Tutorialr.com

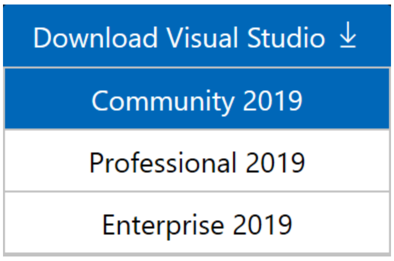
Spotify for Developers

Getting Started

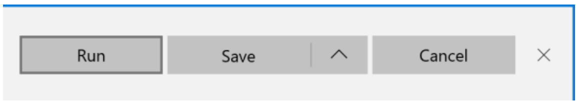
Setup & Start

Step 1

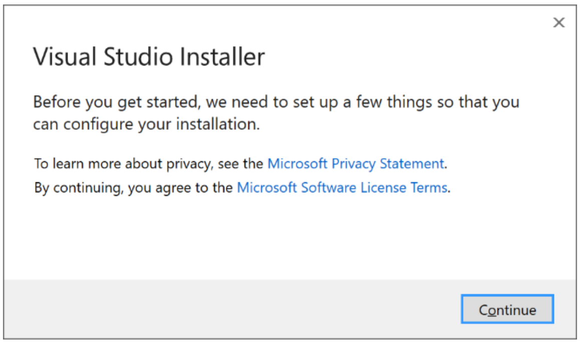
Firstly, you will need to install **Visual Studio 2019 Community**, if not done already by doing the following:



Visit **VisualStudio.com** and then from the **Visual Studio IDE** section choose **Download Visual Studio** then **Community 2019**



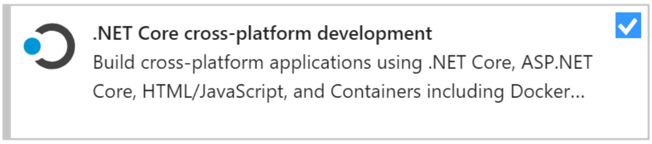
Next on the **Thank you for downloading Visual Studio** page when the download prompt appears, select **Run**


Visual Studio Installer

Before you get started, we need to set up a few things so that you can configure your installation.

To learn more about privacy, see the Microsoft Privacy Statement.
By continuing, you agree to the Microsoft Software License Terms.

Once downloaded, this should start the **Visual Studio Installer** and select **Continue** to begin the installation

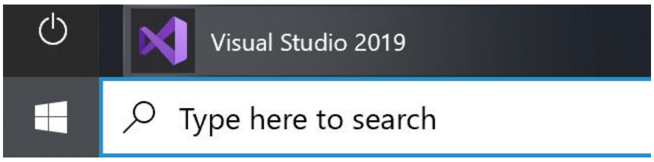

☑ .NET Core cross-platform development

Build cross-platform applications using .NET Core, ASP.NET Core, HTML/JavaScript, and Containers including Docker...

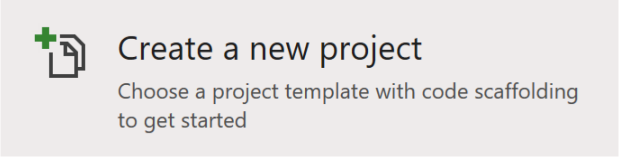
Next once ready select **.NET Core cross-platform development** from the **Workloads** section

Step 2

Next if **Visual Studio 2019 Community** is installed, you can then start **Visual Studio 2019 Community** and **Create a new project**, by doing the following:

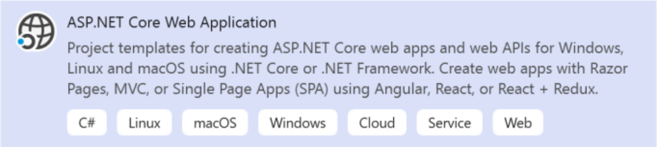


In **Windows 10** choose **Start**, and then from the **Start Menu** find and select **Visual Studio 2019**


Create a new project

Choose a project template with code scaffolding to get started

Once done, from the **Get started** screen for **Visual Studio 2019** select **Create a new project**

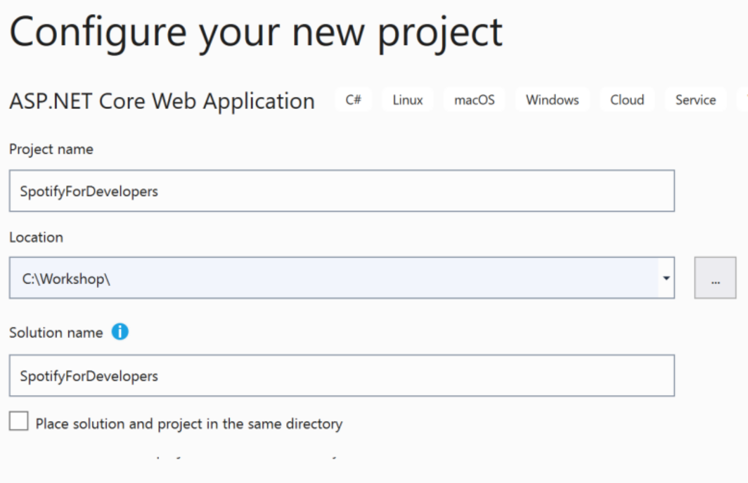

ASP.NET Core Web Application

Project templates for creating ASP.NET Core web apps and web APIs for Windows, Linux and macOS using .NET Core or .NET Framework. Create web apps with Razor Pages, MVC, or Single Page Apps (SPA) using Angular, React, or React + Redux.

Then choose **ASP.NET Core Web Application** and select **Next**

Step 3

Next, in **Configure your new project** enter a **Project name** as **SpotifyForDevelopers** and then choose a **Location** and then select **Create**



Configure your new project

ASP.NET Core Web Application

Project name: SpotifyForDevelopers

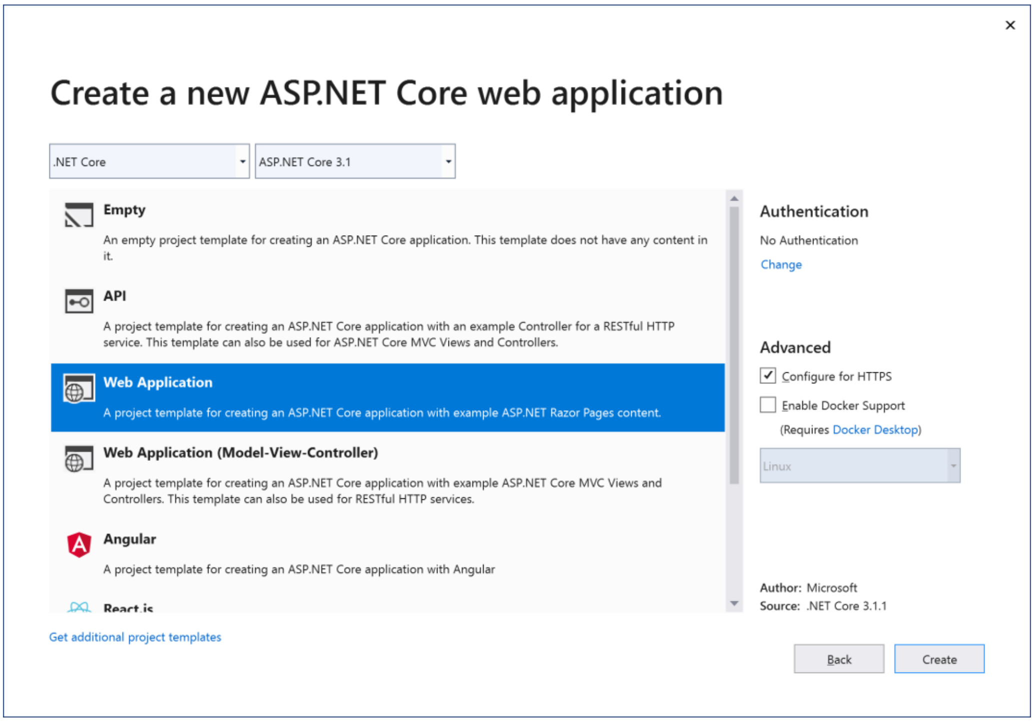
Location: C:\Workshop\

Solution name: SpotifyForDevelopers

□ Place solution and project in the same directory

Step 4

Then, in **Create a new ASP.NET Core web application** make sure from the two dropdowns that **.NET Core** and **ASP.NET Core 3.1** are selected, and **Web Application** is selected from the list and in **Advanced** configure for HTTPS has been selected then select **Create**



Create a new ASP.NET Core web application

.NET Core | ASP.NET Core 3.1

Empty
An empty project template for creating an ASP.NET Core application. This template does not have any content in it.

API
A project template for creating an ASP.NET Core application with an example Controller for a RESTful HTTP service. This template can also be used for ASP.NET Core MVC Views and Controllers.

Web Application
A project template for creating an ASP.NET Core application with example ASP.NET Razor Pages content.

Web Application (Model-View-Controller)
A project template for creating an ASP.NET Core application with example ASP.NET Core MVC Views and Controllers. This template can also be used for RESTful HTTP services.

Angular
A project template for creating an ASP.NET Core application with Angular

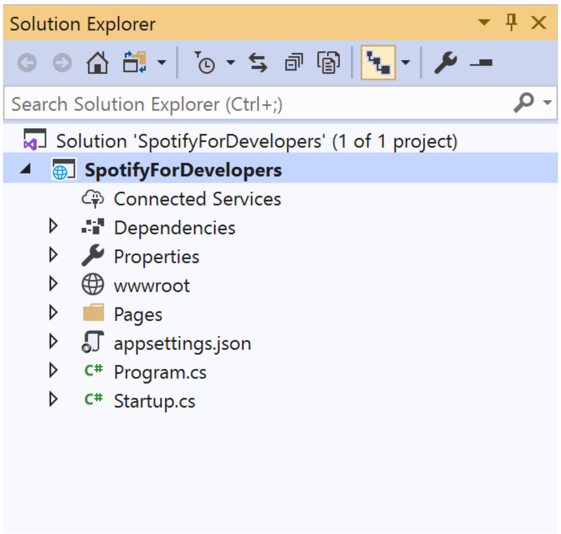
React.js

Authentication
No Authentication
Change

Advanced
☑ Configure for HTTPS
□ Enable Docker Support
(Requires Docker Desktop)
Linux

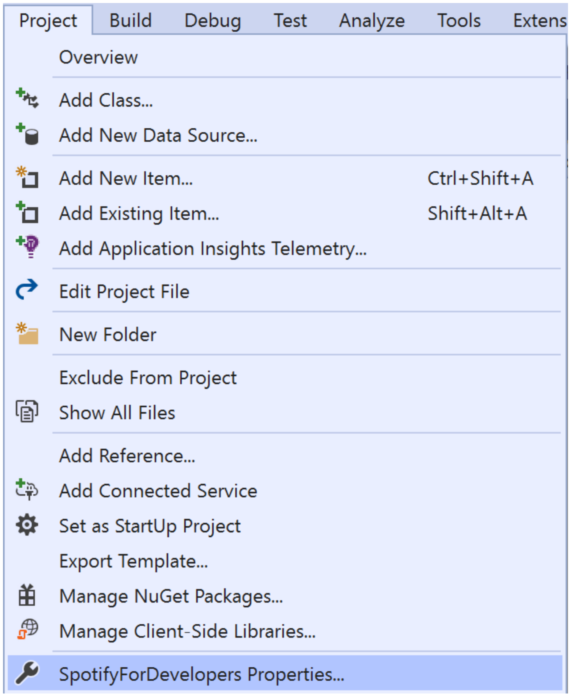
Author: Microsoft
Source: .NET Core 3.1.1

Step 5



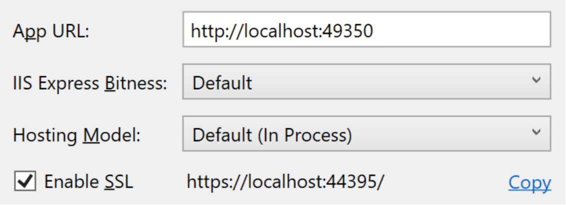
In the **Solution Explorer** of **Visual Studio 2019** select the **Project** for **SpotifyForDevelopers**

Step 6



From the **Menu** of **Visual Studio 2019** select **Project** then **SpotifyForDevelopers Properties...**

Step 7



App URL: http://localhost:49350

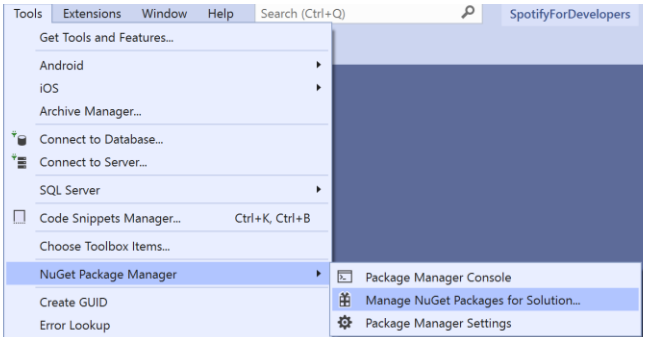
IIS Express Bitness: Default

Hosting Model: Default (In Process)

☑ Enable SSL https://localhost:44395/

Then in **Properties** select the **Debug** section and then in the **Web Server Settings** select the **<u>Copy</u>** option to **Copy** the URL to the **Clipboard** e.g. https://localhost:44395/ - please note that your URL may be different. Then **Paste** the contents into your text editor such as **Notepad** this will be used as a **Redirect URI** for use with the **Spotify API**

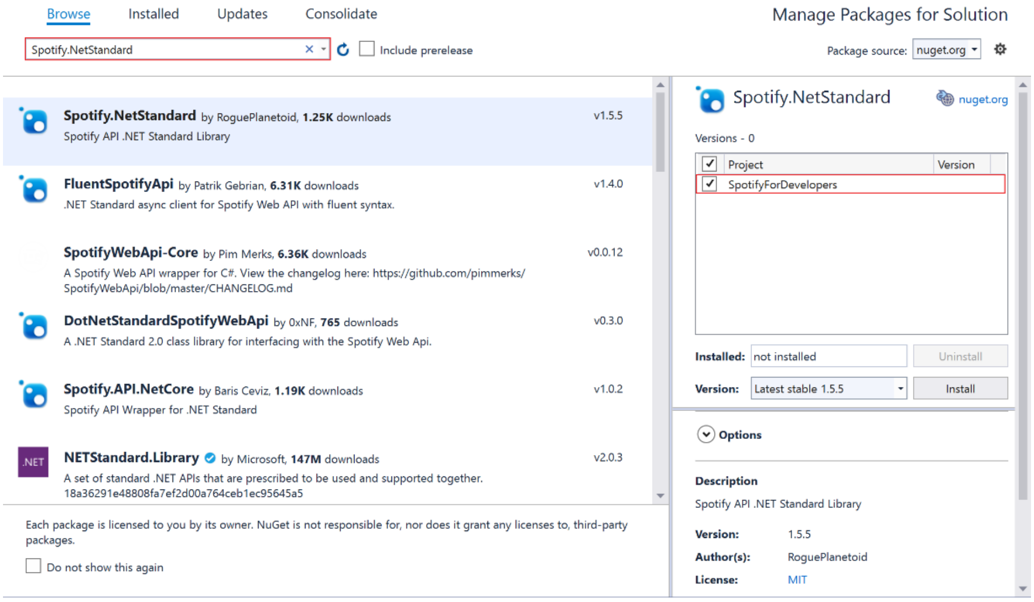
Step 8



From the **Menu** of **Visual Studio 2019** select **Tools** then select **NuGet Package Manager** and **Manage NuGet Packages for Solution...**

Step 9

Then in **NuGet** select **Browse** and search for **Spotify.NetStandard by RoguePlanetoid** as indicated and select **Spotify.NetStandard** then check the box under **Project** as indicated and select **Install**.



Spotify.NetStandard by RoguePlanetoid, **1.25K** downloads v1.5.5
Spotify API .NET Standard Library

FluentSpotifyApi by Patrik Gebrian, **6.31K** downloads v1.4.0
.NET Standard async client for Spotify Web API with fluent syntax.

SpotifyWebApi-Core by Pim Merks, **6.36K** downloads v0.0.12
A Spotify Web API wrapper for C#. View the changelog here: https://github.com/pimmerks/SpotifyWebApi/blob/master/CHANGELOG.md

DotNetStandardSpotifyWebApi by 0xNF, **765** downloads v0.3.0
A .NET Standard 2.0 class library for interfacing with the Spotify Web Api.

Spotify.API.NetCore by Baris Ceviz, **1.19K** downloads v1.0.2
Spotify API Wrapper for .NET Standard

NETStandard.Library by Microsoft, **147M** downloads v2.0.3
A set of standard .NET APIs that are prescribed to be used and supported together. 18a36291e48808fa7ef2d00a764ceb1ec95645a5

Each package is licensed to you by its owner. NuGet is not responsible for, nor does it grant any licenses to, third-party packages.

□ Do not show this again

Spotify.NetStandard nuget.org

Versions - 0

☑ Project	Version
☑ SpotifyForDevelopers	

Installed: not installed Uninstall

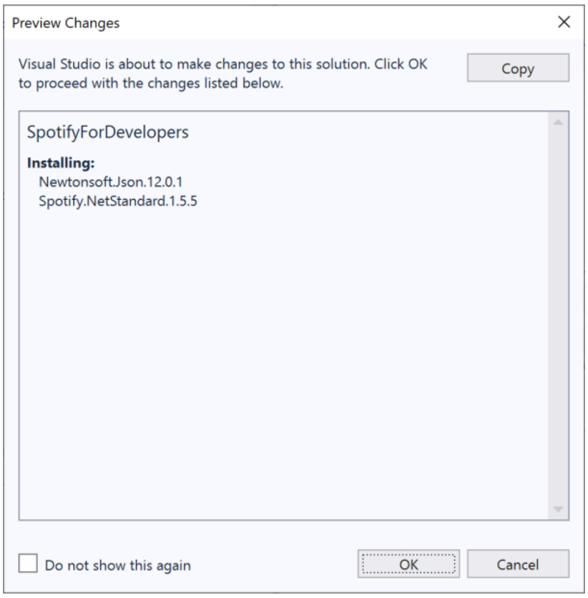
Version: Latest stable 1.5.5 Install

Options

Description
Spotify API .NET Standard Library

Version: 1.5.5
Author(s): RoguePlanetoid
License: MIT

Step 10



Visual Studio is about to make changes to this solution. Click OK to proceed with the changes listed below.

SpotifyForDevelopers
Installing:
Newtonsoft.Json.12.0.1
Spotify.NetStandard.1.5.5

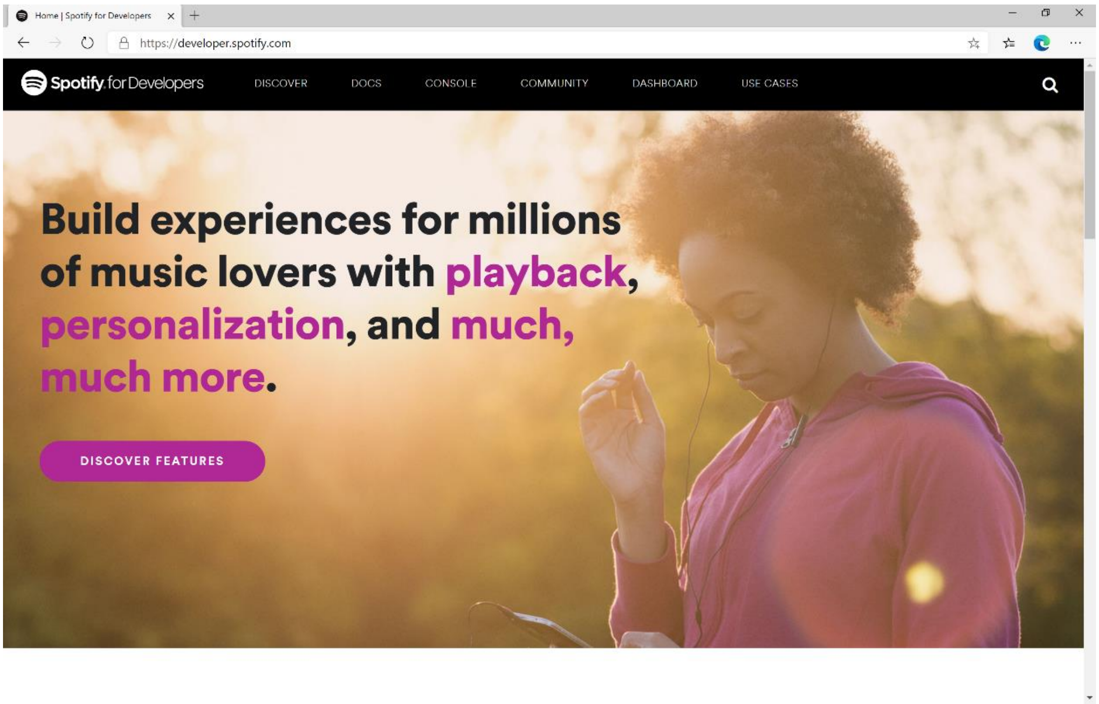
□ Do not show this again

Then, if **Preview Changes** is displayed, select **OK** then or otherwise the **NuGet** package **Spotify.NetStandard** will be installed and then make sure to keep **Visual Studio 2019** open as you'll come back to it later.

Dashboard & Settings

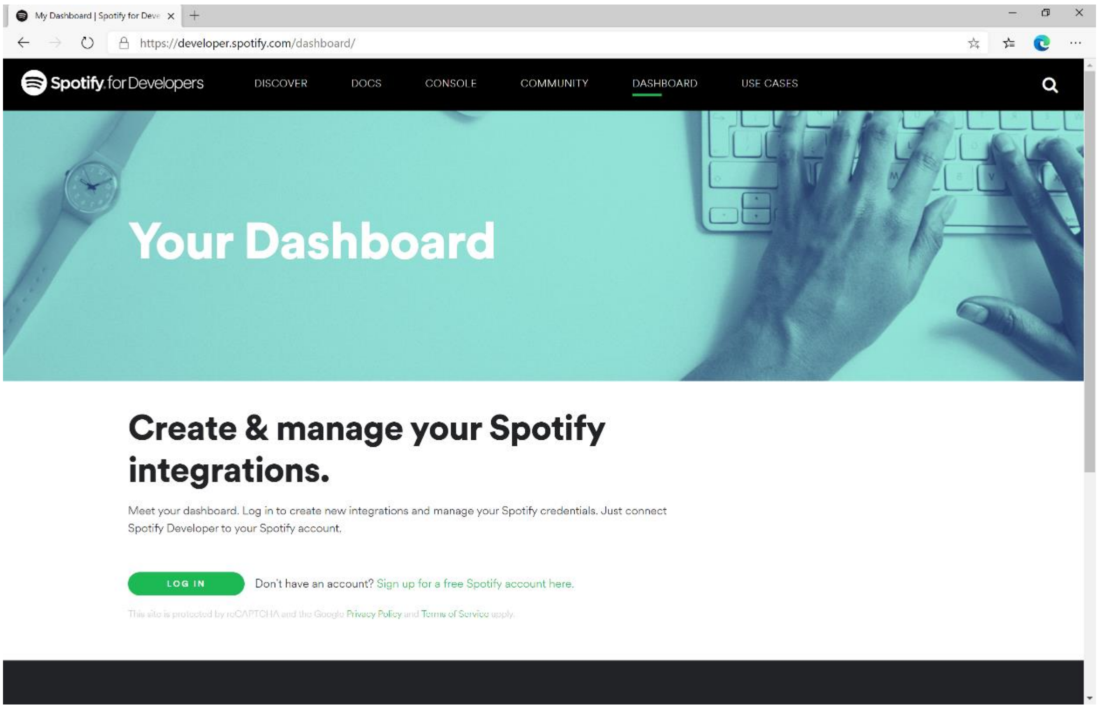
Step 1

Start your favourite **Web Browser** such as **Microsoft Edge** and navigate to **developer.spotify.com** for the **Spotify for Developers** website.


Home | Spotify for Developers
https://developer.spotify.com
Spotify for Developers
DISCOVER
DOCS
CONSOLE
COMMUNITY
DASHBOARD
USE CASES
Build experiences for millions of music lovers with playback, personalization, and much, much more.
DISCOVER FEATURES

Step 2

Then on the **Spotify for Developers** website select **Dashboard** then on the **Your Dashboard Page** choose **Log In** to sign in with a **Spotify Account**



Create & manage your Spotify integrations.

Meet your dashboard. Log in to create new integrations and manage your Spotify credentials. Just connect Spotify Developer to your Spotify account.

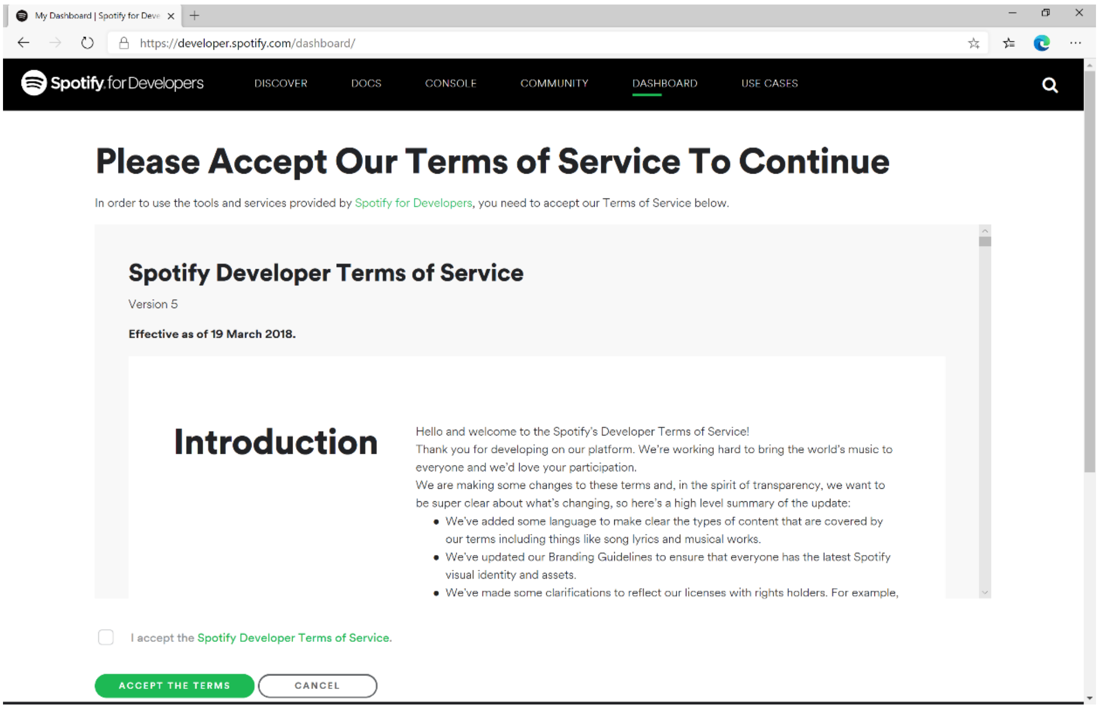
Don't have an account? Sign up for a free Spotify account here.

This site is protected by reCAPTCHA and the Google Privacy Policy and Terms of Service apply.

If you don't have a **Spotify** account already you can use the **Sign up for a free Spotify account here** option to get a **Spotify** account – you will already have one if you have a **Spotify** subscription or already listen to their music service.

Step 3

Once you've signed in with a **Spotify** account for the first time you will need to read through the **Spotify Developer Terms of Service** then once done select the **I accept the Spotify Developer Terms of Service** and then select **Accept the Terms** to continue.



Please Accept Our Terms of Service To Continue

In order to use the tools and services provided by Spotify for Developers, you need to accept our Terms of Service below.

Spotify Developer Terms of Service

Version 5

Effective as of 19 March 2018.

Introduction

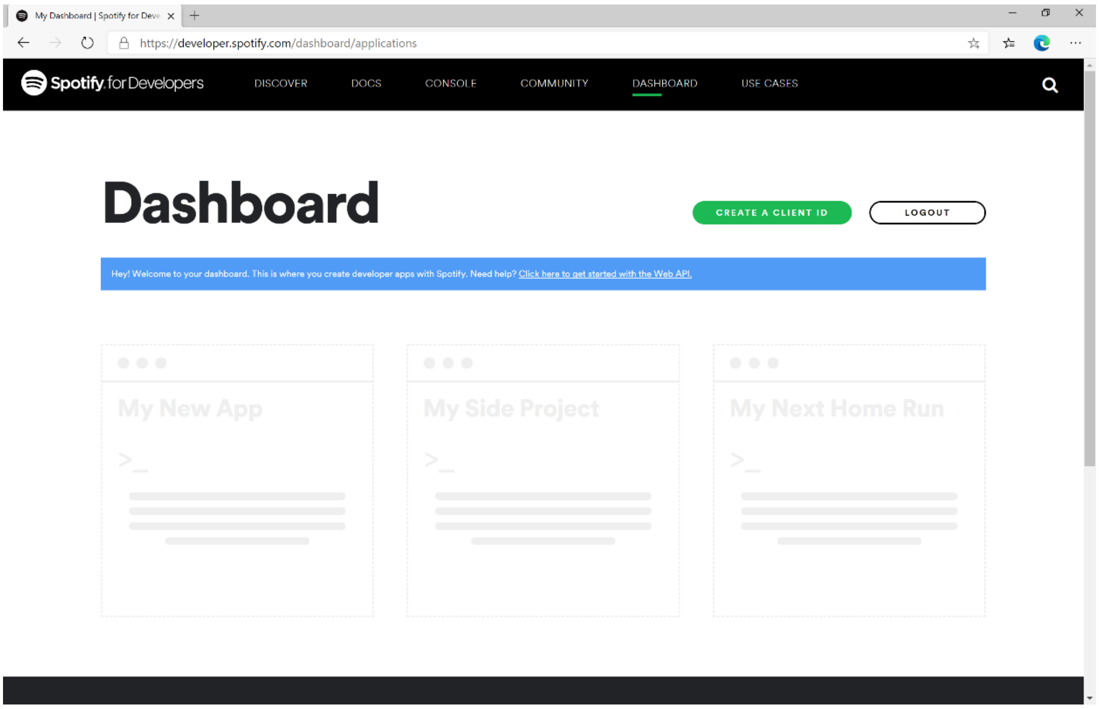
Hello and welcome to the Spotify's Developer Terms of Service!
Thank you for developing on our platform. We're working hard to bring the world's music to everyone and we'd love your participation.
We are making some changes to these terms and, in the spirit of transparency, we want to be super clear about what's changing, so here's a high level summary of the update:

- We've added some language to make clear the types of content that are covered by our terms including things like song lyrics and musical works.
- We've updated our Branding Guidelines to ensure that everyone has the latest Spotify visual identity and assets.
- We've made some clarifications to reflect our licenses with rights holders. For example,

□ I accept the Spotify Developer Terms of Service.

Step 4

Once signed in with a **Spotify** Account and agreed to the **Spotify Developer Terms of Service** you'll be taken to the **Dashboard**

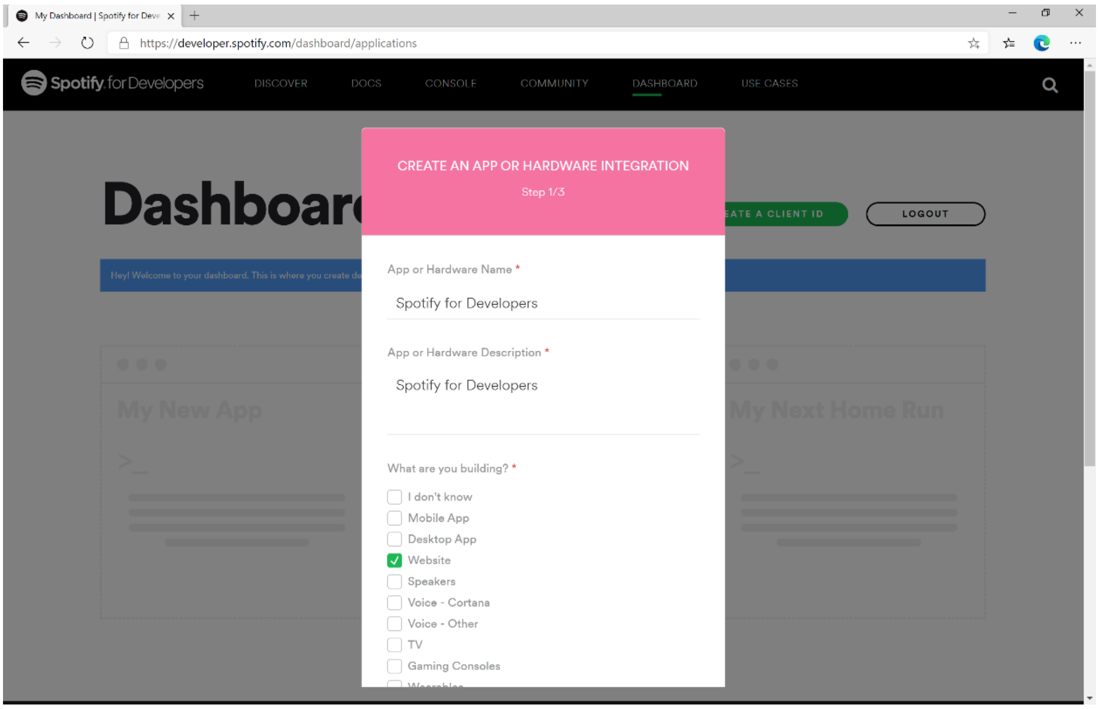


Dashboard

Hey! Welcome to your dashboard. This is where you create developer apps with Spotify. Need help? Click here to get started with the Web API.

Step 5

Then in the **Dashboard** select **Create a Client Id** then in **Step 1/3** of **Create an App or Hardware Integration** you need to enter the **App or Hardware Name** as **Spotify for Developers** and the **App or Hardware Description** as **Spotify for Developers** then under **What are you building** tick the **Website** option and select **Next**.


My Dashboard | Spotify for Deve
https://developer.spotify.com/dashboard/applications
Spotify for Developers
DISCOVER
DOCS
CONSOLE
COMMUNITY
DASHBOARD
USE CASES
Dashboard
EATE A CLIENT ID
LOGOUT
Hey! Welcome to your dashboard. This is where you create d
My New App
My Next Home Run

CREATE AN APP OR HARDWARE INTEGRATION

Step 1/3

App or Hardware Name *

Spotify for Developers

App or Hardware Description *

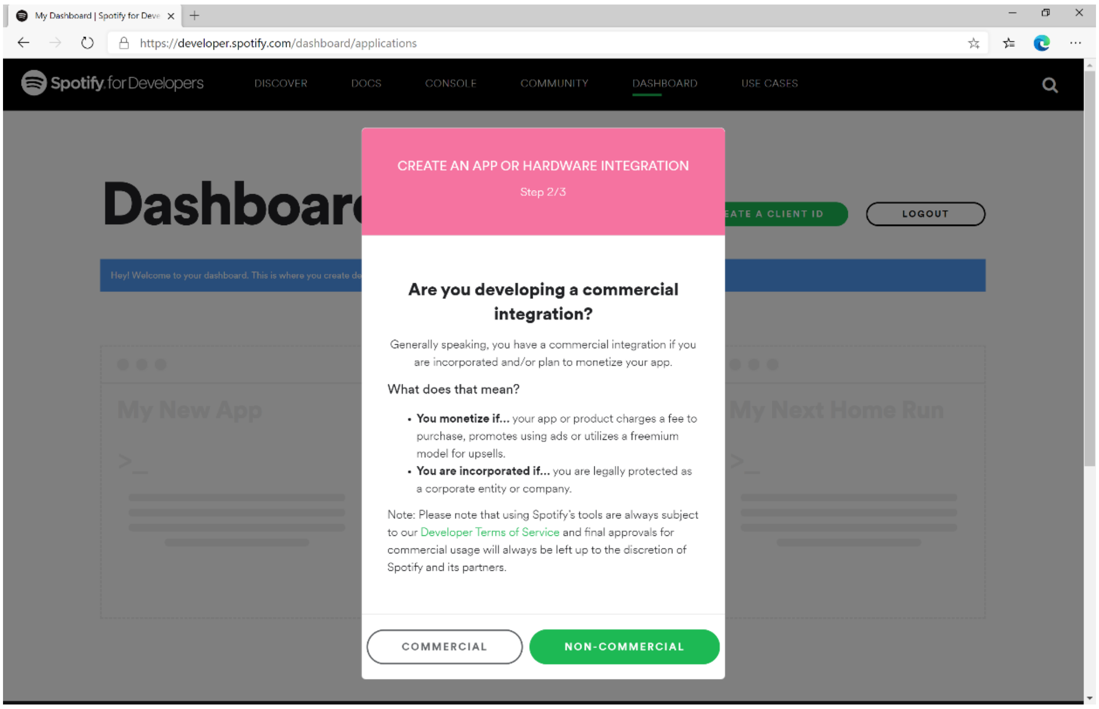
Spotify for Developers

What are you building? *

- □ I don't know
- □ Mobile App
- □ Desktop App
- ☑ Website
- □ Speakers
- □ Voice - Cortana
- □ Voice - Other
- □ TV
- □ Gaming Consoles

Step 6

Then in **Step 2/3** of **Create an App or Hardware Integration** for **Are you developing a commercial integration** select **Non-Commercial**.



CREATE AN APP OR HARDWARE INTEGRATION

Step 2/3

Are you developing a commercial integration?

Generally speaking, you have a commercial integration if you are incorporated and/or plan to monetize your app.

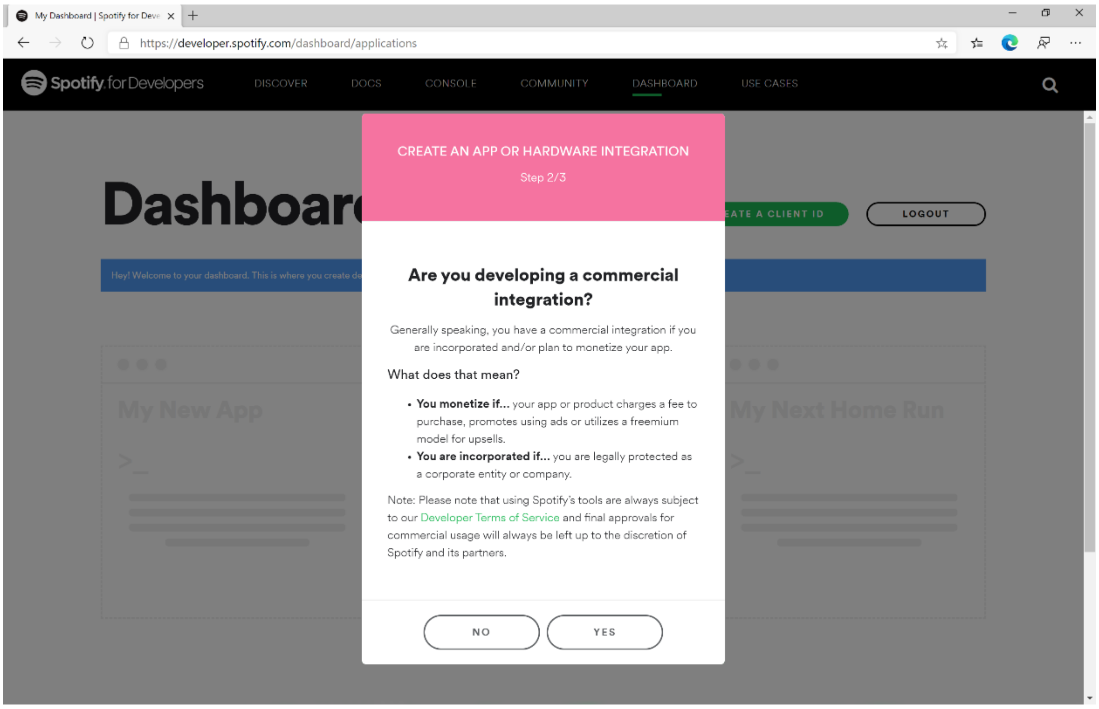
What does that mean?

- **You monetize if...** your app or product charges a fee to purchase, promotes using ads or utilizes a freemium model for upsells.
- **You are incorporated if...** you are legally protected as a corporate entity or company.

Note: Please note that using Spotify's tools are always subject to our Developer Terms of Service and final approvals for commercial usage will always be left up to the discretion of Spotify and its partners.

Step 7

Then in **Step 2/3** of **Create an App or Hardware Integration** for the question **Are you developing a commercial integration** select **No**.


My Dashboard | Spotify for Deve
https://developer.spotify.com/dashboard/applications
Spotify for Developers
DISCOVER
DOCS
CONSOLE
COMMUNITY
DASHBOARD
USE CASES
Dashboard
CREATE A CLIENT ID
LOGOUT
Hey! Welcome to your dashboard. This is where you create
My New App
My Next Home Run

CREATE AN APP OR HARDWARE INTEGRATION

Step 2/3

Are you developing a commercial integration?

Generally speaking, you have a commercial integration if you are incorporated and/or plan to monetize your app.

What does that mean?

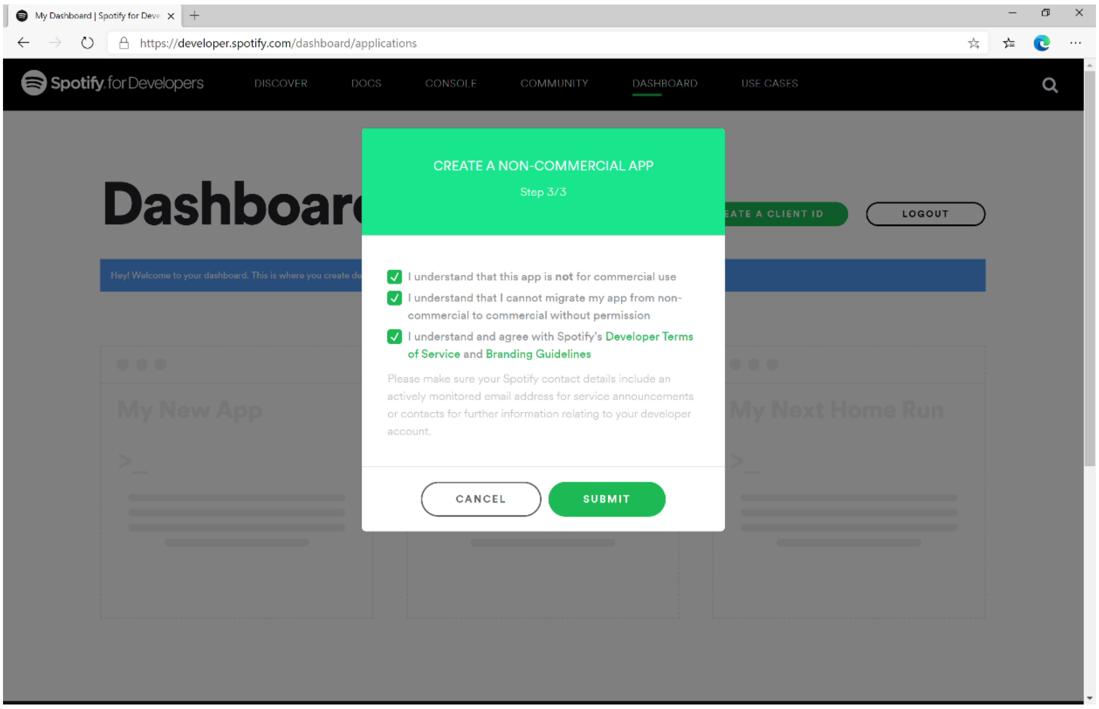
- **You monetize if...** your app or product charges a fee to purchase, promotes using ads or utilizes a freemium model for upsells.
- **You are incorporated if...** you are legally protected as a corporate entity or company.

Note: Please note that using Spotify's tools are always subject to our Developer Terms of Service and final approvals for commercial usage will always be left up to the discretion of Spotify and its partners.

NO YES

Step 8

Then in **Step 3/3** of **Create an App or Hardware Integration** then tick the option for **I understand that this app is not for commercial use**, tick the option for **I understand that I cannot migrate my app from non-commercial to commercial without permission** and follow and read the linked pages then tick the option for **I understand and agree with Spotify's Developer Terms of Service and Branding Guidelines,** then choose **Submit**


My Dashboard | Spotify for Deve
https://developer.spotify.com/dashboard/applications
Spotify for Developers
DISCOVER
DOCS
CONSOLE
COMMUNITY
DASHBOARD
USE CASES
Dashboard
CREATE A CLIENT ID
LOGOUT
Hey! Welcome to your dashboard. This is where you create
My New App
My Next Home Run

CREATE A NON-COMMERCIAL APP

Step 3/3

☑ I understand that this app is **not** for commercial use

☑ I understand that I cannot migrate my app from non-commercial to commercial without permission

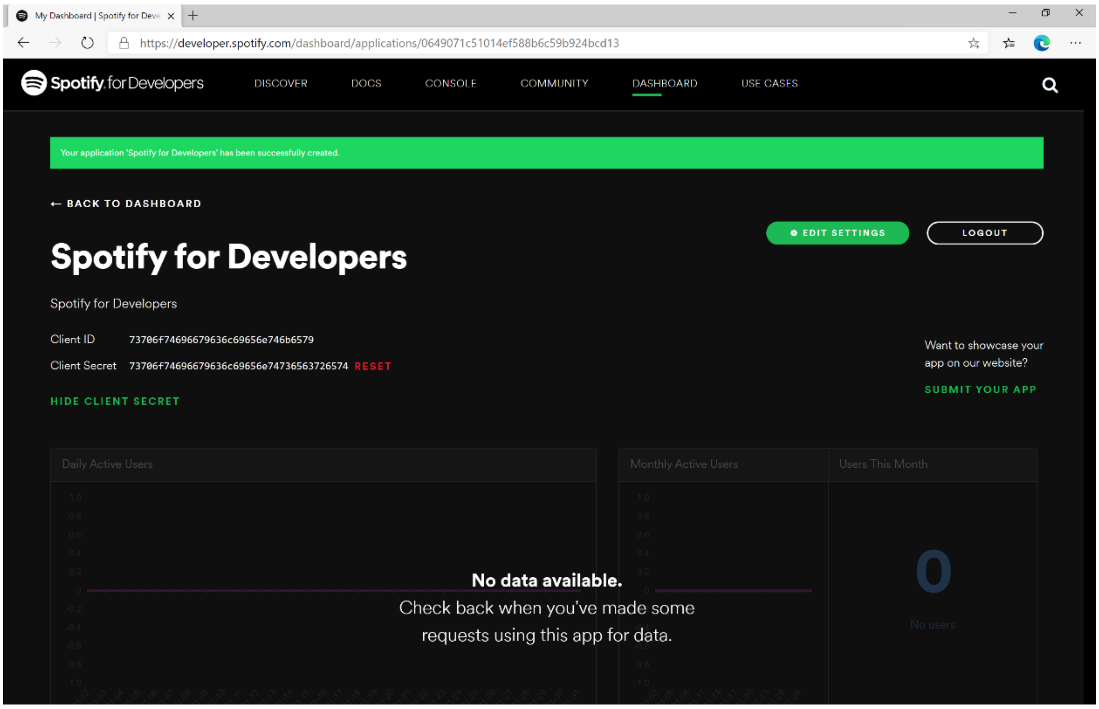
☑ I understand and agree with Spotify's Developer Terms of Service and Branding Guidelines

Please make sure your Spotify contact details include an actively monitored email address for service announcements or contacts for further information relating to your developer account.

CANCEL SUBMIT

Step 9

Once created, you will need to **Copy** the **Client ID** to the **Clipboard** e.g. 73706f74696679636c69656e746b6579 – note that your id will be different, then **Paste** the contents into a text editor such as **Notepad** where you copied the **Redirect URI** previously. Then select **Show Client Secret** and then **Copy** the **Client Secret** to the **Clipboard** e.g. 73706f74696679636c69656e74736563726574 – your secret will be different, then **Paste** this into your text editor.



My Dashboard | Spotify for Deve
https://developer.spotify.com/dashboard/applications/0649071c51014ef588b6c59b924bcd13
Spotify for Developers
DISCOVER
DOCS
CONSOLE
COMMUNITY
DASHBOARD
USE CASES

Your application 'Spotify for Developers' has been successfully created.

← BACK TO DASHBOARD

EDIT SETTINGS

LOGOUT

Spotify for Developers

Spotify for Developers

Client ID 73706f74696679636c69656e746b6579

Client Secret 73706f74696679636c69656e74736563726574 RESET

HIDE CLIENT SECRET

Want to showcase your app on our website?

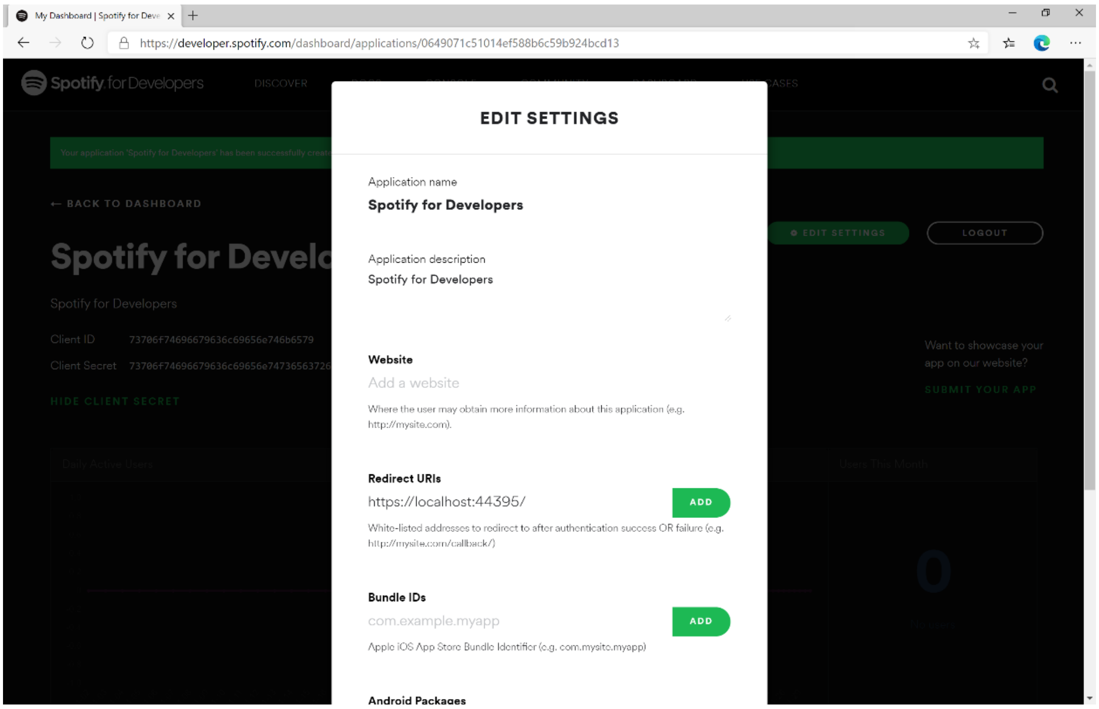
SUBMIT YOUR APP

No data available.

Check back when you've made some requests using this app for data.

Step 10

Finally, in your text editor such as **Notepad** you need to **Copy** the **Redirect URI** e.g. https://localhost:44395/ - please note that your **Redirect URI** may be different. Then in the **Dashboard** choose **Edit Settings** then **Paste** the contents of your **Clipboard** into the **Redirect URIs** and select **Add** then to complete the process select **Save**.



My Dashboard | Spotify for Deve
https://developer.spotify.com/dashboard/applications/0649071c51014ef588b6c59b924bcd13

EDIT SETTINGS

Application name

Spotify for Developers

Application description

Spotify for Developers

Website

Add a website

Where the user may obtain more information about this application (e.g. http://mysite.com).

Redirect URIs

https://localhost:44395/ ADD

White-listed addresses to redirect to after authentication success OR failure (e.g. http://mysite.com/callback/)

Bundle IDs

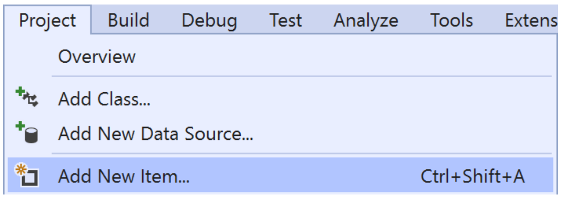
com.example.myapp ADD

Apple iOS App Store Bundle Identifier (e.g. com.mysite.myapp)

Android Packages

Token & Result

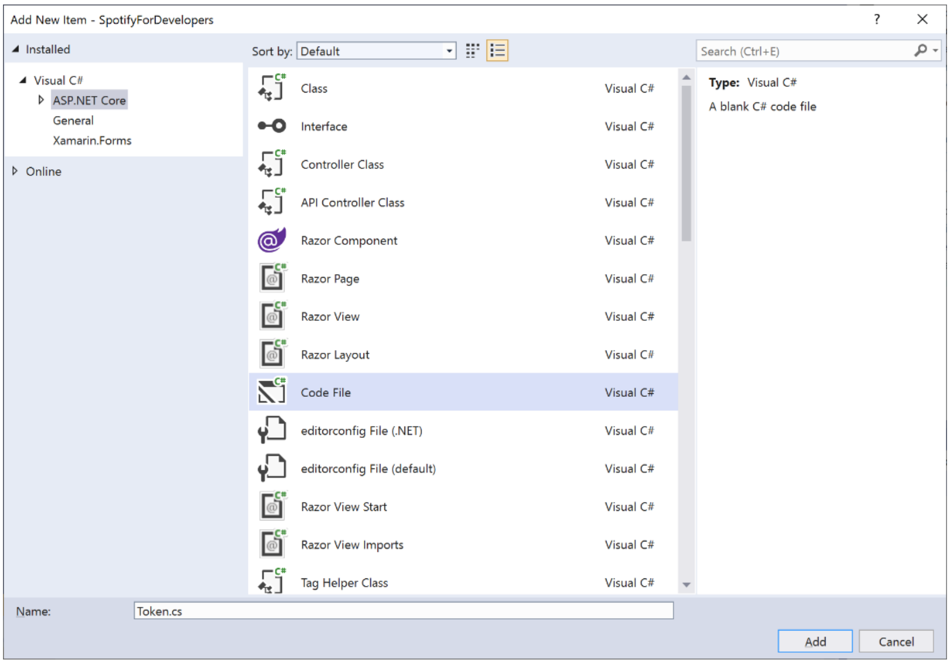
Step 1



First return to **Visual Studio 2019** and then from the **Menu** choose **Project** then **Add New Item...**

Step 2

Then, from the **Add New Item** window from **Installed** select **Visual C#** then **ASP .NET Core** and select **Code File** from the list, then type in the **Name** as **Token.cs** before selecting **Add** to add the file to the **Project**


Add New Item - SpotifyForDevelopers
?
✕
Installed
Visual C#
ASP.NET Core
General
Xamarin.Forms
Online
Sort by: Default
Search (Ctrl+E)
Class Visual C#
Interface Visual C#
Controller Class Visual C#
API Controller Class Visual C#
Razor Component Visual C#
Razor Page Visual C#
Razor View Visual C#
Razor Layout Visual C#
Code File Visual C#
editorconfig File (.NET) Visual C#
editorconfig File (default) Visual C#
Razor View Start Visual C#
Razor View Imports Visual C#
Tag Helper Class Visual C#
Type: Visual C#
A blank C# code file
Add
Cancel

Name: Token.cs

Step 3

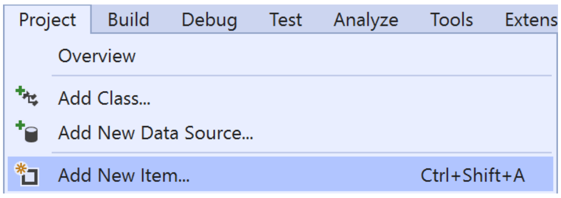
Once in the **Code View** for **Token.cs** the following should be entered:

```
using Spotify.NetStandard.Client.Authentication;
using Spotify.NetStandard.Client.Authentication.Enums;
using System.Text.Json;

public class Token
{
    public AccessToken AccessToken { get; set; }
    public bool HasUserToken => AccessToken?.TokenType == TokenType.User;
    public bool HasToken => AccessToken?.TokenType == TokenType.Access || HasUserToken;
    public Token(AccessToken token) => AccessToken = token;
    public Token(string content) =>
        AccessToken = JsonSerializer.Deserialize<AccessToken>(content);
    public override string ToString() => JsonSerializer.Serialize(AccessToken);
}
```

This will represent an `AccessToken` which is part of the **NuGet** package for **Spotify.NetStandard** and will allow the **AccessToken** to be stored and retrieved in **JSON** format using `System.Text.Json`.

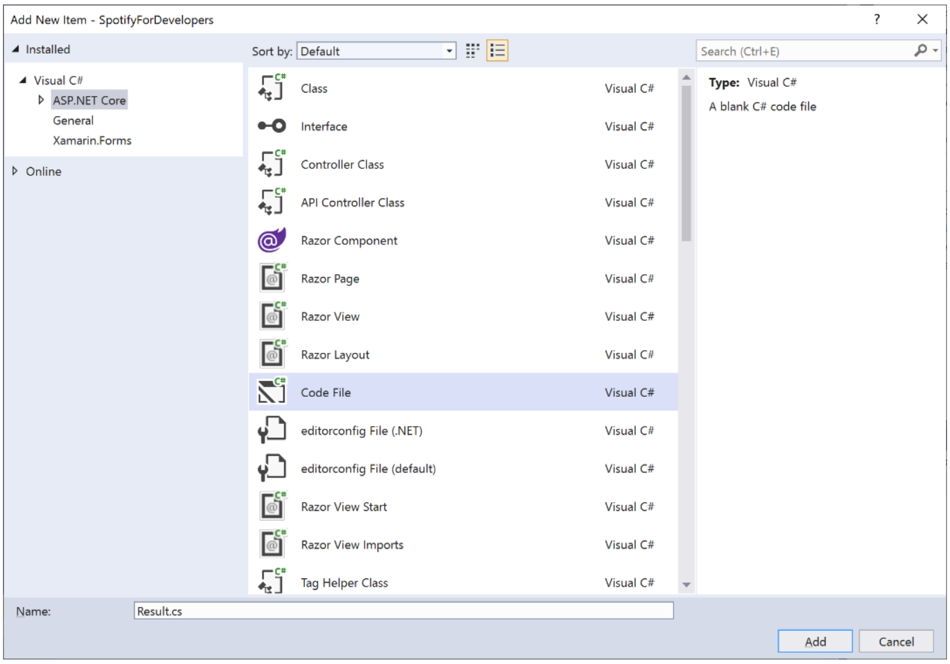
Step 4



Then again from the **Menu** choose **Project** then **Add New Item...**

Step 5

Then, from the **Add New Item** window from **Installed** select **Visual C#** then **ASP .NET Core** and select **Code File** from the list, then type in the **Name** as **Result.cs** before selecting **Add** to add the file to the **Project**


Add New Item - SpotifyForDevelopers
Installed
Visual C#
ASP.NET Core
General
Xamarin.Forms
Online
Sort by: Default
Search (Ctrl+E)
Class Visual C#
Interface Visual C#
Controller Class Visual C#
API Controller Class Visual C#
Razor Component Visual C#
Razor Page Visual C#
Razor View Visual C#
Razor Layout Visual C#
Code File Visual C#
editorconfig File (.NET) Visual C#
editorconfig File (default) Visual C#
Razor View Start Visual C#
Razor View Imports Visual C#
Tag Helper Class Visual C#
Type: Visual C#
A blank C# code file
Add
Cancel

Name: Result.cs

Step 6

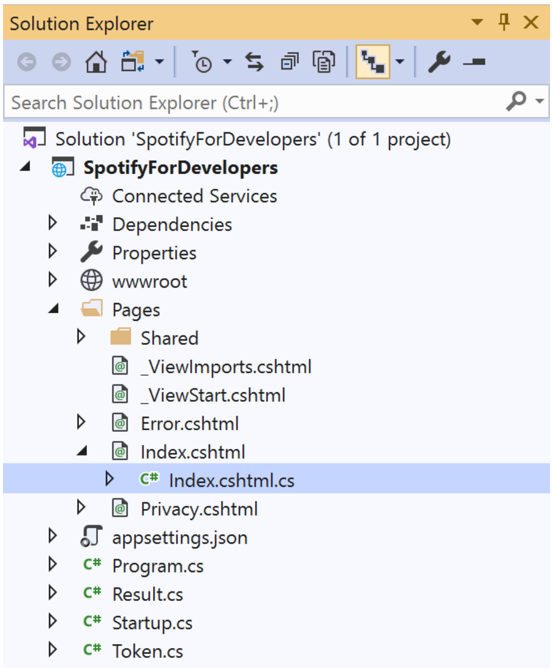
Once in the **Code View** for **Result.cs** the following should be entered:

```
public class Result
{
    public string Id { get; set; }
    public string Name { get; set; }
    public string Image { get; set; }
    public Result Inner { get; set; }

    public Result(
        string id = null,
        string name = null,
        string image = null,
        Result inner = null)
    {
        Id = id;
        Name = name;
        Image = image;
        Inner = inner;
    }
}
```

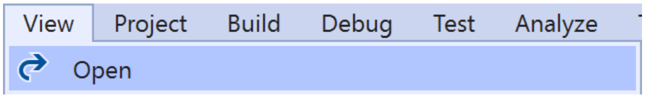
This will represent any response to be displayed from **NuGet** package for **Spotify.NetStandard**.

Step 7



In the **Solution Explorer** open the **Pages** section, then open the **Index.cshtml** section and select **Index.cshtml.cs**

Step 8



Then from the **Menu** choose **View** and then **Open**

Step 9

Once in the **Code View** for **Index.cshtml.cs** above the `namespace SpotifyForDevelopers.Pages` add the following `using` statements:

```
using Microsoft.AspNetCore.Http;
using Spotify.NetStandard.Client;
using Spotify.NetStandard.Client.Interfaces;
using Spotify.NetStandard.Requests;
using Spotify.NetStandard.Enums;
```

Then below `private readonly ILogger<IndexModel> _logger;` add the following `const` and `readonly` values:

```
private const string client = "clientid";
private const string secret = "clientsecret";
private const string state = "spotify.workshop";
private const string token = "token";
private const string country = "GB";

public static readonly ISpotifyApi Api = SpotifyClientFactory.CreateSpotifyClient(
    client, secret).Api;
```

You will need to **Copy** the **Client ID** to the **Clipboard** that you saved in your text editor e.g. **Notepad** and then **Paste** to replace `clientid` e.g. 73706f74696679636c69656e746b6579 - note that your id will be different. You will then need to **Copy** the **Client Secret** to the **Clipboard** from your text editor then **Paste** to replace `clientsecret` e.g. 73706f74696679636c69656e74736563726574 - note that your secret will be different. If done correctly those values should appear like the following:

```
private const string client = "73706f74696679636c69656e746b6579";
private const string secret = "73706f74696679636c69656e74736563726574";
```

You can also set the `country` to your e.g. `US` for United States or any supported Country such as `GB` for Great Britain.

Step 10

While still in the **Code View** for **Index.cshtml.cs** above `public IndexModel(ILogger<IndexModel> logger)` add the following **properties**:

```
public Token Token { get; set; }
public string Value { get; set; }
public string Option { get; set; }
public bool Flag { get; set; }
public IFormFile Upload { get; set; }
public IEnumerable<Result> Results { get; set; }
public Uri RedirectUri => new
Uri($"{HttpContext.Request.Scheme}://{HttpContext.Request.Host}");
public Uri CurrentUri => new
Uri($"{HttpContext.Request.Scheme}://{HttpContext.Request.Host}{HttpContext.Request.Path}{Htt
pContext.Request.QueryString}");
```

These **properties** will represent the **Token** and **Result** plus other values that will come in useful later.

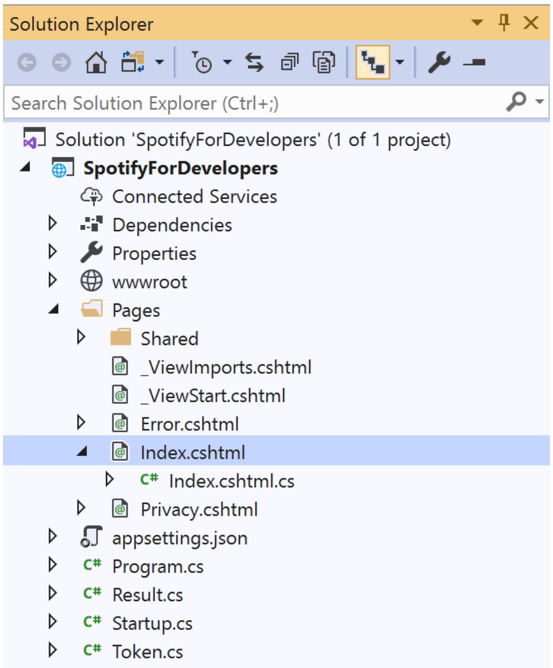
Then below the `public Uri CurrentUri` **property** add the following **methods**:

```
public void LoadToken()
{
    if (Request.Cookies[token] != null)
        Token = new Token(Request.Cookies[token]);
}

public void SaveToken()
{
    if (Token != null)
        Response.Cookies.Append(token, Token.ToString());
}
```

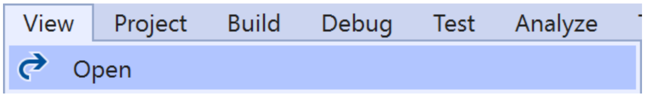
The `LoadToken` **method** will get a `Token` from a **Cookie** and the `SaveToken` **method** will add a `Token` to a given **Cookie**, a **Cookie** is a way to save information for a while in a Web Browser.

Step 11



In the **Solution Explorer** in the **Pages** section select **Index.cshtml**

Step 12



Then from the **Menu** choose **View** and then **Open**

Step 13

Once in the **Code View** for **Index.cshtml** remove all the existing content, which should be like the following:

```
@page
@model IndexModel
@{
    ViewData["Title"] = "Home page";
}

<div class="text-center">
    <h1 class="display-4">Welcome</h1>
    <p>Learn about <a href="https://docs.microsoft.com/aspnet/core">building Web apps with
ASP.NET Core</a>.</p>
</div>
```

Then, once removed you need to replace it with the following:

```
@page
@model IndexModel
@{
    ViewData["Title"] = "Spotify for Developers";
}
<h1>Spotify for Developers</h1>
<div class="container">
    <div class="row mb-2">
        <div class="mx-auto">

            <!—Authorisation Guide -->
        </div>
    </div>
    <div class="row mb-2">
        <div class="col-sm">
            @if (Model?.Token?.HasToken == true)
            {
                <h2 class="text-center">Spotify Web API App Authorisation</h2>

                <!-- Spotify Web API App Authorisation -->
            }
        </div>
        <div class="col-sm">
            @if (Model?.Token?.HasUserToken == true)
            {
                <h2 class="text-center">Spotify Web API User Authorisation</h2>

                <!-- Spotify Web API User Authorisation -->
            }
        </div>
    </div>

    <!-- Results -->
</div>
```

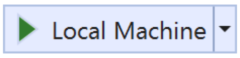
This represents the basic layout where you'll place the various elements in subsequent parts, and with the **<!-- -->** statements which will help you place things correctly as you'll always be adding anything new above those statements.

Step 14

Then, while still in the **Code View** for **Index.cshtml** above `<!-- Results -->` enter the following:

```
<div class="row mb-2">
    @if (Model?.Results?.Count() > 0)
    {
        var first = Model?.Results.First();
        <h2>Results</h2>
        <table class="table">
            <thead>
                <tr>
                    @if (first.Image != null)
                    {
                        <th>Image</th>
                    }
                    @if (first.Id != null)
                    {
                        <th>Id</th>
                    }
                    @if (first.Name != null)
                    {
                        <th>Name</th>
                    }
                    @if (first?.Inner?.Id != null)
                    {
                        <th>Id</th>
                    }
                    @if (first?.Inner?.Name != null)
                    {
                        <th>Name</th>
                    }
                </tr>
            </thead>
            <tbody>
                @foreach (var result in Model.Results)
                {
                    <tr>
                        @if (result.Image != null)
                        {
                            <td><img height="64" width="64" src="@result.Image" /></td>
                        }
                        @if (result.Id != null)
                        {
                            <td>@result.Id</td>
                        }
                        @if (result.Name != null)
                        {
                            <td>@result.Name</td>
                        }
                        @if (result?.Inner?.Id != null)
                        {
                            <td>@result.Inner.Id</td>
                        }
                        @if (result?.Inner?.Name != null)
                        {
                            <td>@result.Inner.Name</td>
                        }
                    </tr>
                }
            </tbody>
        </table>
    }
</div>
```

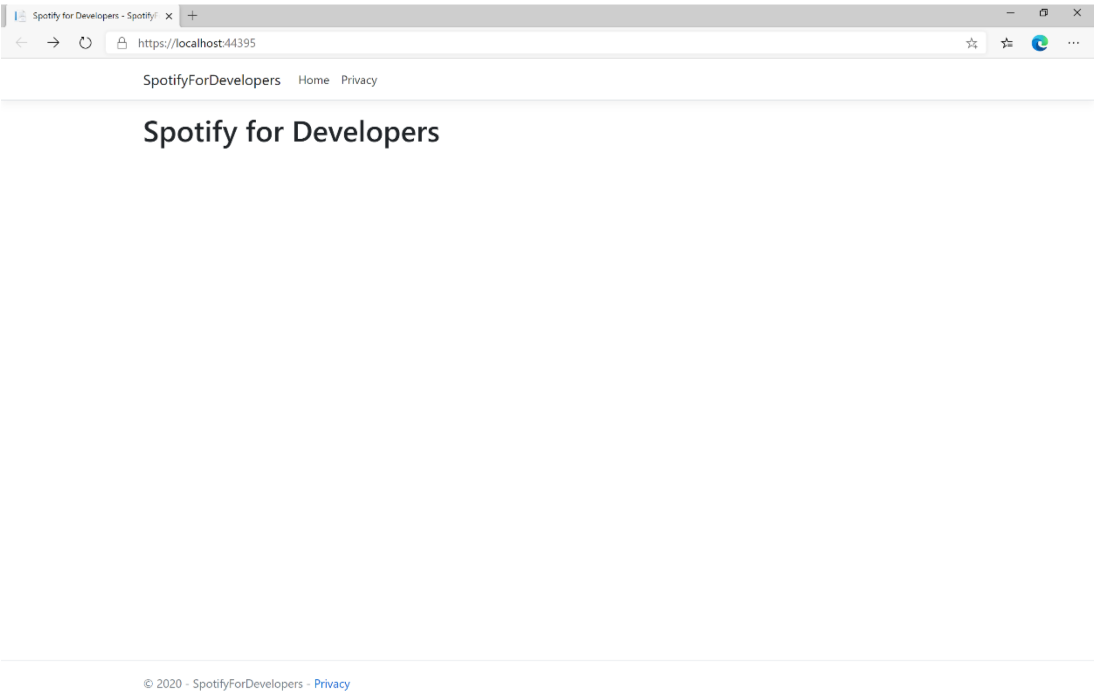
Step 15



Finally, in **Visual Studio 2019** select **IIS Application** to run the **Web Application**

Step 16

Once the **Web Application** is running it should appear something like the following:



Spotify for Developers - Spotify

https://localhost:44395

SpotifyForDevelopers Home Privacy

Spotify for Developers

© 2020 - SpotifyForDevelopers - Privacy

Step 17

You can stop the **web application** in **Visual Studio 2019** by selecting the **Stop debugging** button

Step 18

You can exit **Visual Studio 2019** by selecting the **Close** button in the top right of the **application** as that completes this part of the workshop

Authorisation Guide

Authorisation guide will show how to authorise your application to get data or to allow an end user to approve your application to access their Spotify data or features. Authorised requests to Spotify require permission to be granted to access data. In accordance with OAuth 2.0 the parties involved in the authorisation process are the End User, Application Client and the Spotify Server. Your application can be authorised in one of two ways. **Spotify API App Authorisation** where you authorise your app to access the Spotify Platform e.g. APIs, SDKs and Widgets and **Spotify API User Authorisation** where you grant your app permission to access or modify the user's own data.

Authorisation Scopes

Authorisation scopes allow your application to access specific API endpoints on behalf of a user. The set of scopes you can pass from your application include **Listening History**, **Library**, **Follow**, **Playlists**, **User & Images**, **Spotify Connect** and **Playback** for iOS, Android and Web Playback SDKs.

Listening History

user-top-read	
Endpoints	
Get a User's Top Artists / Get a User's Top Tracks	
Description	**Visible to users**
Read access to a user's top artists and tracks	Read your top artists and content.

user-read-playback-position	
Endpoints	
Get an Episodes, Get Multiple Episodes, Get a Show, Get Multiple Shows and Get a Show's Episodes	
Description	**Visible to users**
Read access to a user's playback position in a content.	Read your position in content you have played.

user-read-recently-played	
Endpoints	
Get Current User's Recently Played Tracks	
Description	**Visible to users**
Read access to a user's recently played tracks.	Access your recently played items.

Library

user-library-modify	
Endpoints	
Remove Albums for Current User, Remove User's Saved Tracks, Remove User's Saved Shows, Save Albums for Current User, Save Tracks for User and Save Shows for Current User	
Description	**Visible to users**
Write/delete access to a user's "Your Music" library.	Manage your saved content.

user-library-read	
Endpoints	
Check User's Saved Albums, Check User's Saved Tracks, Check User's Saved Shows, Get Current User's Saved Albums, Get a User's Saved Tracks and Get Users Saved Shows	
Description	**Visible to users**
Read access to a user's "Your Music" library.	Access your saved content.

Follow

user-follow-read	
Endpoints	
Get Following State for Artists/Users and Get User's Followed Artists	
Description	**Visible to users**
Read access to the list of artists and other users that the user follows.	Access your followers and who you are following.

user-follow-modify	
Endpoints	
Follow Artists or Users and Unfollow Artists or Users	
Description	**Visible to users**
Write/delete access to the list of artists and other users that the user follows.	Manage who you are following.

Playlists

playlist-read-collaborative	
Endpoints	
Get a List of Current User's Playlists and Get a List of a User's Playlists	
Description	**Visible to users**
Include collaborative playlists when requesting a user's playlists.	Access your collaborative playlists.

playlist-modify-public	
Endpoints	
Follow a Playlist, Unfollow a Playlist, Add Tracks to a Playlist, Change a Playlist's Details, Create a Playlist, Remove Tracks from a Playlist, Reorder a Playlist's Tracks, Replace a Playlist's Tracks and Upload a Custom Playlist Cover Image	
Description	**Visible to users**
Write access to a user's public playlists.	Manage your public playlists.

playlist-read-private	
Endpoints	
Check if Users Follow a Playlist, Get a List of Current User's Playlists and Get a List of a User's Playlists	
Description	**Visible to users**
Read access to user's private playlists.	Access your private playlists.

playlist-modify-private	
Endpoints	
Follow a Playlist, Unfollow a Playlist, Add Tracks to a Playlist, Change a Playlist's Details, Create a Playlist, Remove Tracks from a Playlist, Reorder a Playlist's Tracks, Replace a Playlist's Tracks and Upload a Custom Playlist Cover Image	
Description	**Visible to users**
Write access to a user's private playlists.	Manage your private playlists.

Users & Images

user-follow-read	
Endpoints	
Get Following State for Artists / Users and Get User's Followed Artists	
Description	**Visible to users**
Read access to the list of artists and other users that the user follows.	Access your followers and who you are following.

user-read-private	
Endpoints	
Search for an Item, Get Current User's Profile	
Description	**Visible to users**
Read access to user’s subscription details (type of user account).	Access your subscription details.

ugc-image-upload	
Endpoints	
Upload a Custom Playlist Cover Image	
Description	**Visible to users**
Write access to user-provided images.	Upload images to Spotify on your behalf.

Spotify Connect

user-read-playback-state	
Endpoints	
Get a User's Available Devices, Get Information About The User's Current Playback and Get the User's Currently Playing Track	
Description	**Visible to users**
Read access to a user’s player state.	Read your currently playing content and Spotify Connect devices information.

user-modify-playback-state	
Endpoints	
Pause a User's Playback, Seek To Position In Currently Playing Track, Set Repeat Mode On User’s Playback, Set Volume For User's Playback, Skip User’s Playback To Next Track, Skip User’s Playback To Previous Track, Start/Resume a User's Playback, Toggle Shuffle For User’s Playback, Transfer a User's Playback and Add an Item To User's Current Playback Queue	
Description	**Visible to users**
Write access to a user’s playback state	Control playback on your Spotify clients and Spotify Connect devices.

user-read-currently-playing	
Endpoints	
Get the User's Currently Playing Track	
Description	**Visible to users**
Read access to a user’s currently playing content.	Read your currently playing content.

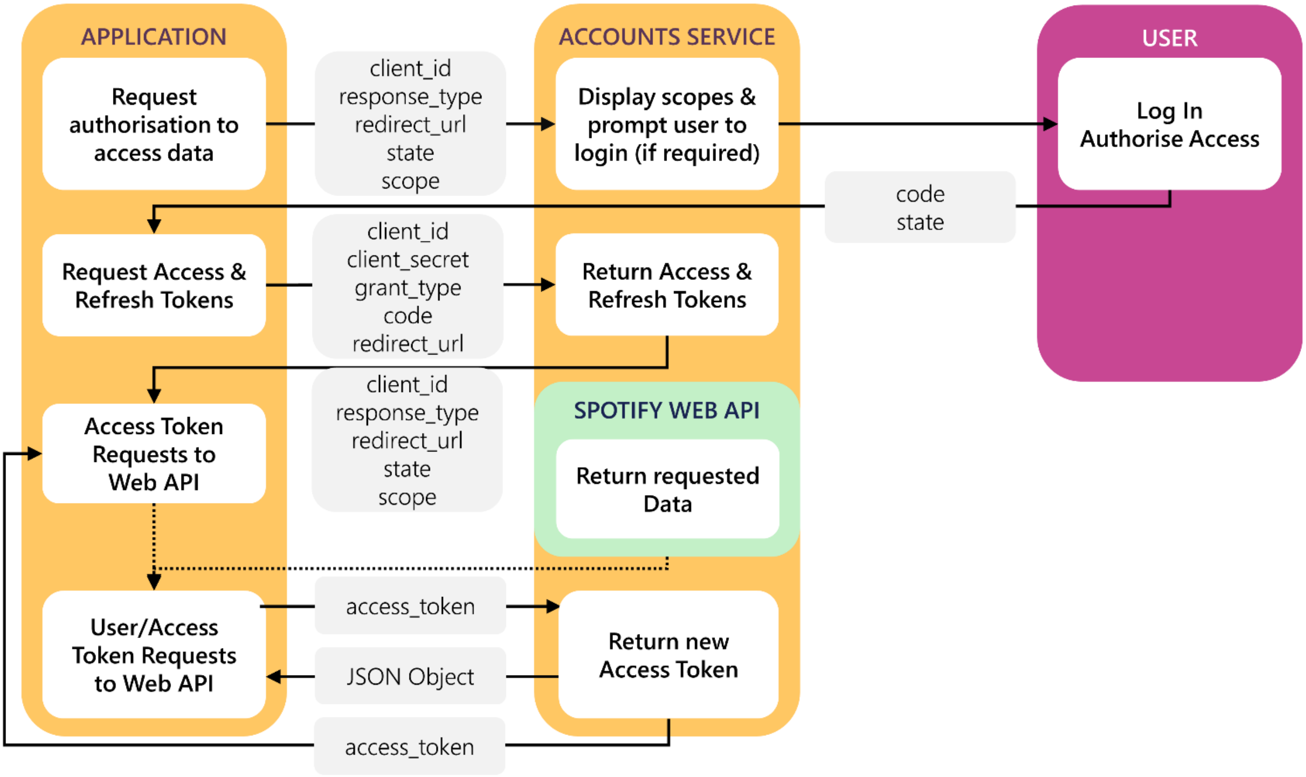
Playback

<table>
<tr><th colspan="2">streaming</th></tr>
<tr><th colspan="2">Endpoints</th></tr>
<tr><td colspan="2">Web Playback SDK</td></tr>
<tr><th>Description</th><th>Visible to users</th></tr>
<tr><td>Control playback of a Spotify track. This scope is currently available to the Web Playback SDK. The user must have a Spotify Premium account.</td><td>Play content and control playback on your other devices.</td></tr>
</table>

<table>
<tr><th colspan="2">app-remote-control</th></tr>
<tr><th colspan="2">Endpoints</th></tr>
<tr><td colspan="2">iOS SDK and Android SDK</td></tr>
<tr><th>Description</th><th>Visible to users</th></tr>
<tr><td>Remote control playback of Spotify. This scope is currently available to Spotify iOS and Android SDKs.</td><td>Communicate with the Spotify app on your device.</td></tr>
</table>

Authorisation Code

Authorisation Code flow is designed for long-running applications where the user grants permission once and provides a Token that can be refreshed where there is an initial request from application to the accounts service, then the user can authorise access to the application followed by any requests to the API to return requested data or new access tokens.



Access Code Request

User logs in and authorises access which includes required parameters including the **Client ID** obtained from the **Spotify Dashboard**, the **Redirect URI** which is the URI to redirect to and you can also provide a **State** value.

Query Parameter	
client_id	Client ID provided to you by Spotify when you registered your application
response_type	Set to "code"
redirect_uri	URI to redirect to after the user grants/denies permission. This URI needs to be entered in the URI whitelist that you specify when you registered your application
state	Strongly recommended as state can be useful for correlating requests and responses
scope	A space-separated list of scopes
show_dialog	Whether or not to force user to approve the app again if they've already done so

Example
https://accounts.spotify.com/authorize?client_id=5fe01282e44241328a84e7c5cc169165&response_type=code&redirect_uri=https%3A%2F%2Fexample.com%2Fcallback&scope=user-read-private%20user-read-email&state=STATE

Access Code Response

If the user accepts the request, you'll be redirected to your **Redirect URI** with the **Access Code** parameter. If the user denies the request or there's a problem, then you'll get an error returned – both will include the **State** value provided.

https://example.com/callback?code=NApCCg..BkWtQ&state=STATE	
User Accepts Request	
code	Authorisation code that can be exchanged for Token
state	Value of the state parameter supplied in the request

https://example.com/callback?error=access_denied&state=STATE	
User Denies Request or Error Occurred	
code	Authorisation code that can be exchanged for Token
state	Reason authorisation failed, for example: "access_denied"

Authorisation Code Request

Your application can request a **Refresh Token** and **Access Token** where you provide a base-64 encoded string containing the **Client ID** and **Client Secret** and **Code** from a previous **Access Code Response** and **Redirect URI** by performing a **POST** request with the following **Header** and **Body** parameters.

POST https://accounts.spotify.com/api/token	
Header	
Authorization	Base 64 encoded string containing Client ID and Client Secret **Authorization: Basic <base64 encoded client_id:client_secret>**
Body Parameter	
grant_type	Set to "authorization_code"
code	Authorisation code returned from Access Code Request
redirect_uri	URI to redirect to after the user grants/denies permission. This URI needs to be entered in the URI whitelist that you specify when you registered your application

Authorisation Code Response

If successful you will get an **Access Token** to use in subsequent calls to the **Spotify API** and a **Refresh Token**.

Result	
access_token	Access token that can be provided in subsequent calls e.g. Spotify Web API services
token_type	How the access token may be used - always "Bearer"
scope	Space-separated list of scopes which have been granted for this access_token
expires_in	Time period (in seconds) for which the access token is valid.
refresh_token	Token that can be sent to Spotify Accounts service in place of authorisation code

Refresh Token Request

You can request a refreshed **Access Token** by providing the **Refresh Token** returned from a previous **Authorisation Code Response** by performing a **POST** request with the following **Header** and **Body** parameters.

POST https://accounts.spotify.com/api/token	
Header	
Authorization	Base 64 encoded string containing Client ID and Client Secret **Authorization: Basic <base64 encoded client_id:client_secret>**
Body Parameter	
grant_type	Set to "refresh_token"
refresh_token	Refresh Token returned from the Authorisation Code exchange

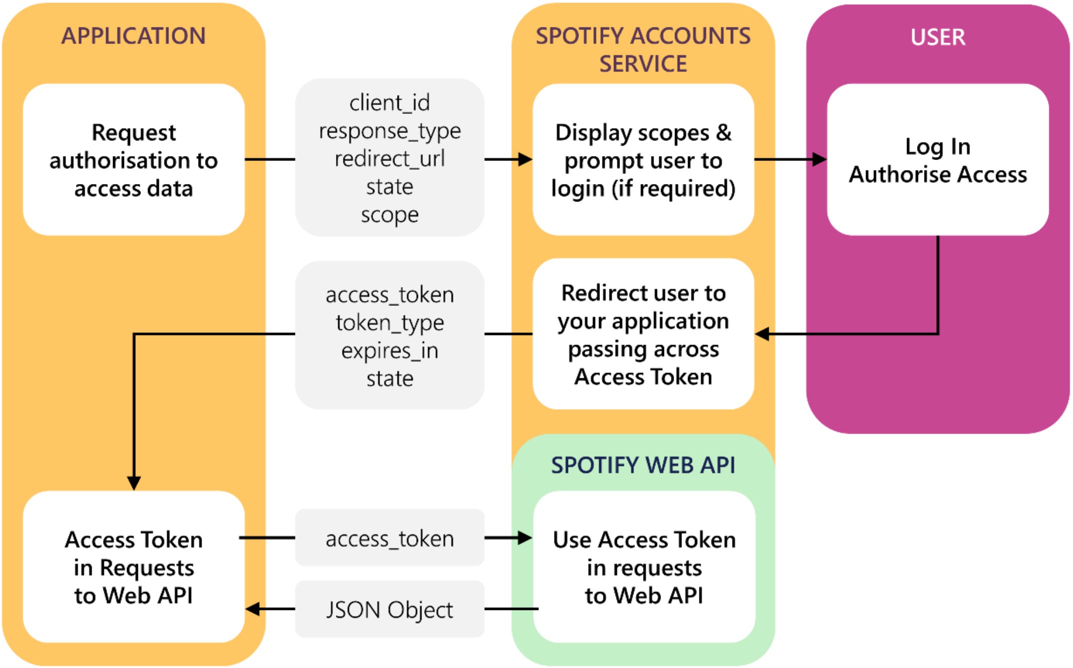
Refresh Token Response

If successful you will get an **Access Token** to use in subsequent calls to the **Spotify API** and can re-use a previously obtained **Refresh Token**.

Result	
access_token	Access token that can be provided in subsequent calls e.g. Spotify Web API services
token_type	How the access token may be used - always "Bearer"
scope	Space-separated list of scopes which have been granted for this access_token
expires_in	Time period (in seconds) for which the access token is valid.

Implicit Grant

Implicit Grant Flow is designed for clients that are implemented such as those using JavaScript and run in a web browser. There is the initial request from the application to the accounts service, then the user can log in and authorise access passing across a short-lived token followed by any requests to the API to return requested data using the token.


APPLICATION
Request authorisation to access data
client_id
response_type
redirect_url
state
scope
SPOTIFY ACCOUNTS SERVICE
Display scopes & prompt user to login (if required)
USER
Log In
Authorise Access
access_token
token_type
expires_in
state
Redirect user to your application passing across Access Token
SPOTIFY WEB API
Access Token in Requests to Web API
access_token
JSON Object
Use Access Token in requests to Web API

Implicit Grant Request

User logs in and authorises access which includes required parameters including the **Client ID** obtained from the **Spotify Dashboard**, the **Redirect URI** which is the URI to redirect to and you can also provide a **State** value.

Query Parameter	
client_id	Client ID provided to you by Spotify when you registered your application
response_type	Set to "token"
redirect_uri	URI to redirect to after the user grants/denies permission. This URI needs to be entered in the URI whitelist that you specify when you registered your application
state	Strongly recommended as state can be useful for correlating requests and responses
scope	A space-separated list of scopes
show_dialog	Whether or not to force user to approve the app again if they've already done so

Example
https://accounts.spotify.com/authorize?client_id=5fe01282e44241328a84e7c5cc169165&response_type=code&redirect_uri=https%3A%2F%2Fexample.com%2Fcallback&scope=user-read-private%20user-read-email&state=STATE

Implicit Grant Response

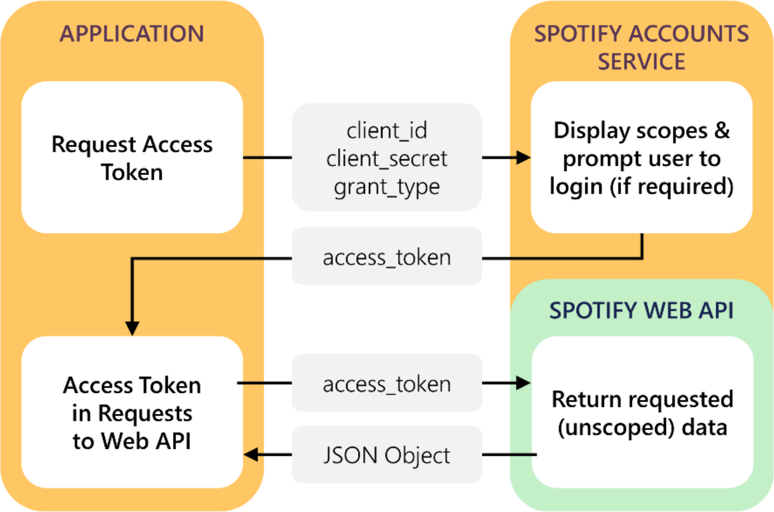
If the user grants access the URI will contain a hash fragment with data encoded as a query string including the **Access Token**, if the user doesn't grant access or there's a problem an error will be returned as a normal query string – both will also include the **State** value provided.

https://example.com/callback#access_token=Nw...2Tk&token_type=Bearer&expires_in=3600&state=STATE	
User Grants Access	
access_token	Token that can be provided in subsequent calls
token_type	Set to "Bearer"
expires_in	The time period (in seconds) for which the access token is valid
state	The value of the state parameter supplied in the request

https://example.com/callback?error=access_denied&state=STATE	
User Denies Access or Error Occurred	
error	Reason authorisation failed, for example: "access_denied"
state	Value of the state parameter supplied in the request

Client Credentials

Client Credentials Flow is designed for server-to-server authentication however only endpoints that do not access user information can be accessed, there is an initial request to obtain a token, you can optionally request the user log in to obtain a higher rate limit, followed by any requests to the API to return requested data using the token.



Client Credentials Request

You can request an **Access Token** by performing a **POST** request with the following **Header** and **Body** parameters.

POST https://accounts.spotify.com/api/token	
Header	
Authorization	Base 64 encoded string containing Client ID and Client Secret **Authorization: Basic <base64 encoded client_id:client_secret>**
Body Parameter	
grant_type	Set to “client_credentials”

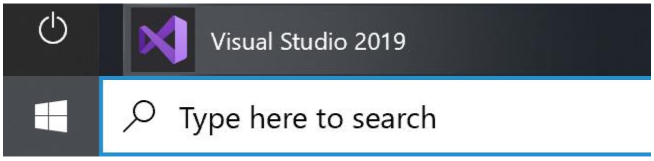
Client Credentials Response

If successful you will get an **Access Token** to use in subsequent calls to the **Spotify API**

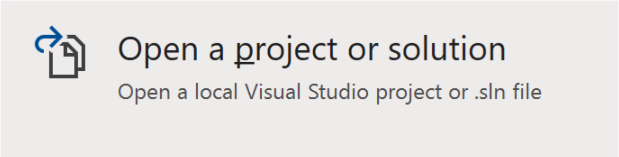
Result	
access_token	Access token that can be provided in subsequent calls e.g. Spotify Web API services
token_type	How the access token may be used - always “bearer”
expires_in	Time period (in seconds) for which the access token is valid.

Authorisation Flows

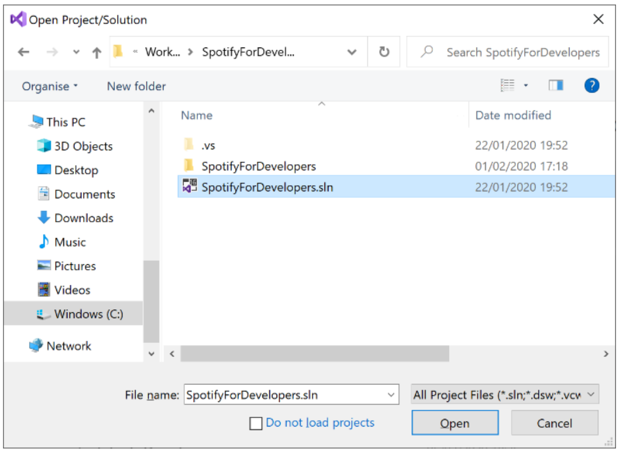
Step 1



In **Windows 10** choose **Start**, and then from the **Start Menu** find and select **Visual Studio 2019**


Open a project or solution
Open a local Visual Studio project or .sln file

Once done, from the **Get started** screen for **Visual Studio 2019** select **Open a project or solution**



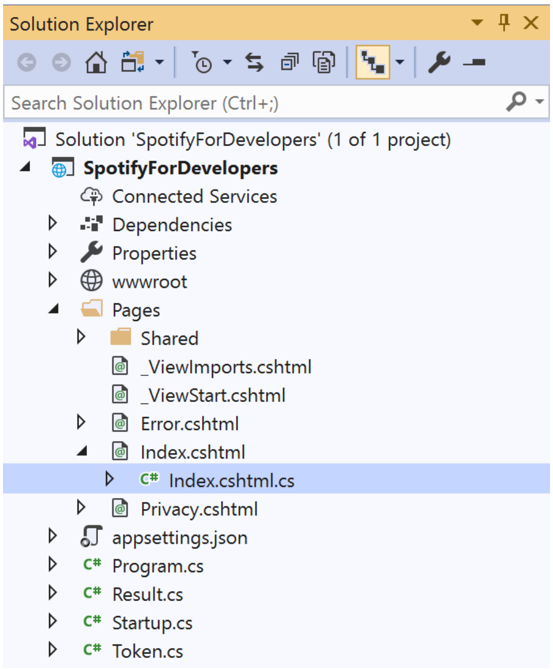
Name	Date modified
.vs	22/01/2020 19:52
SpotifyForDevelopers	01/02/2020 17:18
SpotifyForDevelopers.sln	22/01/2020 19:52

File name: SpotifyForDevelopers.sln

All Project Files (*.sln;*.dsw;*.vcw

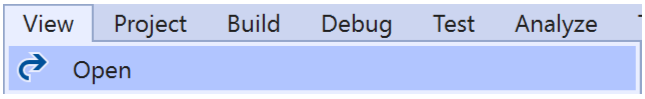
Then locate and select **SpotifyForDevelopers.sln** and select **Open** if you don't have this file already then please follow the previous parts of the workshop including **Getting Started**

Step 2



Once opened, in the **Solution Explorer** open the **Pages** section, then open the **Index.cshtml** section and select **Index.cshtml.cs**

Step 3



Then from the **Menu** choose **View** and then **Open**

Step 4

In the **Code View** for **Index.cshtml.cs** remove the existing `public void OnGet() {}` **method** which will appear like the following:

```
public void OnGet()
{

}
```

Then once removed, replace it with the following **method**:

```
public async Task<IActionResult> OnGetAsync(
    string code = null, string access_token = null)
{
    LoadToken();
    if (code != null)
        Token = new Token(await Api.GetAuthorisationCodeAuthTokenAsync(
            CurrentUri, RedirectUri, state));
    if (access_token != null)
        Token = new Token(Api.GetImplicitGrantAuthToken(
            CurrentUri, RedirectUri, state));
    SaveToken();
    return Page();
}
```

This **method** will be used with the responses from the **Authorisation Code Flow** and **Implicit Grant Flow** it also will read an existing **token** with the `LoadToken()` **method** and write a **token** with the `SaveToken()` **method**.

Step 5

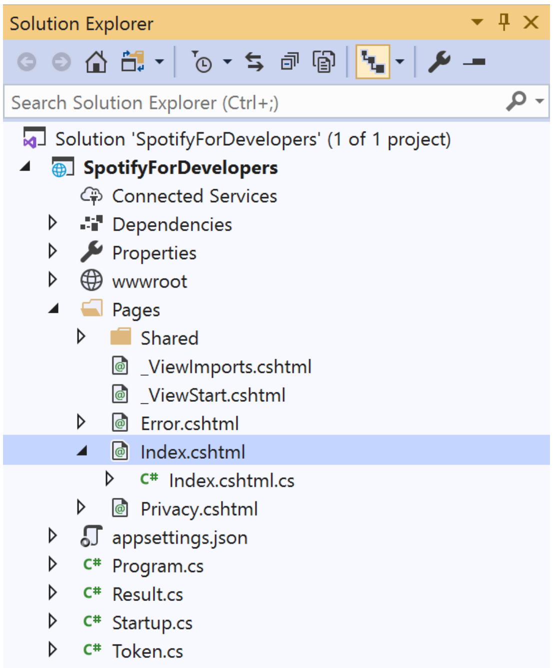
While still in the **Code View** for **Index.cshtml.cs** below the `public async Task<IActionResult> OnGetAsync(...) { ... }` **method** add the following **methods**:

```
public IActionResult OnPostAuthorisationCode() =>
    Redirect(Api.GetAuthorisationCodeAuthUri(
        RedirectUri, state, Scope.AllPermissions).ToString());

public IActionResult OnPostImplicitGrant() =>
    Redirect(Api.GetImplicitGrantAuthUri(
        RedirectUri, state, Scope.AllPermissions).ToString());
```

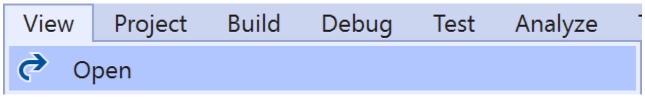
These **methods** will handle the requests for the **Authorisation Code Flow**, **Implicit Grant Flow** and the whole process for the **Client Credentials Flow**.

Step 6



In the **Solution Explorer** in the **Pages** section select **Index.cshtml**

Step 7



Then from the **Menu** choose **View** and then **Open**

Step 8

Once in the **Code View** for **Index.cshtml** above `<!-- Authorisation Guide -->` enter the following:

```
<h2 class="text-center">Authorisation</h2>
<ul class="list-group list-group-horizontal">
    <li class="list-group-item">
        <form asp-page-handler="authorisationcode" method="post">
            <button class="btn btn-success m-md-1">
                Authorisation Code Flow Login
            </button>
        </form>
    </li>
    <li class="list-group-item">
        <form asp-page-handler="implicitgrant" method="post">
            <button class="btn btn-success m-md-1">
                Implicit Grant Flow Login
            </button>
        </form>
    </li>
    <li class="list-group-item">
        <form asp-page-handler="clientcredentials" method="post">
            <button class="btn btn-success m-md-1">
                Client Credentials Flow Login
            </button>
        </form>
    </li>
</ul>
```

The `button` for each of these will trigger the various **Authorisation Flows** for **Authorisation Code**, **Implicit Grant** and **Client Credentials**.

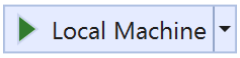
Step 9

While still in the **Code View** at the bottom of **Index.cshtml** enter the following `@section` and `script`:

```
@section Scripts
{
    <script language="javascript">
        $(window).ready(function () {
            var href = $(location).attr("href");
            if (href.indexOf('#') >= 0) {
                href = href.replace('#', '?');
                window.location.replace(href);
            }
        });
    </script>
}
```

The `script` will handle the response from the **Implicit Grant Response** which returns a **hash fragment** and turn this into a **query string** for use with the **web application**.

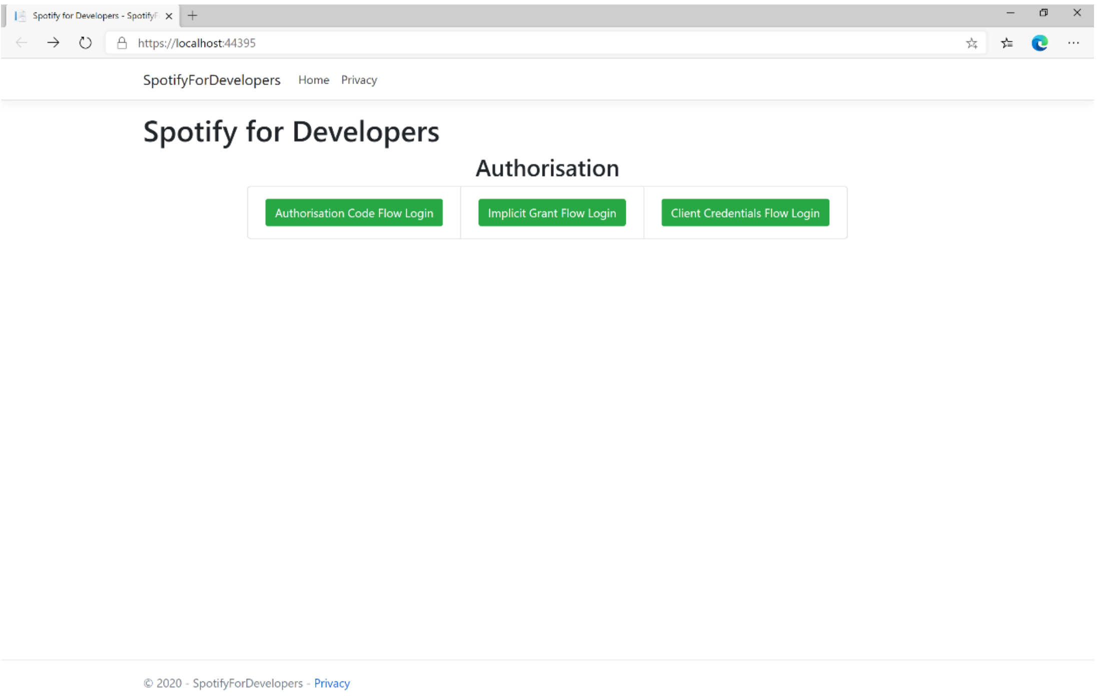
Step 10



Finally, in **Visual Studio 2019** select **IIS Application** to run the **Web Application**

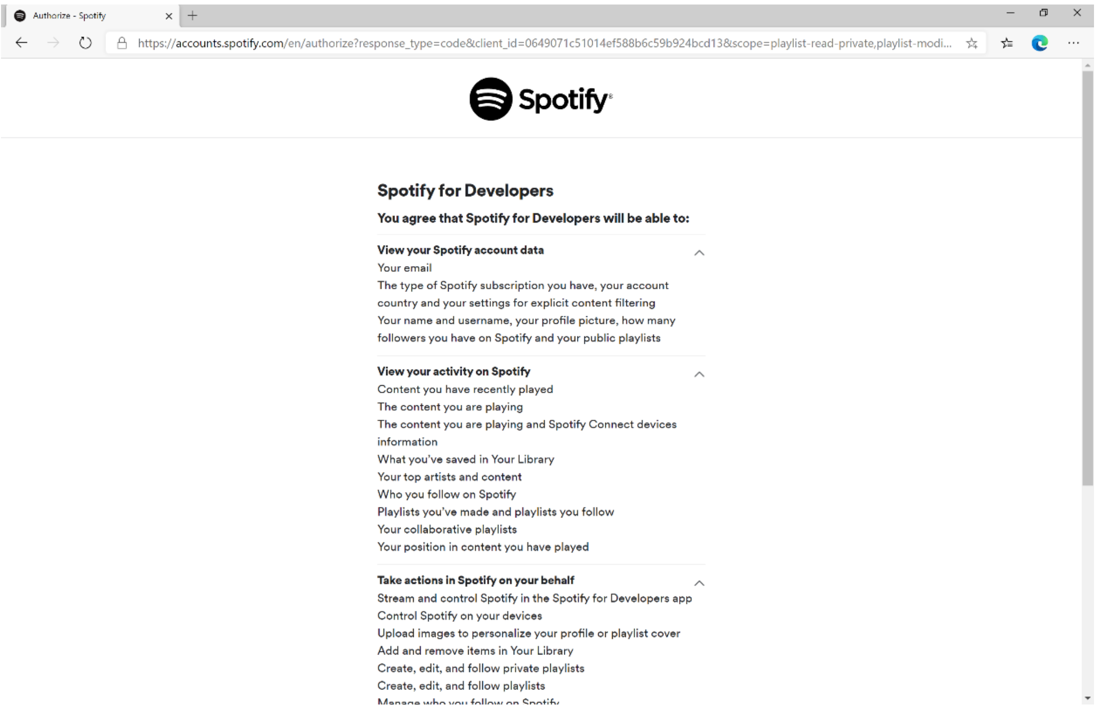
Step 11

Once the **Web Application** is running you should see something like the following:


Spotify for Developers - Spotify
https://localhost:44395
SpotifyForDevelopers
Home
Privacy
Spotify for Developers
Authorisation
Authorisation Code Flow Login
Implicit Grant Flow Login
Client Credentials Flow Login
© 2020 - SpotifyForDevelopers - Privacy

Step 12

Then in the **Web Application** you can select the **Authorisation Code Flow Login** or the **Implicit Grant Flow Login** button then choose **Agree**:



Authorize - Spotify

https://accounts.spotify.com/en/authorize?response_type=code&client_id=0649071c51014ef588b6c59b924bcd13&scope=playlist-read-private,playlist-modi...

Spotify

Spotify for Developers

You agree that Spotify for Developers will be able to:

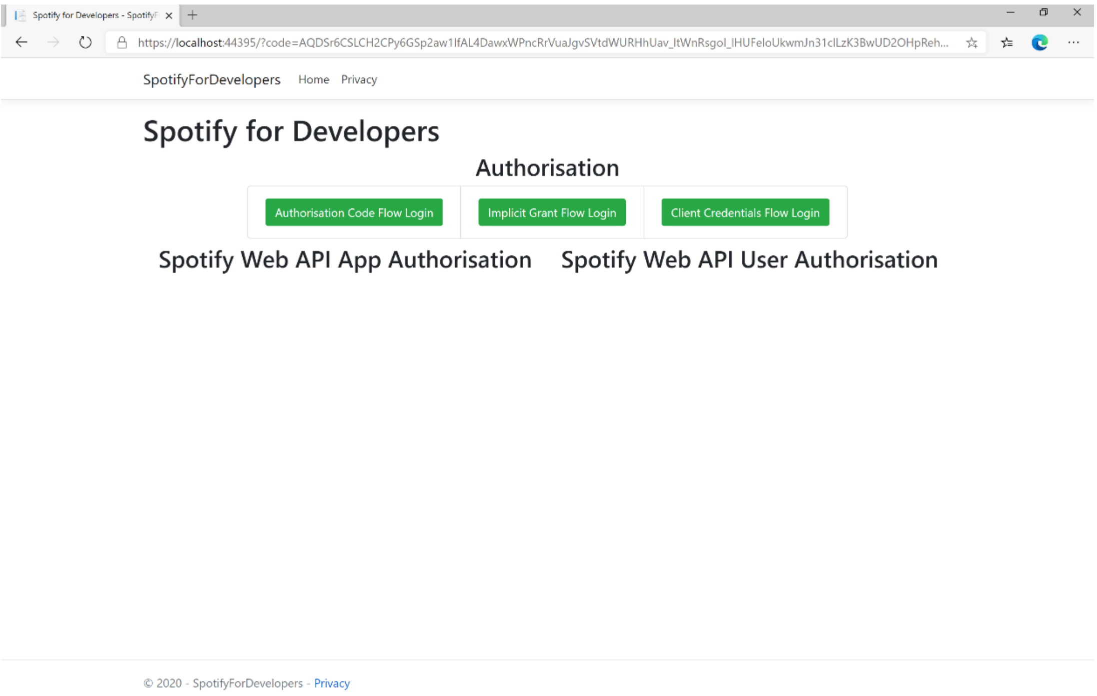
View your Spotify account data
Your email
The type of Spotify subscription you have, your account country and your settings for explicit content filtering
Your name and username, your profile picture, how many followers you have on Spotify and your public playlists

View your activity on Spotify
Content you have recently played
The content you are playing
The content you are playing and Spotify Connect devices information
What you've saved in Your Library
Your top artists and content
Who you follow on Spotify
Playlists you've made and playlists you follow
Your collaborative playlists
Your position in content you have played

Take actions in Spotify on your behalf
Stream and control Spotify in the Spotify for Developers app
Control Spotify on your devices
Upload images to personalize your profile or playlist cover
Add and remove items in Your Library
Create, edit, and follow private playlists
Create, edit, and follow playlists
Manage who you follow on Spotify

Step 13

Once agreed the **web application** if **Authorisation Code Flow Login** or **Implicit Grant Flow Login** was selected should appear like the following:



Spotify for Developers - SpotifyF

https://localhost:44395/?code=AQDSr6CSLCH2CPy6GSp2aw1IfAL4DawxWPncRrVuaJgvSVtdWURHhUav_ItWnRsgol_IHUFeloUkwmJn31clLzK3BwUD2OHpReh...

SpotifyForDevelopers Home Privacy

Spotify for Developers

Authorisation

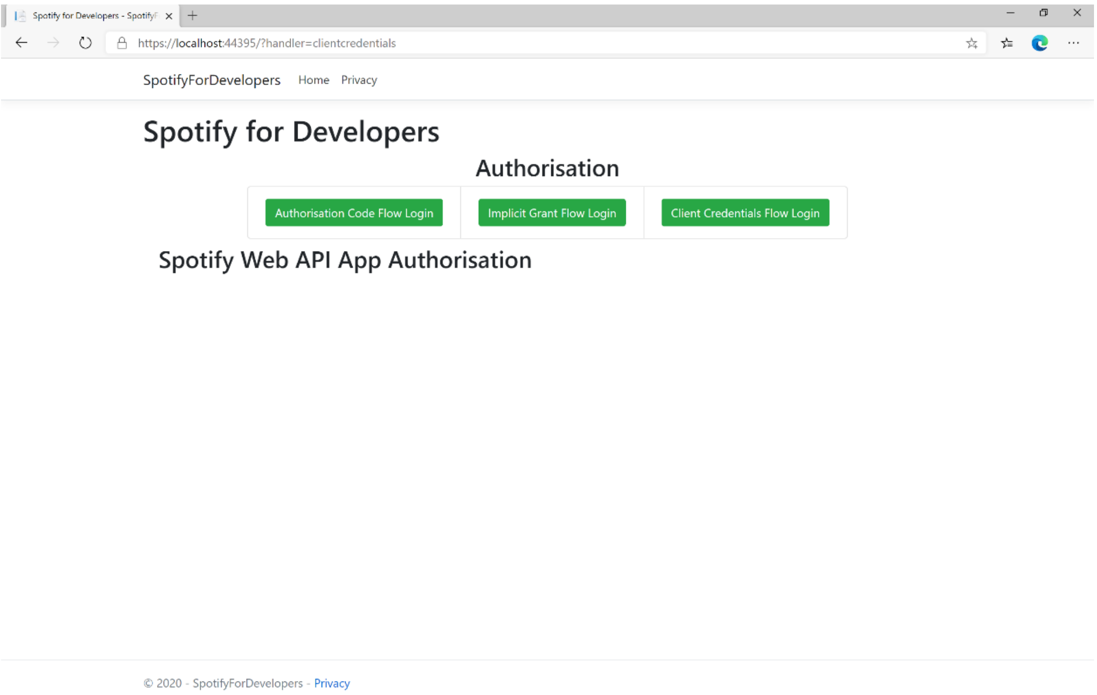
Authorisation Code Flow Login
Implicit Grant Flow Login
Client Credentials Flow Login

Spotify Web API App Authorisation   Spotify Web API User Authorisation

© 2020 - SpotifyForDevelopers - Privacy

Step 14

If the **Client Credentials Flow Login** was selected the **web application** should appear like the following:



Spotify for Developers - Spotify
https://localhost:44395/?handler=clientcredentials
SpotifyForDevelopers
Home
Privacy
Authorisation Code Flow Login
Implicit Grant Flow Login
Client Credentials Flow Login

Spotify for Developers

Authorisation

Spotify Web API App Authorisation

© 2020 - SpotifyForDevelopers - Privacy

Step 15

■

You can stop the **web application** in **Visual Studio 2019** by selecting the **Stop debugging** button

Step 16

×

You can exit **Visual Studio 2019** by selecting the **Close** button in the top right of the **application** as that completes this part of the workshop

Search & Browse

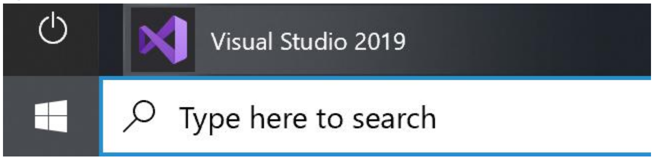
Search for an Item

Get Spotify Catalogue information about albums, artists, playlists, tracks, shows or episodes that match a query string.

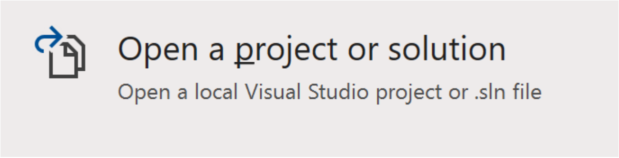
GET https://api.spotify.com/v1/search?q=tania%20bowra&type=artist	
Header	
Authorization	Valid Access Token from Spotify Accounts service
Query Parameter	
q	Query keywords, optional field filters and operators
type	A comma-separated list of item types to search across. Valid types are: album, artist, playlist, track, show and episode
market	ISO 3166-1 alpha-2 country code
limit	Maximum number of results to return
offset	Index of first result to return
include_external	"audio" includes any relevant audio content that is hosted externally

Header	**Response**
Success	
HTTP Status 200 OK	Array of Artist Objects / Simplified Album Objects / Track Objects / Simplified Show Object / Simplified Episode Object wrapped in a Paging Object
Error	
Error Code	Error Object

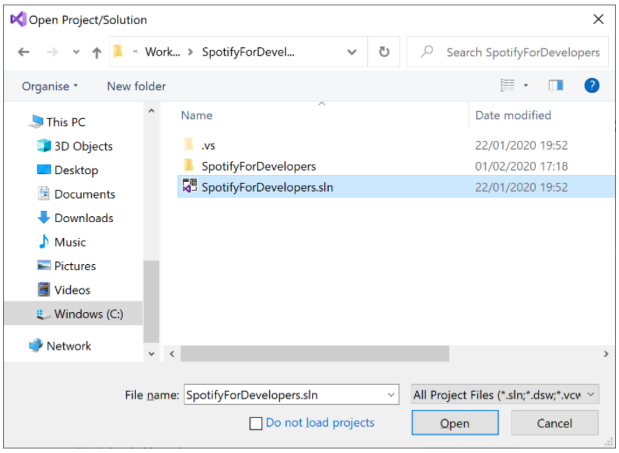
Step 1



In **Windows 10** choose **Start**, and then from the **Start Menu** find and select **Visual Studio 2019**


Open a project or solution
Open a local Visual Studio project or .sln file

Once done, from the **Get started** screen for **Visual Studio 2019** select **Open a project or solution**

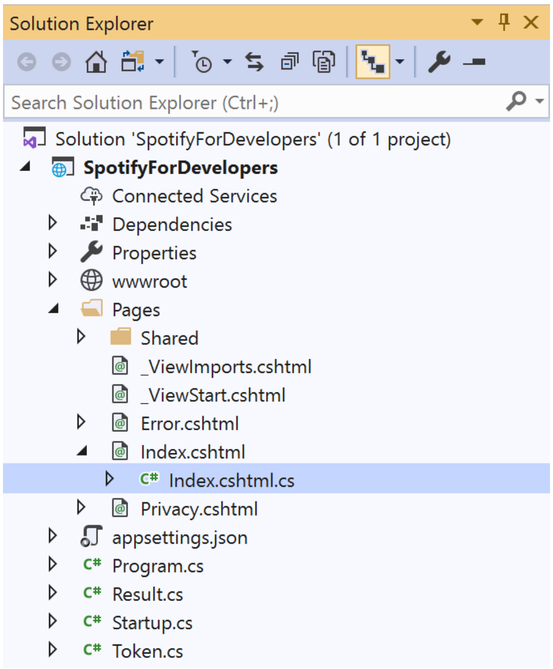


Name	Date modified
.vs	22/01/2020 19:52
SpotifyForDevelopers	01/02/2020 17:18
SpotifyForDevelopers.sln	22/01/2020 19:52

File name: SpotifyForDevelopers.sln

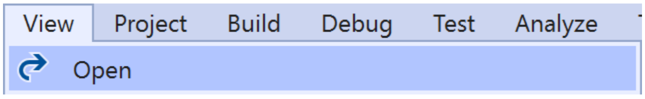
Then locate and select **SpotifyForDevelopers.sln** and select **Open** if you don't have this file already then please follow the previous parts of the workshop including **Getting Started** and **Authorisation Guide**

Step 2



Once opened, in the **Solution Explorer** open the **Pages** section, then open the **Index.cshtml** section and select **Index.cshtml.cs**

Step 3



Then from the **Menu** choose **View** and then **Open**

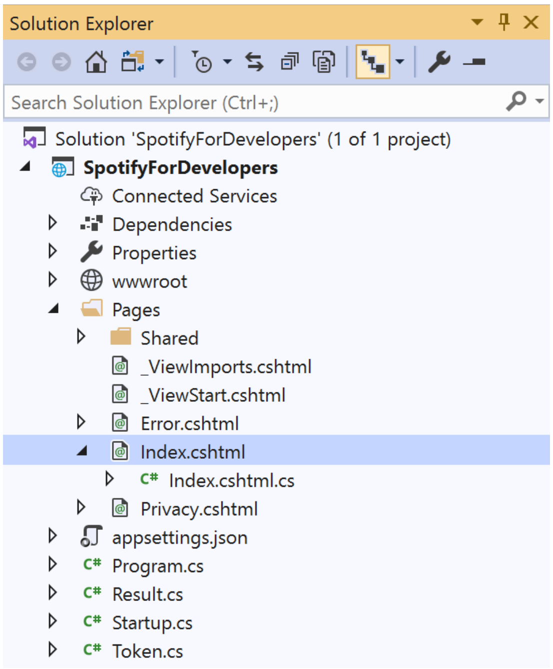
Step 4

In the **Code View** for **Index.cshtml.cs** below the **method** for `public async Task<IActionResult> OnPostClientCredentialsAsync() { ... }` enter the following **method**:

```
public async Task<IActionResult> OnPostSearchForAnItemAsync(string value, string option)
{
    LoadToken();
    var results = new List<Result>();
    var search = await Api.SearchForItemAsync(value,
    new SearchType()
    {
        Album = option.Contains("Album"),
        Track = option.Contains("Track"),
        Artist = option.Contains("Artist"),
        Playlist = option.Contains("Playlist"),
        Show = option.Contains("Show"),
        Episode = option.Contains("Episode")
    }, country);
    if (search?.Albums?.Items != null)
    {
        results.AddRange(search.Albums.Items.Select(result => new Result(
            result.Id, result.Name, result?.Images?.FirstOrDefault()?.Url,
            new Result(result?.Artists[0]?.Id, result?.Artists[0]?.Name))));
    }
    if (search?.Tracks?.Items != null)
    {
        results.AddRange(search.Tracks.Items.Select(result => new Result(
            result.Id, result.Name, result.Album?.Images?.FirstOrDefault()?.Url,
            new Result(result?.Artists[0]?.Id, result?.Artists[0]?.Name))));
    }
    if (search?.Artists?.Items != null)
    {
        results.AddRange(search.Artists.Items.Select(result => new Result(
            result.Id, result.Name, result?.Images?.FirstOrDefault()?.Url)));
    }
    if (search?.Playlists?.Items != null)
    {
        results.AddRange(search.Playlists.Items.Select(result => new Result(
            result.Id, result.Name, result?.Images?.FirstOrDefault()?.Url)));
    }
    if (search?.Shows?.Items != null)
    {
        results.AddRange(search.Shows.Items.Select(result => new Result(
            result.Id, result.Name, result?.Images?.FirstOrDefault()?.Url)));
    }
    if (search?.Episodes?.Items != null)
    {
        results.AddRange(search.Episodes.Items.Select(result => new Result(
            result.Id, result.Name, result?.Images?.FirstOrDefault()?.Url)));
    }
    if (results.Any())
        Results = results;
    return Page();
}
```

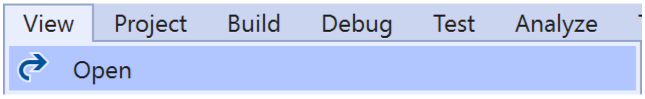
The **method** for `OnPostSearchForAnItemAsync` is used to find any **track**, **artist**, **album** or **playlist** and populates **property** for `Results` accordingly.

Step 5



In the **Solution Explorer** in the **Pages** section select **Index.cshtml**

Step 6



Then from the **Menu** choose **View** and then **Open**

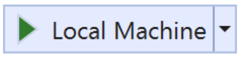
Step 7

Once in the **Code View** for **Index.cshtml** above `<!-- Spotify Web API App Authorisation -->` enter the following:

```
<ul class="list-group mb-2">
    <li class="list-group-item list-group-item-primary">
        <h5 class="list-group-item-heading">Search</h5>
    </li>
    <li class="list-group-item">
        <form asp-page-handler="SearchForAnItem" method="post">
            <select asp-for="Option" class="form-control mb-2">
                <option>Album</option>
                <option>Track</option>
                <option>Artist</option>
                <option>Playlist</option>
                <option>Show</option>
                <option>Episode</option>
            </select>
            <input asp-for="Value" placeholder="Query" class="form-control mb-2" />
            <button class="btn btn-primary mb-2">
                Search For An Item
            </button>
        </form>
    </li>
</ul>
```

This `form` will **post** to the **method** for `SearchForAnItemAsync` and has a `select` list to choose an `option` to search for and an `input` for the `Query`.

