**Datenbankanbindungen, 3-Schichten-Architektur**

**Lernziele:**

* Sie verstehen den Unterschied zwischen einer ODBC und herkömmlichen DB-Verbindung mit mysqli
* Sie können ein MVC-Entwurfsmuster für simple Web-Applikationen umsetzen

# 1 Reprise: DB-Verbindung mit ODBC

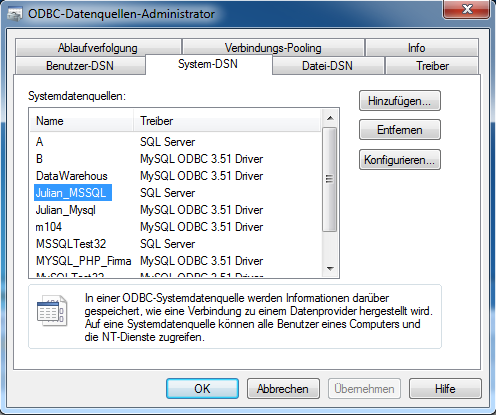
Wir haben letzte Woche eine DB-Verbindung mit *mysqli* umgesetzt und dabe festgestellt, dass wir ein Objekt mit der Verbindung erhalten.

Ein Problem mit dieser Variante ist, dass wir code-mässig nicht unabhängig arbeiten. Wenn wir z.Bsp. die DB auswechseln (von MySQL zu Oracle oder zu MS-Sql), dann müssen wir den PHP-Code anpassen.

Eine elegantere Lösung ist die Verwendung von ODBC (= Open Database Connectivity) mit einem Data Source Name (DSN). Damit Sie mit ODBC arbeiten können, müssen Sie zuerst die notwendige Infrastruktur aufsetzen. Sie benötigen:

* ODBC Treiber für die entsprechende DB
* Ein ODBC Data Source Name (DSN)

Die DSN wird über den ODBC-Datenquellen Admin erstellt:



Die Konfiguration sieht dann so aus, damit Sie mit *odbc\_connect* arbeiten können:

$user = "test";

$pass = "test";

$toReturn = "";

$DBName = "";

$DBPassword = "";

$ServerName = "MySQLTest32"; #DSN festlegen

Wobei „ServerName“ die DSN-Bezeichnung ist. Sie erhalten eine ODBC-ID, welche Sie in der globalen Session-Variable speichern können, z.Bsp. so:

$\_SESSION['DBConnection']['ServerName']=$ServerName;

$\_SESSION['DBConnection']['DBID'] = odbc\_connect($\_SESSION['DBConnection']['ServerName'],"test","test");

Die Notation lautet:

int **odbc\_connect** (string dsn, string user, string password [, int cursor\_type])

Sie dürfen die ODBC-Verbindung gerne ausprobieren oder in Ihrem Projekt verwenden.

# 2 Aufgabe: Datenbankanbindung

Wir wollen uns nochmals mit kleineren Formularen und Datenbankanbindungen beschäftigen. Verwenden Sie dazu die Vorlage „alter.php“ vom BSCW. Wie könnte man dieses Skript verbessern? Machen Sie Vorschläge und setzen Sie diese entsprechend um.

Tipp: die Verwendung von mysql\_connect sollte auf jeden Fall ersetzt werden. Was wird aktuell benützt?

Passen Sie das Skript an, so dass es zeitgemäss ist.

Ziel: Wir wollen eine Artikel-Verwaltung aufbauen. Dabei sollen die Artikel angezeigt, angepasst und gelöscht werden können. Wir wollen in einem separaten Formular auch neue Artikel hinzufügen.

**2b Optionale Aufgabe**

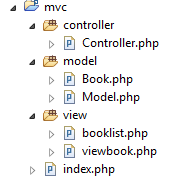
Fügen Sie der Applikation noch ein Login hinzu, so dass die DB nur mit gültigem Login verwendet werden kann.

# 3 Saubere Trennung GUI, Logik, DB-Anbindung

Wir haben einige Beispiele gesehen, die zwar funktionieren, aber nicht sehr elegant umgesetzt sind. Auch bei Web-Applikationen wollen wir darauf achten, dass sie möglichst in sinnvolle Bereiche unterteilt sind.

Das MVC hat sich auch hier als ein Standard durchgesetzt. Es gibt online gute Beispiele, die das zeigen. Eines davon ist im BSCW abgelegt.

Dabei wird in diesem Beispiel objektorientiert das MVC umgesetzt:



Ein Grundprinzip ist, dass alle Fäden beim *Controller* zusammenlaufen. Dh. Dieses Skript kennt das *Model* und die *View* und ruft beide auf:

**public function** \_\_construct()

{

$this->model = **new** Model();

}

**public function** invoke()

{

**if** (!**isset**($\_GET['book']))

{

// no special book is requested, we'll show a list of all available books

$books = $this->model->getBookList();

**include** 'view/booklist.php';

}

**else**

{

// show the requested book

$book = $this->model->getBook($\_GET['book']);

**include** 'view/viewbook.php';

}

}

1. Erstellen Sie ein neues PHP-Projekt und importieren Sie dieses Beispiel. Studieren Sie den Aufbau und vergewissern Sie sich, dass Sie es verstehen. Erstellen Sie eine Skizze (Sequenzdiagramm), wie diese Schichten miteinander kooperieren.
2. Nehmen Sie dieses Beispiel als Vorlage und schreiben Sie die vorige DB-Anbindungs-Applikationen um.

Offene Frage: Sehen Sie eine Möglichkeit, ein solches Design auch nicht-objektorientiert umzusetzen?

Zeigen Sie Ihre Lösungen der Lehrperson.

# 4 Ausblick: Verbesserte Usability mit Ajax

Wir wollen uns auch mit *XMLHttpRequest* beschäftigen und dabei untersuchen, wie wir das in einer Web-Applikation verwenden können.

Als Vorgeschmack probieren Sie folgendes aus:

Erstellen Sie ein PHP-Script welches in einem Array eine Liste von Namen hat.

Erstellen Sie eine HTML-Seite mit einem Text-Input Formular. Diese Seite soll *XMLHttpRequest* verwenden und die PHP-Seite aufrufen und Vorschläge der Namen zurückliefern.