21 Very Useful Htaccess Tips & Tricks

Apache web servers has a great way to manipulate information using **.htaccess** files. **.htaccess (hypertext access)** is the default name of a directory-level configuration file that allows for decentralized management of web server configuration. The **.htaccess file** is placed inside the web tree, and is able to override a subset of the server's global configuration; the extent of this subset is defined by the web server administrator. The original purpose of .htaccess was to allow per-directory access control (e.g. requiring a password to access the content), hence the name. Nowadays .htaccess can override many other configuration settings, mostly related to content control, e.g. content type and character set, CGI handlers, etc.

Following are few very useful htaccess tricks.

1. Custom Directory Index Files

```
DirectoryIndex index.html index.php index.htm
```

You can change a default index file of directory by using above snippet in your htaccess file. If a user request **/foo/**, Apache will serve up **/foo/index.html**, or whatever file you specify.

2. Custom Error Pages

```
ErrorDocument 404 errors/404.html
```

You may want to redirect your users to an error page is any of the http errors like 404 occurs. You can use above snippet in htaccess file to map 404 error to error page errors/404.html. Also you may want to write a common page for all the http errors as follows:

```
ErrorDocument 404 /psych/cgi-bin/error/error?404
```

3. Control access at files & directory level

.htaccess is most often used to restrict or deny access to individual files and folders. A typical example would be an "includes" folder. Your site's pages can call these included scripts all they like, but you don't want users accessing these files directly, over the web. In that case you would drop an .htaccess file in the includes folder with content something like this.

```
# no one gets in here!
deny from all
```

which would deny ALL direct access to ANY files in that folder. You can be more specific with your conditions, for instance limiting access to a particular IP range, here's a handy top-level rule for a local test server.

```
# no nasty crackers in here!
order deny,allow
deny from all
allow from 192.168.0.0/24
# this would do the same thing..
#allow from 192.168.0
```

Generally these sorts of requests would bounce off your firewall anyway, but on a live server they become useful for filtering out undesirable IP blocks, known risks, lots of things.

Sometimes, you will only want to ban one IP, perhaps some persistent robot that doesn't play by the rules.

```
# someone else giving the ruskies a bad name..
order allow,deny
deny from 83.222.23.219
allow from all
```

4. Modifying the Environment Variable

Environment variables contain information used by server-side includes and CGI. Set / Unset environment variables using **SetEnv** and **UnSetEnv**.

```
SetEnv SITE_WEBMASTER "Jack Sprat"
SetEnv SITE_WEBMASTER_URI mailto:Jack.Sprat@characterology.com

UnSetEnv REMOTE_ADDR
```

5. 301 Redirect using htaccess

If you want to redirect from an old document to new:

```
Redirect 301 /old/file.html http://yourdomain.com/new/file.html
```

Use following for redirecting Entire Directory.

```
RedirectMatch 301 /blog(.*) http://yourdomain.com/$1
```

6. Implementing a Caching Scheme with .htaccess

Cache the static files and improve your website’s performance. (read this article: PHP, CSS, JS Compression for full implementation)

```
# year
<FilesMatch "\.(ico|pdf|flv|jpg|jpeg|png|gif|swf|mp3|mp4)$">
Header set Cache-Control "public"
Header set Expires "Thu, 15 Apr 2010 20:00:00 GMT"
Header unset Last-Modified
</FilesMatch>
#2 hours
<FilesMatch "\.(html|htm|xml|txt|xsl)$">
Header set Cache-Control "max-age=7200, must-revalidate"
</FilesMatch>
<FilesMatch "\.(js|css)$">
SetOutputFilter DEFLATE
Header set Expires "Thu, 15 Apr 2010 20:00:00 GMT"
</FilesMatch>
```

7. Compress output using GZIP

Add following snippet into your htaccess file and compress all the css, js, html files with GZip compression.

```
<IfModule mod_gzip.c>
    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.*
</IfModule>
```

Above code works only if mod_gzip module is enabled in your webserver. You may want to add following snippet if your webserver provides mod_deflate support.

```
<Location>
    SetOutputFilter DEFLATE
      SetEnvIfNoCase Request_URI  \
        \.(?:gif|jpe?g|png)$ no-gzip dont-vary
    SetEnvIfNoCase Request_URI  \
        \.(?:exe|t?gz|zip|gz2|sit|rar)$ no-gzip dont-vary
</Location>
```

If your webserver does not support mod_deflate then you may want to use following snippet.

```
<FilesMatch "\.(txt|html|htm|php)">
    php_value output_handler ob_gzhandler
</FilesMatch>
```

Read this articles for more detail: **Compressing PHP, CSS, JavaScript(JS)**.

