

Webentwicklung (WE) - FWPM

Übung 5: "Persistenz"

In dieser Übung geht es darum, unsere Anwendung mit einer Persistenz durch eine MySQL Datenbank auszustatten. Dazu werden wir uns ein einfaches Datenmodell überlegen, dieses als Tabellen anlegen und Daten über eine Zugriffsschicht in unserer Anwendung nutzbar machen.

Datenbankdesign

Wer bereits eine Idee für die Studienarbeit hat, sollte ab hier seine eigenen Designs entwerfen. Im Beispiel wird es um die schon angesprochene Blogsoftware gehen.

1. Designe ein einfaches Datenbankschema zum Start deiner Anwendung. Konzentriere dich auf ein kleines aber zentrales Feature, für einen Blog wäre das z.B. das Anzeigen von Posts.
 - a. Welche Entitäten brauchst du? Zumindestens den Post selbst solltest speichern
 - i. Welche Felder (bzw. Spalten) braucht deine Entität?
2. Lege deine Datenbank und ihre Tabelle an (z.B. per Wizard in MySQL Workbench, oder per CLI mit SQL Queries) und fülle die Tabelle mit Daten.
 - a. Es ist sinnvoll, pro Datenbank einen eigenen Nutzer in deinem Datenbankmanagementsystem zu erstellen, der nur Zugriff auf diese eine Datenbank hat.
 - b. Für Fixture-Textdaten nutzt die Webwelt das sog. Lorem Ipsum. Einfach googeln.

Bonus: Erstelle eine zweite Tabelle für eine Author-Entität und verknüpfe sie mit den Blogbeiträgen. Nutze dazu die *Foreign-Key* Mechanik relationaler Datenbanken.

Datenbankanbindung

Richte dir parallel zur Vorlesung die Anbindung deiner Anwendung an die Datenbank ein. PDO wird empfohlen ist aber kein Muss.

1. Erstelle eine Klasse, deren Aufgabe es ist die Verbindung zu deiner Datenbank aufzubauen.
 - a. Sei dir bewusst, dass die Klasse die Datenbankverbindung anderen Klassen zur Verfügung stellen muss. Wie sie das tut ist dir überlassen.
2. Nutze ein System für Konfigurationsdateien wie DotEnv (siehe Vorlesung) um deine Zugangsdaten außerhalb deiner git Historie zu halten (Sicherheitslücke!).
 - a. Im einfachsten Fall kannst du das DSN System zur Referenzierung von Konfigurationen nutzen.

Zugriffsschicht

Im letzten Schritt sollen deine Daten aus der Datenbank als Instanzen deiner Model-Klassen im Code nutzbar sein.

1. Erstelle eine Klasse die es erlaubt eine einzelne Entität aus der Datenbank abzurufen.
 - a. Sieh dir das Repository Pattern an, du kannst einfach mit der Abfrage starten und weiterführende Funktionen wie das Speichern in späteren Übungen hinzufügen.
2. Erstelle solche Klassen für alle weiteren Models/Tabellen innerhalb deiner Datenstruktur.
3. Binde die so erstellte Zugriffsschicht in deine MVC Architektur ein indem dein Model von deiner Zugriffsschicht geladen wird.
 - a. Die Daten die beim Aufruf deiner Anwendung im Browser angezeigt werden, sollten jetzt aus der Datenbank kommen.

Bonus: Nutze ein ORM Framework wie z.B. Doctrine zur Anbindung deiner Models an die Datenbank.