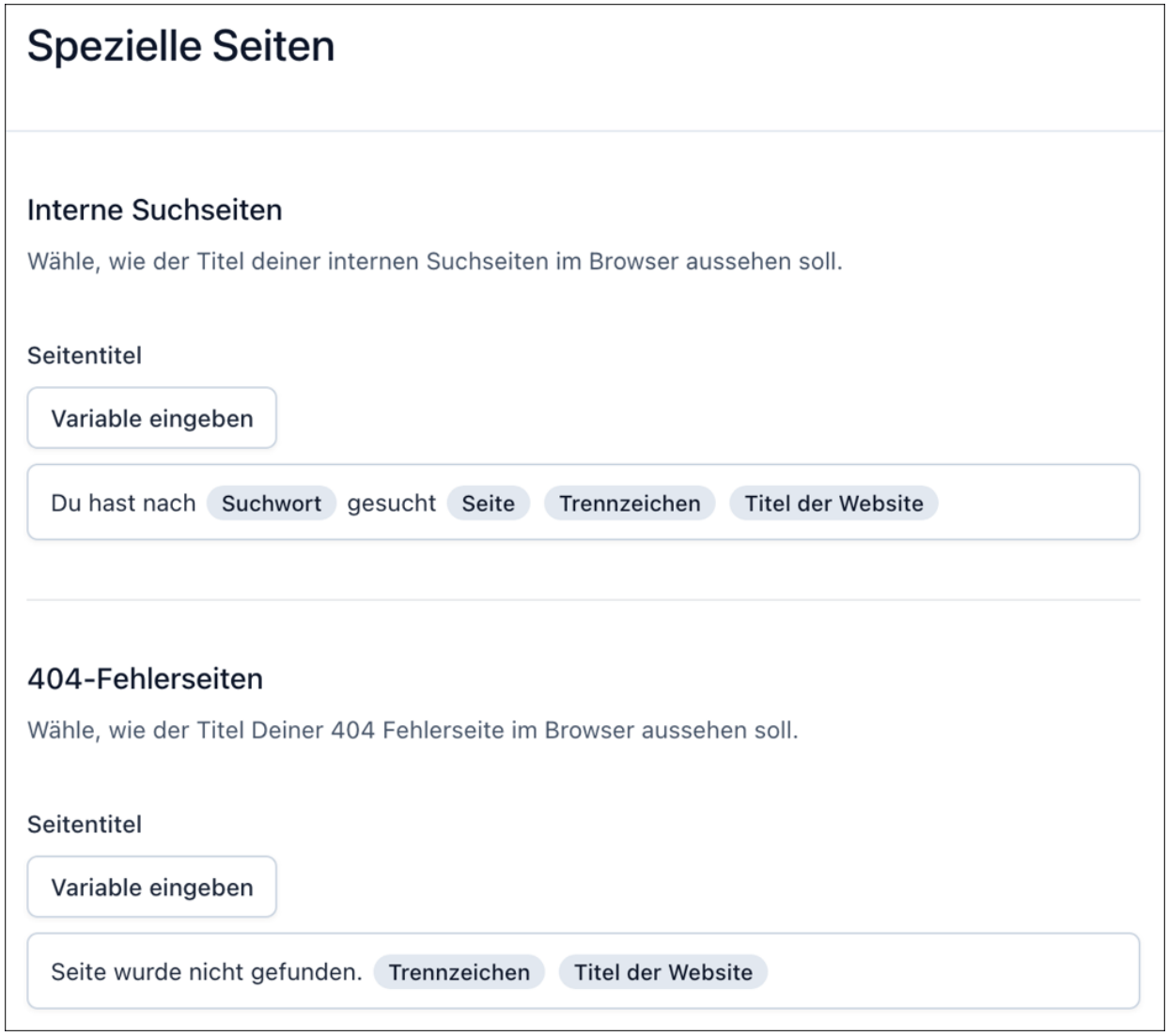


Abbildung 20.45 Hier können Sie Seitentitel für die Suchergebnisseiten und die 404-Fehlerseite vergeben.

Medien Seiten

Wenn du Medien hochlädst (z.B. ein Bild oder Video), erstellt WordPress automatisch eine medien Seite (Anhang-URL) dafür. Diese Seiten sind ziemlich leer und könnten Probleme wegen zu wenig Content verursachen und zu überflüssigen Seiten auf deiner Website führen. Deshalb deaktiviert Yoast SEO sie standardmäßig (und leitet die Anhang-URL auf das Medium selbst um).

Aktiviere medien Seiten

Wir empfehlen, medien-Seiten deaktiviert zu lassen. Dadurch werden alle Anhangs-URLs auf das Medium selbst umgeleitet.

Darstellung in der Suche

Wähle, wie deine medien-Seiten in Suchmaschinen aussehen sollen. Du kannst die Einstellungen für einzelne medien-Seiten jederzeit in der Yoast SEO-Metabox anpassen.

medien Seiten in den Suchergebnissen anzeigen

Wenn du diese Option deaktivierst, werden durch WordPress erstellte medien Seiten nicht von Suchmaschinen indexiert und aus den XML-Sitemaps ausgeschlossen.
Erfahre mehr über die Einstellungen der Suchergebnisse.

SEO-Titel

Variable eingeben

Titel | Seite | Trennzeichen | Titel der Website

Meta-Beschreibung

Variable eingeben

Abbildung 20.46 WordPress generiert für hochgeladene Medien eigene Anhangseiten, für die Sie hier Einstellungen vornehmen können.

RSS

RSS-Feed

Automatisch Content zu deinem RSS hinzufügen. Dies erlaubt es dir, Links zurück zu deinem Blog und zu deinen Beiträgen zu setzen, damit Suchmaschinen sie als die Originalquelle des Contents erkennen.

Inhalt, der vor jeden Eintrag im RSS-Feed gesetzt wird

Inhalt, der an jeden Beitrag im Feed angehängt wird

Der Beitrag %%POSTLINK%% erschien zuerst auf %%BLOGLINK%%.

Verfügbare Variablen

Du kannst folgende Variablen innerhalb des Inhalts verwenden, diese werden durch den Wert auf der rechten Seite ersetzt.

Variable	Beschreibung
%%AUTHORLINK%%	Ein Link zum Archiv des Beitragsautors mit dem Namen des Autors als Ankertext.
%%POSTLINK%%	Link zum Artikel mit dem Artikelnamen als Linktext.

Abbildung 20.47 Hier können Sie Inhalte vor oder nach jedem Beitrag im Feed einfügen.

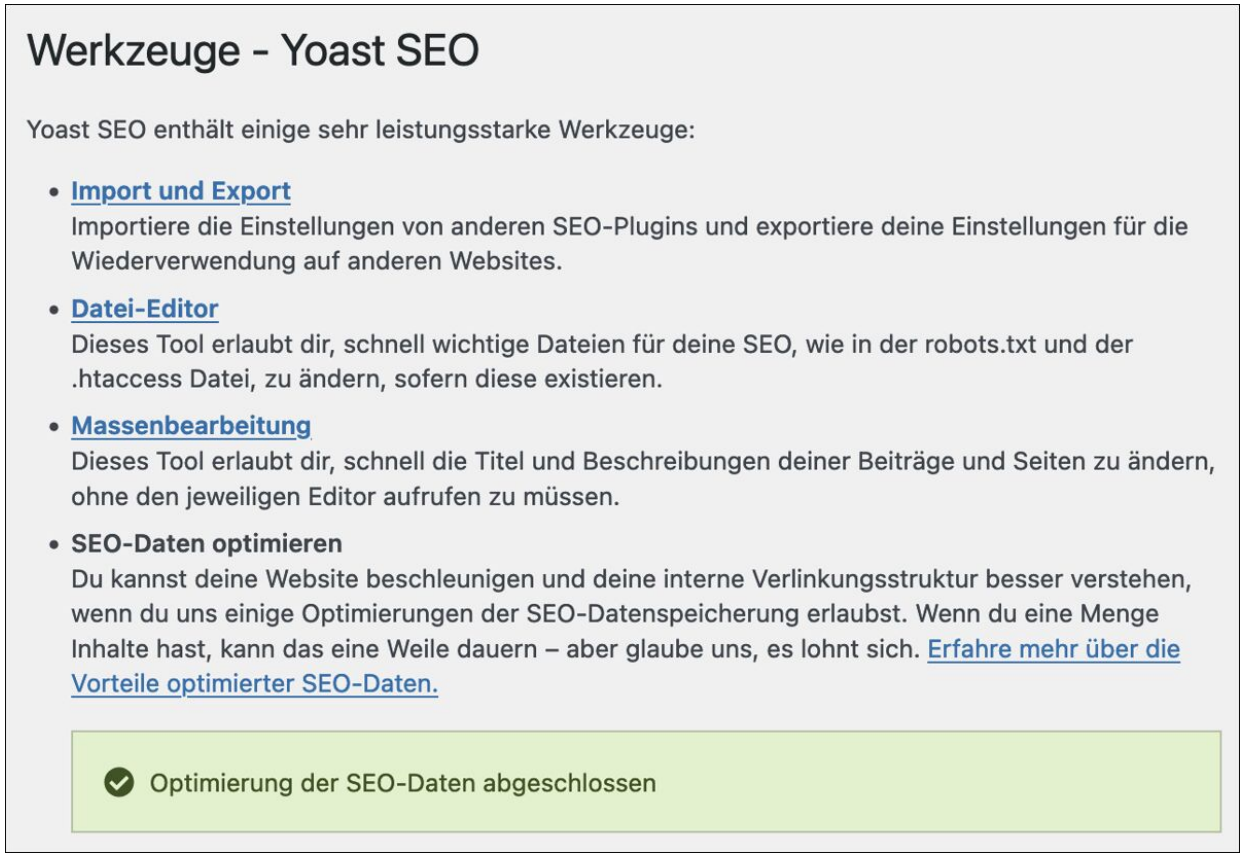


Abbildung 20.48 Yoast bringt ein paar nützliche Werkzeuge mit, beispielsweise zum Import und Export der Einstellungen.

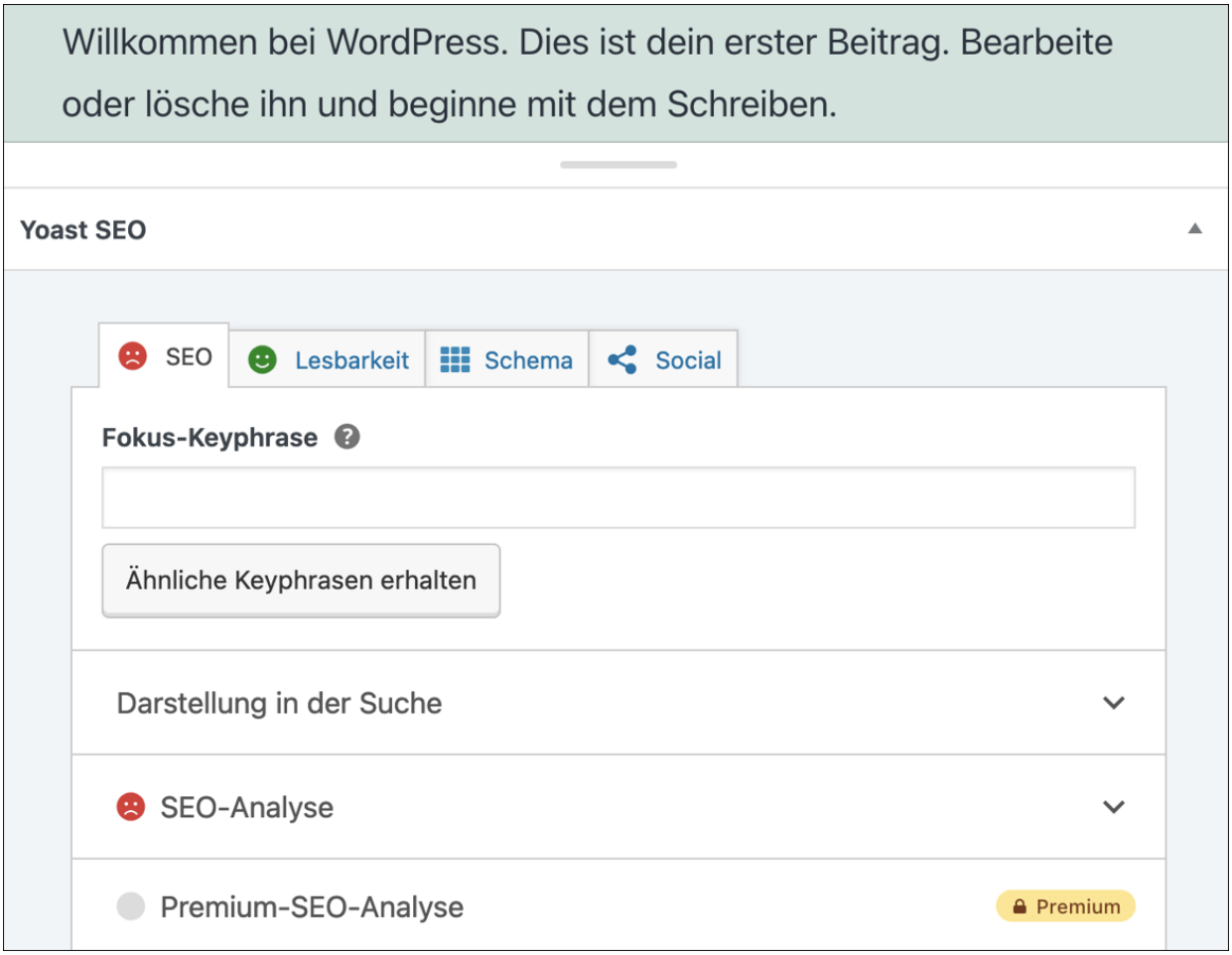


Abbildung 20.49 Unter dem Inhaltseditor finden Sie die Funktionen und Einstellungen von Yoast SEO.

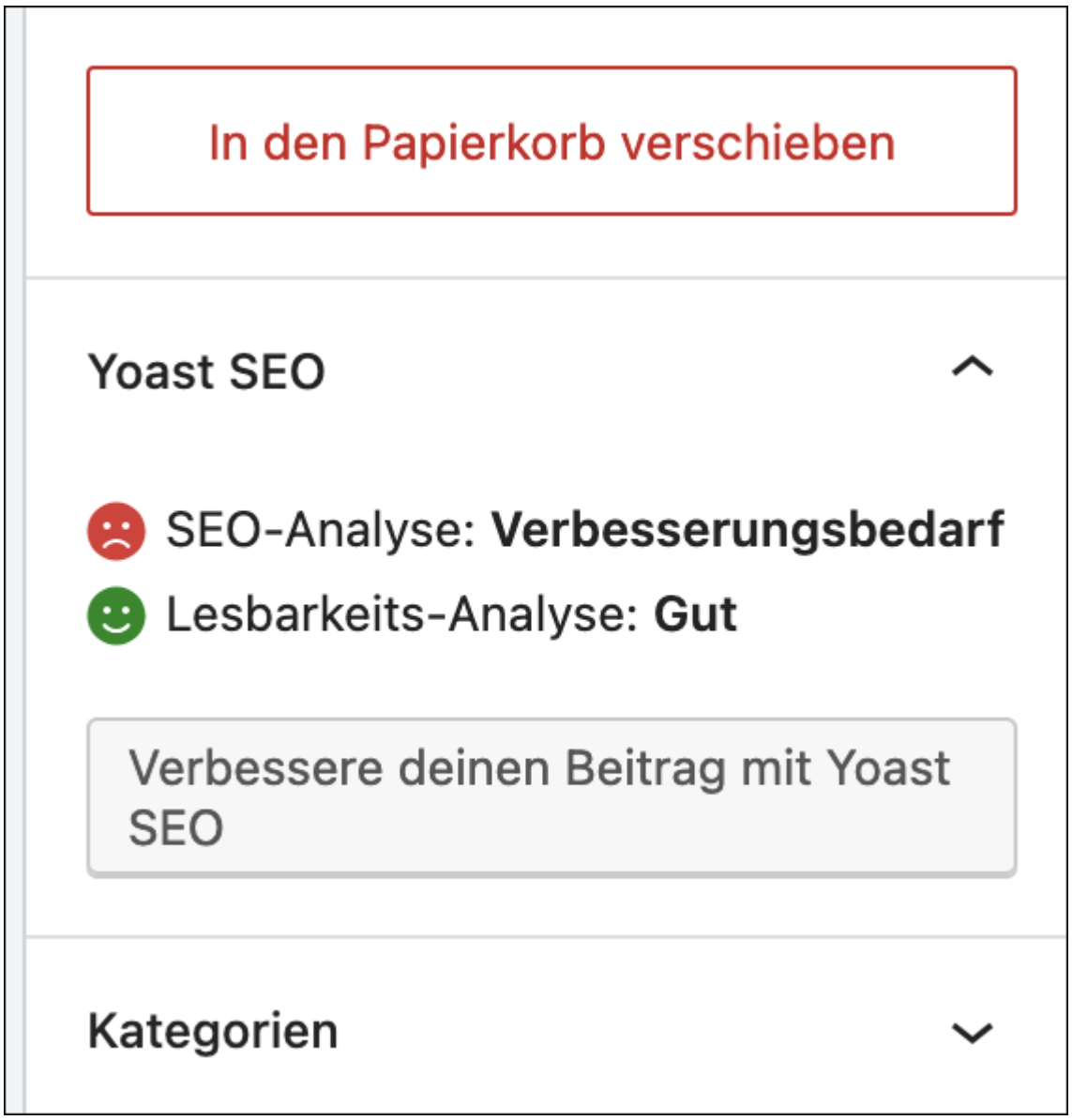


Abbildung 20.50 In der Dokumenten-Seitenleiste sehen Sie den aktuellen Stand von Lesbarkeits- und SEO-Analyse.

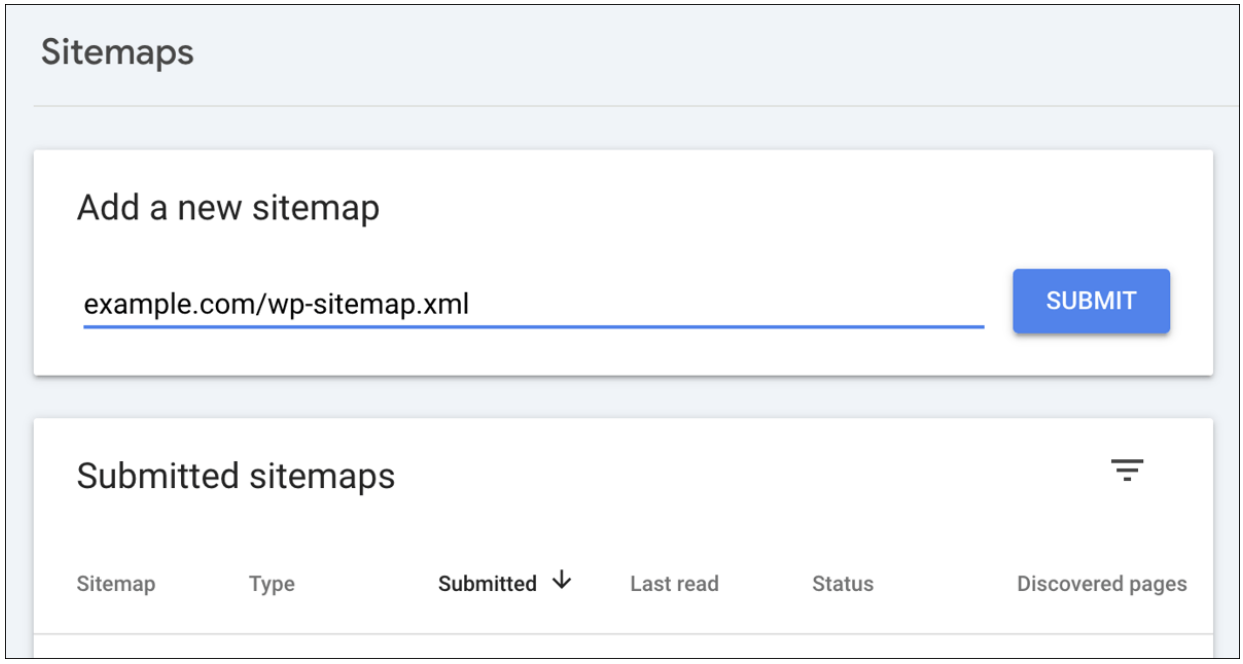


Abbildung 20.51 Die Sitemap(s) für Ihre Website reichen Sie bei Google über die Search Console ein.

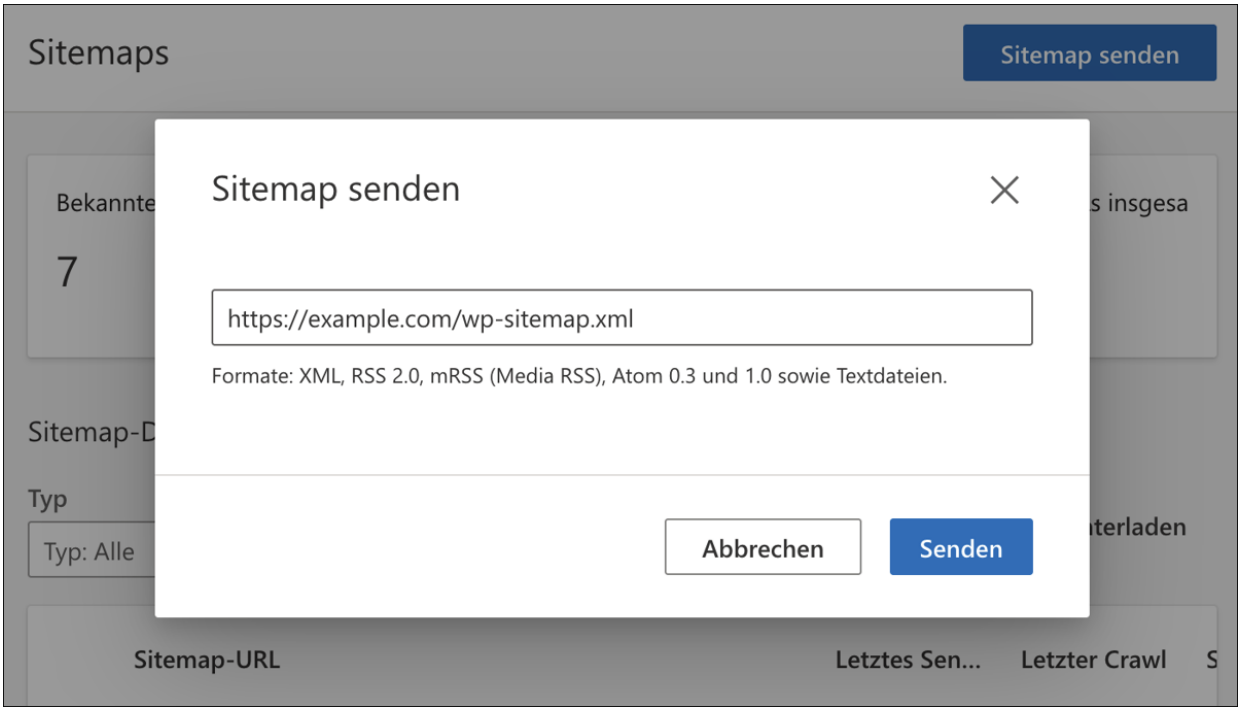


Abbildung 20.52 Hinzufügen einer Sitemap in den Bing Webmaster Tools

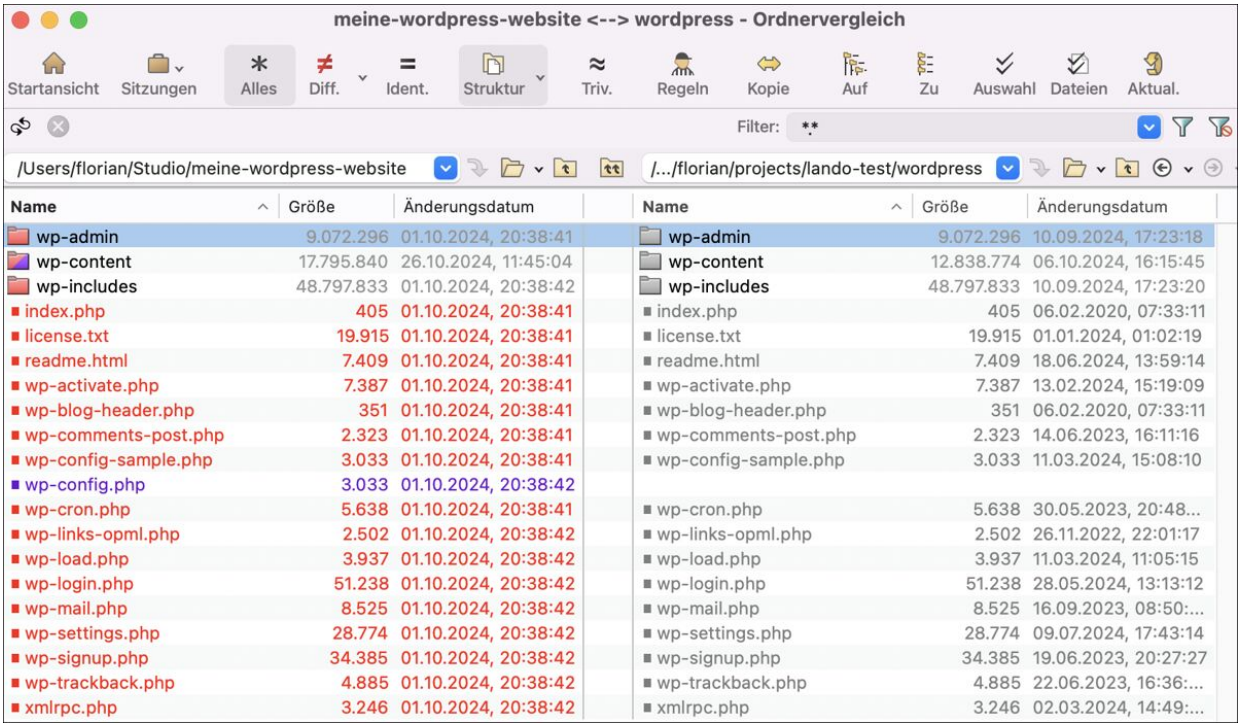


Abbildung 21.1 Beim Vergleich des Ordners Ihrer Installation mit dem Ordner der WordPress- und Plugin-Originaldateien werden Ihnen in entsprechenden Programmen Unterschiede angezeigt, wie hier in BeyondCompare.

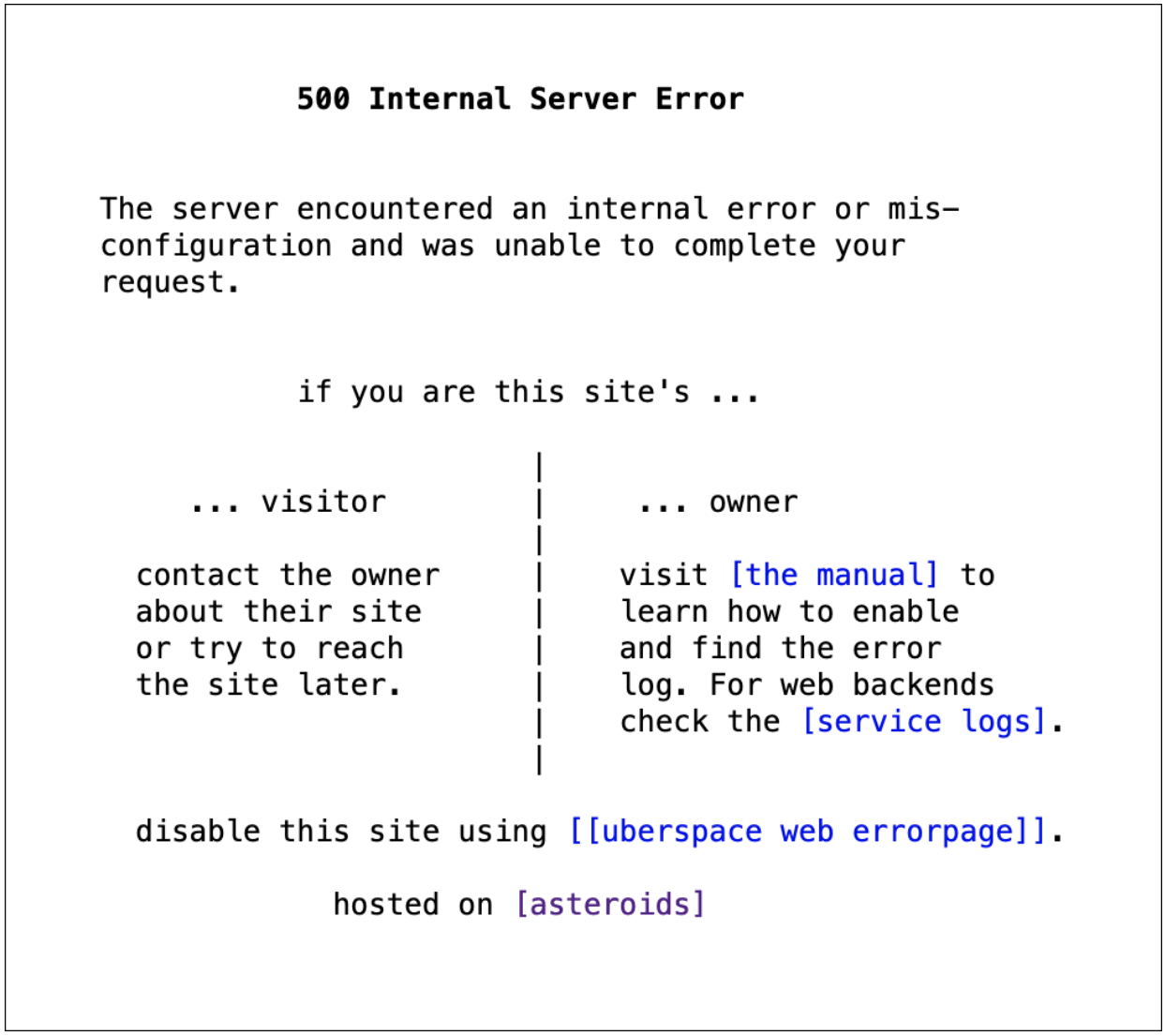


Abbildung 21.2 Wenn bei einer Fehlermeldung der Webserver-Name mit ausgegeben wird (Apache oder nginx), liegt die Ursache nicht bei WordPress, sondern auf Webserver-Ebene.

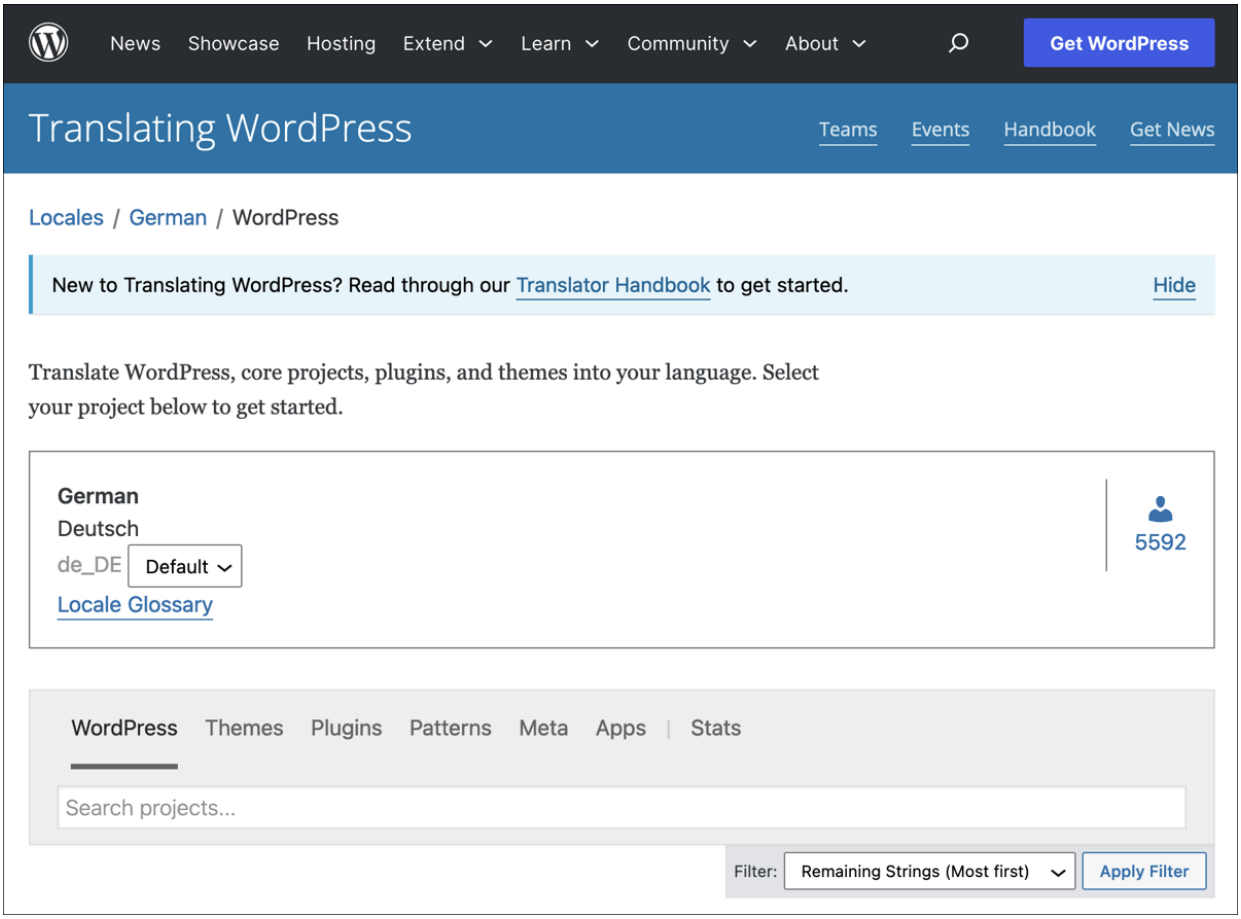


Abbildung 22.1 Auf translate.WordPress.org werden unter anderem die Übersetzungen für alle Plugins und Themes verwaltet, die auf WordPress.org gehostet werden.

