Engineer

WordPress

ein quelloffenes System zur Inhaltsverwaltung von Websites

Was es ist

WordPress ist ein quelloffenes Content-Management-System: PHP-Software, die Beiträge, Seiten, Mediendatensätze, Benutzer und Einstellungen einer Website in einer MySQL- oder MariaDB-Datenbank speichert und daraus auf Anfrage Seiten zusammenstellt. Hochgeladene Mediendateien liegen normalerweise außerhalb der Datenbank. Der Name wird oft zu WP verkürzt. Dasselbe Präfix tragen die eigenen Dateien und Verzeichnisse des Systems: wp-config.php, wp-content und wp-admin. Die Software steht unter der GPLv2 oder einer späteren Version. Herunterladen, Ausführen und Ändern kosten daher nichts.

Eine WordPress-Website besteht aus drei Schichten. Der Kern ist der gemeinsame Code, den alle herunterladen und aktualisieren. Ein Theme steuert die Darstellung, Plugins ergänzen Verhalten. Genau dort zieht auch das Theme-Handbuch des Projekts die Grenze. Beide greifen über Hooks auf den Kern zu: benannte Stellen im WordPress-Code, an denen anderer Code ausgeführt werden darf. Hooks gibt es als Aktionen und Filter. Eine Aktion läuft zu einem bestimmten Zeitpunkt, etwa beim Speichern eines Beitrags, und gibt nichts zurück. Ein Filter erhält einen Wert, ändert ihn und gibt ihn zurück. Dieser verkürzt automatische Beitragsauszüge von den standardmäßigen 55 Wörtern auf 30:

```
add_filter( 'excerpt_length', function ( $length ) {
    return 30;
} );
```

Themes haben sich im Lauf der Jahre verändert. Klassische Themes verwenden das mit WordPress 1.5 im Jahr 2005 eingeführte PHP-Vorlagensystem und sind weiterhin weit verbreitet. Der aktuelle Ansatz sind Block-Themes: Ihre Vorlagen sind aus Blöcken aufgebaute HTML-Dateien, die sich im Site Editor bearbeiten

lassen. Farben, Schriften und Abstände liegen in einer theme.json-Datei. Hybride Themes sind klassische Themes, die einige Block-Funktionen übernommen haben.

Inhalte verlassen die Website auch als JSON über die REST-API, auf der der Block-Editor selbst aufbaut. Eine Anfrage an /wp-json/ liefert eine Übersicht aller angebotenen Routen. Öffentliche Inhalte sind ohne Anmeldung lesbar; alles Private erfordert eine Authentifizierung. Jedes Programm kann die drei neuesten Beiträge so abrufen:

```
curl -s 'https://example.com/wp-json/wp/v2/posts?per_page=3'
```

Für den Betrieb empfiehlt das Projekt PHP ab 8.3, MySQL ab 8.0 oder MariaDB ab 10.11 sowie HTTPS, vorzugsweise auf Apache oder Nginx. Joomla und Drupal sind Content-Management-Systeme derselben Art. Zwei weitere Alternativen für den Vergleich sind eine eigene Anwendung auf einem Framework wie Django und ein Generator für statische Websites.

Das Konzept für eine Entscheidung nutzen

Nutzen Sie „Was ist WordPress?“ für eine konkrete Entscheidung. Wir können helfen, das Konzept mit dem System in Verbindung zu bringen, das Sie entwickeln oder einkaufen. Eine schnelle, auffindbare Website, die Ihre Mitarbeitenden selbst bearbeiten können, ohne für jede Textänderung eine Rechnung zu bekommen.

Unsere Leistungen entdecken Kontakt aufnehmen →

Die Annahmen hinter dem Begriff prüfen

Nutzen Sie die Erklärung und die verlinkten Arbeiten, um Fragen für Ihr Projekt zu formulieren. Wir können die Annahmen gemeinsam mit Ihnen prüfen, bevor ein Werkzeug ausgewählt wird. 100 Webanwendungen mit HTML/HTML5 entwickelt — 100 entwickelte Webanwendungen

100 Webanwendungen mit HTML/HTML5 entwickelt Kontakt aufnehmen →

Warum es wichtig ist

Die eigentliche Entscheidung hinter WordPress lautet, wer die Website nach dem Start ändert und wer währenddessen für ihre Sicherheit sorgt. WordPress gibt Mitarbeitern ohne technischen Hintergrund einen vollständigen Editor: Sie können eine Seite veröffentlichen, ein Bild austauschen, einen Benutzer hinzufügen oder ein Menü überarbeiten, ohne ein Ticket anzulegen. Bei einer wöchentlich geänderten Website ist die günstigste Änderung die, für die kein Entwickler nötig ist.

