

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

MATHEMATISCH-
NATURWISSENSCHAFTLICHE FAKULTÄT
FACHBEREICH INFORMATIK
INFORMATIONSDIENSTE

Grundlagen der Web-Entwicklung

INF3172

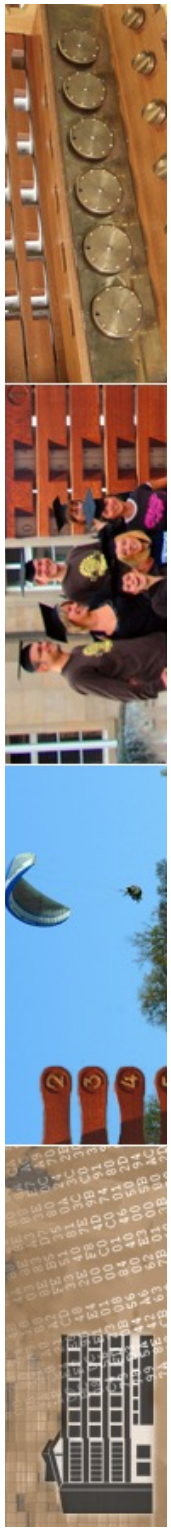
fastCGI und ServerSideIncludes

Thomas Walter

26.11.2026

Version 1.0

FastCGI



das Weihnachtsrätsel

- am **1. Advent 29.11.2026 ab 11.00h:**
  das Weihnachtsrätsel!



- Preise für schnellste Lösung und Verlosung weiterer
- Auflösung und Verlosung der Gewinne in der Vorlesung vor Weihnachten

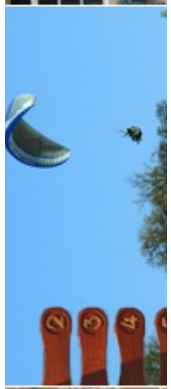
© 2026 Universität Tübingen

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Kritische Sicherheitslücke in Apache2 ermöglicht Codeeinschleusung

Nur Server mit aktiviertem HTTP/2 sind betroffen, Updates sind angeraten. Ein Exploit ist möglich, aber schwierig.







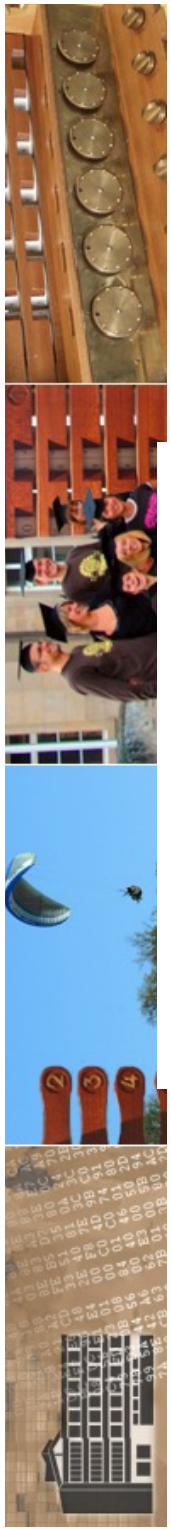
© 2026 Universität Tübingen

12.11.2026, 16:22 Uhr Lesezeit: 1 Min. | Security

Von Dr. Christopher Kunz

Im beliebten Webserver Apache2 steckt eine Sicherheitslücke, die Angreifern die Einschleusung von Schadcode ermöglichen könnte. Der Fehler versteckt sich in einer Funktion des HTTP/2-Moduls mod_http2, wo eine Speichermanipulation zur Remote Code Execution (RCE) führen könnte. Wie Sicherheitsexperten konstatieren, ist ein Exploit wegen der in modernen Betriebssystemen integrierten Zufälligkeit der Speicheradressierung (ASLR) sehr schwierig.

Die in der Apache-Sicherheitsübersicht gelistete Sicherheitslücke CVE-2026-119618 (Schweregrad **kritisch**, kein CVSS-Wert bekannt) steckt in der Funktion „assign_headers()“. Gefunden haben sie Anthropics OSS-Scanner und zwei Mitarbeiter von Google Cloud Vulnerability Research. Da sich die Lücke im standardmäßig nicht aktivierten Modul mod_http2 verbirgt, sind nur Server betroffen, die das Protokoll HTTP/2 unterstützen.



Apache HTTP Server 2.4 vulnerabilities

This page lists all security vulnerabilities fixed in released versions of Apache HTTP Server 2.4. Each vulnerability is given a security impact rating by the Apache security team - please note that this rating may well vary from platform to platform. We also list the versions the flaw is known to affect, and where a flaw has not been verified list the version with a question mark.

Please note that if a vulnerability is shown below as being fixed in a "-dev" release then this means that a fix has been applied to the development source tree and will be part of an upcoming full release.

Please send comments or corrections for these vulnerabilities to the Security Team.

The initial GA release, Apache httpd 2.4.1, includes fixes for all vulnerabilities which have been resolved in Apache httpd 2.2.22 and all older releases. Consult the Apache httpd 2.2 vulnerabilities list for more information.

Recent releases: 2.4.71 (Nov 2026, 3 CVEs) · 2.4.69 (Oct 2026, 20 CVEs) · 2.4.68 (Jun 2026, 13 CVEs) · 2.4.67 (May 2026, 11 CVEs) · 2.4.66 (Dec 2025, 5 CVEs)

▶ All 44 releases

Version 1.0: NGINX - Mehrere kritische Schwachstellen "NGINX2Root" bedrohen Webserver

Datum 19.11.2026



Am 18. November 2026 veröffentlichte die US-Firma F5 Inc. mehrere Advisories zu kritischen Schwachstellen in NGINX Open Source und NGINX Plus. Diese betreffen unterschiedliche Komponenten und Verarbeitungswege, darunter HTTP/2, HTTP/3, den DNS-Resolver sowie den Mail-Proxy. Die Sicherheitsforscher von Xint fassen fünf NGINX-Schwachstellen unter der Bezeichnung "NGINX2Root" mit einer weiteren Linux Schwachstelle zur Rechteausweitung zusammen.

Die fünf NGINX-Schwachstellen werden unter den Kennungen CVE-2026-123007, CVE-2026-123634, CVE-2026-123633, CVE-2026-121810 und CVE-2026-113425 geführt. Sie können abhängig von der eingesetzten Konfiguration unter anderem Denial-of-Service (DoS), Speicherbeschädigungen und die Ausführung von Code aus der Ferne ohne Authentifizierung ermöglichen. Die Codeausführung erfolgt mit den Rechten des jeweiligen NGINX-Worker-Prozesses. Die fünf Schwachstellen werden mit einem CVSSv3.1 Basescore von 8.1/10 ("hoch") sowie nach CVSSv4.0 mit 9.2/10 ("kritisch") geführt.

Xint demonstrierte eine nicht-authentifizierte Codeausführung über CVE-2026-123007 mit den Rechten des NGINX-Worker-Prozesses. Ein funktionsfähiges Proof-of-Concept Exploit wurde von Rapid7 für CVE-2026-123007 veröffentlicht.

© 20
Tübi

TLP:CLEAR

Bundesamt
für Sicherheit in der
Informationstechnik

Nationales
IT-Lagezentrum
BSI

PHP: Kritische Schwachstellen bedrohen Webanwendungen weltweit

BITS-H Nr. 2026-297981-1332, Version 1.3, 17.11.2026

Kritikalität*: 3 / Orange

Achtung: Für die schriftliche und mündliche Weitergabe dieses Dokumentes und der darin enthaltenen Informationen gelten gemäß dem Traffic Light Protokoll (TLP) die folgenden Einschränkungen:

TLP:CLEAR: Unbegrenzte Weitergabe

Abgesehen von urheberrechtlichen Aspekten, die das TLP explizit nicht adressiert, dürfen Informationen der Stufe TLP:CL Einschränkungen frei weitergegeben werden.

Das Dokument ist durch den Empfänger entsprechend den vereinbarten „Ausführungsbestimmungen zum sicheren Informationsaustausch mit TLP" zu verarbeiten, aufzubewahren, weiterzugeben und zu vernichten. Weitere Informa finden Sie am Ende dieses Dokumentes.

Bitte prüfen Sie selbstständig, für welche Stellen in Ihrer Organisation die hier genannten Informationen relevant sir das Schreiben entsprechend weiter – sofern die TLP-Einstufung dies zulässt.

Übersicht

Betroffenes Produkt	PHP	**Gefährdung**	Aktive Ausnut
Betroffene Versionen	alle PHP-Versionen	**Schutzmaßnahmen**	Patch + Worka
Behebende Versionen	• 8.5.12 • 8.4.27 • 8.3.36 • 8.2.35	**CVEs**	• CVE-2026-109618 (9.2) • CVE-2026-109619 (9.2)

Alert!

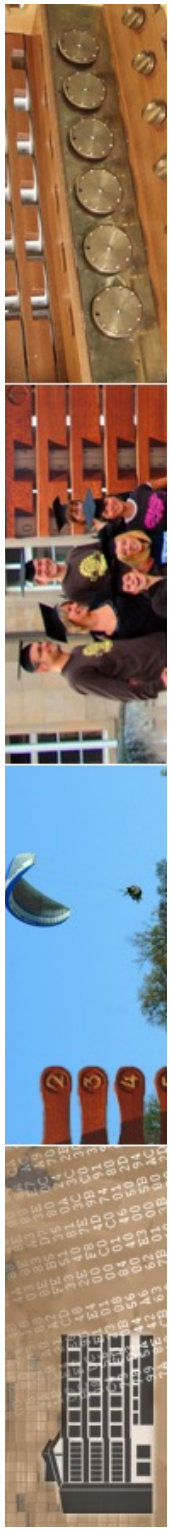
PHP behebt 17 Sicherheitslücken, fünf erlauben Ausführung von Schadcode

Die kritischen Sicherheitslücken sind nur mit Vorbedingungen und nicht auf Standardinstallationen ausnutzbar; Admins sollten dennoch updaten.

19.11.2026, 15:42 Uhr **Lesezeit: 4 Min.** | **Security**

Von Dr. Christopher Kunz

Vier Wochen nach dem Rummel um die „php2shell"-Sicherheitslücken legt das Entwicklerteam der freien Skriptsprache eine neue Aktualisierung für vier Versionszweige vor. Sie behebt 17 Sicherheitslücken, sieben davon kritisch und fünf mit Potenzial zur Ausführung von Schadcode. Admins sollten ihre PHP-Intallationen aktualisieren. Die Veröffentlichung ist Teil des monatlichen PHP-Updates.



550 Jahre seit 1477 UNIVERSITÄT TÜBINGEN

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

SUPPORT SERVICE-KATALOG FORSCHUNG UND LEHRE DAS URZ

URZ UNIVERSITÄTS-RECHENZENTRUM

UNIVERSITÄT HEIDELBERG ZUKUNFT SEIT 1386

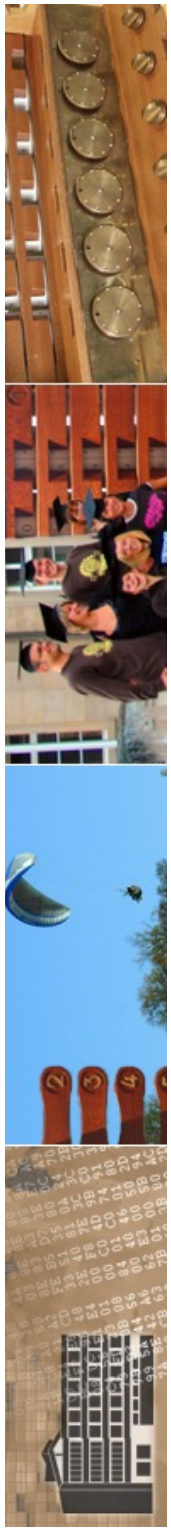
BETRIEBSMELDUNG - 19. NOVEMBER 2025

ANGRIFF AUF DIE IT-INFRASTRUKTUR: VORSICHTSMASSNAHMEN

ZUR SOFORTIGEN UMSETZUNG

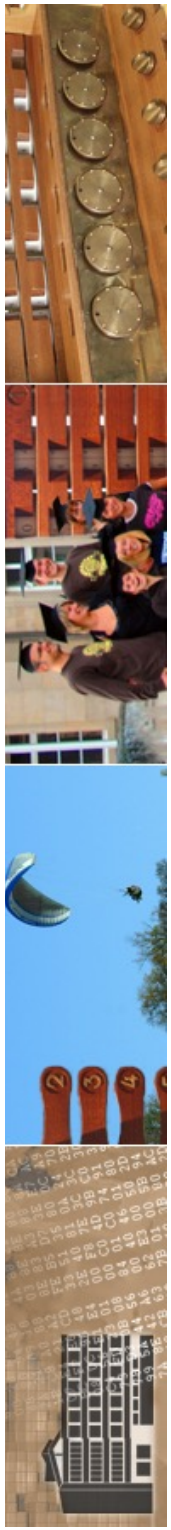
Die Universität Heidelberg hat Vorbereitungen auf einen weitreichenden Angriff auf die IT-Infrastruktur identifiziert. Die Abwehr erfordert die **sofortige Umsetzung notwendiger Vorsichtsmaßnahmen**. Diese Maßnahmen beinhalten unter anderem eine umgehende **Änderung der Passwörter.** Zudem sind ab sofort zahlreiche I**T-Dienste nur noch über VPN oder aus dem Campusnetzwerk erreichbar.**
Es wird mit Nachdruck daran gearbeitet, Schaden von der Universität abzuwenden. Für technische Rückfragen wenden Sie sich bitte an den IT-Service.

weitere Informationen Vorsichtsmaßnahmen →



FastCGI

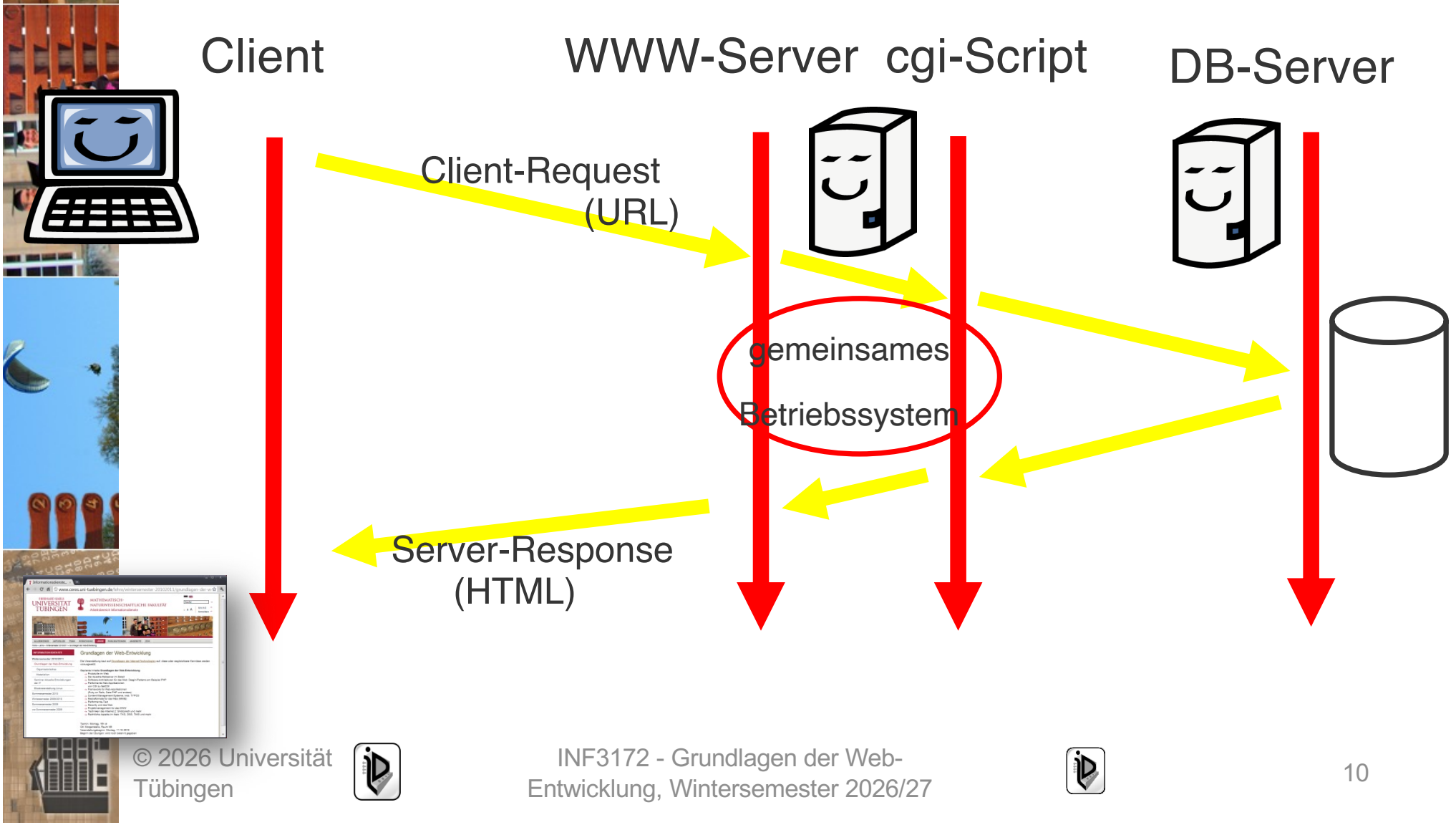
- Ansatz zum *wesentlichen Beschleunigen von Server-Anwendungen*
- hat sich zunächst selbst nicht direkt durchgesetzt, aber *Basis für viele ähnlichen Techniken* gesetzt
- für Ruby on Rails sehr wichtig
- *seit Apache 2.4 Standardmodul*

