

IUPUI

---

Webmasters Hack-a-thon

Genre.fy

Documentation

Andy Jang, Nathan Kohlman, Gregory Long

---

## Goal

The website's goal is to recommend to users some artists based on the top artists listened to. To do this, we must establish authentication with the Spotify API and then use the API to get information back and display it to the user. Theoretically, this web app should be infinite and the user should be able to keep clicking on recommended artists and their songs until they feel like they want to stop.

## Page Info

The prototypes for the pages can be found at:

<https://xd.adobe.com/view/741014f2-d084-4c06-8965-7e99a44fb2fa-2143/>

This section will go into further detail about the purpose of each page.

### *Page 1: Login*

This is the landing page for the website and the one that all users will be brought to as soon as they try to access the web application. From here there are 2 different routes that the user should be able to click.

1. The button "start here" will send the user to the official Spotify login page for us to request access to their information.
2. [MAYBE] The text "need help?" will send the user to a help page that we will design to explain what the website does and how to navigate through the website.
3. [EXTRA] Adding links in the footer of the website will allow us to add sections such as source code and about us.

### *Page 2: Initial Recommendation*

This page is the result of a successful login to Spotify. The artist in the middle of the recommendation is the one that the user has listened to the most. From here, some branches connect the artists to smaller sub-portraits that allow the user to click on them to change the artist in the middle to the one clicked.

For the artist in the middle, they are to have their:

1. Name
2. Top 5 songs

Sub-portraits do not need the top 5 songs, just their name.