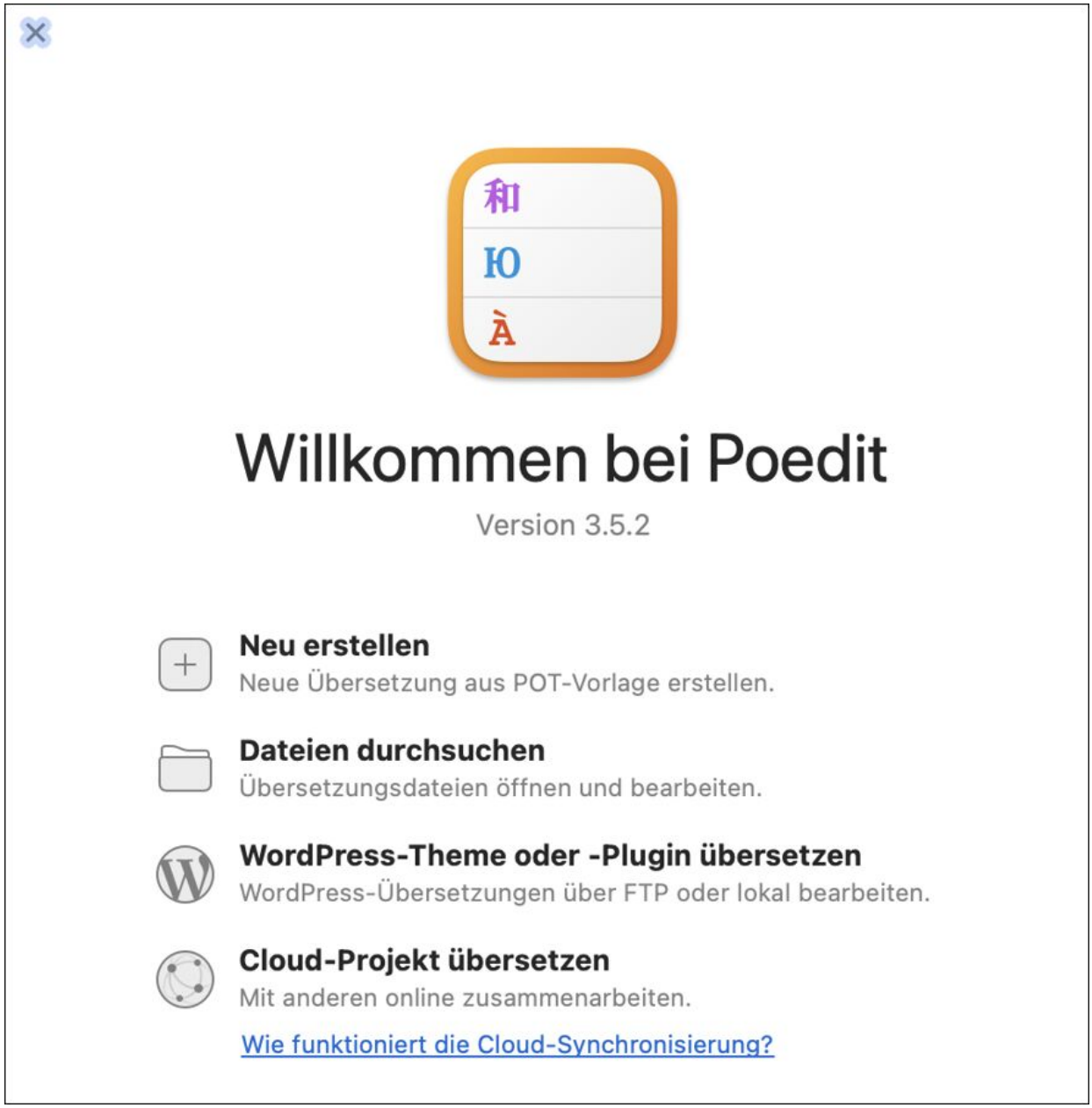


Abbildung 22.2 Poedit ist eine Software zur Übersetzung von Zeichenketten in Programmcode.

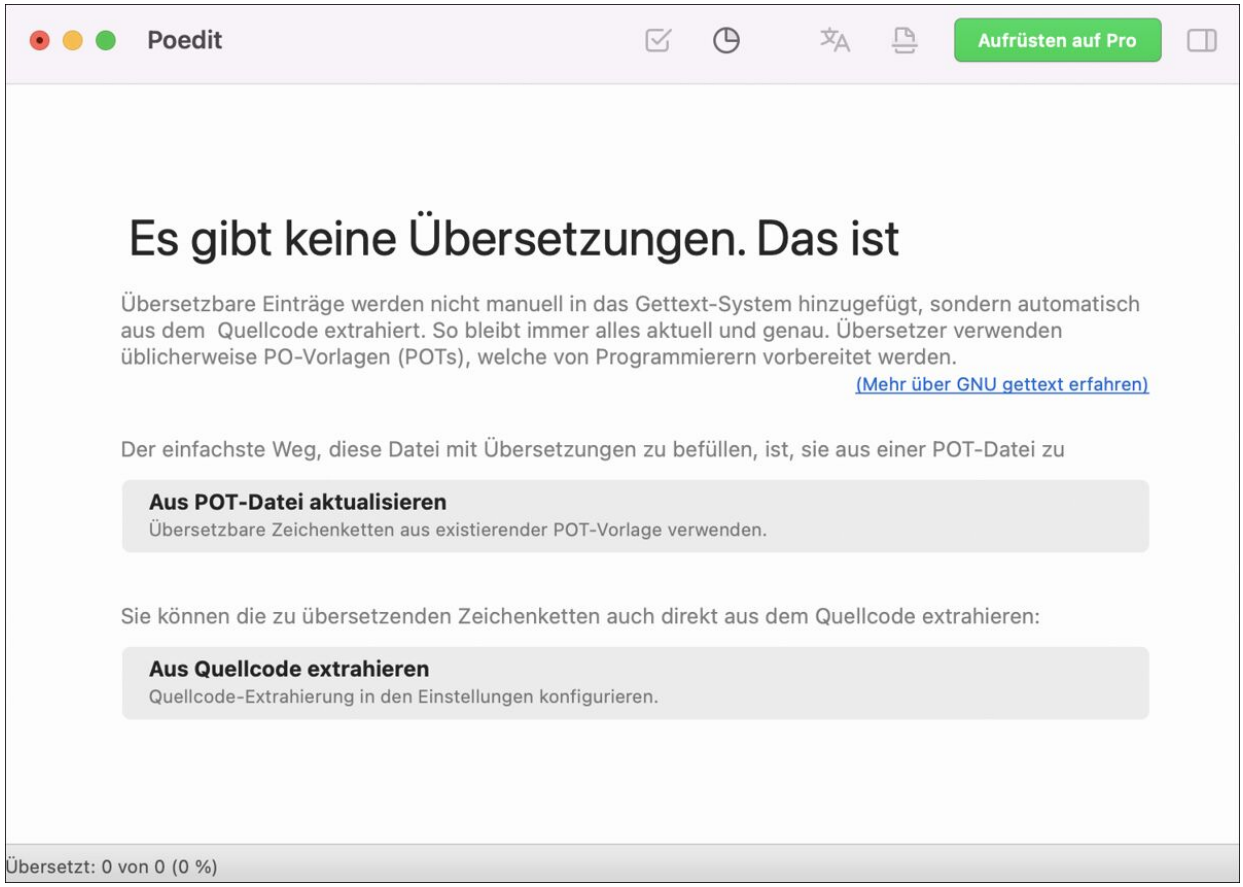


Abbildung 22.3 Poedit-Dialog zum Anlegen einer neuen Übersetzung

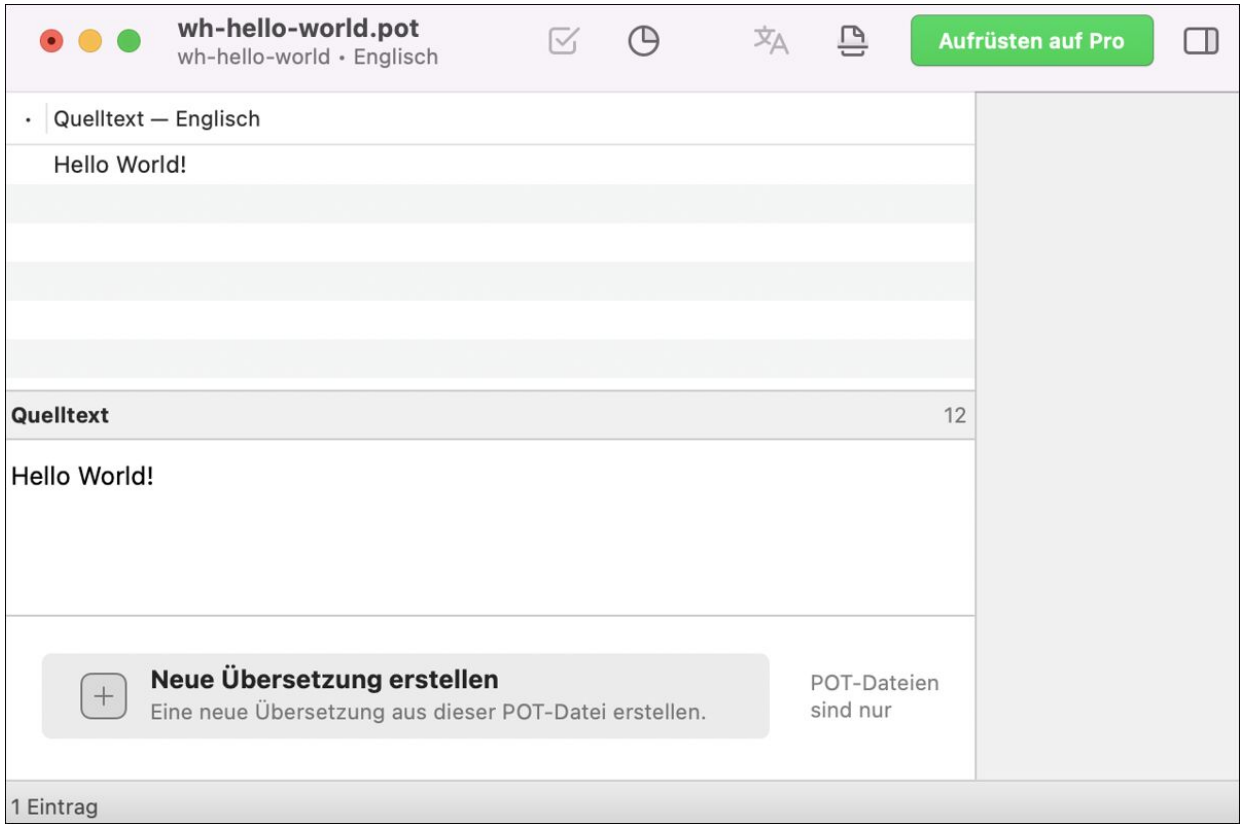


Abbildung 22.4 Das Übersetzungstemplate mit dem »Hello World!«-String

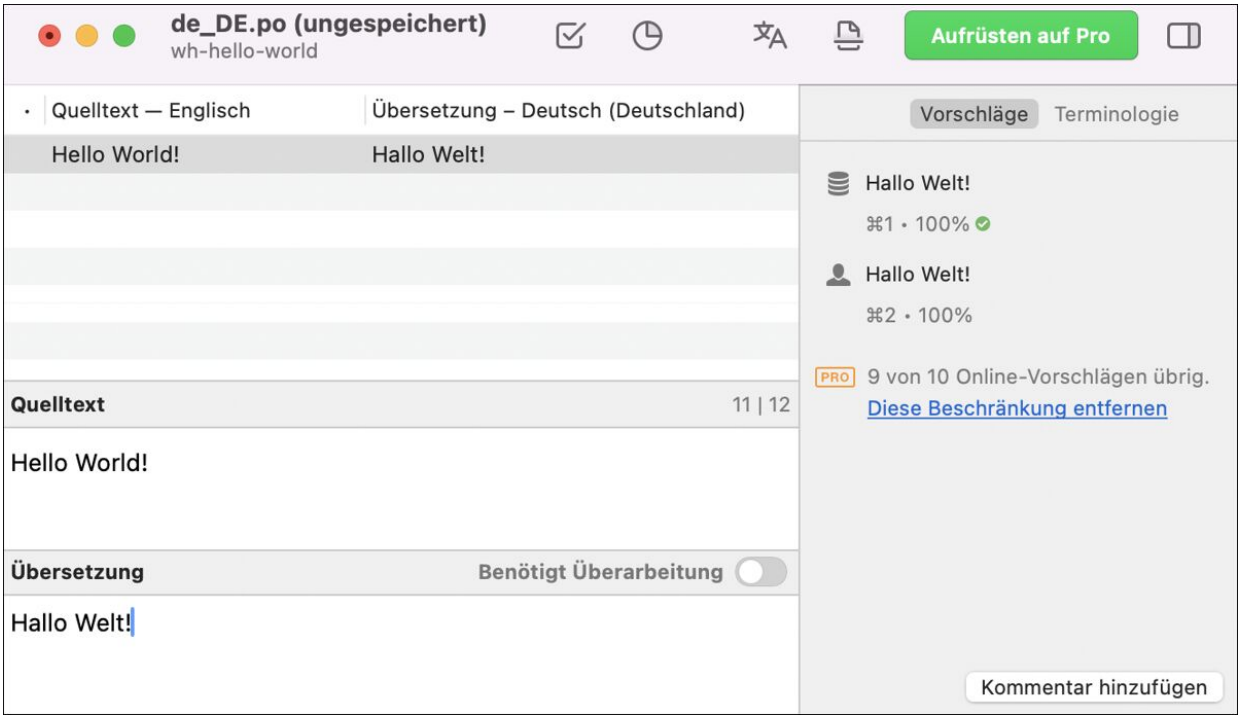


Abbildung 22.5 Ansicht in Poedit, in der Sie die Übersetzungen erstellen können

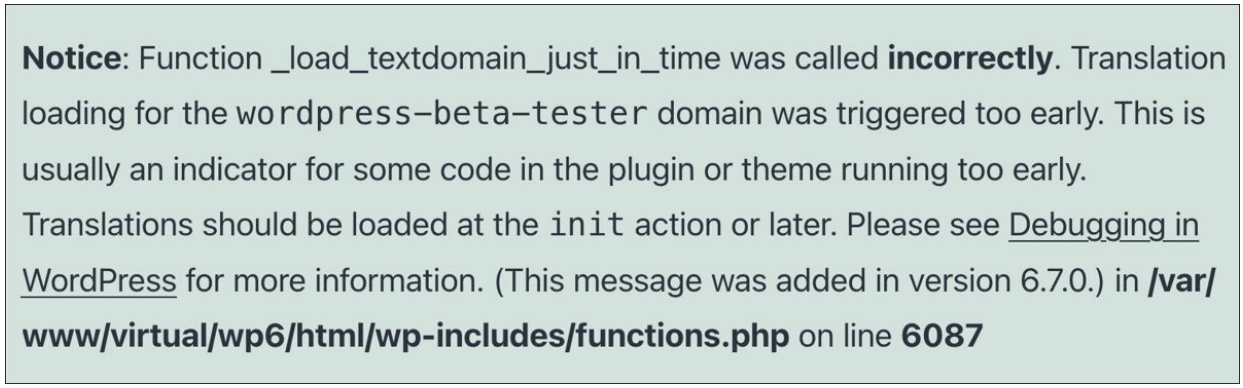


Abbildung 22.6 Ausgabe einer PHP-Warnung auf der Website

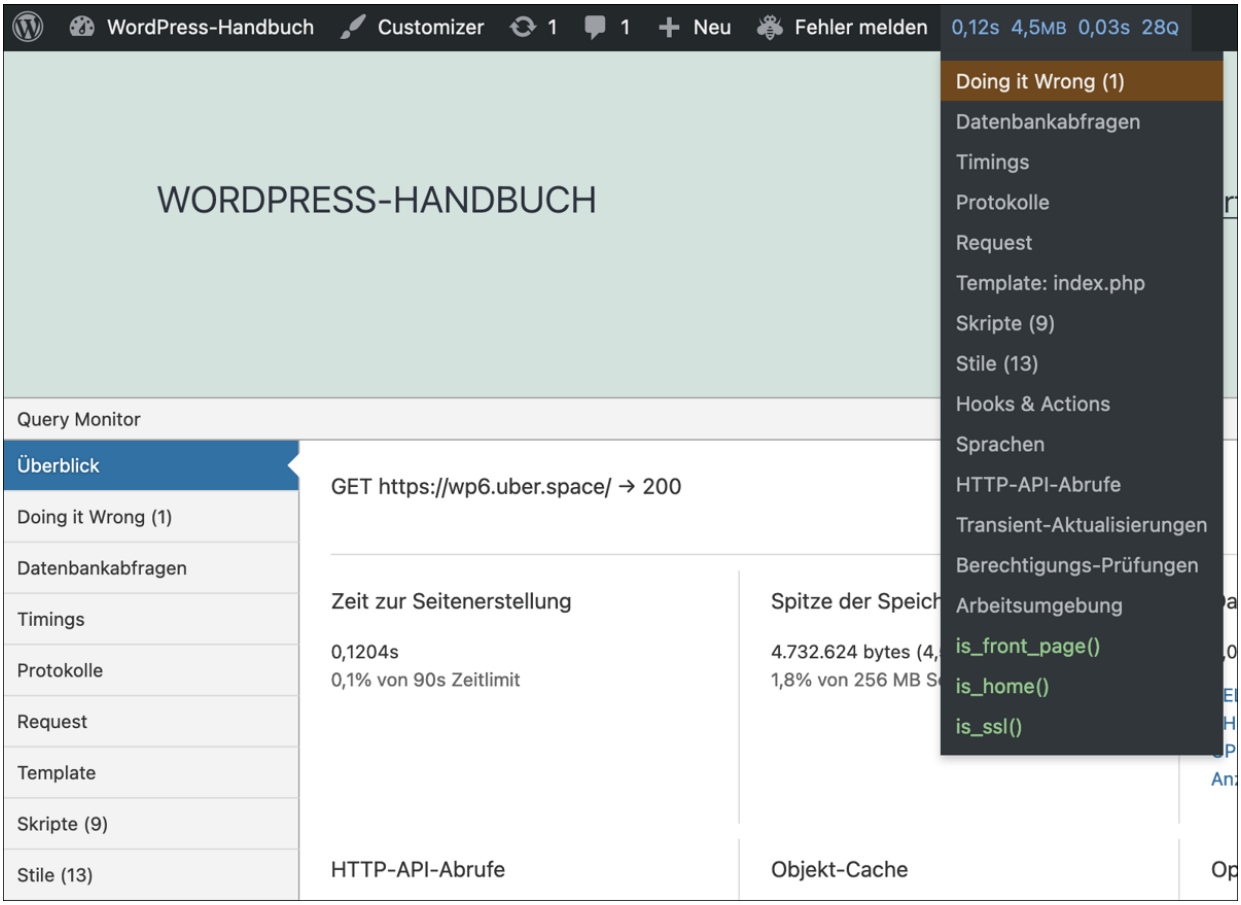


Abbildung 22.7 Das Query-Monitor-Plugin gibt Ihnen nützliche Einblicke hinter die Kulissen eines Requests.

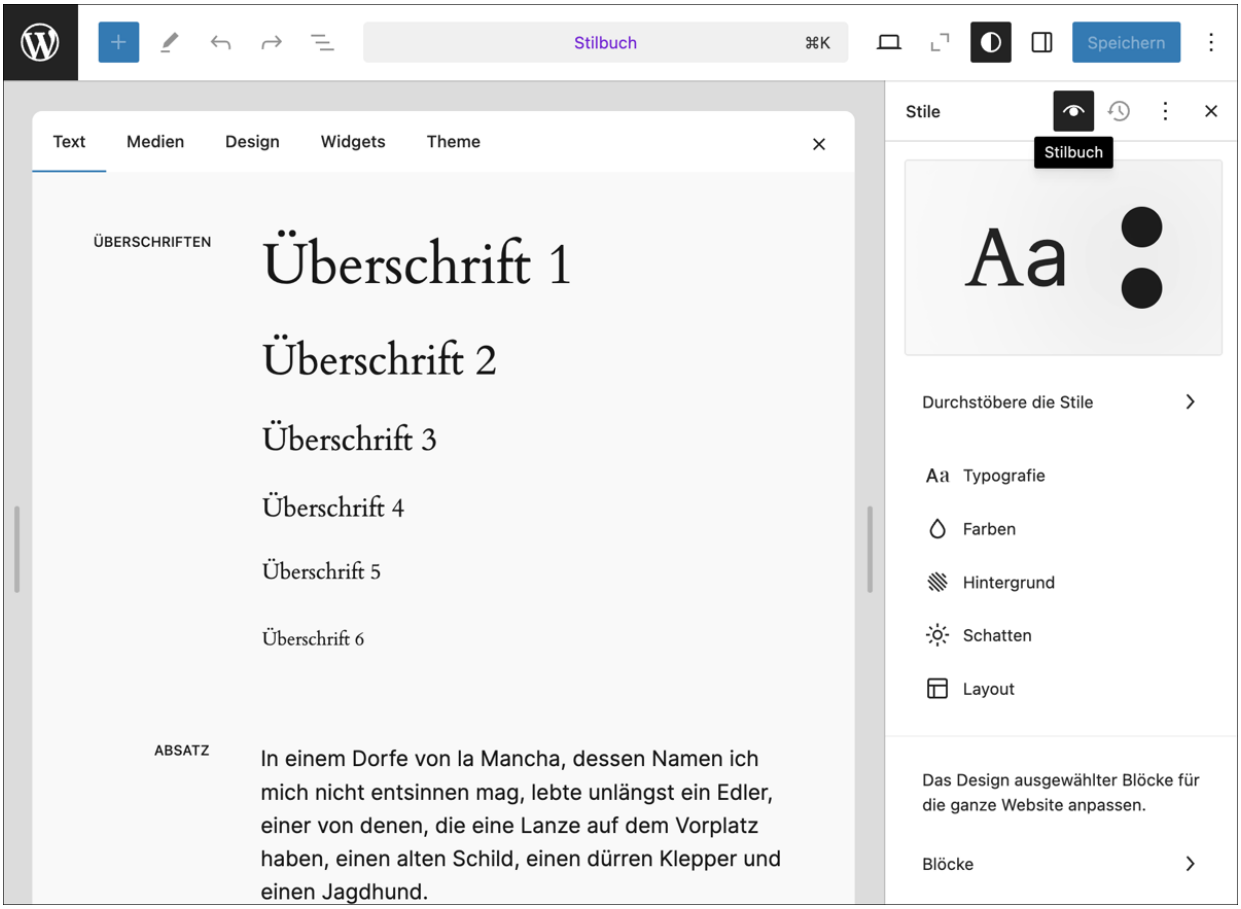


Abbildung 23.1 Mit dem Stilbuch haben Sie eine schöne Übersicht über alle verfügbaren Blöcke und deren Aussehen.

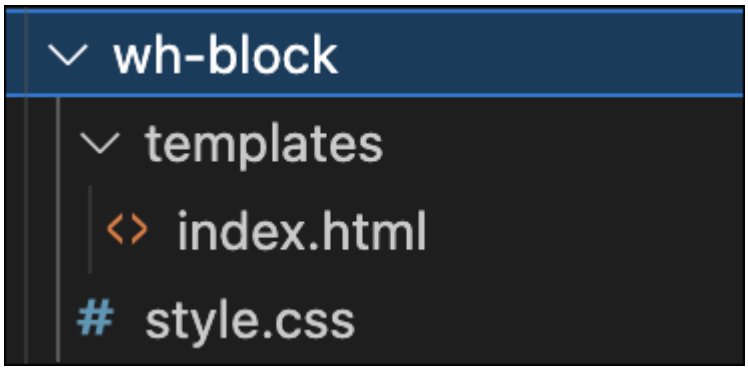


Abbildung 23.2 So muss die Datei- und Ordnerstruktur des Block-Themes aussehen, damit WordPress es erkennen kann.

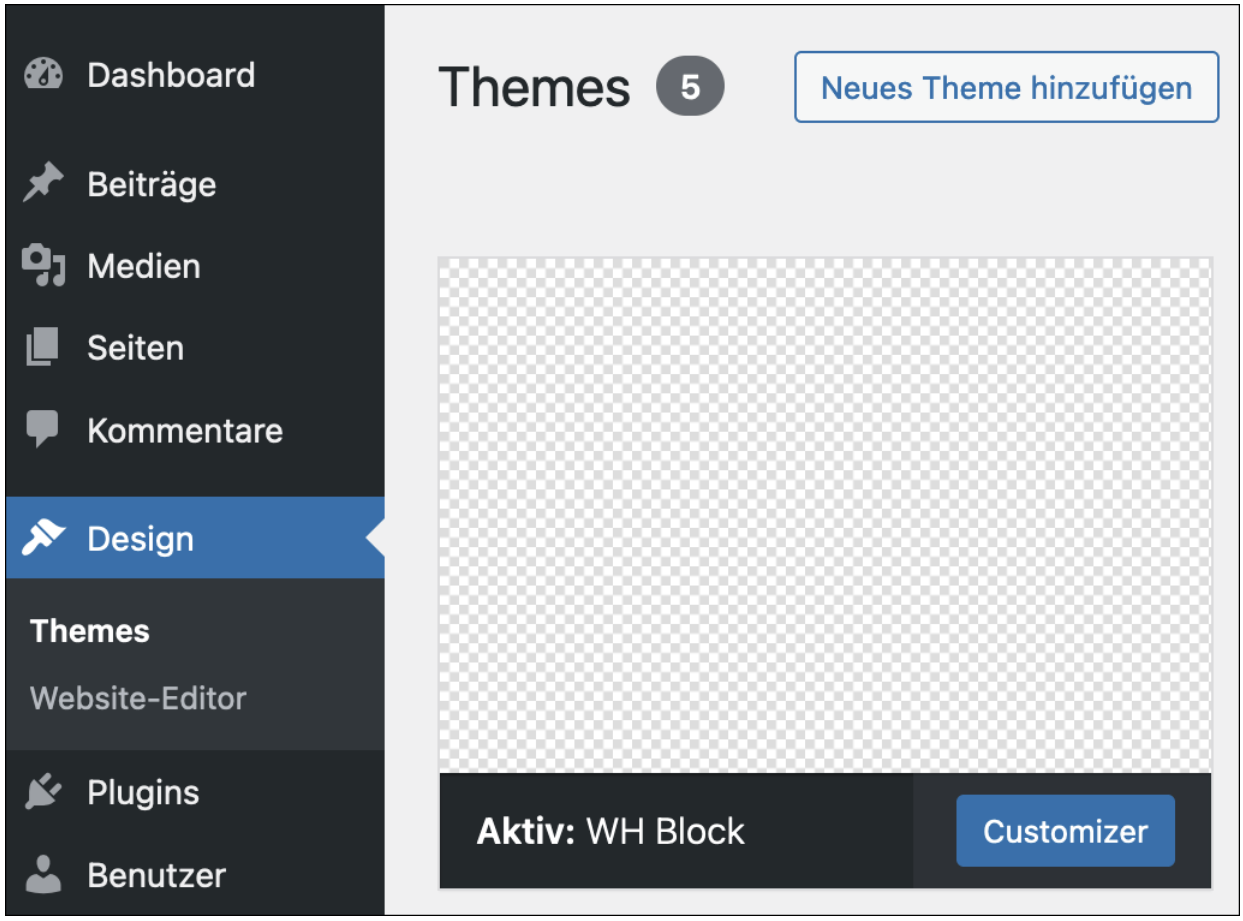


Abbildung 23.3 Das Theme-Grundgerüst kann bereits in WordPress aktiviert werden. Dass es ein Block-Theme ist, fällt direkt aufgrund des Eintrags »Design« • »Website-Editor« links im Admin-Menü auf.

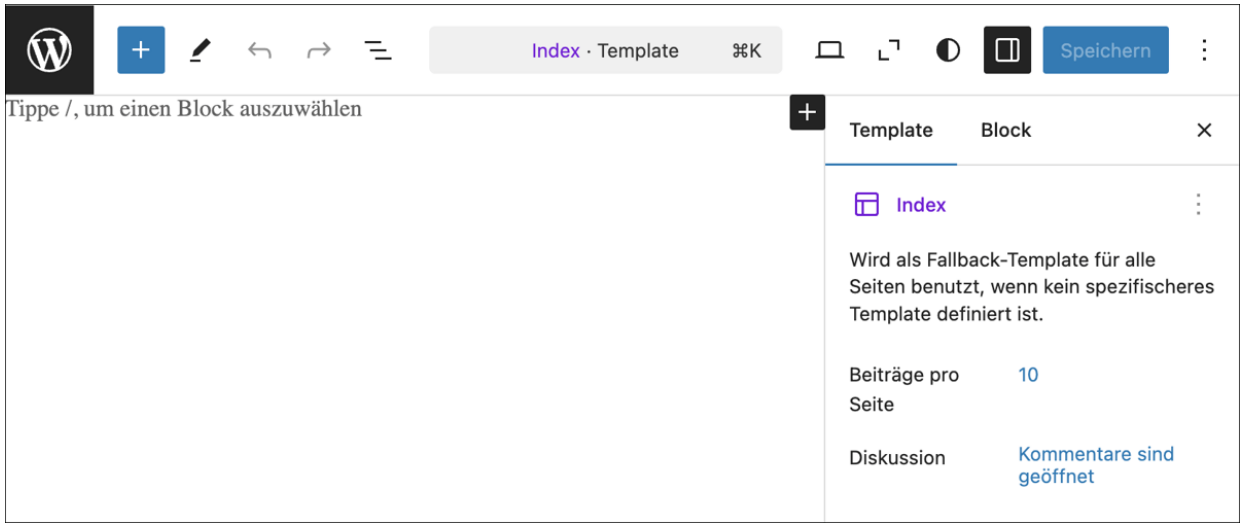


Abbildung 23.4 Nach dem Öffnen des Website-Editors werden Sie von einem leeren Index-Template begrüßt.

