

Website Performance

Daniel Koch



schnell + kompakt

Daniel Koch

Website Performance

Daniel Koch

Website Performance

schnell+kompakt

entwickler.press

Daniel Koch
Website Performance
schnell+kompakt
ISBN: 978-3-86802-269-8

© 2012 entwickler.press
ein Imprint der Software & Support Media GmbH

<http://www.entwickler-press.de>
<http://www.software-support.biz>

Ihr Kontakt zum Verlag und Lektorat: lektorat@entwickler-press.de

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

Lektorat: Sebastian Burkart
Korrektur: Redaktion ALUAN
Satz: Dominique Kalbassi
Umschlaggestaltung: Maria Rudi

Alle Rechte, auch für Übersetzungen, sind vorbehalten. Reproduktion jeglicher Art (Fotokopie, Nachdruck, Mikrofilm, Erfassung auf elektronischen Datenträgern oder andere Verfahren) nur mit schriftlicher Genehmigung des Verlags. Jegliche Haftung für die Richtigkeit des gesamten Werks, kann, trotz sorgfältiger Prüfung durch Autor und Verlag, nicht übernommen werden. Die im Buch genannten Produkte, Warenzeichen und Firmennamen sind in der Regel durch deren Inhaber geschützt.

Inhaltsverzeichnis

Vorwort	7
Kapitel 1: Performance-Optimierung: Das neue SEO	9
1.1 Den Ist-Zustand überprüfen	10
1.2 Die perfekte Analyse mit YSlow	19
1.3 WebPagetest	20
1.4 loads.in	27
Kapitel 2: HTTP-Requests verringern	31
2.1 Die Anzahl der Requests verringern	32
2.2 Dateien zusammenführen	38
2.3 Dateien in der richtigen Reihenfolge einbinden	45
2.4 Dateien optimieren	49
Kapitel 3: CSS optimieren	57
3.1 CSS perfekt einbinden	57
3.2 Vorsicht bei den Selektoren	61
3.3 Kurz-CSS: Shorthands verwenden	64
3.4 Farbdefinitionen kürzen	70
3.5 Browser-unabhängiges CSS	71
3.6 Mit CSS-Sprites Bilder einsparen	78
3.7 Hardware-Beschleunigung verwenden	85
3.8 Transition als JavaScript-Alternative	87

Kapitel 4: Bilder optimieren	89
4.1 Grafikformate: Ein Schnelleinstieg	90
4.2 Die optimale Qualität	92
4.3 Größenangaben	96
4.4 HTML oder CSS?	98
4.5 Farbtiefe und Farbpalette	100
4.6 Grafiken optimieren	102
4.7 Inline-Bilder verwenden	105
4.8 Favicons	109
4.9 Bilder vorladen	112
4.10 CSS3 ausreizen	114
4.11 Perfekte Schriften	121
4.12 Speed mit HTML5	126
Kapitel 5: Weitere Möglichkeiten ausreizen	129
5.1 Die Performance von Code-Abschnitten messen	129
5.2 JavaScript optimieren	130
5.3 PHP optimieren	137
5.4 Content Delivery Networks	141
5.5 ETags: Änderungen an der Seite erkennen	142
5.6 Inhalte aus der Retorte: Caching aktivieren	144
5.7 Inhalte komprimieren	145
Stichwortverzeichnis	149

Vorwort

Wenn Google Maps plötzlich 30 Prozent mehr Kartenmaterial ausliefert und Mozilla 15 Prozent mehr Downloads verzeichnet, könnte man von perfekter Suchmaschinenoptimierung oder erhöhtem Werbeetat ausgehen. Die beiden Erfolgsgeschichten von Mozilla und Google sind allerdings nicht auf diese klassischen Maßnahmen zurückzuführen. Vielmehr ist ein ganz anderer Aspekt dafür verantwortlich, der tatsächlich für den Erfolg von Webseiten immer wichtiger wird: die Website-Performance-Optimierung (WPO). Das zeigt auch die Ankündigung von Google, die Ladezeiten von Webseiten als Parameter für das Suchmaschinen-Ranking zu verwenden.

