Software Engineering and Management

DIT009 - Fundamentals of Programming

Assignment 2 - Music Analytics

You've been assigned to **experience the tasks of a data scientist**. Data science unites the knowledge of statistics, programming, and data visualisation to gather information and help others answer questions related to things or events around us. Here we will use an API (Application Programming Interface) to gather data
around music.

This assignment has the following learning goals:

- Basic operations in Programming
- Functions
- Lists and Dictionaries
- Files and Parsing File Formats (e.g., CSV, JSON, TXT)
- Regular Expressions
- Reusing libraries
- Good practices of programming

Your submission is composed of two parts submitted to Canvas:

- ☐ The Python code created is submitted as a ZIP file.
    - Your code will be graded based on functionality, readability, efficiency, and following good practices and Python conventions.
- ☐ The Project Report following the template shared in Canvas.
    - Your report will be graded based on clarity of the text, organisation, and cohesion.

Grading Criteria:

- ☐ **Grade 3:** Complete Part 1 with functional, readable, and maintainable code. The report should also include clear and cohesive text based on what is asked from the template.
- ☐ **Grade 4:** All criteria for 3, and **one** of the additional parts (Part 2 or 3).
- ☐ **Grade 5:** All criteria for 3, and **both** of the additional parts (Part 2 and 3).

Getting Started: Setting up Spotify API

The Spotify API allows you to retrieve data from Spotify, including your own account data (e.g., songs that you listen to often, your playlists, etc.). The documentation is extensive and overwhelming, so we recommend taking some time to browse examples.

- Browse through the Spotify API Documentation.

To make authentication easier, we use an application created by the awesome team of teaching assistants from DIT009 H24 **(❤️👏 special thanks to Răzvan Albu and Parisa Babaei).** Use the following URL format when sending your requests:

`https://dit009-spotify-assignment.vercel.app/api/v1/<query>`

Important: For security, and following Spotify's regulations, **you can only make 3 requests every 30 seconds**. Save the outcome of your requests in JSON files to ensure you don't lose data.

Examples of a request to get the albums of a specific artist:

`https://dit009-spotify-assignment.vercel.app/api/v1/artists/1HY2Jd0NmPuamShAr6KMms/albums`

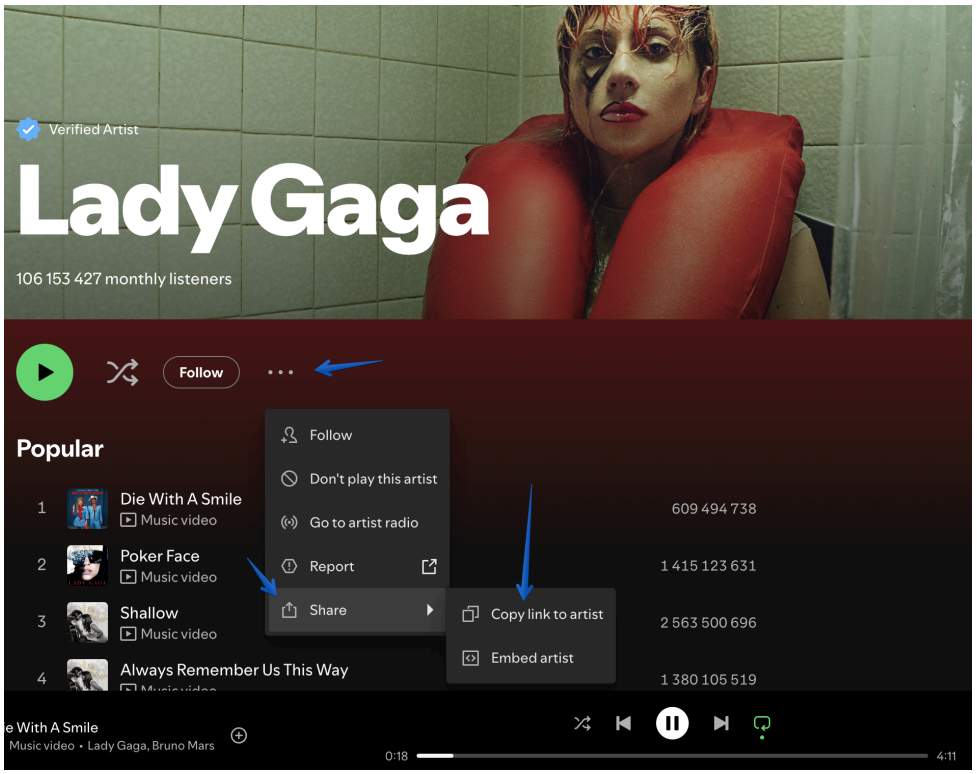
Example of code to using the dit009 application to retrieve a json:

```
import requests
import json

service = "https://dit009-spotify-assignment.vercel.app/api/v1"
url = f"{service}/artists/1HY2Jd0NmPuamShAr6KMms/albums"
response = requests.get(url)
data = response.json()
# Save that data into a JSON file
```

