

Suchmaschinenoptimierung (SEO)

Suchmaschinenfreundliche URLs

Einführung / Vorwort

In diesem kurzen Artikel möchte ich auf drei einfach umzusetzende Möglichkeiten eingehen, die einem Webmaster die Verwendung von Suchmaschinenfreundlichen URLs ermöglichen, bzw. näher bringen. Wir gehen davon aus, dass Dateien oder Ordner in der Form `ergebnis/bundestagswahl/2005.html` oder `ergebnis;bundestagswahl;2005` an eine zentrale Datei weitergeleitet werden sollen, die die angefragte URL (`REQUEST_URI`) analysiert und anschließend die entsprechenden Inhalte ausgibt. Auf die Auswertung der `REQUEST_URI` und Sicherheitsaspekte wird hier nur am Rande eingegangen.

Übersicht

Es gibt mindestens drei Arten der Nutzung bzw. Erstellung von SEO-URLs, unter Verwendung von

1. `mod_rewrite`
2. **ErrorDocument und Header-Manipulation**
3. „Erweiterter Index-Datei“ und `REQUEST_URI`

1. MOD_REWRITE

`Mod_Rewrite` ist ein Modul des Apache-Webservers, welches für die Umschreibung von URLs vorgesehen ist (engl. `rewrite` = umschreiben).

Wichtig zu beachten ist, dass die Voraussetzung zur Verwendung des Moduls ist, dass es installiert ist und man es benutzen darf. Was wohl, insbesondere bei Hostingpaketen, nicht immer der Fall sein wird. Daher eignet sich diese, aus professioneller Sicht, optimale Art der URL-Manipulation nicht für jeden Webmaster.

Sofern einem `mod_rewrite` zur Verfügung steht, gibt es neben einigen Basics unendliche Möglichkeiten der Umleitungen usw., die den Umfang dieses Artikels sprengen würden.

Befehle des `mod_rewrite` werden in einer `.htaccess`-Datei abgelegt und beziehen sich dann auf alle Ordner unter ihr, inklusive dem, indem sie selbst liegt.

Eine einfache Umleitung könnte folgendermaßen aussehen:

```
.htaccess (File)  
RewriteEngine On  
RewriteRule !index\.php - [C]  
RewriteRule .* /index.php
```

Damit würden sämtliche Anfragen auf die Datei `index.php` (im selben Ordner wie die `.htaccess`) umgeleitet werden. Ein guter Anfang, doch womöglich soll auf gewisse Dateien, z.B. Bilder, PDFs direkt zugreifen werden. Hierfür wäre also denkbar, gewisse Ordner von einer Weiterleitung auszuschließen, was mit folgendem Code möglich wäre:

```
.htaccess (File)  
RewriteEngine On  
RewriteRule ^(images|upload) - [L]  
RewriteRule !index\.php - [C]  
RewriteRule .* /index.php
```

Hier würden alle Anfragen die mit `images` oder `upload` beginnen von der Umleitung ausgeschlossen.

Es ist zu beachten, dass die Anfrage der Datei `upload_formular_in_php.html` nicht umgeleitet wird, denn sie beginnt ebenfalls mit `upload`.

Weitere Infos und eine aktive Community: <http://www.modrewrite.de>

2. ErrorDocument und Header-Manipulation

Falls einem das `mod_rewrite`-Modul nicht zur Verfügung steht, bietet diese Methode eine vergleichbare Alternative. Fehler-Dokumente werden vom Webserver in allerlei Fehlerfällen ausgegeben. Sie sind in verschiedene Nummernkreise unterteilt, von denen der 400er für die Rubrik „nicht auffindbar“ zuständig ist.

Auch hier gilt, dass unter Umständen nicht jeder Hoster die nötigen Funktionen zur Verfügung stellt. Es werden 1. selbst definierbare Fehlerseiten und 2. Header-Manipulationen vorausgesetzt. (Strato ermöglicht z.B. keine Header-Manipulation!)

Ein Fehler-Dokument wird in einer `.htaccess`-Datei definiert, die folgendermaßen aussehen kann:

```
.htaccess (File)  
ErrorDocument 404 /index.php
```

Der HTTP-Fehler Nr. 404 bedeutet „angefragtes Dokument nicht gefunden“, als Alternative dafür wird hier eine Datei angegeben, die den Benutzer über den Fehler informiert, oder in unserem Falle den angefragten Inhalt ausgibt. (Für andere Fehler wäre ebenfalls eine Definition denkbar für unsere Belange jedoch nicht nötig.)

Wesentlicher Vorteil bei Fehlerdokumenten ist, dass sich alle Dateien die tatsächlich existieren nach wie vor ansprechen lassen.

Allerdings wird standardmäßig im Dokumenten-Header übermittelt, dass das eigentlich angefragte Dokument nicht gefunden wurde (`HTTP/1.1 404 not found`). Was wiederum zur Folge hat, dass eine Suchmaschine diese Seite nicht indizieren wird, denn „sie existiert ja nicht“.

Hier muss man also noch eine Header-Manipulation vornehmen – in einem PHP-Script wäre dafür (vor jeglicher anderen Ausgabe) folgender Code nötig:

```
index.php (File)  
// ..  
header("HTTP/1.1 200 OK");  
// ..
```

Damit gaukeln wir der Suchmaschine vor, dass die Seite vorhanden und alles Okay ist. (Über die Definition einer tatsächlichen Fehlerseite sollte jedoch auch nachgedacht werden.)

3. Erweiterte Index-Datei

Ob es für diese Methode einen tatsächlichen Namen gibt, sei in den Raum gestellt. Wie genial sie ist ebenso. Fakt ist, sie wird von Suchmaschinen akzeptiert, funktioniert im Browser und sollte wohl auch bei jedem Hostingpaket laufen.

Jede URL wird einfach in der Form

`index.php/upload_formular_in_php` oder
`main.php/upload/formular/in/php`

angegeben. Die als PHP-Datei angegebene Datei bekommt die Anfrage zugeteilt und nun ist es am Programmierer (wie übrigens auch bei Methode 1 und 2) die `REQUEST_URI` auszuwerten und den Inhalt zum angefragten Thema „PHP-Upload-Formular“ auszugeben.

Schlusswort / Fazit

Was haben wir gelernt? – Grundsätzlich gilt: Suchmaschinenfreundliche URLs sind leicht gemacht. Was aber hier bisher nicht erwähnt wurde: Es gibt z.B. zu beachten, dass es bei derart optimierten Seiten sinnvoll ist, Bilder oder andere Links relativ zum `root` anzugeben, also mit vorangestelltem Slash und dem kompletten Zielpfad (``).

Außerdem sollte bei der Verarbeitung der `REQUEST_URI` bedacht werden, dass hier eine potentielle Angriffstelle zum einschleusen von böswilligem Code liegt.

Viel Erfolg.

Hinweis: Sämtliche Informationen ohne Garantie und Anspruch auf Vollständigkeit, jedoch nach bestem Wissen des Autors zum Zeitpunkt der Erstellung. Verwendung von Teilen nur nach Genehmigung, komplette Weitergabe des unveränderten Dokuments erlaubt.