

Martin Günther

»Suchmaschinenoptimierung für TYPO3 – Teil 3

Im letzten Teil dieser Serie wurde gezeigt, wie Sie mit [TYPO3*](#) sprechende URLs erstellen. Dieser Teil knüpft daran an und erklärt Ihnen das Generieren sprechender URLs für Bilder. Außerdem lesen Sie, wie Sie mit TYPO3 echte statische Seiten erstellen und somit noch mehr Geschwindigkeit aus dem System herausholen können.

Sprechende URLs für Bilder

Die Vorteile sprechender URLs wurden bereits in der letzten Ausgabe erläutert und es wurde gezeigt, wie man diese mit TYPO3 generieren kann. Webseiten enthalten neben der URL zahlreiche weitere Inhalte (META-Tags, Überschriften, Seiteninhalte etc.), die es der Suchmaschine ermöglichen, das Thema einer Webseite zu erkennen. Bei Bildern ist das nicht der Fall. Hier steht den Suchmaschinen nur eine begrenzte Auswahl an Informationen zur Verfügung, z. B. das sog. Alt-Tag für die Alternativbezeichnung, Linktexte interner und externer Backlinks, das Bild umfließender Text und der Dateiname bzw. die URL des Bildes. Deshalb ist es bei Bildern besonders wichtig, einen keywordrelevanten und sprechenden Dateinamen zu hinterlegen und dieses Potenzial nicht durch kryptische, vom CMS generierte Dateinamen zu verschweigen – so wie es TYPO3 leider in der Standardkonfiguration anbietet. Ohne Anpassungen werden die Dateinamen von Bildern in der Form `typo3temp/pics/69e32eab93.jpg` generiert. Dies geschieht immer dann, wenn die Abmessungen des Bildes beim Upload verändert werden.

RealURL für Bilder installieren & konfigurieren

Analog zur Installation von RealURL muss auch die Erweiterung „Image RealURL“ aus dem TYPO3 Extension Repository zunächst heruntergeladen und dann installiert werden – die aktuelle Version finden Sie unter <http://einfach.st/t3bilder>. Nach der Installation liefert die Erweiterung bereits erste Ergebnisse. Ab sofort werden Dateinamen in der Form `typo3temp/fl_realurl_image/sprechender-dateiname-xx.jpg` generiert. Das „xx“ am Ende des Da-

teinamens wird dabei durch einen sog. Hash-Wert ersetzt. Der Hash-Wert, dessen Länge über [TypoScript*](#) angepasst werden kann, soll sicherstellen, dass der Dateiname über das gesamte System hinweg auch wirklich eindeutig identifizierbar ist.

In der Standardkonfiguration der Erweiterung ist es so, dass der im TYPO3 hinterlegte Alt-Tag eines Bildes herangezogen wird, um den sprechenden Dateinamen zu generieren. Wurde kein Alt-Tag vergeben, wird der Dateiname des hochgeladenen Bildes verwendet. Vergibt der Nutzer zu einem späteren Zeitpunkt doch einen Alt-Text oder ändert diesen nachträglich, so ändert sich auch automatisch der Dateiname des Bildes in TYPO3 bzw. auf der Website. Diese Einstellung sollte daher dringend angepasst werden. Vermutlich fragen Sie sich, ob der Dateiname, analog zu den sprechenden URLs, nicht einfach frei vergeben werden kann. Das ist in der Form leider nicht möglich. Die einzige Möglichkeit der Einflussnahme besteht darin, den Wert anzupassen, der zur Generierung des Dateinamens herangezogen wird. Dies geschieht über TypoScript im Konstanten-Editor (Abbildung 1).

Über das Attribut `imageLinkName.data` (rote Markierung) wird diese Einstellung vorgenommen. Dabei gilt: Die einzelnen Werte, getrennt durch „/“, werden in der angegebenen Reihenfolge herangezogen, um den Dateinamen zu generieren. Ist der erste Wert nicht belegt, wird der zweite verwendet und so weiter. Hier sollte der Wert `file:origFile` an die erste Stelle vorgezogen werden. Damit hat man die Möglichkeit, durch Vergabe eines sinnvollen Dateinamens vor dem Upload den durch TYPO3 generierten Dateinamen gezielt zu beeinflussen.



DER AUTOR



Martin Günther arbeitet als Website-Boosting-

Engineer für die tms development GmbH in Nürnberg und beschäftigt sich seit mehreren Jahren professionell mit den Themen Online-Marketing und Web-Programmierung.

* siehe Glossar Seite 96-98

Den Pfad der Bilder anpassen

Wie im Beispiel weiter oben zu sehen ist, werden die Bilder im Verzeichnis `typo3temp/fl_realurl_image/` abgelegt. Auch hier besteht die Möglichkeit der Einflussnahme. Man kann die Bild-URL beispielsweise so generieren lassen, dass alle Bilder scheinbar im [Root-Verzeichnis*](#) (`www.eine-domain.de/sprechender-dateiname-xx.jpg`) der Domain liegen. Hierfür muss zunächst das Attribut `virtualPathRemove` in den Erweiterungs-Manager Einstellungen von Image RealURL auf `„typo3temp/fl_realurl_image/“` gesetzt werden. Als Nächstes muss die nun ungültige Bild-URL (physikalisch liegen die Bilder nach wie vor im Unterverzeichnis `typo3temp/fl_realurl_image/`) in der `.htaccess`-Datei umgeschrieben werden. Die nötigen Code-Zeilen sehen Sie in dem unten stehenden Code-Schnippssel.