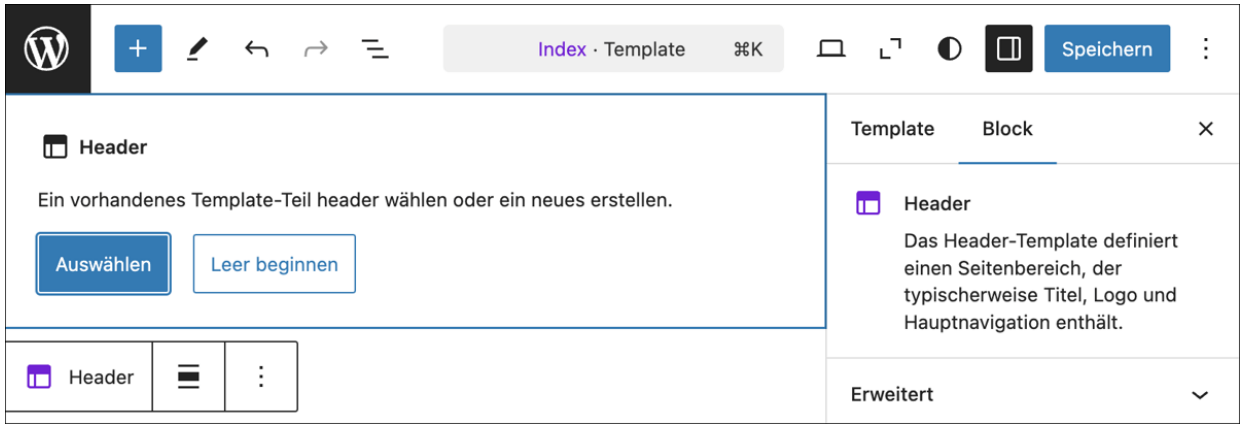


Abbildung 23.5 Mit dem »Header«-Block wird der Template-Teil für den Kopfbereich der Website erstellt.



Abbildung 23.6 Der Header-Template-Teil im Website-Editor

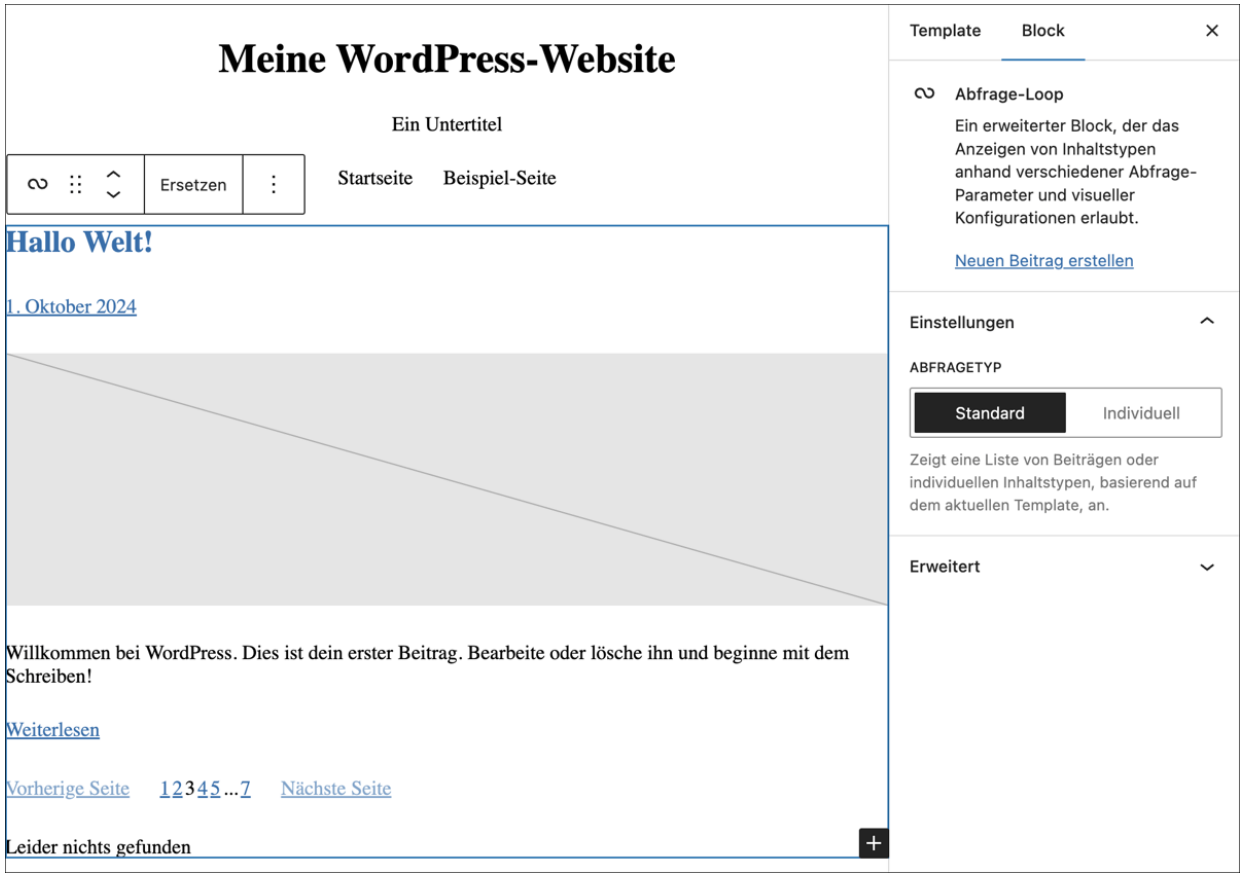


Abbildung 23.7 Mit dem »Abfrage-Loop«-Block erstellen Sie die Liste der neuesten Beiträge.

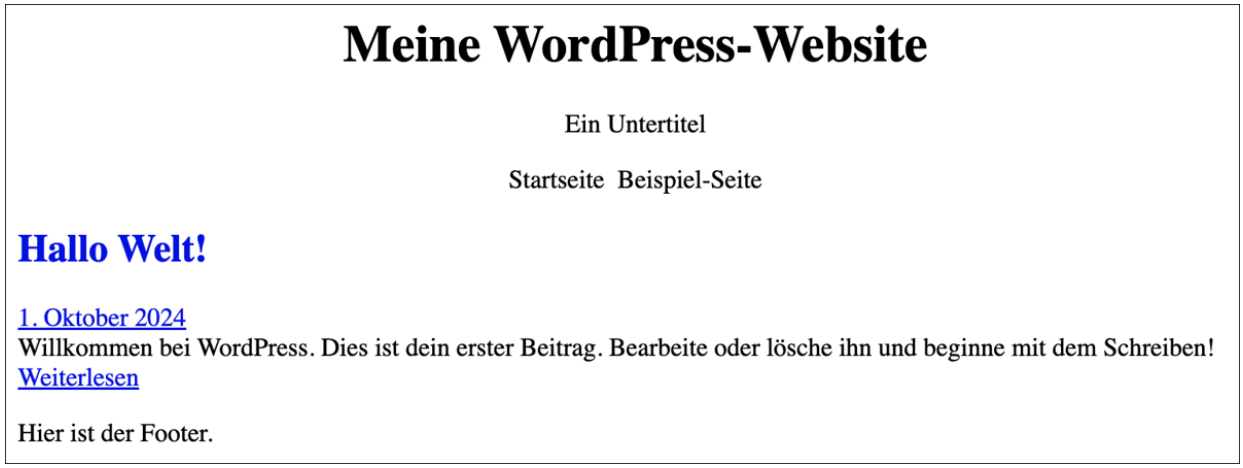


Abbildung 23.8 Das Index-Template mit Header, Beitragsliste und Footer im Frontend

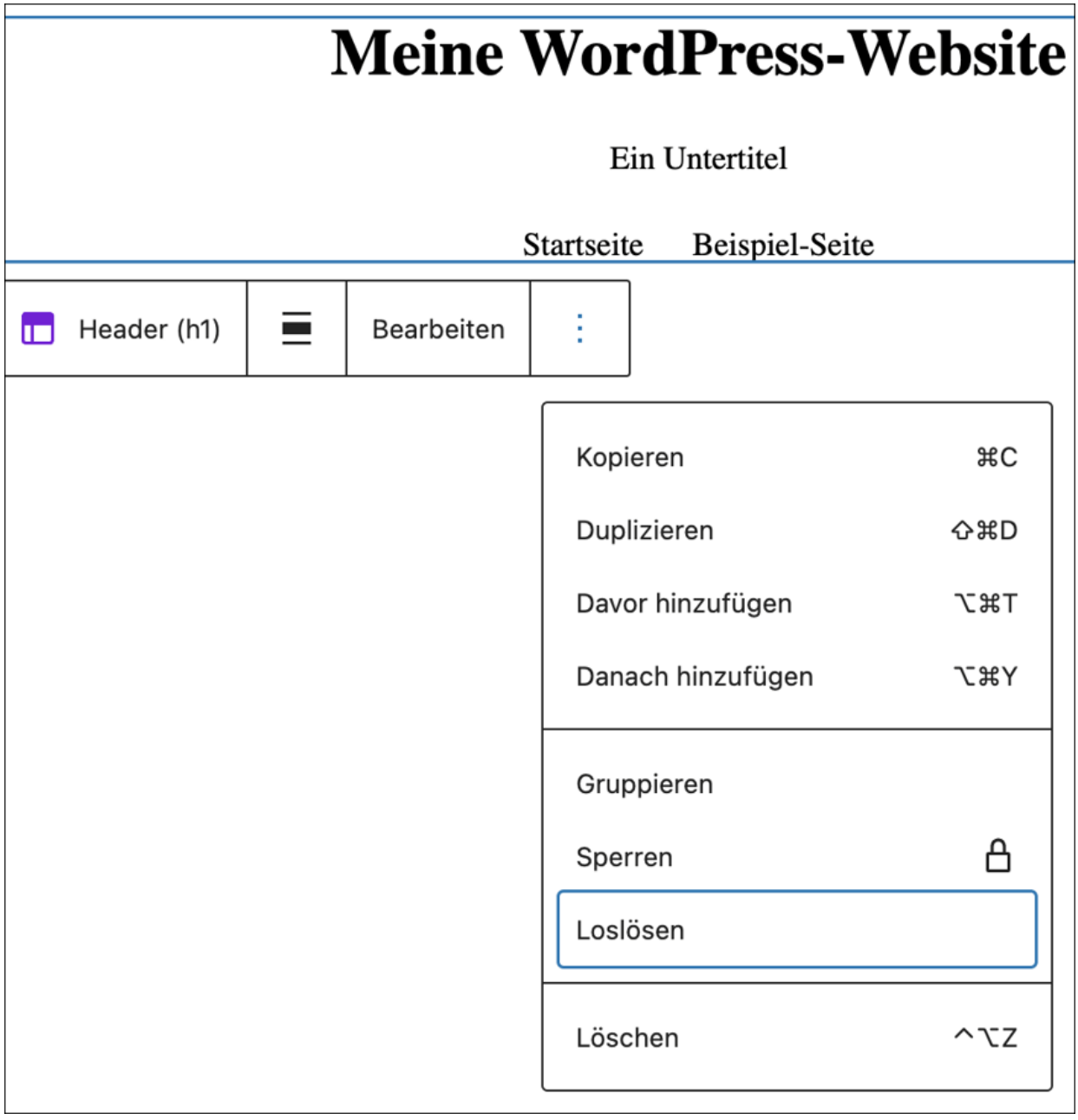


Abbildung 23.9 Ein Template-Teil kann recht einfach in seine Blöcke aufgelöst werden, sodass sie verändert werden können, ohne den Template-Teil selbst anpassen zu müssen.

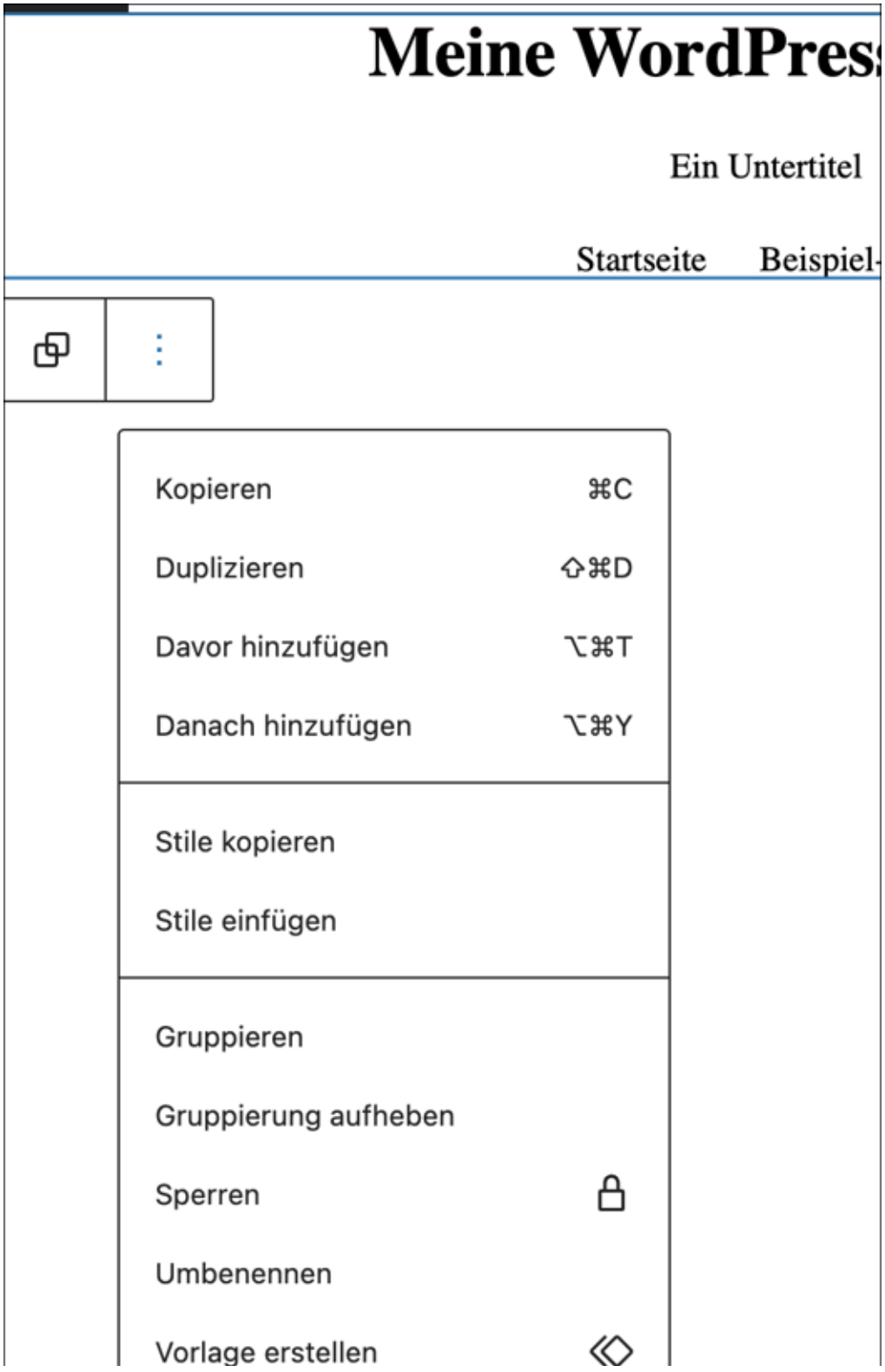

Meine WordPres
Ein Untertitel
Startseite
Beispiel
Kopieren ⌘C
Duplizieren ⇧⌘D
Davor hinzufügen ⌥⌘T
Danach hinzufügen ⌥⌘Y
Stile kopieren
Stile einfügen
Gruppieren
Gruppierung aufheben
Sperren
Umbenennen
Vorlage erstellen



Abbildung 23.10 Ausgewählte Blöcke können in einen Template-Teil umgewandelt werden.

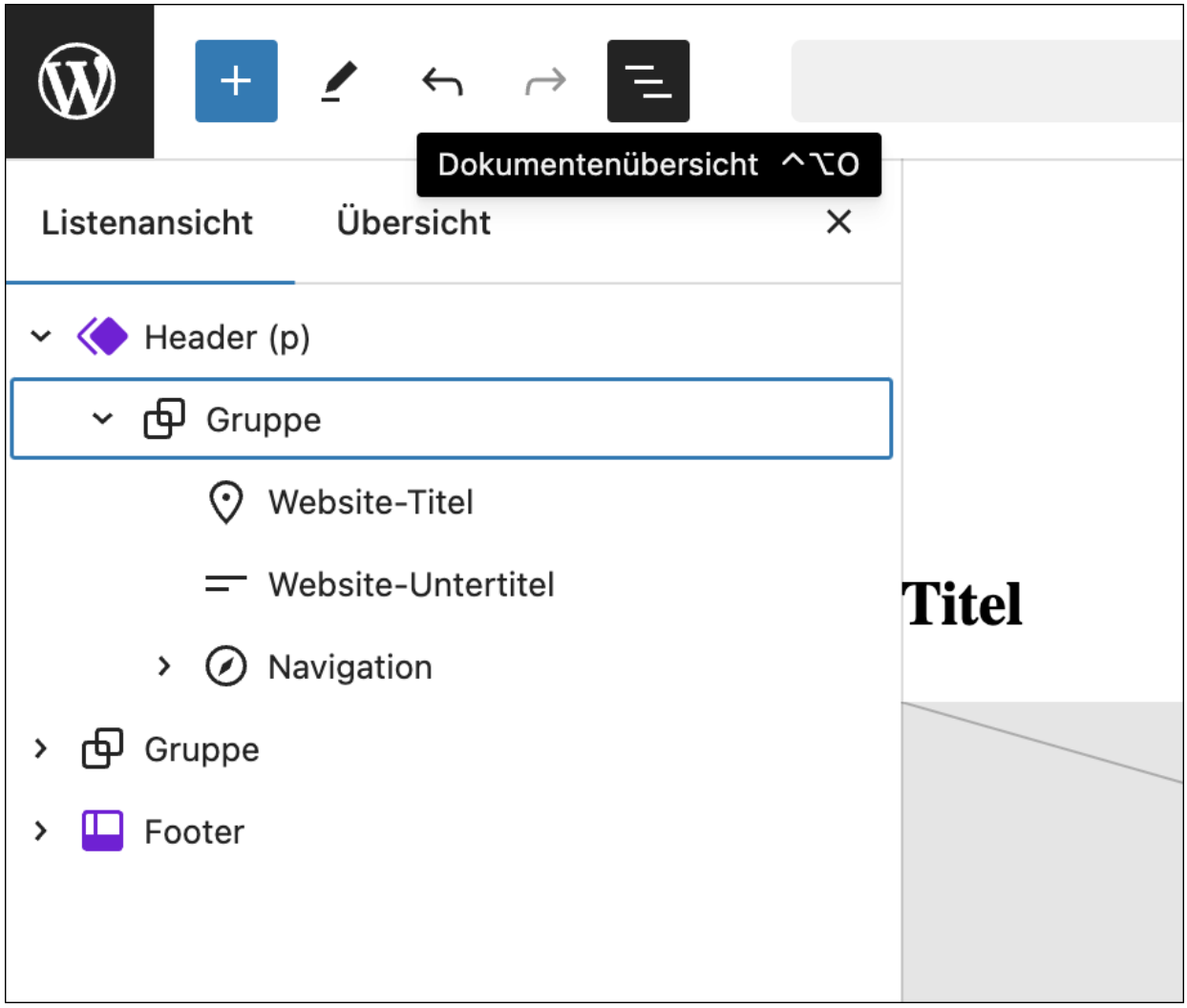


Abbildung 23.11 Die Listenansicht der eingefügten Blöcke hilft bei der Übersicht mit verschachtelten Blöcken.



Abbildung 23.12 Das Template für Seiten in Aktion



Abbildung 23.13 Die Einzelansicht eines Beitrags im Frontend

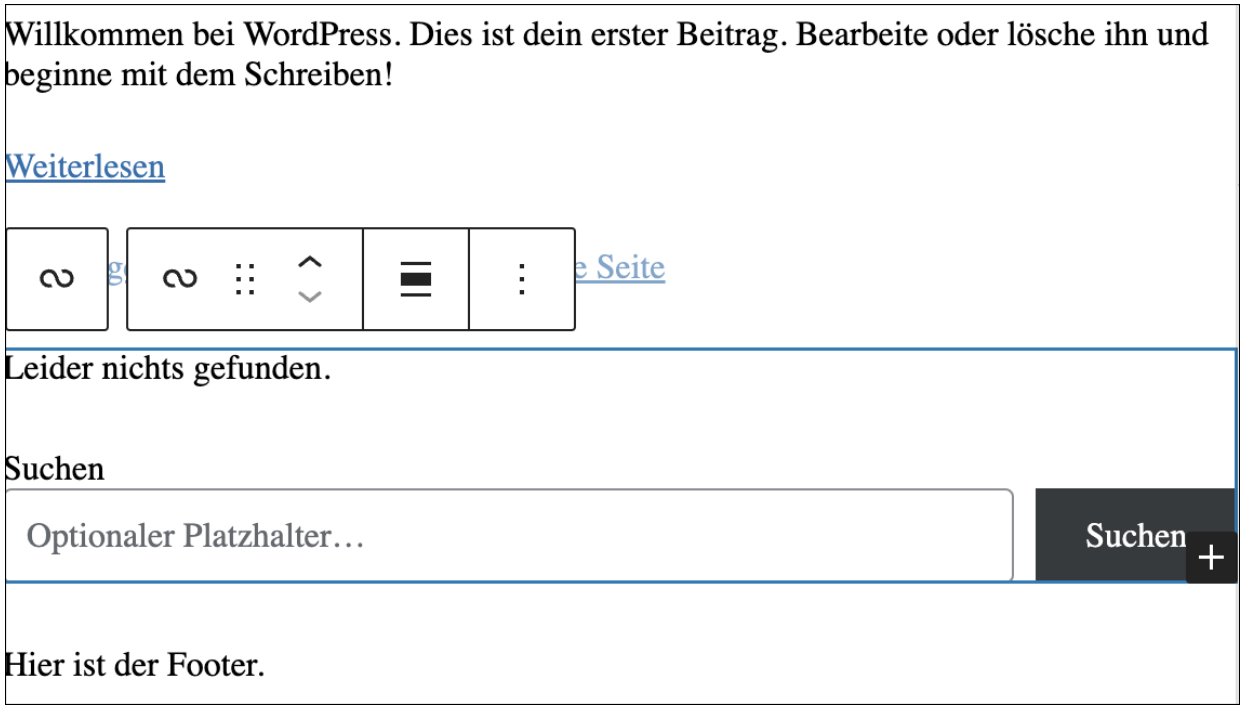


Abbildung 23.14 Wenn keine Beiträge für eine Suche gefunden wurden, wird ein Suchformular angeboten, um es noch einmal zu versuchen.

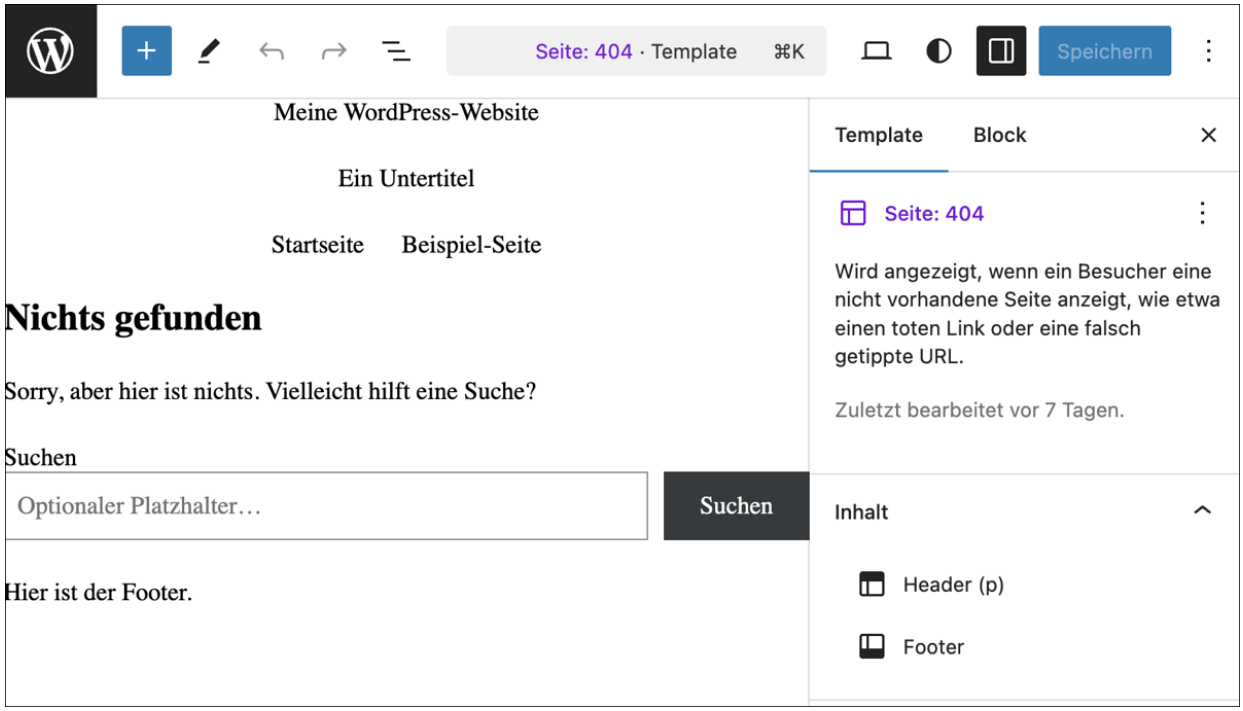


Abbildung 23.15 Mit dem 404-Template sind alle Templates erstellt, die unser Theme haben soll.

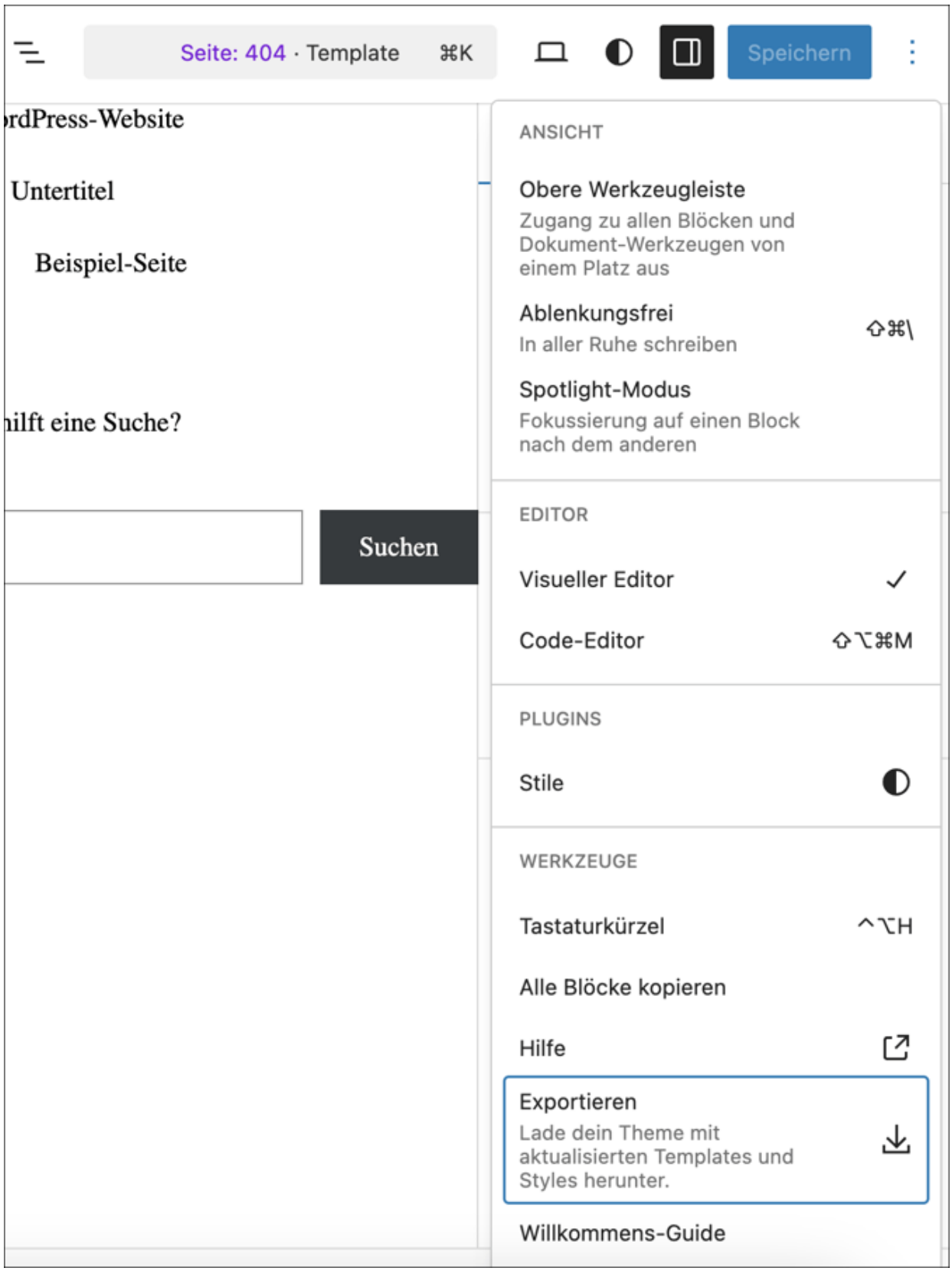

Seite: 404 · Template ⌘K
Speichern
rdPress-Website
Untertitel
Beispiel-Seite
ilft eine Suche?
Suchen
ANSICHT
Obere Werkzeugleiste
Zugang zu allen Blöcken und Dokument-Werkzeugen von einem Platz aus
Ablenkungsfrei
In aller Ruhe schreiben
⇧⌘\
Spotlight-Modus
Fokussierung auf einen Block nach dem anderen
EDITOR
Visueller Editor
Code-Editor
⇧⌥⌘M
PLUGINS
Stile
WERKZEUGE
Tastaturkürzel
^⌥H
Alle Blöcke kopieren
Hilfe
Exportieren
Lade dein Theme mit aktualisierten Templates und Styles herunter.
Willkommens-Guide

Abbildung 23.16 Der aktuelle Stand des Themes lässt sich über den Website-Editor exportieren.

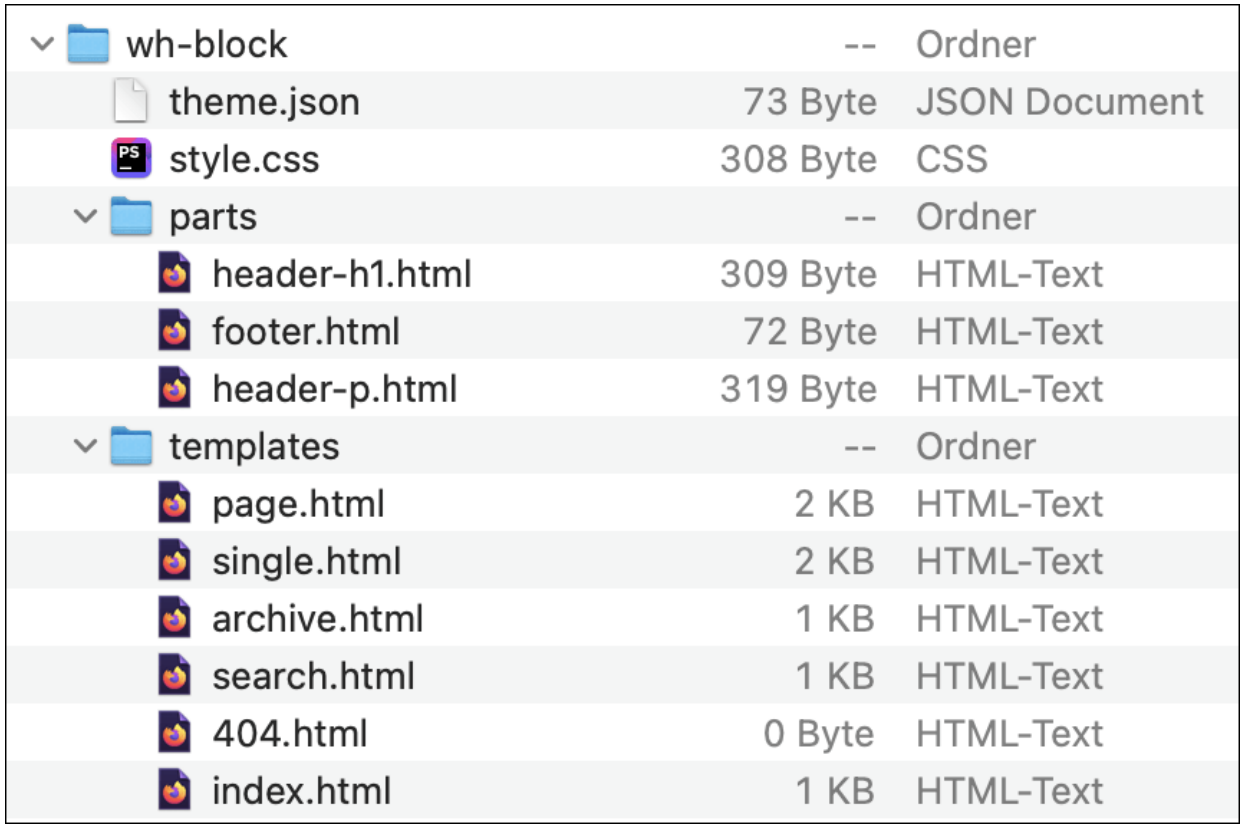


Abbildung 23.17 Der entpackte Ordner enthält alle Dateien, die für ein Block-Theme notwendig sind.



