Natsukasiy (Documentation)

General Design and Description

Description: The ASP .NET Core (Razor Pages) application is based on the **Spotify API** (Spotify for Developers - a special site from Spotify creators for young developers that requires strict adherence to the terms of service, including the prohibition of commercial use of the application). On this site, after registering their application, a developer receives their own **Client_ID** and **Secret_ID** (these IDs are keys to accessing the Spotify database as well as authorization and other essential functions).
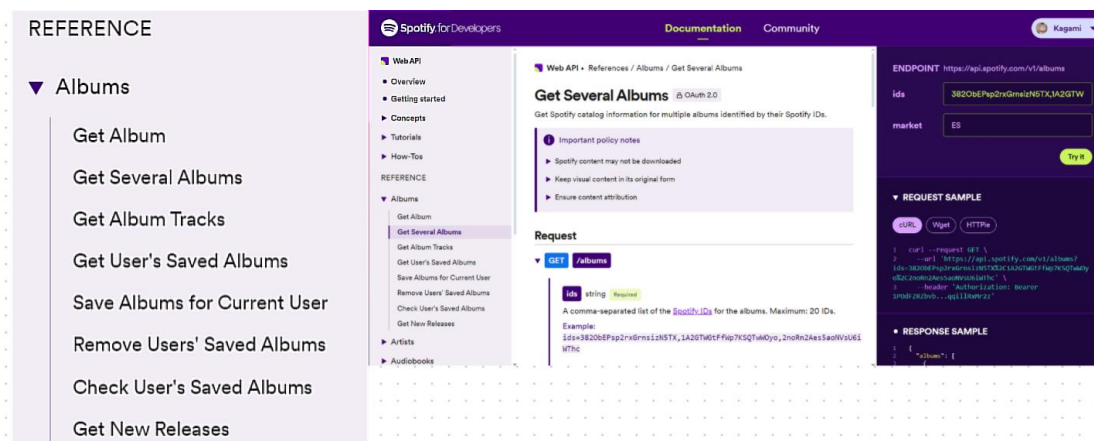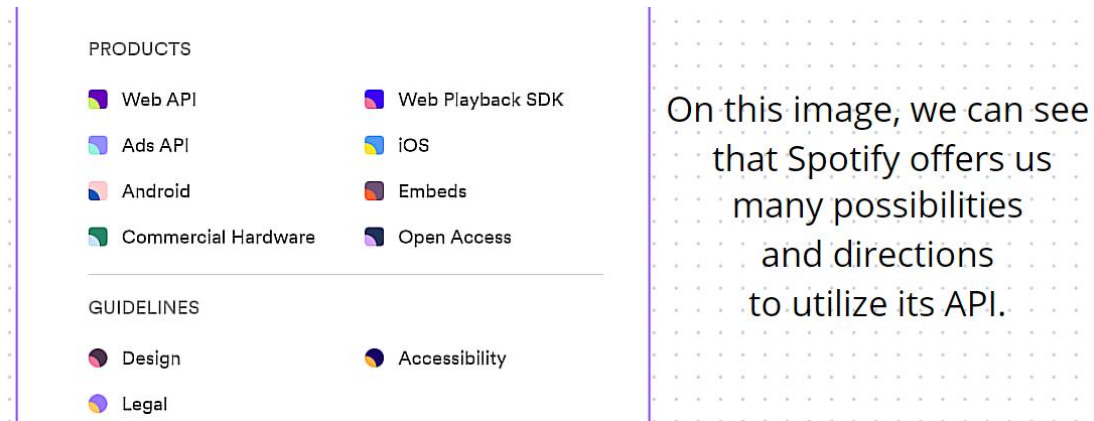
Below, I present the general theory of **how the program works** using the **Spotify API**:

To use the Spotify API, we need:

- ✓ Premium Account
- ✓ Register our application on the Spotify For Developers website
- ✓ Receive the necessary IDs to use the API.
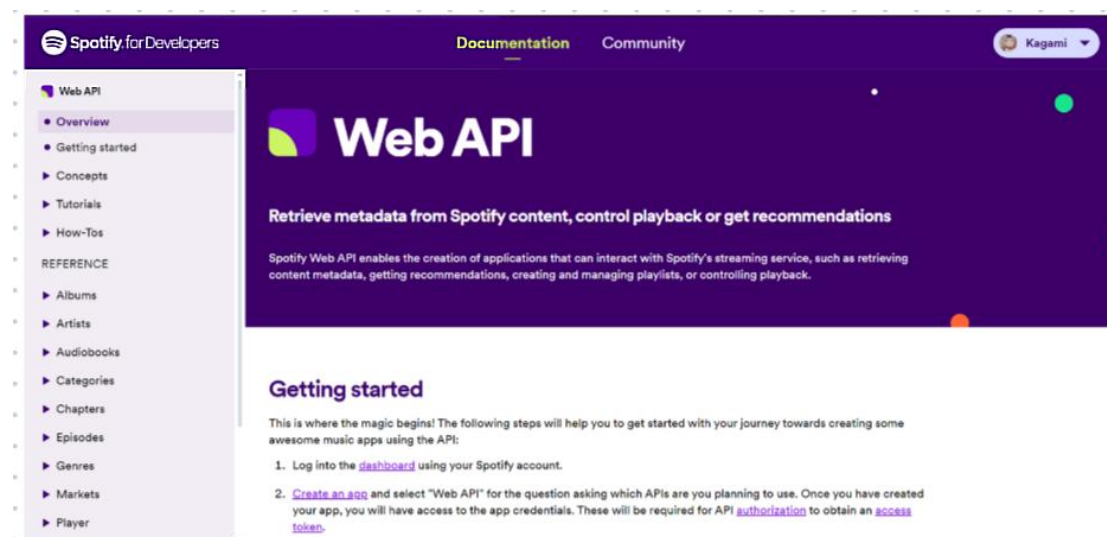
In each section, we have a list of what we can use with instructions on how it works and what we need.

PRODUCTS

Web API   Web Playback SDK

Ads API   iOS

Android   Embeds

Commercial Hardware   Open Access

GUIDELINES

Design   Accessibility

Legal

On this image, we can see that Spotify offers us many possibilities and directions to utilize its API.

REFERENCE

▼ Albums

Get Album

Get Several Albums

Get Album Tracks

Get User's Saved Albums

Save Albums for Current User

Remove Users' Saved Albums

Check User's Saved Albums

Get New Releases

In the section with our application, we can see various statistics:

Dashboard

Create app

Condensed view

N Natsukasiy
Project for university

Daily Active Users

Monthly Active Users
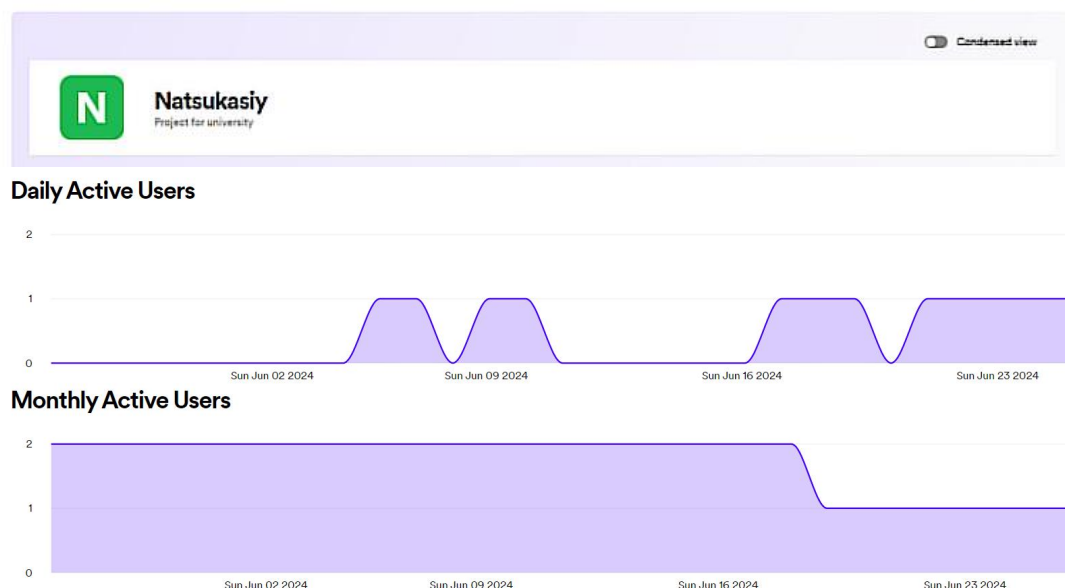
API requests by endpoint

