

-
- [SEO Werkzeuge](#)
- [SEO Lektüren](#)
- [Impressum / Kontakt](#)

[Anmelden](#)

SEO-Snippets

Suchergebnisse

Ergebnis 1 - 1 von 1 für **htaccess Tutorial – Was man als SEO kennen sollte.** (0,164 Sekunden)

[SEO-Snippets by Marina Eins](#)

[Marina Eins](#) ist Head of Online Marketing bei der web care LBJ GmbH.

Auf diesem Blog berichte ich von Suchmaschinenoptimierung und sonstigen Online Themen, die mir bei der Arbeit im Netz ins Netz gehen.



Kategorien

- [\(Sub\)domains und Verzeichnisse](#)
- [Allgemein](#)
- [htaccess](#)
- [JavaScript](#)
- [Links](#)
- [PageRank](#)
- [Wordpress](#)

 [Abonniere den RSS Feed](#)

htaccess Tutorial – Was man als SEO kennen sollte

Auch wenn SEO-Beratung und deren genaue technische Umsetzung oft zweierlei Paar Schuhe sind, und die Umsetzung meist bei den Programmieren liegt, so ist es auch für SEOs unabdingbar einige der meistgenutzten .htaccess-Befehle zu kennen. Aus diesem Grunde gibt es heute einmal eine kleine Einführung von mir in die Welt der .htaccess.

Inhalt:

1. [Was ist die .htaccess und wofür brauche ich sie?](#)
2. [Wie erstelle ich die .htaccess-Datei?](#)

3. [Oft verwendete Beispiele für .htaccess-Codes:](#)
 - [Bestimmen einer Startdatei als Startseite](#)
 - [Umleitung der Startdatei auf das Stammverzeichnis](#)
 - [301 Weiterleitung einer Domain-URL von ohne auf mit www](#)
 - [301 Weiterleitung einer Domain-URL von mit auf ohne www](#)
 - [301 Weiterleitung einer alten URL auf eine neue URL](#)
 - [301 Weiterleitung aller Dateien eines Verzeichnisses, auf eine andere Datei, bis auf eine Ausnahme](#)
 - [Umbenennung und Weiterleitung dynamischer URL auf eine statische Seite](#)
 - [Entfernen eines Zwischenverzeichnisses aus der URL-Struktur](#)
 - [Umbenennen von angezeigten Browser-URL-Pfaden](#)
 - [Fehlerseiten bestimmen](#)
 - [Expires Header verwenden \(Caching für einen bestimmten Zeitraum erzwingen\)](#)
 - [Gzip aktivieren \(zum komprimieren von Dateien auf dem Server\)](#)
 - [Nur bestimmten IPs den Zugriff auf den Server gewähren](#)
 - [Nur bestimmte IPs-Sperren](#)
 - [Sperren für bestimmte Host-Websites und Anfangs-IPs](#)
 - [Datei\(en\) sperren](#)
 - [PHP-Befehle in .html-Seiten ausführen](#)
4. [Mehr zum Thema .htaccess ...](#)

Was ist die .htaccess und wofür brauche ich sie?

Die sogenannte .htaccess-Datei steht für "hypertext access"- bzw. "Hypertext-Zugriffs"-Datei und ist eine Konfigurationsdatei, in der verzeichnisspezifische Einstellungen auf NCSA-kompatiblen Webservern, wie bspw. Apache, vorgenommen werden können. Umsetzen lassen sich mit ihr Zugriffsberechtigungen für Dateien und Verzeichnisse oder aber die Einstellungen für interne Verknüpfungen und Weiterleitungen, sowie für Fehlerseiten. Und dies alles wohlgermerkt ohne die Notwendigkeit den Server neu starten zu müssen. Die Änderungen in der .htaccess treten dabei sofort in Kraft, da sie bei allen Webserver-Anfragen sofort ausgewertet werden. Wirkung zeigen die Befehle der .htaccess nur in dem Verzeichnis, in dem sie abgespeichert wurde, sowie in den zugehörigen Unterverzeichnissen, in denen die Befehle aber auch wieder überschrieben werden können.

Wie erstelle ich die .htaccess-Datei?

Die .htaccess selbst kann im stinknormalen Windows-Editor oder einem anderen frei wählbaren Editor erstellt werden. Wichtig beim Abspeichern ist nur, dass kein Name für die Datei vergeben wird und die Endung .htaccess lautet. Abgespeichert wird die .htaccess-Datei im Hauptverzeichnis/Root des Webserver oder einfach in dem Überordner, von dem aus sie Wirkung zeigen soll.

Oft verwendete Beispiele für .htaccess-Codes

Anmerkung:

Die Zeile "RewriteEngine On" brauchen wir, damit das sogenannte mod_rewrite Modul von Apache läuft, sie muss aber insgesamt nur einmal in der htaccess angegeben werden, wenn Rewrite-Befehle verwendet werden.

