

Music Database

Introduction

You are creating a Music Database system where music lovers from all over the world can find information about their favourite bands and songs. You start small, with just basic information and operations that can be executed on the systems, but you have big plans to grow!

Objectives

Create a simple application with a REST API to store and retrieve information about Artists and Albums. Unless explicitly noted below, you have a freedom of modelling the details of the API and the data. You can choose and use any frameworks and libraries that you find helpful. Be ready to justify your choices, though.

Timebox your work to 3 hours. It is not required to implement each and every feature, so if you hit a snag, don't dwell on it for too long.

Technical stack requirements

- Maven
- JDK 8+
- Spring Boot 2.0+

README file

Attach a README file describing:

- choices and decisions you made
- parts you found easy and difficult
- parts you skipped and the ones that you implemented extra

Minimal requirements

- REST API
 - `/artists` endpoint to manage Artist data. Required information: artist name
 - `POST /artists` to save a new artist
 - `PUT /artists/{artistId}` to update an existing artist
 - `GET /artists` lists all artists. Implement:
 - filtering by a part of artist name
 - sorting by artist name (asc/desc)

- paging
- `/artists/{artistId}/albums` endpoint to manage Album data. Required information: title, year of release, genres (list of tags).
 - `POST /artists/{artistId}/albums` to add a new album to an existing artist
 - `PUT /artists/{artistId}/albums/{albumId}` to update an existing album
 - `GET /artists/{artistId}/albums` lists all albums by the given artist. Implement:
 - filtering by genre(s)
 - sorting by album name and release year (asc/desc)
- Data persistence (embedded DB of your choice)
- Tests should be an integral part of your solution

Album details

Once you have successfully implemented the above API, extend it with an endpoint to retrieve more details about an individual album using a 3rd party service. Implement a `GET /artists/{artistId}/albums/{albumId}` endpoint where you return the album information from your system together with a track listing taken from [Discogs API](#).

- Create an Discogs account and set up a new application [link](#)
- Authenticate using Consumer key/secret [link](#)
- Find the album by Artist name and Album name (if more than one is found, you can use the one of your choice; hint: search for master releases only) [link](#)
- Get the track listing using the `resource_url` of the found album and expose it in your API (track number and title); If there are no matches, produce an empty listing

Extra requirements

Implement (a choice of) these only if you have time left or if you fancy an extra challenge

- Expose metrics:
 - Number of calls per endpoint
 - Average processing time per endpoint
 - Time taken to call Discogs
- Reactive implementation
- Dockerize your application