Utilizar .HTACCESS

Que es un .htaccess?

El .htaccess (Acceso de Hiper-Texto) es el nombre por defecto del archivo de configuración de directorios de Apache. Se utiliza para personalizar la configuración de directivas y parámetros definidos en el archivo de configuración principal del alojamiento. Tiene una gran variedad de usos y utilidades que le pueden resultar útiles en su web. En el siguiente tutorial le mostramos algunas de las funciones más utilizadas por el .htaccess.

Como crear y donde colocar un fichero .htaccess?

Para crear un fichero .htaccess, abra el bloc de notas e introduzca el código necesario. Guarde el fichero como fichero de texto (.txt), por ejemplo "fichero_htaccess.txt", y súbalo por FTP en la carpeta donde tiene que utilizarse. Una vez en el servidor, modifique el nombre del fichero "fichero_htaccess.txt" por el de ".htaccess".

El .htaccess debe colocarse en el interior de la carpeta donde queremos que tenga efecto. Por ejemplo, si queremos proteger con contraseña una carpeta llamada "privado", colocaremos el .htaccess dentro de la carpeta "privado".

Utilidades comunes del .htaccess

El .htaccess se utiliza para un gran número de utilidades, en este tutorial le mostramos algunas de las utilidades más comunes y como debe configurarse en el fichero .htaccess:

Control de acceso a carpetas

Podrías querer **deshabilitar totalmente el acceso a una carpeta** (por ejemplo, una carpeta con librerías de programación que se incluyen en los archivos principales. En este caso sólo los archivos principales accederána ellos mediante el sistema de archivos, pero no se podrán acceder via web). Bueno, simplemente crea un archivo .htaccess en esa carpeta que contenga:

```
#deny all access
deny from all
```

Si se quiere permitir el acceso desde una IP específica

```
#deny all access
deny from all
allow from 10.0.0.1
```

o para un rango específico de IPs (forzado mediante la máscara de red)

```
allow from 192.168.0.0/24
```

también se puede bloquear el acceso a un archivo específico

```
<Files privado.html>
Order allow,deny
```

```
Deny from all
```

Listado de carpetas

Si se quiere hacer las carpetas navegables, entonces necesitamos agregar esta línea al archivo .htaccess

```
 Options +Indexes +MultiViews +FollowSymlinks
```

Y esta si se tiene el módulo apropiado en el servidor web

```
IndexOptions FancyIndexing
```

También se podría querer prevenir el listado de carpetas

```
IndexIgnore *
```

Activar compresión

Se puede habilitar la compresión de datos inherente de PHP para ahorrar ancho de banda

```
php_value zlib.output_compression 16386
```

Escondiendo archivos

Para deshabilitar el acceso a un archivo en particula se puede utilizar una expresión regular y la directiva **Files** para denegar acceso a cualquier archivo que comience con .ht

Se puede modificar esto para restringir un archivo en particular (como archivos de configuración, robots.txt, archivos de logs o lo que se desee).

```
Order allow,deny
Deny from all
Satisfy All
```

Páginas de error HTTP 404 personalizadas

Si se quisiera redireccionar los visitantes cada vez que se encuentran con una página de error HTTP 404, utiliza éste código:

```
ErrorDocument 404 /errores/noencontrado.html
```

Esto redirije el usuario hacia /errores/noencontrado.html cada vez que sucede un error 404. Obviamente, se puede redefinir para que capture otros errores http (403, 5000, etc). Sigue leyendo para ver lo que encontré

Consejo: Internet Explorer tiene una “funcionalidad” poco documentada que previene la utilización de cualquier página de error 404 personalizada que sea menor a 512 bytes de largo. Los visitantes serán enviados, en cambio, a la página propia de IE, que es genérica y sugiere que utilizen una búsqueda en MSN

para buscar la información en internet. ¡Esa es una forma de perder visitantes! Asegúrate que tu página personalizada esté por sobre este límite - algo así como 10 líneas completas de texto y HTML deberían ser suficientes.

