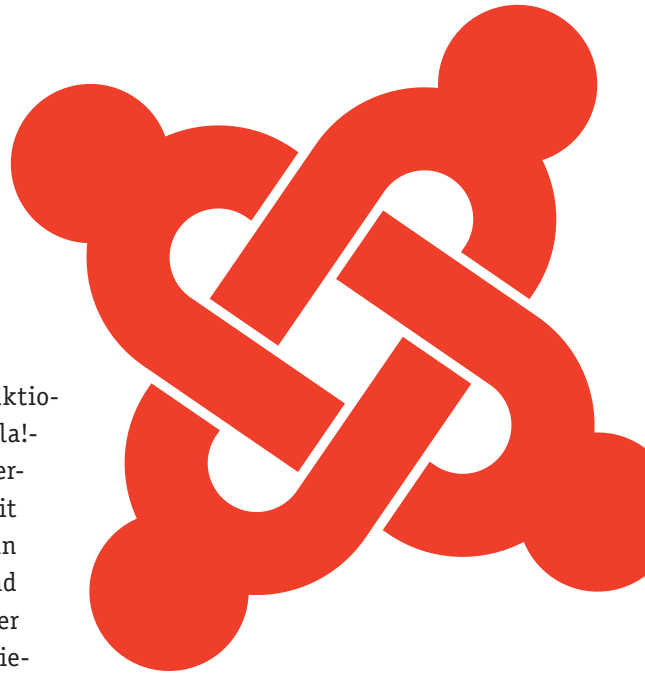


Martin Schirnbacher

SEO mit Joomla!

Teil 3/4

Drehte sich im ersten Teil in Ausgabe #36 alles um die SEO-Grundfunktionen und im zweiten Teil in Ausgabe #37 um Erweiterungen der Joomla!-Grundfunktionen, so geht in diesem Teil primär um das Thema Performance. Joomla!-Experte Stefan Wendhausen erläutert, wie man mit dem beliebten CMS mit Bordmitteln das Caching aktiviert und was ein unscheinbares Plug-in zur weiteren Steigerung der Performance und Webseitengeschwindigkeit beitragen kann. Wurde im ersten Teil der Serie bereits das Caching-Verhalten von Joomla! angerissen, wird dieses für das bessere Verständnis noch mal aufgegriffen und vertieft. Dies ist unter anderem notwendig, da seit Erscheinen der beiden ersten Artikel Joomla! mittlerweile in der neuen Major-Version 3.5 vorliegt und sich auch im Bereich Caching einiges getan hat.



PHP 7 verdoppelt die Ladegeschwindigkeit

Eine der wichtigsten Neuerungen in Joomla! 3.5 ist die native Unterstützung von PHP 7 und der damit verbundene Performance-Boost rein durch eine moderne und aktuelle PHP-Version. In Abhängigkeit von den Inhalten sind Ladezeiten von teilweise < 1 Sekunde zu erwarten.

In den meisten Fällen kann von einer Verdoppelung der Leistung (Anfragen/Sekunde) und Halbierung des Speicherverbrauchs als Anhaltspunkt ausgegangen werden, was sich in der Praxis beweisen muss.

Profiling mit Joomla!

Joomla! hat im Core ein integriertes Profiling-Tool, welches über die Konfiguration aktiviert werden kann. System » Konfiguration » Tab System » System debuggen auf Ja stellen. Auf der Modulposition debug wird nach Aktivierung im Back- und Frontend die Joomla!-Debug-Konsole ausgegeben (Abbildung 2). Die Konsole enthält Informationen zur Sitzung, das Profil zum Laufzeitverhalten, die Speichernutzung und die für die gerenderte Seite benutzten Datenbankabfragen.

Testszenario

Die Ausgangssituation sind drei identische Joomla!-3.5.1-Seiten mit Beispieldaten wie in Abbildung 1 zu sehen.

