

Configuración y uso de Server Side Includes (SSI) en Linux

Paco Brufal, <pbrufal@linuxfreak.com>

Versión 0.1, Octubre 1999

Este pequeño documento explica de manera sencilla cómo configurar y usar las instrucciones Server Side Includes (SSI) para crear documentos HTML de forma dinámica y sencilla. Así mismo, este documento se distribuye SIN NINGUNA GARANTIA bajo licencia GPL. No hace falta decir que se admiten sugerencias, ejemplos, correcciones, etc. para posteriores revisiones.

1. Introducción.

El "lenguaje" SSI (Server Side Includes) son una serie de "tags" o etiquetas con un formato determinado, que se incluyen dentro del código HTML de una página, dándole dinamismo y ahorrándonos algo de trabajo. El formato de los tags tiene esta forma:

```
[codigo HTML...]  
Hora local: <!--#echo var="$DATE_LOCAL"-->  
[codigo HTML...]
```

El servidor, al enviar esa orden, la sustituiría por la variable \$DATE_LOCAL, quedando más o menos así:

```
Hora local: Tuesday, 14-Sep-1999 09:35:23 CEST
```

El servidor web, antes de enviar éstas páginas, sustituye las etiquetas SSI por sus valores correspondientes, es decir, crea las páginas "on-the-fly" y las envía al navegador. No voy a extenderme más, puesto que el SSI no tiene demasiada dificultad de aprendizaje. Si deseas saber más sobre SSI, visita <http://www.apache.org>.

2. Configuración de Apache

Vamos a configurar el servidor Apache. La distribución de Linux que he usado yo es Debian 2.1 (Slink), aunque no creo que difiera mucho de otras distribuciones. Revisa los ficheros de configuración de tu servidor y familiarízate con ellos. Si ves que algo de los que pongo no lo tienes en tus ficheros, añádelo. En caso que lo tengas, pero comentado, descomentalo. Lo primero que necesitamos es decirle que acepte etiquetas SSI descomentando la siguiente línea en la sección de los LoadModule, en el fichero /etc/apache/httpd.conf:

```
LoadModule includes_module /usr/lib/apache/1.3/mod_include.so
```

Para que el servidor reconozca la extensión .shtml (la que llevan las páginas que incluyen SSI), añadiremos esto al fichero /etc/apache/srm.conf:

```
# To use server-parsed HTML files
AddType text/html .shtml
AddHandler server-parsed .shtml
```

También podemos poner la opción "XBitHack". Poniendo esta opción, y haciendo ejecutables los ficheros HTML, el servidor comprobará si esos fichero tienen código SSI.

Ahora vamos al fichero /etc/apache/access.conf. Miramos dentro del bloque "<Directory /var/www>" (o lo que es lo mismo, el DocumentRoot, donde se alojan las páginas del servidor) y buscamos una línea que empiece por "Options", que estará al principio del bloque. Bien, pues si esa línea es algo parecido a esto:

```
Options Indexes FollowSymLinks
```

la dejaremos así

```
Options Indexes FollowSymLinks +Includes
```

Con esto ya podemos incluir etiquetas SSI en nuestras páginas web. Para hacer este documento un poco más útil ;) voy a poner unos ejemplos de uso del SSI:

1. Datos de la conexión
2. Restringir el acceso a determinadas partes del servidor
3. Personalizar las páginas de error

2.1 Datos de la conexión

Este ejemplo servirá para familiarizarnos con SSI. Se trata de una simple página en la que pondremos todos los tags SSI que queramos, y al verla desde un navegador, el servidor sustituirá cada variable por su valor correspondiente, así sabremos qué es y para que sirve cada variable.

```
DATE_GMT <!--#echo var="DATE_GMT" --> <br>
DATE_LOCAL <!--#echo var="DATE_LOCAL" --> <br>
DOCUMENT_NAME <!--#echo var="DOCUMENT_NAME" --> <br>
DOCUMENT_PATH_INFO <!--#echo var="DOCUMENT_PATH_INFO" --> <br>
DOCUMENT_ROOT <!--#echo var="DOCUMENT_ROOT" --> <br>
DOCUMENT_URI <!--#echo var="DOCUMENT_URI" --> <br>
LAST_MODIFIED <!--#echo var="LAST_MODIFIED" --> <br>
GATEWAY_INTERFACE <!--#echo var="GATEWAY_INTERFACE" --> <br>
HTTP_ACCEPT <!--#echo var="HTTP_ACCEPT" --> <br>
HTTP_ACCEPT_CHARSET <!--#echo var="HTTP_ACCEPT_CHARSET" --> <br>
HTTP_ACCEPT_ENCODING <!--#echo var="HTTP_ACCEPT_ENCODING" --> <br>
HTTP_ACCEPT_LANGUAGE <!--#echo var="HTTP_ACCEPT_LANGUAGE" --> <br>
HTTP_CONNECTION <!--#echo var="HTTP_CONNECTION" --> <br>
HTTP_COOKIE <!--#echo var="HTTP_COOKIE" --> <br>
HTTP_HOST <!--#echo var="HTTP_HOST" --> <br>
HTTP_IF_MODIFIED_SINCE <!--#echo var="HTTP_IF_MODIFIED_SINCE" --> <br>
HTTP_PRAGMA <!--#echo var="HTTP_PRAGMA" --> <br>
HTTP_USER_AGENT <!--#echo var="HTTP_USER_AGENT" --> <br>
PATH <!--#echo var="PATH" --> <br>
QUERY_STRING <!--#echo var="QUERY_STRING" --> <br>
REMOTE_ADDR <!--#echo var="REMOTE_ADDR" --> <br>
REMOTE_HOST <!--#echo var="REMOTE_HOST" --> <br>
REMOTE_PORT <!--#echo var="REMOTE_PORT" --> <br>
REQUEST_METHOD <!--#echo var="REQUEST_METHOD" --> <br>
SCRIPT_FILENAME <!--#echo var="SCRIPT_FILENAME" --> <br>
SERVER_ADMIN <!--#echo var="SERVER_ADMIN" --> <br>
SERVER_NAME <!--#echo var="SERVER_NAME" --> <br>
SERVER_PORT <!--#echo var="SERVER_PORT" --> <br>
```

```
SERVER_PROTOCOL <!--#echo var="SERVER_PROTOCOL" --> <br>
SERVER_SIGNATURE <!--#echo var="SERVER_SIGNATURE" --> <br>
SERVER_SOFTWARE <!--#echo var="SERVER_SOFTWARE" --> <br>
SCRIPT_NAME <!--#echo var="SCRIPT_NAME" --> <br>
USER_NAME <!--#echo var="USER_NAME" --> <br>
```

2.2 Restringir el acceso a determinadas partes del servidor

Con este pequeño trozo de código podemos ocultar un enlace según desde donde venga la petición de la página. Es decir, si alguien de nuestra red local solicita una página, el servidor comprobará la IP, y si cumple con la regla SSI, mostrará el enlace en la página web. En caso contrario, no lo mostrará.

El código sería algo así:

```
[...resto de la página...]
<!--#if expr="$REMOTE_ADDR = /^192.168.1./" -->
  <a href="boletin_interno.html">Boletin Interno de la Empresa</a>
<!--#endif -->
[...resto de la página...]
```

Si la IP que solicita la página está en el rango 192.168.1.xxx, podrá ver el enlace. En cambio, si la IP 212.23.125.56 (p. ej.) solicita la misma página, no verá el enlace.

2.3 Personalizar las páginas de error.

Con este ejemplo, lo que pretendemos es que cada vez que se produzca un error, el servidor nos diga qué clase de error se ha producido de una manera que cualquier persona lo pueda entender, y no con el típico error "Error 404" y esas cosas. Además, las haremos en varios idiomas, para que cada persona que vea un error, lo pueda leer según con el idioma preferente que tenga definido en el navegador.