Abbildung 23.18 So sieht ein Navigations-Block aus, für den in der Template-Datei eine Menü-ID angegeben wurde, die es auf dieser Website nicht gibt.

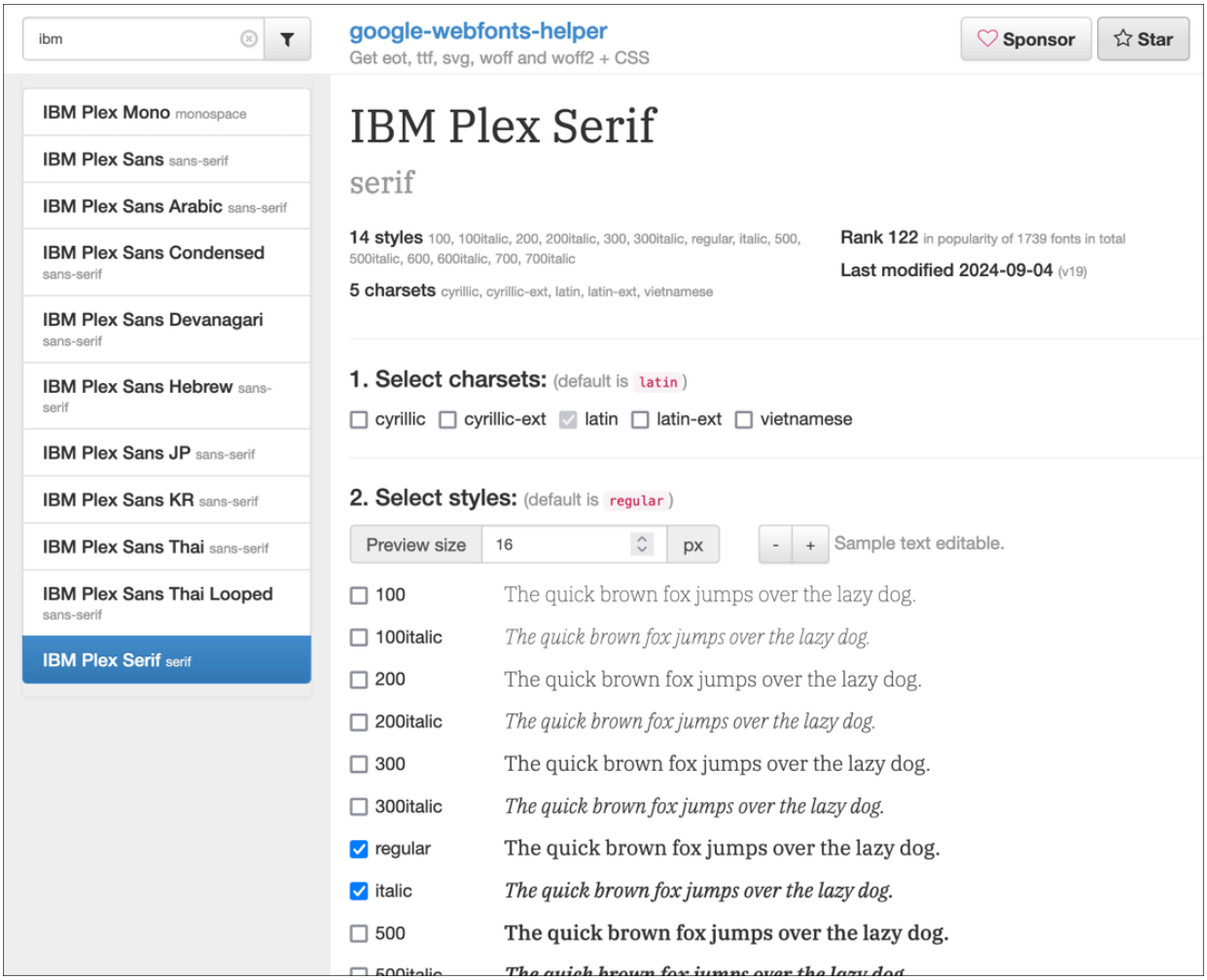


Abbildung 23.19 Mit dem »google-webfonts-helper« lassen sich Schriftdateien für das lokale Hosting auf dem Webserver der eigenen Website einfach herunterladen.

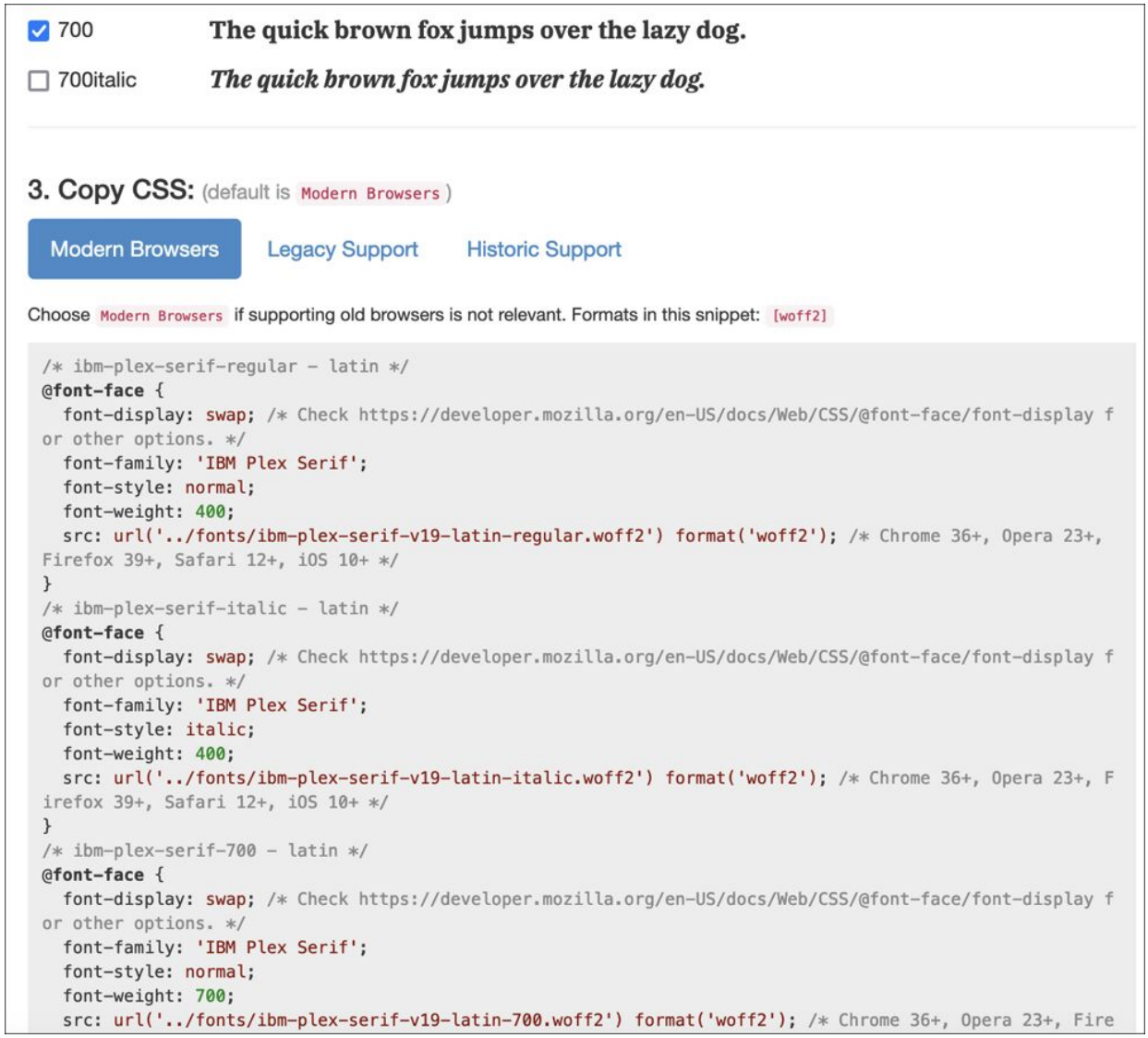


Abbildung 23.20 Die Option »Modern Browsers« sollte eigentlich für alle Websites ausreichen.



Abbildung 23.21 Nach Anpassen der Inhaltsbreite und Hinzufügen der Schrift sieht das Template im Editor schon besser aus.

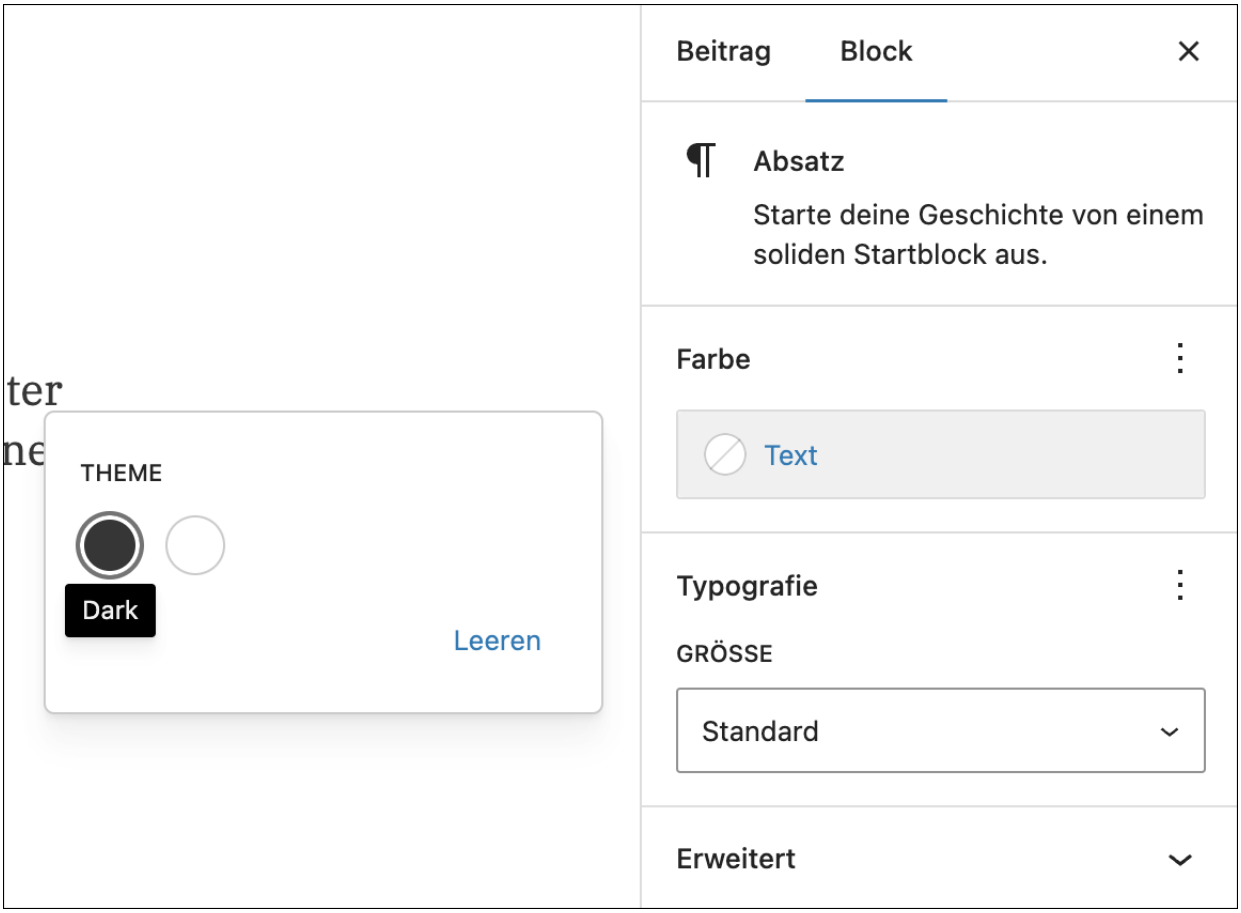


Abbildung 23.22 Die Theme-Farbpalette wird bei allen Blöcken mit Farboptionen angezeigt.

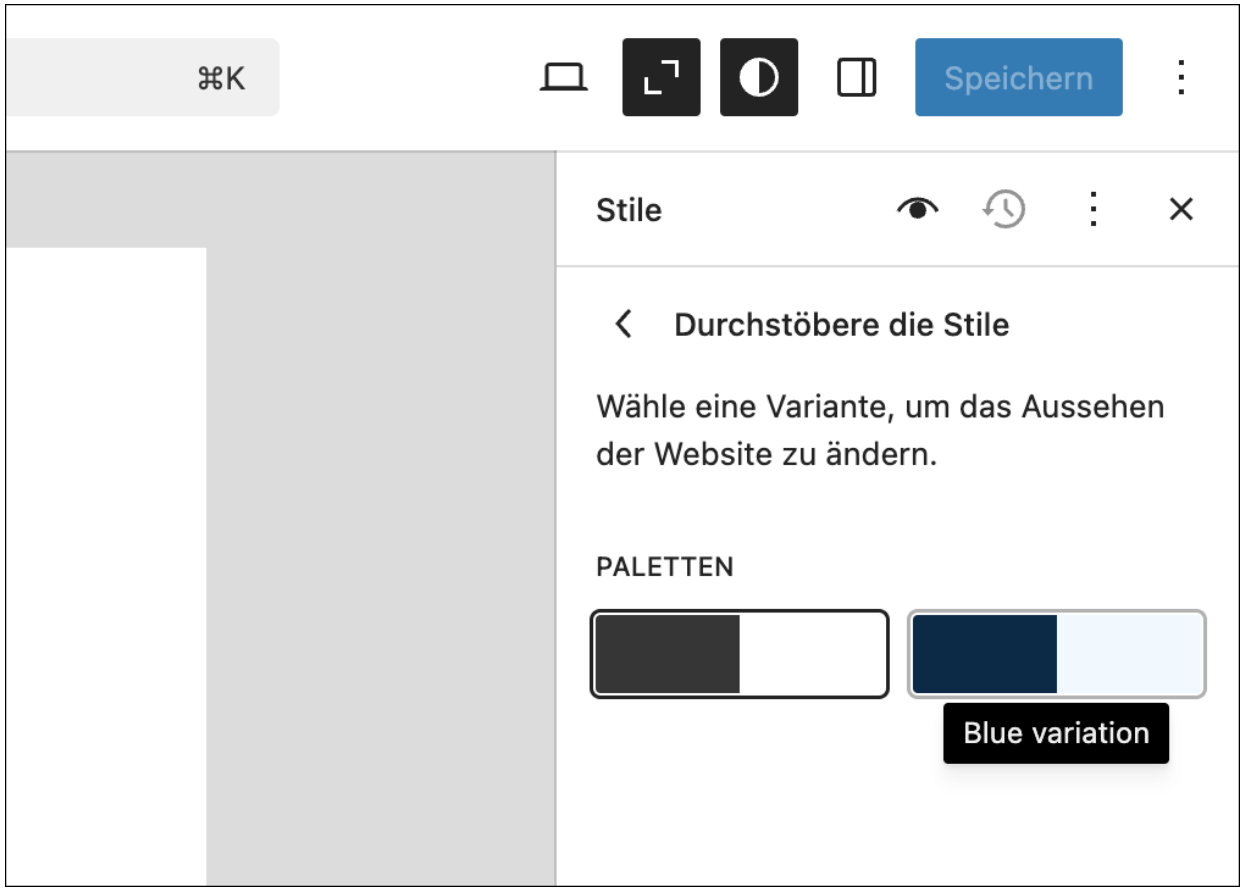


Abbildung 23.23 In der »Stile«-Seitenleiste können vorhandene andere Stilkombinationen über den Punkt »Durchstöbere die Stile« ausgewählt werden.

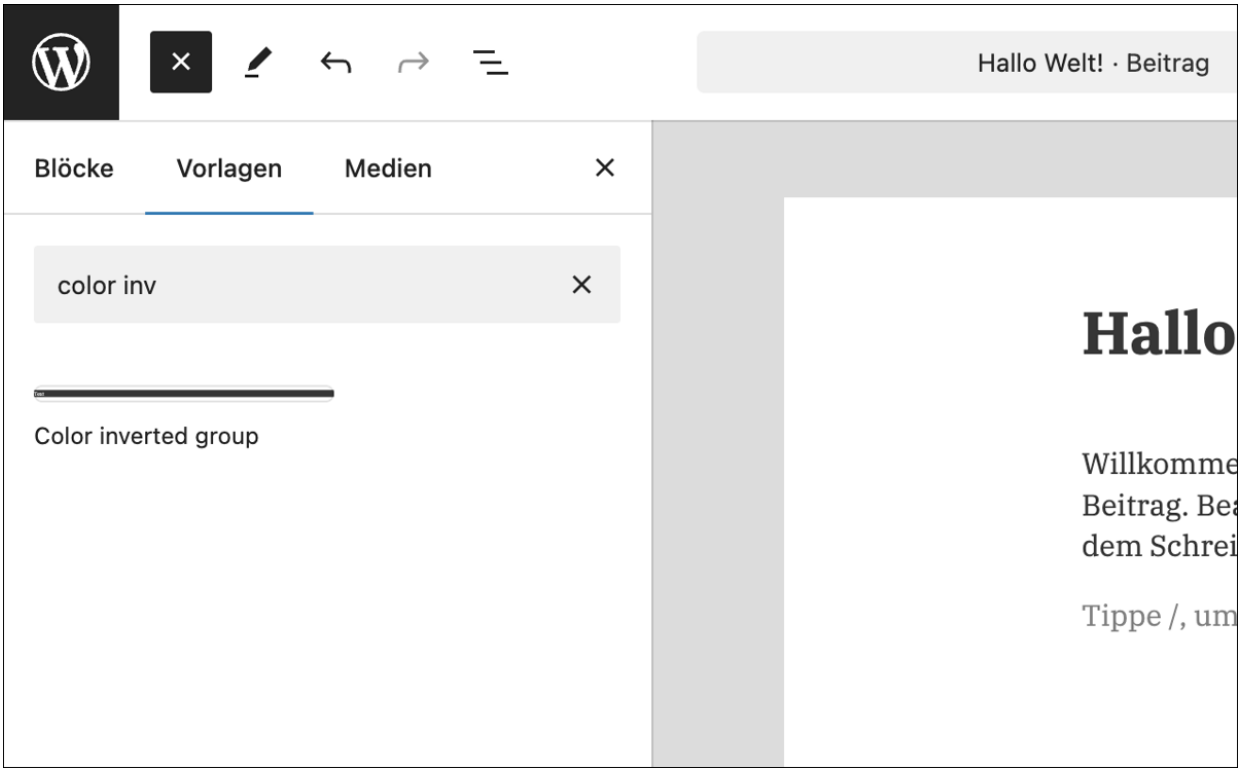


Abbildung 23.24 Im Block-Inserter können die vorhandenen Block-Vorlagen durchgeschaut und eingefügt werden.

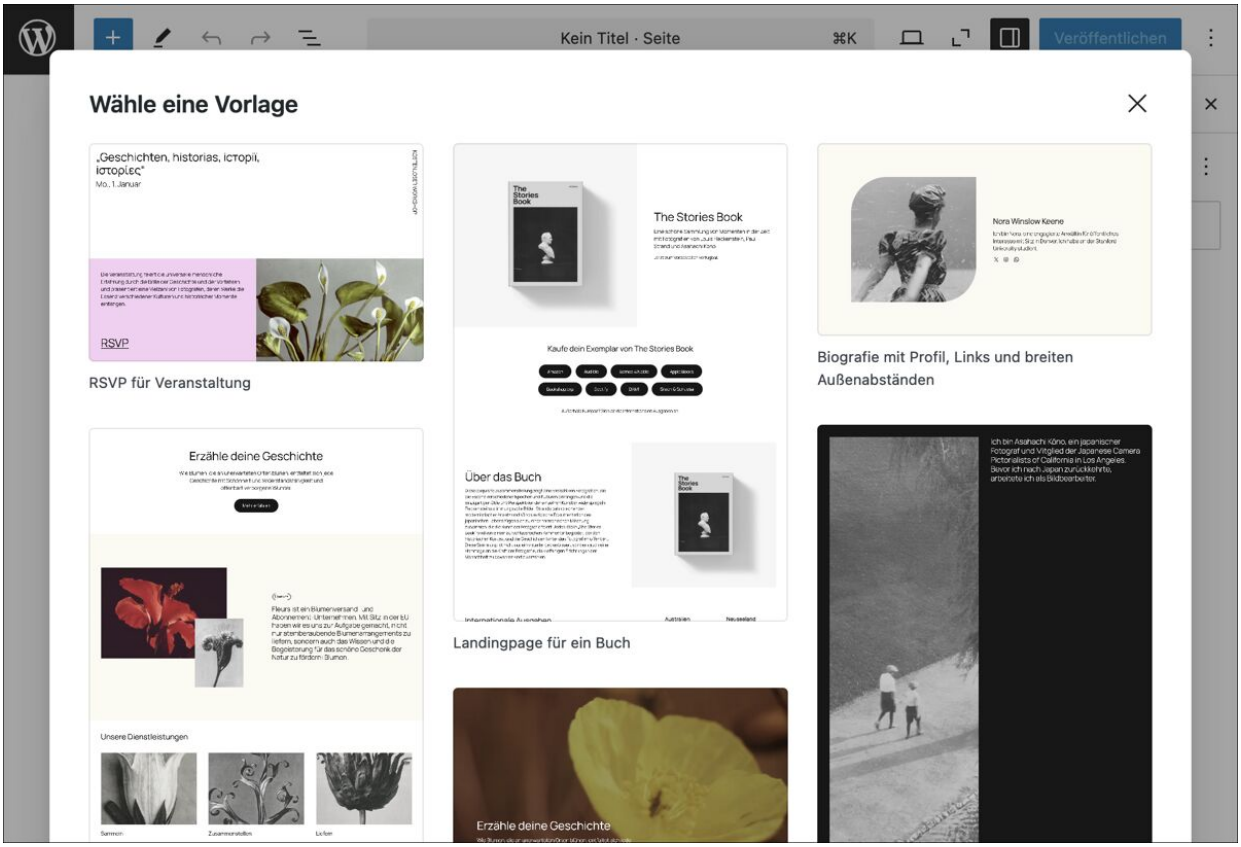


Abbildung 23.25 Seitenvorlagen in dem Theme Twenty Twenty-Five

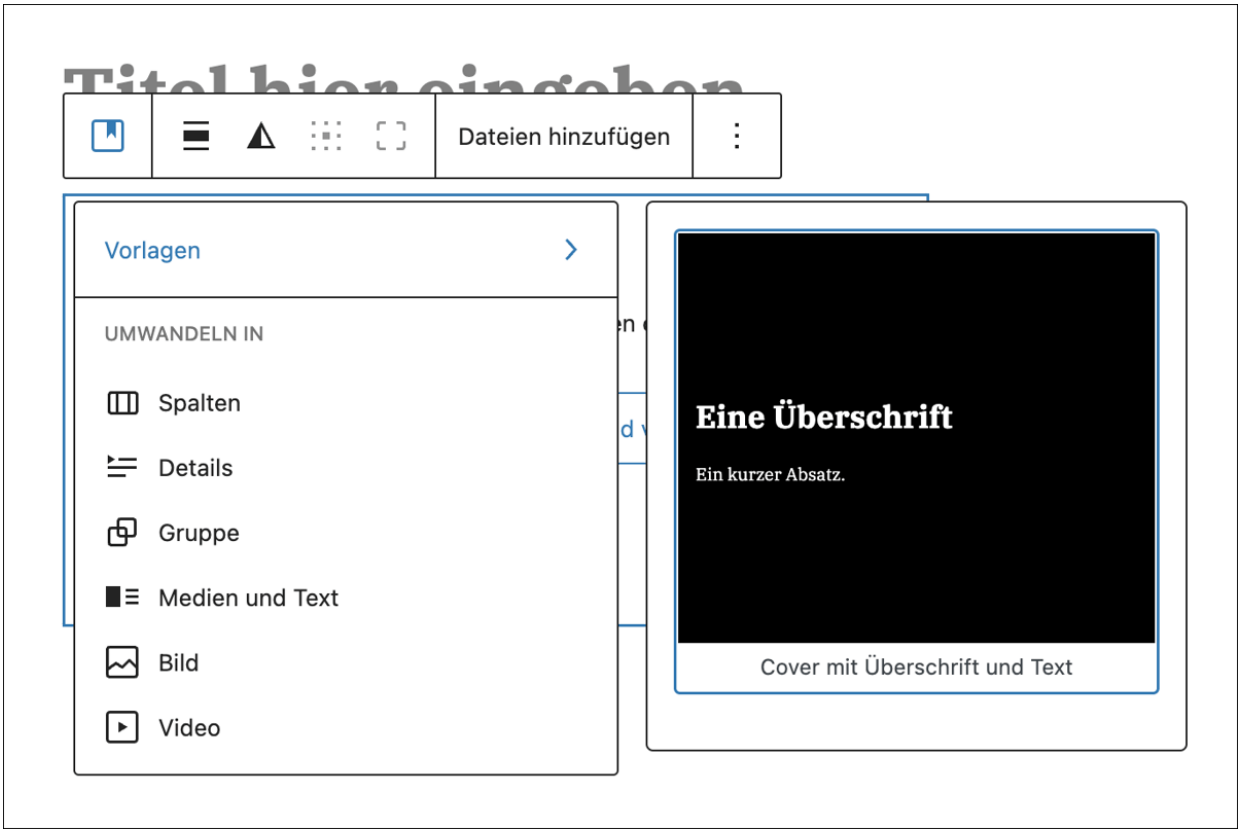


Abbildung 23.26 Block-Vorlagen können als Variante für bestimmte Blöcke markiert werden.

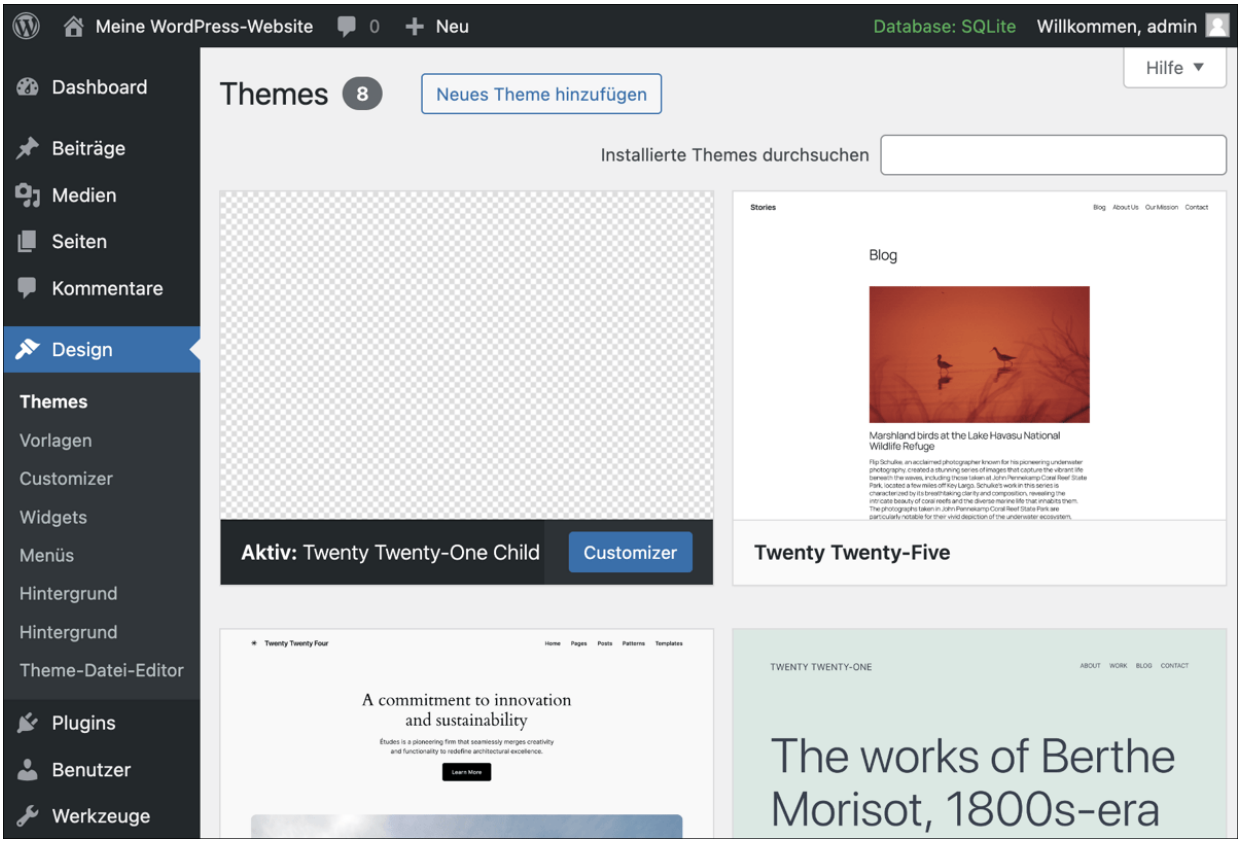


Abbildung 23.27 Nach Anlegen der Datei »style.css« erscheint das Child-Theme bereits im WordPress-Backend und kann aktiviert werden. Jetzt kann es losgehen mit den Anpassungen. (Soll unter »Design« • »Themes« ein Vorschaubild erscheinen, legen Sie einen Screenshot namens »screenshot.png« ins Child-Theme-Verzeichnis.)


News Showcase Hosting Extend Learn Community About Get WordPress
Make WordPress Themes
Blog Trac (Theme Reviews)
Login
Timeline View Tickets Browse Source Ticket Graph
Search
Reverse Diff
Changes from twentytwentyone/2.2 at r223030 to twentytwentyone/2.3 at r235022
Location: twentytwentyone/2.3
Files: 27 edited
assets/css/ie-editor.css (10 diffs)
assets/css/ie-editor.css.map (1 diff)
assets/css/ie.css (34 diffs)
assets/css/ie.css.map (1 diff)
assets/css/style-editor.css (12 diffs)
assets/css/style-editor.css.map (1 diff)
assets/sass/01-settings/file-header.scss (1 diff)
assets/sass/03-generic/breakpoints.scss (5 diffs)
assets/sass/03-generic/vertical-margins.scss (1 diff)
assets/sass/04-elements/blockquote.scss (3 diffs)
assets/sass/04-elements/misc.scss (1 diff)
assets/sass/05-blocks/latest-comments/_editor.scss (1 diff)
assets/sass/05-blocks/latest-comments/_style.scss (2 diffs)
assets/sass/05-blocks/media-text/_style.scss (1 diff)
assets/sass/05-blocks/preformatted/_editor.scss (1 diff)
assets/sass/05-blocks/quote/_editor.scss (6 diffs)
assets/sass/05-blocks/quote/_style.scss (5 diffs)
assets/sass/05-blocks/search/_style.scss (1 diff)
assets/sass/06-components/header.scss (1 diff)
inc/menu-functions.php (1 diff)
inc/template-tags.php (2 diffs)
package-lock.json (70 diffs)
package.json (2 diffs)
readme.txt (2 diffs)
style-rtl.css (36 diffs)
style.css (36 diffs)
style.css.map (1 diff)
View differences inline
Show 2 lines around each change
Show the changes in full context
Ignore:
Blank lines
Case changes
White space changes
Update
Unmodified Added Removed
twentytwentyone/2.3/assets/css/ie-editor.css
Tabular Unified
r223030 r235022
185 185 blockquote:before {
186 186 content: "“";
187 font-size: 1.25rem;
188 line-height: 1.7;
189 187 position: absolute;
190 188 left: -12px;
195 193 blockquote footer {
196 194 color: #28303d;
197 font-size: 1rem;
198 195 font-style: normal;
199 196 }
200 @media only screen and (max-width: 481px) {
197 @media only screen and (max-width: 481.98px) {
201 198
202 199 blockquote {
1004 1001 }
1005 1002
1006 .wp-block-latest-comments,

Abbildung 23.28 Für Themes aus dem WordPress.org-Verzeichnis können Sie sich Änderungen zwischen Versionen anschauen, hier beispielsweise zwischen Twenty Twenty-One 2.2 und 2.3.