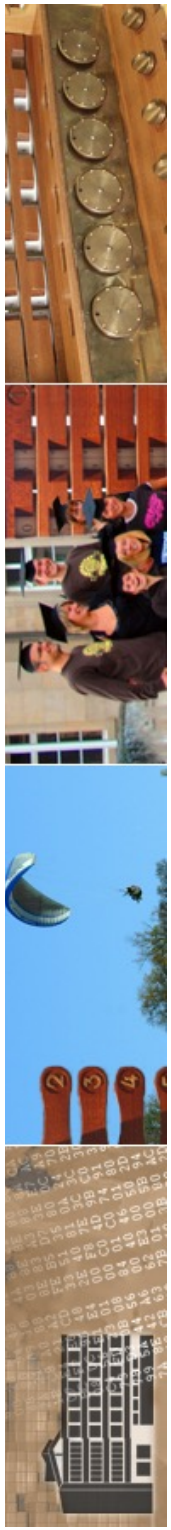


CGI Serverstruktur

- es sind **drei** Rechner im Spiel:
  - der Web-Client, der den HTTP-Request sendet
  - der Webserver, der zugleich auch den CGI-Prozeß ausführen muß
  - optional der Datenbankserver
  - je nach Fall sind es weniger Server, insbesondere ist häufig der Datenbank- und der Webserver die gleiche Maschine

Struktur einer HTTP-Transaktion mit cgi *und Datenbank*

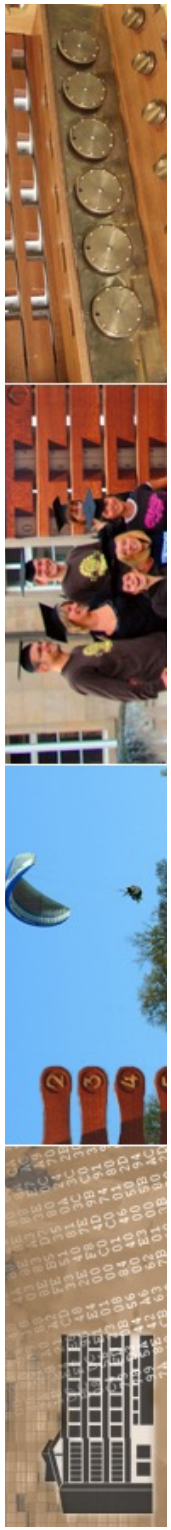




CGI: Umgebungsvariablen

- Server legt bei Aufruf des cgi-Programms spezielle **Umgebungsvariablen** fest, die dem cgi-Programm die notwendigen Informationen liefern
  (interner Mechanismus der Datenübergabe)

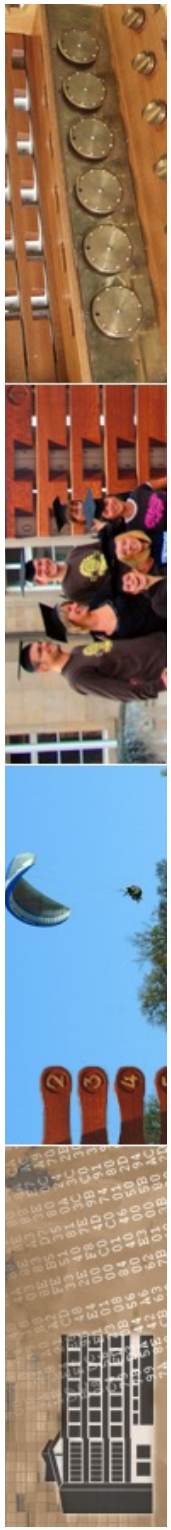
  einige dieser Umgebungsvariablen:
  - **`SERVER_NAME, SERVER_PROTOCOL, SERVER_PORT, ...`**
  - **`PATH_INFO, SCRIPT_NAME, QUERY_NAME`**
  - **`REMOTE_HOST, REMOTE_ADDR, REMOTE_USER`**



thomas — ssh ◂ -bash — 56×14

```
zrvwa01@infodienste =>
zrvwa01@infodienste => cd apache_test/cgi-bin/
zrvwa01@infodienste => ll
insgesamt 24
drwxr-xr-x  2 zrvwa01 4096 29. Okt 09:06 ./
drwxr-xr-x 14 zrvwa01 4096 29. Okt 09:06 ../
-rw-r--r--  1 zrvwa01  820 17. Dez 2012  printenv
-rw-r--r--  1 zrvwa01 1074 17. Dez 2012  printenv.vbs
-rw-r--r--  1 zrvwa01 1133 17. Dez 2012  printenv.wsf
-rw-r--r--  1 zrvwa01 1261 17. Dez 2012  test-cgi
zrvwa01@infodienste =>
zrvwa01@infodienste =>
zrvwa01@infodienste =>
zrvwa01@infodienste =>
```

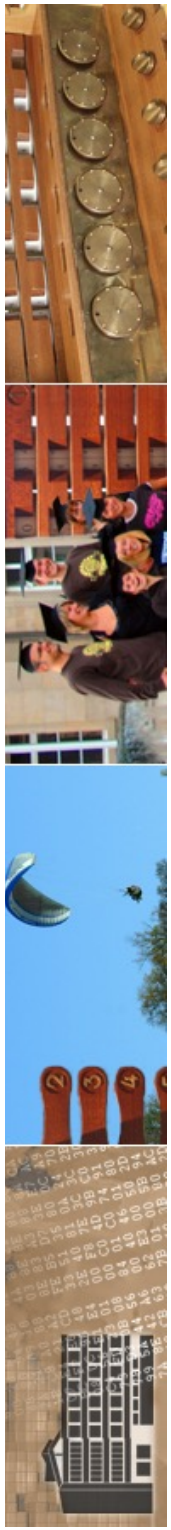
© 2026 Universität Tübingen



thomas — ssh ◂ -bash — 83×30

```
[zrvwa01@infodienste =>                                                    ]
zrvwa01@infodienste => more printenv
#

# To permit this cgi, replace # on the first line above with the
# appropriate #!/path/to/perl shebang, and on Unix / Linux also
# set this script executable with chmod 755.
#
# ***** !!! WARNING !!! *****
# This script echoes the server environment variables and therefore
# leaks information - so NEVER use it in a live server environment!
# It is provided only for testing purpose.
# Also note that it is subject to cross site scripting attacks on
# MS IE and any other browser which fails to honor RFC2616.

##
##  printenv -- demo CGI program which just prints its environment
##
use strict;
use warnings;

print "Content-type: text/plain; charset=iso-8859-1\n\n";
foreach my $var (sort(keys(%ENV))) {
    my $val = $ENV{$var};
    $val =~ s|\n|\\n|g;
    $val =~ s|"|\\"|g;
    print "${var}=\"${val}\"\n";
}

zrvwa01@infodienste =>
```

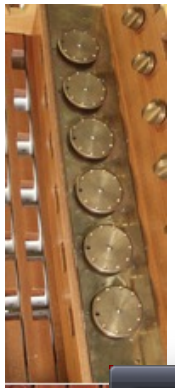
© 2026 Universität Tübingen



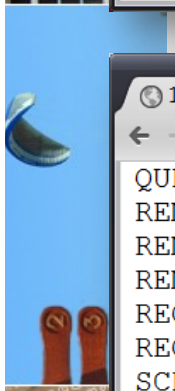
thomas — ssh ◂ -bash — 83×34

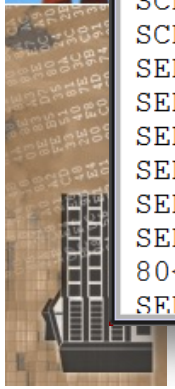
```
[zrvwa01@infodienste =>                                                 ]
[zrvwa01@infodienste =>                                                 ]
[zrvwa01@infodienste => ./printenv                                      ]
Content-type: text/plain; charset=iso-8859-1

CC="gcc"
DBUS_SESSION_BUS_ADDRESS="unix:path=/run/user/605/bus"
HOME="/home-link/zrvwa01"
LANG="de_DE.UTF-8"
LOGNAME="zrvwa01"
MOTD_SHOWN="pam"
OLDPWD="/home-link/zrvwa01"
PATH="/Library/Frameworks/Python.framework/Versions/3.7/bin:/Library/Frameworks/Pyt
hon.framework/Versions/3.6/bin:/home-link/zrvwa01/.cargo/bin:/Library/Frameworks/Py
thon.framework/Versions/3.6/bin:/usr/local/bin:/usr/bin:/bin:/usr/local/games:/usr/
games:/Library/Frameworks/Python.framework/Versions/3.4/bin:."
PS1="\u@\h => "
PS2=" => "
PWD="/home-link/zrvwa01/apache_test/cgi-bin"
SHELL="/bin/bash"
SHLVL="1"
SSH_CLIENT="134.2.246.4 52730 22"
SSH_CONNECTION="134.2.246.4 52730 134.2.6.167 22"
SSH_TTY="/dev/pts/1"
TERM="xterm-256color"
USER="zrvwa01"
XDG_RUNTIME_DIR="/run/user/605"
XDG_SESSION_CLASS="user"
XDG_SESSION_ID="7386"
XDG_SESSION_TYPE="tty"
_="./printenv"
[zrvwa01@infodienste =>                                                 ]
[zrvwa01@infodienste =>                                                 ]
zrvwa01@infodienste =>
```

Script printenv.pl









127.0.0.1/cgi-bin/print ×

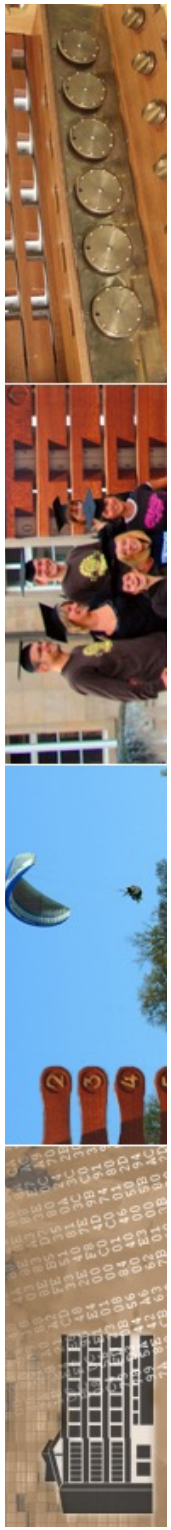
127.0.0.1/cgi-bin/printenv.pl

```
DOCUMENT_ROOT="D:/www"
GATEWAY_INTERFACE="CGI/1.1"
HTTP_ACCEPT="text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8"
HTTP_ACCEPT_CHARSET="ISO-8859-1,utf-8;q=0.7,*;q=0.3"
HTTP_ACCEPT_ENCODING="gzip,deflate,sdch"
HTTP_ACCEPT_LANGUAGE="de-DE,de;q=0.8,en-US;q=0.6,en;q=0.4"
HTTP_CONNECTION="keep-alive"
HTTP_HOST="127.0.0.1"
HTTP_USER_AGENT="Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/535.1 (KHTML, like Gecko)
```

127.0.0.1/cgi-bin/print ×

127.0.0.1/cgi-bin/printenv.pl

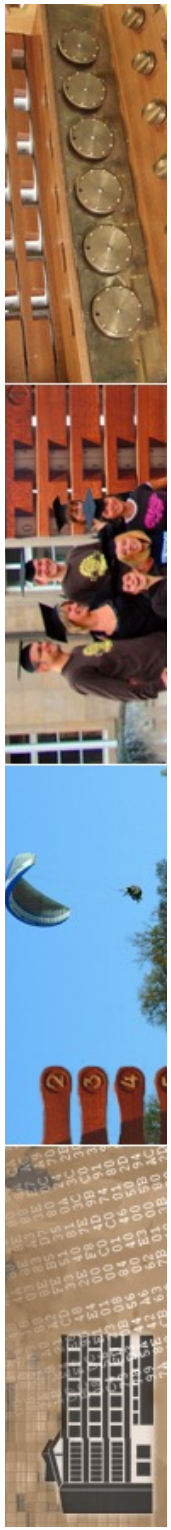
```
QUERY_STRING=""
REMOTE_ADDR="127.0.0.1"
REMOTE_HOST="mouton"
REMOTE_PORT="58323"
REQUEST_METHOD="GET"
REQUEST_URI="/cgi-bin/printenv.pl"
SCRIPT_FILENAME="C:/Program Files (x86)/Apache Software Foundation/Apache2.2/cgi-bin/printenv.pl"
SCRIPT_NAME="/cgi-bin/printenv.pl"
SERVER_ADDR="127.0.0.1"
SERVER_ADMIN="thomas.walter@uni-tuebingen.de"
SERVER_NAME="127.0.0.1"
SERVER_PORT="80"
SERVER_PROTOCOL="HTTP/1.1"
SERVER_SIGNATURE="<address>Apache/2.2.15 (Win32) PHP/5.3.2 Server at 127.0.0.1 Port
80</address>\n"
SERVER_SOFTWARE="Apache/2.2.15 (Win32) PHP/5.3.2"
```



Vorteile CGI

- einfach
- Sprachunabhängigkeit
- offener Standard
- Architekturunabhängigkeit
- Kapselung in einzelnen Prozessen
- läuft mit den meisten (allen?) Webservern, wenn entsprechend konfiguriert

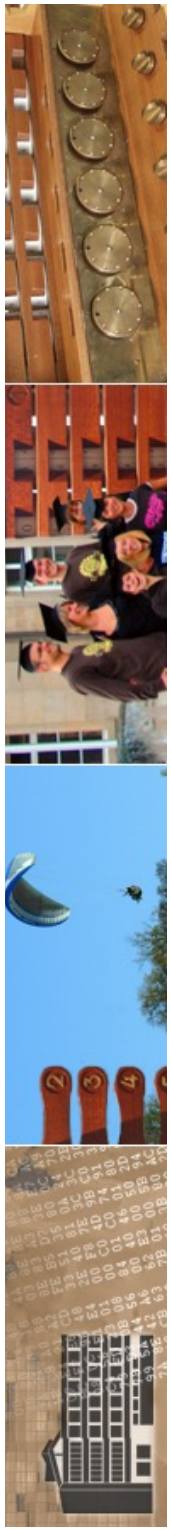
© 2026 Universität Tübingen



Nachteile CGI

- bei jeder Anfrage wird CGI-Prozeß gestartet, nach abarbeiten wieder beendet
  - Beispielsweise wird bei jeder Anfrage für ein Python-CGI der Python-Interpreter neu gestartet
- ➔ keine optimale Performance

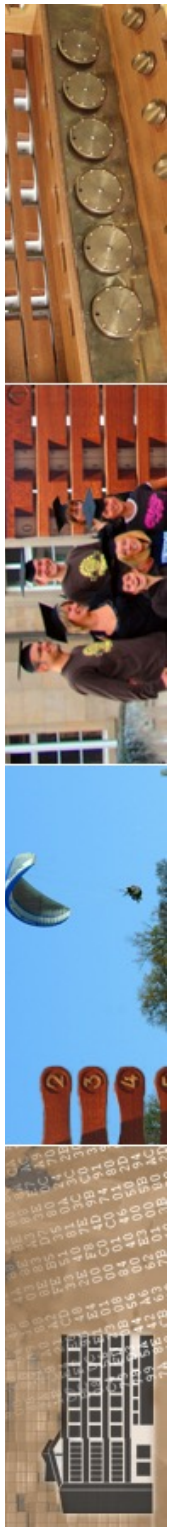
- Idee: »**Daemonize It**«
  - Anstelle eines einzelnen Prozesses **Hintergrundprozess** (Daemon) verwenden
  - Daemon wird nur **einmal** gestartet und wartet dauerhaft auf Anfragen



Grundkonzept fastCGI

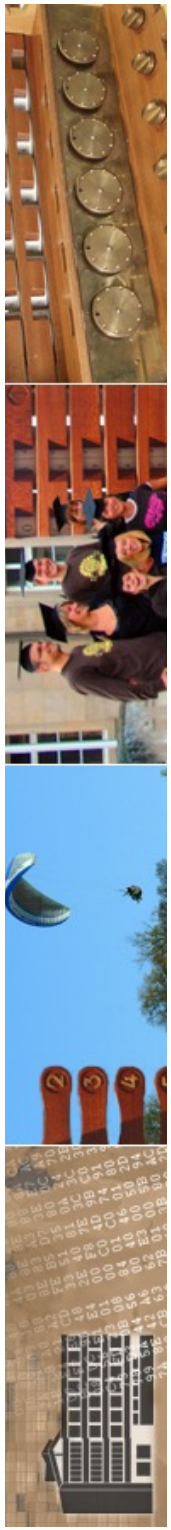
- das um 1995 eingeführte fastCGI-Konzept greift an dieser Stelle die Vorteile von CGI auf, um einen Schritt weiter zu gehen.