- » 1 Content mit Bild (com_content)
- » 8 Module (mod_MODULENAME)

Die Seiten liegen auf dem gleichen Server und unterscheiden sich durch eine andere PHP-Version. Das Caching wurde deaktiviert und es handelt sich um einen rein objektiven Frontend-Test ohne Anspruch auf Vollständigkeit. Bereits nach dem afterRenderComponent com_content hängte die Installation unter PHP 7 mit < 100 ms die anderen beiden ab. Insgesamt spart bei dieser Mini-Seite der Einsatz von PHP 7 ca. 65 ms an

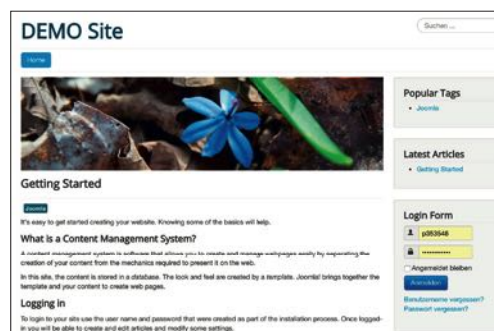


Abb.1: Joomla! 3.5.1 DEMO Site

DER AUTOR



Stefan Wendhausen ist geschäftsführender Gesellschafter der tec-promotion GmbH in Fürth und spezialisiert auf das Hosting von Joomla!-basierten Webseiten.

TIPP

Wenn die Möglichkeit besteht, sollte ein Update auf Joomla! 3.5 und ein Update auf PHP 7 umgehend durchgeführt werden. Der Unterschied bei größeren Seiten ist im Bereich der Seiten-Lade-Geschwindigkeit sofort spür- und messbar.

Ladezeit gegenüber PHP 5.5 und ca. 38 ms gegenüber PHP 5.6 ein.

Aus Tabelle 1 geht PHP 7 als deutlicher Sieger hervor. Die Halbierung der Speichernutzung konnte in diesem Testzenario allerdings nicht bewiesen werden. Bei umfangreicheren Seiten wird dies allerdings mehr ins Gewicht fallen.

Bei der Umstellung auf PHP 7 gilt der Grundsatz, ein Back-up anzulegen und auf einer Testumgebung ausführlich zu testen. Es müssen alle Core-Erweiterungen sowie Drittanbieter-Erweiterungen vorher auf den neuesten Stand gebracht werden.

Zwischenspeicher (Cache) – Einstellungsmöglichkeiten

Im Folgenden sollen die drei derzeit implementierten Caching-Möglichkeiten in Joomla! 3.5 im Einzelnen näher betrachtet und auf die unterschiedlichen Cache-Speicher eingegangen werden.

Cache

Die Cache-Einstellungen sind unter System » Konfiguration » Tab System zu finden. Im Bereich Zwischenspeicher (Cache) sind im Drop-down-Menü Cache drei Optionen möglich: Aus – Cache deaktiviert, An – Normales Caching, An – Erweitertes Caching (siehe Abbildung 5).

Aus – Cache deaktiviert schaltet den Cache für die Seite aus und es werden keine Inhalte im Cache vorgehalten. Dies kann unter System Cache leeren kurz überprüft werden und sollte, wie in Abbildung 6 zu sehen, keine passenden Ergebnisse auflisten.

An – Normales Caching schaltet den Cache ein und bringt die ersten Performance-Ergebnisse. Ein kurzer Check unter Cache leeren kann wie in

