

PHP-Kurs 3: Datenbankbindung, PEAR

09.08.2005

Christoph Hermann, Dominik Benz
{hermann,dbenz}@informatik.uni-freiburg.de

Übersicht

1. **Wiederholung**
2. Datenbankbindung
3. PEAR

Inhalte

Montag	Dienstag	Mittwoch
<ul style="list-style-type: none">- Einführung- Orga- Einrichtung der Arbeitsumgebung- PHP Grundlagen	<ul style="list-style-type: none">- PHP Grundlagen 2- Objekte in PHP	<ul style="list-style-type: none">- Datenbankanbindung- PEAR

Kontrollfragen

- ▶ Wie kann ich das aktuelle Datum in der Form tt.mm.jj ausgeben?
- ▶ Wie kann ich einen String an eine Datei anhängen? (z.B. bei Log-Files)
- ▶ Kann ich Objekte als Keys für Arrays verwenden? Kann ich Objekte in Arrays ablegen?
- ▶ Wozu braucht man Sessions?
- ▶ Wie rufe den Konstruktor der Vaterklasse auf?
- ▶ Wie erzeuge ich eine identische Kopie eines Objekts?
- ▶ Kennst Du ein Beispiel eines objektorientierten Codes, der unter PHP5, aber nicht unter PHP4 funktioniert?

Musterlösung Aufg. 1

```
<?php
class Stopwatch
{
    var $started = 0;
    var $elapsed = 0;    // speichert die vergangene Zeit
    function __construct() {
        $this->start();    // bei Instanziierung gestartet
    }

    function start() {    // Zeitmessung starten
        $this->elapsed = 0;
        $this->cont();
    }

    function cont() {    // Zeitmessung fortsetzen
        $this->started = microtime(true);
    }

    function pause() {    // Zeitmessung unterbrechen
        $this->elapsed += microtime(true) - $this->started;
        return $this->elapsed;    // vergangene Zeit liefern
    }

    function stop() {    // Zeitmessung stoppen
        return $this->pause();
    }
} ?>
```

Übersicht

1. Wiederholung
2. **Datenbankanbindung**
 - Grundlegendes
 - Beispiel MySQL
3. PEAR

Grundlagen

- ▶ Aufgabe eines Datenbanksystems: grosse Datenmengen sicher speichern und für Abfragen effizient bereitzustellen
- ▶ relationale Datenbanken: speichern Daten in flachen, zweidimensionalen Tabellen, die über Schlüssel miteinander verknüpft werden
- ▶ bekannte Systeme: Oracle, MySQL, MSSQL, PostgreSQL, ...
- ▶ viele weitere Features (Trigger, stored procedures, Indexe, ...) → würde Rahmen sprengen ;-/

Table *student*

<u>matNr</u>	name	geboren
12345	<i>hans</i>	1.1.80
67890	<i>peter</i>	2.2.82

Tabelle *kurs*

<u>kursID</u>	titel
1	<i>php</i>
2	<i>c++</i>

Tabelle *belegung*

<u>kursID</u>	<u>matNr</u>
1	12345
1	67890

Einfügen / Update / Löschen

- ▶ geschieht mittels SQL (Structured Query Language)
- ▶ grundlegendste Befehle: SELECT, UPDATE, INSERT, DELETE:

```
// Abfragen von Daten  
SELECT name, geboren  
FROM student  
WHERE matNr = '12345'
```

```
// Ersetzen von Werten  
UPDATE kurs  
SET titel = 'php-kurs'  
WHERE titel = 'php'
```

```
// Löschen von Daten  
DELETE FROM student  
WHERE name = 'Hans'
```

```
// Daten einfügen  
INSERT INTO kurs  
VALUES ('3', 'perl')
```