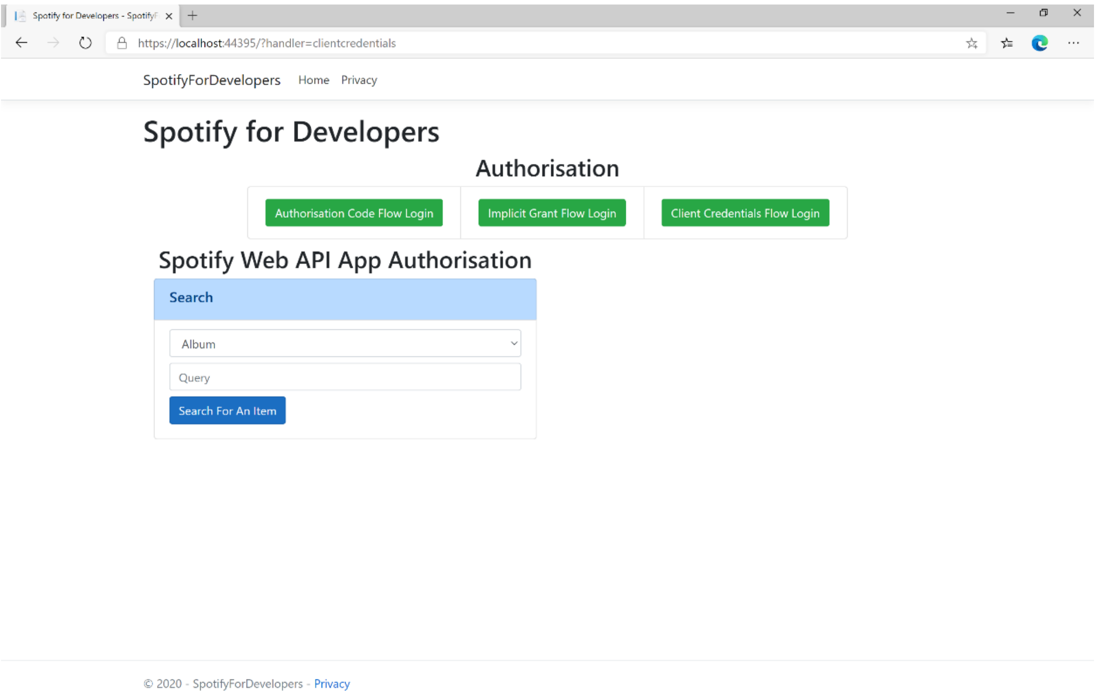
Step 8



Finally, in **Visual Studio 2019** select **IIS Application** to run the **Web Application**

Step 9

Once the **Web Application** is running and you select **Client Credentials Flow Login** you should see something like the following:



https://localhost:44395/?handler=clientcredentials

SpotifyForDevelopers Home Privacy

Spotify for Developers

Authorisation

Authorisation Code Flow Login | Implicit Grant Flow Login | Client Credentials Flow Login

Spotify Web API App Authorisation

Search

Album

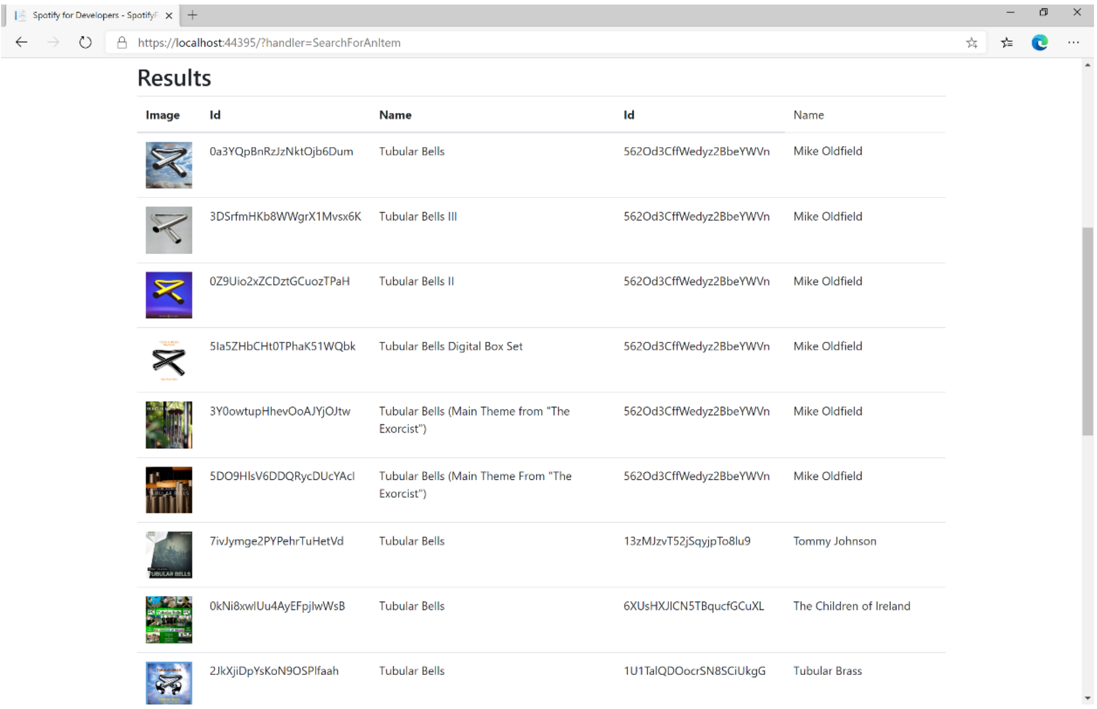
Query

Search For An Item

© 2020 - SpotifyForDevelopers - Privacy

Step 10

You can then search for something, such as an **Album** and select **Search For An Item** then scroll down to view **Results** like the following:



https://localhost:44395/?handler=SearchForAnItem

Results

Image	Id	Name	Id	Name
	0a3YQpBnRzJzNktOjb6Dum	Tubular Bells	562Od3CffWedyz2BbeYWVn	Mike Oldfield
	3DSrfmHKb8WWgrX1Mvsx6K	Tubular Bells III	562Od3CffWedyz2BbeYWVn	Mike Oldfield
	0Z9Uio2xZCDztGCuozTPaH	Tubular Bells II	562Od3CffWedyz2BbeYWVn	Mike Oldfield
	5Ia5ZHbCHt0TPhaK51WQbk	Tubular Bells Digital Box Set	562Od3CffWedyz2BbeYWVn	Mike Oldfield
	3Y0owtupHhevOoAJYjOJtw	Tubular Bells (Main Theme from "The Exorcist")	562Od3CffWedyz2BbeYWVn	Mike Oldfield
	5DO9HIsV6DDQRycDUcYAcl	Tubular Bells (Main Theme From "The Exorcist")	562Od3CffWedyz2BbeYWVn	Mike Oldfield
	7ivJymge2PYPehrTuHetVd	Tubular Bells	13zMJzvT52jSqyjpTo8lu9	Tommy Johnson
	0kNi8xwlUu4AyEFpjlwWsB	Tubular Bells	6XUsHXJICN5TBqucfGCuXL	The Children of Ireland
	2JkXjiDpYsKoN9OSPlfaah	Tubular Bells	1U1TalQDOocrSN8SCiUkgG	Tubular Brass

Step 11

You can stop the **web application** in **Visual Studio 2019** by selecting the **Stop debugging** button

Step 12

You can choose to exit **Visual Studio 2019** by selecting the **Close** button in the top right of the **application** as that completes this part of the workshop

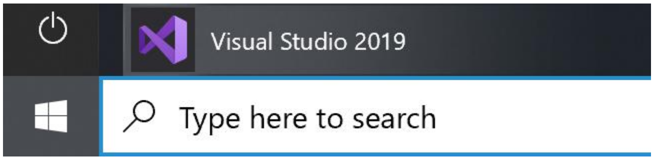
Get All Categories

Get a list of categories used to tag items in Spotify, as shown in the Spotify player's "Browse" tab.

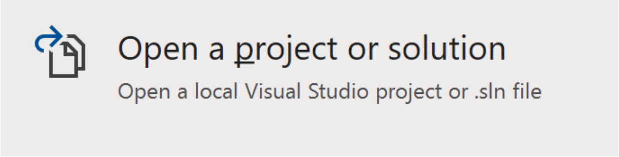
GET https://api.spotify.com/v1/browse/categories	
Header	
Authorization	Valid Access Token from Spotify Accounts service
Query Parameter	
country	ISO 3166-1 alpha-2 country code e.g. "GB". If omitted globally relevant items returned
locale	Desired language, consisting of ISO 639-1 language code and ISO 3166-1 alpha-2 country code, joined by an underscore e.g. "en_GB". If omitted results returned in American English
limit	Maximum number of results to return
offset	Index of first result to return

Header	**Response**
Success	
HTTP Status 200 OK	Categories field with an array of Category Objects wrapped in a Paging Object
Error	
Error Code	Error Object

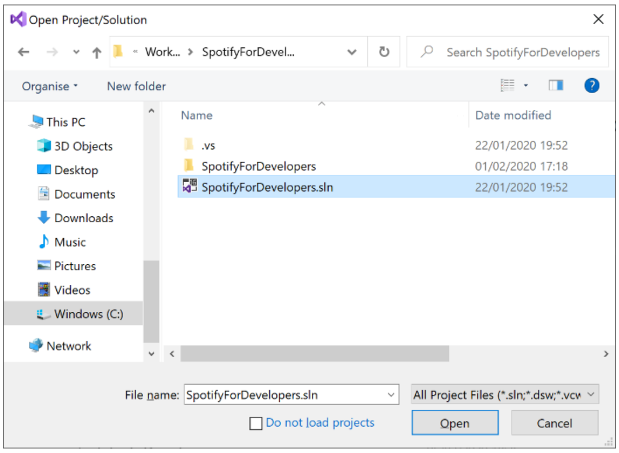
Step 1



If you chose to close **Visual Studio 2019** previously, in **Windows 10** choose **Start**, and then from the **Start Menu** find and select **Visual Studio 2019**


Open a project or solution
Open a local Visual Studio project or .sln file

Once done, from the **Get started** screen for **Visual Studio 2019** select **Open a project or solution**

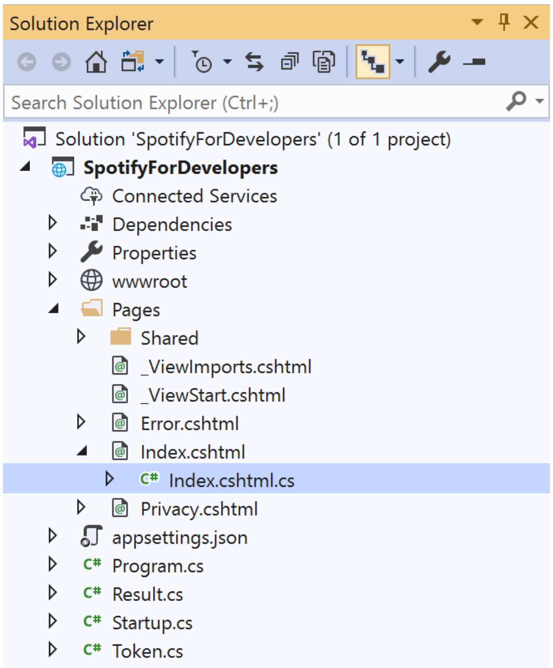


Name	Date modified
.vs	22/01/2020 19:52
SpotifyForDevelopers	01/02/2020 17:18
SpotifyForDevelopers.sln	22/01/2020 19:52

File name: SpotifyForDevelopers.sln

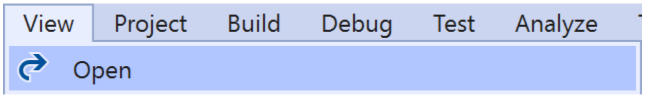
Then locate and select **SpotifyForDevelopers.sln** and select **Open** if you don't have this file already then please follow the previous parts of the workshop including **Getting Started**, **Authorisation Guide** and **Search & Browse**

Step 2



Once opened, in the **Solution Explorer** open the **Pages** section, then open the **Index.cshtml** section and select **Index.cshtml.cs**

Step 3



Then from the **Menu** choose **View** and then **Open**

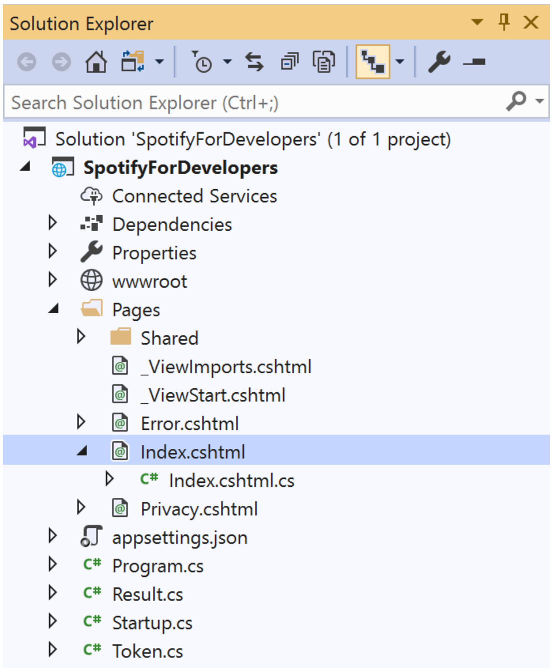
Step 4

In the **Code View** for **Index.cshtml.cs** below the **method** for `public async Task<IActionResult> OnPostSearchForAnItemAsync(...) { ... }` enter the following **method**:

```
public async Task<IActionResult> OnPostGetAllCategoriesAsync()
{
    LoadToken();
    var results = await Api.GetAllCategoriesAsync(country);
    if (results?.Items != null)
    {
        Results = results.Items.Select(result => new Result()
        {
            Id = result.Id,
            Name = result.Name,
            Image = result.Images?.FirstOrDefault()?.Url
        });
    }
    return Page();
}
```

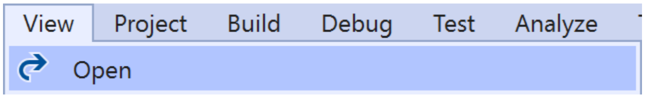
The **method** for `OnPostGetAllCategoriesAsync` is used to get all the **categories** on Spotify and populate the **property** for `Results` accordingly.

Step 5



In the **Solution Explorer** in the **Pages** section select **Index.cshtml**

Step 6



Then from the **Menu** choose **View** and then **Open**

Step 7

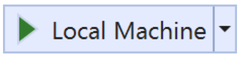
Once in the **Code View** for **Index.cshtml** above `<!-- Spotify Web API App Authorisation -->` enter the following:

```
<ul class="list-group mb-2">
    <li class="list-group-item list-group-item-primary">
        <h5 class="list-group-item-heading">Browse</h5>
    </li>
    <li class="list-group-item">
        <form asp-page-handler="GetAllCategories" method="post">
            <button class="btn btn-primary mb-2">
                Get All Categories
            </button>
        </form>
    </li>

    <!-- Browse -->
</ul>
```

This `form` will **post** to the **method** for `GetAllCategoriesAsync` and will output to the **Results**.

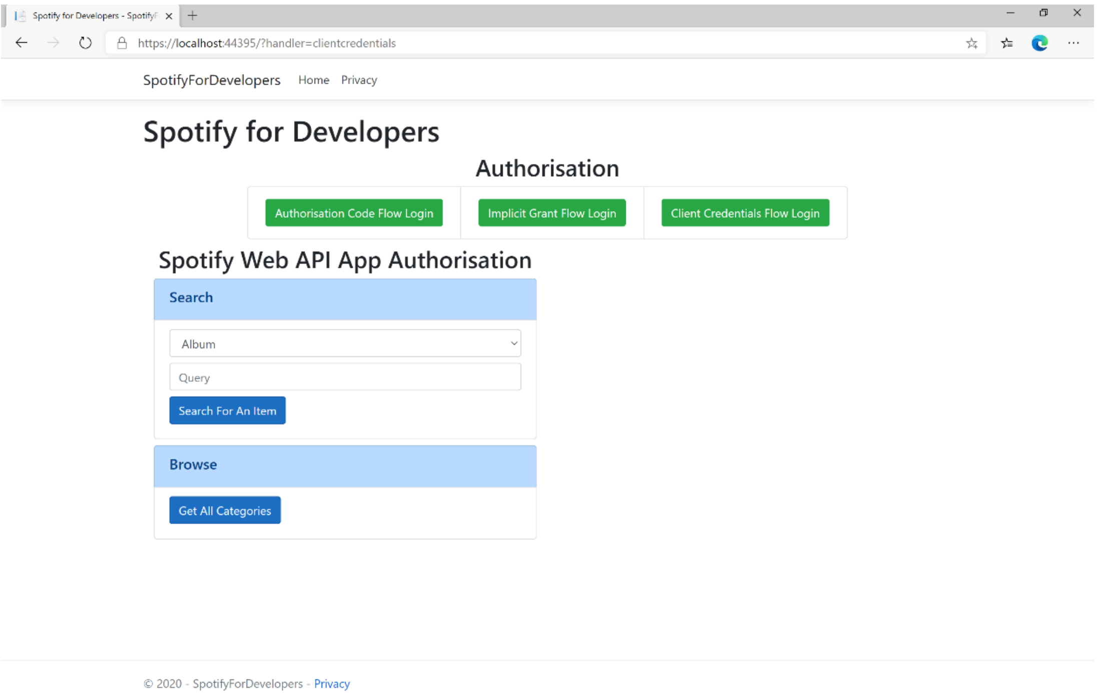
Step 8



Finally, in **Visual Studio 2019** select **IIS Application** to run the **Web Application**

Step 9

Once the **Web Application** is running and you select **Client Credentials Flow Login** you should see something like the following:



Spotify for Developers - Spotify
https://localhost:44395/?handler=clientcredentials
SpotifyForDevelopers
Home
Privacy

Spotify for Developers

Authorisation

Authorisation Code Flow Login
Implicit Grant Flow Login
Client Credentials Flow Login

Spotify Web API App Authorisation

Search

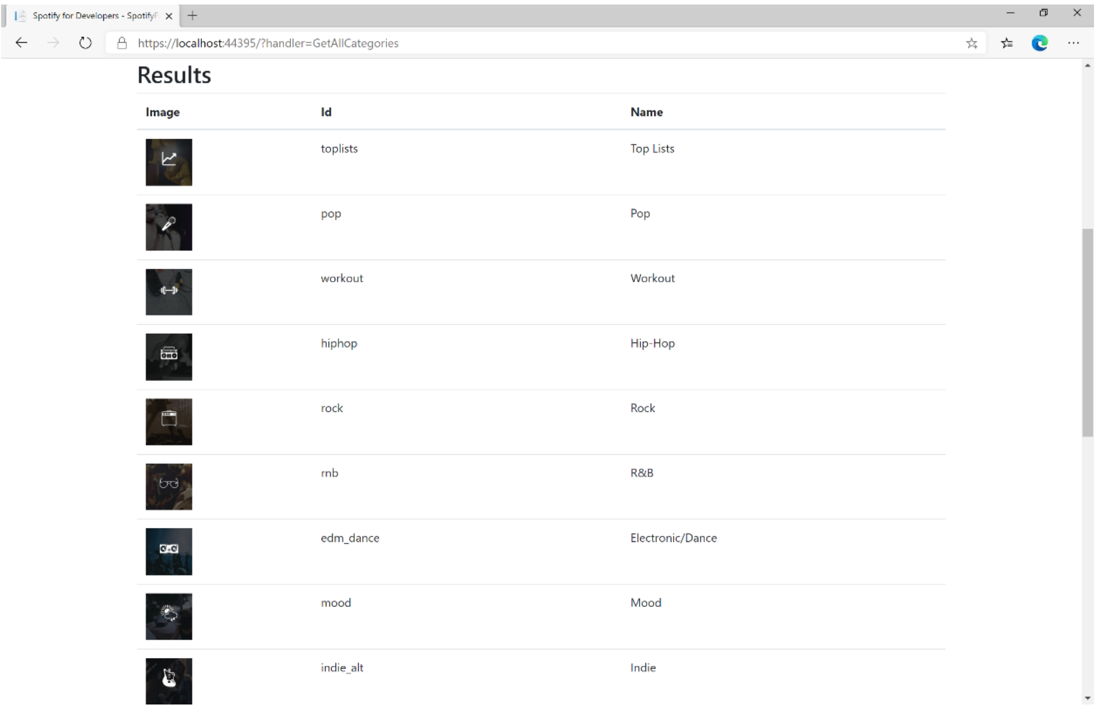
Album
Query
Search For An Item

Browse

© 2020 - SpotifyForDevelopers - Privacy

Step 10

You can then select **Get All Categories** and scroll down to view **Results** like the following:



Spotify for Developers - Spotify
https://localhost:44395/?handler=GetAllCategories

Results

Image	Id	Name
	toplists	Top Lists
	pop	Pop
	workout	Workout
	hiphop	Hip-Hop
	rock	Rock
	rnb	R&B
	edm_dance	Electronic/Dance
	mood	Mood
	indie_alt	Indie

Step 11

■

You can stop the **web application** in **Visual Studio 2019** by selecting the **Stop debugging** button

Step 12

✕

You can choose to exit **Visual Studio 2019** by selecting the **Close** button in the top right of the **application** as that completes this part of the workshop

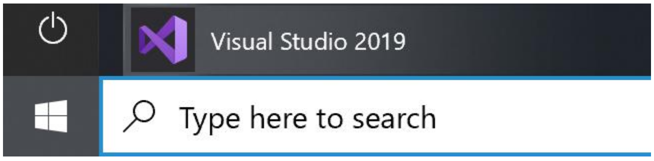
Get a Category

Get a single category used to tag items in Spotify, as shown in the Spotify player's "Browse" tab.

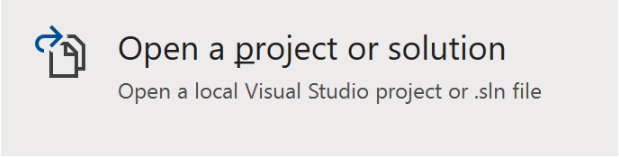
GET https://api.spotify.com/v1/browse/categories/{category_id}	
Header	
Authorization	Valid Access Token from Spotify Accounts service
Path Parameter	
category_id	Spotify category ID for the category
Query Parameter	
country	ISO 3166-1 alpha-2 country code e.g. "GB". If omitted globally relevant items returned
locale	Desired language, consisting of ISO 639-1 language code and ISO 3166-1 alpha-2 country code joined by an underscore e.g. "en_GB". If omitted results returned in American English

Header	Response
Success	
HTTP Status 200 OK	Category Object
Error	
Error Code	Error Object

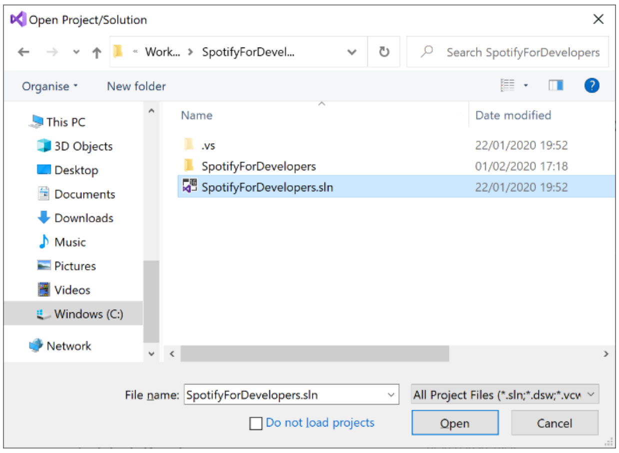
Step 1



If you chose to close **Visual Studio 2019** previously, in **Windows 10** choose **Start**, and then from the **Start Menu** find and select **Visual Studio 2019**


Open a project or solution
Open a local Visual Studio project or .sln file

Once done, from the **Get started** screen for **Visual Studio 2019** select **Open a project or solution**

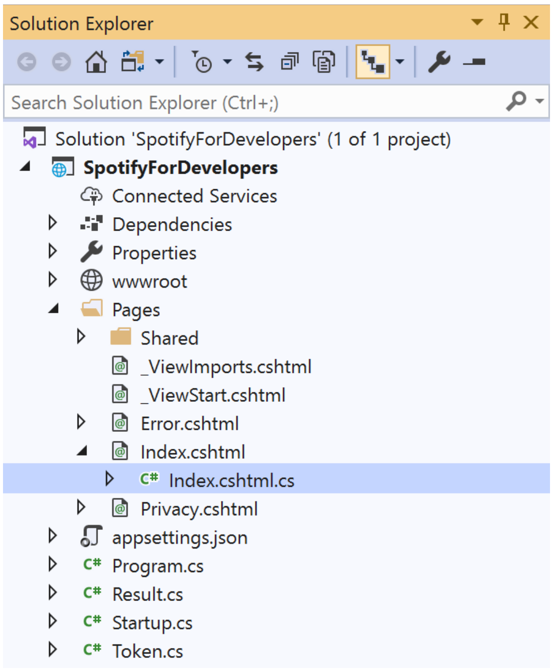


Name	Date modified
.vs	22/01/2020 19:52
SpotifyForDevelopers	01/02/2020 17:18
SpotifyForDevelopers.sln	22/01/2020 19:52

File name: SpotifyForDevelopers.sln

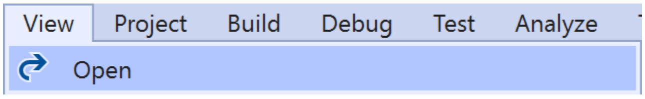
Then locate and select **SpotifyForDevelopers.sln** and select **Open** if you don't have this file already then please follow the previous parts of the workshop including **Getting Started**, **Authorisation Guide** and **Search & Browse**

Step 2



Once opened, in the **Solution Explorer** open the **Pages** section, then open the **Index.cshtml** section and select **Index.cshtml.cs**

Step 3



Then from the **Menu** choose **View** and then **Open**

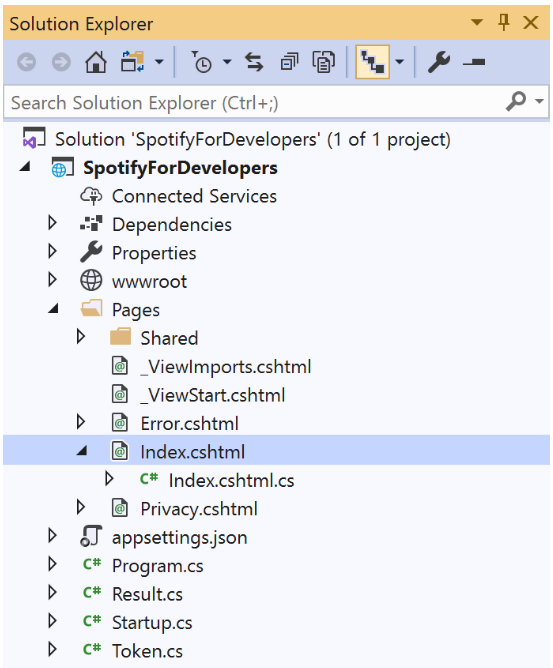
Step 4

In the **Code View** for **Index.cshtml.cs** below the **method** for `public async Task<IActionResult> OnPostGetAllCategoriesAsync() { ... }` enter the following **method**:

```
public async Task<IActionResult> OnPostGetCategoryAsync(string value)
{
    LoadToken();
    var result = await Api.GetCategoryAsync(value);
    if (result != null)
    {
        Results = new List<Result> { new Result()
        {
            Id = result.Id,
            Name = result.Name,
            Image = result.Images?.FirstOrDefault()?.Url
        }};
    }
    return Page();
}
```

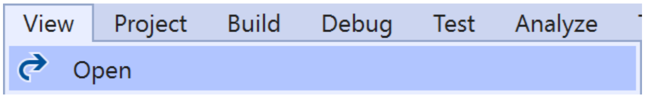
The **method** for `OnPostGetCategoryAsync` is used to get a **category** by **Category Id** on Spotify with the `value` and populate the **property** for `Results` accordingly.

Step 5



In the **Solution Explorer** in the **Pages** section select **Index.cshtml**

Step 6



Then from the **Menu** choose **View** and then **Open**

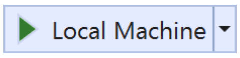
Step 7

Once in the **Code View** for **Index.cshtml** above `<!-- Browse -->` enter the following:

```
<li class="list-group-item">
    <form asp-page-handler="GetCategory" method="post">
        <input asp-for="Value" placeholder="Category Id" class="form-control mb-2" />
        <button class="btn btn-primary mb-2">
            Get Category
        </button>
    </form>
</li>
```

This `form` will **post** to the **method** for `OnPostGetCategoryAsync` with the `Value` as the **Category Id** and will output to the **Results**.

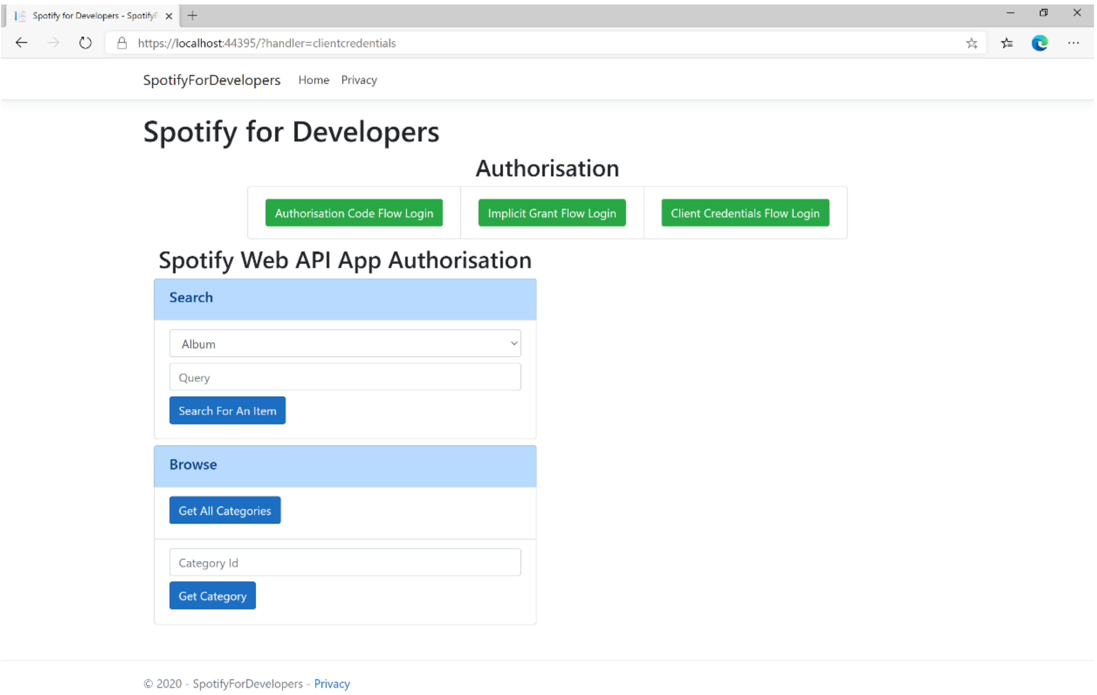
Step 8



Finally, in **Visual Studio 2019** select **IIS Application** to run the **Web Application**

Step 9

Once the **Web Application** is running and you select **Client Credentials Flow Login** you should see something like the following:



Spotify for Developers - Spotify

https://localhost:44395/?handler=clientcredentials

SpotifyForDevelopers Home Privacy

Spotify for Developers

Authorisation

Authorisation Code Flow Login | Implicit Grant Flow Login | Client Credentials Flow Login

Spotify Web API App Authorisation

Search

Album

Query

Search For An Item

Browse

Get All Categories

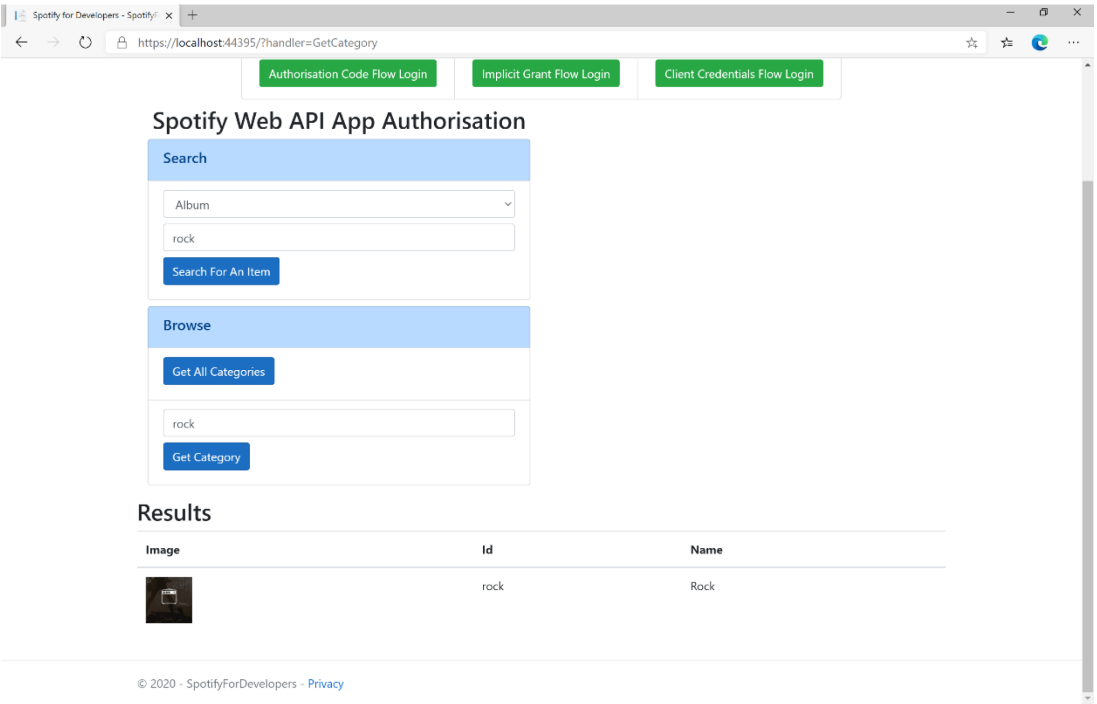
Category Id

Get Category

© 2020 - SpotifyForDevelopers - Privacy

Step 10

You can then enter a **Category Id** from **Get All Categories** and select **Get Category** and scroll down to view **Results** like the following:



Spotify for Developers - Spotify

https://localhost:44395/?handler=GetCategory

Authorisation Code Flow Login | Implicit Grant Flow Login | Client Credentials Flow Login

Spotify Web API App Authorisation

Search

Album

rock

Search For An Item

Browse

Get All Categories

rock

Get Category

Results

Image	Id	Name
	rock	Rock

© 2020 - SpotifyForDevelopers - Privacy

Step 11

■

You can stop the **web application** in **Visual Studio 2019** by selecting the **Stop debugging** button

Step 12

×

You can choose to exit **Visual Studio 2019** by selecting the **Close** button in the top right of the **application** as that completes this part of the workshop

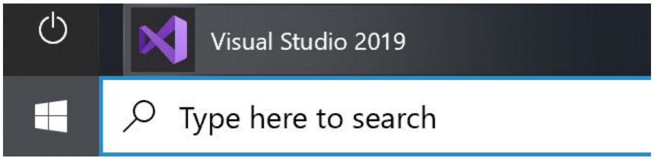
Get a Category's Playlists

Get a list of Spotify playlists tagged with a category.

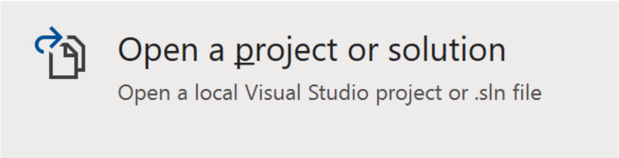
GET https://api.spotify.com/v1/browse/categories/{category_id}/playlists	
Header	
Authorization	Valid Access Token from Spotify Accounts service
Path Parameter	
category_id	Spotify category ID for the category
Query Parameter	
country	ISO 3166-1 alpha-2 country code e.g. “GB”. If omitted globally relevant items returned
limit	Maximum number of results to return
offset	Index of first result to return

Header	Response
Success	
HTTP Status 200 OK	Array of Simplified Playlist Objects wrapped in a Paging Object
Error	
Error Code	Error Object

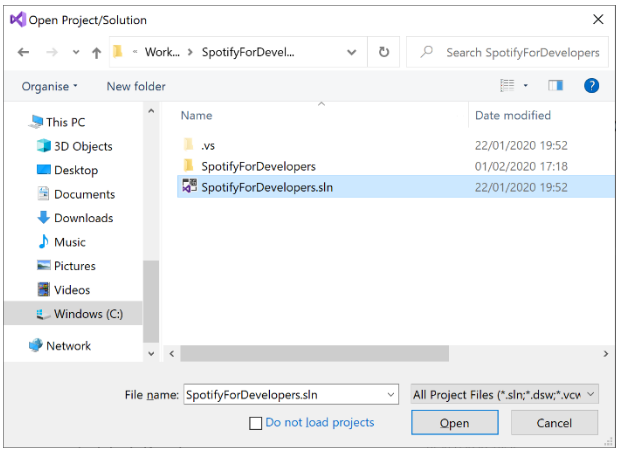
Step 1



If you chose to close **Visual Studio 2019** previously, in **Windows 10** choose **Start**, and then from the **Start Menu** find and select **Visual Studio 2019**


Open a project or solution
Open a local Visual Studio project or .sln file

Once done, from the **Get started** screen for **Visual Studio 2019** select **Open a project or solution**

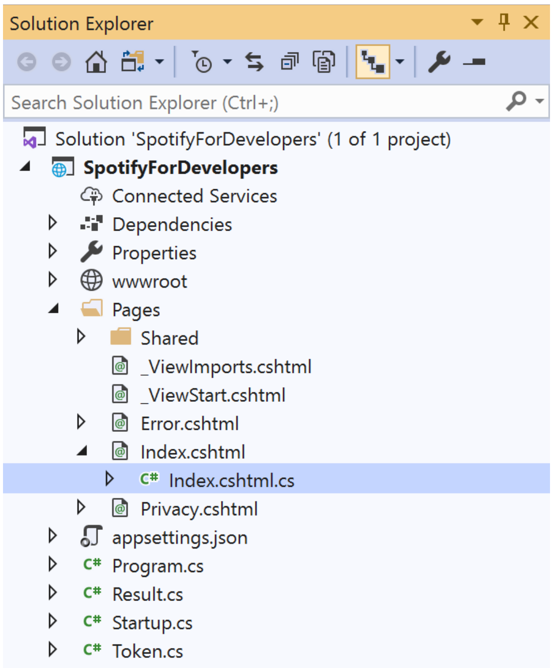


Name	Date modified
.vs	22/01/2020 19:52
SpotifyForDevelopers	01/02/2020 17:18
SpotifyForDevelopers.sln	22/01/2020 19:52

File name: SpotifyForDevelopers.sln

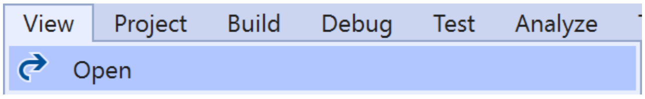
Then locate and select **SpotifyForDevelopers.sln** and select **Open** if you don’t have this file already then please follow the previous parts of the workshop including **Getting Started**, **Authorisation Guide** and **Search & Browse**

Step 2



Once opened, in the **Solution Explorer** open the **Pages** section, then open the **Index.cshtml** section and select **Index.cshtml.cs**

Step 3



Then from the **Menu** choose **View** and then **Open**

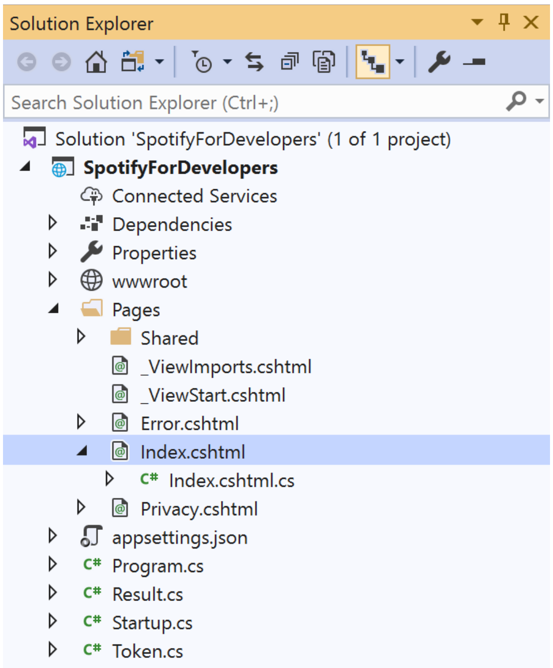
Step 4

In the **Code View** for **Index.cshtml.cs** below the **method** for `public async Task<IActionResult> OnPostGetCategoryAsync() { ... }` enter the following **method**:

```
public async Task<IActionResult> OnPostGetCategoryPlaylistsAsync(string value)
{
    LoadToken();
    var results = await Api.GetCategoryPlaylistsAsync(value);
    if (results?.Items != null)
    {
        Results = results.Items.Select(result => new Result()
        {
            Id = result.Id,
            Name = result.Name,
            Image = result?.Images?.FirstOrDefault()?.Url
        });
    }
    return Page();
}
```

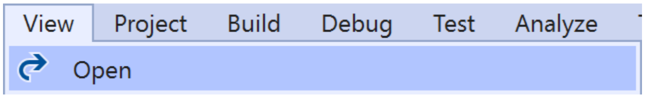
The **method** for `OnPostGetCategoryPlaylistsAsync` is used to get the **playlists** for a **category** by **Category Id** on Spotify with the `value` and populate the **property** for `Results` accordingly.

Step 5



In the **Solution Explorer** in the **Pages** section select **Index.cshtml**

Step 6



Then from the **Menu** choose **View** and then **Open**

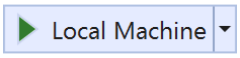
Step 7

Once in the **Code View** for **Index.cshtml** above `<!-- Browse -->` enter the following:

```
<li class="list-group-item">
    <form asp-page-handler="GetCategoryPlaylists" method="post">
        <input asp-for="Value" placeholder="Category Id" class="form-control mb-2" />
        <button class="btn btn-primary mb-2">
            Get Category Playlists
        </button>
    </form>
</li>
```

This `form` will **post** to the **method** for `OnPostGetCategoryPlaylistsAsync` with the `Value` as the **Category Id** and will output to the **Results**.

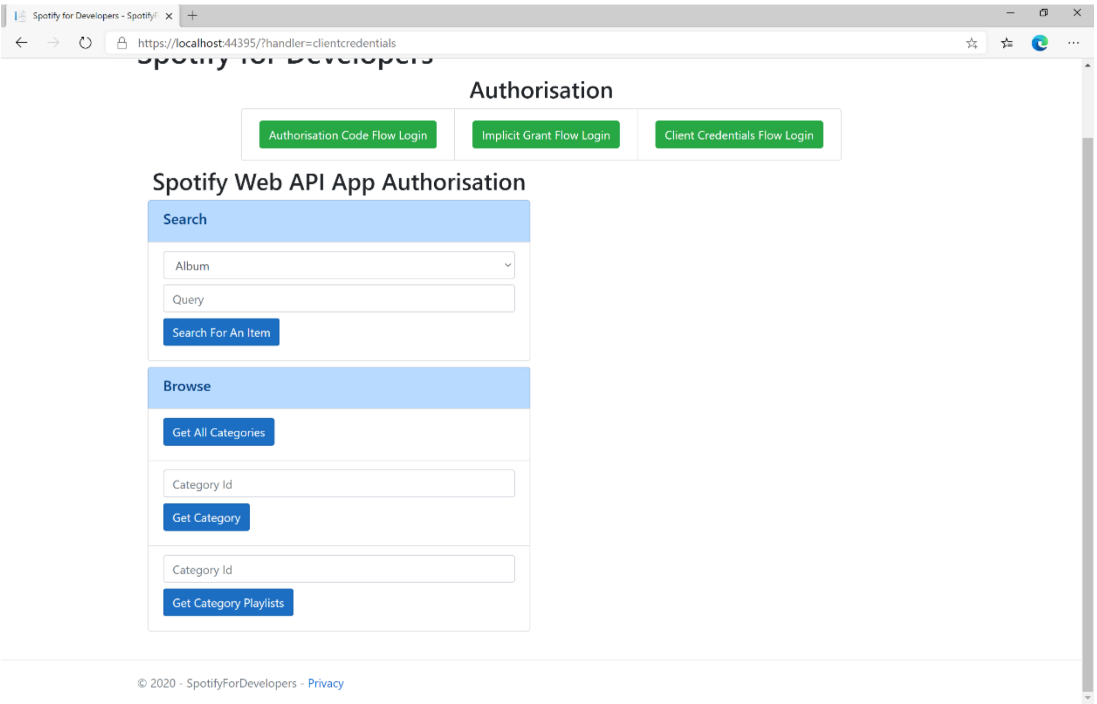
Step 8



Finally, in **Visual Studio 2019** select **IIS Application** to run the **Web Application**

Step 9

Once the **Web Application** is running and you select **Client Credentials Flow Login** and scroll down you should see something like the following:



Spotify for Developers - SpotifyF

https://localhost:44395/?handler=clientcredentials

Spotify for Developers

Authorisation

Authorisation Code Flow Login
Implicit Grant Flow Login
Client Credentials Flow Login

Spotify Web API App Authorisation

Search

Album

Query

Browse

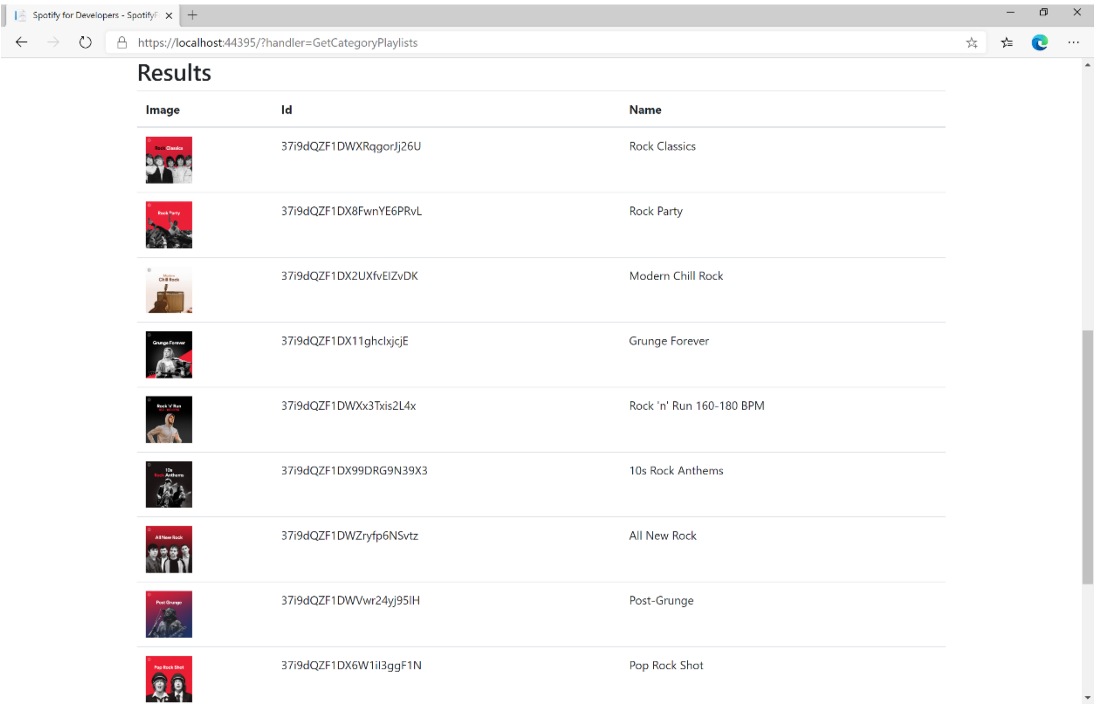
Category Id

Category Id

© 2020 - SpotifyForDevelopers - Privacy

Step 10

You can then enter a **Category Id** from **Get All Categories** and select **Get Category Playlists** and scroll down to view **Results** like the following:



Spotify for Developers - SpotifyF

https://localhost:44395/?handler=GetCategoryPlaylists

Results

Image	Id	Name
	37i9dQZF1DWXRqgorJj26U	Rock Classics
	37i9dQZF1DX8FwnYE6PRvL	Rock Party
	37i9dQZF1DX2UXfvEIZvDK	Modern Chill Rock
	37i9dQZF1DX11ghclxjcjE	Grunge Forever
	37i9dQZF1DWXx3Txis2L4x	Rock 'n' Run 160-180 BPM
	37i9dQZF1DX99DRG9N39X3	10s Rock Anthems
	37i9dQZF1DWZryfp6NSvtz	All New Rock
	37i9dQZF1DWVwr24yj95IH	Post-Grunge
	37i9dQZF1DX6W1il3ggF1N	Pop Rock Shot

Step 11

■

You can stop the **web application** in **Visual Studio 2019** by selecting the **Stop debugging** button

Step 12

×

You can choose to exit **Visual Studio 2019** by selecting the **Close** button in the top right of the **application** as that completes this part of the workshop

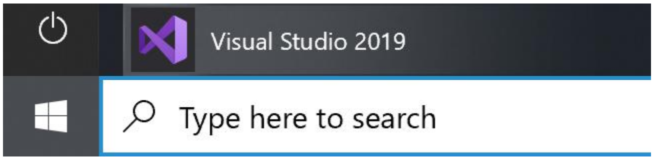
Get Recommendation Genres

Retrieve a list of available genres seed parameter values for recommendations.

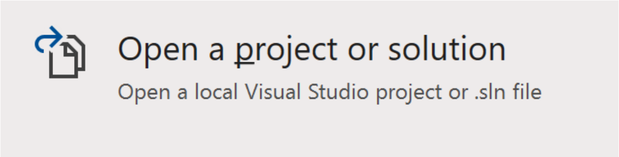
GET https://api.spotify.com/v1/recommendations/available-genre-seeds	
Header	
Authorization	Valid Access Token from Spotify Accounts service

Header	Response
Success	
HTTP Status 200 OK	Array of Genre Seeds
Error	
Error Code	Error Object

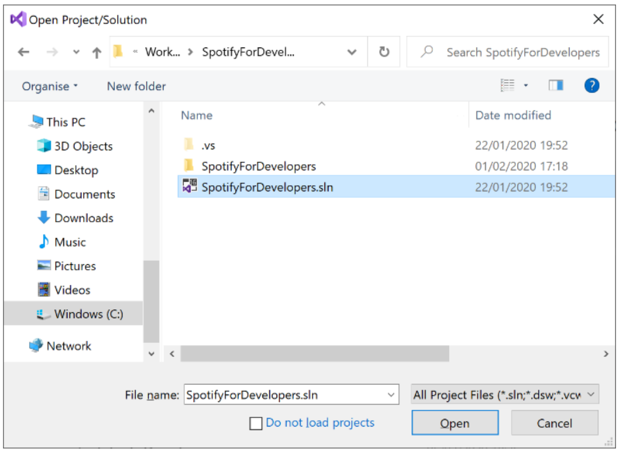
Step 1



If you chose to close **Visual Studio 2019** previously, in **Windows 10** choose **Start**, and then from the **Start Menu** find and select **Visual Studio 2019**


Open a project or solution
Open a local Visual Studio project or .sln file

Once done, from the **Get started** screen for **Visual Studio 2019** select **Open a project or solution**

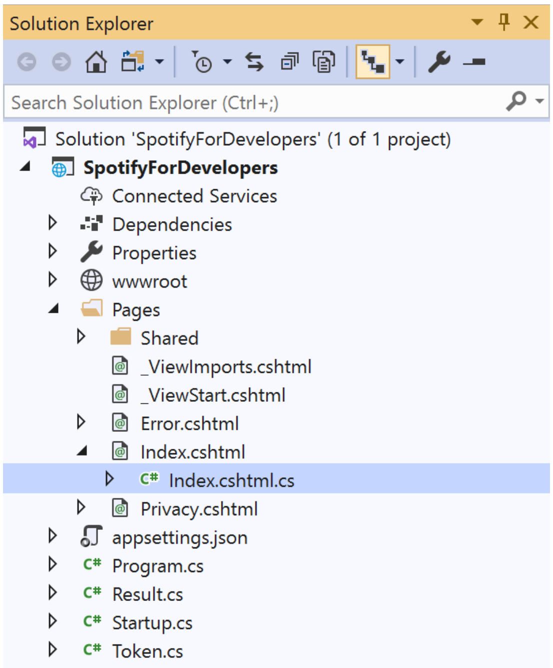


Name	Date modified
.vs	22/01/2020 19:52
SpotifyForDevelopers	01/02/2020 17:18
SpotifyForDevelopers.sln	22/01/2020 19:52

File name: SpotifyForDevelopers.sln

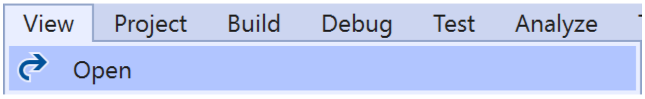
Then locate and select **SpotifyForDevelopers.sln** and select **Open** if you don't have this file already then please follow the previous parts of the workshop including **Getting Started**, **Authorisation Guide** and **Search & Browse**

Step 2



Once opened, in the **Solution Explorer** open the **Pages** section, then open the **Index.cshtml** section and select **Index.cshtml.cs**

Step 3



Then from the **Menu** choose **View** and then **Open**

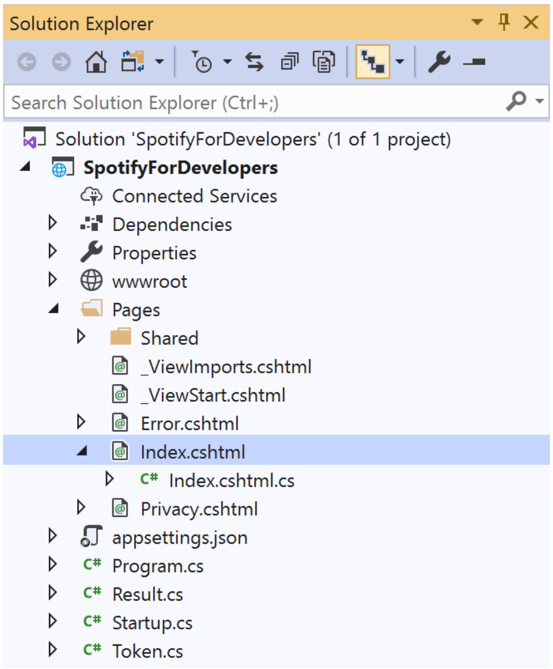
Step 4

In the **Code View** for **Index.cshtml.cs** below the **method** for `public async Task<IActionResult> OnPostGetCategoryPlaylistsAsync() { ... }` enter the following **method**:

```
public async Task<IActionResult> OnPostGetRecommendationGenresAsync()
{
    LoadToken();
    var results = await Api.GetRecommendationGenresAsync();
    if (results?.Genres != null)
    {
        Results = results.Genres.Select(result => new Result()
        {
            Id = result
        });
    }
    return Page();
}
```

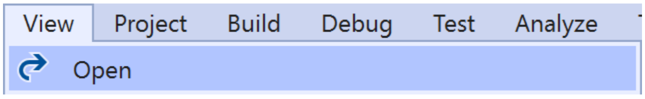
The **method** for `OnPostGetRecommendationGenresAsync` is used to get the **genres** for **recoomendations** on Spotify and populate the **property** for `Results` accordingly.

Step 5



In the **Solution Explorer** in the **Pages** section select **Index.cshtml**

Step 6



Then from the **Menu** choose **View** and then **Open**

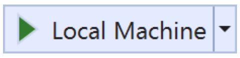
Step 7

Once in the **Code View** for **Index.cshtml** above `<!-- Browse -->` enter the following:

```
<li class="list-group-item">
    <form asp-page-handler="GetRecommendationGenres" method="post">
        <button class="btn btn-primary mb-2">
            Get Recommendation Genres
        </button>
    </form>
</li>
```

This `form` will **post** to the **method** for `OnPostGetRecommendationGenresAsync` and will output to the **Results**.

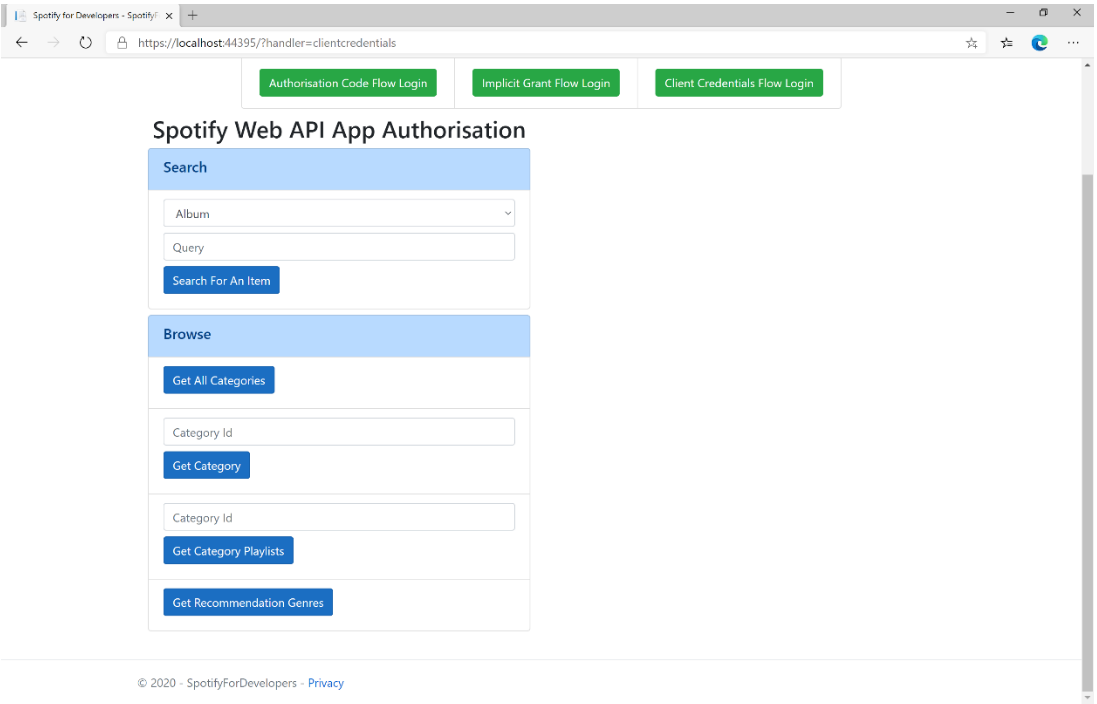
Step 8



Finally, in **Visual Studio 2019** select **IIS Application** to run the **Web Application**

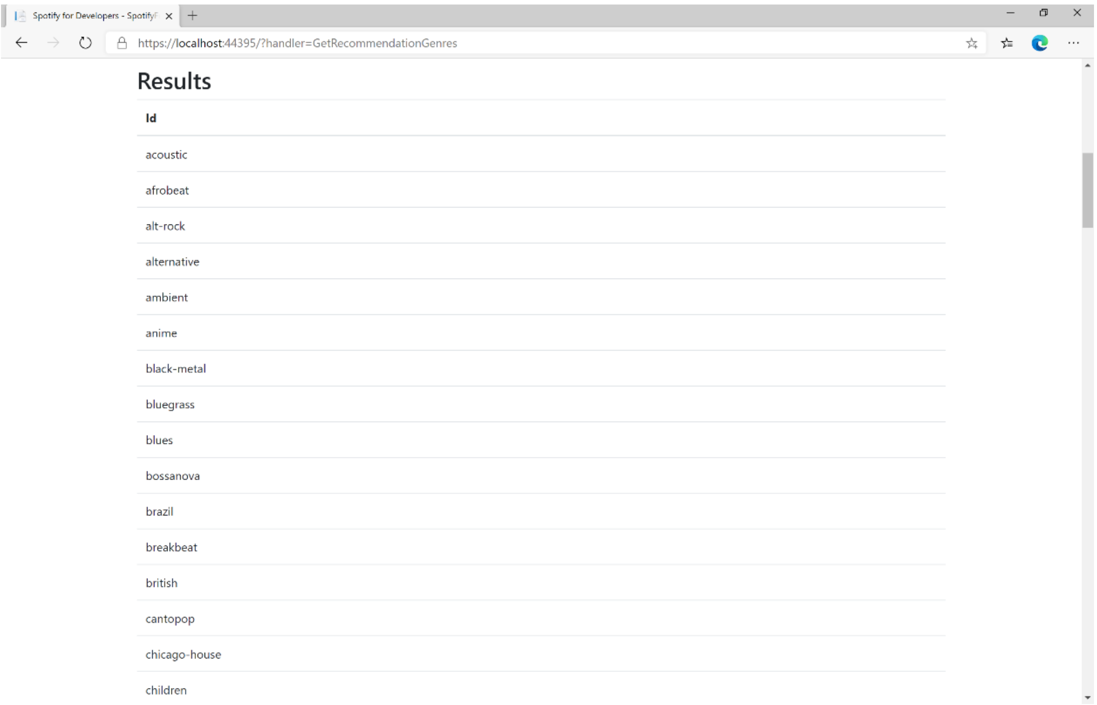
Step 9

Once the **Web Application** is running and you select **Client Credentials Flow Login** and scroll down you should see something like the following:


Spotify for Developers - Spotify
https://localhost:44395/?handler=clientcredentials
Authorisation Code Flow Login
Implicit Grant Flow Login
Client Credentials Flow Login
Spotify Web API App Authorisation
Search
Album
Query
Search For An Item
Browse
Get All Categories
Category Id
Get Category
Category Id
Get Category Playlists
Get Recommendation Genres
© 2020 - SpotifyForDevelopers - Privacy

Step 10

You can select **Get Recommendation Genres** and scroll down to view **Results** like the following:


Spotify for Developers - Spotify
https://localhost:44395/?handler=GetRecommendationGenres

Results

Id
acoustic
afrobeat
alt-rock
alternative
ambient
anime
black-metal
bluegrass
blues
bossanova
brazil
breakbeat
british
cantopop
chicago-house
children

Step 11

■

You can stop the **web application** in **Visual Studio 2019** by selecting the **Stop debugging** button

Step 12

×

You can choose to exit **Visual Studio 2019** by selecting the **Close** button in the top right of the **application** as that completes this part of the workshop

Get Recommendations

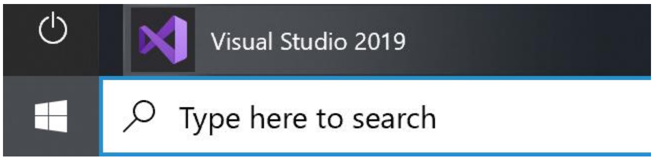
Recommendations are generated based on the available information for a given seed entity and matched against similar artists and tracks. If there is enough information about the provided seeds, a list of tracks will be returned together with pool size details.

GET https://api.spotify.com/v1/recommendations	
Header	
Authorization	Valid Access Token from Spotify Accounts service
Query Parameter	
seed_artists	Comma separated list of Spotify IDs for seed artists. Up to 5 seed values may be provided in any combination of seed_artists, seed_tracks and seed_genres
seed_genres	Comma separated list of any genres in the set of available genre seeds. Up to 5 seed values may be provided in any combination of seed_artists, seed_tracks and seed_genres
seed_tracks	Comma separated list of Spotify IDs for a seed track. Up to 5 seed values may be provided in any combination of seed_artists, seed_tracks and seed_genres
limit	Maximum number of results to return
market	ISO 3166-1 alpha-2 country code e.g. “GB”. Provide to enable Track Relinking
min_*	For each Tuneable Track attribute, a hard floor on the selected track attribute’s value can be provided
max_*	For each Tuneable Track attribute, a hard ceiling on the selected track attribute’s value can be provided
target_*	For each of the Tuneable Track attributes a target value may be provided. Tracks with the attribute values nearest to the target values will be preferred

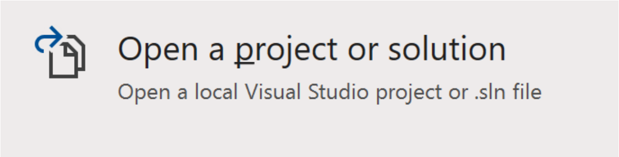
Tuneable Track Attributes	
acousticness	A confidence measure from 0.0 to 1.0 of whether the track is acoustic.
danceability	Describes how suitable a track is for dancing based on a combination of musical elements including tempo, rhythm stability, beat strength, and overall regularity
duration_ms	Duration of the track in milliseconds
energy	Measure from 0.0 to 1.0 and represents a perceptual measure of intensity and activity
instrumentalness	Predicts whether a track contains no vocal
key	Key the track is in. Integers map to pitches using standard Pitch Class notation
liveness	Detects presence of an audience in the recording
loudness	Overall loudness of a track in decibels (dB)
mode	Mode indicates the modality of a track (major or minor)
popularity	Popularity of track, between 0 and 100, with 100 being the most popular
speechiness	Detects the presence of spoken words in a track
tempo	Estimated tempo of a track in beats per minute (BPM)
time_signature	Estimated overall time signature of a track
valence	Measure from 0.0 to 1.0 describing the musical positiveness conveyed by a track

Header	Response
Success	
HTTP Status 200 OK	Recommendations Response Object
Error	
Error Code	Error Object

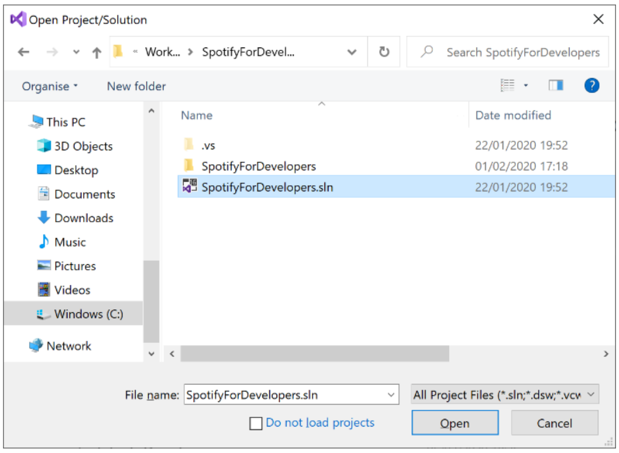
Step 1



If you chose to close **Visual Studio 2019** previously, in **Windows 10** choose **Start**, and then from the **Start Menu** find and select **Visual Studio 2019**

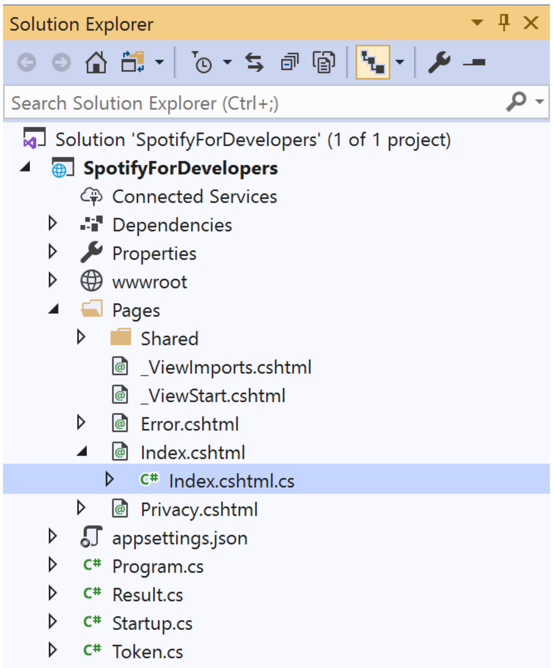

Open a project or solution
Open a local Visual Studio project or .sln file

Once done, from the **Get started** screen for **Visual Studio 2019** select **Open a project or solution**


Open Project/Solution
Work... › SpotifyForDevel...
Search SpotifyForDevelopers
Organise
New folder
This PC
3D Objects
Desktop
Documents
Downloads
Music
Pictures
Videos
Windows (C:)
Network
Name
Date modified
.vs 22/01/2020 19:52
SpotifyForDevelopers 01/02/2020 17:18
SpotifyForDevelopers.sln 22/01/2020 19:52
File name: SpotifyForDevelopers.sln
All Project Files (*.sln;*.dsw;*.vcw
□ Do not load projects
Open
Cancel

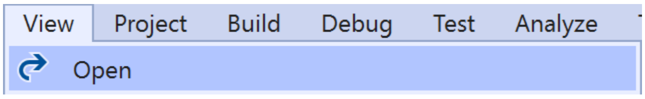
Then locate and select **SpotifyForDevelopers.sln** and select **Open** if you don't have this file already then please follow the previous parts of the workshop including **Getting Started**, **Authorisation Guide** and **Search & Browse**

Step 2



Once opened, in the **Solution Explorer** open the **Pages** section, then open the **Index.cshtml** section and select **Index.cshtml.cs**

Step 3



Then from the **Menu** choose **View** and then **Open**

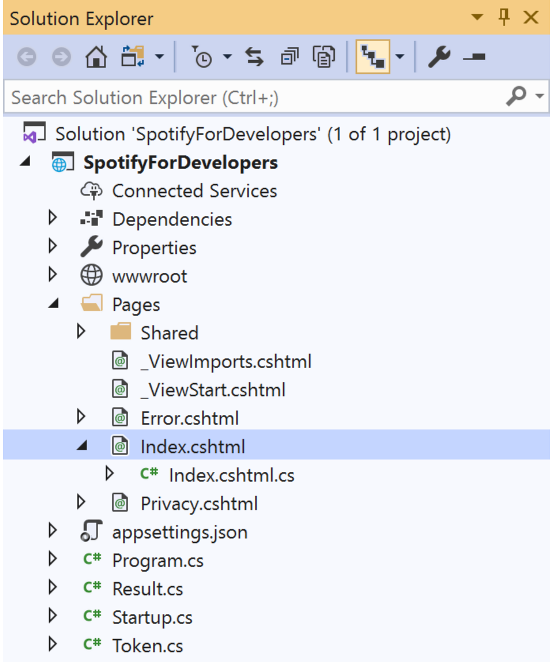
Step 4

In the **Code View** for **Index.cshtml.cs** below the **method** for `public async Task<IActionResult> OnPostGetRecommendationGenresAsync() { ... }` enter the following **method**:

```
public async Task<IActionResult> OnPostGetRecommendationsAsync(string value)
{
    LoadToken();
    var results = await Api.GetRecommendationsAsync(
        seedGenres: new List<string> { value });
    if (results?.Tracks != null)
    {
        Results = results.Tracks.Select(result => new Result()
        {
            Id = result.Id,
            Name = result.Name,
            Inner = new Result()
            {
                Id = result?.Artists?.FirstOrDefault()?.Id,
                Name = result?.Artists?.FirstOrDefault()?.Name
            }
        });
    }
    return Page();
}
```

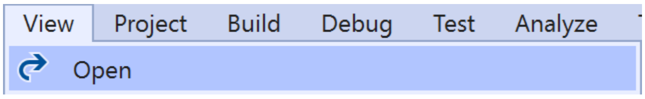
The **method** for `OnPostGetRecommendationsAsync` is used to get **recommendations** by **genre** on Spotify with the `value` and populate the **property** for `Results` accordingly.

Step 5



In the **Solution Explorer** in the **Pages** section select **Index.cshtml**

Step 6



Then from the **Menu** choose **View** and then **Open**

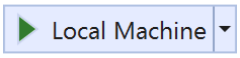
Step 7

Once in the **Code View** for **Index.cshtml** above `<!-- Browse -->` enter the following:

```
<li class="list-group-item">
    <form asp-page-handler="GetRecommendations" method="post">
        <input asp-for="Value" placeholder="Genre" class="form-control mb-2" />
        <button class="btn btn-primary mb-2">
            Get Recommendations
        </button>
    </form>
</li>
```

This `form` will **post** to the **method** for `OnPostGetRecommendationsAsync` with the `Value` as the **Genre** and will output to the **Results**.

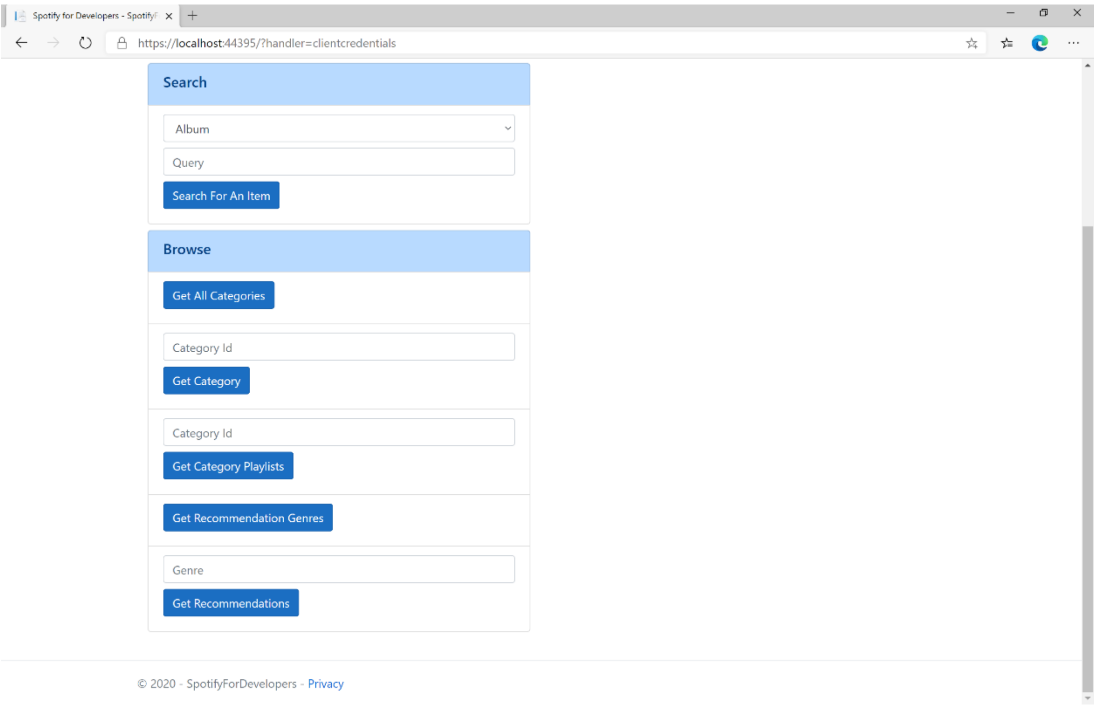
Step 8



Finally, in **Visual Studio 2019** select **IIS Application** to run the **Web Application**

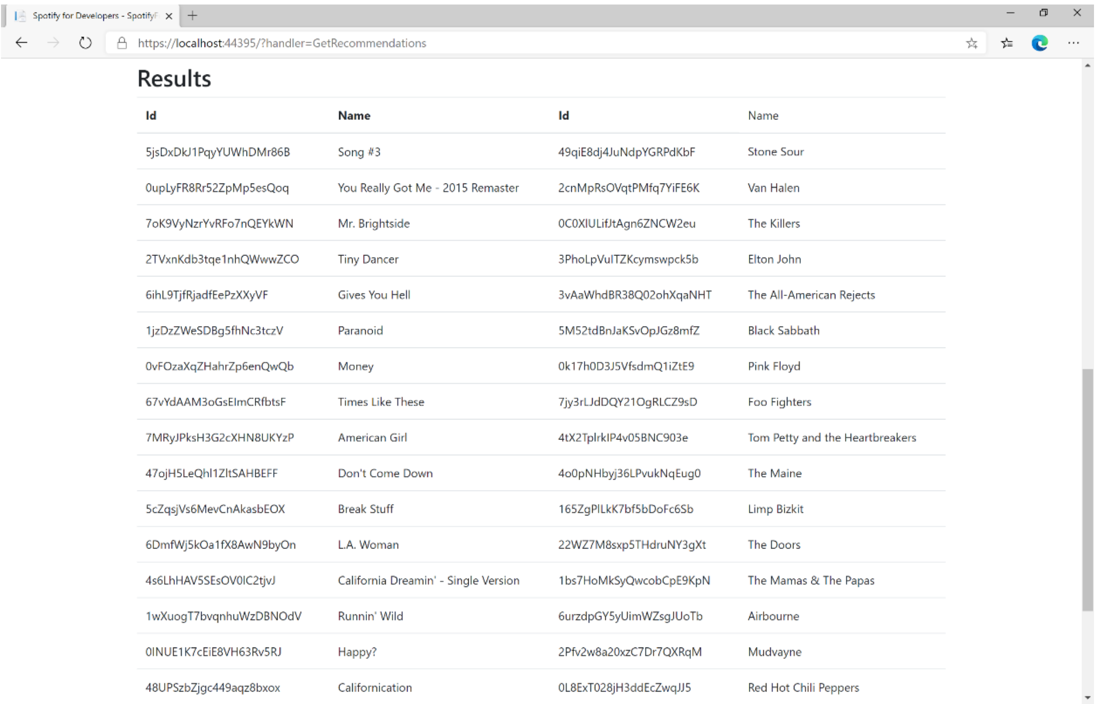
Step 9

Once the **Web Application** is running and you select **Client Credentials Flow Login** and scroll down you should see something like the following:


Spotify for Developers - Spotify
https://localhost:44395/?handler=clientcredentials
Search
Album
Query
Search For An Item
Browse
Get All Categories
Category Id
Get Category
Category Id
Get Category Playlists
Get Recommendation Genres
Genre
Get Recommendations
© 2020 - SpotifyForDevelopers - Privacy

Step 10

You can then enter a **Genre** from **Get Recommendation Genres** and select **Get Recommendations** and scroll down to view **Results** like the following:



https://localhost:44395/?handler=GetRecommendations

Results

Id	Name	Id	Name
5jsDxDkJ1PqyYUWhDMr86B	Song #3	49qiE8dj4JuNdpYGRPdKbF	Stone Sour
0upLyFR8Rr52ZpMp5esQoq	You Really Got Me - 2015 Remaster	2cnMpRsOVqtPMfq7YiFE6K	Van Halen
7oK9VyNzrYvRFo7nQEYkWN	Mr. Brightside	0C0XlULifJtAgn6ZNCW2eu	The Killers
2TVxnKdb3tqe1nhQWwwZCO	Tiny Dancer	3PhoLpVuITZKcymswpck5b	Elton John
6ihL9TjfRjadfEePzXXyVF	Gives You Hell	3vAaWhdBR38Q02ohXqaNHT	The All-American Rejects
1jzDzZWeSDBg5fhNc3tczV	Paranoid	5M52tdBnJaKSvOpJGz8mfZ	Black Sabbath
0vFOzaXqZHahrZp6enQwQb	Money	0k17h0D3J5VfsdmQ1iZtE9	Pink Floyd
67vYdAAM3oGsElmCRfbtsF	Times Like These	7jy3rLJdDQY21OgRLCZ9sD	Foo Fighters
7MRyJPksH3G2cXHN8UKYzP	American Girl	4tX2TplrkIP4v05BNC903e	Tom Petty and the Heartbreakers
47ojH5LeQhl1ZltSAHBEFF	Don't Come Down	4o0pNHbyj36LPvukNqEug0	The Maine
5cZqsjVs6MevCnAkasbEOX	Break Stuff	165ZgPlLkK7bf5bDoFc6Sb	Limp Bizkit
6DmfWj5kOa1fX8AwN9byOn	L.A. Woman	22WZ7M8sxp5THdruNY3gXt	The Doors
4s6LhHAV5SEsOV0lC2tjvJ	California Dreamin' - Single Version	1bs7HoMkSyQwcobCpE9KpN	The Mamas & The Papas
1wXuogT7bvqnhuWzDBNOdV	Runnin' Wild	6urzdpGY5yUimWZsgJUoTb	Airbourne
0INUE1K7cEiE8VH63Rv5RJ	Happy?	2Pfv2w8a20xzC7Dr7QXRqM	Mudvayne
48UPSzbZjgc449aqz8bxox	Californication	0L8ExT028jH3ddEcZwqJJ5	Red Hot Chili Peppers

Step 11

■

You can stop the **web application** in **Visual Studio 2019** by selecting the **Stop debugging** button

Step 12

×

You can choose to exit **Visual Studio 2019** by selecting the **Close** button in the top right of the **application** as that completes this part of the workshop

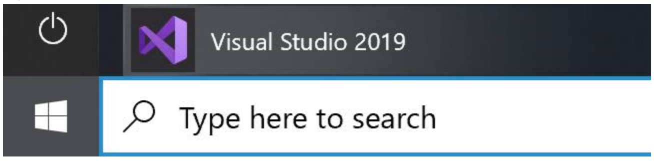
Get All New Releases

Get a list of new album releases featured in Spotify as shown in the Spotify player's "Browse" tab.

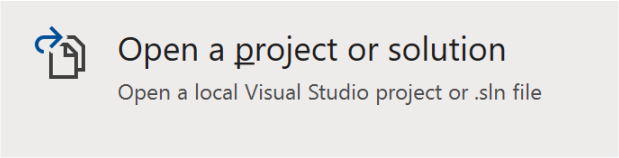
GET https://api.spotify.com/v1/browse/new-releases	
Header	
Authorization	Valid Access Token from Spotify Accounts service
Query Parameter	
country	ISO 3166-1 alpha-2 country code e.g. "GB". If omitted globally relevant items returned
limit	Maximum number of results to return
offset	Index of first result to return

Header	Response
Success	
HTTP Status 200 OK	Array of Simplified Album Objects wrapped in a Paging Object
Error	
Error Code	Error Object

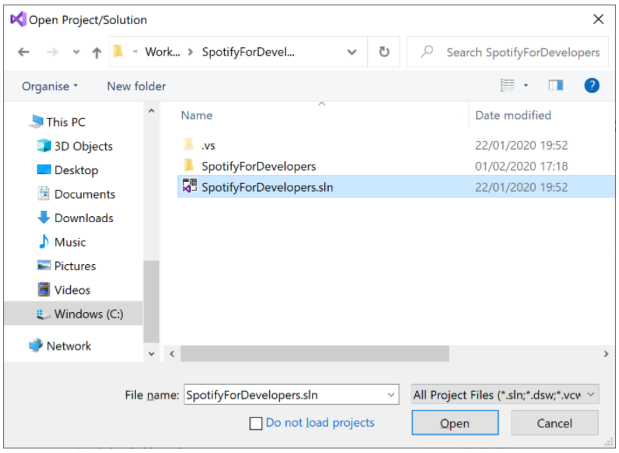
Step 1



If you chose to close **Visual Studio 2019** previously, in **Windows 10** choose **Start**, and then from the **Start Menu** find and select **Visual Studio 2019**

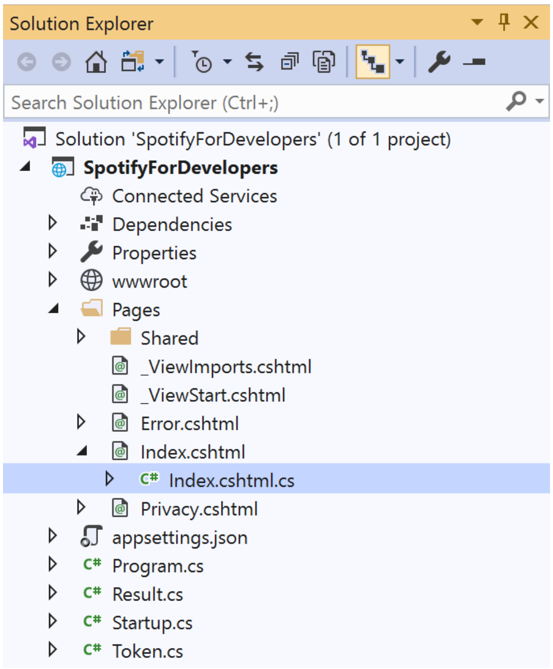

Open a project or solution
Open a local Visual Studio project or .sln file

Once done, from the **Get started** screen for **Visual Studio 2019** select **Open a project or solution**



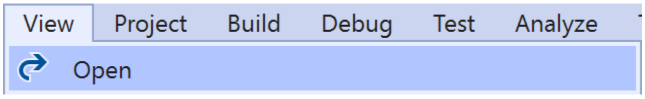
Then locate and select **SpotifyForDevelopers.sln** and select **Open** if you don't have this file already then please follow the previous parts of the workshop including **Getting Started**, **Authorisation Guide** and **Search & Browse**

Step 2



Once opened, in the **Solution Explorer** open the **Pages** section, then open the **Index.cshtml** section and select **Index.cshtml.cs**

Step 3



Then from the **Menu** choose **View** and then **Open**

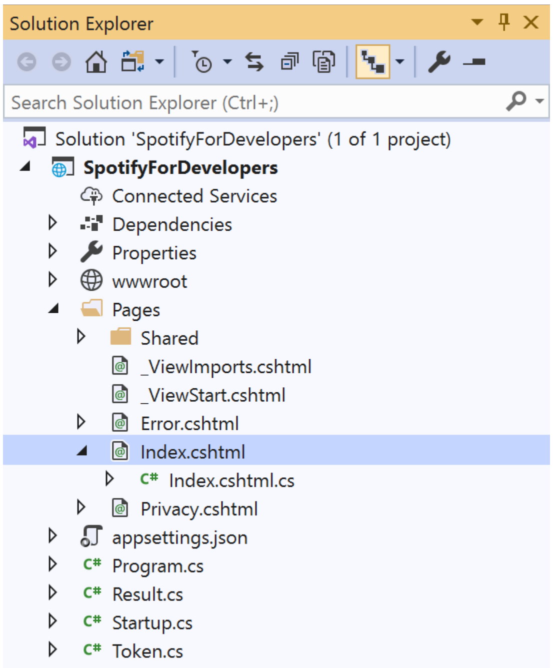
Step 4

In the **Code View** for **Index.cshtml.cs** below the **method** for `public async Task<IActionResult> OnPostGetRecommendationsAsync() { ... }` enter the following **method**:

```
public async Task<IActionResult> OnPostGetAllNewReleasesAsync()
{
    LoadToken();
    var results = await Api.GetAllNewReleasesAsync(country);
    if (results?.Items != null)
    {
        Results = results.Items.Select(result => new Result()
        {
            Id = result.Id,
            Name = result.Name,
            Image = result.Images?.FirstOrDefault()?.Url,
            Inner = new Result()
            {
                Id = result?.Artists?.FirstOrDefault()?.Id,
                Name = result?.Artists?.FirstOrDefault()?.Name
            }
        });
    }
    return Page();
}
```

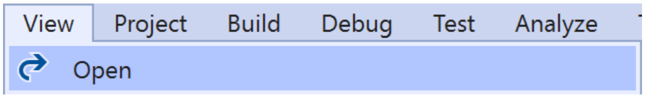
The **method** for `OnPostGetAllNewReleasesAsync` is used to get **all new releases** by `country` on Spotify and populate the **property** for `Results` accordingly.

Step 5



In the **Solution Explorer** in the **Pages** section select **Index.cshtml**

Step 6



Then from the **Menu** choose **View** and then **Open**

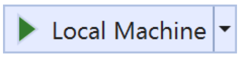
Step 7

Once in the **Code View** for **Index.cshtml** above `<!-- Browse -->` enter the following:

```
<li class="list-group-item">
    <form asp-page-handler="GetAllNewReleases" method="post">
        <button class="btn btn-primary mb-2">
            Get All New Releases
        </button>
    </form>
</li>
```

This `form` will **post** to the **method** for `OnPostGetAllNewReleasesAsync` and will output to the **Results**.

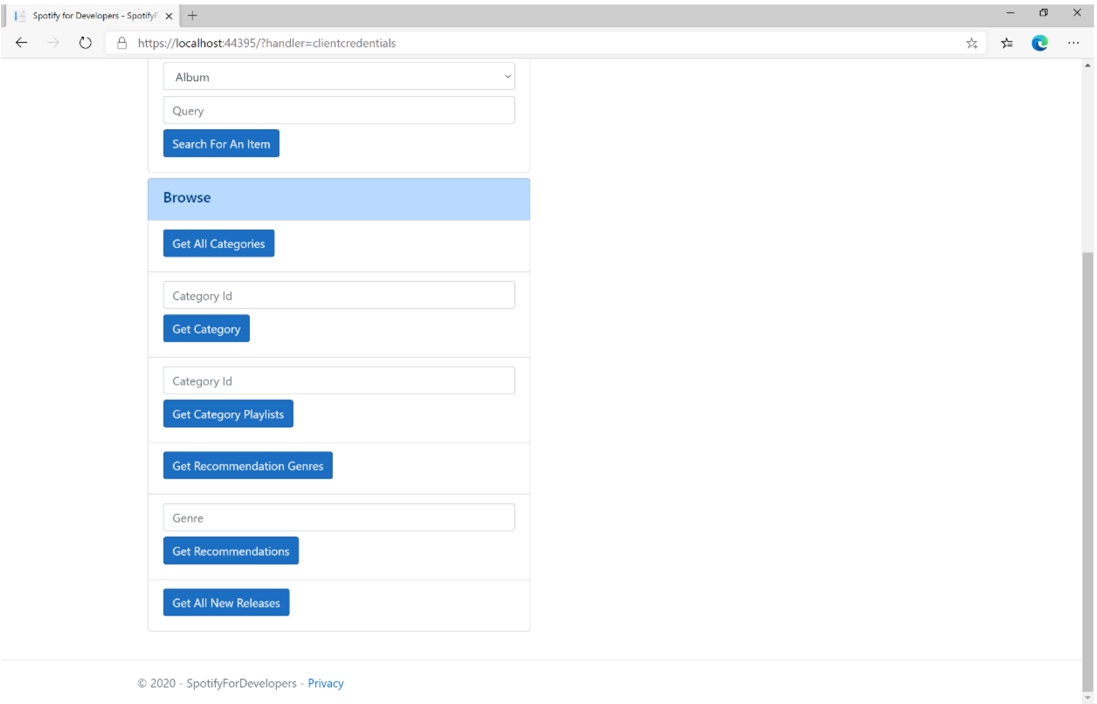
Step 8



Finally, in **Visual Studio 2019** select **IIS Application** to run the **Web Application**

Step 9

Once the **Web Application** is running and you select **Client Credentials Flow Login** and scroll down you should see something like the following:



Spotify for Developers - SpotifyF
https://localhost:44395/?handler=clientcredentials

Album

Query

Search For An Item

Browse

Get All Categories

Category Id

Get Category

Category Id

Get Category Playlists

Get Recommendation Genres

Genre

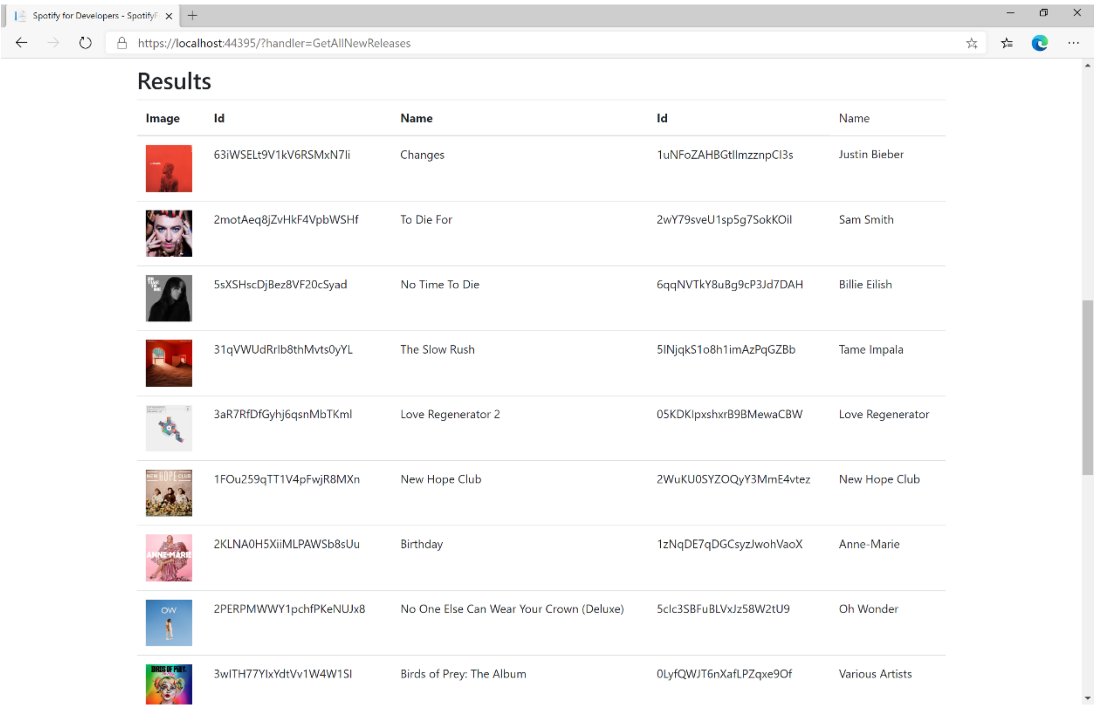
Get Recommendations

Get All New Releases

© 2020 - SpotifyForDevelopers - Privacy

Step 10

You can then select **Get All New Releases** and scroll down to view **Results** like the following:



Spotify for Developers - SpotifyF
https://localhost:44395/?handler=GetAllNewReleases

Results

Image	Id	Name	Id	Name
	63iWSELt9V1kV6RSMxN7li	Changes	1uNFoZAHBGtllmzznpCl3s	Justin Bieber
	2motAeq8jZvHkF4VpbWSHf	To Die For	2wY79sveU1sp5g7SokKOil	Sam Smith
	5sXSHscDjBez8VF20cSyad	No Time To Die	6qqNVTkY8uBg9cP3Jd7DAH	Billie Eilish
	31qVWUdRrlb8thMvts0yYL	The Slow Rush	5INjqkS1o8h1imAzPqGZBb	Tame Impala
	3aR7RfDfGyhj6qsnMbTKml	Love Regenerator 2	05KDKlpxshxrB9BMewaCBW	Love Regenerator
	1FOu259qTT1V4pFwjR8MXn	New Hope Club	2WuKU0SYZOQyY3MmE4vtez	New Hope Club
	2KLNA0H5XiiMLPAWSb8sUu	Birthday	1zNqDE7qDGCsyzJwohVaoX	Anne-Marie
	2PERPMWWY1pchfPKeNUJx8	No One Else Can Wear Your Crown (Deluxe)	5clc3SBFuBLVxJz58W2tU9	Oh Wonder
	3wITH77YlxYdtVv1W4W1SI	Birds of Prey: The Album	0LyfQWJT6nXafLPZqxe9Of	Various Artists

Step 11

You can stop the **web application** in **Visual Studio 2019** by selecting the **Stop debugging** button

Step 12

You can choose to exit **Visual Studio 2019** by selecting the **Close** button in the top right of the **application** as that completes this part of the workshop

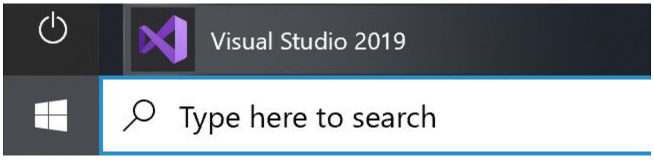
Get All Featured Playlists

Get a list of Spotify featured playlists as shown in the Spotify player's "Browse" tab.

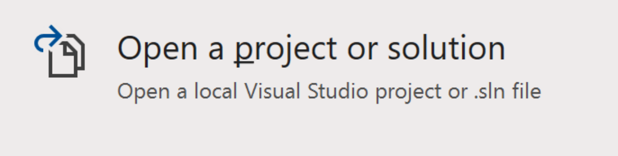
GET https://api.spotify.com/v1/browse/featured-playlists	
Header	
Authorization	Valid Access Token from Spotify Accounts service
Query Parameter	
country	ISO 3166-1 alpha-2 country code e.g. "GB". If omitted globally relevant items returned
locale	Desired language, consisting of ISO 639-1 language code and ISO 3166-1 alpha-2 country code, joined by an underscore e.g. "en_GB". If omitted results returned in American English
timestamp	Timestamp in ISO 8601 format: yyyy-MM-ddTHH:mm:ss. Use this parameter to specify the user's local time to get results tailored for that specific date and time in the day
limit	Maximum number of results to return
offset	Index of first result to return

Header	**Response**
Success	
HTTP Status 200 OK	Playlists Object contains an Array of Simplified Playlist Objects
Error	
Error Code	Error Object

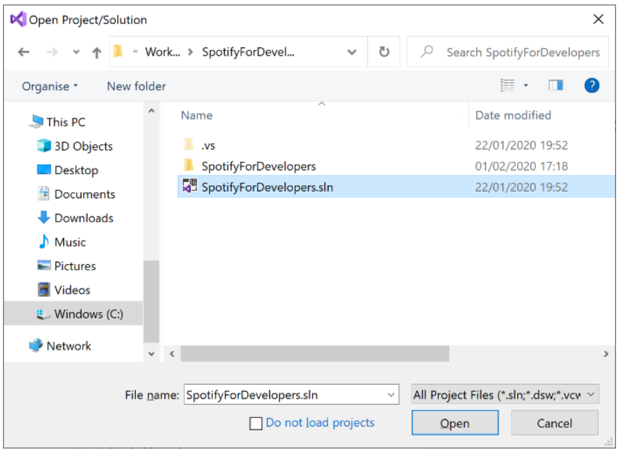
Step 1



If you chose to close **Visual Studio 2019** previously, in **Windows 10** choose **Start**, and then from the **Start Menu** find and select **Visual Studio 2019**


Open a project or solution
Open a local Visual Studio project or .sln file

Once done, from the **Get started** screen for **Visual Studio 2019** select **Open a project or solution**



Name	Date modified
.vs	22/01/2020 19:52
SpotifyForDevelopers	01/02/2020 17:18
SpotifyForDevelopers.sln	22/01/2020 19:52

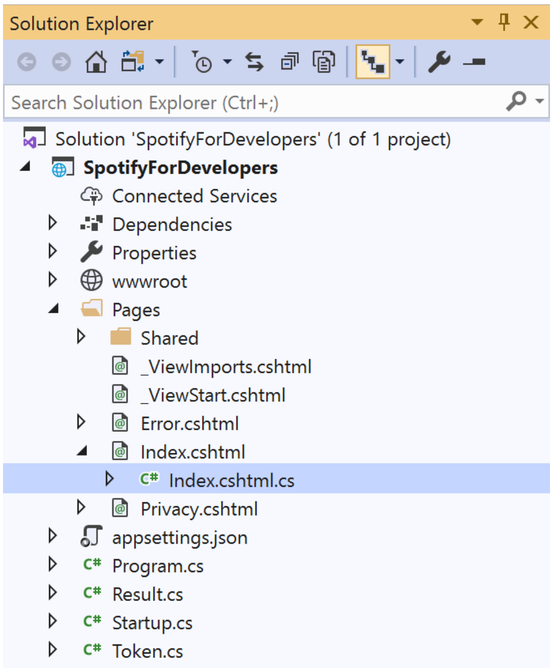
File name: SpotifyForDevelopers.sln

All Project Files (*.sln;*.dsw;*.vcw

□ Do not load projects

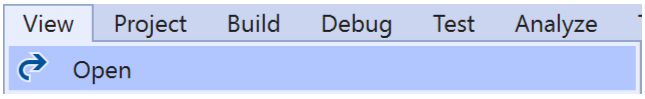
Then locate and select **SpotifyForDevelopers.sln** and select **Open** if you don't have this file already then please follow the previous parts of the workshop including **Getting Started**, **Authorisation Guide** and **Search & Browse**

Step 2



Once opened, in the **Solution Explorer** open the **Pages** section, then open the **Index.cshtml** section and select **Index.cshtml.cs**

Step 3



Then from the **Menu** choose **View** and then **Open**

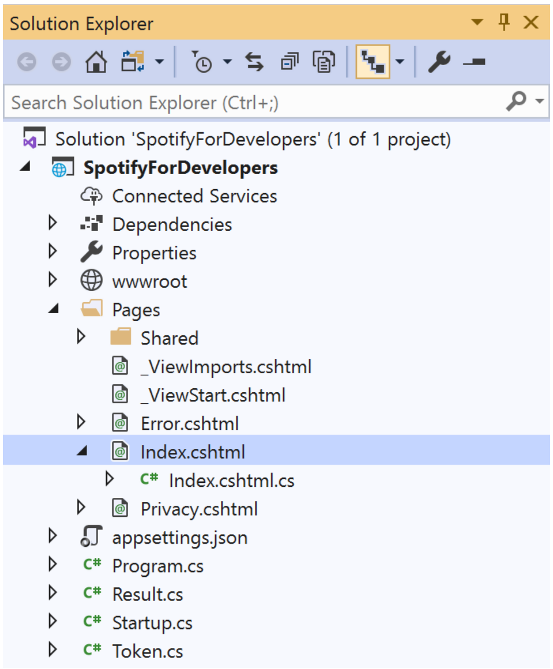
Step 4

In the **Code View** for **Index.cshtml.cs** below the **method** for `public async Task<IActionResult> OnPostGetAllNewReleasesAsync() { ... }` enter the following **method**:

```
public async Task<IActionResult> OnPostGetAllFeaturedPlaylistsAsync()
{
    LoadToken();
    var results = await Api.GetAllFeaturedPlaylistsAsync(country);
    if (results?.Items != null)
    {
        Results = results.Items.Select(result => new Result()
        {
            Id = result.Id,
            Name = result.Name,
            Image = result?.Images?.FirstOrDefault()?.Url
        });
    }
    return Page();
}
```

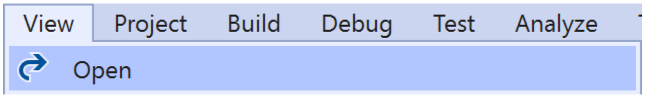
The **method** for `OnPostGetAllFeaturedPlaylistsAsync` is used to get **all featured playlists** by `country` on Spotify and populate the **property** for `Results` accordingly.

Step 5



In the **Solution Explorer** in the **Pages** section select **Index.cshtml**

Step 6



Then from the **Menu** choose **View** and then **Open**

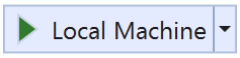
Step 7

Once in the **Code View** for **Index.cshtml** above `<!-- Browse -->` enter the following:

```
<li class="list-group-item">
    <form asp-page-handler="GetAllFeaturedPlaylists" method="post">
        <button class="btn btn-primary mb-2">
            Get All Featured Playlists
        </button>
    </form>
</li>
```

This `form` will **post** to the **method** for `OnPostGetAllFeaturedPlaylistsAsync` and will output to the **Results**.

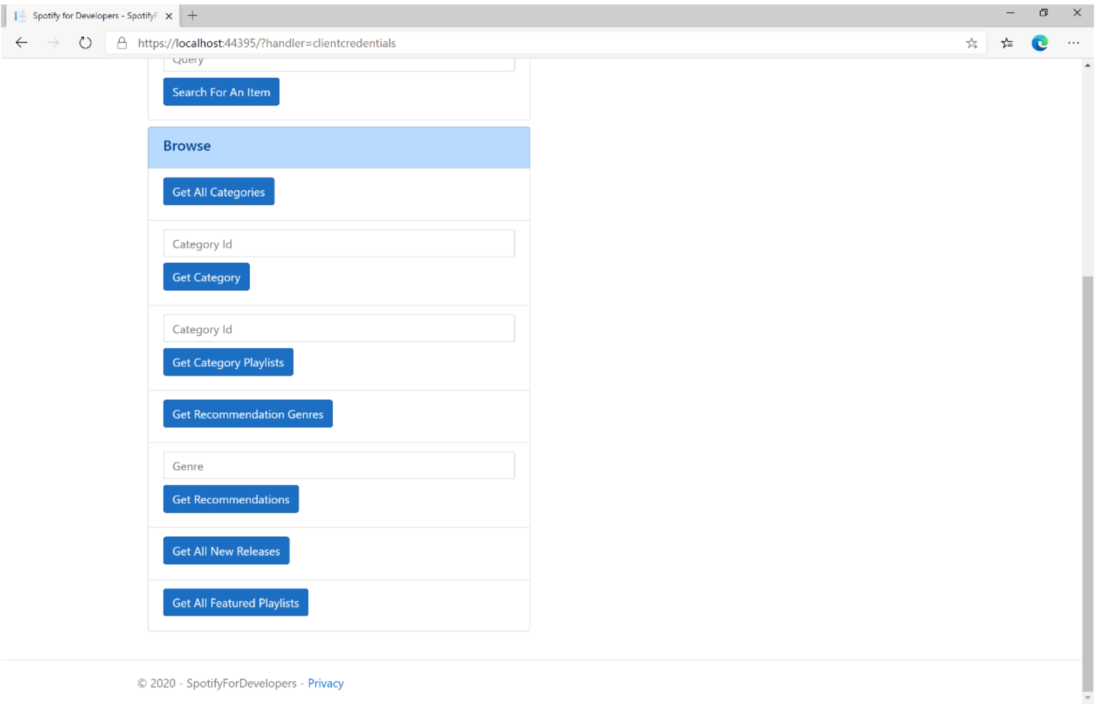
Step 8



Finally, in **Visual Studio 2019** select **IIS Application** to run the **Web Application**

Step 9

Once the **Web Application** is running and you select **Client Credentials Flow Login** and scroll down you should see something like the following:



https://localhost:44395/?handler=clientcredentials

Query

Search For An Item

Browse

Get All Categories

Category Id

Get Category

Category Id

Get Category Playlists

Get Recommendation Genres

Genre

Get Recommendations

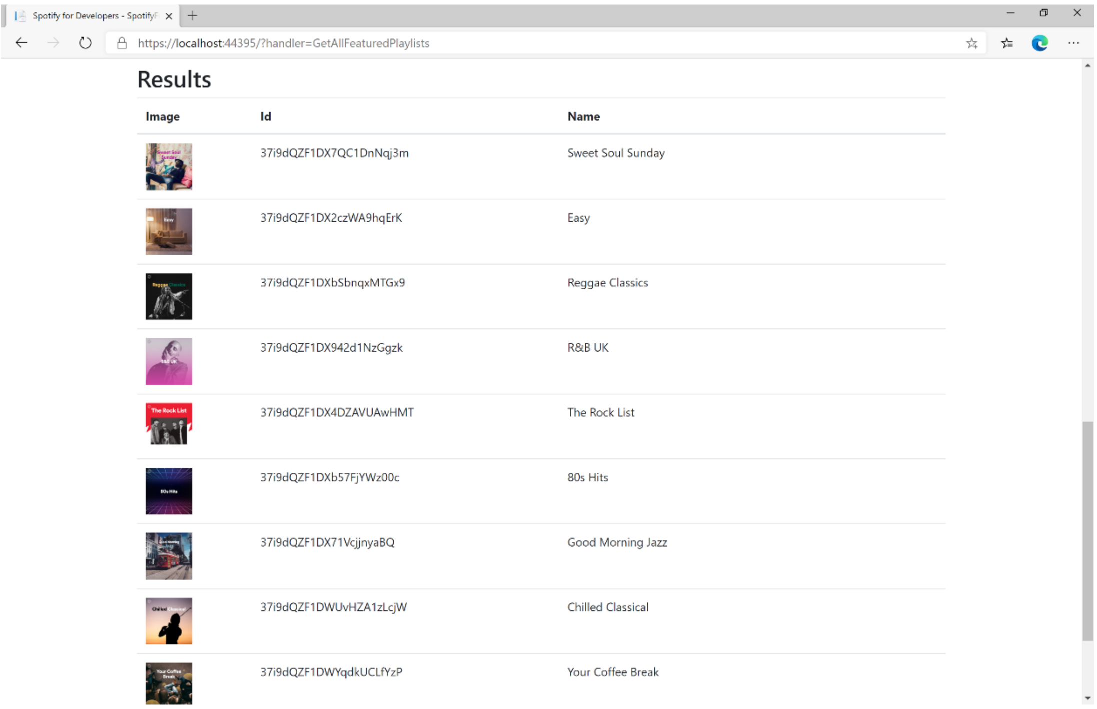
Get All New Releases

Get All Featured Playlists

© 2020 - SpotifyForDevelopers - Privacy

Step 10

You can then select **Get All Featured Playlists** and scroll down to view **Results** like the following:



https://localhost:44395/?handler=GetAllFeaturedPlaylists

Results

Image	Id	Name
	37i9dQZF1DX7QC1DnNqj3m	Sweet Soul Sunday
	37i9dQZF1DX2czWA9hqErK	Easy
	37i9dQZF1DXbSbnqxMTGx9	Reggae Classics
	37i9dQZF1DX942d1NzGgzk	R&B UK
	37i9dQZF1DX4DZAVUAwHMT	The Rock List
	37i9dQZF1DXb57FjYWz00c	80s Hits
	37i9dQZF1DX71VcjjnyaBQ	Good Morning Jazz
	37i9dQZF1DWUvHZA1zLcjW	Chilled Classical
	37i9dQZF1DWYqdkUCLfYzP	Your Coffee Break

Step 11

■

You can stop the **web application** in **Visual Studio 2019** by selecting the **Stop debugging** button

Step 12

×

You can exit **Visual Studio 2019** by selecting the **Close** button in the top right of the **application** as that completes this part of the workshop

Artists, Albums & Tracks

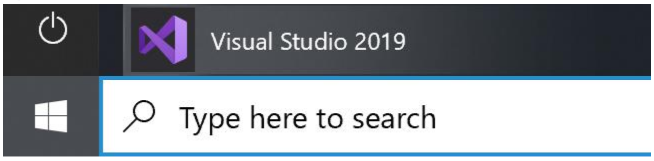
Get Multiple Artists

Get Spotify catalogue information for several artists based on their Spotify Ids.

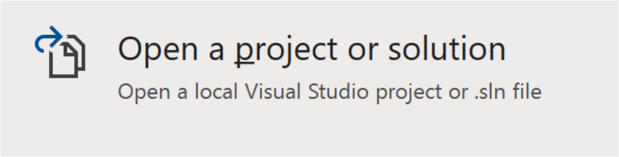
GET https://api.spotify.com/v1/artists	
Header	
Authorization	Valid Access Token from Spotify Accounts service
Query Parameter	
ids	Maximum of 50 comma-separated list of artist Spotify Ids

Header	Response
Success	
HTTP Status 200 OK	Artists as Array of Artist Objects
Error	
Error Code	Error Object

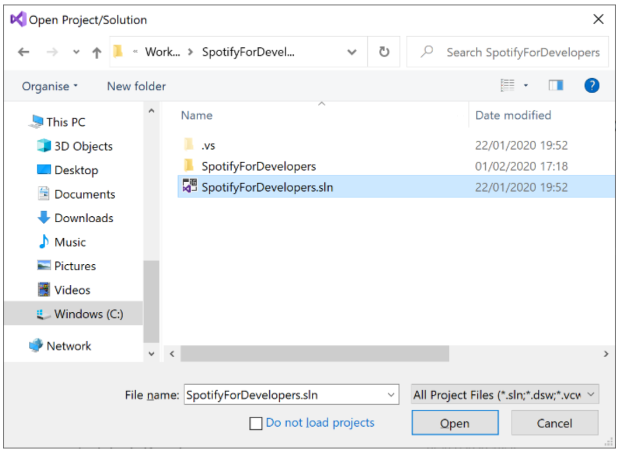
Step 1



In **Windows 10** choose **Start**, and then from the **Start Menu** find and select **Visual Studio 2019**

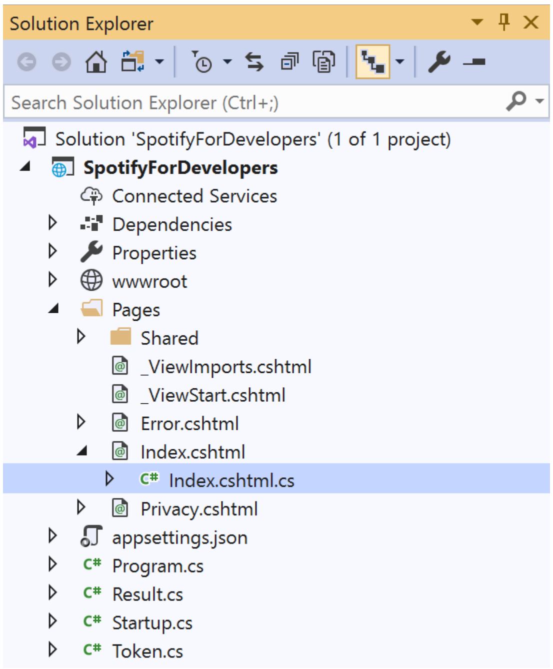

Open a project or solution
Open a local Visual Studio project or .sln file

Once done, from the **Get started** screen for **Visual Studio 2019** select **Open a project or solution**



Then locate and select **SpotifyForDevelopers.sln** and select **Open** if you don't have this file already then please follow the previous parts of the workshop including **Getting Started**, **Authorisation Guide**, **Search & Browse**

Step 2



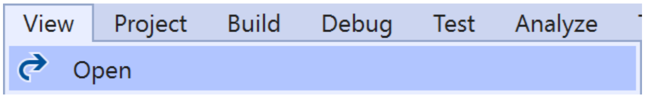
Solution Explorer

Search Solution Explorer (Ctrl+;)

Solution 'SpotifyForDevelopers' (1 of 1 project)
- **SpotifyForDevelopers**
  - Connected Services
  - Dependencies
  - Properties
  - wwwroot
  - Pages
    - Shared
    - _ViewImports.cshtml
    - _ViewStart.cshtml
    - Error.cshtml
    - Index.cshtml
      - Index.cshtml.cs
    - Privacy.cshtml
  - appsettings.json
  - Program.cs
  - Result.cs
  - Startup.cs
  - Token.cs

Once opened, in the **Solution Explorer** open the **Pages** section, then open the **Index.cshtml** section and select **Index.cshtml.cs**

Step 3



Then from the **Menu** choose **View** and then **Open**

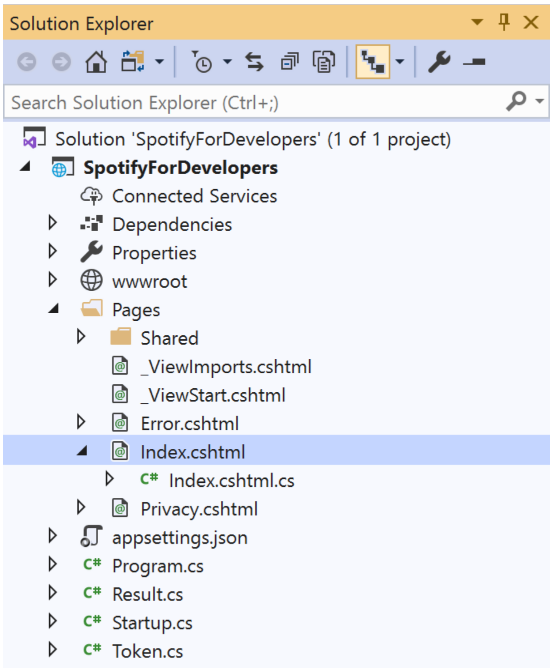
Step 4

In the **Code View** for **Index.cshtml.cs** below the **method** for `public async Task<IActionResult> OnPostGetAllFeaturedPlaylistsAsync() { ... }` enter the following **method**:

```
public async Task<IActionResult> OnPostGetMultipleArtistsAsync(string value)
{
    LoadToken();
    var values = value.Split(",").ToList();
    var results = await Api.GetMultipleArtistsAsync(values);
    if (results != null)
    {
        Results = results.Select(result => new Result()
        {
            Id = result.Id,
            Name = result.Name,
            Image = result?.Images?.FirstOrDefault()?.Url
        });
    }
    return Page();
}
```

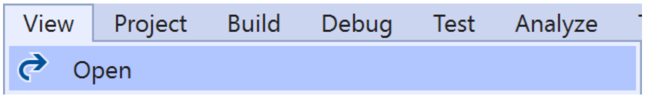
The **method** for `OnPostGetMultipleArtistsAsync` is used to get **artists** by multiple **Artist Id** on Spotify with the `value` and populate the **property** for `Results` accordingly.

Step 5



In the **Solution Explorer** in the **Pages** section select **Index.cshtml**

Step 6



Then from the **Menu** choose **View** and then **Open**

Step 7

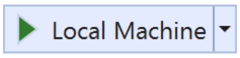
Once in the **Code View** for **Index.cshtml** above `<!-- Spotify Web API App Authorisation -->` enter the following:

```
<ul class="list-group mb-2">
    <li class="list-group-item list-group-item-primary">
        <h5 class="list-group-item-heading">Artists</h5>
    </li>
    <li class="list-group-item">
        <form asp-page-handler="GetMultipleArtists" method="post">
            <input asp-for="Value" placeholder="Artist Ids" class="form-control mb-2" />
            <button class="btn btn-primary mb-2">
                Get Multiple Artists
            </button>
        </form>
    </li>

    <!-- Artists -->
</ul>
```

This `form` will **post** to the **method** for `OnPostGetMultipleArtistsAsync` with the `Value` as the **Artist Ids** and will output to the **Results**.

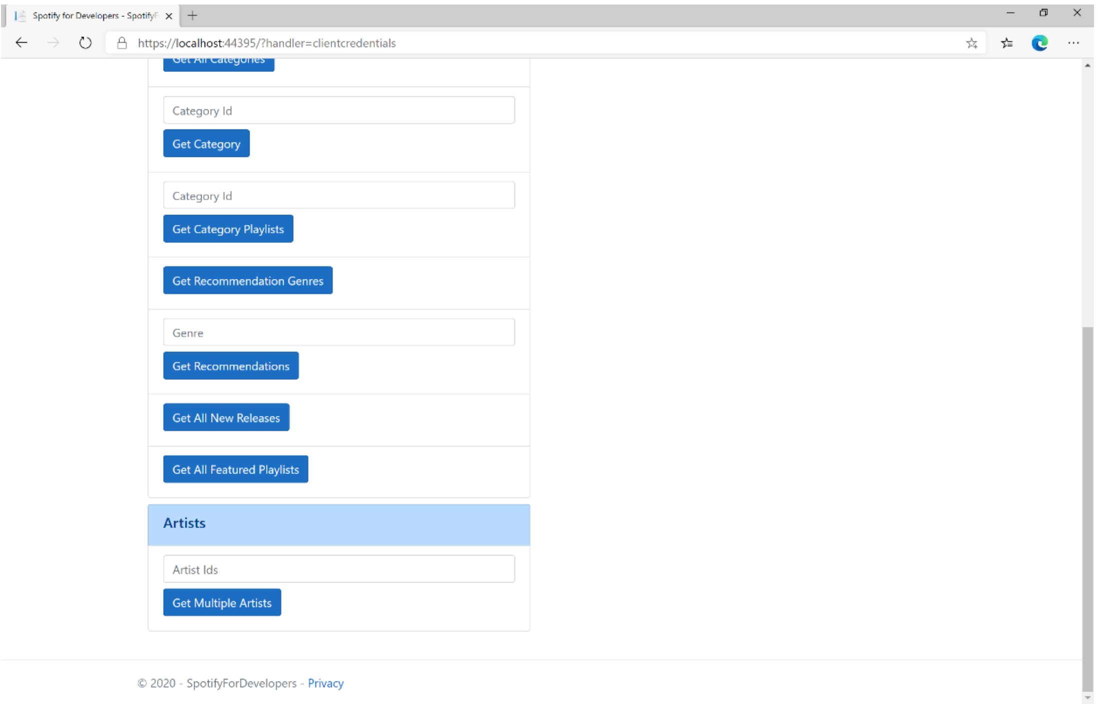
Step 8



Finally, in **Visual Studio 2019** select **IIS Application** to run the **Web Application**

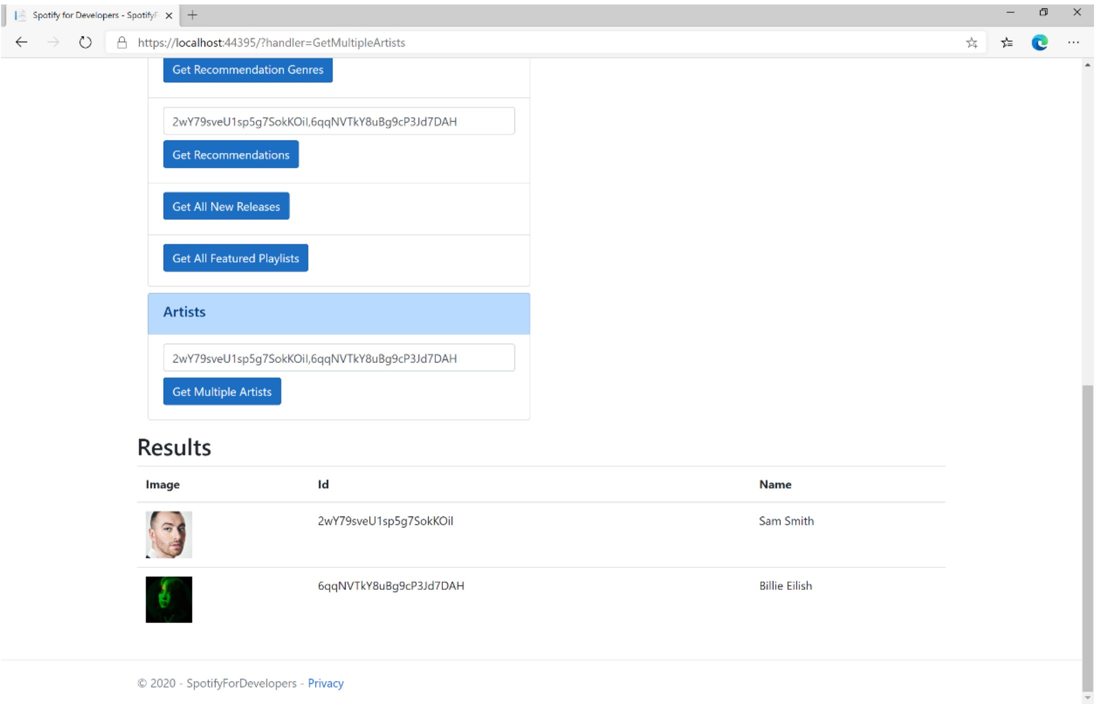
Step 9

Once the **Web Application** is running and you select **Client Credentials Flow Login** and scroll down you should see something like the following:


Spotify for Developers - Spotify
https://localhost:44395/?handler=clientcredentials
Get All Categories
Category Id
Get Category
Category Id
Get Category Playlists
Get Recommendation Genres
Genre
Get Recommendations
Get All New Releases
Get All Featured Playlists
Artists
Artist Ids
Get Multiple Artists
© 2020 - SpotifyForDevelopers - Privacy

Step 10

You can then enter an **Artist Id** or if multiple ones separate them with a **comma** from **Get All New Releases** and select **Get Multiple Artists** and scroll down to view **Results** like the following:


Spotify for Developers - Spotify
https://localhost:44395/?handler=GetMultipleArtists
Get Recommendation Genres
Get Recommendations
Get All New Releases
Get All Featured Playlists
Artists
Get Multiple Artists
© 2020 - SpotifyForDevelopers - Privacy

2wY79sveU1sp5g7SokKOil,6qqNVTkY8uBg9cP3Jd7DAH

2wY79sveU1sp5g7SokKOil,6qqNVTkY8uBg9cP3Jd7DAH

Results

Image	Id	Name
	2wY79sveU1sp5g7SokKOil	Sam Smith
	6qqNVTkY8uBg9cP3Jd7DAH	Billie Eilish

Step 11

■

You can stop the **web application** in **Visual Studio 2019** by selecting the **Stop debugging** button

Step 12

×

You can exit **Visual Studio 2019** by selecting the **Close** button in the top right of the **application** as that completes this part of the workshop

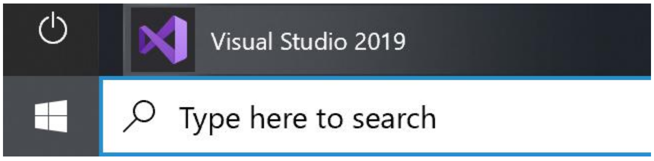
Get an Artist

Get Spotify catalogue information for a single artist identified by their unique Spotify Id.

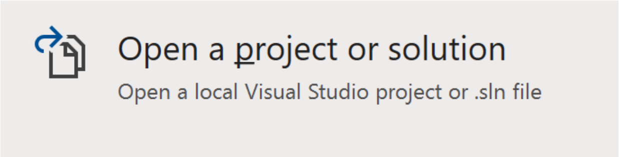
GET https://api.spotify.com/v1/artists/{id}	
Header	
Authorization	Valid Access Token from Spotify Accounts service
Path Parameter	
id	Spotify Id of the artist

Header	Response
Success	
HTTP Status 200 OK	Artist Object
Error	
Error Code	Error Object

Step 1



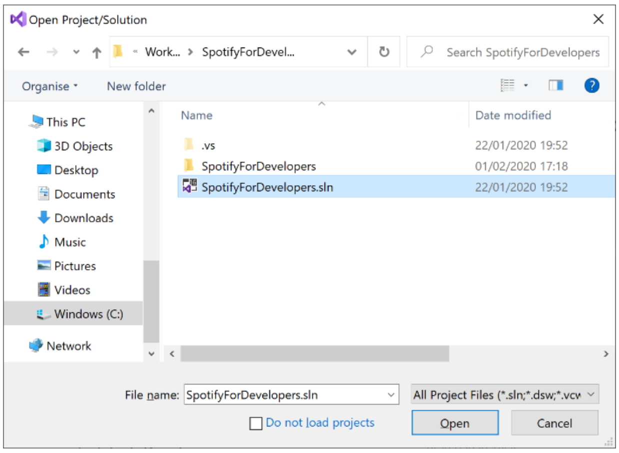
If you chose to close **Visual Studio 2019** previously, in **Windows 10** choose **Start**, and then from the **Start Menu** find and select **Visual Studio 2019**



Open a project or solution

Open a local Visual Studio project or .sln file

Once done, from the **Get started** screen for **Visual Studio 2019** select **Open a project or solution**

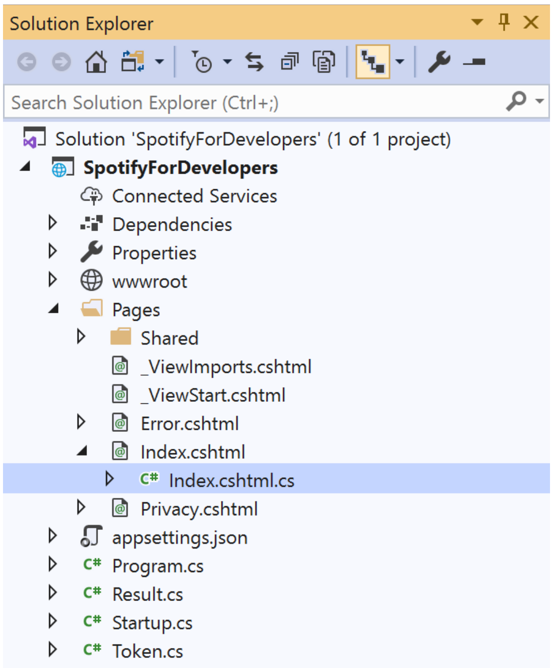


Name	Date modified
.vs	22/01/2020 19:52
SpotifyForDevelopers	01/02/2020 17:18
SpotifyForDevelopers.sln	22/01/2020 19:52

File name: SpotifyForDevelopers.sln

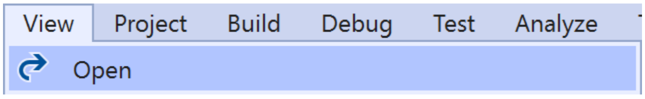
Then locate and select **SpotifyForDevelopers.sln** and select **Open** if you don't have this file already then please follow the previous parts of the workshop including **Getting Started**, **Authorisation Guide**, **Search & Browse** and **Artists, Albums & Tracks**

Step 2



Once opened, in the **Solution Explorer** open the **Pages** section, then open the **Index.cshtml** section and select **Index.cshtml.cs**

Step 3



Then from the **Menu** choose **View** and then **Open**

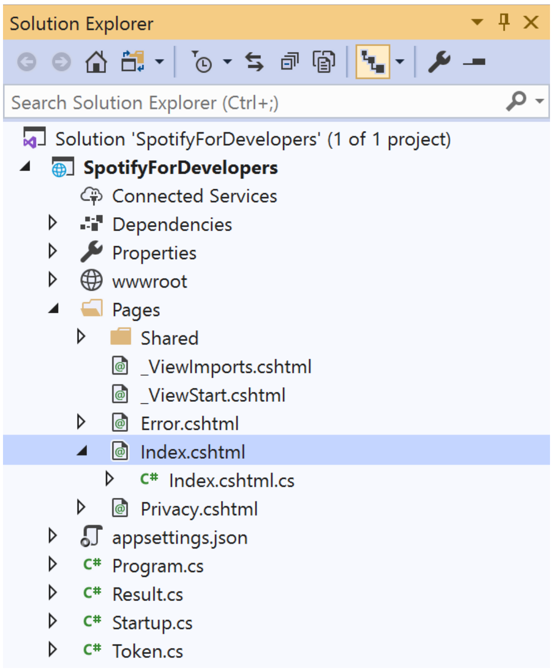
Step 4

In the **Code View** for **Index.cshtml.cs** below the **method** for `public async Task<IActionResult> OnPostGetMultipleArtistsAsync(...) { ... }` enter the following **method**:

```
public async Task<IActionResult> OnPostGetArtistAsync(string value)
{
    LoadToken();
    var result = await Api.GetArtistAsync(value);
    if (result != null)
    {
        Results = new List<Result> { new Result()
        {
            Id = result.Id,
            Name = result.Name,
            Image = result?.Images?.FirstOrDefault()?.Url
        }};
    }
    return Page();
}
```

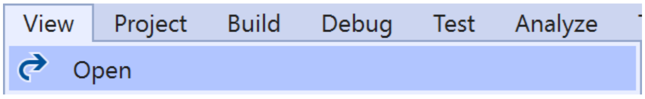
The **method** for `OnPostGetArtistAsync` is used to get an **artist** by **Artist Id** on Spotify with the `value` and populate the **property** for `Results` accordingly.

