



*MOD-REWRITING*

*EASY-LINE*

## Der Schnelleinstieg

Die jeweils aktuelle Version finden Sie auf der Homepage  
[www.honigs-home.de](http://www.honigs-home.de)

Stand: 11.08.2004

© 2001–2004  
by Holger Niggemann

## Was ist Mod-Rewriting?

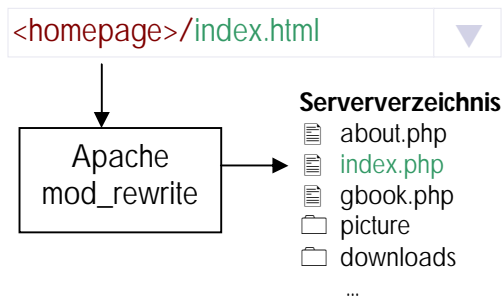
Technisch gesehen handelt es sich hierbei um ein im Apache-Server implementiertes Modul. Mit dieser „Erweiterung“ können URL's auf einfache Weise modifiziert und Serverseitig weiterverarbeitet werden. Die Art und Weise in der diese Modifizierung stattfindet wird durch Reguläre Ausdrücke bestimmt.

## Wozu das Ganze?

### Skriptsprachen verbergen

Ein Anwendungsfall ist das Verbergen von Skriptsprachen. Mit mod\_rewrite lassen sich alle Dateiendungen von z.B. php auf html setzen. Hierdurch entsteht beim User und für die Suchmaschinen der Eindruck, es handle sich um statische Seiten. Dieses ist was Suchmaschinen-Ranking angeht nicht ganz unerheblich. Bei einigen Suchmaschinen stehen statische Seiten deutlich höher im Kurs.

### Hierzu ein einfaches Beispiel



Durch Eingabe der URL **index.html** im Browser wird die Anfrage serverseitig **auf index.php** umgeleitet. Die **index.html** existiert dabei nicht auf dem Server. Durch Eingabe von **index.php** ist diese Seite natürlich ebenfalls noch erreichbar. Ausnahme der Programmierer verhindert dieses durch ein Skript (Umleitung bei bestimmten Endungen...)..

## Parameterübergabe

Einige Skripte und Seiten übergeben Daten mittels der **GET-Methode** (Variablen werden durch ? und & getrennt einfach an die URL gehängt). Hierdurch werden URL's schnell lang und unübersichtlich. Lassen Sie stattdessen doch einfach die zu übergebenen Werte ein Teil des „virtuellen“ Dateinamens werden.

### Beispiel

wird schnell

do=view steckt nun in view

seite=1 wird so zu 1

< > = Homepage, Domain

### kurze URL's

Lange URL's lassen sich verkürzen und sind somit leichter zu merken. Ein weiterer Nebeneffekt ist die bessere Optik. Ihre Benutzer werden dieses zu schätzen wissen, zumindest dann, wenn diese die URL per Hand eingeben müssen.

## Funktioniert das auch bei mir?

Diese lässt sich am einfachsten mittels eines kurzen PHP-Skript klären.

### Gehen Sie folgendermaßen vor:

1. Starten Sie einen Text-Editor Ihrer Wahl
2. Tippen Sie folgende Zeilen ein

```
<?php
    phpinfo();
?>
```

3. Speichern Sie diese Datei unter dem Namen **info.php** ab. Eventuell die Endung auf .php3 oder .php4 ändern, dieses ist abhängig von Ihrem Provider. Die Endung .php sollte in den meisten Fällen jedoch funktionieren

4. Spielen Sie diese Datei per FTP auf Ihre Homepage hoch.
5. Rufen Sie diese dann über `<homepage>/info.php` in Ihrem Browser auf. Wenn alles funktioniert hat, sollten Sie eine Übersicht mit vielen Informationen sehen. Sollten Sie jedoch den unter Punkt 2 eingetippten Quellcode sehen, so wurde das PHP-Skript nicht verarbeitet. Versuchen Sie es dann mit der Dateiendung `.php3` bzw. `.php4`. Sollte auch dieses nicht funktionieren überprüfen Sie ob PHP-Skripte überhaupt ausgeführt werden können.
6. Wenn alles problemlos funktioniert hat, suchen Sie in der Übersicht nun nach **mod\_rewrite**. Meist ist der Eintrag unter **Loaded Moduls** zu finden. Sollten Sie `mod_rewrite` nicht finden, so wird das Modul wahrscheinlich nicht geladen und steht somit nicht zur Verfügung. Abhilfe kann dann nur noch Ihr Provider schaffen...

