

Oliver Thiele

TYPO3 7: Fit für SEO?

Das Content-Management-System TYPO3 bietet eine enorme Flexibilität in vielen Bereichen. Trotzdem nutzt kaum jemand diese Flexibilität, um seine Website für Google & Co. zu optimieren – dabei ist es für einen TYPO3-Entwickler oft nur ein geringer Aufwand, die wichtigsten Einstellungen für die OnPage-Optimierung vorzunehmen. Die neue TYPO3-Version 7 bietet zudem einige neue und interessante Optimierungsmöglichkeiten.

Im Gegensatz zu WordPress erscheint nach der Installation von TYPO3 nicht sofort ein schönes Layout, das einfach durch ein neues Theme ausgetauscht werden kann, sondern nur eine mehr oder weniger leere Seite. „Out of the box“ wird bei TYPO3 in Sachen SEO also nichts vorkonfiguriert. Das TYPO3 Extension Repository (TER) ist ein zentrales Verzeichnis, in dem Extensions gespeichert sind, welche über den Extension-Manager in einem TYPO3-System installiert werden können. Entwickler haben die Möglichkeit, eigene Extensions hochzuladen und diese somit der gesamten TYPO3-Gemeinde zur Verfügung zu stellen. Im TER gibt es einige, teilweise umfangreiche SEO-Extensions, die jedoch nicht immer optimal zu den eigenen Anforderungen passen oder so veraltet sind, dass sie in aktuellen TYPO3-Versionen nicht mehr funktionieren. Wenn die Website beispielsweise auch auf Anforderungen wie Lokalisierung, Internationalisierung und umfangreiche Extensions mit dynamischen Listenansichten optimiert werden soll, ist der Entwickler bei TYPO3 gezwungen, sich selbst um die OnPage-Optimierung zu kümmern. Man könnte das als Nachteil ansehen – im Grunde ist es aber ein Vorteil, da so die Optimierung auf die Gegebenheiten der eigenen Website feinjustiert werden kann und der Webentwickler zeitnah auf Änderungen der Suchalgorithmen reagieren kann. Um einen Einblick in die OnPage-Optimierung in TYPO3 zu geben, geht dieser Artikel auf viele OnPage-Basics wie Meta-Angaben, URL-Handling, Speed-Optimierung

oder semantische Auszeichnung ein.

Eines vorweg: Im Artikel finden Sie Links zu Code-Beispielen und zu einer selbst geschriebenen Extension, in der einige der genannten Punkte umgesetzt sind. Zur Weiterbildung kann auch der Source-Code der SEO-Extensions im TYPO3 Extension Repository (TER) dienen.

1. Die wichtigsten SEO-beeinflussenden Neuerungen der TYPO3-Versionen 6 und 7

Ende des Jahres erscheint das nächste Major Release von TYPO3, die Long-Term-Support-Version TYPO3 7.6. Der folgende Artikel bezieht sich auf die Version 7.4. Viele Tipps und Tricks werden in älteren TYPO3-Versionen aber auch so oder ähnlich durchgeführt.

Ab der TYPO3-Version 6.0 wurde z. B. der „File Abstraction Layer“ (FAL) eingeführt. Damit können die Meta-Daten von Dateien nun zentral in der Dateiliste im Backend gepflegt werden. So entfällt für Redakteure die Notwendigkeit, die Meta-Daten für z. B. Bilddateien selbst einzupflegen. Eine manuelle Überschreibung pro Verwendung ist natürlich immer möglich. Damit sollten Bilder ohne Alt-Tags der Vergangenheit angehören. Darüber hinaus ist es mit dem FAL möglich, statische Dateien wie Bilder, Stylesheets und JavaScripts in einem Content Delivery Network (CDN) zu speichern. Die Daten können dann meist schneller ausgeliefert werden.

Die optimale Darstellung einer Website auf mobilen Endgeräten ist auch für die Suchmaschi-

DER AUTOR



Oliver Thiele ist „Certified TYPO3 Integrator“ und seit 2003 Dozent für die TYPO3-Schulungen beim Agenturhoster Mittwald. Er ist mehrfach im Jahr auf diversen TYPO3-Events anzutreffen.