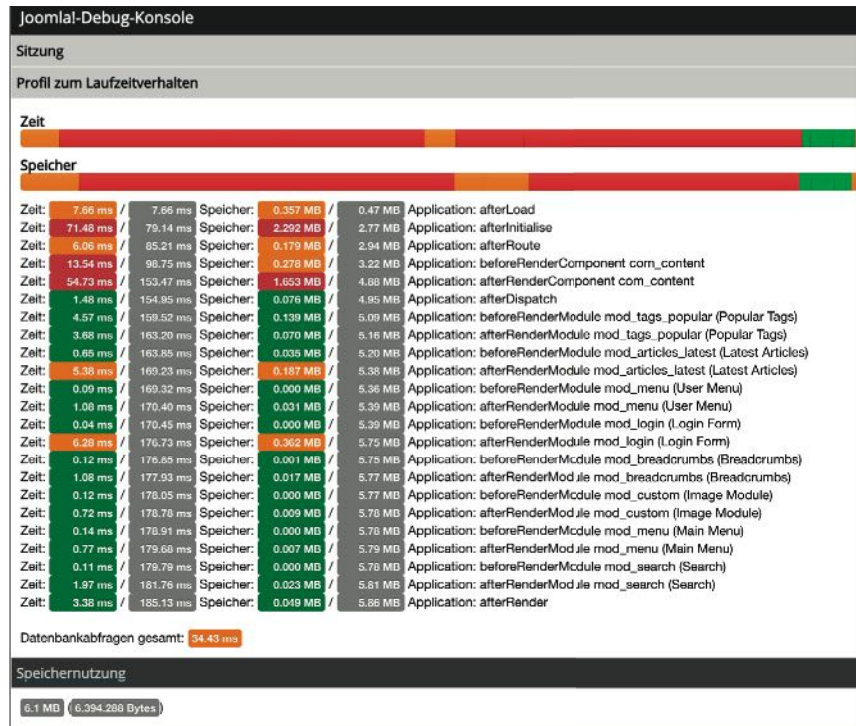


Abb.2: Profil zum Laufzeitverhalten unter PHP 5.5, Joomla! 3.5.1

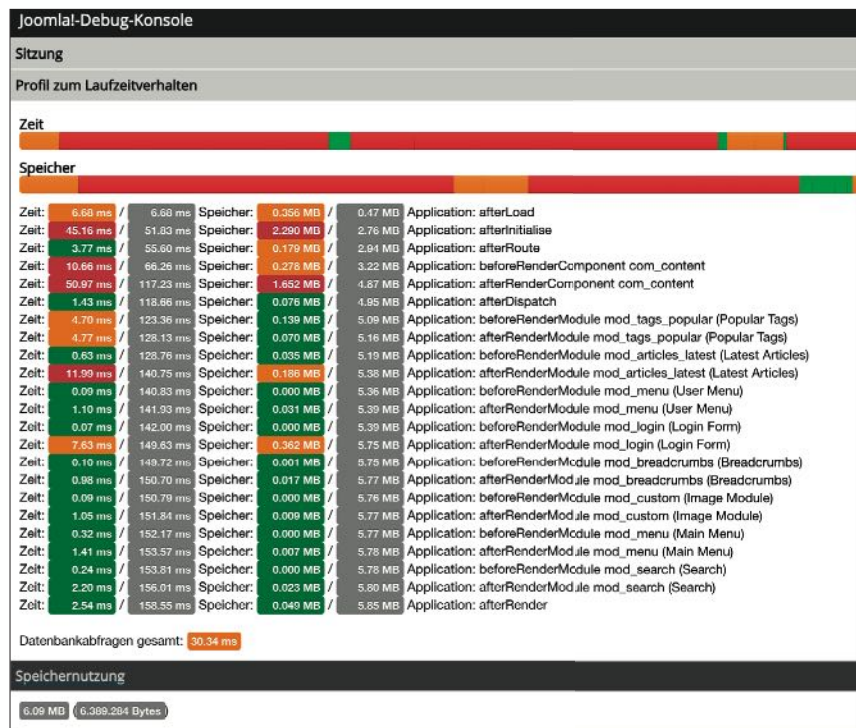


Abb.3: Profil zum Laufzeitverhalten unter PHP 5.6, Joomla! 3.5.1

Abbildung 7 aussehen und gibt eine Auflistung über die gecachten Dateien. Das normale Caching eignet sich hervorragend bei Seiten mit sehr vielen unterschiedlichen Modulen je Seite und für Internetauftritte ohne Log-in-Bereich. Beim normalen Caching-Verfahren wird jedes Modul einzeln und pro Besucher

session in einer eigenen Cache-Datei gespeichert. Das heißt natürlich auch bei vielen Besuchern und sehr vielen Modulen, dass man ggf. sehr viele Cache-Dateien hat. Von Haus aus werden Joomla!-Standard-Komponenten wie com_content, com_languages, com_templates und einige weitere unterstützt.

