

Programmierung von Webapplikationen mit Perl / MySQL

Markus Schade
clt5@markusschade.com

Matthias Szusdziara
matthias.szusdziara@s1999.tu-chemnitz.de

2. März 2003
5. Chemnitzer LinuxTag

Inhalt



Für das Programmieren von Datenbank-gestützten Webapplikationen wird oft nur PHP genannt. Zu den Eltern PHPs gehört auch die Skriptsprache Perl. Dabei ist Perl keinesfalls ein alter Hase, sondern kann locker mit dem Newcomer mithalten.

Dieser Workshop zeigt, wie man mit Perl CGI Skripte schreibt und Informationen aus einer MySQL Datenbank einbindet.

Wir bedanken uns für die freundliche Unterstützung durch die MySQL GmbH.

Geschichte von Perl



- 1986 von Larry Wall aus einer Not heraus entwickelt,
- Name zunächst *Gloria* nach seiner Frau, dann *Pearl*, schließlich *Perl*
- **P**ractical **E**xtraction *And* **R**eport **L**anguage
- Tool für Systemadministration
- Weiterentwicklung durch die OpenSource Gemeinde

Über Perl



- mächtige Skriptsprache für fast alle Anwendungsgebiete selbst Spiele in Perl (z.B.: Frozen Bubble)
- portiert auf alle (fast) alle Plattformen (<http://www.cpan.org/ports/index.html>)
- CPAN - **C**omprehensive **P**erl **A**rchive **N**etwork (www.cpan.org)
- aktuelle Version: Perl 5.8.0

Über MySQL



- Mitte der 80er u.a. von Michael 'Monty' Widenius entwickelt
- GPL Lizenz, wenn man damit kein Geld verdienen will.
- Bei kommerziellen Projekten ist eine entsprechende Lizenz erhältlich
- schnelle Datenbank besonders geeignet für Web-Anwendungen
- stable: MySQL 3.23
- beta: MySQL 4.x

Perl ist einfach



- `print "Hello World!";`
- `$text = "Hello World!";`
`print $text`
- `$foo = "123";`
`print $foo + 1;`
- `$foo` - Skalar

Datentypen



- `@array` - Array
- `%hash` - Hash, oder assoziatives Array
- `&sub` - ein Stück Perl-Code
- `*bla` - alles namens bla

Perl-Einführung - Skalare



- `$a = 42;`
- `$pi = 3.14159265;`
- `$loesung = $a * $pi;` - ein Ausdruck
- `$cmd = 'pwd'` - Ausgabe eines Befehls

Perl-Einführung - Skalare



- `$foo = "bla";` - String
- `$bar = "das ist $foo";` - String mit Interpolation
Ergebnis: das ist bla
- `$bar = 'Preis: $100';` - String ohne Interpolation

Perl-Einführung - Arrays



- Arrays

- `@array = ('42', 'bla', 'foo');`
- `$foo = pop @array;`
- `push (@array, $bar);`
- `$array[0] = 123;`

Perl-Einführung - Hashes



- Hashes

- `%hash = (
 'Jan' => 'Januar',
 'Feb' => 'Feburar'
);`
- `$hash{'Jan'} = 'Januar';`

Perl-Einführung - Das erste Skript



```
#!/usr/bin/perl  
my $hello = "Hello World!";  
print $hello;
```

Das CGI.pm-Modul



- CGI - Common Gateway Interface

```
#!/usr/bin/perl
use strict;
use CGI;
my $cgi = new CGI;
#oder
my $cgi2 = CGI::new();
```

Das CGI Grundfunktionen



- Erzeugen eines HTTP-Headers: `header`
- Erzeugen eines Redirect-Headers: `redirect`
- Ermitteln von Parametern: `param`
- Ermitteln der Werte eines Cookies: `cookie`

HTML Objekte I



- Beginn eines HTML-Dokumentes: `start_html`
- Ende eines HTML-Dokumentes: `end_html`
- Formular beginnen: `start_form`
- Formulate beenden: `end_form`

HTML Objekte II



- Formularobjekte: `button`, `hidden`, `textfield`, `checkbox`, `submit`
- Namen der HTML-Tag-Methoden entsprechen den Namen der HTML-Tags
- z.B.: `h1` für `<H1></H1>` oder auch `img(src=>"bild.jpg")` \Rightarrow ``

