

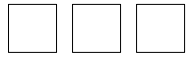


Web Engineering

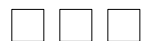
Übung CGI mit Perl

Stefan Schlott
Universität Ulm

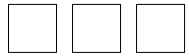
Common Gateway Interface



- Dynamische Generierung von Seiten mit externen Programmen
- Einfache Schnittstelle
- Externes Programm in bel. Sprache
- Parameter in Umgebungsvariablen

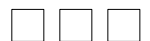


CGI-Ausführung konfigurieren

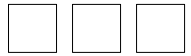


- Default: Server liest nur Daten, keine Ausführung
- Voraussetzung: mod_cgi (Standardmodul)
- Separater Directory-Eintrag, der Ausführung zuläßt, anlegen:

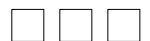
```
<Directory "/home/ss7/webeng/apache/htdocs/mycgi">  
  AllowOverride AuthConfig  
  Options ExecCGI  
# AddType application/x-http-cgi .pl .sh .cgi  
  AddHandler cgi-script .cgi .pl .sh  
</Directory>
```



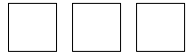
CGI-Ausführung konfigurieren



- Ausführung von CGIs im Verzeichnis, ...
 - das mit **ScriptAlias** festgelegt wurde
 - dessen **Option ExecCGI** gesetzt ist
- Es werden nur Programme mit bestimmten Endungen ausgeführt. Festlegen mit:
 - `AddHandler cgi-script .pl .sh .cgi` oder
 - `AddType application/x-httpd-cgi .pl .sh .cgi`
- Programmdateien müssen zusätzlich ausführbar sein (x-Flag gesetzt)

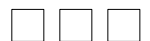


Beispiel 1: Echo-Shellskript

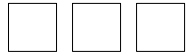


- echo.cgi: Ausgabe der Umgebungsvariablen

```
#!/bin/bash
echo Content-Type: text/plain
echo
echo Benutzer-ID:
echo -----
id
echo
echo Umgebungsvariablen:
echo -----
set
echo
echo Ausgabe von stdin:
echo -----
cat -
```



Beispiel 1: Echo-Shellskript



<http://.../mycgi/echo.cgi?zeige=news&dummy=1>

Benutzer-ID:

uid=10551(ss7) gid=10000(users) groups=10000(users)

Umgebungsvariablen:

DOCUMENT_ROOT=/home/ss7/webeng/apache/htdocs

GATEWAY_INTERFACE=CGI/1.1

HTTP_ACCEPT=\$'text/xml,application/xml, (...)'

HTTP_ACCEPT_CHARSET=\$'ISO-8859-1,utf-8;q=0.7,*;q=0.7'

HTTP_ACCEPT_ENCODING=gzip,deflate

HTTP_ACCEPT_LANGUAGE=\$'en-us,en;q=0.5'

HTTP_HOST=wsl01.informatik.uni-ulm.de:18080

HTTP_USER_AGENT=\$'Mozilla/5.0 (...)'

QUERY_STRING=\$'zeige=news&dummy=1'

REMOTE_ADDR=134.60.70.39

REQUEST_URI=\$'/mycgi/echo.cgi?zeige=news&dummy=1'

SCRIPT_FILENAME=/home/ss7/webeng/apache/htdocs/mycgi/echo.cgi

(...)

Programm läuft
mit Rechten
des Servers

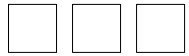
Informationen
über Browser-
fähigkeiten

Parameter der
Anfrage

Zusätzl.
Informationen
vom Server



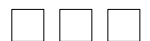
Beispiel 1: Echo-Shellskript



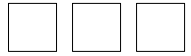
- Bei Authentisierung über .htaccess:
Weitere Umgebungsvariablen

```
AUTH_TYPE=Digest  
REMOTE_USER=hugo
```

- Unterschiedliche Reaktionen je nach Nutzer möglich!



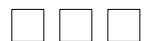
Beispiel 2: Formular



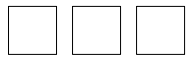
```
<html>
<head><title>Formulartest</title></head>
<body>
<form action="mycgi/echo.cgi" method="post">
Name: <input name="name" type="text" size="30" maxlength="30" /><br />
EMail: <input name="email" type="text" size="30" maxlength="30" /><br />
Kommentar:<br />
<textarea name="kommentar" cols="70" rows="7">
Ihr Kommentar
</textarea>
<br />
<input type="submit" value="Absenden" />
</form>
</body>
</html>
```

Ausgabe von stdin:

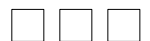
```
name=sadffdsa&email=sadsdf&kommentar=Ihr+Kommentar%0D%0A
```



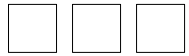
CGIs mit Perl



- perl – Practical Extraction and Report Language
- Einfach zu bedienende Skriptsprache
- Verwendet viele bekannte Konzepte, z.B. aus sed (regular expressions), awk, ...
- Module für (fast) jeden Zweck verfügbar
- Very quick, very dirty (abhängig vom Programmierer)

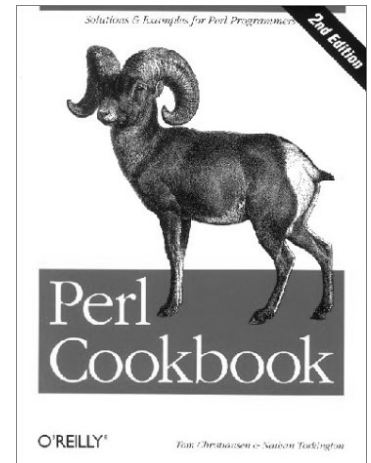


CGIs mit Perl

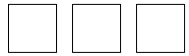


Hilfe zu Perl:

- `man perltoc` – Übersicht über man-Pages
- `man perlfunc` – Eingebaute Funktionen
- `perldoc modulname` – Hilfetext aus Modul anzeigen
- <http://www.cpan.org/> - Comprehensive Perl Archive Network
- Das Handbuch: „Perl Cookbook“



CGIs mit Perl



- CGI-Modul für Perl

- Bei CPAN verfügbar

http://theoryx5.uwinnipeg.ca/mod_perl/cpan-search?distinfo=1013

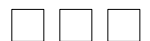
- Viele Tutorials, z.B.