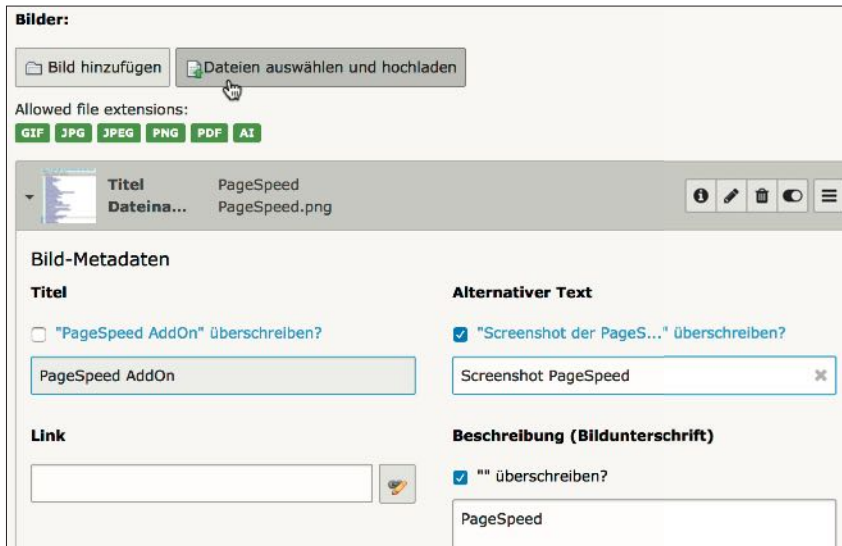


Abb. 1: Meta-Angaben für Bilder können zentral in der Dateiliste definiert, bei Bedarf aber ergänzt/ überschrieben werden

```

1 page {
2     meta {
3         og:site_name = Website Boosting
4         og:site_name.attribute = property
5         description.data = TSFE:page|description
6         og:description.data = TSFE:page|description
7         og:description.attribute = property
8         # ...
9     }
10 }

```

Abb. 2: TypoScript-Auszug für die Open-Graph-Integration.

```

1 <meta property="og:site_name" content="TYPO3-SEO">
2 <meta name="description" content="Beschreibung der Seite">
3 <meta property="og:description" content="Beschreibung der Seite">

```

Abb. 3: Ausgabe der Meta-Daten in HTML

nenoptimierung relevant. Bisher war ein Responsive Design in TYPO3 mit einigen Umwegen (z. B. über die Extension „Grid Elements“) möglich. In einer der nächsten TYPO3 7 Versionen sollen dem TYPO3-Core „Structured Content Containers“ hinzugefügt werden, mit denen Inhaltelemente gruppiert werden können. Einem responsiven Webseiten-Layout sollte damit auch ohne den Einsatz von Erweiterungen nichts mehr im Wege stehen.

2. Title und Meta-Descriptions

Der Seitentitel und die Meta-Description können in TYPO3 relativ einfach ausgegeben werden. Für den Title verwendet

TYPO3 standardmäßig den Seitentitel. Die Meta-Descriptions können im Reiter „Metadaten“ in den Seiteneigenschaften angegeben werden. Damit die Meta-Descriptions (oder weitere Meta-Daten) im HTML-Code der Seite ausgegeben werden, muss jedoch noch etwas TypoScript-Code eingefügt werden, wie im folgenden Beispiel gezeigt wird.

Open-Graph-Unterstützung

Da TYPO3 seit Version 7.4 die Verwendung von Open Graph Attribut-Namen unterstützt, können diese ebenfalls für die Meta-Daten-Auszeichnung verwendet werden. Hier ein Auszug aus dem TypoScript, wie diese auf der Seite ausgegeben werden können (Abb. 2).

In diesem Beispiel wird zunächst für alle Seiten der Titel der Webseite statisch definiert. Danach wird aus der Eigenschaft der aktuellen Webseite die Beschreibung ausgelesen und sowohl für die normale Meta- als auch für die Open-Graph-Description gesetzt. In älteren TYPO3-Versionen musste hier noch mit der TypoScript-Eigenschaft „page.headerData“ gearbeitet werden. Mit dieser Eigenschaft können beliebige HTML-Code-Ausgaben, wie zum Beispiel Angaben zur Altersklassifizierung oder Meta-Angaben für Twitter, im Head der Website definiert werden.

Mit `xxx.data = TSFE:page|xxx` werden die Daten aus einem Array gelesen. Viele Tutorials verwenden hier `xxx.field = xxx`, was den Nachteil hat, dass man in eigenen Extensions in der Detailansicht nicht so einfach die Description der Seite mit der Description des Extension-Datensatzes überschreiben kann.

Das Ergebnis in der HTML-Code-Ausgabe sähe aus, wie in Abbildung 3 zu sehen.

Möchte man weitere Meta-Angaben wie z. B. `noindex`, `nofollow` oder eine manuelle Definition der Canonical-URL machen, muss TYPO3 erst modifiziert werden, damit ein Redakteur bei den Seiteneigenschaften Eingabefelder pflegen kann. Weitere Informationen dazu finden Sie unter der Überschrift URL-Handling und in der Beispiel-Extension „ot_seo“, unter <http://einfach.st/mwseo1>.

