

SSI – Server Side Include

1. Was ist das?

Eine Sammlung von Befehlen, die in die HTML-Seite als Kommentar eingefügt werden, um vom Server mit Hilfe eines Präprozessors interpretiert werden, bevor die Seite an den Client (Browser) geschickt wird.

2. Vorteile

Es ist damit sehr einfach dynamische Webseiten für viele verschiedenen Browsertypen zu erstellen. Da die Befehle auf der Serverseite ausgeführt werden, hat der Client wesentlich weniger Arbeit mit der Darstellung der Seite (er bekommt nur die HTML-Anweisungen, die für ihn bestimmt sind) und es bestehen auch weniger Anforderungen an den Client bzw. an die Fähigkeiten des Browsers (er muß Javascript nicht können, bzw. kann es aus Sicherheitsüberlegungen abgeschaltet sein). Da es sehr wenige Befehle sind, ist SSI leicht erlernbar und wird von vielen Webservern unterstützt. Durch die Flexibilität der Befehle ist trotz der geringen Anzahl der Entwickler in keiner Richtung eingeschränkt.

3. Nachteile

Da die Befehle vom Server ausgeführt werden, ist ein solcher auch lokal zum Testen notwendig, hier existieren aber kostenlose und kleine Webserver auch für den Win32-Client. Unter Linux stehen einem lokalen Test sowieso keine Hindernisse entgegen, da alle Seiten auch über den eigenen Rechner abgerufen werden können (file://pfad1/html-seite → http://127.0.0.1/pfad2/html-seite).

4. Namen

Damit der Präprozessor die Seite interpretiert, muß sie im Namen über die Dateierweiterung shtml (statt htm oder html) verfügen.

5. Einbettung in eine HTML-Seite

Alle SSI-Befehle werden als Kommentar in die HTML-Seite eingefügt, sodaß ein Browser die Seite auch inklusive der SSI-Befehle darstellen kann, diese in einem solchen Fall aber nicht interpretiert werden.

<!--#befehl [attribut=wert] [...] -->

Dabei darf zwischen dem Kommentaranfang „<!--“ und dem „#SSI-Befehl“ kein Leerzeichen sein, hinter dem SSI-Befehl und vor dem Commentarende „-->“ muß aber ein Leerzeichen eingefügt werden.

6. Befehle

<code>#config</code>	Konfiguration von Formaten (z.B.: des Ausgabeformats von Zeit-/Datumsvariablen)
z.B.:	<code><!--#config timefmt="%d.%m%Y" --></code> <code><!--#config errmsg="falsche SSI-Syntax" --></code> <code><!--#config sizefmt="bytes abbrev" --></code>
<code>#echo</code>	Ausgabe einer Variablen
z.B.:	<code><!--#echo var="DATE_LOCAL" --></code>
<code>#exec</code>	Ausführen eines CGI-Programmes oder eines OS-Befehls
z.B.:	<code><!--#exec cgi="/cgi-bin/counter.pl" --></code> <code><!--#exec cmd="ls" --></code>
<code>#filesize</code>	Ausgabe der Dateigröße
z.B.:	<code><!--#filesize file="index.shtml" --></code>
<code>#lastmod</code>	Ausgabe des Datums der letzten Änderung einer Datei
z.B.:	<code><!--#lastmod file="news.html" --></code>
<code>#include</code>	Eine Datei an dieser Stelle einfügen
z.B.:	<code><!--#include file="navigation.html" --></code> (nur relativ zum rufenden Dokument) <code><!--#include virtual="../inc/kopfzeile.txt" --></code> (Absolutangaben möglich)
<code>#printenv</code>	Ausgabe aller Variablen
z.B.:	<code><!--#printenv --></code>
<code>#set</code>	Setzen einer Variablen
z.B.:	<code><!--#set var="autor" value="ssicreator" --></code> <code><!--#set var="lastpage" value="\$HTTP_REFERER" --></code> <code><!--#set var="KOSTEN" value="\\$100" --></code>

und die bedingten Ausdrücke

<code>#if</code>	Beginn einer Bedingung (wenn ...)
z.B.:	<code><!--#if expr="{HTTP_HOST} = www.coufal.org" --></code> <code><!--#if expr="{HTTP_USER_AGENT} = /WebWasher/" --></code> <code><!--#if expr="{REMOTE_ADDR} = /131.130." --></code>
<code>#elif</code>	Wenn Bedingung nicht zutrifft beginne nächste Bedingung (sonst wenn ...)
z.B.:	<code><!--#elif expr="{HTTP_USER_AGENT} = /MSIE/" --></code>
<code>#else</code>	Wenn Bedingung nicht zutrifft (sonst ...)
z.B.:	<code><!--#else --></code>
<code>#endif</code>	Ende einer Bedingung
z.B.:	<code><!--#endif --></code>

7. Einfaches Beispiel

Zum Testen, ob ein bestimmter Server SSI unterstützt, kann folgende Seite unter einem beliebigen Namen auf den Server übertragen werden (z.B. als test.shtml auf den Server www.coufal.org) und mittels Browser und der URL (z.B.: <http://www.coufal.org/test.shtml>) abgerufen werden.

```
<html>
<body>
<PRE><!--#printenv --></PRE>
</body>
</html>
```

