

Spotify API

CSCI 331

Shane Costello
Sage James



How we decided to approach this project:



Using Flask to Implement

efficient and scalable to handle user requests



Creating User Interface

allows users to interact effortlessly



Spotify Access Token

how to gain access, where to find documentation

[Spotify for Developers](#)



WHY SPOTIFY API



User Data Retrieval



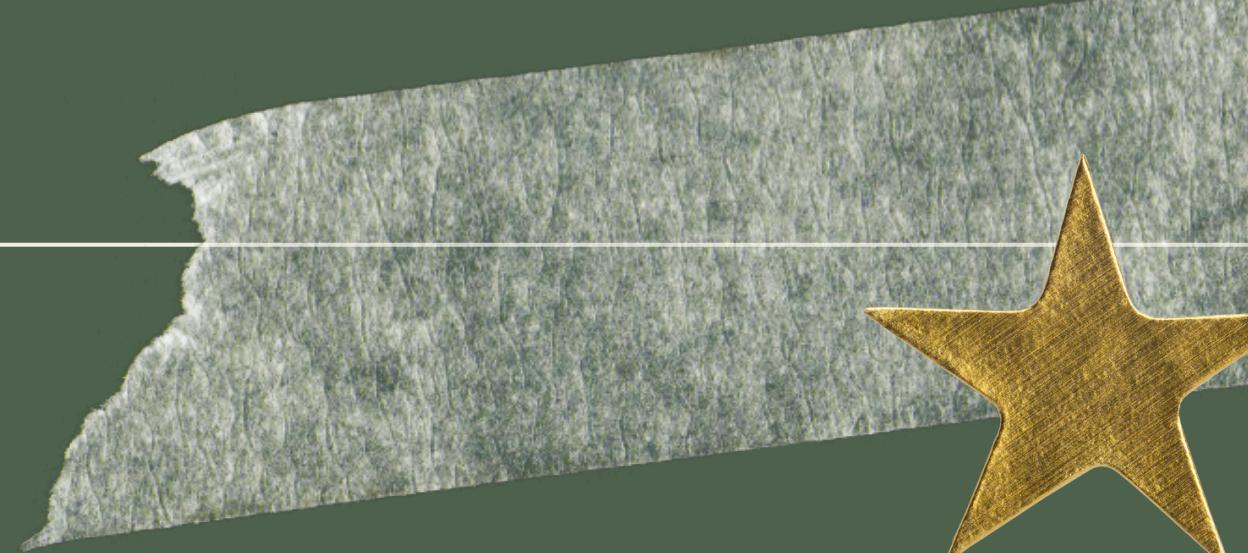
Playlist Management



Access to the Music Catalog



Seamless Integration



Create app

App name

App description

Website

Redirect URI

A URI where users can be redirected after authentication success or failure

Which API/SDK are you planning to use?

Web API

[Read more about Web API](#)

Web Playback SDK

[Read more about Web Playback SDK](#)

Android

[Read more about Android](#)

Ads API

[Read more about Ads API](#)

iOS

[Read more about iOS](#)

I understand and agree with Spotify's [Developer Terms of Service](#) and [Design Guidelines](#)

Save

Cancel

CREATING AN ACCESS TOKEN

Create an App on Spotify

this will get you a client ID and a client secret



Our Approach

742126



Using Flask to implement the API
HTML and CSS to style the site

```
from flask import Flask, request, url_for, session, redirect, render_template
import spotipy
from spotipy.oauth2 import SpotifyOAuth

#DEFINING CONSTS
CLIENT_ID = ...
CLIENT_SECRET = ...
TOKEN_INFO = "token_info"
SECRET_KEY = "asdf"
SHORT_TERM = "short_term"
MEDIUM_TERM = "medium_term"
LONG_TERM = "long_term"

def create_spotify_oauth():
    return SpotifyOAuth(
        client_id = CLIENT_ID,
        client_secret = CLIENT_SECRET,
        redirect_uri = url_for("redirectPage", _external=True),
        scope = "user-top-read user-library-read"
    )

def get_token():
    token_info = session.get(TOKEN_INFO, None)
    return token_info

app = Flask(__name__)
app.secret_key = SECRET_KEY

@app.route("/")
def index():
    name = 'username'
    return render_template('index.html', title='Welcome', username=name)
```

