Export Your Spotify Playlist to a CSV File Using Python

May 17 2022 · 6 min

[python web-scraping]

I love using Python 🐍 to automate boring tasks.

Recently, I got married 🥳 and had to provide a list of songs to the DJ. My wife and I love cheesy songs and put together a *long* playlist on Spotify with our favourite dancing tunes.

Unfortunately, there is no easy way to export the names and artists of the Spotify playlist (e.g. to a csv file) using the Spotify Player user interface.

This meant I would have to manually type out the track names and artist names in order to send the information to the DJ.

As a data engineer, of course that won't do. Why spend 15 minutes doing a boring manual task, when you can spend 30 minutes automating it (and another hour writing a blog post about it)!

Luckily, with some basic Python skills you can automate and extract playlist information using the Spotify API and Python library !

In this post, we create a simple script using the Spotify API to extract the song titles and artists from a Spotify playlist and save it to a nicely formatted csv file.

> 💻 All code for this post is provided in the e4ds-snippets GitHub repository

Getting Started with the Spotify Web API

Before we can start extracting data, we need to register our application and get credentials to access the Spotify Web API.

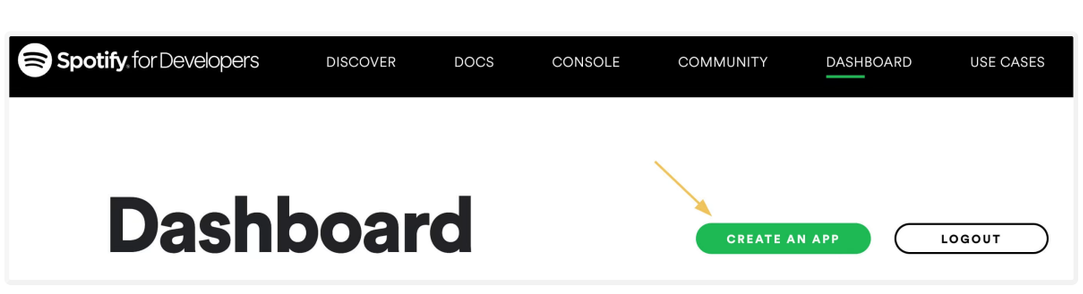
1. Log in to Spotify developer portal

Navigate to the Spotify developer portal and login using your normal Spotify username and password.

If you do not have an account you can create a new one – you don't need to have a paid subscription, the API still works for 'free' account users.

2. Create a new app

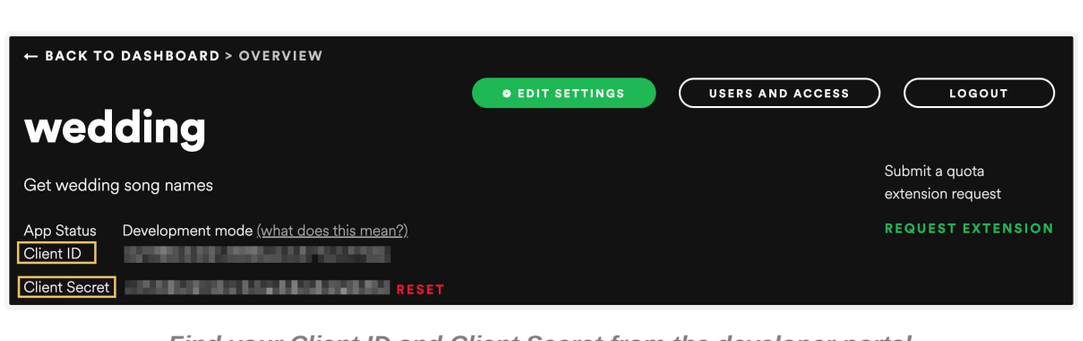
Register a new app in the developer console. It doesn't matter what name you give your app.



Create an app to generate credentials for your script

3. Get your credentials

In order to authenticate your Python application with the Spotify API, you need to get your authentication keys for the app you have just created – 'Client ID' and 'Client Secret'



Find your Client ID and Client Secret from the developer portal

4. Create an `.env` file to store Spotify API credentials

You should not store credentials directly in your scripts or in version control.

Therefore, create a new file called `.env` in your working directory and place your client id and secret. For example:

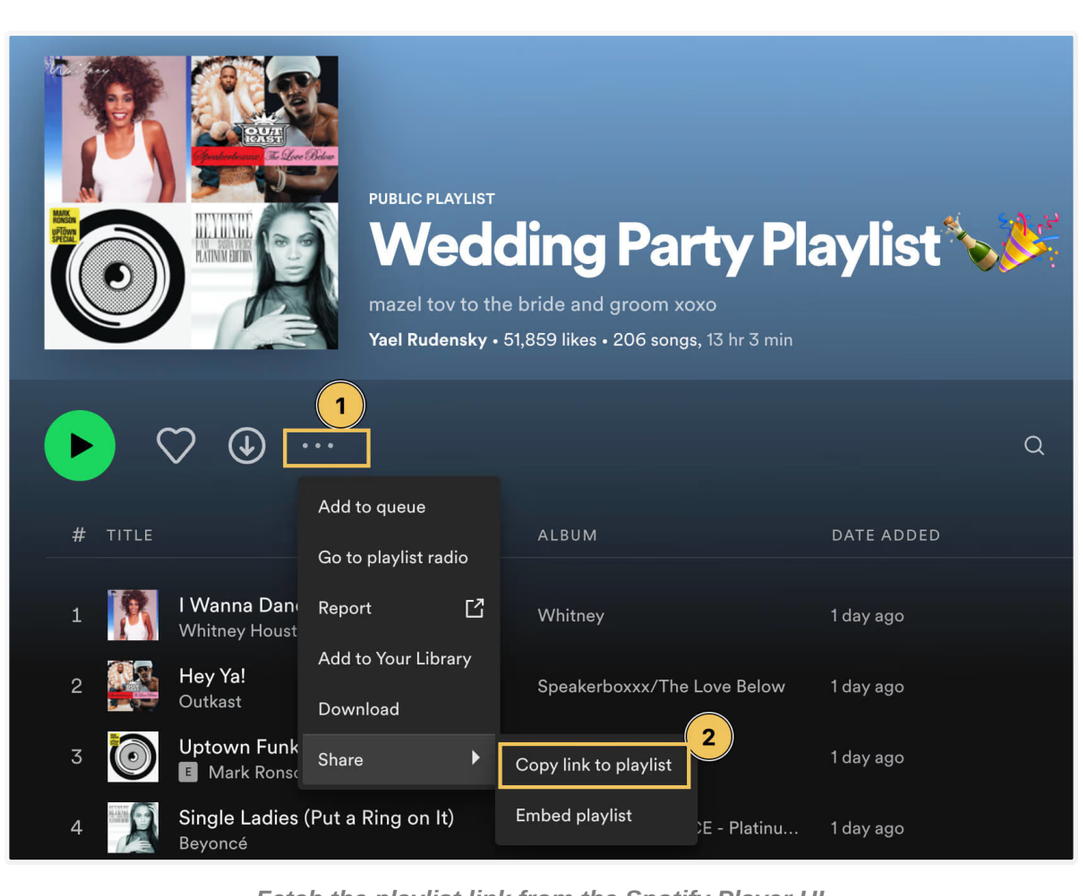
```
# .env example
CLIENT_ID='your_client_id'
CLIENT_SECRET='your_client_secret'
```

These credentials will be loaded into the script at runtime using the dotenv library .

Find Your Playlist Link

We want to use our Python script to extract information from a given playlist.

A playlist can be identified via its web address link, which can be found using the Spotify player user interface:



Fetch the playlist link from the Spotify Player UI

Copy the playlist link for your target playlist and save it for later.

Python Script 🐍

The following script can be used to extract the track names and artists and save into a CSV file.

> The full script can be found at the bottom of this post

1. Library Imports

Spotipy is a Python library which simplifies interacting with the Spotify Web API. We will also use the dotenv library to load our API credentials from the `.env` file.

Install these libraries using `pip` if you do not already have them installed:

```
# install dotenv and spotify
pip install python-dotenv spotipy
```

In addition, we will need to import the following libraries:

```
import csv
import os
import re

import spotipy
from dotenv import load_dotenv
from spotipy.oauth2 import SpotifyClientCredentials
```

2. Load credentials and define constants

Use the `load_dotenv()` function to load your credentials from your `.env` file as environment variables. We can then access the client ID and client secret values using the `os.getenv()` function.

Finally, add the web link to your target playlist.

```
# load credentials from .env file
load_dotenv()

CLIENT_ID = os.getenv("CLIENT_ID", "")
CLIENT_SECRET = os.getenv("CLIENT_SECRET", "")
OUTPUT_FILE_NAME = "track_info.csv"

# change for your target playlist
PLAYLIST_LINK = "https://open.spotify.com/playlist/6jAarBZaMmBLnSIeltPzkz?si=d42be
```

3. Create and authenticate your session

Next, we need to authenticate our session so we can retrieve data from the API.

```
# authenticate
client_credentials_manager = SpotifyClientCredentials(
    client_id=CLIENT_ID, client_secret=CLIENT_SECRET
)

# create spotify session object
session = spotipy.Spotify(client_credentials_manager=client_credentials_manager)
```

4. Extract URI from web link

The Spotify API actually requires us to pass the playlist *uri* to the API endpoint instead of the full web address that we copied earlier.

The playlist uri can be extracted from the web address link and is the part after 'playlist' and before '?'. For example:

```
# playlist web address link
https://open.spotify.com/playlist/6jAarBZaMmBLnSIeltPzkz?si=d42be5c6ec194bb9

# playlist uri
6jAarBZaMmBLnSIeltPzkz
```

We can extract this uri using a regular expression.