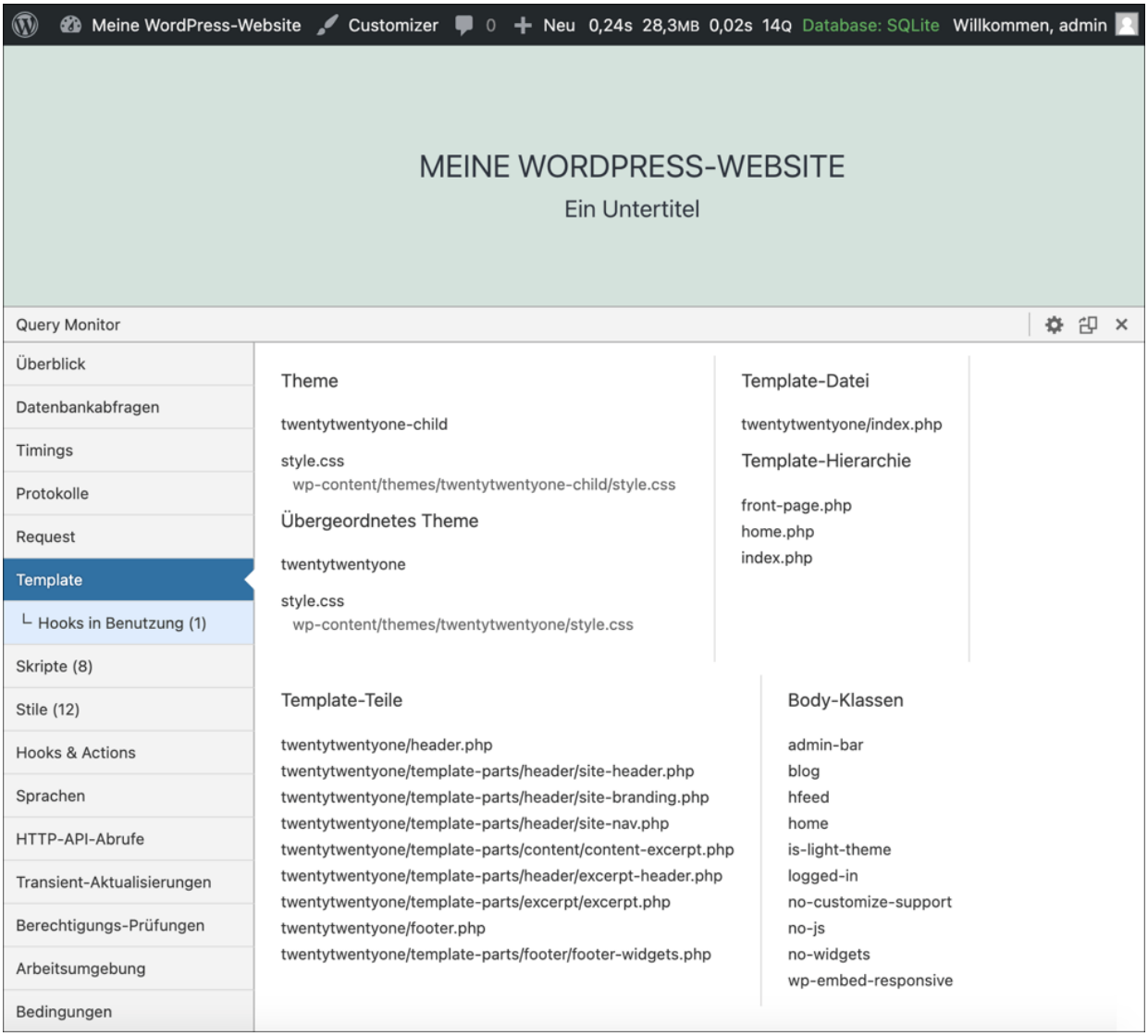


Abbildung 23.29 Das Plugin Query Monitor gibt Ihnen unter anderem Auskunft darüber, welche Template-Datei und welche Template-Teile geladen wurden.



Abbildung 23.30 Den Website-Titel und den Hinweis auf WordPress wollen wir im Footer nicht haben.

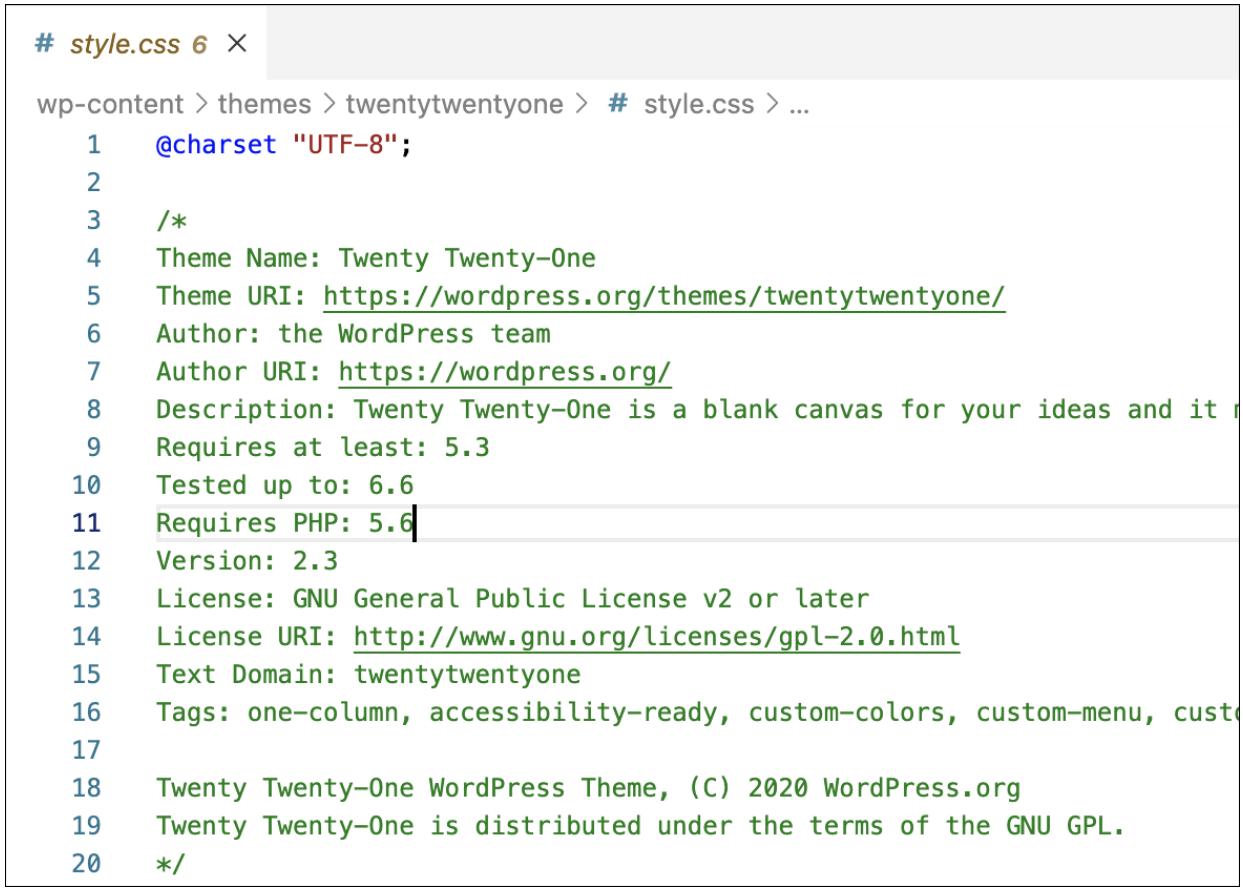


Abbildung 23.31 Der Theme-Meta-Kommentar am Anfang der »style.css« des Standard-Themes Twenty Twenty-One

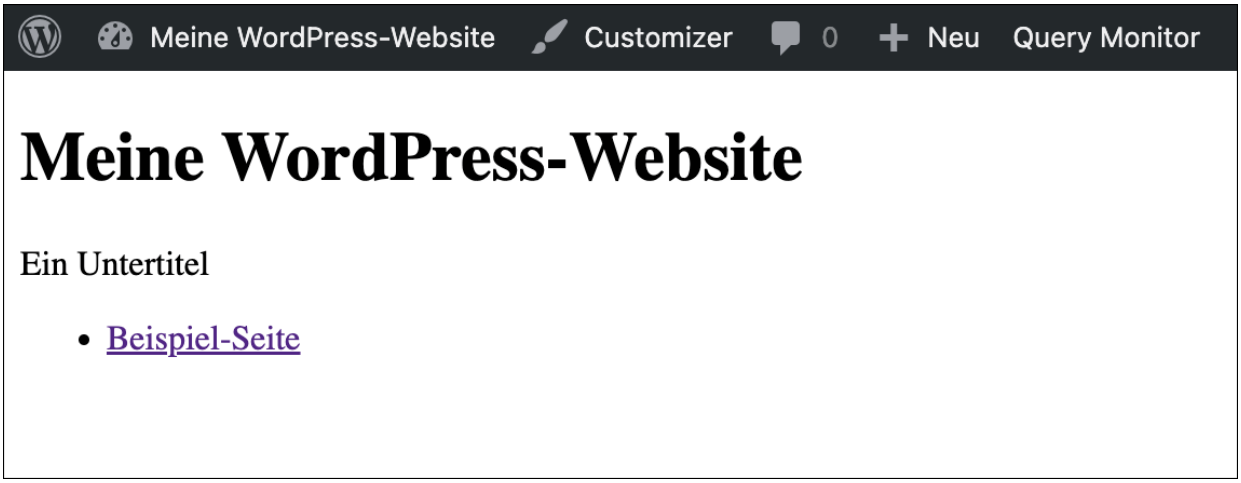


Abbildung 23.32 Nicht schön, aber immerhin schon mal eine Ausgabe nach Anlegen der »header.php« mit den Template-Teilen für Branding-Inhalte und Navigation

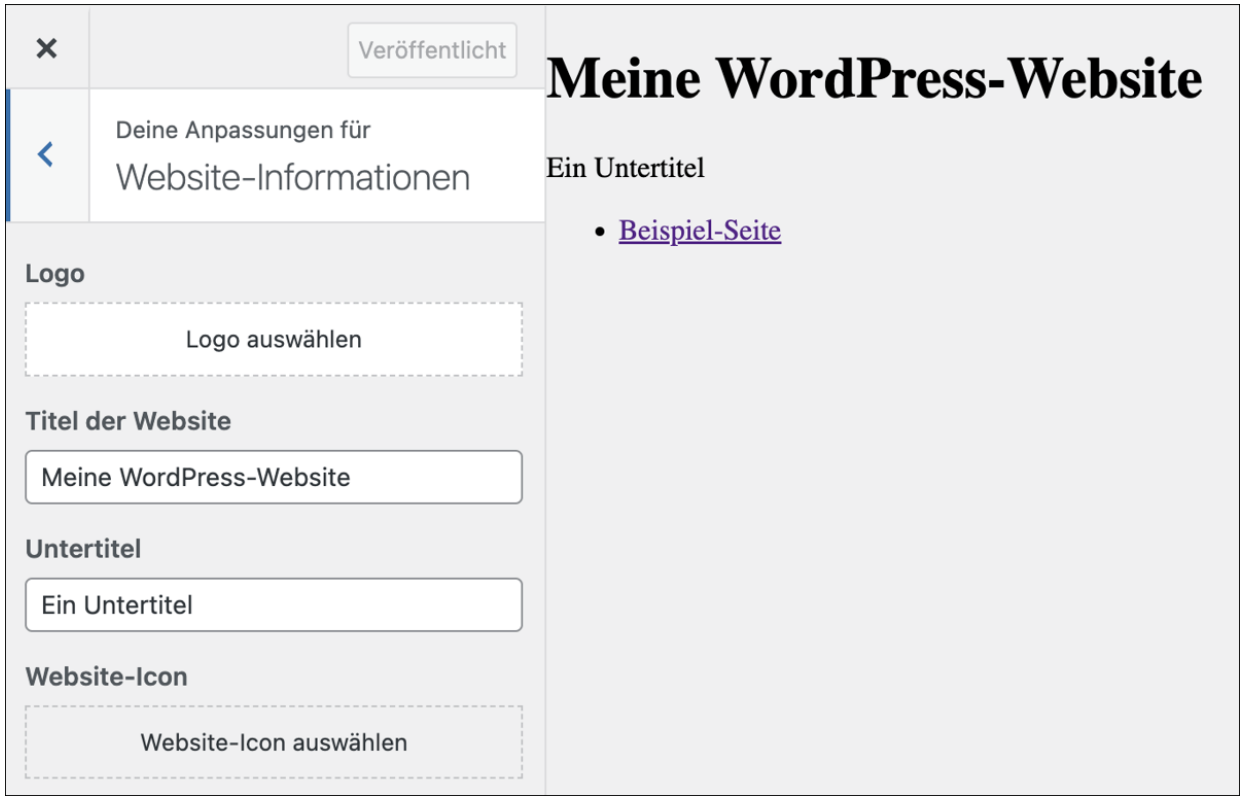


Abbildung 23.33 Nach der Aktivierung des Theme-Supports für die Logo-Funktion ist die Option dafür im Customizer zu finden.

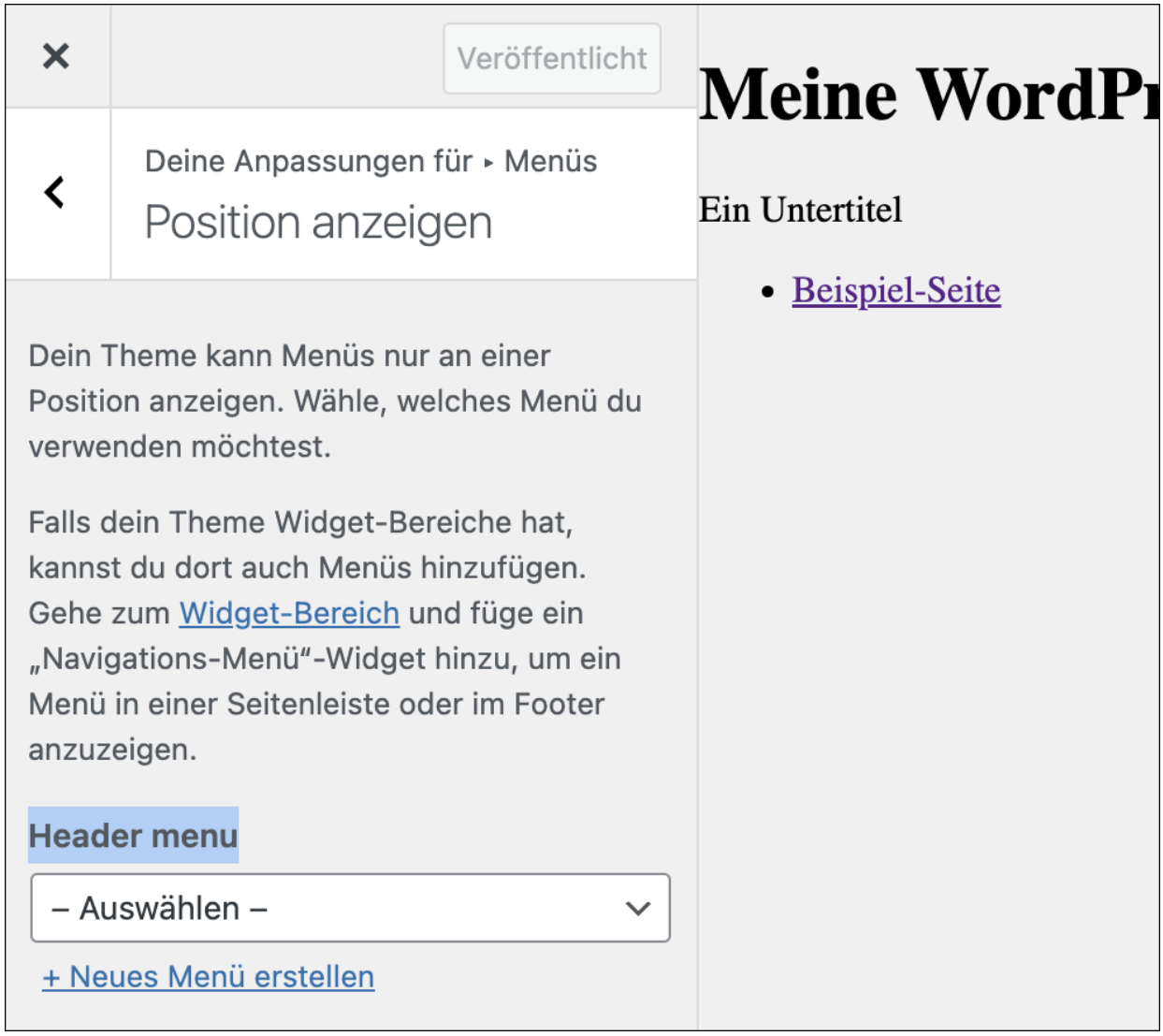


Abbildung 23.34 Die registrierte Menüposition ist in der Menüverwaltung auswählbar.



Abbildung 23.35 Das sieht doch gut aus (vom fehlenden Design abgesehen): Jetzt werden schon die Beiträge in der Blogübersicht angezeigt.

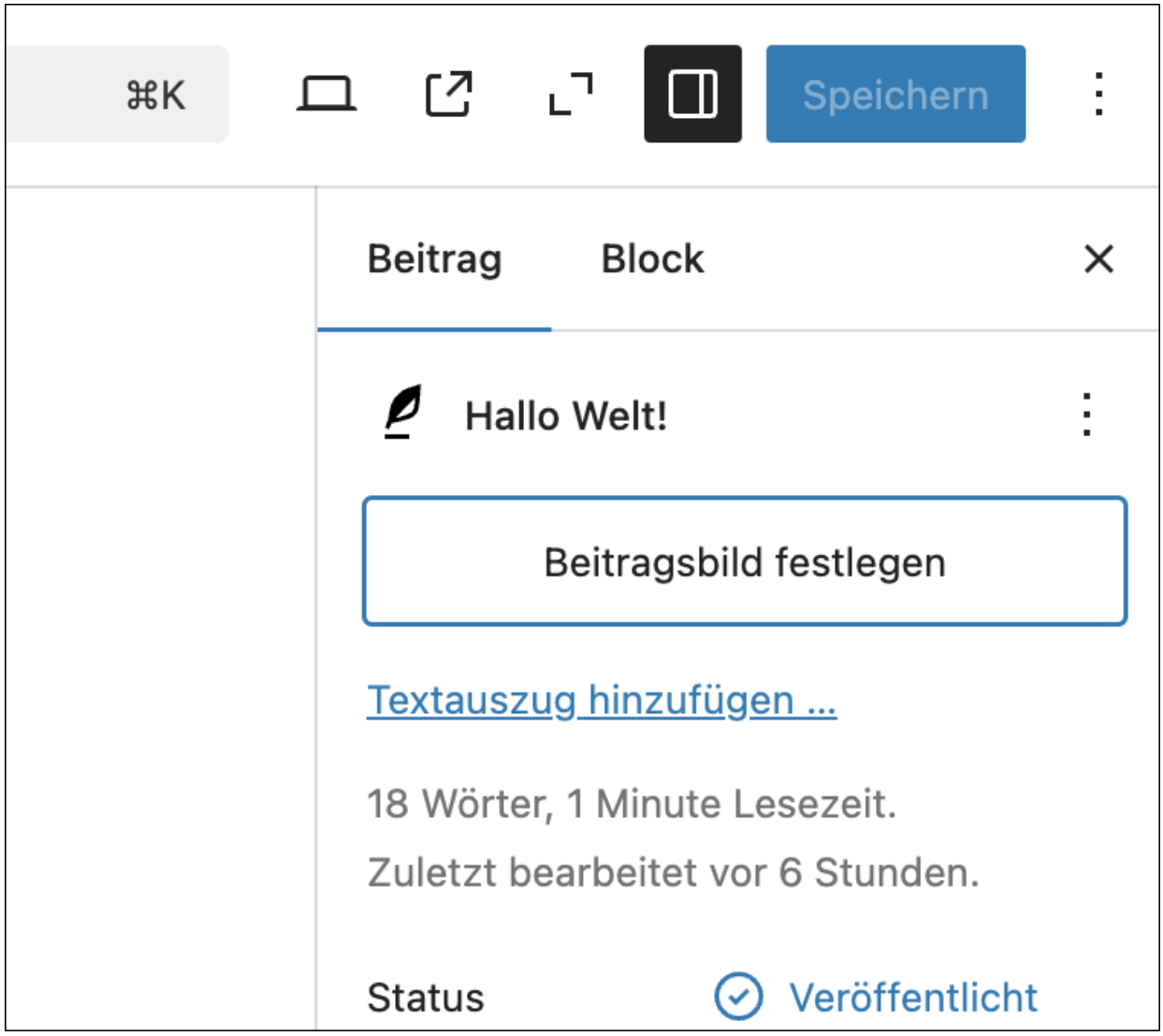


Abbildung 23.36 Wurde die Unterstützung aktiviert, ist die Beitragsbildfunktion im Editor zu sehen.

Meine WordPress-Website

Ein Untertitel

- Beispiel-Seite

Hallo Welt!

1. Oktober 2024

Willkommen bei WordPress. Dies ist dein erster Beitrag. Bearbeite oder lösche ihn und beginne mit dem Schreiben!

Categories: Allgemein

One reaction to »Hallo Welt!«

- **Ein WordPress-Kommentator** sagt:
  1. Oktober 2024 um 20:38 Uhr

  Hallo, dies ist ein Kommentar.
  Um mit dem Freischalten, Bearbeiten und Löschen von Kommentaren zu beginnen, besuche bitte die Kommentare-Ansicht im Dashboard.
  Die Avatare der Kommentatoren kommen von Gravatar.

  Antworten

Schreibe einen Kommentar

Deine E-Mail-Adresse wird nicht veröffentlicht. Erforderliche Felder sind mit * markiert

Kommentar *

Name *

E-Mail *

Website

☐ Meinen Namen, meine E-Mail-Adresse und meine Website in diesem Browser für die nächste Kommentierung speichern.

Submit Comment

Abbildung 23.37 Nach Erstellen der »singular.php«, des »content-single.php«-Template-Teils und der »comments.php« sieht die Einzelansicht des »Hallo-Welt!«-Beitrags ungefähr so aus.

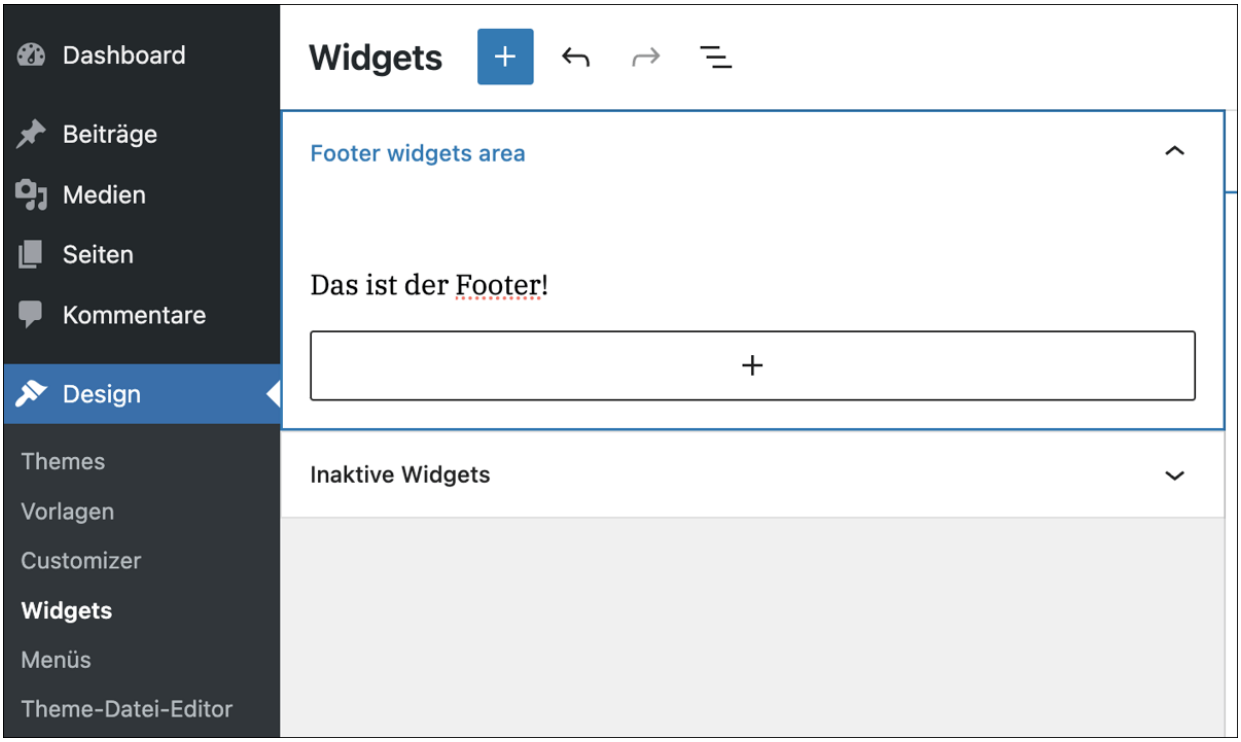


Abbildung 23.38 Der neu erstellte Widget-Bereich im Backend, hier bereits mit einem Absatz-Block befüllt



Abbildung 23.39 Das ist nicht nur ein Absatz, sondern ein Absatz in dem neu registrierten und im Footer eingebundenen Widget-Bereich.



Abbildung 23.40 Die Startseite im gestylten Gewand

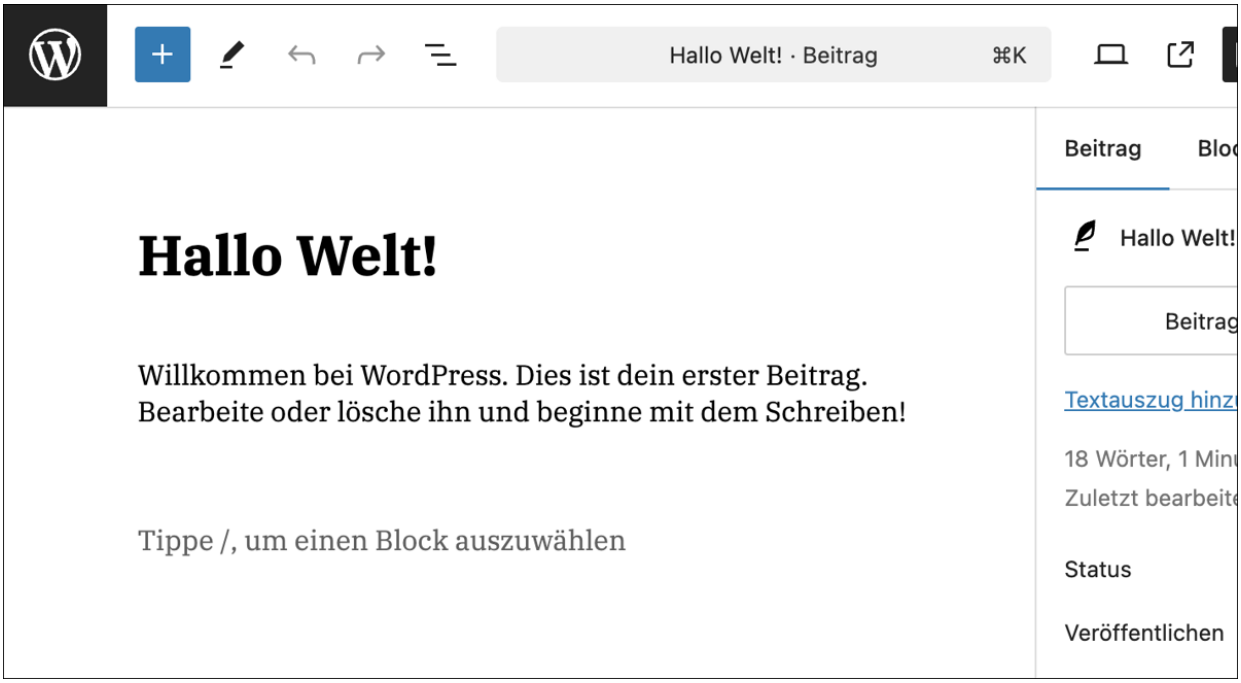


Abbildung 23.41 Jetzt ist auch die Editor-Ansicht zumindest ein bisschen an das Theme angepasst.



Abbildung 23.42 Nach dem Hinzufügen der neuen Größe zu den auswählbaren Bildgrößen taucht sie in der Bild-Block-Option zum Setzen der Größe auf.

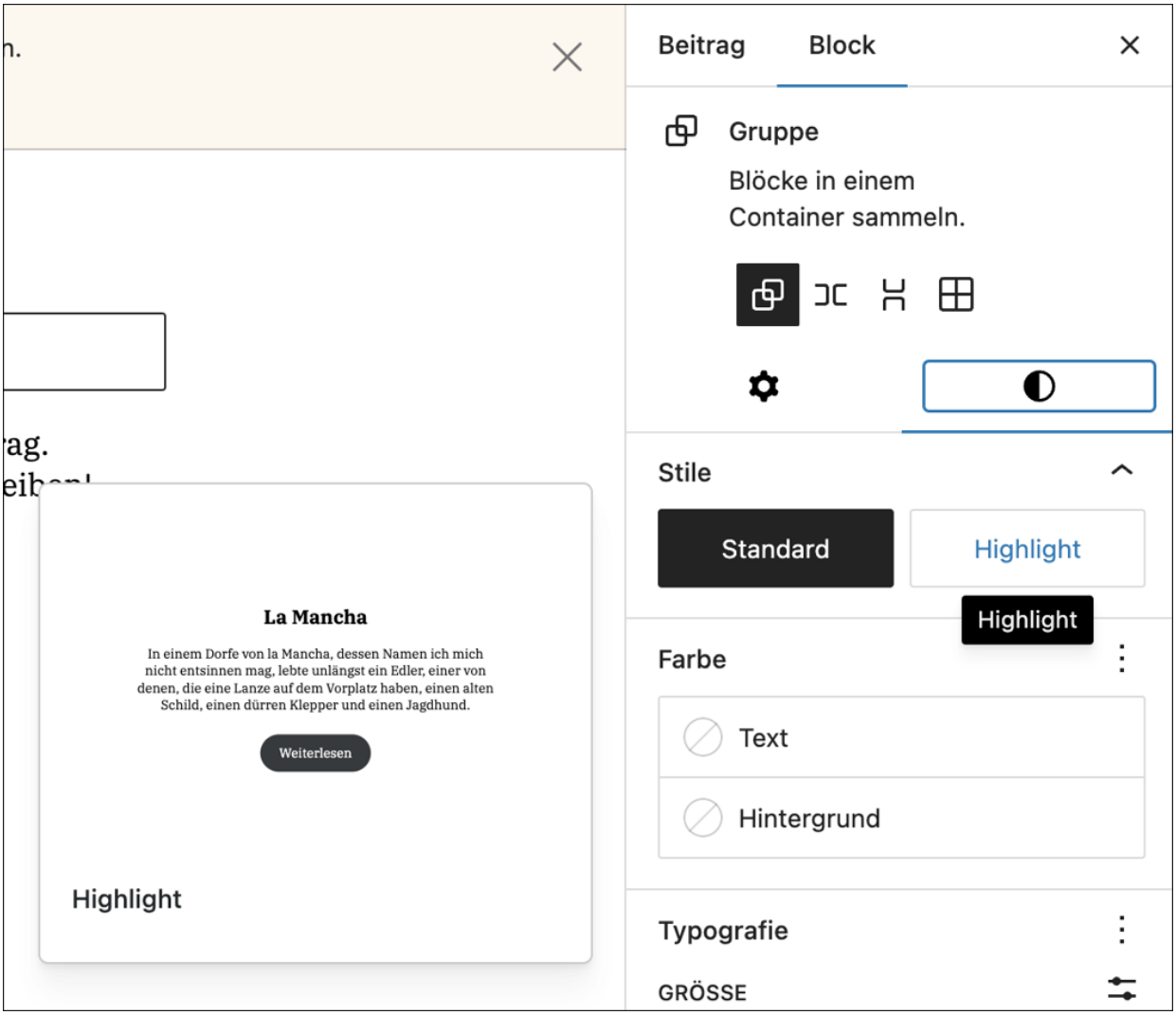


Abbildung 23.43 In der Block-Sidebar kann der neue Stil ausgewählt werden.

