

# Einführung

Im Laufe des praktischen Projekts sollen Sie ein System zur Verwaltung eines Musik-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 09.10.2023, 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 Musik-Streaming-Dienstes. Die explizite Aufgabe für dieses Blatt finden Sie weiter unten.

Es sollen folgende Sachverhalte dargestellt werden:

**Nutzer** haben einen Benutzernamen, eine E-Mail-Adresse, eine IBAN und ein Passwort. Ein Nutzer kann den kostenlosen Dienst nutzen oder für eine bessere Soundqualität auf die Premiumvariante (Premium-Nutzer) zurückgreifen. Jeder Premium-Nutzer hat einen Vermerk über die Gültigkeitsdauer seines Accounts (Datum). Nutzer können Kritik in Form von Text zu einzelnen **Alben** schreiben und zwar so häufig wie sie möchten. Jeder einzelne **Song** kann von einem Nutzer bewertet werden (Skala: 1-5). Mehrfachbewertung sind dabei nicht möglich.

**Künstler** sind **Premium-Nutzer**, jedoch ist nicht jeder Premium-Nutzer zugleich ein Künstler. Ein Künstler ist als Künstler ab einem bestimmten Datum aktiv. Künstler veröffentlichen unter einem Pseudonym **Aufnahmen**. Ein Künstler ist entweder ein **Musiker** oder ein **Podcaster**. Dementsprechend ist eine Aufnahme entweder ein Song oder eine **Podcast-Episode**. Jede Aufnahme hat einen Titel und eine Dauer (in Sek.).

Ein Podcaster kann mit beliebig vielen anderen Podcastern zusammenarbeiten. Manche Podcaster arbeiten professionell.

**Songs** werden von mindestens einem oder in Zusammenarbeit mehrerer Musiker veröffentlicht. Ein Song gehört zu einem bestimmten Genre. Ein Song kann Bestandteil eines Albums, aber auch einer oder mehrerer Playlists sein. Jeder Song liegt in 2 Qualitäten vor (LQ, HQ). Deshalb wird für jeden Song jeweils der Speicherort mitgespeichert.

Eine Podcast-Episode hat eine Nummer und wird in genau einer Sprache veröffentlicht. Beliebige viele Podcast-Episoden bilden einen Podcast, der mindestens einen Themenbereich behandelt.

**Genre** haben eine Bezeichnung (z.B. Pop, Rock,...).

**Playlist** ist eine Zusammenstellung beliebiger Songs oder Podcasts. Diese wird durch einen Premium-Nutzer erstellt und enthält mindestens einen Song. Sie muss einen Namen haben. Die Playlist kann privat oder öffentlich sein. Nutzer haben die Möglichkeit die Playlist zu liken oder zu dislikern.

**Bands** bestehen aus mindestens zwei Musikern. Ein Musiker kann jedoch maximal zu einer Band gehören. Zu jedem Musiker ist bekannt, ob dieser auftrittsfähig ist. Eine Band hat einen Namen und optional einen Vermerk zur Bandgeschichte.

**Album** wird vom einem Musiker oder einer Band veröffentlicht. Es hat eine Bezeichnung und ein Erscheinungsdatum (Jahr). Jedes Album kann bis zu 30 Songs enthalten.

# 4. Aufgabenteil

## Allgemein

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

Für die schnelle Implementierung stehen zwei Templates zur Verfügung:

- [Jersey-Template](#)<sup>1</sup>
- [Spring-Template](#)<sup>2</sup>

Sie dürfen eines dieser Templates zur Implementierung benutzen, sind aber nicht verpflichtet dies zu tun. Bitte bleiben Sie bei Ihrer Wahl.

Eine genaue Erläuterung dieses Aufgabenteils finden Sie auf dem [zusätzlichen Dokument](#)<sup>3</sup>.

## Anforderungen

Es gibt zum einen **Anforderungen zur Technik, zur Funktionalität, als auch zur Abgabe**, die eingehalten werden müssen.

Das Programm soll mit aktuellen Softwareversionen auf der JVM laufen, leicht startbar sein und eine gewisse Grundfunktionalität bieten, die der vorgegebenen Spezifikation genügt. Halten Sie sich bei der Einreichung Ihrer Abgabe an das bereits vom ersten Übungsblatt bekannte Format.

**Ausführliche Informationen dazu finden Sie auf dem bereits erwähnten [zusätzlichen Dokument](#).**

## Hinweise

Teilen Sie sich die Bearbeitungszeit gut ein: Versuchen Sie, so schnell wie möglich 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.

---

<sup>1</sup><https://git.hhu.de/althe100/dbs-propra-phase4-template-jersey>

<sup>2</sup><https://git.hhu.de/althe100/dbs-propra-phase4-template-spring>

<sup>3</sup><https://pad.hhu.de/s/xzHA9BeBn>