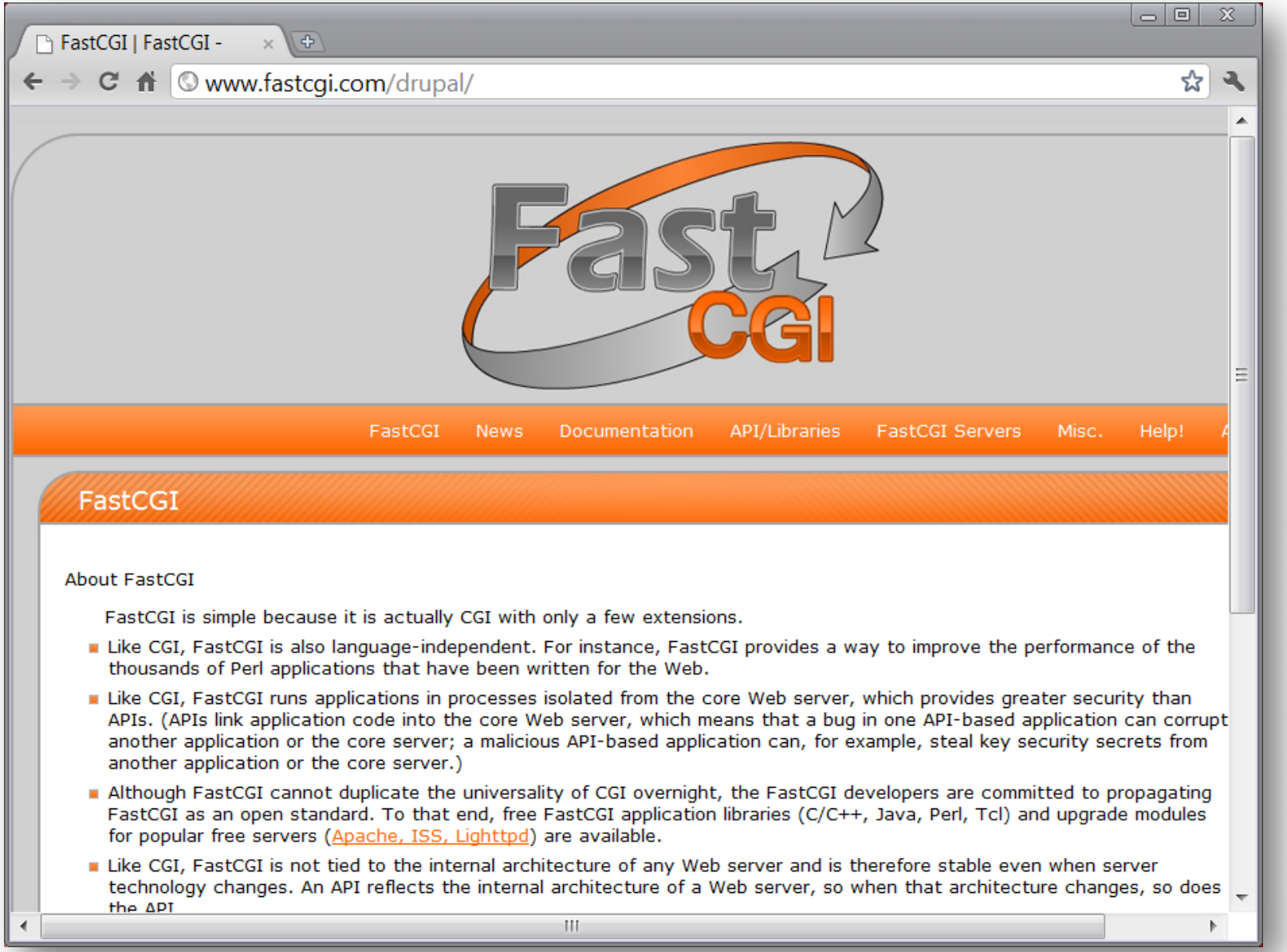
- Wir wollen folgendes erreichen:
  - höhere Performance der CGIs
  - Trennung von Webserver und Applikationsserver (CGI-Server), um mehr Skalierungsmöglichkeiten zu haben

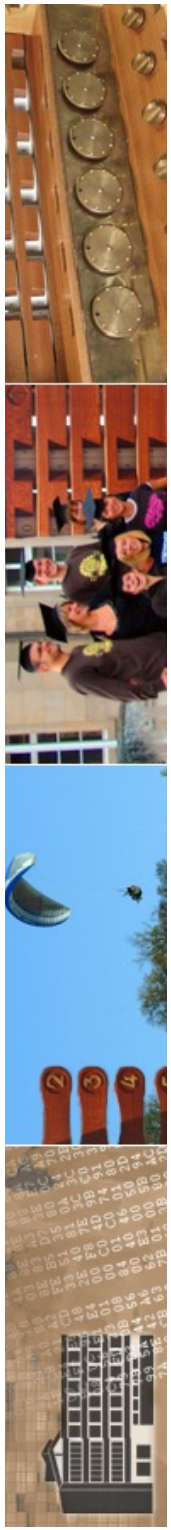


fastCGI

- entwickelt durch Open Market, Inc.
  - **`http://www.fastcgi.com`**
  - **`http://www.openmarket.com/fastcgi`**
- damit auch Trennung von Webserver und cgi-Server möglich, dadurch ideal *skalierbar*
  - cgi: Webserver und CGI kommunizieren über pipes und Umgebungsvariablen: ein einzelner Server für beides
  - fastcgi: Webserver und fastcgi kommunizieren über tcp-Socket: **Trennung der Server möglich**
  - »Umwandlung« eines CGI-Programmes notwendig

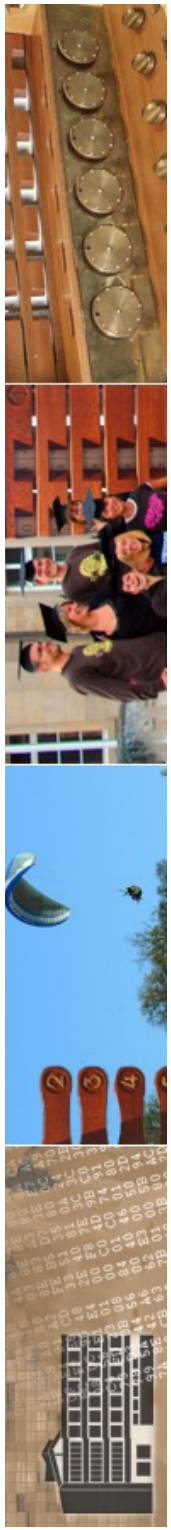



FastCGI | FastCGI -
www.fastcgi.com/drupal/
Fast CGI
FastCGI
News
Documentation
API/Libraries
FastCGI Servers
Misc.
Help!
FastCGI
About FastCGI
FastCGI is simple because it is actually CGI with only a few extensions.
Like CGI, FastCGI is also language-independent. For instance, FastCGI provides a way to improve the performance of the thousands of Perl applications that have been written for the Web.
Like CGI, FastCGI runs applications in processes isolated from the core Web server, which provides greater security than APIs. (APIs link application code into the core Web server, which means that a bug in one API-based application can corrupt another application or the core server; a malicious API-based application can, for example, steal key security secrets from another application or the core server.)
Although FastCGI cannot duplicate the universality of CGI overnight, the FastCGI developers are committed to propagating FastCGI as an open standard. To that end, free FastCGI application libraries (C/C++, Java, Perl, Tcl) and upgrade modules for popular free servers (Apache, ISS, Lighttpd) are available.
Like CGI, FastCGI is not tied to the internal architecture of any Web server and is therefore stable even when server technology changes. An API reflects the internal architecture of a Web server, so when that architecture changes, so does the API



fastCGI

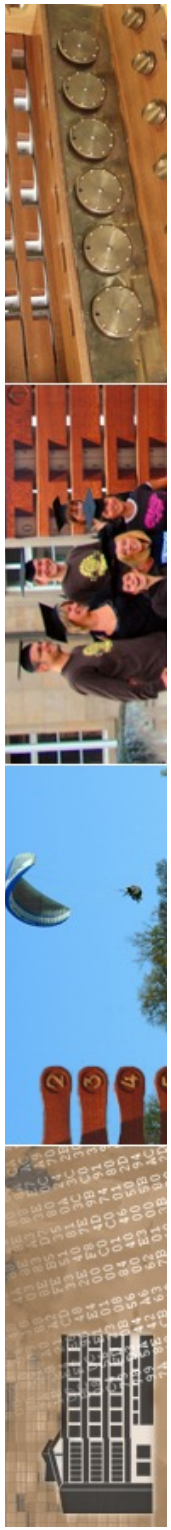
- »fastCGI is a fast, open, and secure Web server interface that solves the performance problems inherent in CGI, without introducing the overhead and complexity of proprietary APIs.«
  - »proprietary APIs«: Anwendungsentwicklung direkt für den jeweiligen Webserver, etwa als Apache-Modul



fastCGI

- Vorteile von fastCGI:
  - Performance
  - Einfachheit, mit einfacher Migration von CGI
    - nur unwesentlich komplexer als native CGI
  - Sprachunabhängigkeit
  - Prozeßisolation
  - offener, nichtproprietärer Standard
  - Architekturunabhängigkeit
  - unterstützt verteilte Systeme

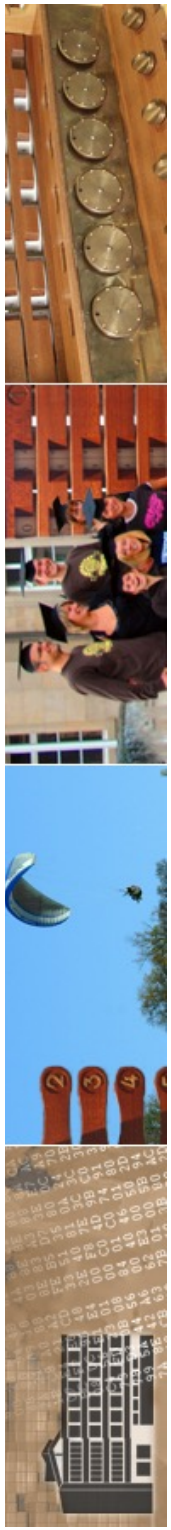
© 2026 Universität Tübingen



Unterschied CGI zu fastCGI

- A: fastCGI-Prozesse sind **persistent**: Nach Beendigung einer Anfrage werden sie *nicht* beendet, sondern warten auf nächste Anfrage

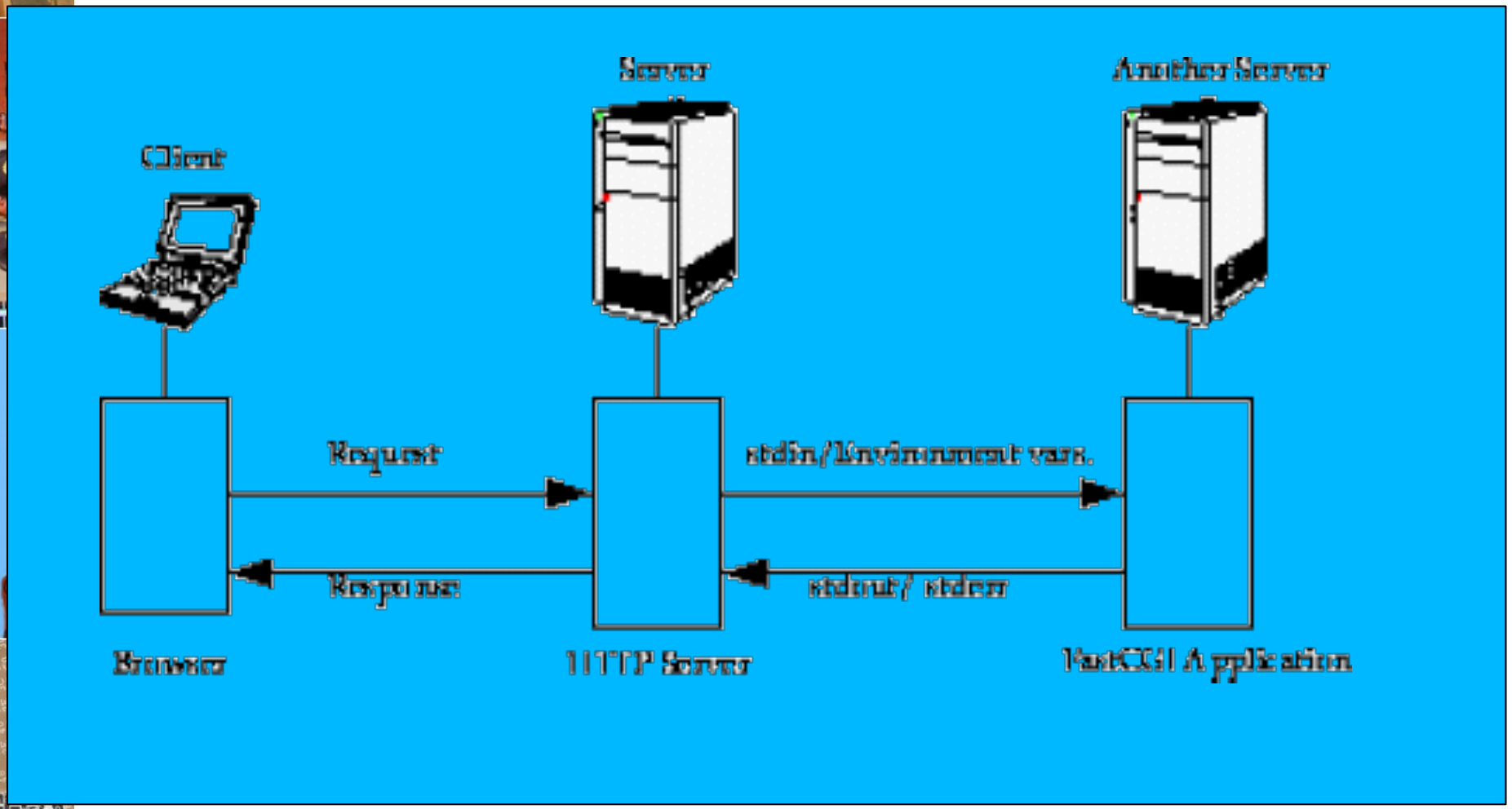
- B: fastCGI verwendet **keine Umgebungsvariablen** und *keine* pipes zur Kommunikation mit dem Webserver, sondern tcp-Verbindung; als Folge können Webserver und fastCGI-Server auch getrennt werden

