



Eine Einführung in die Serverkonfiguration mit .htaccess

Auf zwei Dritteln aller Server steht die Konfigurationsdatei .htaccess zur Verfügung. Mit ihr lässt sich mehr als nur ein Passwortschutz realisieren.

Die Datei kann mit jedem beliebigen Texteditor bearbeitet werden. Der unvertraute Dateiname rührt einfach daher, dass diese Datei aus der Unix-Welt stammt. Eine Warnung vorweg. Hier muss sorgfältig gearbeitet werden. Garantien für eventuelle Schäden übernimmt Dr. Web nicht.

Dieses Dokument ist weder kopiergeschützt, noch gesichert. Bitte respektieren Sie unser Copyright und geben Sie es nicht an Dritte weiter. Es ist nur für unsere Abonnenten bestimmt.

© 2003 drweb.de AG iGr. - Königstraße 97-99 - 23552 Lübeck
Telefon 0451/3072970 - Nachdruck und Weiterverbreitung,
auch auszugsweise, nur mit Genehmigung.

Auch wenn kein eigener Server zur Verfügung steht, ist der Webmaster nicht hilflos. Die Datei `.htaccess` auf NCSA und kompatiblen Servern (Apache) ermöglicht zahlreiche Einstellungen für Verzeichnisse oder einzelne Dateien. Das ist ausgesprochen hilfreich, da auf diese Weise Verzeichnisse mit einem Zugriffsschutz versehen werden können, Umleitungen ermöglicht oder Benutzer gesperrt oder explizit zuzulassen werden können, sogar das Erscheinungsbild der FTP Verzeichnisse kann beeinflusst werden, individuelle Fehlerdokumente, die Verwendung von CGI Scripts und SSI beeinflussen und vieles mehr lässt sich festlegen.

Die Bezeichnung `.htaccess` geht auf den Begriff Hypertext-Access zurück. Die Datei selbst ist eine einfache ASCII Datei.

Auf dem Webserver können für jeden Webespace in jedem Verzeichnis `.htaccess` Dateien definiert werden, wobei sie für alle Unterverzeichnisse gilt, es sei denn dort wurde eine weitere `.htaccess` Datei definiert. Auf diese Weise ist es dem Webmaster möglich, seinen Bereich selbst zu konfigurieren, ohne dass der Server an sich verändert wird oder ein Neustart nötig wäre, da die betroffene `.htaccess` Datei bei jedem

Abruf einer Datei vom Server ausgewertet wird.

Dies alles klappt allerdings nur, wenn der Provider es in seiner generellen Serverkonfiguration ausdrücklich erlaubt hat. Der Provider hat nicht nur die Möglichkeit, einzelne Befehle zu sperren, er könnte die Datei auch beliebig anders benennen, dies geschieht über die `httpd.conf`.

Mitunter ist die Datei zwar vorhanden, aber im FTP-Programm unsichtbar, da der Punkt am Anfang des Dateinamens unter Unix/ Linux eine versteckte Datei kennzeichnet. Da man aber den genauen Namen weiß, kann man direkt auf die Datei zugreifen, unter Telnet beispielsweise mit „`emacs .htaccess`“, bei FTP-Programmen kann man den Dateinamen direkt in den Kopieren-Dialog eingeben.

Unter Windows ist es – im Gegensatz zu Unix/ Linux – ein Problem, eine solche Datei mit Punkt am Anfang zu erzeugen, einfaches Umbenennen klappt nicht. Auch Notepad kann solche Dateien im Normalfall nicht speichern. Hier gibt es einen Trick: den Dateinamen beim Speichern in Anführungszeichen schreiben: `".htaccess"`, dann gibt es keine Probleme. In anderen Editoren klappt es eventuell auch ohne die Anführungszeichen.