Step 5



In the **Solution Explorer** in the **Pages** section select **Index.cshtml**

Step 6



Then from the **Menu** choose **View** and then **Open**

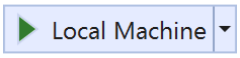
Step 7

Once in the **Code View** for **Index.cshtml** above `<!-- Artists -->` enter the following:

```
<li class="list-group-item">
    <form asp-page-handler="GetArtist" method="post">
        <input asp-for="Value" placeholder="Artist Id" class="form-control mb-2" />
        <button class="btn btn-primary mb-2">
            Get an Artist
        </button>
    </form>
</li>
```

This `form` will **post** to the **method** for `OnPostGetArtistAsync` with the `Value` as the **Artist Id** and will output to the **Results**.

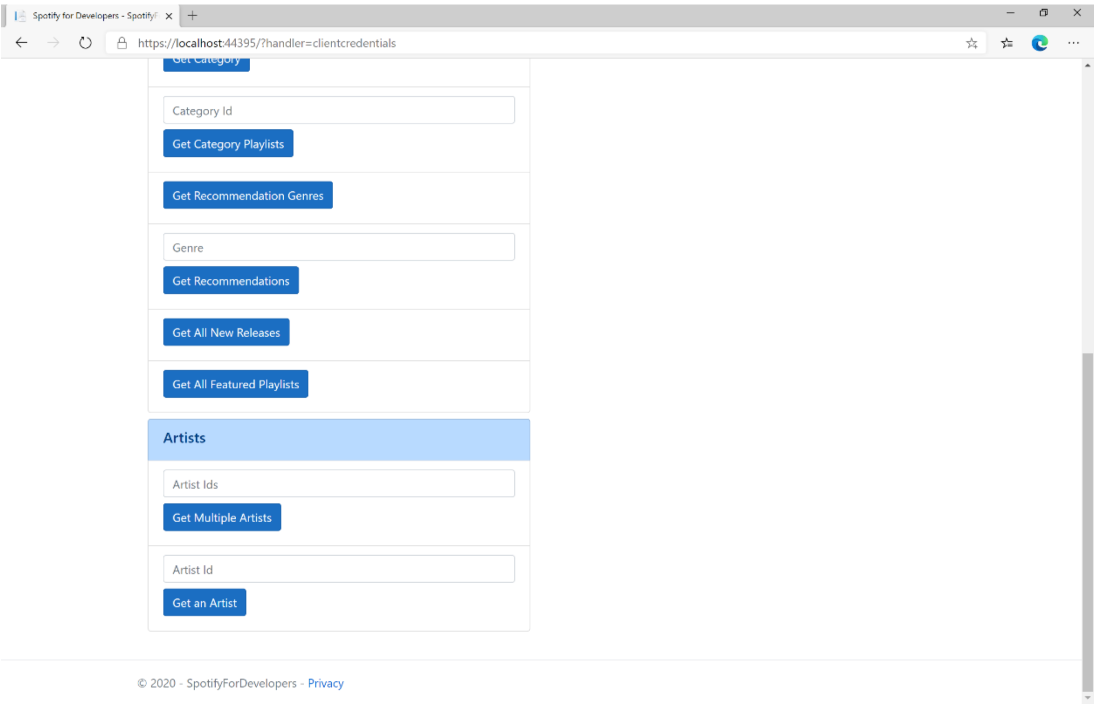
Step 8



Finally, in **Visual Studio 2019** select **IIS Application** to run the **Web Application**

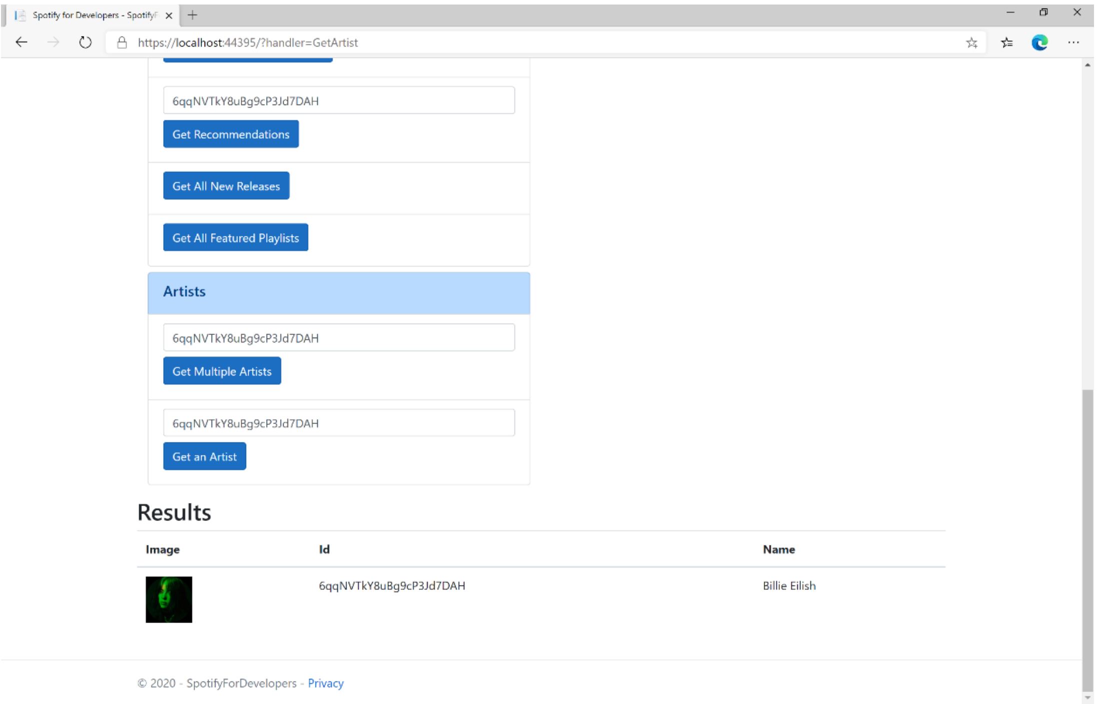
Step 9

Once the **Web Application** is running and you select **Client Credentials Flow Login** and scroll down you should see something like the following:


Spotify for Developers - Spotify
https://localhost:44395/?handler=clientcredentials
Get Category
Category Id
Get Category Playlists
Get Recommendation Genres
Genre
Get Recommendations
Get All New Releases
Get All Featured Playlists
Artists
Artist Ids
Get Multiple Artists
Artist Id
Get an Artist
© 2020 - SpotifyForDevelopers - Privacy

Step 10

You can then enter an **Artist Id** from **Get All New Releases** and select **Get an Artist** and scroll down to view **Results** like the following:


Spotify for Developers - Spotify
https://localhost:44395/?handler=GetArtist
Get Recommendations
Get All New Releases
Get All Featured Playlists
Artists
Get Multiple Artists
Get an Artist
© 2020 - SpotifyForDevelopers - Privacy

6qqNVTkY8uBg9cP3Jd7DAH

6qqNVTkY8uBg9cP3Jd7DAH

6qqNVTkY8uBg9cP3Jd7DAH

Results

Image	Id	Name
	6qqNVTkY8uBg9cP3Jd7DAH	Billie Eilish

Step 11

You can stop the **web application** in **Visual Studio 2019** by selecting the **Stop debugging** button

Step 12

You can choose to exit **Visual Studio 2019** by selecting the **Close** button in the top right of the **application** as that completes this part of the workshop

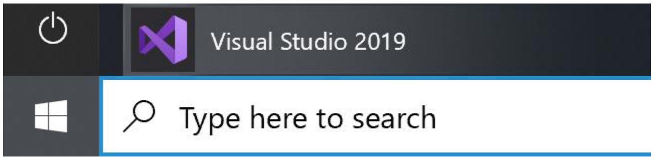
Get an Artist's Albums

Get Spotify catalogue information about an artist’s albums. Optional parameters can be specified to filter and sort the response.

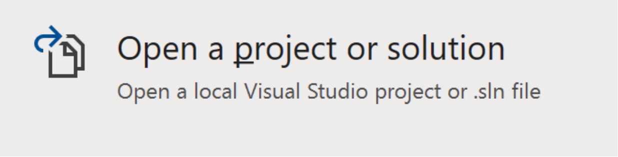
GET https://api.spotify.com/v1/artists/{id}/albums	
Header	
Authorization	Valid Access Token from Spotify Accounts service
Path Parameter	
id	Spotify ID of the artist
Query Parameters	
include_groups	Comma-separated list of keywords that will be used to filter the response from: album, single, appears_on and compilation
market	ISO 3166-1 alpha-2 country code e.g. “GB” or “from_token” if omitted all releases for all countries will be returned which may include duplicates
limit	Maximum number of results to return
offset	Index of first result to return

Header	Response
Success	
HTTP Status 200 OK	Array of Simplified Album Objects wrapped in a Paging Object
Error	
Error Code	Error Object

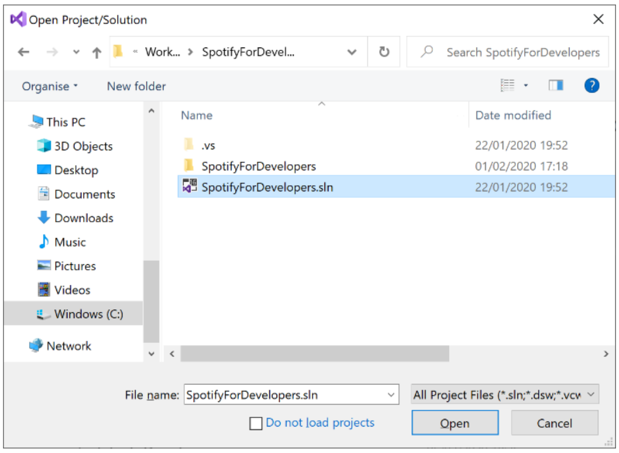
Step 1



If you chose to close **Visual Studio 2019** previously, in **Windows 10** choose **Start**, and then from the **Start Menu** find and select **Visual Studio 2019**


Open a project or solution
Open a local Visual Studio project or .sln file

Once done, from the **Get started** screen for **Visual Studio 2019** select **Open a project or solution**

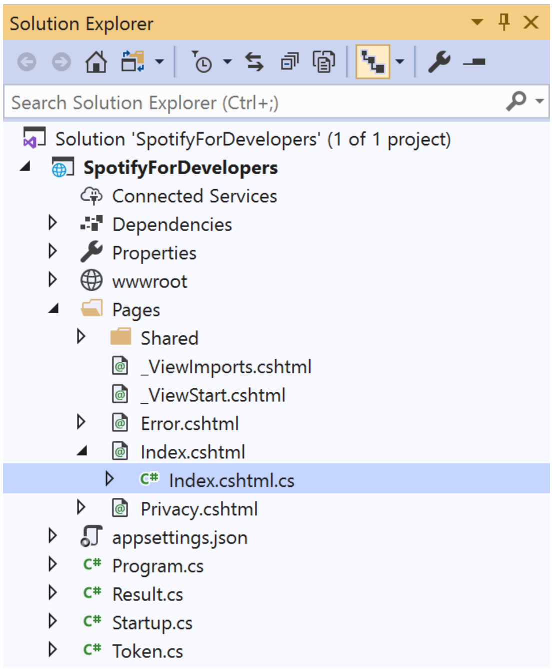


Name	Date modified
.vs	22/01/2020 19:52
SpotifyForDevelopers	01/02/2020 17:18
SpotifyForDevelopers.sln	22/01/2020 19:52

File name: SpotifyForDevelopers.sln

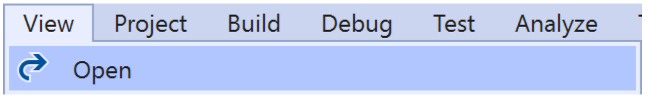
Then locate and select **SpotifyForDevelopers.sln** and select **Open** if you don’t have this file already then please follow the previous parts of the workshop including **Getting Started**, **Authorisation Guide**, **Search & Browse** and **Artists, Albums & Tracks**

Step 2



Once opened, in the **Solution Explorer** open the **Pages** section, then open the **Index.cshtml** section and select **Index.cshtml.cs**

Step 3



Then from the **Menu** choose **View** and then **Open**

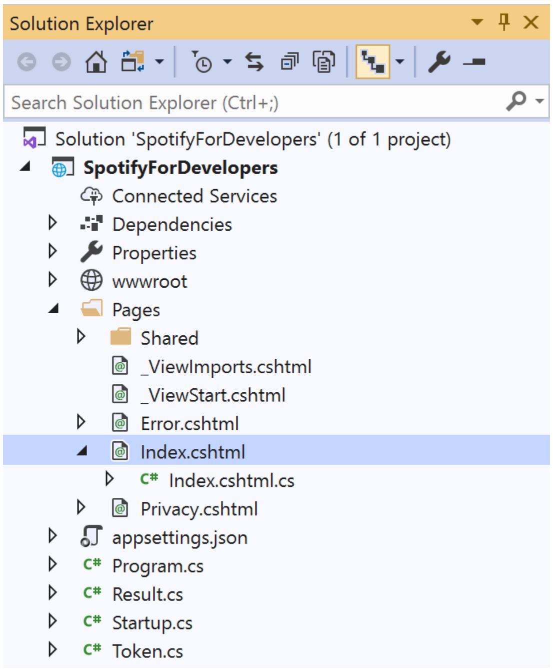
Step 4

In the **Code View** for **Index.cshtml.cs** below the **method** for `public async Task<IActionResult> OnPostGetArtistAsync(...) { ... }` enter the following **method**:

```
public async Task<IActionResult> OnPostGetArtistAlbumsAsync(string value)
{
    LoadToken();
    var results = await Api.GetArtistAlbumsAsync(value);
    if (results?.Items != null)
    {
        Results = results.Items.Select(result => new Result()
        {
            Id = result.Id,
            Name = result.Name,
            Image = result?.Images?.FirstOrDefault()?.Url
        });
    }
    return Page();
}
```

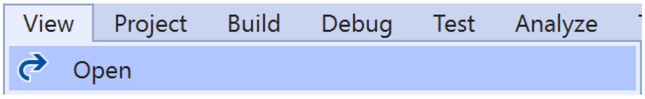
The **method** for `OnPostGetArtistAlbumsAsync` is used to get the **albums** of an **artist** by **Artist Id** on Spotify with the `value` and populate the **property** for `Results` accordingly.

Step 5



In the **Solution Explorer** in the **Pages** section select **Index.cshtml**

Step 6



Then from the **Menu** choose **View** and then **Open**

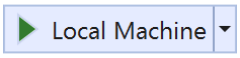
Step 7

Once in the **Code View** for **Index.cshtml** above `<!-- Artists -->` enter the following:

```
<li class="list-group-item">
    <form asp-page-handler="GetArtistAlbums" method="post">
        <input asp-for="Value" placeholder="Artist Id" class="form-control mb-2" />
        <button class="btn btn-primary mb-2">
            Get an Artist's Albums
        </button>
    </form>
</li>
```

This `form` will **post** to the **method** for `OnPostGetArtistAlbumsAsync` with the `Value` as the **Artist Id** and will output to the **Results**.

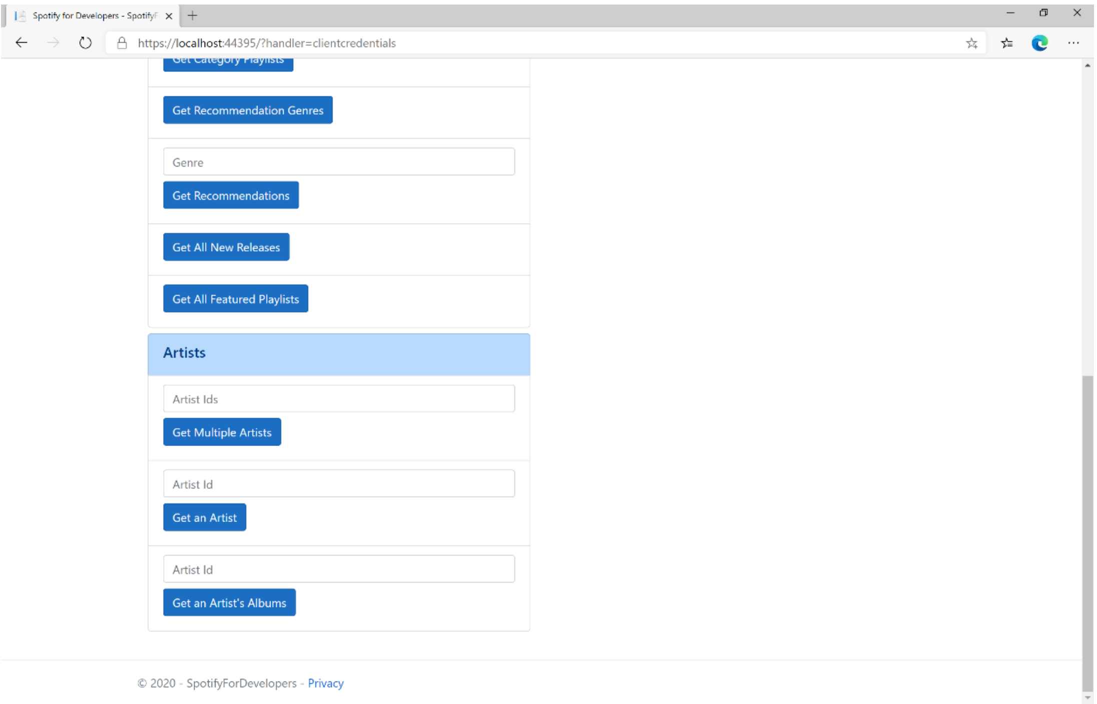
Step 8



Finally, in **Visual Studio 2019** select **IIS Application** to run the **Web Application**

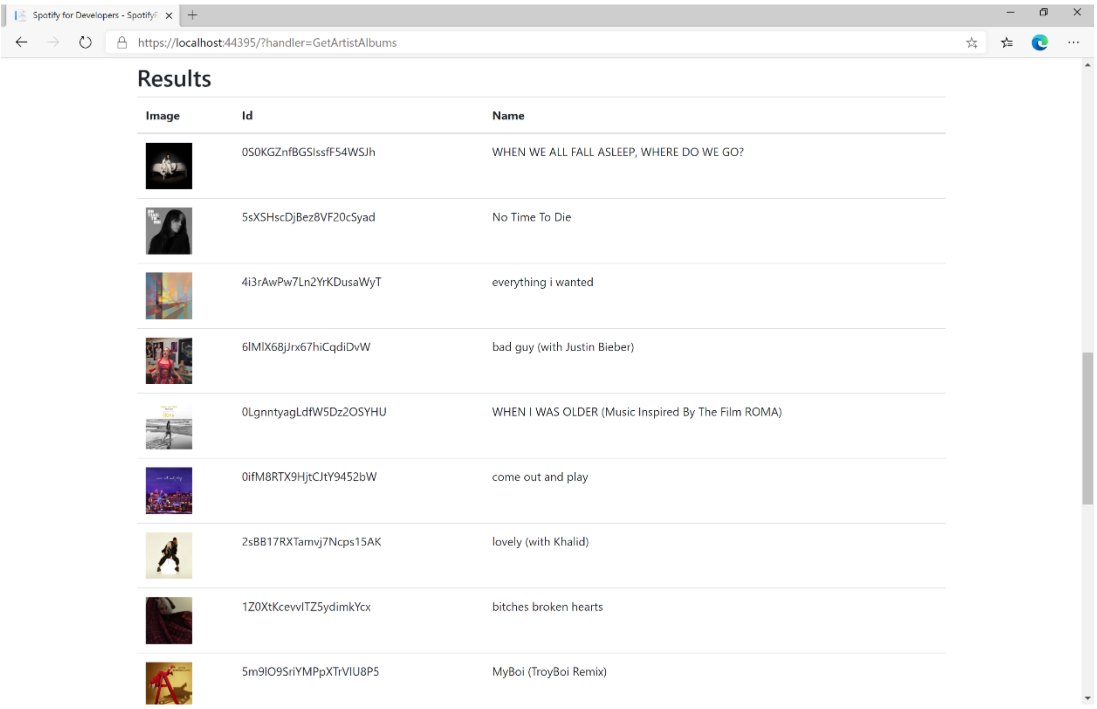
Step 9

Once the **Web Application** is running and you select **Client Credentials Flow Login** and scroll down you should see something like the following:


Spotify for Developers - Spotify
https://localhost:44395/?handler=clientcredentials
Get Category Playlists
Get Recommendation Genres
Genre
Get Recommendations
Get All New Releases
Get All Featured Playlists
Artists
Artist Ids
Get Multiple Artists
Artist Id
Get an Artist
Artist Id
Get an Artist's Albums
© 2020 - SpotifyForDevelopers - Privacy

Step 10

You can then enter an **Artist Id** from **Get All New Releases** and select **Get an Artist's Albums** and scroll down to view **Results** like the following:


Spotify for Developers - Spotify
https://localhost:44395/?handler=GetArtistAlbums

Results

Image	Id	Name
	0S0KGZnfBGSIssfF54WSJh	WHEN WE ALL FALL ASLEEP, WHERE DO WE GO?
	5sXSHscDjBez8VF20cSyad	No Time To Die
	4i3rAwPw7Ln2YrKDusaWyT	everything i wanted
	6lMIX68jJrx67hiCqdiDvW	bad guy (with Justin Bieber)
	0LgnntyagLdfW5Dz2OSYHU	WHEN I WAS OLDER (Music Inspired By The Film ROMA)
	0ifM8RTX9HjtCJtY9452bW	come out and play
	2sBB17RXTamvj7Ncps15AK	lovely (with Khalid)
	1Z0XtKcevvITZ5ydimkYcx	bitches broken hearts
	5m9IO9SriYMPpXTrVIU8P5	MyBoi (TroyBoi Remix)

Step 11

You can stop the **web application** in **Visual Studio 2019** by selecting the **Stop debugging** button

Step 12

You can choose to exit **Visual Studio 2019** by selecting the **Close** button in the top right of the **application** as that completes this part of the workshop

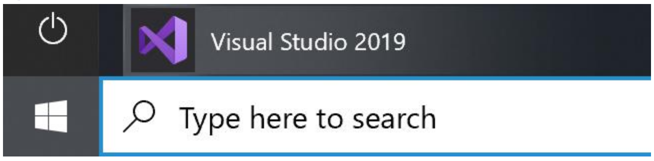
Get an Artist's Top Tracks

Get Spotify catalogue information about an artist's top tracks by country.

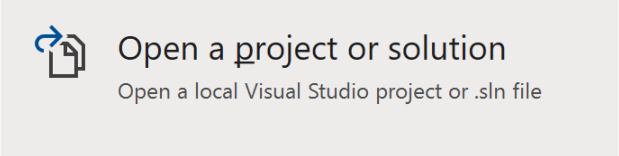
GET https://api.spotify.com/v1/artists/{id}/top-tracks	
Header	
Authorization	Valid Access Token from Spotify Accounts service
Path Parameter	
id	Spotify Id of the artist
Query Parameters	
market	ISO 3166-1 alpha-2 country code e.g. "GB" or "from_token"

Header	**Response**
Success	
HTTP Status 200 OK	Tracks as Array of up to 10 Track Objects
Error	
Error Code	Error Object

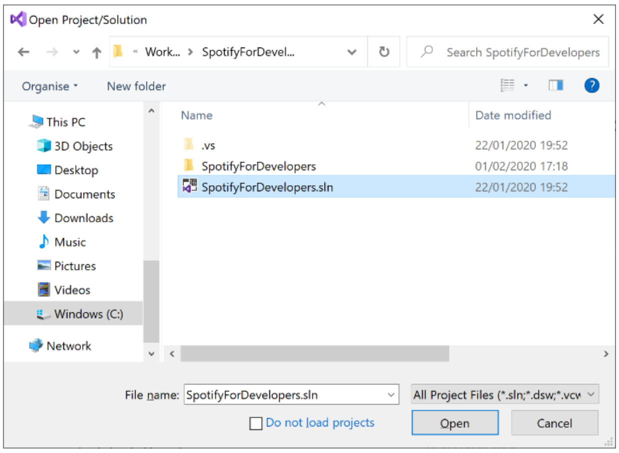
Step 1



If you chose to close **Visual Studio 2019** previously, in **Windows 10** choose **Start**, and then from the **Start Menu** find and select **Visual Studio 2019**


Open a project or solution
Open a local Visual Studio project or .sln file

Once done, from the **Get started** screen for **Visual Studio 2019** select **Open a project or solution**



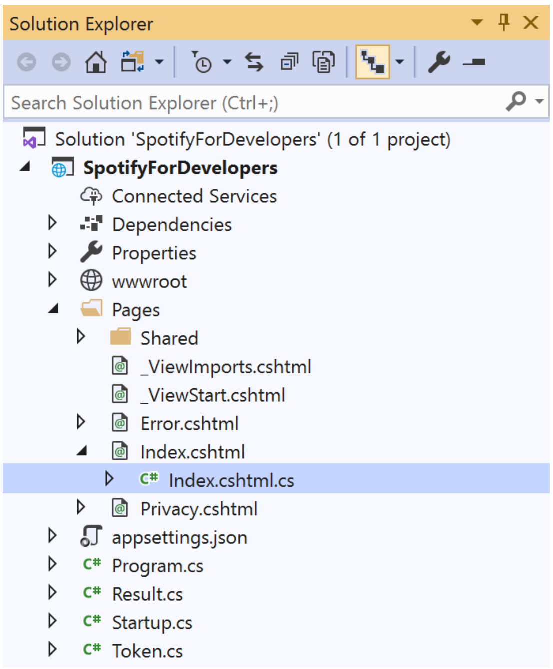
Name	Date modified
.vs	22/01/2020 19:52
SpotifyForDevelopers	01/02/2020 17:18
SpotifyForDevelopers.sln	22/01/2020 19:52

File name: SpotifyForDevelopers.sln

All Project Files (*.sln;*.dsw;*.vcw

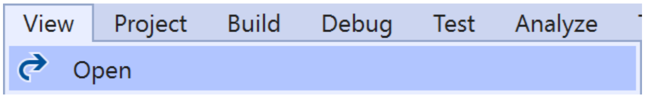
Then locate and select **SpotifyForDevelopers.sln** and select **Open** if you don't have this file already then please follow the previous parts of the workshop including **Getting Started**, **Authorisation Guide**, **Search & Browse** and **Artists, Albums & Tracks**

Step 2



Once opened, in the **Solution Explorer** open the **Pages** section, then open the **Index.cshtml** section and select **Index.cshtml.cs**

Step 3



Then from the **Menu** choose **View** and then **Open**

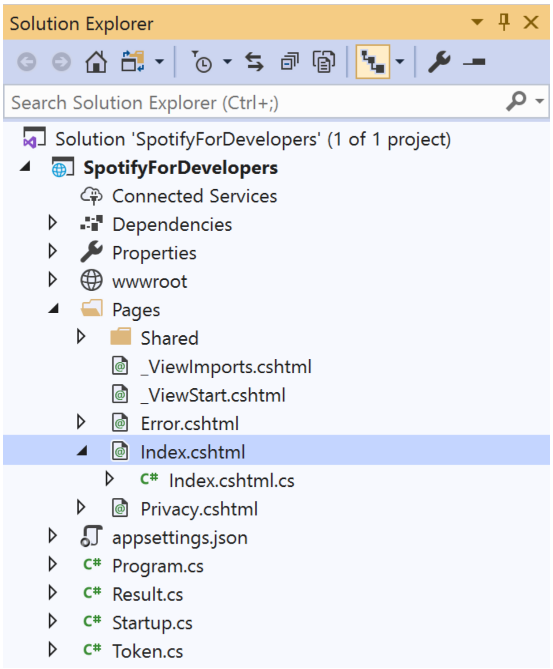
Step 4

In the **Code View** for **Index.cshtml.cs** below the **method** for `public async Task<IActionResult> OnPostGetArtistAlbumsAsync(...) { ... }` enter the following **method**:

```
public async Task<IActionResult> OnPostGetArtistTopTracksAsync(string value)
{
    LoadToken();
    var results = await Api.GetArtistTopTracksAsync(value, country);
    if (results != null)
    {
        Results = results.Select(result => new Result()
        {
            Id = result.Id,
            Name = result.Name,
            Image = result?.Album?.Images?.FirstOrDefault()?.Url,
            Inner = new Result()
            {
                Id = result?.Album?.Id,
                Name = result?.Album?.Name
            }
        });
    }
    return Page();
}
```

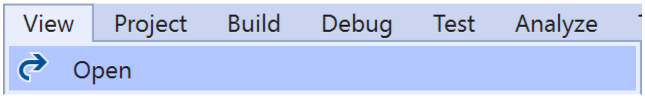
The **method** for `OnPostGetArtistTopTracksAsync` is used to get the top **tracks** of an **artist** by **Artist Id** on Spotify with the `value` and populate the **property** for `Results` accordingly.

Step 5



In the **Solution Explorer** in the **Pages** section select **Index.cshtml**

Step 6



Then from the **Menu** choose **View** and then **Open**

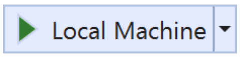
Step 7

Once in the **Code View** for **Index.cshtml** above `<!-- Artists -->` enter the following:

```
<li class="list-group-item">
    <form asp-page-handler="GetArtistTopTracks" method="post">
        <input asp-for="Value" placeholder="Artist Id" class="form-control mb-2" />
        <button class="btn btn-primary mb-2">
            Get an Artist's Top Tracks
        </button>
    </form>
</li>
```

This `form` will **post** to the **method** for `OnPostGetArtistTopTracksAsync` with the `Value` as the **Artist Id** and will output to the **Results**.

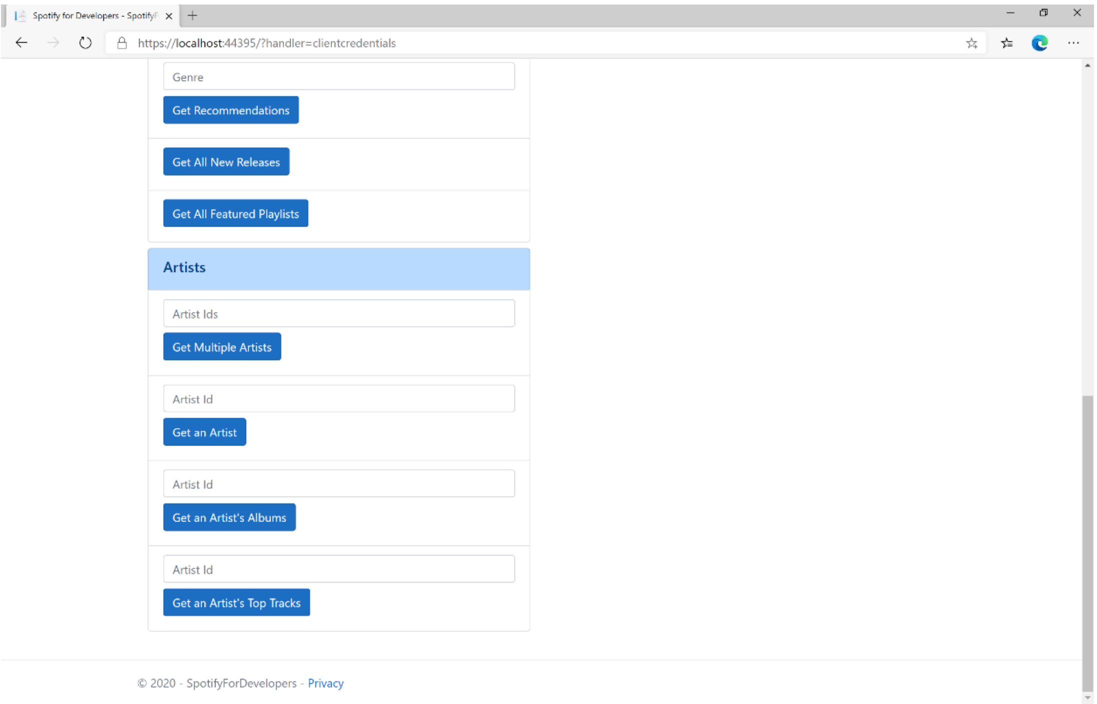
Step 8



Finally, in **Visual Studio 2019** select **IIS Application** to run the **Web Application**

Step 9

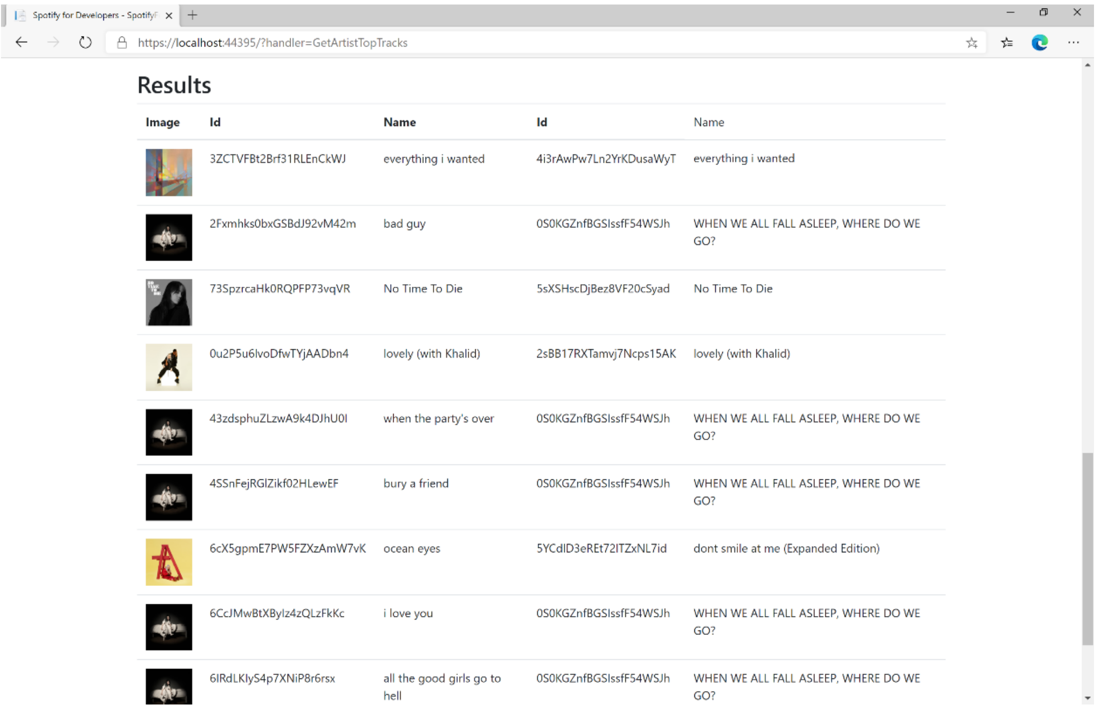
Once the **Web Application** is running and you select **Client Credentials Flow Login** and scroll down you should see something like the following:



Spotify for Developers - Spotify
https://localhost:44395/?handler=clientcredentials
Genre
Get Recommendations
Get All New Releases
Get All Featured Playlists
Artists
Artist Ids
Get Multiple Artists
Artist Id
Get an Artist
Artist Id
Get an Artist's Albums
Artist Id
Get an Artist's Top Tracks
© 2020 - SpotifyForDevelopers - Privacy

Step 10

You can then enter an **Artist Id** from **Get All New Releases** and select **Get an Artist's Top Tracks** and scroll down to view **Results** like the following:



Spotify for Developers - Spotify
https://localhost:44395/?handler=GetArtistTopTracks

Results

Image	Id	Name	Id	Name
	3ZCTVFBt2Brf31RLEnCkWJ	everything i wanted	4i3rAwPw7Ln2YrKDusaWyT	everything i wanted
	2FxmhksObxGSBdJ92vM42m	bad guy	0S0KGZnfBGSIssfF54WSJh	WHEN WE ALL FALL ASLEEP, WHERE DO WE GO?
	73SpzrcaHk0RQPFP73vqVR	No Time To Die	5sXSHscDjBez8VF20cSyad	No Time To Die
	0u2P5u6lvoDfwTYjAADbn4	lovely (with Khalid)	2sBB17RXTamvj7Ncps15AK	lovely (with Khalid)
	43zdsphuZLzwA9k4DJhU0I	when the party's over	0S0KGZnfBGSIssfF54WSJh	WHEN WE ALL FALL ASLEEP, WHERE DO WE GO?
	4SSnFejRGlZikf02HLewEF	bury a friend	0S0KGZnfBGSIssfF54WSJh	WHEN WE ALL FALL ASLEEP, WHERE DO WE GO?
	6cX5gpmE7PW5FZXzAmW7vK	ocean eyes	5YCdlD3eREt72ITZxNL7id	dont smile at me (Expanded Edition)
	6CcJMwBtXBylz4zQLzFkKc	i love you	0S0KGZnfBGSIssfF54WSJh	WHEN WE ALL FALL ASLEEP, WHERE DO WE GO?
	6IRdLKlyS4p7XNiP8r6rsx	all the good girls go to hell	0S0KGZnfBGSIssfF54WSJh	WHEN WE ALL FALL ASLEEP, WHERE DO WE GO?

Step 11

You can stop the **web application** in **Visual Studio 2019** by selecting the **Stop debugging** button

Step 12

You can choose to exit **Visual Studio 2019** by selecting the **Close** button in the top right of the **application** as that completes this part of the workshop

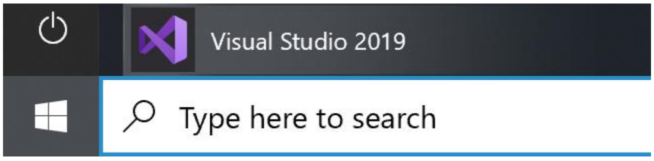
Get an Artist's Related Artists

Get Spotify catalogue information about artists like a given artist. Similarity is based on analysis of the Spotify community's listening history.

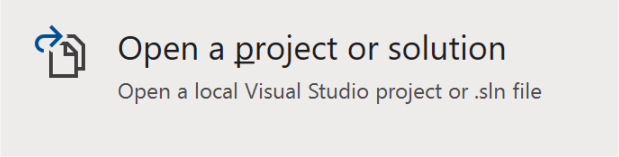
GET https://api.spotify.com/v1/artists/{id}/top-tracks	
Header	
Authorization	Valid Access Token from Spotify Accounts service
Path Parameter	
id	Spotify Id of the artist

Header	Response
Success	
HTTP Status 200 OK	Artists as Array of up to 20 Artist Objects
Error	
Error Code	Error Object

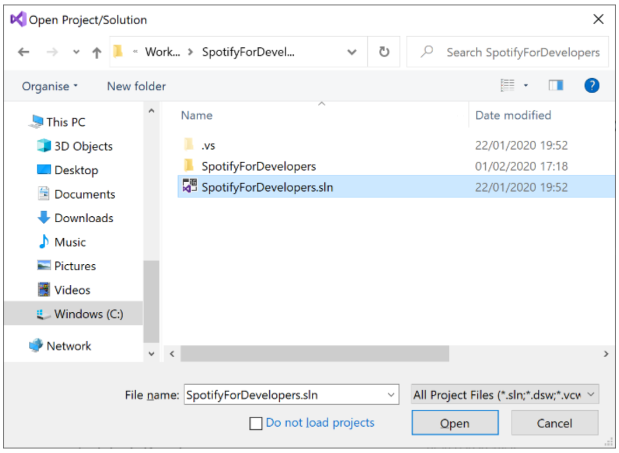
Step 1



If you chose to close **Visual Studio 2019** previously, in **Windows 10** choose **Start**, and then from the **Start Menu** find and select **Visual Studio 2019**


Open a project or solution
Open a local Visual Studio project or .sln file

Once done, from the **Get started** screen for **Visual Studio 2019** select **Open a project or solution**



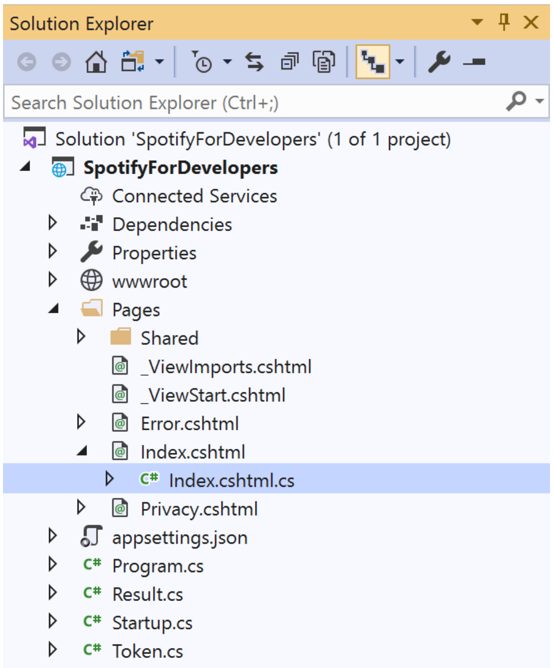
Name	Date modified
.vs	22/01/2020 19:52
SpotifyForDevelopers	01/02/2020 17:18
SpotifyForDevelopers.sln	22/01/2020 19:52

File name: SpotifyForDevelopers.sln

All Project Files (*.sln;*.dsw;*.vcw

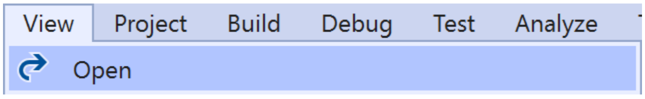
Then locate and select **SpotifyForDevelopers.sln** and select **Open** if you don't have this file already then please follow the previous parts of the workshop including **Getting Started**, **Authorisation Guide**, **Search & Browse** and **Artists, Albums & Tracks**

Step 2



Once opened, in the **Solution Explorer** open the **Pages** section, then open the **Index.cshtml** section and select **Index.cshtml.cs**

Step 3



Then from the **Menu** choose **View** and then **Open**

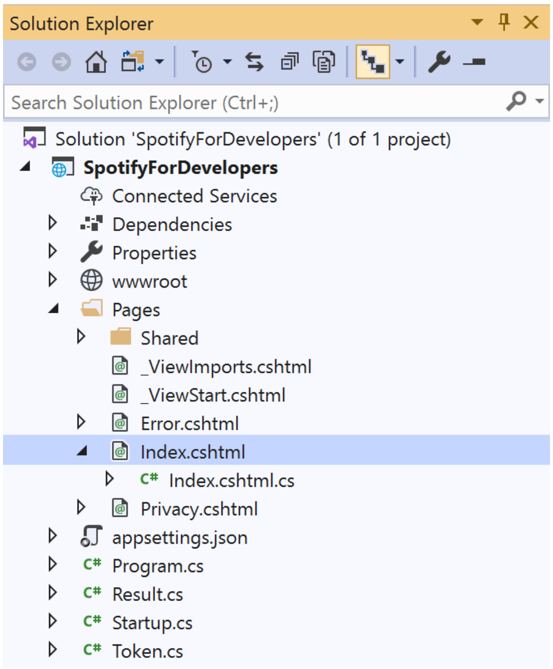
Step 4

In the **Code View** for **Index.cshtml.cs** below the **method** for `public async Task<IActionResult> OnPostGetArtistTopTracksAsync(...) { ... }` enter the following **method**:

```
public async Task<IActionResult> OnPostGetArtistRelatedArtistsAsync(string value)
{
    LoadToken();
    var results = await Api.GetArtistRelatedArtistsAsync(value);
    if (results != null)
    {
        Results = results.Select(result => new Result()
        {
            Id = result.Id,
            Name = result.Name,
            Image = result?.Images?.FirstOrDefault()?.Url
        });
    }
    return Page();
}
```

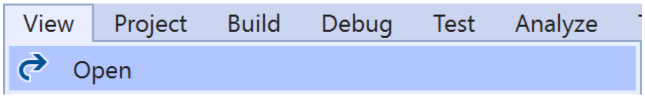
The **method** for `OnPostGetArtistRelatedArtistsAsync` is used to get the related **artists** of an **artist** by **Artist Id** on Spotify with the `value` and populate the **property** for `Results` accordingly.

Step 5



In the **Solution Explorer** in the **Pages** section select **Index.cshtml**

Step 6



Then from the **Menu** choose **View** and then **Open**

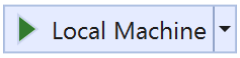
Step 7

Once in the **Code View** for **Index.cshtml** above `<!-- Artists -->` enter the following:

```
<li class="list-group-item">
    <form asp-page-handler="GetArtistRelatedArtists" method="post">
        <input asp-for="Value" placeholder="Artist Id" class="form-control mb-2" />
        <button class="btn btn-primary mb-2">
            Get an Artist's Related Artists
        </button>
    </form>
</li>
```

This `form` will **post** to the **method** for `OnPostGetArtistRelatedArtistsAsync` with the `Value` as the **Artist Id** and will output to the **Results**.

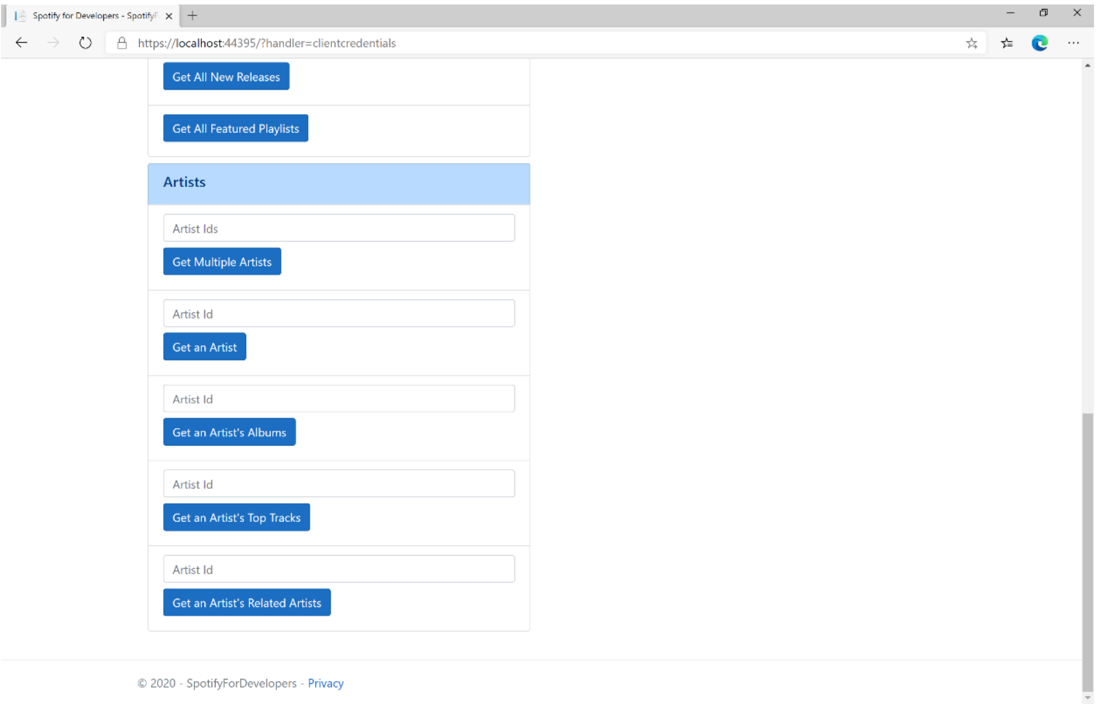
Step 8



Finally, in **Visual Studio 2019** select **IIS Application** to run the **Web Application**

Step 9

Once the **Web Application** is running and you select **Client Credentials Flow Login** and scroll down you should see something like the following:

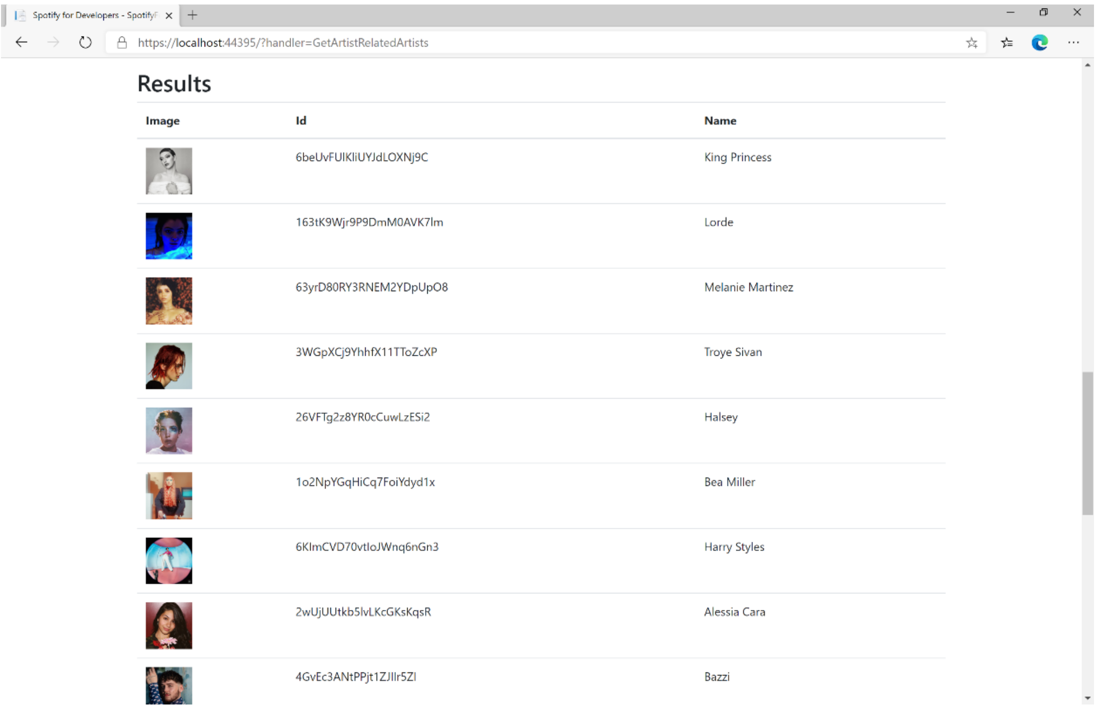


https://localhost:44395/?handler=clientcredentials

© 2020 - SpotifyForDevelopers - Privacy

Step 10

You can then enter an **Artist Id** from **Get All New Releases** and select **Get an Artist's Related Artists** and scroll down to view **Results** like the following:



https://localhost:44395/?handler=GetArtistRelatedArtists

Results

Image	Id	Name
	6beUvFUIKliUYJdLOXNj9C	King Princess
	163tK9Wjr9P9DmM0AVK7lm	Lorde
	63yrD80RY3RNEM2YDpUpO8	Melanie Martinez
	3WGpXCj9YhhfX11TToZcXP	Troye Sivan
	26VFTg2z8YR0cCuwLzESi2	Halsey
	1o2NpYGqHiCq7FoiYdyd1x	Bea Miller
	6KImCVD70vtloJWnq6nGn3	Harry Styles
	2wUjUUtkb5lvLKcGKsKqsR	Alessia Cara
	4GvEc3ANtPPjt1ZJllr5ZI	Bazzi

Step 11

You can stop the **web application** in **Visual Studio 2019** by selecting the **Stop debugging** button

Step 12

You can choose to exit **Visual Studio 2019** by selecting the **Close** button in the top right of the **application** as that completes this part of the workshop

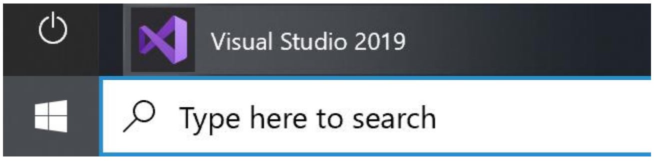
Get Multiple Albums

Get Spotify catalogue information for several albums based on their Spotify Ids.

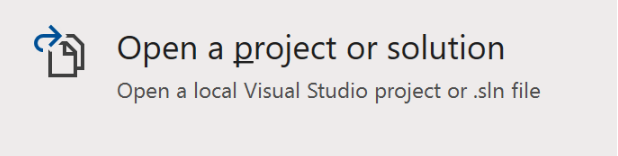
GET https://api.spotify.com/v1/albums	
Header	
Authorization	Valid Access Token from Spotify Accounts service
Query Parameter	
ids	Maximum of 20 comma-separated list of album Spotify Ids
market	ISO 3166-1 alpha-2 country code or "from_token" Provide to apply Track Relinking.

Header	**Response**
Success	
HTTP Status 200 OK	Albums as Array of Album Objects
Error	
Error Code	Error Object

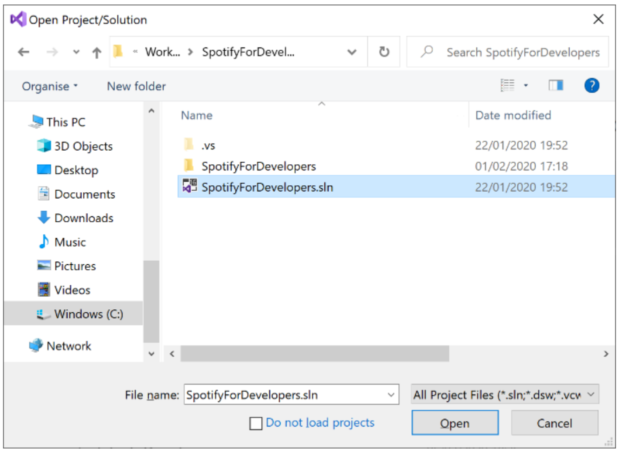
Step 1



If you chose to close **Visual Studio 2019** previously, in **Windows 10** choose **Start**, and then from the **Start Menu** find and select **Visual Studio 2019**


Open a project or solution
Open a local Visual Studio project or .sln file

Once done, from the **Get started** screen for **Visual Studio 2019** select **Open a project or solution**



Name	Date modified
.vs	22/01/2020 19:52
SpotifyForDevelopers	01/02/2020 17:18
SpotifyForDevelopers.sln	22/01/2020 19:52

File name: SpotifyForDevelopers.sln

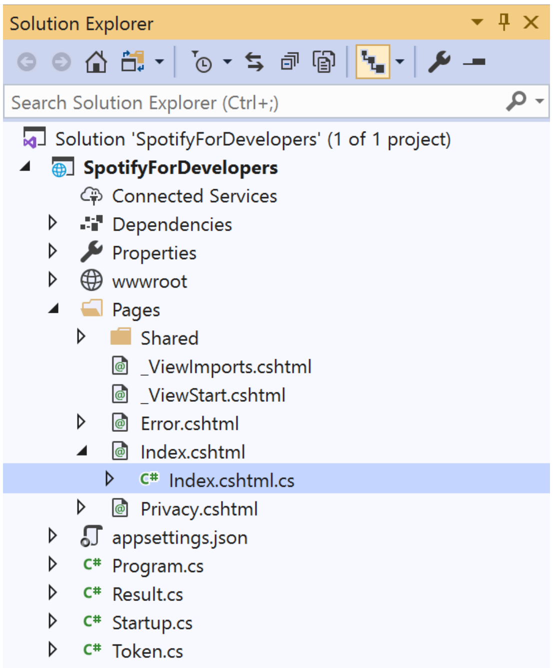
All Project Files (*.sln;*.dsw;*.vcw

□ Do not load projects

Open Cancel

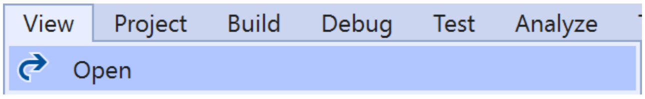
Then locate and select **SpotifyForDevelopers.sln** and select **Open** if you don't have this file already then please follow the previous parts of the workshop including **Getting Started**, **Authorisation Guide**, **Search & Browse** and **Artists, Albums & Tracks**

Step 2



Once opened, in the **Solution Explorer** open the **Pages** section, then open the **Index.cshtml** section and select **Index.cshtml.cs**

Step 3



Then from the **Menu** choose **View** and then **Open**

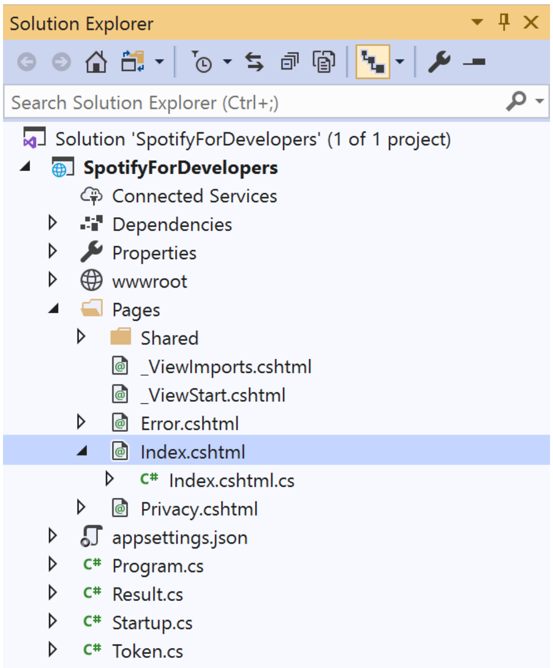
Step 4

In the **Code View** for **Index.cshtml.cs** below the **method** for `public async Task<IActionResult> OnPostGetArtistRelatedArtistsAsync(...) { ... }` enter the following **method**:

```
public async Task<IActionResult> OnPostGetMultipleAlbumsAsync(string value)
{
    LoadToken();
    var values = value.Split(",").ToList();
    var results = await Api.GetMultipleAlbumsAsync(values);
    if (results != null)
    {
        Results = results.Select(result => new Result()
        {
            Id = result.Id,
            Name = result.Name,
            Image = result?.Images?.FirstOrDefault()?.Url
        });
    }
    return Page();
}
```

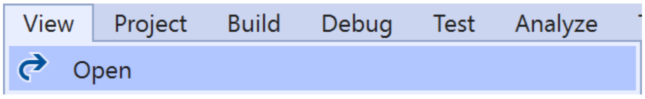
The **method** for `OnPostGetMultipleAlbumsAsync` is used to get **albums** by multiple **Album Id** on Spotify with the `value` and populate the **property** for `Results` accordingly.

Step 5



In the **Solution Explorer** in the **Pages** section select **Index.cshtml**

Step 6



Then from the **Menu** choose **View** and then **Open**

Step 7

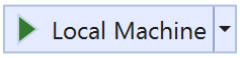
Once in the **Code View** for **Index.cshtml** above `<!-- Spotify Web API App Authorisation -->` enter the following:

```
<ul class="list-group mb-2">
    <li class="list-group-item list-group-item-primary">
        <h5 class="list-group-item-heading">Albums</h5>
    </li>
    <li class="list-group-item">
        <form asp-page-handler="GetMultipleAlbums" method="post">
            <input asp-for="Value" placeholder="Album Ids" class="form-control mb-2" />
            <button class="btn btn-primary mb-2">
                Get Multiple Albums
            </button>
        </form>
    </li>

    <!-- Albums -->
</ul>
```

This `form` will **post** to the **method** for `OnPostGetMultipleAlbumsAsync` with the `Value` as the **Album Ids** and will output to the **Results**.

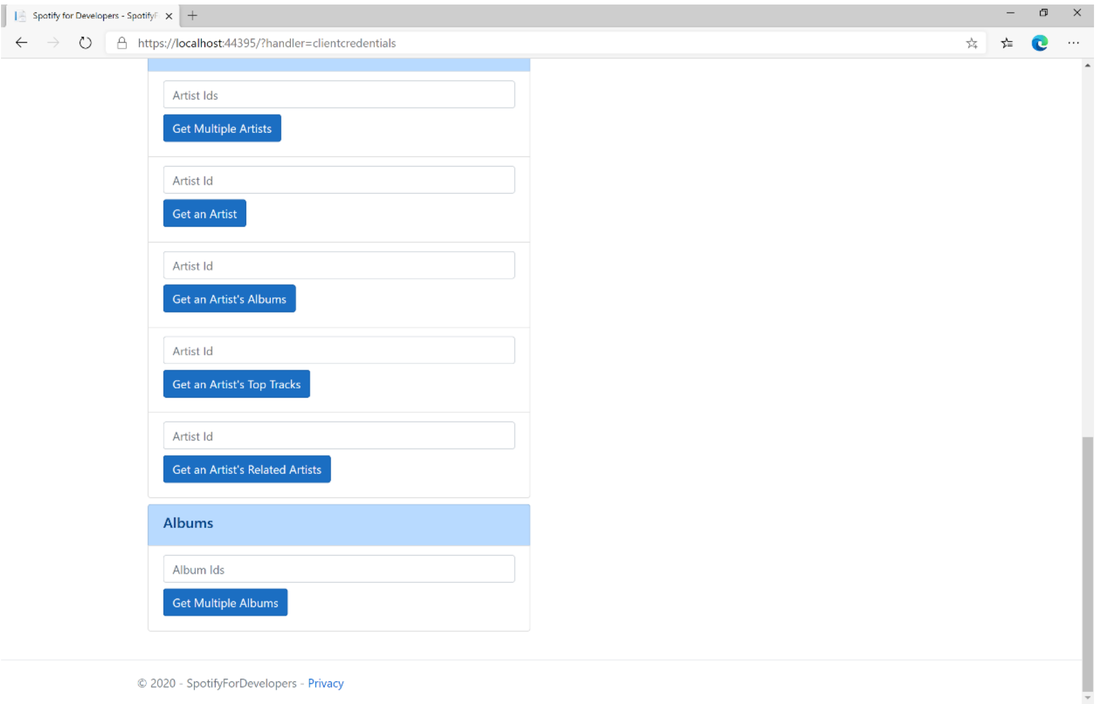
Step 8



Finally, in **Visual Studio 2019** select **IIS Application** to run the **Web Application**

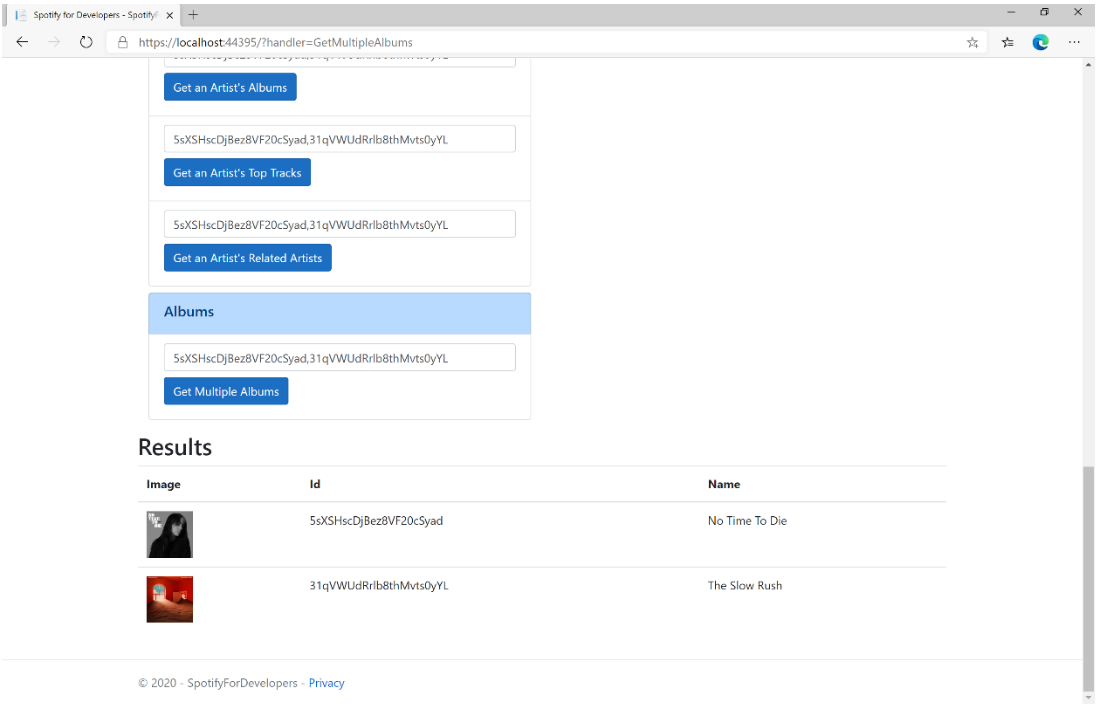
Step 9

Once the **Web Application** is running and you select **Client Credentials Flow Login** and scroll down you should see something like the following:


Spotify for Developers - Spotify
https://localhost:44395/?handler=clientcredentials
Artist Ids
Get Multiple Artists
Artist Id
Get an Artist
Artist Id
Get an Artist's Albums
Artist Id
Get an Artist's Top Tracks
Artist Id
Get an Artist's Related Artists
Albums
Album Ids
Get Multiple Albums
© 2020 - SpotifyForDevelopers - Privacy

Step 10

You can then enter an **Album Id** or if multiple ones separate them with a **comma** from **Get All New Releases** and select **Get Multiple Albums** and scroll down to view **Results** like the following:


Spotify for Developers - Spotify
https://localhost:44395/?handler=GetMultipleAlbums
Get an Artist's Albums
Get an Artist's Top Tracks
Get an Artist's Related Artists
Albums
Get Multiple Albums
© 2020 - SpotifyForDevelopers - Privacy

5sXSHscDjBez8VF20cSyad,31qVWUdRrlb8thMvts0yYL

5sXSHscDjBez8VF20cSyad,31qVWUdRrlb8thMvts0yYL

5sXSHscDjBez8VF20cSyad,31qVWUdRrlb8thMvts0yYL

Results

Image	Id	Name
	5sXSHscDjBez8VF20cSyad	No Time To Die
	31qVWUdRrlb8thMvts0yYL	The Slow Rush

Step 11

■

You can stop the **web application** in **Visual Studio 2019** by selecting the **Stop debugging** button

Step 12

✕

You can choose to exit **Visual Studio 2019** by selecting the **Close** button in the top right of the **application** as that completes this part of the workshop

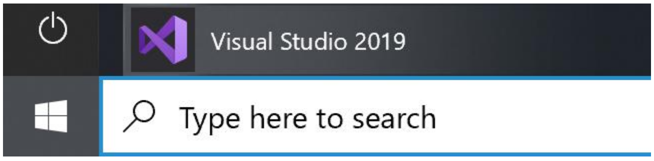
Get an Album

Get Spotify catalogue information for a single album identified by its unique Spotify Id.

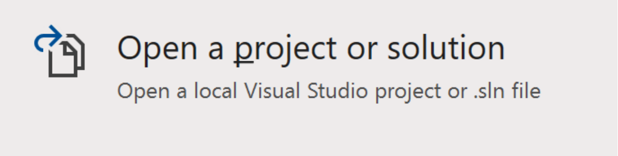
GET https://api.spotify.com/v1/albums/{id}	
Header	
Authorization	Valid Access Token from Spotify Accounts service
Path Parameter	
id	Spotify Id of the album
market	ISO 3166-1 alpha-2 country code e.g. "GB"

Header	Response
Success	
HTTP Status 200 OK	Album Object
Error	
Error Code	Error Object

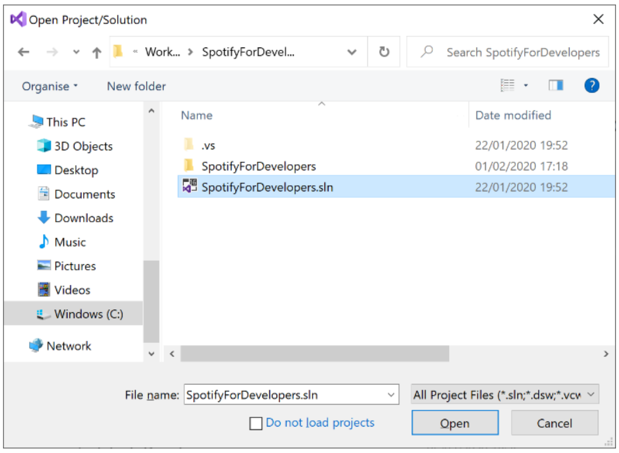
Step 1



If you chose to close **Visual Studio 2019** previously, in **Windows 10** choose **Start**, and then from the **Start Menu** find and select **Visual Studio 2019**


Open a project or solution
Open a local Visual Studio project or .sln file

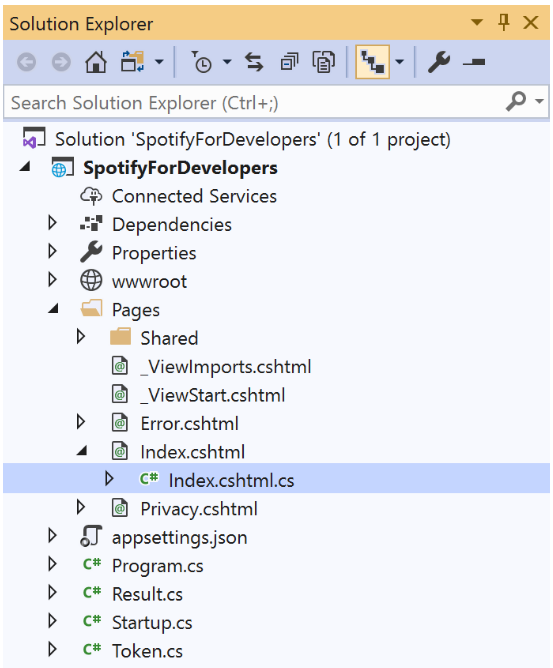
Once done, from the **Get started** screen for **Visual Studio 2019** select **Open a project or solution**



Name	Date modified
.vs	22/01/2020 19:52
SpotifyForDevelopers	01/02/2020 17:18
SpotifyForDevelopers.sln	22/01/2020 19:52

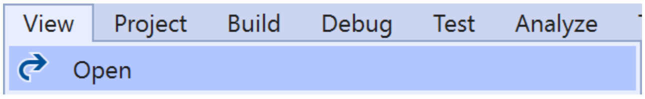
Then locate and select **SpotifyForDevelopers.sln** and select **Open** if you don't have this file already then please follow the previous parts of the workshop including **Getting Started**, **Authorisation Guide**, **Search & Browse** and **Artists, Albums & Tracks**

Step 2



Once opened, in the **Solution Explorer** open the **Pages** section, then open the **Index.cshtml** section and select **Index.cshtml.cs**

Step 3



Then from the **Menu** choose **View** and then **Open**

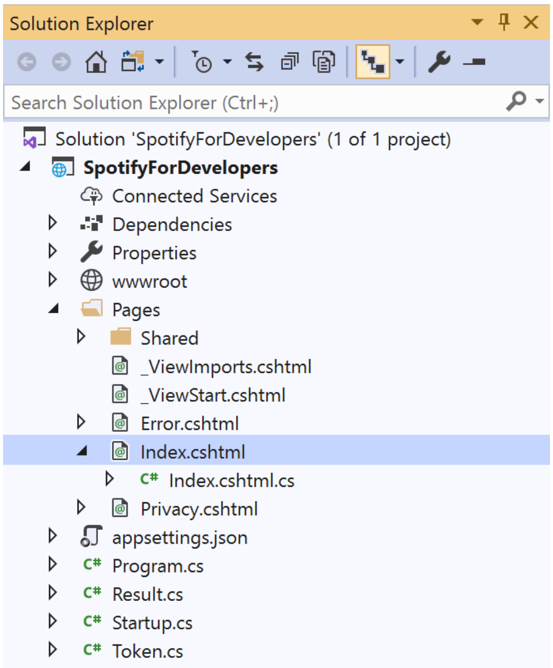
Step 4

In the **Code View** for **Index.cshtml.cs** below the **method** for `public async Task<IActionResult> OnPostGetMultipleAlbumsAsync(...) { ... }` enter the following **method**:

```
public async Task<IActionResult> OnPostGetAlbumAsync(string value)
{
    LoadToken();
    var result = await Api.GetAlbumAsync(value);
    if (result != null)
    {
        Results = new List<Result> { new Result()
        {
            Id = result.Id,
            Name = result.Name,
            Image = result?.Images?.FirstOrDefault()?.Url,
            Inner = new Result()
            {
                Id = result?.Artists?.FirstOrDefault()?.Id,
                Name = result?.Artists?.FirstOrDefault()?.Name
            }
        }};
    }
    return Page();
}
```

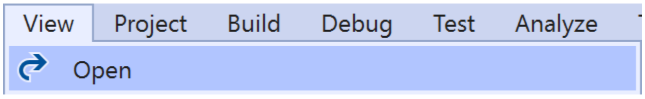
The **method** for `OnPostGetAlbumAsync` is used to get an **album** by **Album Id** on Spotify with the `value` and populate the **property** for `Results` accordingly.

Step 5



In the **Solution Explorer** in the **Pages** section select **Index.cshtml**

Step 6



Then from the **Menu** choose **View** and then **Open**

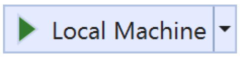
Step 7

Once in the **Code View** for **Index.cshtml** above `<!-- Albums -->` enter the following:

```
<li class="list-group-item">
    <form asp-page-handler="GetAlbum" method="post">
        <input asp-for="Value" placeholder="Album Id" class="form-control mb-2" />
        <button class="btn btn-primary mb-2">
            Get an Album
        </button>
    </form>
</li>
```

This `form` will **post** to the **method** for `OnPostGetAlbumAsync` with the `Value` as the **Album Id** and will output to the **Results**.

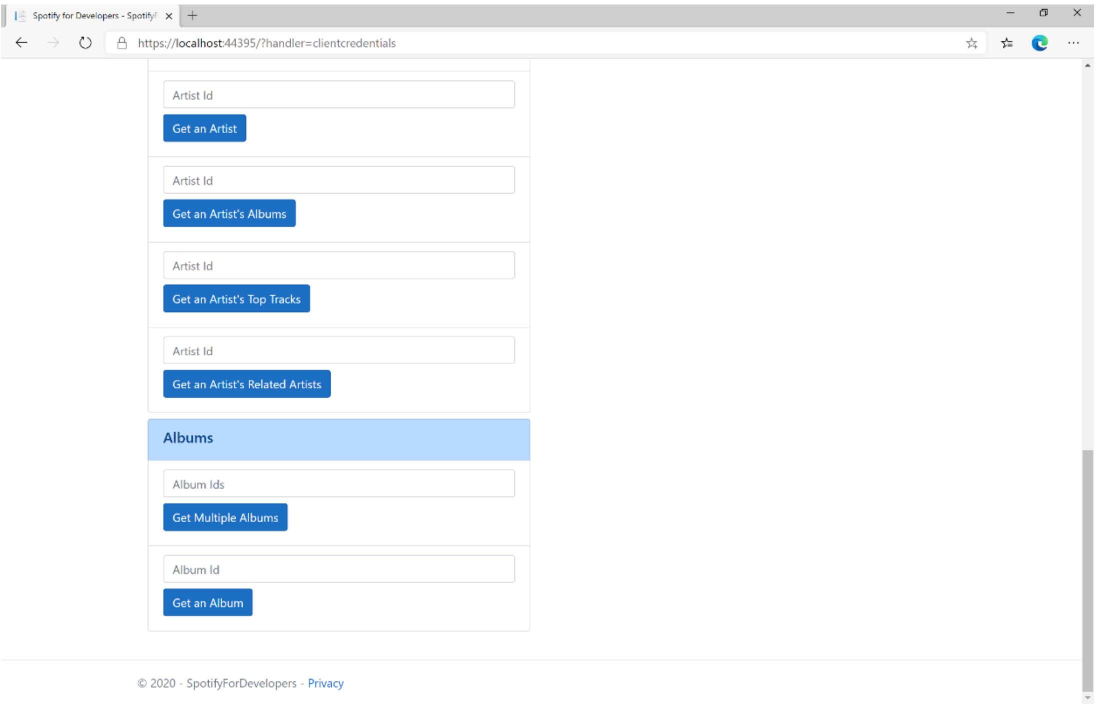
Step 8



Finally, in **Visual Studio 2019** select **IIS Application** to run the **Web Application**

Step 9

Once the **Web Application** is running and you select **Client Credentials Flow Login** and scroll down you should see something like the following:



Spotify for Developers - Spotify
https://localhost:44395/?handler=clientcredentials

Artist Id

Artist Id

Artist Id

Get an Artist's Top Tracks

Artist Id

Get an Artist's Related Artists

Albums

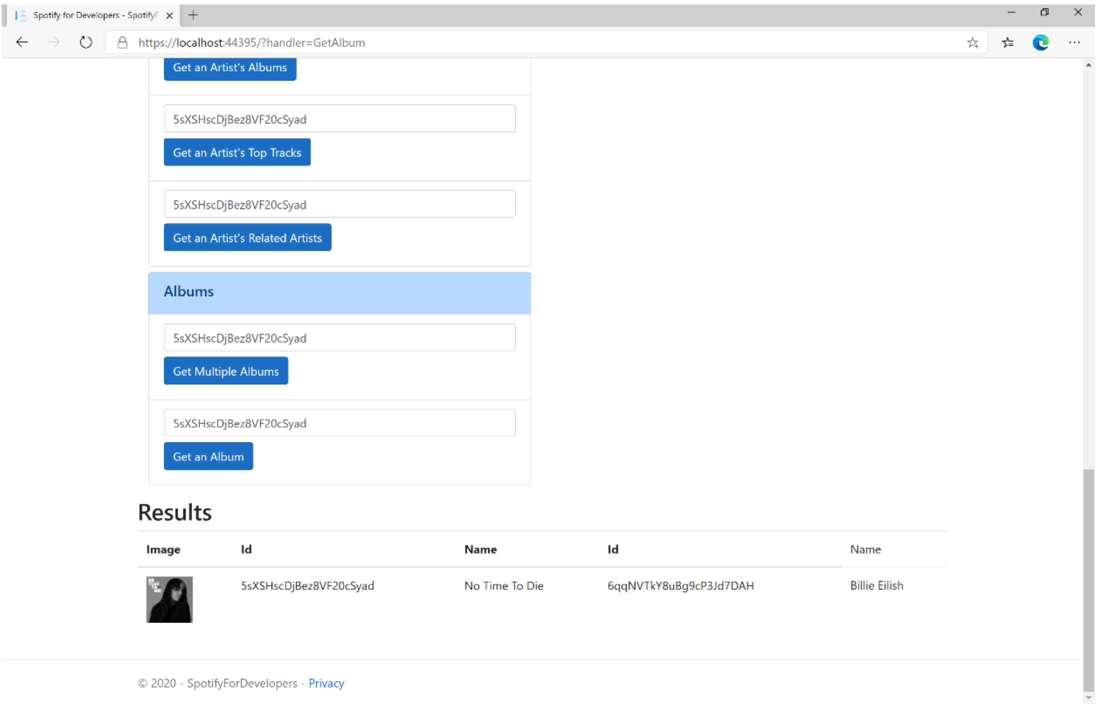
Album Ids

Album Id

© 2020 - SpotifyForDevelopers - Privacy

Step 10

You can then enter an **Album Id** from **Get All New Releases** and select **Get an Album** and scroll down to view **Results** like the following:



Spotify for Developers - Spotify
https://localhost:44395/?handler=GetAlbum
Get an Artist's Albums

5sXSHscDjBez8VF20cSyad

Get an Artist's Top Tracks

5sXSHscDjBez8VF20cSyad

Get an Artist's Related Artists

Albums

5sXSHscDjBez8VF20cSyad

5sXSHscDjBez8VF20cSyad

Results

Image	Id	Name	Id	Name
	5sXSHscDjBez8VF20cSyad	No Time To Die	6qqNVTkY8uBg9cP3Jd7DAH	Billie Eilish

© 2020 - SpotifyForDevelopers - Privacy

Step 11

■

You can stop the **web application** in **Visual Studio 2019** by selecting the **Stop debugging** button

Step 12

×

You can choose to exit **Visual Studio 2019** by selecting the **Close** button in the top right of the **application** as that completes this part of the workshop

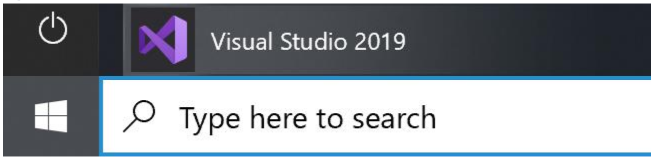
Get an Album's Tracks

Get Spotify catalogue information about an album’s tracks. Optional parameters can be used to limit the number of tracks returned.

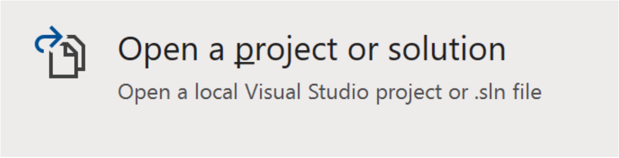
GET https://api.spotify.com/v1/albums/{id}/tracks	
Header	
Authorization	Valid Access Token from Spotify Accounts service
Path Parameter	
id	Spotify Id of the album
Query Parameters	
limit	Maximum number of results to return
offset	Index of first result to return
market	ISO 3166-1 alpha-2 country code e.g. “GB” or “from_token”. Provide to apply Track Relinking

Header	**Response**
Success	
HTTP Status 200 OK	Array of Simplified Track Objects wrapped in a Paging Object
Error	
Error Code	Error Object

Step 1



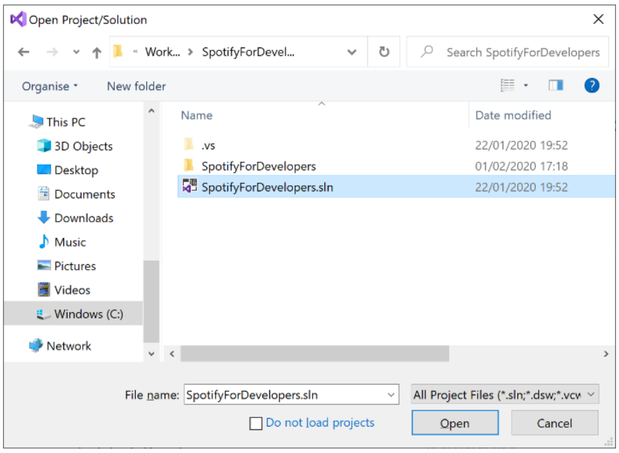
If you chose to close **Visual Studio 2019** previously, in **Windows 10** choose **Start**, and then from the **Start Menu** find and select **Visual Studio 2019**



Open a project or solution

Open a local Visual Studio project or .sln file

Once done, from the **Get started** screen for **Visual Studio 2019** select **Open a project or solution**



Name	Date modified
.vs	22/01/2020 19:52
SpotifyForDevelopers	01/02/2020 17:18
SpotifyForDevelopers.sln	22/01/2020 19:52

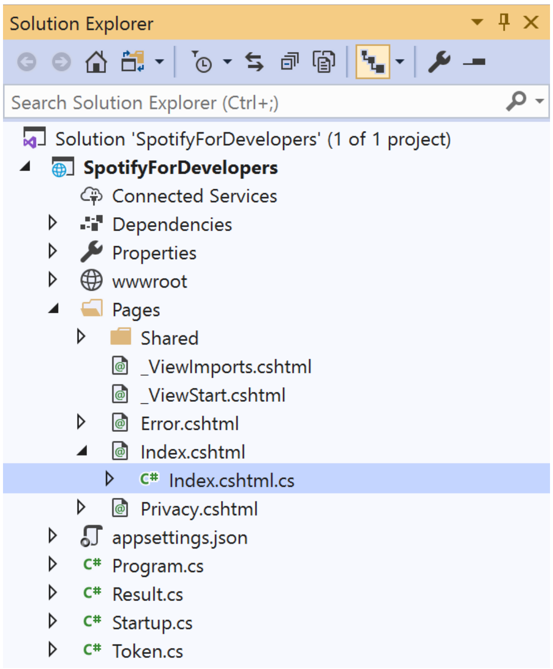
File name: SpotifyForDevelopers.sln

All Project Files (*.sln;*.dsw;*.vcw

□ Do not load projects

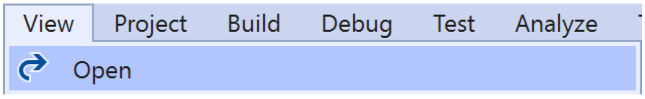
Then locate and select **SpotifyForDevelopers.sln** and select **Open** if you don’t have this file already then please follow the previous parts of the workshop including **Getting Started**, **Authorisation Guide**, **Search & Browse** and **Artists, Albums & Tracks**

Step 2



Once opened, in the **Solution Explorer** open the **Pages** section, then open the **Index.cshtml** section and select **Index.cshtml.cs**

Step 3



Then from the **Menu** choose **View** and then **Open**

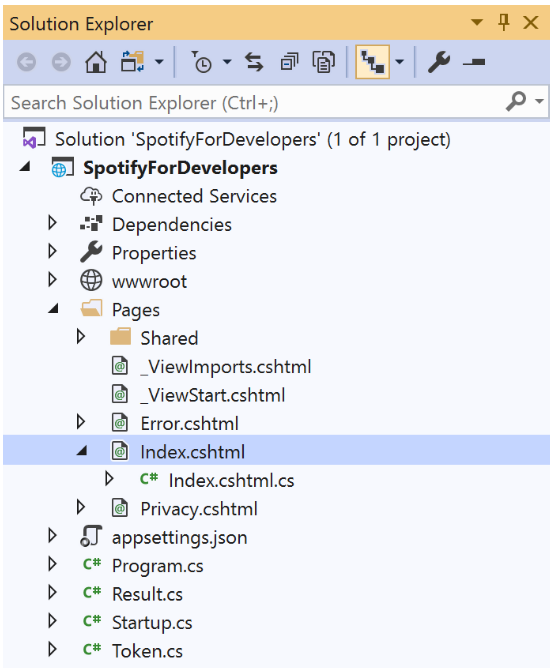
Step 4

In the **Code View** for **Index.cshtml.cs** below the **method** for `public async Task<IActionResult> OnPostGetAlbumAsync(...) { ... }` enter the following **method**:

```
public async Task<IActionResult> OnPostGetAlbumTracksAsync(string value)
{
    LoadToken();
    var results = await Api.GetAlbumTracksAsync(value, country);
    if (results?.Items != null)
    {
        Results = results.Items.Select(result => new Result()
        {
            Id = result.Id,
            Name = result.Name,
            Image = result?.Album?.Images?.FirstOrDefault()?.Url,
            Inner = new Result()
            {
                Id = result?.Artists?.FirstOrDefault()?.Id,
                Name = result?.Artists?.FirstOrDefault()?.Name
            }
        });
    }
    return Page();
}
```

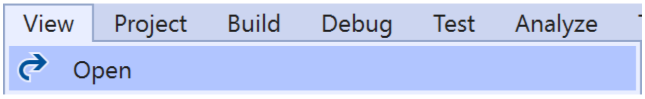
The **method** for `OnPostGetAlbumTracksAsync` is used to get the **tracks** of an **album** by **Album Id** on Spotify with the `value` and populate the **property** for `Results` accordingly.

Step 5



In the **Solution Explorer** in the **Pages** section select **Index.cshtml**

Step 6



Then from the **Menu** choose **View** and then **Open**

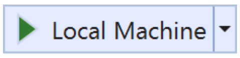
Step 7

Once in the **Code View** for **Index.cshtml** above `<!-- Albums -->` enter the following:

```
<li class="list-group-item">
    <form asp-page-handler="GetAlbumTracks" method="post">
        <input asp-for="Value" placeholder="Album Id" class="form-control mb-2" />
        <button class="btn btn-primary mb-2">
            Get an Album's Tracks
        </button>
    </form>
</li>
```

This `form` will **post** to the **method** for `OnPostGetAlbumTracksAsync` with the `Value` as the **Album Id** and will output to the **Results**.

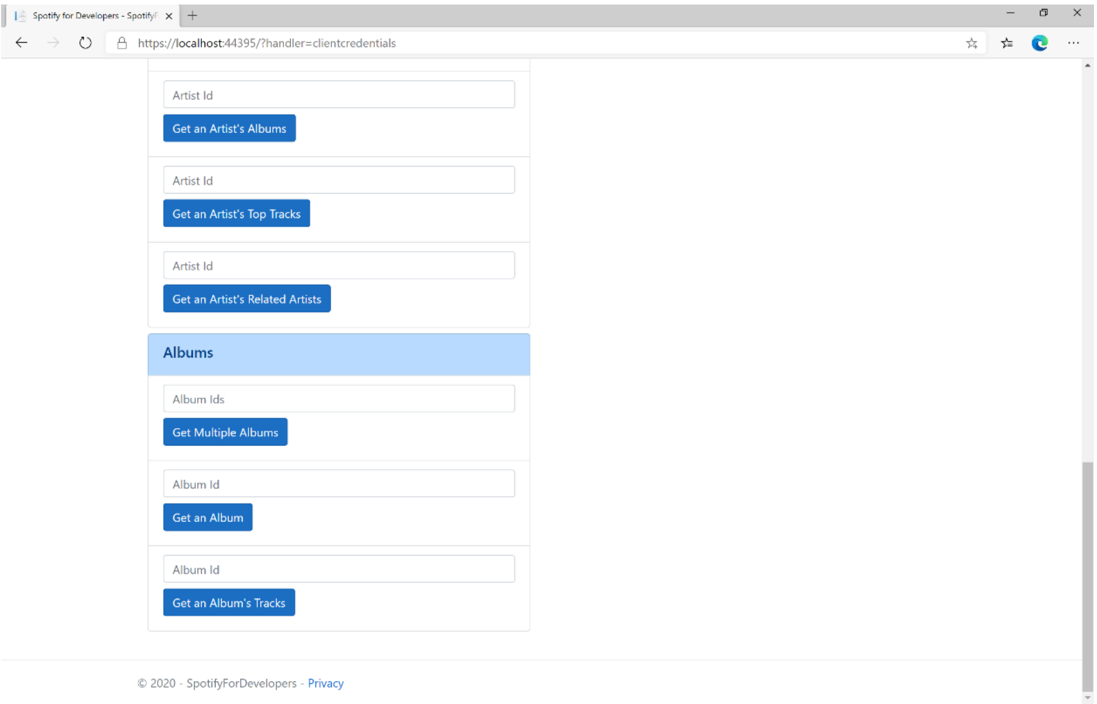
Step 8



Finally, in **Visual Studio 2019** select **IIS Application** to run the **Web Application**

Step 9

Once the **Web Application** is running and you select **Client Credentials Flow Login** and scroll down you should see something like the following:



Spotify for Developers - Spotify
https://localhost:44395/?handler=clientcredentials

Artist Id

Artist Id

Get an Artist's Top Tracks

Artist Id

Get an Artist's Related Artists

Albums

Album Ids

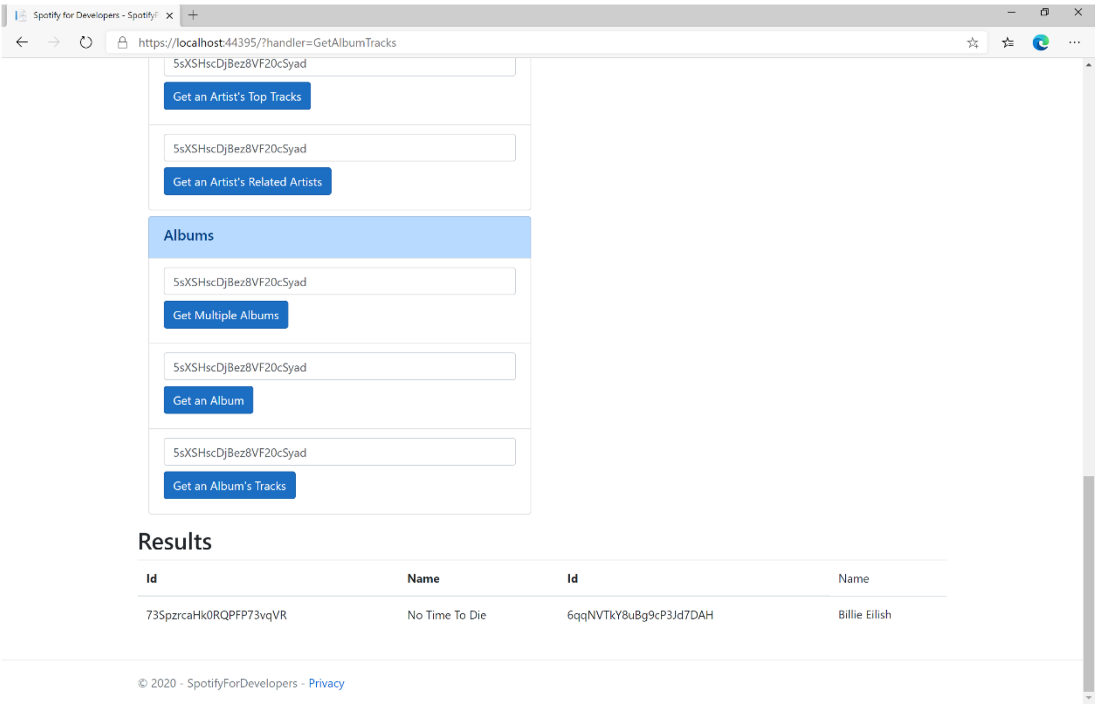
Album Id

Album Id

© 2020 - SpotifyForDevelopers - Privacy

Step 10

You can then enter an **Album Id** from **Get All New Releases** and select **Get an Album's Tracks** and scroll down to view **Results** like the following:



Spotify for Developers - Spotify
https://localhost:44395/?handler=GetAlbumTracks

5sXSHscDjBez8VF20cSyad

Get an Artist's Top Tracks

5sXSHscDjBez8VF20cSyad

Get an Artist's Related Artists

Albums

5sXSHscDjBez8VF20cSyad

5sXSHscDjBez8VF20cSyad

5sXSHscDjBez8VF20cSyad

Results

Id	Name	Id	Name
73SpzrcaHk0RQPFP73vqVR	No Time To Die	6qqNVTkY8uBg9cP3Jd7DAH	Billie Eilish

© 2020 - SpotifyForDevelopers - Privacy

Step 11

■

You can stop the **web application** in **Visual Studio 2019** by selecting the **Stop debugging** button

Step 12

✕

You can choose to exit **Visual Studio 2019** by selecting the **Close** button in the top right of the **application** as that completes this part of the workshop

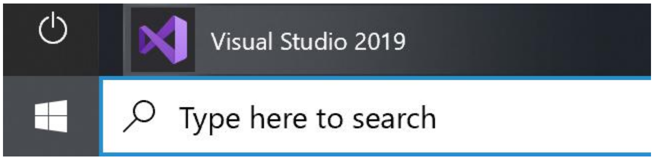
Get Multiple Tracks

Get Spotify catalogue information for multiple tracks based on their Spotify Ids.

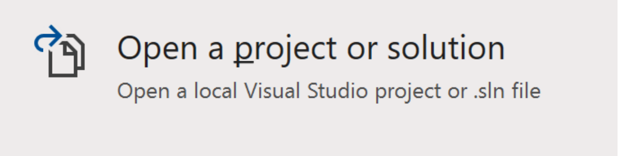
GET https://api.spotify.com/v1/tracks	
Header	
Authorization	Valid Access Token from Spotify Accounts service
Query Parameter	
ids	Maximum of 50 comma-separated list of track Spotify Ids
market	ISO 3166-1 alpha-2 country code e.g. “GB”. Provide to apply Track Relinking

Header	Response
Success	
HTTP Status 200 OK	Tracks as Array of Track Objects
Error	
Error Code	Error Object

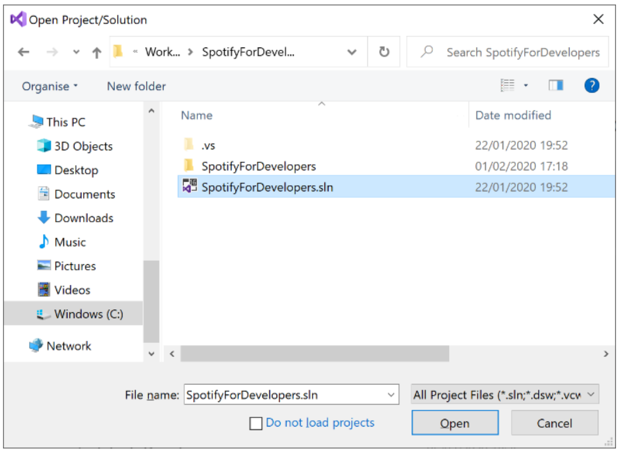
Step 1



If you chose to close **Visual Studio 2019** previously, in **Windows 10** choose **Start**, and then from the **Start Menu** find and select **Visual Studio 2019**

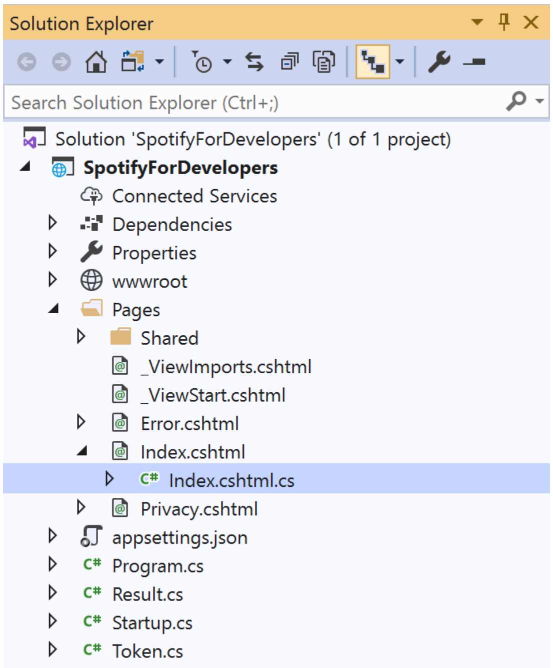

Open a project or solution
Open a local Visual Studio project or .sln file

Once done, from the **Get started** screen for **Visual Studio 2019** select **Open a project or solution**


Open Project/Solution
Work... › SpotifyForDevel...
Search SpotifyForDevelopers
Organise
New folder
This PC
3D Objects
Desktop
Documents
Downloads
Music
Pictures
Videos
Windows (C:)
Network
Name
Date modified
.vs 22/01/2020 19:52
SpotifyForDevelopers 01/02/2020 17:18
SpotifyForDevelopers.sln 22/01/2020 19:52
File name: SpotifyForDevelopers.sln
All Project Files (*.sln;*.dsw;*.vcw
Do not load projects
Open
Cancel

Then locate and select **SpotifyForDevelopers.sln** and select **Open** if you don’t have this file already then please follow the previous parts of the workshop including **Getting Started**, **Authorisation Guide**, **Search & Browse** and **Artists, Albums & Tracks**

Step 2



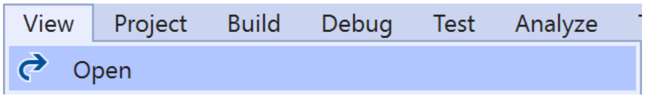
Solution Explorer

Search Solution Explorer (Ctrl+;)

- Solution 'SpotifyForDevelopers' (1 of 1 project)
  - **SpotifyForDevelopers**
    - Connected Services
    - Dependencies
    - Properties
    - wwwroot
    - Pages
      - Shared
      - _ViewImports.cshtml
      - _ViewStart.cshtml
      - Error.cshtml
      - Index.cshtml
        - Index.cshtml.cs
      - Privacy.cshtml
    - appsettings.json
    - Program.cs
    - Result.cs
    - Startup.cs
    - Token.cs

Once opened, in the **Solution Explorer** open the **Pages** section, then open the **Index.cshtml** section and select **Index.cshtml.cs**

Step 3



Then from the **Menu** choose **View** and then **Open**

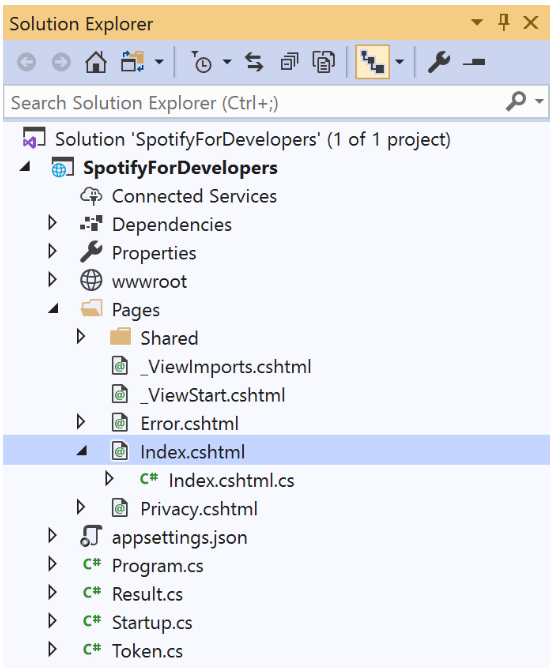
Step 4

In the **Code View** for **Index.cshtml.cs** below the **method** for `public async Task<IActionResult> OnPostGetAlbumTracksAsync(...) { ... }` enter the following **method**:

```
public async Task<IActionResult> OnPostGetSeveralTracksAsync(string value)
{
    LoadToken();
    var values = value.Split(",").ToList();
    var results = await Api.GetSeveralTracksAsync(values);
    if (results != null)
    {
        Results = results.Select(result => new Result()
        {
            Id = result.Id,
            Name = result.Name,
            Image = result?.Album?.Images?.FirstOrDefault()?.Url
        });
    }
    return Page();
}
```

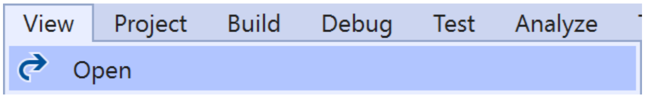
The **method** for `OnPostGetSeveralTracksAsync` is used to get **tracks** by multiple **Track Ids** on Spotify with the `value` and populate the **property** for `Results` accordingly.

Step 5



In the **Solution Explorer** in the **Pages** section select **Index.cshtml**

Step 6



Then from the **Menu** choose **View** and then **Open**

Step 7

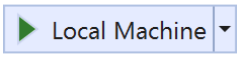
Once in the **Code View** for **Index.cshtml** above `<!-- Spotify Web API App Authorisation -->` enter the following:

```
<ul class="list-group mb-2">
    <li class="list-group-item list-group-item-primary">
        <h5 class="list-group-item-heading">Tracks</h5>
    </li>
    <li class="list-group-item">
        <form asp-page-handler="GetSeveralTracks" method="post">
            <input asp-for="Value" placeholder="Track Ids" class="form-control mb-2" />
            <button class="btn btn-primary mb-2">
                Get Multiple Tracks
            </button>
        </form>
    </li>

    <!-- Tracks -->
</ul>
```

This `form` will **post** to the **method** for `OnPostGetSeveralTracksAsync` with the `Value` as the **Track Ids** and will output to the **Results**.

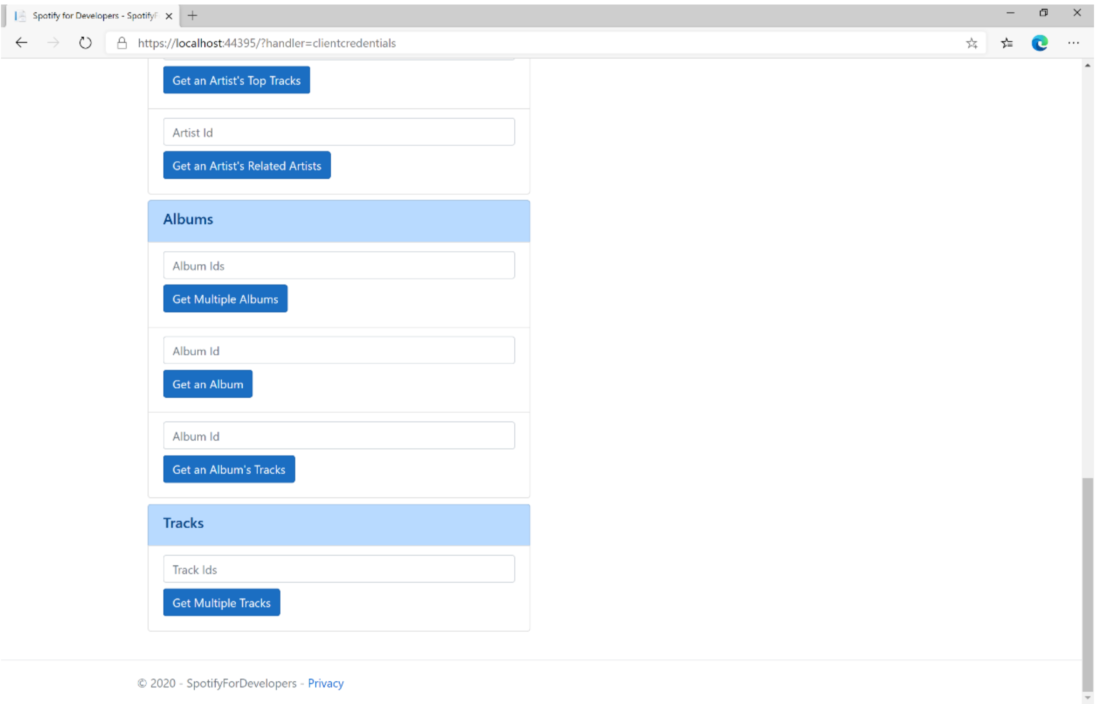
Step 8



Finally, in **Visual Studio 2019** select **IIS Application** to run the **Web Application**

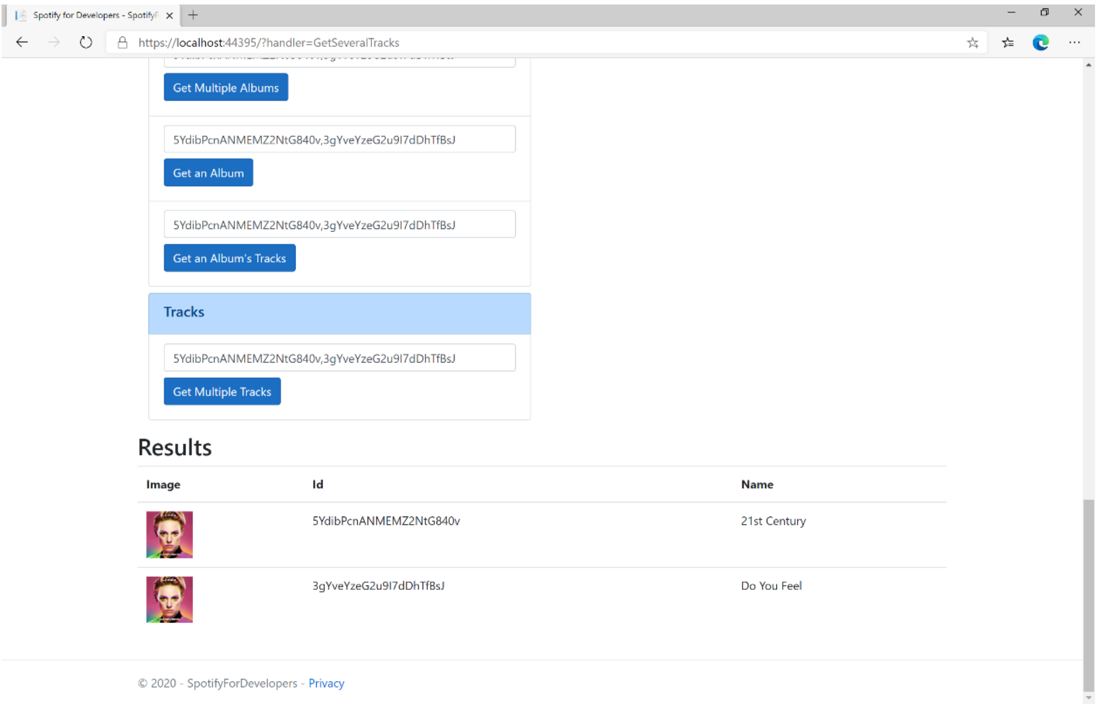
Step 9

Once the **Web Application** is running and you select **Client Credentials Flow Login** and scroll down you should see something like the following:


Spotify for Developers - Spotify
https://localhost:44395/?handler=clientcredentials
Get an Artist's Top Tracks
Artist Id
Get an Artist's Related Artists
Albums
Album Ids
Get Multiple Albums
Album Id
Get an Album
Album Id
Get an Album's Tracks
Tracks
Track Ids
Get Multiple Tracks
© 2020 - SpotifyForDevelopers - Privacy

Step 10

You can then enter an **Album Id** from **Get All New Releases** select **Get an Album's Tracks** and copy some **Track Ids** to use and select **Get Multiple Tracks** and scroll down to view **Results** like the following:


Spotify for Developers - Spotify
https://localhost:44395/?handler=GetSeveralTracks
Get Multiple Albums
Get an Album
Get an Album's Tracks
Tracks
Get Multiple Tracks
© 2020 - SpotifyForDevelopers - Privacy

5YdibPcnANMEMZ2NtG840v,3gYveYzeG2u9I7dDhTfBsJ

5YdibPcnANMEMZ2NtG840v,3gYveYzeG2u9I7dDhTfBsJ

5YdibPcnANMEMZ2NtG840v,3gYveYzeG2u9I7dDhTfBsJ

Results

Image	Id	Name
	5YdibPcnANMEMZ2NtG840v	21st Century
	3gYveYzeG2u9I7dDhTfBsJ	Do You Feel

Step 11

You can stop the **web application** in **Visual Studio 2019** by selecting the **Stop debugging** button

Step 12

You can choose to exit **Visual Studio 2019** by selecting the **Close** button in the top right of the **application** as that completes this part of the workshop

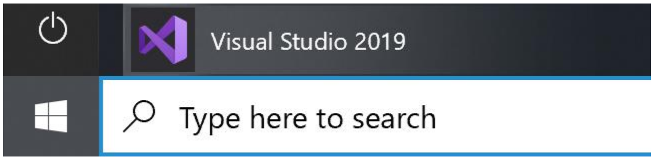
Get a Track

Get Spotify catalogue information for a single track identified by its unique Spotify Id.

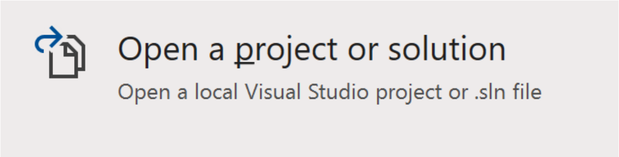
GET https://api.spotify.com/v1/tracks/{id}	
Header	
Authorization	Valid Access Token from Spotify Accounts service
Path Parameter	
id	Spotify Id of the track
market	ISO 3166-1 alpha-2 country code e.g. "GB". Provide to apply Track Relinking

Header	Response
Success	
HTTP Status 200 OK	Track Object
Error	
Error Code	Error Object

Step 1



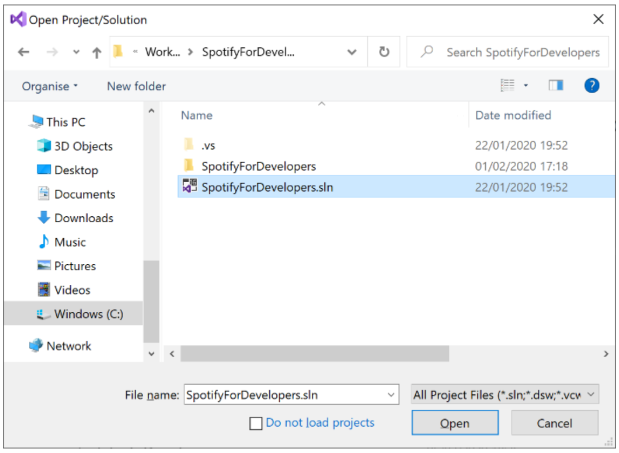
If you chose to close **Visual Studio 2019** previously, in **Windows 10** choose **Start**, and then from the **Start Menu** find and select **Visual Studio 2019**



Open a project or solution

Open a local Visual Studio project or .sln file

Once done, from the **Get started** screen for **Visual Studio 2019** select **Open a project or solution**



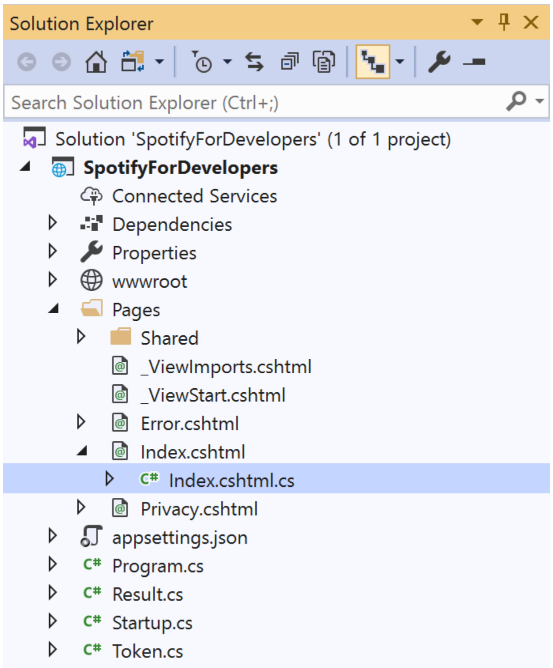
Name	Date modified
.vs	22/01/2020 19:52
SpotifyForDevelopers	01/02/2020 17:18
SpotifyForDevelopers.sln	22/01/2020 19:52

File name: SpotifyForDevelopers.sln

All Project Files (*.sln;*.dsw;*.vcw

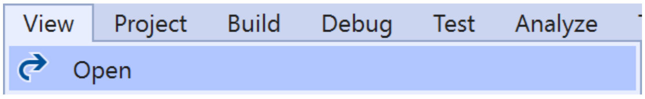
Then locate and select **SpotifyForDevelopers.sln** and select **Open** if you don't have this file already then please follow the previous parts of the workshop including **Getting Started**, **Authorisation Guide**, **Search & Browse** and **Artists, Albums & Tracks**

Step 2



Once opened, in the **Solution Explorer** open the **Pages** section, then open the **Index.cshtml** section and select **Index.cshtml.cs**

Step 3



Then from the **Menu** choose **View** and then **Open**

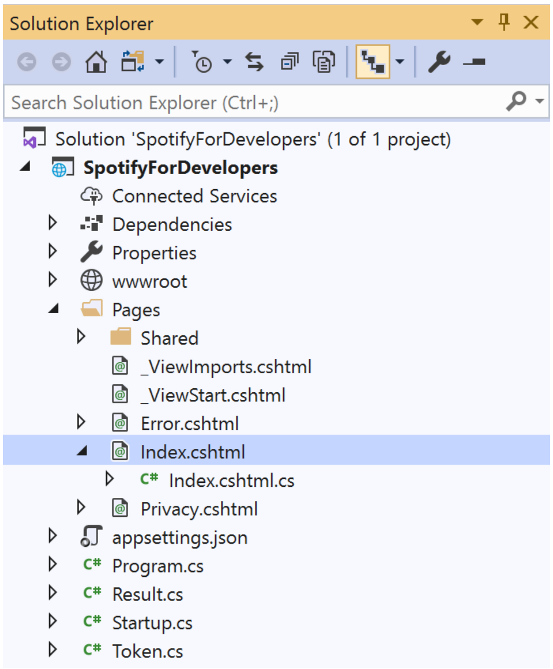
Step 4

In the **Code View** for **Index.cshtml.cs** below the **method** for `public async Task<IActionResult> OnPostGetSeveralTracksAsync(...) { ... }` enter the following **method**:

```
public async Task<IActionResult> OnPostGetTrackAsync(string value)
{
    LoadToken();
    var result = await Api.GetTrackAsync(value);
    if (result != null)
    {
        Results = new List<Result> { new Result()
        {
            Id = result.Id,
            Name = result.Name,
            Image = result?.Album?.Images?.FirstOrDefault()?.Url,
            Inner = new Result()
            {
                Id = result?.Artists?.FirstOrDefault()?.Id,
                Name = result?.Artists?.FirstOrDefault()?.Name
            }
        }};
    }
    return Page();
}
```

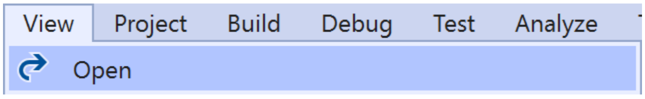
The **method** for `OnPostGetTrackAsync` is used to get a **track** by **Track Id** on Spotify with the `value` and populate the **property** for `Results` accordingly.

Step 5



In the **Solution Explorer** in the **Pages** section select **Index.cshtml**

Step 6



Then from the **Menu** choose **View** and then **Open**

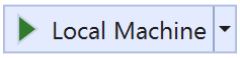
Step 7

Once in the **Code View** for **Index.cshtml** above `<!-- Tracks -->` enter the following:

```
<li class="list-group-item">
    <form asp-page-handler="GetTrack" method="post">
        <input asp-for="Value" placeholder="Track Id" class="form-control mb-2" />
        <button class="btn btn-primary mb-2">
            Get a Track
        </button>
    </form>
</li>
```

This `form` will **post** to the **method** for `OnPostGetTrackAsync` with the `Value` as the **Track Id** and will output to the **Results**.

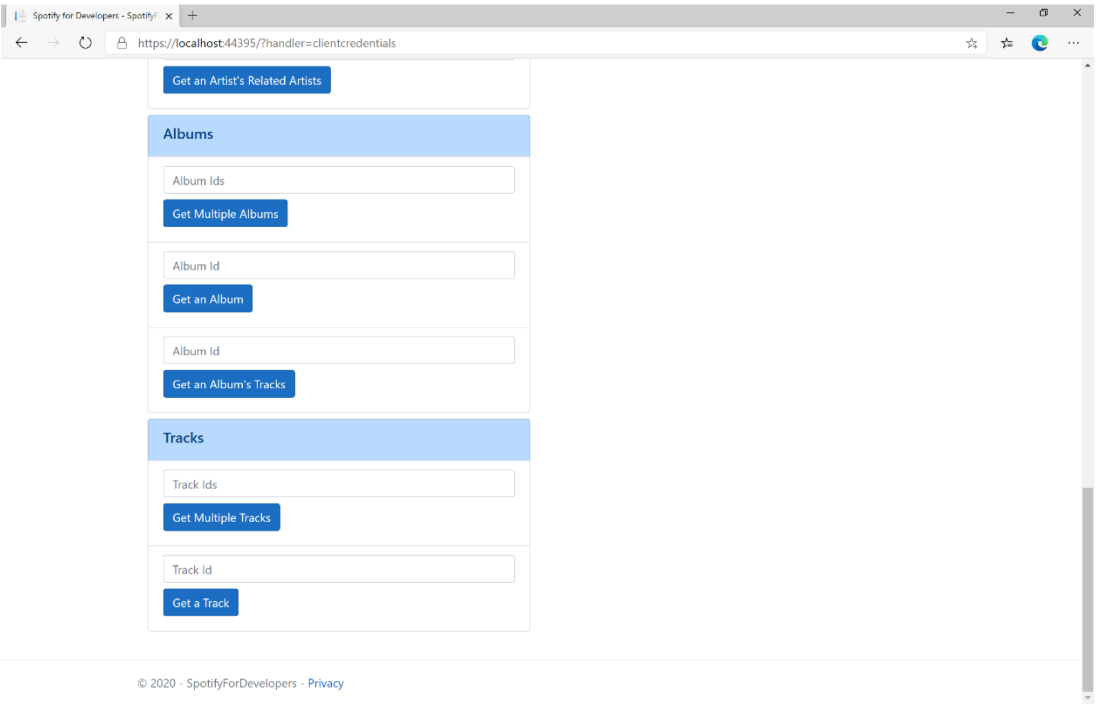
Step 8



Finally, in **Visual Studio 2019** select **IIS Application** to run the **Web Application**

Step 9

Once the **Web Application** is running and you select **Client Credentials Flow Login** and scroll down you should see something like the following:

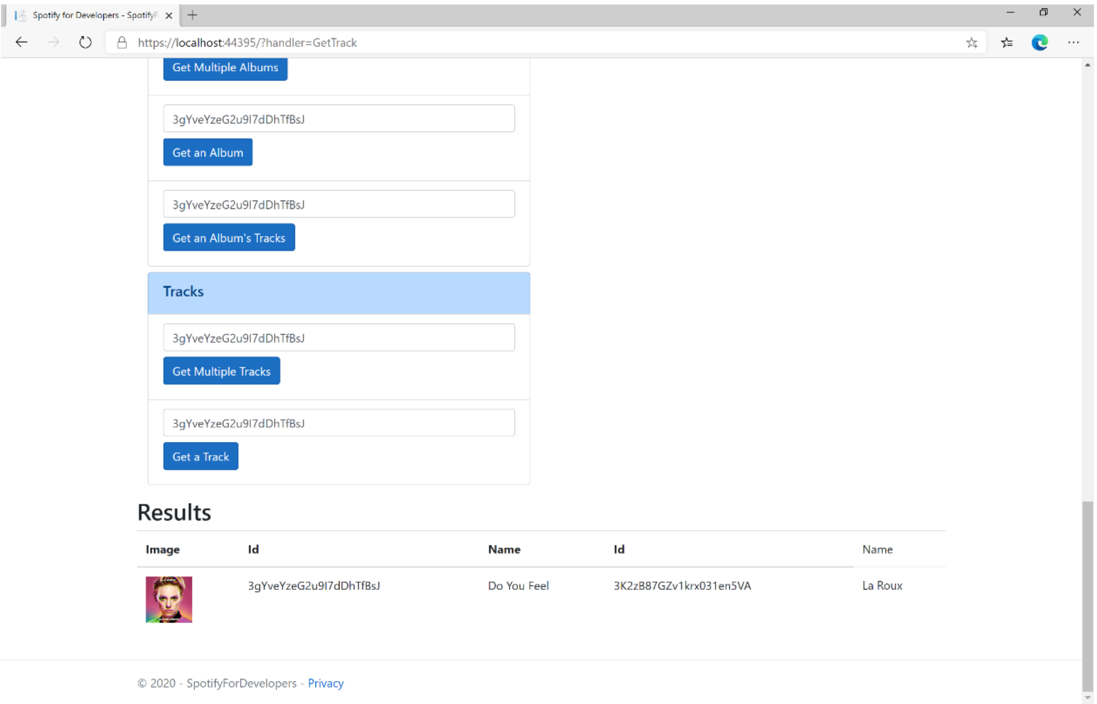


https://localhost:44395/?handler=clientcredentials

© 2020 - SpotifyForDevelopers - Privacy

Step 10

You can then enter an **Album Id** from **Get All New Releases** select **Get an Album's Tracks** and copy a **Track Id** to use and select **Get a Track** and scroll down to view **Results** like the following:



https://localhost:44395/?handler=GetTrack

3gYveYzeG2u9I7dDhTfBsJ

3gYveYzeG2u9I7dDhTfBsJ

3gYveYzeG2u9I7dDhTfBsJ

3gYveYzeG2u9I7dDhTfBsJ

Results

Image	Id	Name	Id	Name
	3gYveYzeG2u9I7dDhTfBsJ	Do You Feel	3K2zB87GZv1krx031en5VA	La Roux

© 2020 - SpotifyForDevelopers - Privacy

Step 11

You can stop the **web application** in **Visual Studio 2019** by selecting the **Stop debugging** button

Step 12

You can choose to exit **Visual Studio 2019** by selecting the **Close** button in the top right of the **application** as that completes this part of the workshop

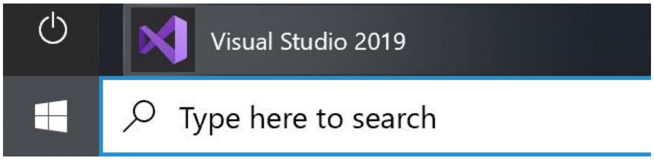
Get Audio Features for Several Tracks

Get audio feature information for a single track identified by its unique Spotify Id.