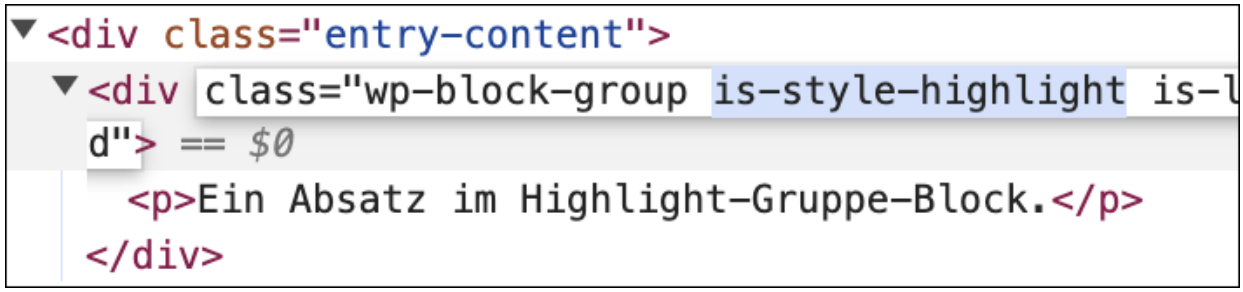


Abbildung 23.44 Die HTML-Klasse für Stile wird aus dem Präfix »is-style-« und dem »name« des Stils gebildet.

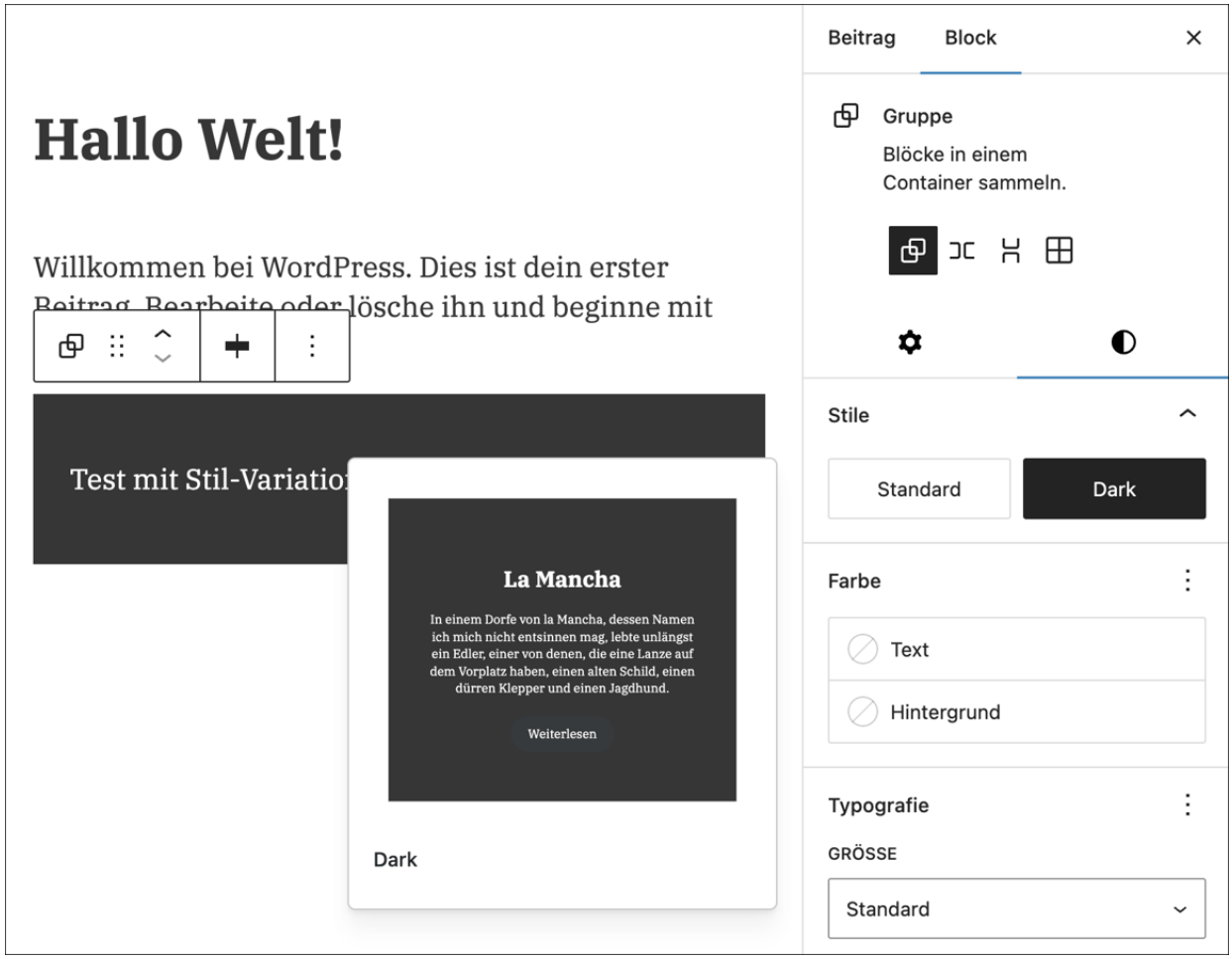
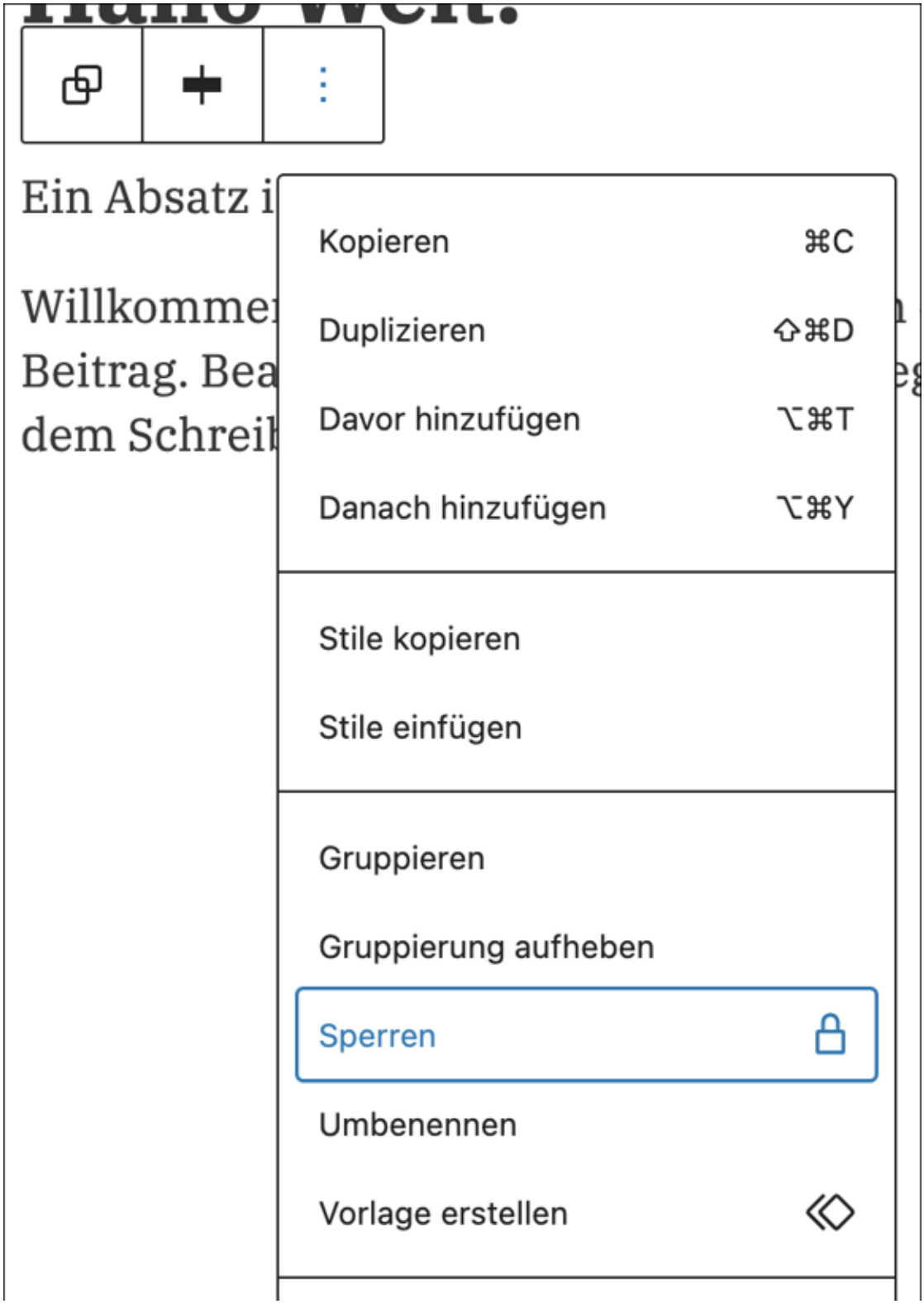


Abbildung 23.45 Die neue Variation inklusive Styles im Editor


Hallo Welt!
Ein Absatz i
Willkomme
Beitrag. Bea
dem Schreib
Kopieren ⌘C
Duplizieren ⇧⌘D
Davor hinzufügen ⌥⌘T
Danach hinzufügen ⌥⌘Y
Stile kopieren
Stile einfügen
Gruppieren
Gruppierung aufheben
Sperren
Umbenennen
Vorlage erstellen

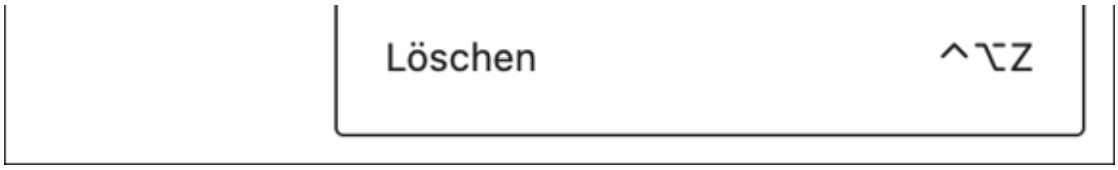


Abbildung 23.46 Über das Drei-Punkte-Menü eines Blocks können im Editor unterschiedliche Optionen zum Sperren genutzt werden.

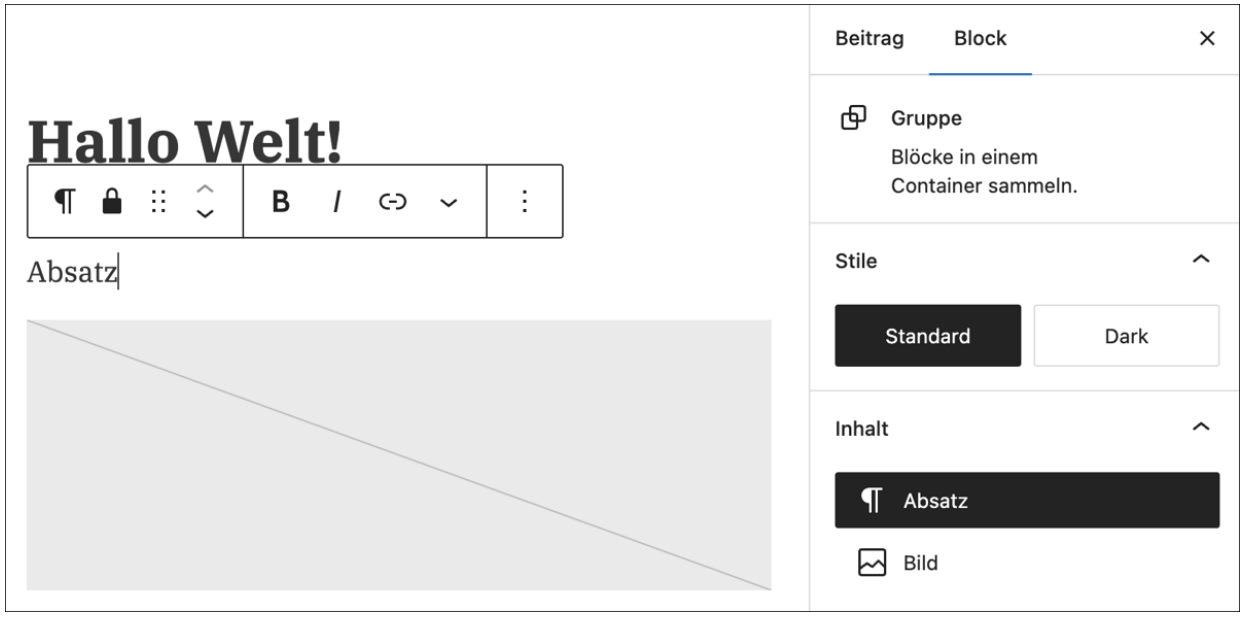


Abbildung 23.47 Bei dem Gruppe-Block kann nur noch der Inhalt verändert werden. Die Einstellungen des Absatz- und Bild-Blocks sind nicht mehr zugänglich.

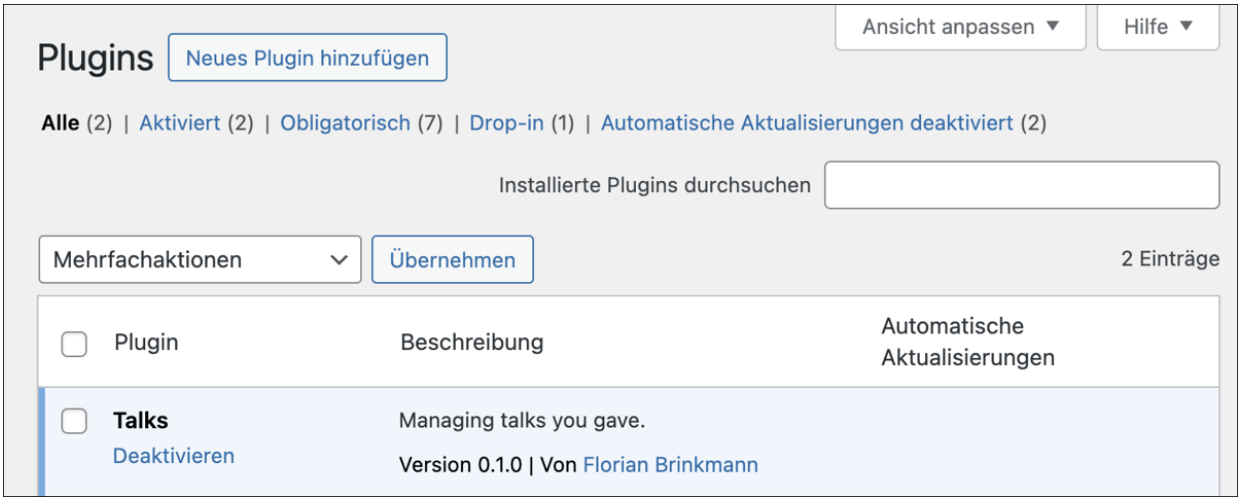


Abbildung 24.1 Mehr als ein PHP-Kommentar mit Metaangaben ist für ein WordPress-Plugin nicht notwendig.

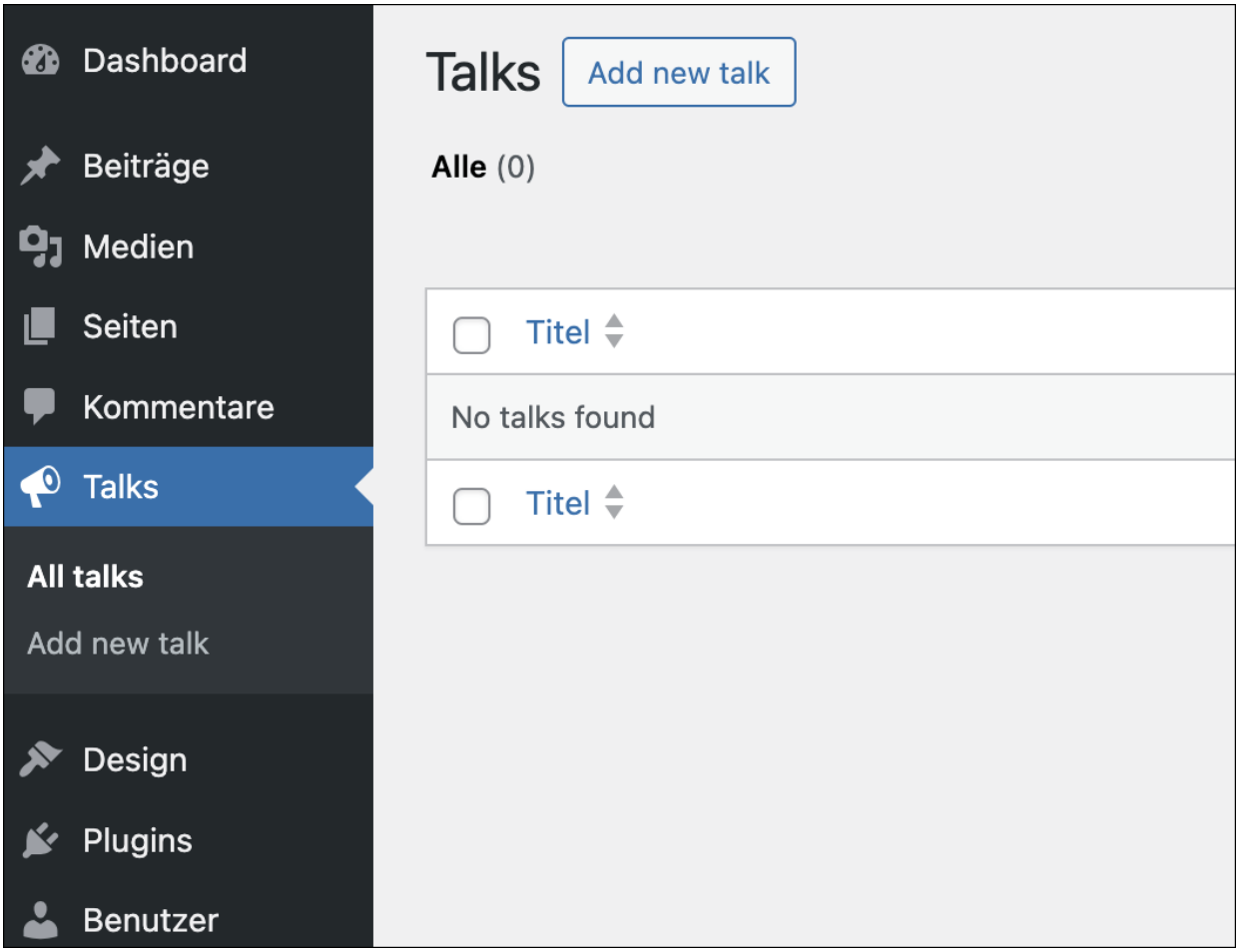


Abbildung 24.2 Nach der Registrierung des Inhaltstyps können Vorträge erstellt werden.

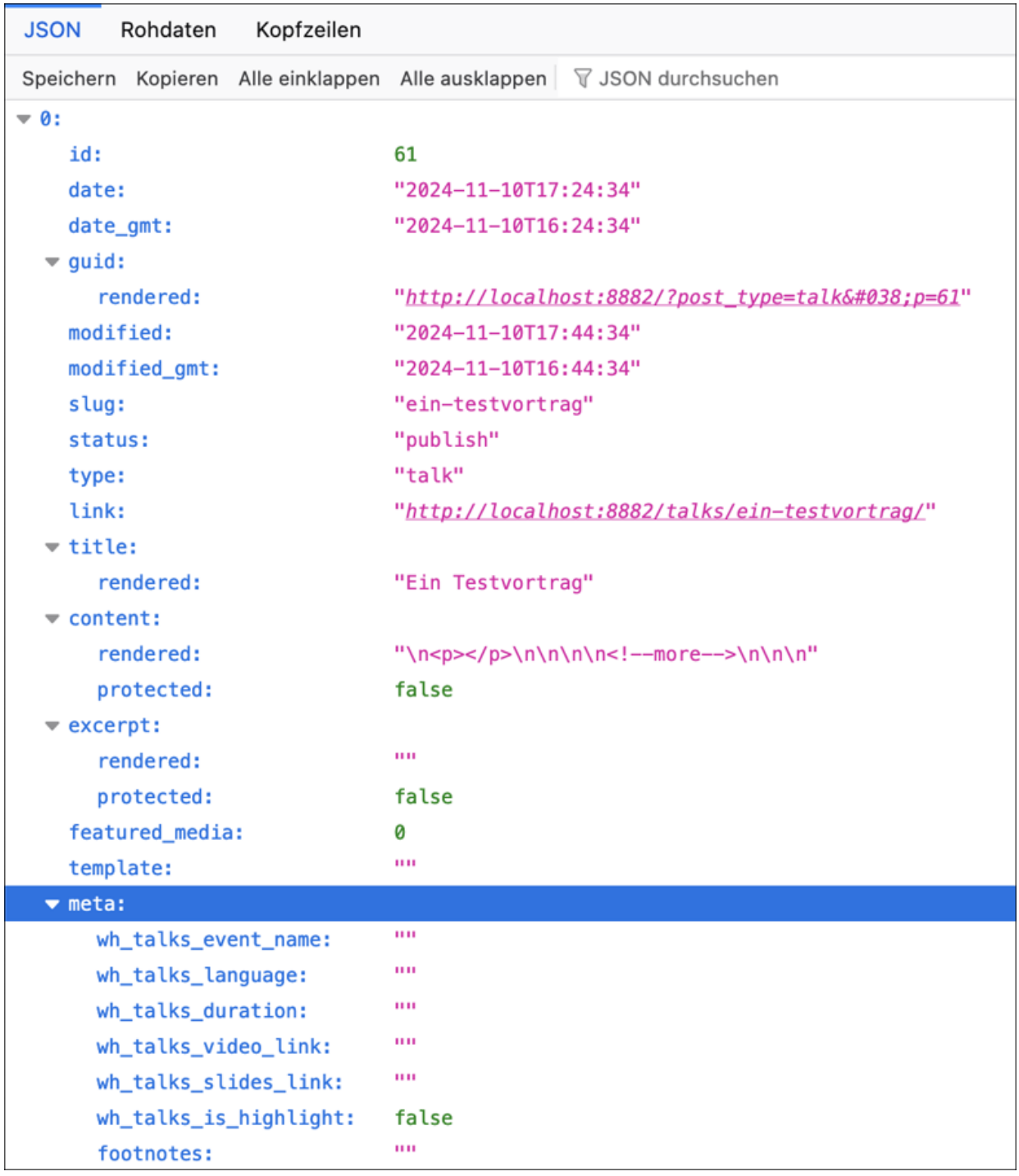

JSON
Rohdaten
Kopfzeilen
Speichern
Kopieren
Alle einklappen
Alle ausklappen
JSON durchsuchen
0:
id: 61
date: "2024-11-10T17:24:34"
date_gmt: "2024-11-10T16:24:34"
guid:
rendered: "http://localhost:8882/?post_type=talk&p=61"
modified: "2024-11-10T17:44:34"
modified_gmt: "2024-11-10T16:44:34"
slug: "ein-testvortrag"
status: "publish"
type: "talk"
link: "http://localhost:8882/talks/ein-testvortrag/"
title:
rendered: "Ein Testvortrag"
content:
rendered: "\n \n\n\n\n<!--more-->\n\n\n"
protected: false
excerpt:
rendered: ""
protected: false
featured_media: 0
template: ""
meta:
wh_talks_event_name: ""
wh_talks_language: ""
wh_talks_duration: ""
wh_talks_video_link: ""
wh_talks_slides_link: ""
wh_talks_is_highlight: false
footnotes: ""

Abbildung 24.3 In der REST-API-Antwort für einen Vortrag sind bereits die registrierten Post-Meta-Einträge zu sehen (Screenshot aus dem Firefox-Browser, der JSON-Antworten etwas schöner darstellt als beispielsweise Chrome).

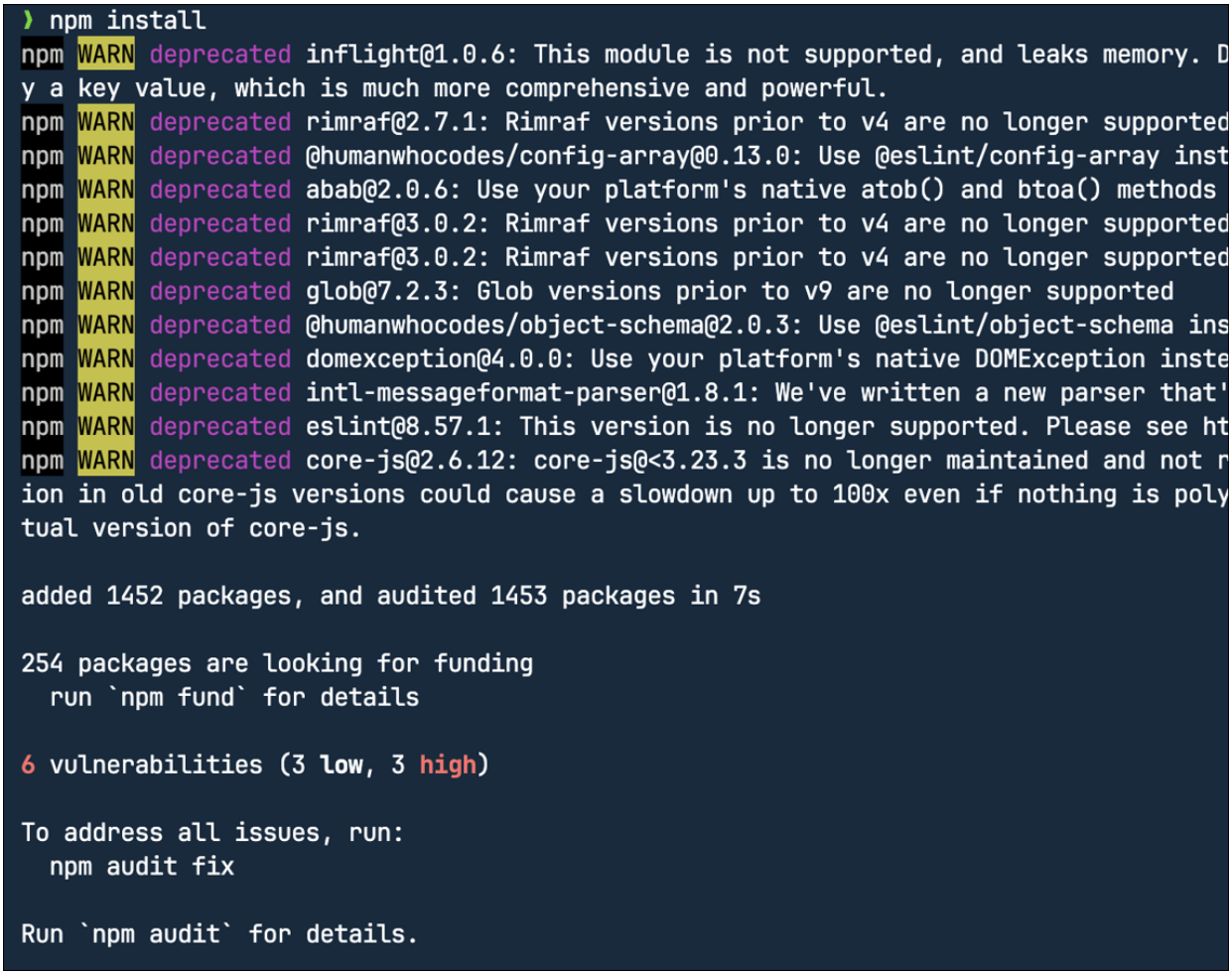
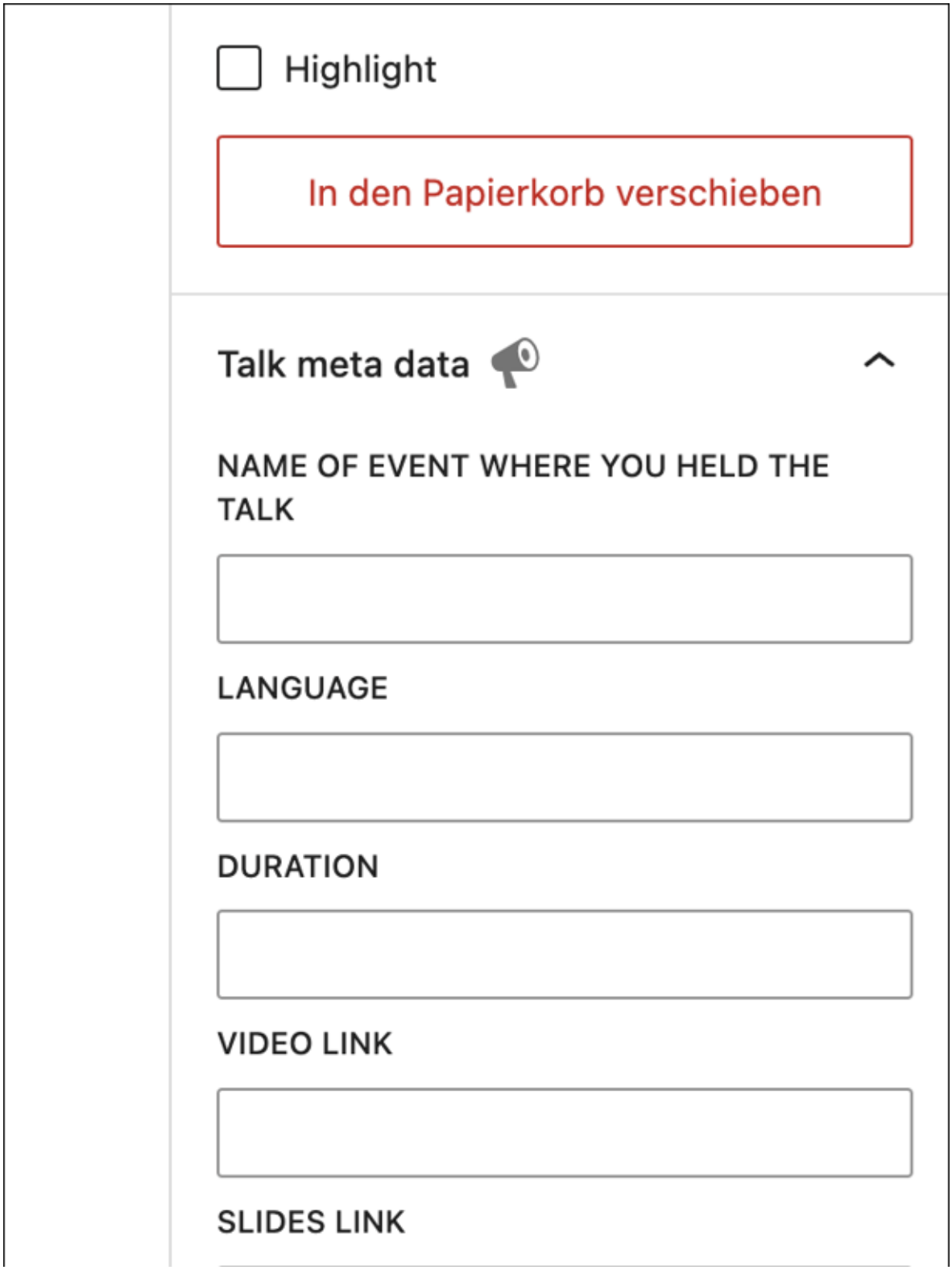


Abbildung 24.4 Nach dem Ausführen von »npm install« werden einige Meldungen und Warnungen angezeigt, aber die Installation läuft erfolgreich durch.


