Institut für Informatik
Datenbanken und Informationssysteme
Prof. Dr. Stefan Conrad
Kirill Bogomasov

WiSe 2021/2021
06.12.2021
Praktisches Projekt

Datenbanksysteme
2. Praktisches Projekt

Einführung

Im Laufe des praktischen Projekts sollen Sie ein System zur Verwaltung eines Audio-Streaming-Dienstes 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 20.12.2021, 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 eines Audio-Streaming-Dienstes. Die explizite Aufgabe für dieses Blatt finden Sie weiter unten.

Es sollen folgende Sachverhalte dargestellt werden:

Jeder **Nutzer** hat einen Benutzernamen, eine E-Mail-Adresse, ein Passwort und eine Wohnadresse (Land, Stadt, PLZ, Strasse, Hausnummer). Ein Nutzer kann den kostenlosen Dienst nutzen oder für eine bessere Tonqualität auf die Premiumvariante (Premium-Nutzer) zurückgreifen.

Ein **Künstler** ist ein Premium-Nutzer, jedoch ist nicht jeder Premium-Nutzer zugleich ein Künstler. Künstler veröffentlichen unter einem Künstlernamen Titel. Beim Premium-Nutzer wird mitgespeichert, zu welchem Zeitpunkt seine Premium-Mitgliedschaft ausläuft.

Titel werden von mindestens einem oder in Zusammenarbeitet mehrerer Künstler veröffentlicht. Ein Titel hat eine Benennung, eine Dauer und gehört zu einem bestimmten Genre. Jedes Genre kann ein anderes Genre empfehlen und von beliebig vielen anderen Genres empfohlen werden. Nutzer können einzelne Titel jeweils einmalig kommentieren. Jeder Titel kann Bestandteil beliebig vieler Playlists sein. Titel können einem Album angehören. Jeder Titel liegt in 2 Qualitäten vor (LQ, HQ). Deshalb wird für jeden Titel jeweils der Speicherort mitgespeichert, wobei hier zwischen den Qualitäten unterschieden wird.

Eine **Playlist** ist eine Zusammenstellung beliebiger Titel. Diese wird durch einen Premium-Nutzer erstellt und enthält mindestens einen Titel. Sie muss eine Bezeichnung besitzen. Die Playlist kann privat oder öffentlich sein. Alle Nutzer haben die Möglichkeit öffentliche Playlists zu bewerten (Bewertungsskala $[1, \ldots, 10]$). Eine Playlist kann ein Coverbild besitzen.

Eine **Band** besteht aus mindestens zwei Künstlern. Ein Künstler kann jedoch maximal zu einer Band gehören. Eine Band hat einen Namen und eine optionale Bandgeschichte.

Ein **Genre** hat eine Bezeichnung {Rock, Rap, Klassik, ...}.

Ein **Album** wird vom genau einem Künstler oder einer Band veröffentlicht. Es hat eine Bezeichnung und ein Erscheinungsdatum (Jahr).

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 Verwaltung von Blogeinträgen 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-ws2122-<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/g6ZV_48Ho

- 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.