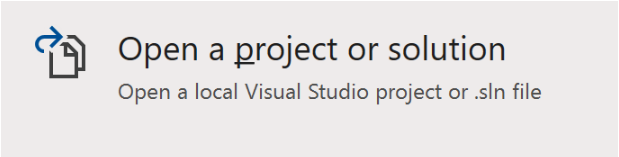
https://api.spotify.com/v1/audio-features	
Header	
Authorization	Valid Access Token from Spotify Accounts service
Path Parameter	
ids	Maximum of 100 comma-separated list of track Spotify Ids

Header	Response
Success	
HTTP Status 200 OK	Audio Features with Array of Audio Features Object
Error	
Error Code	Error Object

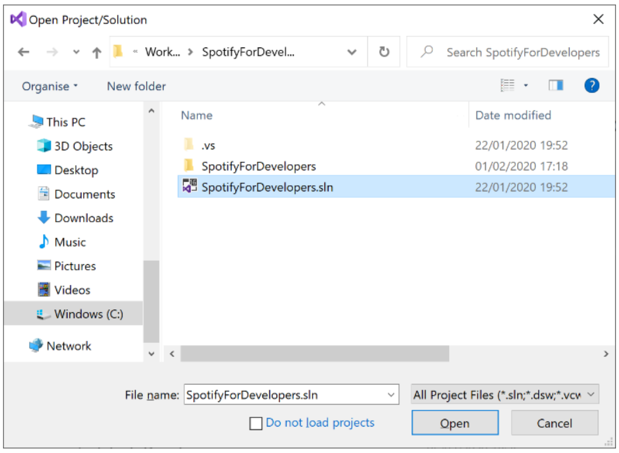
Step 1



If you chose to close **Visual Studio 2019** previously, in **Windows 10** choose **Start**, and then from the **Start Menu** find and select **Visual Studio 2019**

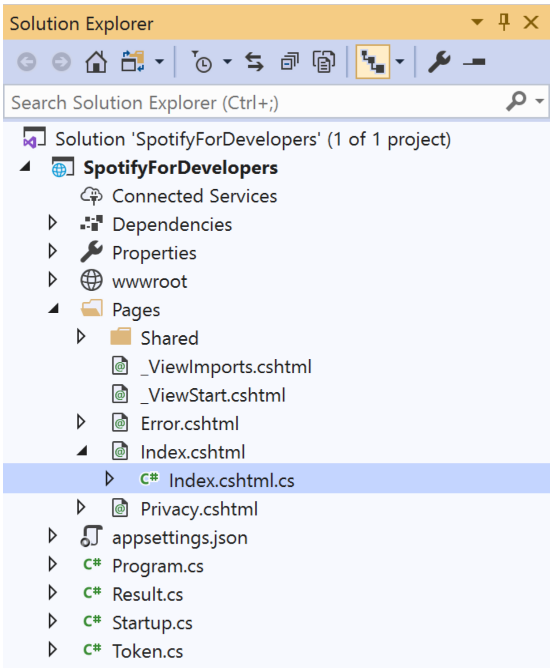

Open a project or solution
Open a local Visual Studio project or .sln file

Once done, from the **Get started** screen for **Visual Studio 2019** select **Open a project or solution**


Open Project/Solution
Work... > SpotifyForDevel...
Search SpotifyForDevelopers
Organise
New folder
This PC
3D Objects
Desktop
Documents
Downloads
Music
Pictures
Videos
Windows (C:)
Network
Name
Date modified
.vs 22/01/2020 19:52
SpotifyForDevelopers 01/02/2020 17:18
SpotifyForDevelopers.sln 22/01/2020 19:52
File name: SpotifyForDevelopers.sln
All Project Files (*.sln;*.dsw;*.vcw
Do not load projects
Open
Cancel

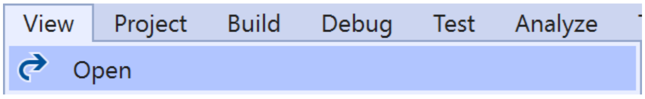
Then locate and select **SpotifyForDevelopers.sln** and select **Open** if you don't have this file already then please follow the previous parts of the workshop including **Getting Started**, **Authorisation Guide**, **Search & Browse** and **Artists, Albums & Tracks**

Step 2



Once opened, in the **Solution Explorer** open the **Pages** section, then open the **Index.cshtml** section and select **Index.cshtml.cs**

Step 3



Then from the **Menu** choose **View** and then **Open**

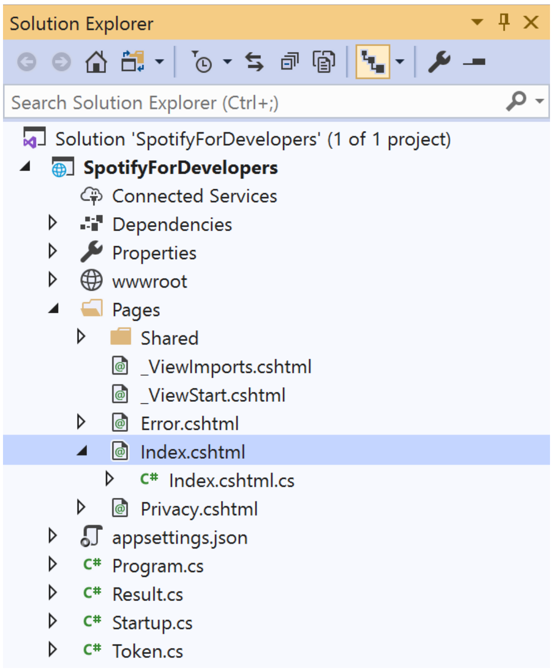
Step 4

In the **Code View** for **Index.cshtml.cs** below the **method** for `public async Task<IActionResult> OnPostGetTrackAsync(...) { ... }` enter the following **method**:

```
public async Task<IActionResult> OnPostGetAudioFeaturesForSeveralTracksAsync(string value)
{
    LoadToken();
    var values = value.Split(",").ToList();
    var results = await Api.GetSeveralTracksAudioFeaturesAsync(values);
    if (results != null)
    {
        Results = results.Select(result => new Result()
        {
            Name = System.Text.Json.JsonSerializer.Serialize(result,
                new System.Text.Json.JsonSerializerOptions()
                { WriteIndented = true })
        });
    }
    return Page();
}
```

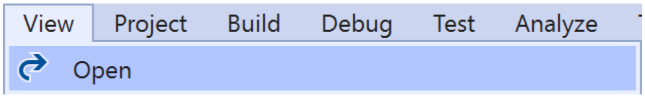
The **method** for `OnPostGetAudioFeaturesForSeveralTracksAsync` is used to get **audio features** for **tracks** by multiple **Track Ids** on Spotify with the `value` and populate the **property** for `Results` accordingly.

Step 5



In the **Solution Explorer** in the **Pages** section select **Index.cshtml**

Step 6



Then from the **Menu** choose **View** and then **Open**

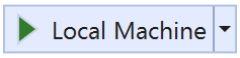
Step 7

Once in the **Code View** for **Index.cshtml** above `<!-- Tracks -->` enter the following:

```
<li class="list-group-item">
    <form asp-page-handler="GetAudioFeaturesForSeveralTracks" method="post">
        <input asp-for="Value" placeholder="Track Ids" class="form-control mb-2" />
        <button class="btn btn-primary mb-2">
            Get Audio Features for Several Tracks
        </button>
    </form>
</li>
```

This `form` will **post** to the **method** for `GetAudioFeaturesForSeveralTracks` with the `Value` as the **Track Ids** and will output to the **Results**.

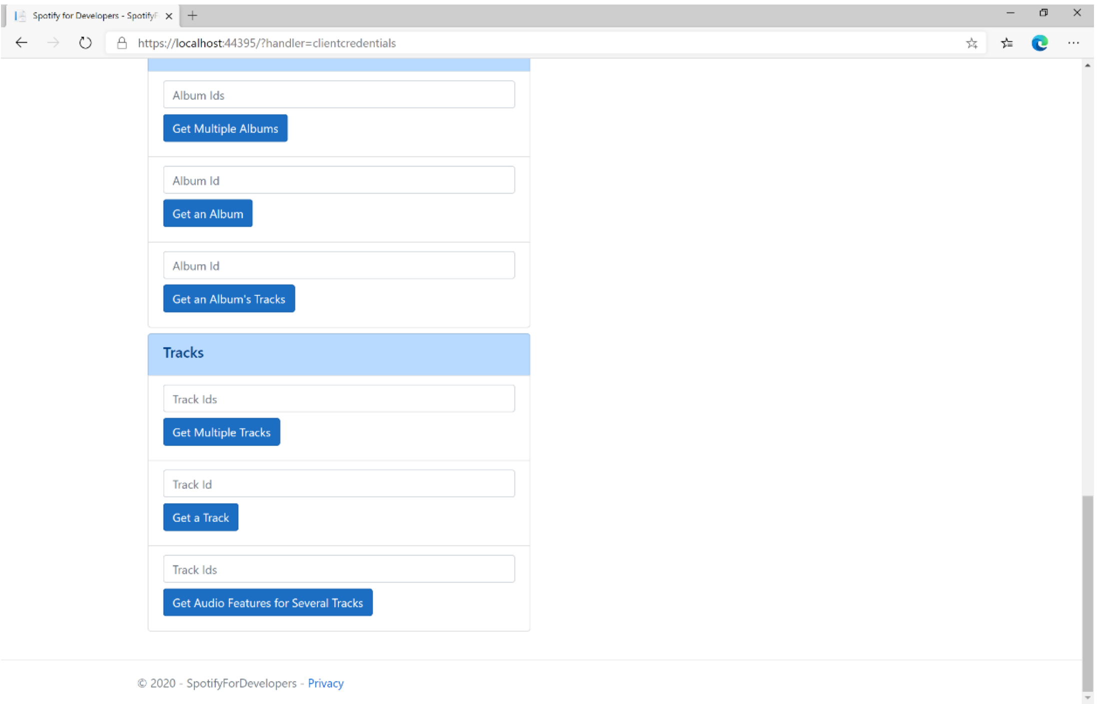
Step 8



Finally, in **Visual Studio 2019** select **IIS Application** to run the **Web Application**

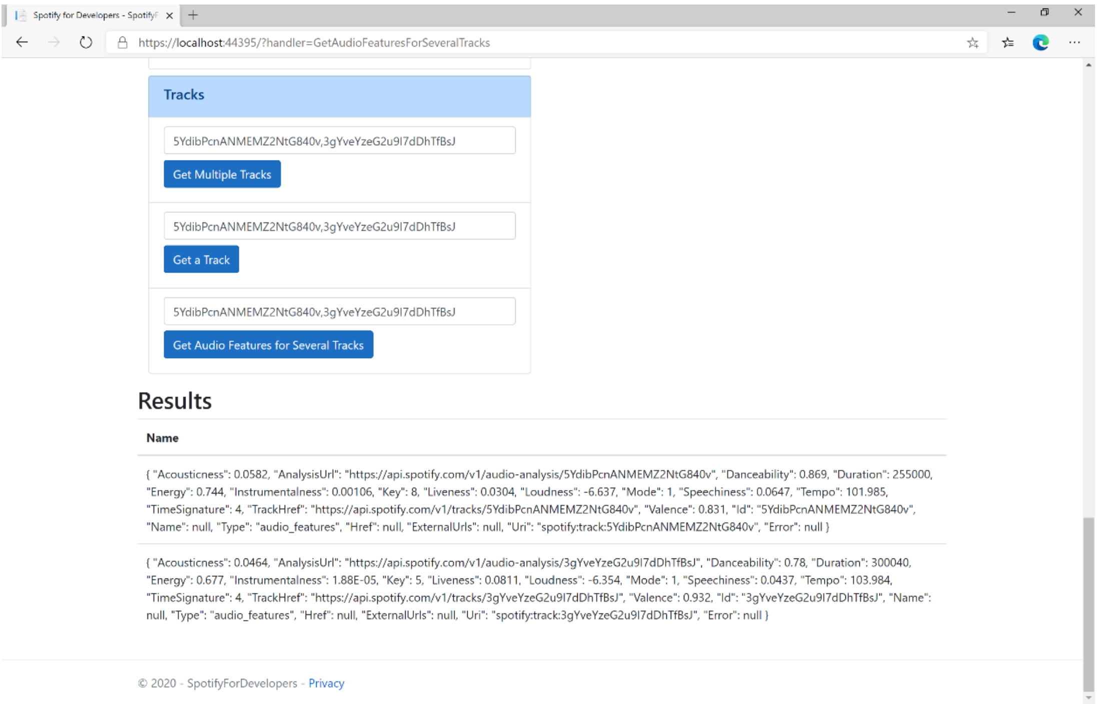
Step 9

Once the **Web Application** is running and you select **Client Credentials Flow Login** and scroll down you should see something like the following:


Spotify for Developers - Spotify
https://localhost:44395/?handler=clientcredentials
Album Ids
Get Multiple Albums
Album Id
Get an Album
Album Id
Get an Album's Tracks
Tracks
Track Ids
Get Multiple Tracks
Track Id
Get a Track
Track Ids
Get Audio Features for Several Tracks
© 2020 - SpotifyForDevelopers - Privacy

Step 10

You can then enter an **Album Id** from **Get All New Releases** select **Get an Album's Tracks** and copy some **Track Ids** to use and select **Get Audio Features for Several Tracks** and scroll down to view **Results** like the following:


Spotify for Developers - Spotify
https://localhost:44395/?handler=GetAudioFeaturesForSeveralTracks
Tracks
Get Multiple Tracks
Get a Track
Get Audio Features for Several Tracks
© 2020 - SpotifyForDevelopers - Privacy

5YdibPcnANMEMZ2NtG840v,3gYveYzeG2u9I7dDhTfBsJ

5YdibPcnANMEMZ2NtG840v,3gYveYzeG2u9I7dDhTfBsJ

5YdibPcnANMEMZ2NtG840v,3gYveYzeG2u9I7dDhTfBsJ

Results

Name
{ "Acousticness": 0.0582, "AnalysisUrl": "https://api.spotify.com/v1/audio-analysis/5YdibPcnANMEMZ2NtG840v", "Danceability": 0.869, "Duration": 255000, "Energy": 0.744, "Instrumentalness": 0.00106, "Key": 8, "Liveness": 0.0304, "Loudness": -6.637, "Mode": 1, "Speechiness": 0.0647, "Tempo": 101.985, "TimeSignature": 4, "TrackHref": "https://api.spotify.com/v1/tracks/5YdibPcnANMEMZ2NtG840v", "Valence": 0.831, "Id": "5YdibPcnANMEMZ2NtG840v", "Name": null, "Type": "audio_features", "Href": null, "ExternalUrls": null, "Uri": "spotify:track:5YdibPcnANMEMZ2NtG840v", "Error": null }
{ "Acousticness": 0.0464, "AnalysisUrl": "https://api.spotify.com/v1/audio-analysis/3gYveYzeG2u9I7dDhTfBsJ", "Danceability": 0.78, "Duration": 300040, "Energy": 0.677, "Instrumentalness": 1.88E-05, "Key": 5, "Liveness": 0.0811, "Loudness": -6.354, "Mode": 1, "Speechiness": 0.0437, "Tempo": 103.984, "TimeSignature": 4, "TrackHref": "https://api.spotify.com/v1/tracks/3gYveYzeG2u9I7dDhTfBsJ", "Valence": 0.932, "Id": "3gYveYzeG2u9I7dDhTfBsJ", "Name": null, "Type": "audio_features", "Href": null, "ExternalUrls": null, "Uri": "spotify:track:3gYveYzeG2u9I7dDhTfBsJ", "Error": null }

Step 11

■

You can stop the **web application** in **Visual Studio 2019** by selecting the **Stop debugging** button

Step 12

×

You can choose to exit **Visual Studio 2019** by selecting the **Close** button in the top right of the **application** as that completes this part of the workshop

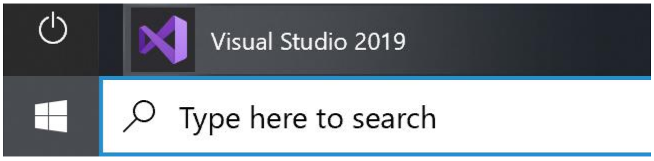
Get Audio Features for a Track

Get audio feature information for a single track identified by its unique Spotify Id.

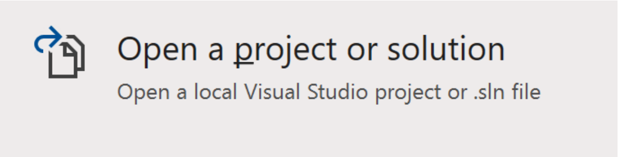
GET https://api.spotify.com/v1/audio-features/{id}	
Header	
Authorization	Valid Access Token from Spotify Accounts service
Path Parameter	
id	Spotify Id of the track

Header	Response
Success	
HTTP Status 200 OK	Audio Features Object
Error	
Error Code	Error Object

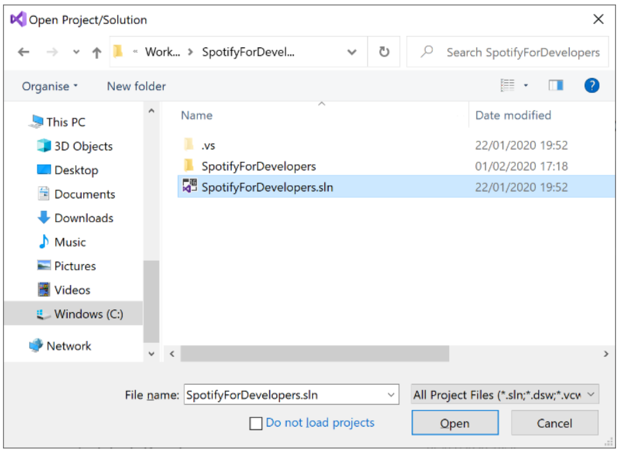
Step 1



If you chose to close **Visual Studio 2019** previously, in **Windows 10** choose **Start**, and then from the **Start Menu** find and select **Visual Studio 2019**


Open a project or solution
Open a local Visual Studio project or .sln file

Once done, from the **Get started** screen for **Visual Studio 2019** select **Open a project or solution**

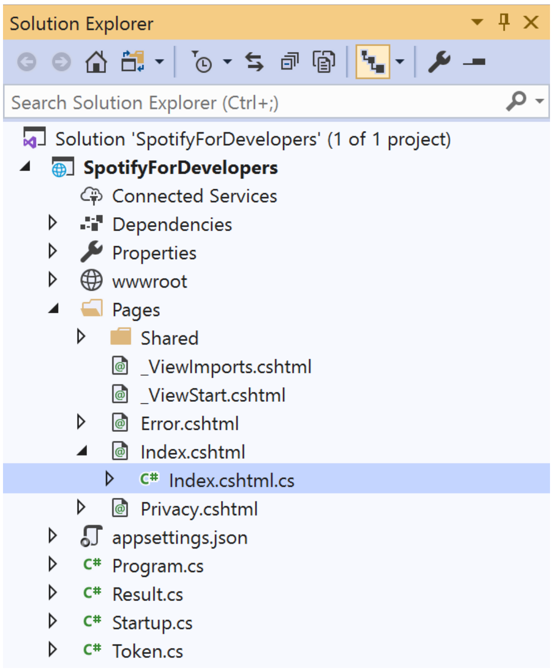


Name	Date modified
.vs	22/01/2020 19:52
SpotifyForDevelopers	01/02/2020 17:18
SpotifyForDevelopers.sln	22/01/2020 19:52

File name: SpotifyForDevelopers.sln

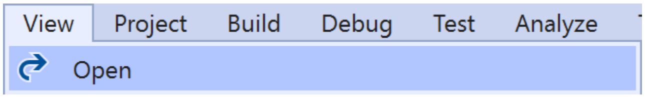
Then locate and select **SpotifyForDevelopers.sln** and select **Open** if you don't have this file already then please follow the previous parts of the workshop including **Getting Started**, **Authorisation Guide**, **Search & Browse** and **Artists, Albums & Tracks**

Step 2



Once opened, in the **Solution Explorer** open the **Pages** section, then open the **Index.cshtml** section and select **Index.cshtml.cs**

Step 3



Then from the **Menu** choose **View** and then **Open**

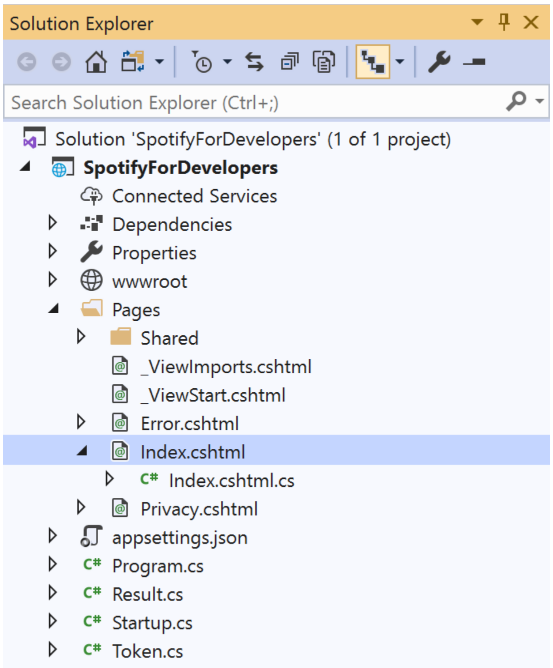
Step 4

In the **Code View** for **Index.cshtml.cs** below the **method** for `public async Task<IActionResult> OnPostGetAudioFeaturesForSeveralTracksAsync(...) { ... }` enter the following **method**:

```
public async Task<IActionResult> OnPostGetAudioFeaturesForTrackAsync(string value)
{
    LoadToken();
    var result = await Api.GetTrackAudioFeaturesAsync(value);
    if (result != null)
    {
        Results = new List<Result>() { new Result()
        {
            Name = System.Text.Json.JsonSerializer.Serialize(result,
                new System.Text.Json.JsonSerializerOptions()
                { WriteIndented = true })
        }};
    }
    return Page();
}
```

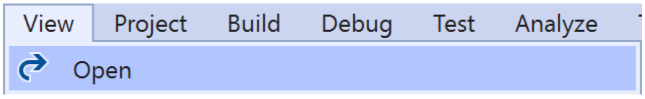
The **method** for `OnPostGetAudioFeaturesForTrackAsync` is used to get **audio features** for **track** by **Track Id** on Spotify with the `value` and populate the **property** for `Results` accordingly.

Step 5



In the **Solution Explorer** in the **Pages** section select **Index.cshtml**

Step 6



Then from the **Menu** choose **View** and then **Open**

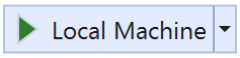
Step 7

Once in the **Code View** for **Index.cshtml** above `<!-- Tracks -->` enter the following:

```
<li class="list-group-item">
    <form asp-page-handler="GetAudioFeaturesForTrack" method="post">
        <input asp-for="Value" placeholder="Track Id" class="form-control mb-2" />
        <button class="btn btn-primary mb-2">
            Get Audio Features for a Track
        </button>
    </form>
</li>
```

This `form` will **post** to the **method** for `OnPostGetAudioFeaturesForTrackAsync` with the `Value` as **Track Id** and will output to the **Results**.

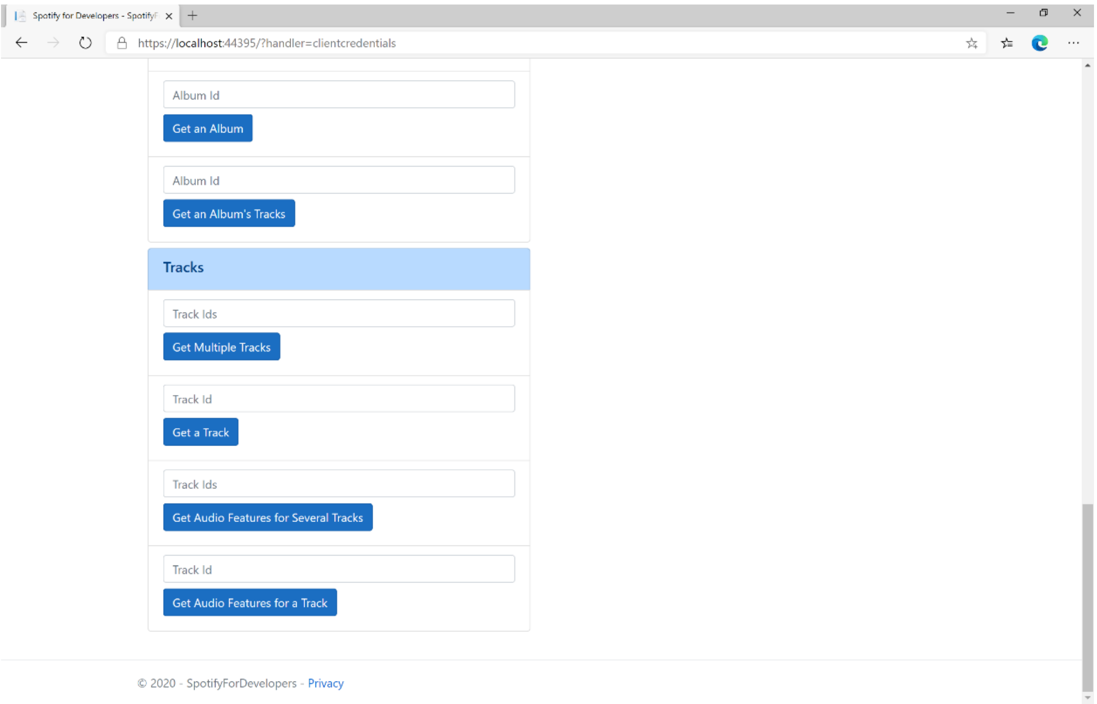
Step 8



Finally, in **Visual Studio 2019** select **IIS Application** to run the **Web Application**

Step 9

Once the **Web Application** is running and you select **Client Credentials Flow Login** and scroll down you should see something like the following:



Spotify for Developers - Spotify
https://localhost:44395/?handler=clientcredentials

Album Id

Get an Album

Album Id

Get an Album's Tracks

Tracks

Track Ids

Get Multiple Tracks

Track Id

Get a Track

Track Ids

Get Audio Features for Several Tracks

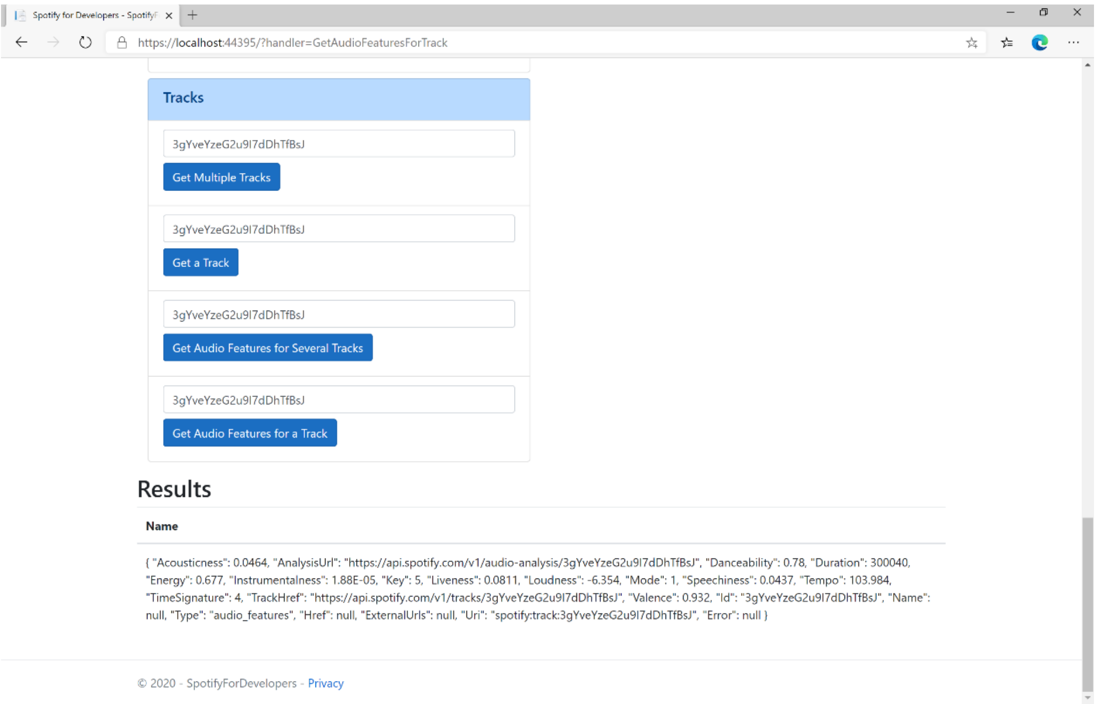
Track Id

Get Audio Features for a Track

© 2020 - SpotifyForDevelopers - Privacy

Step 10

You can then enter an **Album Id** from **Get All New Releases** select **Get an Album's Tracks** and copy a **Track Id** to use and select **Get Audio Features for a Track** and scroll down to view **Results** like the following:



Spotify for Developers - Spotify
https://localhost:44395/?handler=GetAudioFeaturesForTrack

Tracks

3gYveYzeG2u9I7dDhTfBsJ

Get Multiple Tracks

3gYveYzeG2u9I7dDhTfBsJ

Get a Track

3gYveYzeG2u9I7dDhTfBsJ

Get Audio Features for Several Tracks

3gYveYzeG2u9I7dDhTfBsJ

Get Audio Features for a Track

Results

Name
{ "Acousticness": 0.0464, "AnalysisUrl": "https://api.spotify.com/v1/audio-analysis/3gYveYzeG2u9I7dDhTfBsJ", "Danceability": 0.78, "Duration": 300040, "Energy": 0.677, "Instrumentalness": 1.88E-05, "Key": 5, "Liveness": 0.0811, "Loudness": -6.354, "Mode": 1, "Speechiness": 0.0437, "Tempo": 103.984, "TimeSignature": 4, "TrackHref": "https://api.spotify.com/v1/tracks/3gYveYzeG2u9I7dDhTfBsJ", "Valence": 0.932, "Id": "3gYveYzeG2u9I7dDhTfBsJ", "Name": null, "Type": "audio_features", "Href": null, "ExternalUrls": null, "Uri": "spotify:track:3gYveYzeG2u9I7dDhTfBsJ", "Error": null }

© 2020 - SpotifyForDevelopers - Privacy

Step 11

■

You can stop the **web application** in **Visual Studio 2019** by selecting the **Stop debugging** button

Step 12

✕

You can choose to exit **Visual Studio 2019** by selecting the **Close** button in the top right of the **application** as that completes this part of the workshop

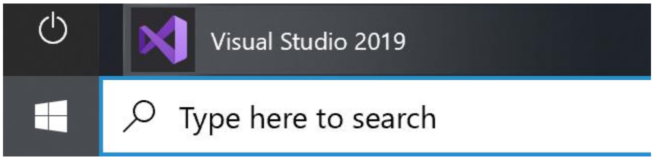
Get Audio Analysis for a Track

Get a detailed audio analysis for a single track identified by its unique Spotify Id.

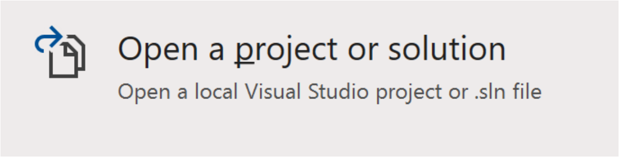
GET https://api.spotify.com/v1/audio-analysis/{id}	
Header	
Authorization	Valid Access Token from Spotify Accounts service
Path Parameter	
id	Spotify Id of the track

Header	Response
Success	
HTTP Status 200 OK	Audio Analysis Object
Error	
Error Code	Error Object

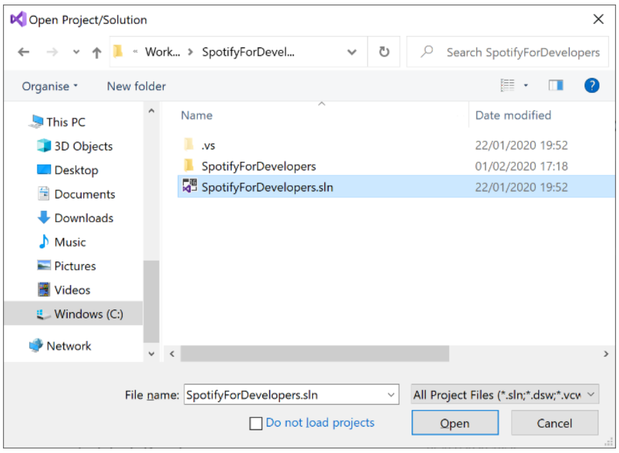
Step 1



If you chose to close **Visual Studio 2019** previously, in **Windows 10** choose **Start**, and then from the **Start Menu** find and select **Visual Studio 2019**

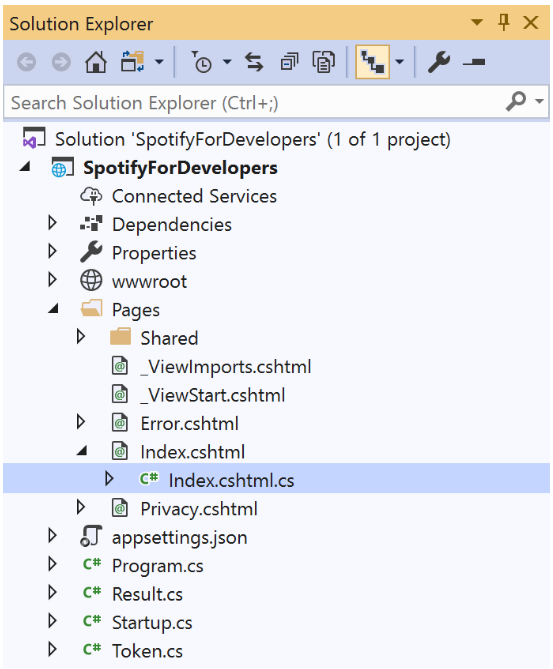

Open a project or solution
Open a local Visual Studio project or .sln file

Once done, from the **Get started** screen for **Visual Studio 2019** select **Open a project or solution**


Open Project/Solution
Work... > SpotifyForDevel...
Search SpotifyForDevelopers
Organise
New folder
This PC
3D Objects
Desktop
Documents
Downloads
Music
Pictures
Videos
Windows (C:)
Network
Name
Date modified
.vs 22/01/2020 19:52
SpotifyForDevelopers 01/02/2020 17:18
SpotifyForDevelopers.sln 22/01/2020 19:52
File name: SpotifyForDevelopers.sln
All Project Files (*.sln;*.dsw;*.vcw
Do not load projects
Open
Cancel

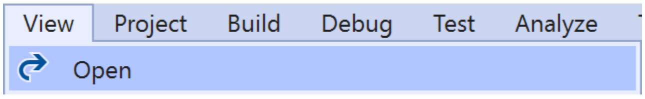
Then locate and select **SpotifyForDevelopers.sln** and select **Open** if you don't have this file already then please follow the previous parts of the workshop including **Getting Started**, **Authorisation Guide**, **Search & Browse** and **Artists, Albums & Tracks**

Step 2



Once opened, in the **Solution Explorer** open the **Pages** section, then open the **Index.cshtml** section and select **Index.cshtml.cs**

Step 3



Then from the **Menu** choose **View** and then **Open**

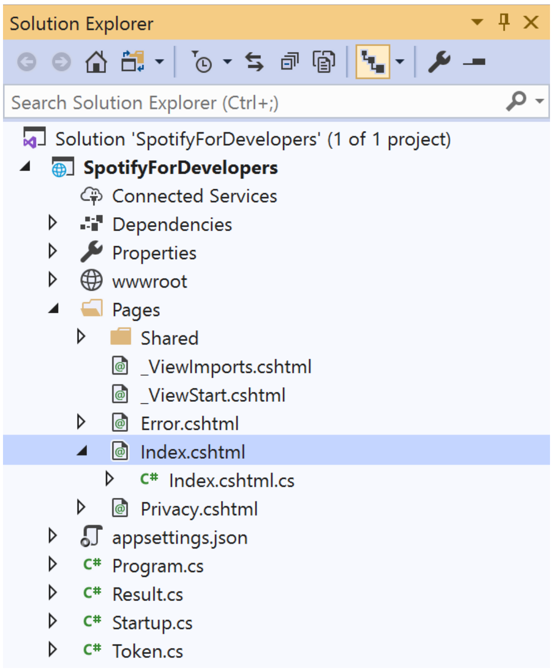
Step 4

In the **Code View** for **Index.cshtml.cs** below the **method** for `public async Task<IActionResult> OnPostGetAudioFeaturesForTrackAsync(...) { ... }` enter the following **method**:

```
public async Task<IActionResult> OnPostGetTrackAudioAnalysisAsync(string value)
{
    LoadToken();
    var result = await Api.GetTrackAudioAnalysisAsync(value);
    if (result != null)
    {
        Results = new List<Result>() { new Result()
        {
            Name = System.Text.Json.JsonSerializer.Serialize(result,
                new System.Text.Json.JsonSerializerOptions()
                { WriteIndented = true })
        }};
    }
    return Page();
}
```

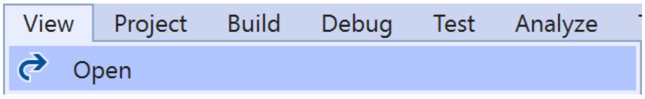
The **method** for `OnPostGetTrackAudioAnalysisAsync` is used to get **audio analysis** for **track** by **Track Id** on Spotify with the `value` and populate the **property** for `Results` accordingly.

Step 5



In the **Solution Explorer** in the **Pages** section select **Index.cshtml**

Step 6



Then from the **Menu** choose **View** and then **Open**

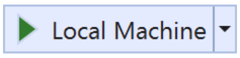
Step 7

Once in the **Code View** for **Index.cshtml** above `<!-- Tracks -->` enter the following:

```
<li class="list-group-item">
    <form asp-page-handler="GetTrackAudioAnalysis" method="post">
        <input asp-for="Value" placeholder="Track Id" class="form-control mb-2" />
        <button class="btn btn-primary mb-2">
            Get Audio Analysis for a Track
        </button>
    </form>
</li>
```

This `form` will **post** to the **method** for `GetTrackAudioAnalysis` with the `Value` as **Track Id** and will output to the **Results**.

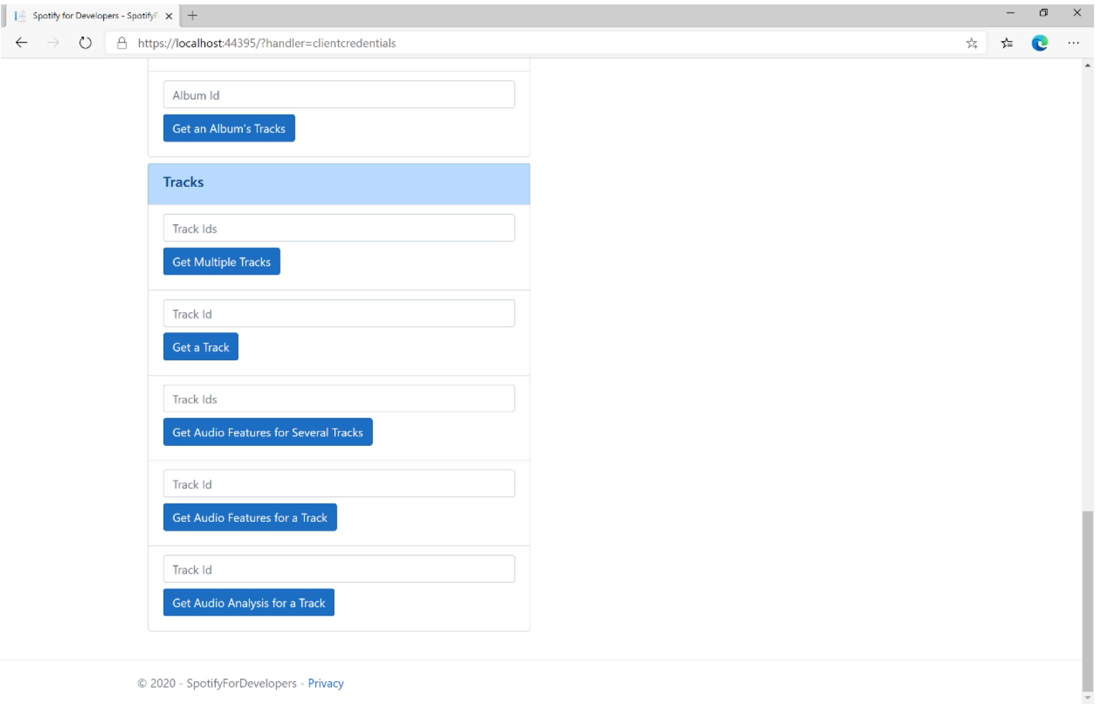
Step 8



Finally, in **Visual Studio 2019** select **IIS Application** to run the **Web Application**

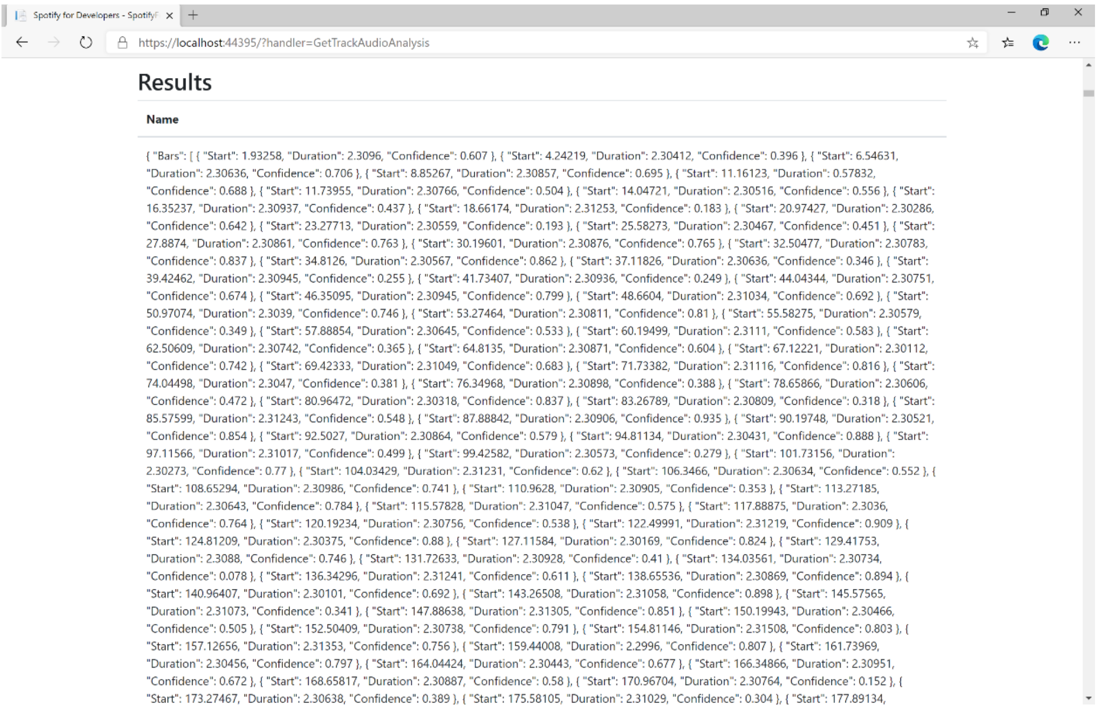
Step 9

Once the **Web Application** is running and you select **Client Credentials Flow Login** and scroll down you should see something like the following:


Spotify for Developers - Spotify
https://localhost:44395/?handler=clientcredentials
Album Id
Get an Album's Tracks
Tracks
Track Ids
Get Multiple Tracks
Track Id
Get a Track
Track Ids
Get Audio Features for Several Tracks
Track Id
Get Audio Features for a Track
Track Id
Get Audio Analysis for a Track
© 2020 - SpotifyForDevelopers - Privacy

Step 10

You can then enter an **Album Id** from **Get All New Releases** select **Get an Album's Tracks** and copy a **Track Id** to use and select **Get Audio Analysis for a Track** and scroll down to view **Results** like the following:


Spotify for Developers - Spotify
https://localhost:44395/?handler=GetTrackAudioAnalysis

Results

Name

```
{ "Bars": [ { "Start": 1.93258, "Duration": 2.3096, "Confidence": 0.607 }, { "Start": 4.24219, "Duration": 2.30412, "Confidence": 0.396 }, { "Start": 6.54631,
"Duration": 2.30636, "Confidence": 0.706 }, { "Start": 8.85267, "Duration": 2.30857, "Confidence": 0.695 }, { "Start": 11.16123, "Duration": 0.57832,
"Confidence": 0.688 }, { "Start": 11.73955, "Duration": 2.30766, "Confidence": 0.504 }, { "Start": 14.04721, "Duration": 2.30516, "Confidence": 0.556 }, { "Start":
16.35237, "Duration": 2.30937, "Confidence": 0.437 }, { "Start": 18.66174, "Duration": 2.31253, "Confidence": 0.183 }, { "Start": 20.97427, "Duration": 2.30286,
"Confidence": 0.642 }, { "Start": 23.27713, "Duration": 2.30559, "Confidence": 0.193 }, { "Start": 25.58273, "Duration": 2.30467, "Confidence": 0.451 }, { "Start":
27.8874, "Duration": 2.30861, "Confidence": 0.763 }, { "Start": 30.19601, "Duration": 2.30876, "Confidence": 0.765 }, { "Start": 32.50477, "Duration": 2.30783,
"Confidence": 0.837 }, { "Start": 34.8126, "Duration": 2.30567, "Confidence": 0.862 }, { "Start": 37.11826, "Duration": 2.30636, "Confidence": 0.346 }, { "Start":
39.42462, "Duration": 2.30945, "Confidence": 0.255 }, { "Start": 41.73407, "Duration": 2.30936, "Confidence": 0.249 }, { "Start": 44.04344, "Duration": 2.30751,
"Confidence": 0.674 }, { "Start": 46.35095, "Duration": 2.30945, "Confidence": 0.799 }, { "Start": 48.6604, "Duration": 2.31034, "Confidence": 0.692 }, { "Start":
50.97074, "Duration": 2.3039, "Confidence": 0.746 }, { "Start": 53.27464, "Duration": 2.30811, "Confidence": 0.81 }, { "Start": 55.58275, "Duration": 2.30579,
"Confidence": 0.349 }, { "Start": 57.88854, "Duration": 2.30645, "Confidence": 0.533 }, { "Start": 60.19499, "Duration": 2.3111, "Confidence": 0.583 }, { "Start":
62.50609, "Duration": 2.30742, "Confidence": 0.365 }, { "Start": 64.8135, "Duration": 2.30871, "Confidence": 0.604 }, { "Start": 67.12221, "Duration": 2.30112,
"Confidence": 0.742 }, { "Start": 69.42333, "Duration": 2.31049, "Confidence": 0.683 }, { "Start": 71.73382, "Duration": 2.31116, "Confidence": 0.816 }, { "Start":
74.04498, "Duration": 2.3047, "Confidence": 0.381 }, { "Start": 76.34968, "Duration": 2.30898, "Confidence": 0.388 }, { "Start": 78.65866, "Duration": 2.30606,
"Confidence": 0.472 }, { "Start": 80.96472, "Duration": 2.30318, "Confidence": 0.837 }, { "Start": 83.26789, "Duration": 2.30809, "Confidence": 0.318 }, { "Start":
85.57599, "Duration": 2.31243, "Confidence": 0.548 }, { "Start": 87.88842, "Duration": 2.30906, "Confidence": 0.935 }, { "Start": 90.19748, "Duration": 2.30521,
"Confidence": 0.854 }, { "Start": 92.5027, "Duration": 2.30864, "Confidence": 0.579 }, { "Start": 94.81134, "Duration": 2.30431, "Confidence": 0.888 }, { "Start":
97.11566, "Duration": 2.31017, "Confidence": 0.499 }, { "Start": 99.42582, "Duration": 2.30573, "Confidence": 0.279 }, { "Start": 101.73156, "Duration":
2.30273, "Confidence": 0.77 }, { "Start": 104.03429, "Duration": 2.31231, "Confidence": 0.62 }, { "Start": 106.3466, "Duration": 2.30634, "Confidence": 0.552 }, {
"Start": 108.65294, "Duration": 2.30986, "Confidence": 0.741 }, { "Start": 110.9628, "Duration": 2.30905, "Confidence": 0.353 }, { "Start": 113.27185,
"Duration": 2.30643, "Confidence": 0.784 }, { "Start": 115.57828, "Duration": 2.31047, "Confidence": 0.575 }, { "Start": 117.88875, "Duration": 2.3036,
"Confidence": 0.764 }, { "Start": 120.19234, "Duration": 2.30756, "Confidence": 0.538 }, { "Start": 122.49991, "Duration": 2.31219, "Confidence": 0.909 }, {
"Start": 124.81209, "Duration": 2.30375, "Confidence": 0.88 }, { "Start": 127.11584, "Duration": 2.30169, "Confidence": 0.824 }, { "Start": 129.41753,
"Duration": 2.3088, "Confidence": 0.746 }, { "Start": 131.72633, "Duration": 2.30928, "Confidence": 0.41 }, { "Start": 134.03561, "Duration": 2.30734,
"Confidence": 0.078 }, { "Start": 136.34296, "Duration": 2.31241, "Confidence": 0.611 }, { "Start": 138.65536, "Duration": 2.30869, "Confidence": 0.894 }, {
"Start": 140.96407, "Duration": 2.30101, "Confidence": 0.692 }, { "Start": 143.26508, "Duration": 2.31058, "Confidence": 0.898 }, { "Start": 145.57565,
"Duration": 2.31073, "Confidence": 0.341 }, { "Start": 147.88638, "Duration": 2.31305, "Confidence": 0.851 }, { "Start": 150.19943, "Duration": 2.30466,
"Confidence": 0.505 }, { "Start": 152.50409, "Duration": 2.30738, "Confidence": 0.791 }, { "Start": 154.81146, "Duration": 2.31508, "Confidence": 0.803 }, {
"Start": 157.12656, "Duration": 2.31353, "Confidence": 0.756 }, { "Start": 159.44008, "Duration": 2.2996, "Confidence": 0.807 }, { "Start": 161.73969,
"Duration": 2.30456, "Confidence": 0.797 }, { "Start": 164.04424, "Duration": 2.30443, "Confidence": 0.677 }, { "Start": 166.34866, "Duration": 2.30951,
"Confidence": 0.672 }, { "Start": 168.65817, "Duration": 2.30887, "Confidence": 0.58 }, { "Start": 170.96704, "Duration": 2.30764, "Confidence": 0.152 }, {
"Start": 173.27467, "Duration": 2.30638, "Confidence": 0.389 }, { "Start": 175.58105, "Duration": 2.31029, "Confidence": 0.304 }, { "Start": 177.89134,
```

Step 11

■

You can stop the **web application** in **Visual Studio 2019** by selecting the **Stop debugging** button

Step 12

×

You can exit **Visual Studio 2019** by selecting the **Close** button in the top right of the **application** as that completes this part of the workshop

Episodes & Shows

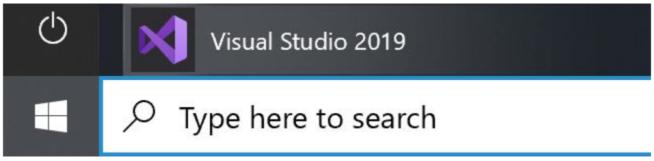
Get an Episode

Get Spotify catalogue information for a single episode identified by its unique Spotify Id.

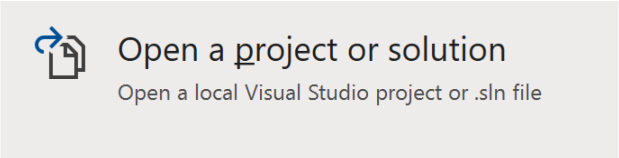
GET https://api.spotify.com/v1/episodes/{id}	
Header	
Authorization	Valid Access Token or User Token from Spotify Accounts service with **user-read-playback-position** scope to read user's resume points on episode objects
Path Parameter	
id	The Spotify Id for the episode
market	An ISO 3166-1 alpha-2 country code. If a country code is specified, only shows and episodes that are available in that market will be returned. If a valid user access token is specified in the request header, the country associated with the user account will take priority over this parameter. If neither market nor user country are provided, the content is considered unavailable for the client. Users can view the country that is associated with their account in the account settings.

Header	**Response**
Success	
HTTP Status 200 OK	Episode Object
Error	
Error Code	Error Object

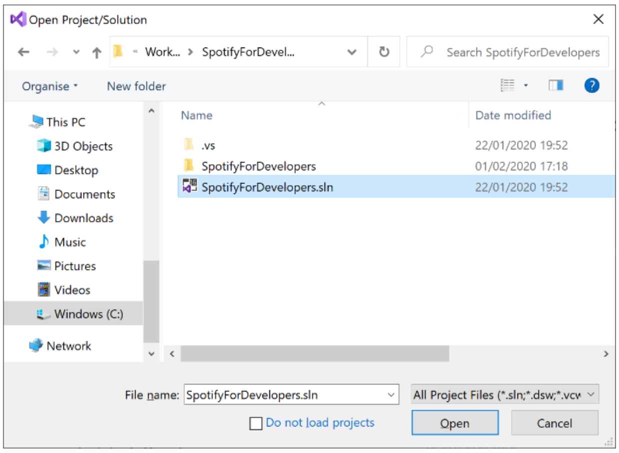
Step 1



In **Windows 10** choose **Start**, and then from the **Start Menu** find and select **Visual Studio 2019**

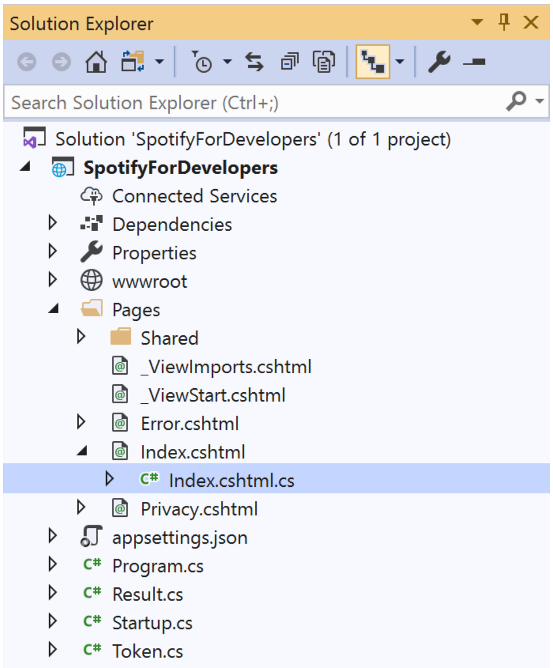

Open a project or solution
Open a local Visual Studio project or .sln file

Once done, from the **Get started** screen for **Visual Studio 2019** select **Open a project or solution**



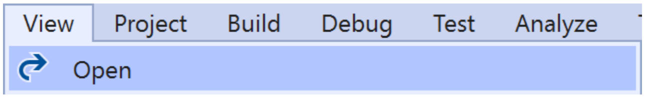
Then locate and select **SpotifyForDevelopers.sln** and select **Open** if you don't have this file already then please follow the previous parts of the workshop including **Getting Started**, **Authorisation Guide**, **Search & Browse** and **Artists, Albums & Tracks**

Step 2



Once opened, in the **Solution Explorer** open the **Pages** section, then open the **Index.cshtml** section and select **Index.cshtml.cs**

Step 3



Then from the **Menu** choose **View** and then **Open**

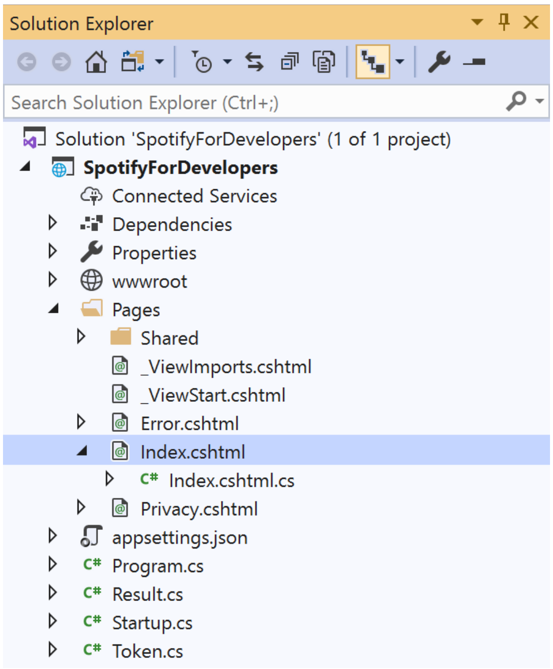
Step 4

In the **Code View** for **Index.cshtml.cs** below the **method** for `public async Task<IActionResult> OnPostGetTrackAudioAnalysisAsync(...) { ... }` enter the following **method**:

```
public async Task<IActionResult> OnPostGetEpisodeAsync(string value)
{
    LoadToken();
    var result = await Api.GetEpisodeAsync(value, country);
    if (result != null)
    {
        Results = new List<Result> { new Result()
        {
            Id = result.Id,
            Name = result.Name,
            Image = result?.Images?.FirstOrDefault()?.Url
        }};
    }
    return Page();
}
```

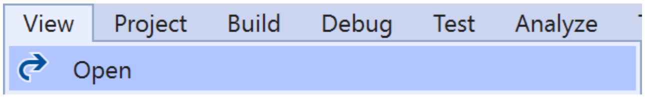
The **method** for `OnPostGetEpisodeAsync` is used to get an **episode** by **Episode Id** on Spotify with the `value` and populate the **property** for `Results` accordingly.

Step 5



In the **Solution Explorer** in the **Pages** section select **Index.cshtml**

Step 6



Then from the **Menu** choose **View** and then **Open**

Step 7

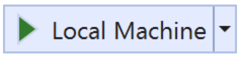
Once in the **Code View** for **Index.cshtml** above `<!-- Spotify Web API App Authorisation -->` enter the following:

```
<ul class="list-group mb-2">
    <li class="list-group-item list-group-item-primary">
        <h5 class="list-group-item-heading">Episodes</h5>
    </li>
    <li class="list-group-item">
        <form asp-page-handler="GetEpisode" method="post">
            <input asp-for="Value" placeholder="Episode Id" class="form-control mb-2" />
            <button class="btn btn-primary mb-2">
                Get an Episode
            </button>
        </form>
    </li>

    <!-- Episodes -->
</ul>
```

This `form` will **post** to the **method** for `GetEpisode` with the `Value` as **Episode Id** and will output to the **Results**.

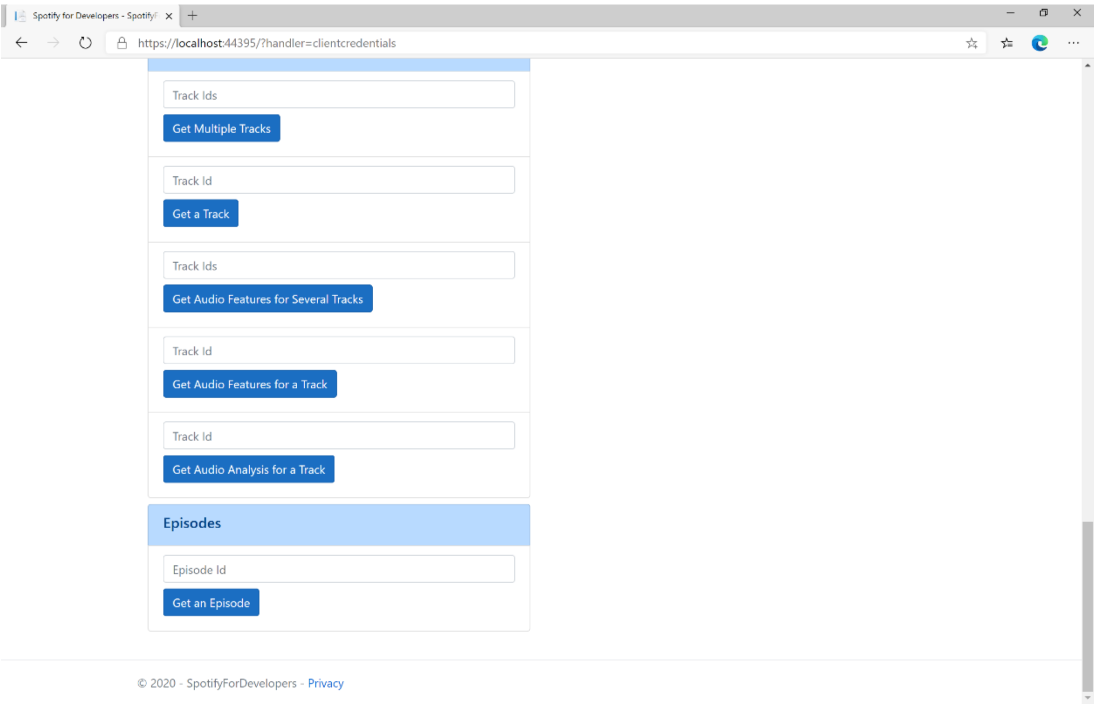
Step 8



Finally, in **Visual Studio 2019** select **IIS Application** to run the **Web Application**

Step 9

Once the **Web Application** is running and you select **Client Credentials Flow Login** and scroll down you should see something like the following:



Spotify for Developers - Spotify

https://localhost:44395/?handler=clientcredentials

Track Ids

Get Multiple Tracks

Track Id

Get a Track

Track Ids

Get Audio Features for Several Tracks

Track Id

Get Audio Features for a Track

Track Id

Get Audio Analysis for a Track

Episodes

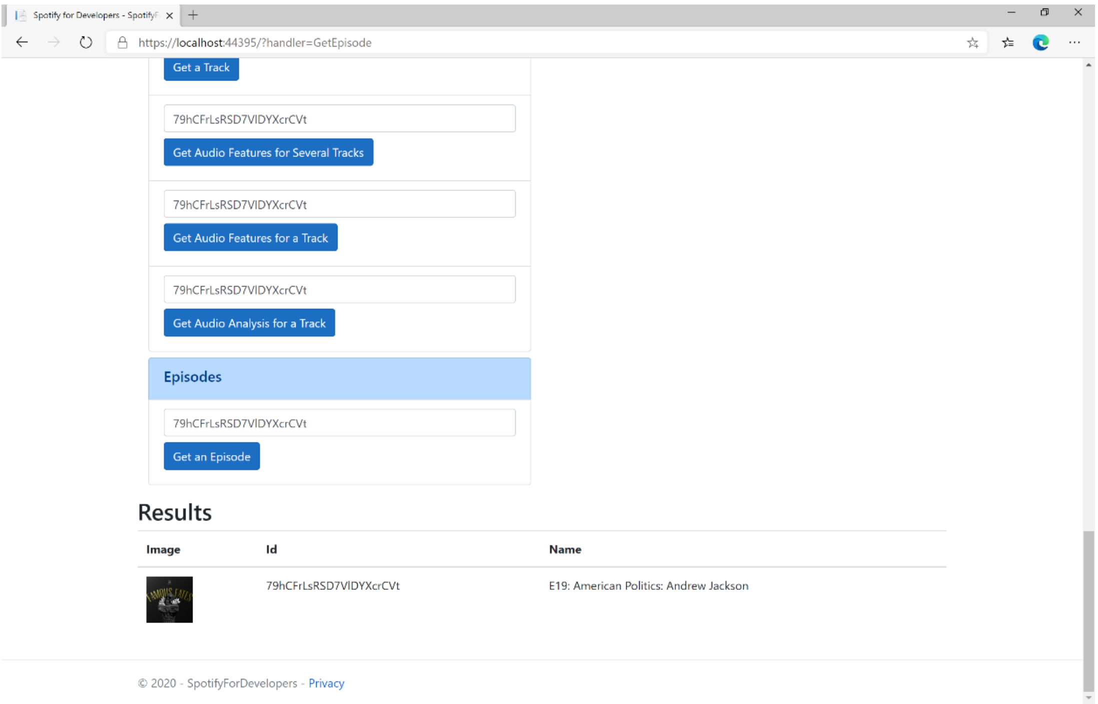
Episode Id

Get an Episode

© 2020 - SpotifyForDevelopers - Privacy

Step 10

You can get an **Episode Id** by using **Search for an Item** and pick **Episodes** then into **Query** enter a **Podcast Episode** then copy an **Episode Id** to use and select **Get an Episode** and scroll down to view **Results** like the following:



Spotify for Developers - Spotify

https://localhost:44395/?handler=GetEpisode

Get a Track

79hCFrLsRSD7VIDYXcrCVt

Get Audio Features for Several Tracks

79hCFrLsRSD7VIDYXcrCVt

Get Audio Features for a Track

79hCFrLsRSD7VIDYXcrCVt

Get Audio Analysis for a Track

Episodes

79hCFrLsRSD7VIDYXcrCVt

Get an Episode

Results

Image	Id	Name
	79hCFrLsRSD7VIDYXcrCVt	E19: American Politics: Andrew Jackson

© 2020 - SpotifyForDevelopers - Privacy

Step 11

You can stop the **web application** in **Visual Studio 2019** by selecting the **Stop debugging** button

Step 12

You can choose to exit **Visual Studio 2019** by selecting the **Close** button in the top right of the **application** as that completes this part of the workshop

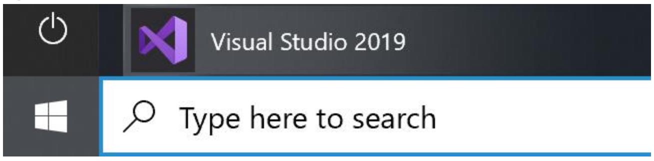
Get Multiple Episodes

Get Spotify catalogue information for several episodes based on their Spotify Ids.

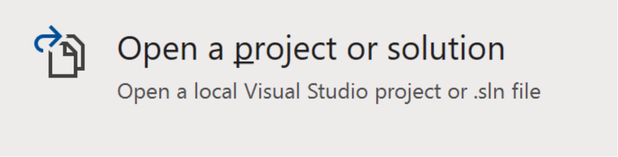
GET https://api.spotify.com/v1/episodes	
Header	
Authorization	Valid Access Token or User Token from Spotify Accounts service with **user-read-playback-position** scope to read user's resume points on episode objects
Path Parameter	
ids	A comma-separated list of the Spotify Ids for the episodes. Maximum: 50 IDs.
market	An ISO 3166-1 alpha-2 country code. If a country code is specified, only shows and episodes that are available in that market will be returned. If a valid user access token is specified in the request header, the country associated with the user account will take priority over this parameter. If neither market nor user country are provided, the content is considered unavailable for the client. Users can view the country that is associated with their account in the account settings.

Header	**Response**
Success	
HTTP Status 200 OK	Episodes as Array of Episode Object
Error	
Error Code	Error Object

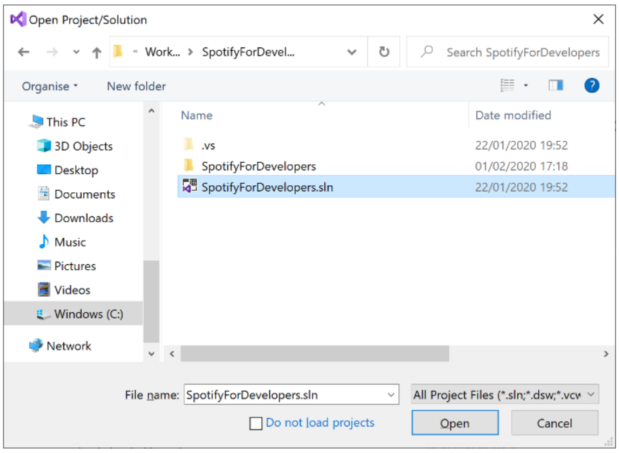
Step 1



If you chose to close **Visual Studio 2019** previously, in **Windows 10** choose **Start**, and then from the **Start Menu** find and select **Visual Studio 2019**

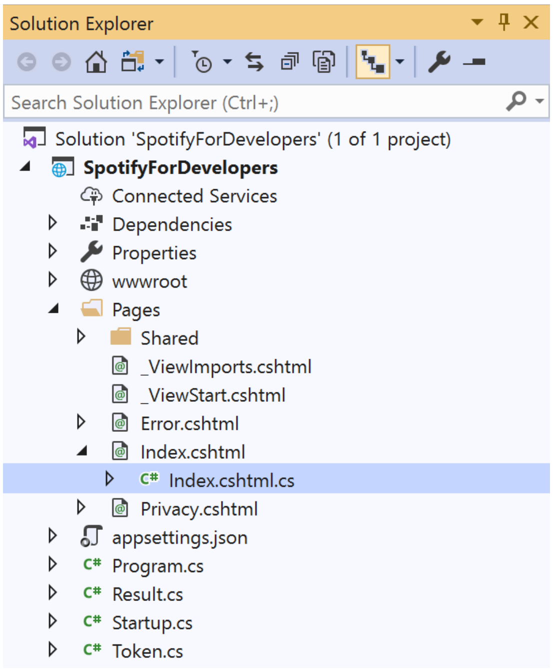

Open a project or solution
Open a local Visual Studio project or .sln file

Once done, from the **Get started** screen for **Visual Studio 2019** select **Open a project or solution**



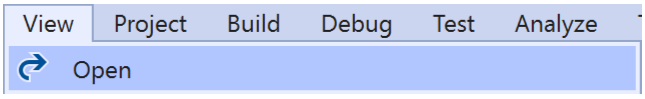
Then locate and select **SpotifyForDevelopers.sln** and select **Open** if you don't have this file already then please follow the previous parts of the workshop including **Getting Started**, **Authorisation Guide**, **Search & Browse**, **Artists, Albums & Tracks** and **Episodes & Shows**

Step 2



Once opened, in the **Solution Explorer** open the **Pages** section, then open the **Index.cshtml** section and select **Index.cshtml.cs**

Step 3



Then from the **Menu** choose **View** and then **Open**

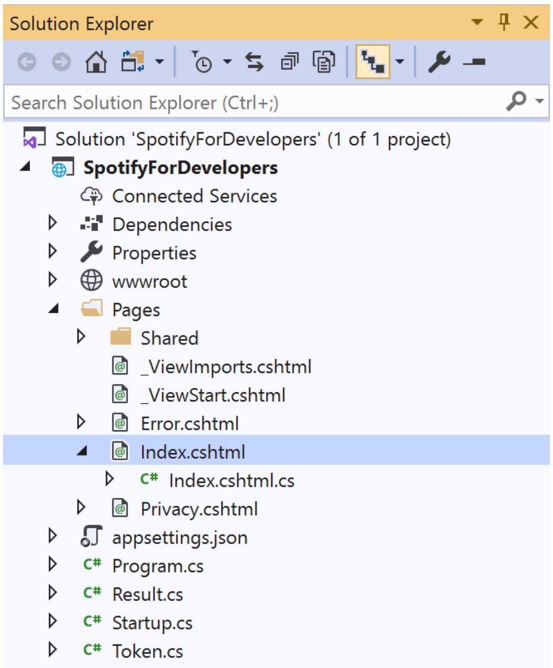
Step 4

In the **Code View** for **Index.cshtml.cs** below the **method** for `public async Task<IActionResult> OnPostGetEpisodeAsync(...) { ... }` enter the following **method**:

```
public async Task<IActionResult> OnPostGetMultipleEpisodesAsync(string value)
{
    LoadToken();
    var values = value.Split(",").ToList();
    var results = await Api.GetMultipleEpisodesAsync(values, country);
    if (results != null)
    {
        Results = results.Select(result => new Result()
        {
            Id = result.Id,
            Name = result.Name,
            Image = result?.Images?.FirstOrDefault()?.Url
        });
    }
    return Page();
}
```

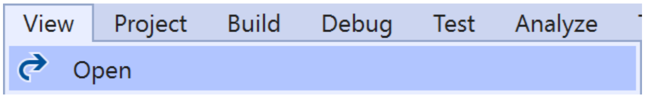
The **method** for `OnPostGetMultipleEpisodesAsync` is used to get **episodes** by multiple **Episode Ids** on Spotify with the `value` and populate the **property** for `Results` accordingly.

Step 5



In the **Solution Explorer** in the **Pages** section select **Index.cshtml**

Step 6



Then from the **Menu** choose **View** and then **Open**

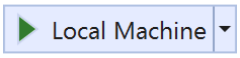
Step 7

Once in the **Code View** for **Index.cshtml** above `<!-- Episodes -->` enter the following:

```
<li class="list-group-item">
    <form asp-page-handler="GetMultipleEpisodes" method="post">
        <input asp-for="Value" placeholder="Episode Ids" class="form-control mb-2" />
        <button class="btn btn-primary mb-2">
            Get Multiple Episodes
        </button>
    </form>
</li>
```

This `form` will **post** to the **method** for `GetMultipleEpisodes` with the `Value` as **Episode Ids** and will output to the **Results**.

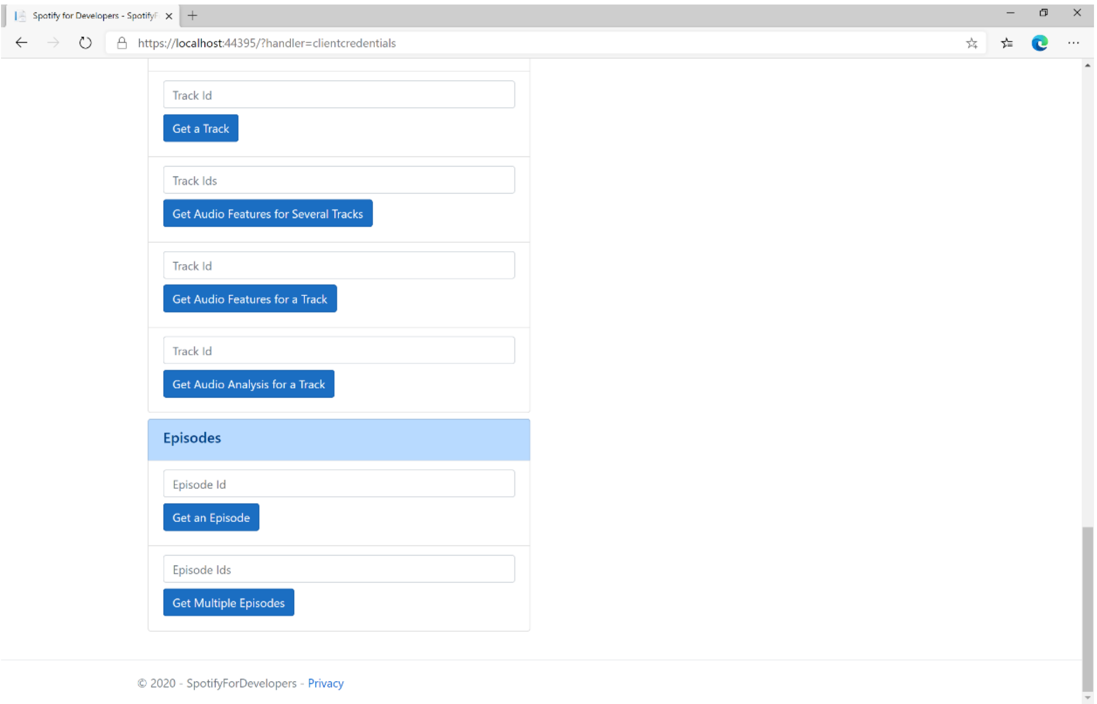
Step 8



Finally, in **Visual Studio 2019** select **IIS Application** to run the **Web Application**

Step 9

Once the **Web Application** is running and you select **Client Credentials Flow Login** and scroll down you should see something like the following:



https://localhost:44395/?handler=clientcredentials

Track Id

Get a Track

Track Ids

Get Audio Features for Several Tracks

Track Id

Get Audio Features for a Track

Track Id

Get Audio Analysis for a Track

Episodes

Episode Id

Get an Episode

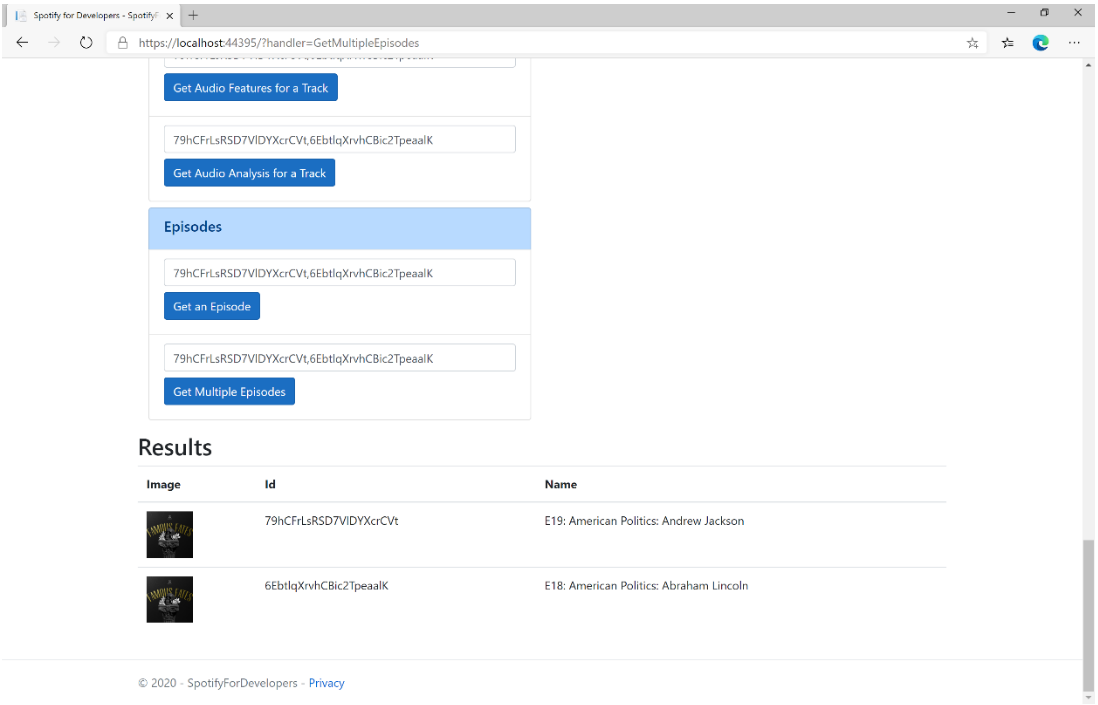
Episode Ids

Get Multiple Episodes

© 2020 - SpotifyForDevelopers - Privacy

Step 10

You can get some **Episode Ids** by using **Search for an Item** and pick **Episodes** then into **Query** enter some **Podcast Episodes** then copy some **Episode Ids** to use and select **Get Multiple Episodes** and scroll down to view **Results** like the following:



https://localhost:44395/?handler=GetMultipleEpisodes

Get Audio Features for a Track

79hCFrLsRSD7VlDYXcrCVt,6EbtlqXrvhCBic2TpeaalK

Get Audio Analysis for a Track

Episodes

79hCFrLsRSD7VlDYXcrCVt,6EbtlqXrvhCBic2TpeaalK

Get an Episode

79hCFrLsRSD7VlDYXcrCVt,6EbtlqXrvhCBic2TpeaalK

Get Multiple Episodes

Results

Image	Id	Name
	79hCFrLsRSD7VlDYXcrCVt	E19: American Politics: Andrew Jackson
	6EbtlqXrvhCBic2TpeaalK	E18: American Politics: Abraham Lincoln

© 2020 - SpotifyForDevelopers - Privacy

Step 11

You can stop the **web application** in **Visual Studio 2019** by selecting the **Stop debugging** button

Step 12

You can choose to exit **Visual Studio 2019** by selecting the **Close** button in the top right of the **application** as that completes this part of the workshop

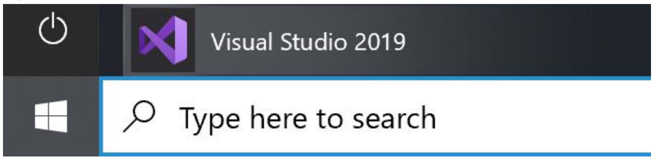
Get a Show

Get Spotify catalogue information for a single show identified by its unique Spotify Id.

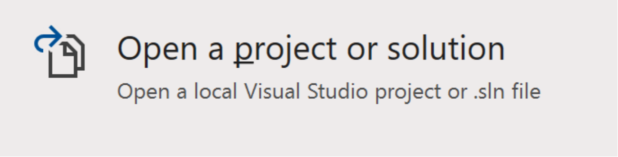
GET https://api.spotify.com/v1/shows/{id}	
Header	
Authorization	Valid Access Token or User Token from Spotify Accounts service with **user-read-playback-position** scope to read user's resume points on episode objects
Path Parameter	
id	The Spotify Id for the show
market	An ISO 3166-1 alpha-2 country code. If a country code is specified, only shows and episodes that are available in that market will be returned. If a valid user access token is specified in the request header, the country associated with the user account will take priority over this parameter. If neither market nor user country are provided, the content is considered unavailable for the client. Users can view the country that is associated with their account in the account settings.

Header	Response
Success	
HTTP Status 200 OK	Show Object
Error	
Error Code	Error Object

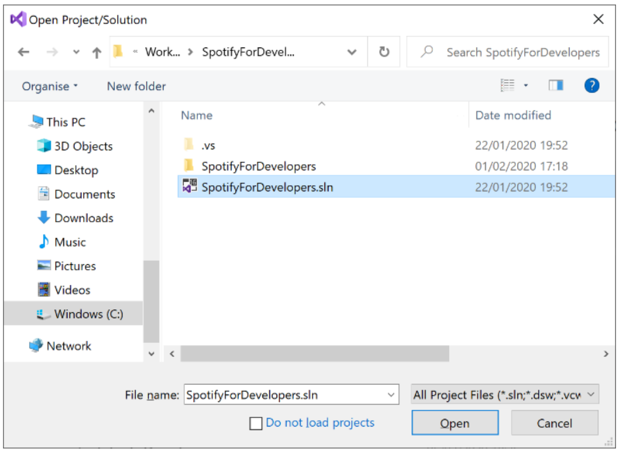
Step 1



If you chose to close **Visual Studio 2019** previously, in **Windows 10** choose **Start**, and then from the **Start Menu** find and select **Visual Studio 2019**


Open a project or solution
Open a local Visual Studio project or .sln file

Once done, from the **Get started** screen for **Visual Studio 2019** select **Open a project or solution**

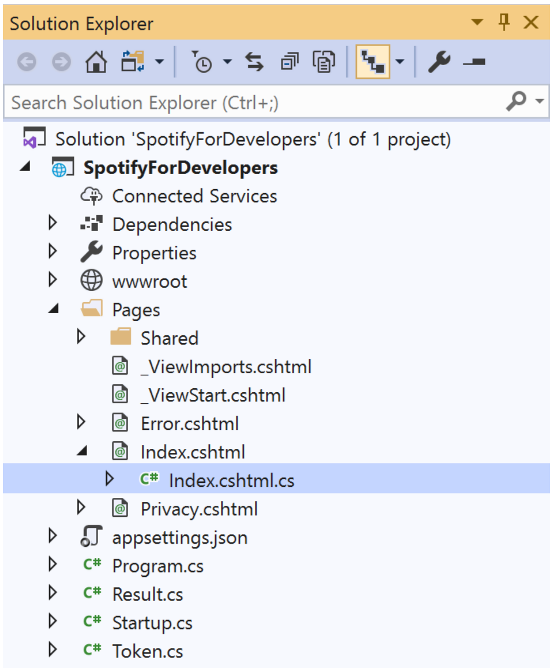


Name	Date modified
.vs	22/01/2020 19:52
SpotifyForDevelopers	01/02/2020 17:18
SpotifyForDevelopers.sln	22/01/2020 19:52

File name: SpotifyForDevelopers.sln

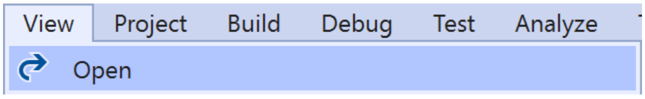
Then locate and select **SpotifyForDevelopers.sln** and select **Open** if you don't have this file already then please follow the previous parts of the workshop including **Getting Started**, **Authorisation Guide**, **Search & Browse**, **Artists, Albums & Tracks** and **Episodes & Shows**

Step 2



Once opened, in the **Solution Explorer** open the **Pages** section, then open the **Index.cshtml** section and select **Index.cshtml.cs**

Step 3



Then from the **Menu** choose **View** and then **Open**

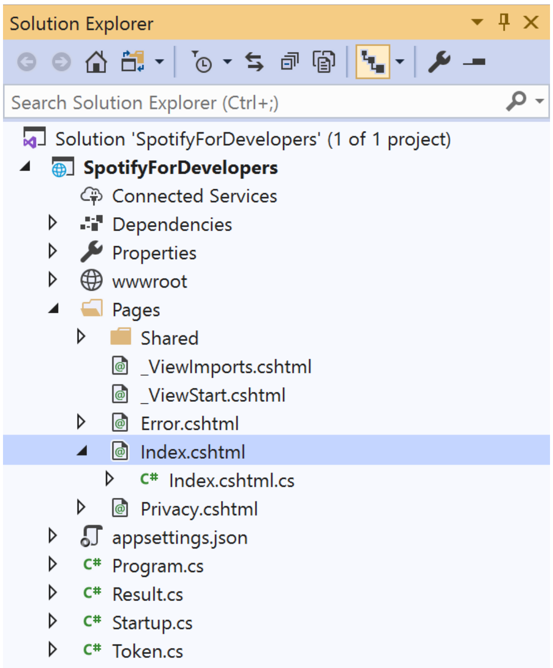
Step 4

In the **Code View** for **Index.cshtml.cs** below the **method** for `public async Task<IActionResult> OnPostGetMultipleEpisodesAsync(...) { ... }` enter the following **method**:

```
public async Task<IActionResult> OnPostGetShowAsync(string value)
{
    LoadToken();
    var result = await Api.GetShowAsync(value, country);
    if (result != null)
    {
        Results = new List<Result> { new Result()
        {
            Id = result.Id,
            Name = result.Name,
            Image = result?.Images?.FirstOrDefault()?.Url
        }};
    }
    return Page();
}
```

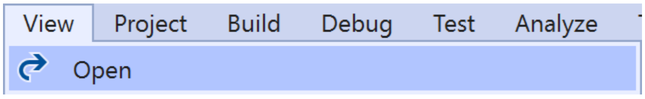
The **method** for `OnPostGetShowAsync` is used to get a **show** by **Show Id** on Spotify with the `value` and populate the **property** for `Results` accordingly.

Step 5



In the **Solution Explorer** in the **Pages** section select **Index.cshtml**

Step 6



Then from the **Menu** choose **View** and then **Open**

Step 7

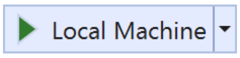
Once in the **Code View** for **Index.cshtml** above `<!-- Spotify Web API App Authorisation -->` enter the following:

```
<ul class="list-group mb-2">
    <li class="list-group-item list-group-item-primary">
        <h5 class="list-group-item-heading">Shows</h5>
    </li>
    <li class="list-group-item">
        <form asp-page-handler="GetShow" method="post">
            <input asp-for="Value" placeholder="Show Id" class="form-control mb-2" />
            <button class="btn btn-primary mb-2">
                Get a Show
            </button>
        </form>
    </li>

    <!-- Shows -->
</ul>
```

This `form` will **post** to the **method** for `GetShow` with the `Value` as **Show Id** and will output to the **Results**.

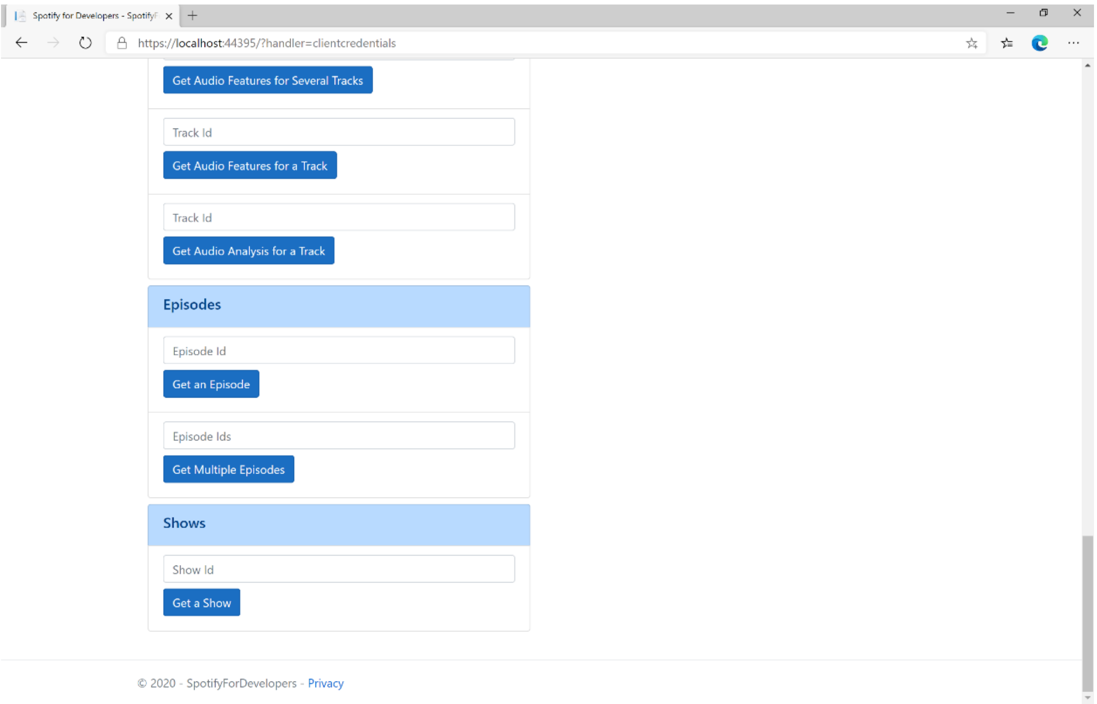
Step 8



Finally, in **Visual Studio 2019** select **IIS Application** to run the **Web Application**

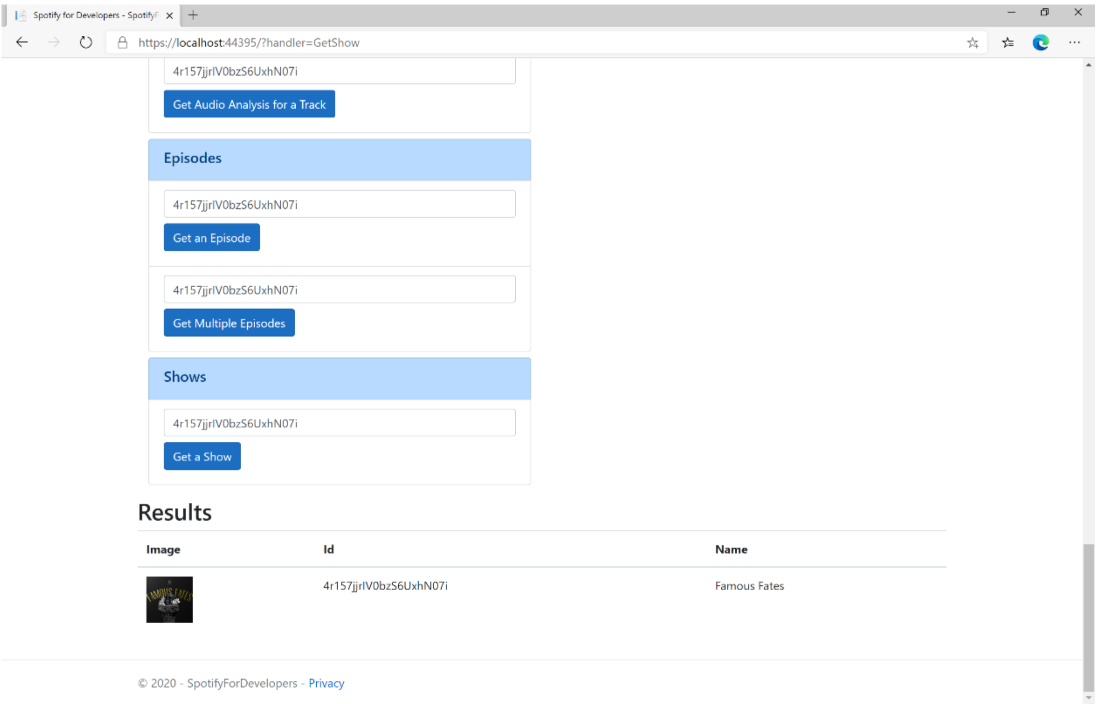
Step 9

Once the **Web Application** is running and you select **Client Credentials Flow Login** and scroll down you should see something like the following:


Spotify for Developers - Spotify
https://localhost:44395/?handler=clientcredentials
Get Audio Features for Several Tracks
Track Id
Get Audio Features for a Track
Track Id
Get Audio Analysis for a Track
Episodes
Episode Id
Get an Episode
Episode Ids
Get Multiple Episodes
Shows
Show Id
Get a Show
© 2020 - SpotifyForDevelopers - Privacy

Step 10

You can get a **Show Id** by using **Search for an Item** and pick **Shows** then into **Query** enter a **Podcast Name** then copy a **Show Id** to use and select **Get a Show** and scroll down to view **Results** like the following:


Spotify for Developers - Spotify
https://localhost:44395/?handler=GetShow
Get Audio Analysis for a Track
Episodes
Get an Episode
Get Multiple Episodes
Shows
Get a Show
© 2020 - SpotifyForDevelopers - Privacy

4r157jjrlV0bzS6UxhN07i

4r157jjrlV0bzS6UxhN07i

4r157jjrlV0bzS6UxhN07i

4r157jjrlV0bzS6UxhN07i

Results

Image	Id	Name
	4r157jjrlV0bzS6UxhN07i	Famous Fates

Step 11

■

You can stop the **web application** in **Visual Studio 2019** by selecting the **Stop debugging** button

Step 12

×

You can choose to exit **Visual Studio 2019** by selecting the **Close** button in the top right of the **application** as that completes this part of the workshop

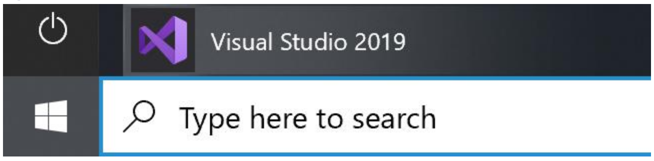
Get Multiple Shows

Get Spotify catalogue information for several shows based on their Spotify Ids.

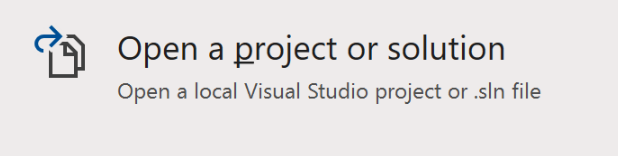
GET https://api.spotify.com/v1/episodes	
Header	
Authorization	Valid Access Token or User Token from Spotify Accounts service with **user-read-playback-position** scope to read user's resume points on episode objects
Path Parameter	
ids	A comma-separated list of the Spotify Ids for the shows. Maximum: 50 Ids.
market	An ISO 3166-1 alpha-2 country code. If a country code is specified, only shows and episodes that are available in that market will be returned. If a valid user access token is specified in the request header, the country associated with the user account will take priority over this parameter. If neither market nor user country are provided, the content is considered unavailable for the client. Users can view the country that is associated with their account in the account settings.

Header	**Response**
Success	
HTTP Status 200 OK	Shows as Array of Show Object
Error	
Error Code	Error Object

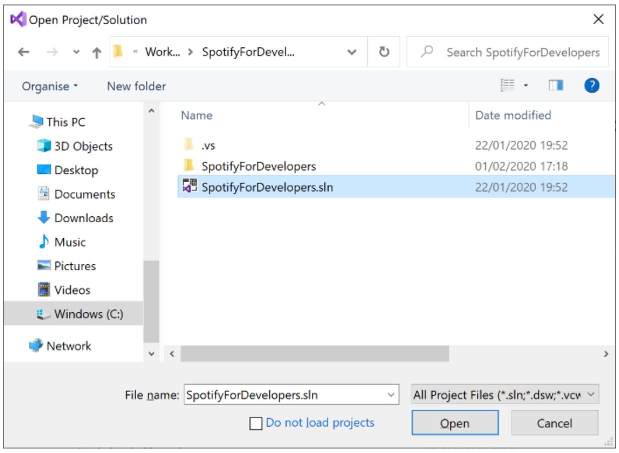
Step 1



If you chose to close **Visual Studio 2019** previously, in **Windows 10** choose **Start**, and then from the **Start Menu** find and select **Visual Studio 2019**


Open a project or solution
Open a local Visual Studio project or .sln file

Once done, from the **Get started** screen for **Visual Studio 2019** select **Open a project or solution**



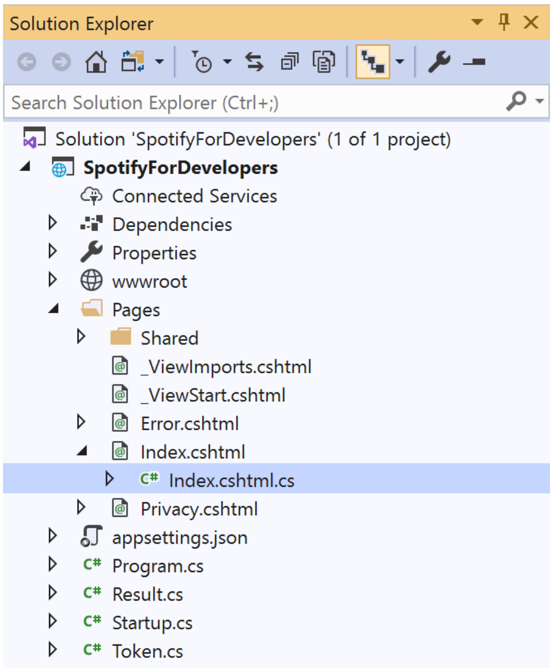
Name	Date modified
.vs	22/01/2020 19:52
SpotifyForDevelopers	01/02/2020 17:18
SpotifyForDevelopers.sln	22/01/2020 19:52

File name: SpotifyForDevelopers.sln

All Project Files (*.sln;*.dsw;*.vcw

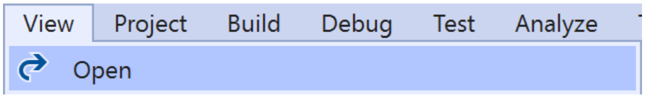
Then locate and select **SpotifyForDevelopers.sln** and select **Open** if you don't have this file already then please follow the previous parts of the workshop including **Getting Started**, **Authorisation Guide**, **Search & Browse**, **Artists, Albums & Tracks** and **Episodes & Shows**

Step 2



Once opened, in the **Solution Explorer** open the **Pages** section, then open the **Index.cshtml** section and select **Index.cshtml.cs**

Step 3



Then from the **Menu** choose **View** and then **Open**

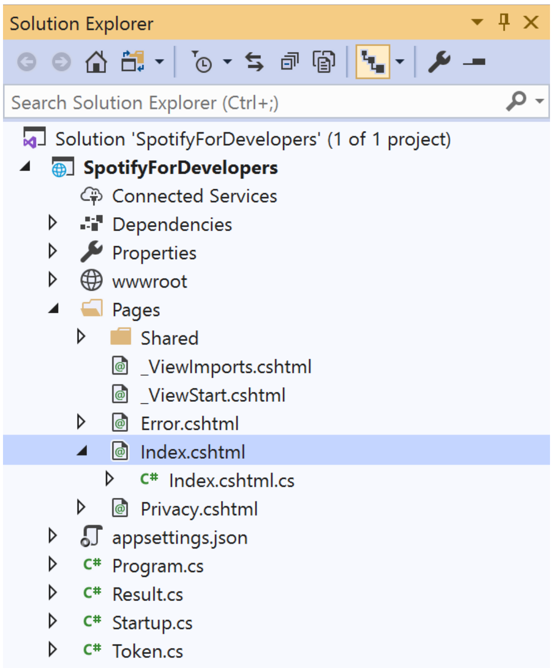
Step 4

In the **Code View** for **Index.cshtml.cs** below the **method** for `public async Task<IActionResult> OnPostGetShowAsync(...) { ... }` enter the following **method**:

```
public async Task<IActionResult> OnPostGetMultipleShowsAsync(string value)
{
    LoadToken();
    var values = value.Split(",").ToList();
    var results = await Api.GetMultipleShowsAsync(values, country);
    if (results != null)
    {
        Results = results.Select(result => new Result()
        {
            Id = result.Id,
            Name = result.Name,
            Image = result?.Images?.FirstOrDefault()?.Url
        });
    }
    return Page();
}
```

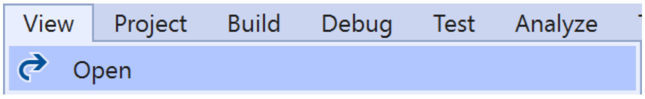
The **method** for `OnPostGetMultipleShowsAsync` is used to get **shows** by multiple **Show Ids** on Spotify with the `value` and populate the **property** for `Results` accordingly.

Step 5



In the **Solution Explorer** in the **Pages** section select **Index.cshtml**

Step 6



Then from the **Menu** choose **View** and then **Open**

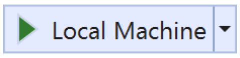
Step 7

Once in the **Code View** for **Index.cshtml** above `<!-- Shows -->` enter the following:

```
<li class="list-group-item">
    <form asp-page-handler="GetMultipleShows" method="post">
        <input asp-for="Value" placeholder="Show Ids" class="form-control mb-2" />
        <button class="btn btn-primary mb-2">
            Get Multiple Shows
        </button>
    </form>
</li>
```

This `form` will **post** to the **method** for `GetMultipleShows` with the `Value` as **Show Ids** and will output to the **Results**.

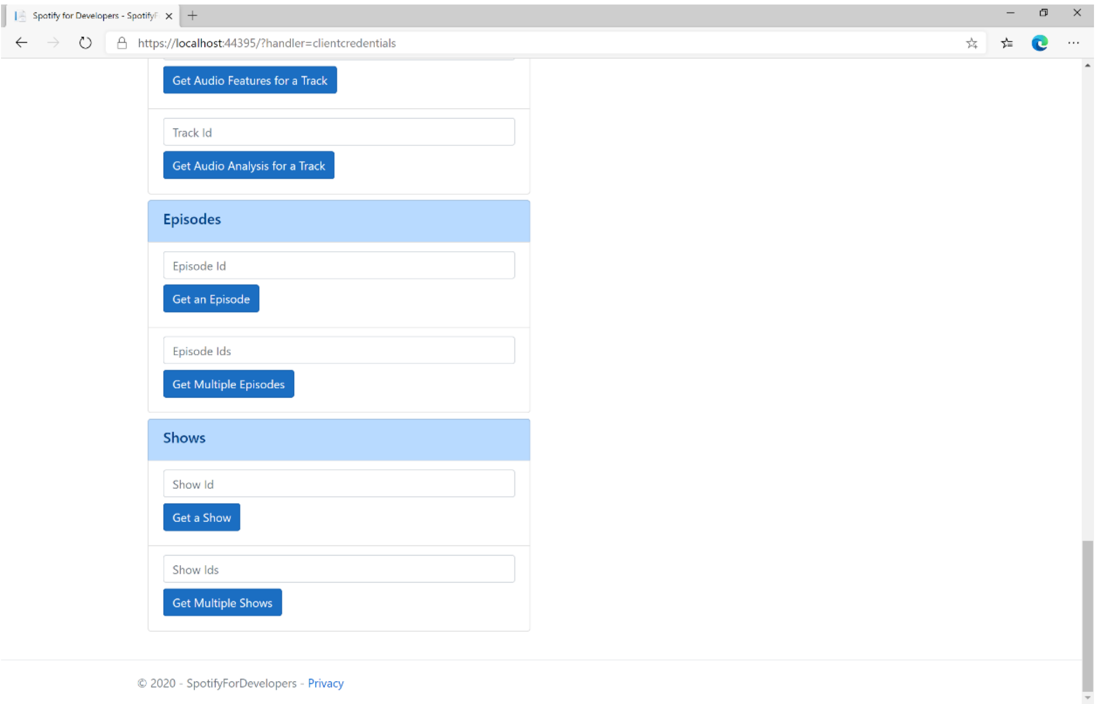
Step 8



Finally, in **Visual Studio 2019** select **IIS Application** to run the **Web Application**

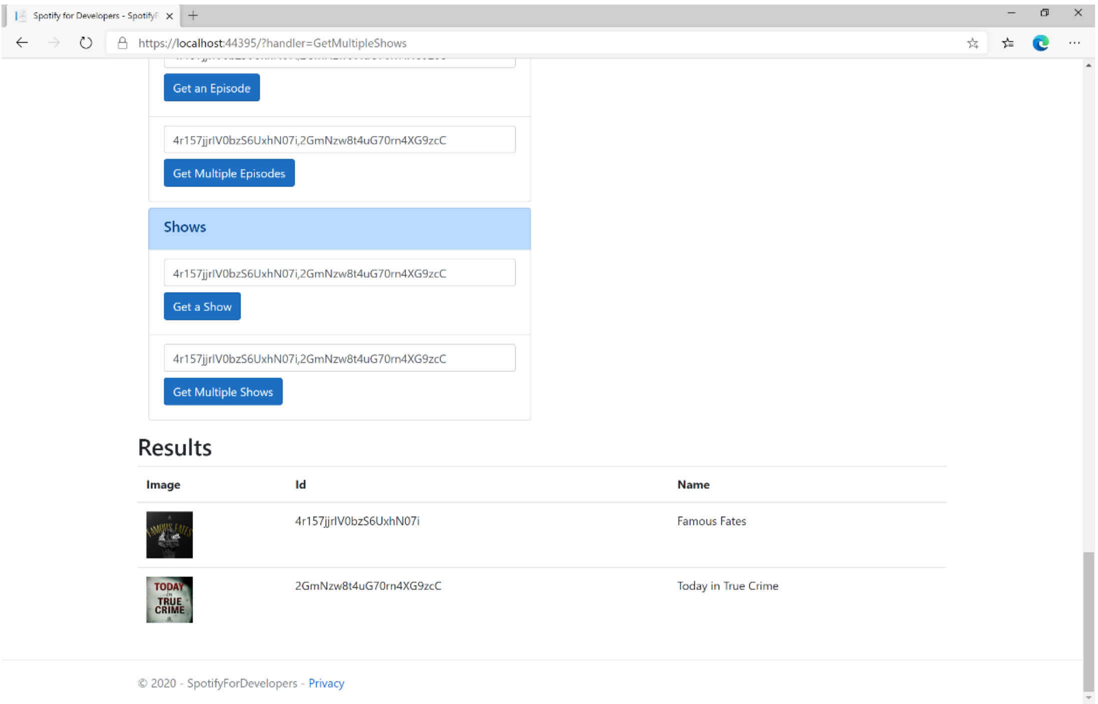
Step 9

Once the **Web Application** is running and you select **Client Credentials Flow Login** and scroll down you should see something like the following:


Spotify for Developers - Spotify
https://localhost:44395/?handler=clientcredentials
Get Audio Features for a Track
Track Id
Get Audio Analysis for a Track
Episodes
Episode Id
Get an Episode
Episode Ids
Get Multiple Episodes
Shows
Show Id
Get a Show
Show Ids
Get Multiple Shows
© 2020 - SpotifyForDevelopers - Privacy

Step 10

You can get some **Shows Ids** by using **Search for an Item** and pick **Shows** then into **Query** enter some **Podcast Names** then copy some **Shows Ids** to use and select **Get Multiple Shows** and scroll down to view **Results** like the following:


Spotify for Developers - Spotify
https://localhost:44395/?handler=GetMultipleShows
Get an Episode
Get Multiple Episodes
Shows
Get a Show
Get Multiple Shows
© 2020 - SpotifyForDevelopers - Privacy

4r157jjrIV0bzS6UxhN07i,2GmNzw8t4uG70rn4XG9zcC

4r157jjrIV0bzS6UxhN07i,2GmNzw8t4uG70rn4XG9zcC

4r157jjrIV0bzS6UxhN07i,2GmNzw8t4uG70rn4XG9zcC

Results

Image	Id	Name
	4r157jjrIV0bzS6UxhN07i	Famous Fates
	2GmNzw8t4uG70rn4XG9zcC	Today in True Crime

Step 11

You can stop the **web application** in **Visual Studio 2019** by selecting the **Stop debugging** button

Step 12

You can choose to exit **Visual Studio 2019** by selecting the **Close** button in the top right of the **application** as that completes this part of the workshop

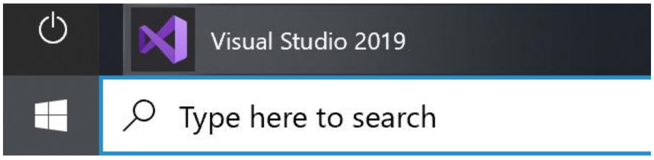
Get a Show's Episodes

Get Spotify catalogue information about a show’s episodes. Optional parameters can be used to limit the number of episodes returned.

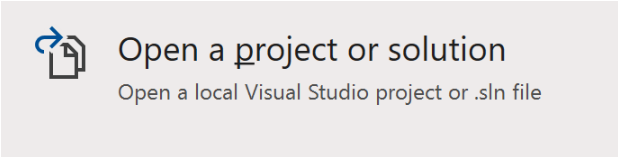
GET https://api.spotify.com/v1/shows/{id}/episodes	
Header	
Authorization	Valid Access Token or User Token from Spotify Accounts service with **user-read-playback-position** scope to read user’s resume points on episode objects
Path Parameter	
id	The Spotify Id for the show
limit	The maximum number of episodes to return. Default: 20. Minimum: 1. Maximum: 50.
offset	The index of the first episode to return. Default: 0 (the first object). Use with limit to get the next set of episodes.
market	An ISO 3166-1 alpha-2 country code. If a country code is specified, only shows and episodes that are available in that market will be returned. If a valid user access token is specified in the request header, the country associated with the user account will take priority over this parameter. Note: If neither market nor user country are provided, the content is considered unavailable for the client. Users can view the country that is associated with their account in the account settings.

Header	Response
Success	
HTTP Status 200 OK	Array of Simplified Episode Objects wrapped in a Paging Object
Error	
Error Code	Error Object

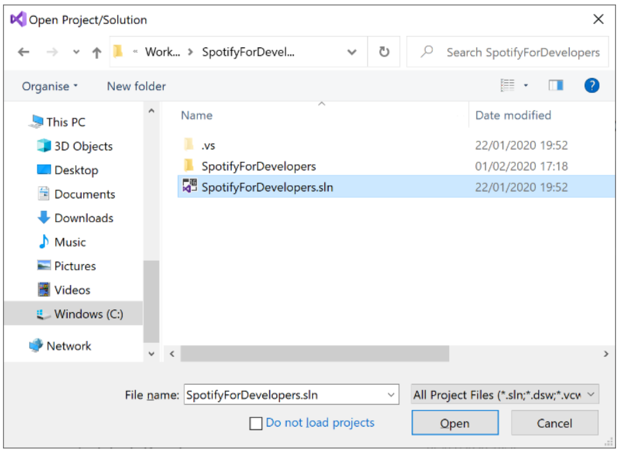
Step 1



If you chose to close **Visual Studio 2019** previously, in **Windows 10** choose **Start**, and then from the **Start Menu** find and select **Visual Studio 2019**

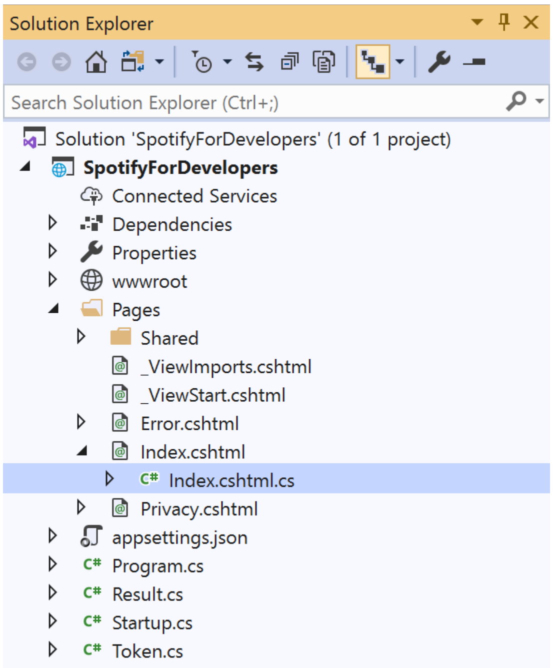

Open a project or solution
Open a local Visual Studio project or .sln file

Once done, from the **Get started** screen for **Visual Studio 2019** select **Open a project or solution**


Open Project/Solution
Work... › SpotifyForDevel...
Search SpotifyForDevelopers
Organise
New folder
This PC
3D Objects
Desktop
Documents
Downloads
Music
Pictures
Videos
Windows (C:)
Network
Name
Date modified
.vs 22/01/2020 19:52
SpotifyForDevelopers 01/02/2020 17:18
SpotifyForDevelopers.sln 22/01/2020 19:52
File name: SpotifyForDevelopers.sln
All Project Files (*.sln;*.dsw;*.vcw
Do not load projects
Open
Cancel

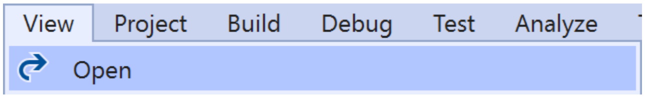
Then locate and select **SpotifyForDevelopers.sln** and select **Open** if you don’t have this file already then please follow the previous parts of the workshop including **Getting Started**, **Authorisation Guide**, **Search & Browse**, **Artists, Albums & Tracks** and **Episodes & Shows**

Step 2



Once opened, in the **Solution Explorer** open the **Pages** section, then open the **Index.cshtml** section and select **Index.cshtml.cs**

Step 3



Then from the **Menu** choose **View** and then **Open**

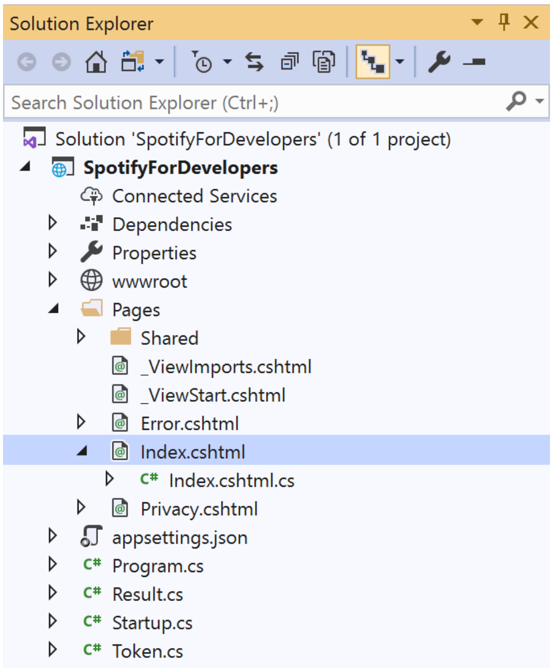
Step 4

In the **Code View** for **Index.cshtml.cs** below the **method** for `public async Task<IActionResult> OnPostGetMultipleShowsAsync(...) { ... }` enter the following **method**:

```
public async Task<IActionResult> OnPostGetShowEpisodesAsync(string value)
{
    LoadToken();
    var results = await Api.GetShowEpisodesAsync(value, country);
    if (results != null)
    {
        Results = results.Items.Select(result => new Result()
        {
            Id = result.Id,
            Name = result.Name,
            Image = result?.Images?.FirstOrDefault()?.Url
        });
    }
    return Page();
}
```

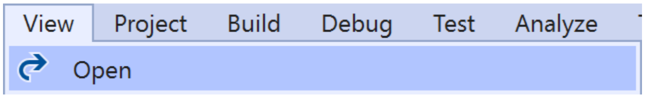
The **method** for `OnPostGetShowEpisodesAsync` is used to get **episodes** by **Show Id** on Spotify with the `value` and populate the **property** for `Results` accordingly.

Step 5



In the **Solution Explorer** in the **Pages** section select **Index.cshtml**

Step 6



Then from the **Menu** choose **View** and then **Open**

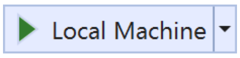
Step 7

Once in the **Code View** for **Index.cshtml** above `<!-- Shows -->` enter the following:

```
<li class="list-group-item">
    <form asp-page-handler="GetShowEpisodes" method="post">
        <input asp-for="Value" placeholder="Show Id" class="form-control mb-2" />
        <button class="btn btn-primary mb-2">
            Get a Show's Episodes
        </button>
    </form>
</li>
```

This `form` will **post** to the **method** for `GetShowEpisodes` with the `Value` as **Show Id** and will output to the **Results**.

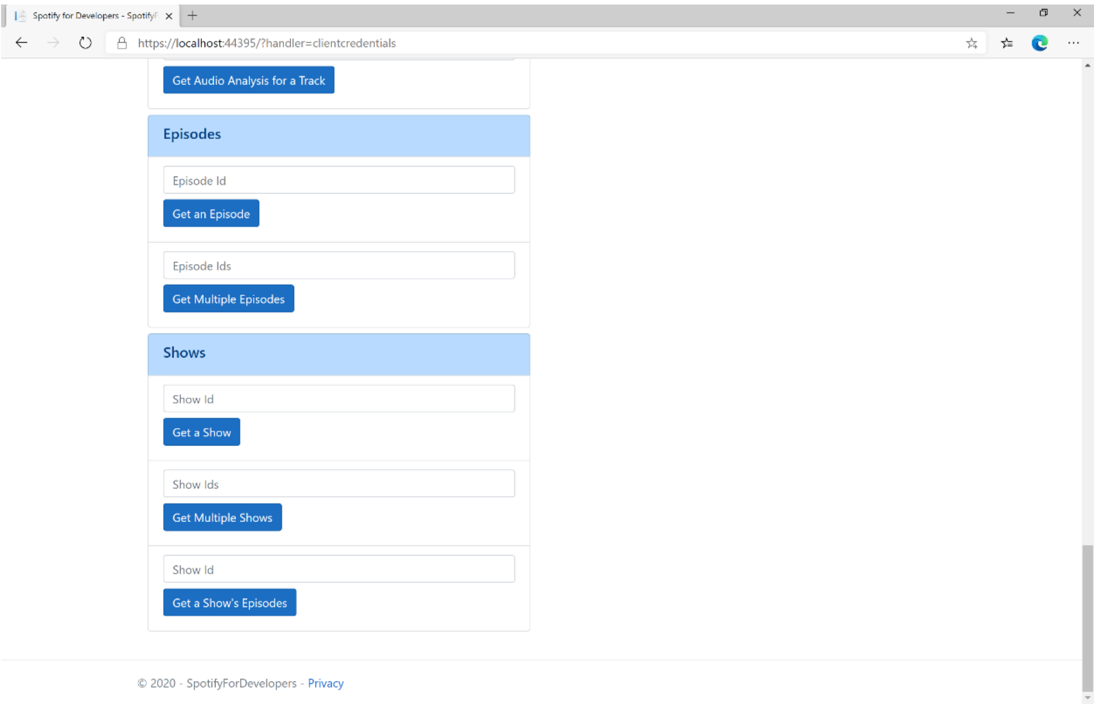
Step 8



Finally, in **Visual Studio 2019** select **IIS Application** to run the **Web Application**

Step 9

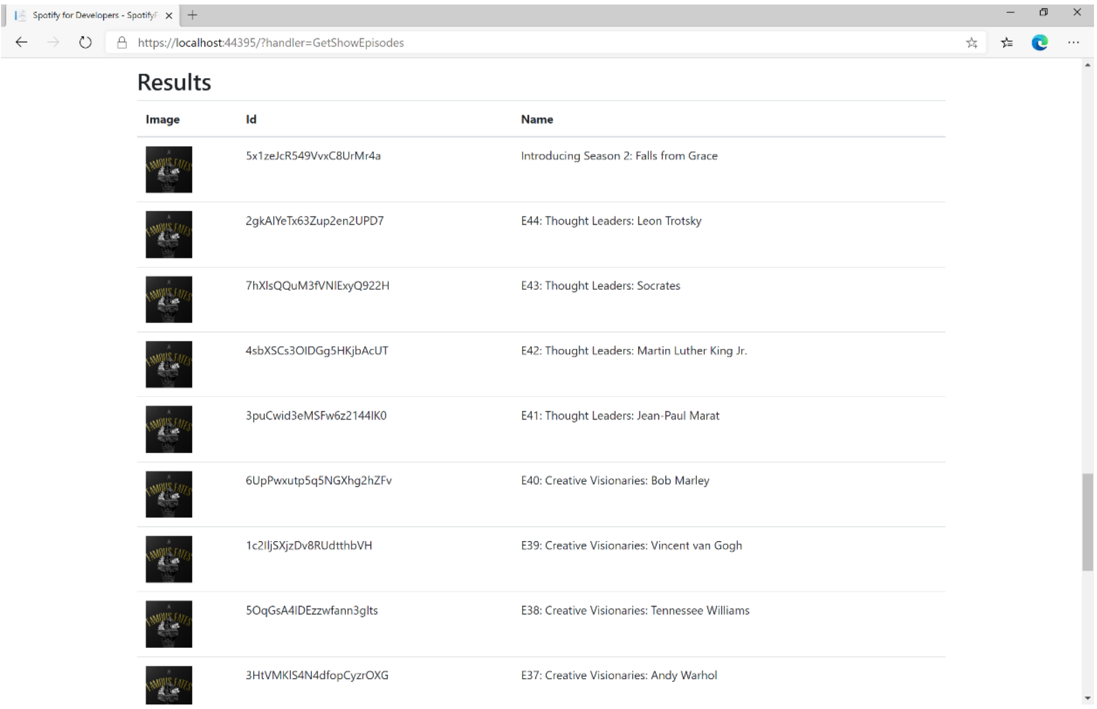
Once the **Web Application** is running and you select **Client Credentials Flow Login** and scroll down you should see something like the following:



Spotify for Developers - Spotify
https://localhost:44395/?handler=clientcredentials
Get Audio Analysis for a Track
Episodes
Episode Id
Get an Episode
Episode Ids
Get Multiple Episodes
Shows
Show Id
Get a Show
Show Ids
Get Multiple Shows
Show Id
Get a Show's Episodes
© 2020 - SpotifyForDevelopers - Privacy

Step 10

You can get a **Show Id** by using **Search for an Item** and pick **Shows** then into **Query** enter a **Podcast Name** then copy a **Show Id** to use and select **Get a Show's Episodes** and scroll down to view **Results** like the following:



Spotify for Developers - Spotify
https://localhost:44395/?handler=GetShowEpisodes

Results

Image	Id	Name
	5x1zeJcR549VvxC8UrMr4a	Introducing Season 2: Falls from Grace
	2gkAlYeTx63Zup2en2UPD7	E44: Thought Leaders: Leon Trotsky
	7hXlsQQuM3fVNlExyQ922H	E43: Thought Leaders: Socrates
	4sbXSCs3OlDGg5HKjbAcUT	E42: Thought Leaders: Martin Luther King Jr.
	3puCwid3eMSFw6z2144lK0	E41: Thought Leaders: Jean-Paul Marat
	6UpPwxutp5q5NGXhg2hZFv	E40: Creative Visionaries: Bob Marley
	1c2IljSXjzDv8RUdtthbVH	E39: Creative Visionaries: Vincent van Gogh
	5OqGsA4lDEzzwfann3glts	E38: Creative Visionaries: Tennessee Williams
	3HtVMKlS4N4dfopCyzrOXG	E37: Creative Visionaries: Andy Warhol

Step 11

■

You can stop the **web application** in **Visual Studio 2019** by selecting the **Stop debugging** button

Step 12

✕

You can exit **Visual Studio 2019** by selecting the **Close** button in the top right of the **application** as that completes this part of the workshop

Follow

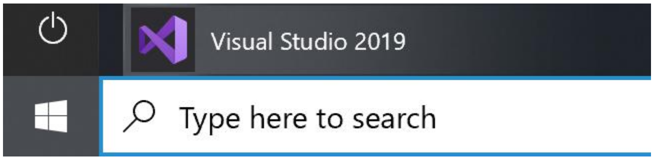
Follow Artists or Users

Add the current user as a follower of one or more artists or other Spotify users.

