

Von Zeit zu Zeit entwickelt plan08 Tutorials zum Thema Neue Medien. Unsere Dokumente versuchen mit Hilfe von Beispielen Schritt für Schritt zu erklären, wie man bestimmte Probleme löst. Dabei spezialisieren wir uns auf die Themengebiete "Softwareentwicklung für das Internet", "Grafikbearbeitung" und "Internetmarketing". Unsere Anleitungen sollen Anfängern ermöglichen, ähnlich wie in einer Schulung an Erfahrung zu gewinnen und dabei ohne Lehrkraft schnell gute Ergebnisse zu erzielen.

Projekt
plan08 Bibliothek

Aufgabe
**Suchmaschinen
Optimierung durch
mod_rewrite**

Datum
22. Juni 2006

Weitere Tutorials finde Sie unter:

<http://www.plan08.de/projekt08/plan08bib>

Suchmaschinen Optimierung durch mod_rewrite

Dieses Tutorial soll einen kleinen Einblick geben, wie man den Apache Webserver dazu bringen kann, dynamische Internetseiten als statische anzuzeigen. Nach dem Lesen dieser Anleitung sollten Sie in der Lage sein eine URL wie

<http://www.meinedomain.de/index.php?ref=13>

in

<http://www.meinedomain.de/referenzen/13.mercedes-benz.htm>

umzuschreiben, ohne eine Zeile PHP-Code zu produzieren. Dazu werden wir einen kleinen Ausflug zu den regulären Ausdrücken machen um uns danach das Apache Modul mod_rewrite genauer anzusehen.

Als erstes wollen wir auf die Idee eingehen, die hinter dieser Art von Suchmaschinen Optimierung steckt. Suchmaschinen versuchen Seiten auf eine Art und Weise zu bewerten, wie der Mensch sie auch bewerten würde. Nur so kann man eine Möglichst hohe Zufriedenheit des „Kunden“ gewährleisten. Welche der zwei oben erwähnten URLs würden Sie denn eher besuchen, wenn Sie auf der Suche nach Referenzen wären? Ganz genau, die Zweite.

Intuitiv bewertet der Mensch eine Adresse, die er lesen oder interpretieren kann für hochwertiger. Suchmaschinen gehen da sehr ähnlich ran. Sie bewerten die erste URL schlechter, da man nicht feststellen kann, ob was der ref=13 Parameter bedeutet. Ändert sich etwas an der Seite, wenn man diesen Parameter ändert? Oder heißt ref=13 einfach, dass man von der Webseite mit dem Index 13 hierher weitergeleitet wurde? Suchmaschinen wissen es nicht und ich denke die meisten Menschen, die ein Content Management System (CMS) benutzen wissen es auch nicht.

Die logische Schlussfolgerung wäre, dass Suchmaschinen „klare“ URLs also höher bewerten, dies ist auch so. Und wenn Sie schon das Hauptkeyword in Ihrem Dateinamen haben, so bekommen Sie bei Google sogar noch einen Bonus. Probieren Sie es einfach aus. Seiten die das Suchwort bereits in der URL versteckt haben, finden Sie immer ganz oben in Suchmaschinen.

Das Umschreiben von URLs hat aber noch weitere Vorteile. Einer davon wäre das Verstecken der eigentlichen Struktur Ihres Systems. Nehmen wir wieder an, Sie verwenden ein CMS, wie Joomla (www.joomla.de) Jeder, der ein wenig Ahnung von Webentwicklung hat, erkennt an der URL, dass die Seiten zu einem CMS gehören. Die Angriffsmöglichkeit steigt somit enorm an. Betrachten Sie nur die ganzen Foren Attacks in der letzten Zeit.

Achten Sie aber bei der URL Optimierung auch auf natürliche Verzeichnis Strukturen. Niemand würde 1000 Dateien in einem Verzeichnis lagern. Führen Sie also Unterverzeichnisse ein, wie wir dies in unserem Beispiel auch getan haben. Sie dürfen nie vergessen:

„Suchmaschinen sind intelligent und werden immer intelligenter“

Tricks die vielleicht vor ein paar Jahren noch funktioniert haben, wie z.B. weiße Schrift auf weißem Hintergrund, würde heute keine Suchmaschinen mehr austricksen.