Beispiele

Kommentare werden mit einem vorangestellten Gatter gekennzeichnet.

```
# Dies ist ein Kommentar
```

Der Webserver wird die Startseiten in dieser Reihenfolge ausgeben. Ist eine index.html vorhanden, wird sie zum Browser geschickt, ist sie nicht vorhanden, wird der Server start.php versuchen und so weiter.

```
DirectoryIndex index.html start.php hallo.htm
```

Ein eigenes Fehlerdokument mit beliebigem Inhalt wird ausgegeben, wenn eine Datei nicht gefunden wird (Error 404). Ebenso können andere Fehlermeldungen abgefangen werden, zu empfehlen sind 403 (Unerlaubter Zugriff: „Authorisation Required“) sowie 500 (Serverfehler).

```
ErrorDocument 404 http://www.heise.de/fehler.shtml
```

Statt der Standardfehlermeldung wird die angegebene angezeigt. Das Anführungszeichen darf nur einmal auftauchen.

```
ErrorDocument 404 "Die angegebene Datei wurde nicht gefunden
```

Besonders viele Möglichkeiten gibt es in Sachen Indexing

```
FancyIndexing On
```

Verändert das Aussehen eines Verzeichnislistings, welches ausgegeben wird falls keine Startseite im Verzeichnis gefunden wird. Beschreibungen können ein- oder ausgeblendet, die Größe und Art der Icons bestimmt, Dateigröße und Datum ausgeblendet werden.

```
Order Allow, Deny
```

```
Order Deny, Allow
```

Order gibt die Rangfolge an, die für die nachfolgenden Allow und Deny Anweisungen gelten soll.

Durch Angabe einer IP Adresse oder einer Domain können bestimmte Nutzer ausgesperrt werden, „all“ sperrt alle Benutzer aus. Sie bekommen den Fehler 403 zu sehen. Sollen verschiedene IPs/ Benutzer gesperrt werden, müssen mehrere Deny Anweisungen angegeben werden.

```
Deny from 111.222
```

```
Deny from .de
```

```
Deny from all
```

Diese folgende Option gibt an, welche Benutzer erlaubt sind. Durch die Kombination mit der Order Anweisung können einzelne Benutzer innerhalb einer ausgeschlossen Gruppe erlaubt werden. Auch hier muss für jeden Benutzer eine Allow Anweisung eingefügt werden.

```
Allow from 111.222.333
Allow from firma.de
Allow from all
```

Mit <Limit> kann festgelegt werden welche Nutzer die HTTP-Befehle GET, PUT, POST ausführen dürfen, es können die Befehle Order, Allow, Deny und Require beliebig verwendet werden.

Alle Benutzer können alles abrufen.

```
<Limit GET>
    Order allow, deny
    Allow from all
    Deny from none
</Limit>
```

Grundsätzlich werden alle Benutzer von den Befehlen PUT und POST ausgeschlossen, allerdings wird durch Order angegeben, dass die Allow Anweisung dies überschreibt, wodurch Benutzer, deren Host mit .de endet zugelassen sind. Zusätzlich müssen sie noch der Benutzergruppe "autoren" angehören.

```
<Limit PUT, POST >
    Order deny, allow
    Deny from all
    Allow from .de
    Require autoren
</Limit>
```

Mit AddType weist man den Server an, Dateien mit den Endungen .html und .abc als SSI-Dokumente zu parsen. Dies bedeutet eine größere Belastung des Servers, da nicht mehr nur .shtml Dateien sondern alle Daten geparkt werden. Nur sinnvoll, wenn tatsächlich alle Dateien SSI-Anweisungen enthalten.

```
AddType text/x-server-parsed-html .html .abc
```

Alle Dateien mit den aufgeführten Endungen werden nun mit dem MIME-Typ „JPEG“ gesendet.

```
AddType image/jpeg .jpg .jpeg .jp
```

Verzeichnisse schützen

Es gibt die Möglichkeit, einzelne Verzeichnisse mit Passwörtern für verschiedene Nutzer zu versehen. Im Browser erscheint ein Dialogfenster, das zur Eingabe von Name und Passwort auffordert. Falls keine oder eine falsche Eingabe erfolgt, bekommt man einen Error 401 ("Authorisation Required").