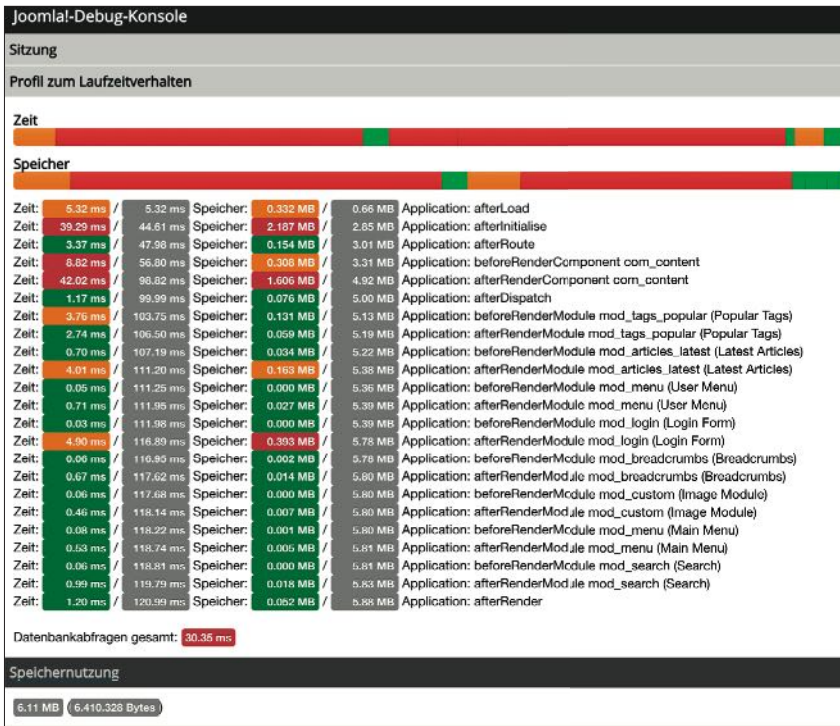


Abb.4: Profil zum Laufzeitverhalten unter PHP 7, Joomla! 3.5.4

	PHP 5.5	PHP 5.6	PHP 7
afterInitialise	71.48 ms	45.16 ms	39.29 ms
afterRender	3.38 ms	2.54 ms	1.20 ms
Gesamtzeit	185.13 ms	158.55 ms	120.99 ms
Datenbankabfragen	34.43 ms	30.34 ms	30.35 ms
Speichernutzung	6.1 MB	6.09 MB	6.11 MB

Tabelle 1: Profil zum Laufzeitverhalten unter PHP 5.5, PHP 5.6 und PHP 7

Konfiguration:

System » Konfiguration » Tab System » Option Zwischenspeicher (Cache) » An – Normales Caching

Cache:

jeden View einer Komponente und jedes Modul separat

Basierend auf:

URL, View, Parametern, ...

Hinweise: Für Extension-Entwickler heißt es an dieser Stelle, einen Blick in die Dokumentation zu werfen und für die eigene Erweiterung das Caching zu implementieren (<http://einfach.st/apijoom2> bzw. <http://einfach.st/jom2>). Einzelne Module können bei Bedarf per Modulparameter individuell konfiguriert werden, um sie vom Caching auszunehmen und so Fehlverhalten zu vermeiden. Das normale Caching wirkt sich nur auf das Frontend aus und ist somit nur für Seitenbesucher/

Gäste und nicht für eingeloggte Benutzer verwendbar.

An – Erweitertes Caching geht einen Schritt weiter und eignet sich für Seiten mit ähnlichen Modulkompositionen. Beim erweiterten Caching werden Modulgruppen in einer Cache-Datei gespeichert und nicht mehr separat. So können Seitenbesucher/Gäste den Cache gemeinsam nutzen. Weiterhin ermöglicht das erweiterte Caching auch für eingeloggte Frontend-Benutzer ab dieser Methode das Caching.