Wie bereits im letzten Artikel finden Sie auch diesmal eine (für Image RealURL angepasste) Vorlage der `.htaccess`-Datei auf unserem Webserver, von wo aus Sie den Code einfach kopieren und anpassen können (<http://einfach.st/htpic>).

Wichtig ist, dass Sie dabei nicht vergessen dürfen, auch die Alt-Tags Ihrer Bilder mit sinnvollen Inhalten zu belegen. Verwenden Sie hierfür das Feld „Alternativer Text“ in den Einstellungen des Content-Elements (siehe Hinweis).

Mit den gezeigten Einstellungen haben Sie eine gute Basis für Ihre Bilderoptimierung. Weitere Einstellungsmöglichkeiten der Erweiterung Image RealURL finden Sie in der Dokumentation von TYPO3 unter <http://einfach.st/t3pd>.

Mit TYPO3 den Turbo einschalten

Wie beschleunigt man den Webseitenaufbau beim Besucher mit TYPO3 wohl am besten? Überspitzt gesagt, indem man es erst gar nicht als CMS nutzt! Genauer gesagt, indem man die typischen Eigenschaften eines CMS umgeht, dass Seiten bzw. deren einzelne Elemente eben erst bei der Anforderung durch den Besucher aus Datenbankfeldern geholt und zusammgebaut werden müssen (jede einzelne Datenbankabfrage kostet Zeit und das kann sich spürbar zu entsprechend störenden Wartezeiten summieren – je nach aktueller Auslastung des Web- und Datenbankservers).

Genau diesen Ansatz verfolgt die TYPO3-Erweiterung Static File Cache (<http://einfach.st/static>). Diese Erweiterung produziert nämlich die Webseiten einmalig beim ersten Aufruf und legt sie dann als echte statische Seiten direkt auf dem Webserver ab. Der Abruf einer einzelnen, bereits fertigen HTML-Seite geht nämlich deutlich schneller als die vielen

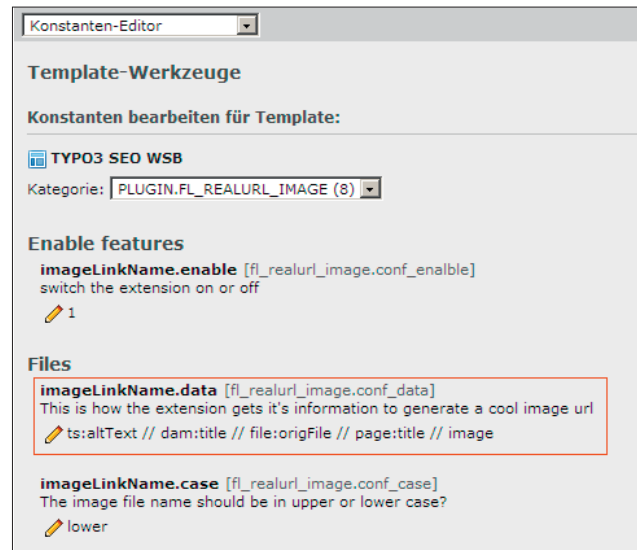


Abbildung 1: Über den Konstanten-Editor kann der Dateiname beeinflusst werden

HINWEIS:

Alternativer Text:

► Hier steht der Alt-Text des ersten Bildes
Hier steht der Alt-Text des dritten Bildes

Für jedes Bild innerhalb des Content-Elements müssen Sie bei der Vergabe des Alternativ-Textes eine eigene Zeile verwenden! Möchten Sie beispielsweise für das erste und dritte, aber nicht für das zweite Bild einen Alternativ-Text vergeben, so müssen Sie, wie in der Abbildung beispielhaft gezeigt, die zweite Zeile leer lassen.

einzelnen Datenbankabfragen, die im Hintergrund laufen und jedes Mal neu ausgeführt werden müssen – bei jedem Seitenaufruf! Dies macht aber nur Sinn, wenn auf den Seiten keine echten dynamischen Inhalte verwendet werden, die wirklich aktuell und sich verändernd aus der Datenbank generiert werden müssen. Bei den meisten Webauftritten ist dies aber nicht der Fall, mit Ausnahme einzelner Bereiche mit dynamischen Inhalten. Einschränkungen gibt es weiterhin bei der zeitgesteuerten Publikation einzelner Seiten. Wer also vorher Webseiten anlegt und diese nicht sofort, sondern zu einem genau definierten Zeitpunkt, automatisch publizieren lassen möchte, muss diese Seiten vom Caching ausschließen. Das gilt auch für Seiten, die sog. TypoScript-Bedingungen enthalten. Der eigentliche Vorteil dieser Erweiterung liegt also

