Tutoriales programación

Tutoriales de programación en PHP y Android

domingo, 30 de junio de 2013

Reglas de reescritura - Parte 2: Directivas

RewriteEngine

Lo primero que tenemos que hacer para empezar a escribir reglas de reescritura es indicar a Apache que vamos a utilizar el módulo mod_rewrite . Y para ello utilizaremos la siguiente directiva en el archivo .htaccess

```
RewriteEngine on
```

Para desactivarlo, y así el servidor ignore el resto de la configuración, sólo tendríamos que poner

```
 RewriteEngine off
```

RewriteBase

Esta directiva se utiliza para indicar la URL base para la reescritura.

```
RewriteBase /
```

Esta directiva no es obligatoria pero si que es útil. Por defecto la ruta al contenido que usamos para la reescritura es relativa al directorio raíz del host (por ejemplo public_html/). Por lo que con esta directiva podemos especificar el directorio (ruta) base para las reglas.

Por ejemplo queremos que cuando se introduzca la URL dominio.com/index.php seamos redireccionados automáticamente a dominio.com/directorio/index2.php. Así que si no especificamos una RewriteBase, la regla de reescritura (se verán a continuación) tendrá que especificarse como:

```
RewriteRule ^index\.php$ /directorio/index2.php [R,L].
```

Sin embargo utilizando la base correcta podremos prescindir del /folder en la regla reescritura.

```
RewriteEngine On
RewriteBase /directorio/
RewriteCond %{HTTP_REFERER} dominio\.com
RewriteRule ^index\.php$ index2.php [R,L]
```

RewriteCond

La directiva RewriteCond nos permite especificar una condición que si se cumple se ejecuta la regla RewriteRule posterior. Se pueden poner varias condiciones con RewriteCond. En este caso, cuando se cumplen todas las condiciones (a no ser que se indique otra cosa) se ejecuta la directiva RewriteRule posterior.

La sintaxis es:

```
RewriteCond variable_apache expresión_regular banderas
```

Aquí tenemos una lista de las principales variables que utiliza el módulo para crear condiciones. Como vemos la mayoría de estas condiciones tienen su equivalencia en PHP

Cabeceras HTTP	Conexión y petición	
HTTP_USER_AGENT	REMOTE_ADDR	
HTTP_REFERER	REMOTE_HOST	
HTTP_COOKIE	REMOTE_PORT	
HTTP_FORWARDED	REMOTE_USER	

HTTP_HOST HTTP_PROXY_CONNECTION HTTP_ACCEPT	REMOTE_IDENT REQUEST_METHOD SCRIPT_FILENAME PATH_INFO QUERY_STRING AUTH_TYPE	
Información servidor	**Fecha y hora**	**Especiales**
DOCUMENT_ROOT SERVER_ADMIN SERVER_NAME SERVER_ADDR SERVER_PORT SERVER_PROTOCOL SERVER_SOFTWARE	TIME_YEAR TIME_MON TIME_DAY TIME_HOUR TIME_MIN TIME_SEC TIME_WDAY TIME	API_VERSION THE_REQUEST REQUEST_URI REQUEST_FILENAME IS_SUBREQ HTTPS REQUEST_SCHEME

A continuación se presenta la lista de las banderas disponibles. Estas servirán para controlar el comportamiento de la directiva en la que sea utilizada.

- R: (redirect) para forzar una redirección HTTP.
- F: (forbidden) para prohibir el acceso.
- G: (gone) para eliminar la URL.
- P: (proxy) para pasar la URL a mod_proxy.
- L: (last) para detener el procesamiento. No se ejecuta nada que haya detrás.
- N: (next) para continuar.
- C: (chain) para encadenar la regla activa con la siguiente.
- F: (forbbiden) devuelve una respuesta de error 403 y para el proceso.
- NS: (nosubreq) para segurarse que la regla sólo se aplica si no se realizan subpeticiones internas.
- NC: (nocase) para que la URL no distinga mayúsculas de minúsculas.
- QSA: (qsappend) para añadir una nueva cadena de consulta (query string) en lugar de sustituirla.
- PT: (passthrough) para asar la URL modificada a otro módulo apache.
- S: (skip) para saltar la siguiente regla.
- E: (env) para asignar una variable de entorno.
- OR: para indicar que para ejecutar la regla de reescritura se tiene que cumplir la condición actual o la siguiente.