3. Semantische Auszeichnung Template-Anpassungen und eigene Inhaltelemente mit Microdata, JSON-LD & RDFa

TYPO3 gibt von Haus aus keine Inhalte mit Schema-Daten aus. Mit eigenen Inhaltelementen und Extensions ist es aber möglich, beliebige Markups zu generieren.

Extbase besteht aus einem Grundgerüst von PHP-Klassen, die zur Entwicklung von TYPO3 Extensions genutzt

werden können. Viele Extbase-basierte Erweiterungen, wie beispielsweise „News“ (nicht zu verwechseln mit „tt_news“), können über eine TypoScript-Konfiguration andere Template-Dateien aus selbst angelegten Ordnern verwenden.

In der finalen TYPO3-Version 7 LTS wird auch das Rendering von Inhaltselementen überarbeitet. In Version 7.4 wurden dafür schon Daten-Prozessoren eingeführt, die über TypoScript aufgerufen werden können. Möchte man z. B. eine große Zahl semantischer Daten in die Datenbank importieren, um beispielsweise einen Produktkatalog auszuzeichnen, können die Daten mithilfe der Daten-Prozessoren über eine Komma-separierte Datei ausgelesen und in Arrays umgewandelt werden. Im Fluid-Template könnte dann mit einer Foreach-Schleife das Array als HTML-Tabelle mit Mikrodaten ausgegeben werden. Dies war vorher nur sehr umständlich möglich.

Breadcrumb mit Schema-Auszeichnung

Die Definition eines Brotkrumenpfads erfolgt bei TYPO3, wie auch bei den meisten Menüs, über TypoScript. Daher ist es ein Leichtes, die Schema-Angaben der Ausgabe des HTML-Codes hinzuzufügen. Es gibt laut Google (<http://einfach.st/gbc3>) verschiedene Möglichkeiten, die Auszeichnung zu realisieren. Da die Website den Breadcrumb i. d. R. ausgeben soll, ist es sinnvoll, die RDFa-Version zu verwenden. Um in TYPO3 einen Brotkrumenpfad wie in der Google-Dokumentation beschrieben auszuzeichnen, haben wir ein Beispiel-Code-Snippet erstellt.

Ein korrekt ausgezeichnete Breadcrumb hilft nicht nur Google dabei, die Struktur einer Website besser zu verstehen, sondern schlägt sich auch in einer optisch ansprechenden Darstellung in den SERPs nieder.