Das Werkzeug, das wir zum Optimieren verwenden werden heißt mod_rewrite und ist ein Modul für den Apache Webserver.

„mod_rewrite is voodoo, damn cool but still voodoo“

Seien Sie also vorsichtig, es sollte kein Problem sein, durch einen einfachen Fehler in Ihren Ausdrücken, den ganzen Server abzuschießen. Aus diesem Grund wurde von vielen Webhostern dieses Feature entfernt. Den Voodoo-Spruch habe ich natürlich nicht ausgesucht, um Ihnen Angst zu machen, jede Änderung oder Umleitung kann in sekundenschnelle wieder rückgängig gemacht werden.

Wir gehen davon aus, dass Sie ein funktionierendes mod_rewrite auf Ihrem System installiert haben, ansonsten gibt es im Internet eine Menge von Tutorials, die Ihnen die Installation vereinfachen.

Um das Modul nutzen zu können, benötigen Sie die Datei .htaccess im Hauptverzeichnis Ihrer Webseite. Der Punkt am Anfang des Dateinamens bedeutet, dass es sich um eine versteckte Datei in einem Unix System handelt. Falls Sie dieses File in Ihrem Editor nicht sehen können, Dreamweaver blendet sie z.B. aus, so öffnen Sie sie einfach per SSH oder lokal und laden sie danach per FTP auf den Webserver. Sie haben nun eine leere .htaccess auf Ihrem System. Tragen Sie die folgenden Zeilen ein:

```
RewriteEngine On

#Regeln
RewriteRule ^index\.htm$ /index.php
```

Mit ein bisschen „informatischem“ Verständnis kann man diese Zeilen schon lesen. Sie bedeuten, dass alle Anfragen, die auf die Seite index.htm zeigen, auf index.php weitergeleitet werden. Erstellen Sie nun eine index.php Datei und versuchen Sie sie über die nicht existierende index.htm über Ihren Browser aufzurufen. Wenn alles geklappt hat, so sollten Sie nun den Inhalt der vom PHP Interpreter verarbeitete index.php sehen. Falls dies nicht geklappt hat, so ist dies ein Indiz dafür, dass das Modul mod_rewrite nicht installiert ist. In diesem Fall sollten Sie Ihrem Webhoster kontaktieren.

Gehen wir nur genauer auf die einzelnen Zeilen Code ein:

```
RewriteEngine On
```

Hiermit sagen Sie dem Apache Webserver, dass wir das Modul verwenden wollen. Sie schalten es sozusagen ein.

```
#Regeln
```

Zeilen, bei denen ein # vorausgeht haben nur die Funktion eines Kommentars. Sie werden vom Server nicht beachtet, Sie sollten sie aber trotzdem verwenden um nach einiger Zeit noch zu wissen, was sie dort eigentlich gemacht haben.

```
RewriteRule ^index\.htm$ /index.php
```

Dies ist die eigentliche Regel, beginnend mit RewriteRule um den Anfang zu kennzeichnen. Als nächstes folgt die Adresse, die umgeschrieben werden soll, eingegrenzt durch ^ und \$, danach das Ziel, als absoluter Pfad, deshalb ist das / am Anfang auch so wichtig.

Dies ist natürlich eine der einfachen Arten URLs umzuschreiben. Falls man aber hunderte solcher Fälle hat, wird das ganze auf diese Weise sehr umständlich. Sie bräuchten hundert Einträge in Ihrer .htaccess um alle Umleitungen abzufangen.

Und genau hier kommen die Regulären Ausdrücke mit ins Spiel, auf die wir nun näher eingehen werden. Diese Ausdrücke werden dazu verwendet eine ganze Klasse von Suchwörtern abzudecken. Ein einfaches Beispiel für reguläre Ausdrücke wird Ihnen bei der Datenbankabfrage Sprache SQL untergekommen sein. Die dort verwendeten Ausdrücke mit dem % Operator sind nichts anderes, als Reguläre Ausdrücke. Er ist eine Art Platzhalter, so wie früher zu alten DOS Zeiten oder unter Linux ein dir *.txt oder ls *.txt.

