# 1 All about Data

## Background

Spotify is a world-wide music streaming service that allows users to listen to music anywhere at any time. The service offers several different APIs that are used to handle anything from playback to data collection. This project utilizes the Web API to import data from Spotify’s category playlists into a locally stored MongoDB database and generate playlists based on user input.

## Data Source

All data is sourced directly from Spotify using their Web API. This iteration of the project solely imports song data from the top 20 categories on Spotify. It is worth noting that certain categories provided by Spotify are deprecated on the platform, but the API will still generate collections for them.

## 1.3 Data Destination

All song data is imported and stored within collections generated based on the category name. Each of the 20 categories is stored in the `spotify` database with each collection currently having anywhere from 1.2k documents (songs) to 4.4k documents.

The Web API used for this project does have a limit on maximum requests per playlist creation scope and therefore a maximum of 100 songs/category can be added to a given playlist.

## 1.4 Data Transformation

No significant transformations were made to the data. The only additional field added to the data is the `import\_date` value which is a timestamp taken at the time of import to ensure that subsequent running of the code will not result in data imports each time.

# 2 Tooling

Jupyter Notebook and MongoDB are the tools currently used for this project. Due to technical issues with the authorization flow of the Spotify API, this project needed to be completed in a local environment and configuration of Spark on this local machine was not possible. In lieu of `pyspark`, the package `pymongo` was utilized to initiate data reads and writes to and from MongoDB. This would be a simple transition in future iterations of this project.

# 3 Code Completion

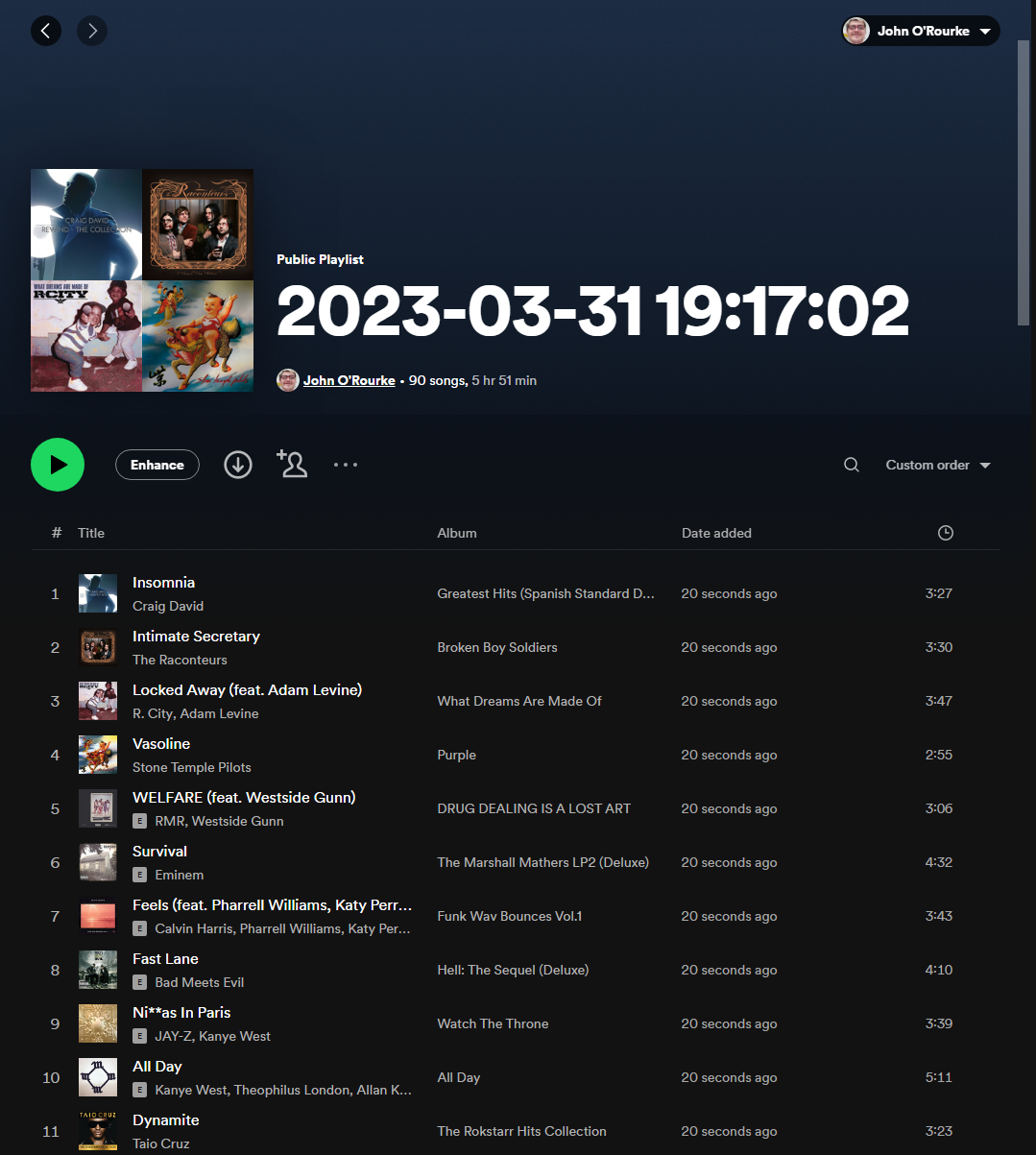
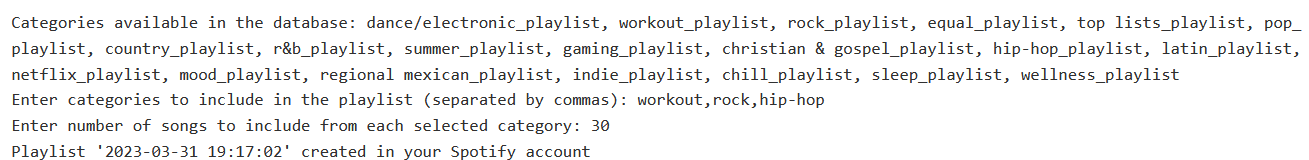
* MongoDB Configuration:Graphical user interface, text

  Description automatically generated
* Spotify API Authentication: Graphical user interface, text, application

  Description automatically generated
* Category collection creation: Graphical user interface, text, application

  Description automatically generated
* Song data import to collections: Text

  Description automatically generated
* Playlist Generation: Text

  Description automatically generated 

# 4 Conclusion

This project was very fulfilling although it had its roadblocks regarding using the local VM. The overall goal of this project was to utilize the Spotify API to import data into a MongoDB database as well as generate a playlist using the API. Both of these goals were achieved and the code used to achieve this goal can be further improved to generate analysis or a full fledged tool for playlist generation.