Bloqueo de referers maliciosos - Nada de hotlinking

Si se desea bloquera algunas partes del sitio de cualquier referer malicioso:

```
RewriteEngine on
RewriteCond %{HTTP_REFERER} ejemplo\.com [NC,OR]
RewriteCond %{HTTP_REFERER} otroejemplo\.com
RewriteRule .* - [F]
```

Utilizando el motor rewrite [de reescritura] se denegará el acceso al sitio a cualquier visitante que venga de chicomalo.com u otrositiodesagradable.com. Para evitar el robo de ancho de banda, se puede bloquear el acceso a un archivo en particular (o extensión de archivos).

```
RewriteEngine on
RewriteCond %{HTTP_REFERER} !^$
RewriteCond %{HTTP_REFERER} !^http://([-a-z0-9]+\.)?example\.com[NC]
RewriteRule .*\.(zip|mp3|avi|wmv|mpg|mpeg)$
http://www.example.com/images/nohotlink.gif [R,NC,L]
```

Esto dice: “si el visitante no proviene de misitio.net, entonce redirije todos los pedidos de archivos (zip,mp3,avi,wmv,mpg,mpeg) a una imagen que dice “No permitimos hotlinking” De esa forma, puedes redirigir a una página, o lo que desees, o p uedes modificar la lista de extensíones de archivo para incluir/quitar otros archivos. **Cuidado**: cuando se decide bloquera el hotlinking de imágenes recuerda que puedes estar bloqueando **todo** tráfico fuera del alcance de tu dominio. Por ejemplo, si se posee un archivo de sindicación tomado por bloglines necesitarás modificar la regla para permitirles a los lectores obtener las imágenes - o el RSS se verá mal.