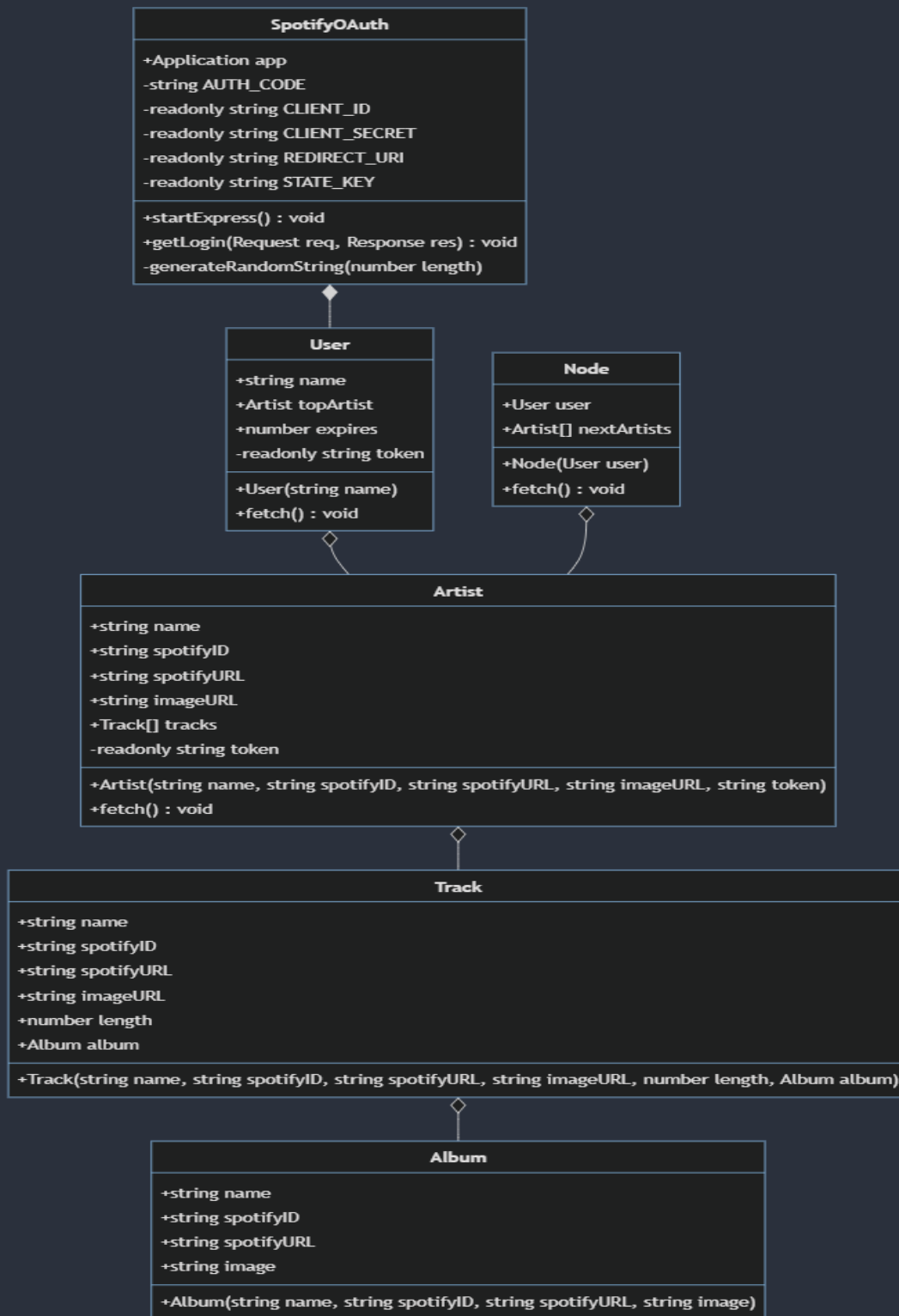
*Page 3: Recommendation after a click*

This page is the result of the user clicking on another artist. The artist that has been clicked will take the focus in the middle and their top 5 songs will be displayed. New recommendations will be generated.

*Page ?: Error page for login*

The page is the result of an unsuccessful login into Spotify. In this case, we should just let the user know of the error and offer a button to send them back.

**Data Structures**



The data of the program was split into classes and used aggregation to share data between each class.

class Album

Album
+string name
+string spotifyID
+string spotifyURL
+string image
+Album(string name, string spotifyID, string spotifyURL, string image)

class Node

Node
+User user
+Artist[] nextArtists
+Node(User user)
+fetch() : void

class Track

Track
+string name
+string spotifyID
+string spotifyURL
+string imageURL
+number length
+Album album
+Track(string name, string spotifyID, string spotifyURL, string imageURL, number length, Album album)

class Artist

Artist
+string name +string spotifyID +string spotifyURL +string imageURL +Track[] tracks -readonly string token
+Artist(string name, string spotifyID, string spotifyURL, string imageURL, string token) +fetch() : void

class Artist

User
+string name +Artist topArtist +number expires -readonly string token
+User(string name) +fetch() : void

class SpotifyOAuth

SpotifyOAuth
+Application app -string AUTH_CODE -readonly string CLIENT_ID -readonly string CLIENT_SECRET -readonly string REDIRECT_URI -readonly string STATE_KEY
+startExpress() : void +getLogin(Request req, Response res) : void -generateRandomString(number length)

## Config

There are some dependencies to start working on this project.

1. Install the latest, **stable** version of Node.js
2. Install TypeScript by running this command in the console (PowerShell, Git Bash, etc.) "npm install -g typescript"
3. TypeScript files are denoted by a .ts extension, to make a JavaScript or .js file out of the TypeScript file run the command "tsc filename.ts"
4. Run "npm install" to install the dependencies for the program.
5. To run the program locally, use the command, "node run test-server"

Note: you can both compile and run a program by using the command:

"tsc filename && node filename"

## GitHub Policy

1. Do not commit directly to the main branch, this will cause a lot of problems down the road.
2. Every change that you make will have to be on your own branch.
3. Once you believe that the branch is ready for merging, start a pull request.
4. After the pull request is approved, then the branch will automatically be merged and deleted.

## Notice

Sadly, as we ran on a limited amount of time, there is no way to publish this website for the public at the moment to get out of Spotify's "development mode" we must submit a quote extension request which would take at the minimum 6 weeks. Instead, for this project, we are running in development mode and only authorized users will be able to run the project.

If you would like to try and run the app, please contact **boooiil#1931** on Discord.

For now, as we are using a public token, the repo is private, if you would like access, contact Andy ([andyjang@iu.edu](mailto:andyjang@iu.edu))