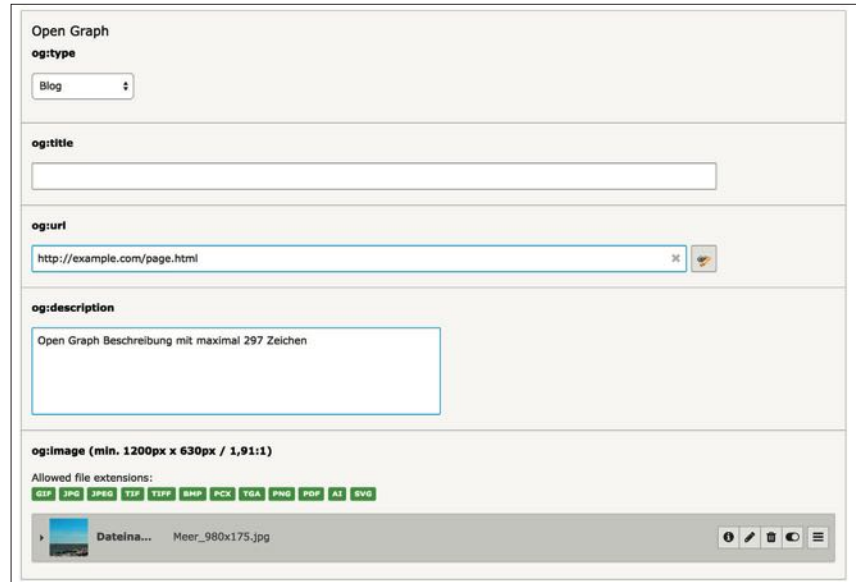


Abb. 4: Eingabefelder in der Extension ot_seo

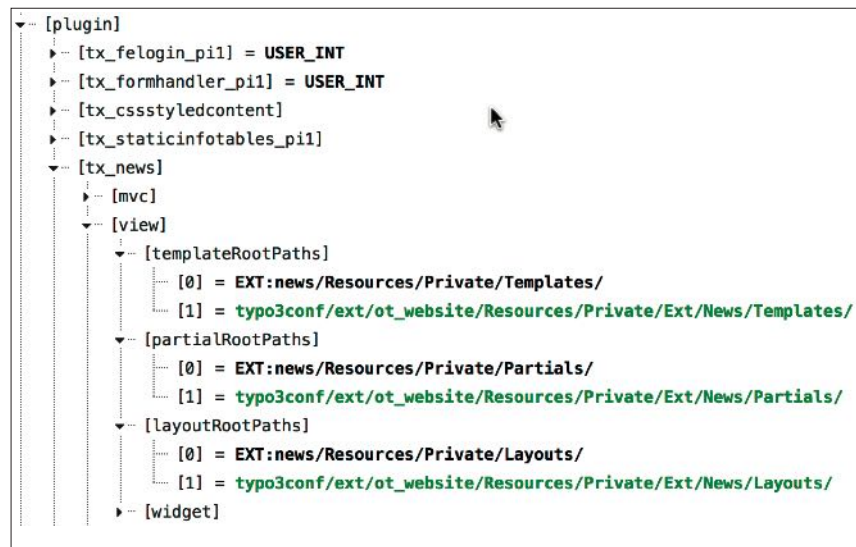


Abb. 5: Template-Dateien können Update-sicher in anderen Ordnern (grün) modifiziert werden; dort können dann auch Schema-Auszeichnungen realisiert werden

```

1 lib {
2   breadcrumb = HMFNU
3   breadcrumb {
4     special = rootline
5     special.range = 0|-1
6     1 = TMFNU
7     1 {
8       wrap = <ol class="breadcrumb" vocab="http://schema.org/" typeof="BreadcrumbList">|</ol>
9       NO = 1
10      NO {
11        before.cObject = LOAD_REGISTER
12        before.cObject {
13          tempN.data = register:LIN
14          tempN.wrap = |+1
15          LIN.data = register:tempN
16          LIN.prioriCalc = 1
17        }
18        wrapItemAndSub = <li property="itemListElement" typeof="ListItem">|</li>
19        stdWrap.wrap = <span property="name">|</span>
20        linkWrap = |<meta property="position" content="{register:1IN}">
21        AllagParams = property-"item" typeof-"WebPage"
22        stdWrap.htmlspecialchars = 1
23        allStdWrap.insertData = 1
24      }
25      CUR = .NO
26      CUR {
27        wrapItemAndSub = <li class="active" property="itemListElement" typeof="ListItem">|</li>
28        doNotlinkIt = 1
29      }
30    }
31  }
32 }
33

```

Abb. 6: Brotkrumenpfad-Auszeichnung in TYPO3 (Code verfügbar unter <http://einfach.st/mwc2>)

Prototypen testen – Design Thinking Part 4 » Mittwald Blog ...
www.mittwald.de » Blog » Arbeitsalltag
 20.08.2015 - Design Thinking ist ein iterativer Prozess, der schnell Ergebnisse liefern kann. Damit das Team und der Prototyp eine gewisse Qualität ...

Abb. 6: Die semantische Auszeichnung des Brotkrumenpfads wirkt sich auf die Darstellung in den SERPs aus

Das Attribut „hreflang“

Mit dem Attribut hreflang können Sie im Head-Bereich der Website oder auch in einer Sitemap.xml-Datei definieren, in welchen Sprachen und für welche Gebiete es Übersetzungen oder Anpassungen einer Seite gibt. Standardmäßig wird bei TYPO3 mit dem GET/POST-Parameter „L“ zwischen den Sprachen umgeschaltet. Sollte in der TypoScript-Konfiguration „config.sys_language_mode = content_fallback“ definiert sein, wird bei einer nicht übersetzten Webseite meist die Standard-Sprache angezeigt. Da somit eine Seite durch die angehängten „L“-Parameter unter mehreren URLs vorhanden wäre, würde es zu Duplicate Content kommen. Aus SEO-Sicht sollten daher alle Seiten übersetzt werden und nicht übersetzte Seiten sollten nicht aufrufbar sein, es sollte also kein Content_fallback-Modus aktiviert werden. Dazu definieren Sie „config.sys_language_mode = strict“.

Für jede Variante einer Seite müsste nun jeweils eine Zeile in der Form

```
<link rel="alternate" href="http://example.com/[Ordner]/[Seite]/" hreflang="[Sprache]-[Gebiet]" />
```

im Head jeder Seite eingefügt werden. Für eine Website mit den Zielgebieten Deutschland, Österreich und England müsste beispielsweise folgender Code auf die Impressum-Seite:

```
<link rel="alternate" href="http://example.de/de-de/impressum/" hreflang="de-DE" />
<link rel="alternate" href="http://example.de/de-at/impressum/" hreflang="de-AT" />
<link rel="alternate" href="http://example.de/en/imprint/" hreflang="en" />
```