<http://stein.cshl.org/WWW/software/CGI/>

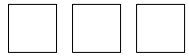
- HTML-Header erzeugen

- Funktionen zum Handeln von Parametern und Feldinhalten

- Funktionen für die Ausgabe der HTML-Elemente



Beispiel 3: Parameter in Perl



■ params.pl: Parameter ausgeben

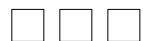
```
#!/usr/bin/perl

use CGI qw(:standard);

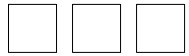
print header;
print start_html('Testformular');
print h1('Übergebene Parameter');
@rows = th(["Name", "Wert"]);
foreach $n (param()) {
    push(@rows, td([$n, param($n)]));
}
print table({-border=>undef, -width=>'25%'}, Tr(\@rows));
print end_html;
```

Import der Funktionen in
den Hauptnamensraum

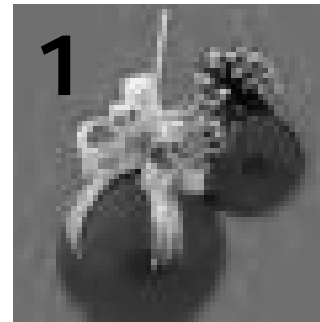
- ...params() übergibt sowohl URL-Parameter als auch Inhalte von Formularfeldern



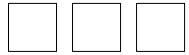
Beispiel 4: Adventskalender



- 24 Türchen zum aufmachen
- Beliebige Türen auf- und zumachen
- Zustand des Kalenders über Sitzungen hinweg sichern
- Kein Beschummeln!
- Idee: Darstellung des Zustandes als Bitfolge (niederwertigstes Bit: 1. Dezember)



Beispiel 4: Adventskalender



```
#!/usr/bin/perl

use CGI;

$q = new CGI;
$offen = 0; # Bitmaske für die Tage
($sec,$min,$hour,$mday,$mon,$year,$wday,$yday,$isdst) = localtime(time);

# Gibt es ein Cookie für den Kalender?
$offen_alt=$q->cookie("AdvKal");
if ($offen_alt=~ /^[0-9]+$/) {
    $offen = $offen_alt;
}

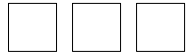
# Gibt es einen Parameter für den Kalender?
$offen_alt=$q->param("offen");
if ($offen_alt=~ /^[0-9]+$/) {
    $offen = $offen_alt;
}

...

```



Beispiel 4: Adventskalender



```
# Schummeln verbieten
if ($mon<11) {
    $offen=0;
}
if ($mon==11) {
    for ($i=$mday; $i<24; $i++) {
        $offen&=~(1<<$i);
    }
}

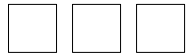
# Cookie erzeugen
$cookie=$q->cookie(-name=>"AdvKal",-value=>$offen,-expires=>" +25d",
    -path=>" /",-domain=>".uni-ulm.de");

# Ausgabe
print $q->header(-cookie=>$cookie)."\n";
print $q->start_html("Advent, Advent, ein Lichtlein brennt")."\n";
print $q->h2("Adventskalender")."\n";
print "<table>\n";
```

...



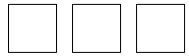
Beispiel 4: Adventskalender



```
for ($zeile=0; $zeile<4; $zeile++) {
    print " <tr>\n";
    for ($spalte=0; $spalte<6; $spalte++) {
        $tag=$zeile*6+$spalte; $zelle="";
        if ($tag+1<=$mday) {
            $zelle="<a href=\"\".$q->url."?offen=".(($offen^(1<<$tag))."\">";
        }
        if (($offen&(1<<$tag))>0) {
            $zelle.="<img src=\"auf\".$tag.\".jpg\" />";
        } else {
            $zelle.="<img src=\"zu\".$tag.\".jpg\" />";
        }
        if ($tag+1<=$mday) {
            $zelle.="</a>";
        }
        print " <td>".$zelle."</td>\n";
    }
    print "</tr>\n";
}
print "</table>\n„.$q->end_html.""\n";
```



Beispiel 4: Adventskalender



- Nettes Feature des Moduls CGI:
Kommandozeilentest

```
ss7@ws101:~/webeng/apache/htdocs/mycgi > ./advkal.pl offen=6
Set-Cookie: AdvKal=6; domain=.uni-ulm.de; path=/; expires=Tue,
  30-Dec-2003 08:40:23 GMT
Date: Fri, 05 Dec 2003 08:40:23 GMT
Content-Type: text/html; charset=ISO-8859-1

<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE html
  PUBLIC "-//W3C//DTD XHTML Basic 1.0//EN"
  "http://www.w3.org/TR/xhtml-basic/xhtml-basic10.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="en-US"><head>
<title>Advent, Advent, ein Lichtlein brennt</title>
</head><body>
<h2>Adventskalender</h2>  (...)
```