Dieses Buch stellt Ihnen die besten Tipps und Tricks vor, mit denen Sie die Performance Ihrer Webseite im Handumdrehen verbessern können. Dabei werden ausschließlich solche Dinge behandelt, die tatsächlich jeder umsetzen kann, der Webseiten erstellt. Auf „abstrakte“ Themen wie beispielsweise den Zukauf teurerer Server oder Tipps für eine Investition in neue Infrastrukturen wird an dieser Stelle bewusst verzichtet.

Sie erfahren, wie Sie Ihre Webseite so optimieren, dass diese auch den gestiegenen Anforderungen von Google & Co. hinsichtlich der Performance entspricht. Die vorgestellten Dinge lassen sich jeweils binnen weniger Minuten umsetzen.

Sollten Sie während oder nach dem Lesen des Buchs Fragen oder Anmerkungen haben, Kontakt zum Autor aufnehmen oder Kritik

bzw. Lob loswerden wollen, können Sie das über kontakt@medienwerke.de tun – Ihre Nachricht ist herzlich willkommen.

Berlin, September 2012

Daniel Koch

Performance-Optimierung: Das neue SEO

1.1 Den Ist-Zustand überprüfen	10
1.2 Die perfekte Analyse mit YSlow	19
1.3 WebPagetest	20
1.4 loads.in	27

Aktuelle Messungen zeigen, wie stark sich eine gute Performance auf den Erfolg von Webseiten auswirkt. So wird im offiziellen Mozilla-Blog von jährlich 60.000.000 heruntergeladenen Firefox-Browsern berichtet. Eine stolze Zahl, keine Frage. Diese wurde aber durch gezielte Performance-Optimierung der Landing-Page um mehr als 15 Prozent gesteigert. Ähnliche Erfolge konnte auch Google verzeichnen. So reduzierte Google Maps das übertragene Dateivolumen um 30 Prozent und konnte gleichzeitig bei den Kartenanfragen einen Anstieg von ebenfalls 30 Prozent verbuchen. Auch die Auswirkungen schlechter Performance wurden hinlänglich untersucht. Für die Suchmaschine Bing bedeutete eine zweisekündige Verzögerung 4,3 Prozent weniger Umsatz pro Nutzer. Auch bei Amazon registrierte man die negativen Auswirkungen verminderter Performance: Dort bedeuten 100 Millisekunden langsamere Ladezeit einen Umsatzrückgang von einem Prozent. Yahoo! macht ähnliche Erfahrungen. Hier führt eine Verzögerung von 400 Millisekunden zu 5 bis 9 Prozent weniger Traffic. Der Erfolg einer Webseite hängt somit offensichtlich auch von deren

Performance ab. Suchmaschinenoptimierung (SEO) allein ist also kein Erfolgsgarant mehr. Daher wird neben dem klassischen SEO auch das Thema Web-Performance-Optimierung immer mehr in den Fokus der Web-Entwickler rücken. Wie beim SEO gibt es auch für die WPO nicht das alleinige Patentrezept. Vielmehr müssen verschiedene Faktoren berücksichtigt werden, um die Webseiten-Performance zu verbessern.

1.1 Den Ist-Zustand überprüfen

Bevor WPO-Maßnahmen ergriffen werden, sollte man den Performance-Ist-Zustand der Webseite überprüfen. Dafür gibt es ganz unterschiedliche Möglichkeiten, von denen einige in diesem Kapitel vorgestellt werden. Zunächst aber noch einige allgemeine Hinweise. Wenn Sie Ihre Webseite hinsichtlich der Performance überprüfen, werden Sie zahlreiche Messergebnisse erhalten. Was aber sagen diese aus? Wie groß sollte eine Webseite sein und in welcher Geschwindigkeit sollte sie angezeigt werden? Pauschal lässt sich das natürlich nicht sagen. Dennoch kann man innerhalb der letzten Jahre eine stetige Zunahme der Webseitengröße erkennen. So hat Riverbed Technology (<http://www.riverbed.com/de/>) eine durchschnittliche Webseitengröße von 1 MB gemessen. Ebenso wie die reine Größe ist aber auch die Anzahl der in einer Webseite eingebundenen bzw. damit verknüpften Elemente während der letzten Jahre gestiegen. Mittlerweile ist man laut Riverbed Technology bei einer durchschnittlichen Anzahl von 85 Elementen angelangt.

