

Tutorial del Servidor Apache HTTP: Ficheros .htaccess

Idiomas disponibles: en | es | fr | ja | ko | pt-br

Los ficheros `.htaccess` facilitan una forma de realizar cambios en la configuración en contexto directorio.

Ficheros .htaccess

Módulos Relacionados

[core](#)
[mod_authn_file](#)
[mod_authz_groupfile](#)
[mod_cgi](#)
[mod_include](#)
[mod_mime](#)

Directivas Relacionadas

[AccessFileName](#)
[AllowOverride](#)
[Options](#)
[AddHandler](#)
[SetHandler](#)
[AuthType](#)
[AuthName](#)
[AuthUserFile](#)
[AuthGroupFile](#)
[Require](#)

- Ficheros .htaccess
- Qué son/Cómo usuarios
- Cuando (no) usar ficheros .htaccess
- How directives are applied
- Ejemplo de Autenticación
- Ejemplo de Server Side Includes
- Reglas de Rewrite en ficheros .htaccess
- Ejemplo de CGI
- Resolución de problemas

Consulte también

- Comentarios

Qué son/Cómo usarlos

Los ficheros `.htaccess` (o "ficheros de configuración distribuida") facilitan una forma de realizar cambios en la configuración en contexto directorio. Un fichero, que contiene una o más directivas, se coloca en un documento específico de un directorio, y estas directivas aplican a ese directorio y todos sus subdirectorios.

Nota:

Si quiere llamar a su fichero `.htaccess` de otra manera, puede cambiar el nombre del fichero usando la directiva `AccessFileName`. Por ejemplo, si usted prefiere llamar al fichero `.config`, entonces puede poner lo siguiente en el fichero de configuración de su servidor:

```
AccessFileName ".config"
```

Generalmente, los ficheros `.htaccess` usan la misma sintaxis que los [ficheros de la configuración principal](#). Lo que puede utilizar en estos ficheros lo determina la directiva `AllowOverride`. Esta directiva especifica, en categorías, qué directivas tendrán efecto si se encuentran en un fichero `.htaccess`. Si se permite una directiva en un fichero `.htaccess`, la documentación para esa directiva contendrá una sección `Override`, especificando qué valor debe ir en `AllowOverride` para que se permita esa directiva.

Por ejemplo, si busca en la documentación la directiva `AddDefaultCharset`, encontrará que se permite en ficheros `.htaccess`. (Vea la línea de Contexto en el sumario de la directiva.) La línea `Override` muestra `FileInfo`. De este modo, debe tener al menos `AllowOverride FileInfo` para que esta directiva se aplique en ficheros `.htaccess`.

Ejemplo:

Context:

server config, virtual host, directory, .htaccess

Override:

FileInfo

Si no está seguro de cuándo, una directiva en concreto, se puede usar en un fichero `.htaccess`, consulte la documentación para esa directiva, y compruebe la línea `Context` buscando `".htaccess"`.

Cuando (no) usar ficheros .htaccess

Generalmente, solo debería usar ficheros `.htaccess` cuando no tiene acceso al fichero principal de configuración del servidor. Hay, por ejemplo, una creencia errónea de que la autenticación de usuario debería hacerse siempre dentro de ficheros `.htaccess`, y, más recientemente, otra creencia errónea de que las directivas de `mod_rewrite` deben ir en ficheros `.htaccess`. Esto sencillamente no es el caso. Puede poner las configuraciones de autenticación de usuario en la configuración principal del servidor, y esto es de hecho, el método preferido de configurar Apache. Del mismo modo, las directivas `mod_rewrite` funcionan mejor, en muchos sentidos, en el fichero de configuración principal del servidor.

Los ficheros `.htaccess` deberían usarse cuando su proveedor de contenidos le permite hacer modificaciones de configuración en contexto directorio, pero usted no tiene acceso de root en el servidor. En el caso de que el administrador no esté dispuesto a hacer cambios frecuentes en la configuración, puede que sea necesario permitir a usuarios individuales realizar estos cambios de configuración en ficheros `.htaccess` por ellos mismos. Lo cual ocurre a menudo, por ejemplo, en casos donde los ISP están albergando múltiples sitios web de usuario en una sola máquina, y quieren que sus usuarios tengan la posibilidad de modificar sus configuraciones.

Aun así, generalmente, el uso de ficheros `.htaccess` debería evitarse cuando sea posible. Cualquier configuración que consideraría poner en un fichero `.htaccess`, puede usarse con la misma efectividad en una sección `<Directory>` en el fichero de configuración del servidor.

Hay dos razones para evitar el uso de ficheros `.htaccess`.

La primera es el rendimiento. Cuando `AllowOverride` está configurado para permitir el uso de ficheros `.htaccess`, httpd buscará ficheros `.htaccess` en cada directorio. Así, permitiendo ficheros `.htaccess` provoca una pérdida de rendimiento, ¡incluso aunque no los use! Además, los ficheros `.htaccess` se cargan cada vez que se solicita un documento.