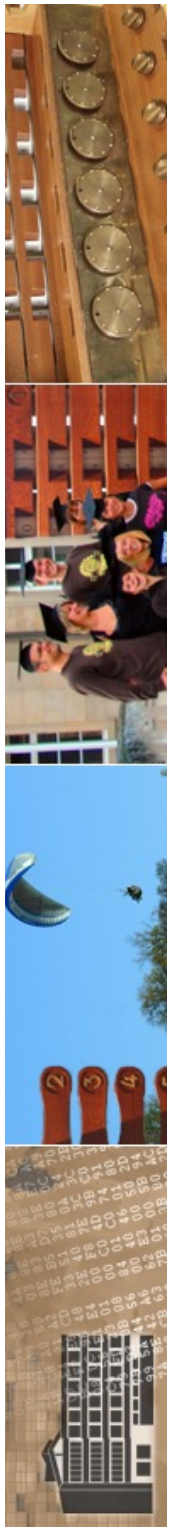


typischer Ablauf

- Webserver erzeugt fastCGI-Anwendungsprozeß
  - beim Start des Webservers oder bei erster Anfrage
- fastCGI-Prozeß initialisiert sich selbst und wartet auf Anfrage
- bei Client-Anfrage an Webserver: Webserver baut Verbindung zu fastCGI-Prozeß auf und sendet darüber CGI-Umgebungsvariablen und stdin
- fastCGI sendet stdout und stderr entsprechend zurück
- fastCGI wartet auf weitere Anfrage

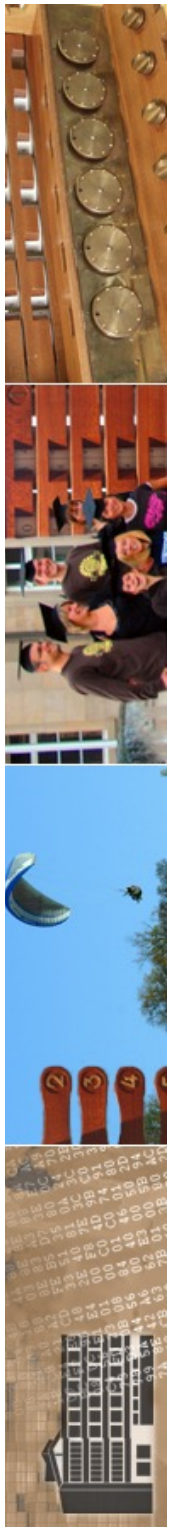
Schema fastCGI





fastCGI-Protokoll

- fastCGI verwendet ein spezielles Protokoll zur Kommunikation zwischen Webserver und Prozess
  - **`FCGI_PARAMS`**: name/value-Paare für CGI-Umgebungsvariablen
  - **`FCGI_STDIN`**
  - **`FCGI_STDOUT`**
  - **`FCGI_STDERR`**
  - **`FCGI_END_REQUEST`**: Ende der Anfrage
  - ...und weitere, siehe Web



für fastCGI notwendig...

- ...ist das (frei verfügbare) »**fastCGI Developer's Kit**«

- aktuell Version 2.4.6 (13.11.2007)
  - umfaßt viele Tools
  - u.a. fastCGI-Module für PERL, Python, C und Java
  - Online-Dokumentation
  - Protokoll-Beschreibung

  - https://fastcgi-archives.github.io/FastCGI_Developers_Kit_FastCGI.html

fastcgi-archives.github.io/FastCGI_Developers_Kit_FastCGI.html

FastCGI Developer's Kit

This FastCGI Developer's Kit is designed to make developing FastCGI applications easy.
The kit currently supports FastCGI applications written in C/C++, Perl, Tcl, and Java.

View on GitHub

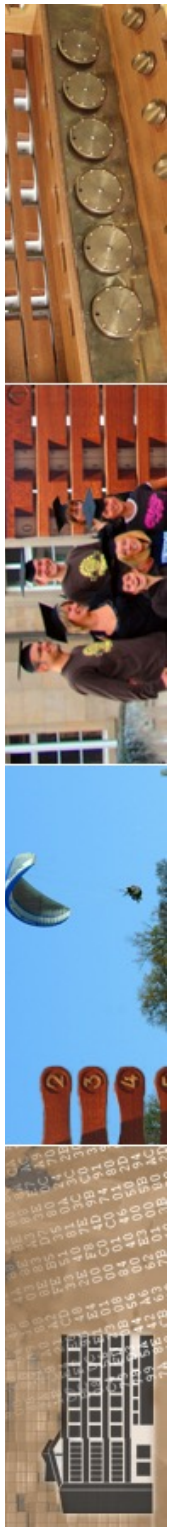
Mark R. Brown

Open Market, Inc.

Document Version: 1.08
11 June 1996

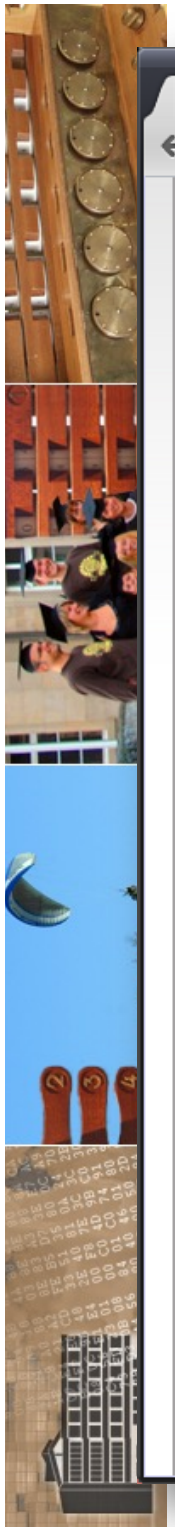
Copyright © 1996 Open Market, Inc. 245 First Street, Cambridge, MA 02142 U.S.A.

- 1. Introduction
- 2. Getting started
- 3. Writing applications
  - 3.1 Using the `fcgi_stdio` library
  - 3.2 Using the `fcgiapp` library
  - 3.3 Using Perl and Tcl
  - 3.4 Using Java



der Webserver und fastCGI

- die Nutzung von fastCGI erfordert serverseitige Voraussetzungen:
  - das entsprechende Modul muss vorhanden sein (für Apache **mod_fastcgi**)
  - der Server muß entsprechend konfiguriert sein
  - fastCGI ist mit zahlreichen Webservern kombinierbar
    - Apache
    - nginx
    - Microsoft IIS, ...



Module mod_fastcg...

www.fastcgi.com/drupal/node/25

Module mod_fastcgi

Module mod_fastcgi

This 3rd party module provides support for the FastCGI protocol. FastCGI is a language independent, scalable, open extension to CGI that provides high performance and persistence without the limitations of server specific APIs.

FastCGI applications are not limited to a particular development language (the protocol is open). FastCGI application libraries currently exist for Perl, C/C++, Java, Python, TCL, SmallEiffel, and Smalltalk.

FastCGI applications use TCP or Unix sockets to communicate with the web server. This scalable architecture allows applications to run on the same platform as the web server or on many machines scattered across an enterprise network.

FastCGI applications are portable to other web server platforms. FastCGI is supported either directly or through commercial extensions by most popular web servers.

FastCGI applications are fast because they're persistent. There is no per-request startup and initialization overhead. This makes possible the development of applications which would otherwise be impractical within the CGI paradigm (e.g. a huge Perl script, or an application which requires a connection to one or more databases).

See the FastCGI website for more information. To receive FastCGI related announcements and notifications of software updates, subscribe to fastcgi-announce. To participate in the discussion of `mod_fastcgi` and FastCGI application development, subscribe to fastcgi-developers.

Summary

For information about building and installing the module, see the INSTALL document that came with the distribution.

FastCGI applications under `mod_fastcgi` are defined as one of three types: static, dynamic, or external. They're configured using the FastCgiServer, FastCgiConfig, and FastCgiExternalServer directives respectively. Any URI that Apache identifies as a FastCGI application and which hasn't been explicitly configured using a FastCgiServer or FastCgiExternalServer directive is handled as a dynamic application (see the FastCgiConfig directive for more information).

FastCGI static and dynamic applications are spawned and managed by the FastCGI Process Manager, fcgi-pm. The process manager is spawned by Apache at server initialization. External applications are presumed to be started and managed independently.

Apache must be configured to identify requests for FastCGI URIs. `mod_fastcgi` registers (with Apache) a handler type of `fastcgi-script` for this purpose.

To configure Apache to handle all files (within the scope of the directive) as FastCGI applications (e.g. for a fcgi-bin directory):

Apache Module mod_proxy_fcgi

Available Languages: en | fr

Description:	FastCGI support module for mod_proxy
Status:	Extension
Module Identifier:	proxy_fcgi_module
Source File:	mod_proxy_fcgi.c
Compatibility:	Available in version 2.3 and later

Summary

This module *requires* the service of mod_proxy. It provides support for the FastCGI protocol.

Thus, in order to get the ability of handling the FastCGI protocol, mod_proxy and mod_proxy_fcgi have to be present in the server.

Unlike mod_fcgid and mod_fastcgi, mod_proxy_fcgi has no provision for starting the application process; fcgistarter is provided (on some platforms) for that purpose. Alternatively, external launching or process management may be available in the FastCGI application framework in use.

Warning

Do not enable proxying until you have secured your server. Open proxy servers are dangerous both to your network and to the Internet at large.

Examples

Remember, in order to make the following examples work, you have to enable mod_proxy and mod_proxy_fcgi.

Single application instance

```
ProxyPass "/myapp/" "fcgi://localhost:4000/"
```

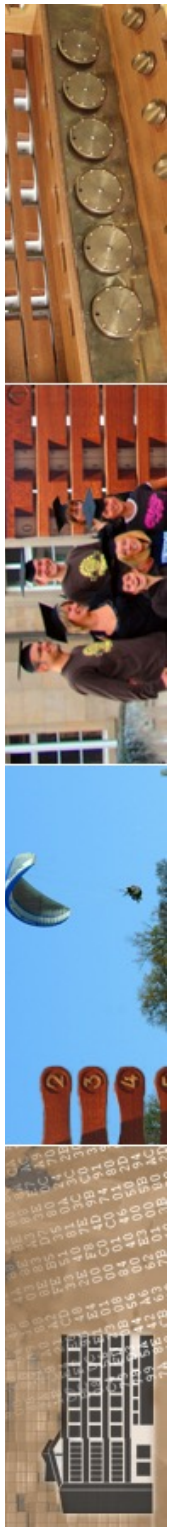
mod_proxy_fcgi disables connection reuse by default, so after a request has been completed the connection will NOT be

Topics
- Examples
- Environment Variables

Directives
ProxyFCGIBackendType
ProxyFCGISetEnvIf

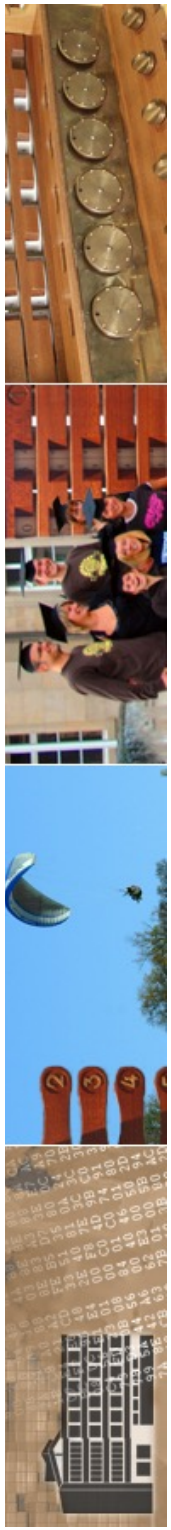
Bugfix checklist
- httpd changelog
- Known issues
- Report a bug

See also
- fcgistarter
- mod_proxy
- mod_authnz_fcgi
- Comments



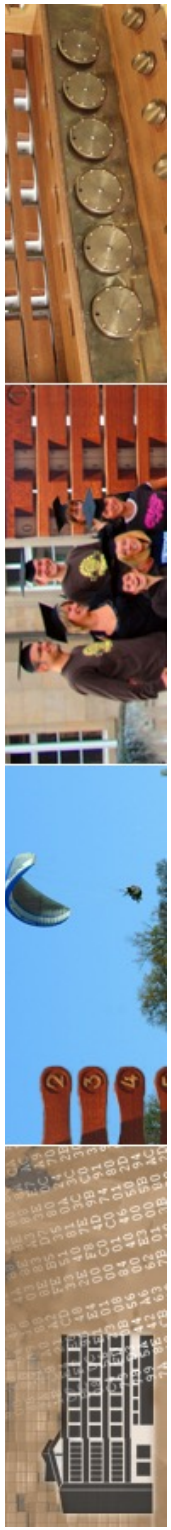
Konfiguration des Apache

- **`SetHandler`** und **`AddHandler`** für fastCGI
- **`fastCGI-Server`**
  (mit sehr zahlreichen Optionen)
- es gibt einige weitere Direktiven für fastCGI:
  - **`FastCgiExternalServer`**
  - **`FastCgiConfig`**
  - **`FastCgiAuthenticator`**
  - **`FastCgiAuthenticatorAuthoritative`**
  - **`FastCgiAuthorizer`**
  - `...`



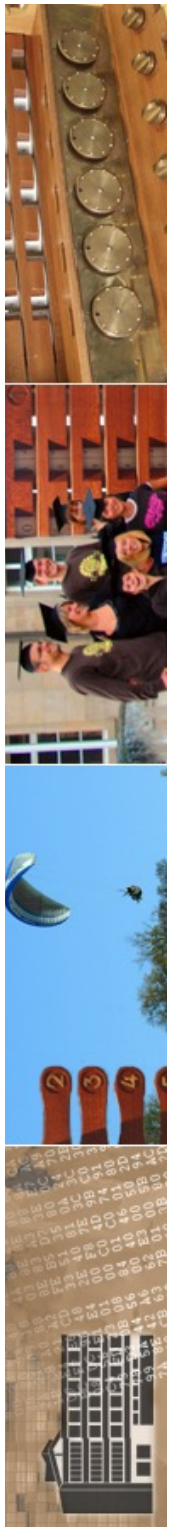
was passiert nun?