Die Passwörter werden in einer speziellen Datei namens `.htpasswd` gespeichert. Natürlich können sie auch in jeder beliebigen anderen Datei gespeichert werden, diese können aber – sofern der Dateiname bekannt ist – im Browser aufgerufen werden. Eine andere Möglichkeit, die Passwortdatei zu schützen ist, sie im zu schützenden Verzeichnis abzulegen.

Das Passwort wird in der Datei verschlüsselt abgelegt, zur Erzeugung gibt es verschiedene Möglichkeiten. Eine ist die Benutzung des Programms `htpasswd`, das beispielsweise per Telnet aufgerufen werden kann.

So wird es benutzt:

```
htpasswd [-cmdps] passwortdatei benutzername  
htpasswd -b[cmdps] passwortdatei benutzername passwort
```

- c Erstellt eine neue Datei
- m MD5-Verschlüsselungsmethode verwenden (Windows/ TPF-Standard).
- d CRYPT-Verschlüsselungsmethode verwenden (Unix/ Linux-Standard)
- p Passwort nicht verschlüsseln (Klartext)
- s SHA-Verschlüsselungsmethode verwenden
- b Übernimmt das Passwort von der Kommandozeile

Nun wird die Passwortdatei per `htpasswd -c .htpasswd sven` angelegt. Das Programm fragt nach einem Passwort, das noch einmal bestätigt werden muss, dann wird es in der Datei gespeichert. Die Eingabe des Passwortes wird nicht angezeigt. Beim Hinzufügen eines weiteren Benutzers darf der Parameter `-c` nicht noch mal verwendet werden, da sonst die bestehende Datei überschrieben werden würde. Sollte der Benutzername schon vorhanden sein, so wird das alte Passwort ersetzt.

```
[user@ns htdocs]$ htpasswd -c .htpasswd klaus
```

```
New password:
```

```
Re-type new password:
```

```
Adding password for user klaus
```

```
[user@ns htdocs]$ htpasswd .htpasswd gerd
```

```
New password:
```

```
Re-type new password:
```

```
Adding password for user gerd
```

Die .htpasswd Datei hat jetzt folgenden Inhalt:

```
klaus:Kvu1.2/mgOmEQ  
gerd:1QdiJ4XJ9H6XQ
```

Man kann die .htpasswd Datei auch von Hand anlegen, zum Verschlüsseln der Passwörter gibt es im Internet Seiten, die nach Eingabe des Passwortes im Klartext die Verschlüsselung zurückliefern. Dieses einfach hinter dem Benutzernamen wie im obigen Beispiel in die Datei hinter dem Benutzernamen einfügen und auf dem Webserver abspeichern.

Um dem Webserver mitzuteilen, welche User zugelassen sind, wird folgendes in die .htaccess Datei eingefügt des zu schützenden Verzeichnisses eingefügt:

```
AuthType Basic  
AuthName "Geschützter Bereich"  
AuthUserFile /htdocs/.htpasswd  
AuthGroupFile /htdocs/.htgroups  
require valid-user
```

AuthName ist eine Bezeichnung des Bereiches, dieser wird im Browser gespeichert, so dass er nicht jedes Mal nach dem Passwort fragt. Falls er Leerzeichen enthält, muss er in Anführungszeichen stehen. AuthUserFile ist die oben erzeugte Datei, in der die Benutzer nebst Passwörtern aufgelistet sind.

AuthGroupFile bezeichnet eine Datei, die Benutzergruppen enthält, in jeder Zeile steht eine Gruppe mit beliebig vielen Nutzern. Beispiel:

```
autoren: klaus gerd
```

Diese Gruppe kann unter require aufgeführt werden:

```
require group autoren
```








Andere Möglichkeiten für require sind: valid-user: irgend ein Benutzer, der in der Passwortdatei vorkommt; user sven: Nur Sven ist zugelassen; user arne sven: Arne und Sven sind zugelassen; group autoren: alle, die der Gruppe „Autoren“ zugehörig sind, sind zugelassen. Es können auch mehrere require-Anweisungen untereinander stehen.

Problembehebung: Stimmt z.B. die Syntax nicht wird ein 'Internal Server Error' angezeigt. Um der Sache auf den Grund zu gehen, hilft oft ein Blick in die Error Logs des Servers.










