Music collection rush

Summary: In this rush, you will use the API of a popular music streaming service to print out a catalog of your music collection.

Chapter 1

Introduction

Spotify is a non-free Swedish -American audio streaming and media services provider founded on 23 April 2006 by Daniel Ek and Martin Lorentzon. It is one of the largest music streaming service providers with over 406 million monthly active users, including 180 million paying subscribers, as of December 2021. Spotify offers digital copyright restricted recorded music and podcasts, including more than 82 million songs, from record labels and media companies.

Deezer is a French online music streaming service. It allows users to listen to music content from record labels, including Universal Music Group, Sony Music and Warner Music Group (owned by Deezer's parent company Access Industries) as well as podcasts on various devices online or offline. Created in Paris, Deezer currently has 90 million licensed tracks in its library, with over 30,000 radio channels, 100 million playlists, 16 million monthly active users, and 7 million paid subscribers as of January 2019.

(Source: Wikipedia)

Chapter 2

Instructions

The popular streaming services Spotify and Deezer offer an API that developers can use to gather information about the music and the users, allowing them to build useful applications.

The objective of this rush is to build a tool (or series of tools) that will allow a Spotify or Deezer user to download a list of their saved albums, and print them out as a nice looking catalog. Your tool only needs to support one of the services.

You can use the languages, libraries, frameworks or tools of your choice. The focus of this exercise is functionality and quality of the end product, rather than the tools' user interface.

You are not required to write the entire project in a single language. You can make multiple tools, each taking care of a specific aspect of the project. As long as the instructions are clear, it is allowed to have the user perform simple steps manually.

2.1 API access

Though the specifics will depend on the service you chose, you will have to:

- Open an account with the chosen streaming service (or use an existing one)
- Register to the developer program
- Create an application on the service's developer website

Then, your tool should:

- Allow a user of your chosen streaming service to sign in with their own credentials
- Fetch the list of albums saved by that user

Notes:

- You may use existing code for the authentication part. However, you **must** have a basic understanding of how it works and be able to explain it during your evaluation.

- Because these services use web authentication, this part of the tool will most likely need to run in a web browser.

2.2 Catalog

Now that you can authenticate a user and collect data about them and their music, you need to present that data in a pleasing manner.

Display all the user’s saved albums with (at least):

- Cover picture
- Title
- Artist(s)
- Year
- Track listing

Your tool should produce a document (or documents) in the format of your choice, that can be browsed easily and printed out.

Chapter 3

Bonuses

3.1 QR codes

Next to each album, display a QR code that would allow a user to quickly start listening to the album on their mobile phone.

3.2 Genres distribution

At the end of the catalog, display a graph of the distribution of genres in the user's album collection.

Chapter 4

Evaluation

As usual, the evaluator will clone your git repository, and should be able to run your project themselves. Since you have the choice of language and tools, make sure that your project can run on the school iMacs, and to include instructions on how to install the requirements.

If you write your project in C, you are not required to follow the Norm. However, regardless of the language, you are still expected to provide source code that compiles and/or runs without errors.