Soll, wie in diesem Beispiel, auf der Seite zwischen den deutschsprachigen Ländern unterschieden werden, so sollte auch eine allgemeine Version für „Deutsch“ definiert werden. Die nächste Zeile wäre laut Google wünschenswert:

```
<link rel="alternate" href="http://example.de/de/impressum/" hreflang="de" />
```

Ist eine weitere Seite z. B. für das Kontaktformular für Österreich nicht angepasst, dann dürfte auf dieser Seite nur Folgendes stehen:

```
<link rel="alternate" href="http://example.de/de/kontakt/" hreflang="de" />
<link rel="alternate" href="http://example.de/en/contact/" hreflang="en" />
```

Gibt es eine Seite mit einer Sprachauswahl für alle nicht näher angegebenen Sprachen, dann sollte die Zeile auch noch ergänzt werden:

```
<link rel="alternate" href="http://example.de/" hreflang="x-default" />
```

Auf dieser Seite könnte dann eine Weltkarte dargestellt werden, mit der ein Besucher sich seine gewünschte Version der Seite aussuchen kann.

Aufpassen bei der Open-Graph-Eigenschaft og:locale! Hier wird ein Unterstrich verwendet:

```
<meta property="og:locale" content="de_DE" />
```

Dateien mit passenden hreflang-Headern

Damit Google auch bei Dateien wie PDF-Dokumenten weiß, für welche Sprach-/Gebietskombination diese verfasst wurden, bietet es sich an, für jedes Land einen Unterordner im Standard-Verzeichnis fileadmin anzulegen. Innerhalb des Verzeichnisses sollte dann ein Ordner für jede Sprache erstellt werden. Ein Beispiel für ein PDF könnte so aussehen:

```
» fileadmin/Pdf/de/datei.pdf
» fileadmin/Pdf/de/de/datei.pdf
» fileadmin/Pdf/de/at/datei.pdf
» fileadmin/Pdf/en/datei.pdf
```

Damit die richtigen HTTP-Header übermittelt werden, könnte der Download über eine selbst geschriebene Extension erfolgen. Diese müsste dann überprüfen, welche Sprach-Varianten der Datei vorhanden sind und dann die Header dynamisch ausliefern.

Eine Alternative zu der Extension-Lösung wäre, dass für jeweils eine Datei ein Ordner angelegt wird, in dem die verschiedenen Varianten abgelegt sind. In diesen Ordner muss eine .htaccess-Datei gelegt werden, in der die passenden HTTP-Header definiert werden, die wie in Abbildung 7 zu sehen aufgebaut sind.

Nachteil dieser Variante ist, dass der Redakteur später diese .htaccess-Dateien selber pflegen muss.

4. URL-Handling

Beim URL-Handling kommt es im Wesentlichen darauf an, dass ein Dokument über eine einzige, im Idealfall sprechende, URL zu erreichen ist. Dadurch wird unter anderem die Gefahr von Duplicate Content minimiert.

Sprechende Pfade mit der Extension RealURL

Da TYPO3 von Haus aus URLs mit Seiten-IDs generiert, muss zunächst eine sprechende URL erzeugt werden. Hierzu lässt sich die Extension RealURL verwenden. Die Extension baut URLs in der Form `http://example.com/index.php?id=1&L=1&news=123` in die suchmaschinenfreundliche Version `http://example.com/de/nachricht/titel-der-nachricht/` um. Es kann eine automatische Konfiguration verwendet werden, sinnvoller ist jedoch meist die manuelle Konfiguration in der Datei `typo3conf/realurl_conf.php`.

Relativ unbekannt ist, dass es innerhalb der Extension-Konfiguration eine Lösung für das Trailing-Slash-Problem gibt. Mit der Zeile „`appendMissingSlash => ,ifNotFile, redirect[301]`“, im `init`-Bereich wird automatisch eine permanente Umleitung mit einem 301-Header gemacht.

Abgesehen davon sollte RealURL nicht für Weiterleitungen verwendet werden. TYPO3 müsste erst langsam den Redirect generieren. Dies könnte bedeutend schneller in der Webserver-Konfiguration oder, falls vorhanden, in einem vorgeschalteten Reverse Proxy (z. B. Varnish, mehr dazu unter `http://einfach.st/mwseo2`) gemacht werden. So sollten Umleitungen von `http://` zu `https://` bzw. `www.example.com` zu `example.com` ebenfalls in dem Webserver oder Reverse-Proxy erfolgen.