Wie schnell eine Webseite geladen werden sollte, damit man von einer guten Performance sprechen kann, ist pauschal schwer zu sagen. Als interessant erweist sich in diesem Zusammenhang aber ein Test von Uptrends (<http://www.uptrends.de/>). Dieser Spezialist für Lösungen zum Web Performance Monitoring hat die

Performance von 20 Online-Blumenshops überprüft. Dabei ging Uptrends davon aus, dass Besucher der Shops nicht länger als zwei Sekunden Ladezeit hinnehmen wollen. Hier einige Testergebnisse:

- floridee.de = 1,10 Sekunden
- valentins.de = 1,14 Sekunden

Das waren die schnellsten beiden Seiten im Test. Die Letztplatzierten sind hingegen:

- fleurop.de = 5,09 Sekunden
- floraprima.de = 5,61 Sekunden

Den Gesamttest finden Sie unter <http://www.uptrends.com/html/de/Uptrends-Ladezeiten-Vergleich-Online-Blumenshops-Muttertag>.

Wie schnell eine Seite laden sollte, hängt übrigens auch vom Thema der Seite ab. So haben Besucher der Seite eines Fotografen sicherlich etwas mehr „Geduld“ als in einem Online-Shop. Sie sollten daher die Geschwindigkeit Ihrer Seite auch in Relation zur Performance der Seiten Ihrer Mitbewerber sehen.

Entscheidend ist zudem, darauf zu achten, welche Seite gemessen wird. Oftmals begeht man nämlich den Fehler, ausschließlich die Startseite zu überprüfen. Das genügt jedoch nicht. Die Performance sollte möglichst auf einer repräsentativen Auswahl der Seiten gemessen werden. Für einen guten Schnelltest sind Web-Dienste eine gute Wahl. Einer davon wird direkt von Google unter <https://developers.google.com/pagespeed/> kostenlos angeboten. Auf der genannten Seite muss lediglich der URL der zu überprüfenden Seite angegeben werden.

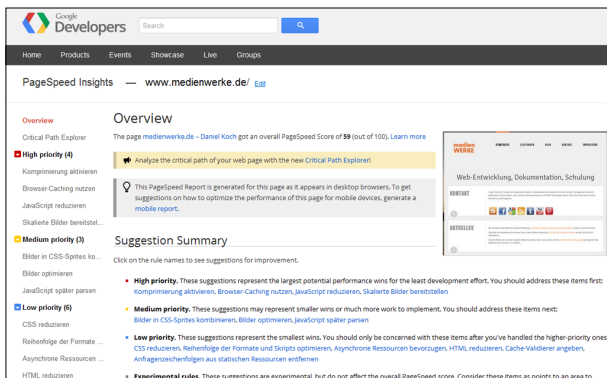


Abbildung 1.1: Hier kann die Geschwindigkeit gemessen werden

Nach erfolgreicher Analyse ist vor allem der linke Fensterbereich entscheidend. Denn dort sind die möglichen Maßnahmen und deren jeweiliges Einsparpotenzial zu sehen. Unterteilt ist der linke Fensterbereich nach der Priorität der vorgeschlagenen Maßnahmen.

- *High priority* – Hier sind die wichtigsten WPO-Maßnahmen aufgeführt. Mehr dazu gleich im Anschluss.
- *Medium priority* – Hier sind ebenfalls wichtige Dinge aufgeführt, die zu einer Verbesserung der Performance beitragen. Typische Aspekte können CSS-Sprites und zu umfangreicher JavaScript-Code sein.
- *Low priority* – In diesem Bereich ist all das aufgeführt, was man zusätzlich tun kann, im Einzelfall aber nur einen geringen Effekt hat. Dennoch sollten Sie überprüfen, ob sich die hier vor-

geschlagenen Maßnahmen nicht vielleicht doch recht schnell umsetzen lassen.