- beim Starten von Apache startet der FastCGI-Prozess **automatisch mit**
- der FastCGI-Prozess wird vom Prozeßmanager des mod_fastcgi gesteuert!!!! (konfigurierbar)
- ...jeweils, wenn beides auf einer Maschine, ansonsten separates Starten des FastCGI-Daemons



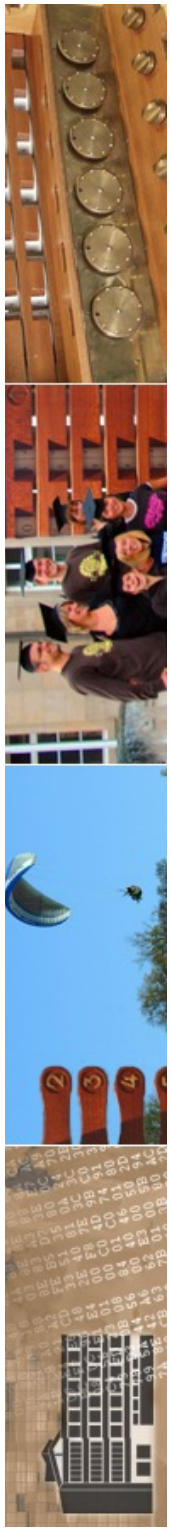
der fastCGI-Prozeßmanager für Apache

- der Prozessmanager startet den fastCGI-Prozess und achtet darauf, daß dieser „durchläuft“
  - ...ist auch noch konfigurierbar...
- beim Beenden des Webservers wird auch fastCGI-Prozess beendet



fastCGI selber programmieren

- fastCGI ist mit allen für CGI »gängigen« Programmiersprachen kombinierbar (alles notwendige ist Bestandteil des Developer's Kits):
  - C/C++
  - **PERL**
  - **Java**
  - Schema
  - Eiffel
  - **Python**
  - **Ruby**
  - TCL
  - Smalltalk



Entwicklung von fastCGI-Anwendungen

- Bestehende CGI-Applikationen können *nicht* direkt übernommen werden, sondern müssen etwas angepasst werden
  - Einbindung der entsprechenden fastCGI-Libraries
  - typisch *Endlos-Schleife* der Art:

```
Initialisierung;

while (FCGI_Accept() >= 0 )  {
   Process request;
}
```

  - Schleife wartet »für immer« auf neue fastCGI-Anfrage

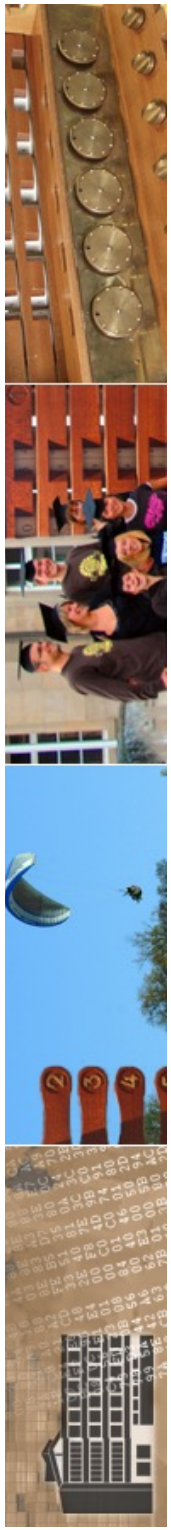
D:\www\webkompendium\material\fcgi\fcgi.pl - Notepad++

Datei Bearbeiten Suchen Ansicht Format Sprachen Einstellungen Makro Ausführen TextFX Erweiterungen Fenster ?

zustaendig.txt | schedule.txt | uebung32.php | webst1_20091_10.tex | ProgressBar.class.php | ajaxtest.html | ajaxparameter.html | parameter.php.txt | fcgi.pl

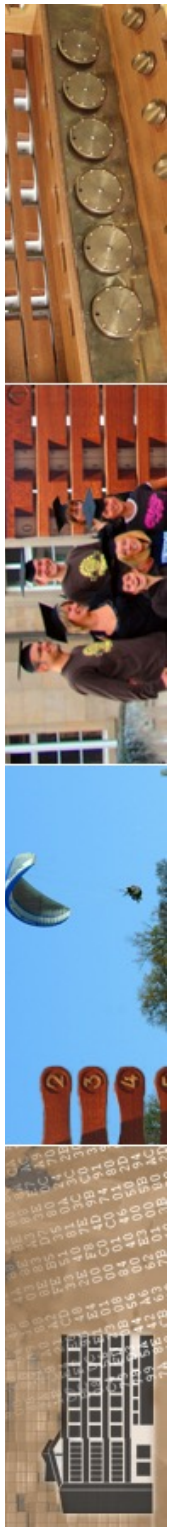
```
 9  use FCGI;    ###  PERL-Modul fuer FastCGI
10
11  $cnt = 0;
12
13  ### klassisches fastCGI-Schleife
14
15  while (FCGI::accept >= 0) {
16      $cnt++;
17      print <<"finish"
18  Content-type: text/html\n\n
19  <HTML>
20      <HEAD>
21          <TITLE>Webkompendium: fastCGI mit PERL</TITLE>
22          <link rel="stylesheet" type="text/css" href="/webkompend
23          <link rel="shortcut icon" href="/webkompendium/images/sp
24      </HEAD>
25      <BODY>
26          <CENTER>
27              <HR><H2>Kompendium der Web-Programmierung<BR>fastCGI
28              <HR><H3>Prozess von Server <CODE>$ENV{SERVER_NAME}</
29              <BR>Dies ist Anfrage Nr. $cnt<HR></H3>
```

Perl source file | nb char : 833 | Ln : 1 Col : 1 Sel : 0 | Dos\Windows ANSI INS



Performancevorteile durch fastCGI

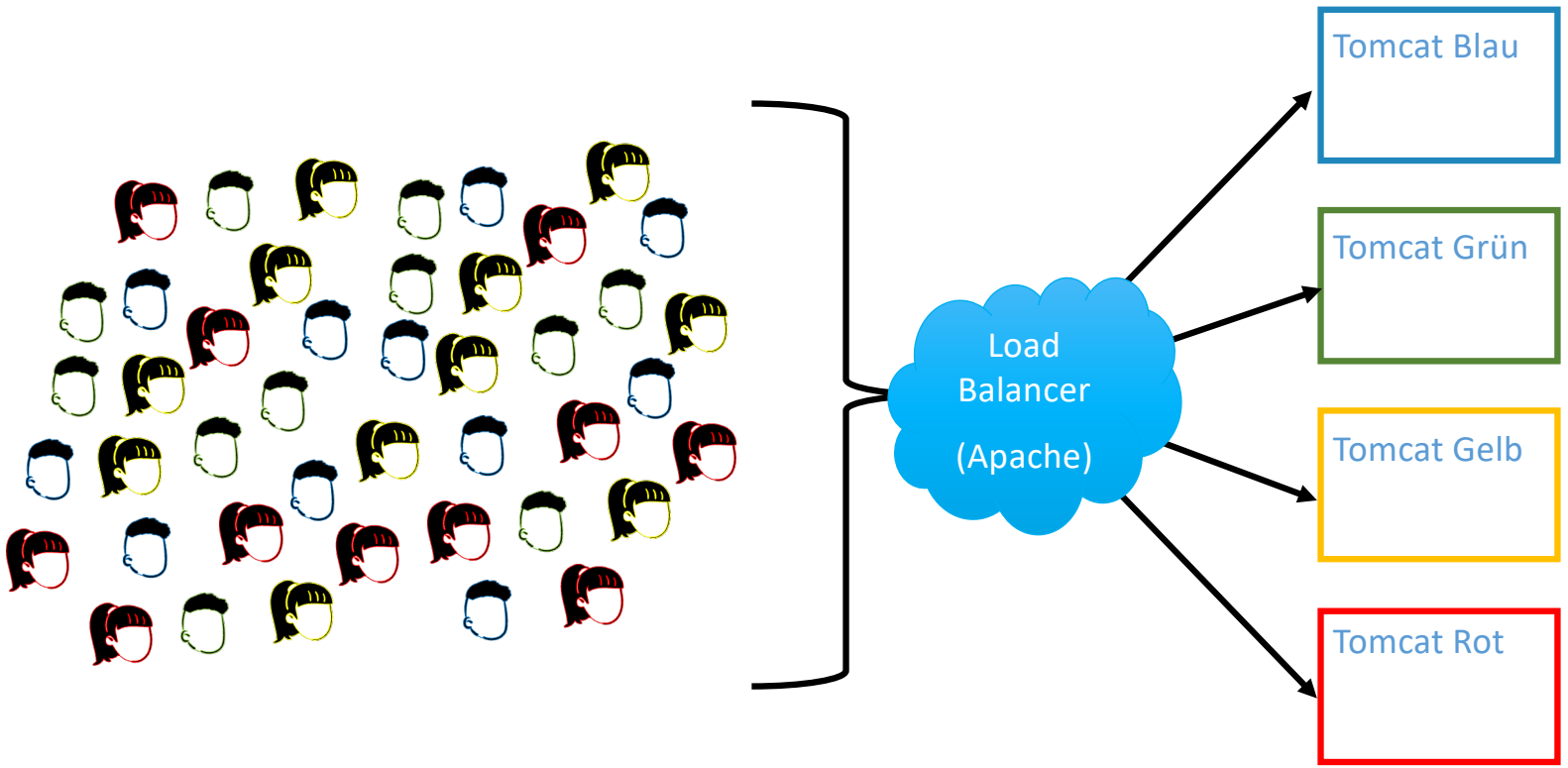
- Literaturangabe:
  - durch Einsatz von FastCGI werden Webanwendungen um bis zu Faktor 5 schneller
  - Beispiel (Herstellerangabe):
    - statische Datei: 21 ms + 0.19 ms / kB
    - fastCGI: 22 ms + 0.28 ms / kB
    - CGI: 59 ms + 0.37 ms / kB

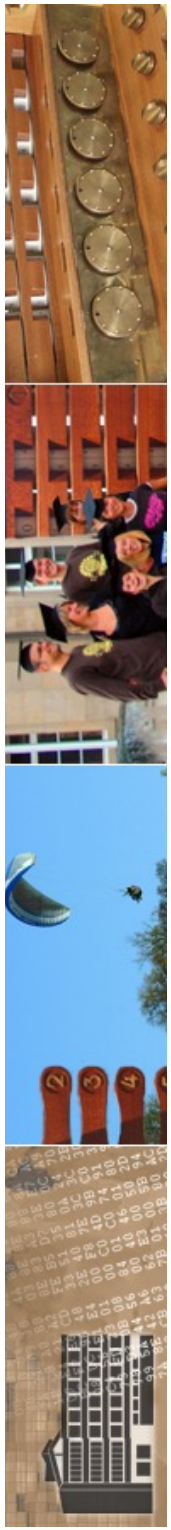


neue Einsatzbereiche

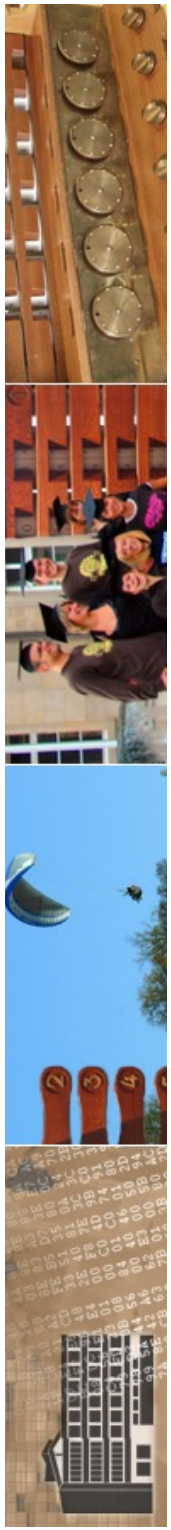
- fastCGI hatte 1995 wichtige neue Ansätze
- •diese sind von neueren Techniken übernommen und erweitert worden: Java Servlets, …
- **fastCGI ist aber für moderne Frameworks wie Ruby on Rails wieder sehr wichtig!**

Tomcat Clusterbetrieb



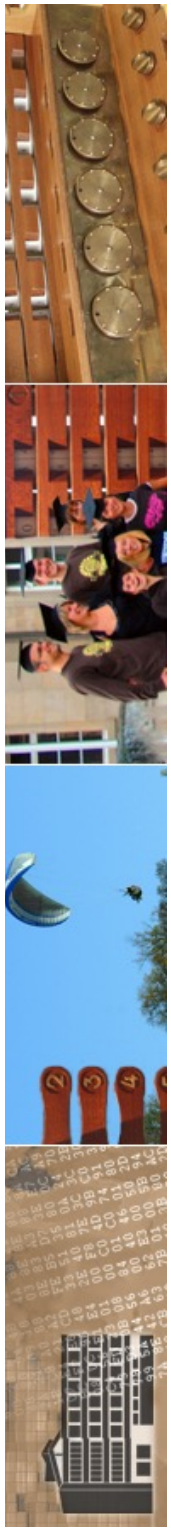






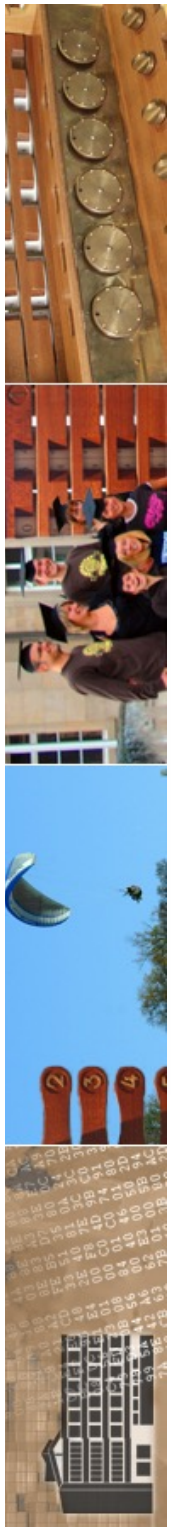
ServerSideIncludes (SSI)

- ServerSideIncludes (SSI) – wofür?
  - einfach(st)e Möglichkeit zur Erstellung dynamischer Seiten
  - vollständig serverbasiert
  - von den Möglichkeiten beschränkt, aber für vieles ausreichend
  - wird durch Standard-Apachemodul **include** ermöglicht
  - wird insb. von Apache2 intensiv selbst genutzt



SSI: Voraussetzungen

- der Webserver Apache muss mit dem Modul **mod_include** betrieben werden
- bekannte Möglichkeiten:
  - statische Integration in Apache-Kernel
  - dynamische Integration über DSO



Auszug aus httpd.conf

thomas — ssh ◂ -bash — 92×19

```
#LoadModule macro_module modules/mod_macro.so
#LoadModule dbd_module modules/mod_dbd.so
#LoadModule dumpio_module modules/mod_dumpio.so
#LoadModule buffer_module modules/mod_buffer.so
#LoadModule ratelimit_module modules/mod_ratelimit.so
LoadModule reqtimeout_module modules/mod_reqtimeout.so
#LoadModule ext_filter_module modules/mod_ext_filter.so
#LoadModule request_module modules/mod_request.so
LoadModule include_module modules/mod_include.so
LoadModule filter_module modules/mod_filter.so
#LoadModule substitute_module modules/mod_substitute.so
#LoadModule sed_module modules/mod_sed.so
#LoadModule deflate_module modules/mod_deflate.so
#LoadModule brotli_module modules/mod_brotli.so
LoadModule mime_module modules/mod_mime.so
#LoadModule ldap_module modules/mod_ldap.so
LoadModule log_config_module modules/mod_log_config.so
#LoadModule log_debug_module modules/mod_log_debug.so
                                                    106,1         18%
```



Apache Module mod_include

Description:	Server-parsed html documents (Server Side Includes)
Status:	Base
Module Identifier:	include_module
Source File:	mod_include.c

Summary

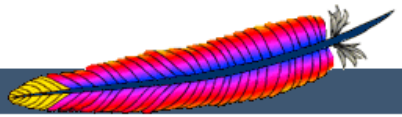
This module provides a filter which will process files before they are sent to the client. The processing is controlled by specially formatted SGML comments, referred to as *elements*. These elements allow conditional text, the inclusion of other files or programs, as well as the setting and printing of environment variables.

Enabling Server-Side Includes

Server Side Includes are implemented by the `INCLUDES` filter. If documents containing server-side include directives are given the extension .shtml, the following directives will make Apache parse them and assign the resulting document the mime type of `text/html`:

```
AddType text/html .shtml
AddOutputFilter INCLUDES .shtml
```





Modules | Directives | FAQ | Glossary | Sitemap

Apache HTTP Server Version 2.4

Apache > HTTP Server > Documentation > Version 2.4 > Modules

Apache Module mod_include

Available Languages: en | fr | ja

Description:	Server-parsed html documents (Server Side Includes)
Status:	Base
Module Identifier:	include_module
Source File:	mod_include.c

Summary

This module provides a filter which will process files before they are sent to the client. The processing is controlled by specially formatted SGML comments, referred to as *elements*. These elements allow conditional text, the inclusion of other files or programs, as well as the setting and printing of environment variables.