Bestimmen einer Startdatei als Startseite

Bei vielen Providern lautet der Name der eigentlichen Startseite index.html, index.htm, index.php oder ähnlich. Um eine bestimmte Seite als Startseite festzulegen, kann folgendes eingetragen werden (.htaccess, dabei im Root):

```
DirectoryIndex diestartdatei
```

Beispiel (für index.html als Startseite):

```
DirectoryIndex index.html
```

Umleitung der Startdatei auf das Stammverzeichnis:

Beispiel (für Umleitung von www.domain.de/index.html -> www.domain.de/):

```
RewriteEngine On
RewriteCond %{THE_REQUEST} ^[A-Z]{3,9}\ /index\.(html?|php)\ HTTP/
RewriteRule ^index\.(html?|php)$ http://www.domain.de/ [R=301,L]
```

301 Weiterleitung einer Domain-URL von ohne auf mit www:

Beispiel (für Weiterleitung von domain.de auf www.domain.de):

```
RewriteEngine On
RewriteCond %{HTTP_HOST} ^domain.de [NC]
RewriteRule ^(.*)$ http://www.domain.de/$1 [L,R=301]
```

Alternative Schreibweisen:

```
RewriteCond %{HTTP_HOST} !^www\.domain\.de$ [NC]
RewriteRule ^(.*)$ http://www.domain.de/$1 [L,R=301]
```

oder

```
RewriteCond %{HTTP_HOST} !^www.domain.de$ [NC]
RewriteRule ^(.*)$ http://www.domain.de/$1 [L,R=301]
```

301 Weiterleitung einer Domain-URL von mit auf ohne www:

Beispiel (für Weiterleitung von www.domain.de auf domain.de):

```
RewriteEngine On
RewriteCond %{HTTP_HOST} ^domain.de [NC]
RewriteRule (.*) http://domain.de/$1 [R=301,L]
```

Alternative Schreibweise:

```
RewriteCond %{HTTP_HOST} ^www.domain.de [NC]
RewriteRule ^(.*)$ http://domain.de/$1 [L,R=301]
```

301 Weiterleitung einer alten URL auf eine neue URL:

Beispiel (Weiterleitung von /relativer-pfad/alte-url.html auf http://www.domain.de/absoluter-pfad/neue-url.html):

```
Redirect 301 /relativer-pfad/alte-url.html http://www.domain.de/absoluter-pfad/neue-url.html
```

Alternative Schreibweise:

```
Redirect permanent /relativer-pfad/alte-url.html http://www.domain.de/absoluter-pfad/neue-url.html
```

301 Weiterleitung aller Dateien eines Verzeichnisses, auf eine andere Datei, bis auf eine Ausnahme

Beispiel (Weiterleitung aller Dateien des Verzeichnisses /produkte/ auf /produkte.php, ausser /produkte/separat.php):

```
RewriteEngine On
RewriteRule ^produkte/separat.php /separat.html [L,R=301]
RewriteRule ^produkte(.*)$ /produkte.php [L,R=301]
```

Umbenennung und Weiterleitung dynamischer URL auf eine statische Seite

Beispiel1:

(Weiterleitung von http://www.domain.de/article.jsp?id=8932 auf http://www.domain.de/neueste-nachrichten.htm)

```
RewriteEngine On
RewriteRule ^article.jsp?id=(.*)$ /neueste-nachrichten.htm [L,R=301]
```

Beispiel2:

(<http://www.domain.de/index.php?go=home> umbenennen in <http://www.domain.de/home.html>)

```
RewriteRule ^(.*)\.html$ index.php?go=$1 [L]
```

Entfernen eines Zwischenverzeichnisses aus der URL-Struktur

Beispiel

(<http://domain.de/category/category-name> anzeigen als <http://domain.de/category-name>):

```
RewriteEngine On
RewriteRule ^category/(.+)$ http://www.domain.de/$1 [R=301,L]
```

Umbenennen von angezeigten Browser-URL-Pfaden

Beispiel:

(<http://domain.de/ordner-mit-keyword/keyword.html> anzeigen statt

http://domain.de/ein_ordner/de/nochn_ordner/index.php)

```
RewriteEngine On
RewriteCond %{REQUEST_URI} ordner-mit-keyword/keyword.html$
RewriteRule (.*) ein_ordner/de/nochn_ordner/index.php
```

Von diesen URL-Paaren können beliebig viele angelegt werden.