Vamos con un ejemplo:

```
RewriteCond %{HTTP_USER_AGENT} ^ejemplo    [OR,NC]
RewriteCond %{HTTP_USER_AGENT} ^google     [NC]
```

- La variable_apache es una variable conocida por el servidor, por ejemplo el nombre del navegador sería %{HTTP_USER_AGENT}.
- La expresión regular se utiliza para determinar cual es el valor de la variable que estamos analizando, por ejemplo si ponemos ^ejemplo estaríamos indicando que el nombre del navegador comienza por ejemplo.
- Para las banderas, el valor NC indica (no case) que no distinga mayúsculas de minúsculas.

Es importante conocer que un signo de exclamación a principio de una expresión regular significa que negamos dicha expresión.

Finalmente vamos a presentar los diferentes "tests" que se pueden hacer con directorios o ficheros. Dichos tests se pueden utilizar en las directivas de condición en el lugar que hasta ahora colocábamos una expresión regular.

- '-d': Trata la cadena como un ruta y comprueba si existe o no y si es un directorio.
- '-f': Trata la cadena como una ruta y comprueba si existe o no y si es un fichero regular.
- '-F': Comprueba si la cadena es una fichero válido accesible vía todos los mecanismo de control (actualmente configurados) de acceso para esa ruta. Hace uso de una subpetición para la comprobación, lo q puede afectar al rendimiento.
- '-H' , -L: Mirar -l.
- '-l': Trata la cadena como una ruta y comprueba si existe o no y si es un enlace simbólico. Se puede usar también la convención Bash -L or -h.
- '-s': Trata la cadena como una ruta y comprueba si existe o no y si es un fichero regular con un tamaño mayor que cero.
- '-U': Comprueba si la cadena es una URL válida accesible vía todos los mecanismo de control (actualmente configurados) de acceso para esa ruta. Hace uso de una subpetición para la comprobación, lo q puede afectar al rendimiento.
- '-x': Trata la cadena como una ruta y comprueba si existe o y si tiene los permisos de ejecución.

En la siguiente sección veremos un ejemplo completo.

RewriteRule

Finalmente tenemos la directiva que efectuará la redirección (reescritura) correspondiente.

La sintaxis es la siguiente

```
RewriteRule expresión_regular redirección banderas
```

Mediante el carácter "-" indicamos que no queremos ninguna página de redirección.

Un tema importante dentro de las reglas de reescritura son las referencias. Esto significa que podemos asignar contenido que se encuentre entre paréntesis en una expresión regular.
Este contenido lo asignaremos a 'variables' del tipo $1, $2... si la expresión regular se encuentra en la directiva RewriteRule o a 'varibles' %1,%2... si estamos en una directiva de tipo RewriteCond.

Nota: Aunque no haya ningún paréntesis en la expresión regular, mediante $1 o %1 (dependiendo de la directiva) nos referiríamos a todo el conjunto que se corresponda con dicha expresión.

Y para comprender dicha directiva vamos a ver unos cuantos ejemplos de utilización:

- Ejemplo 1: Si queremos que todas las páginas terminadas en .bak no se redirijan y se muestre la página de error 403 (prohibido el acceso).

```
RewriteEngine On
RewriteRule ^.*\.bak$ - [F]
```

- Ejemplo 2: Si tenemos un script llamado operacion.php y queremos que exteriormente sea accedido como operacion.html.

```
RewriteEngine On
RewriteRule ^operacion\.html$ operacion.php
```

Ahora se podrá acceder a http://localhost/operacion.html?op=suma&op1=6 y gracias a la regla de reescritura el servidor la interpretará como http://localhost/operacion.php?op=suma&op1=6.

