

Apuntes labo .htaccess – 24/05/2018

Un fichero .htaccess (hypertext access), también conocido como archivo de configuración distribuida, es un fichero especial, popularizado por el Servidor HTTP Apache que permite definir diferentes directivas de configuración para cada directorio (con sus respectivos subdirectorios) sin necesidad de editar el archivo de configuración principal de Apache.

Ahora bien, los permisos de configuración dependen de los establecidos dentro del archivo de configuración de Apache mediante la directiva AllowOverride. Podemos permitir las todas, denegar todo o permitir solo ciertas opciones.

```
# No permitir ninguna directiva
AllowOverride None
```

```
# Permitir todo
AllowOverride All
```

Las directivas de un archivo .htaccess son aplicadas a los subdirectorios, pero si un directorio tiene otro archivo de configuración, dependiendo de estas directivas puede causarse que las segundas sobrescriban las de la carpeta superior.

¿Cómo y cuándo usar un archivo .htaccess?

Se nos recomienda que el uso de archivos .htaccess sea exclusivo en casos donde no se tiene acceso al archivo de configuración principal de Apache. Esto seguramente será porque hasta cierto punto es un riesgo de seguridad el tener archivos de configuración junto con las carpetas en las que navegan los usuarios dentro de nuestro sitio. Ahora, pasando al tema de cómo utilizarlo; decir que basta con crear un archivo con ese nombre, colocar las directivas dentro del mismo y subirlo al servidor. Con eso, cada vez que Apache detecte la presencia de este archivo, ejecutará su contenido.

Pero es importante tener en cuenta el contexto de la directiva de Apache, que determina dónde se puede utilizar: server config | virtual host | directory | .htaccess. Mostrarlo en la documentación de Apache, buscando por ejemplo

- ServerName: <http://httpd.apache.org/docs/current/mod/core.html#servername>
- ErrorDocument: <http://httpd.apache.org/docs/current/mod/core.html#errordocument>

Formato

Por razones históricas, el formato de .htaccess es el mismo que el fichero de configuración global de Apache, incluso en otros servidores web como Sun Java System Web Server y Zeus Web Server, que cuentan con ficheros de configuración globales muy diferentes.

Usos frecuentes

El fichero .htaccess ofrece un universo de posibilidades, detallamos a continuación los usos más frecuentes.

• *Listado de carpetas*

Se podría prevenir el listado de carpetas de esta forma, aunque muestra que la carpeta existe...

```
IndexIgnore *
```

Podemos aprovechar a ocultar la información del servidor:

```
ServerSignature Off
```

Si se quiere prevenir que se listen los índices de las carpetas navegables, entonces creamos un archivo .htaccess en la raíz del directorio y agregamos esta línea:

```
Options -Indexes
```

- ***Cambiar DirectoryIndex***

```
DirectoryIndex aplicacion.php
```

- ***PROBLEMAS DE ENCODING - Cambio de configuraciones de PHP***

El código fuente de nuestro proyecto está en UTF8, mientras que PHP está configurado

```
default_charset="iso-8859-1"
```

Cuando no tenemos acceso al PHP.ini, podemos modificar algunas variables, ejemplo:

```
php_flag display_errors Off

#formato general
php_value|php_flag setting_name setting_value
```

Para resolver el problema del encoding, agregamos:

```
php_value default_charset "utf8"
```

- ***Control de acceso a carpetas***

Se podría 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 accederán a ellos mediante el sistema de archivos, pero no se podrán acceder via web.

Crear un archivo .htaccess en esa carpeta que contenga:

```
deny from 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 /autocompletar_ht/errores/404.html
```

Esto redirige el usuario cada vez que sucede un error 404. Obviamente, se puede redefinir para que capture otros errores http (403, 500, etc.).

- ***Información del entorno del servidor y de ejecución***

\$_SERVER -> Información del entorno del servidor y de ejecución. Las entradas en este array son creadas por el servidor web.

```
SetEnv mail_admin "admin@cs.uns.edu.ar"
```

Con eso se muestra el mail al final del html de aplicacion.php gracias a este código:

```
<?php
if (isset($_SERVER['mail_admin'])) {
    echo "<br/>";
    echo "<span>Mail del Administrador: </span>" .
        $_SERVER['mail_admin'];
}
?>
```

- ***Cabeceras personalizadas***

Prevenir el cacheo de archivos CSS y JavaScript

```
<FilesMatch "\.(js|css)$">
    FileETag None
    Header unset ETag
    Header set Cache-Control "max-age=0, no-cache, no-store, private"
    Header set Pragma "no-cache"
    Header set Expires "0"
</FilesMatch>
```

Borrar la caché, mostrar el tráfico de red.

- ***Cambio de configuraciones de PHP***

Cuando no tenemos acceso al PHP.ini, podemos modificar algunas variables, ejemplo:

```
php_flag display_errors Off

#formato general
php_value|php_flag setting_name setting_value
```

Agregar en la sección <?php ... ?> lo siguiente para que cause un warning:

```
split(',', 'a,b,c');
```

- ***Protección con contraseña mediante .htpasswd***

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

- Crear un archivo .htpasswd en la carpeta a proteger.
- 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. Pueden usar: <http://www.htaccesstools.com/htpasswd-generator/>
- Normalmente, se puede modificar el archivo .htaccess. La autenticación se aplicará a la carpeta en la que se encuentre y las subcarpetas.
- En lo posible, que .htpasswd no esté en un directorio navegable

```
#control de acceso
AuthName "Directorio privado"
```

```
AuthType Basic
AuthUserFile c:/acceso/autocompletar_ht/.htpasswd
Require valid-user
```

Consejos y trucos varios

- Mantener el archivo .htaccess pequeño: este archivo es procesado por el servidor web en cada pedido (pudiendo causar problemas de performance).
- Mantener el archivo .htaccess organizado. Utilizar comentarios (líneas que comienzan en #) y ser lógicamente consistente. Es complicado entender un archivo .htaccess desorganizado una vez que crece lo suficiente.
- Cuidado con la herencia: el archivo .htaccess a nivel raíz es aplicado también en las carpetas, cualquier regla .htaccess en la carpeta puede reemplazar las reglas de la carpeta raíz.