Lo primero que haremos será añadir este bloque al final del fichero `/etc/apache/srm.conf`

```
# Asignamos un alias a la ruta /var/errordocs/,
# es decir, que el path /var/errordocs/ será
# la URL http://localhost/errordocs
# (al igual que /usr/lib/cgi-bin/ es http://localhost/cgi-bin/...)
Alias /errordocs /var/errordocs

# Definimos las propiedades de ese directorio
<Directory /var/errordocs>
  AllowOverride none
  Options MultiViews IncludesNoExec FollowSymLinks
  AddType text/html .shtml
  AddHandler server-parsed .shtml
</Directory>

# Asignamos un documento por cada error (puedes poner tantos como errores
# existan).
ErrorDocument 403 /errordocs/403.shtml
ErrorDocument 404 /errordocs/404.shtml
ErrorDocument 500 /errordocs/500.shtml
ErrorDocument 400 /errordocs/400.shtml
```

Ahora vamos a crear las páginas de error, que se alojarán en el directorio `/var/errordocs`. Los nombres de los ficheros han de ser los que definimos en el fichero `/etc/apache/srm.conf`. El diseño de la página es común, lo único que cambia es el cuerpo y el pie de página. Para elegir el idioma del mensaje, usaremos algo así:

```
<!--#if expr="$HTTP_ACCEPT_LANGUAGE = /^es-ES/" -->
  <!--#include virtual="404_es.html" -->
<!--#endif -->
<!--#if expr="$HTTP_ACCEPT_LANGUAGE = /^en/" -->
  <!--#include virtual="404_en.html" -->
<!--#endif -->
[...todas las condiciones que desees...]
```

Al final de cada página de error, incluiremos la orden

```
<!--#include virtual="foot.shtml" -->
```

para que en el pie de página aparezcan una serie de datos variables y comunes a todas las páginas de error. Con esto lo que nos ahorramos es trabajo, ya que lo escribimos una vez, y lo podemos usar en todos las páginas de error. Aquí pongo un pequeño fichero de ejemplo, concretamente el fichero /var/errordocs/404.shtml, que saldrá cuando alguien pida una página que no existe:

```
<!doctype html public "-//w3c//dtd html 4.0 transitional//en">
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
  <meta name="Author" content="Paco Brufal">
  <title>Error!</title>
</head>

<body text="#000000" bgcolor="#000000" link="#0000EF" vlink="#51188E"
alink="#FF0000">

  <center>

    <!-- Incluimos una imagen de error -->
    
    <hr WIDTH="100%">

    <!-- Condiciones del idioma -->
    <!--#if expr="$HTTP_ACCEPT_LANGUAGE = /^es-ES/" -->
      <!--#include virtual="404_es.html" -->
    <!--#endif -->
    <!--#if expr="$HTTP_ACCEPT_LANGUAGE = /^en/" -->
      <!--#include virtual="404_en.html" -->
    <!--#endif -->
    <!-- [...todas las condiciones que desees...] -->

    <!-- El pie de página es comun -->
    <!--#include virtual="foot.shtml" -->

  </center>
</body>
</html>
```

El contenido del fichero foot.shtml es este:

```
<br>
<a href="mailto:webmaster@dominio.com">WebMaster</a>
<br>
Dirección remota: <!--#echo var="REMOTE_HOST" -->
[<!--#echo var="REMOTE_ADDR" -->]
<br>
Hora local: <!--#echo var="DATE_LOCAL" -->
</font>
</center>
```

Los ficheros de error en distintos idiomas han de ser nombrados siguiendo el patrón "`<numero-de-error>_<idioma>.html`", como por ejemplo: `404_es.html` (mensaje de error 404 en castellano), o `500_en.html` (mensaje de error 500 en inglés). Ver Anexo C para los códigos de idioma.

El fichero `404_es.html` podría ser así:

```
<br>
<font face="Arial,Helvetica">
<font color="#FFFFFF">
<font size="+4">Error 404
</font></font></font>
<p><font face="Arial,Helvetica">
<font color="#FFFFFF"><font size="+4">
Página no encontrada
</font></font></font>
<p><font face="Arial,Helvetica"><font color="#FFFFFF">
La página que usted solicita no se encuentra en este servidor.
</font></font>
<br>
<hr WIDTH="100%">
<br>
<center>
<font color="#FFFFFF">
Si está seguro que este error es fallo del
servidor, y no suyo, contacte con
```

Ahora solo nos falta hacer un buen diseño de las páginas, traducir los mensajes a sus respectivos idiomas y reiniciar el servidor Apache para que acepte los cambios.