Die Lizenz ist kostenlos, die Website nicht. Die GPL gilt für die Software; wordpress.org vertritt die Auffassung, dass Plugins und Themes abgeleitete Werke sind und dieselbe Lizenz übernehmen. Die Ausgaben verlagern sich: auf ein Hosting mit PHP und Datenbank, auf die Stunden für Auswahl, Einrichtung und Test eines Themes und auf die laufende Pflege von Kern, Plugins, Themes und PHP-Version. Diese letzte Ausgabe bleibt während der gesamten Lebensdauer der Website bestehen. Ein Angebot sollte benennen, wer sie trägt.

Der Pflegeaufwand folgt daraus, dass WordPress eine laufende Anwendung ist. Zur Seitenerzeugung ohne Cache führt WordPress PHP-Code aus, der auf eine Datenbank zugreift; ein Cache für ganze Seiten kann eine gespeicherte Seite ausliefern. Plugin-Code läuft mit denselben Zugriffsrechten wie der Kern. Die Anmeldeseite ist aus dem offenen Internet erreichbar, sofern niemand eine zweite Zugangssperre davor setzt. Hintergrundaktualisierungen des Kerns gibt es seit Version 3.7. Bei neuen Installationen ab 5.6 sind standardmäßig sowohl kleinere als auch größere Kernaktualisierungen aktiviert, abhängig von Konfiguration und Eignung der Installation für das jeweilige Update. Ältere Installationen behalten ihre bisherigen Einstellungen, solange diese nicht geändert werden. PHP folgt einem eigenen Aktualisierungsplan.

Eine eigene Entwicklung passt zu Daten und Abläufen, die mit Seiten und Beiträgen wenig gemein haben. Ihre redaktionellen Abläufe müssen entwickelt und gepflegt werden. Eine statische Website nimmt die PHP- und Datenbankanwendung aus der Verarbeitung eingehender Anfragen heraus; ihr Server und ihre Build-Werkzeuge brauchen weiterhin Pflege. Ihre Redakteure arbeiten meist mit Dateien und einem Build-Schritt. WordPress deckt den breiten Bereich dazwischen ab: eine eher gewöhnliche Website, die ihre Eigentümer selbst betreiben, und bei der die Bearbeitungsoberflächen den Pflegeaufwand wert sind. Die Anbieterbindung bleibt überschaubar: Der Code steht unter der GPL, die Inhalte liegen in einer gewöhnlichen SQL-Datenbank, und die REST-API liefert Beiträge, Seiten und Taxonomien als JSON.

Wie es in der Praxis aussieht

In der Arbeit dieses Unternehmens ist WordPress eine bewusst gewählte Antwort unter mehreren. Der Bericht über hundert Webanwendungen deckt 2003–2014 ab und nennt die Entscheidungsregel. Brauchte ein Kunde etwas Individuelles, wurde mit HTML/HTML5, CSS/SCSS und Django entwickelt. Brauchte er eine eher gewöhnliche Website, die er auch selbst pflegen konnte, führten WordPress oder Joomla schneller zum Ziel und waren sinnvoller.

Der Bericht würdigt das Urteilsvermögen, nicht das Werkzeug. Zu der beschriebenen Arbeit gehörte, beide Fälle auseinanderzuhalten: einem Kunden von einer nicht benötigten Maßanfertigung abzuraten und ihm dann dazu zu raten, wenn der Bedarf tatsächlich bestand.

Als Fragen, die ein Käufer schon vor der Bitte um ein Angebot beantworten kann, sieht diese Abwägung so aus:

- Werden Menschen die Website jede Woche ohne Entwickler bearbeiten, oder wird sie nur ein paarmal im Jahr geändert?
- Passen die Inhalte zu Seiten, Beiträgen, Kategorien und Medien, oder brauchen sie ein eigenes Datenmodell mit eigenen Regeln?
- Wer hält Kern, Plugins und PHP aktuell, solange die Website in Betrieb ist?
- Baut die Website auf einer Handvoll gut gepflegter Plugins auf oder auf Dutzenden, die zusammengefügt werden, um eine Anwendung nachzubilden?

Wöchentliche Änderungen, passende Inhalte, ein benannter Verantwortlicher und eine kurze Plugin-Liste sprechen für WordPress oder ein ähnliches CMS. Bei ungewöhnlichen Daten oder wenn niemand das System aktuell hält, bietet sich eine eigene Entwicklung oder eine statische Website an.

Wo es schiefgeht

Ein gewöhnliches Update verursacht Fehler auf der Website

Jemand aktualisiert ein Plugin, und eine Seite zeigt einen Fehler oder das Dashboard lädt nicht mehr. Über Jahre hatten sich Plugins angesammelt, manche von ihren Autoren aufgegeben. Jedes brachte fremden PHP-Code mit vollem Zugriff mit, und zwei gerieten in Konflikt. Halten Sie die Liste kurz und löschen Sie jedes nicht mehr verwendete Plugin, wie das Handbuch zur Absicherung verlangt. Erstellen Sie vor jedem Update ein Backup, testen Sie größere Updates zuerst auf einer Kopie und bevorzugen Sie Plugins, deren Autoren weiterhin neue Versionen veröffentlichen.