Duplicate Content vermeiden

Sobald im TYPO3 die Extension RealURL aktiviert ist, kann eine Seite auf mehreren Wegen aufgerufen werden. Hätte eine Seite mit einer Produktliste die interne ID 321, dann würde der Aufruf je nach eingerichteten Rewrite-Regeln über `index.php?id=321`, `produkte`, `produkte/`, `produkte.html` funktionieren. Um hier die Entstehung von Duplicate Content zu verhindern, sollte die `robots.txt` angepasst und/oder ein Canonical-Link gesetzt werden. In der `robots.txt` sollten zumindest diese Zeilen vorhanden sein:

```
Disallow: /*?id=* #Seiten,
die nicht über sprechende Pfade
aufgerufen werden
Disallow: /*&L=0* #Entspricht
der Standard-Sprache
```

Oft findet man im Internet Beispiele für eine auf TYPO3-optimierte `robots.txt`. Diese sollten Sie genau prüfen, denn es dürfen dabei keine CSS- oder JavaScript-Dateien geblockt werden! Vor allem beim Ausschließen der Dateien im Verzeichnis `typo3temp/`

```
Link: <http://es.example.com/fileadmin/Pdf/datei_es.pdf>; rel "alternate"; hreflang "
```

Abb. 7: .htaccess - Anpassung des http-Headers

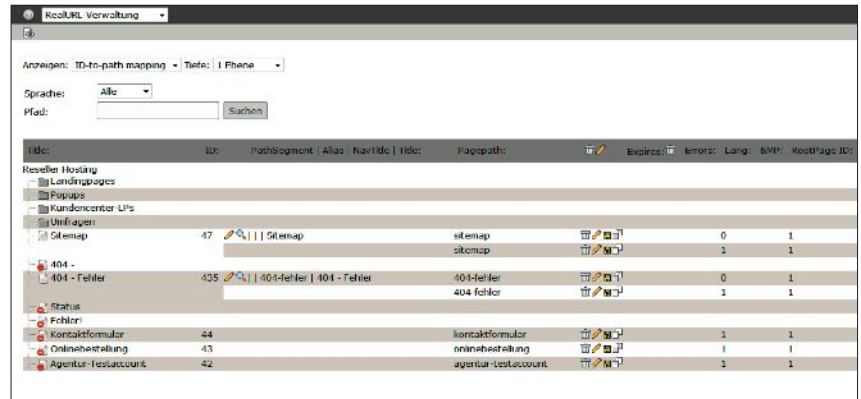


Abb. 8: RealURL wandelt kryptische URLs in sprechende und suchmaschinenfreundliche URLs um



Abb. 9: Mit einer Erweiterung des Backends kann eine individuelle Canonical-URL pro Seite definiert werden

sind dynamisch generierte CSS- & JS-Dateien für Google nicht mehr auswertbar.

Sicherheitshalber sollten Sie alles mit dem Robots-Testing-Tool auf `http://einfach.st/grtt` überprüfen.

Eine SEO-optimierte `robots.txt`-Datei für TYPO3 finden Sie unter `http://einfach.st/mwrob`.

Druckversion über CSS definieren

Druckversionen einer Seite werden üblicherweise durch die Erzeugung einer weiteren Seitendefinition in TypoScript erzeugt. Aufgerufen wird die druckoptimierte Seite dann über einen zusätzlichen URL-Parameter (meist „`&type=98`“). Da der gleiche Seiteninhalt nun über verschiedene URLs zu erreichen ist, kommt es zu Duplicate Content. Um das zu umgehen, sollte in der `robots.txt` z. B. folgende Definition erfolgen `Disallow: /*&type=98*`.

Um dieses Problem komplett zu umgehen, sollten Sie besser mit Media-Querys in der CSS-Datei arbeiten.

Canonical-Link

Sofern eine Seite über mehrere URLs zu erreichen ist, muss Google wissen, welche URL indiziert werden soll und welche nicht. Über die `robots.txt` können schon einige mehrfache Indizierungen abgefangen werden. Oft ist es aber sicherer, die korrekte URL per Canonical-Link zu hinterlegen.

Falls es auf einer Seite noch unterschiedliche Sortierungen (z. B. nach Preis, Bewertung ...) mit einer Listendarstellung gibt, müsste der Seite eine manuelle Konfigurationsmöglichkeit für die Canonical-URL hinzugefügt werden. Ein Eingabefeld gibt es dafür standardmäßig nicht, aber es ist durch wenige Zeilen in einer Extension hinzufügbare. Ein Beispiel finden Sie in einer Demo-SEO-Extension unter: `http://einfach.st/mwseo3`.