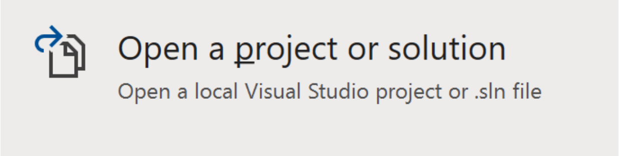
PUT https://api.spotify.com/v1/me/following	
Header	
Authorization	User Token from Spotify Accounts service with **user-follow-modify** scope
Query Parameter	
type	Type of artist or user
ids	Comma-separated list of the artist or the user Spotify Ids - maximum 50

Header	Response
Success	
HTTP Status 204 No Content	Empty Response Body
Error	
Error Code	Error Object

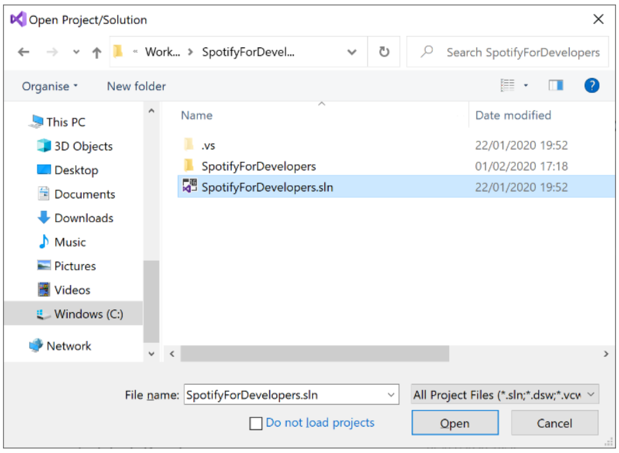
Step 1



In **Windows 10** choose **Start**, and then from the **Start Menu** find and select **Visual Studio 2019**


Open a project or solution
Open a local Visual Studio project or .sln file

Once done, from the **Get started** screen for **Visual Studio 2019** select **Open a project or solution**

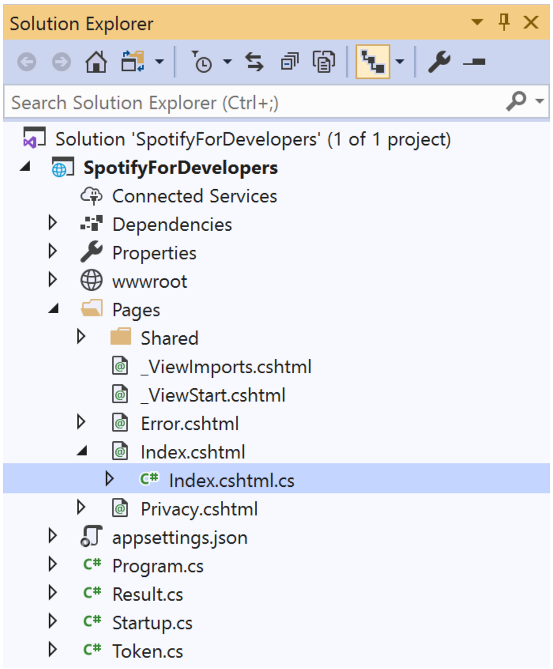


Name	Date modified
.vs	22/01/2020 19:52
SpotifyForDevelopers	01/02/2020 17:18
SpotifyForDevelopers.sln	22/01/2020 19:52

File name: SpotifyForDevelopers.sln
All Project Files (*.sln;*.dsw;*.vcw

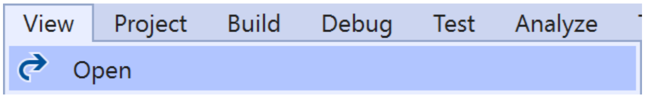
Then locate and select **SpotifyForDevelopers.sln** and select **Open** if you don’t have this file already then please follow the previous parts of the workshop including **Getting Started**, **Authorisation Guide**, **Search & Browse**, **Playlists & Artists**, **Albums & Tracks** and **Episodes & Shows**

Step 2



Once opened, in the **Solution Explorer** open the **Pages** section, then open the **Index.cshtml** section and select **Index.cshtml.cs**

Step 3



Then from the **Menu** choose **View** and then **Open**

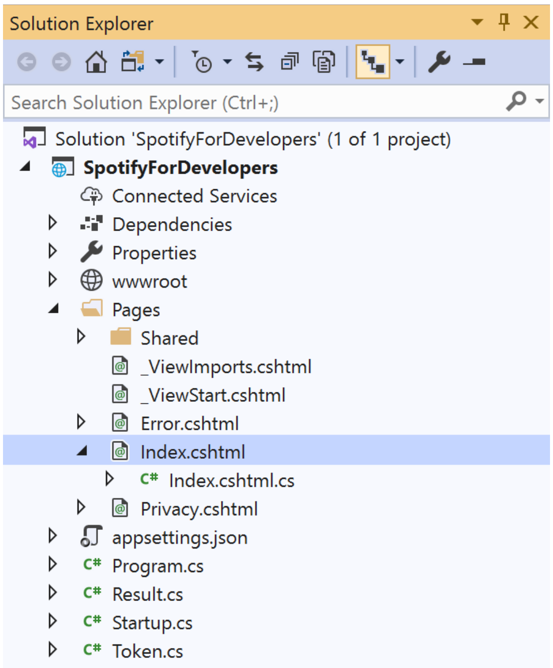
Step 4

In the **Code View** for **Index.cshtml.cs** below the **method** for `public async Task<IActionResult> OnPostGetShowEpisodesAsync(...) { ... }` enter the following **method**:

```
public async Task<IActionResult> OnPostFollowArtistsOrUsersAsync(string value, FollowType
option)
{
    LoadToken();
    var values = value.Split(",").ToList();
    var result = await Api.FollowArtistsOrUsersAsync(values, option);
    if (result != null)
    {
        Results = new List<Result>() { new Result()
        {
            Name = result.Success.ToString()
        }};
    }
    return Page();
}
```

The **method** for `OnPostFollowArtistsOrUsersAsync` is used to set the **following state** for an **artist** or **user** by **Artist Id** or **User Id** on Spotify with the `value` and selected `option` and populate the **property** for `Results` accordingly with the **success** of the operation.

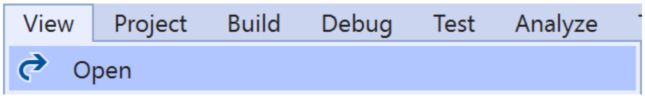
Step 5



Solution 'SpotifyForDevelopers' (1 of 1 project)
- SpotifyForDevelopers
  - Connected Services
  - Dependencies
  - Properties
  - wwwroot
  - Pages
    - Shared
    - _ViewImports.cshtml
    - _ViewStart.cshtml
    - Error.cshtml
    - Index.cshtml
      - Index.cshtml.cs
    - Privacy.cshtml
  - appsettings.json
  - Program.cs
  - Result.cs
  - Startup.cs
  - Token.cs

In the **Solution Explorer** in the **Pages** section select **Index.cshtml**

Step 6



Then from the **Menu** choose **View** and then **Open**

Step 7

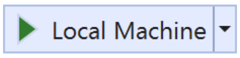
Once in the **Code View** for **Index.cshtml** above `<!-- Spotify Web API User Authorisation -->` enter the following:

```
<ul class="list-group mb-2">
    <li class="list-group-item list-group-item-primary">
        <h5 class="list-group-item-heading">Follow</h5>
    </li>
    <li class="list-group-item">
        <form asp-page-handler="FollowArtistsOrUsers" method="post">
            <select asp-for="Option" class="form-control mb-2"
                asp-items="Html.GetEnumSelectList<Spotify.NetStandard.Enums.FollowType>()">
            </select>
            <input asp-for="Value" placeholder="Ids" class="form-control mb-2" />
            <button class="btn btn-primary mb-2">
                Follow Artists or Users
            </button>
        </form>
    </li>

    <!-- Follow -->
</ul>
```

This `form` will **post** to the **method** for `FollowArtistsOrUsers` with the `Value` of the **Ids** and the `Option` for the **Artist** or **User** and will output to the **Results**.

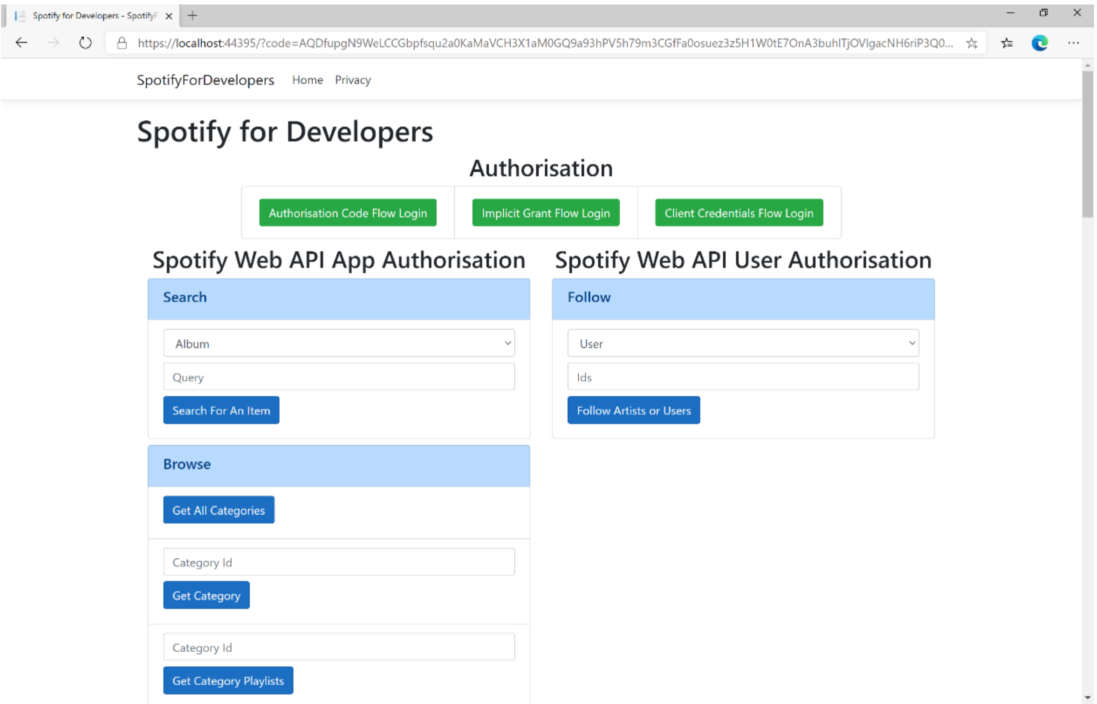
Step 8



Finally, in **Visual Studio 2019** select **IIS Application** to run the **Web Application**

Step 9

Once the **Web Application** is running and you select **Authorisation Code Flow Login** or **Implicit Grant Flow Login** you should see something like the following:



Spotify for Developers - Spotify
https://localhost:44395/?code=AQDfupgN9WeLCCGbpfsqu2a0KaMaVCH3X1aM0GQ9a93hPV5h79m3CGfFa0osuez3z5H1W0tE7OnA3buhlTjOVlgacNH6riP3Q0...
SpotifyForDevelopers
Home
Privacy
Authorisation Code Flow Login
Implicit Grant Flow Login
Client Credentials Flow Login
Search
Album
Query
Search For An Item
Browse
Get All Categories
Category Id
Get Category
Category Id
Get Category Playlists
Follow
User
Ids
Follow Artists or Users

Spotify for Developers

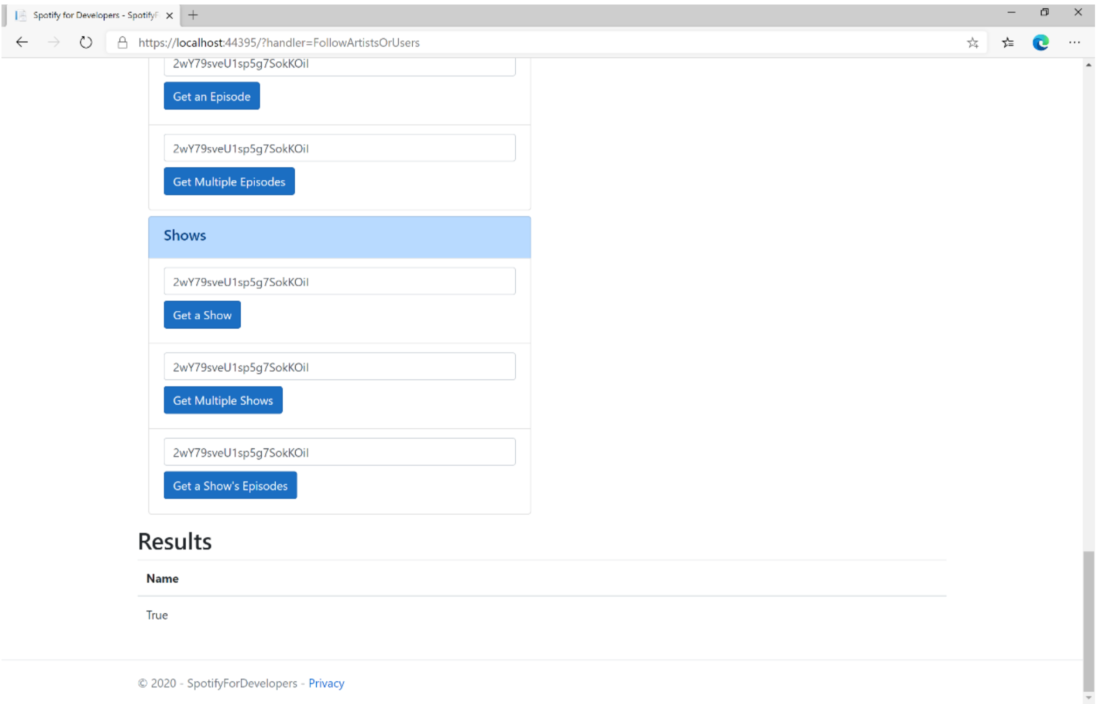
Authorisation

Spotify Web API App Authorisation

Spotify Web API User Authorisation

Step 10

You can then enter an **Artist Id** from **Get All New Releases** and copy an **Artist Id** to use and above **Follow Artists or Users** choose **Artist** and then select **Follow Artists or Users** and scroll down to view **Results** like the following:



2wY79sveU1sp5g7SokKOil

2wY79sveU1sp5g7SokKOil

2wY79sveU1sp5g7SokKOil

2wY79sveU1sp5g7SokKOil

2wY79sveU1sp5g7SokKOil

Results

Name
True

© 2020 - SpotifyForDevelopers - Privacy

Step 11

■

You can stop the **web application** in **Visual Studio 2019** by selecting the **Stop debugging** button

Step 12

×

You can choose to exit **Visual Studio 2019** by selecting the **Close** button in the top right of the **application** as that completes this part of the workshop

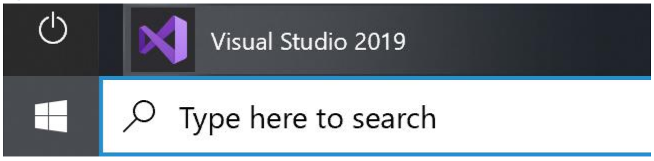
Follow a Playlist

Add the current user as a follower of a playlist.

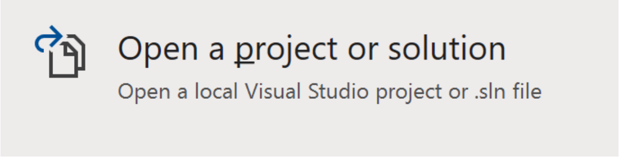
PUT https://api.spotify.com/v1/playlists/{playlist_id}/followers	
Header	
Authorization	User Token from Spotify Accounts service with **user-follow-modify** scope
Path Parameter	
playlist_id	Spotify Id of the playlist
Body Parameter	
public	Defaults to true. If true the Playlist will be included in the User's public Playlists, if false it will remain private

Header	**Response**
Success	
HTTP Status 200 OK	Empty Response Body
Error	
Error Code	Error Object

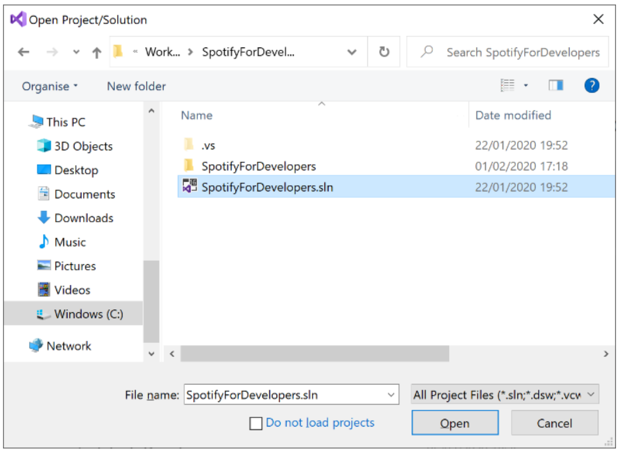
Step 1



If you chose to close **Visual Studio 2019** previously, in **Windows 10** choose **Start**, and then from the **Start Menu** find and select **Visual Studio 2019**


Open a project or solution
Open a local Visual Studio project or .sln file

Once done, from the **Get started** screen for **Visual Studio 2019** select **Open a project or solution**



Name	Date modified
.vs	22/01/2020 19:52
SpotifyForDevelopers	01/02/2020 17:18
SpotifyForDevelopers.sln	22/01/2020 19:52

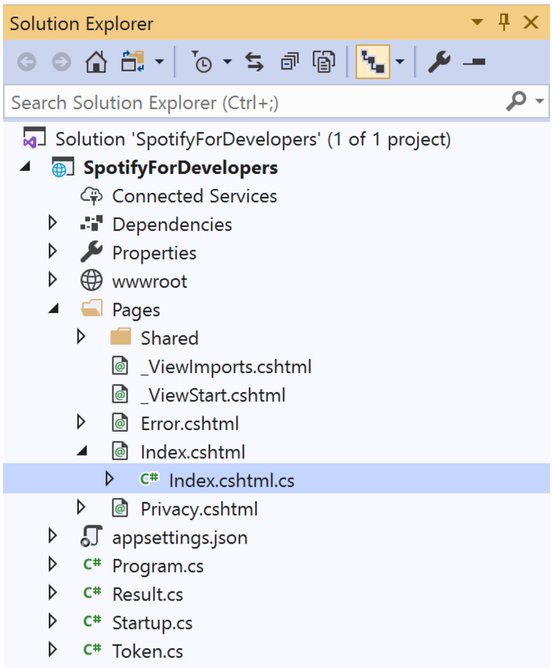
File name: SpotifyForDevelopers.sln

All Project Files (*.sln;*.dsw;*.vcw

☐ Do not load projects

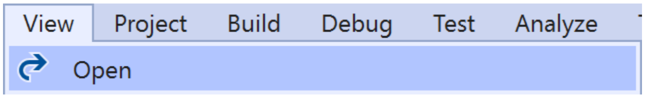
Then locate and select **SpotifyForDevelopers.sln** and select **Open** if you don't have this file already then please follow the previous parts of the workshop including **Getting Started**, **Authorisation Guide**, **Search & Browse**, **Playlists & Artists**, **Albums & Tracks**, **Episodes & Shows** and **Follow**

Step 2



Once opened, in the **Solution Explorer** open the **Pages** section, then open the **Index.cshtml** section and select **Index.cshtml.cs**

Step 3



Then from the **Menu** choose **View** and then **Open**

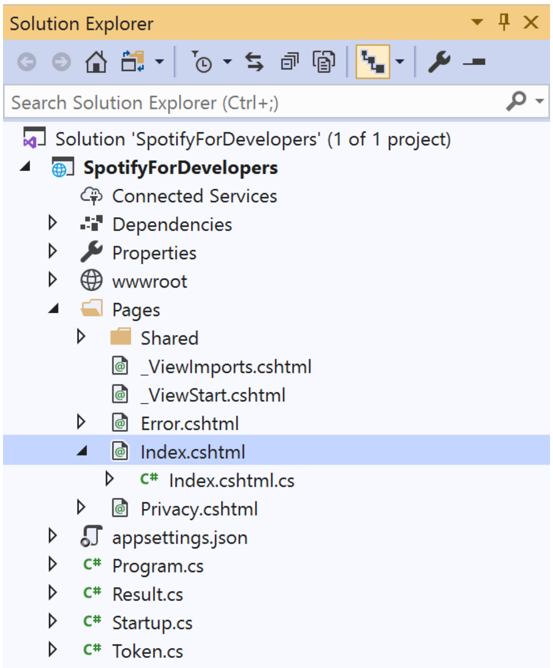
Step 4

In the **Code View** for **Index.cshtml.cs** below the **method** for `public async Task<IActionResult> OnPostFollowArtistsOrUsersAsync(...) { ... }` enter the following **method**:

```
public async Task<IActionResult> OnPostFollowPlaylistAsync(string value)
{
    LoadToken();
    var result = await Api.FollowPlaylistAsync(value);
    if (result != null)
    {
        Results = new List<Result>() { new Result()
        {
            Name = result.Success.ToString()
        }};
    }
    return Page();
}
```

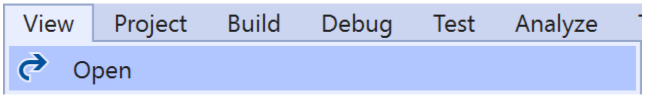
The **method** for `OnPostFollowPlaylistAsync` is used to set the **following state** for a **playlist** by **Playlist Id** on Spotify with the `value` and populate the **property** for `Results` accordingly with the **success** of the operation.

Step 5



In the **Solution Explorer** in the **Pages** section select **Index.cshtml**

Step 6



Then from the **Menu** choose **View** and then **Open**

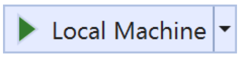
Step 7

Once in the **Code View** for **Index.cshtml** above `<!-- Follow -->` enter the following:

```
<li class="list-group-item">
    <form asp-page-handler="FollowPlaylist" method="post">
        <input asp-for="Value" placeholder="Playlist Id" class="form-control mb-2" />
        <button class="btn btn-primary mb-2">
            Follow a Playlist
        </button>
    </form>
</li>
```

This `form` will **post** to the **method** for `FollowPlaylist` with the `Value` of the **Playlist Id** and will output to the **Results**.

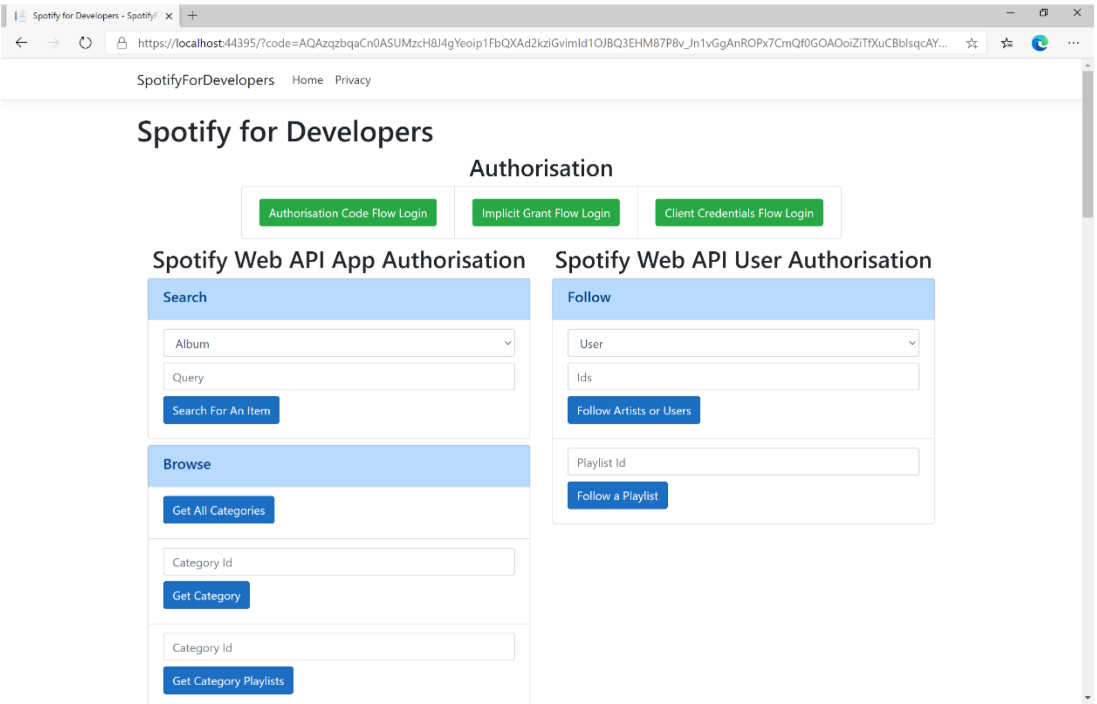
Step 8



Finally, in **Visual Studio 2019** select **IIS Application** to run the **Web Application**

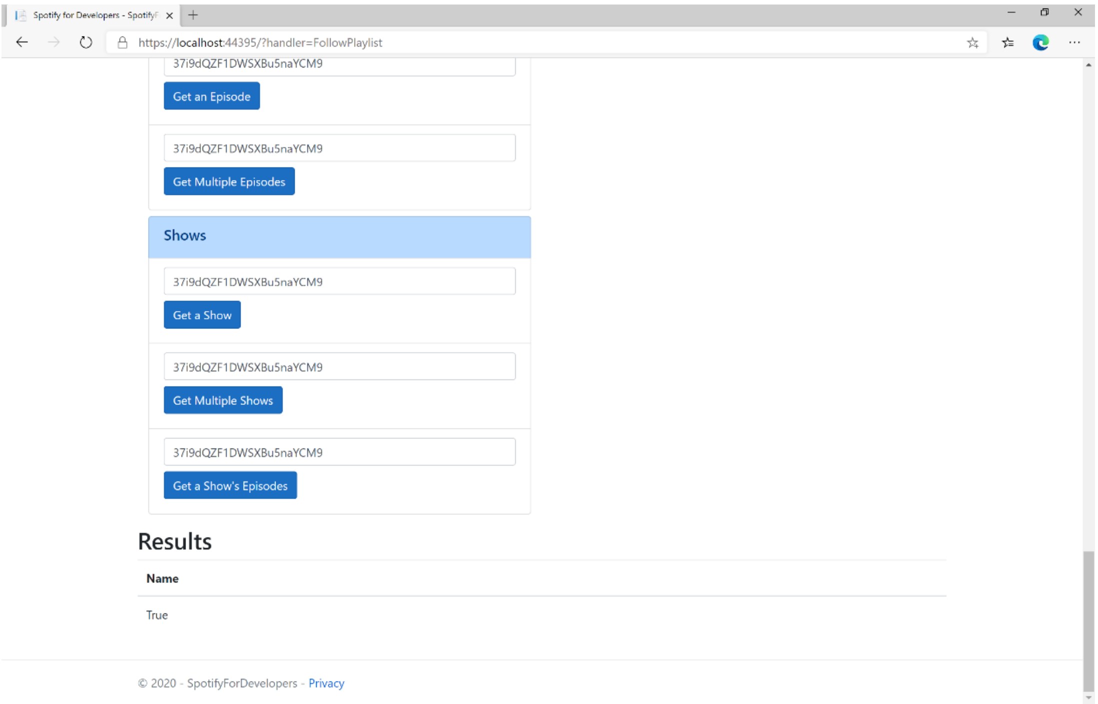
Step 9

Once the **Web Application** is running and you select **Authorisation Code Flow Login** or **Implicit Grant Flow Login** you should see something like the following:


Spotify for Developers - Spotify
https://localhost:44395/?code=AQAzqzbqaCn0ASUMzcH8J4gYeoip1FbQXAd2kziGvimld1OJBQ3EHM87P8v_Jn1vGgAnROPx7CmQf0GOAOoiZiTfXuCBblsqcAY...
SpotifyForDevelopers
Home
Privacy
Spotify for Developers
Authorisation
Authorisation Code Flow Login
Implicit Grant Flow Login
Client Credentials Flow Login
Spotify Web API App Authorisation
Spotify Web API User Authorisation
Search
Album
Query
Search For An Item
Browse
Get All Categories
Category Id
Get Category
Category Id
Get Category Playlists
Follow
User
Ids
Follow Artists or Users
Playlist Id
Follow a Playlist

Step 10

You can then enter a **Playlist Id** from **Get All Featured Playlists** and copy a **Playlist Id** to use and select **Follow a Playlist** and scroll down to view **Results** like the following:



Results

Name
True

© 2020 - SpotifyForDevelopers - Privacy

Step 11

■

You can stop the **web application** in **Visual Studio 2019** by selecting the **Stop debugging** button

Step 12

×

You can choose to exit **Visual Studio 2019** by selecting the **Close** button in the top right of the **application** as that completes this part of the workshop

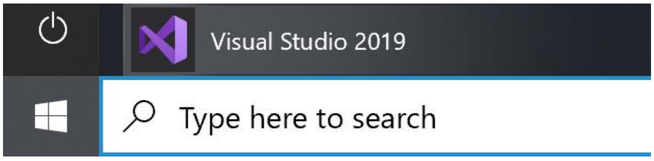
Get Following State for Artists/Users

Check to see if the current user is following one or more artists or other Spotify users.

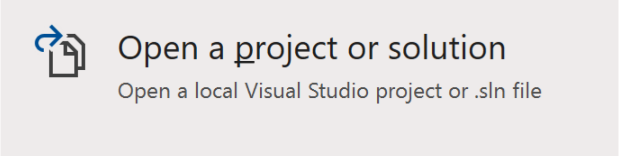
GET https://api.spotify.com/v1/me/following/contains	
Header	
Authorization	User Token from Spotify Accounts service with **user-follow-read** scope
Query Parameter	
type	Type of artist or user
ids	Comma-separated list of the artist or the user Spotify Ids to check

Header	**Response**
Success	
HTTP Status 200 OK	Array of true or false values in same order as in which the Ids were provided
Error	
Error Code	Error Object

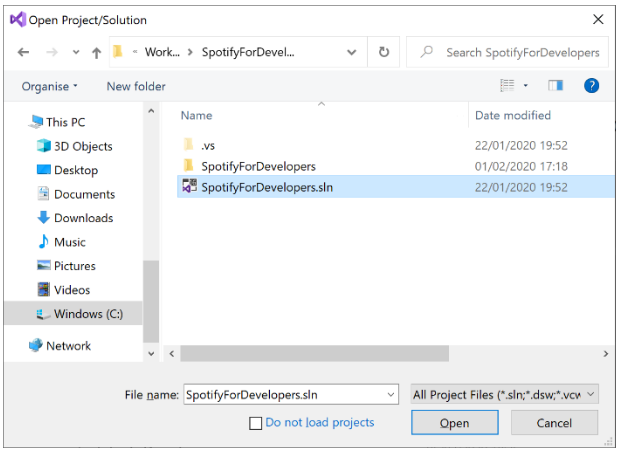
Step 1



If you chose to close **Visual Studio 2019** previously, in **Windows 10** choose **Start**, and then from the **Start Menu** find and select **Visual Studio 2019**

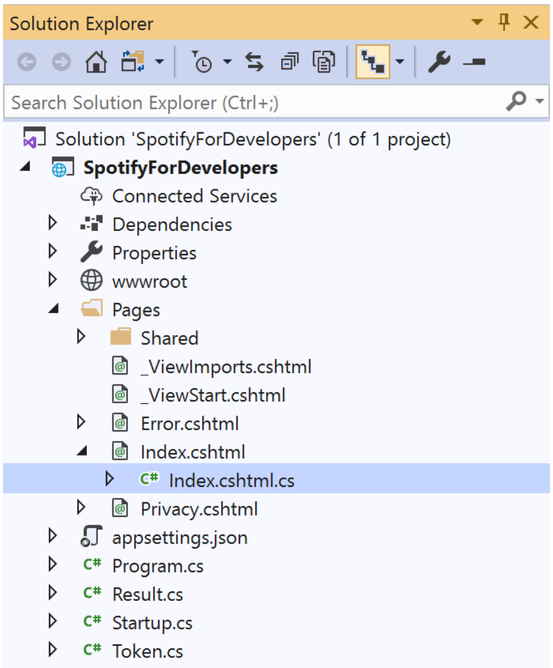

Open a project or solution
Open a local Visual Studio project or .sln file

Once done, from the **Get started** screen for **Visual Studio 2019** select **Open a project or solution**



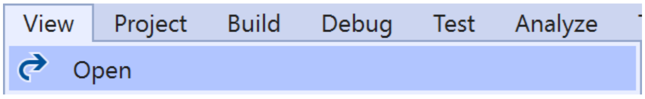
Then locate and select **SpotifyForDevelopers.sln** and select **Open** if you don't have this file already then please follow the previous parts of the workshop including **Getting Started**, **Authorisation Guide**, **Search & Browse**, **Playlists & Artists**, **Albums & Tracks**, **Episodes & Shows** and **Follow**

Step 2



Once opened, in the **Solution Explorer** open the **Pages** section, then open the **Index.cshtml** section and select **Index.cshtml.cs**

Step 3



Then from the **Menu** choose **View** and then **Open**

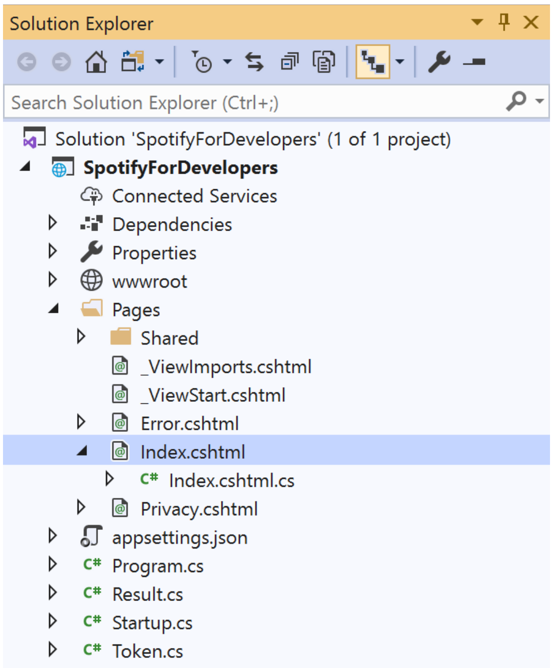
Step 4

In the **Code View** for **Index.cshtml.cs** below the **method** for `public async Task<IActionResult> OnPostFollowPlaylistAsync(...) { ... }` enter the following **method**:

```
public async Task<IActionResult> OnPostGetFollowingStateForArtistsOrUsersAsync(string value,
FollowType option)
{
    LoadToken();
    var values = value.Split(",").ToList();
    var results = await Api.GetFollowingStateForArtistsOrUsersAsync(values, option);
    if (results != null)
    {
        Results = results.Select(result => new Result()
        {
            Name = result.ToString()
        });
    }
    return Page();
}
```

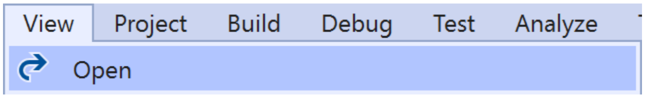
The **method** for `OnPostGetFollowingStateForArtistsOrUsersAsync` is used to get the **following state** for a **artist** or **user** by **Artist Id** or **User Id** on Spotify with the `value` and selected `option` and populate the **property** for `Results` accordingly.

Step 5



In the **Solution Explorer** in the **Pages** section select **Index.cshtml**

Step 6



Then from the **Menu** choose **View** and then **Open**

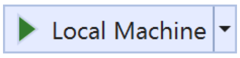
Step 7

Once in the **Code View** for **Index.cshtml** above `<!-- Follow -->` enter the following:

```
<li class="list-group-item">
    <form asp-page-handler="GetFollowingStateForArtistsOrUsers" method="post">
        <select asp-for="Option" class="form-control mb-2"
            asp-items="Html.GetEnumSelectList<Spotify.NetStandard.Enums.FollowType>()">
        </select>
        <input asp-for="Value" placeholder="Ids" class="form-control mb-2" />
        <button class="btn btn-primary mb-2">
            Get Following State for Artists or Users
        </button>
    </form>
</li>
```

This `form` will **post** to the **method** for `GetFollowingStateForArtistsOrUsers` with the `Value` of the **Ids** and the `Option` for the **Artist** or **User** and will output to the **Results**.

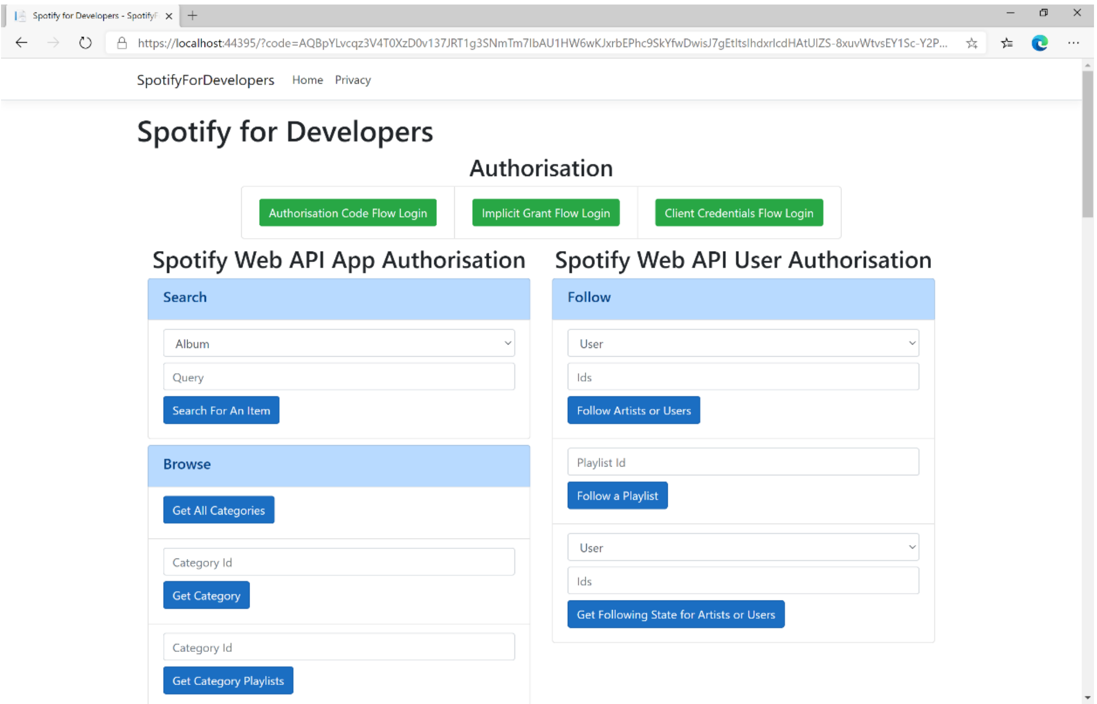
Step 8



Finally, in **Visual Studio 2019** select **IIS Application** to run the **Web Application**

Step 9

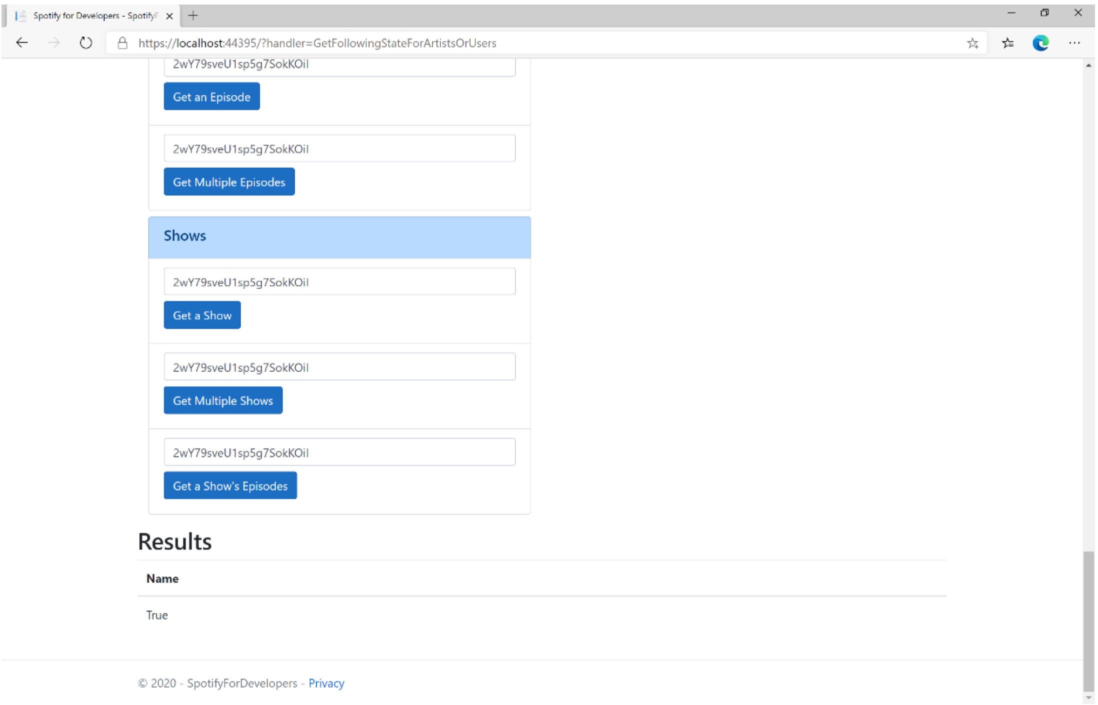
Once the **Web Application** is running and you select **Authorisation Code Flow Login** or **Implicit Grant Flow Login** you should see something like the following:



https://localhost:44395/?code=AQBpYLvcqz3V4T0XzD0v137JRT1g3SNmTm7lbAU1HW6wKJxrbEPhc9SkYfwDwisJ7gEtltslhdxrlcdHAtUIZS-8xuvWtvsEY1Sc-Y2P...

Step 10

You can then enter an **Artist Id** from **Get All New Releases** or copy the same **Artist Id** from **Follow Artists or Users** to use and above **Get Following State for Artists or Users** choose **Artist** and then select **Get Following State for Artists or Users** and scroll down to view **Results** like the following:



https://localhost:44395/?handler=GetFollowingStateForArtistsOrUsers

2wY79sveU1sp5g7SokKOil

2wY79sveU1sp5g7SokKOil

2wY79sveU1sp5g7SokKOil

2wY79sveU1sp5g7SokKOil

2wY79sveU1sp5g7SokKOil

Name
True

© 2020 - SpotifyForDevelopers - Privacy

Step 11

You can stop the **web application** in **Visual Studio 2019** by selecting the **Stop debugging** button

Step 12

You can choose to exit **Visual Studio 2019** by selecting the **Close** button in the top right of the **application** as that completes this part of the workshop

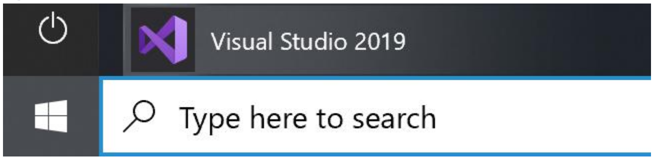
Check if Users Follow a Playlist

Check to see if one or more Spotify users are following a specified playlist.

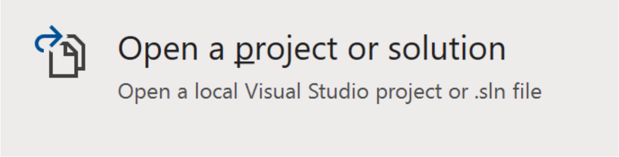
GET https://api.spotify.com/v1/playlists/{playlist_id}/followers/contains	
Header	
Authorization	User Token from Spotify Accounts service with **playlist-read-private** scope if a private playlist is requested
Path Parameter	
playlist_id	Spotify Id of the playlist.
Query Parameter	
ids	Comma-separated list of Spotify User Ids you want to check to see if they follow the playlist

Header	Response
Success	
HTTP Status 200 OK	Array of true or false values in same order as in which the Ids were provided
Error	
Error Code	Error Object

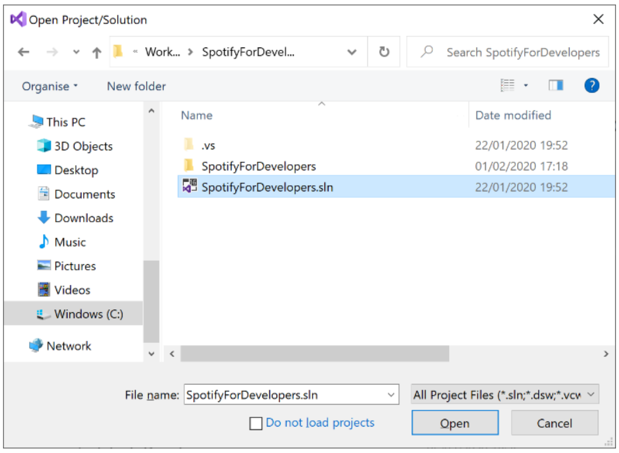
Step 1



If you chose to close **Visual Studio 2019** previously, in **Windows 10** choose **Start**, and then from the **Start Menu** find and select **Visual Studio 2019**


Open a project or solution
Open a local Visual Studio project or .sln file

Once done, from the **Get started** screen for **Visual Studio 2019** select **Open a project or solution**



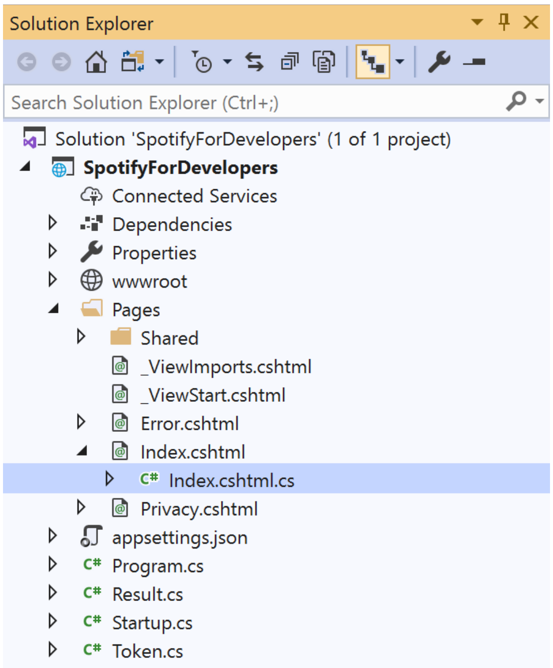
Name	Date modified
.vs	22/01/2020 19:52
SpotifyForDevelopers	01/02/2020 17:18
SpotifyForDevelopers.sln	22/01/2020 19:52

File name: SpotifyForDevelopers.sln

All Project Files (*.sln;*.dsw;*.vcw

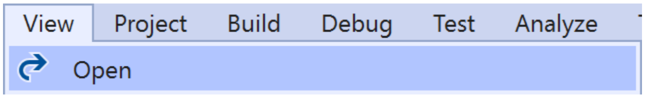
Then locate and select **SpotifyForDevelopers.sln** and select **Open** if you don't have this file already then please follow the previous parts of the workshop including **Getting Started**, **Authorisation Guide**, **Search & Browse**, **Playlists & Artists**, **Albums & Tracks**, **Episodes & Shows** and **Follow**

Step 2



Once opened, in the **Solution Explorer** open the **Pages** section, then open the **Index.cshtml** section and select **Index.cshtml.cs**

Step 3



Then from the **Menu** choose **View** and then **Open**

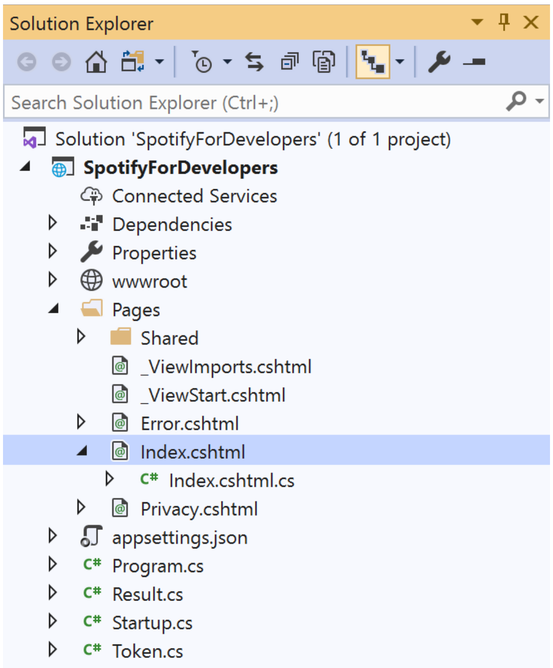
Step 4

In the **Code View** for **Index.cshtml.cs** below the **method** for `public async Task<IActionResult> OnPostGetFollowingStateForArtistsOrUsersAsync(...) { ... }` enter the following **method**:

```
public async Task<IActionResult> OnPostCheckUsersFollowingPlaylistAsync(string value, string
option)
{
    LoadToken();
    var values = value.Split(",").ToList();
    var results = await Api.CheckUsersFollowingPlaylistAsync(values, option);
    if (results != null)
    {
        Results = results.Select(result => new Result()
        {
            Name = result.ToString()
        });
    }
    return Page();
}
```

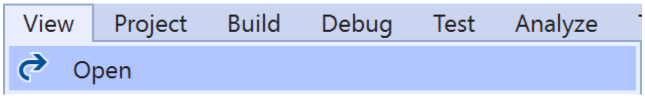
The **method** for `OnPostCheckUsersFollowingPlaylistAsync` is used to get the **following state** for a **playlist** by **Artist Id** for the **users** by **User Id** on Spotify with the `value` and `option` and populate the **property** for `Results` accordingly.

Step 5



In the **Solution Explorer** in the **Pages** section select **Index.cshtml**

Step 6



Then from the **Menu** choose **View** and then **Open**

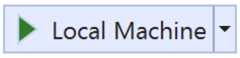
Step 7

Once in the **Code View** for **Index.cshtml** above `<!-- Follow -->` enter the following:

```
<li class="list-group-item">
    <form asp-page-handler="CheckUsersFollowingPlaylist" method="post">
        <input asp-for="Option" placeholder="Playlist Id" class="form-control mb-2" />
        <input asp-for="Value" placeholder="User Ids" class="form-control mb-2" />
        <button class="btn btn-primary mb-2">
            Check if Users Follow a Playlist
        </button>
    </form>
</li>
```

This `form` will **post** to the **method** for `CheckUsersFollowingPlaylist` with the `Value` of the **Playlist Id** and the `Option` for the **User Id** and will output to the **Results**.

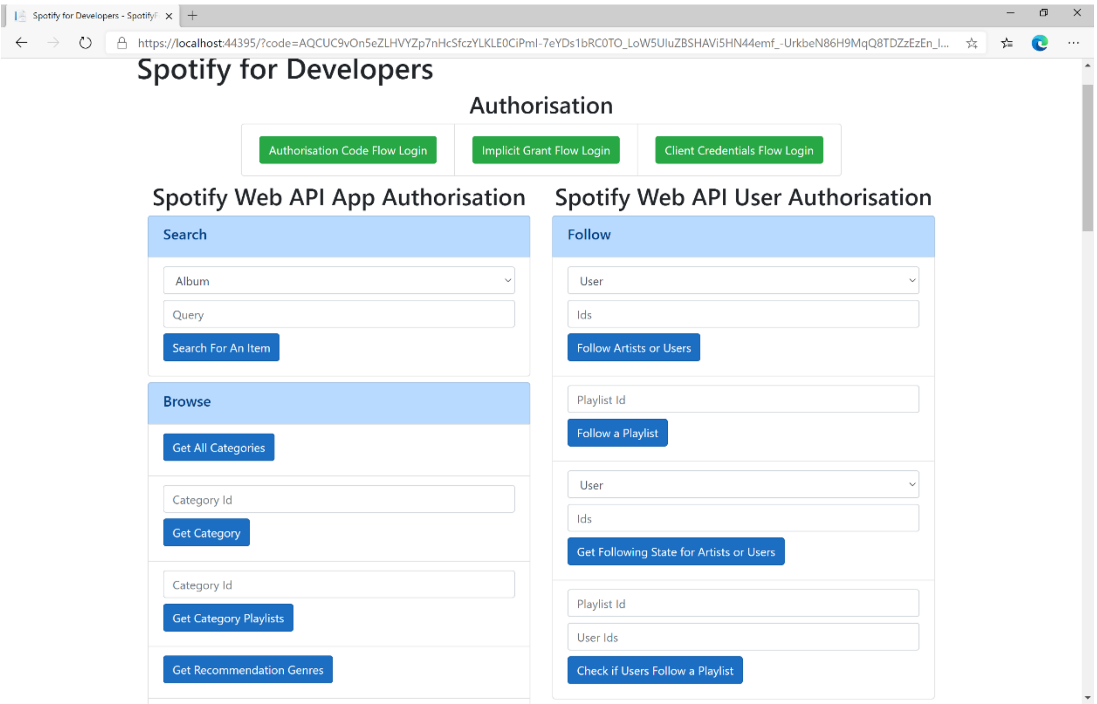
Step 8



Finally, in **Visual Studio 2019** select **IIS Application** to run the **Web Application**

Step 9

Once the **Web Application** is running and you select **Authorisation Code Flow Login** or **Implicit Grant Flow Login** and scroll down you should see something like the following:



Spotify for Developers - SpotifyF
https://localhost:44395/?code=AQCUC9vOn5eZLHVYZp7nHcSfczYLKLE0CiPml-7eYDs1bRC0TO_LoW5UluZBSHAVi5HN44emf_-UrkbeN86H9MqQ8TDZzEzEn_l...

Spotify for Developers

Authorisation

Authorisation Code Flow Login
Implicit Grant Flow Login
Client Credentials Flow Login

Spotify Web API App Authorisation

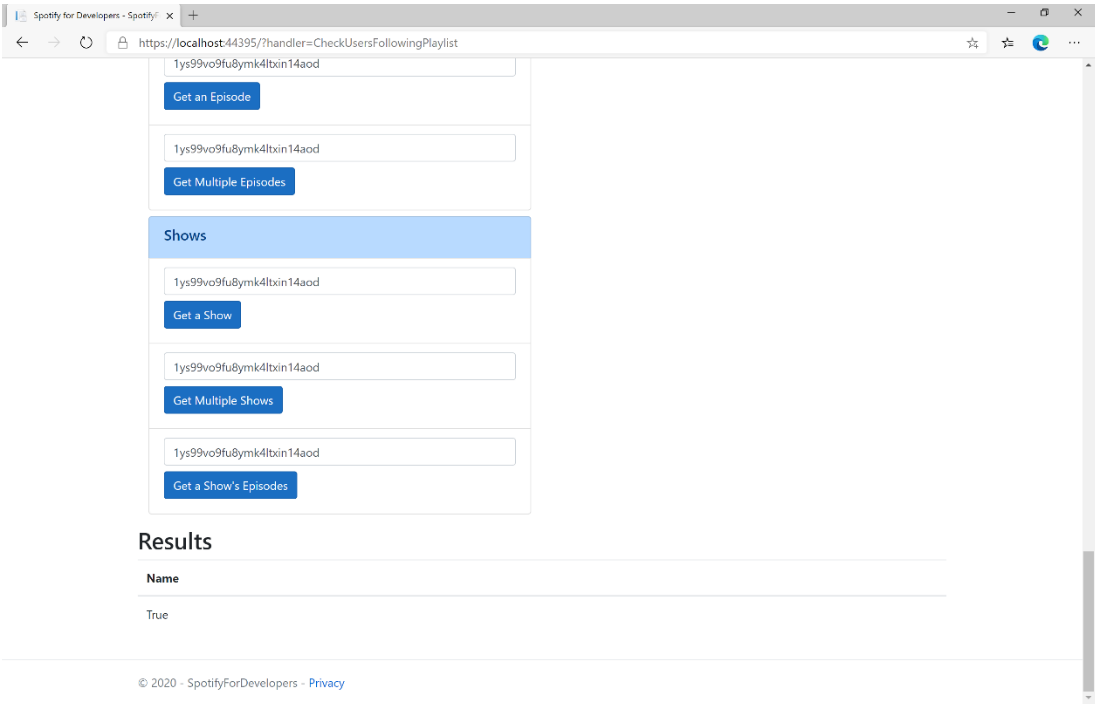
Search
Album
Query
Search For An Item
Browse
Get All Categories
Category Id
Get Category
Category Id
Get Category Playlists
Get Recommendation Genres

Spotify Web API User Authorisation

Follow
User
Ids
Follow Artists or Users
Playlist Id
Follow a Playlist
User
Ids
Get Following State for Artists or Users
Playlist Id
User Ids
Check if Users Follow a Playlist

Step 10

Use the same **Playlist Id** from **Follow a Playlist** and use your own **User Id**, to get this in **Spotify** select your **Username** then **...** and **Copy Spotify URI** your **User Id** will be after **spotify:user:** then enter this above **Check if Users Follow a Playlist** and then select **Check if Users Follow a Playlist** and scroll down to view **Results** like the following:



Spotify for Developers - SpotifyF
https://localhost:44395/?handler=CheckUsersFollowingPlaylist
1ys99vo9fu8ymk4ltxin14aod
Get an Episode
1ys99vo9fu8ymk4ltxin14aod
Get Multiple Episodes
Shows
1ys99vo9fu8ymk4ltxin14aod
Get a Show
1ys99vo9fu8ymk4ltxin14aod
Get Multiple Shows
1ys99vo9fu8ymk4ltxin14aod
Get a Show's Episodes

Results

Name
True

© 2020 - SpotifyForDevelopers - Privacy

Step 11

You can stop the **web application** in **Visual Studio 2019** by selecting the **Stop debugging** button

Step 12

You can choose to exit **Visual Studio 2019** by selecting the **Close** button in the top right of the **application** as that completes this part of the workshop

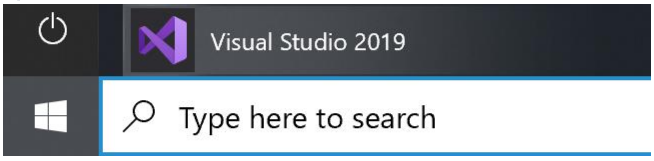
Get User's Followed Artists

Get the current user's followed artists.

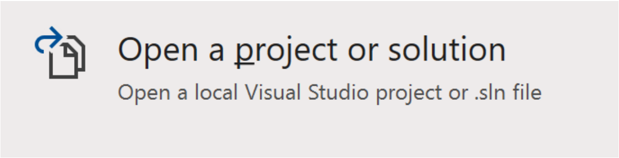
GET https://api.spotify.com/v1/me/following	
Header	
Authorization	User Token from Spotify Accounts service with **user-follow-modify** scope
Query Parameter	
type	Id type, only artist is currently supported
limit	Maximum number of items to return. Default: 20. Minimum: 1. Maximum: 50.
after	Last artist Id retrieved from the previous request.

Header	Response
Success	
HTTP Status 200 OK	Object containing a cursor-based paging object of Artist Objects
Error	
Error Code	Error Object

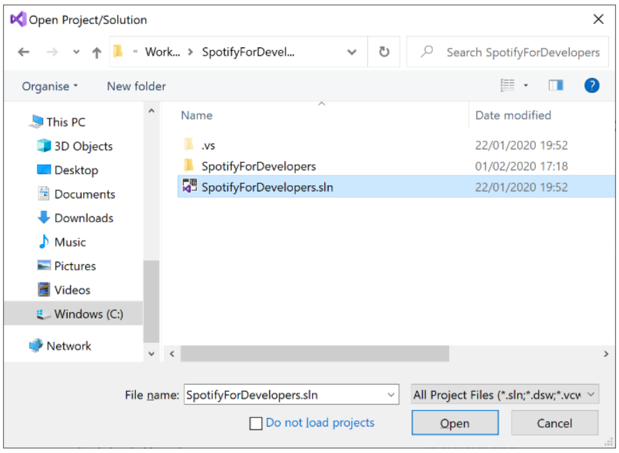
Step 1



If you chose to close **Visual Studio 2019** previously, in **Windows 10** choose **Start**, and then from the **Start Menu** find and select **Visual Studio 2019**


Open a project or solution
Open a local Visual Studio project or .sln file

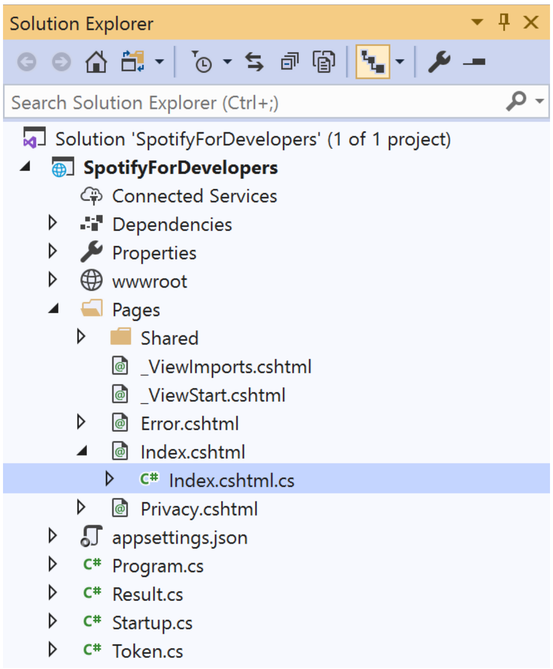
Once done, from the **Get started** screen for **Visual Studio 2019** select **Open a project or solution**



File name: SpotifyForDevelopers.sln

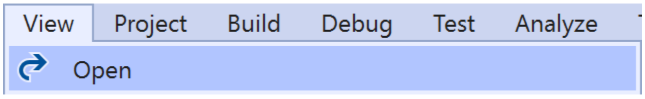
Then locate and select **SpotifyForDevelopers.sln** and select **Open** if you don't have this file already then please follow the previous parts of the workshop including **Getting Started**, **Authorisation Guide**, **Search & Browse**, **Playlists & Artists**, **Albums & Tracks**, **Episodes & Shows** and **Follow**

Step 2



Once opened, in the **Solution Explorer** open the **Pages** section, then open the **Index.cshtml** section and select **Index.cshtml.cs**

Step 3



Then from the **Menu** choose **View** and then **Open**

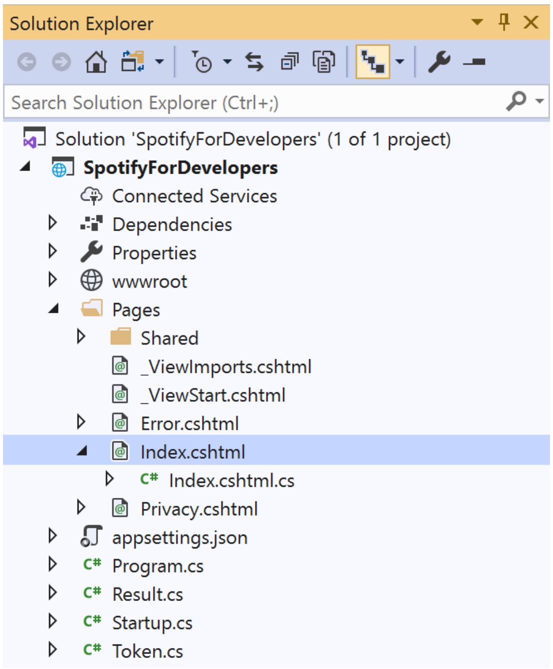
Step 4

In the **Code View** for **Index.cshtml.cs** below the **method** for `public async Task<IActionResult> OnPostCheckUsersFollowingPlaylistAsync(...) { ... }` enter the following **method**:

```
public async Task<IActionResult> OnPostGetUsersFollowedArtistsAsync()
{
    LoadToken();
    var results = await Api.GetUsersFollowedArtistsAsync();
    if (results != null)
    {
        Results = results.Items.Select(result => new Result()
        {
            Id = result.Id,
            Name = result.Name,
            Image = result?.Images?.FirstOrDefault()?.Url
        });
    }
    return Page();
}
```

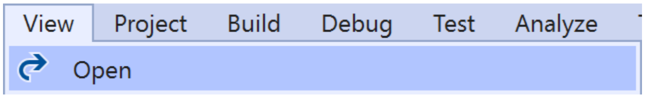
The **method** for `OnPostGetUsersFollowedArtistsAsync` is used to get the **followed artists** for the logged in **user** and populate the **property** for `Results` accordingly.

Step 5



In the **Solution Explorer** in the **Pages** section select **Index.cshtml**

Step 6



Then from the **Menu** choose **View** and then **Open**

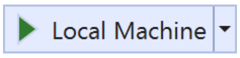
Step 7

Once in the **Code View** for **Index.cshtml** above `<!-- Follow -->` enter the following:

```
<li class="list-group-item">
    <form asp-page-handler="GetUsersFollowedArtists" method="post">
        <button class="btn btn-primary mb-2">
            Get User's Followed Artists
        </button>
    </form>
</li>
```

This `form` will **post** to the **method** for `GetUsersFollowedArtists` and will output to the **Results**.

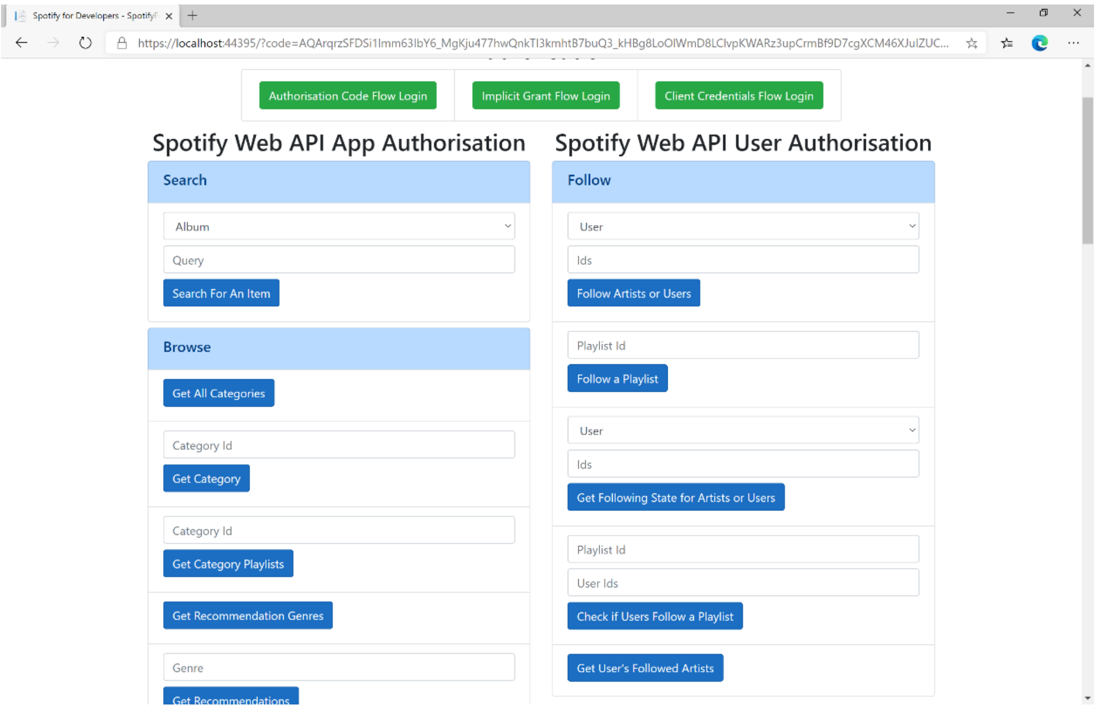
Step 8



Finally, in **Visual Studio 2019** select **IIS Application** to run the **Web Application**

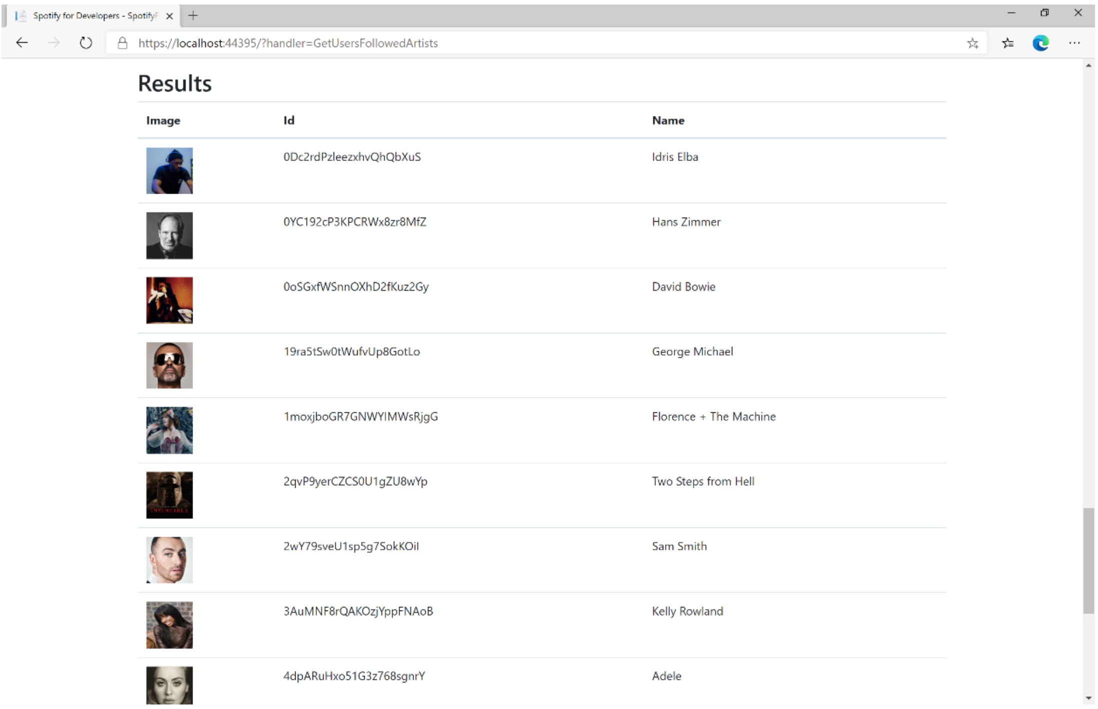
Step 9

Once the **Web Application** is running and you select **Authorisation Code Flow Login** or **Implicit Grant Flow Login** and scroll down you should see something like the following:


Spotify for Developers - Spotify
https://localhost:44395/?code=AQArqrzSFDSi1Imm63lbY6_MgKju477hwQnkTI3kmhtB7buQ3_kHBg8LoOlWmD8LClvpKWARz3upCrmBf9D7cgXCM46XJulZUC...
Authorisation Code Flow Login
Implicit Grant Flow Login
Client Credentials Flow Login
Spotify Web API App Authorisation
Search
Album
Query
Search For An Item
Browse
Get All Categories
Category Id
Get Category
Category Id
Get Category Playlists
Get Recommendation Genres
Genre
Get Recommendations
Spotify Web API User Authorisation
Follow
User
Ids
Follow Artists or Users
Playlist Id
Follow a Playlist
User
Ids
Get Following State for Artists or Users
Playlist Id
User Ids
Check if Users Follow a Playlist
Get User's Followed Artists

Step 10

You can select **Get User's Followed Artists** and scroll down to view **Results** like the following:


Spotify for Developers - Spotify
https://localhost:44395/?handler=GetUsersFollowedArtists

Results

Image	Id	Name
	0Dc2rdPzleezxhvQhQbXuS	Idris Elba
	0YC192cP3KPCRWx8zr8MfZ	Hans Zimmer
	0oSGxfWSnnOXhD2fKuz2Gy	David Bowie
	19ra5tSw0tWufvUp8GotLo	George Michael
	1moxjboGR7GNWYIMWsRjgG	Florence + The Machine
	2qvP9yerCZCS0U1gZU8wYp	Two Steps from Hell
	2wY79sveU1sp5g7SokKOil	Sam Smith
	3AuMNF8rQAKOzjYppFNAoB	Kelly Rowland
	4dpARuHxo51G3z768sgnrY	Adele

Step 11

You can stop the **web application** in **Visual Studio 2019** by selecting the **Stop debugging** button

Step 12

You can choose to exit **Visual Studio 2019** by selecting the **Close** button in the top right of the **application** as that completes this part of the workshop

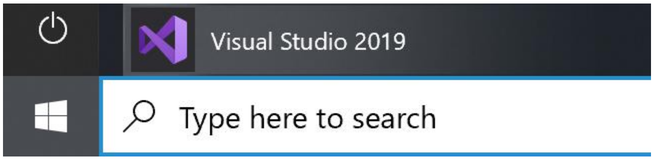
Unfollow Artists or Users

Remove the current user as a follower of one or more artists or other Spotify users.

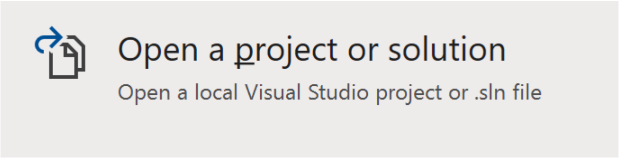
DELETE https://api.spotify.com/v1/me/following	
Header	
Authorization	User Token from Spotify Accounts service with **user-follow-modify** scope
Query Parameter	
type	Type of artist or user
ids	Comma-separated list of the artist or the user Spotify Ids - maximum 50

Header	**Response**
Success	
HTTP Status 204 No Content	Empty Response Body
Error	
Error Code	Error Object

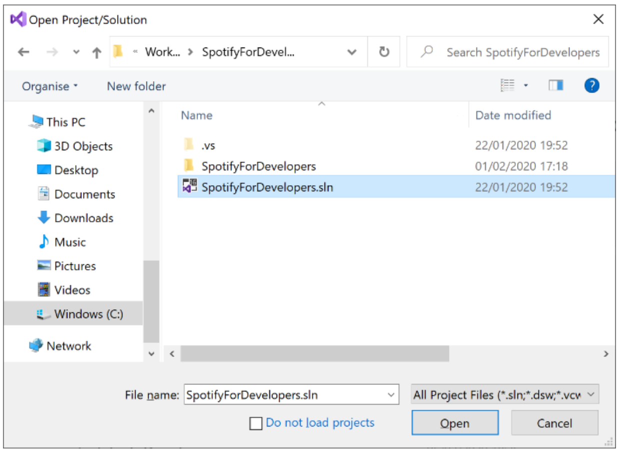
Step 1



If you chose to close **Visual Studio 2019** previously, in **Windows 10** choose **Start**, and then from the **Start Menu** find and select **Visual Studio 2019**


Open a project or solution
Open a local Visual Studio project or .sln file

Once done, from the **Get started** screen for **Visual Studio 2019** select **Open a project or solution**

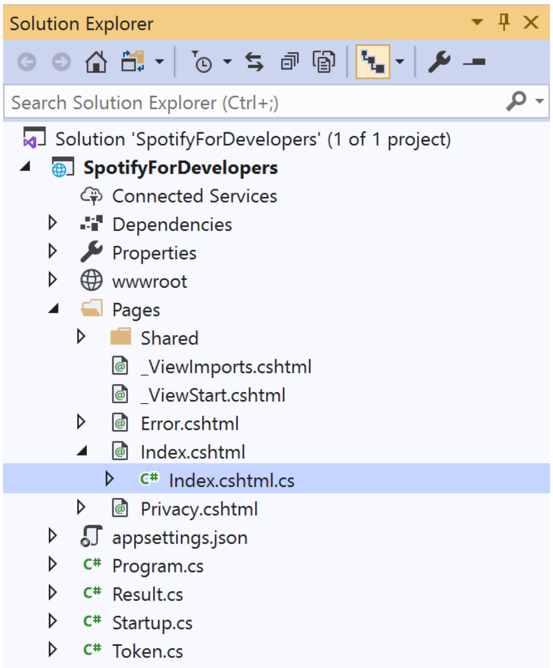


Name	Date modified
.vs	22/01/2020 19:52
SpotifyForDevelopers	01/02/2020 17:18
SpotifyForDevelopers.sln	22/01/2020 19:52

File name: SpotifyForDevelopers.sln

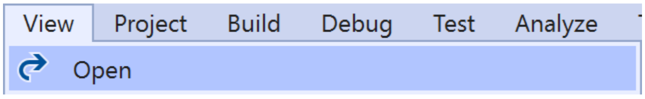
Then locate and select **SpotifyForDevelopers.sln** and select **Open** if you don't have this file already then please follow the previous parts of the workshop including **Getting Started**, **Authorisation Guide**, **Search & Browse**, **Playlists & Artists**, **Albums & Tracks**, **Episodes & Shows** and **Follow**

Step 2



Once opened, in the **Solution Explorer** open the **Pages** section, then open the **Index.cshtml** section and select **Index.cshtml.cs**

Step 3



Then from the **Menu** choose **View** and then **Open**

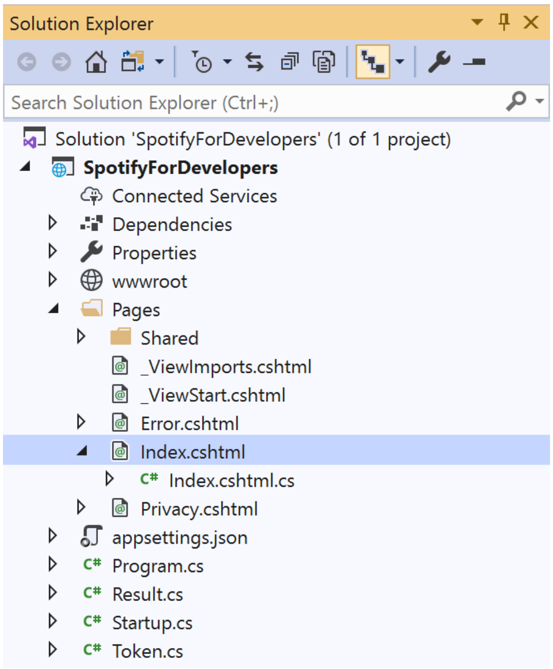
Step 4

In the **Code View** for **Index.cshtml.cs** below the **method** for `public async Task<IActionResult> OnPostGetUsersFollowedArtistsAsync(...) { ... }` enter the following **method**:

```
public async Task<IActionResult> OnPostUnfollowArtistsOrUsersAsync(string value, FollowType
option)
{
    LoadToken();
    var values = value.Split(",").ToList();
    var result = await Api.UnfollowArtistsOrUsersAsync(values, option);
    if (result != null)
    {
        Results = new List<Result>() { new Result()
        {
            Name = result.Success.ToString()
        }};
    }
    return Page();
}
```

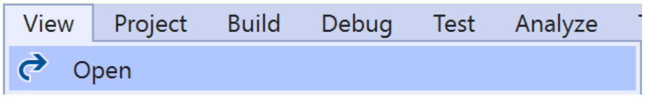
The **method** for `OnPostUnfollowArtistsOrUsersAsync` is used to **unfollow** an **artist** or **user** by **Artist Id** or **User Id** on Spotify with the `value` and selected `option` and populate the **property** for `Results` accordingly with the **success** of the operation.

Step 5



In the **Solution Explorer** in the **Pages** section select **Index.cshtml**

Step 6



Then from the **Menu** choose **View** and then **Open**

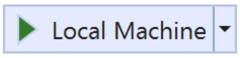
Step 7

Once in the **Code View** for **Index.cshtml** above `<!-- Follow -->` enter the following:

```
<li class="list-group-item">
    <form asp-page-handler="UnfollowArtistsOrUsers" method="post">
        <select asp-for="Option" class="form-control mb-2"
            asp-items="Html.GetEnumSelectList<Spotify.NetStandard.Enums.FollowType>()">
        </select>
        <input asp-for="Value" placeholder="Id" class="form-control mb-2" />
        <button class="btn btn-primary mb-2">
            Unfollow Artists or Users
        </button>
    </form>
</li>
```

This `form` will **post** to the **method** for `UnfollowArtistsOrUsers` with the `Value` of the **Ids** and the `Option` for the **Artist** or **User** and will output to the **Results**.

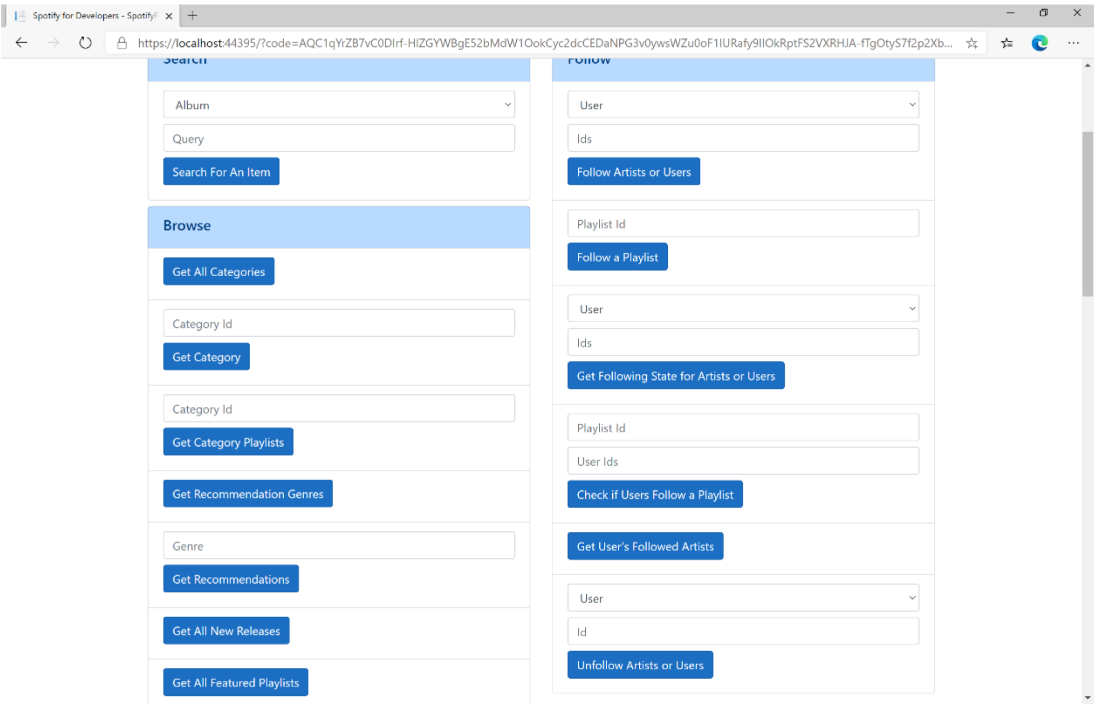
Step 8



Finally, in **Visual Studio 2019** select **IIS Application** to run the **Web Application**

Step 9

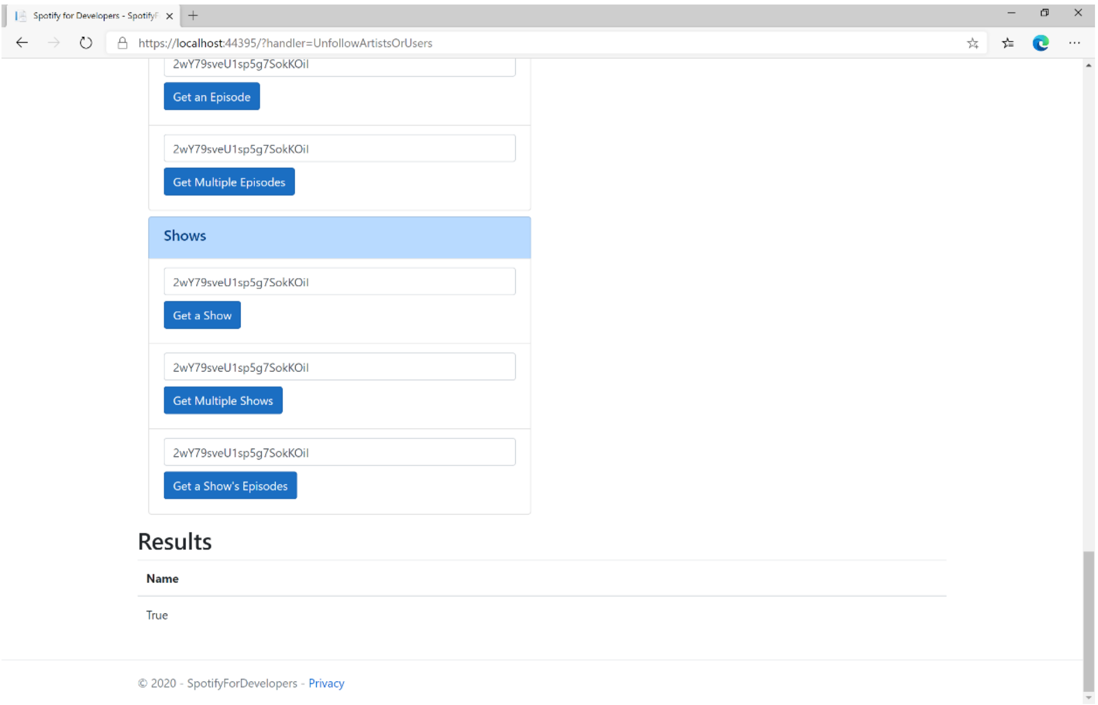
Once the **Web Application** is running and you select **Authorisation Code Flow Login** or **Implicit Grant Flow Login** and scroll down you should see something like the following:



https://localhost:44395/?code=AQC1qYrZB7vC0DIrf-HIZGYWBgE52bMdW1OokCyc2dcCEDaNPG3v0ywsWZu0oF1IURafy9lIOkRptFS2VXRHJA-fTgOtyS7f2p2Xb...

Step 10

You can then enter an **Artist Id** from **Get All New Releases** or the same **Artist Id** from **Follow Artists or Users** to use and above **Unfollow Artists or Users** choose **Artist** and then select **Unfollow Artists or Users** and scroll down to view **Results** like the following:



https://localhost:44395/?handler=UnfollowArtistsOrUsers

2wY79sveU1sp5g7SokKOil

Results

Name
True

© 2020 - SpotifyForDevelopers - Privacy

Step 11

■

You can stop the **web application** in **Visual Studio 2019** by selecting the **Stop debugging** button

Step 12

×

You can choose to exit **Visual Studio 2019** by selecting the **Close** button in the top right of the **application** as that completes this part of the workshop

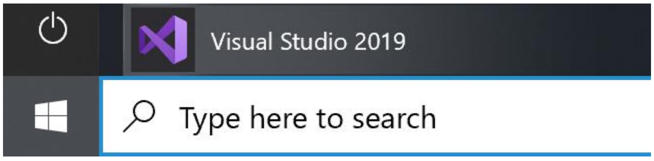
Unfollow Playlist

Remove the current user as a follower of a playlist.

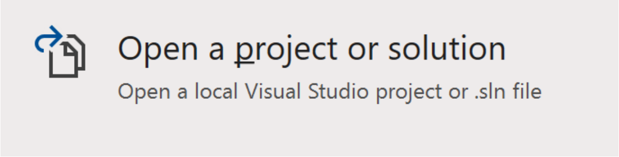
DELETE https://api.spotify.com/v1/playlists/{playlist_id}/followers	
Header	
Authorization	User Token from Spotify Accounts service with **playlist-modify-public** scope for publicly followed playlist and **playlist-modify-private** for a privately followed playlist
Path Parameter	
playlist_id	Spotify Id of the playlist

Header	Response
Success	
HTTP Status 200 OK	Empty Response Body
Error	
Error Code	Error Object

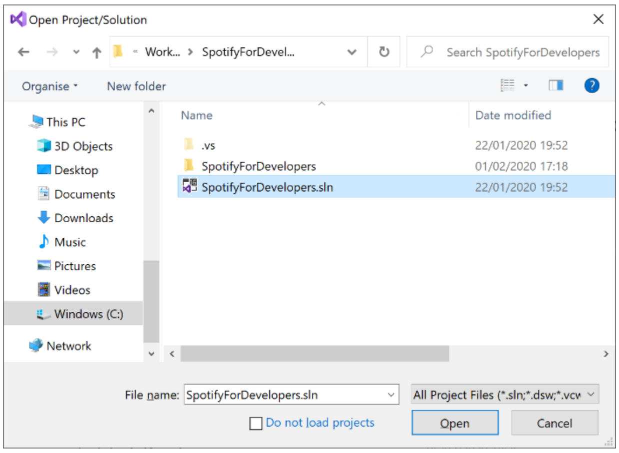
Step 1



If you chose to close **Visual Studio 2019** previously, in **Windows 10** choose **Start**, and then from the **Start Menu** find and select **Visual Studio 2019**


Open a project or solution
Open a local Visual Studio project or .sln file

Once done, from the **Get started** screen for **Visual Studio 2019** select **Open a project or solution**



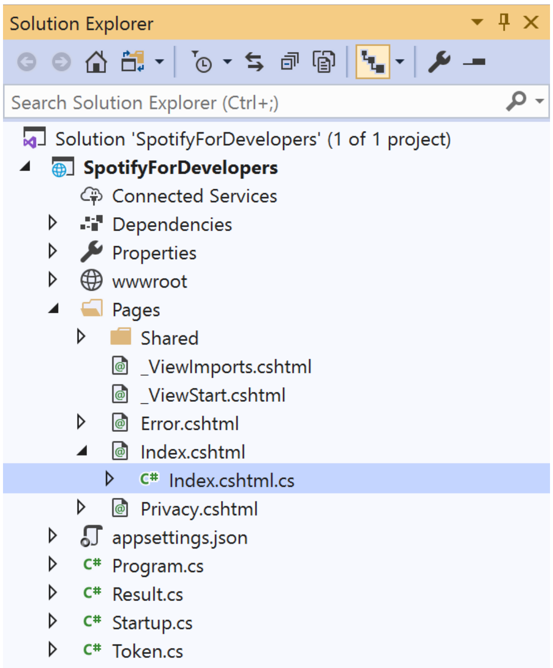
Name	Date modified
.vs	22/01/2020 19:52
SpotifyForDevelopers	01/02/2020 17:18
SpotifyForDevelopers.sln	22/01/2020 19:52

File name: SpotifyForDevelopers.sln

All Project Files (*.sln;*.dsw;*.vcw

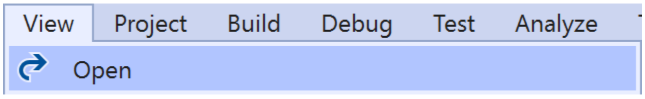
Then locate and select **SpotifyForDevelopers.sln** and select **Open** if you don't have this file already then please follow the previous parts of the workshop including **Getting Started**, **Authorisation Guide**, **Search & Browse**, **Playlists & Artists**, **Albums & Tracks**, **Episodes & Shows** and **Follow**

Step 2



Once opened, in the **Solution Explorer** open the **Pages** section, then open the **Index.cshtml** section and select **Index.cshtml.cs**

Step 3



Then from the **Menu** choose **View** and then **Open**

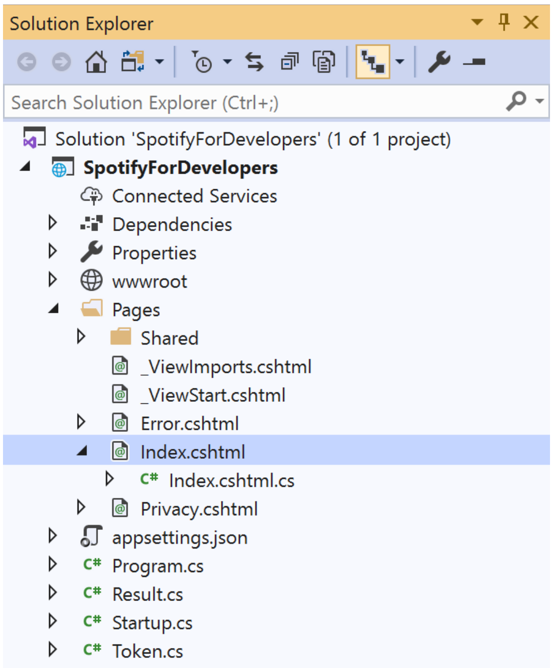
Step 4

In the **Code View** for **Index.cshtml.cs** below the **method** for `public async Task<IActionResult> OnPostUnfollowArtistsOrUsersAsync(...) { ... }` enter the following **method**:

```
public async Task<IActionResult> OnPostUnfollowPlaylistAsync(string value)
{
    LoadToken();
    var result = await Api.UnfollowPlaylistAsync(value);
    if (result != null)
    {
        Results = new List<Result>() { new Result()
        {
            Name = result.Success.ToString()
        }};
    }
    return Page();
}
```

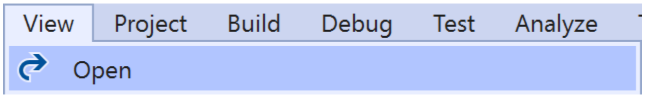
The **method** for `OnPostUnfollowPlaylistAsync` is used to **unfollow** a **playlist** by **Playlist Id** on Spotify with the `value` and populate the **property** for `Results` accordingly with the **success** of the operation.

Step 5



In the **Solution Explorer** in the **Pages** section select **Index.cshtml**

Step 6



Then from the **Menu** choose **View** and then **Open**

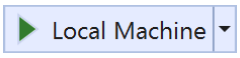
Step 7

Once in the **Code View** for **Index.cshtml** above `<!-- Follow -->` enter the following:

```
<li class="list-group-item">
    <form asp-page-handler="UnfollowPlaylist" method="post">
        <input asp-for="Value" placeholder="Playlist Id" class="form-control mb-2" />
        <button class="btn btn-primary mb-2">
            Unfollow a Playlist
        </button>
    </form>
</li>
```

This `form` will **post** to the **method** for `UnfollowPlaylist` with the `Value` of the **Playlist Id** and will output to the **Results**.

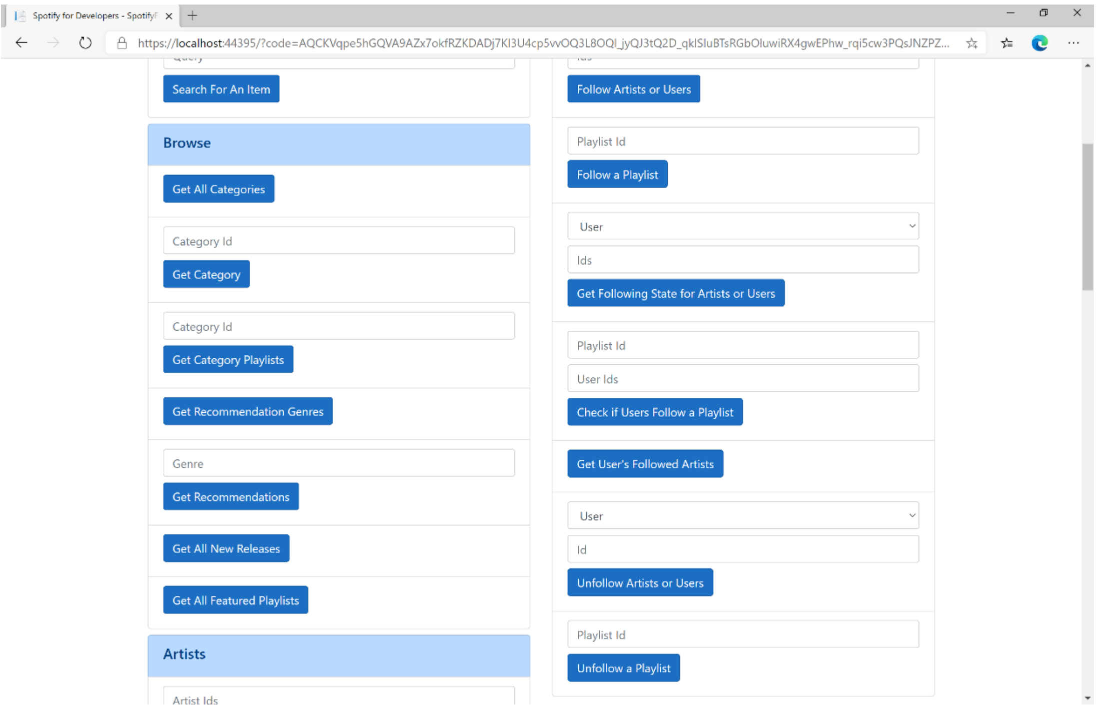
Step 8



Finally, in **Visual Studio 2019** select **IIS Application** to run the **Web Application**

Step 9

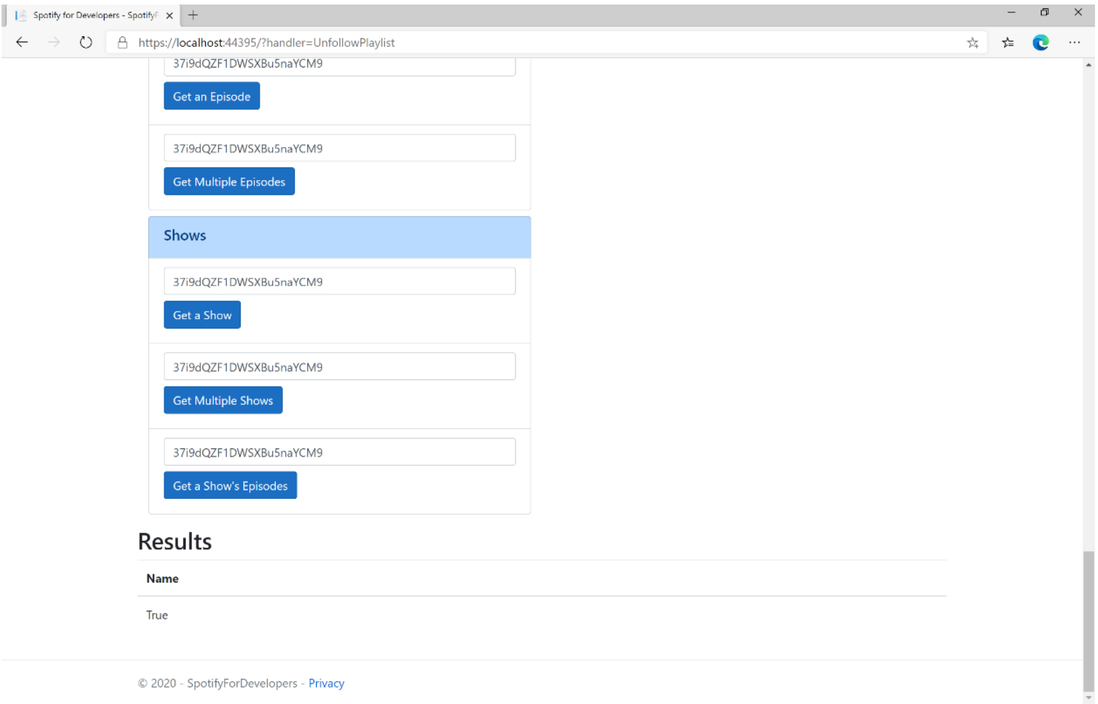
Once the **Web Application** is running and you select **Authorisation Code Flow Login** or **Implicit Grant Flow Login** and scroll down you should see something like the following:



https://localhost:44395/?code=AQCKVqpe5hGQVA9AZx7okfRZKDADj7KI3U4cp5vvOQ3L8OQl_jyQJ3tQ2D_qklSluBTsRGbOluwiRX4gwEPhw_rqi5cw3PQsJNZPZ...

Step 10

You can then enter a **Playlist Id** from **Get All Featured Playlists** or the same **Playlist Id** from **Follow a Playlist** to use and select **Unfollow a Playlist** and scroll down to view **Results** like the following:



https://localhost:44395/?handler=UnfollowPlaylist

37i9dQZF1DWSXBu5naYCM9

37i9dQZF1DWSXBu5naYCM9

37i9dQZF1DWSXBu5naYCM9

37i9dQZF1DWSXBu5naYCM9

37i9dQZF1DWSXBu5naYCM9

Results

Name
True

© 2020 - SpotifyForDevelopers - Privacy

Step 11

You can stop the **web application** in **Visual Studio 2019** by selecting the **Stop debugging** button

Step 12

You can exit **Visual Studio 2019** by selecting the **Close** button in the top right of the **application** as that completes this part of the workshop

Playlists

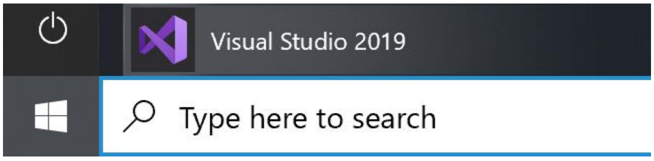
Get a Playlist

Get a playlist owned by a Spotify user.

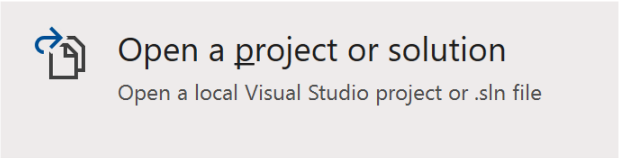
https://api.spotify.com/v1/playlists/{playlist_id}	
Header	
Authorization	Valid Access Token from Spotify Accounts service
Path Parameter	
playlist_id	Spotify Id for the playlist
Query Parameter	
fields	Filters for the query: comma-separated list of fields to return e.g. description, uri
market	ISO 3166-1 alpha-2 country code e.g. "GB". Provide to apply Track Relinking

Header	Response
Success	
HTTP Status 200 OK	Playlist Object
Error	
Error Code	Error Object

Step 1



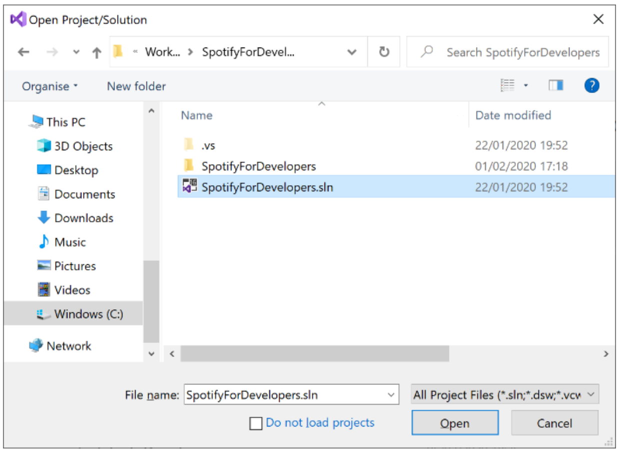
In **Windows 10** choose **Start**, and then from the **Start Menu** find and select **Visual Studio 2019**



Open a project or solution

Open a local Visual Studio project or .sln file

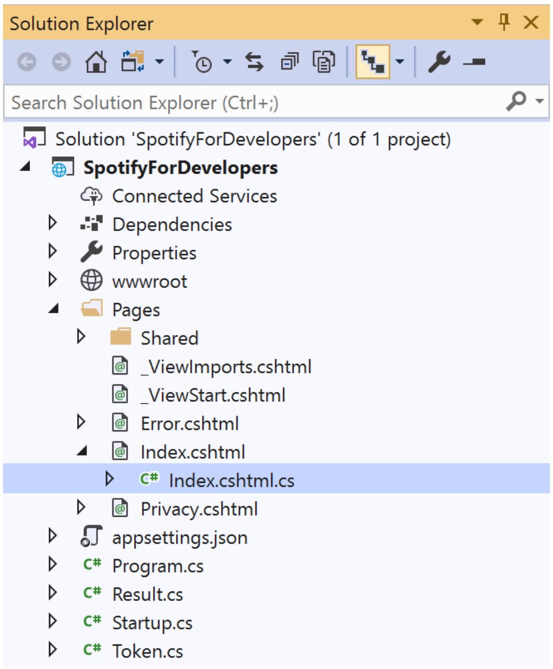
Once done, from the **Get started** screen for **Visual Studio 2019** select **Open a project or solution**



File name: SpotifyForDevelopers.sln

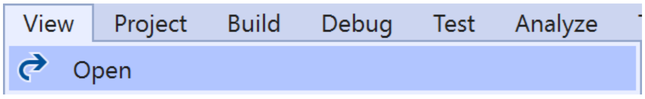
Then locate and select **SpotifyForDevelopers.sln** and select **Open** if you don't have this file already then please follow the previous parts of the workshop including **Getting Started**, **Authorisation Guide**, **Search & Browse**, **Playlists & Artists**, **Albums & Tracks**, **Episodes & Shows** and **Follow**

Step 2



Once opened, in the **Solution Explorer** open the **Pages** section, then open the **Index.cshtml** section and select **Index.cshtml.cs**

Step 3



Then from the **Menu** choose **View** and then **Open**

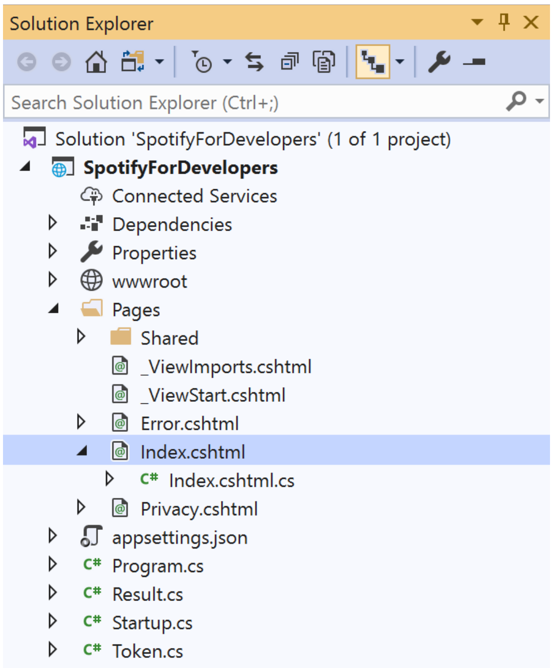
Step 4

In the **Code View** for **Index.cshtml.cs** below the **method** for `public async Task<IActionResult> OnPostUnfollowPlaylistAsync() { ... }` enter the following **method**:

```
public async Task<IActionResult> OnPostGetPlaylistAsync(string value)
{
    LoadToken();
    var result = await Api.GetPlaylistAsync(value);
    if (result != null)
    {
        Results = new List<Result> { new Result()
        {
            Id = result.Id,
            Name = result.Name,
            Image = result?.Images?.FirstOrDefault()?.Url
        }};
    }
    return Page();
}
```

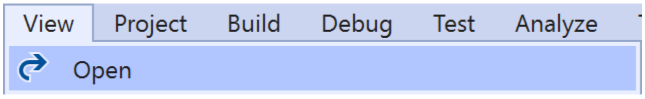
The **method** for `OnPostGetPlaylistAsync` is used to get a **playlist** by **Playlist Id** on Spotify with the `value` and populate the **property** for `Results` accordingly.

Step 5



In the **Solution Explorer** in the **Pages** section select **Index.cshtml**

Step 6



Then from the **Menu** choose **View** and then **Open**

Step 7

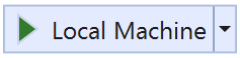
Once in the **Code View** for **Index.cshtml** above `<!-- Spotify Web API User Authorisation -->` enter the following:

```
<ul class="list-group mb-2">
    <li class="list-group-item list-group-item-primary">
        <h5 class="list-group-item-heading">Playlists</h5>
    </li>
    <li class="list-group-item">
        <form asp-page-handler="GetPlaylist" method="post">
            <input asp-for="Value" placeholder="Playlist Id" class="form-control mb-2" />
            <button class="btn btn-primary mb-2">
                Get a Playlist
            </button>
        </form>
    </li>

    <!-- Playlists -->
</ul>
```

This `form` will **post** to the **method** for `OnPostGetPlaylistAsync` with the `Value` as the **Playlist Id** and will output to the **Results**. This `form` could also be used with **Spotify Web API App Authorisation** as it does not require a **User Token**

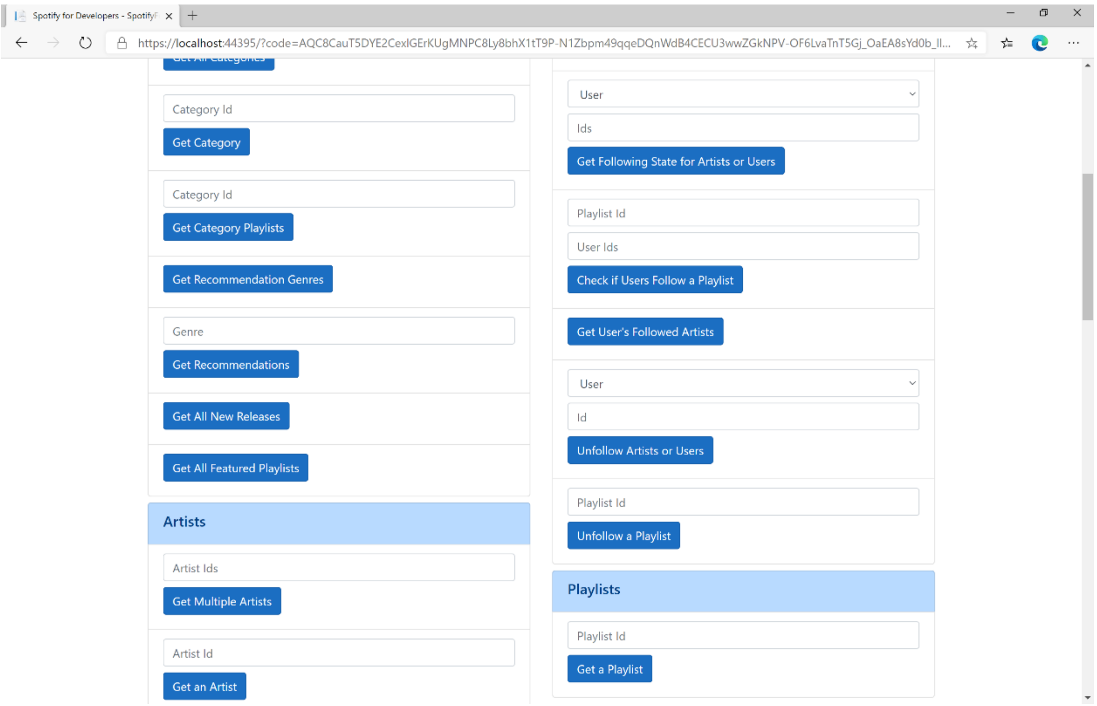
Step 8



Finally, in **Visual Studio 2019** select **IIS Application** to run the **Web Application**

Step 9

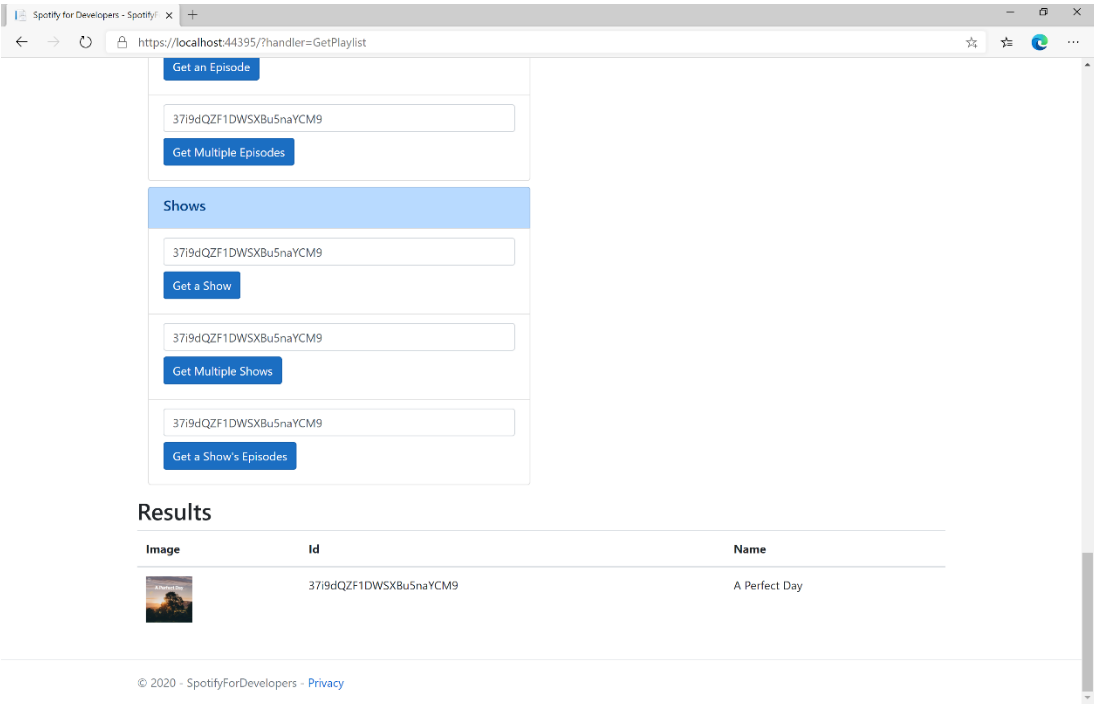
Once the **Web Application** is running and you select **Authorisation Code Flow Login** or **Implicit Grant Flow Login** and scroll down you should see something like the following:



https://localhost:44395/?code=AQC8CauT5DYE2CexlGErKUgMNPC8Ly8bhX1tT9P-N1Zbpm49qqeDQnWdB4CECU3wwZGkNPV-OF6LvaTnT5Gj_OaEA8sYd0b_Il...

Step 10

You can then enter a **Playlist Id** from **Get All Featured Playlists** and select **Get a Playlist** and scroll down to view **Results** like the following:



https://localhost:44395/?handler=GetPlaylist

37i9dQZF1DWSXBu5naYCM9

37i9dQZF1DWSXBu5naYCM9

37i9dQZF1DWSXBu5naYCM9

37i9dQZF1DWSXBu5naYCM9

Results

Image	Id	Name
	37i9dQZF1DWSXBu5naYCM9	A Perfect Day

© 2020 - SpotifyForDevelopers - Privacy

Step 11

■

You can stop the **web application** in **Visual Studio 2019** by selecting the **Stop debugging** button

Step 12

×

You can choose to exit **Visual Studio 2019** by selecting the **Close** button in the top right of the **application** as that completes this part of the workshop

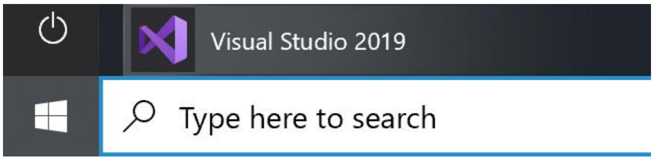
Get a Playlist's Tracks

Get full details of the tracks of a playlist owned by a Spotify user.

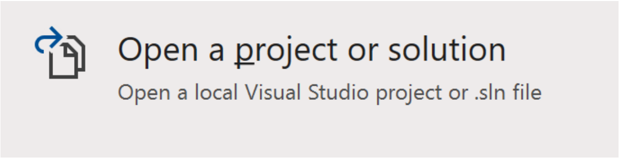
GET https://api.spotify.com/v1/playlists/{playlist_id}/tracks	
Header	
Authorization	Valid Access Token from Spotify Accounts service
Path Parameter	
playlist_id	Spotify Id for the playlist
Query Parameter	
fields	Filters for the query: comma-separated list of fields to return e.g. total, limit
limit	Maximum number of results to return
offset	Index of first result to return
market	ISO 3166-1 alpha-2 country code e.g. "GB". Provide to apply Track Relinking

Header	Response
Success	
HTTP Status 200 OK	Array of Playlist Track Objects wrapped in a Paging Object
Error	
Error Code	Error Object

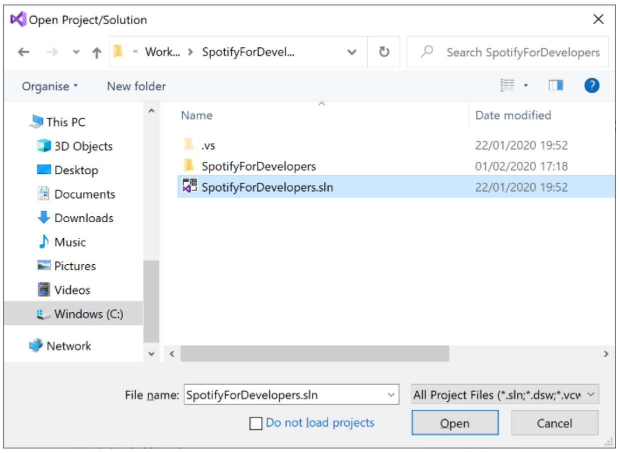
Step 1



If you chose to close **Visual Studio 2019** previously, in **Windows 10** choose **Start**, and then from the **Start Menu** find and select **Visual Studio 2019**


Open a project or solution
Open a local Visual Studio project or .sln file

Once done, from the **Get started** screen for **Visual Studio 2019** select **Open a project or solution**

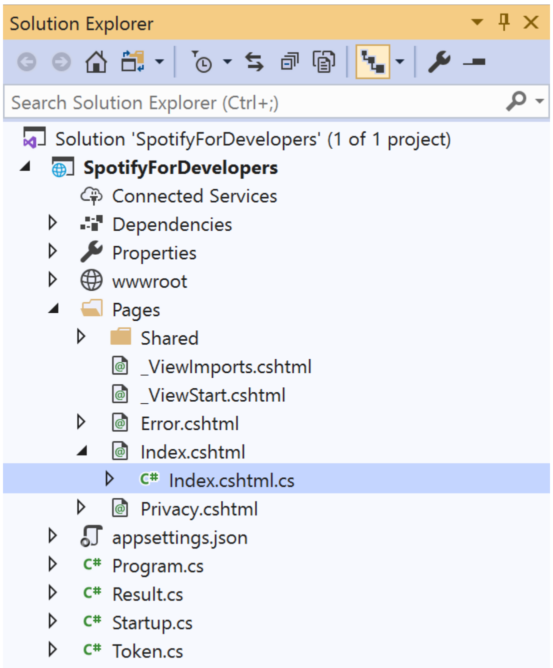


Name	Date modified
.vs	22/01/2020 19:52
SpotifyForDevelopers	01/02/2020 17:18
SpotifyForDevelopers.sln	22/01/2020 19:52

File name: SpotifyForDevelopers.sln

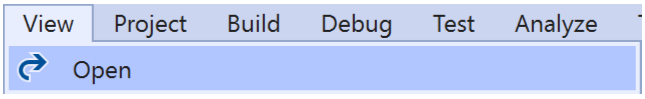
Then locate and select **SpotifyForDevelopers.sln** and select **Open** if you don't have this file already then please follow the previous parts of the workshop including **Getting Started**, **Authorisation Guide**, **Search & Browse**, **Playlists & Artists**, **Albums & Tracks**, **Episodes & Shows**, **Follow** and **Playlists**

Step 2



Once opened, in the **Solution Explorer** open the **Pages** section, then open the **Index.cshtml** section and select **Index.cshtml.cs**

Step 3



Then from the **Menu** choose **View** and then **Open**

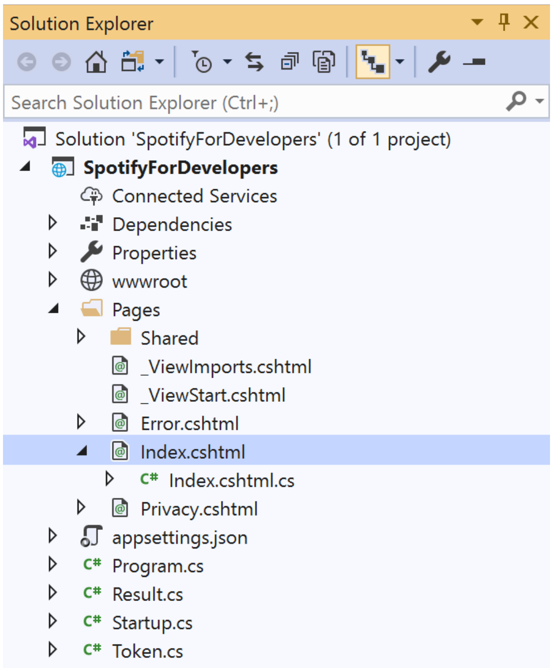
Step 4

In the **Code View** for **Index.cshtml.cs** below the **method** for `public async Task<IActionResult> OnPostGetPlaylistAsync(...) { ... }` enter the following **method**:

```
public async Task<IActionResult> OnPostGetPlaylistTracksAsync(string value)
{
    LoadToken();
    var results = await Api.GetPlaylistTracksAsync(value);
    if (results?.Items != null)
    {
        Results = results.Items.Select(result => new Result()
        {
            Id = result?.Track.Id,
            Name = result?.Track.Name,
            Image = result?.Track.Album?.Images?.FirstOrDefault()?.Url,
            Inner = new Result()
            {
                Id = result?.Track?.Artists?.FirstOrDefault()?.Id,
                Name = result?.Track?.Artists?.FirstOrDefault()?.Name
            }
        });
    }
    return Page();
}
```

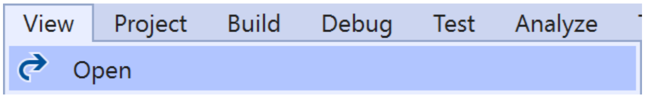
The **method** for `OnPostGetPlaylistTracksAsync` is used to get the **tracks** for a **playlist** by **Playlist Id** on Spotify with the `value` and populate the **property** for `Results` accordingly.

Step 5



In the **Solution Explorer** in the **Pages** section select **Index.cshtml**

Step 6



Then from the **Menu** choose **View** and then **Open**

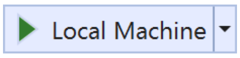
Step 7

Once in the **Code View** for **Index.cshtml** above `<!-- Playlists -->` enter the following:

```
<li class="list-group-item">
    <form asp-page-handler="GetPlaylistTracks" method="post">
        <input asp-for="Value" placeholder="Playlist Id" class="form-control mb-2" />
        <button class="btn btn-primary mb-2">
            Get a Playlist's Tracks
        </button>
    </form>
</li>
```

This `form` will **post** to the **method** for `OnPostGetPlaylistTracksAsync` with the `Value` as the **Playlist Id** and will output to the **Results**. This `form` could also be used with **Spotify Web API App Authorisation** as it does not require a **User Token**.