MySQL-Abfrage in PHP

```
<?php // Connecting, selecting database (host,user,pwd)
$link = mysql_connect('localhost', 'soca', 'soca')
    or die('Could not connect: ' . mysql_error());
echo 'Connected successfully';
mysql_select_db('SOCA07') or die('Could not select database');

// Performing SQL query
$query = 'SELECT * FROM students';
$result = mysql_query($query) or die('Error: ' . mysql_error());

// Printing results in HTML
echo "<table>\n";
while ($line = mysql_fetch_array($result, MYSQL_ASSOC)) {
    echo "\t<tr>\n";
    foreach ($line as $col_value) {
        echo "\t\t<td>$col_value</td>\n";
    }
    echo "\t</tr>\n";
}
echo "</table>\n";  ?>
```

Datenbanken und PHP

- ▶ PHP bietet vordefinierte Funktion zur Arbeit mit verschiedenen Datenbanken an
- ▶ Voraussetzung: bei der Installation von PHP muss die Unterstützung der jeweiligen Datenbank mitinstalliert werden (→ `phpinfo()`)
- ▶ Zugriff auf Datenbank auf lokalem Rechner (`localhost`) oder remote möglich
- ▶ verschiedene Funktionen für verschiedene Datenbanken (`mysql_connect`, `mssql_connect`, `pgsql_connect`, ...) → beim Wechsel der Datenbank Änderung des Codes notwendig!
- ▶ Abhilfe: Datenbank-Abstraktionsschicht (z.B. MDB, siehe später)
- ▶ auch ohne dies: Kapselung von Datenbankfunktionen in einer Klasse sehr sinnvoll (siehe Skript)

Übersicht

1. Wiederholung
2. Datenbankbindung
3. PEAR
 - Allgemeines
 - Datenbankabstraktion: MDB
 - Internationalisierung: i18n
 - Authentifizierung: Auth



Was ist PEAR

- ▶ PHP Extension and Application Repository; bietet:
 - eine Bibliothek von PHP-Klassen (Quellcode)
 - eine Infrastruktur für den Vertrieb und die Betreuung von Quellcode
 - einen einheitlichen Stil für Quellcode
 - PHP Foundation Classes (PFC): PEAR-Klassen, die besonders hohen Standards entsprechen bzgl. Qualität, Anwendungsfeld, Verträglichkeit mit anderen Packages, Kompatibilität
 - PHP Extension Community Library (PECL): Erweiterungen in C
- ▶ PEAR stellt viele Lösungen (bzw. Lösungsvorschläge) für Standard-Aufgaben (Authentifikation, Datenbankzugriff, Internationalisierung, Templates...) zur Verfügung
 - aufgeteilt in "packages" (reiner sourcecode oder binaries + evtl. sourcecode)
 - organisiert in einem "package tree"; jeder "Knoten" ist Teil des package-Namens, getrennt mit _ (Beispiel: Mail_IMAP, XML_Beautifier, ...)



PEAR Package Tree: Top Level

Authentication	Benchmarking	Caching
Auth...		
Configuration	Console	Database
		MDB...
Date and Time	Encryption	Event
File Formats	File System	Gtk
HTML	HTTP	Images
Internationalization	Logging	Mail
Math	Networking	Numbers
Payment	PEAR	PHP
Science	Streams	Structures
System	Text	Tools and Utilities
XML	Web Services	

gesamt: 483 packages in 41 Kategorien (Stand: Oktober 2005)



Installation

- ▶ Package Manager für Installation neuer / Upgrade vorhandener Pakete
 - Kommandozeilen-Version:
`pear install <package>`
(ab PHP 4.3 mitgeliefert, vergleichbar mit Debian's `apt-get`); notwendig hierfür: command line interface von PHP installiert (`php-cli`)
 - web installer
 - Installation per ssh, ftp oder lokal möglich
<http://pear.php.net/manual/en/installation.php>
 - manuelles Downloaden und Kopieren der PEAR-Dateien auch möglich, aber nicht empfohlen; niemals: einzelne Files aus Packages herauskopieren (Abhängigkeiten!)
- ▶ Packages und Klassen bestehen aus reinem PHP-Quellcode, der z.B. mit `require_once('PEAR.php')` eingebunden wird
 - `include_path` muss zur PEAR-Installation zeigen