Eine sehr oft eingesetzte Extension ist die News Extension. Auf verschiedenen Seiten können unterschiedliche Versionen der Liste ausgegeben werden. Hier sollte man sich entscheiden,

welche Listenansicht indiziert werden sollte (Top-News, Standard-Liste mit Sortierreihenfolge, Erstellungsdatum, Archiv ...). Auf den Seiten, die nicht indiziert werden sollen, müsste vom Redakteur wie schon bei der Canonical-URL eine Eingabemöglichkeit hinzugefügt werden, mit dem die Meta-Angabe `<meta name="robots" content="no-index">` konfiguriert werden kann. Den nötigen Programm-Code finden Sie in der schon genannten Extension „ot_seo“.

Startseite optimieren

In vielen Dokumentationen steht beschrieben, dass als erste Seite eine „Root“-Seite für die Vererbung des TypoScript-Templates angelegt werden soll. Dem untergeordnet solle die echte Startseite mit einem gegebenenfalls angepassten TypoScript-Template, welches dann nur für die Startseite gültig ist, angelegt werden. Dies führt aber dazu, dass beim Aufruf von beispielsweise <http://example.com> direkt eine Umleitung auf <http://example.com/startseite/> erfolgt. In alten Versionen wurde beim Aufruf der Root-Seite einfach der Inhalt der Startseite gezeigt, was natürlich zu Duplicate Content führte. Optimal ist heute, dass die Root-Seite entfernt und stattdessen direkt die Startseite angezeigt wird. Soll dann die Startseite eine abweichende Ausgabe haben, helfen „Backend Layouts“, „Conditions“ und If-Abfragen im Fluidtemplate weiter.

Performance-Optimierungen

Eine Website sollte immer so schnell wie möglich ausgeliefert werden. Bei Seiten, die mehr oder weniger statische Inhalte haben, kann ein Reverse Proxy wahre Wunder bewirken. TYPO3 muss in diesem Fall keine Rechenzeit darauf verwenden, um eine gecachte Seite auszuliefern. Müssen dem Besucher der Webseite allerdings dynamische Inhalte ausgeliefert werden, muss dafür gesorgt werden, dass die Webseiten möglichst

schnell generiert werden. Helfen kann hier eine aktuelle PHP-FPM-Version. Es sollte darauf geachtet werden, dass ein Byte-Code-Cache wie OpCache oder APC mit mindestens 100 MB Shared Memory sowie ein Key-Value-Speicher wie der User Cache „APCu“ verwendet werden. Als Festplatten sollten SSDs eingesetzt werden, das gilt vor allem für die MySQL-Datenbank.

Auch in TYPO3 können durch geschickten Einsatz der TypoScript-Funktion „cache“ (mehr dazu unter <http://einfach.st/typo3d>) und der in TYPO3 7.4 hinzugekommenen TypoScript-Eigenschaft „cache“ Render-Zeiten eingespart werden. Die neu hinzugefügte Eigenschaft funktioniert auch mit einem „Content Object Array“ (COA). Ein COA ist eine Gruppierung von mehreren Content-Elementen wie z. B. TEXT oder IMAGE. Mit einem solchen COA lässt sich z. B. eine umfangreiche Mega-Navigation erstellen. Der Zeitaufwand für die Generierung kann dabei schon mal sehr hoch sein. Wenn die Seite aber durch den richtigen Einsatz des Cache-Befehls nur einmal pro Sprache generiert werden müsste, könnte nach dem Löschen der Seitencaches der Cache durch ein Aufrufen einer einzelnen Seite pro Sprache „vorgewärmt“ werden. Jeder Aufruf einer Seite würde somit teilweise um Sekunden beschleunigt.

Personenbezogene Daten sollten, wenn möglich, immer über AJAX nachgeladen werden. TYPO3 kann dann Seiten entweder komplett oder zumindest teilweise cachen. Dies vereinfacht auch später den Einsatz eines Reverse-Proxy-Servers.

Die mit TYPO3 gelieferte Extension „Indexsuche“ sollte nur bei sehr kleinen Webseiten eingesetzt werden. Schon bei mittelgroßen Webseiten wird das gesamte System äußerst langsam. Besser ist hier der Einsatz von Apache Solr (<http://einfach.st/mwsolr>), Elasticsearch (<http://einfach.st/mwes>) oder der benutzerdefinierten Google-Suche.