Konfiguration:

System » Konfiguration » Tab System » Option Zwischenspeicher (Cache) » An – Erweitertes Caching

Cache:

größerer System-Cache und Modul-Cache

Basierend auf:

URL, View, Parametern, ...

Cache-Speicher

Joomla! bietet neben der Möglichkeit, den Cache in einer Datei zu speichern, einige weitere spannende Speicherunterstützungen wie APC, Eaccelerator, Memcache, Redis oder XCache an. Die Auswahl kann wie in Abbildung 8 aussehen und ist abhängig vom Internet-Hoster, der den Speicher anbieten muss. Gängig sind APC-Derivate, Memcache und Redis.

Hinweis: Den Cache in einer Datei zu speichern, sollte bei jedem Hoster funktionieren. Wichtig ist, darauf zu achten, dass das Cache-Verzeichnis Joomla!-ROOT/cache nicht schreibgeschützt ist. Das Zwischenspeichern in einer Datei ist der „langsamste Speicher“, da immer von der Festplatte gelesen werden muss. Bei vielen Besuchern auf einer umfangreichen Seite mit vielen Cache-Dateien kann dies zum Flaschenhals werden. Von allen Möglichkeiten sind Memcache und Redis die schnellsten.

Plattformspezifischer Cache

Aktiviert oder deaktiviert den plattformspezifischen Cache. Aktivieren, wenn das generierte HTML sich für verschiedene Geräte unterscheidet (Standard: deaktiviert).

Cache-Dauer

Die maximale Dauer in Minuten, die eine Cache-Datei gespeichert bleibt, bevor sie aktualisiert wird. Ändert sich kaum etwas an der Seite, ist die Cache-Dauer für den Livebetrieb höher einzustellen. Standard sind 15 Minuten.

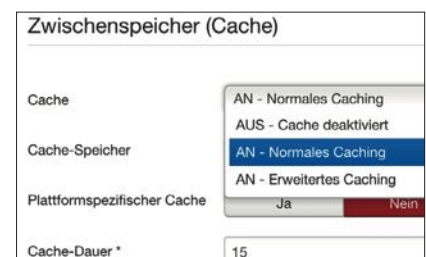


Abbildung 5: Zwischenspeicher (Cache) – Einstellungen in der Konfiguration von Joomla!

TIPP

Entwickler aufgepasst. Bei Aktivierung des Plug-in-Seitencache ist kein Profiling (System debuggen aktiviert) im Frontend mehr möglich.

Während der Entwicklung empfiehlt es sich, die Cache-Dauer gering zu halten bzw. das Caching zu deaktivieren. Andernfalls muss man über das Backend hin und

wieder den Cache leeren, um Änderungen sofort zu sehen. Will man sich die Navigation durch das Backend sparen, gibt es zwei grundsätzlich ähnlich funktionierende Drittanbieter-Plug-ins, welche die „Cache-Leeren-Funktion“ per Klick im Backend von jeder Stelle aus ermöglichen und Zeit sparen: zum einen das Cache-Cleaner-Plug-in von RegularLabs (<http://einfach.st/jcc2>) und als weiteres das Plug-in Quick Cache Cleaning von Kubik Rubik (<http://einfach.st/joomext>). Beide sind in Free-Versionen erhältlich.

Core-Plug-in-System – Seitencache

Das bereits mit einer Joomla!-Standardinstallation vom Core zur Verfügung gestellte System-Plug-in „System – Seitencache“ ist standardmäßig deaktiviert und kann über Erweiterungen » Plugins » System – Seitencache aktiviert werden. Bis zur Version 3.5 war dies leider nur für statische Seiten ohne Log-in-Bereich oder Formulare anzuraten. Nach Aktivierung des Plug-ins wird jede Seite als komplettes HTML-Dokument im Speicher abgelegt, was bei dynamischen Seiten schon mal zum handfesten Problem führen kann. Dynamische Inhalte wie bspw. Formulare können nicht mehr auf Benutzereingaben reagieren und versendet werden. Schlimmer noch sind Formularinhalte vom ersten Absender. Hierbei ist noch zu beachten, dass das normale und das erweiterte Caching mit Aktivierung des Plug-ins außer Kraft gesetzt und übergangen werden, auch wenn sie konfiguriert und aktiviert sind.