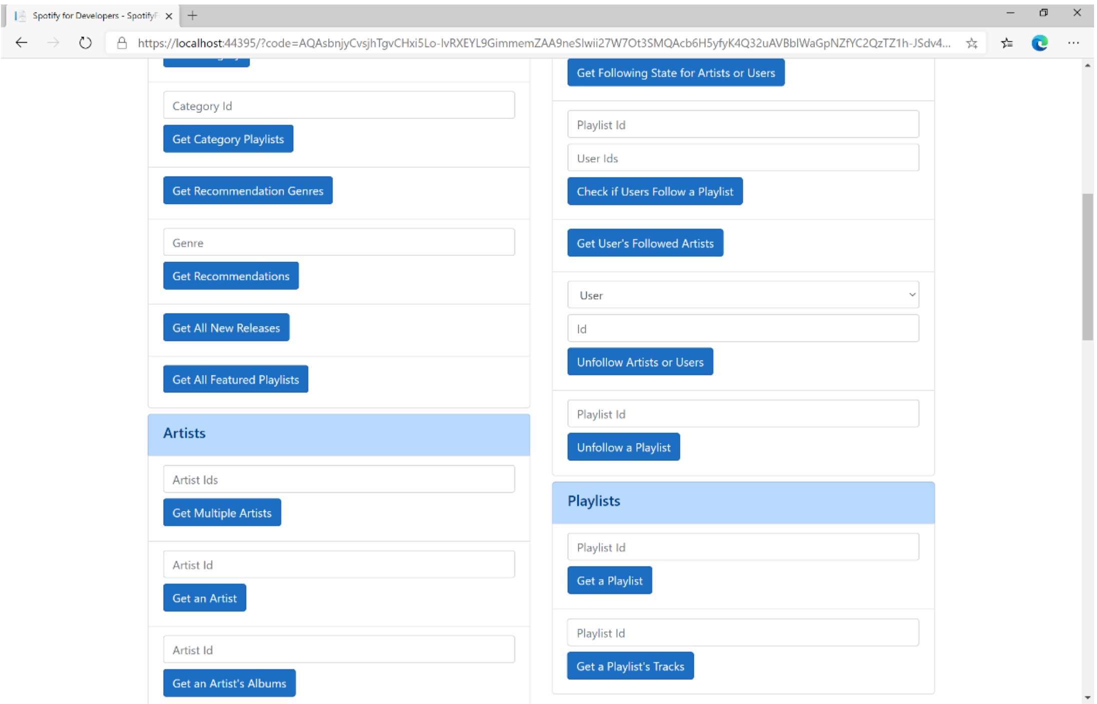
Step 8



Finally, in **Visual Studio 2019** select **IIS Application** to run the **Web Application**

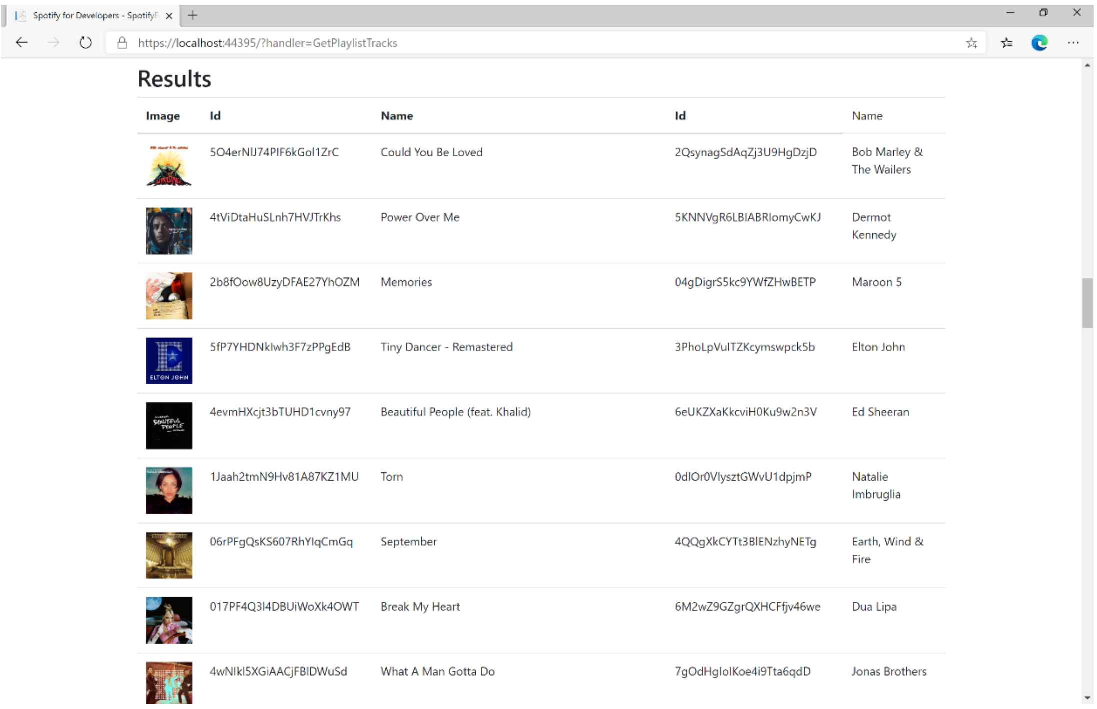
Step 9

Once the **Web Application** is running and you select **Authorisation Code Flow Login** or **Implicit Grant Flow Login** and scroll down you should see something like the following:


Spotify for Developers - Spotify
https://localhost:44395/?code=AQAsbnjyCvsjhTgvCHxi5Lo-lvRXEYL9GimmemZAA9neSlwii27W7Ot3SMQAcb6H5yfyK4Q32uAVBblWaGpNZfYC2QzTZ1h-JSdv4...
Category Id
Get Category Playlists
Get Recommendation Genres
Genre
Get Recommendations
Get All New Releases
Get All Featured Playlists
Artists
Artist Ids
Get Multiple Artists
Artist Id
Get an Artist
Artist Id
Get an Artist's Albums
Get Following State for Artists or Users
Playlist Id
User Ids
Check if Users Follow a Playlist
Get User's Followed Artists
User
Id
Unfollow Artists or Users
Playlist Id
Unfollow a Playlist
Playlists
Playlist Id
Get a Playlist
Playlist Id
Get a Playlist's Tracks

Step 10

You can then enter a **Playlist Id** from **Get All Featured Playlists** and select **Get Playlist Tracks** and scroll down to view **Results** like the following:


Spotify for Developers - Spotify
https://localhost:44395/?handler=GetPlaylistTracks

Results

Image	Id	Name	Id	Name
	5O4erNIJ74PIF6kGol1ZrC	Could You Be Loved	2QsynagSdAqZj3U9HgDzjD	Bob Marley & The Wailers
	4tViDtaHuSLnh7HVJTrKhs	Power Over Me	5KNNVgR6LBIABRlomyCwKJ	Dermot Kennedy
	2b8fOow8UzyDFAE27YhOZM	Memories	04gDigrS5kc9YWfZHwBETP	Maroon 5
	5fP7YHDNklwh3F7zPPgEdB	Tiny Dancer - Remastered	3PhoLpVuITZKcymswpck5b	Elton John
	4evmHXcjt3bTUHD1cvny97	Beautiful People (feat. Khalid)	6eUKZXaKkcviH0Ku9w2n3V	Ed Sheeran
	1Jaah2tmN9Hv81A87KZ1MU	Torn	0dlOr0VlysztGWvU1dpjmP	Natalie Imbruglia
	06rPFgQsKS607RhYlqCmGq	September	4QQgXkCYTt3BlENzhyNETg	Earth, Wind & Fire
	017PF4Q3l4DBUiWoXk4OWT	Break My Heart	6M2wZ9GZgrQXHCFfjv46we	Dua Lipa
	4wNIkl5XGiAACjFBIDWuSd	What A Man Gotta Do	7gOdHglolKoe4i9Tta6qdD	Jonas Brothers

Step 11

You can stop the **web application** in **Visual Studio 2019** by selecting the **Stop debugging** button

Step 12

You can choose to exit **Visual Studio 2019** by selecting the **Close** button in the top right of the **application** as that completes this part of the workshop

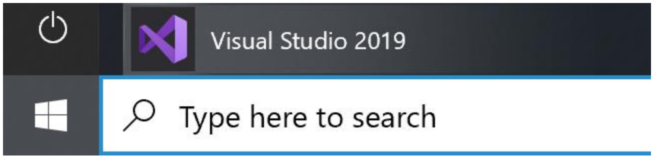
Create a Playlist

Create a playlist for a Spotify user

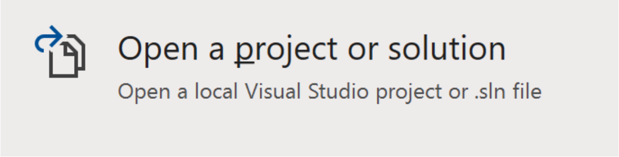
POST https://api.spotify.com/v1/users/{user_id}/playlists	
Header	
Authorization	User Token from Spotify Accounts service with **playlist-modify-public** scope for publicly followed playlist and **playlist-modify-private** for a privately followed playlist
Path Parameter	
user_id	User's Spotify User Id
Body Parameter	
name	Name for the new playlist
public	If true, the playlist will be public, if false it will be private
collaborative	If true, the playlist will be collaborative and other users will be able to modify the playlist
description	Playlist description

Header	Response
Success	
HTTP Status 200 OK	
Http Status 201 Created	
Error	
HTTP Status 403 Forbidden	Returned when not Authorised
Error Code	Error Object

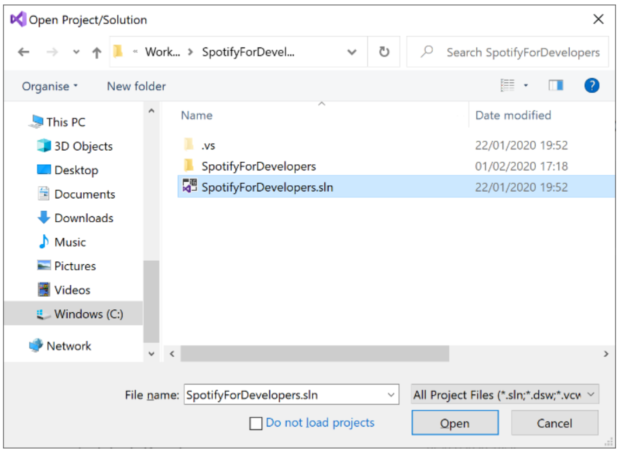
Step 1



If you chose to close **Visual Studio 2019** previously, in **Windows 10** choose **Start**, and then from the **Start Menu** find and select **Visual Studio 2019**

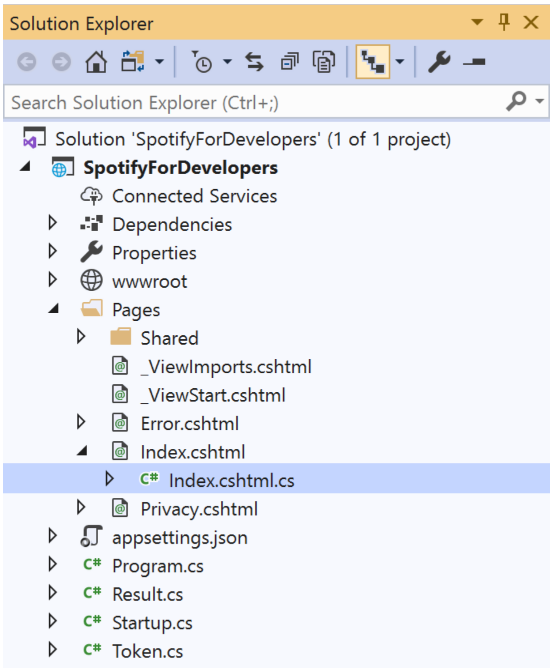

Open a project or solution
Open a local Visual Studio project or .sln file

Once done, from the **Get started** screen for **Visual Studio 2019** select **Open a project or solution**



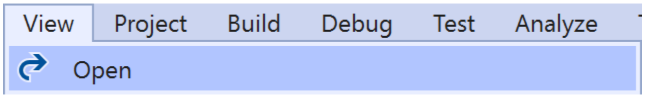
Then locate and select **SpotifyForDevelopers.sln** and select **Open** if you don't have this file already then please follow the previous parts of the workshop including **Getting Started**, **Authorisation Guide**, **Search & Browse**, **Playlists & Artists**, **Albums & Tracks**, **Episodes & Shows**, **Follow** and **Playlists**

Step 2



Once opened, in the **Solution Explorer** open the **Pages** section, then open the **Index.cshtml** section and select **Index.cshtml.cs**

Step 3



Then from the **Menu** choose **View** and then **Open**

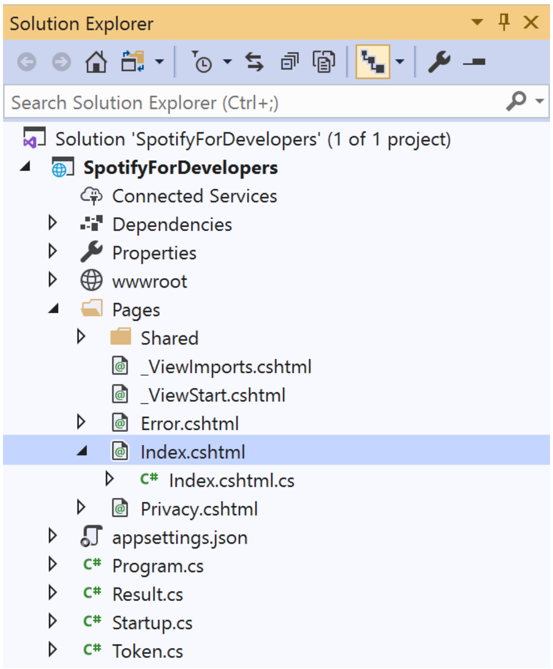
Step 4

In the **Code View** for **Index.cshtml.cs** below the **method** for `public async Task<IActionResult> OnPostGetPlaylistTracksAsync(...) { ... }` enter the following **method**:

```
public async Task<IActionResult> OnPostCreatePlaylistAsync(string value, string option)
{
    LoadToken();
    var result = await Api.CreatePlaylistAsync(value, option);
    if (result != null)
    {
        Results = new List<Result> { new Result()
        {
            Id = result.Id,
            Name = result.Name
        }};
    }
    return Page();
}
```

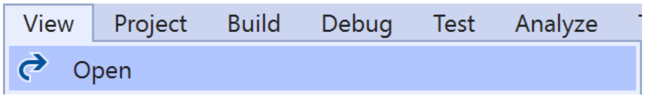
The **method** for `OnPostCreatePlaylistAsync` is used to **create** a **playlist** for a **user** by **User Id** with the `value` and `option` with a given **Name** on Spotify and populate the **property** for `Results` accordingly.

Step 5



In the **Solution Explorer** in the **Pages** section select **Index.cshtml**

Step 6



Then from the **Menu** choose **View** and then **Open**

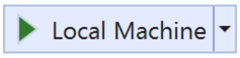
Step 7

Once in the **Code View** for **Index.cshtml** above `<!-- Playlists -->` enter the following:

```
<li class="list-group-item">
    <form asp-page-handler="CreatePlaylist" method="post">
        <input asp-for="Value" placeholder="User Id" class="form-control mb-2" />
        <input asp-for="Option" placeholder="Name" class="form-control mb-2" />
        <button class="btn btn-primary mb-2">
            Create a Playlist
        </button>
    </form>
</li>
```

This `form` will **post** to the **method** for `CreatePlaylist` with the `Value` of the **User Id** and the `Option` of the **Name** and will output to the **Results**.

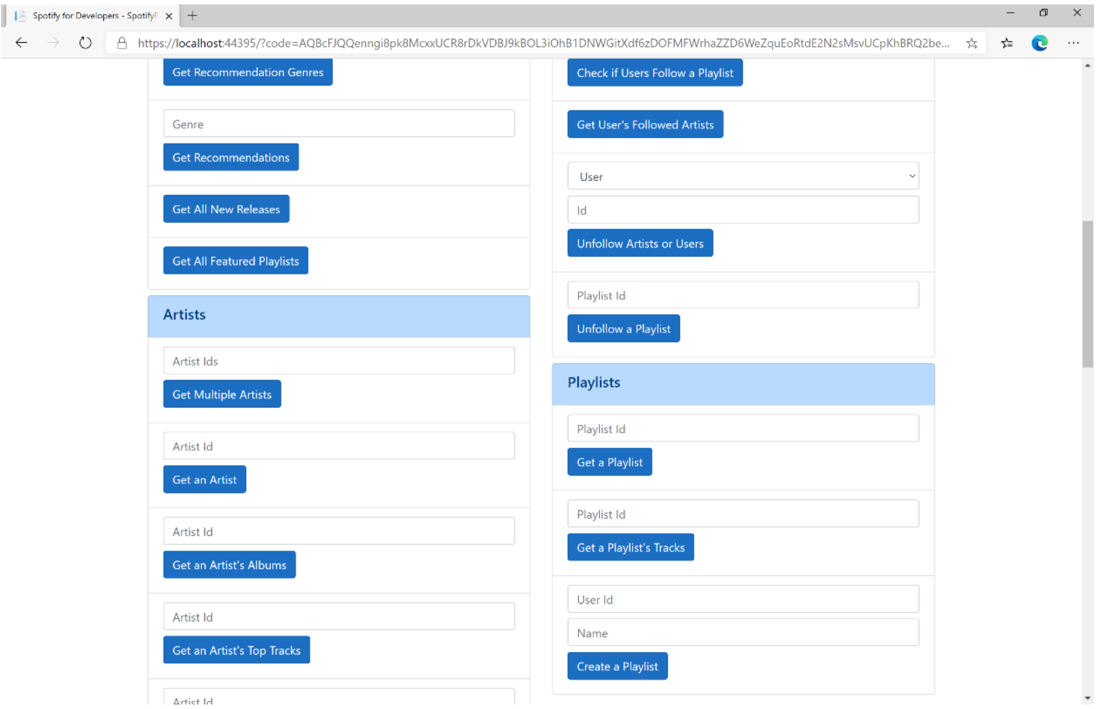
Step 8



Finally, in **Visual Studio 2019** select **IIS Application** to run the **Web Application**

Step 9

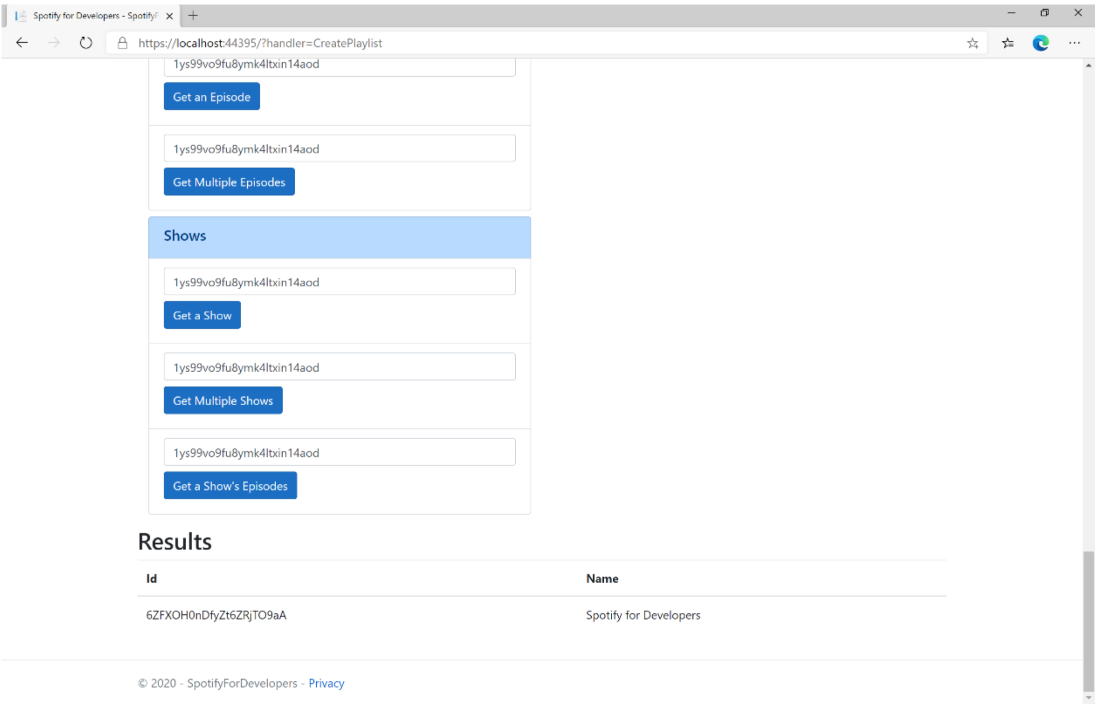
Once the **Web Application** is running and you select **Authorisation Code Flow Login** or **Implicit Grant Flow Login** and scroll down you should see something like the following:



https://localhost:44395/?code=AQBcFJQQenngi8pk8McxxUCR8rDkVDBJ9kBOL3iOhB1DNWGitXdf6zDOFMFWrhaZZD6WeZquEoRtdE2N2sMsvUCpKhBRQ2be...

Step 10

You can then enter above **Create a Playlist** your **User Id**, to get this in **Spotify** select your **Username** then **...** and **Copy Spotify URI** your **User Id** will be after **spotify:user:**, and a **Name** then select **Create a Playlist** and scroll down to view **Results** like the following and copy the **Id** for use later:



https://localhost:44395/?handler=CreatePlaylist

Spotify for Developers - Spotify
1ys99vo9fu8ymk4ltxin14aod
Get an Episode
1ys99vo9fu8ymk4ltxin14aod
Get Multiple Episodes
Shows
1ys99vo9fu8ymk4ltxin14aod
Get a Show
1ys99vo9fu8ymk4ltxin14aod
Get Multiple Shows
1ys99vo9fu8ymk4ltxin14aod
Get a Show's Episodes

Results

Id	Name
6ZFXOH0nDfyZt6ZRjTO9aA	Spotify for Developers

© 2020 - SpotifyForDevelopers - Privacy

Step 11

■

You can stop the **web application** in **Visual Studio 2019** by selecting the **Stop debugging** button

Step 12

✕

You can choose to exit **Visual Studio 2019** by selecting the **Close** button in the top right of the **application** as that completes this part of the workshop

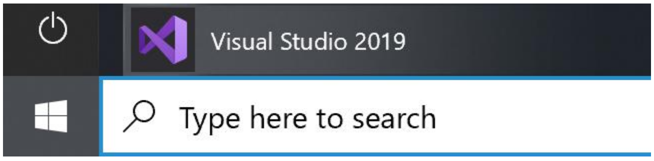
Get a List of Current User's Playlists

Get a list of the playlists owned or followed by the current Spotify user.

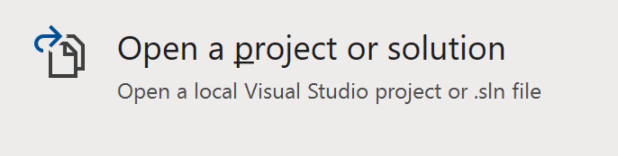
GET https://api.spotify.com/v1/me/playlists	
Header	
Authorization	User Token from Spotify Accounts service with **playlist-read-private** scope for playlists for the current user and **playlist-read-collaborative** for collaborative playlists
Query Parameter	
limit	Maximum number of playlists to return. Default: 20. Minimum: 1. Maximum: 50
offset	Index of the first playlist to return. Default: 0 (the first object). Maximum offset: 100,000. Use with limit to get the next set of playlists.

Header	Response
Success	
HTTP Status 200 OK	Array of Simplified Playlist Object wrapped in a Paging Object
Error	
Error Code	Error Object

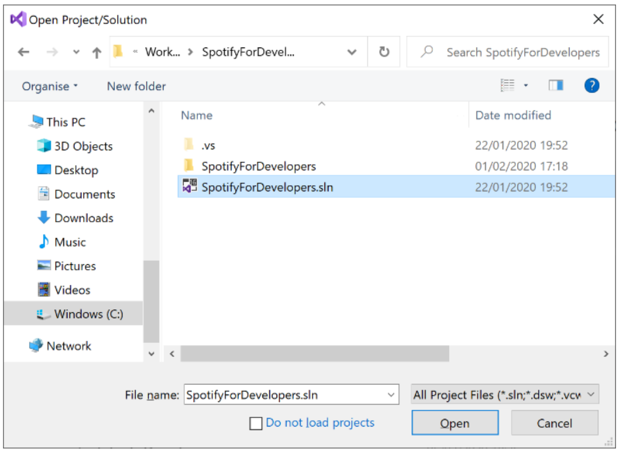
Step 1



If you chose to close **Visual Studio 2019** previously, in **Windows 10** choose **Start**, and then from the **Start Menu** find and select **Visual Studio 2019**


Open a project or solution
Open a local Visual Studio project or .sln file

Once done, from the **Get started** screen for **Visual Studio 2019** select **Open a project or solution**



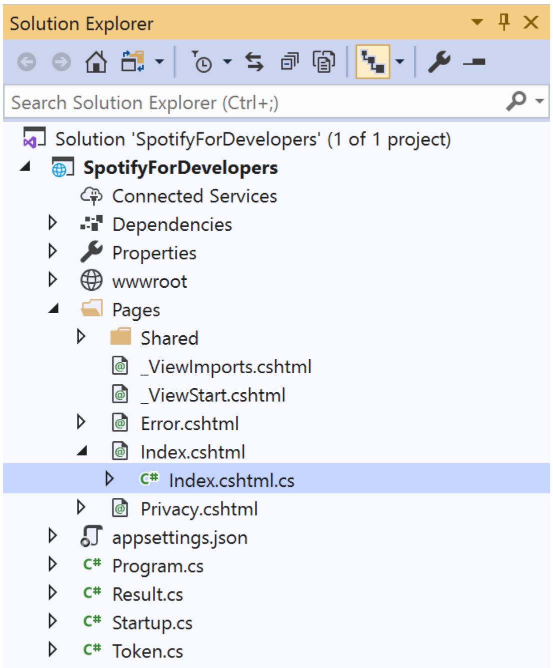
Name	Date modified
.vs	22/01/2020 19:52
SpotifyForDevelopers	01/02/2020 17:18
SpotifyForDevelopers.sln	22/01/2020 19:52

File name: SpotifyForDevelopers.sln

All Project Files (*.sln;*.dsw;*.vcw

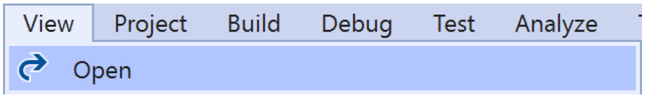
Then locate and select **SpotifyForDevelopers.sln** and select **Open** if you don't have this file already then please follow the previous parts of the workshop including **Getting Started**, **Authorisation Guide**, **Search & Browse**, **Playlists & Artists**, **Albums & Tracks**, **Episodes & Shows**, **Follow** and **Playlists**

Step 2



Once opened, in the **Solution Explorer** open the **Pages** section, then open the **Index.cshtml** section and select **Index.cshtml.cs**

Step 3



Then from the **Menu** choose **View** and then **Open**

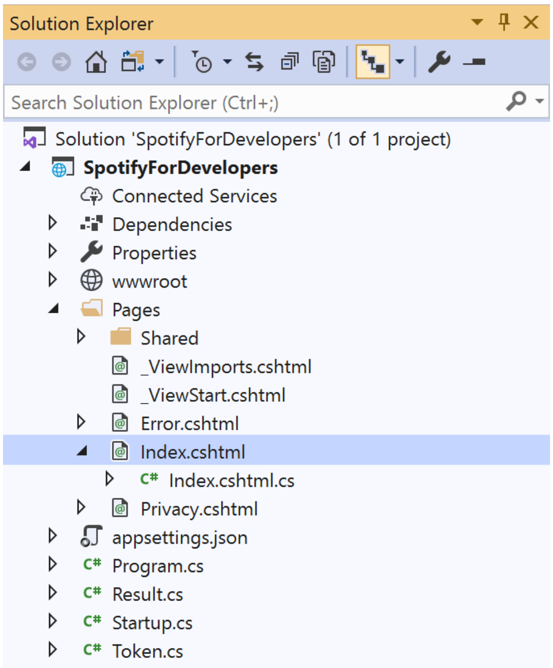
Step 4

In the **Code View** for **Index.cshtml.cs** below the **method** for `public async Task<IActionResult> OnPostCreatePlaylistAsync(...) { ... }` enter the following **method**:

```
public async Task<IActionResult> OnPostGetCurrentUserPlaylistsAsync()
{
    LoadToken();
    var results = await Api.GetUserPlaylistsAsync();
    if (results?.Items != null)
    {
        Results = results.Items.Select(result => new Result()
        {
            Id = result.Id,
            Name = result.Name,
            Image = result?.Images?.FirstOrDefault()?.Url
        });
    }
    return Page();
}
```

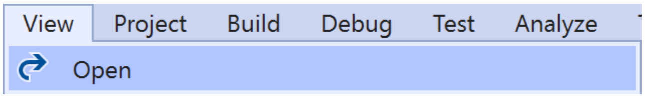
The **method** for `OnPostGetCurrentUserPlaylistsAsync` is used to list the **playlists** for the logged in **user** on Spotify and populate the **property** for `Results` accordingly.

Step 5



In the **Solution Explorer** in the **Pages** section select **Index.cshtml**

Step 6



Then from the **Menu** choose **View** and then **Open**

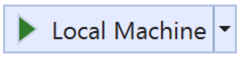
Step 7

Once in the **Code View** for **Index.cshtml** above `<!-- Playlists -->` enter the following:

```
<li class="list-group-item">
    <form asp-page-handler="GetCurrentUserPlaylists" method="post">
        <button class="btn btn-primary mb-2">
            Get a List of Current User's Playlists
        </button>
    </form>
</li>
```

This `form` will **post** to the **method** for `GetCurrentUserPlaylists` and will output to the **Results**.

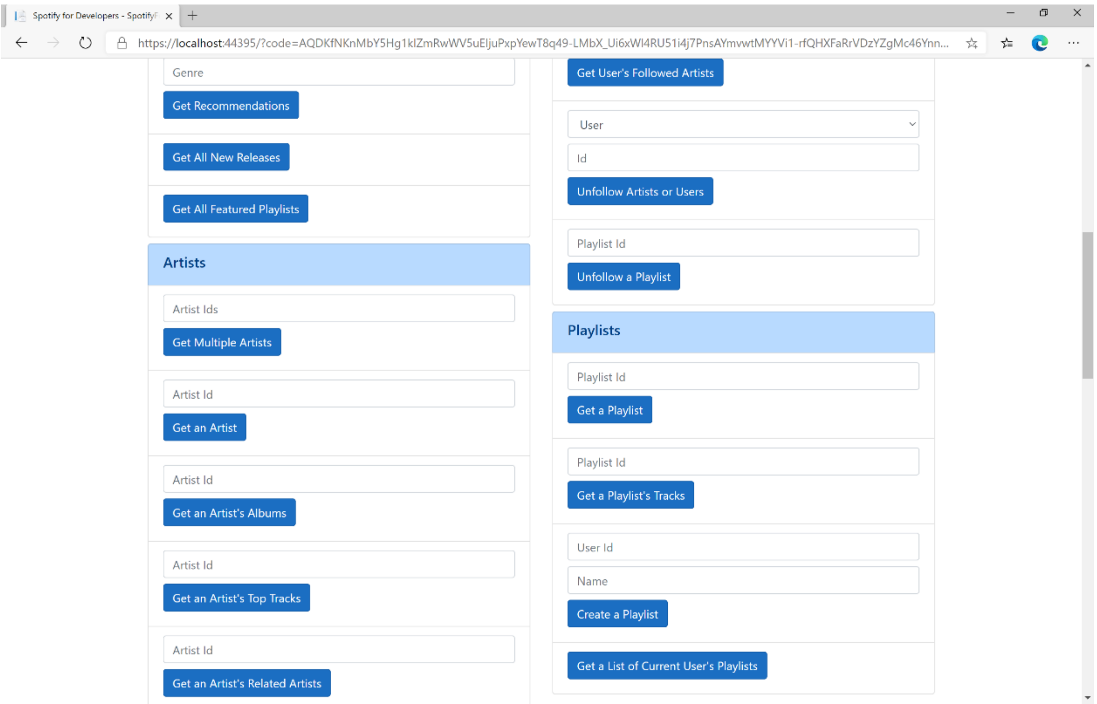
Step 8



Finally, in **Visual Studio 2019** select **IIS Application** to run the **Web Application**

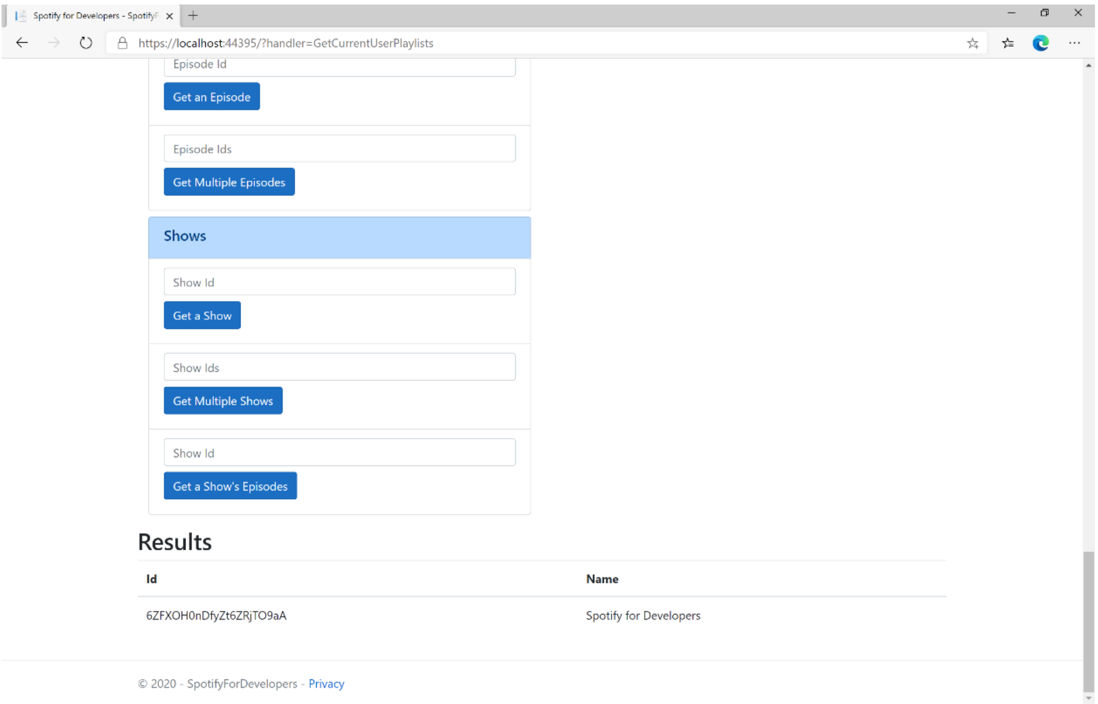
Step 9

Once the **Web Application** is running and you select **Authorisation Code Flow Login** or **Implicit Grant Flow Login** and scroll down you should see something like the following:


Spotify for Developers - Spotify
https://localhost:44395/?code=AQDKfNKnMbY5Hg1klZmRwWV5uEljuPxpYewT8q49-LMbX_Ui6xWI4RU51i4j7PnsAYmvwtMYYVi1-rfQHXFaRrVDzYZgMc46Ynn...
Genre
Get Recommendations
Get All New Releases
Get All Featured Playlists
Artists
Artist Ids
Get Multiple Artists
Artist Id
Get an Artist
Artist Id
Get an Artist's Albums
Artist Id
Get an Artist's Top Tracks
Artist Id
Get an Artist's Related Artists
Get User's Followed Artists
User
Id
Unfollow Artists or Users
Playlist Id
Unfollow a Playlist
Playlists
Playlist Id
Get a Playlist
Playlist Id
Get a Playlist's Tracks
User Id
Name
Create a Playlist
Get a List of Current User's Playlists

Step 10

You can then select **Get a List of Current User's Playlists** and scroll down to view **Results** like the following:



Results

Id	Name
6ZFXOH0nDfyZt6ZRjTO9aA	Spotify for Developers

© 2020 - SpotifyForDevelopers - Privacy

Step 11

You can stop the **web application** in **Visual Studio 2019** by selecting the **Stop debugging** button

Step 12

You can choose to exit **Visual Studio 2019** by selecting the **Close** button in the top right of the **application** as that completes this part of the workshop

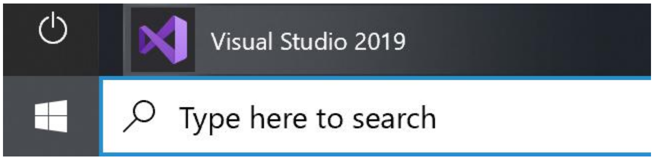
Get a List of a User's Playlists

Get a list of the playlists owned or followed by a Spotify user.

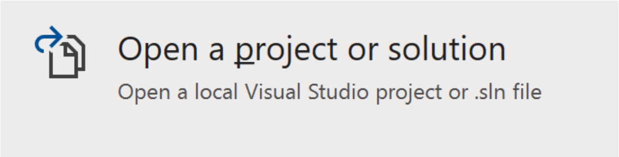
GET https://api.spotify.com/v1/users/{user_id}/playlists	
Header	
Authorization	User Token from Spotify Accounts service with **playlist-read-private** scope for playlists for the current user and **playlist-read-collaborative** for collaborative playlists
Path Parameter	
user_id	User's Spotify User Id
Query Parameter	
limit	Maximum number of playlists to return. Default: 20. Minimum: 1. Maximum: 50
offset	Index of the first playlist to return. Default: 0 (the first object). Maximum offset: 100,000. Use with limit to get the next set of playlists.

Header	Response
Success	
HTTP Status 200 OK	Array of Simplified Playlist Object wrapped in a Paging Object
Error	
Error Code	Error Object

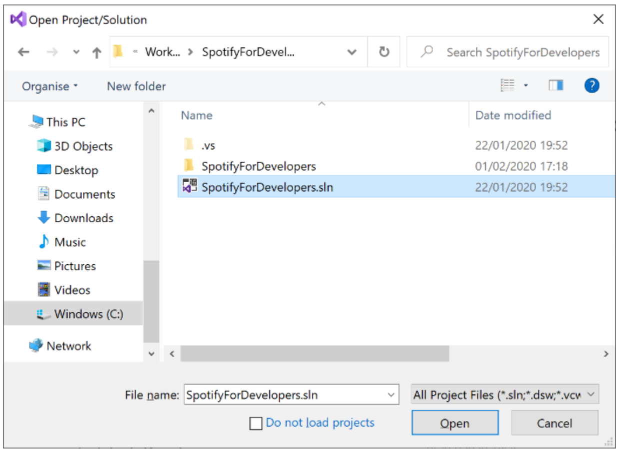
Step 1



If you chose to close **Visual Studio 2019** previously, in **Windows 10** choose **Start**, and then from the **Start Menu** find and select **Visual Studio 2019**


Open a project or solution
Open a local Visual Studio project or .sln file

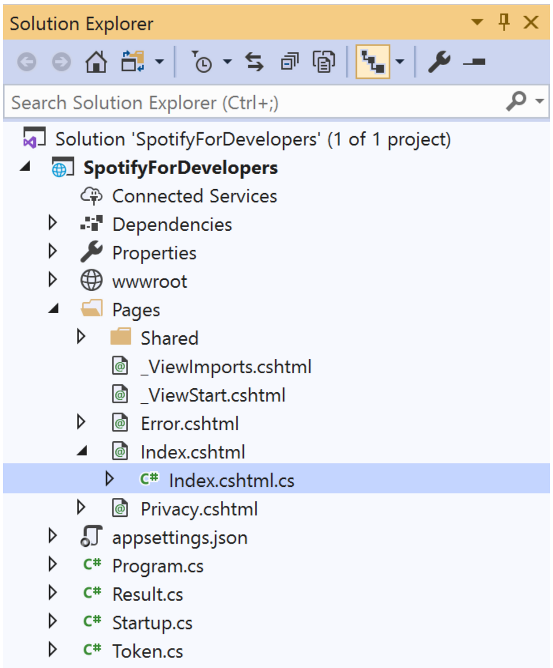
Once done, from the **Get started** screen for **Visual Studio 2019** select **Open a project or solution**



File name: SpotifyForDevelopers.sln

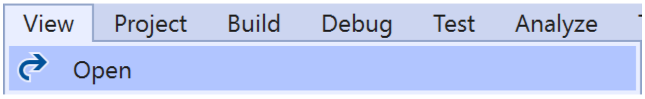
Then locate and select **SpotifyForDevelopers.sln** and select **Open** if you don't have this file already then please follow the previous parts of the workshop including **Getting Started**, **Authorisation Guide**, **Search & Browse**, **Playlists & Artists**, **Albums & Tracks**, **Episodes & Shows**, **Follow** and **Playlists**

Step 2



Once opened, in the **Solution Explorer** open the **Pages** section, then open the **Index.cshtml** section and select **Index.cshtml.cs**

Step 3



Then from the **Menu** choose **View** and then **Open**

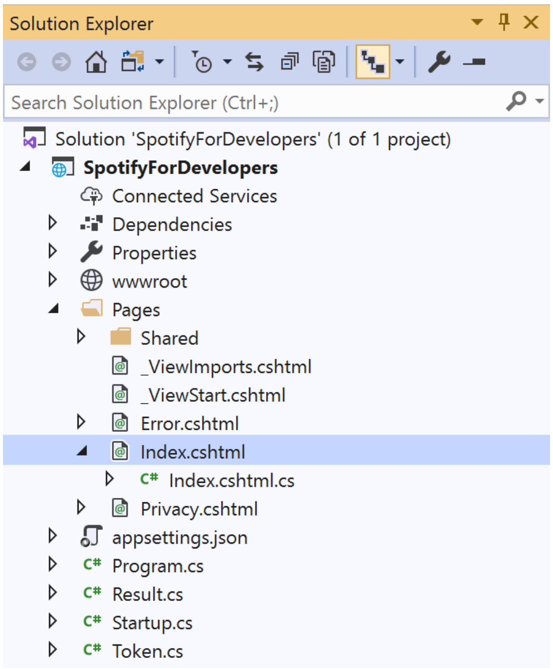
Step 4

In the **Code View** for **Index.cshtml.cs** below the **method** for `public async Task<IActionResult> OnPostGetCurrentUserPlaylistsAsync() { ... }` enter the following **method**:

```
public async Task<IActionResult> OnPostGetUserPlaylistsAsync(string value)
{
    LoadToken();
    var results = await Api.GetUserPlaylistsAsync(value);
    if (results?.Items != null)
    {
        Results = results.Items.Select(result => new Result()
        {
            Id = result.Id,
            Name = result.Name,
            Image = result?.Images?.FirstOrDefault()?.Url
        });
    }
    return Page();
}
```

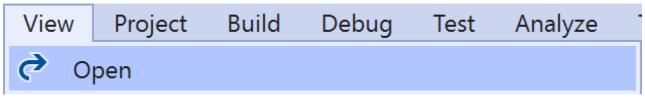
The **method** for `OnPostGetUserPlaylistsAsync` is used to list the **playlists** for a **user** by **User Id** on Spotify and populate the **property** for `Results` accordingly.

Step 5



In the **Solution Explorer** in the **Pages** section select **Index.cshtml**

Step 6



Then from the **Menu** choose **View** and then **Open**

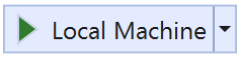
Step 7

Once in the **Code View** for **Index.cshtml** above `<!-- Playlists -->` enter the following:

```
<li class="list-group-item">
    <form asp-page-handler="GetUserPlaylists" method="post">
        <input asp-for="Value" placeholder="User Id" class="form-control mb-2" />
        <button class="btn btn-primary mb-2">
            Get a List of a User's Playlists
        </button>
    </form>
</li>
```

This `form` will **post** to the **method** for `GetUserPlaylists` with the `Value` of the **User Id** and will output to the **Results**.

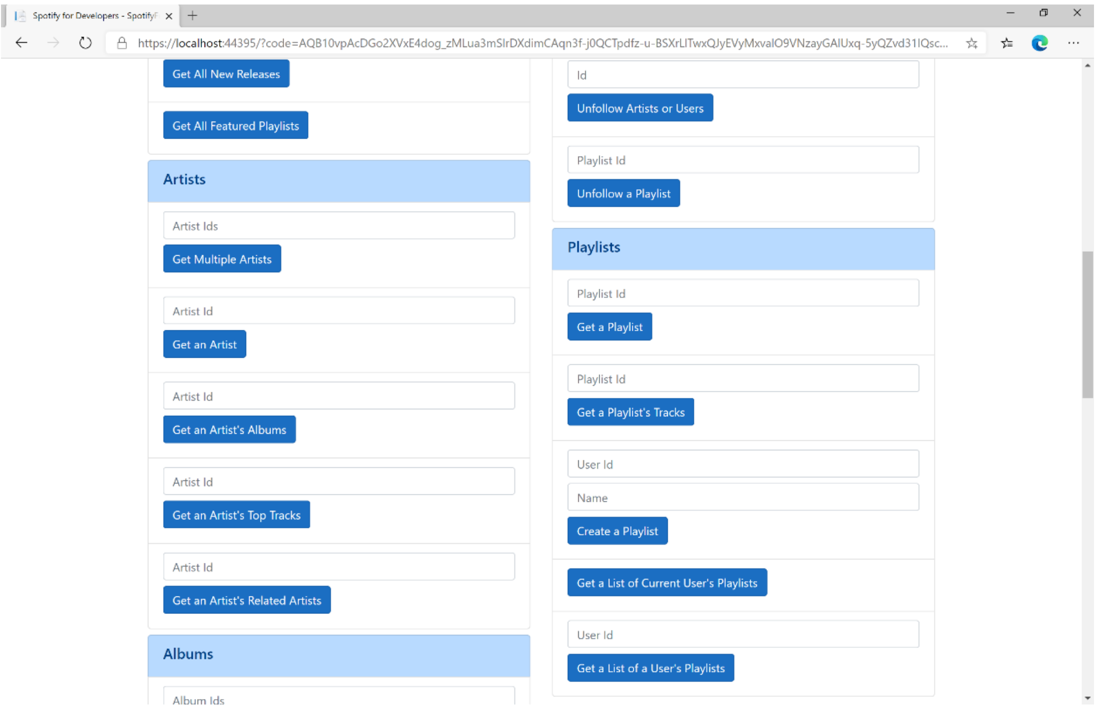
Step 8



Finally, in **Visual Studio 2019** select **IIS Application** to run the **Web Application**

Step 9

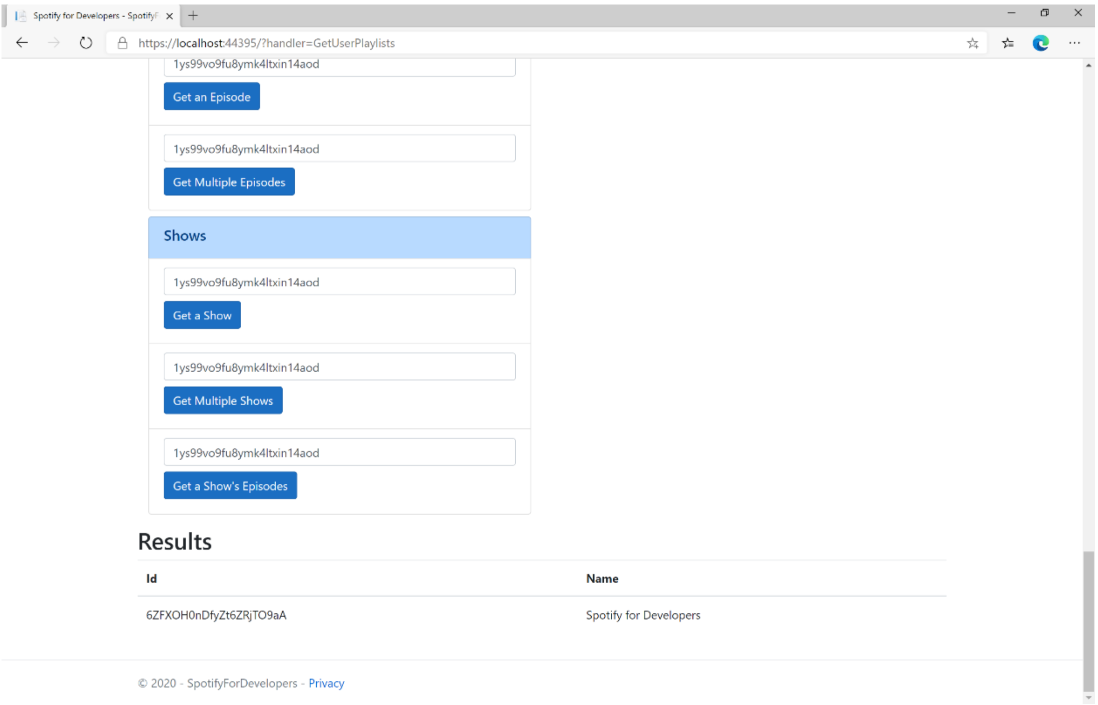
Once the **Web Application** is running and you select **Authorisation Code Flow Login** or **Implicit Grant Flow Login** and scroll down you should see something like the following:



Spotify for Developers - Spotify
https://localhost:44395/?code=AQB10vpAcDGo2XVxE4dog_zMLua3mSlrDXdimCAqn3f-j0QCTpdfz-u-BSXrLITwxQJyEVyMxvalO9VNzayGAlUxq-5yQZvd31lQsc...
Get All New Releases
Get All Featured Playlists
Artists
Artist Ids
Get Multiple Artists
Artist Id
Get an Artist
Artist Id
Get an Artist's Albums
Artist Id
Get an Artist's Top Tracks
Artist Id
Get an Artist's Related Artists
Albums
Album Ids
Id
Unfollow Artists or Users
Playlist Id
Unfollow a Playlist
Playlists
Playlist Id
Get a Playlist
Playlist Id
Get a Playlist's Tracks
User Id
Name
Create a Playlist
Get a List of Current User's Playlists
User Id
Get a List of a User's Playlists

Step 10

You can then above **Get a List of a User's Playlists** enter your own **User Id**, to get this in **Spotify** select your **Username** then **...** and **Copy Spotify URI** your **User Id** will be after **spotify:user**:, and then select **Get a List of a User's Playlists** and scroll down to view **Results** like the following:



Spotify for Developers - Spotify
https://localhost:44395/?handler=GetUserPlaylists
1ys99vo9fu8ymk4ltxin14aod
Get an Episode
1ys99vo9fu8ymk4ltxin14aod
Get Multiple Episodes
Shows
1ys99vo9fu8ymk4ltxin14aod
Get a Show
1ys99vo9fu8ymk4ltxin14aod
Get Multiple Shows
1ys99vo9fu8ymk4ltxin14aod
Get a Show's Episodes
© 2020 - SpotifyForDevelopers - Privacy

Results

Id	Name
6ZFXOH0nDfyZt6ZRjTO9aA	Spotify for Developers

Step 11

You can stop the **web application** in **Visual Studio 2019** by selecting the **Stop debugging** button

Step 12

You can choose to exit **Visual Studio 2019** by selecting the **Close** button in the top right of the **application** as that completes this part of the workshop

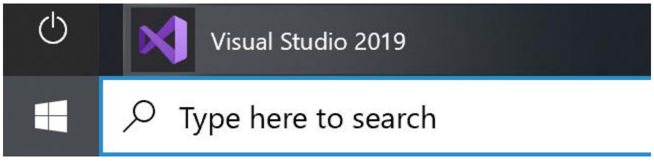
Add Tracks to a Playlist

Add one or more tracks to a user's playlist.

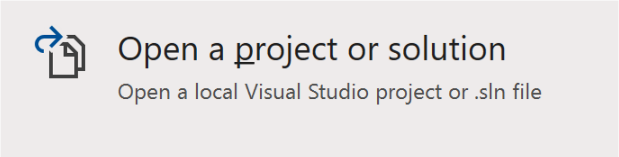
POST https://api.spotify.com/v1/playlists/{playlist_id}/tracks	
Header	
Authorization	User Token from Spotify Accounts service with **playlist-modify-public** scope for publicly followed playlist and **playlist-modify-private** for a privately followed playlist
Path Parameter	
playlist_id	Spotify Id for the Playlist
Query Parameter	
uri	Comma-separated list of Spotify Track URIs to add – maximum 100
position	Position to insert the tracks, a zero-based index.

Header	**Response**
Success	
HTTP Status 201 Created	snapshot_id
Error	
HTTP Status 403 Forbidden	Returned when not Authorised or more than 10,000 tracks in Playlist
Error Code	Error Object

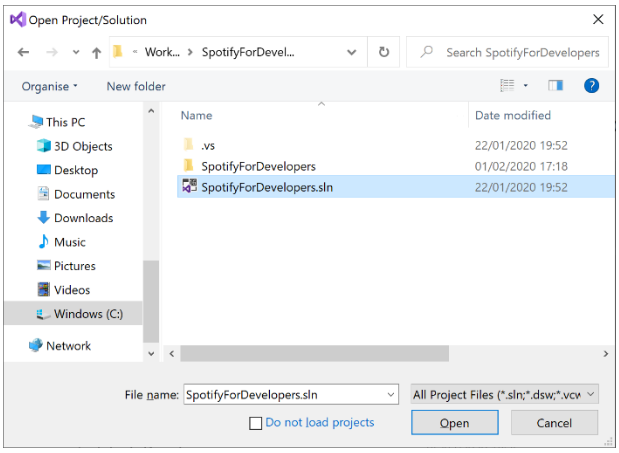
Step 1



If you chose to close **Visual Studio 2019** previously, in **Windows 10** choose **Start**, and then from the **Start Menu** find and select **Visual Studio 2019**


Open a project or solution
Open a local Visual Studio project or .sln file

Once done, from the **Get started** screen for **Visual Studio 2019** select **Open a project or solution**

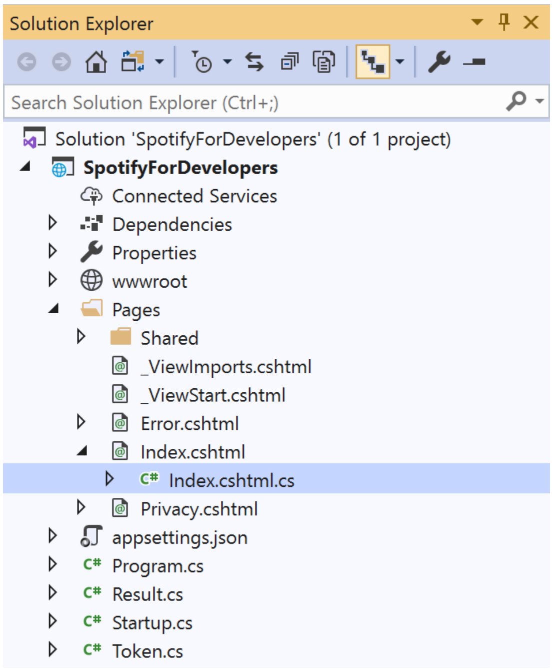


Name	Date modified
.vs	22/01/2020 19:52
SpotifyForDevelopers	01/02/2020 17:18
SpotifyForDevelopers.sln	22/01/2020 19:52

File name: SpotifyForDevelopers.sln

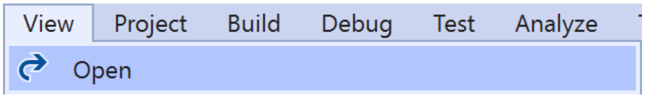
Then locate and select **SpotifyForDevelopers.sln** and select **Open** if you don't have this file already then please follow the previous parts of the workshop including **Getting Started**, **Authorisation Guide**, **Search & Browse**, **Playlists & Artists**, **Albums & Tracks**, **Episodes & Shows**, **Follow** and **Playlists**

Step 2



Once opened, in the **Solution Explorer** open the **Pages** section, then open the **Index.cshtml** section and select **Index.cshtml.cs**

Step 3



Then from the **Menu** choose **View** and then **Open**

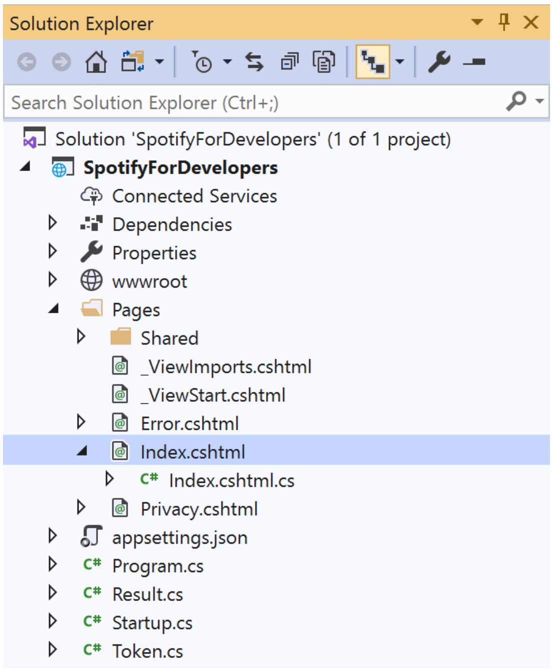
Step 4

In the **Code View** for **Index.cshtml.cs** below the **method** for `public async Task<IActionResult> OnPostGetUserPlaylistsAsync(...) { ... }` enter the following **method**:

```
public async Task<IActionResult> OnPostAddTracksToPlaylistAsync(string value, string option)
{
    LoadToken();
    var values = value.Split(",").Select(value => $"spotify:track:{value}").ToList();
    var result = await Api.AddTracksToPlaylistAsync(option, values);
    if (result != null)
    {
        Results = new List<Result>() { new Result()
        {
            Id = result.SnapshotId,
            Name = result.Success.ToString()
        }};
    }
    return Page();
}
```

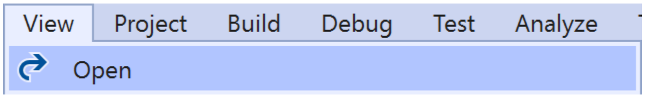
The **method** for `OnPostAddTracksToPlaylistAsync` is used add **tracks** by **Track Ids** to a **playlist** by **Playlist Id** on Spotify and populate the **property** for `Results` accordingly.

Step 5



In the **Solution Explorer** in the **Pages** section select **Index.cshtml**

Step 6



Then from the **Menu** choose **View** and then **Open**

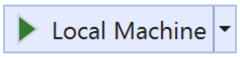
Step 7

Once in the **Code View** for **Index.cshtml** above `<!-- Playlists -->` enter the following:

```
<li class="list-group-item">
    <form asp-page-handler="AddTracksToPlaylist" method="post">
        <input asp-for="Value" placeholder="Track Ids" class="form-control mb-2" />
        <input asp-for="Option" placeholder="Playlist Id" class="form-control mb-2" />
        <button class="btn btn-primary mb-2">
            Add Tracks to a Playlist
        </button>
    </form>
</li>
```

This `form` will **post** to the **method** for `AddTracksToPlaylist` with the `Value` of the **Track Ids** and the `Option` of the **Playlist Id** and will output to the **Results**.

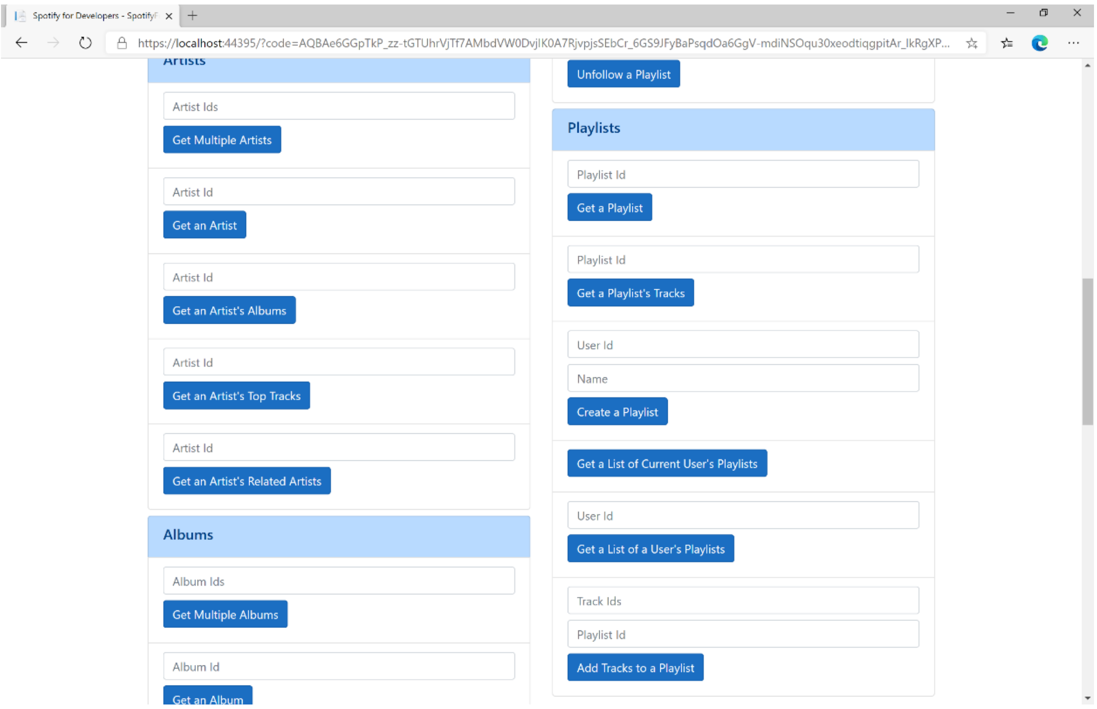
Step 8



Finally, in **Visual Studio 2019** select **IIS Application** to run the **Web Application**

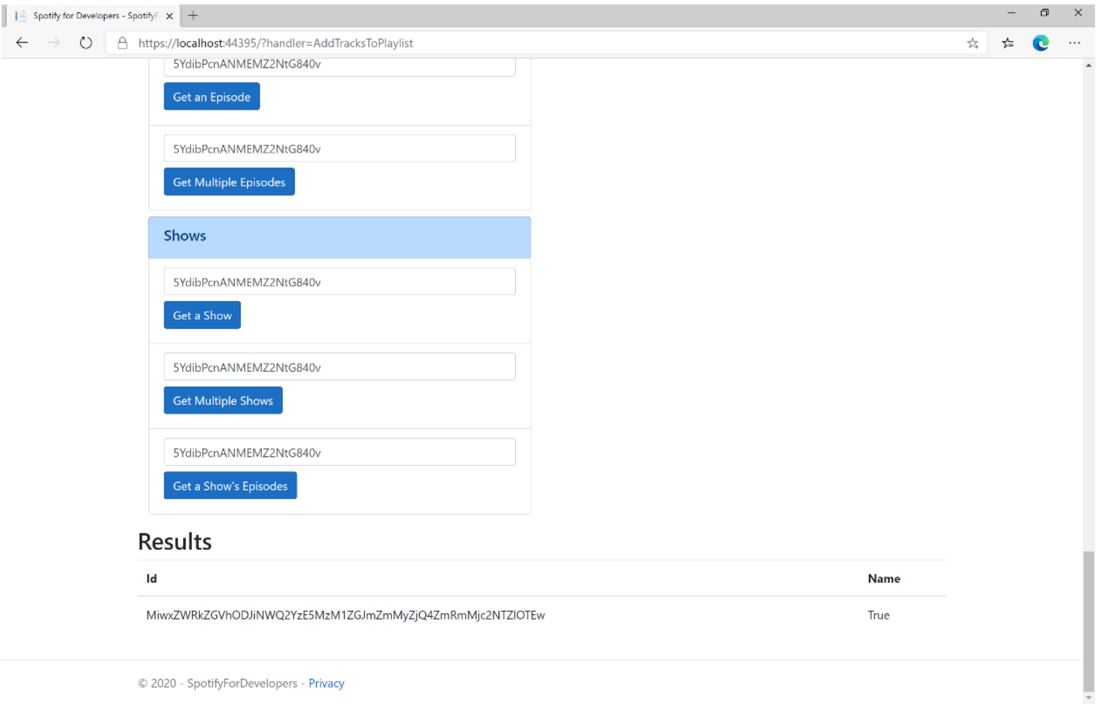
Step 9

Once the **Web Application** is running and you select **Authorisation Code Flow Login** or **Implicit Grant Flow Login** and scroll down you should see something like the following:



Step 10

You can then enter some **Track Ids** from an **Album Id** from **Get All New Releases** using **Get an Album's Tracks** into **Track Ids** above **Add Tracks to a Playlist** and the **Playlist Id** from **Create a Playlist** then select **Add Tracks to a Playlist** and scroll down to view **Results** like the following:



Results

Id	Name
MiwxZWRkZGVhODJiNWQ2YzE5MzM1ZGJmZmMyZjQ4ZmRmMjc2NTZlOTEw	True

© 2020 - SpotifyForDevelopers - Privacy

Step 11

You can stop the **web application** in **Visual Studio 2019** by selecting the **Stop debugging** button

Step 12

You can choose to exit **Visual Studio 2019** by selecting the **Close** button in the top right of the **application** as that completes this part of the workshop

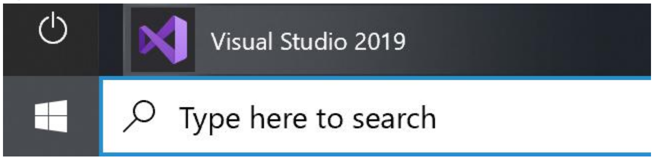
Remove Tracks from a Playlist

Remove one or more tracks from a user's playlist.

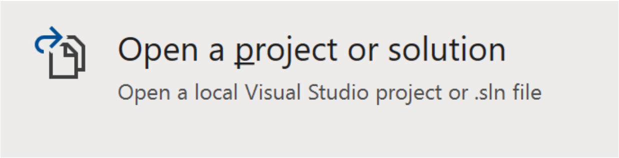
DELETE https://api.spotify.com/v1/playlists/{playlist_id}/tracks	
Header	
Authorization	User Token from Spotify Accounts service with **playlist-modify-public** scope for publicly followed playlist and **playlist-modify-private** for a privately followed playlist
Path Parameter	
playlist_id	Spotify Id for the Playlist
Query Parameter	
tracks	Comma-separated list of Spotify Track URIs to add – maximum 100
snapshot_id	Playlist's snapshot Id against which you want to make the changes

Header	**Response**
Success	
HTTP Status 200 OK	snapshot_id
Error	
HTTP Status 403 Forbidden	Returned when not Authorised
HTTP Status 400 Bad Request	Client Errors including specified invalid positions
Error Code	Error Object

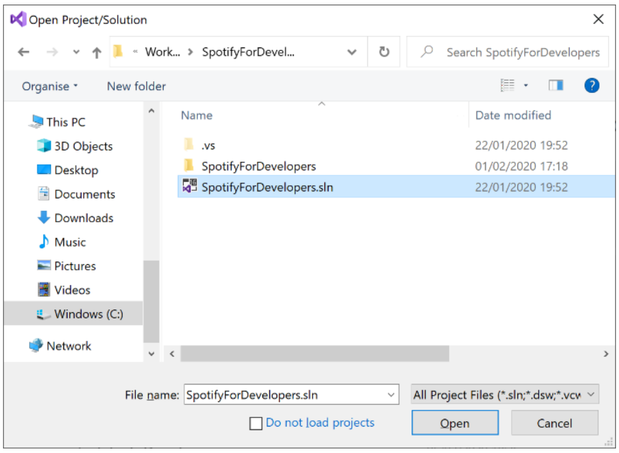
Step 1



If you chose to close **Visual Studio 2019** previously, in **Windows 10** choose **Start**, and then from the **Start Menu** find and select **Visual Studio 2019**


Open a project or solution
Open a local Visual Studio project or .sln file

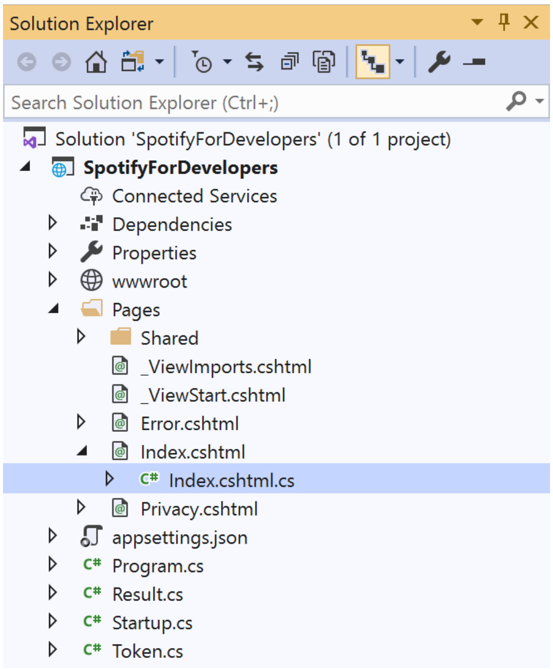
Once done, from the **Get started** screen for **Visual Studio 2019** select **Open a project or solution**



File name: SpotifyForDevelopers.sln

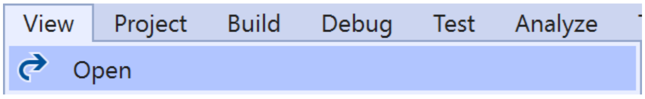
Then locate and select **SpotifyForDevelopers.sln** and select **Open** if you don't have this file already then please follow the previous parts of the workshop including **Getting Started**, **Authorisation Guide**, **Search & Browse**, **Playlists & Artists**, **Albums & Tracks**, **Episodes & Shows**, **Follow** and **Playlists**

Step 2



Once opened, in the **Solution Explorer** open the **Pages** section, then open the **Index.cshtml** section and select **Index.cshtml.cs**

Step 3



Then from the **Menu** choose **View** and then **Open**

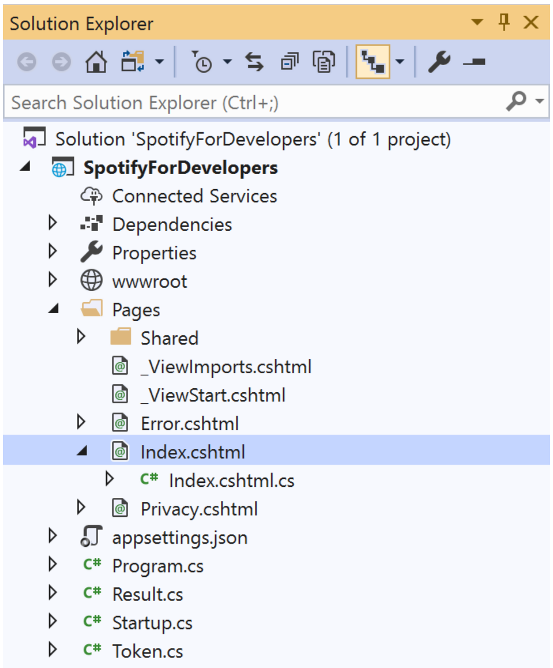
Step 4

In the **Code View** for **Index.cshtml.cs** below the **method** for `public async Task<IActionResult> OnPostAddTracksToPlaylistAsync(...) { ... }` enter the following **method**:

```
public async Task<IActionResult> OnPostRemoveTracksFromPlaylistAsync(string value, string
option)
{
    LoadToken();
    var values = value.Split(",").Select(value => $"spotify:track:{value}").ToList();
    var result = await Api.RemoveTracksFromPlaylistAsync(option, values);
    if (result != null)
    {
        Results = new List<Result>() { new Result()
        {
            Id = result.SnapshotId,
            Name = result.Success.ToString()
        }};
    }
    return Page();
}
```

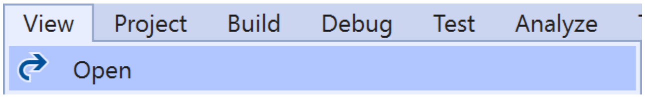
The **method** for `OnPostRemoveTracksFromPlaylistAsync` is used remove **tracks** by **Track Ids** from a **playlist** by **Playlist Id** on Spotify and populate the **property** for `Results` of the **success** of the operation accordingly.

Step 5



In the **Solution Explorer** in the **Pages** section select **Index.cshtml**

Step 6



Then from the **Menu** choose **View** and then **Open**

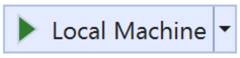
Step 7

Once in the **Code View** for **Index.cshtml** above `<!-- Playlists -->` enter the following:

```
<li class="list-group-item">
    <form asp-page-handler="RemoveTracksFromPlaylist" method="post">
        <input asp-for="Value" placeholder="Track Ids" class="form-control mb-2" />
        <input asp-for="Option" placeholder="Playlist Id" class="form-control mb-2" />
        <button class="btn btn-primary mb-2">
            Remove Tracks from a Playlist
        </button>
    </form>
</li>
```

This `form` will **post** to the **method** for `RemoveTracksFromPlaylist` with the `Value` of the **Track Ids** and the `Option` of the **Playlist Id** and will output to the **Results**.

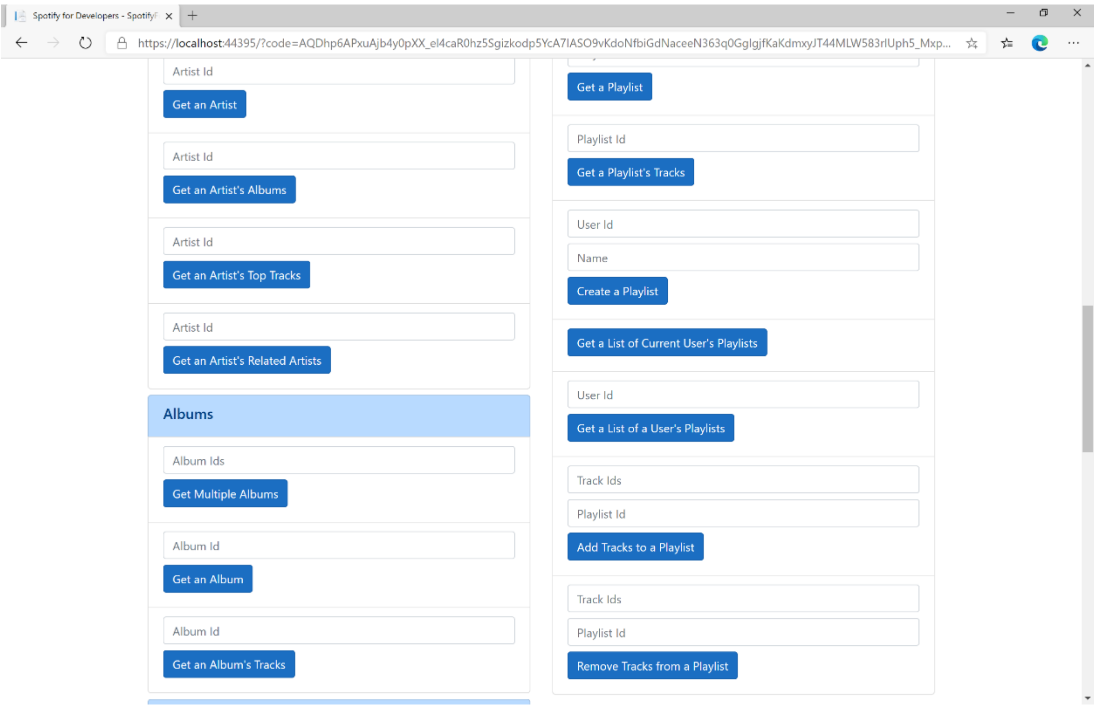
Step 8



Finally, in **Visual Studio 2019** select **IIS Application** to run the **Web Application**

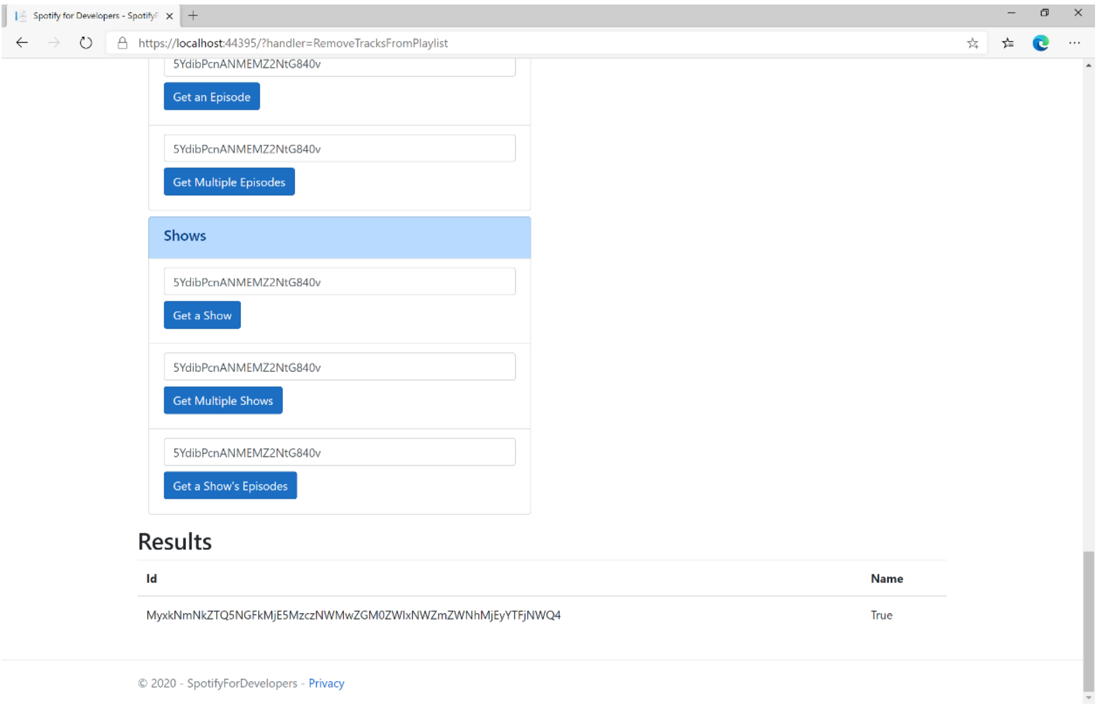
Step 9

Once the **Web Application** is running and you select **Authorisation Code Flow Login** or **Implicit Grant Flow Login** and scroll down you should see something like the following:


Spotify for Developers - Spotify
https://localhost:44395/?code=AQDhp6APxuAjb4y0pXX_el4caR0hz5Sgizkodp5YCA7IASO9vKdoNfbiGdNaceeN363q0GglgjfKaKdmxyJT44MLW583rIUph5_Mxp...
Artist Id
Get an Artist
Artist Id
Get an Artist's Albums
Artist Id
Get an Artist's Top Tracks
Artist Id
Get an Artist's Related Artists
Albums
Album Ids
Get Multiple Albums
Album Id
Get an Album
Album Id
Get an Album's Tracks
Get a Playlist
Playlist Id
Get a Playlist's Tracks
User Id
Name
Create a Playlist
Get a List of Current User's Playlists
User Id
Get a List of a User's Playlists
Track Ids
Playlist Id
Add Tracks to a Playlist
Track Ids
Playlist Id
Remove Tracks from a Playlist

Step 10

You can use **Get a Playlist's Tracks** and the **Playlist Id** from **Create a Playlist** to get the **Track Ids** to enter into **Track Ids** above **Remove Tracks from a Playlist** and the **Playlist Id** then select **Remove Tracks from a Playlist** and scroll down to view **Results** like the following:


Spotify for Developers - Spotify
https://localhost:44395/?handler=RemoveTracksFromPlaylist
Get an Episode
Get Multiple Episodes
Shows
Get a Show
Get Multiple Shows
Get a Show's Episodes

5YdibPcnANMEMZ2NtG840v

5YdibPcnANMEMZ2NtG840v

5YdibPcnANMEMZ2NtG840v

5YdibPcnANMEMZ2NtG840v

5YdibPcnANMEMZ2NtG840v

Results

Id	Name
MyxkNmNkZTQ5NGFkMjE5MzczNWMwZGM0ZWIxNWZmZWNhMjEyYTFjNWQ4	True

© 2020 - SpotifyForDevelopers - Privacy

Step 11

You can stop the **web application** in **Visual Studio 2019** by selecting the **Stop debugging** button

Step 12

You can choose to exit **Visual Studio 2019** by selecting the **Close** button in the top right of the **application** as that completes this part of the workshop

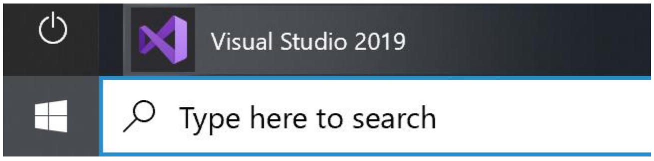
Upload a Custom Playlist Cover Image

Replace the image used to represent a specific playlist.

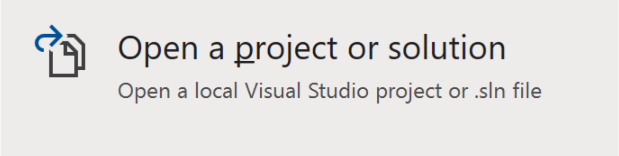
PUT https://api.spotify.com/v1/playlists/{playlist_id}/images	
Header	
Authorization	User Token from Spotify Accounts service with **ugc-image-upload**, **playlist-modify-public** scope for publicly followed playlist and **playlist-modify-private** for a privately followed playlist
Path Parameter	
playlist_id	Spotify Id for the Playlist
Body	
	Base 64 Encoded JPEG with a maximum size of 256KB

Header	Response
Success	
HTTP Status 200 OK	
Error	
HTTP Status 429 Too Many Requests	Returned when have sent too many requests
Error Code	Error Object

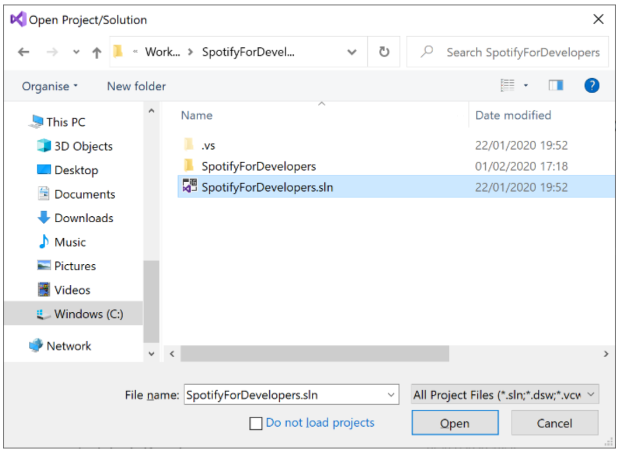
Step 1



If you chose to close **Visual Studio 2019** previously, in **Windows 10** choose **Start**, and then from the **Start Menu** find and select **Visual Studio 2019**


Open a project or solution
Open a local Visual Studio project or .sln file

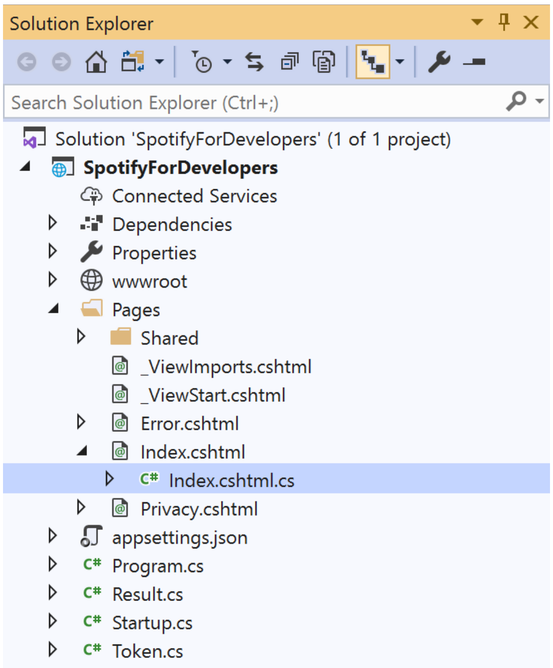
Once done, from the **Get started** screen for **Visual Studio 2019** select **Open a project or solution**



File name: SpotifyForDevelopers.sln

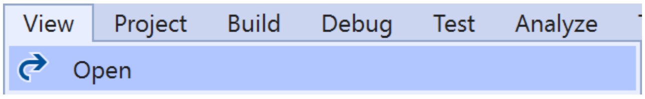
Then locate and select **SpotifyForDevelopers.sln** and select **Open** if you don't have this file already then please follow the previous parts of the workshop including **Getting Started**, **Authorisation Guide**, **Search & Browse**, **Playlists & Artists**, **Albums & Tracks**, **Episodes & Shows**, **Follow** and **Playlists**

Step 2



Once opened, in the **Solution Explorer** open the **Pages** section, then open the **Index.cshtml** section and select **Index.cshtml.cs**

Step 3



Then from the **Menu** choose **View** and then **Open**

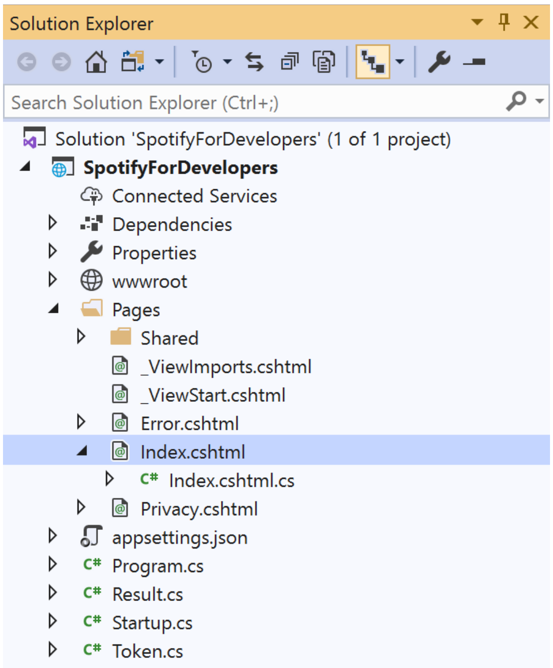
Step 4

In the **Code View** for **Index.cshtml.cs** below the **method** for `public async Task<IActionResult> OnPostRemoveTracksFromPlaylistAsync(...) { ... }` enter the following **method**:

```
public async Task<IActionResult> OnPostUploadCustomPlaylistCoverImageAsync(string value)
{
    LoadToken();
    System.IO.MemoryStream stream = new System.IO.MemoryStream();
    Upload.OpenReadStream().CopyTo(stream);
    var file = stream.ToArray();
    var result = await Api.UploadCustomPlaylistCoverImageAsync(value, file);
    if (result != null)
    {
        Results = new List<Result>() { new Result()
        {
            Id = result.Code.ToString(),
            Name = result.Success.ToString()
        }};
    }
    return Page();
}
```

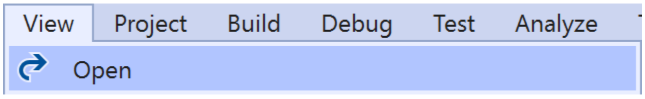
The **method** for `OnPostUploadCustomPlaylistCoverImageAsync` is **upload** an **image** with `Upload` for a **Playlist** by **Playlist Id** with `Value` on Spotify and populate the **property** for `Results` of the **success** of the operation accordingly.

Step 5



In the **Solution Explorer** in the **Pages** section select **Index.cshtml**

Step 6



Then from the **Menu** choose **View** and then **Open**

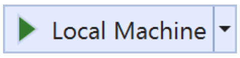
Step 7

Once in the **Code View** for **Index.cshtml** above `<!-- Playlists -->` enter the following:

```
<li class="list-group-item">
    <form asp-page-handler="UploadCustomPlaylistCoverImage"
        enctype="multipart/form-data" method="post">
        <input asp-for="Value" placeholder="Playlist Id" class="form-control mb-2" />
        <input asp-for="Upload" class="form-control mb-2" />
        <button class="btn btn-primary mb-2">
            Upload a Custom Playlist Cover Image
        </button>
    </form>
</li>
```

This `form` will **post** to the **method** for `UploadCustomPlaylistCoverImage` with the `Value` of the **Playlist Id** and the `Upload` of the **File** and will output to the **Results**.

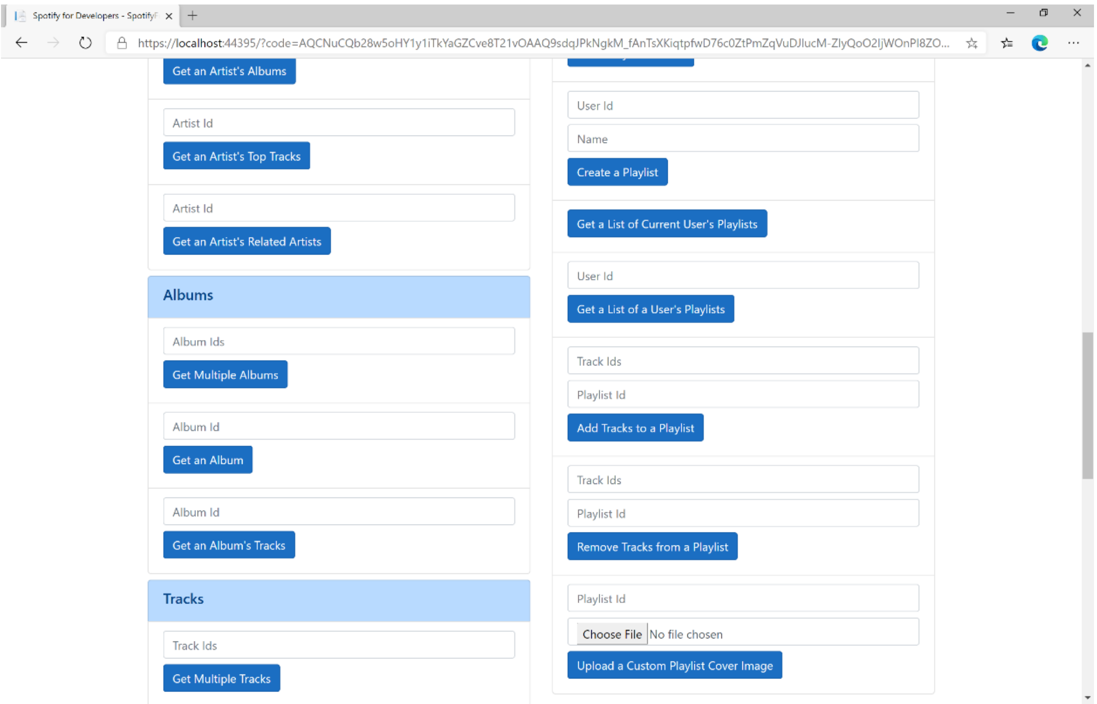
Step 8



Finally, in **Visual Studio 2019** select **IIS Application** to run the **Web Application**

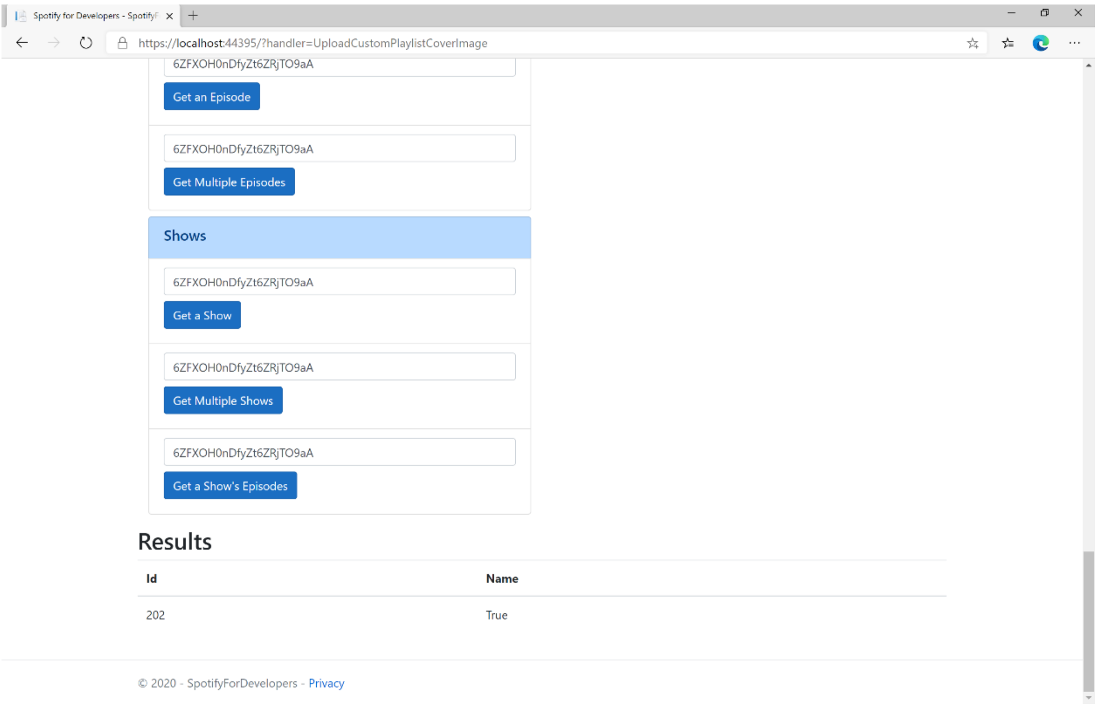
Step 9

Once the **Web Application** is running and you select **Authorisation Code Flow Login** or **Implicit Grant Flow Login** and scroll down you should see something like the following:


Spotify for Developers - SpotifyF
https://localhost:44395/?code=AQCNuCQb28w5oHY1y1iTkYaGZCve8T21vOAAQ9sdqJPkNgkM_fAnTsXKiqtpfwD76c0ZtPmZqVuDJlucM-ZlyQoO2lJWOnPl8ZO...
Get an Artist's Albums
Artist Id
Get an Artist's Top Tracks
Artist Id
Get an Artist's Related Artists
Albums
Album Ids
Get Multiple Albums
Album Id
Get an Album
Album Id
Get an Album's Tracks
Tracks
Track Ids
Get Multiple Tracks
User Id
Name
Create a Playlist
Get a List of Current User's Playlists
User Id
Get a List of a User's Playlists
Track Ids
Playlist Id
Add Tracks to a Playlist
Track Ids
Playlist Id
Remove Tracks from a Playlist
Playlist Id
Choose File No file chosen
Upload a Custom Playlist Cover Image

Step 10

You can then enter the **Playlist Id** from **Create a Playlist** into **Playlist Id** above **Upload a Custom Playlist Cover Image** and choose a **JPEG** file of **256KB** or smaller to upload and **Playlist Id** from **Create a Playlist** then select **Upload a Custom Playlist Cover Image** and scroll down to view **Results** like the following:



Results

Id	Name
202	True

© 2020 - SpotifyForDevelopers - Privacy

Step 11

You can stop the **web application** in **Visual Studio 2019** by selecting the **Stop debugging** button

Step 12

You can choose to exit **Visual Studio 2019** by selecting the **Close** button in the top right of the **application** as that completes this part of the workshop

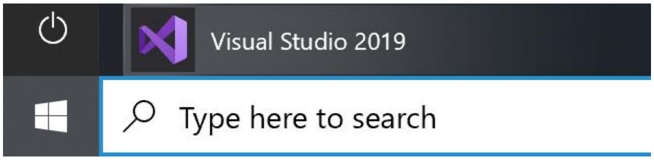
Get a Playlist Cover Image

Get the current image associated with a specific playlist.

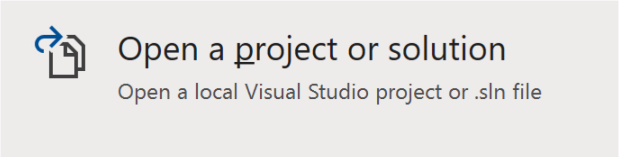
GET https://api.spotify.com/v1/playlists/{playlist_id}/images	
Header	
Authorization	User Token from Spotify Accounts service
Path Parameter	
playlist_id	Spotify Id for the Playlist

Header	Response
Success	
HTTP Status 200 OK	List of Image Objects
Error	
Error Code	Error Object

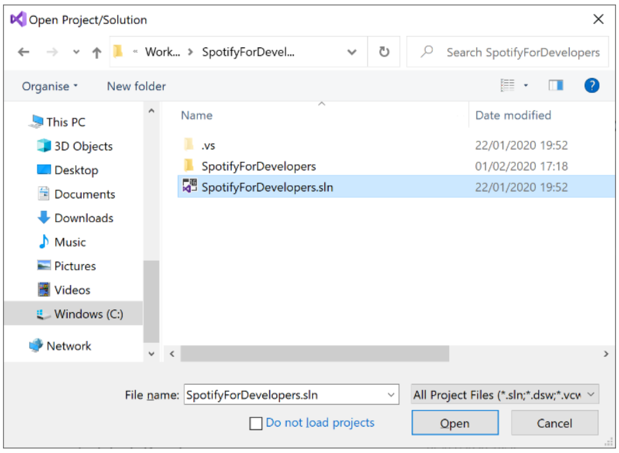
Step 1



If you chose to close **Visual Studio 2019** previously, in **Windows 10** choose **Start**, and then from the **Start Menu** find and select **Visual Studio 2019**


Open a project or solution
Open a local Visual Studio project or .sln file

Once done, from the **Get started** screen for **Visual Studio 2019** select **Open a project or solution**



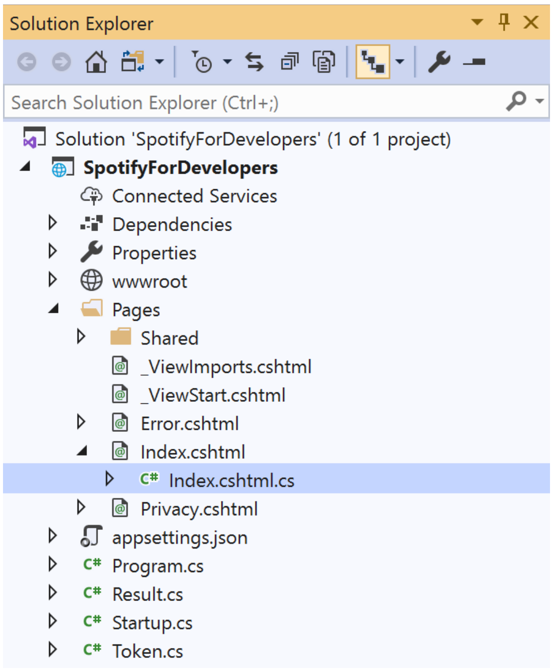
Name	Date modified
.vs	22/01/2020 19:52
SpotifyForDevelopers	01/02/2020 17:18
SpotifyForDevelopers.sln	22/01/2020 19:52

File name: SpotifyForDevelopers.sln

All Project Files (*.sln;*.dsw;*.vcw

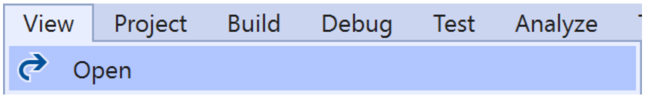
Then locate and select **SpotifyForDevelopers.sln** and select **Open** if you don't have this file already then please follow the previous parts of the workshop including **Getting Started**, **Authorisation Guide**, **Search & Browse**, **Playlists & Artists**, **Albums & Tracks**, **Episodes & Shows**, **Follow** and **Playlists**

Step 2



Once opened, in the **Solution Explorer** open the **Pages** section, then open the **Index.cshtml** section and select **Index.cshtml.cs**

Step 3



Then from the **Menu** choose **View** and then **Open**

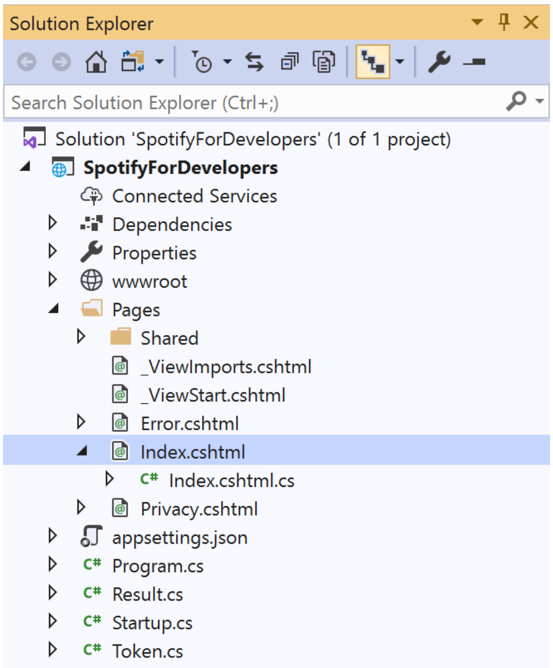
Step 4

In the **Code View** for **Index.cshtml.cs** below the **method** for `public async Task<IActionResult> OnPostUploadCustomPlaylistCoverImageAsync(...) { ... }` enter the following **method**:

```
public async Task<IActionResult> OnPostGetPlaylistCoverImageAsync(string value)
{
    LoadToken();
    var results = await Api.GetPlaylistCoverImageAsync(value);
    if (results != null)
    {
        Results = results.Select(result => new Result()
        {
            Image = result.Url
        });
    }
    return Page();
}
```

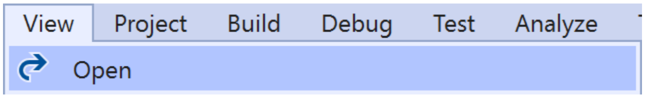
The **method** for `OnPostGetPlaylistCoverImageAsync` gets an **image** for a **Playlist** by **Playlist Id** with `Value` on Spotify and populate the **property** for `Results` accordingly.

Step 5



In the **Solution Explorer** in the **Pages** section select **Index.cshtml**

Step 6



Then from the **Menu** choose **View** and then **Open**

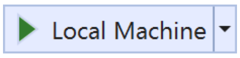
Step 7

Once in the **Code View** for **Index.cshtml** above `<!-- Playlists -->` enter the following:

```
<li class="list-group-item">
    <form asp-page-handler="GetPlaylistCoverImage" method="post">
        <input asp-for="Value" placeholder="Playlist Id" class="form-control mb-2" />
        <button class="btn btn-primary mb-2">
            Get a Playlist Cover Image
        </button>
    </form>
</li>
```

This `form` will **post** to the **method** for `GetPlaylistCoverImage` with the `Value` of the **Playlist Id** and the `Upload` of the **File** and will output to the **Results**.

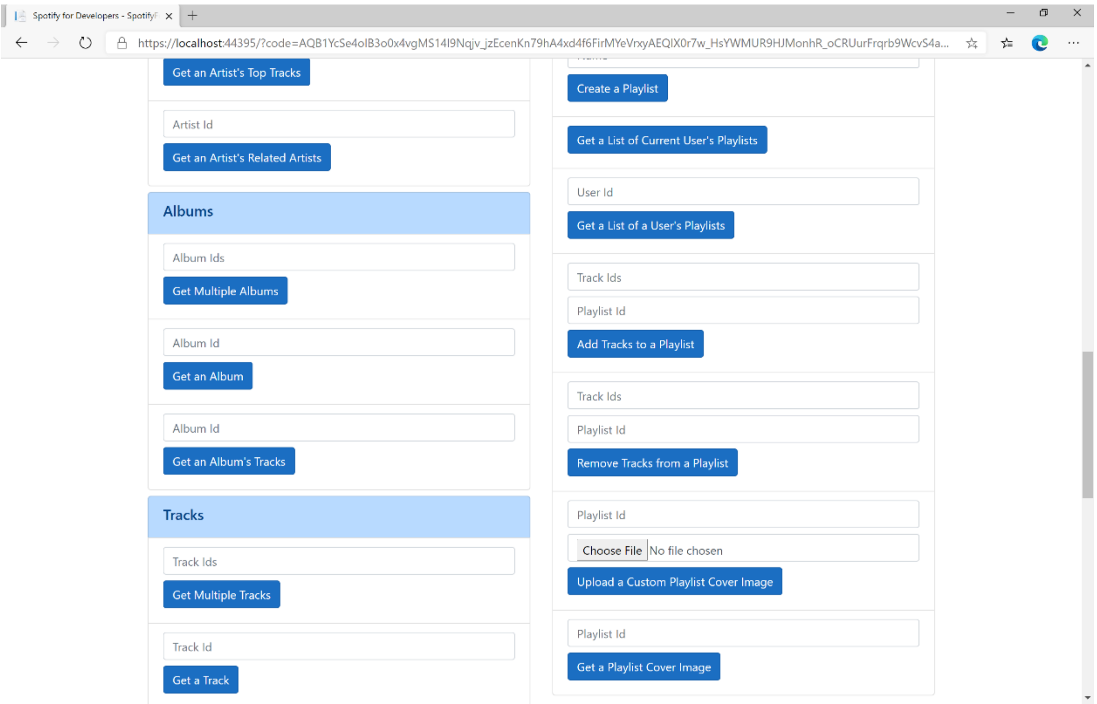
Step 8



Finally, in **Visual Studio 2019** select **IIS Application** to run the **Web Application**

Step 9

Once the **Web Application** is running and you select **Authorisation Code Flow Login** or **Implicit Grant Flow Login** and scroll down you should see something like the following:

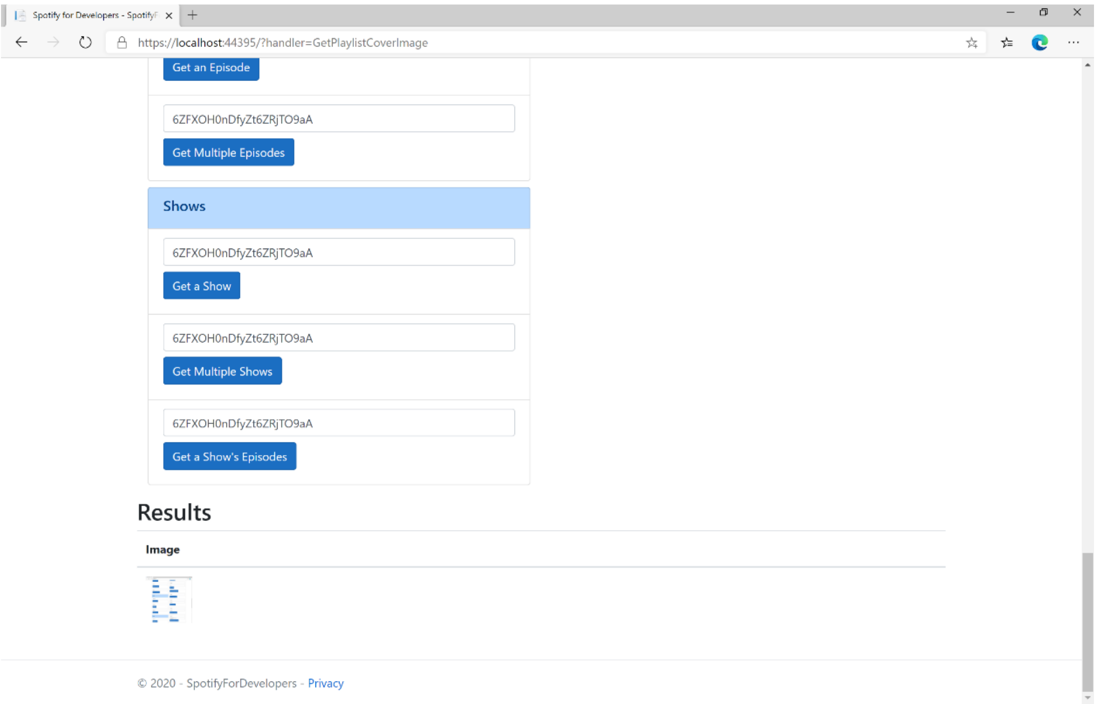


https://localhost:44395/?code=AQB1YcSe4olB3o0x4vgMS14I9Nqjv_jzEcenKn79hA4xd4f6FirMYeVrxyAEQIX0r7w_HsYWMUR9HJMonhR_oCRUurFrqrb9WcvS4a...

Spotify for Developers - SpotifyF
Get an Artist's Top Tracks
Artist Id
Get an Artist's Related Artists
Albums
Album Ids
Get Multiple Albums
Album Id
Get an Album
Album Id
Get an Album's Tracks
Tracks
Track Ids
Get Multiple Tracks
Track Id
Get a Track
Create a Playlist
Get a List of Current User's Playlists
User Id
Get a List of a User's Playlists
Track Ids
Playlist Id
Add Tracks to a Playlist
Track Ids
Playlist Id
Remove Tracks from a Playlist
Playlist Id
Choose File No file chosen
Upload a Custom Playlist Cover Image
Playlist Id
Get a Playlist Cover Image

Step 10

You can then enter the **Playlist Id** from **Create a Playlist** into **Playlist Id** above **Get a Playlist Cover Image** then select **Get a Playlist Cover Image** and scroll down to view **Results** like the following:



https://localhost:44395/?handler=GetPlaylistCoverImage

6ZFXOH0nDfyZt6ZRjTO9aA

6ZFXOH0nDfyZt6ZRjTO9aA

6ZFXOH0nDfyZt6ZRjTO9aA

6ZFXOH0nDfyZt6ZRjTO9aA

© 2020 - SpotifyForDevelopers - Privacy

Step 11

■

You can stop the **web application** in **Visual Studio 2019** by selecting the **Stop debugging** button

Step 12

×

You can choose to exit **Visual Studio 2019** by selecting the **Close** button in the top right of the **application** as that completes this part of the workshop

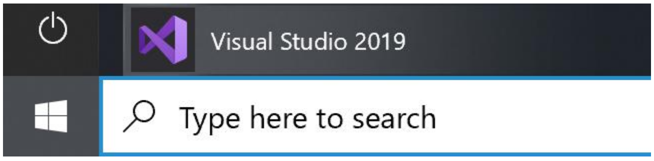
Change a Playlist's Details

Change a playlist’s name and public/private state of a playlist owned by a user

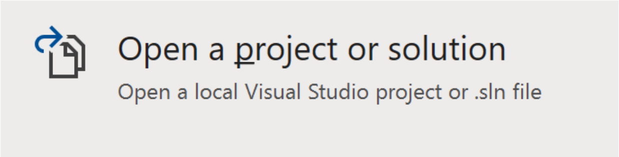
PUT https://api.spotify.com/v1/playlists/{playlist_id}	
Header	
Authorization	User Token from Spotify Accounts service with **playlist-modify-public** scope for publicly followed playlist and **playlist-modify-private** for a privately followed playlist
Path Parameter	
playlist_id	Spotify Id for the Playlist
Body Parameter	
name	New name for the playlist
public	If true, the playlist will be public, if false it will be private
collaborative	If true, the playlist will be collaborative and other users will be able to modify the playlist
description	Playlist description

Header	**Response**
Success	
HTTP Status 200 OK	
Error	
HTTP Status 403 Forbidden	Returned when not Authorised
Error Code	Error Object

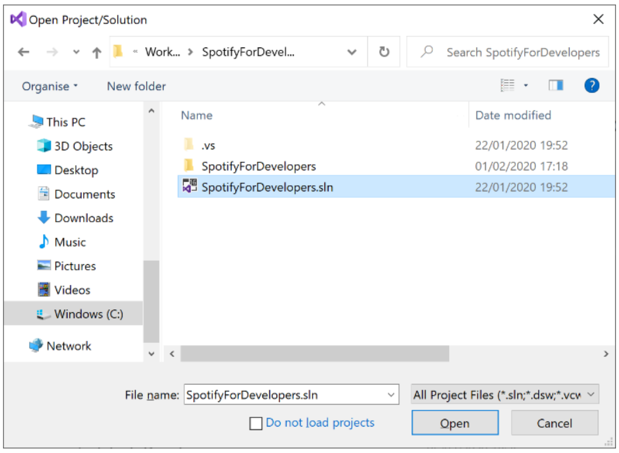
Step 1



If you chose to close **Visual Studio 2019** previously, in **Windows 10** choose **Start**, and then from the **Start Menu** find and select **Visual Studio 2019**

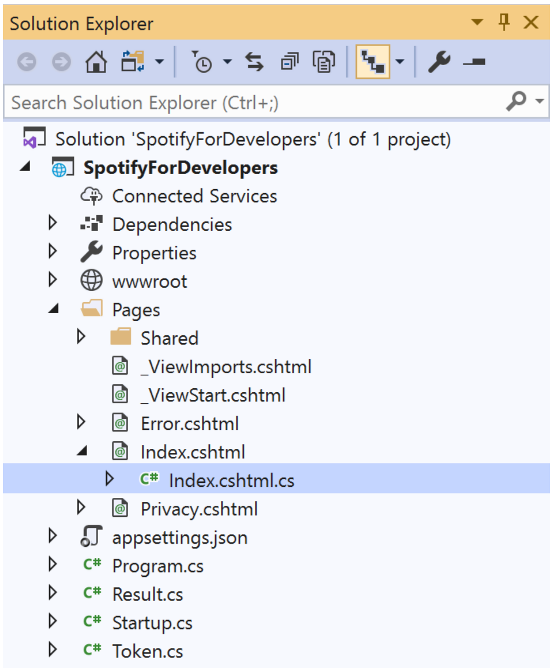

Open a project or solution
Open a local Visual Studio project or .sln file

Once done, from the **Get started** screen for **Visual Studio 2019** select **Open a project or solution**



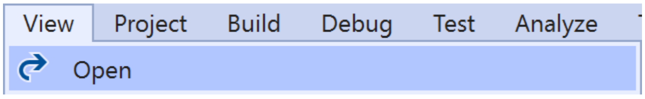
Then locate and select **SpotifyForDevelopers.sln** and select **Open** if you don’t have this file already then please follow the previous parts of the workshop including **Getting Started**, **Authorisation Guide**, **Search & Browse**, **Playlists & Artists**, **Albums & Tracks**, **Episodes & Shows**, **Follow** and **Playlists**

Step 2



Once opened, in the **Solution Explorer** open the **Pages** section, then open the **Index.cshtml** section and select **Index.cshtml.cs**

Step 3



Then from the **Menu** choose **View** and then **Open**

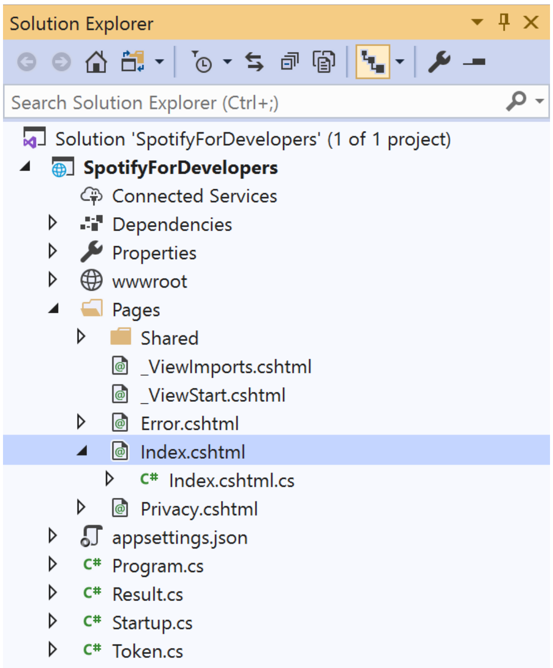
Step 4

In the **Code View** for **Index.cshtml.cs** below the **method** for `public async Task<IActionResult> OnPostGetPlaylistCoverImageAsync(...) { ... }` enter the following **method**:

```
public async Task<IActionResult> OnPostChangePlaylistDetailsAsync(string value, string
option)
{
    LoadToken();
    var result = await Api.ChangePlaylistDetailsAsync(value, option);
    if (result != null)
    {
        Results = new List<Result>() { new Result()
        {
            Id = result.Code.ToString(),
            Name = result.Success.ToString()
        }};
    }
    return Page();
}
```

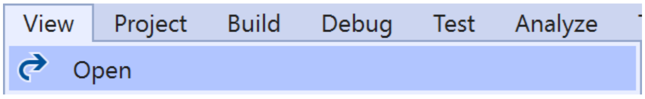
The **method** for `OnPostChangePlaylistDetailsAsync` sets the **Name** for a **Playlist** of `Option` by **Playlist Id** with `Value` on Spotify and populate the **property** for `Results` of the **success** of the operation accordingly.

Step 5



In the **Solution Explorer** in the **Pages** section select **Index.cshtml**

Step 6



Then from the **Menu** choose **View** and then **Open**

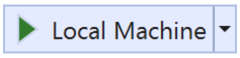
Step 7

Once in the **Code View** for **Index.cshtml** above `<!-- Playlists -->` enter the following:

```
<li class="list-group-item">
    <form asp-page-handler="ChangePlaylistDetails" method="post">
        <input asp-for="Value" placeholder="Playlist Id" class="form-control mb-2" />
        <input asp-for="Option" placeholder="Name" class="form-control mb-2" />
        <button class="btn btn-primary mb-2">
            Change a Playlist's Details
        </button>
    </form>
</li>
```

This `form` will **post** to the **method** for `ChangePlaylistDetails` with the `Value` of the **Playlist Id** and the `Option` of the **Name** and will output to the **Results**.

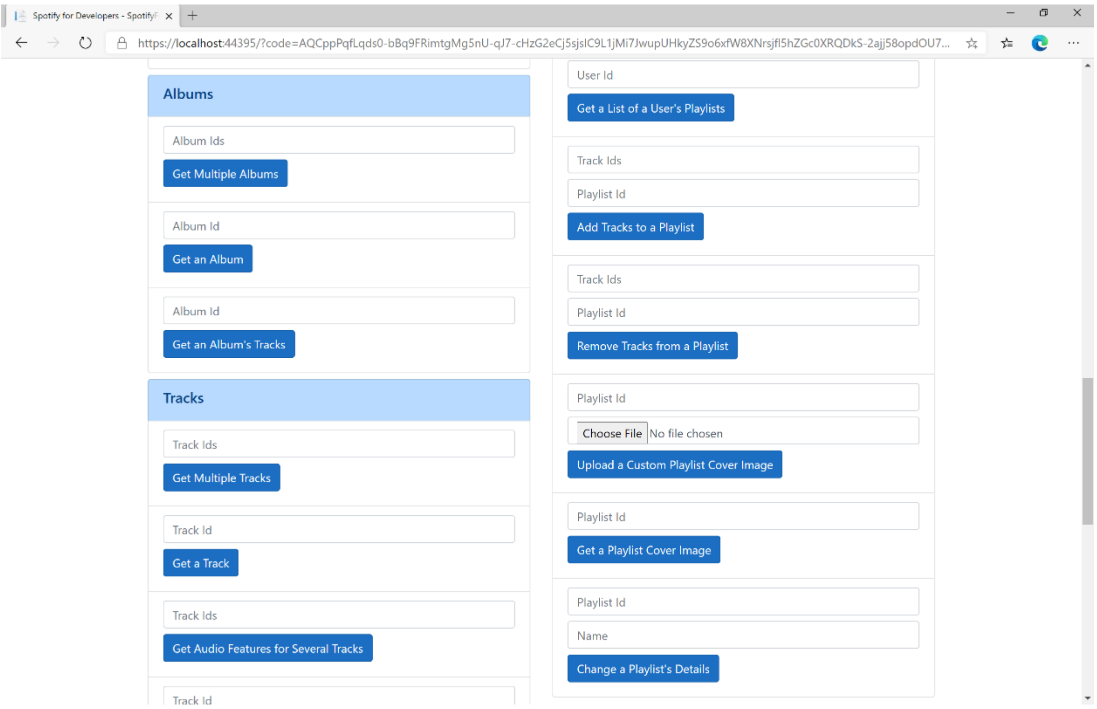
Step 8



Finally, in **Visual Studio 2019** select **IIS Application** to run the **Web Application**

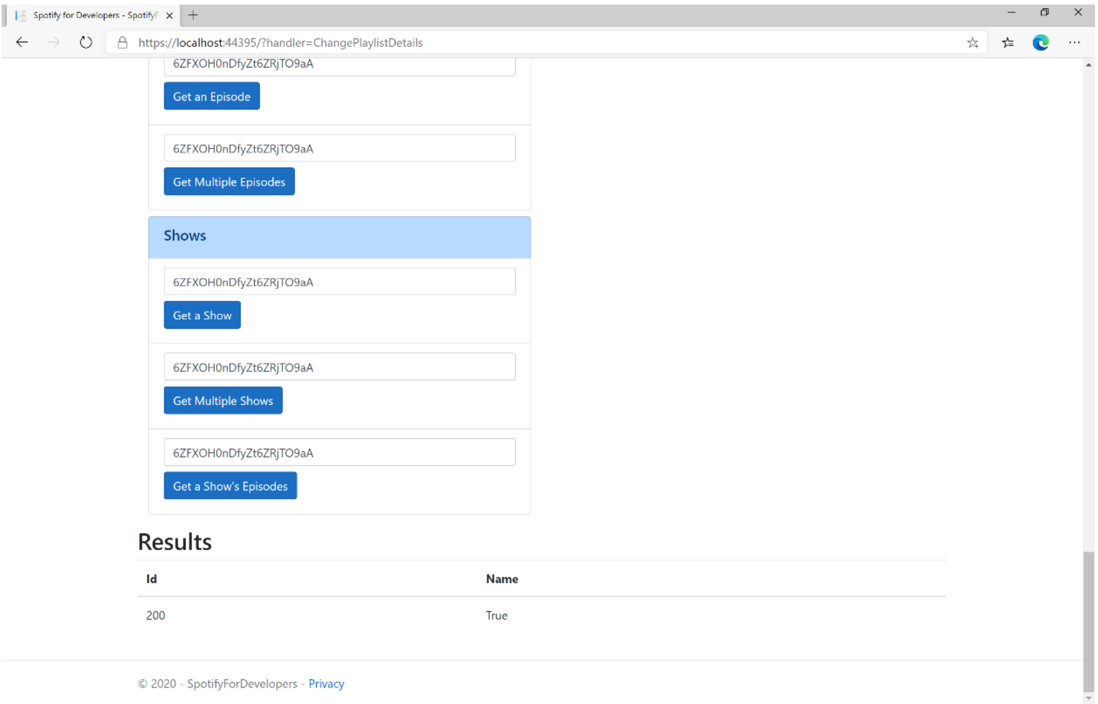
Step 9

Once the **Web Application** is running and you select **Authorisation Code Flow Login** or **Implicit Grant Flow Login** and scroll down you should see something like the following:


Spotify for Developers - Spotify
https://localhost:44395/?code=AQCppPqfLqds0-bBq9FRimtgMg5nU-qJ7-cHzG2eCj5sjsIC9L1jMi7JwupUHkyZS9o6xfW8XNrsjfl5hZGc0XRQDkS-2ajj58opdOU7...
Albums
Album Ids
Get Multiple Albums
Album Id
Get an Album
Album Id
Get an Album's Tracks
Tracks
Track Ids
Get Multiple Tracks
Track Id
Get a Track
Track Ids
Get Audio Features for Several Tracks
Track Id
User Id
Get a List of a User's Playlists
Track Ids
Playlist Id
Add Tracks to a Playlist
Track Ids
Playlist Id
Remove Tracks from a Playlist
Playlist Id
Choose File No file chosen
Upload a Custom Playlist Cover Image
Playlist Id
Get a Playlist Cover Image
Playlist Id
Name
Change a Playlist's Details

Step 10

You can then enter the **Playlist Id** from **Create a Playlist** into **Playlist Id** above **Change a Playlist's Details** and a new **Name** then select **Change a Playlist's Details** and scroll down to view **Results** like the following:



Results

Id	Name
200	True

© 2020 - SpotifyForDevelopers - Privacy

Step 11

■

You can stop the **web application** in **Visual Studio 2019** by selecting the **Stop debugging** button

Step 12

×

You can exit **Visual Studio 2019** by selecting the **Close** button in the top right of the **application** as that completes this part of the workshop

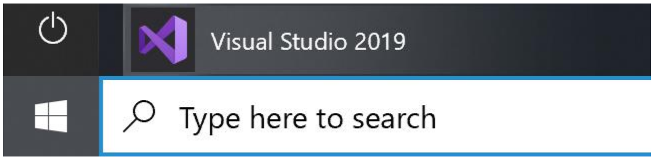
Replace a Playlist's Tracks

Replace all the tracks in a playlist, overwriting its existing tracks.

