

Módulo Cuarto

Conceptos básicos de Apache: mod_rewrite

Técnicas Avanzadas Web

Modulo 4: Apache

- Desarrollado inicialmente por Rob McCool (NCSA)
- Es desde 1996 el servidor web HTTP más utilizado en Internet.
- Es multiplataforma (Unix, Microsoft, Novell)
- La versión 2 de apache se ha reescrito completamente.
- Es extremadamente estable.
- Modularizado para evitar mantener un código extremadamente complejo y monolítico.
- Es software libre aunque no GPL

Técnicas Avanzadas Web

Módulo 4: Apache

- Apache es un programa modular que permite elegir que funcionalidades se incluyen.
- Las funcionalidades adicionales se incluyen mediante la selección de módulos.
- Los módulos pueden seleccionarse:
 - Compilados de manera estática (httpd)
 - Como Objetos Dinámicos Compartidos (DSO)

Compilados de manera estática

- Incluidos en la distribución de apache
 - Apache 1.3
 - En tiempo de compilación en el archivo:
`<fuente_apache>/src/Configuration.tpl`
 - En tiempo de compilación en `./configure`
`./configure --enable-module=<modulo>`

Módulos cargados externamente:

- Apache permite cargar módulos independientes al archivo binario `httpd`.
- Es posible gracias al módulo `mod_so`, que debe compilarse estáticamente en el núcleo de Apache y a la herramienta `apxs` (APache eXtenSion).
- Apache necesita ser compilado previamente para poder utilizar la herramienta `apxs`.
- Para usar este mecanismo el sistema también debe soportar DSO.

- La configuración de Apache se realiza a través de varios ficheros de texto que son autoexplicativos y sencillos de configurar
- El archivo de configuración se compone de **directrices**, que serán utilizadas en el contexto correspondiente.
- El contexto indica en qué parte del servidor se puede usar una directiva.

Contexto de Configuración de Servidor

- Pueden aparecer en cualquier fichero, pero no dentro de cualquier contenedor o archivo de configuración por directorio.

- Ejemplo:

```
ServerRoot "C:/AppServ/Apache"  
Port 80
```

Contexto de Directorio o Contenedores

- Tipos de contenedores:

- `<VirtualHost>`
- `<Directory>`
- `<Files>`
- `<Limit>`

- Ejemplo:

```
<Files ~ "^\.ht">  
    Order allow,deny  
    Deny from all  
    Satisfy All  
</Files>
```

Contexto de .htaccess

- Es posible incluir directrices dentro de los ficheros de configuración por directorio (`.htaccess`)
 - Se puede cambiar el nombre del archivo a través de la directiva `AccessFileName`
- Es necesario que esté activada la opción de sobrescritura (`AllowOverride`) para que se procesen esas directivas.

Contexto de .htaccess

- Valores de la directiva `AllowOverride`:
 - `None`: Desabilita el uso de `.htaccess`
 - `AuthConfig`: Uso de directivas de autenticación
 - `FileInfo`: Uso de directivas de control de tipos de documentos
 - `Indexes`: El uso de directivas de control de indexación
 - `Limit`: El uso de directivas de control de acceso
 - `Options`: El uso de directivas de características de directorio
 - `All`: Se permite usar todas las directivas aplicables.

Contexto de .htaccess

- En general los archivos .htaccess no se deben usar nunca, a menos que no se tenga acceso al archivo de configuración del servidor (Ej: ISP)
- Se debe usar el contenedor `<Directory>`
- Razones:
 - Eficiencia: Apache debe buscar ficheros .htaccess por todo el arbol de directorios, en busca de herencias
 - Seguridad: Se permite a los usuarios hacer cambios sobre las directivas del servidor. Cambios sobre los que no se tiene control.

Contexto de .htaccess

- Ambos ejemplo son equivalentes. Sin embargo es más recomendable usar el primero:

- httpd.conf

```
<Directory /var/www>  
    AddType text/example .exm  
</Directory>
```

- .htaccess (en /var/www/)

```
AddType text/example .exm
```

- Además sería necesario que el directorio /var/www o algún padre incluyese (como mínimo):

```
<Directory /var/www>  
    AllowOverride FileInfo  
</Directory>
```

- Ejemplo de módulo externo: **mod_rewrite**
 - El modulo de re-escritura en apache, permite a Apache reescribir una URL en base a unas reglas determinadas, antes de procesar la petición.

URL Peticioo:

<http://www.server.com/productos/10.html>

URL Procesada:

<http://www.server.com/productos.php?id=10>

- Instalacion
 - Editar el fichero de configuración de apache httpd.conf:

```
LoadModule rewrite_module  
modules/mod_rewrite.so
```

- Directivas de configuración:
 - **RewriteEngine** on | off
 - Activa / desactiva el motor de re-escritura en el contenedor donde es indicado.
 - **RewriteRule** Patrón Re-escritura
 - Regla que indica al apache en función a que patrón de URL, tiene que reescribir la petición.
 - Soporta expresiones regulares.
 - **RewriteCond** CadenaEvaluada Patron
 - En función a su resultado (true|false), ejecutará la regla que le suceda.
 - La cadena evaluada puede tomar su valor de determinadas variables de entorno a las que apache nos da acceso.

- Variables de Apache más importantes..
 - REQUEST_URI
 - HTTP_USER_AGENT
 - HTTP_REFERER
 - HTTP_COOKIE
 - HTTP_FORWARDED
 - HTTP_HOST
 - HTTP_ACCEPT
 - REMOTE_ADDR
 - QUERY_STRING

- Modificadores
 - flags que se insertan al final de un RewriteRule, para modificar su comportamiento:
 - **[QSA]**: Si está presente, se mantendrán los valores query_string que “traiga” la petición, añadiéndose los que indiquemos (o no), en nuestra regla.
 - **[L]**: Parará el proceso de reescritura después de aplicar la regla donde se encuentre.
 - **[T=application/pdf]**: fuerza a modificar el mime-type de la petición cursada.
 - **[E=variable:valor]**: Setea una variable de entorno, accesible más tarde desde la página dinámica a la que se esté accediendo.

- Modo de uso:
 - Las directivas `mod_rewrite`, deberán ir incluidas bien en un contexto de directorio, o bien en un fichero `.htaccess`.
 - Es importante tener en cuenta, que estas directivas, tendrán influencia en el directorio donde se sitúe el `.htaccess` o al que nos refiramos dentro del contexto directorio del `httpd.conf`, así como en sus directorio hijos, si éstos no tuvieran otras directivas `mod_rewrite`.

- Ejemplo de uso:

```
# Si el REQUEST_URI empieza con /es/ o /en/ o /ru/  
RewriteCond %{REQUEST_URI} /(es|en|ru)/.*
```

```
#Re-escribo la url por con la variable de idioma  
pasada como argumento:
```

```
RewriteRule /(.*)/(.*) /$2?idioma=$1 [QSA]
```

Técnicas Avanzadas Web

Módulo 4: Apache

- Enlaces interesantes
 - <http://www.google.com>
 - http://httpd.apache.org/docs/1.3/mod/mod_rewrite.html
 - <http://httpd.apache.org/docs/misc/rewriteguide.html>
 - http://www.irontec.com/~jabi/mod4_apache.pdf