- *Experimental rules* – Während des Tests können nicht alle Aspekte untersucht werden. Diese werden in einem Experimental-Bereich gelistet. Google schlägt Ihnen vor, die dort verzeichneten Maßnahmen zu Testzwecken auszuprobieren.
- *Already done!* – Dieser Bereich zeigt, welche Optimierungsmaßnahmen bereits umgesetzt wurden.

Je nachdem, wie gut die Seite bereits optimiert ist, sind dort nicht alle Punkte aufgeführt. Innerhalb der einzelnen Bereiche sind dann die jeweiligen Vorschläge zur Optimierung zu finden. Wenn Sie beispielsweise viel JavaScript verwenden – möglicherweise durch den Einsatz eines JavaScript-Frameworks –, findet sich der Punkt *JavaScript reduzieren*. Mit Anklicken dieses Links werden zusätzliche Informationen angezeigt:

- der URL des Skripts,
- die KB, die eingespart werden können,
- die Prozent, die eingespart werden können.

Im Fall von JavaScript sind jeweils die URLs zu den einzelnen Skripten aufgeführt. Mit *JavaScript reduzieren* ist in diesem Fall übrigens das Beseitigen überflüssiger Elemente aus den Skript-Dateien gemeint. Dazu gehören Leerzeichen, Kommentare etc. Ausführliche Informationen zur Optimierung von JavaScript finden Sie in Kapitel 5.

Arbeiten Sie zunächst die unter *High priority* zusammengefassten Punkte ab. So erscheint dort beispielsweise regelmäßig die Empfehlung *Komprimierung aktivieren*. Folgt man diesem Link, werden die Dateien angezeigt, für die eine Komprimierung sinnvoll erscheint.

Overview

Critical Path Explorer

High priority (4)

Komprimierung aktivieren

Browser-Caching nutzen

JavaScript reduzieren

Skalierte Bilder bereitstellen

Medium priority (3)

Bilder in CSS-Sprites ko...

Bilder optimieren

JavaScript später parsen

Low priority (6)

CSS reduzieren

Hebelpunkt der Homeba...

Asynchrone Ressourcen la...

Komprimierung aktivieren

Compressing resources with gzip or the Late can reduce the number of bytes sent over the network.
[Learn more](#)

Suggestions for this page

Die Komprimierung der folgenden Ressourcen mit gzip könnte Ihre Übertragungsgröße um 422,6 KiB verringern (Reduzierung um 73 %).

- Durch die Komprimierung von http://www.medienwerke.de/_/prototype.js... können Sie 105,6 KiB einsparen (Reduzierung um 77 %).
- Durch die Komprimierung von http://www.medienwerke.de/_/jquery.cookie.js... können Sie 30 KiB einsparen (Reduzierung um 81 %).
- Durch die Komprimierung von http://www.medienwerke.de/_/jquery.js... können Sie 29,1 KiB einsparen (Reduzierung um 77 %).
- Durch die Komprimierung von http://www.medienwerke.de/_/controls.js... können Sie 25,1 KiB einsparen (Reduzierung um 74 %).
- Durch die Komprimierung von http://www.medienwerke.de/_/jquery.js... können Sie 22 KiB einsparen (Reduzierung um 75 %).
- Durch die Komprimierung von http://www.medienwerke.de/_/jquery.js... können Sie 20,3 KiB einsparen (Reduzierung um 80 %).
- Durch die Komprimierung von http://www.medienwerke.de/_/jquery.prettyPhoto.js... können Sie 18,2 KiB einsparen (Reduzierung um 74 %).
- Durch die Komprimierung von http://www.medienwerke.de/_/jquery.form.js... können Sie 17,5 KiB einsparen (Reduzierung um 66 %).
- Durch die Komprimierung von http://www.medienwerke.de/_/prettyPhoto.swf... können Sie 16,4 KiB einsparen (Reduzierung um 85 %).
- Durch die Komprimierung von http://www.medienwerke.de/_/jquery.jquery.js... können Sie 13,2 KiB einsparen (Reduzierung um 68 %).
- Durch die Komprimierung von http://www.medienwerke.de/_/jquery.js... können Sie 12,1 KiB einsparen (Reduzierung um 82 %).
- Durch die Komprimierung von http://www.medienwerke.de/_/jquery.mobile.customized.min.js... können Sie 10,8 KiB einsparen (Reduzierung um 62 %).
- Durch die Komprimierung von http://www.medienwerke.de/_/base.css... können Sie 9,3 KiB einsparen (Reduzierung um 73 %).
- Durch die Komprimierung von http://www.medienwerke.de/_/main.js... können Sie 8,7 KiB einsparen (Reduzierung um 67 %).
- Durch die Komprimierung von http://www.medienwerke.de/_/slider.js... können Sie 8,1 KiB einsparen (Reduzierung um 72 %).