- Ejemplo 3: Si queremos usar extensión do, para las páginas del servidor, en lugar de html podríamos utilizar la siguiente regla.

```
RewriteEngine On
RewriteRule ^(.+)\.do$ $1.html [NC]
```

Recuerda que podemos usar referencias. Por lo tanto tomamos todo lo que va delante de .do (único paréntesis) y lo utilizaremos ($1) junto a la terminación .html para la redirección. Así el servidor transformará internamente www.loquesea.com/pagina.do a www.loquesea.com/pagina.html. Que será lo que entenderá y procesará.

Sin embargo existe un problema con esta regla y es que tal como está tendremos un problema SEO ya que únicamente estamos haciendo una redirección interna y no de URL (HTTP). Por lo que tenemos dos URL's para acceder al mismo recurso (www.loquesea.com/pagina.do y www.loquesea.com/pagina.html). Lo que puede ser penalizado por los motores de búsqueda.

La solución es simple. Añadir la bandera [R] para obligar el redireccionamiento.

```
RewriteEngine On
 RewriteRule ^(.+)\.do$ $1.html [R,NC]
```

Ahora si escribimos en el navegador www.loquesea.com/pagina.do, este nos redireccionará a la URL www.loquesea.com/pagina.html.

- Ejemplo 4: Supongamos que hemos cambiado el nombre de un directorio en el servidor pero sin embargo queremos que sigan funcionando las peticiones que llegan sin que se produzca en mensaje 404 no encontrado.

```
RewriteEngine On
RewriteRule ^(.*)/antiguo/(.*)$ /nuevo/$2/$1 [R,L]
```

Las banderas, expuestas en la tabla anterior, indican un forzado de redirección [R] y que no se ejecutará ninguna regla de reescritura posterior a la actual [L].

- Ejemplo 5: Supongamos que dentro de nuestro directorio raíz tenemos una carpeta /busqueda con un fichero buscar.php. Este fichero me permite obtener la página de búsqueda de Google con el parámetro dado, de esta forma: http://localhost/busqueda/buscar.php?id=hola. Sin embargo nos gustaría poder crear una URL más corta que haga lo mismo:

```
RewriteEngine On
RewriteRule ^buscar busqueda/buscar.php   [NC]
```

De esta forma accederíamos por medio de la URL http://localhost/buscar?id=hola. En este caso no necesitamos una redirección HTTP (bandera [R]).

-Ejemplo 6: Vamos a ver un ejemplo en el que usemos los anteriores tests. Queremos cachear miniaturas de imágenes y si no existen (las miniaturas) las crearemos automáticamente. Imaginemos que para esto último tenemos el script thumb.php.

```
RewriteEngine On
RewriteCond %{REQUEST_FILENAME} !-f
RewriteCond %{REQUEST_FILENAME} -([0-9]+)x([0-9]+)\.(jpe?g|gif|png)$ [NC]
RewriteRule ^(.*)-(.+)x(.+)\.(.{3,4})$ /thumb.php?file=$1.$4&ancho=$2&alto=$3
```

La primera directiva comprueba si el fichero no existe, si no es así continúa.
La segunda directiva comprueba si el final del nombre del fichero pedido termina en guión, seguido por dimensiones y una extensión. Por ejemplo podría ser ejemplo-256x128.jpg. Aceptaría la condición y pasaríamos a la regla de reescritura.
Como podemos ver, el patrón de la regla de reescritura tiene 4 paréntesis por lo que utilizaremos 4 referencias. Redireccionaremos al script thumb.php pasando las referencias anteriores:
- $1 será la cadena anterior al guión. El nombre de la imagen (ejemplo).
- $2 será el valor que utilizaremos para el ancho de la miniatura (256).
- $3 será el valor que utilizaremos para el alto de la miniatura (128).
- $4 será el valor que utilizaremos para el tipo de imagen que generaremos (jpg).

Entradas relacionadas

Reglas de reescritura - Parte 1: Introducción
Reglas de reescritura - Parte 3: Ejemplos completos

Publicado por Iván Posilio Gellida en domingo, junio 30, 2013
Etiquetas: apache, reescritura URL

No hay comentarios:

Publicar un comentario

Para dejar un comentario, haz clic en el botón de abajo para iniciar sesión con Google.

INICIAR SESIÓN CON GOOGLE

Entrada más reciente Inicio Entrada antigua

Suscribirse a: Enviar comentarios (Atom)