Gebe mir alle Dateien zurück, die mit .txt enden. Dies würden die beiden dir und ls Befehle ausführen. Und genau einen solchen Ausdruck brauchen wir ja: Nimm alle .htm Dateien und leite Sie auf index.php um. Versuchen wir uns also an unserer .htaccess Datei.

```
RewriteRule ^(.*)\.htm$ /index.php
```

Der (-*) Operator ist das mod_rewrite Pendant zu dem Stern auf der Dateiebene und dem % in SQL Ausdrücken. Nachdem Sie diese Zeilen Code eingebaut haben, versuchen Sie mal eine kontakt.htm auszuführen. Sie werden merken, dass Sie nun in Ihrer index.php landen.

Natürlich sind wir noch lange nicht an den Grenzen des mod_rewrite. Aber wir nähern uns so langsam an unser Ziel an. Der Große Vorteil des (-*) Operators ist, dass die Werte, die dort „abgefangen“ werden Ihnen jetzt auch zur Verfügung stehen. Sie können Sie also in Ihre Ziel URL einbauen.

```
RewriteRule ^/(.*)\.htm$ /$1.php
```

Unser erstes Teilziel ist nun erreicht, eine kontakt.htm wird auf kontakt.php umgeleitet, genauso wie eine impressum.htm auf impressum.php weitergeleitet wird.

Mit dem Wissen, dass Sie jetzt erlang haben, können Sie schon eine ganze Menge machen, wie die folgend aufgeführten Beispiele Ihnen demonstrieren werden.

```
RewriteRule ^(.*)\.htm$ /index.php?action=$1
```

CMS verstecken

```
RewriteRule ^/(.*)/index\.htm$ /$1.php  
RewriteRule^ /user/(.*)\.htm$ /showUser.php?user=$1
```

Über den ersten Ausdruck können Sie dem Surfer und den Suchmaschinen eine vorhandene Verzeichnisstruktur vorgaukeln.

Den zweiten würde ich verwenden um jedem Mitglied Ihrer Community eine eigene „schöne“ URL zu geben.

Um letztendlich die URL wie Sie in unserem ersten Beispiel erwähnt ist umzuschreiben, fehlt uns aber noch ein kleines Bruchstück. In Regulären Ausdrücken ist es möglich mehrere Platzhalter mit (-*) zu definieren. Daher kommt auch die \$1, die Sie verwenden um die ausgeschnittenen Werte wieder einzubauen. \$1 wäre der erste Platzhalter, \$2 der Zweite usw. Dabei zählen wir von Links.

```
RewriteRule ^referenzen/(.*)\.(.*)\.htm$ /index.php?ref=$1
```

Dieser Ausdruck würde jetzt genau dass machen, was wir vorhatten.

Natürlich gehört der ganze Ausdruck in eine Zeile. Er schreibt die URL

<http://www.meinedomain.de/index.php?ref=13>

in

<http://www.meinedomain.de/referenzen/13.mercedes-benz.htm>

um, genau so, wie wir es gewollt haben. Dabei ist es uns ganz egal, was hinter der 13 steht. Würde statt „mercedes-benz“ dort „homepage-mercedes-benz“ stehen, wäre der Effekt der gleiche.

Als letzten Tipp, den ich Ihnen mit auf den Weg geben möchte, wäre noch anzumerken, dass Sie versuchen sollten Umlaute und Sonderzeichen aus Ihren URLs fernzuhalten, da sie dadurch keine Probleme mit anderen Zeichenkodierungen bekommen.

Falls Sie Fragen oder Anregungen haben, Sie melden Sie sich unter

<http://www.plan08.de/projekt08/plan08bib>

und kommentieren Sie das Tutorial. Natürlich können Sie auch unter

<http://www.plan08.de/index.php?action=projekt08&command=showplan08bib>

Ihre Anregungen äußern. Wie diese Umschreibung funktioniert, sollten Sie ja jetzt wissen.