.htaccess

```
RewriteCond %{DOCUMENT_ROOT}/typo3temp/fl_realurl_image%{REQUEST_URI} -f [OR]
RewriteCond %{DOCUMENT_ROOT}/typo3temp/fl_realurl_image%{REQUEST_URI} -l
RewriteRule ^(.*)$ typo3temp/fl_realurl_image/$1 [L]
```

* siehe Glossar Seite 96-98

Konfiguration:

(Hinweis: Nach der Konfiguration der Erweiterung kann ein Leeren des Cache erforderlich sein. Dies ist dar

Clear cache for all domains in tree [clearCacheForAllDomains]
When the TYPO3 tree hosts more than one domain, each domain has it's own cache branch. When clearing the cache, only the cache for the domain you are logged on to will be cleared. If you would like the cache for all domains to be cleared when clearing the frontend cache, then check this box.

Send Cache-Control header [sendCacheControlHeader]
If your apache installation has 'mod_expires' loaded, you can use it to send Cache-Control headers. For each statically cached page a .htaccess file will be created with a mod_expires rule. This rule will make apache send a cache header when delivering the cached page to the visitor. Please note that this will only work with a clean realurl setup (paths ending with a '/').

Show generation timestamp signature [showGenerationSignature]
This appends a comment at the end of the static html file. This can be usefull for debugging purposes (look at the source and see if the comment is present).

Timestamp format [strftime]
The date format, formatted according to your locale.

Log debug information to cc_devlog [debug]
If enabled, logging info will be written to cc_devlog. So make sure you have this installed if you have this option checked.

Abbildung 3: Einstellungsmöglichkeiten bei der Installation von Static File Cache.

Seite "Startseite - TYPO3 SEO" bearbeiten

Allgemein Metadaten Ressourcen Optionen Zugriff **Erweitert**

Speichere Seitencache als statische Datei

Abbildung 4: Option in den Seiteneigenschaften nach Installation von Static File Cache.

darin, den trägen TYPO3-Apparat zu umgehen und die Seiten direkt und als fertige HTML-Datei zur Verfügung zu stellen.

Während der Installation über den Erweiterungsmanager sollten zwei Einstellungen vorgenommen werden (Abbildung 3). Wer mit „Cache-Control Headers“ nicht hundertprozentig vertraut ist, sollte „Send Cache-Control header“ unbedingt deaktivieren, da es sonst zu unerwünschten Nebenwirkungen kommen kann. Die Einstellung „Show generation timestamp signature“ führt dazu, dass im Quellcode der gecachten Seite ein Zeitstempel (`<!-- cached statically on: 13-10-10 19:05 -->`) mit ausgegeben wird, und zeigt so, dass die Erweiterung korrekt arbeitet. (Hin-

weis: Die gecachte Seite wird nicht angezeigt, solange man am TYPO3 Backend angemeldet ist. Es empfiehlt sich daher, einen anderen Browser für den Funktionstest zu verwenden!) Nach der Installation muss noch eine Weiche eingerichtet werden, die das TYPO3-System außen vor lässt und stattdessen die gecachte Seite an den Browser übermittelt. Hierfür wird die .htaccess-Datei erweitert, da die Weiche bereits auf der Webserverebene eingreifen muss. Die Erweiterung bringt eine ausführlich kommentierte Beispieldatei mit sich (.htaccess im Ordner `nc_staticfilecache/doc`), aus der alle notwendigen Einstellungen entnommen werden können.

Mit der Installation dieser Erweiterung ist eine weitere Option in den Sei-



teigenschaften hinzugekommen. Diese findet man unter „Erweitert“, mit der Aufschrift „Speichere Seitencache als statische Datei“ (Abbildung 4). Diese ist standardmäßig aktiviert. Wie der Name schon vermuten lässt, kann hier festgelegt werden, ob die jeweilige Seite als statische Datei gecached werden soll oder nicht. Falls eine Seite echte dynamische Inhalte enthält, was die Erweiterung im Regelfall bis auf zwei Ausnahmen zuverlässig erkennt, kann sie damit explizit vom Caching ausgeschlossen werden. Diese Ausnahmen sind, wie oben schon erwähnt, Seiten mit zeitgesteuerter Publikation und/oder TypoScript-Bedingungen, die manuell vom Caching mittels Deaktivierung ausgenommen werden müssen.

Die Dokumentation zu Static File Cache (<http://einfach.st/filecache>) beschreibt detailliert alle wesentlichen Einstellungsmöglichkeiten und verspricht ein maximales Geschwindigkeits-Boosting um den Faktor 230. Ob man diesen Wert tatsächlich mit allen Systemen erreicht, hängt dabei von vielen Faktoren ab. Tests der Redaktion haben gezeigt, dass sich eine Steigerung um den Faktor 50 bis 100 jedoch recht einfach erreichen lässt. Webseiten, die sich hundertmal schneller aufbauen als vorher, erhalten nicht nur von Suchmaschinen, sondern auch von den Besuchern deutlich mehr Liebe. ¶