Fehlerseiten bestimmen:

```
ErrorDocument [derStatusCode] [/[derdokumentenpfad]/dasdokument.html
```

Beispiel (für Zuweisung einer 404-Seite):

```
ErrorDocument 404 /notfound.html
```

([Übersicht der Statuscodes für Fehlerseiten](#))

Expires Header verwenden (Caching für einen bestimmten Zeitraum erzwingen):

Beispiel

(Default-Einstellung auf 1 Tag, für einzelne Dateitypen auf x;
statt seconds kann auch hour, week, month, etc. gewählt werden):

```
ExpiresActive on
ExpiresDefault "access plus 86400 seconds"

ExpiresByType image/gif "access plus x seconds"
ExpiresByType image/jpeg "access plus x seconds"
ExpiresByType image/png "access plus x seconds"
ExpiresByType image/x-icon "access plus x seconds"ExpiresByType text/css "access plus x seconds"
ExpiresByType application/javascript "access plus x seconds"
ExpiresByType application/x-javascript "access plus x seconds"
ExpiresByType application/pdf "access plus x seconds"
ExpiresByType text/plain "access plus x seconds"
```

Gzip aktivieren (zum komprimieren von Dateien auf dem Server)

Klappt nur, wenn das Apache-Modul mod_gzip installiert ist

```
mod_gzip_on Yes
mod_gzip_dechunk Yes
mod_gzip_item_include file \.(html?|txt|css|js|php|pl)$
mod_gzip_item_include handler ^cgi-script$
mod_gzip_item_include mime ^text/*
mod_gzip_item_include mime ^application/x-javascript.*
mod_gzip_item_exclude mime ^image/*
```

```
mod_gzip_item_exclude rspheader ^Content-Encoding:.*gzip.*
```

Falls das Modul nicht auf dem Server installiert ist, dann kann alternativ PHP die Arbeit übernehmen, dazu folgenden Code in die `.htaccess` eintragen:

```
php_value output_handler ob_gzhandler
```

Diese Variante frisst allerdings auch mehr CPU.

Nur bestimmten IPs den Zugriff auf den Server gewähren

```
order deny,allow
deny from all
allow from xx.xx.xx.xx
```

Weitere IPs können bei Bedarf durch weitere "allow from"-Zeilen hinzugefügt werden

Nur bestimmte IPs-Sperren

```
order allow,deny
allow from all
deny from 27.101.84.2001
```

Sperren für bestimmte Host-Websites und Anfangs-IPs

Beispiel:

(Sperren aller Hosts aus der Domain `domain.de`, den Host `traffic.website.com` und IPs beginnend mit `192.168`):

```
order allow,deny
deny from .domain.de
deny from traffic.website.com
deny from 192.168
allow from all
```

Datei(en) sperren

Beispiel1 (Sperren nur der `.htaccess`; eingesetzt werden kann hier eine beliebige Seite, File oder Dateityp)

```
<Files .htaccess>
order allow,deny
deny from all
</Files>
```

Beispiel2 (Sperren aller Dateien, die mit einem `.` beginnen, wie z.B. die `.htaccess`)

```
<FilesMatch "^\. " >
deny from all
</FilesMatch>
```

Beispiel3 (Sperren von Dateien mit Endung `.tpl`, `.ini`, `.htaccess`)

```
<FilesMatch "\.(tpl|ini|htaccess)$">
order allow,deny
deny from all
</FilesMatch>
```

PHP-Befehle in .html-Seiten ausführen

```
AddType application/x-httpd-php .html
AddType application/x-httpd-php .htm
```

oder:

```
AddType application/x-httpd-php .html .htm
```

oder für neueren Versionen von PHP bzw. Apache:

```
AddHandler application/x-httpd-php .html .htm
```

Und wenn gar nichts passiert, kann als Letztes dieses probiert werden:

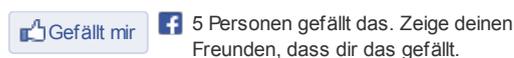
```
AddHandler x-httpd-php .html .htm
```

Mehr zum Thema .htaccess ...

gibt's auf [SELFHTML](#) oder in diesen netten Artikeln von:

- [seobook.com](#)
- [methode-web.de](#)
- [makeitrank.com](#)
- [perfekte-website.de](#)
- [hp-profi.info](#)

Wenn euch noch weitere nützliche .htaccess-Spielereien einfallen, die man als SEO ebenfalls kennen sollte, so tut euch keinen Zwang an und kommentiert gerne. Oder teilt mir einfach mit, ob ihr dieses Tutorial hilfreich fandet oder nicht. 😊



Kategorie: [htaccess](#)

« [PageRank Sculpting – Von Nofollow, JavaScript und JS Crypting für Links](#)
[Unterverzeichnis, Subdomain oder Domain – Die Website erweitern](#) »

Hinterlasse einen Kommentar

Name (required)

Mail (will not be published) (required)

Website

Submit Comment

- [Design: Premium Wordpress Themes](#)