Playlist App

CSE 438

Spring 2019

Introduction

In this assignment, you will use a third party API to build a playlist application to search for music.

Screenshot here

Details

- Due Date:
- Grading: This entire lab is X points. The distribution of points is listed below in the requirements section.
- Submission: Please put your project into your forked repository. The repository can be found here.

Last.fm API

You will be using the Last.fm API to query for music data. The API itself has a lot to offer, so consider using different functions for the creative portion (and documenting the appropriate API calls). You can find instructions for setting up an account and getting an API key here. We recommend getting your account setup and making a few sample calls in the browser to see what the JSON output looks like.

Description

When the app launches, there should be a GridView displaying the current top tracks (**Hint**: Take a look at the What's New section of the studio code and the API documentation for how to do this). The GridView should display at least 20 tracks if possible (with scrolling), and should display the track cover (if available) and title.

The app should have two tabs - the first should give the user the ability to search for different tracks by artist name. After the user searches for the artist, the app should display the top tracks by that artist in a GridView with the same format as with the top tracks view (track cover and title, at least 20 results if available). If no result were returned, the app should indicate that to the user.

Screenshot here

If a user clicks on one of the entries into the GridView, a new activity should open up with that tracks cover, title, and three other pieces of information you can pick (**Hint**: You may need to make another API call for this information). Additionally, there should be an option to add the track to the playlist.

The second tab should have a users playlist. These should be songs that the user saved from searching for artists. The playlist should have the songs in ListView with the name of each saved song and it's artist. This data should be stored and fetched from a SQLite database.

Screenshot here

Requirements

- (10 points) The app displays the current top tracks in a GridView on startup
- (10 points) The app uses a tab bar with two tabs, one for searching for tracks and one for looking at the playlist
- (15 points) Data is pulled from the API and processed into a GridView on the main page. Makes use of a Fragment to display the results seamlessly.
- (15 points) Selecting a track from the GridView opens a new activity with the track cover, title, and 3 other pieces of information as well as the ability to save it to the playlist.
- (5 points) User can change search query by editing text field.
- (10 points) User can save a track to their playlist, and the track is saved using SQLite.
- (5 points) User can delete a track from the playlist (deleting it from the SQLite database itself).
- (10 points) Playlist is maintained between app launches.
- (4 points) App is visually appealing
- (1 point) Properly attribute Last.fm API as source of data.
- (5 points) Code is well formatted and commented.
- (10 points) All API calls are done asynchronously and do not stall the application.
- (15 points) Creative portion: Be creative! (**Hint**: You have an entire API available to you - use it!)

Helpful Advice

- Take a look at the studio code for how to make external API calls and process the JSON result.
- The API documentation is your friend - make sure to read it fully. Specifically, make sure you take a look at how to get your result formatted as a JSON.
- To populate a GridView, you will end up using an Adapter just like you did with a ListView with the last lab.
- This app will take a considerable amount of time. For testing purposes, consider using some dummy JSON data you save locally to test your GridView features instead of making an API call every single time.