Abbildung 6: Cache leeren (Site) – keine passenden Ergebnisse bei deaktiviertem Cache

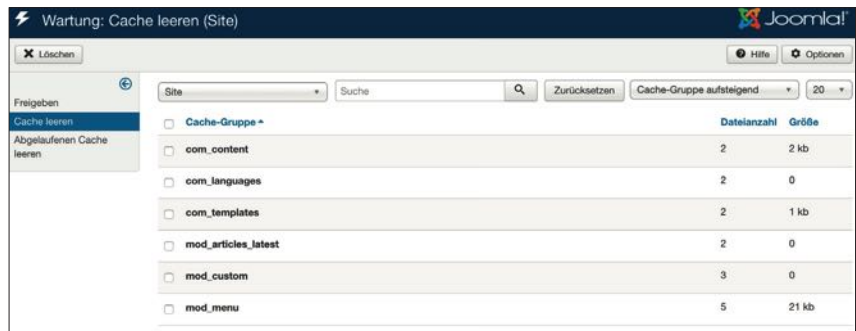


Abbildung 7: Cache leeren (Site) – erste Cache-Inhalte bei An - Normales Caching

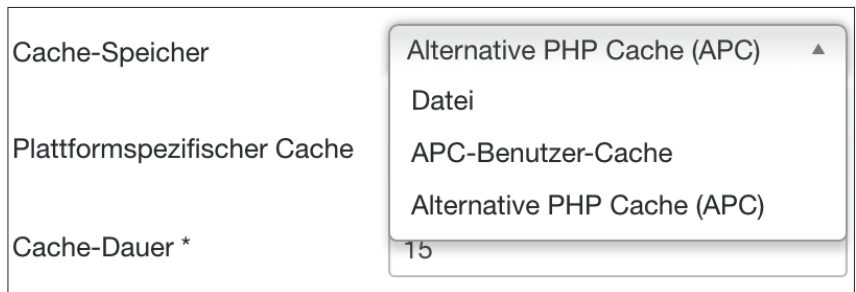


Abbildung 8: Cache-Speicher mit den vom Hostler unterstützten Optionen

Neu ab Joomla! 3.5 sind nun die Möglichkeiten, im Plug-in direkt Menüeinträge und im Tab Erweitert zeilenweise URLs vom Caching auszuschließen. So können zum Beispiel der Menüeintrag mit dem Formular und die Dankeseite vom Caching ausgeschlossen werden und man erhält so eine sehr schnelle Seite mit dynamischen Elementen.

Konfiguration:

Erweiterungen »

Plugins » System – Seitencache

Cache:

jede einzelne Seite als komplette Datei

Basierend auf:

URL

Hinweise: Die optionale Einstellung Browser-Cache wird berücksichtigt. Der Cache ist nur für Seitenbesucher/Gäste und nicht für eingeloggte Benutzer im Frontend nutzbar.

JCH-Optimize

Erhältlich als Free- und Paid-Version ist das Plug-in JCH-Optimize von Entwickler Samuel Marshall, ein wirkliches Caching-Turbo-Magic-Tool. Durch das Ausführen einer Anzahl an Front-End-Optimierungen beschleunigt es signifikant die Ladezeit der Webseite. Zum Hauptfunktionsumfang gehört das Zusammenfassen von CSS- und JavaScript-Dateien, Minifizieren des HTML-Outputs, automatisiertes Erstellen von Sprites, Lazy Load für Bilder, CDN und Cookieless-Domain-Unterstützung und vieles mehr. Einen großen Vorteil bietet das Plug-in durch sechs vordefinierte, aber anpassbare Konfigurationssets, wie in Abbildung 9 und 10 zu sehen.

Eine vollständige Liste der Plug-in-Optionen mit detaillierten Beschreibungen ist unter <http://einfach.st/jplug3> zu finden.