The code below extracts the uri from the provided web address and will raise a ValueError if no matches are found as this indicates the web address is not valid.

```
# get uri from https link
if match := re.match(r"https://open.spotify.com/playlist/(.*)\?", PLAYLIST_LINK):
    playlist_uri = match.groups()[0]
else:
    raise ValueError("Expected format: https://open.spotify.com/playlist/...")
```

5. Get playlist track information

We can retrieve the information about each track in a playlist using the `playlist_tracks` method of our session object and passing it the playlist uri.

```
# get list of tracks in a given playlist (note: max playlist length 100)
tracks = session.playlist_tracks(playlist_uri)["items"]
```

6. Extract data and save to CSV file

Now we can we can extract the information and save it in a nice format to a csv file.

The code below will open a csv file and write a header row. Then it loops through each track in the playlist, extracting the name and concatenating the artists into a single value (if there are multiple artists for a track). The extracted information is saved to a csv file called `track_information.csv` (defined by the `OUTPUT_FILE_NAME` constant above).

```
# create csv file
with open(OUTPUT_FILE_NAME, "w", encoding="utf-8") as file:
    writer = csv.writer(file)

    # write header column names
    writer.writerow(["track", "artist"])

    # extract name and artist
    for track in tracks:
        name = track["track"]["name"]
        artists = ", ".join(
            [artist["name"] for artist in track["track"]["artists"]]
        )

        # write to csv
        writer.writerow([name, artists])
```

Putting it all together: Full python script

```
"""Get song titles and artists from Spotify playlist"""

import csv
import os
import re

import spotipy
from dotenv import load_dotenv
from spotipy.oauth2 import SpotifyClientCredentials

# load credentials from .env file
load_dotenv()

CLIENT_ID = os.getenv("CLIENT_ID", "")
CLIENT_SECRET = os.getenv("CLIENT_SECRET", "")
OUTPUT_FILE_NAME = "track_info.csv"

# change for your target playlist
PLAYLIST_LINK = "https://open.spotify.com/playlist/6jAarBZaMmBLnSIeltPzkz?si=d42be5c6ec194bb9"

# authenticate
client_credentials_manager = SpotifyClientCredentials(
    client_id=CLIENT_ID, client_secret=CLIENT_SECRET
)

# create spotify session object
session = spotipy.Spotify(client_credentials_manager=client_credentials_manager)

# get uri from https link
if match := re.match(r"https://open.spotify.com/playlist/(.*)\?", PLAYLIST_LINK):
    playlist_uri = match.groups()[0]
else:
    raise ValueError("Expected format: https://open.spotify.com/playlist/...")

# get list of tracks in a given playlist (note: max playlist length 100)
tracks = session.playlist_tracks(playlist_uri)["items"]

# create csv file
with open(OUTPUT_FILE_NAME, "w", encoding="utf-8") as file:
    writer = csv.writer(file)

    # write header column names
    writer.writerow(["track", "artist"])

    # extract name and artist
    for track in tracks:
        name = track["track"]["name"]
        artists = ", ".join(
            [artist["name"] for artist in track["track"]["artists"]]
        )

        # write to csv
        writer.writerow([name, artists])
```

extract_spotify_playlist_info.py hosted with ❤ by GitHub view raw

After running the script, your playlist track names and artist information will be available in the `track_info.csv` file.

> See the e4ds-snippets GitHub Repo for the full example

Extensions

The script described in this post is very simple and only extracts the track name and artists.

The data available from the Spotify API is rich and contains a lot more information which can be used for a variety of interesting use cases and analyses. I recommend looking through the API docs and Python library docs to find out what information you can extract.

Happy coding!

Further Reading

- Matplotlib: Make Impactful Charts using plt.suptitle
- Reproducible ML: Maybe you shouldn't be using Sklearn's train_test_split
- Why is machine learning deployment so difficult in large companies
- How to set up an amazing terminal for data science with oh-my-zsh plugins
- Data Science Setup on MacOS (Homebrew, pyenv, VSCode, Docker)
- Five Tips to Elevate the Readability of your Python Code
- SQL-like Window Functions in Pandas
- Do Programmers Need to be able to Type Fast?

Tags: [python web-scraping]

« PREV PAGE
🎂 One year of blogging: Growing the blog to 10k monthly visitors

NEXT PAGE »
Matplotlib: Make Impactful Charts by Adding Subtitles with plt.suptitle

2 Comments - *powered by utteranc.es*

MakePerceive commented on May 11, 2023

thank you for posting

EricWhite8452 commented 2 months ago

Thank you so much for sharing. Your post was so detailed. Only exporting Spotify playlist to Excel/CSV may not satisfy me. I also export the Spotify playlists to local MP3 files via AMusicSoft Spotify Music Converter for eternal offline access.