Enabling Server-Side Includes

Server Side Includes are implemented by the `INCLUDES` filter. If documents containing server-side include directives are given the extension .shtml, the following directives will make Apache parse them and assign the resulting document the mime type of `text/html`:

```
AddType text/html .shtml
AddOutputFilter INCLUDES .shtml
```

The following directive must be given for the directories containing the shtml files (typically in a `<Directory>` section, but this directive is also valid in `.htaccess` files if `AllowOverride` `Options` is set):

```
Options +Includes
```

For backwards compatibility, the `server-parsed` handler also activates the INCLUDES filter. As well, Apache will activate the INCLUDES filter for any document with mime type `text/x-server-parsed-html` or `text/x-server-parsed-html3` (and the resulting output will have the mime type `text/html`).

For more information, see our Tutorial on Server Side Includes.

PATH_INFO with Server Side Includes

Files processed for server-side includes no longer accept requests with `PATH_INFO` (trailing pathname information) by default. You can use the `AcceptPathInfo`

Topics

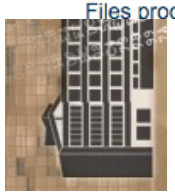
- Enabling Server-Side Includes
- PATH_INFO with Server Side Includes
- Available Elements
- Include Variables
- Variable Substitution
- Flow Control Elements
- Legacy expression syntax

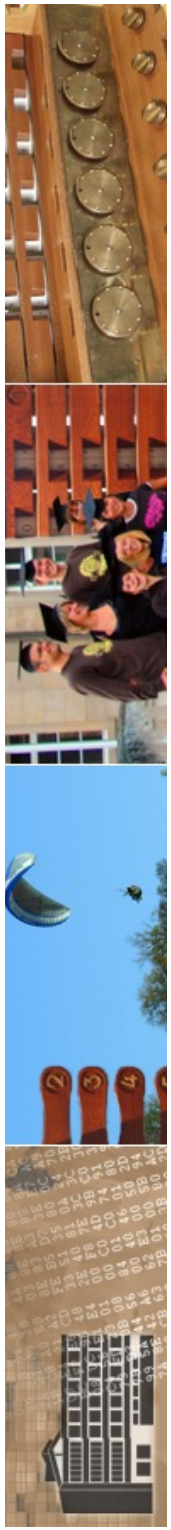
Directives

SSIEndTag
SSIErrorMsg
SSIETag
SSILastModified
SSILegacyExprParser
SSIStartTag
SSITimeFormat
SSIUndefinedEcho
XBitHack

See also

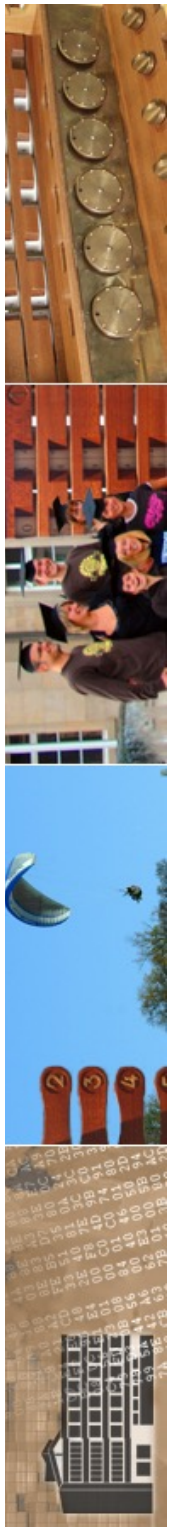
- `Options`
- `AcceptPathInfo`
- Filters
- SSI Tutorial
- Comments





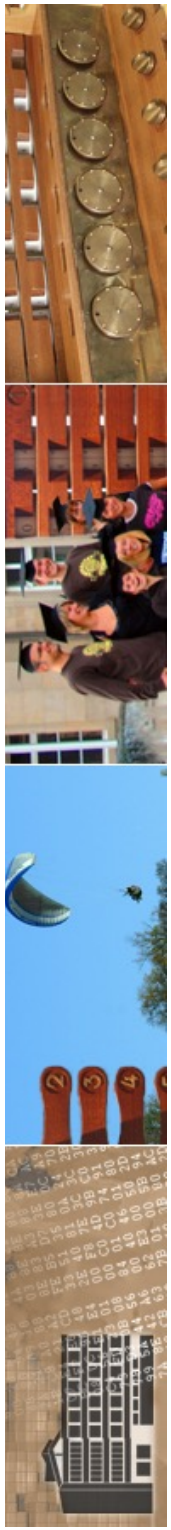
was sind SSI

- HTML-Anweisungen
- werden serverseitig bei Auslieferung der Seite ausgewertet
- dabei wird kein CGI-Prozeß etc. gestartet
- eine schnellere Alternative zu CGI
  - „SSI is a great way to add small pieces of information, such as the current time. But if a majority of your page is being generated at the time that it is served, you need to look for some other solution.“
- eine **sichere** Option für Dynamik im Web



SSI: Beispiele für Einsatz

- Beispiele:
  - Last-Modified des Dokuments
    - auch mit JavaScript möglich – aber mit SSI *clientunabhängig* (Format) und unbedenklich für Client
  - Einbinden einer Datei (etwa footer.html)
    - auch mit PHP möglich, aber größerer Aufwand und nicht mehr Datei-Endung ".html"

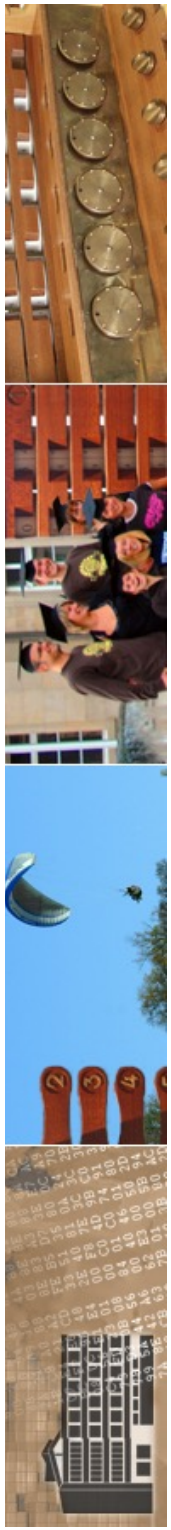


SSI: Voraussetzungen I

- in Apache-Konfiguration http.conf muß für ein Verzeichnis des Webservers und alle Unterverzeichnisse SSI erlaubt werden
  - Erlauben von SSI im Verzeichnis:
    - Direktive
      `Options +Includes`
    - Konkretes Beispiel

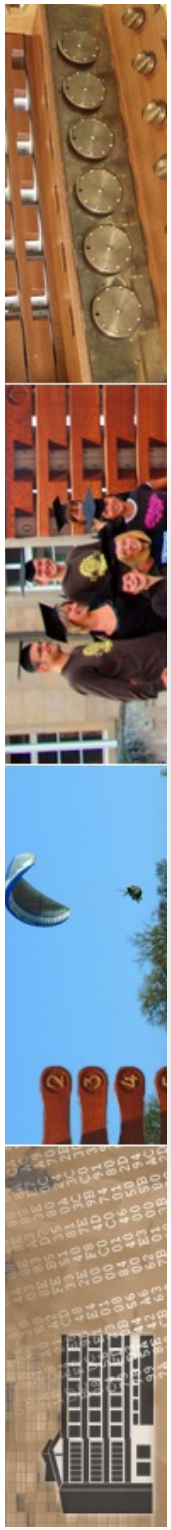
```
<Directory "d:/www/myssi">
        Options +Includes
</Directory>
```

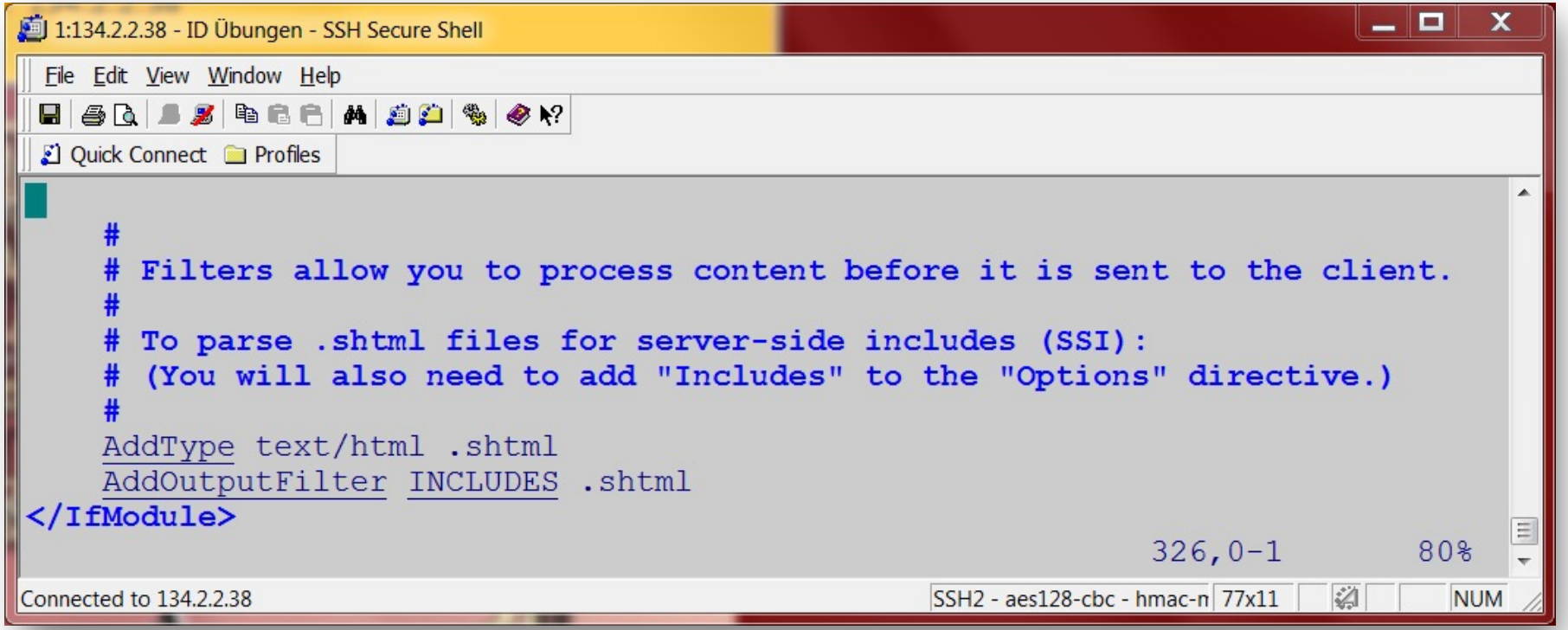
  - Alternative:
    - **`Options all`**

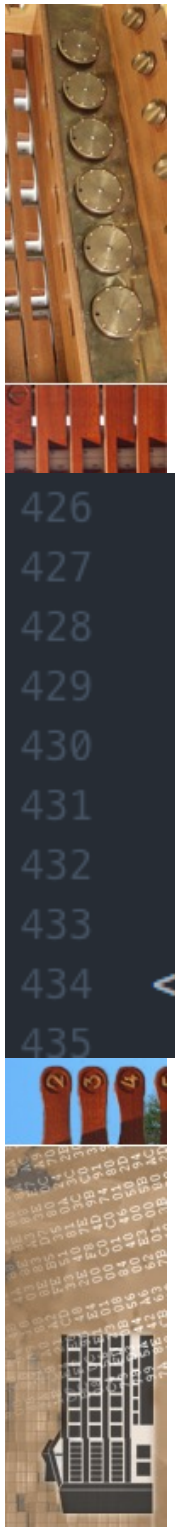


SSI: Voraussetzungen II

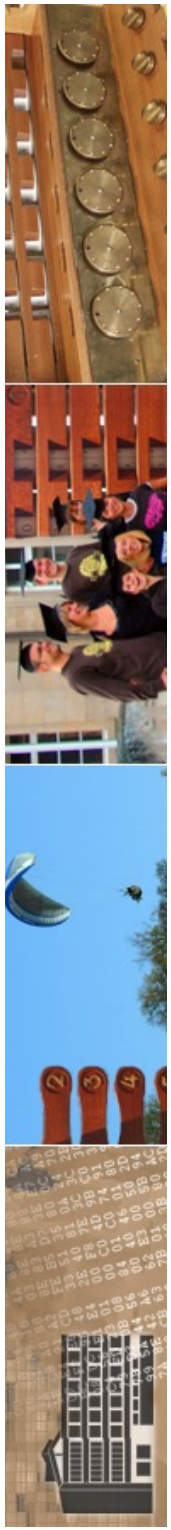
- Kennzeichnung einer HTML-Datei als SSI in http.conf:
  - Dateiendung shtml und Konfiguration
    - **`AddType text/html .shtml`**
      `AddHandler server-parsed .shtml`
  - oder *(nur* auf Unix): mit Direktive **`XBitHack on`** werden alle ausführbaren Dateien als SSI betrachtet (x-bit gesetzt)
    - **`chmod +x pagename.html`**
    - Voraussetung: eine weitere Direktive **`XBitHack`**




1:134.2.2.38 - ID Übungen - SSH Secure Shell
File Edit View Window Help
Quick Connect Profiles
#
Filters allow you to process content before it is sent to the client.
#
To parse .shtml files for server-side includes (SSI):
(You will also need to add "Includes" to the "Options" directive.)
#
AddType text/html .shtml
AddOutputFilter INCLUDES .shtml
</IfModule>
326,0-1 80%
Connected to 134.2.2.38
SSH2 - aes128-cbc - hmac-n 77x11
NUM



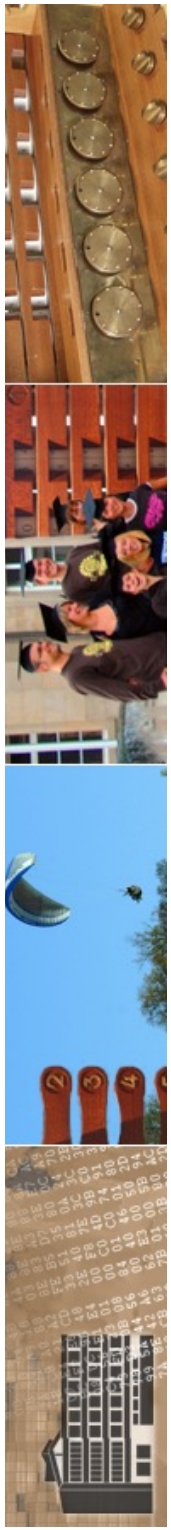
```
426       #
427       # Filters allow you to process content before it is sent to the client.
428       #
429       # To parse .shtml files for server-side includes (SSI):
430       # (You will also need to add "Includes" to the "Options" directive.)
431       #
432       #AddType text/html .shtml
433       #AddOutputFilter INCLUDES .shtml
434   </IfModule>
435
```



Syntax von SSI

- Grundaufbau:
  - SSI-Befehle selbst sind **HTML-Kommentare!**
    - `<!--#element attribute="value" ... -->`
  - Beispiel: Ausgabe der Serverzeit:

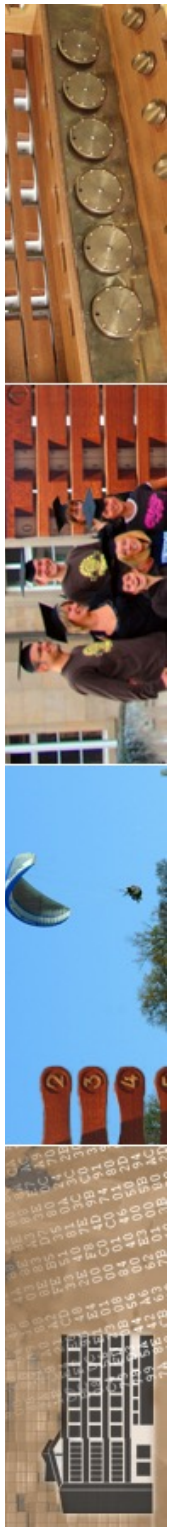
```
<!--#echo var="DATE_LOCAL" -->
```



Beispiel

- wir erweitern unsere Startseite auf dem eigenen Apache
  - als erstes wollen wir die Serverzeit und Namen ausgeben
  - als zweites die Zeit der Aktualisierung der Seite ausgeben
    (über SSI, nicht wie häufig über JavaScript)
  - und letztlich noch eine footer-Datei einfügen