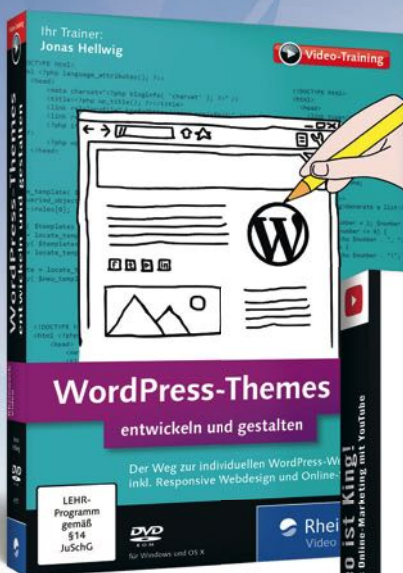
Abbildung 9: Sechs vordefinierte Konfigurationssets von jch-optimize

	Minimum	Intermediate	Average	Deluxe	Premium	Optimum
Automatic Basic Settings						
Combine CSS Files	Yes	Yes	Yes	Yes	Yes	Yes
Combine Javascript Files	Yes	Yes	Yes	Yes	Yes	Yes
Gzip javascript and css	No	Yes	Yes	Yes	Yes	Yes
Minify CSS	No	Yes	Yes	Yes	Yes	Yes
Minify javascript	No	Yes	Yes	Yes	Yes	Yes
Minify HTML	No	Yes	Yes	Yes	Yes	Yes
Defer javascript	No	No	No	No	No	Yes
Manage combined files	Preserve execution order	Preserve execution order	Preserve execution order	Combine files in one	Place javascript at bottom	Place javascript at bottom
Automatic Exclude Settings						
Include all extensions	No	No	Yes	Yes	Yes	Yes
Automatic Pro Settings						
Replace @imports in CSS	Yes	Yes	Yes	Yes	Yes	Yes
Include PHP and external files	Yes	Yes	Yes	Yes	Yes	Yes
Include inline CSS files	Yes	Yes	Yes	Yes	Yes	Yes
Include inline javascript files	No	No	No	Yes	Yes	Yes
Search body	No	No	No	No	Yes	Yes
Load javascript asynchronously	No	No	No	No	No	Yes

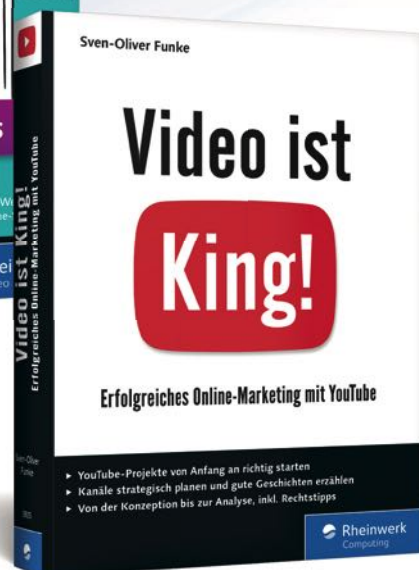
Abbildung 10: Übersicht der automatisch gesetzten Parameter von jch-optimize per Set

Fazit

Das Aktivieren von Cache im Allgemeinen und im Speziellen durch Nutzen von Core oder Drittanbieter-Plug-ins lohnt sich bereits bei kleinen Unerfahrenen Seitenbetreibern, die nicht auf die Optimierung verzichten wollen, sei das Plug-in JCH-Optimize ans Herz gelegt. Die Konfigurationssets ermöglichen den einfachen Einstieg in das Thema. ¶



Know-how für Online-Marketer und Webentwickler!



Entdecken Sie unser neues Programm: Online-Marketern empfehlen wir *Video ist King* – so gelingt Ihnen erfolgreiches Marketing auf YouTube. Webentwickler können sich über *WordPress-Themes, PHP 7 und MySQL* sowie *Node.js* freuen. Unsere Bücher und Video-Trainings sind auch als E-Book bzw. Download erhältlich.

www.rheinwerk-verlag.de

