

# Website Performance: Die Geschwindigkeit von WordPress Seiten verbessern

Date : 7. Juli 2015

Eine langsame WordPress Seite ist ein Ärgernis für den Betreiber: Besucher wenden sich möglicherweise genervt ab, wenn die Seiten nicht sofort laden und auch Suchmaschinen zieht die Performance einer Webseite mit in das Ranking ein. Unser Artikel präsentiert erprobte Vorgehensweisen, um die Ladegeschwindigkeit von WordPress Seiten zu erhöhen.

Webseiten, die mit neueren WordPress Versionen betrieben werden, sollten standardmäßig keine Performance Probleme verursachen. Laden sie trotzdem zu langsam, so ist die Ursache meist im Laden zu vieler Scripts (JavaScript, CSS), zu großer Bilddateien oder einem zu schlechten Webpace Paket zu finden. Orientierung bietet das [PageSpeed Insights Tool](#) von Google. URL der Webseite eingeben, die Analyse abwarten - und Google teilt nach wenigen Sekunden mit, welche Fehler die Ursache einer langen Ladezeit sind.

Google definiert [10 Regeln für eine höhere Geschwindigkeit](#). Um diese Empfehlungen in der WordPress Seite umzusetzen, gibt es zwei Möglichkeiten: Man bedient sich entsprechender Plugins oder man nimmt manuelle Anpassungen vor. Serverseitige Probleme (z.B. langsame Antwortzeit des Servers) werden in diesem Artikel nicht besprochen und liegen in der Zuständigkeit des Webpace Anbieters.

## Was ohne großen Aufwand möglich ist

Folgende Regeln lassen sich ohne größere Eingriffe umsetzen:

- **Ungenutzte Plugins deaktivieren:** Wer die Funktionen eines Plugins nicht nutzt, sollte dieses deaktivieren, sodass im Hintergrund keine Scripts über das Plugin geladen werden.
- **Auf unnötige Plugins verzichten:** Viele Plugins sind überflüssig. Ein Plugin zum Einbinden von YouTube Videos etwa ist nicht erforderlich, da der entsprechende Code von der YouTube Seite auch manuell kopiert und in den Beitrag im Text-Modus eingefügt werden kann.
- **Bilder vor dem Hochladen verkleinern:** Wer Bilder in eine Webseite einbindet, sollte diese mit einem Bildbearbeitungsprogramm zunächst auf maximal 1024px Breite verkleinern und anschließend mit einem Tool zur Bildkompression die Dateigröße reduzieren. Am einfachsten geht das mit einem [lokalen Tool](#) für Mac oder Windows.

## Caching

Caching sorgt dafür, dass bestimmte Dinge nicht immer erneut angefragt werden müssen, sondern zwischengespeichert werden und somit schneller ausgeliefert werden können. Caching sollte man in zweierlei Hinsicht verwenden: Um die Datenbankabfragen zu reduzieren und um Dateien wie Bilder, JavaScripts und CSS zwischenzuspeichern.

Das **Reduzieren der Datenbankabfragen** gelingt am besten mit dem Plugin [Cachify](#). Das Plugin ermöglicht es, dass die von WordPress generierten HTML Seiten serverseitig zwischengespeichert werden und in komprimierter Form ausgegeben werden. Somit wird die Zahl der Datenbankabfragen deutlich reduziert und die Komprimierung sorgt für eine reduzierte Dateigröße der HTML Datei.

*Cachify minimiert die Ladezeiten deutlich und reduziert die Datenbankabfragen*

Um das **Zwischenspeichern von Dateien** wie Bildern und Scripts zu ermöglichen, wird dem Browser in Verfallsdatum für diese Dateien mitgeteilt. Das erledigt das Apache Modul [mod\\_expires](#), das über die .htaccess Datei aktiviert wird. Ein Plugin wird dafür nicht benötigt: Einfach .htaccess Datei vom WordPress Root Verzeichnis herunterladen, in einem Editor öffnen und die Expires setzen. Anschließendes Hochladen und Testen nicht vergessen. Bei manchen Webhosting Anbietern muss mod\_expires möglicherweise erst noch freigeschaltet werden.

## **Asynchrone Skripts für schnelleres Rendern**

Wenn CSS Dateien und JavaScripts schon im Kopfbereich der durch WordPress generierten HTML nachgeladen werden, verzögert dies den Seitenaufbau. Das [Plugin Async JS and CSS](#) setzt

alle Scripts in den Footer.

Standardmäßig platziert das Plugin auch CSS Dateien im Footer. Diese Funktion kann nachteilig sein, da die Seite in manchen Browsern möglicherweise doppelt rendert und zudem die W3C Standards verletzt, die das Einbinden von Stylesheets nur im Head Bereich erlauben (siehe dazu auch [diesen Artikel von Blog IT-Solutions](#)). Daher: *Load CSS asynchronously* in den Einstellungen deaktivieren.

## HTTP Requests reduzieren

Beim Laden einer Webseite sind verschiedene HTTP Requests erforderlich: Externe CSS und JavaScript Ressourcen werden eingebunden und Bilder zur Anzeige geladen. Besonders bei der Verwendung von Plugins werden häufig seitenweit externe Ressourcen geladen, die eigentlich nur auf einigen wenigen Unterseiten erforderlich wären. Mit einigen kleinen Anpassungen kann dies verhindert werden.

Ein Eingriff in die Ausgabe von externen Scripts ist nur dann sinnvoll, wenn absehbar ist, dass nicht ständig neue Themes und Plugins verwendet werden. Auch der Update Prozess von Plugins wird etwas erschwert, da dieser nun nicht mehr per Mausklick möglich ist. Prinzipiell geht es nämlich darum, die CSS und JavaScript Dateien in jeweils einer Datei zusammenzufassen und diese komprimiert an den Browser zu übertragen.

Dafür sind mehrere Schritte erforderlich:

- **1. Plugins mit externen Scripts und CSS finden:** Am schnellsten gelingt das, indem der Quelltext der Seite aufgerufen und per Suchfunktion nach .js bzw. .css gesucht wird. Über die generierte URL lässt sich schnell ablesen, welches Plugin das Script generiert.
- **2. Scripts und CSS abmelden:** Wenn das Plugin sauber programmiert wurde, sollten das Script per `wp_dequeue_script` ([Codex](#)) und CSS Dateien per `wp_dequeue_style` ([Codex](#)) von der Ausgabe ausgeschlossen werden können. Eine genaue Anleitung folgt in einem separaten Artikel
- **3. Jeweils eine CSS und eine JavaScript Datei anlegen**
- **4. Dateien komprimieren:** Beide Dateien mit [gzip komprimieren](#), hochladen und in der .htaccess Datei den Browser anweisen, vorzugsweise die komprimierte Datei zu laden.

## Fazit

Wenn die WordPress Seite zu langsam ist, sind "aufgeblasene" Seiten häufig die Ursache. Die oben beschriebenen Schritte erhöhen den Arbeitsaufwand, reduzieren die Ladezeiten aber spürbar.