© 2026 Universität Tübingen

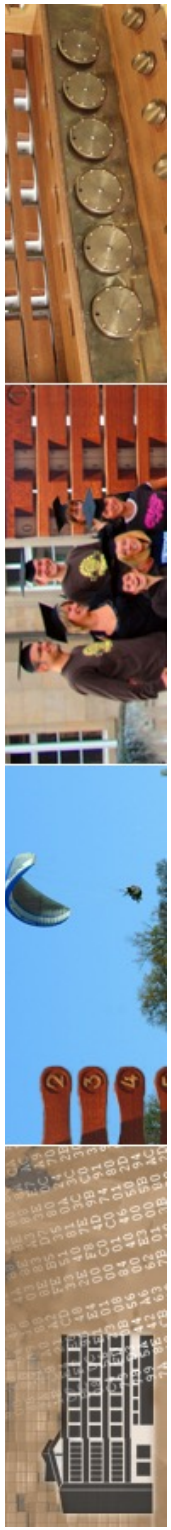


Ausgabe der Serverzeit und mehr

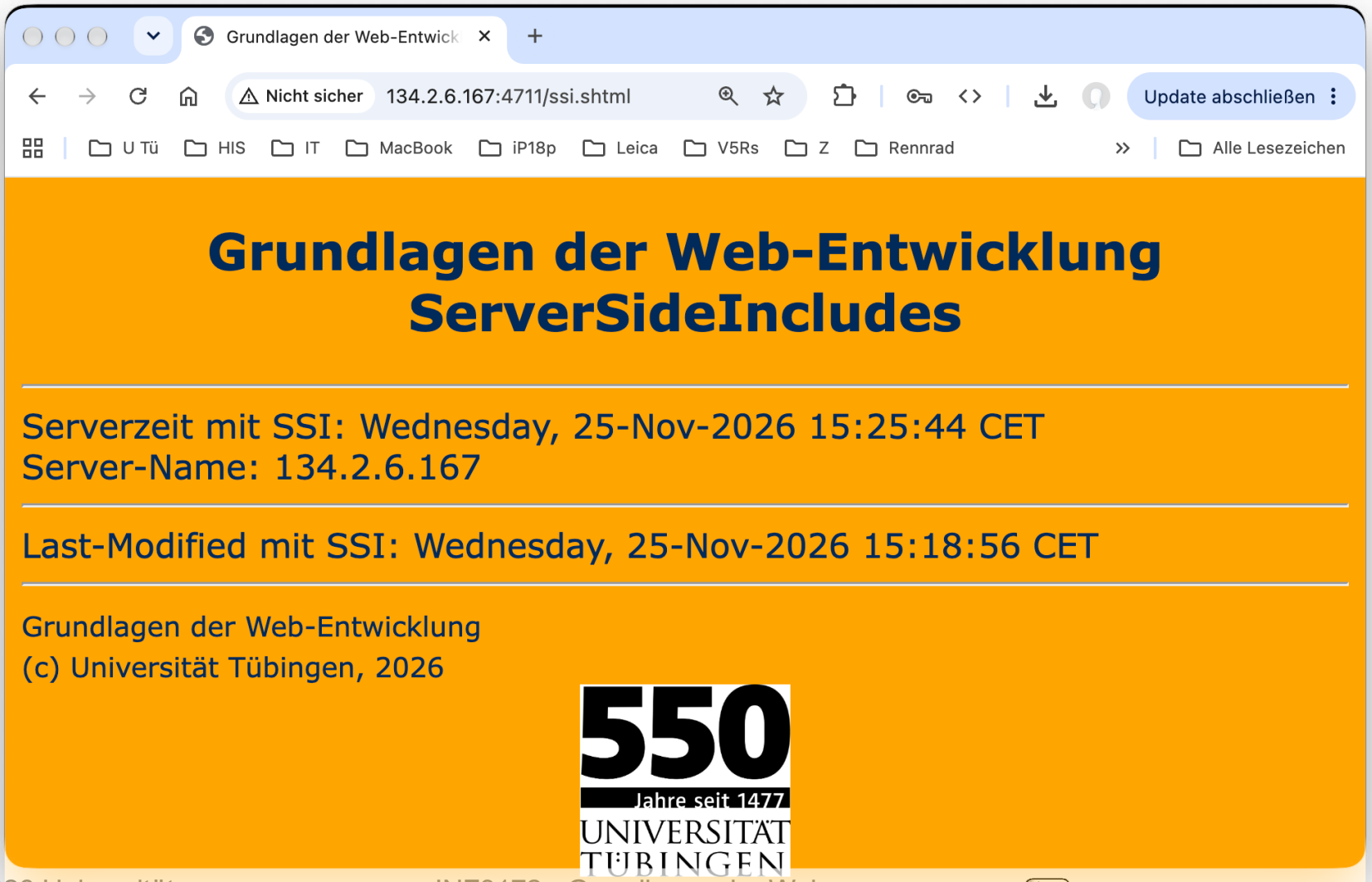
thomas — ssh ◂ -bash — 92×32

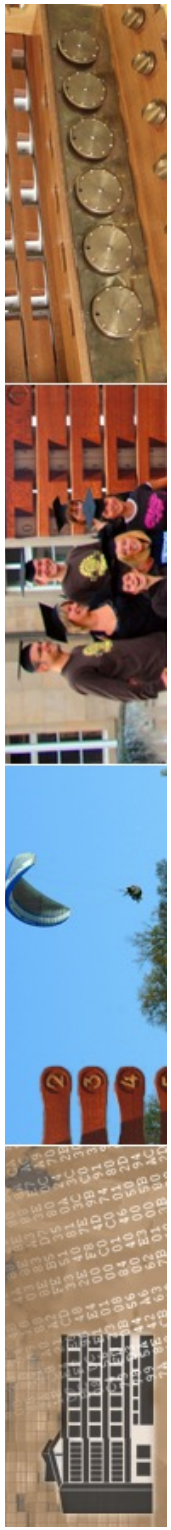
/Users/thomas — ssh ◂ -bash /Users/thomas — ssh ◂ -bash

```
zrvwa01@infodienste =>
zrvwa01@infodienste => more ssi.shtml
<!DOCTYPE html>
<html>
  <head>
      <meta http-equiv="Content-type" content="text/html;charset=UTF-8" />
      <meta name="author" content="Thomas Walter" />
          <link rel="stylesheet" type="text/css" media="screen" href="css/mycss.css" />
      <title>Grundlagen der Web-Entwicklung - SSI</title>
  </head>
  <body bgcolor="orange">
      <center>
           <h2>Grundlagen der Web-Entwicklung<br />ServerSideIncludes</h2>
      </center>

                <hr />

                Serverzeit mit SSI: <!--#echo var="DATE_LOCAL"-->
                <BR />
                Server-Name: <!--#echo var="SERVER_NAME"-->

                <hr />

                Last-Modified mit SSI: <!--#flastmod file="ssi.shtml" -->
                <hr />

                <!--#include virtual="footer.html"-->
   </body>

</html>
```



Ergebnis

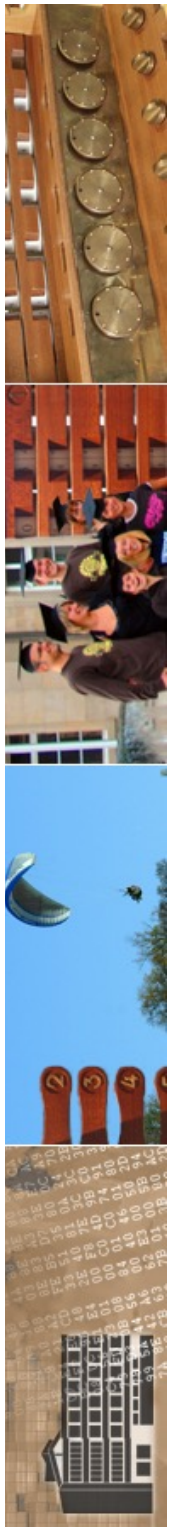

Grundlagen der Web-Entwick
Nicht sicher
134.2.6.167:4711/ssi.shtml
Update abschließen
U Tü
HIS
IT
MacBook
iP18p
Leica
V5Rs
Z
Rennrad
Alle Lesezeichen
Grundlagen der Web-Entwicklung
ServerSideIncludes
Serverzeit mit SSI: Wednesday, 25-Nov-2026 15:25:44 CET
Server-Name: 134.2.6.167
Last-Modified mit SSI: Wednesday, 25-Nov-2026 15:18:56 CET
Grundlagen der Web-Entwicklung
(c) Universität Tübingen, 2026
550
Jahre seit 1477
UNIVERSITÄT
TÜBINGEN



Zeitformate

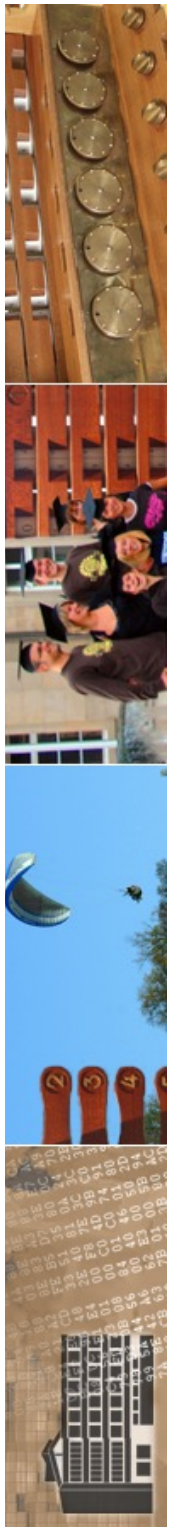
- SSI stellt einfache Möglichkeit zur Zeitformatierung bereit:
  - `<!--#config timefmt="%d. %b %Y, %H.%mh" -->`
    `Today is <!--#echo var="DATE_LOCAL" -->`
  - Bedeutung:
    d: Tag, Y: Jahr b: Monat (Kurzform) H: Stunde, m: Minute

© 2026 Universität Tübingen



weitere Konfigurationen

- mit **`config`** kann neben dem Zeitformat noch festgelegt werden:
  - **`ermsg`** : die Fehlermeldung, die der Client erhält
  - **`sizefmt`** : bei Angabe einer Dateigröße das Format der Anzeige (Kb oder Mb)



last modified

- Ersetzen des JavaScripts für “LastModified” durch SSI:

```
<!--#flastmod file="index.html" -->
```

...damit LastModified auch *ohne* JavaScript!

  - die Zeitformatierung gilt auch hier

D:\users\thomas\Eberhardina\InformationsDienste\20102\Source\SSI\ssi.shtml - Notepad++

Datei Bearbeiten Suchen Ansicht Kodierung Sprachen Einstellungen Makro Ausführen TextFX Erweiterungen Fenster ? X

SimpleHttpd.java | ssi.shtml

```
15
16  Serverzeit mit SSI: <!--#echo var="DATE_LOCAL"-->
17  <BR />
18  Server-Name: <!--#echo var="SERVER_NAME"-->
19
20  <hr />
21
22  Last-Modified mit SSI: <!--#flastmod file="ssi.shtml" -->
23  <br />
24  Last-Modified mit JavaScript:
25  <script type="text/javascript">
26      document.write(document.lastModified);
27  </script>
28
```

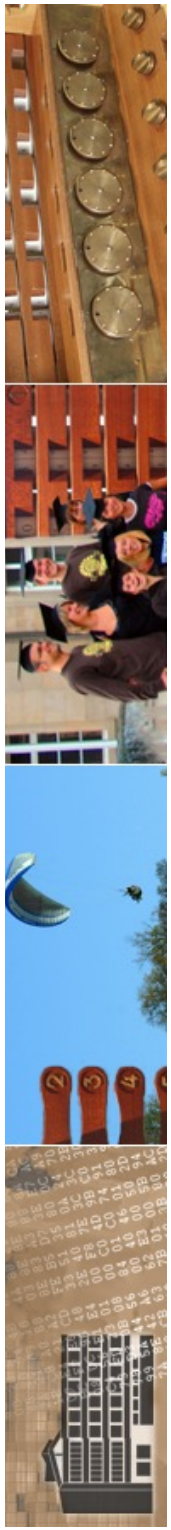
Hyper Text Markup Language file | 937 chars | 969 bytes | 33 lines | Ln : 2 | Col : 14 | Sel : 0 (0 bytes) in 0 ranges | UNIX | ANSI | INS

Grundlagen der Web-Entwicklung
ServerSideIncludes

Serverzeit mit SSI: Wednesday, 25-Nov-2026 13:00:28 CET
Server-Name: 134.2.6.167

Last-Modified mit SSI: Wednesday, 25-Nov-2026 12:57:49 CET

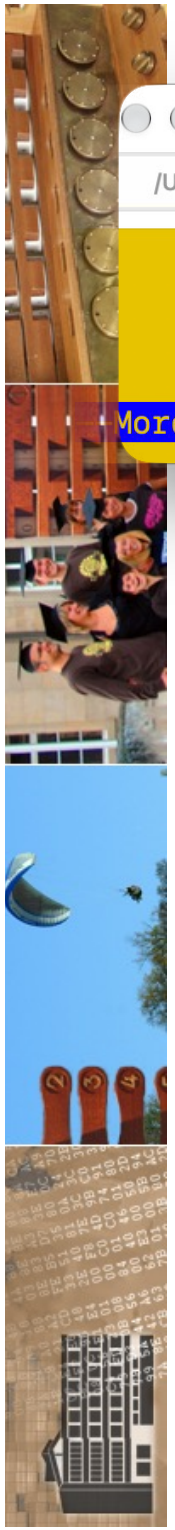
© 2026 Universität Tübingen



Einfügen einer footer-Datei

- alle Seiten sollen einheutlichen Abschluss bekommen
- es wäre sehr ungeschickt, diesen in jeder einzelnen HTML-Datei zu codieren
- -> Einfügen einer HTML-Datei

```
<!--#include virtual="/footer.html" -->
```



thomas — ssh ◂ -bash — 76×5

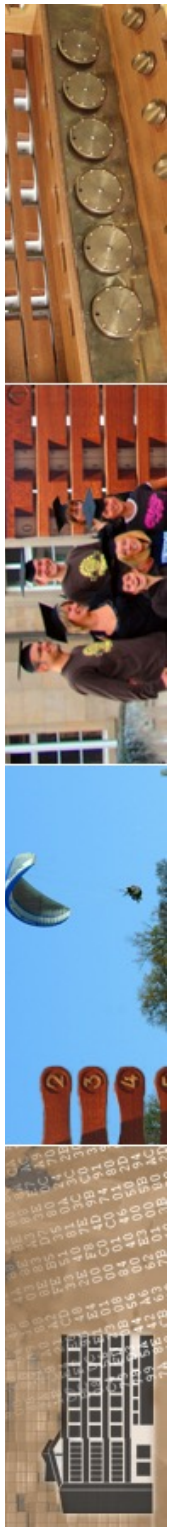
/Users/thomas — ssh ◂ -bash | ~ — ssh ◂ -bash | ~ — ssh ◂ -bash