8. Redirect browser to https (ssl)

Add following snippet to your htaccess and redirect entire website to https.

```
RewriteEngine On
RewriteCond %{HTTPS} !on
RewriteRule (.*) https://%{HTTP_HOST}%{REQUEST_URI}
```

9. Rewrite URLs using htacccess

Rewriting product.php?id=12 to product-12.html

```
RewriteEngine on
RewriteRule ^product-([0-9]+)\.html$ product.php?id=$1
```

Rewriting product.php?id=12 to product/ipod-nano/12.html

```
RewriteEngine on
RewriteRule ^product/([a-zA-Z0-9_-]+)/([0-9]+)\.html$ product.php?id=$2
```

Redirecting non www URL to www URL

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

Rewriting yoursite.com/user.php?username=xyz to yoursite.com/xyz

```
RewriteEngine On
RewriteRule ^([a-zA-Z0-9_-]+)$ user.php?username=$1
RewriteRule ^([a-zA-Z0-9_-]+)/$ user.php?username=$1
```

Redirecting the domain to a new subfolder of inside public_html

```
RewriteEngine On
RewriteCond %{HTTP_HOST} ^test\.com$ [OR]
RewriteCond %{HTTP_HOST} ^www\.test\.com$
RewriteCond %{REQUEST_URI} !^/new/
RewriteRule (.*) /new/$1
```

10. Prevent Directory Listing

Add any of the following snippet to avoid directory listing.

```
Options -Indexes
```

or

```
IndexIgnore *
```

Read this article on more details on **Denying/Allowing directory listing**.

11. Adding new MIME types

The type of file depends on the filename extension. Unrecognized file extensions are treated as text data, and corrupted on download.

```
AddType application/x-endnote-connection enz
AddType application/x-endnote-filter enf
AddType application/x-spss-savefile sav
```

12. Deny access to static file data

Denies any request for static files (images, css, etc) if referrer is not local site or empty.

```
RewriteCond %{HTTP_REFERER} !^$
RewriteCond %{REQUEST_URI} !^/(wp-login.php|wp-admin/|wp-content/plugins/|wp-in
RewriteCond %{HTTP_REFERER} !^http://www.askapache.com.*$ [NC]
RewriteRule \.(ico|pdf|flv|jpg|jpeg|mp3|mpg|mp4|mov|wav|wmv|png|gif|swf|css|js)
```

13. Specify Upload file limit for PHP in htaccess

```
php_value upload_max_filesize 20M
php_value post_max_size 20M
php_value max_execution_time 200
php_value max_input_time 200
```

In the above .htaccess file, uploading capability is increased by the four parameter first one is maximum file size for uploading, second one is maximum size of the post data , third one is maximum time in seconds a script is allowed to run before it is terminated by the parser and last one is maximum time in seconds a script is allowed to parse input data such as like file uploads, POST and GET data.

14. Disallow Script Execution

```
Options -ExecCGI
AddHandler cgi-script .php .pl .py .jsp .asp .htm .shtml .sh .cgi
```

15. Change Charset and Language headers

```
AddDefaultCharset UTF-8
DefaultLanguage en-US
```

16. Set Timezone of the Server (GMT)

```
SetEnv TZ America/Indianapolis
```

17. Force “File Save As” Prompt

```
AddType application/octet-stream .avi .mpg .mov .pdf .xls .mp4
```

18. Protecting a single file

Normally .htaccess applies to the entire directory. With the directive you can restrict it to specific files:

```
<Files quiz.html>
order deny,allow
deny from all
AuthType Basic
AuthName "Characterology Student Authcate"
AuthLDAP on
AuthLDAPServer ldap://directory.characterology.com/
AuthLDAPBase "ou=Student, o=Characterology University, c=au"
require valid-user
satisfy any
</Files>
```

19. Set Cookie using htaccess

Set Cookie with environment variable

```
Header set Set-Cookie "language=%{lang}e; path=/;" env=lang
```

Set Cookie based on Request. This code sends the Set-Cookie header to create a cookie on the client with the value of a matching item in 2nd parentheses.

```
RewriteEngine On
RewriteBase /
RewriteRule ^(.*)(de|es|fr|it|ja|ru|en)/$ - [co=lang:$2:.yourserver.com:7200:/]
```

20. Send Custom Headers

```
Header set P3P "policyref=\"http://www.askapache.com/w3c/p3p.xml\""
Header set X-Pingback "http://www.askapache.com/xmlrpc.php"
Header set Content-Language "en-US"
Header set Vary "Accept-Encoding"
```

21. Blocking request based on User-Agent Header

```
SetEnvIfNoCase ^User-Agent$ .*(craftbot|download|extract|stripper|sucker|ninja|
SetEnvIfNoCase ^User-Agent$ .*(libwww-perl|aesop_com_spiderman) HTTP_SAFE_BADBO
Deny from env=HTTP_SAFE_BADBOT
```

Feel free to bookmark this article.

Related Articles

1. **Directory listing in htaccess. Allow, Deny, Disable, Enable Directory Listing in .htaccess**
2. **PHP Fatal Error Maximum Execution Time Issue & htaccess**
3. **Password Protect your webpages using htaccess**
4. **How To Avoid Image Hotlinking & Bandwidth Theft In Your Website**
5. **Compress PHP, CSS, JavaScript(JS) & Optimize website performance.**
6. **20 Very Useful CSS Stylesheet Tips & Tricks**
7. **Redirect your homepage /+ URL to your Google + profile**