## Wie verwende ich Mod - Rewriting?

Das Zauberwort lautet **.htaccess**. In dieser Textdatei werden die Regeln für `mod_rewrite` festgelegt. Den meisten dürfte dieser Dateiname im Zusammenhang mit Passwortschutz ein Begriff sein. Diese Datei wird dann mittels FTP einfach auf dem Webserver hochgeladen.

### Hier ein Beispiel für eine .htaccess: :

```

1: #-----
2: # SAMPLE MOD-REWRITING
3: #-----
4:
5: RewriteEngine On
6: Option +FollowSymLinks
7:
8: RewriteRule ^*(.*) (\.html?)
9:

```

Zeile 1-3 sind reine Kommentare.

Mit **RewriteEngine On** wird `mod_rewrite` aktiviert. Damit das Ganze funktioniert müssen zusätzlich noch **symbolische Links** aktiviert werden. Dieses geschieht in der Zeile 6. Je nach Rechtevergabe des Providers ist dieses **nicht** immer in der `.htaccess`-Datei gestattet. Sollte `mod_rewrite` nicht funktionieren und Sie sonst kein Fehler finden, fragen Sie einfach mal bei Ihrem Provider nach. Vielleicht aktiviert dieser ja die Optionen in der Config-Datei seines Servers.

Mit `RewriteRule <Ausdruck1> <Ausdruck2>` werden die Regeln für das Rewriting festgelegt.

`<Ausdruck 1>` für die Originaldatei  
`<Ausdruck 2>` für die virtuelle Datei

## Reguläre Ausdrücke

Hier eine Übersicht der wichtigsten Parameter. Mit diesen sollten sich die meisten Aufgabenstellungen erfolgreich lösen lassen.

\	Das darauf folgende Zeichen besitzt keine Sonderzeichenbedeutung, selbst wenn es ein Sonderzeichen ist.
^	Markiert den Zeilenanfang
\$	Markiert das Zeilenende
	Musteralternative
()	Gruppieren, Subpatterns definieren
*	Der links davon stehende Ausdruck darf beliebig oft oder auch gar nicht vorkommen
+	Der links davon stehende Ausdruck darf beliebig oft muss aber mindestens einmal vorkommen
?	Das vorherige Zeichen darf genau einmal vorkommen
[ ]	Definiert eine Menge von Zeichen. Beginnt die Menge mit ^ am Anfang, so enthält die Menge alle Zeichen außer den angegebenen.
\d	Steht für ein Zahl zwischen 0-9
\w	alle alphanumerischen Zeichen inklusive dem Unterstrich

## Beispiele

Zum Abschluss noch ein paar konkrete Anwendungsfälle

### *Dateiendung php auf html ändern*

```
RewriteRule ^(.*)\.html$ $1.php
```

`^(.*)` Am Anfang  
`^(.*)` Beliebig viele Zeichen Parameter \$1  
`.html` Muss einmal vorkommen  
`$` und am Ende stehen  
`$1` Inhalt von (.\* ) an dieser Stelle setzen  
`.php` wirklich Existierende Endung

index.php => index.html  
gbuch.php => gbuch.html ...

### *Übergabe einer Variable*

```
RewriteRule ^viewPage (.*) (\\.html?)$ person.php?seite=$1
```

person.php?seite1 => viewPage1.html

### *Übergabe zweier Variablen*

```
RewriteRule ^fotos(.)(.)(\.html?)$ wallpapers.php?view=$2&sort=$1
```

wallpapers.php?view=1^&sort=N => fotosN1.html