Einbindung

Der `include_path` wird in der PHP-Konfigurationsdatei `php.ini` gesetzt; Anzeige z.B. mit `phpinfo()`:

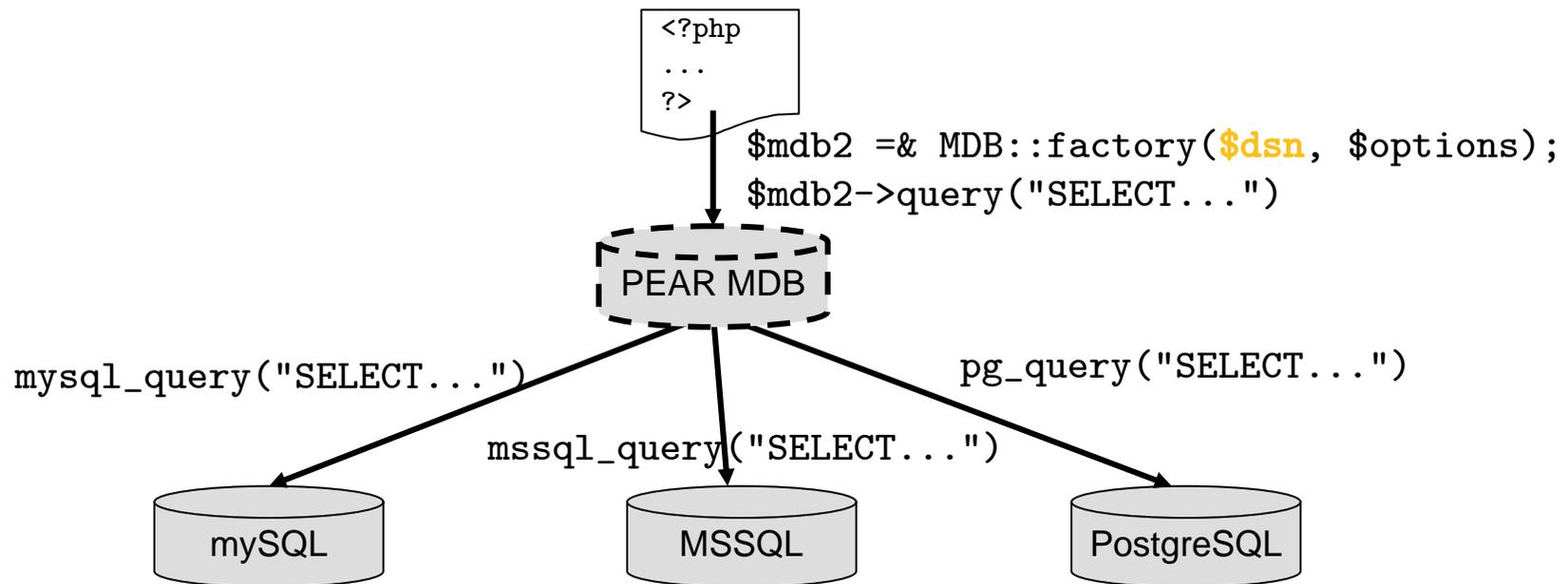
<code>implicit_flush</code>	On
<code>include_path</code>	<code>./usr/share/php:..</code>
<code>log_errors</code>	Off

- ▶ beinhaltet Verzeichnisse (getrennt mit `:`), in denen nach Dateien gesucht wird, die mit `include` oder `require` eingebunden werden
 - auch `.` und `..` möglich (current / parent directory)
- ▶ kann von Hand in `php.ini` verändert werden, aber auch dynamisch zur Laufzeit:

```
ini_set("include_path", '<path_to_pear>' . PATH_SEPARATOR .  
    ini_get("include_path"));
```
- ▶ Test: `require_once('PEAR.php')` funktioniert ohne error