Abbildung 1.2: Hier lohnt sich die Komprimierung

Zusätzlich ist zu erkennen, in welchem Maße sich die Übertragungsgröße der einzelnen Dateien durch eine Komprimierung reduzieren lässt. Das können Sie in vielen Systemen selbst aktivieren. So stehen dafür beispielsweise in Joomla! oder WordPress Bordmittel zur Verfügung. Ausführliche Informationen zur Komprimierung finden Sie in Kapitel 5.

Möglicherweise taucht bei Ihnen außerdem der Punkt *JavaScript reduzieren* auf. Dahinter verbirgt sich die Empfehlung, unnötige Elemente aus den JavaScript-Dateien bzw. -Bereichen zu entfernen. Dafür gibt es einige Tools, die in Kapitel 2 vorgestellt werden.

Sollte der Punkt *JavaScript später parsen* bei Ihnen erscheinen, müssen Sie überprüfen, an welcher Stelle Ihrer Seite JavaScript momentan eingebunden ist. Auf vielen Webseiten wird JavaScript nämlich direkt im *head*-Element aufgerufen. Das blockiert allerdings den Seitenaufbau und senkt somit die Performance. Informationen zu diesem Thema finden Sie in Kapitel 2.

Über den Punkt *Browser-Caching nutzen* werden Sie darauf hingewiesen, für statische Ressourcen im HTTP-Header ein Ablaufdatum anzugeben. Was es damit auf sich hat, wird in Kapitel 5 gezeigt.

Auch um die in der Webseite integrierten Bilder kümmert sich der Speed-Test von Google. Kein Wunder, schließlich können Bilder echte Performance-Bremsen sein.

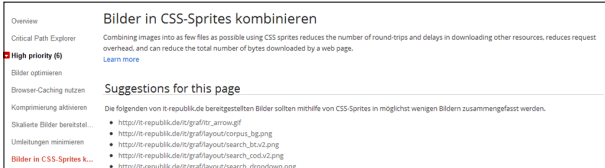


Abbildung 1.3: Bilder können ebenfalls optimiert werden

Die von PageSpeed vorgeschlagenen Punkte für die Bildoptimierung betreffen ganz unterschiedliche Bereiche.

- *Bilder optimieren* – Durch die Wahl des richtigen Grafikformats und eine gute Komprimierung lassen sich Bildgrößen minimieren.
- *Skalierte Bilder bereitstellen* – Dieses Phänomen ist auf erstaunlich vielen Seiten anzutreffen. Bilder werden mittels HTML oder CSS skaliert, also auch die richtige Größe gebracht. Das gilt es zu verhindern. Bilder sollten immer vorher in einem Grafikprogramm auf die richtige Größe gebracht werden. Mehr zu diesem Thema erfahren Sie in Kapitel 4.
- *Bilder in CSS-Sprites kombinieren* – Dabei werden mehrere Bilder zu einem Gesamtbild zusammengefasst. Mittels CSS lässt sich dann jeweils ein kleiner Teil dieses Gesamtbilds einblenden.

Die CSS-Sprites werden in Kapitel 3 noch einmal detailliert unter die Lupe genommen. Begegnet sind Ihnen diese Sprites allerdings schon an vielen Stellen im Web, beispielsweise auch bei YouTube.