

Datenbanksysteme

1. Praktisches Projekt

Einführung

Im Laufe des praktischen Projekts sollen Sie ein System zur Verwaltung von Einträgen einer Bibliothek entwickeln. Die Entwicklung dieses Projekts soll dabei in vier Schritten ablaufen:

1. ER-Modellierung der Datenbank
2. Überführung der ER-Modellierung in ein relationales Modell
3. Implementierung der Datenbank in SQLite
4. Umsetzung eines dazugehörigen RESTful Web Services

Die einzelnen Arbeitsschritte bauen hierbei aufeinander auf. Die aktuelle Aufgabenstellung wird im letzten Abschnitt des Kapitels „Aufgabenstellung“ beschrieben.

Wichtige allgemeine Hinweise

Für die ersten zwei Schritte haben Sie jeweils einen Bearbeitungszeitraum von einer Woche, für die Implementierung der Datenbank in SQLite und die abschließende Umsetzung eines dazugehörigen RESTful Web Services haben Sie jeweils zwei Wochen Zeit. Um die praktischen Übungen zu bestehen, müssen Sie *jeden* der vier Abschnitte bestehen. Dies bedeutet, dass Sie den zweiten Schritt nur bearbeiten können, wenn Sie den ersten *bestanden* haben. Klären Sie daher kritische Punkte und Fragen *vor* der Abgabe in den Sprechstunden, da es aufgrund des Zeitplans keine Nachbearbeitungszeit geben kann.

Sie müssen alle Arbeitsschritte *alleine* und selbstständig bearbeiten. Gruppenarbeiten, auch bei nicht ganz identischen Abgaben, führen zum Ausschluss aller Beteiligten.

Die Abgabe dieses vierten Teils ist bis Montag, den 22.03.2021, um 10:00 Uhr über das Abgabesystem möglich. Achten Sie auf die **aktuelle Aufgabenstellung** am Ende des Übungsblattes.

Anwendungsszenario

Das Ziel der gesamten praktischen Übung besteht in der Implementierung eines Systems zur Verwaltung von Einträgen einer Bibliothek. Die explizite Aufgabe für dieses Blatt finden Sie weiter unten.

Es sollen folgende Sachverhalte dargestellt werden:

In unserer Datenbank sind **Nutzer** hinterlegt. Jeder Nutzer hat einen Vornamen, einen Nachnamen, ein Geburtsdatum, eine eindeutige E-Mail Adresse, sowie ein Passwort. Das Passwort und die E-Mail Adresse dienen der Authentifizierung.

Ein **Bibliothekar** ist ein Nutzer. Dieser besitzt eine einmalige Telefonnummer.

Ein **Kunde** ist ebenfalls ein Nutzer. Er besitzt ein Guthaben (Betrag in €) und kann optional vom Jahresbeitrag befreit sein.

Der Kunde kann sich beliebig viele **Exemplare** eines **Artikels** ausleihen. Dabei wird in der Datenbank zu jeder **Ausleihe** ein Zeitraum der Ausleihe mit Beginn und Ende gespeichert, sowie ein Vermerk darüber, ob die Ausleihe zurückgegeben wurde. Eine Ausleihe bezieht sich auf ein Exemplar. Der Kunde kann sich beliebig viele Exemplare ausleihen. Jede Ausleihe kann beliebig oft verlängert werden.

Ein Kunde hat genau eine **Adresse**. Jede Adresse besteht aus der Angabe zu Stadt, Postleitzahl, Straße und Hausnummer. Es können beliebig viele Kunden dieselbe Adresse teilen.

Ein Exemplar ist ein **Artikel**, der zu einem bestimmten Anschaffungspreis erworben wurde und unter genau einem **Standort** zu finden ist. Dabei unterscheiden sich die Standorte durch einen Vermerk zur Etage, sowie zum Regal (Regal Nr.).

Ein Artikel besitzt eine eindeutige ISBN Nummer, eine Bezeichnung, eine Beschreibung und ein optionales Coverbild. Zusätzlich liegt zu jedem Artikel ein Erscheinungsdatum vor.

Jeder Artikel wird in Form genau eines **Mediums** angeboten. Auf diesem Wege kann beispielsweise ein Buch als Softcover und Hardcover angeboten werden.

Jeder Artikel kann beliebig viele andere Artikel empfehlen, zeitgleich können zu einem Artikel beliebig viele Artikel empfohlen sein.