Bloqueo de robots maliciosos

En algunos casos se querrá bloquear algunos robots maliciosos, como spiders o descargadores. Para ello utilizaremos mod_rewrite nuevamente. Normalmente los robots maliciosos ignoran el archivo de directivas robots.txt por lo que se podría querer forzar un error 403 cada vez que quieran recorrer o descargar tu sitio:

```
RewriteEngine On
RewriteCond %{HTTP_USER_AGENT} ^BlackWidow [OR]
RewriteCond %{HTTP_USER_AGENT} ^Bot\ mailto:craftbot@yahoo.com [OR]
RewriteCond %{HTTP_USER_AGENT} ^ChinaClaw [OR]
RewriteCond %{HTTP_USER_AGENT} ^Custo [OR]
RewriteCond %{HTTP_USER_AGENT} ^DISCo [OR]
RewriteCond %{HTTP_USER_AGENT} ^Download\ Demon [OR]
RewriteCond %{HTTP_USER_AGENT} ^eCatch [OR]
RewriteCond %{HTTP_USER_AGENT} ^EirGrabber [OR]
RewriteCond %{HTTP_USER_AGENT} ^EmailSiphon [OR]
RewriteCond %{HTTP_USER_AGENT} ^EmailWolf [OR]
RewriteCond %{HTTP_USER_AGENT} ^Express\ WebPictures [OR]
RewriteCond %{HTTP_USER_AGENT} ^ExtractorPro [OR]
RewriteCond %{HTTP_USER_AGENT} ^EyeNetIE [OR]
RewriteCond %{HTTP_USER_AGENT} ^FlashGet [OR]
RewriteCond %{HTTP_USER_AGENT} ^GetRight [OR]
RewriteCond %{HTTP_USER_AGENT} ^GetWeb! [OR]
RewriteCond %{HTTP_USER_AGENT} ^Go!Zilla [OR]
```

```
RewriteCond %{HTTP_USER_AGENT} ^Go-Ahead-Got-It [OR]
RewriteCond %{HTTP_USER_AGENT} ^GrabNet [OR]
RewriteCond %{HTTP_USER_AGENT} ^Grafula [OR]
RewriteCond %{HTTP_USER_AGENT} ^HMView [OR]
RewriteCond %{HTTP_USER_AGENT} HTTrack [NC,OR]
RewriteCond %{HTTP_USER_AGENT} ^Image\ Stripper [OR]
RewriteCond %{HTTP_USER_AGENT} ^Image\ Sucker [OR]
RewriteCond %{HTTP_USER_AGENT} Indy\ Library [NC,OR]
RewriteCond %{HTTP_USER_AGENT} ^InterGET [OR]
RewriteCond %{HTTP_USER_AGENT} ^Internet\ Ninja [OR]
RewriteCond %{HTTP_USER_AGENT} ^JetCar [OR]
RewriteCond %{HTTP_USER_AGENT} ^JOC\ Web\ Spider [OR]
RewriteCond %{HTTP_USER_AGENT} ^larbin [OR]
RewriteCond %{HTTP_USER_AGENT} ^LeechFTP [OR]
RewriteCond %{HTTP_USER_AGENT} ^Mass\ Downloader [OR]
RewriteCond %{HTTP_USER_AGENT} ^MIDown\ tool [OR]
RewriteCond %{HTTP_USER_AGENT} ^Mister\ PiX [OR]
RewriteCond %{HTTP_USER_AGENT} ^Navroad [OR]
RewriteCond %{HTTP_USER_AGENT} ^NearSite [OR]
RewriteCond %{HTTP_USER_AGENT} ^NetAnts [OR]
RewriteCond %{HTTP_USER_AGENT} ^NetSpider [OR]
RewriteCond %{HTTP_USER_AGENT} ^Net\ Vampire [OR]
RewriteCond %{HTTP_USER_AGENT} ^NetZIP [OR]
RewriteCond %{HTTP_USER_AGENT} ^Octopus [OR]
RewriteCond %{HTTP_USER_AGENT} ^Offline\ Explorer [OR]
RewriteCond %{HTTP_USER_AGENT} ^Offline\ Navigator [OR]
RewriteCond %{HTTP_USER_AGENT} ^PageGrabber [OR]
RewriteCond %{HTTP_USER_AGENT} ^Papa\ Foto [OR]
RewriteCond %{HTTP_USER_AGENT} ^pavuk [OR]
RewriteCond %{HTTP_USER_AGENT} ^pcBrowser [OR]
RewriteCond %{HTTP_USER_AGENT} ^RealDownload [OR]
RewriteCond %{HTTP_USER_AGENT} ^ReGet [OR]
RewriteCond %{HTTP_USER_AGENT} ^SiteSnagger [OR]
RewriteCond %{HTTP_USER_AGENT} ^SmartDownload [OR]
RewriteCond %{HTTP_USER_AGENT} ^SuperBot [OR]
RewriteCond %{HTTP_USER_AGENT} ^SuperHTTP [OR]
RewriteCond %{HTTP_USER_AGENT} ^Surfbot [OR]
RewriteCond %{HTTP_USER_AGENT} ^tAkeOut [OR]
RewriteCond %{HTTP_USER_AGENT} ^Teleport\ Pro [OR]
RewriteCond %{HTTP_USER_AGENT} ^VoidEYE [OR]
RewriteCond %{HTTP_USER_AGENT} ^Web\ Image\ Collector [OR]
RewriteCond %{HTTP_USER_AGENT} ^Web\ Sucker [OR]
RewriteCond %{HTTP_USER_AGENT} ^WebAuto [OR]
RewriteCond %{HTTP_USER_AGENT} ^WebCopier [OR]
RewriteCond %{HTTP_USER_AGENT} ^WebFetch [OR]
RewriteCond %{HTTP_USER_AGENT} ^WebGo\ IS [OR]
RewriteCond %{HTTP_USER_AGENT} ^WebLeacher [OR]
RewriteCond %{HTTP_USER_AGENT} ^WebReaper [OR]
RewriteCond %{HTTP_USER_AGENT} ^WebSauger [OR]
RewriteCond %{HTTP_USER_AGENT} ^Website\ eXtractor [OR]
RewriteCond %{HTTP_USER_AGENT} ^Website\ Quester [OR]
RewriteCond %{HTTP_USER_AGENT} ^WebStripper [OR]
RewriteCond %{HTTP_USER_AGENT} ^WebWhacker [OR]
RewriteCond %{HTTP_USER_AGENT} ^WebZIP [OR]
RewriteCond %{HTTP_USER_AGENT} ^Wget [OR]
RewriteCond %{HTTP_USER_AGENT} ^Widow [OR]
RewriteCond %{HTTP_USER_AGENT} ^WWWOFFLE [OR]
RewriteCond %{HTTP_USER_AGENT} ^Xaldon\ WebSpider [OR]
RewriteCond %{HTTP_USER_AGENT} ^Zeus
RewriteRule .* - [F]
```