MySQL Einführung -



- als MySQL root DB anlegen
`CREATE DATABASE sample`
- Nutzer anlegen und Zugriff gestatten
`GRANT ALL on sample.* TO marks@'%' IDENTIFIED BY "geheim"`
- mit Datenbank verbinden
`mysql -h dbhost -u marks -p`
- Beenden mit QUIT

Tabellen I



- Tabellen anlegen

```
CREATE TABLE tabelle1 ( Spaltendef. )
```

- Format: Name Typ Optionen

```
Bsp: vorname VARCHAR(15) NOT NULL
```

oder

```
id INT PRIMARY KEY NOT NULL AUTO_INCREMENT
```

- Tabelle löschen

```
DROP TABLE tabelle1
```

Tabellen II



- Tabellen ändern `ALTER TABLE`
- Tabellen anzeigen `SHOW TABLES`
- Spalten anzeigen
`SHOW COLUMNS FROM tabelle1`
oder
`DESCRIBE tabelle1`

SQL Befehle



- `SELECT vorname FROM tabelle1 WHERE vorname='Markus'`
- `UPDATE tabelle SET vorname='Matthias' where id=1`
- `INSERT INTO tabelle1 (vorname) VALUES ('Markus')`
- `DELETE FROM tabelle1 WHERE ID=1`

Perl-DBI



- DBI Modul laden mit `use DBI;`

Name	Bedeutung
<code>\$dsn</code>	Datenquelle (Data Source Name)
<code>\$dbh</code>	Handle auf ein Datenbankobjekt
<code>\$sth</code>	Handle auf eine Anfrage
<code>\$rc</code>	Rückgabecode von Booleschen Operationen
<code>\$rv</code>	Rückgabewert bei numerischen Operationen
<code>\$rows</code>	Anzahl der Zeilen

Ein DBI-Skript



```
#!/usr/bin/perl
use strict;
use DBI;
my ($dsn) = "DBI:mysql:sample:localhost:3306";
my ($user) = "marks";
my ($password) = "geheim";
my $dbh = DBI->connect ($dsn,$user,$password) or
error("DB connect failed: $DBI::errstr");
```



```
my $sql = qq{SELECT vorname FROM tabelle1 where  
id=1};  
my $sth = $dbh->prepare ($sql);  
$sth->execute();  
my $vorname = $sth->fetchrow_array();  
$sth->finish();  
$dbh->disconnect();
```

Perl-DBI continued



- Was tun bei SQL Statements ohne Rückgabewerte?
`$dbh->do($sql)` ohne `prepare`

- Rückgabemethoden

Methode	Rückgabewerte
<code>fetchrow_array()</code>	Array
<code>fetchrow_arrayref()</code>	Referenz auf ein Array
<code>fetch()</code>	identisch zu <code>fetchrow_array()</code>
<code>fetchrow_hashref()</code>	Referenz auf einen Hash mit Zeilenwerten, angegeben durch den Spaltennamen

Zusammenfassung



- Perl steht PHP in den Punkten Funktionalität und Umfang in nichts nach.
- Es existieren eine Unmenge an fertigen Modulen im CPAN.
- Die Sprache selbst ist nicht aufs Web beschränkt.
- PHP ist nicht schneller bei Verwendung von mod_perl im Apache.
- Ergo: `use perl;`

Literatur Perl



- `perldoc`
- CGI - kurz & gut / Linda Mui (O'Reillys Taschenbibliothek)
- Programmieren mit Perl / Wall, Christiansen, Schwartz (O'Reilly)
- Perl Kochbuch / Christiansen, Torkington (O'Reilly)
- Reguläre Ausdrücke / Jeffrey E.F. Friedl (O'Reilly)

Literatur MySQL



- www.mysql.de/doc/
- MySQL - das offizielle Handbuch / Widenius, Axmark, MySQL AB (MITP)
- MySQL - entwicklung, implementierung und referenz / Paul Dubois (Markt+Technik)

MySQL ist ein eingetragenes Warenzeichen von MySQL AB.

Danksagung



Vielen Dank für ihre Aufmerksamkeit.

created with \LaTeX and PPower4