Además tenga en cuenta que httpd debe buscar ficheros `.htaccess` en todos los directorios de mayor jerarquía, para poder tener la lista completa de directivas que debe aplicar. (Vea la sección sobre [Cómo se aplican las directivas](#).) Así, si se solicita un fichero de un directorio `/www/htdocs/example`, httpd debe buscar los siguientes ficheros:

```
/.htaccess
/www/.htaccess
/www/htdocs/.htaccess
/www/htdocs/example/.htaccess
```

De esta manera, por cada acceso a un fichero de ese directorio, hay 4 accesos adicionales al sistema de ficheros, incluso si ninguno de esos ficheros está presente. (Tenga en cuenta que este caso solo se daría si los ficheros `.htaccess` están activados en `/`, que generalmente no es el caso.).

En el caso de las directivas `RewriteRule`, en el contexto de `.htaccess` estas expresiones regulares deben recompilarse con cada solicitud a ese directorio, cuando en el contexto de configuración del servidor solo se compilan una vez y se cachean. Adicionalmente, las reglas en sí mismas son más complicadas, puesto que uno debe sortear las restricciones que vienen acompañadas del contexto directorio y `mod_rewrite`. Consulte la [Guía de Rewrite](#) para un mayor detalle sobre este tema.

La segunda consideración es de seguridad. Estará permitiendo que usuarios modifiquen la configuración del servidor, lo cual puede dar lugar a cambios sobre los que usted no tendrá ningún control. Medite profundamente si debe dar a sus usuarios ese privilegio. Además tenga en cuenta que dar a los usuarios menos privilegios de los que necesitan dará lugar a más peticiones de soporte. Asegúrese de que le indica a sus usuarios claramente el nivel de privilegios que les está dando. Especificando exactamente cómo ha configurado `AllowOverride`, e invítele a revisar la documentación relacionada, lo cual le ahorrará bastantes confusiones más adelante.

Tenga en cuenta que esto es equivalente por completo a poner un fichero `.htaccess` en un directorio `/www/htdocs/example` con una directiva, y poner la misma directiva en una sección `Directory` `<Directory "/www/htdocs/example">` en su configuración principal del servidor:

Fichero `.htaccess` en `/www/htdocs/example`:

Contenido de fichero .htaccess en /www/htdocs/example

AddType text/example ".exm"

Sección de su fichero httpd.conf

<Directory "/www/htdocs/example">
 AddType text/example ".exm"
</Directory>

Aun así, poniendo ésta en el fichero de configuración dará como resultado una menor pérdida de rendimiento, y como la configuración se carga una vez cuando el httpd arranca, en lugar de cada vez que se solicita un fichero.

El uso de ficheros `.htaccess` puede desactivarse por completo configurando la directiva `AllowOverride` a `none`:

AllowOverride

None

How directives are applied

Las directivas de configuración que se encuentran en el fichero `.htaccess` se aplican al directorio en el que el fichero `.htaccess` se encuentra, y a todos sus subdirectorios. Sin embargo, es importante recordar que puede haber otros ficheros `.htaccess` en directorios previos. Las directivas se aplican en el orden en el que se encuentran. Por lo tanto, un fichero `.htaccess` puede sobrescribir directivas que se encuentran en ficheros `.htaccess` que se encuentran en directorios previos del árbol de directorios. Y estos, en cambio, pueden haber sobrescrito directivas que se encontraban más arriba, o en el fichero principal de configuración del servidor mismo.

Ejemplo:

En el directorio `/www/htdocs/example1` tenemos un fichero `.htaccess` que contiene lo siguiente:

Options

+ExecCGI

(Nota: debe tener `"AllowOverride Options"` configurado para permitir el uso de la directiva `"Options"` en ficheros `.htaccess` files.)

En el directorio `/www/htdocs/example1/example2` tenemos un fichero `.htaccess` que contiene:

Options

Includes

Por este segundo fichero `.htaccess`, en el directorio `/www/htdocs/example1/example2`, la ejecución de CGI `execution` no está permitida, porque solo se ha definido `Options Includes`, que sobrescribe completamente una configuración previa que se pudiera haber definido.

Incorporando el .htaccess en los ficheros de configuración principal

Como se ha comentado en la documentación en las [Secciones de Configuración](#), los ficheros `.htaccess` pueden sobrescribir las secciones `<Directory>` por el directorio correspondiente, pero se sobrescribirán por otros tipos de secciones de configuración de los ficheros de configuración principal. Este hecho se puede usar para forzar ciertas configuraciones, incluso en presencia de una configuración laxa de `AllowOverride`. Por ejemplo, para prevenir la ejecución de un script mientras se permite cualquier otra cosa en `.htaccess` puede usar:

```
<Directory "/www/htdocs">
    AllowOverride All
</Directory>

<Location "/">
    Options +IncludesNoExec -ExecCGI
</Location>
```

Este ejemplo asume que su `DocumentRoot` es `/www/htdocs`.