Jeder Artikel wurde von mindestens einem **Autor** verfasst. Ein Autor besitzt einen Vornamen und einen Nachnamen.

Jeder Artikel gehört zu beliebig vielen **Genres** (z. B.: Kriminalroman).

4. Aufgabenteil

Allgemein

Ihre Aufgabe besteht nun darin, ein Java-Programm zu schreiben, durch das Anwender die Möglichkeit haben, mit Ihrer Datenbank zu kommunizieren.

Dieses Programm muss eine REST-API in Form eines [RESTful](#) Web Services sein und soll den gelisteten Anforderungen genügen.

Anforderungen

Das Programm soll die Grundfunktionalität eines Systems zur Buchung von Festivals bieten. **Die genaue Auflistung der Anforderungen finden Sie auf dem [Zusatzblatt](#)¹.**

Hilfsmittel

Sie dürfen Tools wie beispielsweise [Gradle](#), Frameworks wie beispielsweise [Jersey](#) oder [Spring](#) und Libraries wie beispielsweise [Lombok](#) benutzen, die Ihnen die Entwicklung vereinfachen, solange Sie dadurch noch immer **alle Anforderungen einhalten**.

Es steht Ihnen ein [Template für die REST-API](#) zur Verfügung, das Sie benutzen können, aber nicht müssen.

Abgabe

Die Abgabe soll in Form eines ZIP-Archivs erfolgen, welches das Programm als Quellcode bzw. IDE-Projekt und die korrigierte und mit Beispieleinträgen befüllte Datenbank enthält, mit der das Programm kommuniziert. Halten Sie sich bei der Einreichung Ihrer Abgabe an das bereits vom ersten Übungsblatt bekannte Format.

Bitte geben Sie zusätzlich eine Datei (curl.txt) ab, die jeweils einen auf Ihrer Datenbank korrekt funktionierenden cURL-Befehl für jeden Endpunkt der API enthält. Filterparameter sollen hierbei nicht angegeben werden.

Die Abgabe soll ein als ZIP-Archiv komprimierter Ordner mit Namen dbs-propra-ws2020-<vorname>-<nachname>, wobei <vorname> Ihr Vorname und <nachname> Ihr Nachname ist, und folgender Ordnerstruktur sein:

- phase1 (Ordner mit Inhalt aus Phase 1)
- phase2 (Ordner mit Inhalt aus Phase 2)
- phase3 (Ordner mit Inhalt aus Phase 3)
- phase4 (Ordner für Phase 4)
- phase4/data (Ordner für Datenbank spezifische Daten)
- phase4/data/schema.sql (nur SQLite-DDL-Anweisungen für SQL-Schema)
- phase4/data/data.sql (nur SQLite-DML-Anweisungen für Beispieleinträge)

¹<https://pad.hhu.de/s/S1tt1j9hB>

- phase4/data/database.db (SQLite-Datenbank, welche aus schema.sql und data.sql erstellt werden kann)
- phase4/gradle (Ordner für Gradle spezifische Dateien)
- phase4/gradle/wrapper (Ordner für Gradle Wrapper spezifische Dateien)
- phase4/gradle/wrapper/gradle-wrapper.jar (Gradle Wrapper)
- phase4/gradle/wrapper/gradle-wrapper.properties (Einstellungen für Gradle Wrapper)
- phase4/src (Ordner für Quellcode)
- phase4/build.gradle (Buildskript für Gradle)
- phase4/gradlew (ausführbares Skript für Gradle Wrapper)
- phase4/gradlew.bat (ausführbares Skript für Gradle Wrapper)
- phase4/settings.gradle (Einstellungen für Gradle)
- phase4/README.adoc (kritische Entscheidungen und die verwendete SQLite-Version)
- phase4/curl.txt (cURL-Befehle)

Hinweise

Teilen Sie sich die Bearbeitungszeit von 2 Wochen gut ein: Versuchen Sie, so gut es geht, in der ersten Woche fertig zu werden, um für eventuell auftretende Bugs oder Probleme noch genug Zeit zu haben.

Die Korrektur wird zum Teil automatisiert ablaufen. Es ist folglich **zwingend erforderlich, sich genauestens an die Anforderungen zu halten!** Bei Fragen dazu, wenden Sie sich bitte an die Tutoren.