Highlight
In den Papierkorb verschieben
Talk meta data
NAME OF EVENT WHERE YOU HELD THE TALK
LANGUAGE
DURATION
VIDEO LINK
SLIDES LINK

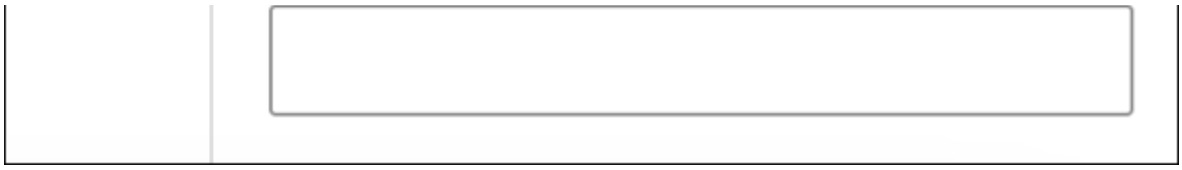


Abbildung 24.5 Das neue Panel in der Seitenleiste eines Talk-Eintrags für die Änderung der Metadaten

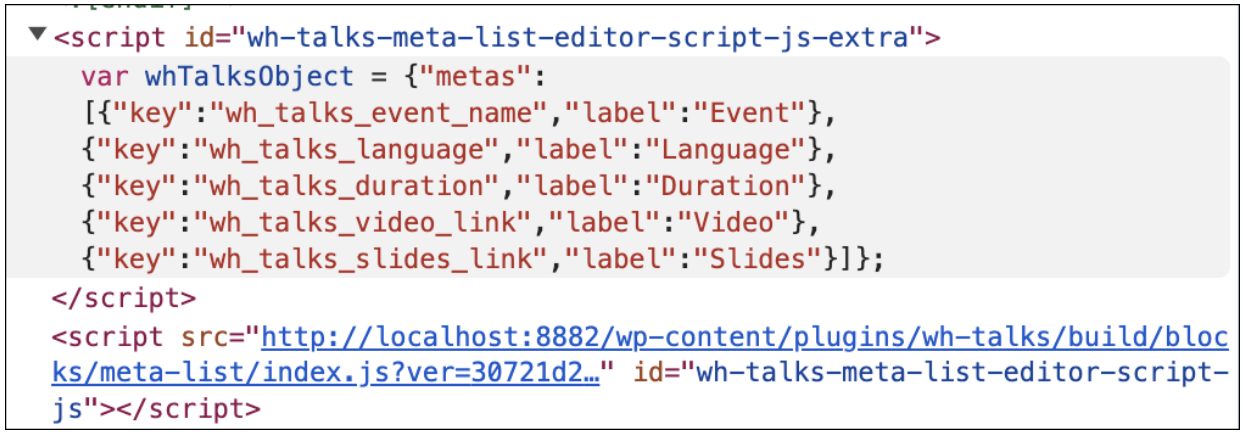


Abbildung 24.6 »wp_localize_script()« fügt ein Inline Script vor dem Script ein, dessen Handle als erster Parameter übergeben wird.

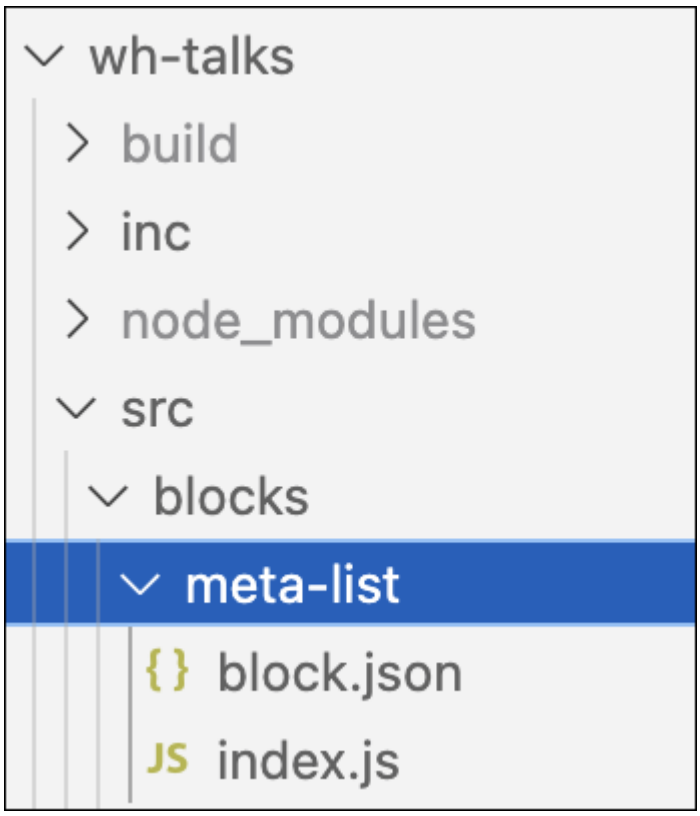


Abbildung 24.7 Die Verzeichnisstruktur für den ersten Block

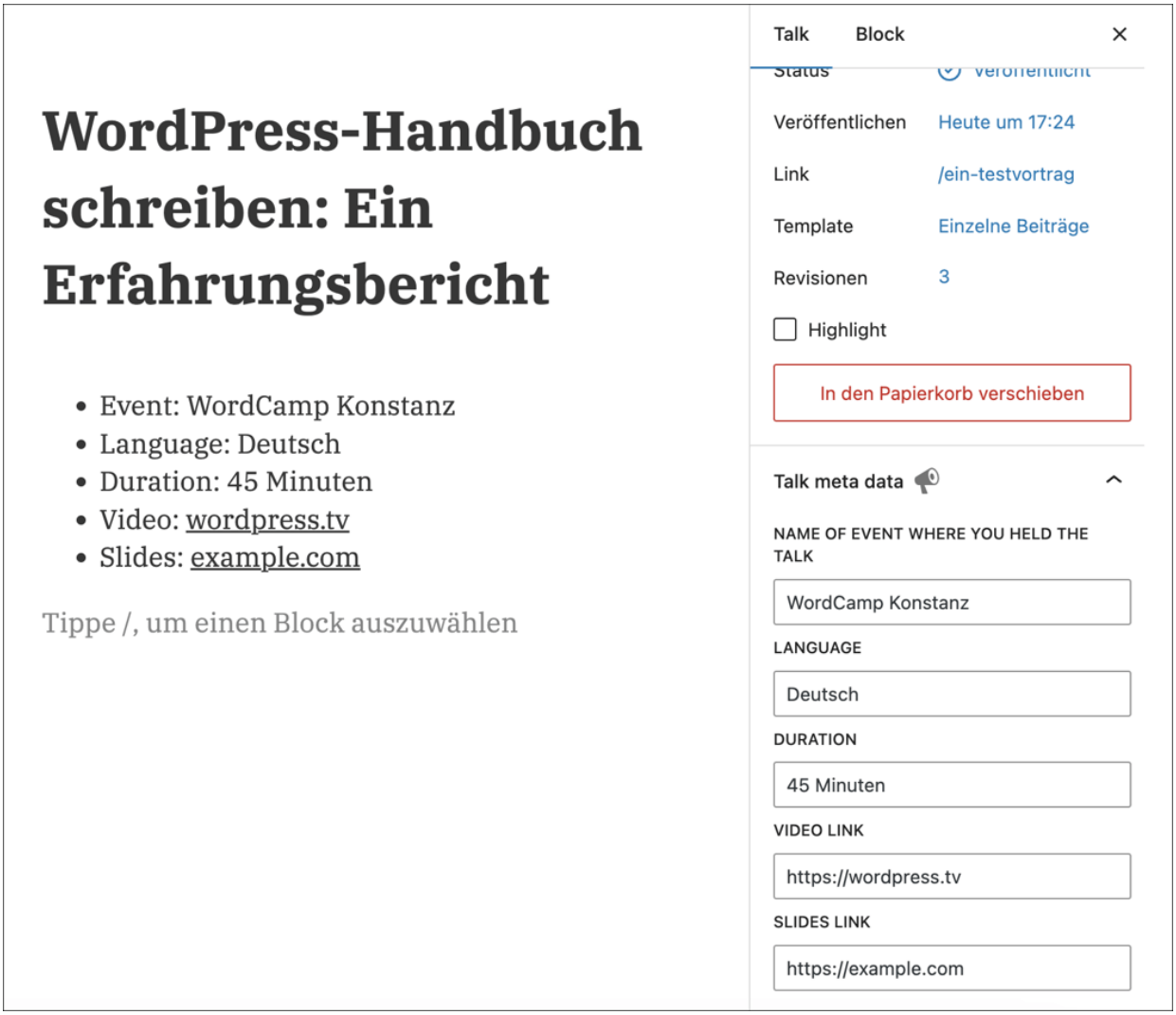


Abbildung 24.8 Der Block zur Darstellung der Vortrags-Metawerte in Aktion



Abbildung 24.9 Der Metadaten-Liste-Block im Frontend

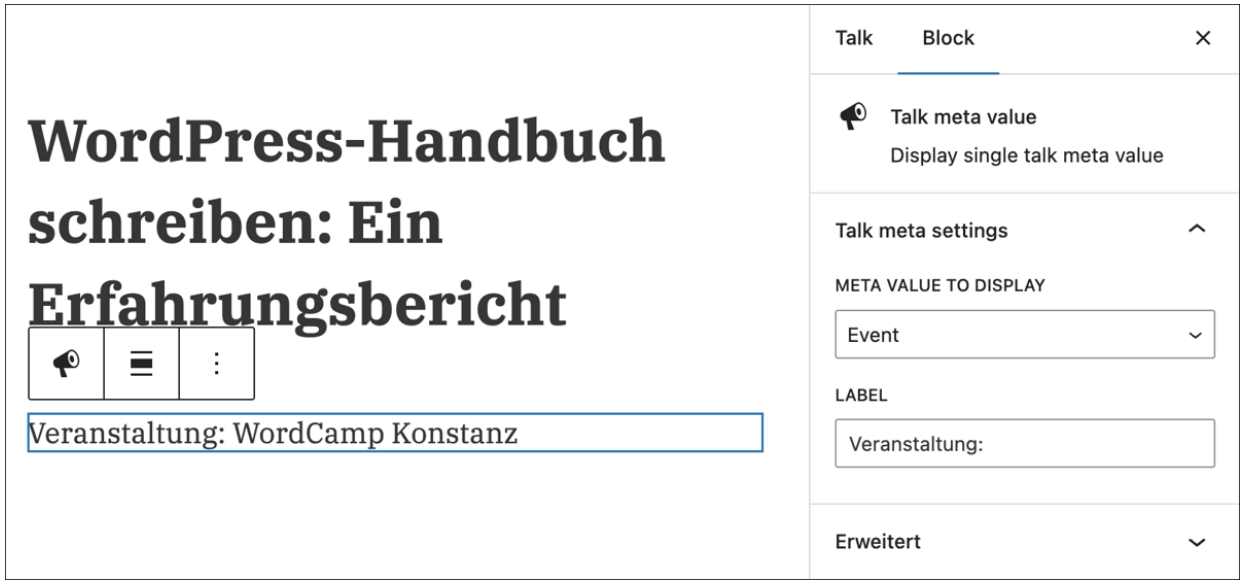


Abbildung 24.10 Nachdem Sie den »Talk meta value«-Block eingefügt haben, finden Sie rechts in der Seitenleiste die Optionen.



Abbildung 24.11 Der neue Block im Frontend

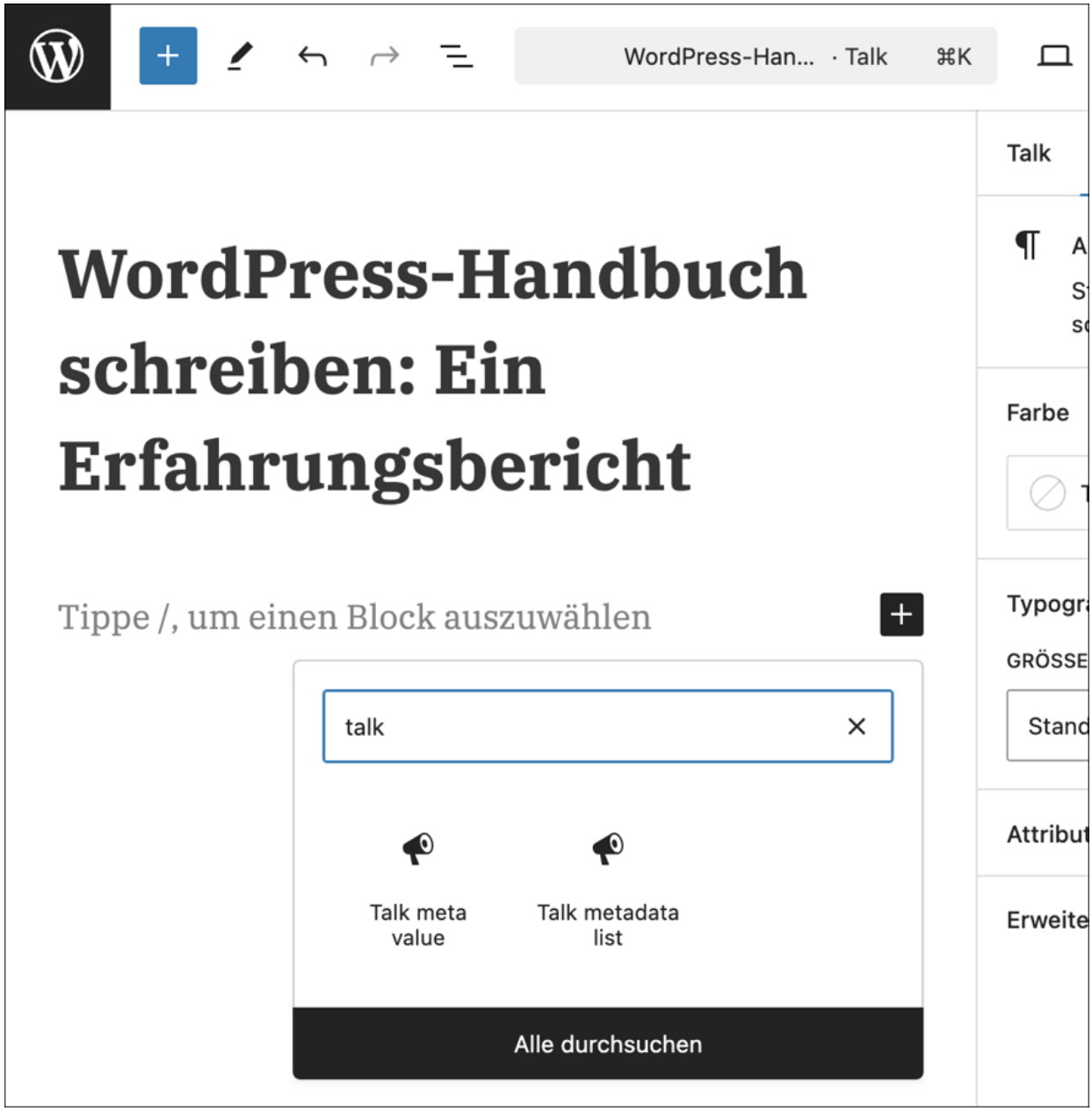


Abbildung 24.12 In der Bearbeitungsansicht eines Vortrags sind die beiden Blöcke weiter verfügbar.

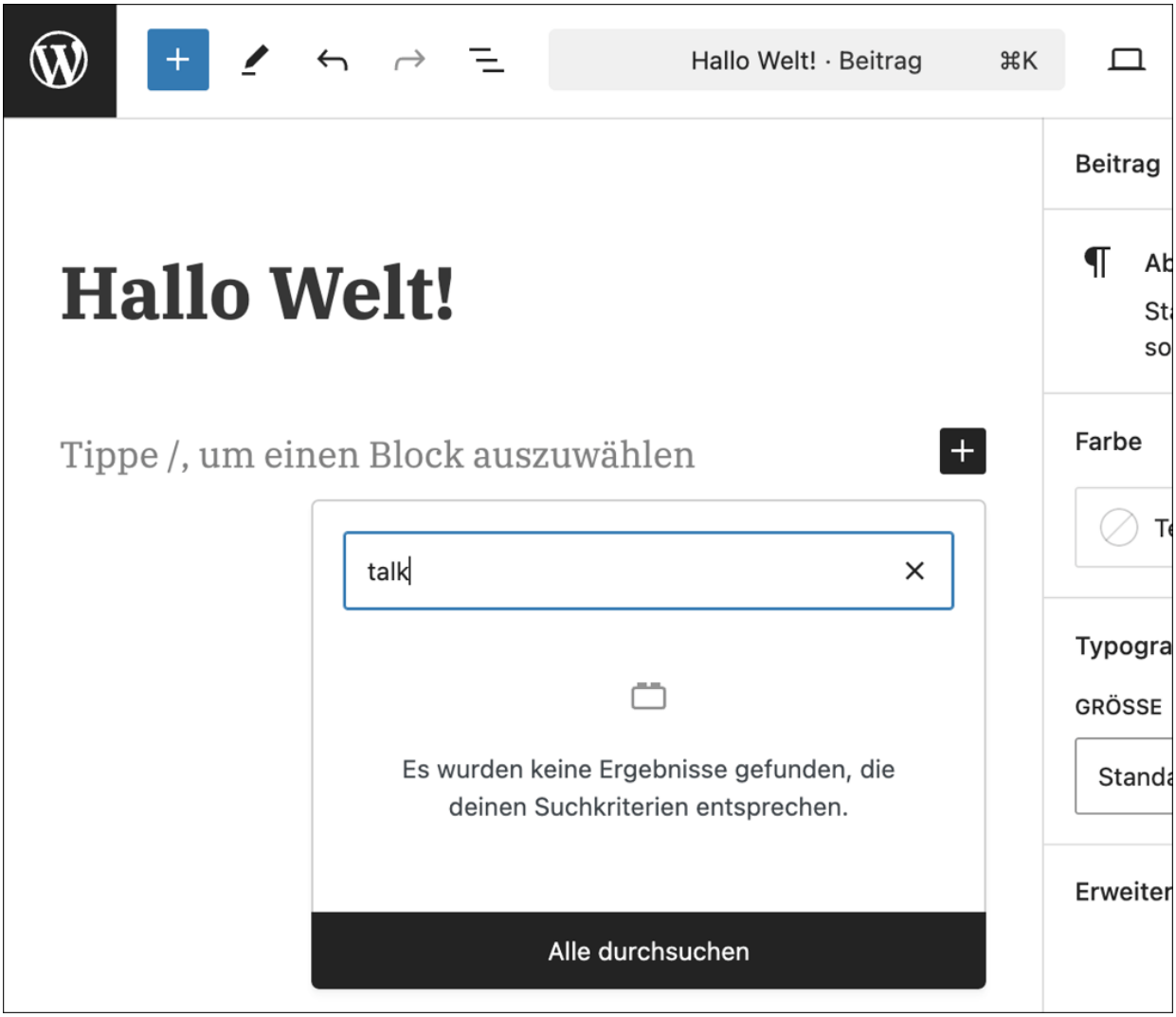


Abbildung 24.13 In einen Beitrag können die Blöcke nicht mehr eingefügt werden.



Abbildung 24.14 So soll das Ergebnis am Ende aussehen: eine Checkbox, mit der ein Vortrag als Highlight markiert werden kann.

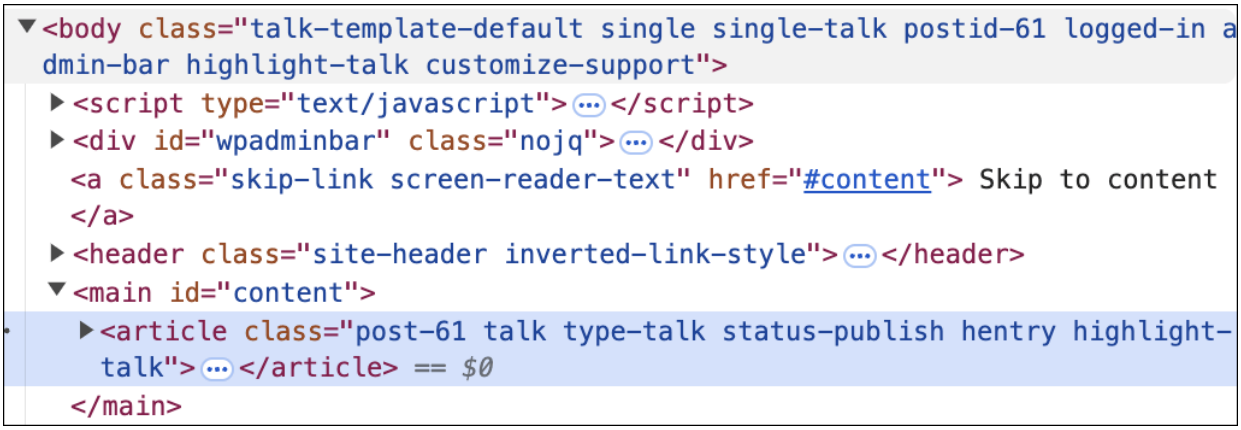


Abbildung 24.15 Ganz am Ende der Klassen im »class«-Attribut des »body«-Elements können Sie die Klasse »highlight-talk« sehen.

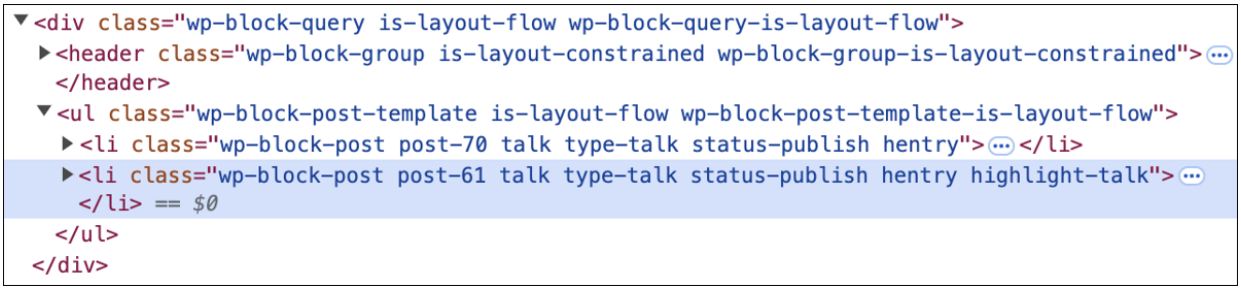


Abbildung 24.16 In der Übersicht der Talks wird im Abfrage-Loop-Block für das Wrapper-Element des hervorgehobenen Beitrags am Ende die Klasse eingefügt.

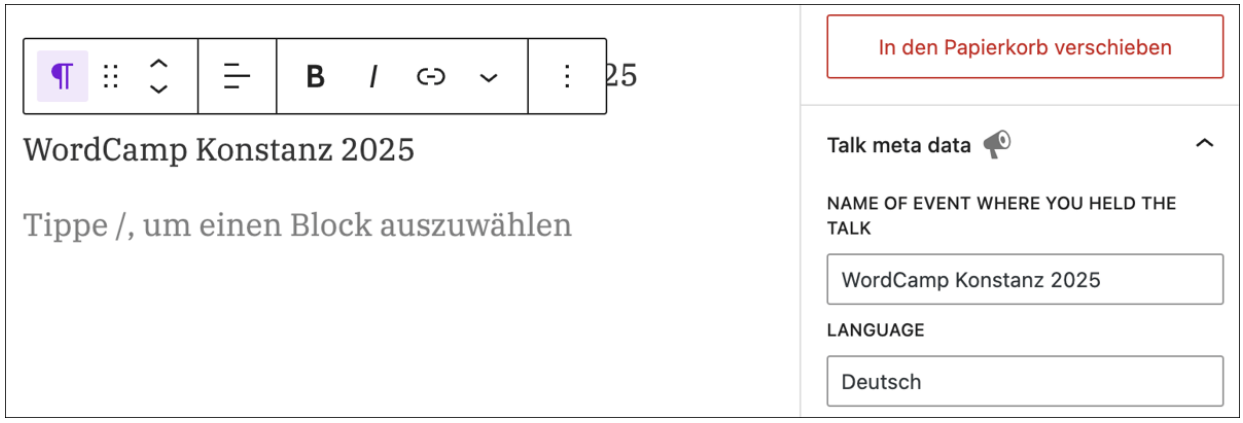


Abbildung 24.17 Der Wert des Absatz-Blocks spiegelt den Meta-Wert wider, und Sie können den Metawert sogar über den Absatzblock bearbeiten.

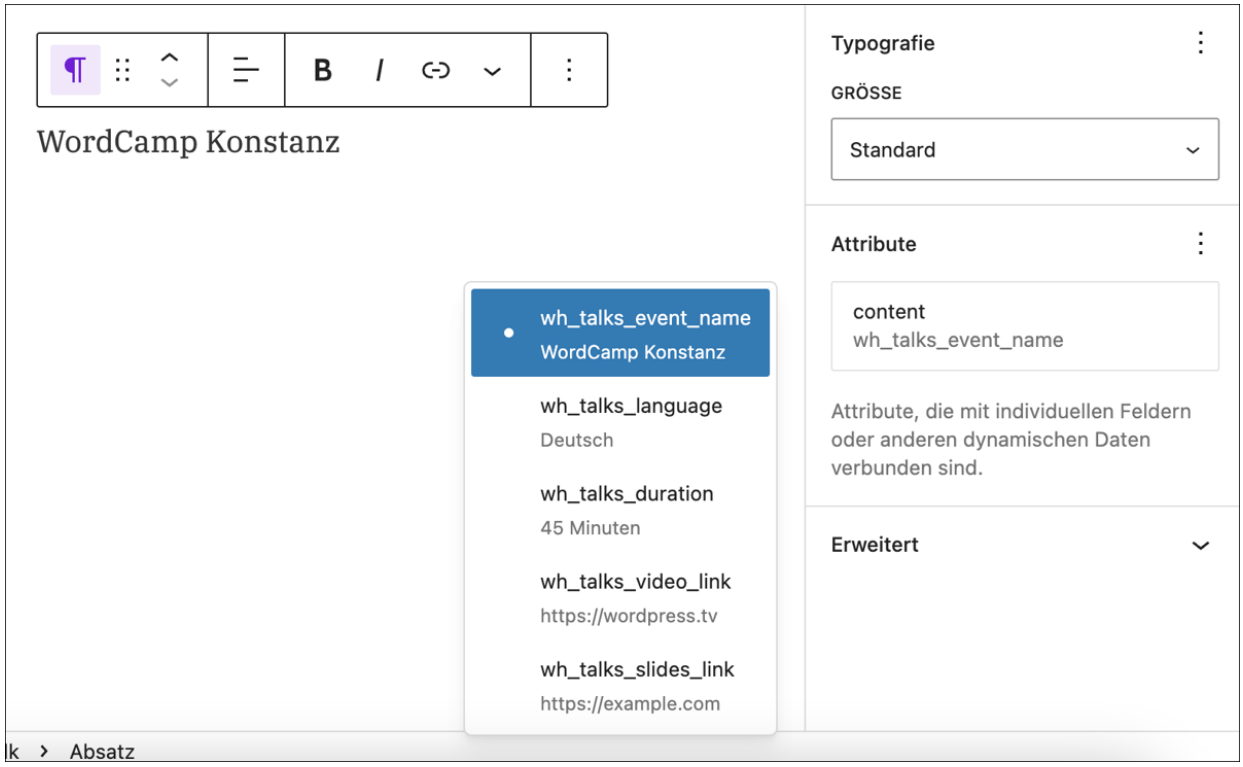


Abbildung 24.18 Seit WordPress 6.7 kann die Block Bindings API über ein User-Interface genutzt werden.

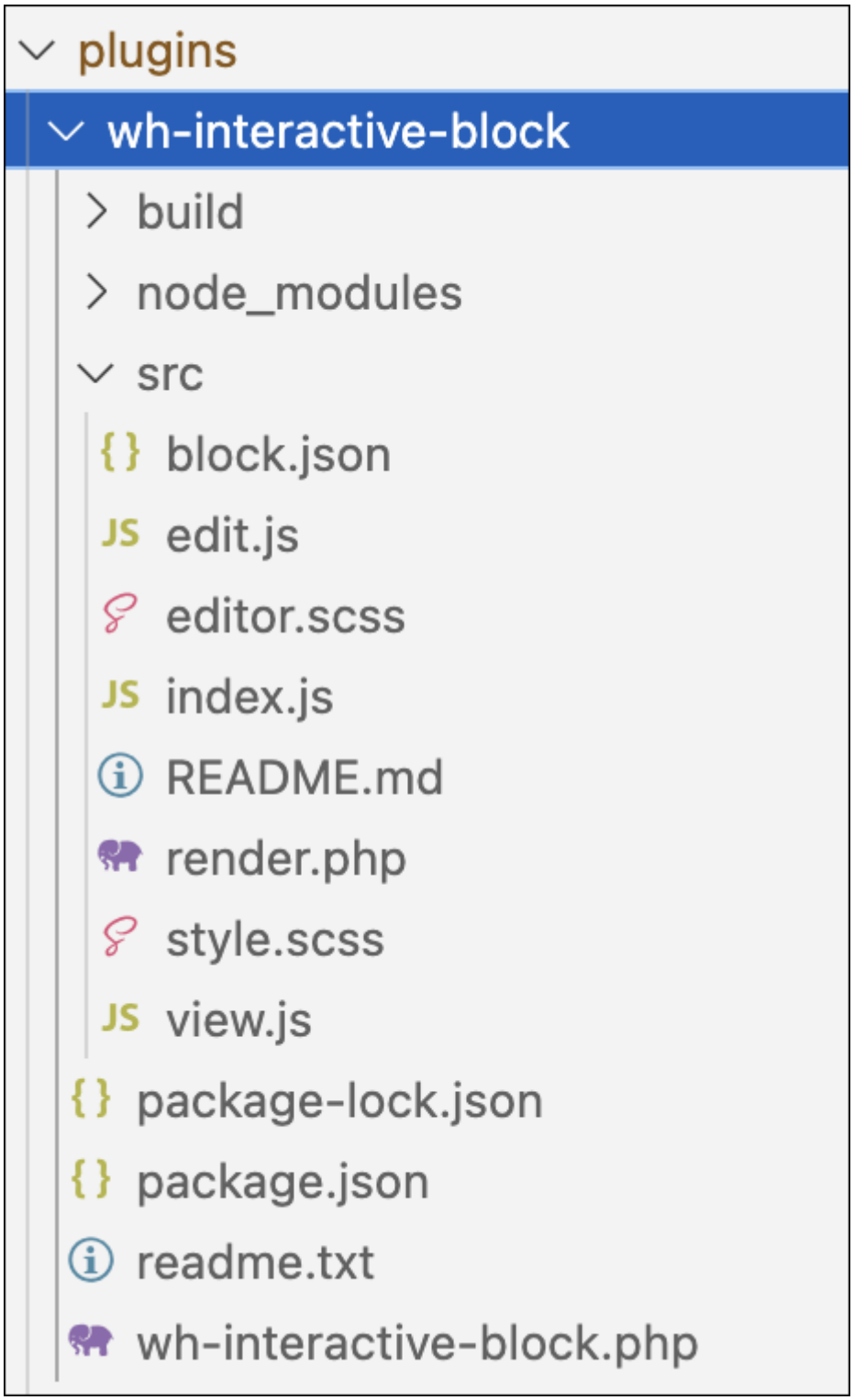

plugins
wh-interactive-block
build
node_modules
src
block.json
edit.js
editor.scss
index.js
README.md
render.php
style.scss
view.js
package-lock.json
package.json
readme.txt
wh-interactive-block.php

Abbildung 24.19 Nach Ausführen des Befehls haben Sie ein kleines fertiges Plugin im Ordner »wh-interactive-block«.