Index-Optionen

Ist in einem Verzeichnis keine Index-Datei vorhanden (etwa index.html oder eine durch DirectoryIndex definierte), wird im Browser ein Verzeichnislisting angezeigt. Um diese Ansicht zu formatieren, gibt es einige Einstellungen, vorausgesetzt sie wurde nicht durch „Options-Indexes“ komplett unterdrückt. In diesem Falle wird ein Error 403 gesendet.

Mit „FancyIndexing On“ wird dem Server mitgeteilt, dass die individuellen Einstellungen zu beachten sind, diese Angabe ist aber nicht unbedingt immer notwendig, hier muss man ausprobieren. Das Listing besteht aus fünf Spalten: ein Icon (Bildsymbol), der Name der Datei, das Datum der letzten Änderung, die Dateigröße sowie eine Beschreibung.

Index of /images				
<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Desc</u>	
 Parent Directory	24-Oct-2001 18:17	-		
 alien_468_06.gif	24-Oct-2001 18:15	7k		
 belleandsebastian.gif	24-Oct-2001 18:15	12k		
 bulo.gif	24-Aug-2001 11:49	1k		
 fold2.gif	24-Oct-2001 18:15	20k		
 illu_doc.jpg	24-Oct-2001 18:15	4k		
 illu_everlast.jpg	24-Oct-2001 18:15	4k		

Standard Listing

Index of /images				
<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Desc</u>	
 Parent Directory	24-Oct-2001 18:17	-		
 link.gif	24-Oct-2001 18:21	1k	GIF-	
 tindersticks.jpg	24-Oct-2001 18:15	24k	JPG-	
 stuartpidgeon.jpg	24-Oct-2001 18:15	13k	JPG-	
 schnittke1.gif	24-Oct-2001 18:15	11k	GIF-	
 schnittke.gif	24-Oct-2001 18:15	21k	GIF-	
 opix.gif	24-Oct-2001 18:15	1k	GIF-	
 illu_music.jpg	24-Oct-2001 18:15	3k	JPG-	
 illu_geld.jpg	24-Oct-2001 18:15	4k	JPG-	

Verändertes Listing

AddIcon Icon Datei

Das Standardicon der Datei wird zu durch das angegebene Icon ersetzt. Für das Icon kann ein relativer oder absoluter Pfad angegeben werden, die Datei kann eine bestimmte Datei sein oder auch durch Wildcards definiert werden, auch für ein Verzeichnis kann ein Icon gesetzt werden.

AddIconByEncoding Icon MIME-Encoding-Type

Ebenso wie AddIcon kann auch hier das Icon festgelegt werden, dieses Mal ist das Kriterium der MIME-Encoding-Typ der Datei. Dieser kann durch „AddType“ festgelegt werden.

AddIconByType Icon MIME-Typ

Ändert das Icon aller Dateien, die dem angegebenen MIME-Typ entsprechen.

DefaultIcon

legt für alle Icons, für die kein Icon zugewiesen wurde, ein Standardicon fest.

AddAlt Text Datei

AddAltByEncoding Text MIME-Encoding-Type

AddAltByType

Ändert die Standardbeschreibung der Datei („Alt-Text“) von „[IMG]“ bzw. „[DIR]“ in den angegebenen Text. Beispiel:

AddAlt Grafik *.gif

Zeigt als Alternativ Text statt [ALT] den Text [Grafik]

IndexIgnore Dateinamen

Alle Dateien, die nicht im Listing aufgeführt werden sollen, können hier angegeben werden, es sind Dateinamen und Wildcards erlaubt.

IndexOrderDefault Reihenfolge Spalte

Gibt die Standardeinstellung der Sortierung vor, die Reihenfolge kann aufsteigend („Ascending“) oder absteigend („Descending“) sein, die Spaltennamen sind: „Name“, „Size“, „Date“, „Description“.