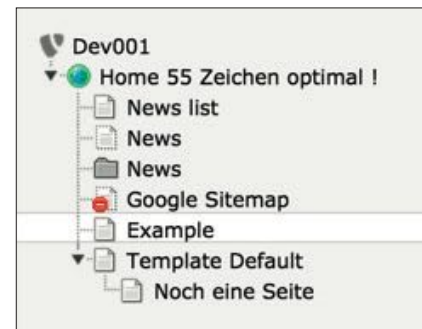


Abb. 10 Seitenbaum in TYPO3

Sitemap für Google erzeugen

Es gibt einige Extensions, die eine Sitemap für Google generieren. Eine performante Extension, die auch schon bis zur TYPO3 7.3 funktionierte, hat den Extension-Key „ddgooglesitemap“. Diese eignet sich zumindest für „normale“ Webseiten. Wenn Sie aber zum Beispiel mithilfe einer Extension einen Produktkatalog abbilden, würde die Sitemap nur die erste Seite der Listenansicht zeigen. Die Detailansicht der Produkte würde komplett fehlen, da diese erst zur Laufzeit durch die Extension generiert werden und damit von den meisten Sitemap-Extensions nicht „gesehen“ werden. Zum Zeitpunkt der Artikelstellung erzeugt „ddgooglesitemap“ in TYPO3 7.4 leider noch eine Fehlermeldung, die hoffentlich bis zur finalen TYPO3-Version 7 LTS nicht mehr auftritt.

Eine weitere Extension, die aber zum jetzigen Zeitpunkt nur mit TYPO3 6 lauffähig ist, hat den Extension-Key „metaseo“. Diese hat sehr viele Konfigurationsmöglichkeiten und kann auch eine Sitemap für Google generieren – dem Manual nach funktioniert diese aber nur mit gecachten Seiten. Diese Extension bietet zumindest für das Selbststudium einige Ansätze, was an Optimierungen eingebaut werden kann.

Als guter Ansatz wäre noch die Extension „google_services“ zu nennen. (<http://einfach.st/github2>). Eine kurze (ältere) Anleitung findet man bei TYPO3 Blogger: <http://einfach.st/t3bl2>.

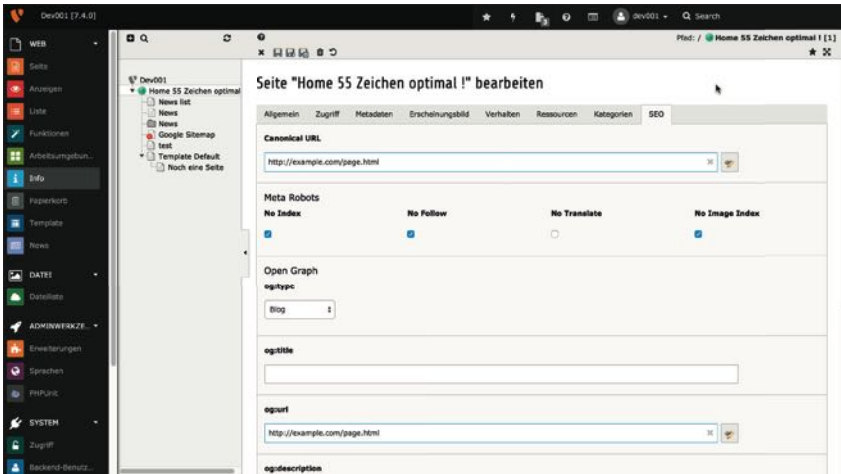


Abb. 11: Das Backend des neuen TYPO3 7 kommt aufgeräumter und farbenfroher daher; unter der Haube sind viele SEO-Anpassungen möglich

Der Vorteil dieser Extension ist, dass relativ leicht eigene „SitemapProvider“ gebaut oder angepasst werden können, die dann nicht nur mit Seiten, sondern auch mit News und anderen Extensions zusammenarbeiten.

Die perfekte fertige Sitemap-Extension gibt es also nicht. Keine der genannten Erweiterungen kennt das

hreflang-Attribut. Spätestens jetzt wird klar, dass Erweiterungen „von der Stange“ nicht immer weiterhelfen. Eine perfekte Sitemap, die Ihre Webseiten oder Produkt-Listen womöglich für verschiedene Länder, Sprachen und Währungen mit passendem hreflang-Attribut anzeigen, kann im Endeffekt nur selbst geschrieben werden.

Fazit: SEO in TYPO3 7

Wie gezeigt, ist TYPO3 „out of the box“ nicht sofort suchmaschinenfreundlich. Mit den im Artikel vorgestellten Beispielen sieht man aber wohl recht deutlich, wie flexibel TYPO3 ist. Es sind zwar einige Eingriffe nötig, bis TYPO3 suchmaschinenfreundlich arbeitet, aber dafür lässt sich eine Lösung für so ziemlich jede OnPage-SEO-Problematik finden. In den nächsten TYPO3 7 Versionen halten mit den Structured Content Containers oder der Open-Graph-Unterstützung zudem einige Neuerungen Einzug, die das Leben der TYPO3-Entwickler in puncto SEO noch ein Stück einfacher machen.¶

Einzelheftbestellung* unter:

www.websiteboosting.com/einzelheftbestellung/
 *so lange der Vorrat reicht...



Oder gleich im rückwirkenden Abo unter:
www.websiteboosting.com/abo