2.4 Consideraciones

Lo explicado aquí es muy básico, las posibilidades que ofrece SSI son muy grandes, así que te recomiendo que visites <http://www.apache.org> para informarte más ampliamente.

3. Anexos

3.1 Anexo A.- Algunas etiquetas interesantes de SSI

```
SSI:<!--#config errmsg=" -->
SSI:<!--#config sizefmt=" -->
SSI:<!--#config timefmt=" -->
SSI:<!--#echo var=" -->
SSI:<!--#exec cgi=" -->
SSI:<!--#exec cmd=" -->
SSI:<!--#fsize file=" -->
SSI:<!--#fsize virtual=" -->
SSI:<!--#flastmod file=" -->
SSI:<!--#flastmod virtual=" -->
SSI:<!--#if expr=" -->
SSI:<!--#elif expr=" -->
SSI:<!--#else -->
SSI:<!--#endif -->
SSI:<!--#include file=" -->
SSI:<!--#include virtual=" -->
SSI:<!--#printenv -->
SSI:<!--#set var=" value=" -->
```

3.2 Anexo B.- Algunas variables que se pueden usar con SSI

Variables:\$DATE_GMT
Variables:\$DATE_LOCAL
Variables:\$DOCUMENT_NAME
Variables:\$DOCUMENT_PATH_INFO
Variables:\$DOCUMENT_ROOT
Variables:\$DOCUMENT_URI
Variables:\$LAST_MODIFIED
Variables:\$GATEWAY_INTERFACE
Variables:\$HTTP_ACCEPT
Variables:\$HTTP_ACCEPT_CHARSET
Variables:\$HTTP_ACCEPT_ENCODING
Variables:\$HTTP_ACCEPT_LANGUAGE
Variables:\$HTTP_CONNECTION
Variables:\$HTTP_COOKIE
Variables:\$HTTP_HOST
Variables:\$HTTP_IF_MODIFIED_SINCE
Variables:\$HTTP_PRAGMA
Variables:\$HTTP_USER_AGENT
Variables:\$PATH
Variables:\$QUERY_STRING
Variables:\$REMOTE_ADDR
Variables:\$REMOTE_HOST
Variables:\$REMOTE_PORT
Variables:\$REQUEST_METHOD
Variables:\$SCRIPT_FILENAME
Variables:\$SERVER_ADMIN
Variables:\$SERVER_NAME
Variables:\$SERVER_PORT
Variables:\$SERVER_PROTOCOL
Variables:\$SERVER_SIGNATURE
Variables:\$SERVER_SOFTWARE
Variables:\$SCRIPT_NAME
Variables:\$USER_NAME

3.3 Anexo C.- Códigos de Idioma

afrikaans	af
albanian	sq
basque	eu
bulgarian	bg
byelorussian	be
catalan	ca
chinese	zh
chinese/China	zh-CN
chinese/Taiwan	zh-TW
croatian	hr
czech	cs
danish	da
dutch	nl
dutch/Belgium	nl-BE
english	en
english/UK	en-GB
english/USA	en-US
fearoese	fo
finnish	fi
french	fr
french/belgium	fr-BE
french/canada	fr-CA
french/france	fr-FR
french/switzerland	fr-CH
galician	gl
german	de
german/austria	de-AT
german/germany	de-DE
german/switzerland	de-CH
greek	el
hungarian	hu
icelandic	is
indonesian	id
irish	ga
italian	it

japanese	ja
korean	ko
macedonia	mk
norwegian	no
polish	pl
portuguese	pt
portuguese/brazil	pt-BR
romanian	ro
russian	ru
scots gaelic	gd
serbian	sr
slovak	sk
slovenian	sl
spanish	es
spanish/argentina	es-AR
spanish/colombia	es-CO
spanish/mexico	es-MX
spanish/spain	es-ES
swedish	sv
turkish	tr
ukrainian	uk