Datenbankabstraktion: MDB





Datenbankabstraktion: MDB 2

```
<?php
require_once 'MDB2.php';
$dsn = 'pgsql://user:pass@hostname/mydb';
$options = array( 'debug' => 2,
                 'result_buffering' => false);

$mdb2 =& MDB2::factory($dsn, $options);
if (PEAR::isError($mdb2)) {
    die($mdb2->getMessage());
}

// ... Die mdb-klasse verwenden: Select
$res =& $mdb2->query('SELECT * FROM table1');

// Zeilenweise Daten holen
while (($row = $res->fetchRow())) {
    echo $row[0] . "\n";
}
$mdb2->disconnect();    ?>
```



Internationalisierung: i18n

- ▶ Textdaten werden in externen Dateien abgelegt und dynamisch geladen
- ▶ Landesspezifische Konventionen (Datum, Währung, ...) werden unterstützt
- ▶ im Quellcode: Kapselung von auszugebenden Strings mit einer speziellen Funktion, oft `echo _('HELLO');`
- ▶ zu übersetzende Strings können dann mit Tools wie `xgettext` direkt aus dem Quellcode extrahiert und separat übersetzt werden → "einsprachiger" Quellcode, vielsprachige Ausgabe
- ▶ Beispiel: siehe Skript



Authentifizierung: Auth

Das Package `Auth` bietet ein Framework für Benutzerauthentifizierung auf `username/password`-Basis

- ▶ Unterstützung einiger "Speicher-Container" für die Login-Daten:
 - Datenbank (basiert auf DB)
 - Textdateien (basiert auf `File_Passwd`)
 - IMAP, LDAP, POP3, ...
 - ausserdem eigene Speichercontainer definierbar
- ▶ bietet eine Basisklasse: `Auth`, Funktionen:
 - `getStatus`, `getUsername`, `checkAuth`, ...
- ▶ hier: Login-Daten im Standardformat in Datenbank gespeichert (MySQL):

```
CREATE TABLE auth (  
    username VARCHAR(50) default '' NOT NULL,  
    password VARCHAR(32) default '' NOT NULL,  
    PRIMARY KEY (username),  
    KEY (password)  
);
```



Auth: Beispiel

```
<?php
require_once "Auth.php";
function loginFunction()
{    // HTML Form for login
}
// define the Storage Container
$dsn = "mysql://soca:soca@localhost/SOCA07";
$a = new Auth("DB", $dsn, "loginFunction");
// Start Authentication
$a->start();
// Logout?
if ($_GET['action'] == "logout" && $a->checkAuth()) {
    $a->logout();
    $a->start();
}
if ($a->checkAuth()) {
    // only visible for logged-in users
    echo "Welcome to our member's section!\n";
    echo "<a href=\"03-auth.php?action=logout\">logout\n";
}
```

Kontrollfragen

- ▶ Was heisst SQL?
- ▶ Wie heisst das SQL-Sprachkonstrukt um Daten aus einer relationalen Datenbank abzufragen?
- ▶ Welche Datenbanken unterstützt PHP?
- ▶ Welche Möglichkeiten gibt es die eigenen PHP-Skripte unabhängig von einer bestimmten Datenbank zu machen?
- ▶ Was heisst I18N? Was heisst L10N?
- ▶ Welche Funktionen bietet Pear? Kann man das essen?

Übungsaufgabe

- ▶ Benutze die Klasse Pear::Auth mit Datenbankauthentifizierung um eine passwortgeschützte Seite zu realisieren.

PHPKurs - Abschluss

Danke für die Aufmerksamkeit :)

- ▶ Feedback-Formulare ausfüllen (bloss nicht nett sein wollen!!)
- ▶ weitere Fragen / Anregungen / Kommentare:
hermann,dbenz@informatik.uni-freiburg.de

```
foreach ($phpkursTeilnehmer as $teilnehmer) {  
    becomeProfiIn('PHP');  
    if (is_interested($teilnehmer)) {  
        $fortgeschrittenenKurs[] = $teilnehmer;  
    }  
}  
exit('Goodbye □ and □ good □ luck!!!');
```