Eigene Anpassungen verschwinden nach einem Update oder einer Neugestaltung

Eine monatelang funktionierende Anpassung verschwindet mit dem Theme-Update. Oder eine für das Geschäft nötige Funktion fehlt, nachdem die Website ein neues Aussehen bekommen hat. Im ersten Fall wurden die Dateien eines fremden Themes bearbeitet und durch das Update ersetzt. Das Theme-

Handbuch warnt, dass eine direkt in der functions.php eines Themes ergänzte Funktion beim nächsten Update verschwindet. Im zweiten Fall steckte das für die Website unverzichtbare Verhalten im Theme und ging mit ihm. Legen Sie Änderungen an fremden Themes in einem Child-Theme ab und Funktionen in einem eigenen kleinen Plugin, das sich über Hooks in den Kern einklinkt. So trägt das Theme die Darstellung und nichts anderes.

Ein Administratorpasswort ermöglicht Codeausführung auf dem Server

Wer das Passwort eines Administrators stiehlt, kann vom Bearbeiten von Beiträgen zur Ausführung eigenen PHP-Codes übergehen: Das Dashboard enthält einen Editor für Theme- und Plugin-Dateien. Eine Zeile in wp-config.php schaltet ihn ab. Das hat denselben Effekt, als würden jedem Benutzer die Fähigkeiten edit_themes, edit_plugins und edit_files entzogen:

```
define( 'DISALLOW_FILE_EDIT', true );
```

Das schließt einen von mehreren Wegen. Das Handbuch sagt ausdrücklich, dass es einen Angreifer nicht aufhält, der bösartige Dateien hochladen kann. Verbinden Sie es mit starken Passwörtern, einer Zwei-Schritt-Authentifizierung für jeden Administrator und so wenigen Administratorkonten, wie die Arbeit erlaubt.

Die Website bleibt hinter der aktuellen Version zurück

Das Dashboard zeigt monatelang den Hinweis auf eine neue Version, während die Website auf einer stark veralteten Hauptversion oder PHP-Version läuft. Eine nur für kleinere Kernaktualisierungen konfigurierte Website kann sich weiter mit Patches versorgen, während Hauptversion und PHP zurückfallen, bis ein benötigtes Plugin eine neuere Version verlangt. Bei neuen Installationen ab 5.6 sind auch größere Kernaktualisierungen standardmäßig aktiviert; die Konfiguration kann das jedoch ändern. Planen Sie größere Updates fest ein, gleichen Sie dabei die PHP-Version mit der Anforderungsseite des Projekts ab und testen Sie beides auf einer Kopie, bevor Sie die produktive Website aktualisieren.

Weiterführende Lektüre

- WordPress-Dokumentation — das Handbuch des Projekts zum Websitebetrieb, vom Veröffentlichen und den Medien bis zur Installation und Pflege.
- Anforderungen — die aktuell vom Projekt empfohlenen Versionen von PHP, MySQL und MariaDB.
- WordPress aktualisieren — Aktualisierungen im Hintergrund, per Klick und von Hand sowie das vorher anzulegende Backup.
- WordPress absichern — Dateirechte, Datenbankprivilegien, der Schalter für den Dateieditor und der sorgfältige Umgang mit Plugins.
- Plugin-Handbuch: Hooks — wie Code über Aktionen und Filter WordPress verändern kann, ohne den Kern anzutasten.
- REST-API-Handbuch — Routen, Endpunkte und Authentifizierung der API, auf der der Block-Editor aufbaut.



Glossar

Für Ihre tatsächlichen Rahmenbedingungen auswählen

Beschreiben Sie den Arbeitsablauf, die vorhandenen Schnittstellen und die Nutzer, die die Veränderung benötigen. Wir können eine prüfbare Lieferung definieren und die Übergabe so gestalten, dass Ihr Team die gelieferten Bestandteile warten kann.

Preise ansehen Kontakt aufnehmen →

Die Begriffe praktisch nutzbar machen

Fragen Sie, wie „Was ist WordPress?“ auf Ihre Organisation zutrifft. Ein konkretes Beispiel aus Ihrem bestehenden System reicht aus, um das Gespräch zu beginnen.

Unsere Leistungen entdecken

welcome@engineer.company +45 50 10 61 84 Kopenhagen, Dänemark

Nehmen Sie diese Seite mit

Kontaktkarte

Mit Sorgfalt und Liebe gestaltet von D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Copyright © 2025 – heute · Engineer ApS · Alle Rechte vorbehalten.

Gedruckt von https://engineer.company/de/glossary/wordpress/