



Take-Home Assignment

Senior Backend Engineer - Royalty Processing

Introduction

We're excited to have you as part of our interview process!

This take-home assignment is designed to evaluate your technical skills, system design approach, and overall software craftsmanship.

Context

Imagine you are part of the team building a music metadata service for a streaming platform akin to Spotify or Apple Music, that serves millions of users worldwide. Our final goal is to provide a Music Metadata Service that will provide behaviour detailed in requirements. Your task is to design and implement a core module that models the behaviours, interfaces with repository and provides an interface for api.

Task Requirements

We need the following behaviours:

1. **add an album with an artist**, artist is not a separate concept in this model;
2. **add tracks to album**;
3. **set album release date**;
4. **a way to check if album is released**;
5. **search for album by title using Levenshtein distance algorithm**.

These requirements brief provide an outline for the core. However, how you approach these requirements, how you design and implement the system, and how you anticipate and plan for potential issues is entirely up to you. We encourage you to make assumptions where necessary, but please ensure that you document and justify those assumptions. Please note that you are not expected to submit a real database integration or packaged service api. At this stage we want to see how you approach the domain modeling, service core design and behaviour implementation. We will be exploring system design options and infrastructure details during follow up technical interview.

Extra Information

- **Language:** Your application should be implemented using JVM language (we use Scala and favour it for its power of expression, which is great for tasks like this assignment).
- **Freedom of Choice:** You are free to choose what you want to use: the build tools, libraries, etc.
- **Keep it simple:** You don't need to dedicate days to this; simply demonstrate your ability to craft excellent software.
- **Functional:** Please ensure that the code in the submission is fully functional on a local machine, and include instructions for building and running it.
- **Dependencies:** Avoid any external dependencies (remote cloud databases, remote search engines, etc.) to solve the task. This exercise is intended to assess your problem-solving skills rather than creating a system reliant on cloud connections.
- **MVP:** Use this exercise as a guide for design decisions, considering it as the initial prototype of a Minimum Viable Product (MVP) that will evolve into production ready deliverable. Be prepared for further discussions regarding how to transition and scale the prototype for future deployment. We understand that "production-ready" may have different interpretations, and we look forward to discussing what it means to you during the review.

Deliverables

- Please submit your solution either as a link to a public repository or as a zip file with all the necessary components for local building and running.
- Be prepared to present and explain your design and implementation.
- If there are additional artifacts (e.g., slides, Miro-board, etc.) that you'll use during the presentation, please include them in your submission.

If you have any questions or require further clarification, feel free to reach out to us. We're looking forward to receiving your solution!