

## MISC

- set up spotify for developers dashboard, provides client id and client secret for api calls
- created .env file, stores client id, client secret, and app secret key as variables for reference in flask app
- created .gitignore file so that .env file is not public, client secret cannot be public

## app.py

```
1 from flask import Flask, redirect, url_for, render_template
2 from dotenv import load_dotenv
3 import os
4
5 #load .env file, get client secret, id
6 load_dotenv()
7 client_id = os.getenv("CLIENT_ID")
8 client_secret = os.getenv("CLIENT_SECRET")
9
10 REDIRECT_URI = "http://localhost:5000/callback"
11
12 #initialize web app
13 app = Flask(__name__)
14 app.secret_key = os.getenv("APP_SECRET_KEY")
```

- imported necessary libraries and packages
- client secret and id are pulled from .env file via load\_dotenv()
- redirect uri = the address where the user is sent back after successfully logging in, spotify sends the user here with the access token which is used to make api calls, will come in handy later
- flask app is initialized via Flask(\_\_name\_\_), app secret key is then assigned to it via env file, app secret key is necessary for security

```
15
16 #spotify log in page
17 AUTH_URL = "https://accounts.spotify.com/authorize"
18
19 #url to refresh token
20 TOKEN_URL = "https://accounts.spotify.com/api/token"
21
22 #homepage
23 @app.route('/')
24 def index():
25     return render_template('index.html')
26
27 #run app
28 if __name__ == '__main__':
29     app.run(host='0.0.0.0', debug=True)
```

- set up spotify authorization url and token url for later use
- set up home page (just shows the message “hi” for now)
- app is run locally with debug mode on

## base.html/index.html

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1.0">
6     {% block head %} {% endblock %}
7 </head>
8 <body>
9     {% block body %} {% endblock %}
10 </body>
11 </html>
```

- base.html has all the boilerplate html code with a bit of jinja (jinja refers to the stuff like “block head” and “endblock”) so that it can be transferred to other html files without having to type it out all over again for each file

```
1 {% extends "base.html" %}
2
3 {% block head %}
4 <title>Semicolon Expected</title>
5 {% endblock %}
6
7 {% block body %}
8 <h1>hi</h1>
9 {% endblock %}
```

- index.html displays the message “hi”

## **SUMMARY**

I set up the basics of flask, the spotify api dashboard, and a the urls needed to authorize the user via spotify/keep them logged in with a refresh token