(Buscar Listas Actualizadas)

No mostrar 'wwww'

Para hacer esto, basta con una simple regla de re-escritura:

```
Options +FollowSymlinks
RewriteEngine on
RewriteCond %{http_host} ^www\.example\.com[nc]
RewriteRule ^(.*)$ http://example.com/$1 [r=301,nc]
```

Escondiendo la extensión del lenguaje de scripting

Se puede aumentar la seguridad cambiando la extensión de los scripts para que los visitantes desconozcan qué lenguaje estás utilizando:

```
# Make PHP code look like unknown types
AddType application/x-httpd-php .cool
```

De esta forma, los archivos .cool serán tratados como si fuesen archivos PHP. Se deben renombrar los archivos que se quiera con esta nueva extensión.

Consejos y trucos varios

- Mantiene el archivo .htaccess pequeño: este archivo es procesado por el servidor web en cada pedido (pudiendo causar problemas de performance).
- Mantiene tu archivo .htaccess organizado. Utiliza comentarios (líneas que comienzan en #) y se lógicamente consistente. Es complicado entender un archivo .htaccess desorganizado una vez que crece lo suficiente.
- Cuando se utilicen reglas de reescritura de URLs, agrega la opción [L] a aquellas páginas finales (como la de hotlinking y demás). Esto le dirá al servidor que no procese más reglas (aumentando la performance).
- Cuidado con la herencia: el archivo .htaccess a nivel raíz es aplicado también en las carpetas, cuaquier regla .htaccess en la carpeta puede reemplazar las reglas de la carpeta raíz.

Protección con contraseña mediante .htpasswd

Esto es útil cuando se quiere agregar una contraseña a ciertas páginas y/o archivos

- Crea un archivo **.htpasswd** en la carpeta a protejer.
- El archivo contendrá la información de registro de la forma **usuario:contraseña**. El nombre de usuario es en texto plano. La contraseña debe de estar encriptada o no funcionará. Utiliza esta herramienta para saber qué texto agregar.
- Si se crea el archivo en la PC local, acuérdate de subirlo al servidor en **modo ASCII**.
- Normalmente, se puede modificar el archivo .htaccess. La autenticación se aplicará a la carpeta en la que se encuentre y las subcarpetas.

  ```
  AuthUserFile /home/pathto/.htpasswd
  AuthType Basic
  AuthName "Mi Carpeta Secreta"


  require valid-user
  ```

  Se puede protejer un sólo archivo incluyendo este código dentro de una directiva

- Asegúrate de proteger el acceso al archivo .htaccess utilizando el primer consejo.

Activando SSI

Utiliza éstas instrucciones para activar la interpretación SSI

```
AddType text/html .html
AddType text/html .shtml
AddHandler server-parsed .html
AddHandler server-parsed .shtml
```

Cambiando la página por defecto

Se puede utilizar esta instrucción para cambiar la página por defecto (el orden es importante)

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

Evitando el error 500

Pasando el juego de caracteres se evita el mostrar un error 500

```
AddDefaultCharset utf-8
```

Directiva CheckSpelling [Control de Ortografía]

Esta directiva puede ser útil para auto-corregir errores de ortografía simple en la URL

```
CheckSpelling On
```

Agregar sumario MD5

Si no se está preocupado por problemas de performance, se peude agregar un cálculo de llave MD5 para agregar un **MIC** [Control de Integridad de Mensaje en inglés] en cada pedido. Esto es útil para controlar la integridad del mensaje.

```
ContentDigest On
```

Para obtener mas información sobre HTACCESS :
http://en.wikipedia.org/wiki/Htaccess

VIGUNU.com