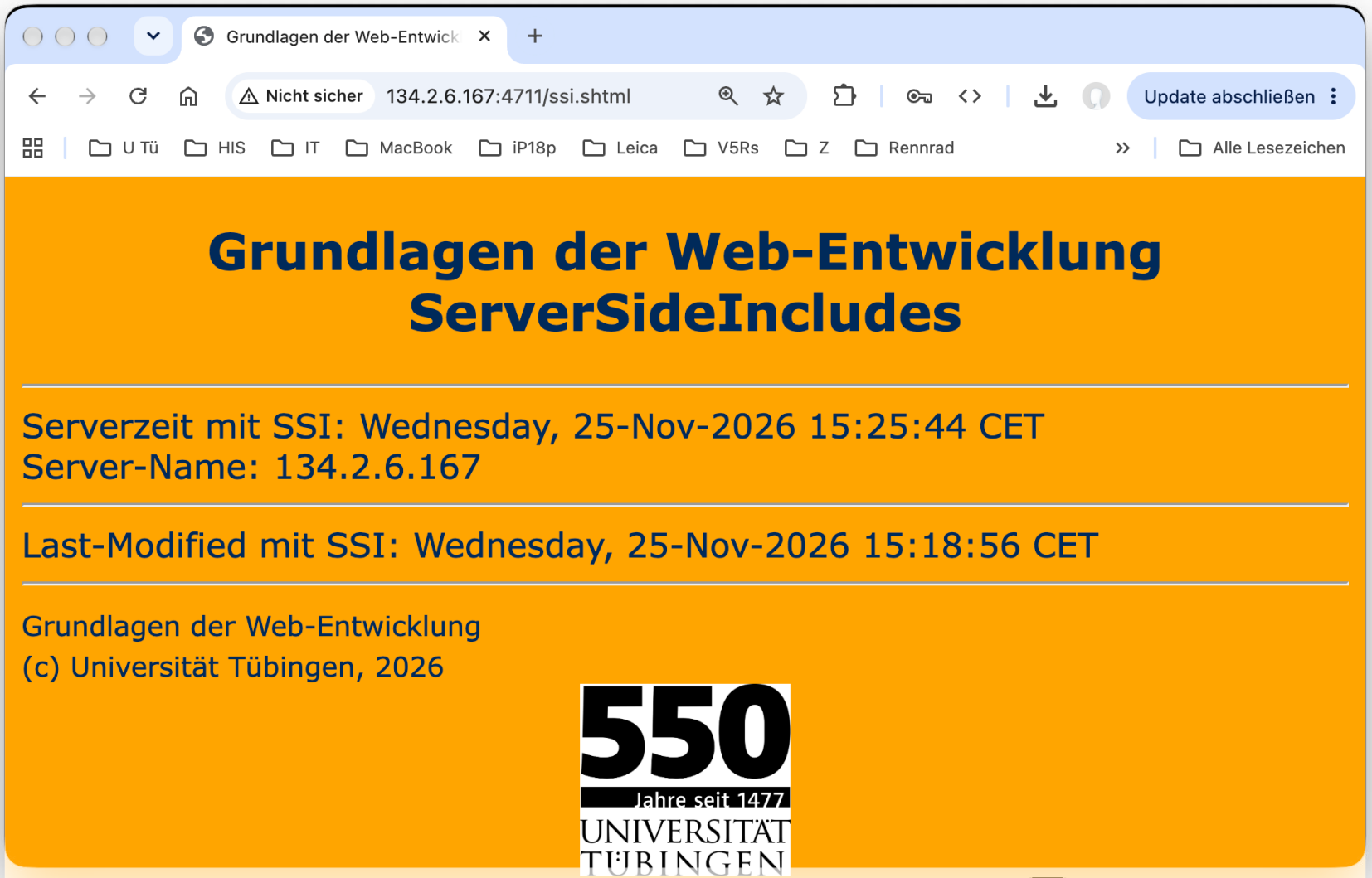
```
            Last-Modified mit SSI: <!--#flastmod file="ssi.shtml" -->
            <hr />

            <!--#include virtual="footer.html"-->
--More--(97%)
```

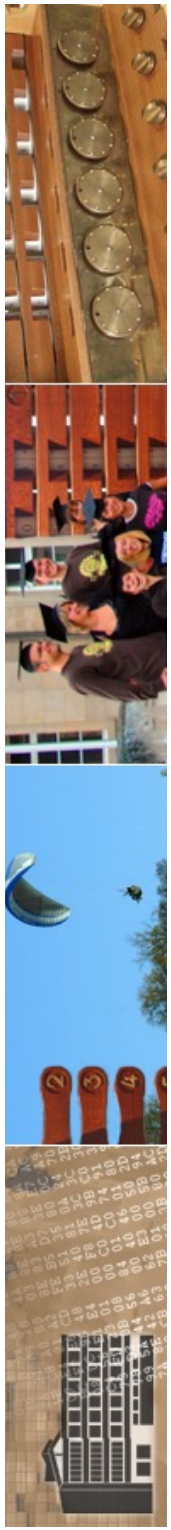
thomas — ssh ◂ -bash — 64×12

/Users/thomas — ssh ◂ -bash | /Users/thomas — ssh ◂ -bash

```
zrvwa01@infodienste =>
zrvwa01@infodienste => more footer.html
<font size="-1">
        Grundlagen der Web-Entwicklung
        <BR />
        (c) Universit&auml;t T&uuml;bingen, 2026
  <center>
          <img src="/css/JL550_Schwarz_300dpi.jpg" width="100">
  </center>
</font>
zrvwa01@infodienste =>
zrvwa01@infodienste =>
```

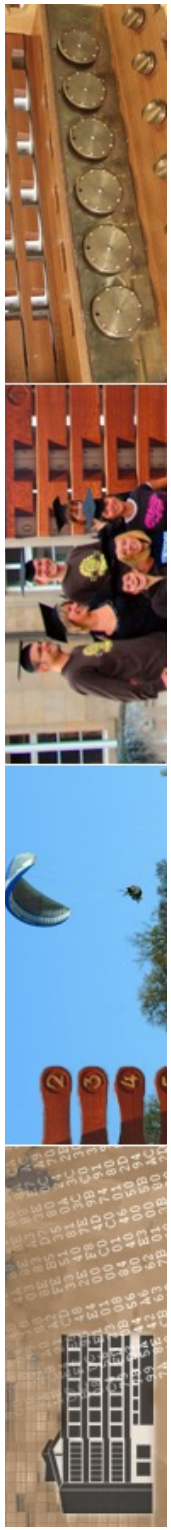



Grundlagen der Web-Entwick
Nicht sicher
134.2.6.167:4711/ssi.shtml
Update abschließen
U Tü
HIS
IT
MacBook
iP18p
Leica
V5Rs
Z
Rennrad
Alle Lesezeichen
Grundlagen der Web-Entwicklung
ServerSideIncludes
Serverzeit mit SSI: Wednesday, 25-Nov-2026 15:25:44 CET
Server-Name: 134.2.6.167
Last-Modified mit SSI: Wednesday, 25-Nov-2026 15:18:56 CET
Grundlagen der Web-Entwicklung
(c) Universität Tübingen, 2026
550
Jahre seit 1477
UNIVERSITÄT
TÜBINGEN



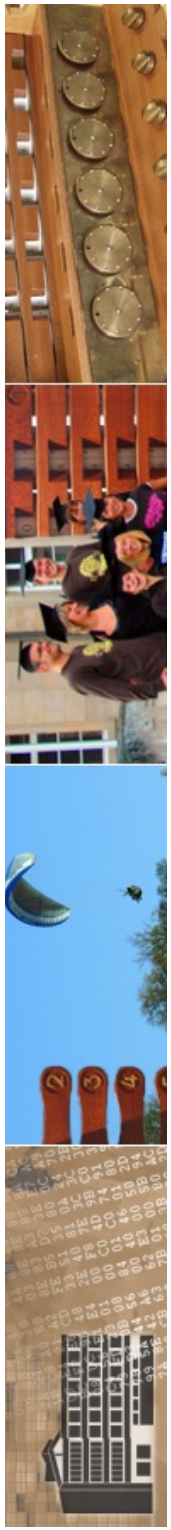
Kombination

- in der »includeten«-Datei kann selbst auch wieder SSI-Code enthalten sein, der dann entprechend ausgeführt wird
- »geschachtelte« SSIs sind möglich!
  - Beispiel: Serverzeit in footer.html



Ausführen eines Programms

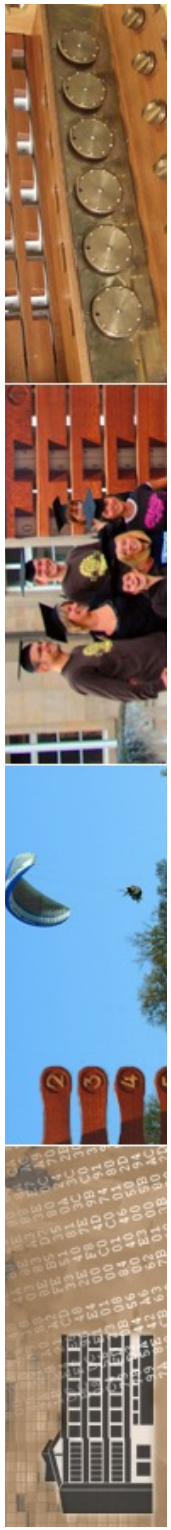
- SSI kann direkt ein ausführbares Programm auf dem Server starten
  - Anweisung
  - `<!--#exec cmd="ls -alp"  -->`
- damit etwa auch Python-Scripte mit Shebang-Zeile einbindbar



Ausführen eines cgi-Scriptes

- mittels **`#exec cmd=`**… wird ein auf der Shell ausführbares Programm ausgeführt

- Alternative:
  - **`#exec cgi=`**…
    führt dierkt ein vollwertiges CGI aus

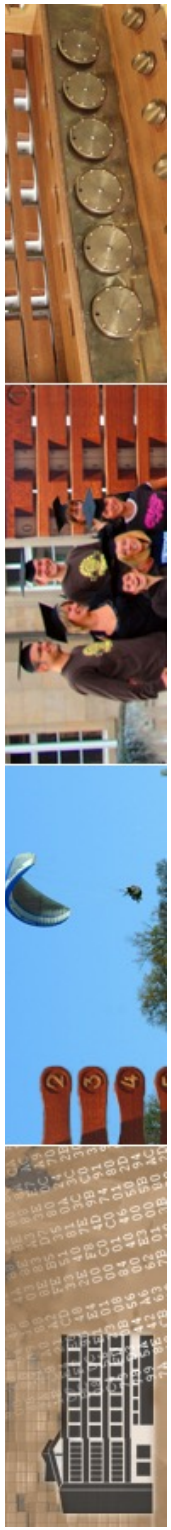


Was gibt es noch?

- noch ein paar »Kleinigkeiten«:
  - einfache (typenlose) Variablen

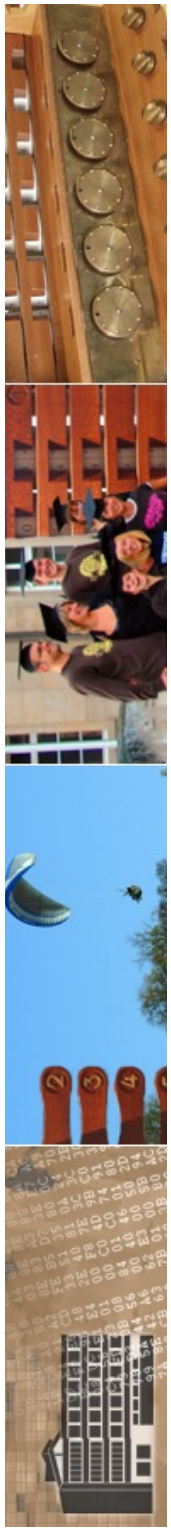
    ```
    <!--#set var="cost"  value="€100"  -->
    ```

  - Vergleichsoperationen (if - else)
  - Abfrage Browser-Typ
  - Ausgabe der Größe einer Datei: fsize



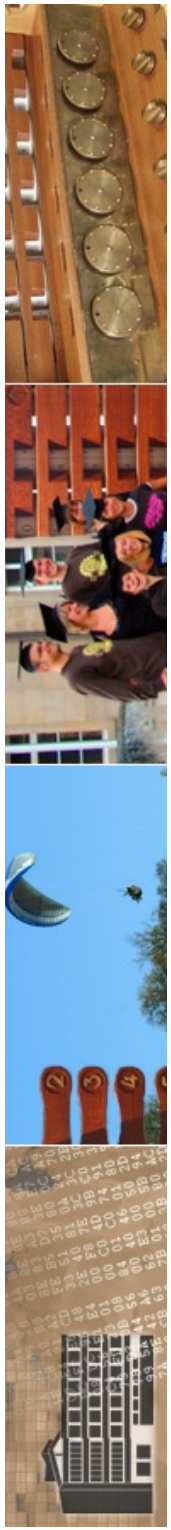
SSI-Standardvariablen

- `DATE_GMT`
- `DATE_LOCAL`
- `DOCUMENT_NAME`
- `DOCUMENT_URI`
- `LAST_MODIFIED`
- `HTTP_USER_AGENT`
- `HTTP_REFERER`
- `SERVER_NAME`
- `SERVER_SOFTWARE`
- `REMOTE_ADDR`



SSI Commands

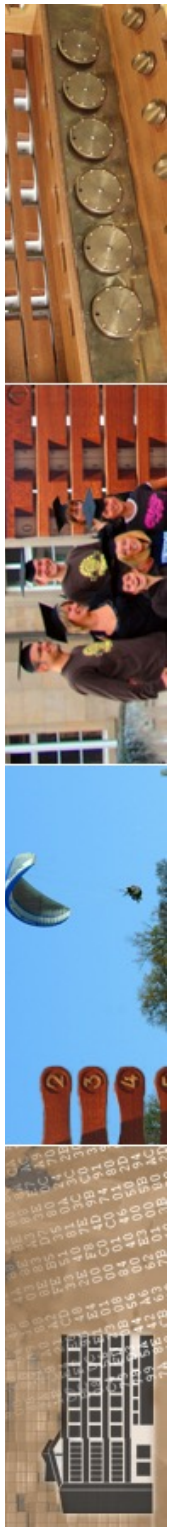
- `include`
- `config`
- `echo`
- `fsize`
- `flastmod`
- `exec`
- `set`



Beispielanwendung

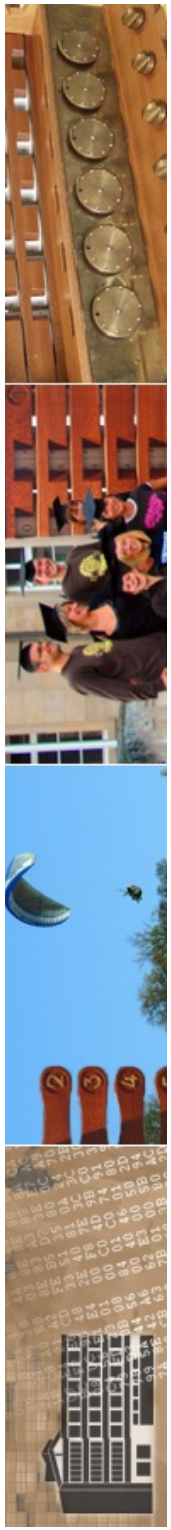
- Site mit beim Neuladen austauschenden Bildern (Zufallsazswahl)
  - über SSI, welches PERL-Script zur Zufallsberechnung der Bilder enthält
- modularer Seitenaufbau
  - Auslagerung des Menüs in Datei, die von SSI importiert wird

© 2026 Universität Tübingen



eingeschränkte Ausführung

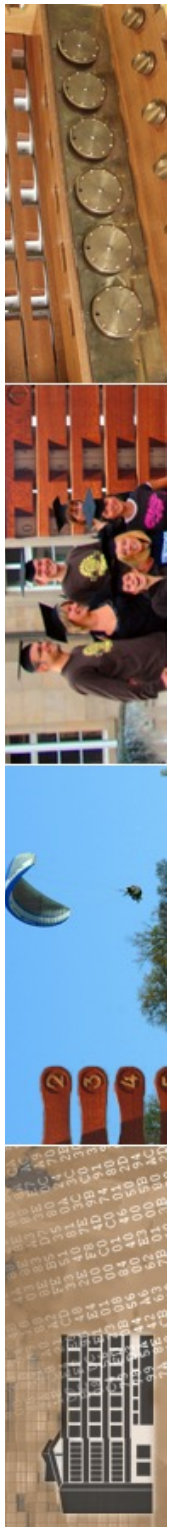
- besondere Apache-Direktive
  - **`IncludesNOEXEC`**
- diese erlaubt SSI *ohne* **`#include`** und **`#exec`**



Zusammenfassung SSI

- „SSI is certainly not a replacement for CGI, or other technologies used for generating dynamic web pages. But it is a great way to add small amounts of dynamic content to pages, without doing a lot of extra work.“

© 2026 Universität Tübingen



…und nun…

- haben wir zwei spezielle Apache-Module genauer kennen gelernt: fastCGI und Includes



- als nächstes: Content-Management-Systeme, insbesondere TYPO3

© 2026 Universität Tübingen