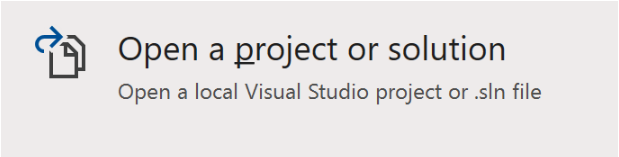
PUT https://api.spotify.com/v1/playlists/{playlist_id}/tracks	
Header	
Authorization	User Token from Spotify Accounts service with **playlist-modify-public** scope for publicly followed playlist and **playlist-modify-private** for a privately followed playlist
Path Parameter	
playlist_id	Spotify Id for the playlist
Query Parameter	
uris	Comma-separated list of Spotify track URIs to set – maximum of 100

Header	Response
Success	
Http Status 201 Created	
Error	
HTTP Status 403 Forbidden	Returned when not Authorised
Error Code	Error Object

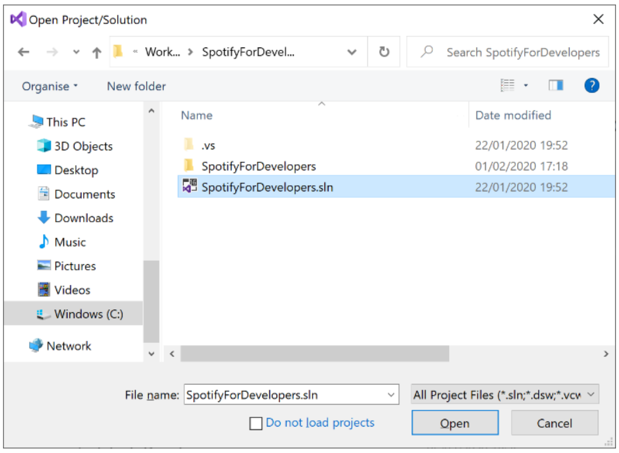
Step 1



If you chose to close **Visual Studio 2019** previously, in **Windows 10** choose **Start**, and then from the **Start Menu** find and select **Visual Studio 2019**


Open a project or solution
Open a local Visual Studio project or .sln file

Once done, from the **Get started** screen for **Visual Studio 2019** select **Open a project or solution**

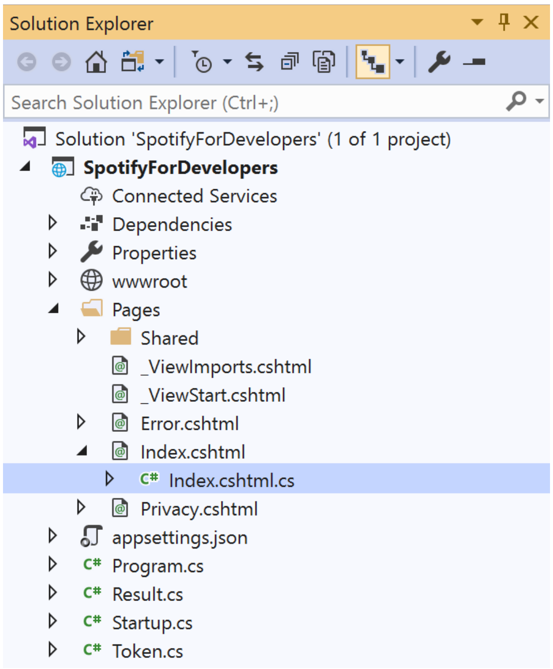


Name	Date modified
.vs	22/01/2020 19:52
SpotifyForDevelopers	01/02/2020 17:18
SpotifyForDevelopers.sln	22/01/2020 19:52

File name: SpotifyForDevelopers.sln

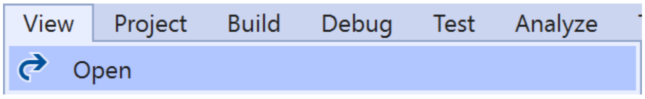
Then locate and select **SpotifyForDevelopers.sln** and select **Open** if you don't have this file already then please follow the previous parts of the workshop including **Getting Started**, **Authorisation Guide**, **Search & Browse**, **Playlists & Artists**, **Albums & Tracks**, **Episodes & Shows**, **Follow** and **Playlists**

Step 2



Once opened, in the **Solution Explorer** open the **Pages** section, then open the **Index.cshtml** section and select **Index.cshtml.cs**

Step 3



Then from the **Menu** choose **View** and then **Open**

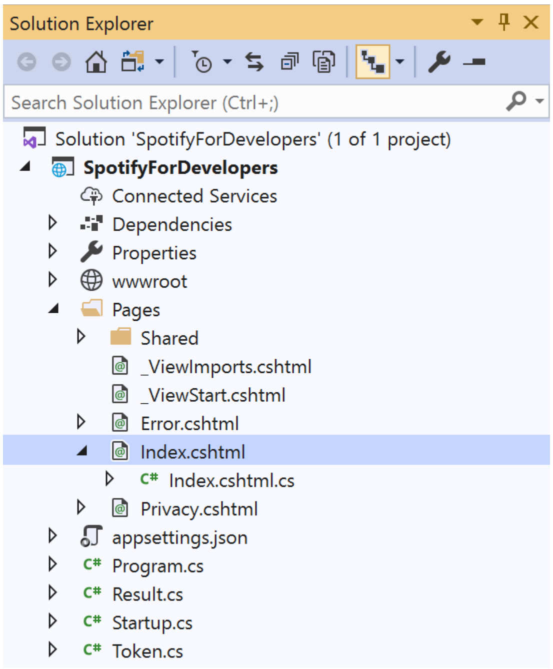
Step 4

In the **Code View** for **Index.cshtml.cs** below the **method** for `public async Task<IActionResult> OnPostChangePlaylistDetailsAsync(...) { ... }` enter the following **method**:

```
public async Task<IActionResult> OnPostReplacePlaylistTracksAsync(string value, string
option)
{
    LoadToken();
    var values = value.Split(",").Select(value => $"spotify:track:{value}").ToList();
    var result = await Api.ReplacePlaylistTracksAsync(option, values);
    if (result != null)
    {
        Results = new List<Result>() { new Result()
        {
            Id = result.Code.ToString(),
            Name = result.Success.ToString()
        }};
    }
    return Page();
}
```

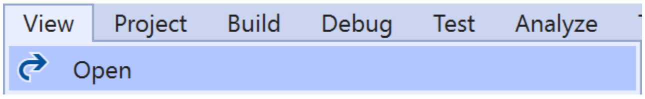
The **method** for `OnPostReplacePlaylistTracksAsync` replaces the **tracks** for a **Playlist** with `Value` of the **Track Ids** and the **Playlist Id** of `Option` on Spotify and populate the **property** for `Results` of the **success** of the operation accordingly.

Step 5



In the **Solution Explorer** in the **Pages** section select **Index.cshtml**

Step 6



Then from the **Menu** choose **View** and then **Open**

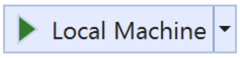
Step 7

Once in the **Code View** for **Index.cshtml** above `<!-- Playlists -->` enter the following:

```
<li class="list-group-item">
    <form asp-page-handler="ReplacePlaylistTracks" method="post">
        <input asp-for="Value" placeholder="Track Ids" class="form-control mb-2" />
        <input asp-for="Option" placeholder="Playlist Id" class="form-control mb-2" />
        <button class="btn btn-primary mb-2">
            Replace a Playlist's Tracks
        </button>
    </form>
</li>
```

This `form` will **post** to the **method** for `ReplacePlaylistTracks` with the `Value` of the **Track Ids** and the `Option` of the **Playlist Id** and will output to the **Results**.

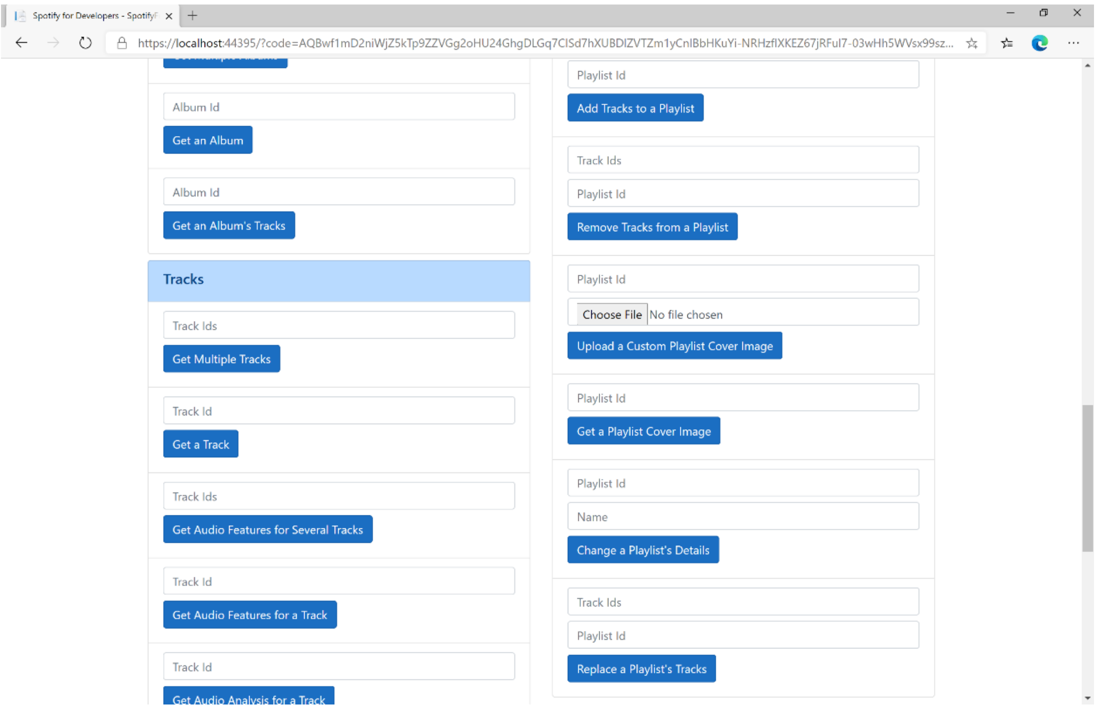
Step 8



Finally, in **Visual Studio 2019** select **IIS Application** to run the **Web Application**

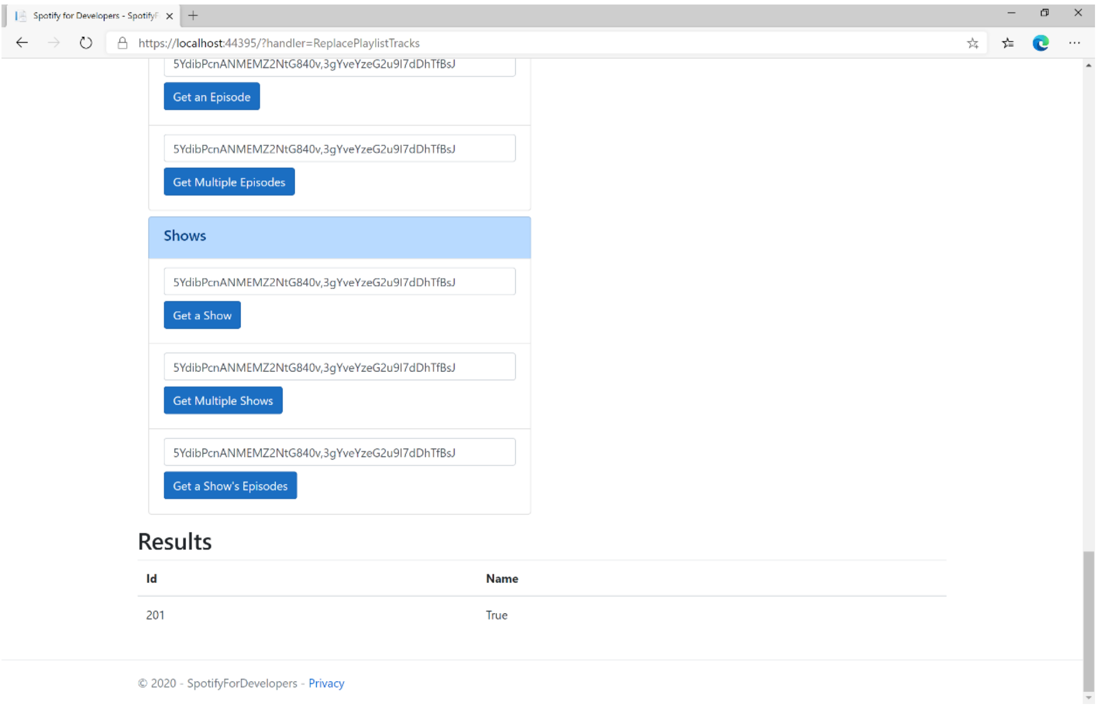
Step 9

Once the **Web Application** is running and you select **Authorisation Code Flow Login** or **Implicit Grant Flow Login** and scroll down you should see something like the following:


Spotify for Developers - Spotify
https://localhost:44395/?code=AQBwf1mD2niWjZ5kTp9ZZVGg2oHU24GhgDLGq7ClSd7hXUBDIZVTZm1yCnlBbHKuYi-NRHzflXKEZ67jRFul7-03wHh5WVsx99sz...
Album Id
Get an Album
Album Id
Get an Album's Tracks
Tracks
Track Ids
Get Multiple Tracks
Track Id
Get a Track
Track Ids
Get Audio Features for Several Tracks
Track Id
Get Audio Features for a Track
Track Id
Get Audio Analysis for a Track
Playlist Id
Add Tracks to a Playlist
Track Ids
Playlist Id
Remove Tracks from a Playlist
Playlist Id
Choose File No file chosen
Upload a Custom Playlist Cover Image
Playlist Id
Get a Playlist Cover Image
Playlist Id
Name
Change a Playlist's Details
Track Ids
Playlist Id
Replace a Playlist's Tracks

Step 10

You can then enter some **Track Ids** from an **Album Id** from **Get All New Releases** using **Get an Album's Tracks** into **Track Ids** above **Replace a Playlist's Tracks** and enter the **Playlist Id** from **Create a Playlist** into **Playlist Id** and then select **Replace a Playlist's Tracks** and scroll down to view **Results** like the following:


Spotify for Developers - Spotify
https://localhost:44395/?handler=ReplacePlaylistTracks
Get an Episode
Get Multiple Episodes
Shows
Get a Show
Get Multiple Shows
Get a Show's Episodes
© 2020 - SpotifyForDevelopers - Privacy

5YdibPcnANMEMZ2NtG840v,3gYveYzeG2u9I7dDhTfBsJ

5YdibPcnANMEMZ2NtG840v,3gYveYzeG2u9I7dDhTfBsJ

5YdibPcnANMEMZ2NtG840v,3gYveYzeG2u9I7dDhTfBsJ

5YdibPcnANMEMZ2NtG840v,3gYveYzeG2u9I7dDhTfBsJ

5YdibPcnANMEMZ2NtG840v,3gYveYzeG2u9I7dDhTfBsJ

Results

Id	Name
201	True

Step 11

You can stop the **web application** in **Visual Studio 2019** by selecting the **Stop debugging** button

Step 12

You can choose to exit **Visual Studio 2019** by selecting the **Close** button in the top right of the **application** as that completes this part of the workshop

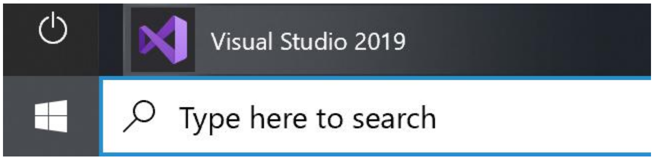
Reorder a Playlist's Tracks

Reorder a track or a group of tracks in a playlist.

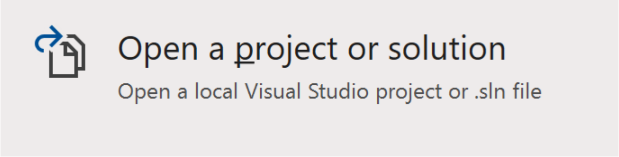
PUT https://api.spotify.com/v1/playlists/{playlist_id}/tracks	
Header	
Authorization	User Token from Spotify Accounts service with **playlist-modify-public** scope for publicly followed playlist and **playlist-modify-private** for a privately followed playlist
Path Parameter	
playlist_id	Spotify Id for the playlist
Body Parameter	
range_start	Position of the first track to be reordered
insert_before	Position where the tracks should be inserted
range_length	Number of tracks to be reordered. Defaults to 1 if not set.
snapshot_id	Playlist's snapshot ID against which you want to make the changes

Header	Response
Success	
Http Status 200 OK	
Error	
Error Code	Error Object

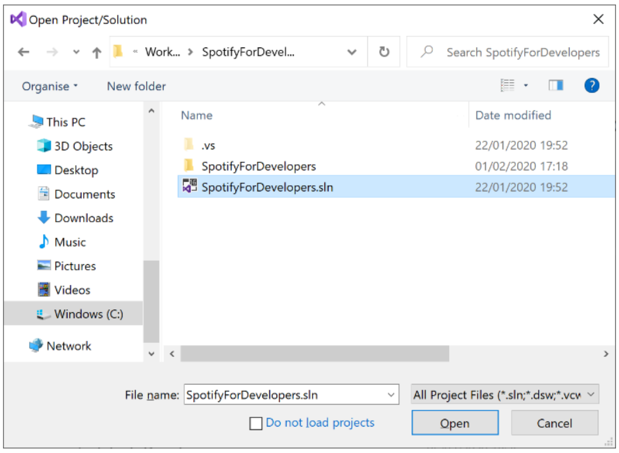
Step 1



If you chose to close **Visual Studio 2019** previously, in **Windows 10** choose **Start**, and then from the **Start Menu** find and select **Visual Studio 2019**


Open a project or solution
Open a local Visual Studio project or .sln file

Once done, from the **Get started** screen for **Visual Studio 2019** select **Open a project or solution**



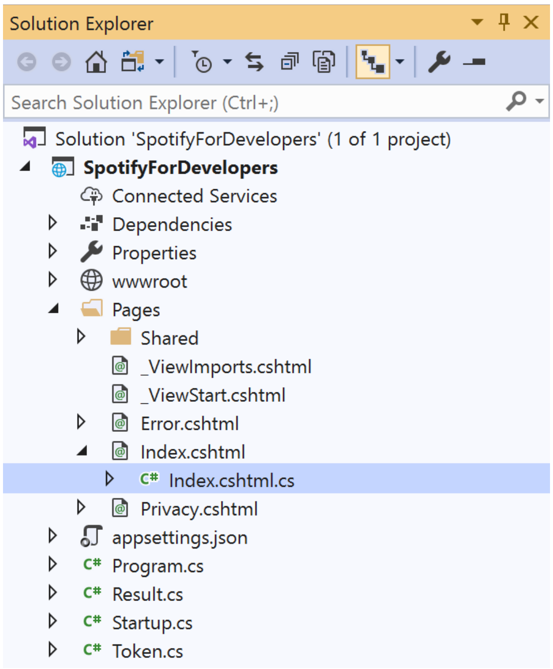
Name	Date modified
.vs	22/01/2020 19:52
SpotifyForDevelopers	01/02/2020 17:18
SpotifyForDevelopers.sln	22/01/2020 19:52

File name: SpotifyForDevelopers.sln

All Project Files (*.sln;*.dsw;*.vcw

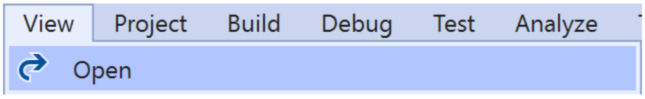
Then locate and select **SpotifyForDevelopers.sln** and select **Open** if you don't have this file already then please follow the previous parts of the workshop including **Getting Started**, **Authorisation Guide**, **Search & Browse**, **Playlists & Artists**, **Albums & Tracks**, **Episodes & Shows**, **Follow** and **Playlists**

Step 2



Once opened, in the **Solution Explorer** open the **Pages** section, then open the **Index.cshtml** section and select **Index.cshtml.cs**

Step 3



Then from the **Menu** choose **View** and then **Open**

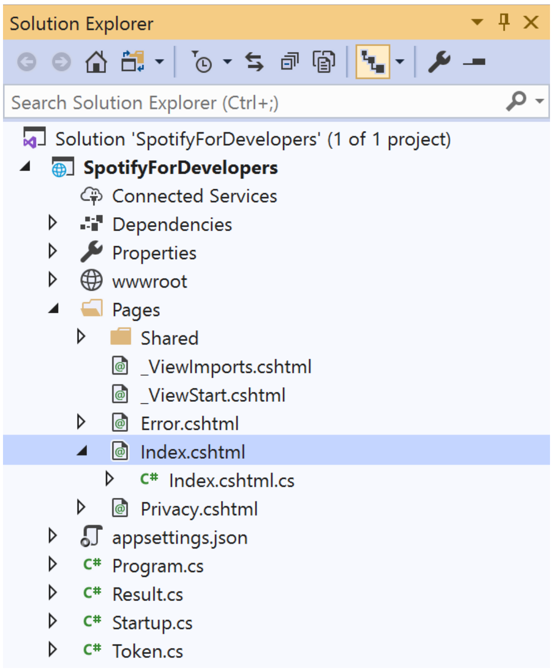
Step 4

In the **Code View** for **Index.cshtml.cs** below the **method** for `public async Task<IActionResult> OnPostReplacePlaylistTracksAsync(...) { ... }` enter the following **method**:

```
public async Task<IActionResult> OnPostReorderPlaylistTracksAsync(string value, string
option)
{
    LoadToken();
    var index = int.Parse(option);
    var result = await Api.ReorderPlaylistTracksAsync(value, index, 0, 1);
    if (result != null)
    {
        Results = new List<Result>() { new Result()
        {
            Id = result.SnapshotId.ToString(),
            Name = result.Success.ToString()
        }};
    }
    return Page();
}
```

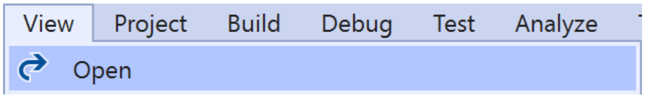
The **method** for `OnPostReorderPlaylistTracksAsync` **reorders** the **tracks** in a **Playlist** with the `Value` of the **Playlist Id** on Spotify and the `Option` is the **Index** of the **track** to move to the first position and populate the **property** for `Results` of the **success** of the operation accordingly.

Step 5



In the **Solution Explorer** in the **Pages** section select **Index.cshtml**

Step 6



Then from the **Menu** choose **View** and then **Open**

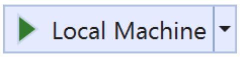
Step 7

Once in the **Code View** for **Index.cshtml** above `<!-- Playlists -->` enter the following:

```
<li class="list-group-item">
    <form asp-page-handler="ReorderPlaylistTracks" method="post">
        <input asp-for="Value" placeholder="Playlist Id" class="form-control mb-2" />
        <input asp-for="Option" placeholder="Index" class="form-control mb-2" />
        <button class="btn btn-primary mb-2">
            Reorder a Playlist's Tracks
        </button>
    </form>
</li>
```

This `form` will **post** to the **method** for `ReorderPlaylistTracks` with the `Value` of the **Playlist Id** and the `Option` of **Index** and will output to the **Results**.

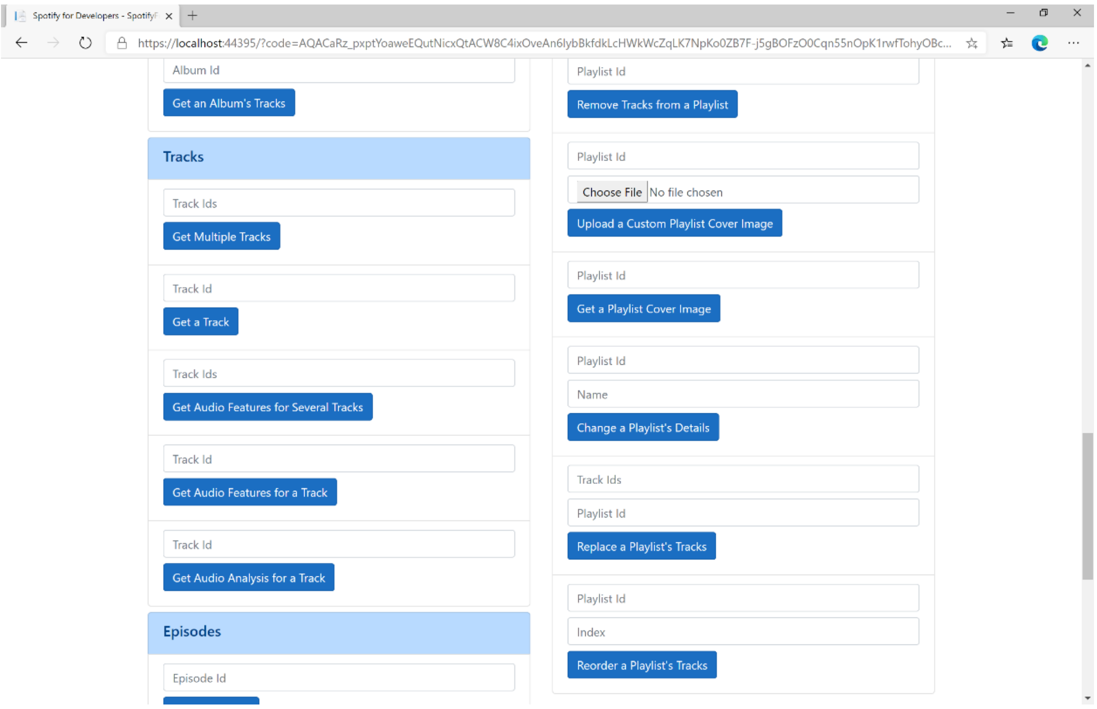
Step 8



Finally, in **Visual Studio 2019** select **IIS Application** to run the **Web Application**

Step 9

Once the **Web Application** is running and you select **Authorisation Code Flow Login** or **Implicit Grant Flow Login** and scroll down you should see something like the following:

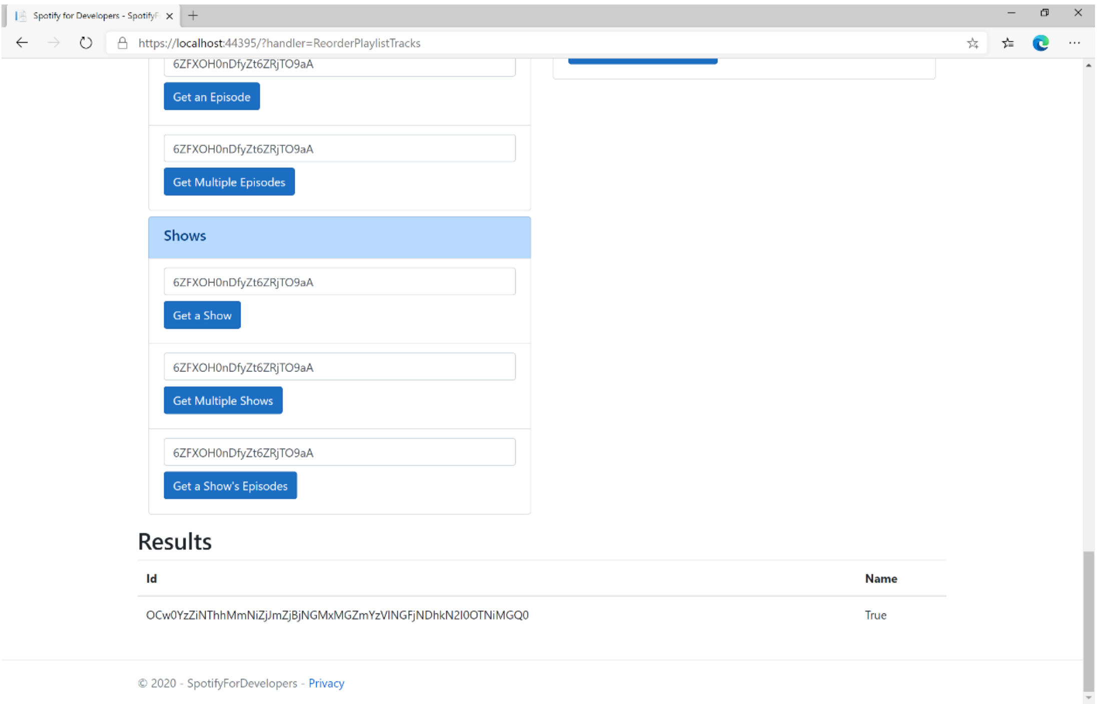


https://localhost:44395/?code=AQACaRz_pxptYoaweEQutNicxQtACW8C4ixOveAn6lybBkfdkLcHWkWcZqLK7NpKo0ZB7F-j5gBOFzO0Cqn55nOpK1rwfTohyOBc...

Spotify for Developers - SpotifyF
Album Id
Get an Album's Tracks
Tracks
Track Ids
Get Multiple Tracks
Track Id
Get a Track
Track Ids
Get Audio Features for Several Tracks
Track Id
Get Audio Features for a Track
Track Id
Get Audio Analysis for a Track
Episodes
Episode Id
Playlist Id
Remove Tracks from a Playlist
Playlist Id
Choose File No file chosen
Upload a Custom Playlist Cover Image
Playlist Id
Get a Playlist Cover Image
Playlist Id
Name
Change a Playlist's Details
Track Ids
Playlist Id
Replace a Playlist's Tracks
Playlist Id
Index
Reorder a Playlist's Tracks

Step 10

You can use **Get a Playlist's Tracks** and **Playlist Id** from **Create a Playlists** to check the **Order** of the **Tracks** in the **Playlist**. You can enter above **Reorder a Playlist's Tracks** the **Playlist Id** from **Create a Playlist** into **Playlist Id**, then enter an **Index** e.g. **1** and select **Reorder a Playlist's Tracks** and scroll down to view **Results** like the following:



https://localhost:44395/?handler=ReorderPlaylistTracks

Results

Id	Name
OCw0YzZiNThhMmNiZjJmZjBjNGMxMGZmYzVlNGFjNDhkN2I0OTNiMGQ0	True

© 2020 - SpotifyForDevelopers - Privacy

Step 11

■

You can stop the **web application** in **Visual Studio 2019** by selecting the **Stop debugging** button

Step 12

×

You can exit **Visual Studio 2019** by selecting the **Close** button in the top right of the **application** as that completes this part of the workshop

Library

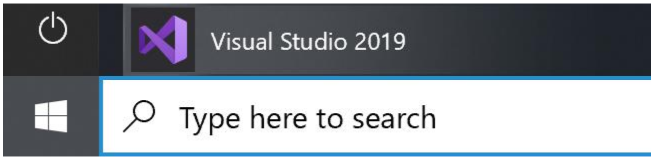
Save Albums for Current User

Save one or more albums to the current user's 'Your Music' library.

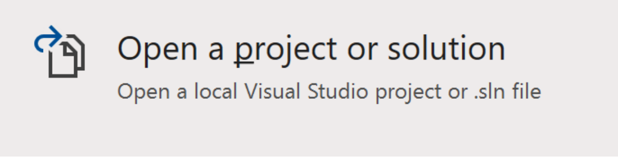
PUT https://api.spotify.com/v1/me/albums?ids={ids}	
Header	
Authorization	User Token from Spotify Accounts service with **user-library-modify** scope
Query Parameter	
ids	Comma-separated list of the Spotify Ids for the Albums. Maximum: 50 Ids.

Header	Response
Success	
Http Status 201 Created	
Error	
HTTP Status 403 Forbidden	Returned when not Authorised
Error Code	Error Object

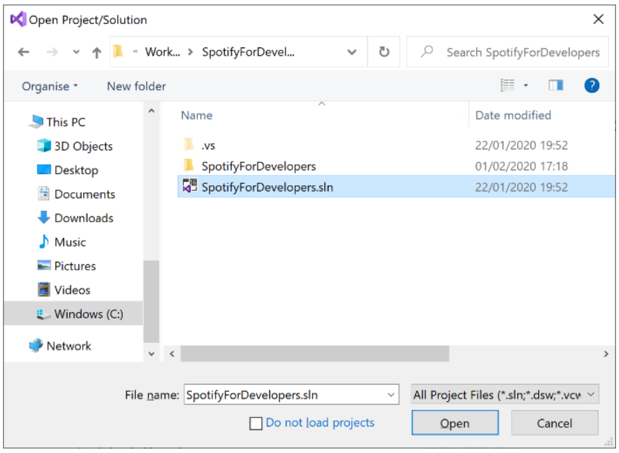
Step 1



In **Windows 10** choose **Start**, and then from the **Start Menu** find and select **Visual Studio 2019**

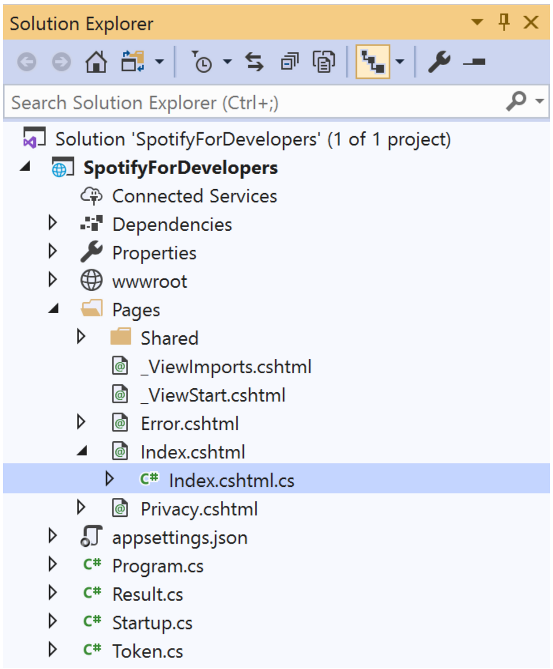

Open a project or solution
Open a local Visual Studio project or .sln file

Once done, from the **Get started** screen for **Visual Studio 2019** select **Open a project or solution**



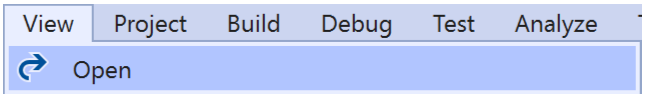
Then locate and select **SpotifyForDevelopers.sln** and select **Open** if you don't have this file already then please follow the previous parts of the workshop including **Getting Started**, **Authorisation Guide**, **Search & Browse**, **Playlists & Artists**, **Albums & Tracks**, **Episodes & Shows**, **Follow** and **Playlists**

Step 2



Once opened, in the **Solution Explorer** open the **Pages** section, then open the **Index.cshtml** section and select **Index.cshtml.cs**

Step 3



Then from the **Menu** choose **View** and then **Open**

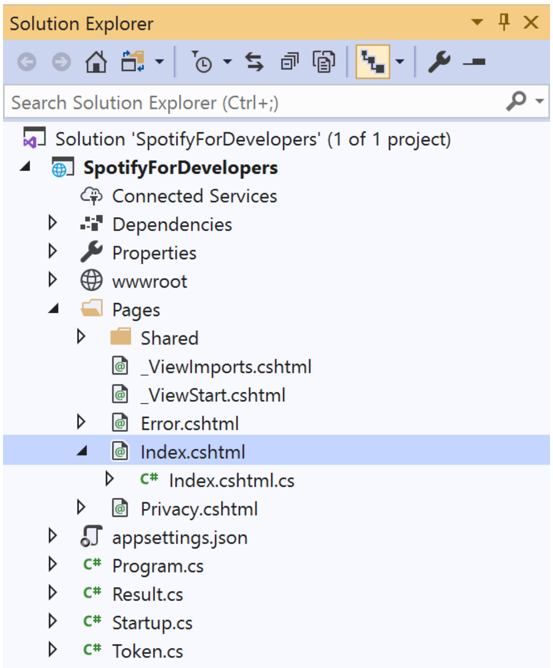
Step 4

In the **Code View** for **Index.cshtml.cs** below the **method** for `public async Task<IActionResult> OnPostReorderPlaylistTracksAsync(...) { ... }` enter the following **method**:

```
public async Task<IActionResult> OnPostSaveUserAlbumsAsync(string value)
{
    LoadToken();
    var values = value.Split(",").ToList();
    var result = await Api.SaveUserAlbumsAsync(values);
    if (result != null)
    {
        Results = new List<Result>() { new Result()
        {
            Id = result.Code.ToString(),
            Name = result.Success.ToString()
        }};
    }
    return Page();
}
```

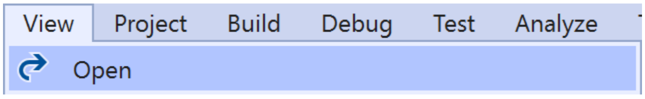
The **method** for `OnPostSaveUserAlbumsAsync` **saves** the **albums** to the **Library** with `Value` of the **Album Ids** of the **album** and populate the **property** for `Results` of the **success** of the operation accordingly.

Step 5



In the **Solution Explorer** in the **Pages** section select **Index.cshtml**

Step 6



Then from the **Menu** choose **View** and then **Open**

Step 7

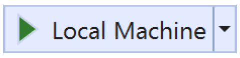
Once in the **Code View** for **Index.cshtml** above `<!-- Spotify Web API User Authorisation -->` enter the following:

```
<ul class="list-group mb-2">
    <li class="list-group-item list-group-item-primary">
        <h5 class="list-group-item-heading">Library</h5>
    </li>
    <li class="list-group-item">
        <form asp-page-handler="SaveUserAlbums" method="post">
            <input asp-for="Value" placeholder="Album Ids" class="form-control mb-2" />
            <button class="btn btn-primary mb-2">
                Save Albums for Current User
            </button>
        </form>
    </li>

    <!-- Library -->
</ul>
```

This `form` will **post** to the **method** for `SaveUserAlbums` with the `Value` of the **Album Ids** and will output to the **Results**.

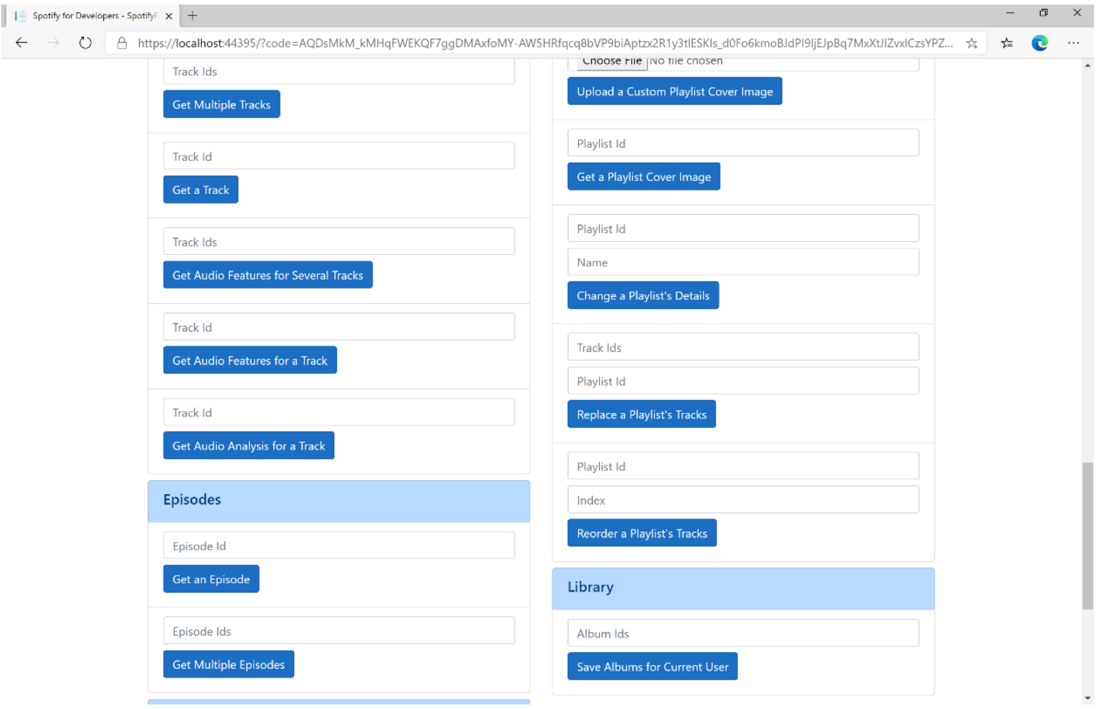
Step 8



Finally, in **Visual Studio 2019** select **IIS Application** to run the **Web Application**

Step 9

Once the **Web Application** is running and you select **Authorisation Code Flow Login** or **Implicit Grant Flow Login** and scroll down you should see something like the following:

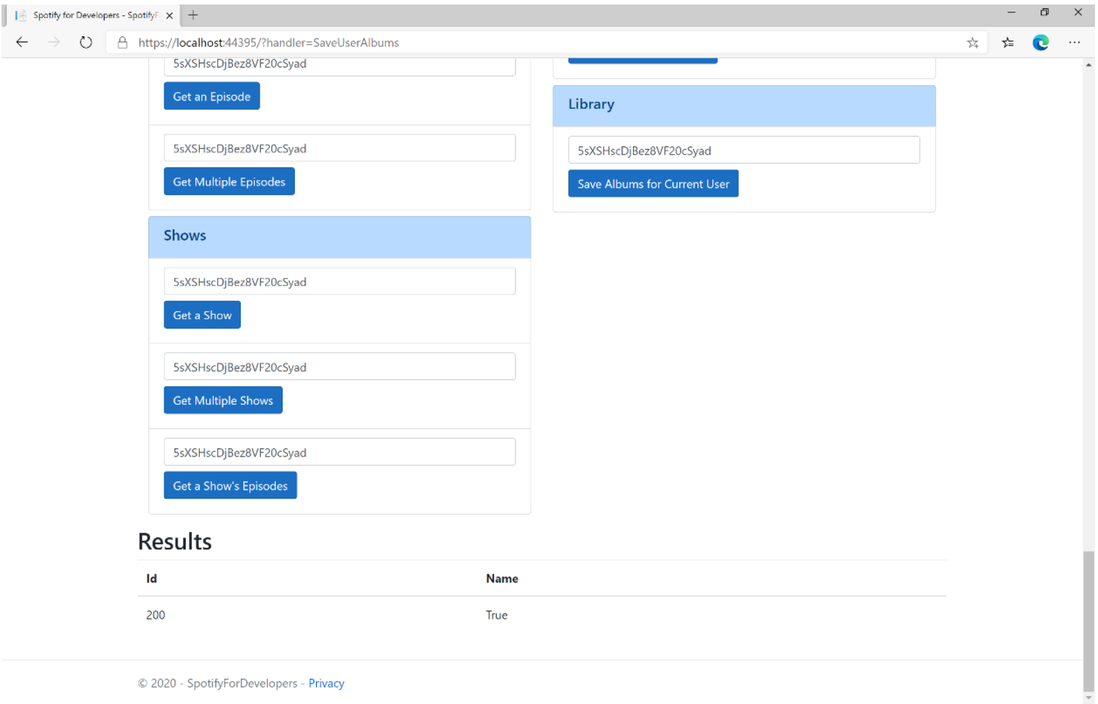


https://localhost:44395/?code=AQDsMkM_kMHqFWEKQF7ggDMAxfoMY-AW5HRfqcq8bVP9biAptzx2R1y3tlESKIs_d0Fo6kmoBJdPI9IjEJpBq7MxXtJIZvxlCzsYPZ...

Spotify for Developers - Spotify
Track Ids
Get Multiple Tracks
Track Id
Get a Track
Track Ids
Get Audio Features for Several Tracks
Track Id
Get Audio Features for a Track
Track Id
Get Audio Analysis for a Track
Episodes
Episode Id
Get an Episode
Episode Ids
Get Multiple Episodes
Choose File No file chosen
Upload a Custom Playlist Cover Image
Playlist Id
Get a Playlist Cover Image
Playlist Id
Name
Change a Playlist's Details
Track Ids
Playlist Id
Replace a Playlist's Tracks
Playlist Id
Index
Reorder a Playlist's Tracks
Library
Album Ids
Save Albums for Current User

Step 10

You can then enter some **Album Ids** from **Get All New Releases** above **Save Albums for Current User** and then select **Save Albums for Current User** and scroll down to view **Results** like the following:



https://localhost:44395/?handler=SaveUserAlbums

5sXSHscDjBez8VF20cSyad

Results

Id	Name
200	True

© 2020 - SpotifyForDevelopers - Privacy

Step 11

■

You can stop the **web application** in **Visual Studio 2019** by selecting the **Stop debugging** button

Step 12

×

You can choose to exit **Visual Studio 2019** by selecting the **Close** button in the top right of the **application** as that completes this part of the workshop

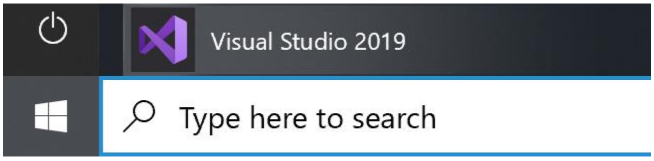
Save Tracks for User

Save one or more tracks to the current user's 'Your Music' library.

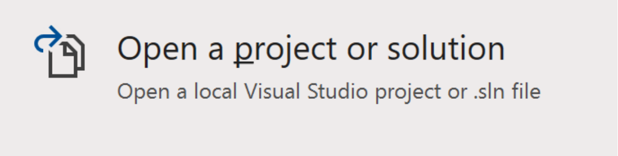
PUT https://api.spotify.com/v1/me/tracks	
Header	
Authorization	User Token from Spotify Accounts service with **user-library-modify** scope
Query Parameter	
ids	Comma-separated list of the Spotify Ids for the Tracks. Maximum: 50 Ids.

Header	Response
Success	
Http Status 200 OK	
Error	
HTTP Status 403 Forbidden	Returned when not Authorised or more than 10,000 tracks in Your Music
Error Code	Error Object

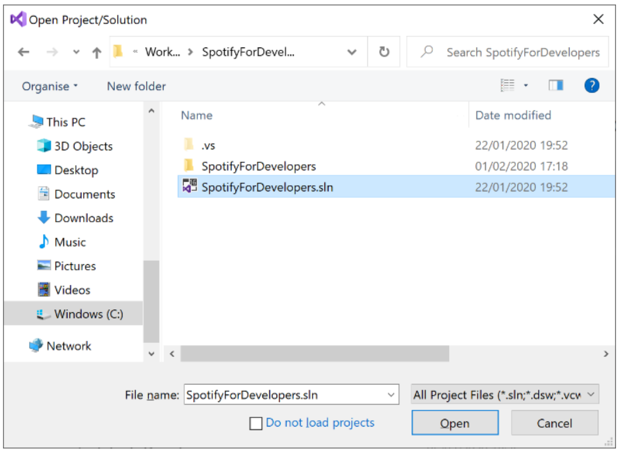
Step 1



If you chose to close **Visual Studio 2019** previously, in **Windows 10** choose **Start**, and then from the **Start Menu** find and select **Visual Studio 2019**

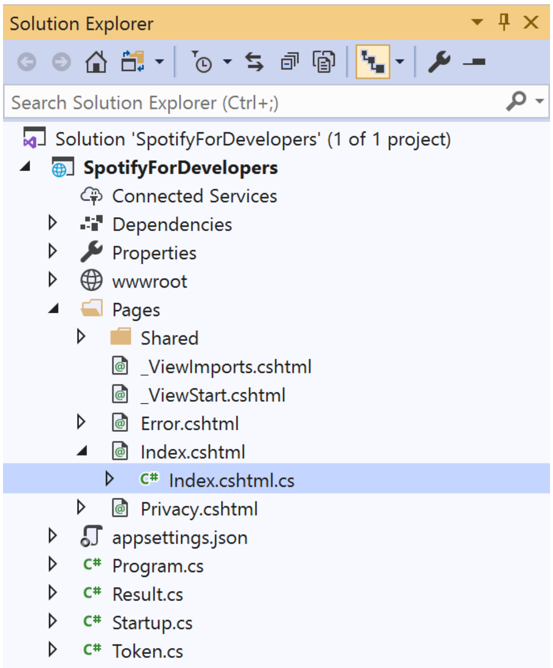

Open a project or solution
Open a local Visual Studio project or .sln file

Once done, from the **Get started** screen for **Visual Studio 2019** select **Open a project or solution**



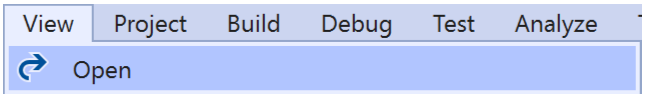
Then locate and select **SpotifyForDevelopers.sln** and select **Open** if you don't have this file already then please follow the previous parts of the workshop including **Getting Started**, **Authorisation Guide**, **Search & Browse**, **Playlists & Artists**, **Albums & Tracks**, **Episodes & Shows**, **Follow**, **Playlists** and **Library**

Step 2



Once opened, in the **Solution Explorer** open the **Pages** section, then open the **Index.cshtml** section and select **Index.cshtml.cs**

Step 3



Then from the **Menu** choose **View** and then **Open**

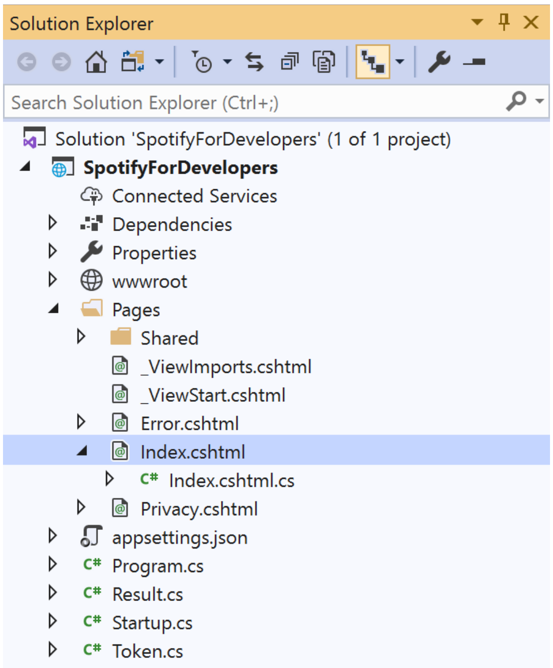
Step 4

In the **Code View** for **Index.cshtml.cs** below the **method** for `public async Task<IActionResult> OnPostSaveUserAlbumsAsync(...) { ... }` enter the following **method**:

```
public async Task<IActionResult> OnPostSaveUserTracksAsync(string value)
{
    LoadToken();
    var values = value.Split(",").ToList();
    var result = await Api.SaveUserTracksAsync(values);
    if (result != null)
    {
        Results = new List<Result>() { new Result()
        {
            Id = result.Code.ToString(),
            Name = result.Success.ToString()
        }};
    }
    return Page();
}
```

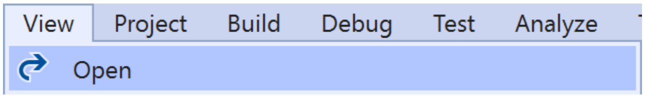
The **method** for `OnPostSaveUserTracksAsync` **saves** the **tracks** to the **Library** with `Value` of the **Track Ids** of the **track** and populate the **property** for `Results` of the **success** of the operation accordingly.

Step 5



In the **Solution Explorer** in the **Pages** section select **Index.cshtml**

Step 6



Then from the **Menu** choose **View** and then **Open**

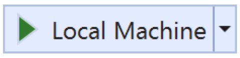
Step 7

Once in the **Code View** for **Index.cshtml** above `<!-- Library -->` enter the following:

```
<li class="list-group-item">
    <form asp-page-handler="SaveUserTracks" method="post">
        <input asp-for="Value" placeholder="Track Ids" class="form-control mb-2" />
        <button class="btn btn-primary mb-2">
            Save Tracks for User
        </button>
    </form>
</li>
```

This `form` will **post** to the **method** for `SaveUserTracks` with the `Value` of the **Track Ids** and will output to the **Results**.

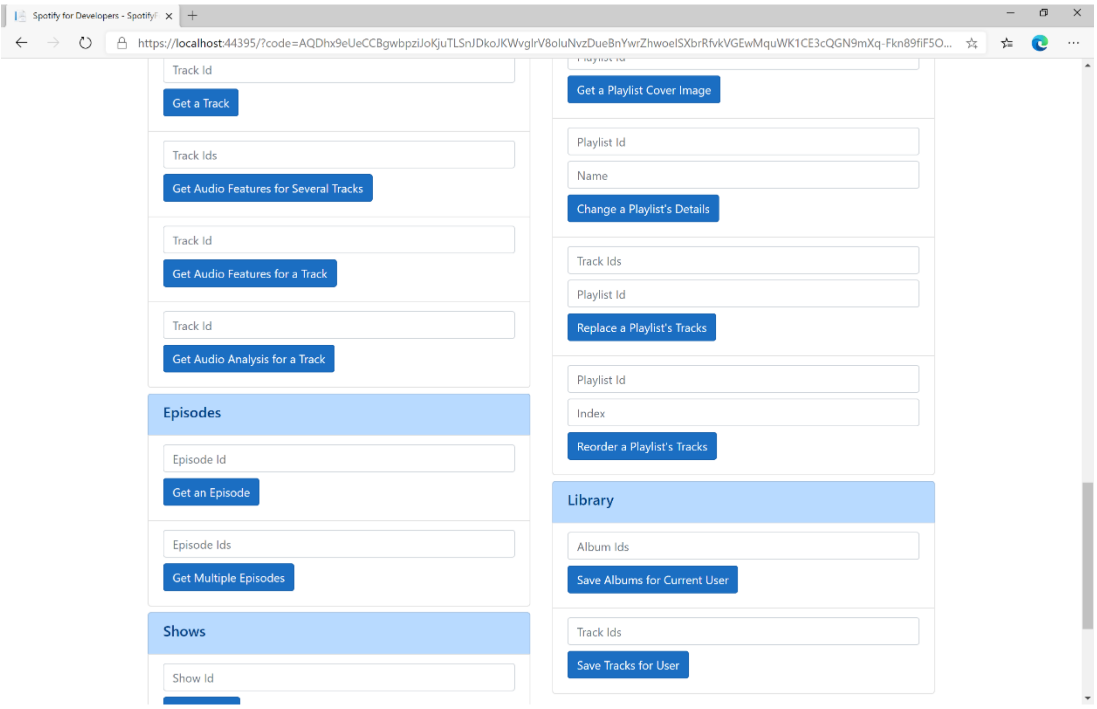
Step 8



Finally, in **Visual Studio 2019** select **IIS Application** to run the **Web Application**

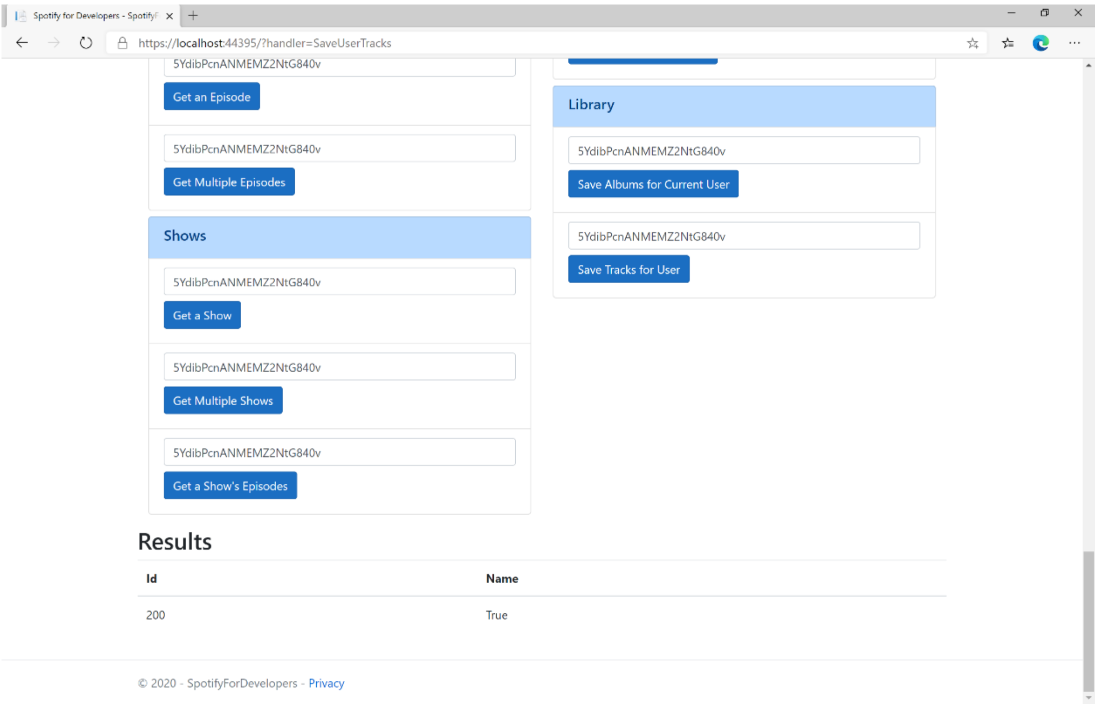
Step 9

Once the **Web Application** is running and you select **Authorisation Code Flow Login** or **Implicit Grant Flow Login** and scroll down you should see something like the following:


Spotify for Developers - Spotify
https://localhost:44395/?code=AQDhx9eUeCCBgwbpziJoKjuTLSnJDkoJKWvglrV8oluNvzDueBnYwrZhwoelSXbrRfvkVGEwMquWK1CE3cQGN9mXq-Fkn89fiF5O...
Track Id
Get a Track
Track Ids
Get Audio Features for Several Tracks
Track Id
Get Audio Features for a Track
Track Id
Get Audio Analysis for a Track
Episodes
Episode Id
Get an Episode
Episode Ids
Get Multiple Episodes
Shows
Show Id
Get a Playlist Cover Image
Playlist Id
Name
Change a Playlist's Details
Track Ids
Playlist Id
Replace a Playlist's Tracks
Playlist Id
Index
Reorder a Playlist's Tracks
Library
Album Ids
Save Albums for Current User
Track Ids
Save Tracks for User

Step 10

You can then enter some **Track Ids** from an **Album Id** from **Get All New Releases** using **Get an Album's Tracks** into **Track Ids** above **Save Tracks for User** and then select **Save Tracks for User** and scroll down to view **Results** like the following:



5YdibPcnANMEMZ2NtG840v

5YdibPcnANMEMZ2NtG840v

5YdibPcnANMEMZ2NtG840v

5YdibPcnANMEMZ2NtG840v

5YdibPcnANMEMZ2NtG840v

5YdibPcnANMEMZ2NtG840v

5YdibPcnANMEMZ2NtG840v

Results

Id	Name
200	True

© 2020 - SpotifyForDevelopers - Privacy

Step 11

You can stop the **web application** in **Visual Studio 2019** by selecting the **Stop debugging** button

Step 12

You can choose to exit **Visual Studio 2019** by selecting the **Close** button in the top right of the **application** as that completes this part of the workshop

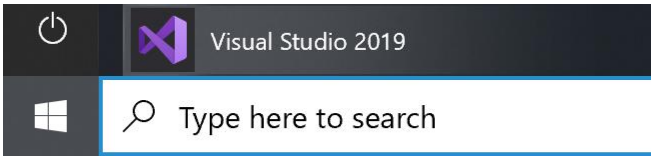
Save Shows for Current User

Save one or more shows to current Spotify user's library.

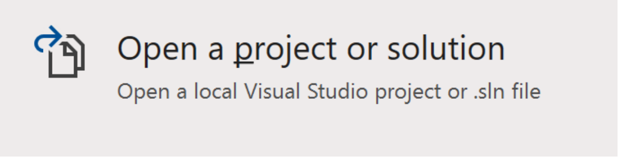
PUT https://api.spotify.com/v1/me/shows	
Header	
Authorization	User Token from Spotify Accounts service with **user-library-modify** scope
Query Parameter	
ids	Comma-separated list of Spotify Ids for the shows to be added to the user's library.

Header	Response
Success	
Http Status 200 OK	
Error	
HTTP Status 403 Forbidden	Returned when not Authorised or more than 10,000 items saved in library
Error Code	Error Object

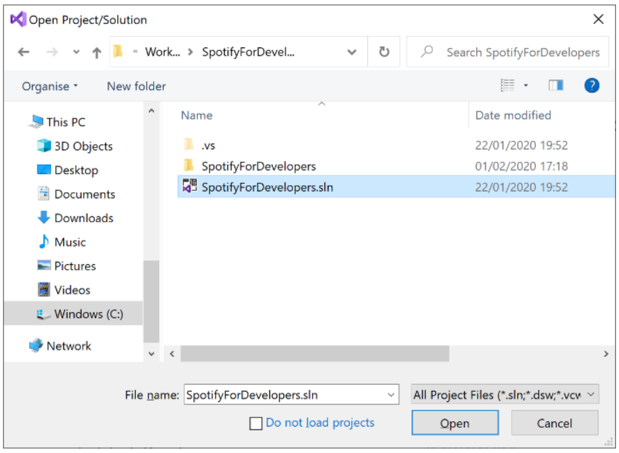
Step 1



If you chose to close **Visual Studio 2019** previously, in **Windows 10** choose **Start**, and then from the **Start Menu** find and select **Visual Studio 2019**

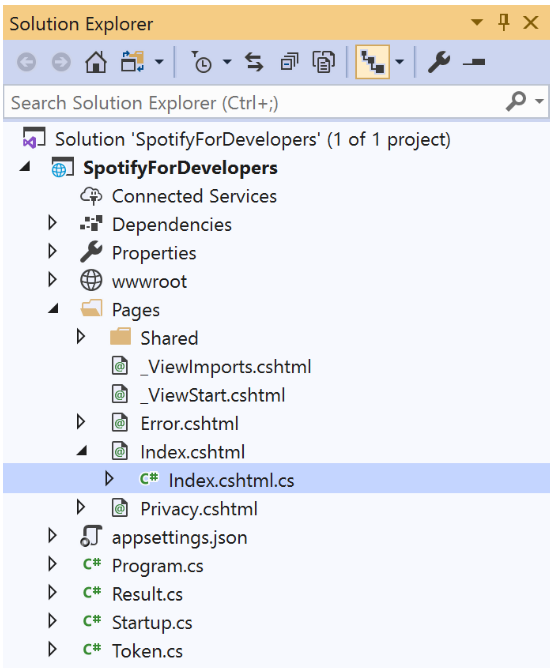

Open a project or solution
Open a local Visual Studio project or .sln file

Once done, from the **Get started** screen for **Visual Studio 2019** select **Open a project or solution**



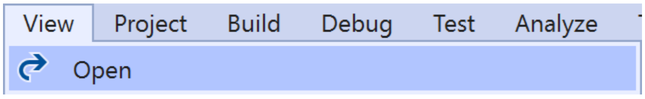
Then locate and select **SpotifyForDevelopers.sln** and select **Open** if you don't have this file already then please follow the previous parts of the workshop including **Getting Started**, **Authorisation Guide**, **Search & Browse**, **Playlists & Artists**, **Albums & Tracks**, **Episodes & Shows**, **Follow**, **Playlists** and **Library**

Step 2



Once opened, in the **Solution Explorer** open the **Pages** section, then open the **Index.cshtml** section and select **Index.cshtml.cs**

Step 3



Then from the **Menu** choose **View** and then **Open**

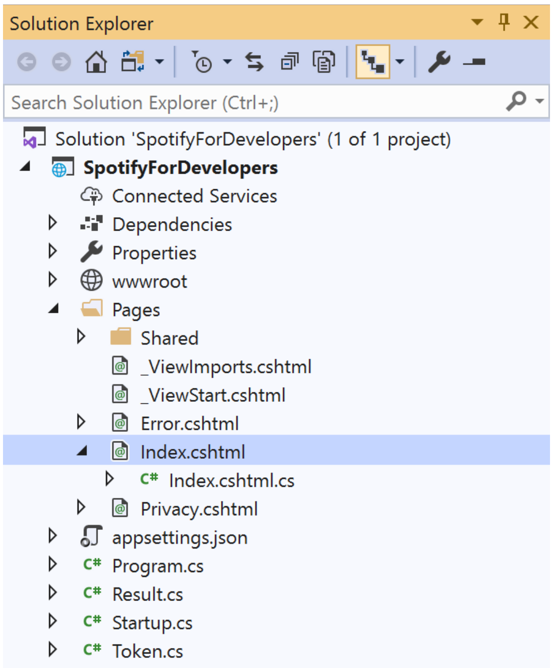
Step 4

In the **Code View** for **Index.cshtml.cs** below the **method** for `public async Task<IActionResult> OnPostSaveUserTracksAsync(...) { ... }` enter the following **method**:

```
public async Task<IActionResult> OnPostSaveUserShowsAsync(string value)
{
    LoadToken();
    var values = value.Split(",").ToList();
    var result = await Api.SaveUserShowsAsync(values);
    if (result != null)
    {
        Results = new List<Result>() { new Result()
        {
            Id = result.Code.ToString(),
            Name = result.Success.ToString()
        }};
    }
    return Page();
}
```

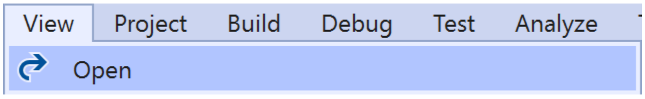
The **method** for `OnPostSaveUserShowsAsync` **saves** the **shows** to the **Library** with `Value` of the **Show Ids** of the **show** and populate the **property** for `Results` of the **success** of the operation accordingly.

Step 5



In the **Solution Explorer** in the **Pages** section select **Index.cshtml**

Step 6



Then from the **Menu** choose **View** and then **Open**

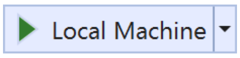
Step 7

Once in the **Code View** for **Index.cshtml** above `<!-- Library -->` enter the following:

```
<li class="list-group-item">
    <form asp-page-handler="SaveUserShows" method="post">
        <input asp-for="Value" placeholder="Show Ids" class="form-control mb-2" />
        <button class="btn btn-primary mb-2">
            Save Shows for User
        </button>
    </form>
</li>
```

This `form` will **post** to the **method** for `SaveUserTracks` with the `Value` of the **Track Ids** and will output to the **Results**.

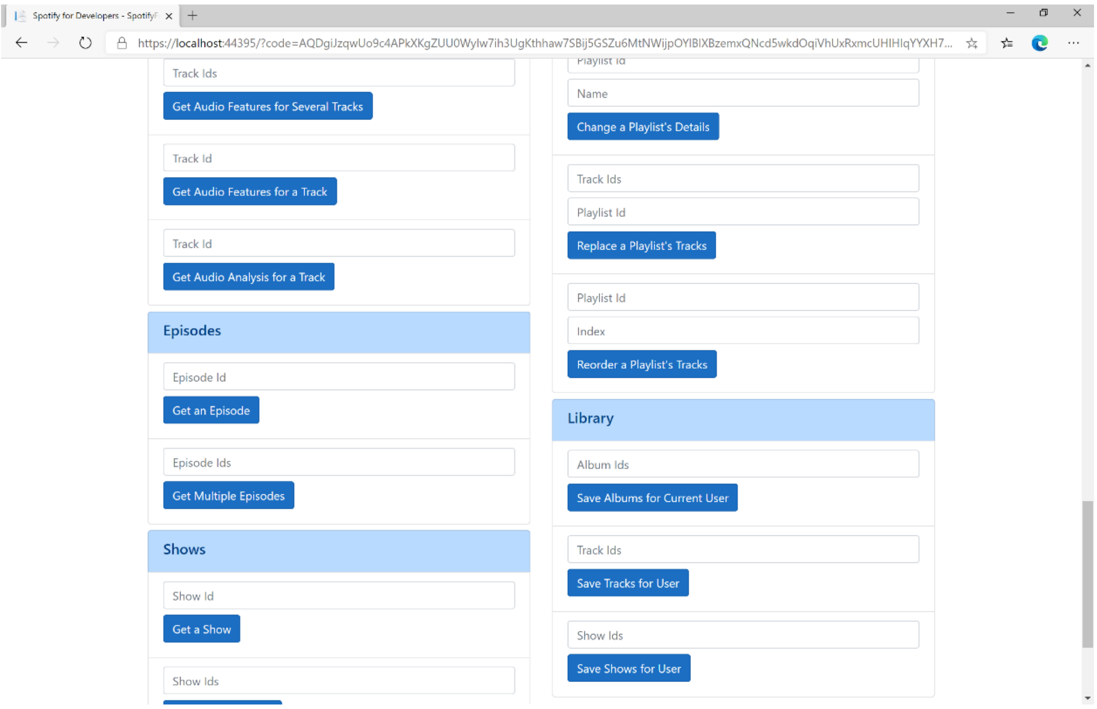
Step 8



Finally, in **Visual Studio 2019** select **IIS Application** to run the **Web Application**

Step 9

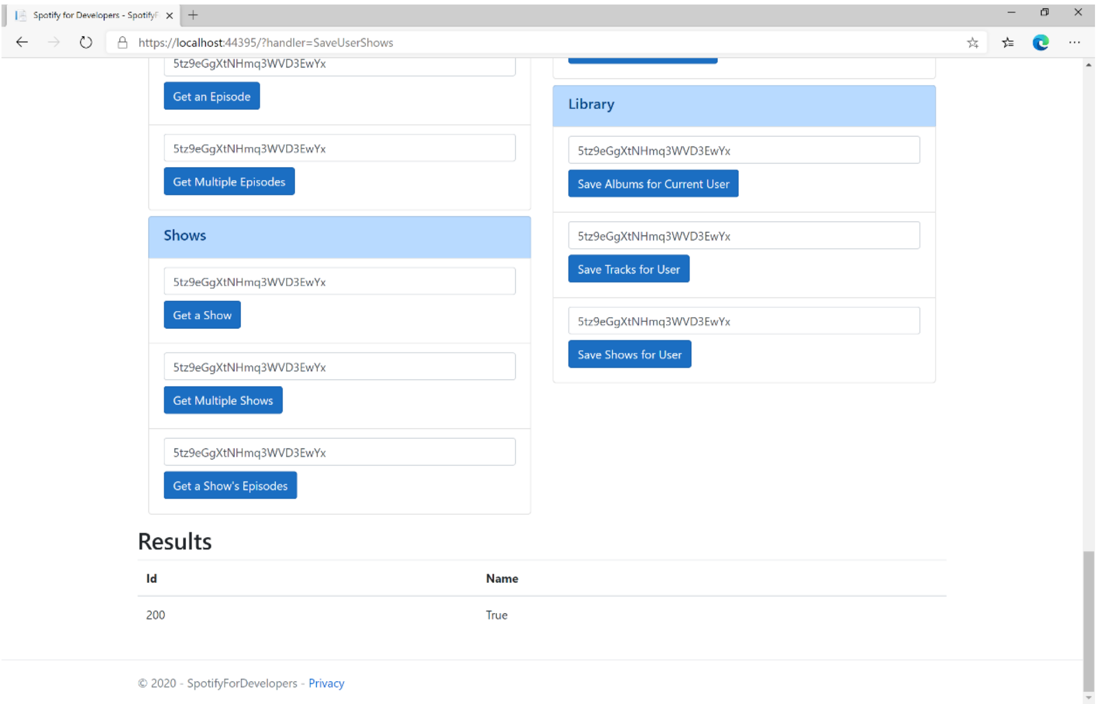
Once the **Web Application** is running and you select **Authorisation Code Flow Login** or **Implicit Grant Flow Login** and scroll down you should see something like the following:



Spotify for Developers - Spotify
https://localhost:44395/?code=AQDgiJzqwUo9c4APkXKgZUU0WyIw7ih3UgKthhaw7ih3UgKthhaw7SBij5GSZu6MtNWijpOYIBIXBzemxQNcd5wkdOqiVhUxRxmcUHIHlqYYXH7...
Track Ids
Get Audio Features for Several Tracks
Track Id
Get Audio Features for a Track
Track Id
Get Audio Analysis for a Track
Episodes
Episode Id
Get an Episode
Episode Ids
Get Multiple Episodes
Shows
Show Id
Get a Show
Show Ids
Playlist Id
Name
Change a Playlist's Details
Track Ids
Playlist Id
Replace a Playlist's Tracks
Playlist Id
Index
Reorder a Playlist's Tracks
Library
Album Ids
Save Albums for Current User
Track Ids
Save Tracks for User
Show Ids
Save Shows for User

Step 10

You can get a **Show Id** from **Search For An Item** using **Shows** and then enter this into **Show Ids** above **Save Shows for User** and then select **Save Shows for User** and scroll down to view **Results** like the following:



Spotify for Developers - Spotify
https://localhost:44395/?handler=SaveUserShows

5tz9eGgXtNHmq3WVD3EwYx

Get an Episode

5tz9eGgXtNHmq3WVD3EwYx

Get Multiple Episodes

Shows

5tz9eGgXtNHmq3WVD3EwYx

Get a Show

5tz9eGgXtNHmq3WVD3EwYx

Get Multiple Shows

5tz9eGgXtNHmq3WVD3EwYx

Get a Show's Episodes

Library

5tz9eGgXtNHmq3WVD3EwYx

Save Albums for Current User

5tz9eGgXtNHmq3WVD3EwYx

Save Tracks for User

5tz9eGgXtNHmq3WVD3EwYx

Save Shows for User

Results

Id	Name
200	True

© 2020 - SpotifyForDevelopers - Privacy

Step 11

You can stop the **web application** in **Visual Studio 2019** by selecting the **Stop debugging** button

Step 12

You can choose to exit **Visual Studio 2019** by selecting the **Close** button in the top right of the **application** as that completes this part of the workshop

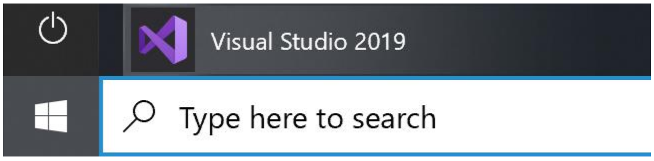
Check User's Saved Albums

Check if one or more albums is already saved in the current Spotify user's 'Your Music' library.

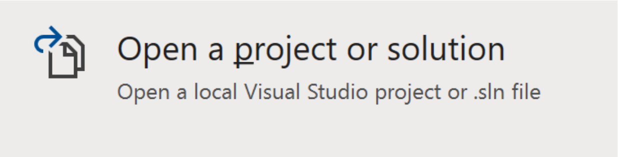
GET https://api.spotify.com/v1/me/albums/contains	
Header	
Authorization	User Token from Spotify Accounts service with **user-library-read** scope
Query Parameter	
ids	Comma-separated list of the Spotify Ids for the Albums. Maximum: 50 Ids.

Header	Response
Success	
Http Status 200 OK	Array of true or false values in same order as in which the Ids were provided
Error	
Error Code	Error Object

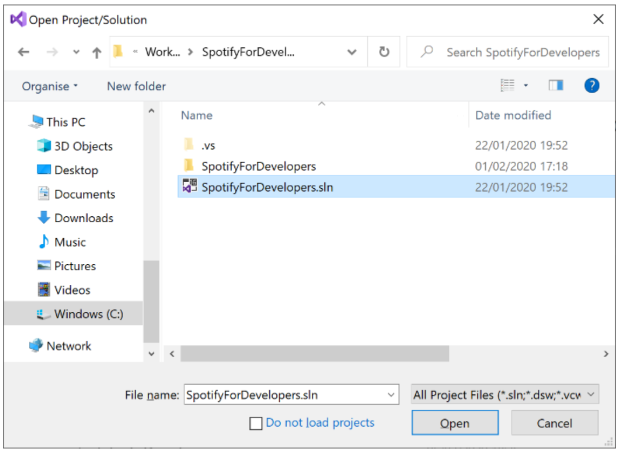
Step 1



If you chose to close **Visual Studio 2019** previously, in **Windows 10** choose **Start**, and then from the **Start Menu** find and select **Visual Studio 2019**

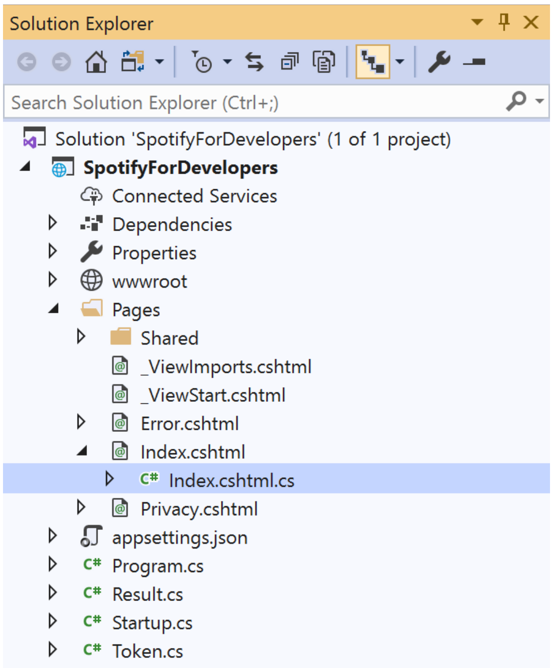

Open a project or solution
Open a local Visual Studio project or .sln file

Once done, from the **Get started** screen for **Visual Studio 2019** select **Open a project or solution**



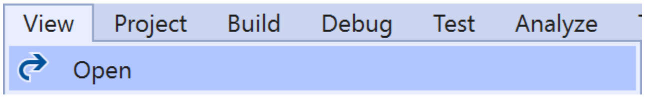
Then locate and select **SpotifyForDevelopers.sln** and select **Open** if you don't have this file already then please follow the previous parts of the workshop including **Getting Started**, **Authorisation Guide**, **Search & Browse**, **Playlists & Artists**, **Albums & Tracks**, **Episodes & Shows**, **Follow**, **Playlists** and **Library**

Step 2



Once opened, in the **Solution Explorer** open the **Pages** section, then open the **Index.cshtml** section and select **Index.cshtml.cs**

Step 3



Then from the **Menu** choose **View** and then **Open**

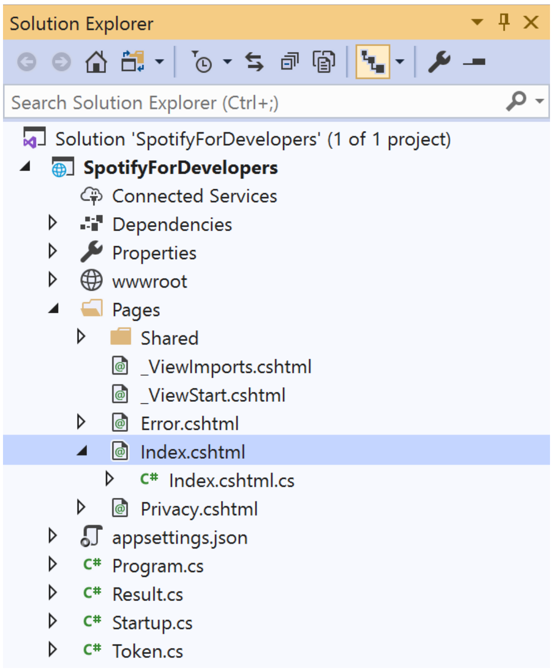
Step 4

In the **Code View** for **Index.cshtml.cs** below the **method** for `public async Task<IActionResult> OnPostSaveUserShowsAsync(...) { ... }` enter the following **method**:

```
public async Task<IActionResult> OnPostCheckUserSavedAlbumsAsync(string value)
{
    LoadToken();
    var values = value.Split(",").ToList();
    var results = await Api.CheckUserSavedAlbumsAsync(values);
    if (results != null)
    {
        Results = results.Select(result => new Result()
        {
            Name = result.ToString()
        });
    }
    return Page();
}
```

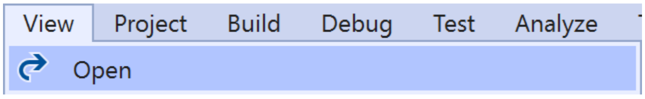
The **method** for `OnPostCheckUserSavedAlbumsAsync` is used to get the **saved state** of **albums** in the **Library** with `Value` of the **Album Ids** of the **track** and populate the **property** for `Results` accordingly.

Step 5



In the **Solution Explorer** in the **Pages** section select **Index.cshtml**

Step 6



Then from the **Menu** choose **View** and then **Open**

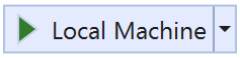
Step 7

Once in the **Code View** for **Index.cshtml** above `<!-- Library -->` enter the following:

```
<li class="list-group-item">
    <form asp-page-handler="CheckUserSavedAlbums" method="post">
        <input asp-for="Value" placeholder="Album Ids" class="form-control mb-2" />
        <button class="btn btn-primary mb-2">
            Check User's Saved Albums
        </button>
    </form>
</li>
```

This `form` will **post** to the **method** for `CheckUserSavedAlbums` with the `Value` of the **Album Ids** and will output to the **Results**.

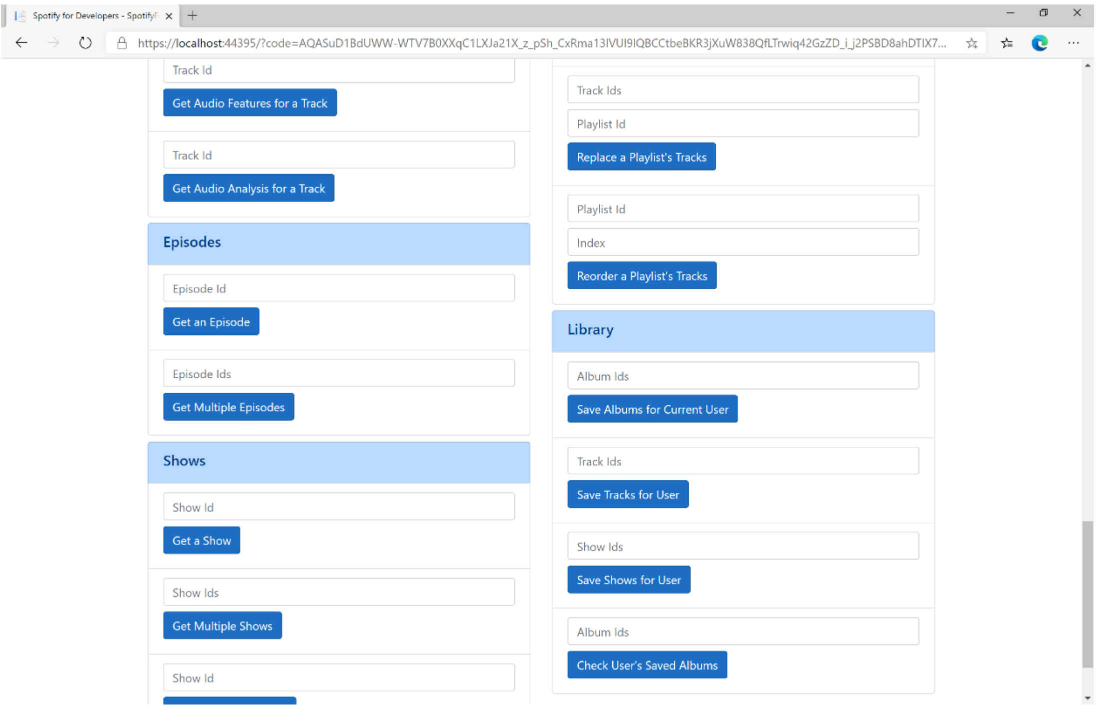
Step 8



Finally, in **Visual Studio 2019** select **IIS Application** to run the **Web Application**

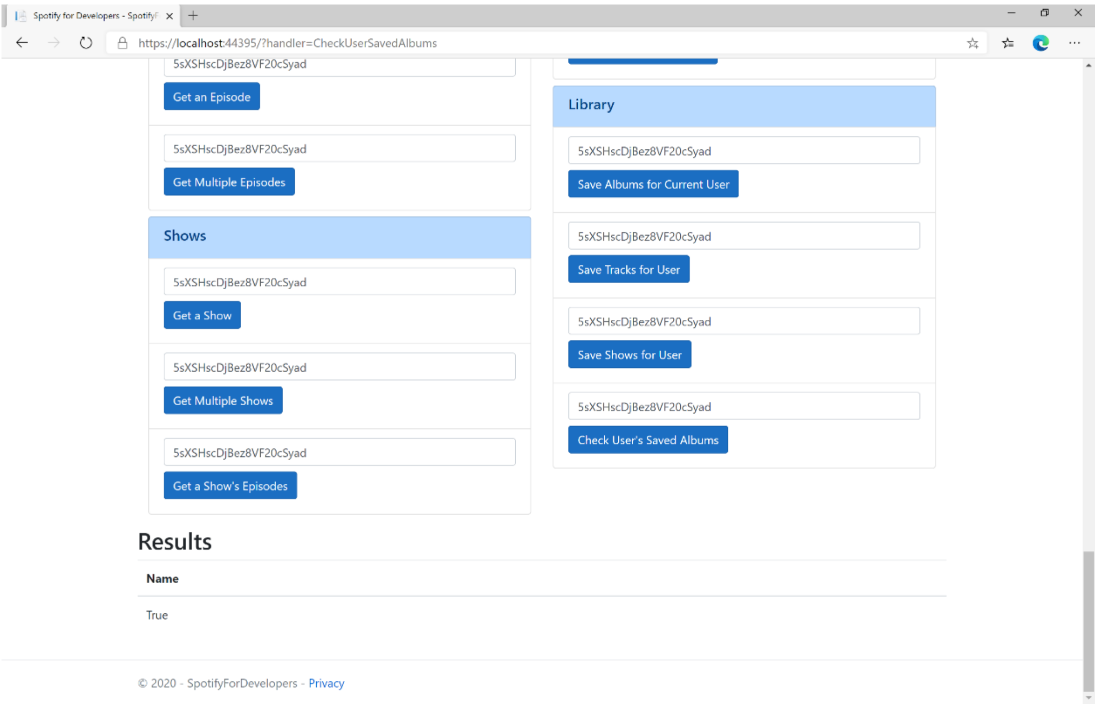
Step 9

Once the **Web Application** is running and you select **Authorisation Code Flow Login** or **Implicit Grant Flow Login** and scroll down you should see something like the following:


Spotify for Developers - Spotify
https://localhost:44395/?code=AQASuD1BdUWW-WTV7B0XXqC1LXJa21X_z_pSh_CxRma13IVUI9IQBCCtbeBKR3jXuW838QfLTrwiq42GzZD_i_j2PSBD8ahDTIX7...
Track Id
Get Audio Features for a Track
Track Id
Get Audio Analysis for a Track
Episodes
Episode Id
Get an Episode
Episode Ids
Get Multiple Episodes
Shows
Show Id
Get a Show
Show Ids
Get Multiple Shows
Show Id
Track Ids
Playlist Id
Replace a Playlist's Tracks
Playlist Id
Index
Reorder a Playlist's Tracks
Library
Album Ids
Save Albums for Current User
Track Ids
Save Tracks for User
Show Ids
Save Shows for User
Album Ids
Check User's Saved Albums

Step 10

You can then enter some **Album Ids** from **Get All New Releases** or copy the same **Album Ids** used in **Save Albums for Current User** into **Album Ids** above **Check User's Saved Albums** and then select **Check User's Saved Albums** and scroll down to view **Results** like the following:



Results

Name
True

© 2020 - SpotifyForDevelopers - Privacy

Step 11

You can stop the **web application** in **Visual Studio 2019** by selecting the **Stop debugging** button

Step 12

You can choose to exit **Visual Studio 2019** by selecting the **Close** button in the top right of the **application** as that completes this part of the workshop

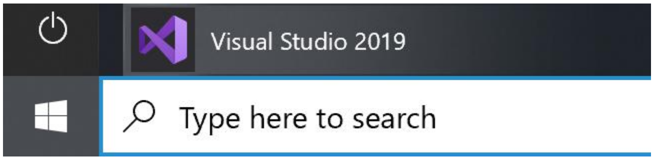
Check User's Saved Tracks

Check if one or more tracks is already saved in the current Spotify user's 'Your Music' library.

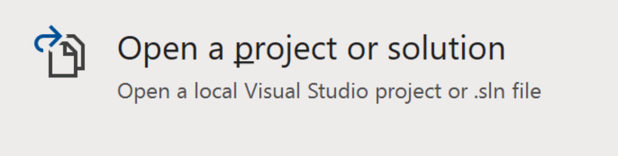
GET https://api.spotify.com/v1/me/tracks/contains	
Header	
Authorization	User Token from Spotify Accounts service with **user-library-read** scope
Query Parameter	
ids	Comma-separated list of the Spotify Ids for the Tracks. Maximum: 50 Ids.

Header	Response
Success	
Http Status 200 OK	Array of true or false values in same order as in which the Ids were provided
Error	
Error Code	Error Object

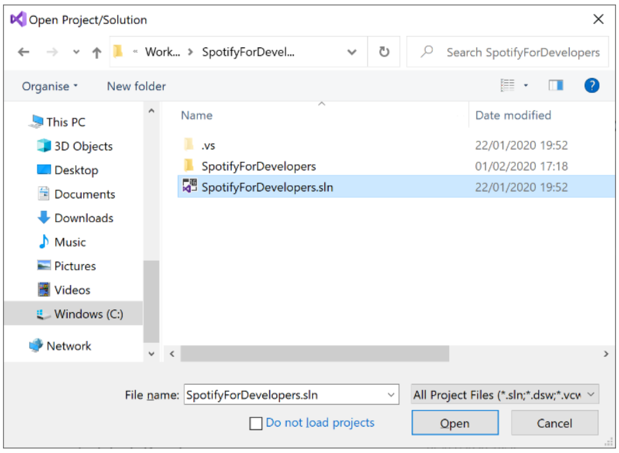
Step 1



If you chose to close **Visual Studio 2019** previously, in **Windows 10** choose **Start**, and then from the **Start Menu** find and select **Visual Studio 2019**


Open a project or solution
Open a local Visual Studio project or .sln file

Once done, from the **Get started** screen for **Visual Studio 2019** select **Open a project or solution**



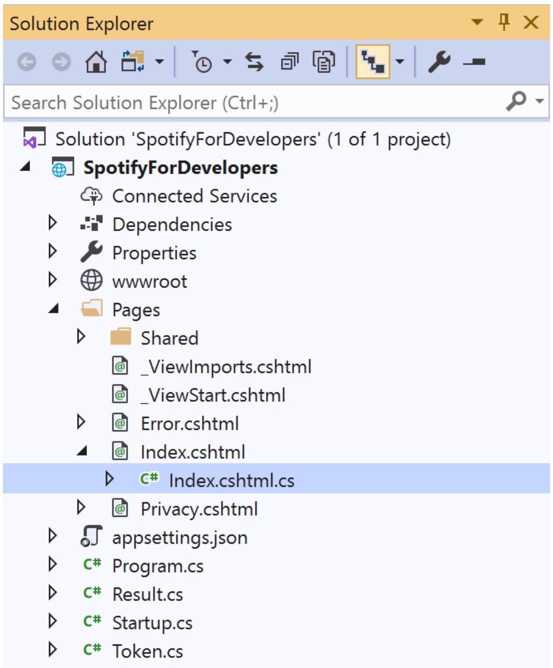
Name	Date modified
.vs	22/01/2020 19:52
SpotifyForDevelopers	01/02/2020 17:18
SpotifyForDevelopers.sln	22/01/2020 19:52

File name: SpotifyForDevelopers.sln   All Project Files (*.sln;*.dsw;*.vcw

□ Do not load projects

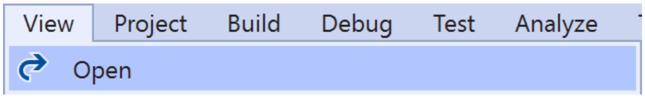
Then locate and select **SpotifyForDevelopers.sln** and select **Open** if you don't have this file already then please follow the previous parts of the workshop including **Getting Started**, **Authorisation Guide**, **Search & Browse**, **Playlists & Artists**, **Albums & Tracks**, **Episodes & Shows**, **Follow**, **Playlists** and **Library**

Step 2



Once opened, in the **Solution Explorer** open the **Pages** section, then open the **Index.cshtml** section and select **Index.cshtml.cs**

Step 3



Then from the **Menu** choose **View** and then **Open**

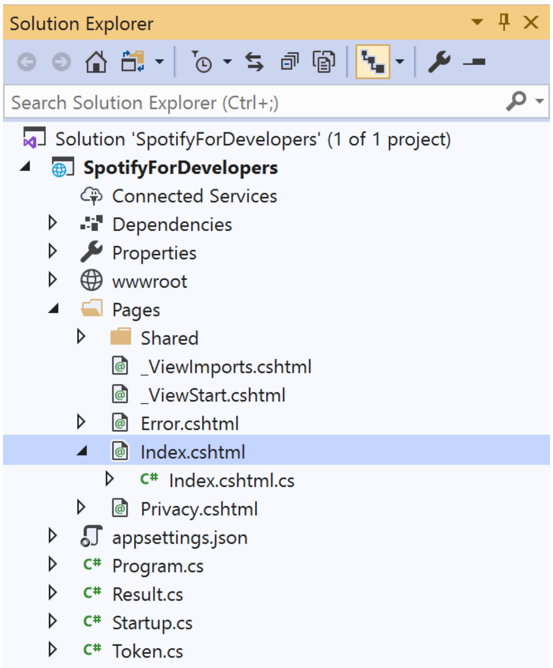
Step 4

In the **Code View** for **Index.cshtml.cs** below the **method** for `public async Task<IActionResult> OnPostCheckUserSavedAlbumsAsync(...) { ... }` enter the following **method**:

```
public async Task<IActionResult> OnPostCheckUserSavedTracksAsync(string value)
{
    LoadToken();
    var values = value.Split(",").ToList();
    var results = await Api.CheckUserSavedTracksAsync(values);
    if (results != null)
    {
        Results = results.Select(result => new Result()
        {
            Name = result.ToString()
        });
    }
    return Page();
}
```

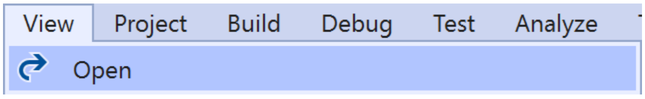
The **method** for `OnPostCheckUserSavedTracksAsync` is used to get the **saved state** of **tracks** in the **Library** with `Value` of the **Track Ids** of the **track** and populate the **property** for `Results` accordingly.

Step 5



In the **Solution Explorer** in the **Pages** section select **Index.cshtml**

Step 6



Then from the **Menu** choose **View** and then **Open**

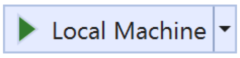
Step 7

Once in the **Code View** for **Index.cshtml** above `<!-- Library -->` enter the following:

```
<li class="list-group-item">
    <form asp-page-handler="CheckUserSavedTracks" method="post">
        <input asp-for="Value" placeholder="Track Ids" class="form-control mb-2" />
        <button class="btn btn-primary mb-2">
            Check User's Saved Tracks
        </button>
    </form>
</li>
```

This `form` will **post** to the **method** for `CheckUserSavedTracks` with the `Value` of the **Track Ids** and will output to the **Results**.

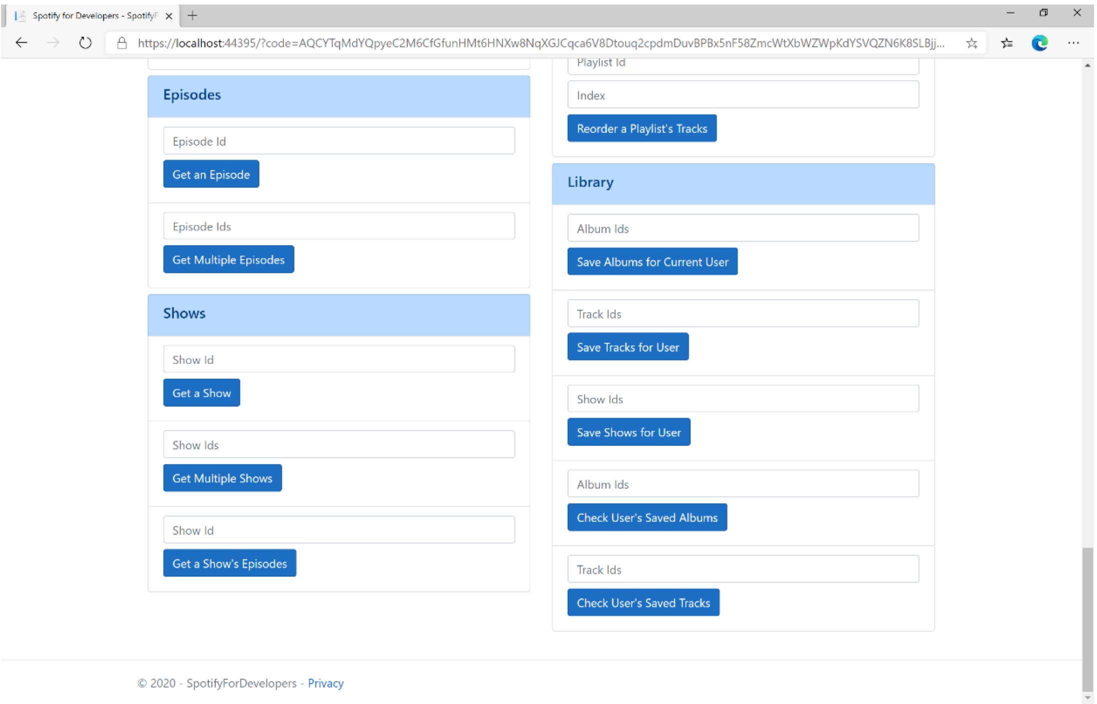
Step 8



Finally, in **Visual Studio 2019** select **IIS Application** to run the **Web Application**

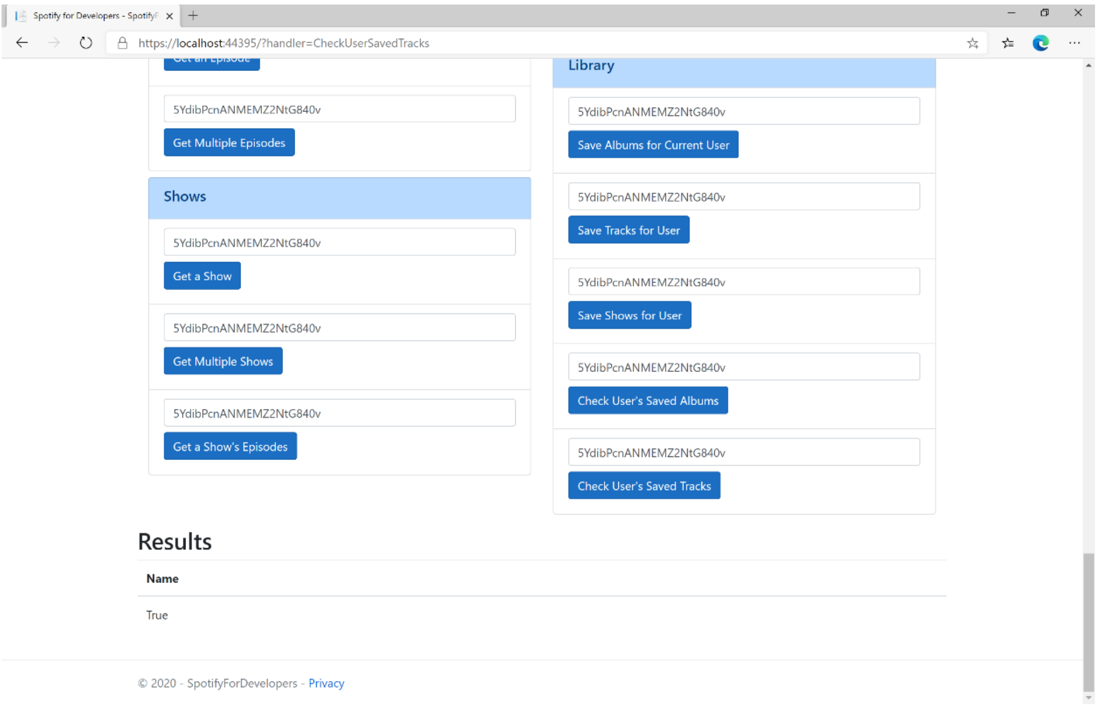
Step 9

Once the **Web Application** is running and you select **Authorisation Code Flow Login** or **Implicit Grant Flow Login** and scroll down you should see something like the following:


Spotify for Developers - Spotify
https://localhost:44395/?code=AQCYTqMdYQpyeC2M6CfGfunHMt6HNXw8NqXGJCqca6V8Dtouq2cpdmDuvBPBx5nF58ZmcWtXbWZWpKdYSVQZN6K8SLBjj...
Episodes
Episode Id
Get an Episode
Episode Ids
Get Multiple Episodes
Shows
Show Id
Get a Show
Show Ids
Get Multiple Shows
Show Id
Get a Show's Episodes
Playlist Id
Index
Reorder a Playlist's Tracks
Library
Album Ids
Save Albums for Current User
Track Ids
Save Tracks for User
Show Ids
Save Shows for User
Album Ids
Check User's Saved Albums
Track Ids
Check User's Saved Tracks
© 2020 - SpotifyForDevelopers - Privacy

Step 10

You can then enter some **Track Ids** from an **Album Id** from **Get All New Releases** using **Get an Album's Tracks** or copy the same **Track Ids** used in **Save Tracks for User** in **Track Ids** above **Check User's Saved Tracks** and then select **Check User's Saved Tracks** and scroll down to view **Results** like the following:



Results

Name
True

Step 11

■

You can stop the **web application** in **Visual Studio 2019** by selecting the **Stop debugging** button

Step 12

×

You can choose to exit **Visual Studio 2019** by selecting the **Close** button in the top right of the **application** as that completes this part of the workshop

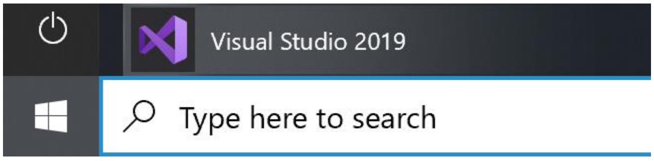
Check User's Saved Shows

Check if one or more shows is already saved in the current Spotify user's library

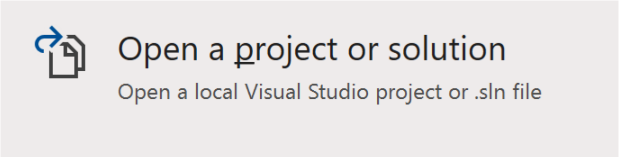
GET https://api.spotify.com/v1/me/shows/contains	
Header	
Authorization	User Token from Spotify Accounts service with **user-library-read** scope
Query Parameter	
ids	Comma-separated list of the Spotify Ids for the shows. Maximum: 50 ids

Header	Response
Success	
Http Status 200 OK	Array of true or false values in same order as in which the Ids were provided
Error	
Error Code	Error Object

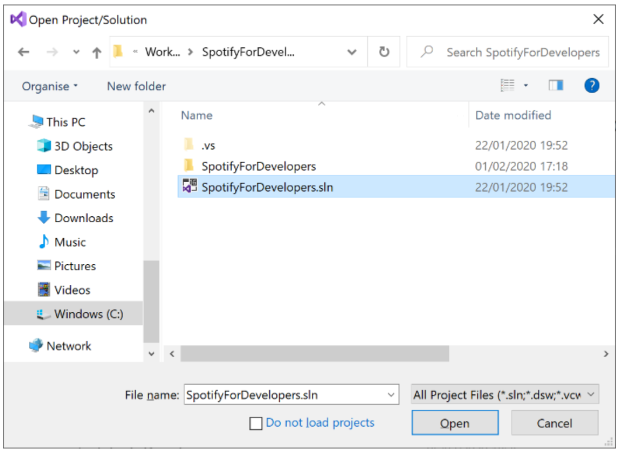
Step 1



If you chose to close **Visual Studio 2019** previously, in **Windows 10** choose **Start**, and then from the **Start Menu** find and select **Visual Studio 2019**


Open a project or solution
Open a local Visual Studio project or .sln file

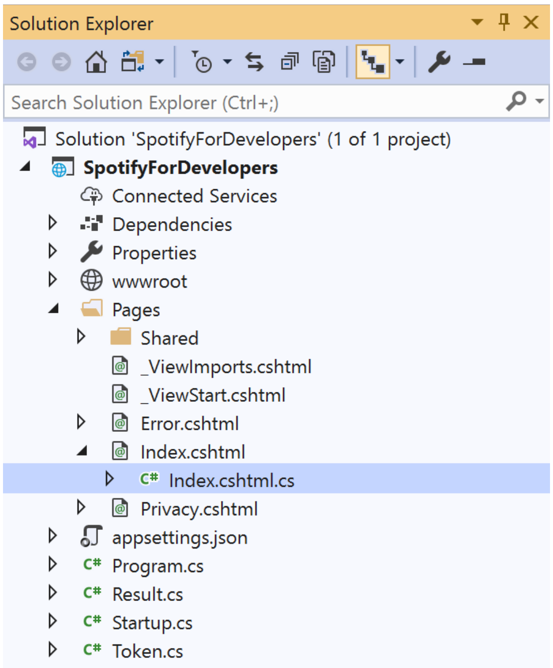
Once done, from the **Get started** screen for **Visual Studio 2019** select **Open a project or solution**



File name: SpotifyForDevelopers.sln

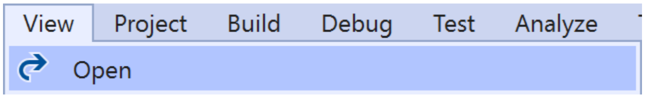
Then locate and select **SpotifyForDevelopers.sln** and select **Open** if you don't have this file already then please follow the previous parts of the workshop including **Getting Started**, **Authorisation Guide**, **Search & Browse**, **Playlists & Artists**, **Albums & Tracks**, **Episodes & Shows**, **Follow**, **Playlists** and **Library**

Step 2



Once opened, in the **Solution Explorer** open the **Pages** section, then open the **Index.cshtml** section and select **Index.cshtml.cs**

Step 3



Then from the **Menu** choose **View** and then **Open**

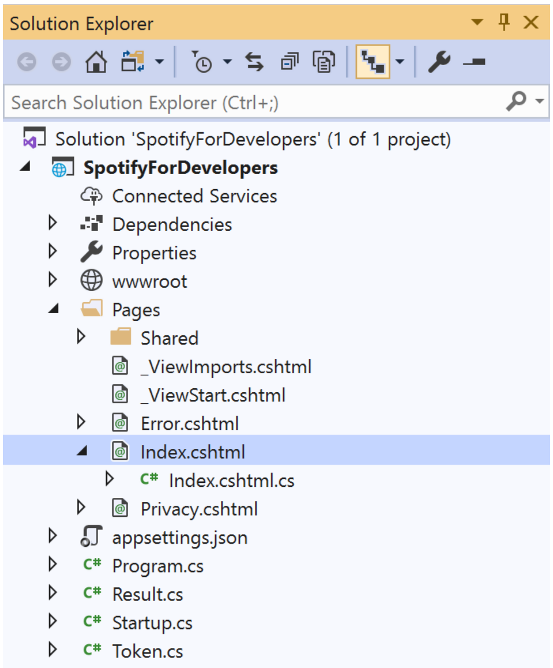
Step 4

In the **Code View** for **Index.cshtml.cs** below the **method** for `public async Task<IActionResult> CheckUserSavedTracksAsync(...) { ... }` enter the following **method**:

```
public async Task<IActionResult> OnPostCheckUserSavedShowsAsync(string value)
{
    LoadToken();
    var values = value.Split(",").ToList();
    var results = await Api.CheckUserSavedShowsAsync(values);
    if (results != null)
    {
        Results = results.Select(result => new Result()
        {
            Name = result.ToString()
        });
    }
    return Page();
}
```

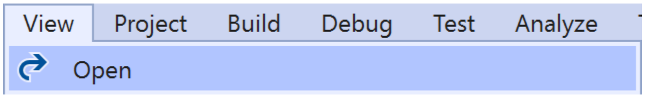
The **method** for `OnPostCheckUserSavedShowsAsync` is used to get the **saved state** of **shows** in the **Library** with `Value` of the **Shows Ids** of the **show** and populate the **property** for `Results` accordingly.

Step 5



In the **Solution Explorer** in the **Pages** section select **Index.cshtml**

Step 6



Then from the **Menu** choose **View** and then **Open**

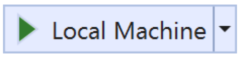
Step 7

Once in the **Code View** for **Index.cshtml** above `<!-- Library -->` enter the following:

```
<li class="list-group-item">
    <form asp-page-handler="CheckUserSavedShows" method="post">
        <input asp-for="Value" placeholder="Show Ids" class="form-control mb-2" />
        <button class="btn btn-primary mb-2">
            Check User's Saved Shows
        </button>
    </form>
</li>
```

This `form` will **post** to the **method** for `CheckUserSavedShows` with the `Value` of the **Show Ids** and will output to the **Results**.

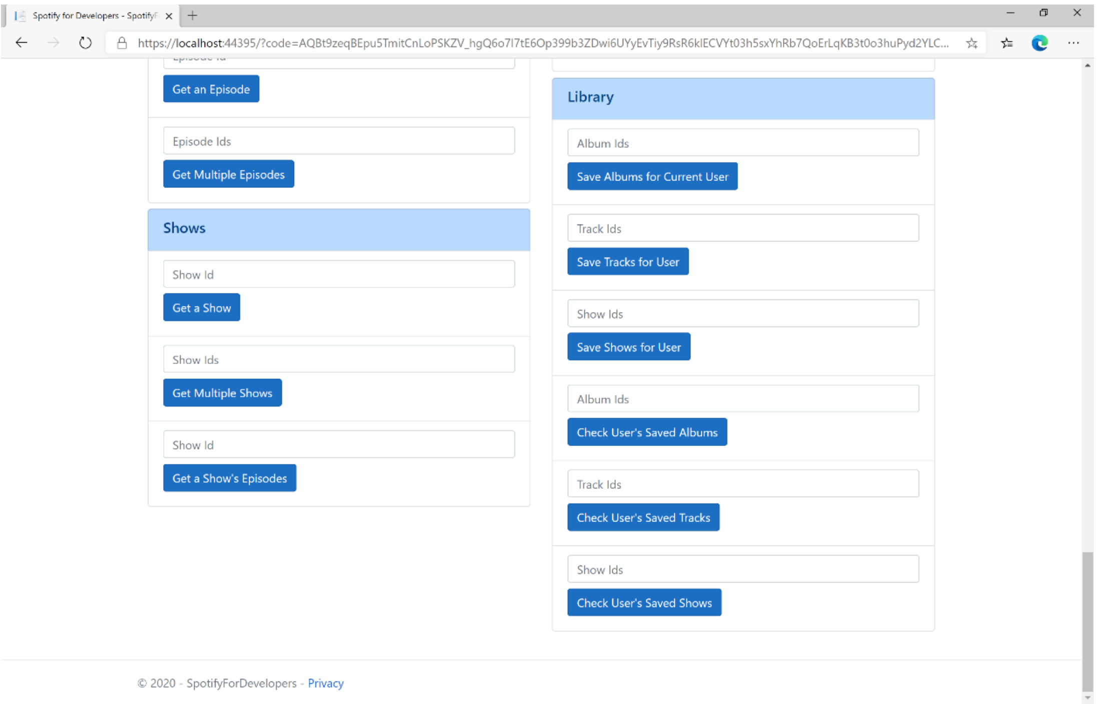
Step 8



Finally, in **Visual Studio 2019** select **IIS Application** to run the **Web Application**

Step 9

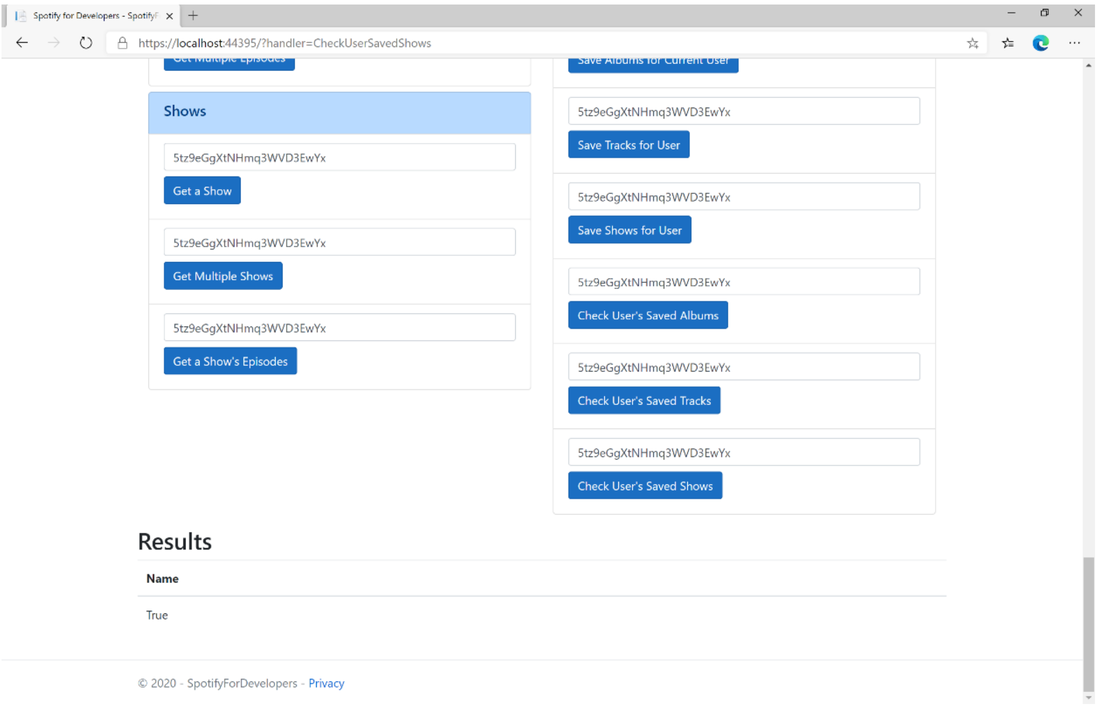
Once the **Web Application** is running and you select **Authorisation Code Flow Login** or **Implicit Grant Flow Login** and scroll down you should see something like the following:



https://localhost:44395/?code=AQBt9zeqBEpu5TmitCnLoPSKZV_hgQ6o7I7tE6Op399b3ZDwi6UYyEvTiy9RsR6klECVYt03h5sxYhRb7QoErLqKB3t0o3huPyd2YLC...

Step 10

You can get a **Show Id** from **Search For An Item** using **Shows** or copy the same **Shows Ids** used in **Save Shows for User** in **Show Ids** above **Check User's Saved Shows** and then select **Check User's Saved Shows** and scroll down to view **Results** like the following:



https://localhost:44395/?handler=CheckUserSavedShows

Shows

5tz9eGgXtNHmq3WVD3EwYx

Get a Show

5tz9eGgXtNHmq3WVD3EwYx

Get Multiple Shows

5tz9eGgXtNHmq3WVD3EwYx

Get a Show's Episodes

5tz9eGgXtNHmq3WVD3EwYx

Save Tracks for User

5tz9eGgXtNHmq3WVD3EwYx

Save Shows for User

5tz9eGgXtNHmq3WVD3EwYx

Check User's Saved Albums

5tz9eGgXtNHmq3WVD3EwYx

Check User's Saved Tracks

5tz9eGgXtNHmq3WVD3EwYx

Check User's Saved Shows

Results

Name
True

Spotify for Developers - Spotify
Get Multiple Episodes
Save Albums for Current User
© 2020 - SpotifyForDevelopers - Privacy

Step 11

You can stop the **web application** in **Visual Studio 2019** by selecting the **Stop debugging** button

Step 12

You can choose to exit **Visual Studio 2019** by selecting the **Close** button in the top right of the **application** as that completes this part of the workshop

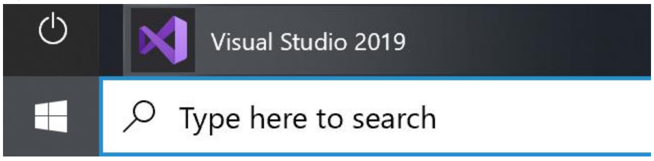
Get User's Saved Albums

Get a list of the albums saved in the current Spotify user's 'Your Music' library.

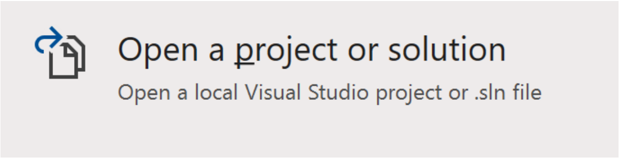
GET https://api.spotify.com/v1/me/albums	
Header	
Authorization	User Token from Spotify Accounts service with **user-library-read** scope
Query Parameter	
limit	Maximum number of objects to return. Default: 20. Minimum: 1. Maximum: 50
offset	Index of the first object to return. Default: 0
market	ISO 3166-1 alpha-2 country code. Provide to apply Track Relinking

Header	**Response**
Success	
Http Status 200 OK	Array of Album Objects accompanied by a Timestamp wrapped in a Paging Object
Error	
Error Code	Error Object

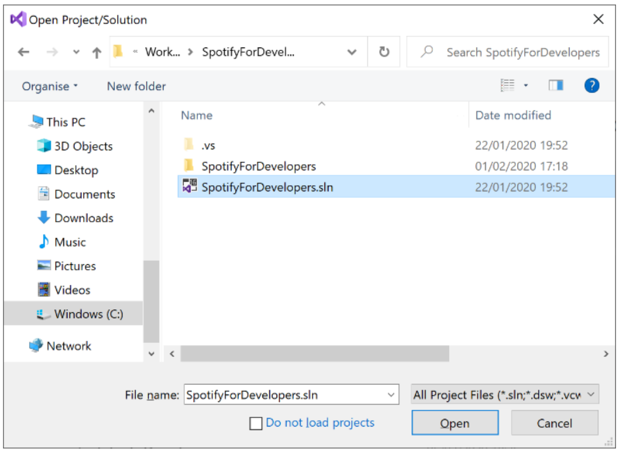
Step 1



If you chose to close **Visual Studio 2019** previously, in **Windows 10** choose **Start**, and then from the **Start Menu** find and select **Visual Studio 2019**


Open a project or solution
Open a local Visual Studio project or .sln file

Once done, from the **Get started** screen for **Visual Studio 2019** select **Open a project or solution**



Name	Date modified
.vs	22/01/2020 19:52
SpotifyForDevelopers	01/02/2020 17:18
SpotifyForDevelopers.sln	22/01/2020 19:52

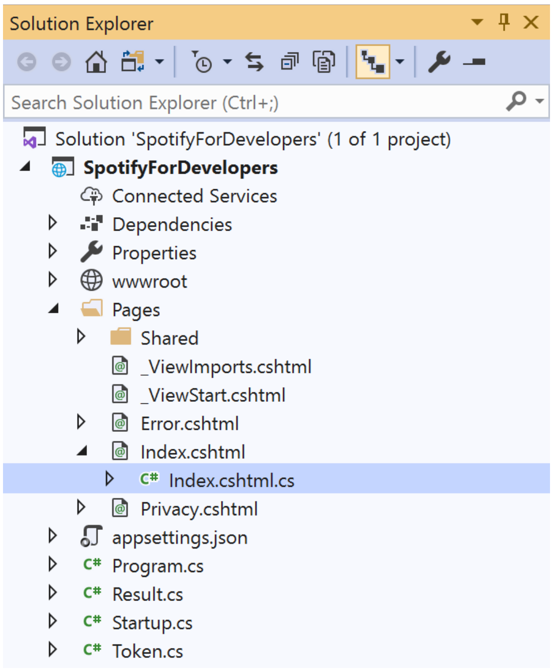
File name: SpotifyForDevelopers.sln

All Project Files (*.sln;*.dsw;*.vcw

□ Do not load projects

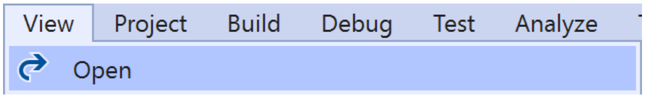
Then locate and select **SpotifyForDevelopers.sln** and select **Open** if you don't have this file already then please follow the previous parts of the workshop including **Getting Started**, **Authorisation Guide**, **Search & Browse**, **Playlists & Artists**, **Albums & Tracks**, **Episodes & Shows**, **Follow**, **Playlists** and **Library**

Step 2



Once opened, in the **Solution Explorer** open the **Pages** section, then open the **Index.cshtml** section and select **Index.cshtml.cs**

Step 3



Then from the **Menu** choose **View** and then **Open**

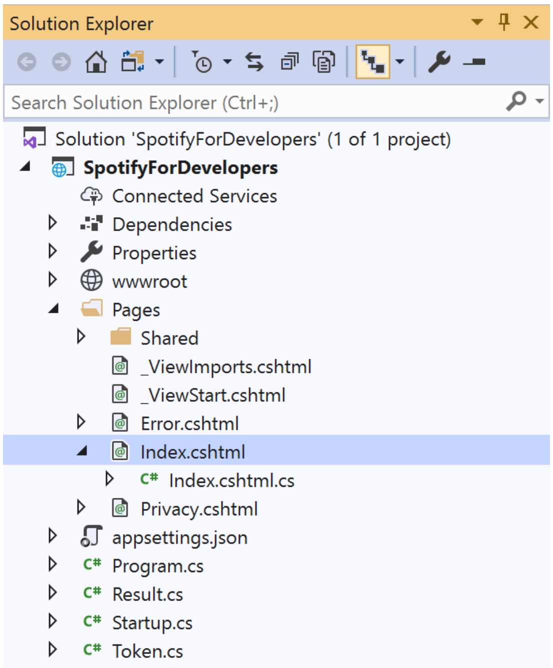
Step 4

In the **Code View** for **Index.cshtml.cs** below the **method** for `public async Task<IActionResult> OnPostCheckUserSavedShowsAsync(...) { ... }` enter the following **method**:

```
public async Task<IActionResult> OnPostGetUserSavedAlbumsAsync()
{
    LoadToken();
    var results = await Api.GetUserSavedAlbumsAsync();
    if (results?.Items != null)
    {
        Results = results.Items.Select(result => new Result()
        {
            Id = result.Album.Id,
            Name = result.Album.Name,
            Image = result?.Album?.Images?.FirstOrDefault()?.Url,
            Inner = new Result()
            {
                Id = result?.Album?.Artists?.FirstOrDefault()?.Id,
                Name = result?.Album?.Artists?.FirstOrDefault()?.Name
            }
        });
    }
    return Page();
}
```

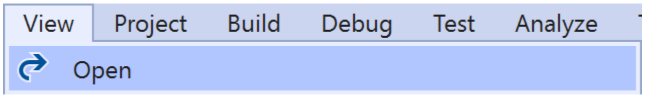
The **method** for `OnPostGetUserSavedAlbumsAsync` is used to get the **albums** in the **Library** and populate the **property** for `Results` accordingly.

Step 5



In the **Solution Explorer** in the **Pages** section select **Index.cshtml**

Step 6



Then from the **Menu** choose **View** and then **Open**

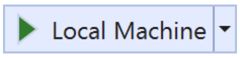
Step 7

Once in the **Code View** for **Index.cshtml** above `<!-- Library -->` enter the following:

```
<li class="list-group-item">
    <form asp-page-handler="GetUserSavedAlbums" method="post">
        <button class="btn btn-primary mb-2">
            Get User's Saved Albums
        </button>
    </form>
</li>
```

This `form` will **post** to the **method** for `GetUserSavedAlbums` and will output to the **Results**.

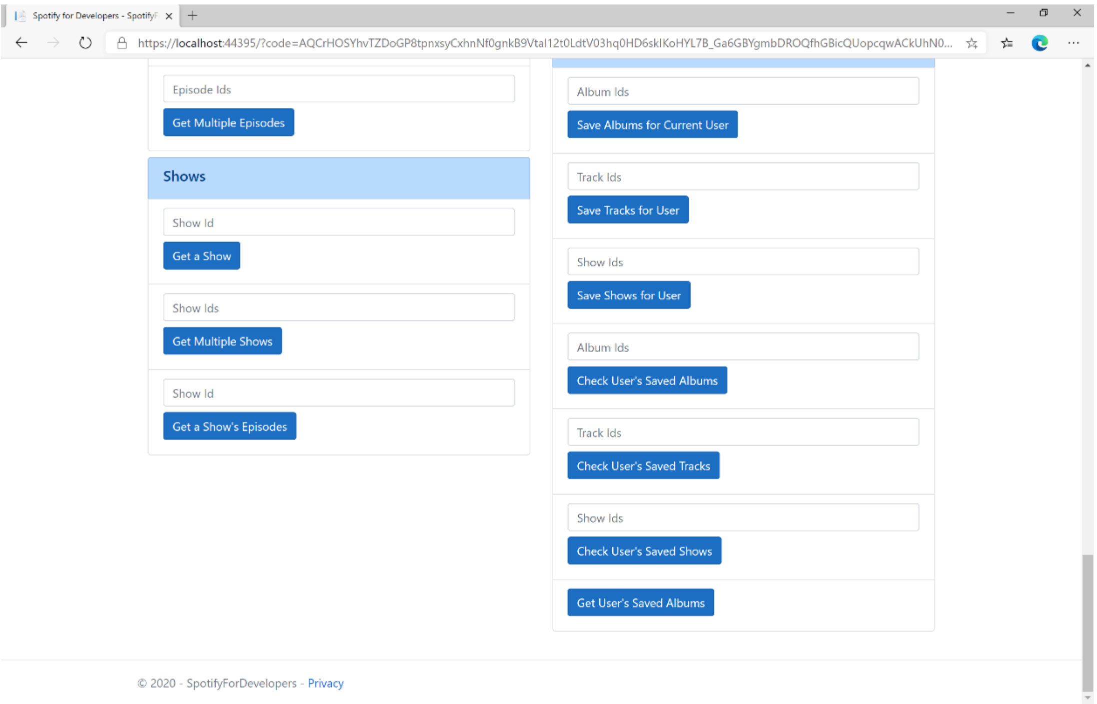
Step 8



Finally, in **Visual Studio 2019** select **IIS Application** to run the **Web Application**

Step 9

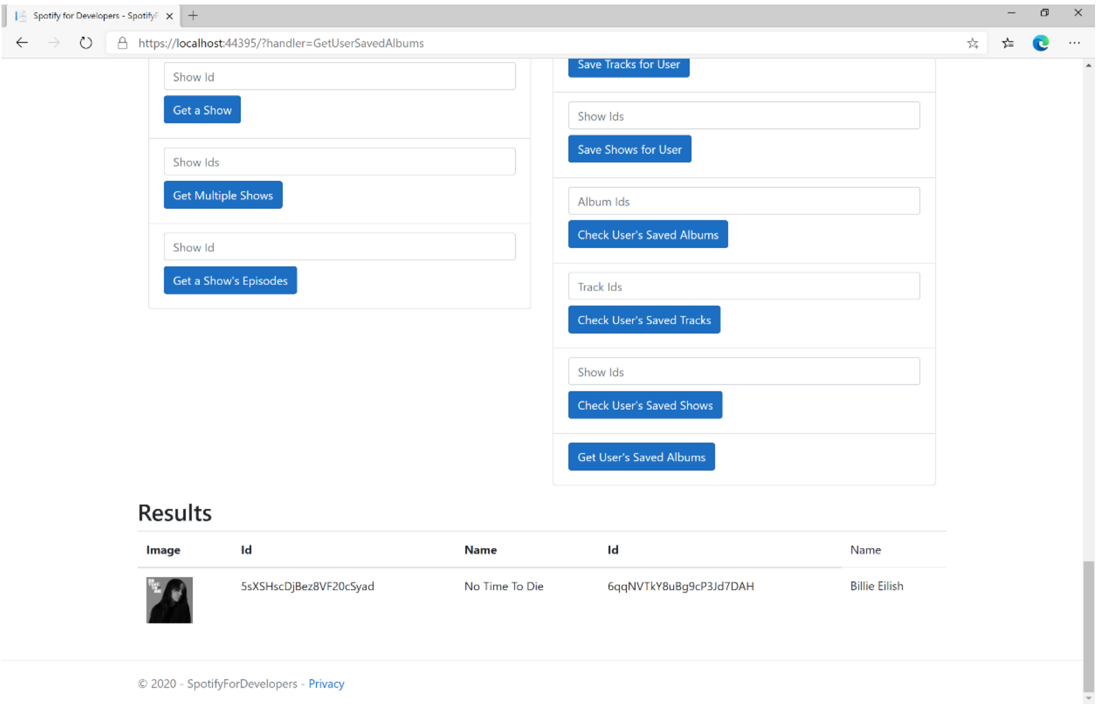
Once the **Web Application** is running and you select **Authorisation Code Flow Login** or **Implicit Grant Flow Login** and scroll down you should see something like the following:



https://localhost:44395/?code=AQCrHOSYhvTZDoGP8tpnxsyCxhnNf0gnkB9Vtal12t0LdtV03hq0HD6skIKoHYL7B_Ga6GBYgmbDROQfhGBicQUopcqwACkUhN0...

