

Beykoz University

Department of “Computer Engineering”

“Systems Programming”

Final Project Report

Music Playlist Manager

Lecturer: Mustafa Yıldız

Leyla Abdullayeva - 1904010038

Music Playlist Manager - Python

Project name: Music Playlist Manager built in Python

Software Requirement -

Programming language – Python 3.8 (latest version is recommended)

Libraries - Requests & Json & OS library for music recommendation, SpotifyClient

library for getting all the Spotify Web API including access to all endpoints, and support for user authorization.

Project Description:

In this Python Project, my aim is to develop a “Music Playlist Generator” rather than manager. Spotify does a great job at providing us with many (and always fresh) playlists, to the point that it is basically 100% guaranteed that it satisfies us on any occasion. But, sometimes it is also lots of fun to generate our own playlists, according to whatever criteria we can think of. My project is about a simple Python application to create custom Spotify playlists using the Spotify Web API, based on seed tracks.

What development tools that I used in my project:

Python 3.8 (the latest version)

Google Colab

Requests & JSON & SpotifyClient and other machine learning libraries

Spotipy library

Spotify Web API (OAuth token) for developers

Functional Requirements:

- 1.The system requires Python language (the latest version)
- 2.The system requires the “Requests & Spotipy” library.
- 3.Project requires Spotify’s Web API for developers.

Overview:

The first thing we need to do is use the API which is a token and that allows us to authenticate our requests. To obtain this, I used this website

<https://developer.spotify.com/console/get-recommendations/> and got to the bottom and clicked on the “Get Token” button.

After selecting the scope that I need and logging in, I’ve got the code that I can copy-paste and use to authenticate the requests.

By doing this, I’ve setted some filters and query the Recommendations and Playlists endpoints to create and fill our playlist.

I worked directly with invoking the Web APIs, using the “Requests” library to make HTTP requests needed.

```
"Music Playlist Manager / Generator with Spotify API"
"Leyla Abdullayeva - 1904010038 - Computer Engineering"

import requests

# SETTINGS
endpoint_url = "https://api.spotify.com/v1/recommendations?"
token = "BQCyfBr2q7p_6gaXcFI2P7owSIk6DP0AHIn_i5AFPxXLzx85Uaar7IltAe8tmDbHAgrpPfcMFFQ_0RmBXrCuib2ERgPjGq_bw-0g2YKuuCVIjmQgB:"
user_id = "316u2o6h73qemlfdlbccen5ok4iy"

# OUR FILTERS
limit=30
market="US&UK"
seed_genres="slow"
target_danceability=1.0
uris = []
seed_artists = '4dpARuHxo5lG3z768sgnrY'
seed_tracks='3bNv3VuUOKgrf5hu3YouRo'

# PERFORM THE QUERY
query = f'{endpoint_url}limit={limit}&market={market}&seed_genres={seed_genres}&target_danceability={target_danceability}'
query += f'&seed_artists={seed_artists}'
query += f'&seed_tracks={seed_tracks}'

response = requests.get(query,
                        headers={"Content-Type": "application/json",
                                "Authorization": f"Bearer {token}"})
json_response = response.json()

print('Recommended Songs:')
for i,j in enumerate(json_response['tracks']):
    uris.append(j['uri'])
    print(f'{i+1}) \'{j['name']}\' by {j['artists'][0]['name']}')

```

```
Recommended Songs:
1) "Me Too" by Meghan Trainor
2) "Shape of You" by Ed Sheeran
3) "Santo" by Christina Aguilera
4) "Right As Rain" by Adele
5) "Don't Stop The Music" by Rihanna
6) "Rumors (feat. Cardi B)" by Lizzo
7) "California Gurls" by Katy Perry
8) "Memories" by Maroon 5
9) "Thinking out Loud" by Ed Sheeran
10) "Daydreamer" by Adele
11) "Only Girl (In The World)" by Rihanna
12) "What Makes You Beautiful" by One Direction
13) "If I Die Young (Glee Cast Version)" by Glee Cast
14) "Sugar" by Maroon 5
15) "Just Give Me a Reason (feat. Nate Ruess)" by P!nk
16) "Tired" by Adele
17) "Bitch Better Have My Money" by Rihanna
18) "Little Things" by One Direction
19) "Black Magic" by Little Mix
20) "Never Really Over" by Katy Perry
21) "Big Girls Don't Cry (Personal)" by Fergie
22) "My Same" by Adele
23) "S&M" by Rihanna
24) "Juice" by Lizzo
25) "Shout Out to My Ex" by Little Mix
26) "The One That Got Away" by Katy Perry
27) "Hey, Soul Sister" by Train
28) "Cry Your Heart Out" by Adele
29) "Pon de Replay" by Rihanna
30) "To Be So Lonely" by Harry Styles

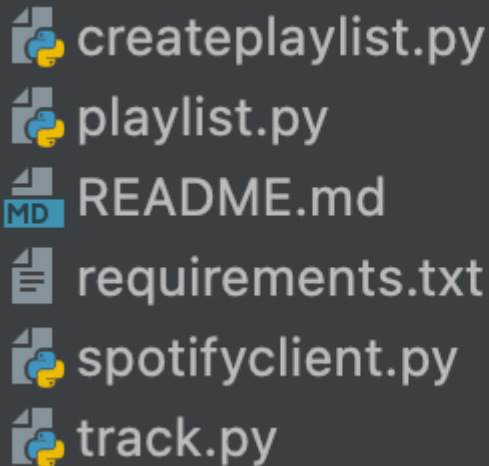
```

As you can see from the screenshot above, we have to access our token and user id of our Spotify account and we can also change filters which limit, market, genres and seed of artists & tracks & danceability. After arranging filters, we can perform our query with the help of json library.

By running our code, we can see the recommended songs which we limited to 30. Here’s the list:

I also tried to do this project by using different interference which means I've also used different libraries which are OS & SpotifyClient.

I have 4 different .py files total. After getting some music recommendations, we have to create and manage our playlist. For this I used playlist.py for representing the Spotify playlist. "track.py" represents the tracks which will be included in our playlist. In "spotifyclient.py" I also used the json & requests library in order to create a playlist. Our future playlist will include 50 songs. It's limited but we also can define the actual limit. I used the SpotifyClient class for this operation. Final step is doing operations in our "createplaylist.py" folder.



- createplaylist.py
- playlist.py
- README.md
- requirements.txt
- spotifyclient.py
- track.py

In this part below, I've imported SpotifyClient. Then, we have to run this file from the terminal by writing our Spotify Authorization token which we gained from the first step. Also we have to know our user ID after registering for spotify developer feature.

```
import os

from spotifyclient import SpotifyClient

def main():
    spotify_client = SpotifyClient(os.getenv("SPOTIFY_AUTHORIZATION_TOKEN"),
                                   os.getenv("SPOTIFY_USER_ID"))

    # get last played tracks
    num_tracks_to_visualise = int(input("How many tracks would you like to visualise? "))
    last_played_tracks = spotify_client.get_last_played_tracks(num_tracks_to_visualise)

    print(f"\n Here are the last {num_tracks_to_visualise} tracks you listened to on Spotify:")
    for index, track in enumerate(last_played_tracks):
        print(f"{index+1}- {track}")
```