To get the ID of playlists, artists, or albums, you can use your Spotify web client. See the picture below for a reference. The Lady Gaga link is below, and highlighted you will find the id to use in the API request:

- https://open.spotify.com/artist/**1HY2Jd0NmPuamShAr6KMms**?si=ISbyMfLGRQy692xmt7QLMQ



Remember: The goal of the assignment is to **test your knowledge of programming, dictionaries, JSON, lists, file parsing, etc**. The API is mainly to gather data, and is not something you will be evaluated on. Rather, we just provide it for you to be creative on what type of data you can get and use it for a fun project. 🙂

Part 1: Analysis of Music Data

Goal: Create and answer 3 questions using the Spotify API data. You are open to use any artist, playlist, or album that you desire. Use this to experiment with your creativity. 🙂

Important:

- Due to the API authentication restrictions, **you cannot get personalised data** (e.g., your private playlists, etc.) You should use only Spotify public data.
- **Read Part 2** as well, in case you want to use multiple data sources and aim for a higher assignment grade.

To complete Part 1, your submission must:

- ☐ Save and read from JSON files. Try to minimise the API requests.
  - ☐ Tip: Use the **json.dump()** function from the json library to get the json as a string and write it in the file.
- ☐ Separate files between the API access and Data Analysis:
  - ☐ **spotify_api_miner.py:** Access the API and Save Files.
  - ☐ **music_data_analyser.py:** Load the saved files to do the analysis.
- ☐ **Include user interaction**, such as asking for the user to type an artist name, year, album, or the information related to your question.
- ☐ Submit the **code as a group** (See submission instructions in Canvas).
- ☐ Submit an **individual report**. The report **must include** the sections and information from this part required by the template (link in Canvas)

Tips and Advice:

- **Spotify offers live data**, so retrieving top artists or albums will likely change every day (or in the location). Be mindful of that when testing your outputs.
- Read the API documentation to see which attributes you can get for tracks, or albums. For instance, the tracks have **danceability**, each **artist has genres** associated with them, and you can **get related artists** or recommendations.
- If you are struggling to come up with questions, try using statistical measures or data indicating time trends. For instance, popularity over the years, during each month summer (Jun - Aug) for the past 5 years, etc.

Part 2: Using Datasets from Multiple Sources

Goal: Combine data from multiple sources to combine or correlate data. Here you should use data from **at least one other API** in your questions. For instance, lyrics API, or WikiMedia for page view access in Wikipedia pages (download JSON)

If you find a useful API then share it with your colleagues on Discord, or teachers. 🙂

To complete Part 2, your submission must:

- ☐ Use another API data to answer **at least 2 of your 3 questions**.
- ☐ Have code that properly reuses reading JSON files from different APIs, as well as independent functions to extract unique data from those additional data sources.
- ☐ The individual report should include the sections and information from this part required by the template.

Part 3: Improving Your Program

Goal: Use existing libraries and techniques to improve and expand your program.
Here you will play with two additional features of your project: Error handling, and Plotting.

To complete Part 3, your submission must include:

- ☐ The report should include the sections and information from this part required by the template.
- ☐ **Error Handling:** Create Error handling for your program. Namely:
  - ☐ JSON file is empty, or has the wrong format
  - ☐ Check for user input errors for the interactivity provided in Part 1.

- ☐ **Plotting:** Use the matplotlib library plot visualisation for your analysis.
  - ☐ Create at least one plot to visualise the data to help answer at least 2 questions from Part 1.
  - ☐ Save the plots as image files in your project resources folder (use the **png format**).