IMPORTS & CONSTANTS

#DEFINING CONSTS

```
CLIENT_ID =  
CLIENT_SECRET =  
TOKEN_INFO = "token_info"  
SECRET_KEY = "asdf"  
SHORT_TERM = "short_term"  
MEDIUM_TERM = "medium_term"  
LONG_TERM = "long_term"
```

```
from flask import Flask, request, url_for, session, redirect, render_template  
  
import spotipy  
from spotipy.oauth2 import SpotifyOAuth
```



```
@app.route("/")
def index():
    name = 'username'
    return render_template('index.html', title='Welcome', username=name)
```

1

```
@app.route("/login")
def login():
    sp_oauth = create_spotify_oauth() *
```

3

```
def create_spotify_oauth():
    return SpotifyOAuth(
        client_id = CLIENT_ID,
        client_secret = CLIENT_SECRET,
        redirect_uri = url_for("redirectPage", _external=True),
        scope = "user-top-read user-library-read"
    )
```



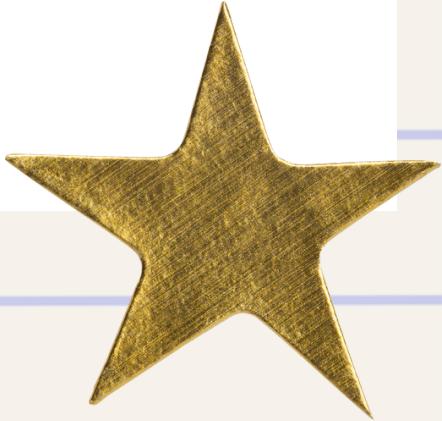
GETTING
STARTED...



Spotify Integration

Top Track Generator

Login in with spotify



index.html

```
</head>
<body>
<center>
    <h1>Spotify Integration</h1>
    <h2>Top Track Generator</h2>

    <button class="loginButton" onclick="location.href = '/login';">Login in with spotify</button>
</center>
</body>
</html>
```

2

Now that we are logged in, we need to create access tokens

```
@app.route("/redirectPage")
def redirectPage():
    sp_oauth = create_spotify_oauth()
    session.clear()
    code = request.args.get('code')
    token_info = sp_oauth.get_access_token(code)
    session[TOKEN_INFO] = token_info
    return redirect(url_for("getTracks", _external=True))
```

*session shows the unique instance we are currently in

```
@app.route("/getTracks")
def getTracks():
    user_token = get_token()

    sp = spotipy.Spotify(
        auth = user_token['access_token']
    )

    current_user_name = sp.current_user()['display_name']

    short_term = sp.current_user_top_tracks(
        limit=10,
        offset=0,
        time_range=SHORT_TERM
    )

    medium_term = sp.current_user_top_tracks(
        limit=10,
        offset=0,
        time_range=MEDIUM_TERM
    )

    long_term = sp.current_user_top_tracks(
        limit=10,
        offset=0,
        time_range=LONG_TERM,
    )

    return render_template('tracks.html', user_name = current_user_name, short_term=short_term, medium_term=medium_term, long_term=long_term)
```

HOW TO GET THE DATA



```
def get_token():
    token_info = session.get(TOKEN_INFO, None)
    return token_info
```



```
<h1>Spotify Integration</h1>

<h2>Hello, {{ user_name }}</h2>

<button onclick="showLastMonthTracks()">Last Month's Top Tracks</button>
<button onclick="showLastSixMonthsTracks()">Last 6 Months' Top Tracks</button>
<button onclick="showAllTimeTracks()">All Time Top Tracks</button>

<div id="lastMonth">
  <h3>Last Month Top Tracks:</h3>
  <div class="container">
    <ul>
      {% for track in short_term['items'] %}
        <div class="item">
          <br>
          {{ track['name'] }} <br> {{ track['artists'][0]['name'] }}
        </div>
      {% endfor %}
    </ul>
  </div>
</div>
```

TRACKS.HTML



JavaScript

```
<body onload="showLastMonthTracks()">

<script>
  function showLastMonthTracks() {
    document.getElementById('lastMonth').style.display = 'block';
    document.getElementById('6months').style.display = 'none';
    document.getElementById('allTime').style.display = 'none';
  }
</script>
```