AddDescription Text Datei

Legt für die angegebene Datei eine Beschreibung fest, die in der letzten Spalte angezeigt wird. Beispiel:

AddDescription GIF-Bild *.gif

Der Text "GIF-Bild" wird als Beschreibung für alle Dateien mit der Endung .gif angezeigt. Siehe auch Abbildung. Innerhalb des Textes sind keine Leerzeichen erlaubt.

HeaderName Dateiname

Normalerweise wird ganz oben beim Listing der aktuelle Pfad ausgegeben, durch die Angabe einer Datei wird diese stattdessen angezeigt.

`ReadmeName` `Dateiname`

So wie die Headerdatei ganz oben angezeigt wird, wird die Readme-Datei unterhalb des Listings angezeigt.

`IndexOptions` [Optionen]

Dieser Befehl erlaubt es, das Verzeichnislisting sehr stark zu beeinflussen. Die verschiedenen Optionen:

`IconHeight=xx`

legt eine einheitliche Höhe für alle Icons fest.

`IconWidth=xx`

legt analog dazu die Breite fest.

Beispiel:

`IndexOptions IconHeight=16 IconWidth=16`

Alle Icons sind 16 Pixel hoch und 16 Pixel breit

`IconsAreLinks`

stellt alle Icons als Links auf die entsprechenden Dateien dar.

`NameWidth=xx`

setzt die Spaltenbreite in Zeichen, bei Angabe eines „*“ richtet sich der Server nach dem längsten Dateinamen.

`ScanHTMLTitles`

gibt in der Spalte Description den Titel von HTML-Dateien aus. Hinweis: hierdurch wird der Server stark belastet.

`SuppressColumnSorting`

Die Spalten können nicht mehr durch Klicken auf die Überschrift sortiert werden.

`SuppressDescription`, `SuppressLastModified`, `SuppressSize`

unterdrückt das Anzeigen jeweiligen Spalten des Verzeichnislistings.

URL Weiterleitung

Eine ausgesprochen nützliche Funktion bietet die so genannten Redirection. Damit ist es möglich Zugriffe auf Verzeichnisse oder Einzeldateien umzuleiten. Sogar auf externe Dateien oder Verzeichnisse kann dabei verwiesen werden.

Die Syntax ist einfach zu handhaben

Redirect /styles http://www.url.de

Leitet jeden Zugriff auf das Verzeichnis "styles" auf die Domain heise.de um.
Versuchen Sie einmal http://www.drwebmagazin.de/styles

Redirect /seite1.html seite2.html

Leitet die Anfrage auf seite1.html auf seite2.html um

Redirect / http://www.url.de

Leitet jede Anfrage auf die Domain url.de um

Eine solche Funktion kann durchaus Sinn machen, beispielsweise während größerer Bauarbeiten oder technischer Probleme. Außerdem lassen sich nicht mehr vorhandene oder umbenannte Dateien bequem ersetzen, ohne das dazu Dummy-Dateien angelegt werden müssen.

Tipp

Wer ein Problem damit hat, die Datei .htaccess unter Windows anzulegen (der Punkt macht Schwierigkeiten), benenne sie irgendwie und verleihe ihr nach dem Upload auf dem Server einen neuen Namen. Oder man nehme das gute alte DOS.

- Dos-Box öffnen
- copy con .htaccess eingeben
- F6 drücken und mit Return bestätigen
- Voilà, es existiert eine echte .htaccess im gewählten Verzeichnis, die mit dem Editor anschließend bearbeitet werden kann.

Befehlsübersicht

- Allow
- Deny
- Order
- Require
- AddType
- AuthUserFile
- AuthGroupFile
- AuthType
- AuthName

- DefaultType
- ErrorDocument
- ForceType
- Options
- Redirect
- Satisfy
- <Files> </Files>

Drei weiterführende Links zu Originaldokumenten in englischer Sprache

Apache 1.3 URL Rewriting Guide

<http://httpd.apache.org/docs/misc/rewriteguide.html>

A Users Guide to URL Rewriting with the Apache Webserver

<http://www.engelschall.com/pw/apache/rewriteguide/#ToC16>

Authentication, Authorization, and Access Control

<http://httpd.apache.org/docs/howto/auth.html>

© 2003 <http://www.drweb.de>