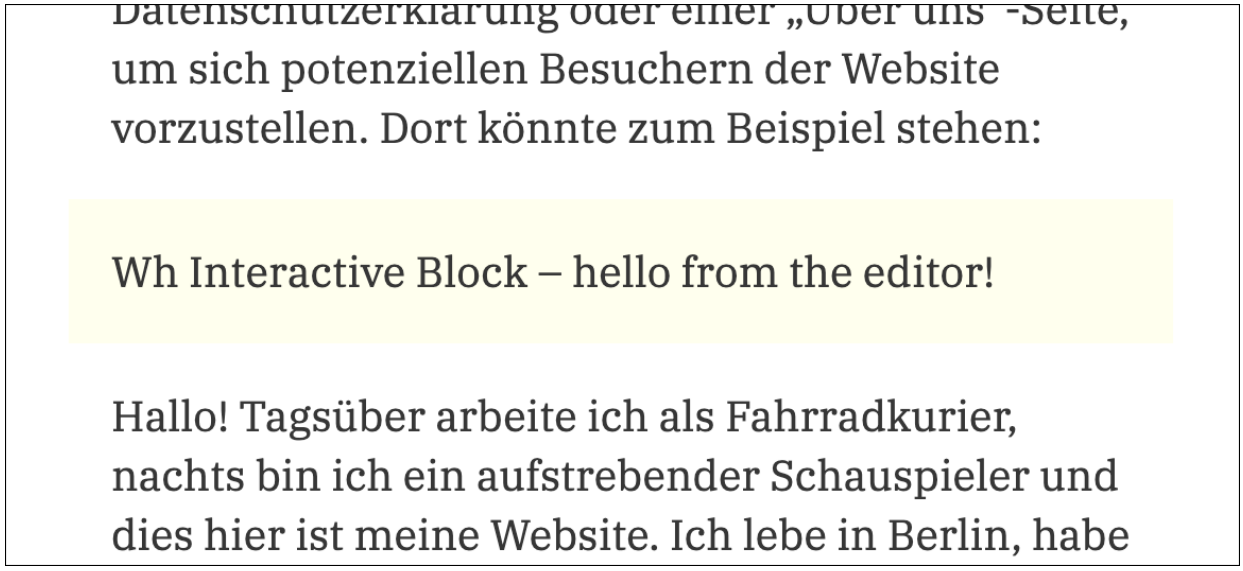


Abbildung 24.20 Im Backend ist der Block nur ein Platzhalter.



Abbildung 24.21 Im Frontend kommt der interessante, der interaktive Teil.

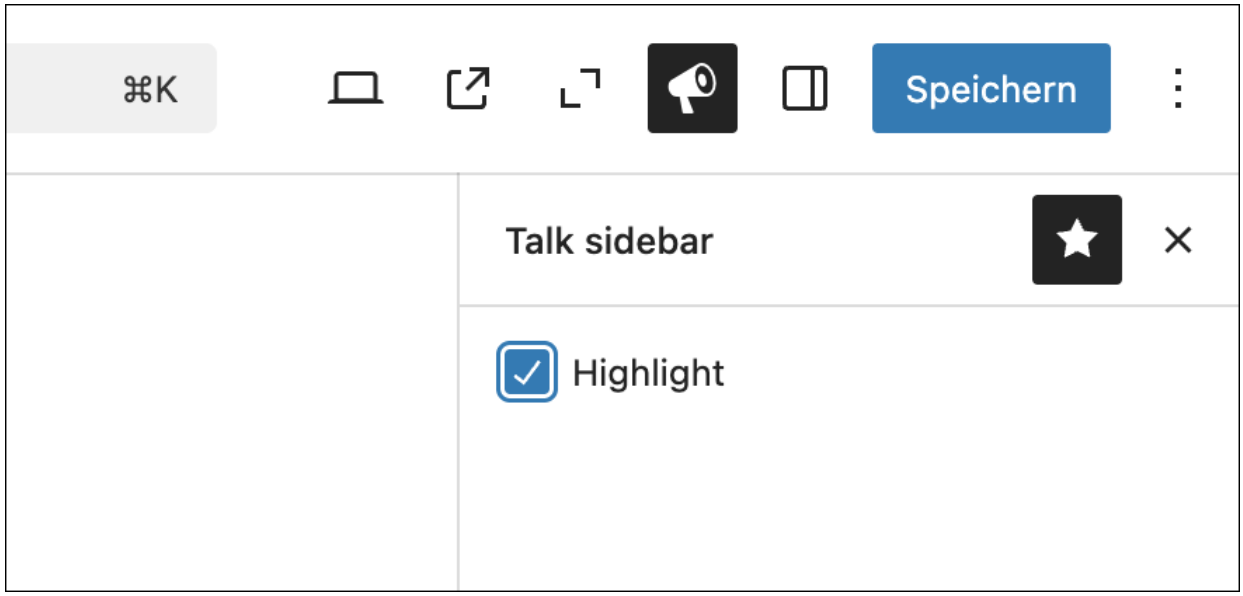


Abbildung 24.22 Eigene Seitenleiste in der Bearbeitungsansicht eines Vortrags

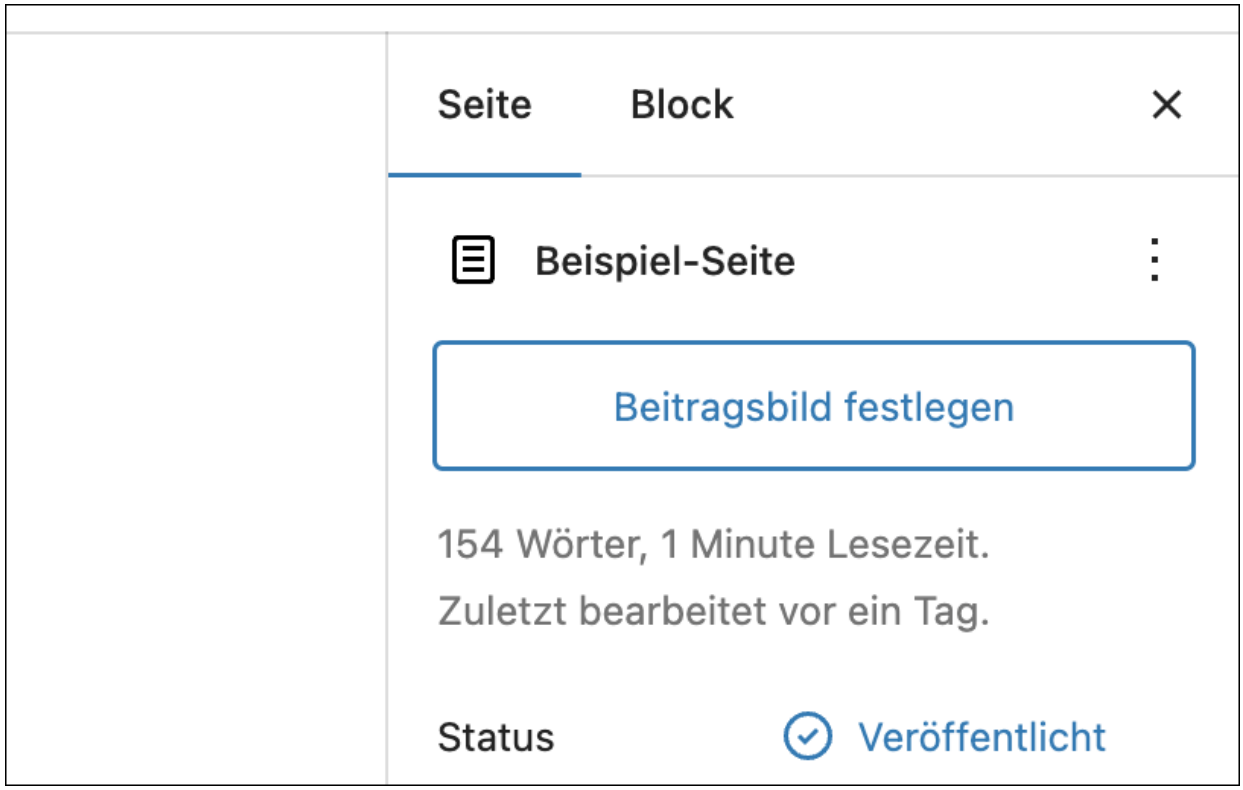


Abbildung 24.23 Die Editor-Sidebar für Seiten vor der Anpassung mit der Option für das Beitragsbild

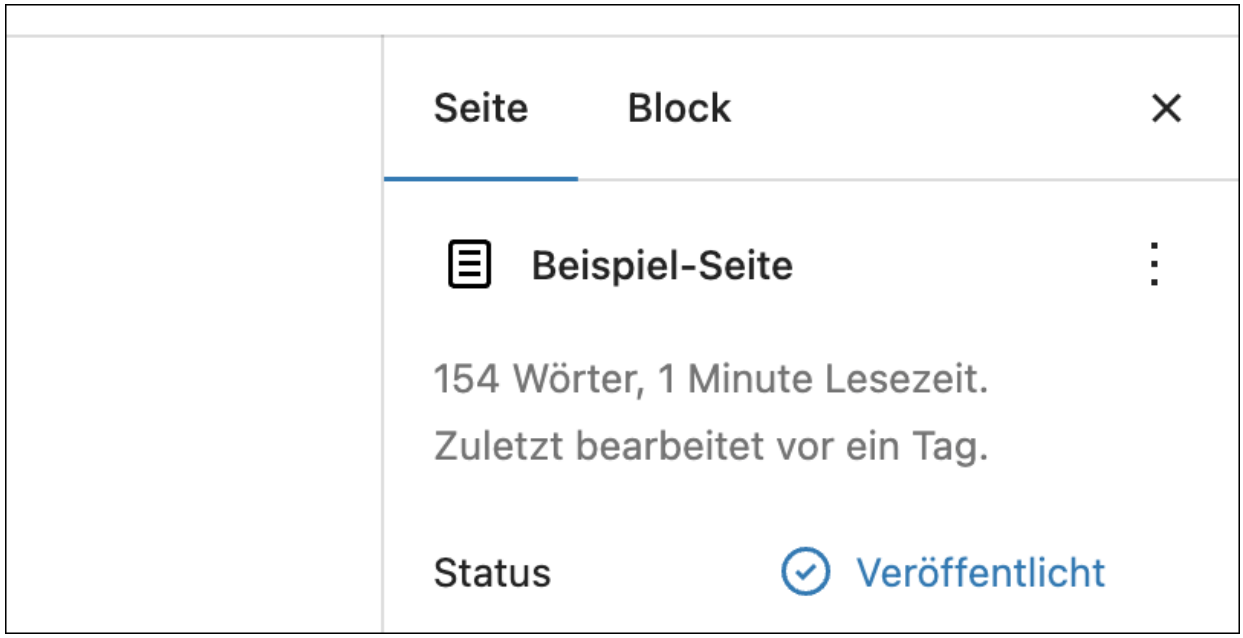


Abbildung 24.24 Nach der Anpassung wird die Beitragsbild-Option nicht mehr angezeigt.

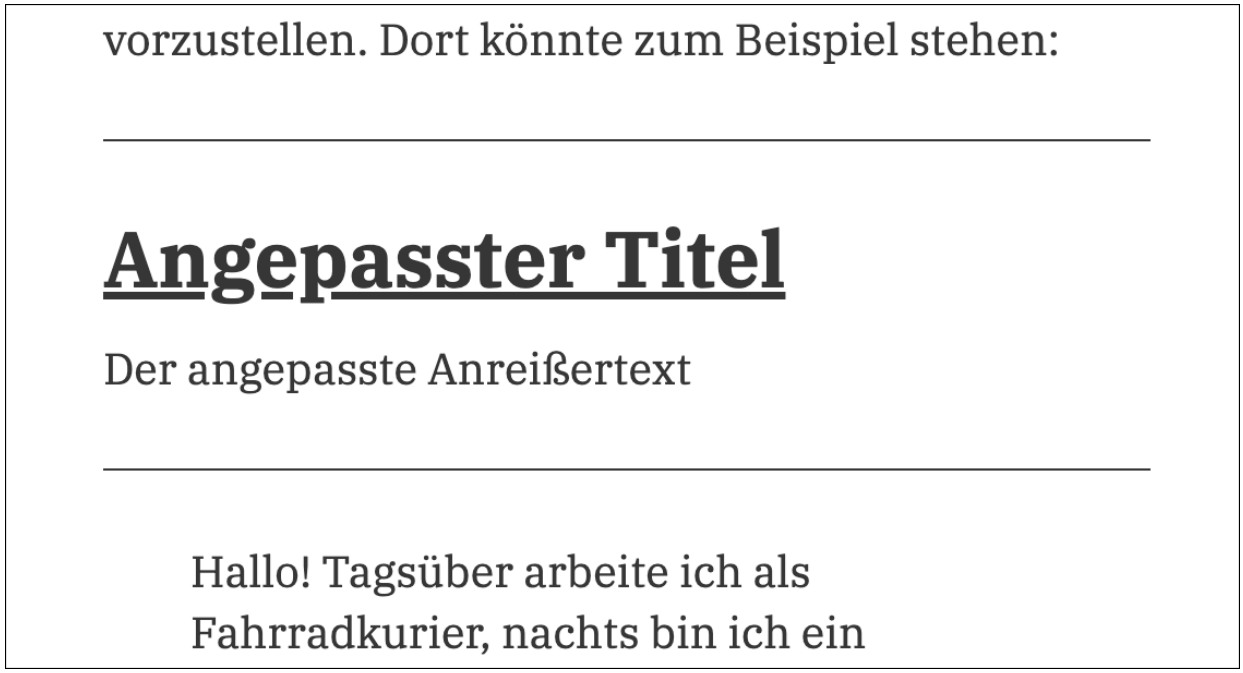


Abbildung 24.25 Ein Beitrag mit einem Post-Teaser für den Hallo-Welt-Beitrag, zur besseren Unterscheidung mit einem visuellen Trenner oben und unten

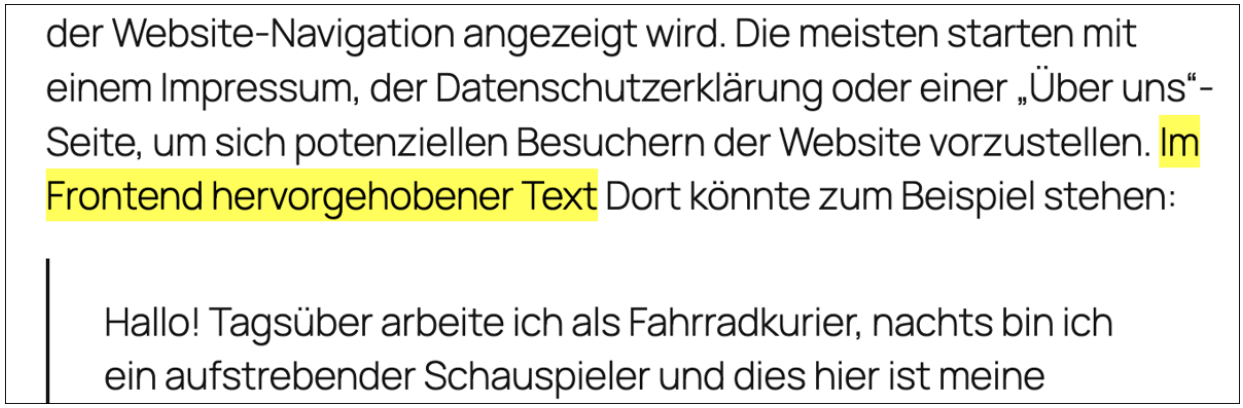


Abbildung 24.26 Nicht schön, aber deutlich hervorgehoben: das Ergebnis des Shortcodes im Frontend

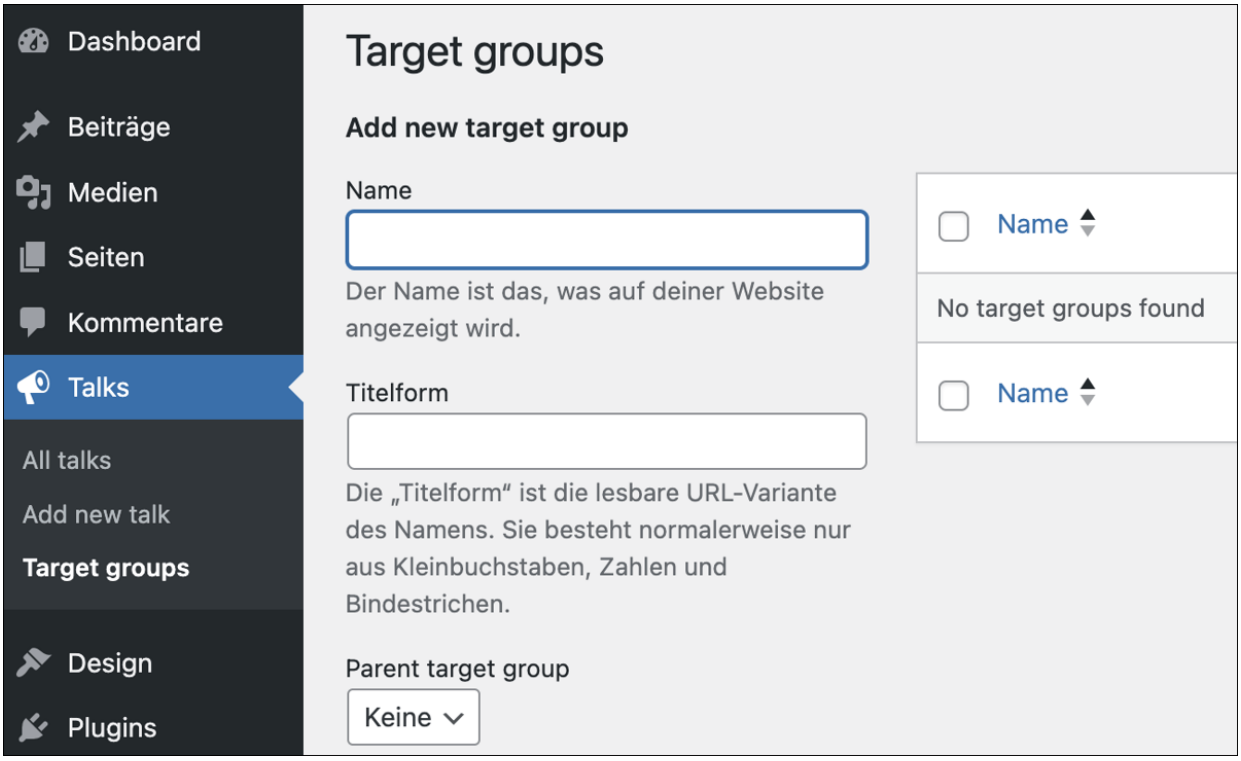


Abbildung 24.27 Die neue Taxonomie im Backend

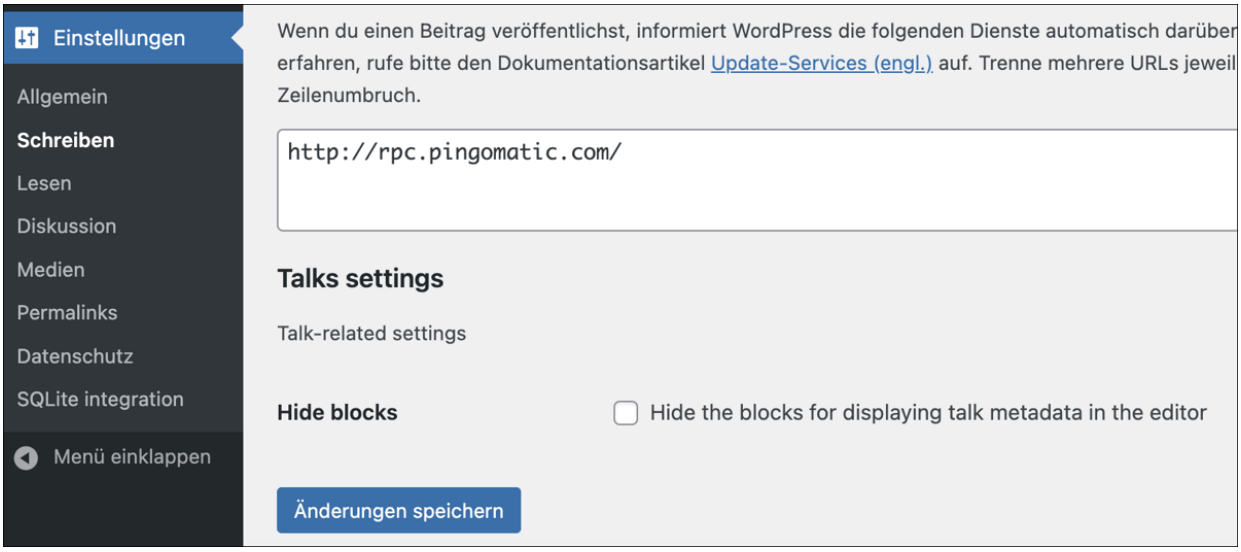


Abbildung 24.28 Die neue Option auf der bestehenden Einstellungsseite

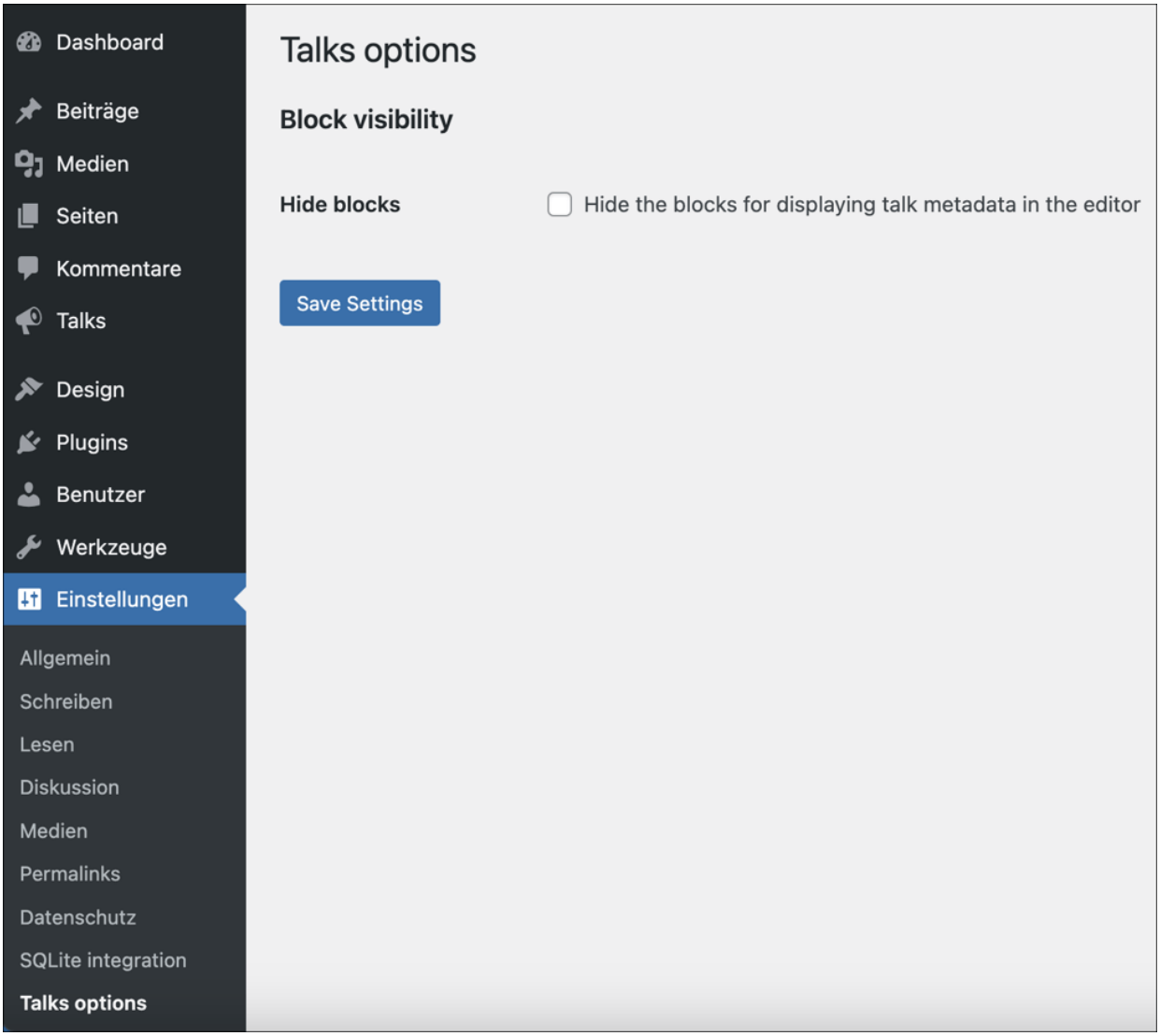


Abbildung 24.29 Eine eigene Seite bietet sich an, wenn ein paar mehr Optionen angezeigt werden sollen als in diesem Beispiel.

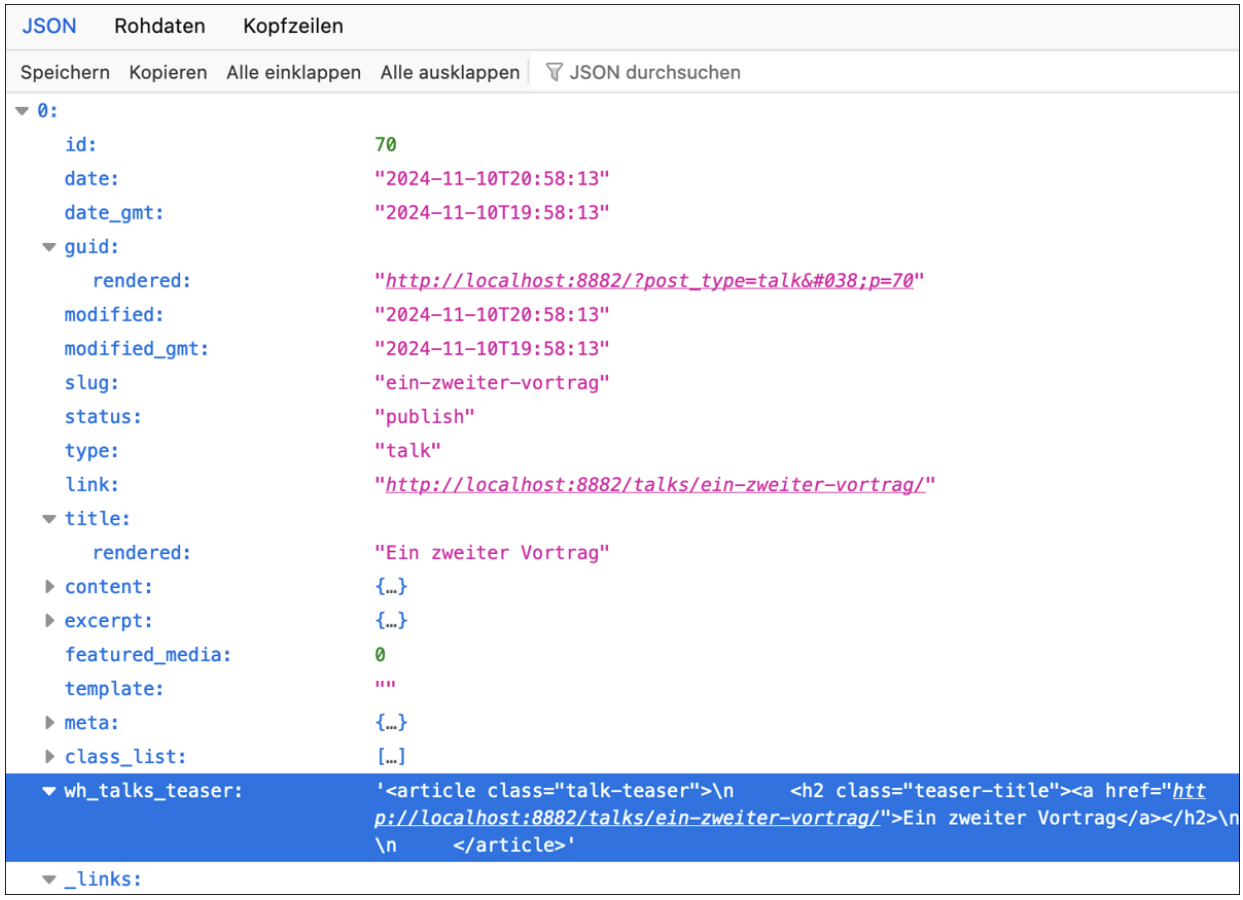


Abbildung 24.30 Das veränderte Objekt eines Vortrags in der REST-API



Abbildung 24.31 Die Antwort mit dem Objekt, das auf das »wh_talks_teaser«- Feld eingeschränkt wurde

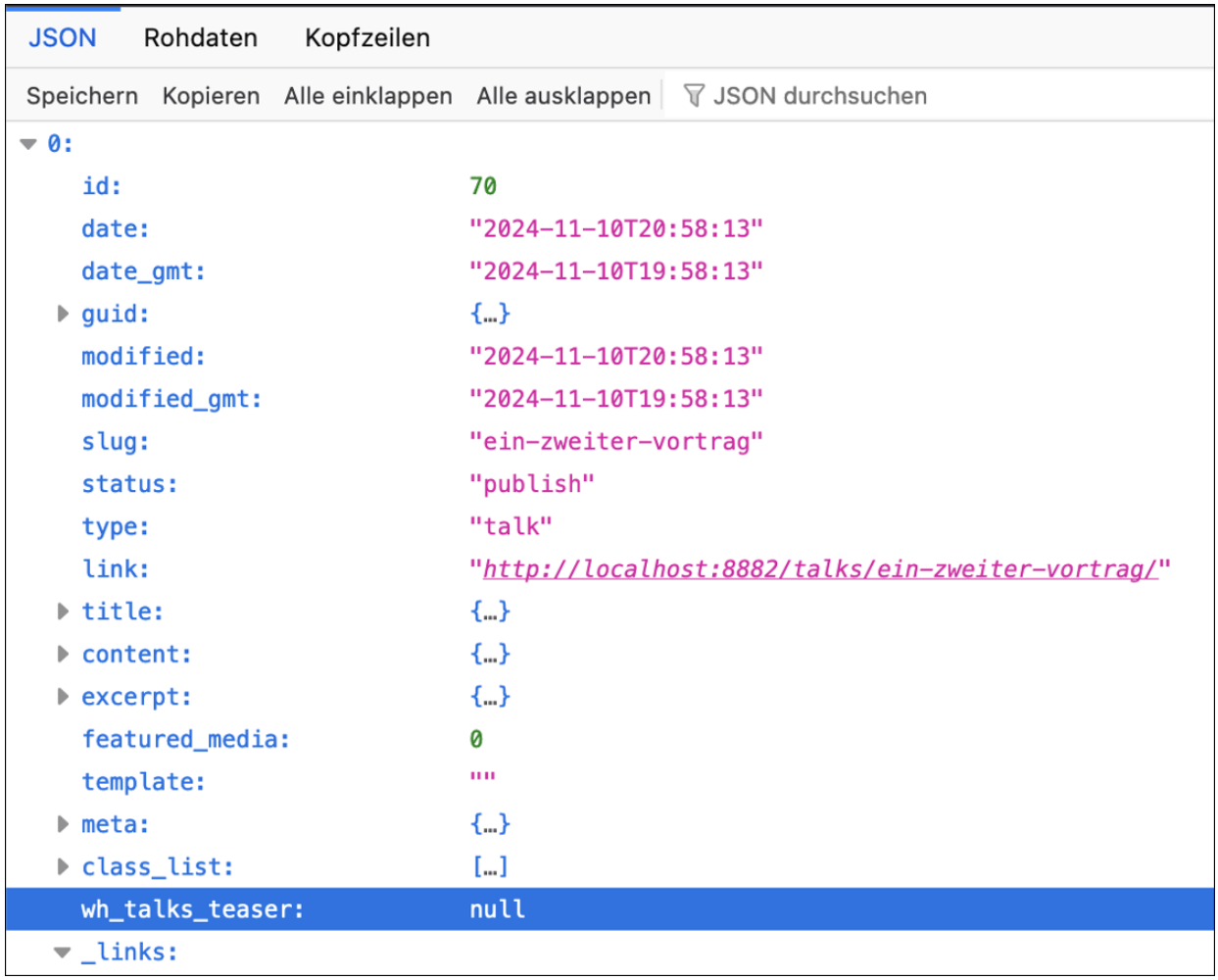


Abbildung 24.32 Wenn Sie den Endpunkt ohne Einschränkung auf das Feld aufrufen, ist das hinzugefügte Feld leer.



Abbildung 24.33 Die Ausgabe eines gültigen Aufrufs des neuen REST-Endpunkts für die Ausgabe des Videolinks zu einem Vortrag