Ejemplo de Autenticación

Si saltó directamente a esta parte del documento para averiguar como hacer la autenticación, es important que tenga en cuenta una cosa. Hay una creencia errónea de que necesita usar ficheros `.htaccess` para configurar autenticación con contraseña. Este no es el caso. Colocar las directivas de autenticación en una sección `<Directory>`, en su fichero de configuración principal, es el método recomendado para configurar esto, y los ficheros `.htaccess` deberían usarse solamente si no tiene acceso al fichero de configuración principal del servidor. Vea [más arriba](#) una explicación de cuando debería y cuando no debería usar ficheros `.htaccess`.

Dicho esto, si todavía cree que debe usar el fichero `.htaccess`, podrá ver que una configuración como la que sigue podría servirle.

Contenido del fichero `.htaccess`:

```
AuthType Basic
AuthName "Password Required"
AuthUserFile "/www/passwords/password.file"
AuthGroupFile "/www/passwords/group.file"
Require group admins
```

Tenga en cuenta que `AllowOverride AuthConfig` debe estar habilitado para que estas directivas tengan algún efecto.

Por favor vea el [tutorial de autenticación](#) para una explicación más completa de la autenticación y la autorización.

Ejemplo de Server Side Includes

Otro uso común de ficheros `.htaccess` es activar `Server Side Includes` para un directorio en particular. Esto puede hacerse con las siguientes directivas de configuración, colocadas en un fichero `.htaccess` y el directorio deseado:

```
Options +Includes
AddType text/html "shtml"
AddHandler server-parsed shtml
```

Tenga en cuenta que `AllowOverride Options` y `AllowOverride FileInfo` deben estar activadas para que estas directivas tengan efecto.

Por favor vea el [tutorial de SSI](#) para una explicación más completa de server-side includes.

Reglas de Rewrite en ficheros .htaccess

Cuando use `RewriteRule` en ficheros `.htaccess`, tenga en cuenta que el contexto directorio cambia las cosas un poco. En concreto, las reglas son relativas al directorio actual, en lugar de serlo de la petición de URI solicitada originalmente. Considere los siguientes ejemplos:

```
# En httpd.conf
RewriteRule "^/images/(.+)\.jpg" "/images/$1.png"

# En .htaccess en el directorio raíz
RewriteRule "^images/(.+)\.jpg" "images/$1.png"

# En .htaccess en images/
RewriteRule "^(.+)\.jpg" "$1.png"
```

En un `.htaccess` en `images/` (donde el valor `DocumentRoot`, la barra ("/") inicial se elimina del valor facilitado a `RewriteRule`, y en el subdirectorio `images`, se elimina `/images/` también de este valor. Así, su expresión regular necesita omitir también esa parte.

Consulte la [documentación de mod_rewrite](#) para más detalles al usar `mod_rewrite`.

Ejemplo de CGI

Finalmente, puede que quiera usar un fichero `.htaccess` para permitir la ejecución de programas CGI en un directorio en particular. Esto se puede implementar con la siguiente configuración:

```
Options +ExecCGI
AddHandler cgi-script ".cgi" ".pl"
```

Alternativamente, si quiere considerar como programas CGI todos los ficheros de un directorio concreto, esto se puede conseguir con la siguiente configuración:

```
Options +ExecCGI
SetHandler cgi-script
```

Tenga en cuenta que `AllowOverride Options` y `AllowOverride FileInfo` deben estar ambas activadas para que estas directivas tengan efecto.

Por favor vea el [tutorial CGI](#) para mayor detalle sobre programación y configuración de CGI.

Resolución de problemas

Cuando pone directivas en un fichero `.htaccess` y no obtiene el efecto deseado hay una serie de cosas que pueden haber ido mal.

El problema más común es que `AllowOverride` no está configurada para que sus directivas puedan surtir efecto. Asegúrese de que no tiene `AllowOverride None` configurado en el directorio en cuestión. Una buena forma de probar esto es poner "basura" en su fichero `.htaccess` y recargar la página. Si no se genera un error en el servidor, casi seguro que tiene configurado `AllowOverride None`.

Si, por otro lado, obtiene errores de servidor al intentar acceder a documentos, compruebe el log de errores de httpd. Seguramente le indiquen que la directiva en uso en su fichero `.htaccess` no está permitida.

[Fri Sep 17 18:43:16 2010] [alert] [client 192.168.200.51] /var/www/html/.htaccess: DirectoryIndex not allowed here

Esto indicará que o bien ha usado una directiva que no se permite nunca en ficheros `.htaccess`, o que simplemente no tiene `AllowOverride` configurado a un nivel suficiente para la directiva que ha usado. Consulte la documentación para esa directiva en particular para determinar cual es el caso.

Alternativamente, puede que le indique que hay un error de sintaxis en el uso de la propia directiva.

[Sat Aug 09 16:22:34 2008] [alert] [client 192.168.200.51] /var/www/html/.htaccess: RewriteCond: bad flag delimiters

En este caso, el mensaje de error debería ser específico para el error de sintaxis concreto que ha cometido.

Comentarios

Notice:

This is not a Q&A section. Comments placed here should be pointed towards suggestions on improving the documentation or server, and may be removed again by our moderators if they are either implemented or considered invalid/off-topic. Questions on how to manage the Apache HTTP Server should be directed at either our IRC channel, #httpd, on Freenode, or sent to our [mailing lists](#).