Die Ausgabe könnte dann z.B. so aussehen (Auszug):

```
DOCUMENT_ROOT=/home/coufal/coufal-www
HTTP_ACCEPT_CHARSET=iso-8859-1,* ,utf-8
HTTP_ACCEPT_ENCODING=gzip
HTTP_ACCEPT_LANGUAGE=de,en-US,en
HTTP_CONNECTION=Keep-Alive
HTTP_HOST=www.coufal.org
HTTP_USER_AGENT=Mozilla/4.7 [de] (WinNT; I)
PATH=/usr/bin:/bin:/usr/local/bin:/usr/X11R6/bin:/home/dom/bin
SCRIPT_FILENAME=/home/coufal/coufal-www/test.shtml
SERVER_ADDR=64.176.199.161
SERVER_ADMIN=webmaster@coufal.org
SERVER_NAME=coufal.org
SERVER_PORT=80
SERVER_SOFTWARE=Apache/1.3.20 (Unix) ApacheJServ/1.1.2 PHP/4.0.4p11
FrontPage/5.0.2.2510 Rewrit/1.1a
GATEWAY_INTERFACE=CGI/1.1
SERVER_PROTOCOL=HTTP/1.0
REQUEST_METHOD=GET
QUERY_STRING=
REQUEST_URI=/test.shtml
SCRIPT_NAME=/test.shtml
DATE_LOCAL=Friday, 16-Nov-2001 15:31:02 EST
DATE_GMT=Friday, 16-Nov-2001 20:31:02 GMT
LAST_MODIFIED=Friday, 05-Oct-2001 02:31:56 EDT
DOCUMENT_URI=/test.shtml
DOCUMENT_PATH_INFO=
USER_NAME=coufal
DOCUMENT_NAME=test.shtml
```

8. Timeformat-Variablen

- %a abgekürzter Name des Wochentages, abhängig von der gesetzten Umgebung
- %A ausgeschriebener Name des Wochentages, abhängig von der gesetzten Umgebung
- %b abgekürzter Name des Monats, abhängig von der gesetzten Umgebung
- %B ausgeschriebener Name des Monats, abhängig von der gesetzten Umgebung
- %c Wiedergabewerte für Datum und Zeit, abhängig von der gesetzten Umgebung
- %C Jahrhundert (Jahr geteilt durch 100, gekürzt auf Integer, Wertebereich 00 bis 99)
- %d Tag des Monats als Zahl (Bereich 00 bis 31)
- %D so wie %m/%d/%y
- %e Tag des Monats als Dezimal-Wert, einstelligen Werten wird ein Leerzeichen voran gestellt (Wertebereich ´ 0´ bis ´31´)
- %g Zur Wochennummer gehörige 2-stellige Jahreszahl als Dezimalwert
- %G Zur Wochennummer gehörige 4-stellige Jahreszahl als Dezimalwert

%h	so wie %b
%H	Stunde als Zahl im 24-Stunden-Format (Bereich 00 bis 23)
%I	Stunde als Zahl im 12-Stunden-Format (Bereich 01 bis 12)
%j	Tag des Jahres als Zahl (Bereich 001 bis 366)
%k	Stunde als Zahl im 24-Stunden-Format (einstellig mit führendem Leerzeichen)
%l	Stunde als Zahl im 12-Stunden-Format (einstellig mit führendem Leerzeichen)
%m	Monat als Zahl (Bereich 01 bis 12)
%M	Minute als Dezimal-Wert
%n	neue Zeile
%p	entweder `AM' oder `PM' oder die entsprechenden Zeichenketten der gesetzten Umgebung
%P	entweder `am' oder `pm' oder die entsprechenden Zeichenketten der gesetzten Umgebung
%r	Zeit in "am"- oder "pm"-Formatierung
%R	Zeit in der 24-Stunden-Formatierung
%s	Sekunden seit 1.1.1970 0:00 UTC
%S	Sekunden als Dezimal-Wert
%t	Tabulator
%T	aktuelle Zeit, genau wie %H:%M:%S
%u	Tag der Woche als Dezimal-Wert [1,7], dabei ist 1 der Montag
%U	Nummer der Woche des aktuellen Jahres als Dezimal-Wert, beginnend mit dem ersten Sonntag als erstem Tag der ersten Woche
%V	Kalenderwoche (nach ISO 8601:1988) des aktuellen Jahres. Als Dezimal-Zahl mit dem Wertebereich 01 bis 53, wobei die Woche 01 die erste Woche mit mindestens 4 Tagen im aktuellen Jahr ist. Die Woche beginnt am Montag (nicht am Sonntag)
%w	Wochentag als Dezimal-Wert, Sonntag ist 0
%W	Nummer der Woche des aktuellen Jahres, beginnend mit dem ersten Montag als erstem Tag der ersten Woche
%x	bevorzugte Datumwiedergabe (ohne Zeit), abhängig von der gesetzten Umgebung
%X	bevorzugte Zeitwiedergabe (ohne Datum), abhängig von der gesetzten Umgebung
%y	Jahr als 2-stellige-Zahl (Bereich 00 bis 99)
%Y	Jahr als 4-stellige-Zahl
%z	Zeitzone als Stundendifferenz zu UTC
%Z	Zeitzone als Text
%+	MMDDhhmmCCYY.ss
%%	ein %-Zeichen

9. Quellen

c't Heft 20/2001

<http://httpd.apache.org/docs/howto/ssi.html>

http://apachetoday.com/news_story.php3?ltsn=2000-06-12-001-01-PS

http://apachetoday.com/news_story.php3?ltsn=2000-06-19-002-01-NW-LF-SW

http://apachetoday.com/news_story.php3?ltsn=2000-06-26-001-01-NW-LF-SW