Legend endpoints:
`/v1/me/player` `/v1/me/player/pause` `/v1/recommendations` `/v1/search` `/v1/browse/featured-playlists` `/v1/me` `/v1/shows/<show-id>` `/v1/browse/categories/<category-id>/playlists` `/v1/browse/categories` `/v1/playlists/<playlist-id>/tracks` `/v1/me/player/devices` `/v1/me/player/play` `/v1/me/player/currently-playing` `/v1/tracks/<track-id>` `/v1/me/player/next` `/v1/episodes/<episode-id>` `/v1/me/player/previous` `/v1/me/player/volume` `/v1/me/player/seek`
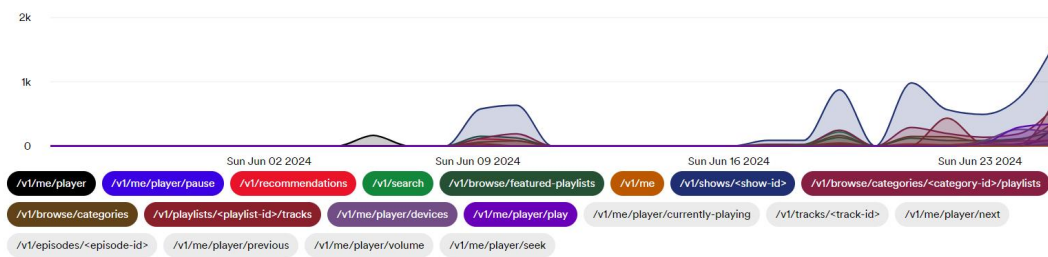
We also have application settings with IDs and necessary items (redirectUri):

Client ID

7eae723d9e134d2b9a4516d0ca88a7d9

App Status

Development mode

View client secret

App name

Natsukasiy

App description

Project for university

Website

Redirect URIs

- https://localhost:7297/callback
- https://ivanbezushko.github.io/Uri/index.html
- https://ivanbezushko.github.io/Uri/index_web.html

I would like to take this opportunity to explain how the application works:

✓ Once the program has been initiated, the authorization process is initiated as well.

✓ The user then has the opportunity to confirm their identity.

✓ Spotify then uses the redirect URI to send the response.

✓ In my case, the redirect URI is an index_web.html file on GitHub (specifically shown in the folder), which processes this response into a code and sends it to the appropriate localhost port.

✓ The program then receives this code and converts it into an access token, which grants the program access to a full range of functions.

I would like to bring to your attention a few issues that I have encountered.

WPF

➢ In the case of WPF, it seems that track ID interception does not always work, resulting in the wrong track being played.
➢ Handling episodes is a challenge because Spotify does not treat them as tracks, making some functions difficult to adapt.
➢ XAML is not HTML/CSS, so there is room for improvement.

Razor Pages

➢ Razor Pages have their own particular nuances, and without experience, it can be challenging to fully comprehend how the final program should look.
➢ While not all functions are yet fully compatible with episodes, there has been encouraging progress.

In conclusion...

In conclusion, I feel that this group of projects (WPF+Razor) provided me with valuable experience and presented a significant challenge for someone new to these technologies. While some aspects could have been refined further, I am overall satisfied with this enjoyable challenge. I hope you will appreciate it too.

Below is the appearance of the application after Spotify authorization. **The application's functionality will be demonstrated in recordings** that **will be included in the folder**.

Zezwól na połączenie Spotify z:

Natsukasiy

Kagami
To nie Ty?

Wyrażasz zgodę na to, by Natsukasiy mógł:

👁 Wyświetlanie danych Twojego konta Spotify
- Twój adres e-mail
- Twoja subskrypcja Spotify, kraj konta i ustawienia filtrowania treści nieprzyzwoitych
- Twoje imię i nazwisko, nazwa użytkownika, zdjęcie profilowe, obserwujący w Spotify oraz publiczne playlisty

📊 Wyświetlanie Twojej aktywności w Spotify
- Odtwarzana zawartość oraz urządzenia Spotify Connect

📝 Podejmowanie w Twoim imieniu działań w Spotify
- Sterowanie Spotify na Twoim urządzeniu

Zgadzam się

Anuluj

W każdej chwili możesz odebrać te uprawnienia w ustawieniach swojego konta.
Polityka prywatności Natsukasiy zawiera szczegółowe informacje na temat

---

Natsukasiy

🏠 Home
🔍 Browse

Internet People
Internet People

ViralHits

Home Page

RAP GE... | Hot Hit... | Hity z r... | Hity Lat... | viral rap | Viral Hity | MEGA ... | Po...

Loyal — Chris Brown	4:25
Sinner — Deaf Havana	3:27
One Last Breath — Creed	3:58
Here — Alessia Cara	3:19

Register
Login

Loyal
Chris Brown

Loyal
Chris Brown

Browse page

Natsukasiy

Home
Browse

All Out 80s

Przeglądaj muzykę

RAP GENERAC...   Hot Hits Polska   Hity z radia 20...   Hity Lata 2024   viral rap   Vir...

Made For You   New Releases   Summer   Hip-Hop   Pop   Mo...

Register
Login

Loyal
Chris Brown

Loyal
Chris Brown

Registration Page

Natsukasiy

Home
Browse

Letni chill

Registration

User Nick

Email

Password

Confirm Password

Register
Login

Loyal
Chris Brown

Loyal
Chris Brown

Login Page

Natsukasiy

Home
Browse

Release Radar

Logining

Email

Password

Login

Register
Login

Loyal
Chris Brown

Loyal
Chris Brown