Step 10

You can then select **Get User's Saved Albums** and scroll down to view **Results** like the following:



https://localhost:44395/?handler=GetUserSavedAlbums

Results

Image	Id	Name	Id	Name
	5sXSHscDjBez8VF20cSyad	No Time To Die	6qqNVTkY8uBg9cP3Jd7DAH	Billie Eilish

Step 11

■

You can stop the **web application** in **Visual Studio 2019** by selecting the **Stop debugging** button

Step 12

✕

You can choose to exit **Visual Studio 2019** by selecting the **Close** button in the top right of the **application** as that completes this part of the workshop

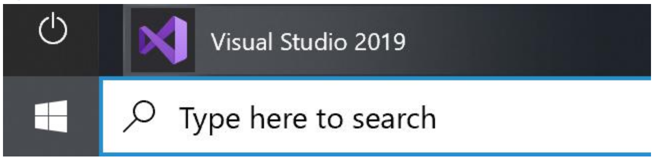
Get User's Saved Tracks

Get a list of the songs saved in the current Spotify user’s ‘Your Music’ library.

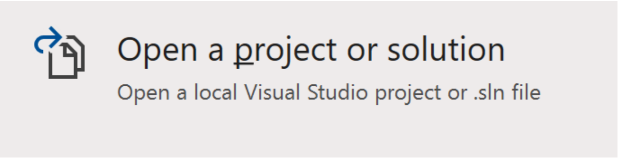
GET https://api.spotify.com/v1/me/tracks	
Header	
Authorization	User Token from Spotify Accounts service with **user-library-read** scope
Query Parameter	
limit	Maximum number of objects to return. Default: 20. Minimum: 1. Maximum: 50
offset	Index of the first object to return. Default: 0
market	ISO 3166-1 alpha-2 country code. Provide to apply Track Relinking

Header	**Response**
Success	
Http Status 200 OK	Array of Saved Track Objects wrapped in a Paging Object
Error	
Error Code	Error Object

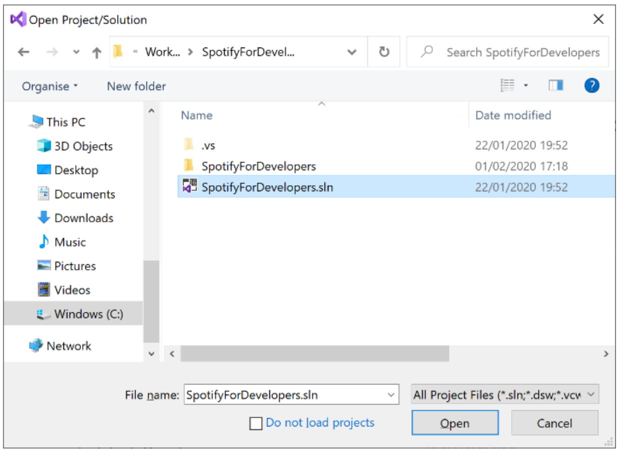
Step 1



If you chose to close **Visual Studio 2019** previously, in **Windows 10** choose **Start**, and then from the **Start Menu** find and select **Visual Studio 2019**

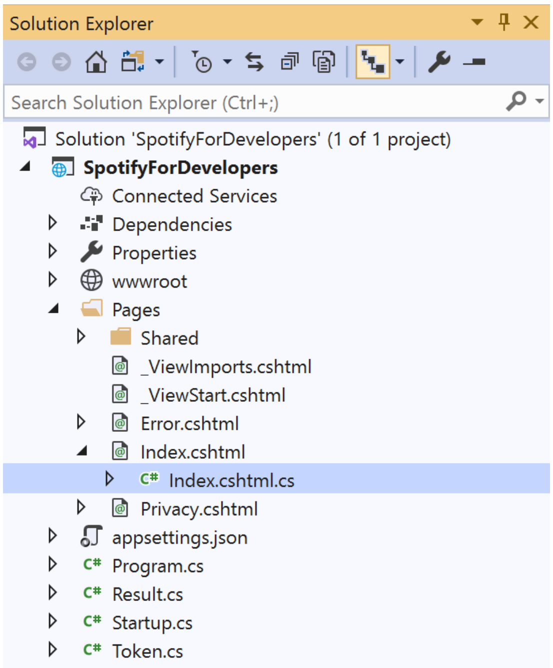

Open a project or solution
Open a local Visual Studio project or .sln file

Once done, from the **Get started** screen for **Visual Studio 2019** select **Open a project or solution**



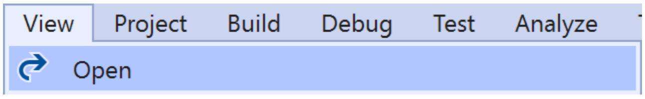
Then locate and select **SpotifyForDevelopers.sln** and select **Open** if you don’t have this file already then please follow the previous parts of the workshop including **Getting Started**, **Authorisation Guide**, **Search & Browse**, **Playlists & Artists**, **Albums & Tracks**, **Episodes & Shows**, **Follow**, **Playlists** and **Library**

Step 2



Once opened, in the **Solution Explorer** open the **Pages** section, then open the **Index.cshtml** section and select **Index.cshtml.cs**

Step 3



Then from the **Menu** choose **View** and then **Open**

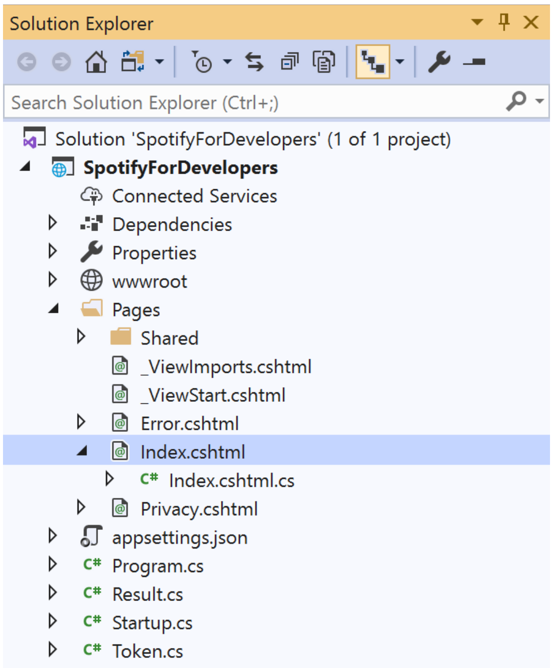
Step 4

In the **Code View** for **Index.cshtml.cs** below the **method** for `public async Task<IActionResult> OnPostGetUserSavedAlbumsAsync() { ... }` enter the following **method**:

```
public async Task<IActionResult> OnPostGetUserSavedTracksAsync()
{
    LoadToken();
    var results = await Api.GetUserSavedTracksAsync();
    if (results?.Items != null)
    {
        Results = results.Items.Select(result => new Result()
        {
            Id = result.Track.Id,
            Name = result.Track.Name,
            Image = result?.Track?.Album?.Images?.FirstOrDefault()?.Url,
            Inner = new Result()
            {
                Id = result?.Track?.Album?.Artists?.FirstOrDefault()?.Id,
                Name = result?.Track?.Album?.Artists?.FirstOrDefault()?.Name
            }
        });
    }
    return Page();
}
```

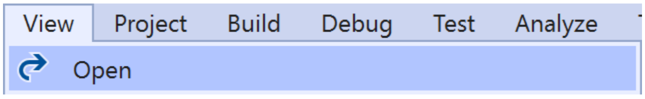
The **method** for `OnPostGetUserSavedTracksAsync` is used to get the **tracks** in the **Library** and populate the **property** for `Results` accordingly.

Step 5



In the **Solution Explorer** in the **Pages** section select **Index.cshtml**

Step 6



Then from the **Menu** choose **View** and then **Open**

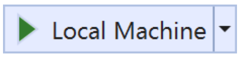
Step 7

Once in the **Code View** for **Index.cshtml** above `<!-- Library -->` enter the following:

```
<li class="list-group-item">
    <form asp-page-handler="GetUserSavedTracks" method="post">
        <button class="btn btn-primary mb-2">
            Get User's Saved Tracks
        </button>
    </form>
</li>
```

This `form` will **post** to the **method** for `GetUserSavedTracks` and will output to the **Results**.

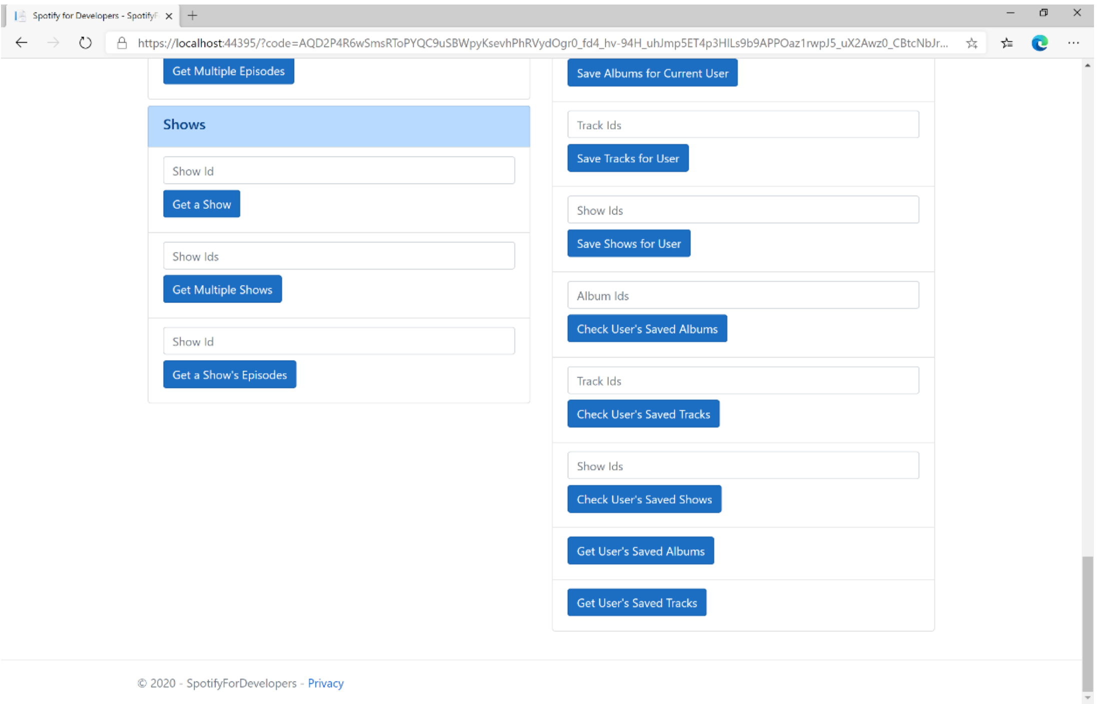
Step 8



Finally, in **Visual Studio 2019** select **IIS Application** to run the **Web Application**

Step 9

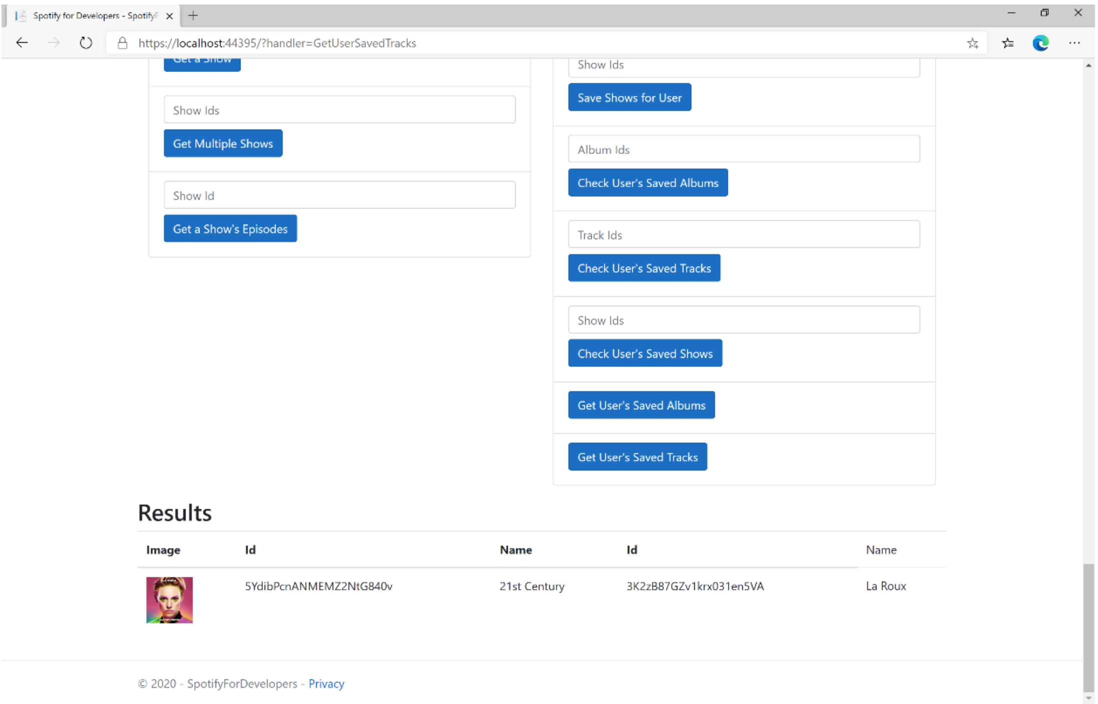
Once the **Web Application** is running and you select **Authorisation Code Flow Login** or **Implicit Grant Flow Login** and scroll down you should see something like the following:



© 2020 - SpotifyForDevelopers - Privacy

Step 10

You can then select **Get User's Saved Tracks** and scroll down to view **Results** like the following:



Results

Image	Id	Name	Id	Name
	5YdibPcnANMEMZ2NtG840v	21st Century	3K2zB87GZv1krx031en5VA	La Roux

© 2020 - SpotifyForDevelopers - Privacy

Step 11

■

You can stop the **web application** in **Visual Studio 2019** by selecting the **Stop debugging** button

Step 12

×

You can choose to exit **Visual Studio 2019** by selecting the **Close** button in the top right of the **application** as that completes this part of the workshop

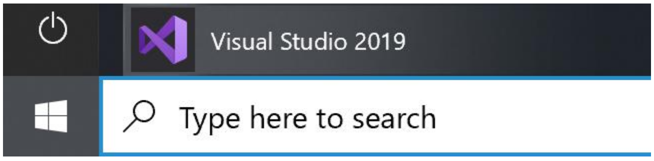
Get User's Saved Shows

Get a list of shows saved in the current Spotify user's library.

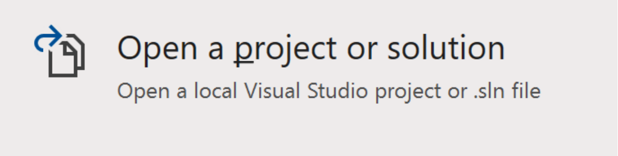
GET https://api.spotify.com/v1/me/shows	
Header	
Authorization	User Token from Spotify Accounts service with **user-library-read** scope
Query Parameter	
limit	Maximum number of shows to return. Default: 20. Minimum: 1. Maximum: 50
offset	Index of the first object to return. Default: 0

Header	**Response**
Success	
Http Status 200 OK	Array of Saved Show Objects wrapped in a Paging Object
Error	
Error Code	Error Object

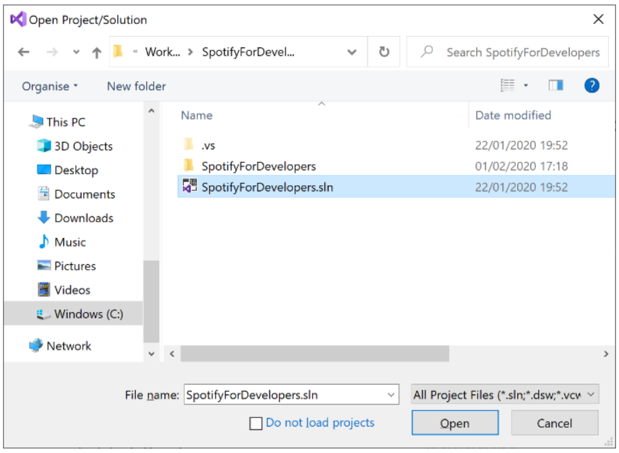
Step 1



If you chose to close **Visual Studio 2019** previously, in **Windows 10** choose **Start**, and then from the **Start Menu** find and select **Visual Studio 2019**


Open a project or solution
Open a local Visual Studio project or .sln file

Once done, from the **Get started** screen for **Visual Studio 2019** select **Open a project or solution**



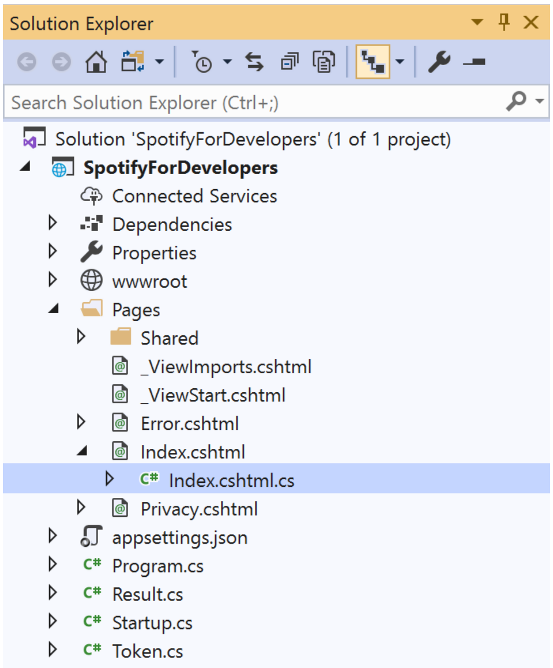
Name	Date modified
.vs	22/01/2020 19:52
SpotifyForDevelopers	01/02/2020 17:18
SpotifyForDevelopers.sln	22/01/2020 19:52

File name: SpotifyForDevelopers.sln

All Project Files (*.sln;*.dsw;*.vcw

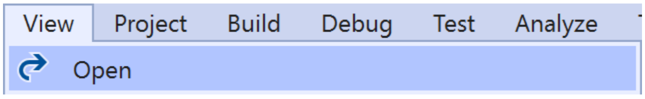
Then locate and select **SpotifyForDevelopers.sln** and select **Open** if you don't have this file already then please follow the previous parts of the workshop including **Getting Started**, **Authorisation Guide**, **Search & Browse**, **Playlists & Artists**, **Albums & Tracks**, **Episodes & Shows**, **Follow**, **Playlists** and **Library**

Step 2



Once opened, in the **Solution Explorer** open the **Pages** section, then open the **Index.cshtml** section and select **Index.cshtml.cs**

Step 3



Then from the **Menu** choose **View** and then **Open**

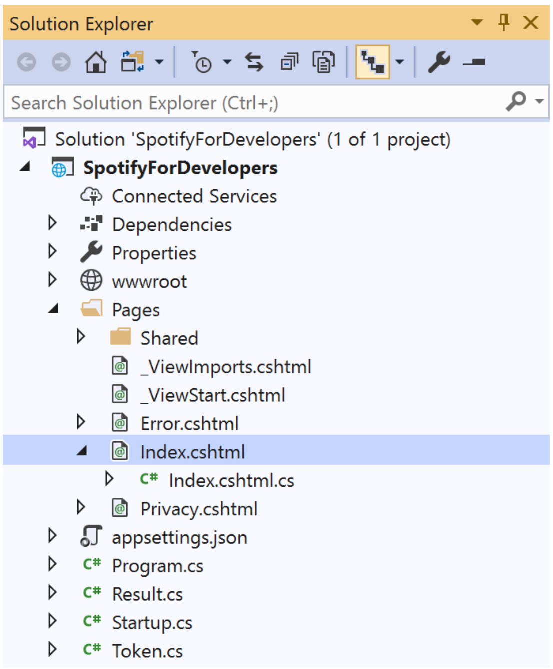
Step 4

In the **Code View** for **Index.cshtml.cs** below the **method** for `public async Task<IActionResult> OnPostGetUserSavedTracksAsync() { ... }` enter the following **method**:

```
public async Task<IActionResult> OnPostGetUserSavedShowsAsync()
{
    LoadToken();
    var results = await Api.GetUserSavedShowsAsync();
    if (results?.Items != null)
    {
        Results = results.Items.Select(result => new Result()
        {
            Id = result.Show.Id,
            Name = result.Show.Name,
            Image = result?.Show?.Images?.FirstOrDefault()?.Url
        });
    }
    return Page();
}
```

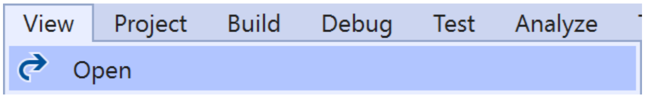
The **method** for `OnPostGetUserSavedShowsAsync` is used to get the **shows** in the **Library** and populate the **property** for `Results` accordingly.

Step 5



In the **Solution Explorer** in the **Pages** section select **Index.cshtml**

Step 6



Then from the **Menu** choose **View** and then **Open**

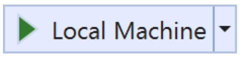
Step 7

Once in the **Code View** for **Index.cshtml** above `<!-- Library -->` enter the following:

```
<li class="list-group-item">
    <form asp-page-handler="GetUserSavedShows" method="post">
        <button class="btn btn-primary mb-2">
            Get User's Saved Shows
        </button>
    </form>
</li>
```

This `form` will **post** to the **method** for `GetUserSavedShows` and will output to the **Results**.

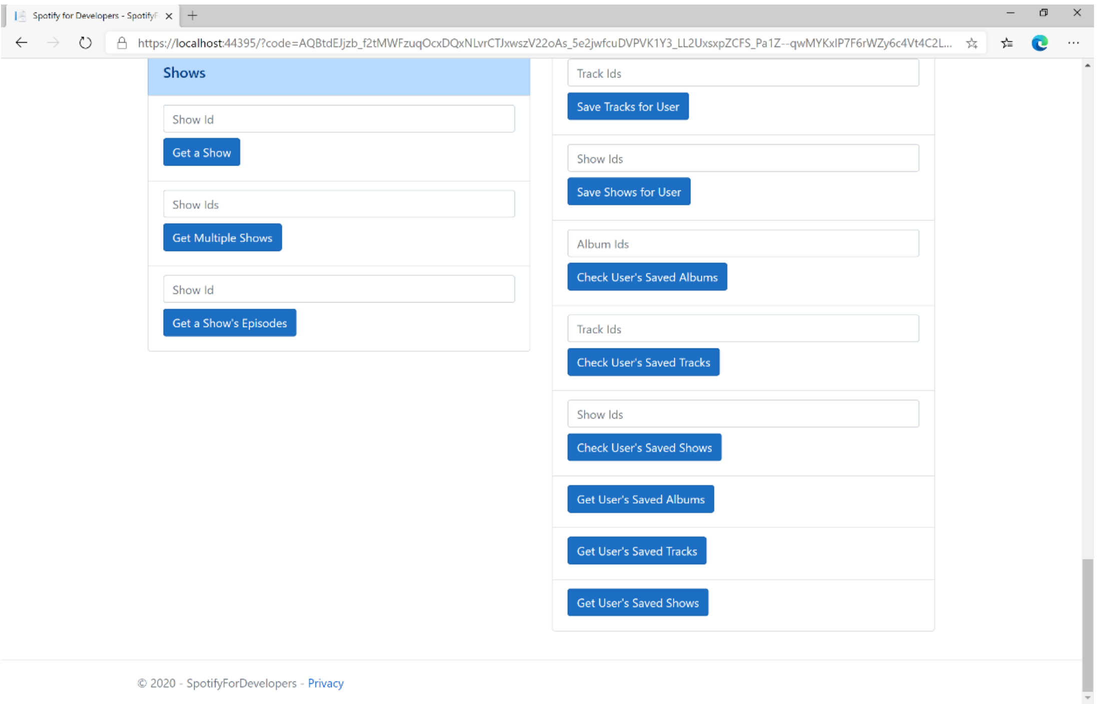
Step 8



Finally, in **Visual Studio 2019** select **IIS Application** to run the **Web Application**

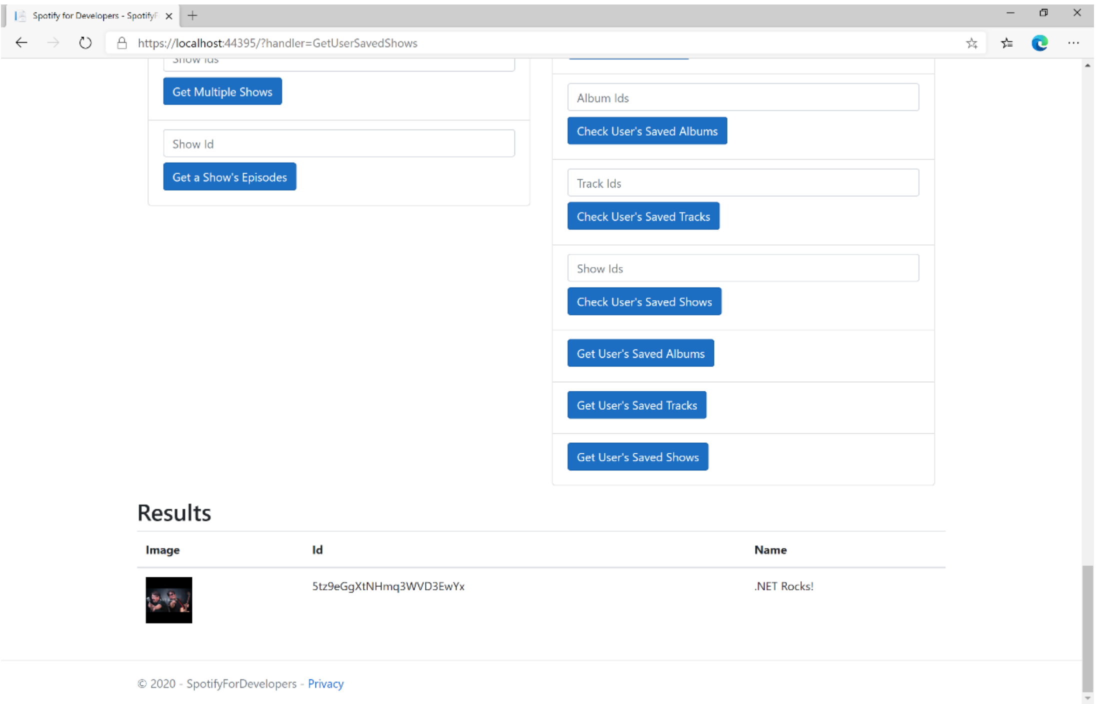
Step 9

Once the **Web Application** is running and you select **Authorisation Code Flow Login** or **Implicit Grant Flow Login** and scroll down you should see something like the following:


Spotify for Developers - Spotify
https://localhost:44395/?code=AQBtdEJjzb_f2tMWFzuqOcxDQxNLvrCTJxwszV22oAs_5e2jwfcuDVPVK1Y3_LL2UxsxpZCFS_Pa1Z--qwMYKxlP7F6rWZy6c4Vt4C2L...
Shows
Show Id
Get a Show
Show Ids
Get Multiple Shows
Show Id
Get a Show's Episodes
Track Ids
Save Tracks for User
Show Ids
Save Shows for User
Album Ids
Check User's Saved Albums
Track Ids
Check User's Saved Tracks
Show Ids
Check User's Saved Shows
Get User's Saved Albums
Get User's Saved Tracks
Get User's Saved Shows
© 2020 - SpotifyForDevelopers - Privacy

Step 10

You can then select **Get User's Saved Shows** and scroll down to view **Results** like the following:



Results

Image	Id	Name
	5tz9eGgXtNHmq3WVD3EwYx	.NET Rocks!

Step 11

■

You can stop the **web application** in **Visual Studio 2019** by selecting the **Stop debugging** button

Step 12

×

You can choose to exit **Visual Studio 2019** by selecting the **Close** button in the top right of the **application** as that completes this part of the workshop

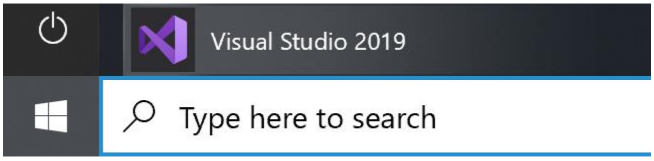
Remove Albums for Current User

Remove one or more albums from the current user's 'Your Music' library.

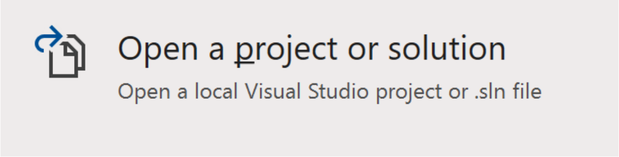
DELETE https://api.spotify.com/v1/me/albums?ids={ids}	
Header	
Authorization	User Token from Spotify Accounts service with **user-library-modify** scope
Query Parameter	
ids	Comma-separated list of the Spotify Ids for the Albums. Maximum: 50 Ids.

Header	**Response**
Success	
Http Status 201 Created	
Error	
HTTP Status 403 Forbidden	Returned when not Authorised
Error Code	Error Object

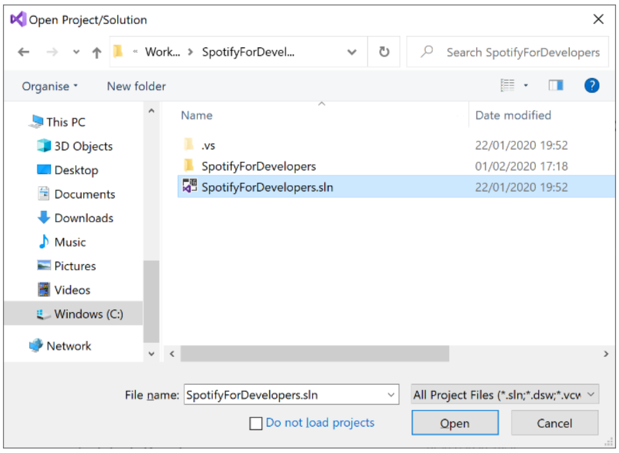
Step 1



If you chose to close **Visual Studio 2019** previously, in **Windows 10** choose **Start**, and then from the **Start Menu** find and select **Visual Studio 2019**


Open a project or solution
Open a local Visual Studio project or .sln file

Once done, from the **Get started** screen for **Visual Studio 2019** select **Open a project or solution**

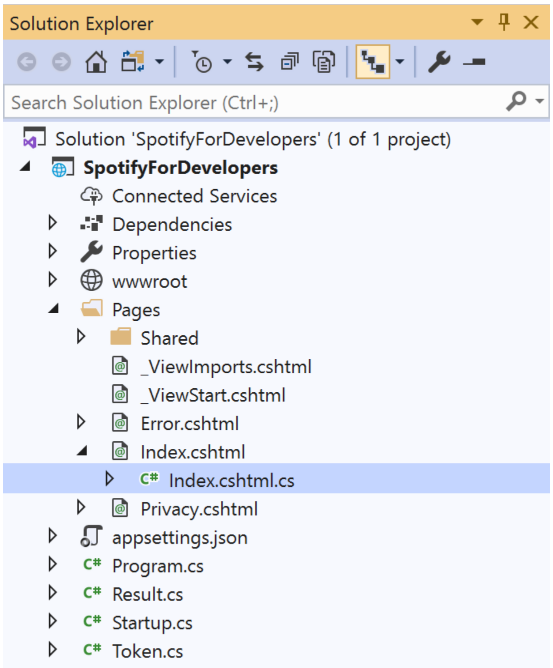


Name	Date modified
.vs	22/01/2020 19:52
SpotifyForDevelopers	01/02/2020 17:18
SpotifyForDevelopers.sln	22/01/2020 19:52

File name: SpotifyForDevelopers.sln

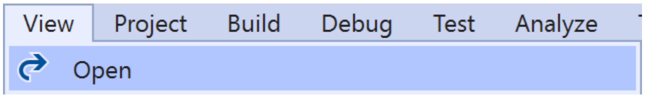
Then locate and select **SpotifyForDevelopers.sln** and select **Open** if you don't have this file already then please follow the previous parts of the workshop including **Getting Started**, **Authorisation Guide**, **Search & Browse**, **Playlists & Artists**, **Albums & Tracks**, **Episodes & Shows**, **Follow**, **Playlists** and **Library**

Step 2



Once opened, in the **Solution Explorer** open the **Pages** section, then open the **Index.cshtml** section and select **Index.cshtml.cs**

Step 3



Then from the **Menu** choose **View** and then **Open**

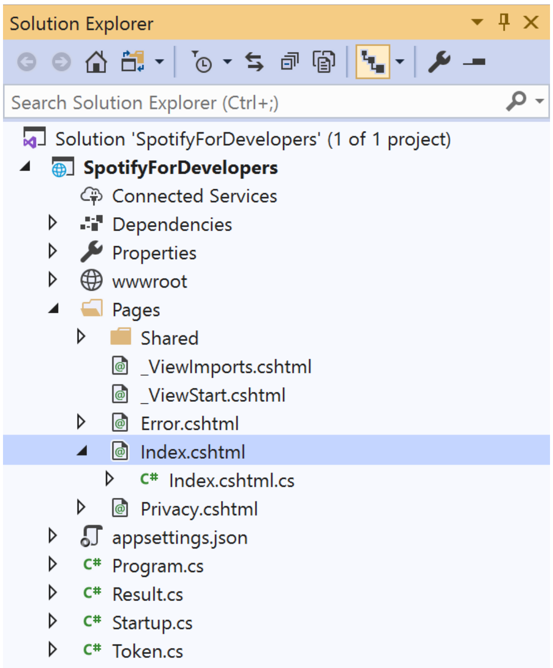
Step 4

In the **Code View** for **Index.cshtml.cs** below the **method** for `public async Task<IActionResult> OnPostGetUserSavedShowsAsync() { ... }` enter the following **method**:

```
public async Task<IActionResult> OnPostRemoveUserAlbumsAsync(string value)
{
    LoadToken();
    var values = value.Split(",").ToList();
    var result = await Api.RemoveUserAlbumsAsync(values);
    if (result != null)
    {
        Results = new List<Result>() { new Result()
        {
            Id = result.Code.ToString(),
            Name = result.Success.ToString()
        }};
    }
    return Page();
}
```

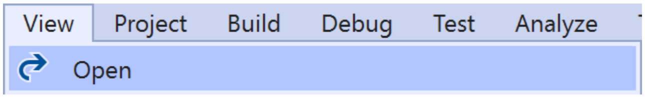
The **method** for `OnPostRemoveUserAlbumsAsync` **removes** the **saved albums** from the **Library** with `Value` of the **Album Ids** of the **album** and populate the **property** for `Results` of the **success** of the operation accordingly.

Step 5



In the **Solution Explorer** in the **Pages** section select **Index.cshtml**

Step 6



Then from the **Menu** choose **View** and then **Open**

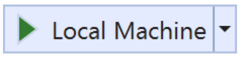
Step 7

Once in the **Code View** for **Index.cshtml** above `<!-- Library -->` enter the following:

```
<li class="list-group-item">
    <form asp-page-handler="RemoveUserAlbums" method="post">
        <input asp-for="Value" placeholder="Album Ids" class="form-control mb-2" />
        <button class="btn btn-primary mb-2">
            Remove Albums for Current User
        </button>
    </form>
</li>
```

This `form` will **post** to the **method** for `RemoveUserAlbums` with the `Value` of the **Album Ids** and will output to the **Results**.

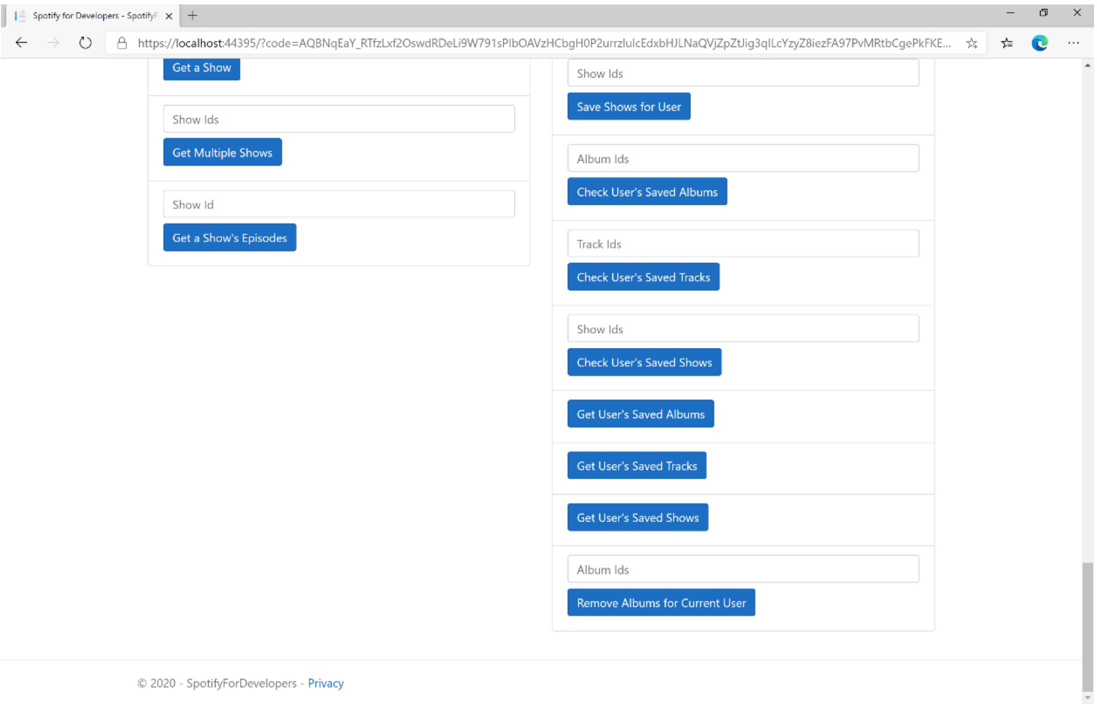
Step 8



Finally, in **Visual Studio 2019** select **IIS Application** to run the **Web Application**

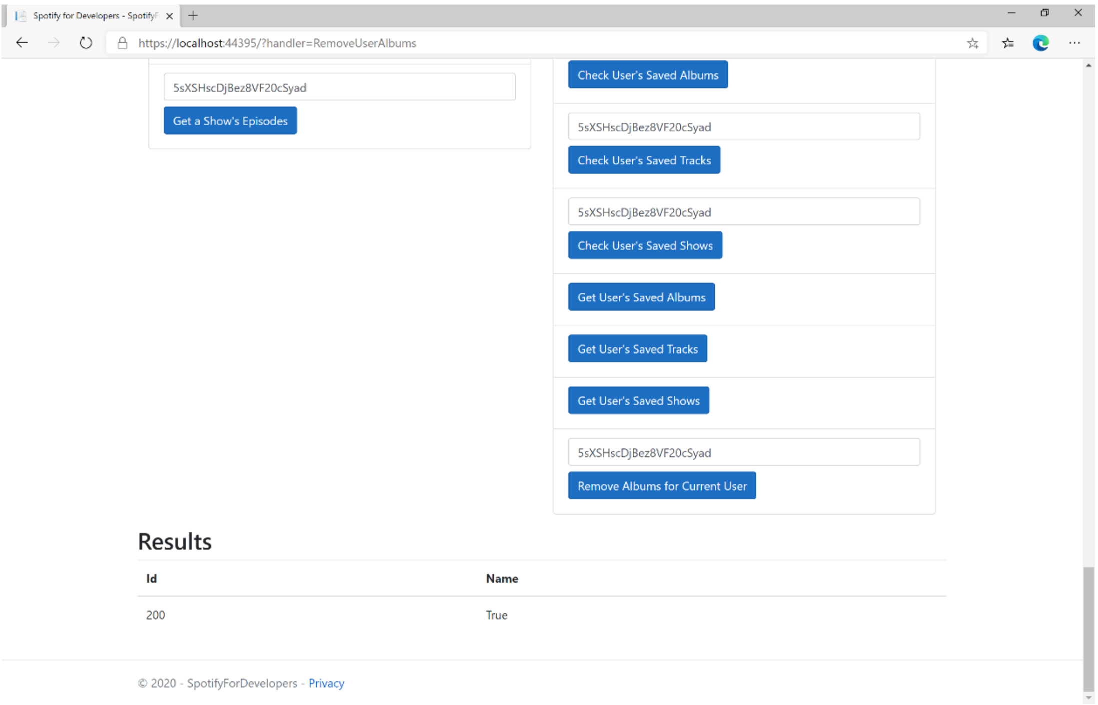
Step 9

Once the **Web Application** is running and you select **Authorisation Code Flow Login** or **Implicit Grant Flow Login** and scroll down you should see something like the following:


Spotify for Developers - Spotify
https://localhost:44395/?code=AQBNqEaY_RTfzLxf2OswdRDeLi9W791sPlbOAVzHCbgH0P2urrzlulcEdxbHJLNaQVjZpZtJig3qlLcYzyZ8iezFA97PvMRtbCgePkFKE...
Get a Show
Show Ids
Get Multiple Shows
Show Id
Get a Show's Episodes
Show Ids
Save Shows for User
Album Ids
Check User's Saved Albums
Track Ids
Check User's Saved Tracks
Show Ids
Check User's Saved Shows
Get User's Saved Albums
Get User's Saved Tracks
Get User's Saved Shows
Album Ids
Remove Albums for Current User
© 2020 - SpotifyForDevelopers - Privacy

Step 10

You can then enter some **Album Ids** from **Get All New Releases** or copy the same **Album Ids** used in **Save Albums for Current User** into **Album Ids** above **Remove Albums for Current User** and then select **Remove Albums for Current User** and scroll down to view **Results** like the following:


Spotify for Developers - Spotify
https://localhost:44395/?handler=RemoveUserAlbums
Get a Show's Episodes
Check User's Saved Albums
Check User's Saved Tracks
Check User's Saved Shows
Get User's Saved Albums
Get User's Saved Tracks
Get User's Saved Shows
Remove Albums for Current User
© 2020 - SpotifyForDevelopers - Privacy

5sXSHscDjBez8VF20cSyad

5sXSHscDjBez8VF20cSyad

5sXSHscDjBez8VF20cSyad

5sXSHscDjBez8VF20cSyad

Results

Id	Name
200	True

Step 11

■

You can stop the **web application** in **Visual Studio 2019** by selecting the **Stop debugging** button

Step 12

×

You can choose to exit **Visual Studio 2019** by selecting the **Close** button in the top right of the **application** as that completes this part of the workshop

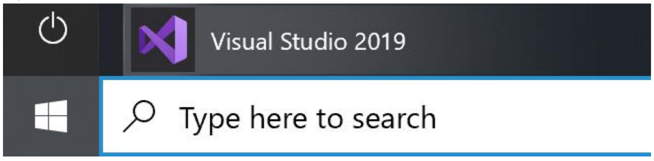
Remove User's Saved Tracks

Remove one or more tracks from the current user's 'Your Music' library.

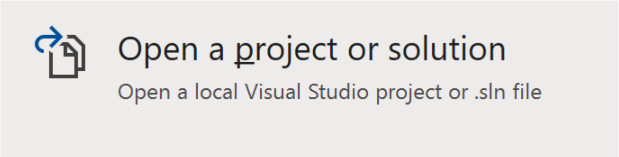
DELETE https://api.spotify.com/v1/me/tracks	
Header	
Authorization	User Token from Spotify Accounts service with **user-library-modify** scope
Query Parameter	
ids	Comma-separated list of the Spotify Ids for the Tracks. Maximum: 50 Ids.

Header	Response
Success	
Http Status 200 OK	
Error	
HTTP Status 403 Forbidden	Returned when not Authorised
Error Code	Error Object

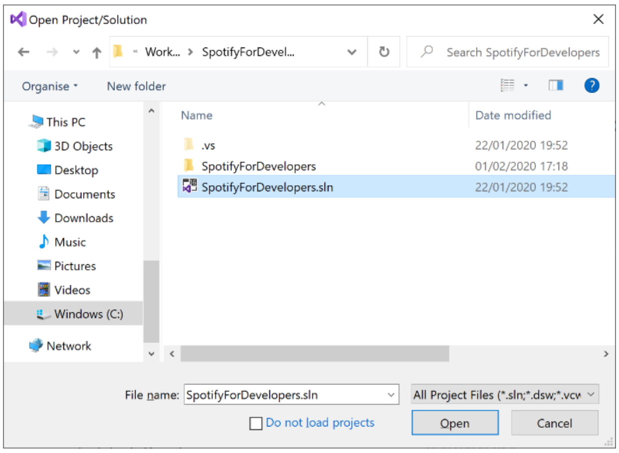
Step 1



If you chose to close **Visual Studio 2019** previously, in **Windows 10** choose **Start**, and then from the **Start Menu** find and select **Visual Studio 2019**


Open a project or solution
Open a local Visual Studio project or .sln file

Once done, from the **Get started** screen for **Visual Studio 2019** select **Open a project or solution**

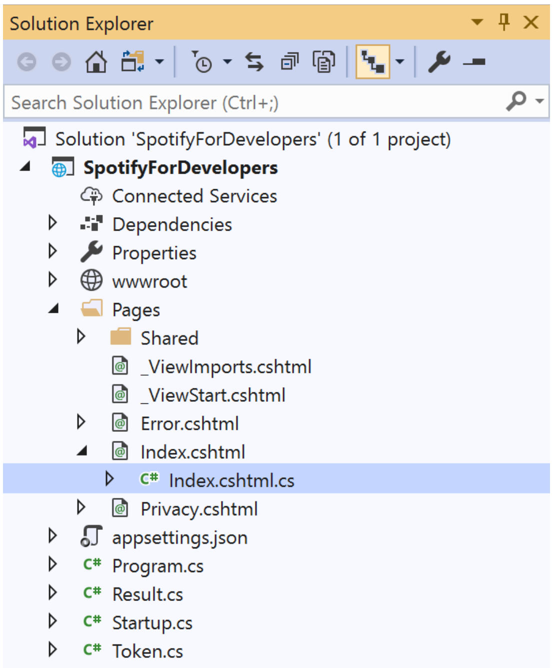


Name	Date modified
.vs	22/01/2020 19:52
SpotifyForDevelopers	01/02/2020 17:18
SpotifyForDevelopers.sln	22/01/2020 19:52

File name: SpotifyForDevelopers.sln
All Project Files (*.sln;*.dsw;*.vcw

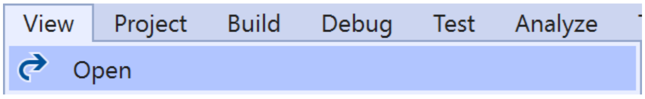
Then locate and select **SpotifyForDevelopers.sln** and select **Open** if you don't have this file already then please follow the previous parts of the workshop including **Getting Started**, **Authorisation Guide**, **Search & Browse**, **Playlists & Artists**, **Albums & Tracks**, **Episodes & Shows**, **Follow**, **Playlists** and **Library**

Step 2



Once opened, in the **Solution Explorer** open the **Pages** section, then open the **Index.cshtml** section and select **Index.cshtml.cs**

Step 3



Then from the **Menu** choose **View** and then **Open**

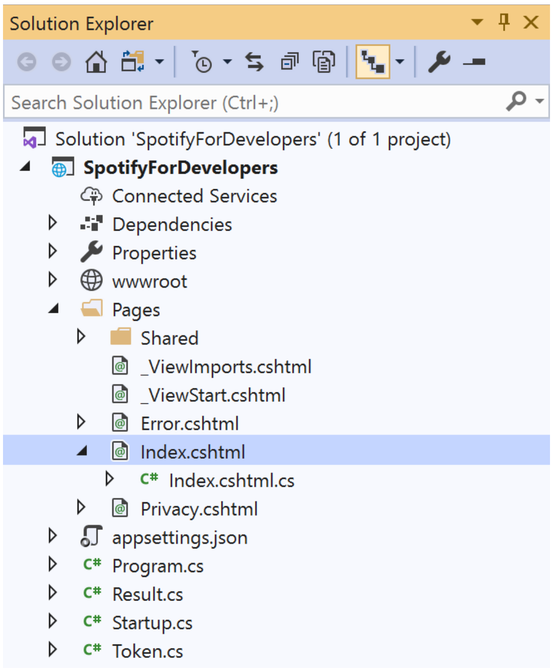
Step 4

In the **Code View** for **Index.cshtml.cs** below the **method** for `public async Task<IActionResult> OnPostRemoveUserAlbumsAsync(...) { ... }` enter the following **method**:

```
public async Task<IActionResult> OnPostRemoveUserTracksAsync(string value)
{
    LoadToken();
    var values = value.Split(",").ToList();
    var result = await Api.RemoveUserTracksAsync(values);
    if (result != null)
    {
        Results = new List<Result>() { new Result()
        {
            Id = result.Code.ToString(),
            Name = result.Success.ToString()
        }};
    }
    return Page();
}
```

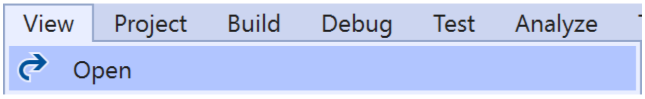
The **method** for `OnPostRemoveUserTracksAsync` **removes** the **saved track** from the **Library** with `Value` of the **Track Ids** of the **track** and populate the **property** for `Results` of the **success** of the operation accordingly.

Step 5



In the **Solution Explorer** in the **Pages** section select **Index.cshtml**

Step 6



Then from the **Menu** choose **View** and then **Open**

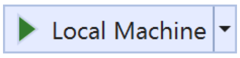
Step 7

Once in the **Code View** for **Index.cshtml** above `<!-- Library -->` enter the following:

```
<li class="list-group-item">
    <form asp-page-handler="RemoveUserTracks" method="post">
        <input asp-for="Value" placeholder="Track Ids" class="form-control mb-2" />
        <button class="btn btn-primary mb-2">
            Remove User's Saved Tracks
        </button>
    </form>
</li>
```

This `form` will **post** to the **method** for `RemoveUserTracks` with the `Value` of the **Track Ids** and will output to the **Results**.

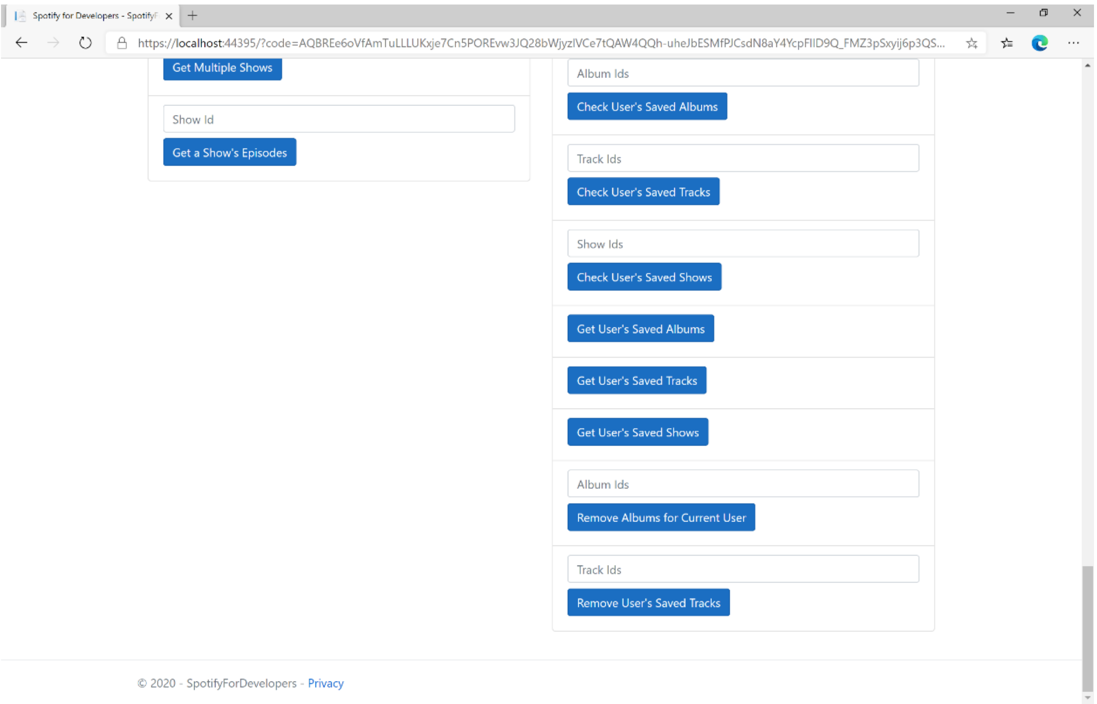
Step 8



Finally, in **Visual Studio 2019** select **IIS Application** to run the **Web Application**

Step 9

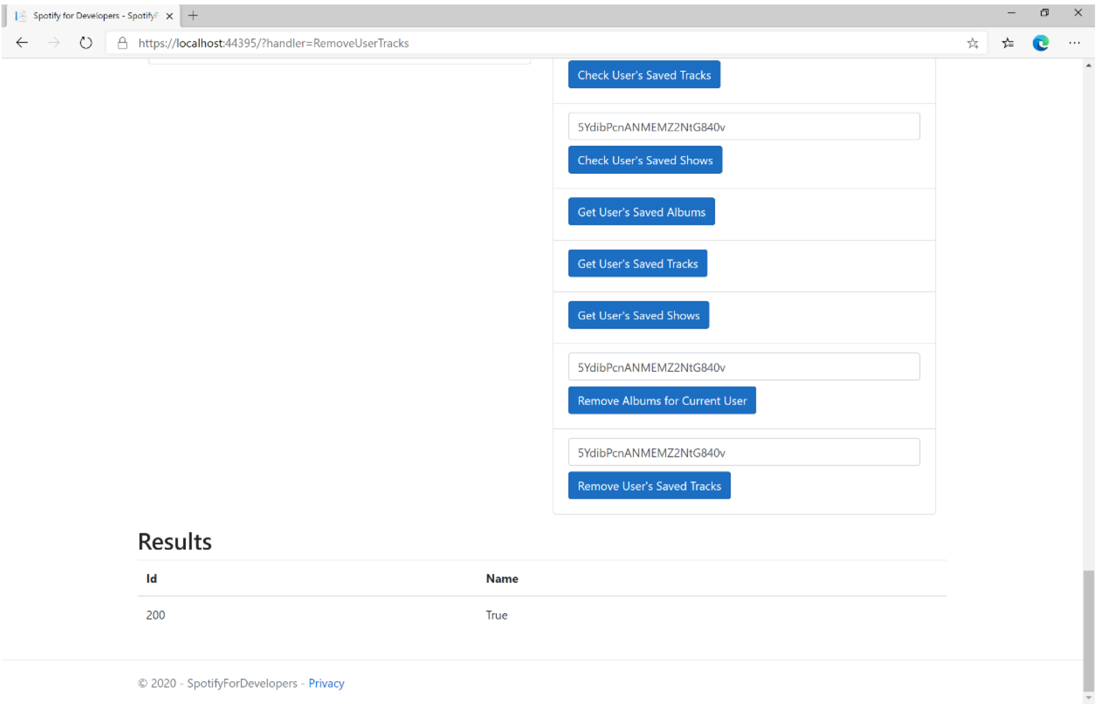
Once the **Web Application** is running and you select **Authorisation Code Flow Login** or **Implicit Grant Flow Login** and scroll down you should see something like the following:



Spotify for Developers - SpotifyF
https://localhost:44395/?code=AQBREe6oVfAmTuLLLUKxje7Cn5POREvw3JQ28bWjyzlVCe7tQAW4QQh-uheJbESMfPJCsdN8aY4YcpFIID9Q_FMZ3pSxyij6p3QS...
Get Multiple Shows
Show Id
Get a Show's Episodes
Album Ids
Check User's Saved Albums
Track Ids
Check User's Saved Tracks
Show Ids
Check User's Saved Shows
Get User's Saved Albums
Get User's Saved Tracks
Get User's Saved Shows
Album Ids
Remove Albums for Current User
Track Ids
Remove User's Saved Tracks
© 2020 - SpotifyForDevelopers - Privacy

Step 10

You can then enter some **Track Ids** from an **Album Id** from **Get All New Releases** using **Get an Album's Tracks** or copy the same **Track Ids** used in **Save Tracks for User** in **Track Ids** above **Remove User's Saved Tracks** and then select **Remove User's Saved Tracks** and scroll down to view **Results** like the following:



Spotify for Developers - SpotifyF
https://localhost:44395/?handler=RemoveUserTracks
Check User's Saved Tracks
5YdibPcnANMEMZ2NtG840v
Check User's Saved Shows
Get User's Saved Albums
Get User's Saved Tracks
Get User's Saved Shows
5YdibPcnANMEMZ2NtG840v
Remove Albums for Current User
5YdibPcnANMEMZ2NtG840v
Remove User's Saved Tracks

Results

Id	Name
200	True

© 2020 - SpotifyForDevelopers - Privacy

Step 11

You can stop the **web application** in **Visual Studio 2019** by selecting the **Stop debugging** button

Step 12

You can choose to exit **Visual Studio 2019** by selecting the **Close** button in the top right of the **application** as that completes this part of the workshop

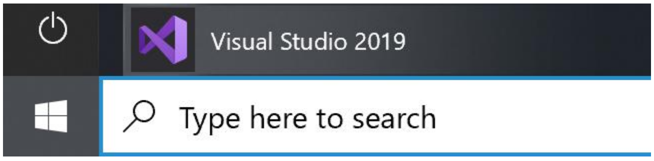
Remove User's Saved Shows

Delete one or more shows from current Spotify user's library.

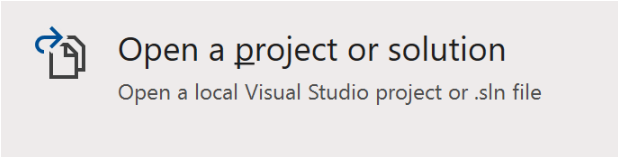
DELETE https://api.spotify.com/v1/me/shows	
Header	
Authorization	User Token from Spotify Accounts service with **user-library-modify** scope
Query Parameter	
ids	Comma-separated list of Spotify Ids for the shows to be deleted from the user's library
market	An ISO 3166-1 alpha-2 country code. If a country code is specified, only shows that are available in that market will be removed. If a valid user access token is specified in the request header, the country associated with the user account will take priority over this parameter. If neither market nor user country are provided, the content is considered unavailable for the client. Users can view the country that is associated with their account in the account settings

Header	Response
Success	
Http Status 200 OK	
Error	
HTTP Status 403 Forbidden	Returned when not Authorised
Error Code	Error Object

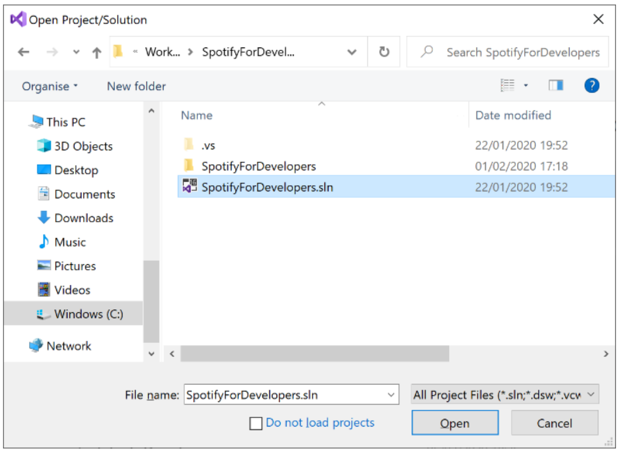
Step 1



If you chose to close **Visual Studio 2019** previously, in **Windows 10** choose **Start**, and then from the **Start Menu** find and select **Visual Studio 2019**


Open a project or solution
Open a local Visual Studio project or .sln file

Once done, from the **Get started** screen for **Visual Studio 2019** select **Open a project or solution**

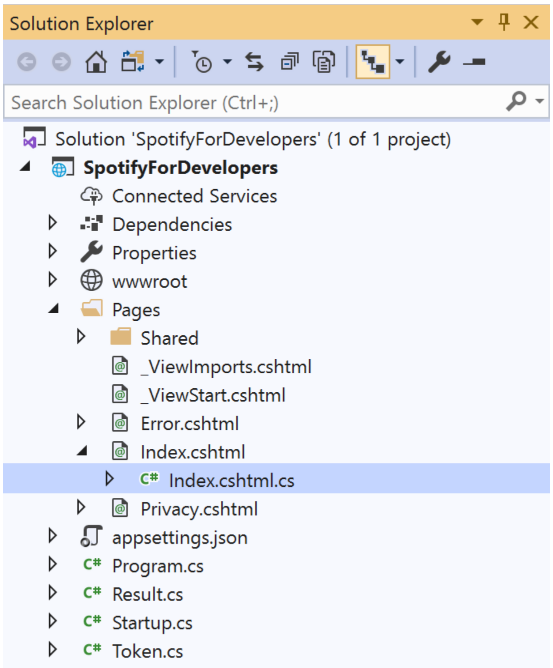


Name	Date modified
.vs	22/01/2020 19:52
SpotifyForDevelopers	01/02/2020 17:18
SpotifyForDevelopers.sln	22/01/2020 19:52

File name: SpotifyForDevelopers.sln

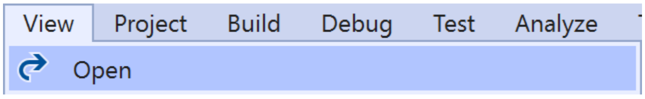
Then locate and select **SpotifyForDevelopers.sln** and select **Open** if you don't have this file already then please follow the previous parts of the workshop including **Getting Started**, **Authorisation Guide**, **Search & Browse**, **Playlists & Artists**, **Albums & Tracks**, **Episodes & Shows**, **Follow**, **Playlists** and **Library**

Step 2



Once opened, in the **Solution Explorer** open the **Pages** section, then open the **Index.cshtml** section and select **Index.cshtml.cs**

Step 3



Then from the **Menu** choose **View** and then **Open**

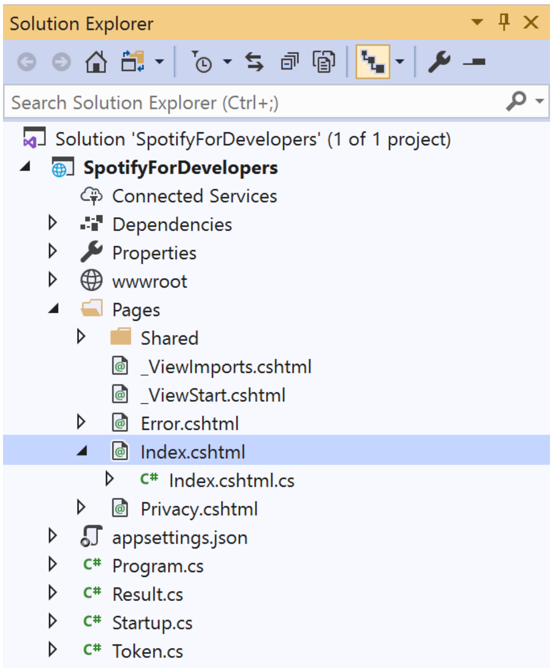
Step 4

In the **Code View** for **Index.cshtml.cs** below the **method** for `public async Task<IActionResult> OnPostRemoveUserTracksAsync(...) { ... }` enter the following **method**:

```
public async Task<IActionResult> OnPostRemoveUserShowsAsync(string value)
{
    LoadToken();
    var values = value.Split(",").ToList();
    var result = await Api.RemoveUserShowsAsync(values);
    if (result != null)
    {
        Results = new List<Result>() { new Result()
        {
            Id = result.Code.ToString(),
            Name = result.Success.ToString()
        }};
    }
    return Page();
}
```

The **method** for `OnPostRemoveUserShowsAsync` **removes** the **saved show** from the **Library** with `Value` of the **Show Ids** of the **show** and populate the **property** for `Results` of the **success** of the operation accordingly.

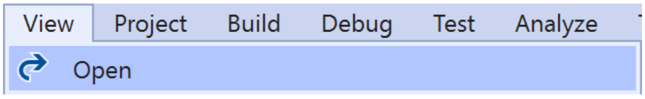
Step 5



```
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'SpotifyForDevelopers' (1 of 1 project)
  SpotifyForDevelopers
    Connected Services
    Dependencies
    Properties
    wwwroot
    Pages
      Shared
      _ViewImports.cshtml
      _ViewStart.cshtml
      Error.cshtml
      Index.cshtml
        Index.cshtml.cs
      Privacy.cshtml
    appsettings.json
    Program.cs
    Result.cs
    Startup.cs
    Token.cs
```

In the **Solution Explorer** in the **Pages** section select **Index.cshtml**

Step 6



Then from the **Menu** choose **View** and then **Open**

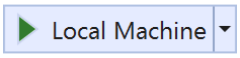
Step 7

Once in the **Code View** for **Index.cshtml** above `<!-- Library -->` enter the following:

```
<li class="list-group-item">
    <form asp-page-handler="RemoveUserShows" method="post">
        <input asp-for="Value" placeholder="Show Ids" class="form-control mb-2" />
        <button class="btn btn-primary mb-2">
            Remove User's Saved Shows
        </button>
    </form>
</li>
```

This `form` will **post** to the **method** for `RemoveUserShows` with the `Value` of the **Show Ids** and will output to the **Results**.

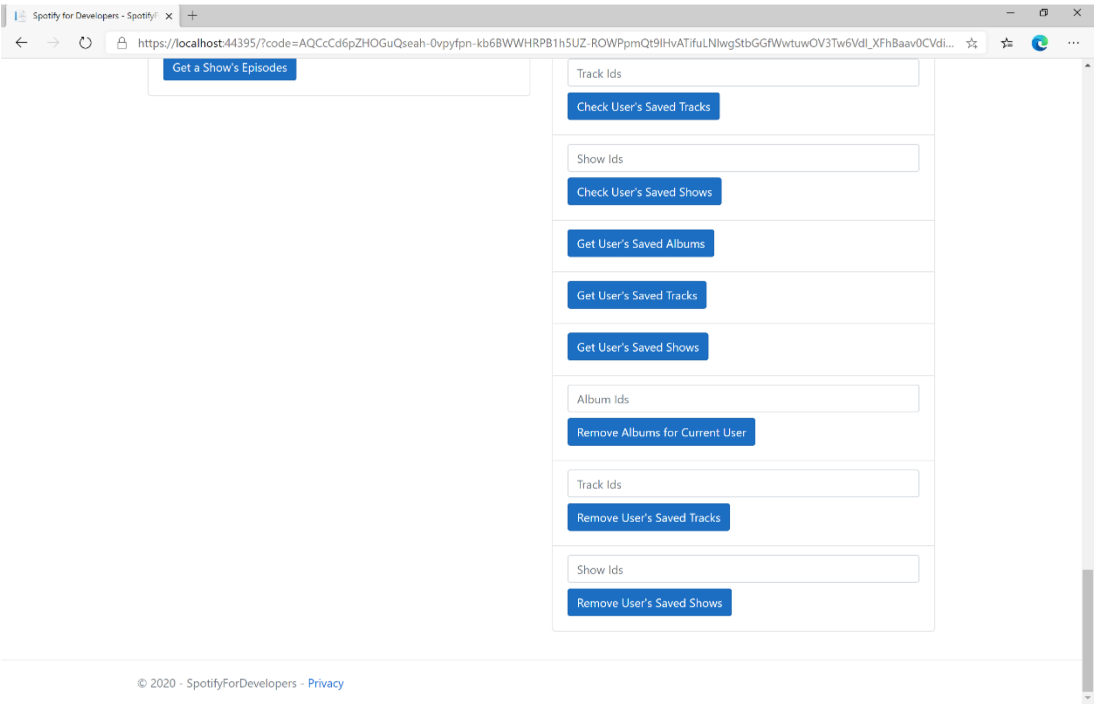
Step 8



Finally, in **Visual Studio 2019** select **IIS Application** to run the **Web Application**

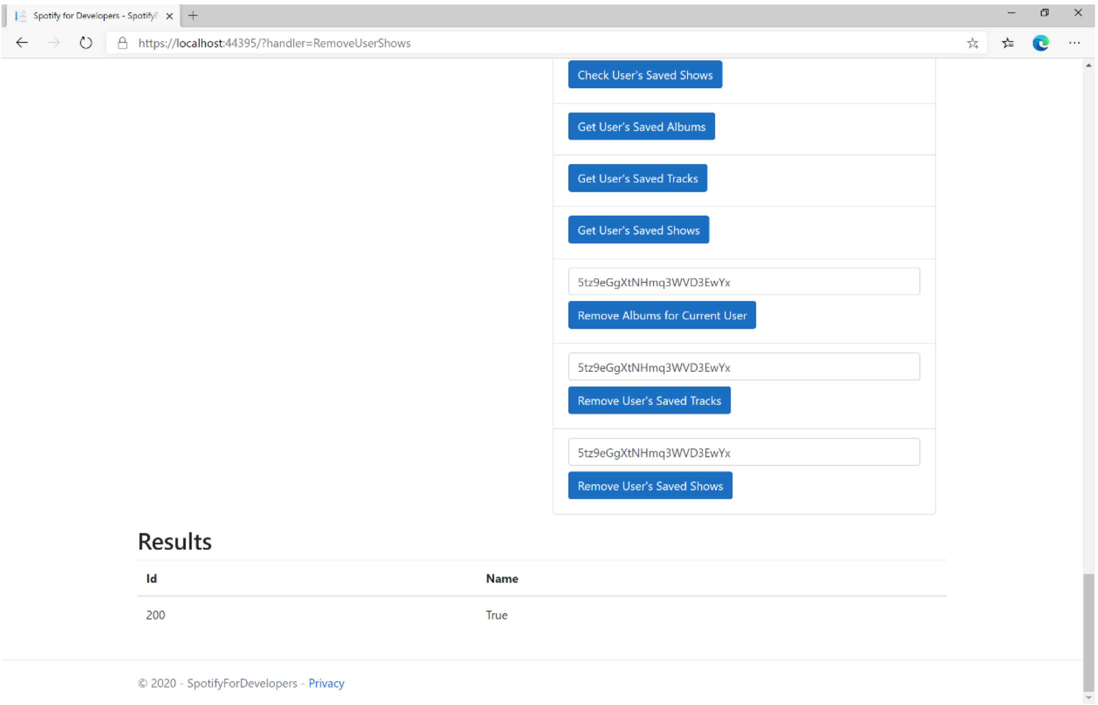
Step 9

Once the **Web Application** is running and you select **Authorisation Code Flow Login** or **Implicit Grant Flow Login** and scroll down you should see something like the following:


Spotify for Developers - Spotify
https://localhost:44395/?code=AQCcCd6pZHOGuQseah-0vpyfpn-kb6BWWHRPB1h5UZ-ROWPpmQt9lHvATifuLNlwgStbGGfWwtuwOV3Tw6Vdl_XFhBaav0CVdi...
Get a Show's Episodes
Track Ids
Check User's Saved Tracks
Show Ids
Check User's Saved Shows
Get User's Saved Albums
Get User's Saved Tracks
Get User's Saved Shows
Album Ids
Remove Albums for Current User
Track Ids
Remove User's Saved Tracks
Show Ids
Remove User's Saved Shows
© 2020 - SpotifyForDevelopers - Privacy

Step 10

You can get a **Show Id** from **Search For An Item** using **Shows** or copy the same **Show Ids** used in **Save Shows for User** in **Show Ids** above **Remove User's Saved Shows** and then select **Remove User's Saved Shows** and scroll down to view **Results** like the following:


Spotify for Developers - Spotify
https://localhost:44395/?handler=RemoveUserShows
Check User's Saved Shows
Get User's Saved Albums
Get User's Saved Tracks
Get User's Saved Shows
Remove Albums for Current User
Remove User's Saved Tracks
Remove User's Saved Shows
© 2020 - SpotifyForDevelopers - Privacy

5tz9eGgXtNHmq3WVD3EwYx

5tz9eGgXtNHmq3WVD3EwYx

5tz9eGgXtNHmq3WVD3EwYx

Results

Id	Name
200	True

Step 11

You can stop the **web application** in **Visual Studio 2019** by selecting the **Stop debugging** button

Step 12

You can exit **Visual Studio 2019** by selecting the **Close** button in the top right of the **application** as that completes this part of the workshop

Player

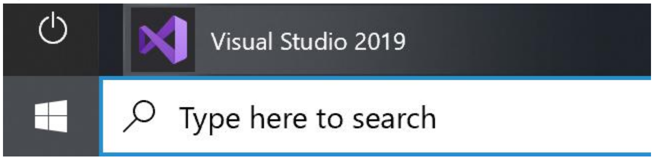
Add an Item to the User's Playback Queue

Add an item to the end of the user's current playback queue.

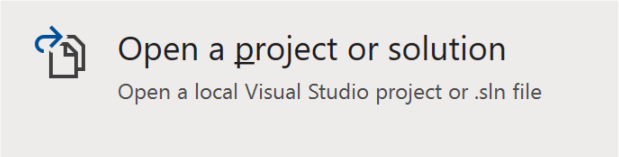
POST https://api.spotify.com/v1/me/player/queue	
Header	
Authorization	User Token from Spotify Accounts service with **user-modify-playback-state** scope
Query Parameter	
uri	Uri of the item to add to the queue. Must be a track or an episode uri
device_id	Device Id to target, if not supplied the user's current active device is the target

Header	Response
Success	
Http Status 204 No Content	
Error	
HTTP Status 404 Not Found	NO_ACTIVE_DEVICE - Device was not found
HTTP Status 403 Forbidden	PREMIUM_REQUIRED - Returned when account non-premium
Error Code	Error Object

Step 1



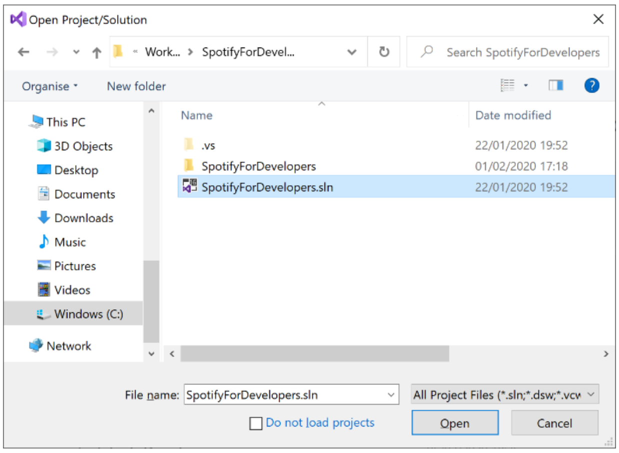
In **Windows 10** choose **Start**, and then from the **Start Menu** find and select **Visual Studio 2019**



Open a project or solution

Open a local Visual Studio project or .sln file

Once done, from the **Get started** screen for **Visual Studio 2019** select **Open a project or solution**



Name	Date modified
.vs	22/01/2020 19:52
SpotifyForDevelopers	01/02/2020 17:18
SpotifyForDevelopers.sln	22/01/2020 19:52

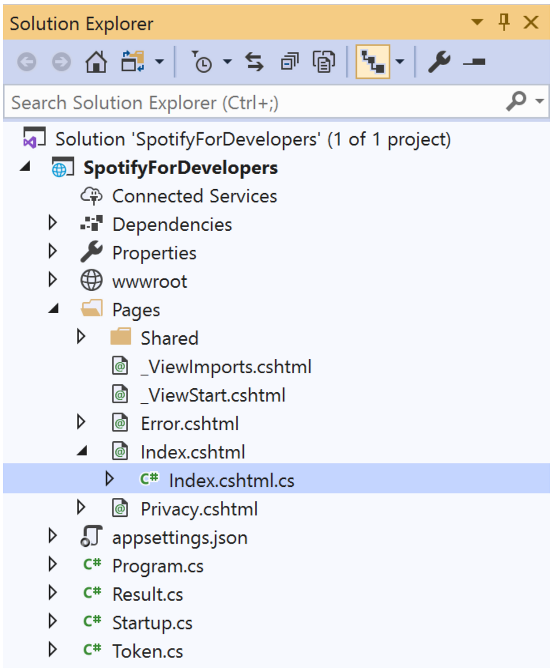
File name: SpotifyForDevelopers.sln

All Project Files (*.sln;*.dsw;*.vcw

□ Do not load projects

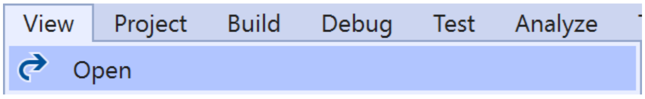
Then locate and select **SpotifyForDevelopers.sln** and select **Open** if you don't have this file already then please follow the previous parts of the workshop including **Getting Started**, **Authorisation Guide**, **Search & Browse**, **Playlists & Artists**, **Albums & Tracks**, **Episodes & Shows**, **Follow**, **Playlists** and **Library**

Step 2



Once opened, in the **Solution Explorer** open the **Pages** section, then open the **Index.cshtml** section and select **Index.cshtml.cs**

Step 3



Then from the **Menu** choose **View** and then **Open**

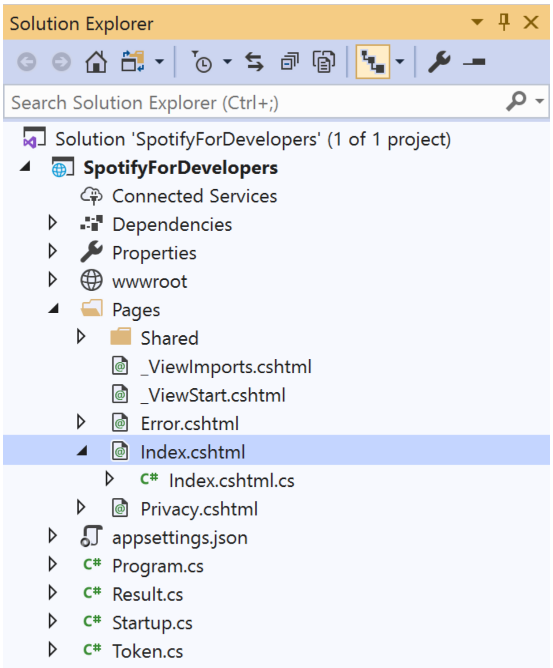
Step 4

In the **Code View** for **Index.cshtml.cs** below the **method** for `public async Task<IActionResult> OnPostRemoveUserShowsAsync(...) { ... }` enter the following **method**:

```
public async Task<IActionResult> OnPostUserPlaybackAddToQueueAsync(string value, string
option)
{
    LoadToken();
    var uri = $"spotify:{option.ToLower()}:{value}";
    var result = await Api.UserPlaybackAddToQueueAsync(uri);
    if (result != null)
    {
        Results = new List<Result>() { new Result()
        {
            Id = result.Code.ToString(),
            Name = result.Success.ToString()
        }};
    }
    return Page();
}
```

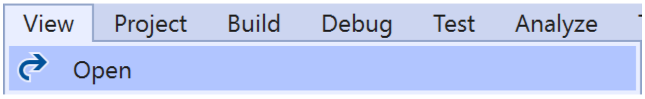
The **method** for `OnPostUserPlaybackAddToQueueAsync` **adds** the **track** or **episode** to the **playback queue** with `Value` of the **Item Id** of the **Track Id** or **Episode Id** and the `Option` of **Track** or **Episode** and populate the **property** for `Results` of the **success** of the operation accordingly.

Step 5



In the **Solution Explorer** in the **Pages** section select **Index.cshtml**

Step 6



Then from the **Menu** choose **View** and then **Open**

Step 7

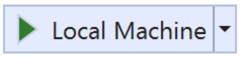
Once in the **Code View** for **Index.cshtml** above `<!-- Spotify Web API User Authorisation -->` enter the following:

```
<ul class="list-group mb-2">
    <li class="list-group-item list-group-item-primary">
        <h5 class="list-group-item-heading">Player</h5>
    </li>
    <li class="list-group-item">
        <form asp-page-handler="UserPlaybackAddToQueue" method="post">
            <select asp-for="Option" class="form-control mb-2">
                <option>Track</option>
                <option>Episode</option>
            </select>
            <input asp-for="Value" placeholder="Item Id" class="form-control mb-2" />
            <button class="btn btn-primary mb-2">
                Add an Item to the User's Playback Queue
            </button>
        </form>
    </li>

    <!-- Player -->
</ul>
```

This `form` will **post** to the **method** for `UserPlaybackAddToQueue` with the `Value` of the **Item Id** and the `Option` of **Track** or **Episode** and will output to the **Results**.

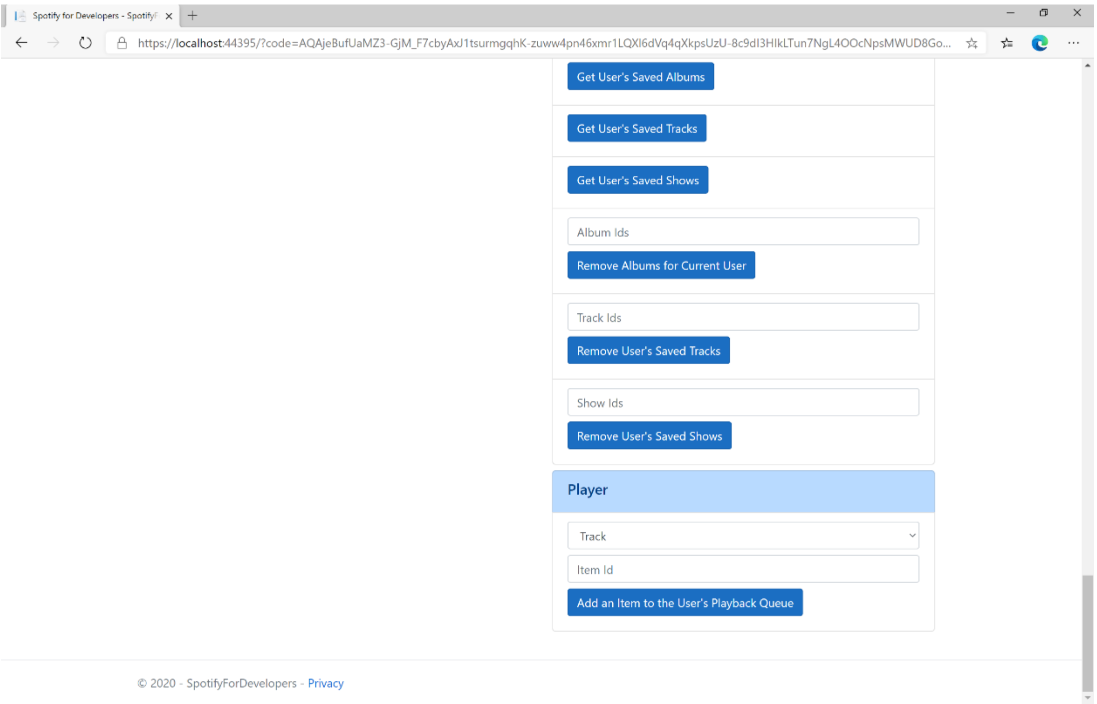
Step 8



Finally, in **Visual Studio 2019** select **IIS Application** to run the **Web Application**

Step 9

Once the **Web Application** is running and you select **Authorisation Code Flow Login** or **Implicit Grant Flow Login** and scroll down you should see something like the following:

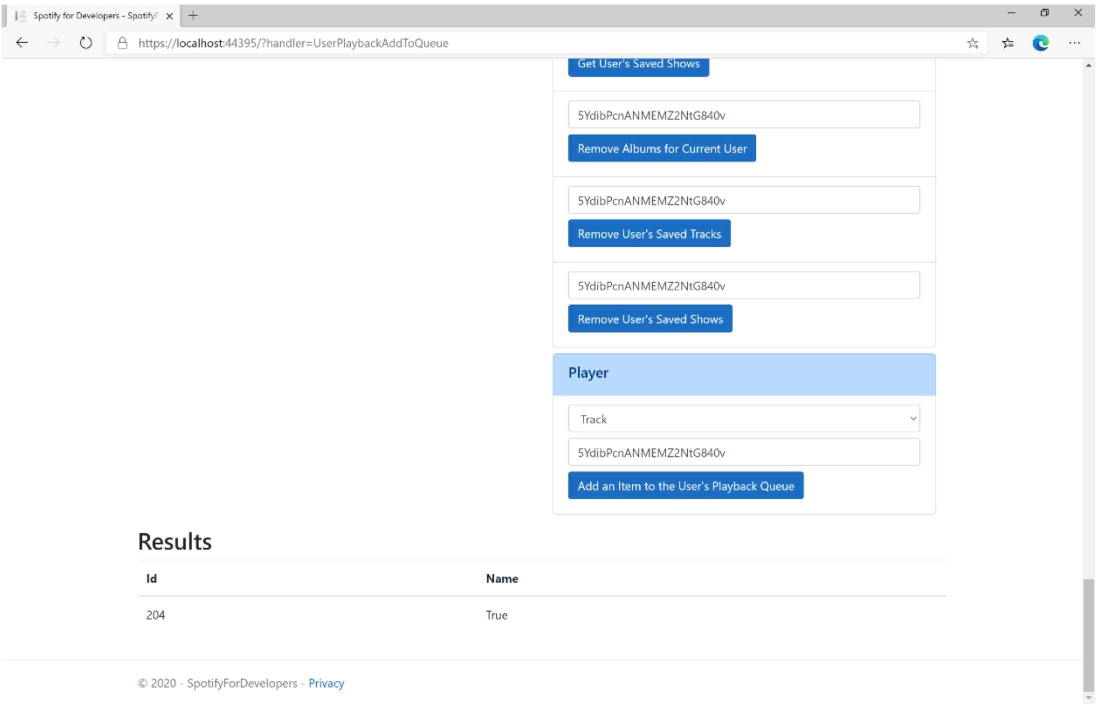

Spotify for Developers - Spotify
Get User's Saved Albums
Get User's Saved Tracks
Get User's Saved Shows
Album Ids
Remove Albums for Current User
Track Ids
Remove User's Saved Tracks
Show Ids
Remove User's Saved Shows
Player
Track
Item Id
Add an Item to the User's Playback Queue
Privacy

https://localhost:44395/?code=AQAjeBufUaMZ3-GjM_F7cbyAxJ1tsurmgqhK-zuww4pn46xmr1LQXl6dVq4qXkpsUzU-8c9dI3HIkLTun7NgL4OOcNpsMWUD8Go...

© 2020 - SpotifyForDevelopers - Privacy

Step 10

Make sure **Spotify** is running then you can get a **Track Id** from **Search For An Item** using **Tracks** or an **Episode Id** using **Episodes** and enter this in **Item Id** above **Add an Item to the User's Playback Queue** and then select **Add an Item to the User's Playback Queue** and also scroll down to view **Results** like the following:


Spotify for Developers - Spotify
Get User's Saved Shows
Remove Albums for Current User
Remove User's Saved Tracks
Remove User's Saved Shows
Player
Track
Add an Item to the User's Playback Queue
Privacy

https://localhost:44395/?handler=UserPlaybackAddToQueue

5YdibPcnANMEMZ2NtG840v

5YdibPcnANMEMZ2NtG840v

5YdibPcnANMEMZ2NtG840v

5YdibPcnANMEMZ2NtG840v

Results

Id	Name
204	True

© 2020 - SpotifyForDevelopers - Privacy

Step 11

You can stop the **web application** in **Visual Studio 2019** by selecting the **Stop debugging** button

Step 12

You can choose to exit **Visual Studio 2019** by selecting the **Close** button in the top right of the **application** as that completes this part of the workshop

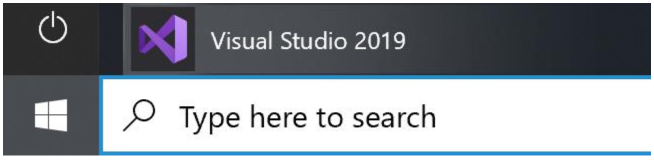
Start/Resume a User's Playback

Start a new context or resume current playback on the user's active device.

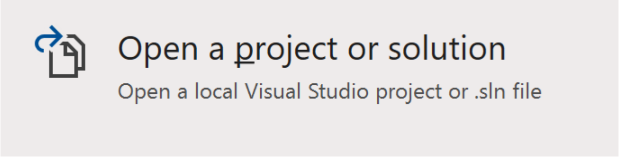
PUT https://api.spotify.com/v1/me/player/play	
Header	
Authorization	User Token from Spotify Accounts service with **user-modify-playback-state** scope
Query Parameter	
device_id	Device Id to target, if not supplied the user's current active device is the target
Body Parameter	
context_uri	Spotify URI of the context to play. Valid contexts are albums, artists, playlists
uris	Spotify track URIs to play
offset	Indicates from where in the context playback should start. Only available when context_uri corresponds to an album or playlist object, or when the uris parameter is used.
position_ms	Indicates from what position to start playback

Header	Response
Success	
Http Status 204 No Content	
Error	
HTTP Status 404 Not Found	NO_ACTIVE_DEVICE - Device was not found
HTTP Status 403 Forbidden	PREMIUM_REQUIRED - Returned when account non-premium
Error Code	Error Object

Step 1



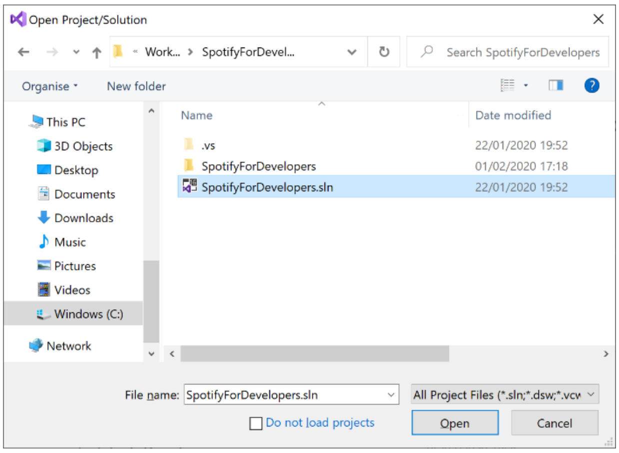
If you chose to close **Visual Studio 2019** previously, in **Windows 10** choose **Start**, and then from the **Start Menu** find and select **Visual Studio 2019**



Open a project or solution

Open a local Visual Studio project or .sln file

Once done, from the **Get started** screen for **Visual Studio 2019** select **Open a project or solution**



Name	Date modified
.vs	22/01/2020 19:52
SpotifyForDevelopers	01/02/2020 17:18
SpotifyForDevelopers.sln	22/01/2020 19:52

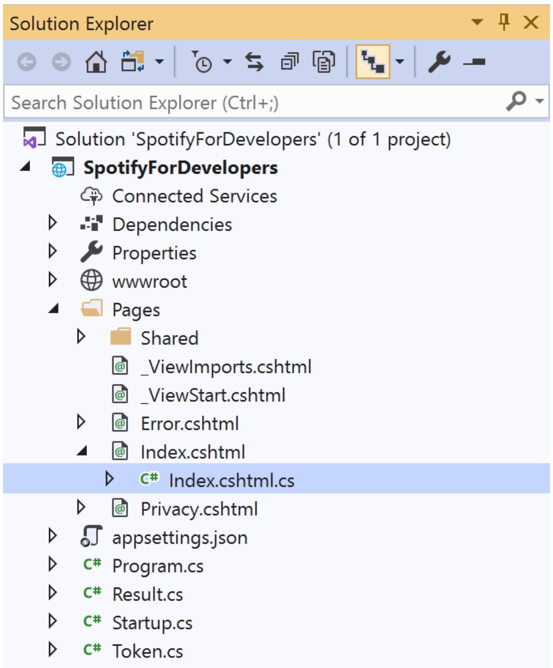
File name: SpotifyForDevelopers.sln

All Project Files (*.sln;*.dsw;*.vcw

□ Do not load projects

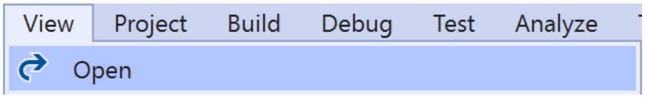
Then locate and select **SpotifyForDevelopers.sln** and select **Open** if you don't have this file already then please follow the previous parts of the workshop including **Getting Started**, **Authorisation Guide**, **Search & Browse**, **Playlists & Artists**, **Albums & Tracks**, **Episodes & Shows**, **Follow**, **Playlists**, **Library** and **Player**

Step 2



Once opened, in the **Solution Explorer** open the **Pages** section, then open the **Index.cshtml** section and select **Index.cshtml.cs**

Step 3



Then from the **Menu** choose **View** and then **Open**

Step 4

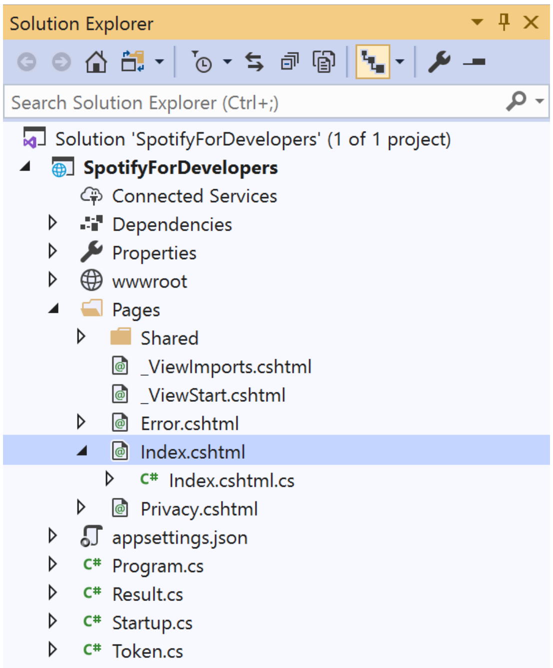
In the **Code View** for **Index.cshtml.cs** below the **method** for `public async Task<IActionResult> OnPostUserPlaybackAddToQueueAsync(...) { ... }` enter the following **method**:

```
public async Task<IActionResult> OnPostUserPlaybackStartResumeAsync(string value, string
option)
{

    var isTrack = option.Equals("Track");
    var uri = $"spotify:{option.ToLower()}:{value}";
    var uris = new List<string> { uri };
    var result = await Api.UserPlaybackStartResumeAsync(
        contextUri: isTrack ? null : uri,
        uris: isTrack ? uris : null,
        offsetPosition: 0, position: 0);
    if (result != null)
    {
        Results = new List<Result>() { new Result()
        {
            Id = result.Code.ToString(),
            Name = result.Success.ToString()
        }};
    }
    return Page();
}
```

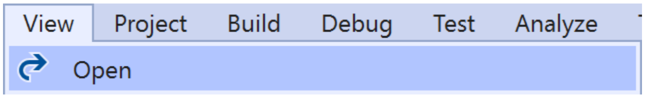
The **method** for `OnPostUserPlaybackStartResumeAsync` will **start** the **playback** with the `Value` of the **Item Id** of the `Option` of an **Album**, **Track**, **Artist** or **Playlist** and populate the **property** for `Results` of the **success** of the operation accordingly.

Step 5



In the **Solution Explorer** in the **Pages** section select **Index.cshtml**

Step 6



Then from the **Menu** choose **View** and then **Open**

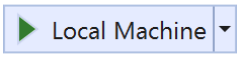
Step 7

Once in the **Code View** for **Index.cshtml** above `<!-- Player -->` enter the following:

```
<li class="list-group-item">
    <form asp-page-handler="UserPlaybackStartResume" method="post">
        <select asp-for="Option" class="form-control mb-2">
            <option>Album</option>
            <option>Track</option>
            <option>Artist</option>
            <option>Playlist</option>
        </select>
        <input asp-for="Value" placeholder="Item Id" class="form-control mb-2" />
        <button class="btn btn-primary mb-2">
            Start/Resume a User's Playback
        </button>
    </form>
</li>
```

This `form` will **post** to the **method** for `UserPlaybackStartResume` with the `Value` of the **Item Id** and the `Option` of **Album**, **Track**, **Artist** and **Playlist** will output to the **Results**.

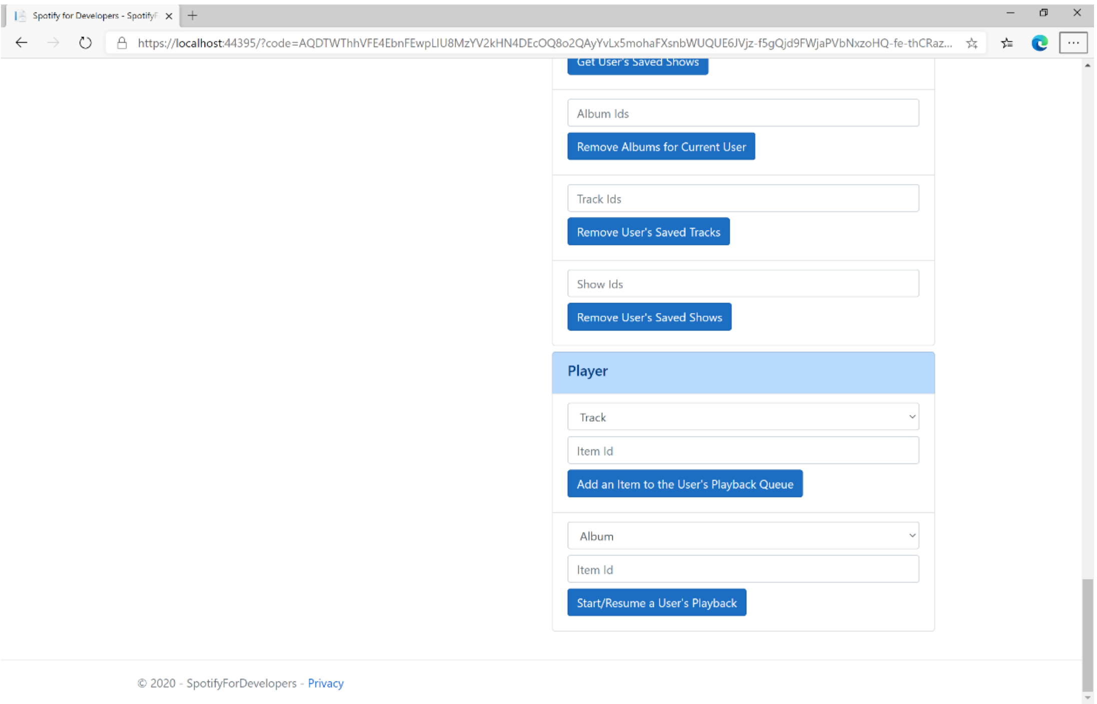
Step 8



Finally, in **Visual Studio 2019** select **IIS Application** to run the **Web Application**

Step 9

Once the **Web Application** is running and you select **Authorisation Code Flow Login** or **Implicit Grant Flow Login** and scroll down you should see something like the following:

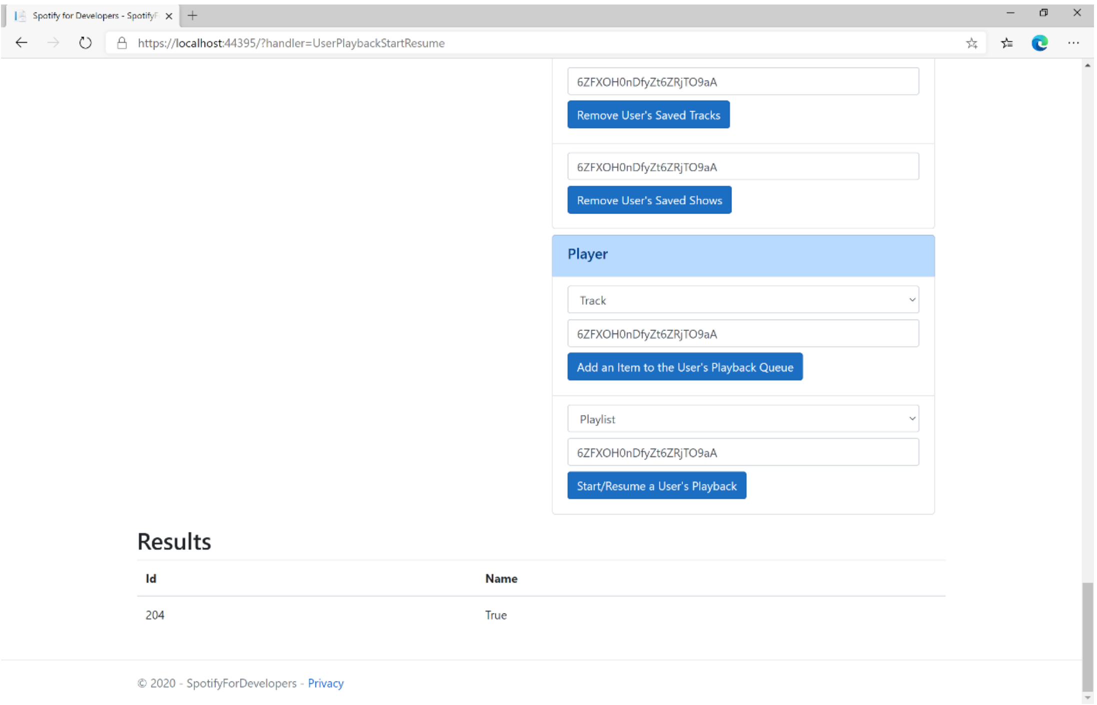

Spotify for Developers - SpotifyF

https://localhost:44395/?code=AQDTWThhVFE4EbnFEwpLIU8MzYV2kHN4DEcOQ8o2QAyYvLx5mohaFXsnbWUQUE6JVjz-f5gQjd9FWjaPVbNxzoHQ-fe-thCRaz...

Get User's Saved Shows
Album Ids
Remove Albums for Current User
Track Ids
Remove User's Saved Tracks
Show Ids
Remove User's Saved Shows
Player
Track
Item Id
Add an Item to the User's Playback Queue
Album
Item Id
Start/Resume a User's Playback
© 2020 - SpotifyForDevelopers - Privacy

Step 10

Make sure **Spotify** is running then get an **Album Id** from **Search For An Item** using **Albums** or an **Track Id** using **Tracks**, or an **Artist Id** using **Artists**, or a **Playlist Id** using **Playlists** and enter this in **Item Id** above **Start/Resume a User's Playback** and then select **Start/Resume a User's Playback** and scroll down to view **Results** like the following:


Spotify for Developers - SpotifyF

https://localhost:44395/?handler=UserPlaybackStartResume

6ZFXOH0nDfyZt6ZRjTO9aA

Remove User's Saved Tracks

6ZFXOH0nDfyZt6ZRjTO9aA

Remove User's Saved Shows
Player

Track

6ZFXOH0nDfyZt6ZRjTO9aA

Add an Item to the User's Playback Queue

Playlist

6ZFXOH0nDfyZt6ZRjTO9aA

Start/Resume a User's Playback

Results

Id	Name
204	True

© 2020 - SpotifyForDevelopers - Privacy

Step 11

■

You can stop the **web application** in **Visual Studio 2019** by selecting the **Stop debugging** button

Step 12

×

You can choose to exit **Visual Studio 2019** by selecting the **Close** button in the top right of the **application** as that completes this part of the workshop

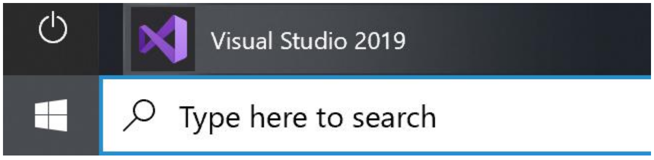
Pause a User's Playback

Pause playback on the user’s account.

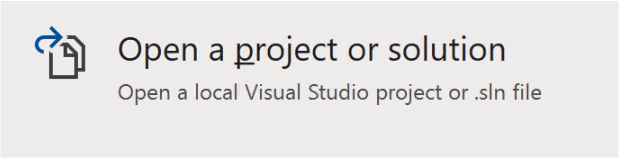
PUT https://api.spotify.com/v1/me/player/pause	
Header	
Authorization	User Token from Spotify Accounts service with **user-modify-playback-state** scope
Query Parameter	
device_id	Device Id to target, if not supplied the user’s current active device is the target

Header	**Response**
Success	
Http Status 204 No Content	
Error	
HTTP Status 404 Not Found	NO_ACTIVE_DEVICE - Device was not found
HTTP Status 403 Forbidden	PREMIUM_REQUIRED - Returned when account non-premium
Error Code	Error Object

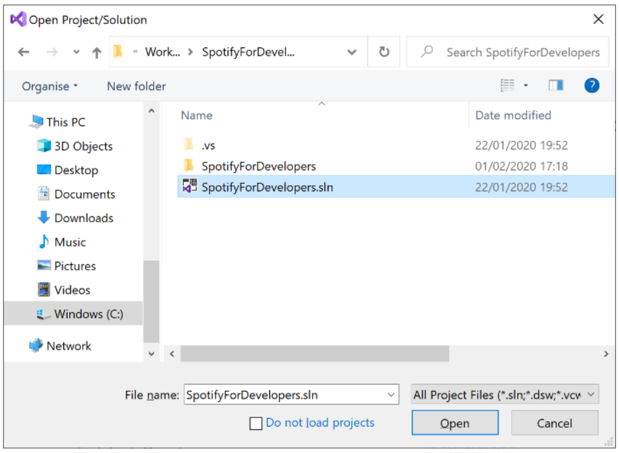
Step 1



If you chose to close **Visual Studio 2019** previously, in **Windows 10** choose **Start**, and then from the **Start Menu** find and select **Visual Studio 2019**


Open a project or solution
Open a local Visual Studio project or .sln file

Once done, from the **Get started** screen for **Visual Studio 2019** select **Open a project or solution**

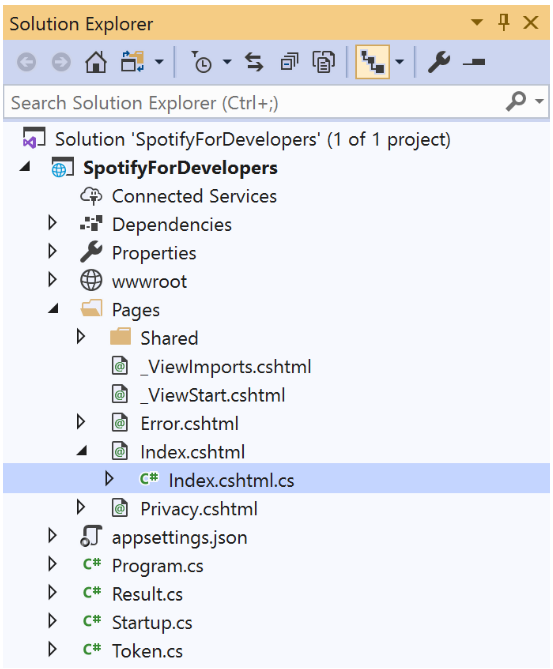


Name	Date modified
.vs	22/01/2020 19:52
SpotifyForDevelopers	01/02/2020 17:18
SpotifyForDevelopers.sln	22/01/2020 19:52

File name: SpotifyForDevelopers.sln

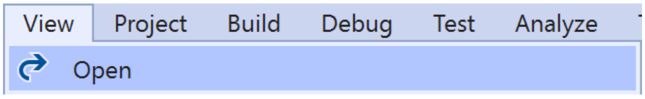
Then locate and select **SpotifyForDevelopers.sln** and select **Open** if you don’t have this file already then please follow the previous parts of the workshop including **Getting Started**, **Authorisation Guide**, **Search & Browse**, **Playlists & Artists**, **Albums & Tracks**, **Episodes & Shows**, **Follow**, **Playlists**, **Library** and **Player**

Step 2



Once opened, in the **Solution Explorer** open the **Pages** section, then open the **Index.cshtml** section and select **Index.cshtml.cs**

Step 3



Then from the **Menu** choose **View** and then **Open**

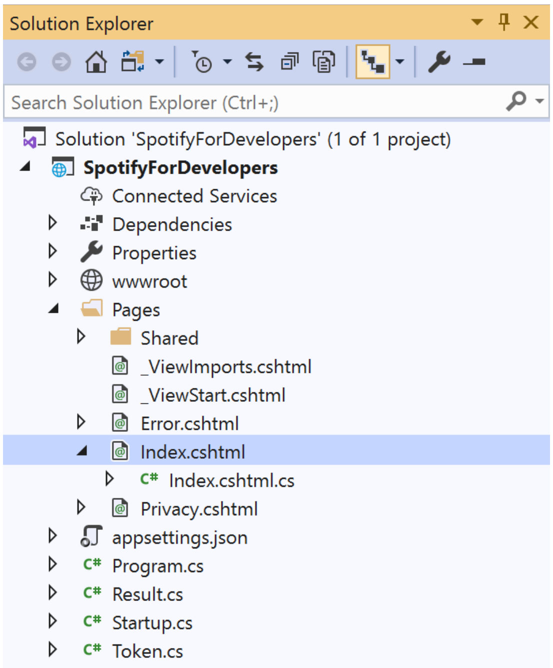
Step 4

In the **Code View** for **Index.cshtml.cs** below the **method** for `public async Task<IActionResult> OnPostUserPlaybackStartResumeAsync(...) { ... }` enter the following **method**:

```
public async Task<IActionResult> OnPostUserPlaybackPauseAsync()
{
    LoadToken();
    var result = await Api.UserPlaybackPauseAsync();
    if (result != null)
    {
        Results = new List<Result>() { new Result()
        {
            Id = result.Code.ToString(),
            Name = result.Success.ToString()
        }};
    }
    return Page();
}
```

The **method** for `OnPostUserPlaybackPauseAsync` will **pause** any **playback** and populate the **property** for `Results` of the **success** of the operation accordingly.

Step 5



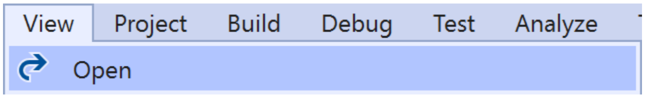
Solution Explorer

Search Solution Explorer (Ctrl+;)

- Solution 'SpotifyForDevelopers' (1 of 1 project)
  - SpotifyForDevelopers
    - Connected Services
    - Dependencies
    - Properties
    - wwwroot
    - Pages
      - Shared
      - _ViewImports.cshtml
      - _ViewStart.cshtml
      - Error.cshtml
      - Index.cshtml
        - Index.cshtml.cs
      - Privacy.cshtml
    - appsettings.json
    - Program.cs
    - Result.cs
    - Startup.cs
    - Token.cs

In the **Solution Explorer** in the **Pages** section select **Index.cshtml**

Step 6



Then from the **Menu** choose **View** and then **Open**

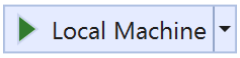
Step 7

Once in the **Code View** for **Index.cshtml** above `<!-- Player -->` enter the following:

```
<li class="list-group-item">
    <form asp-page-handler="UserPlaybackPause" method="post">
        <button class="btn btn-primary mb-2">
            Pause a User's Playback
        </button>
    </form>
</li>
```

This `form` will **post** to the **method** for `UserPlaybackPause` and will output to the **Results**.

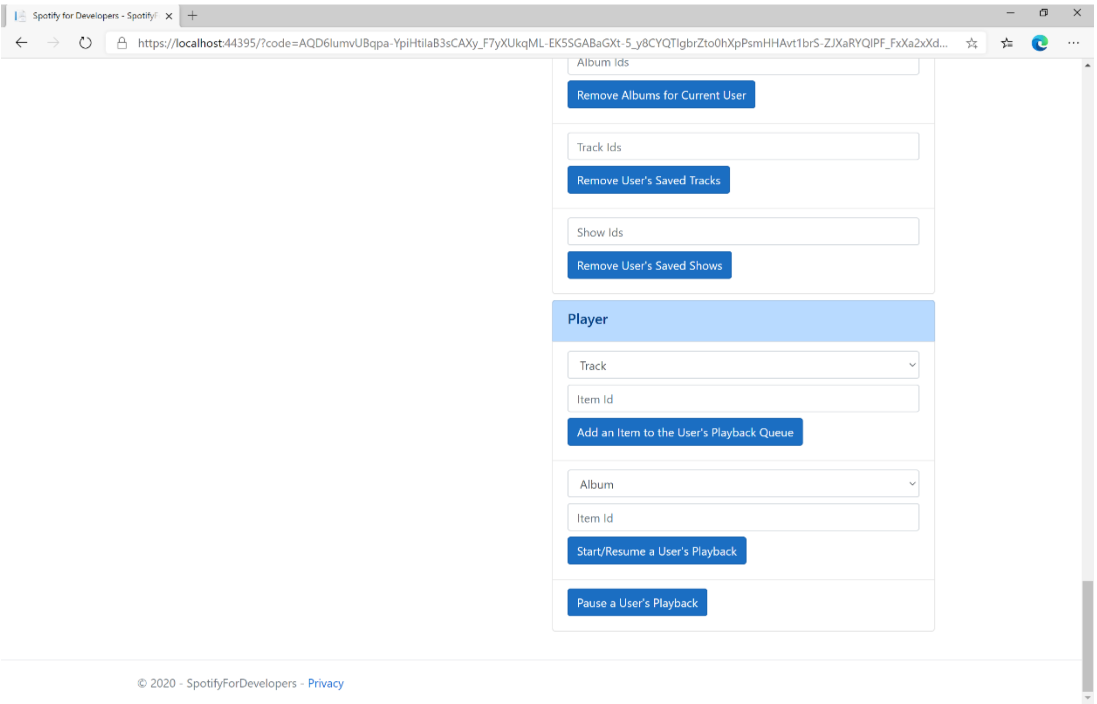
Step 8



Finally, in **Visual Studio 2019** select **IIS Application** to run the **Web Application**

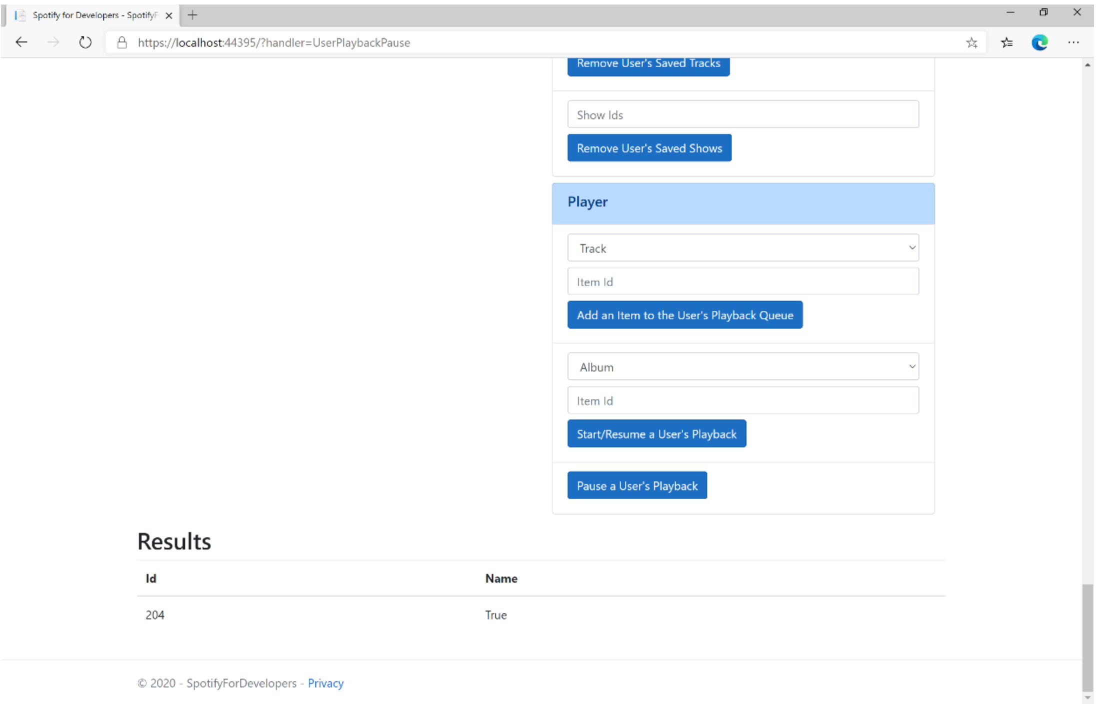
Step 9

Once the **Web Application** is running and you select **Authorisation Code Flow Login** or **Implicit Grant Flow Login** and scroll down you should see something like the following:


Spotify for Developers - SpotifyF
https://localhost:44395/?code=AQD6lumvUBqpa-YpiHtilaB3sCAXy_F7yXUkqML-EK5SGABaGXt-5_y8CYQTlgbrZto0hXpPsmHHAvt1brS-ZJXaRYQIPF_FxXa2xXd...
Album Ids
Remove Albums for Current User
Track Ids
Remove User's Saved Tracks
Show Ids
Remove User's Saved Shows
Player
Track
Item Id
Add an Item to the User's Playback Queue
Album
Item Id
Start/Resume a User's Playback
Pause a User's Playback
© 2020 - SpotifyForDevelopers - Privacy

Step 10

Make sure **Spotify** is running then you can select **Pause a User's Playback** and then scroll down to view **Results** like the following:


Spotify for Developers - SpotifyF
https://localhost:44395/?handler=UserPlaybackPause
Remove User's Saved Tracks
Show Ids
Remove User's Saved Shows
Player
Track
Item Id
Add an Item to the User's Playback Queue
Album
Item Id
Start/Resume a User's Playback
Pause a User's Playback

Results

Id	Name
204	True

© 2020 - SpotifyForDevelopers - Privacy

Step 11

■

You can stop the **web application** in **Visual Studio 2019** by selecting the **Stop debugging** button

Step 12

✕

You can choose to exit **Visual Studio 2019** by selecting the **Close** button in the top right of the **application** as that completes this part of the workshop

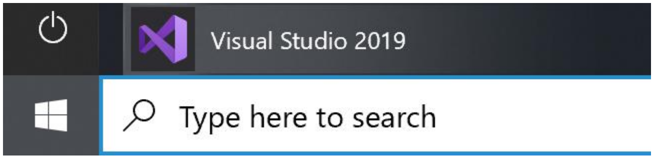
Skip User’s Playback to Next Track

Skips to next track in the user’s queue.

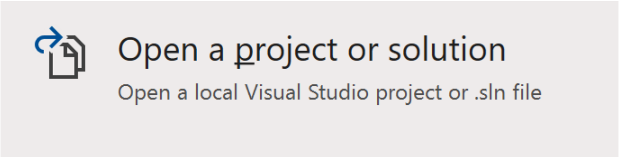
POST https://api.spotify.com/v1/me/player/next	
Header	
Authorization	User Token from Spotify Accounts service with **user-modify-playback-state** scope
Query Parameter	
device_id	Device Id to target, if not supplied the user’s current active device is the target

Header	Response
Success	
HTTP Status 204 No Content	
Error	
HTTP Status 404 Not Found	NO_ACTIVE_DEVICE - Device was not found
HTTP Status 403 Forbidden	PREMIUM_REQUIRED - Returned when account non-premium
Error Code	Error Object

Step 1

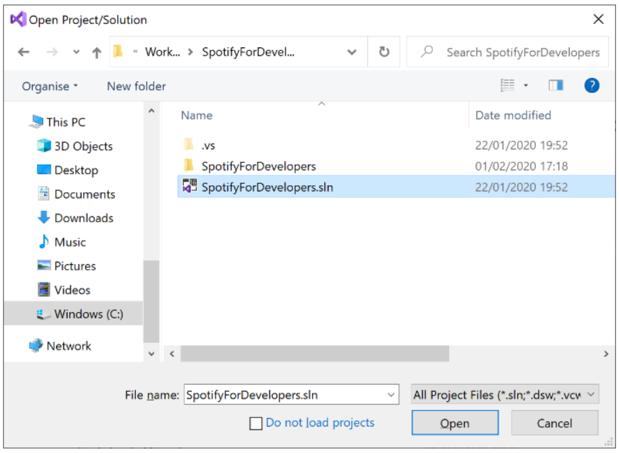


If you chose to close **Visual Studio 2019** previously, in **Windows 10** choose **Start**, and then from the **Start Menu** find and select **Visual Studio 2019**


Open a project or solution

Open a local Visual Studio project or .sln file

Once done, from the **Get started** screen for **Visual Studio 2019** select **Open a project or solution**

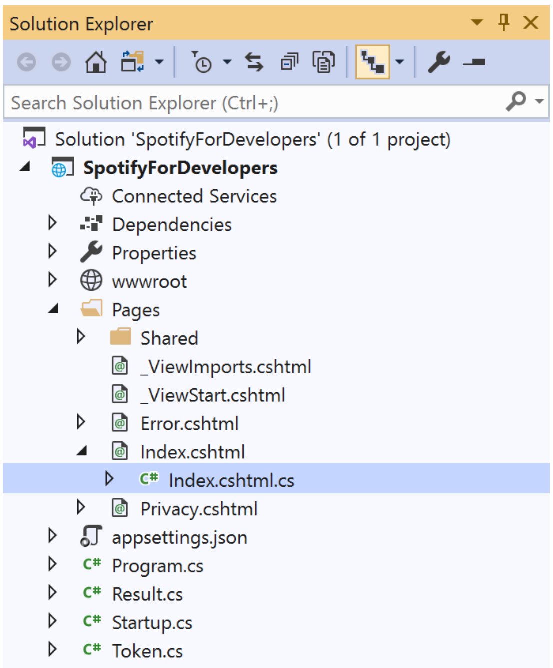


Name	Date modified
.vs	22/01/2020 19:52
SpotifyForDevelopers	01/02/2020 17:18
SpotifyForDevelopers.sln	22/01/2020 19:52

File name: SpotifyForDevelopers.sln

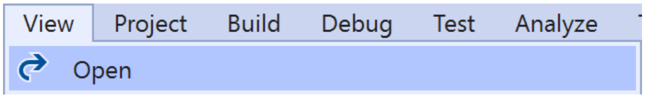
Then locate and select **SpotifyForDevelopers.sln** and select **Open** if you don’t have this file already then please follow the previous parts of the workshop including **Getting Started**, **Authorisation Guide**, **Search & Browse**, **Playlists & Artists**, **Albums & Tracks**, **Episodes & Shows**, **Follow**, **Playlists**, **Library** and **Player**

Step 2



Once opened, in the **Solution Explorer** open the **Pages** section, then open the **Index.cshtml** section and select **Index.cshtml.cs**

Step 3



Then from the **Menu** choose **View** and then **Open**

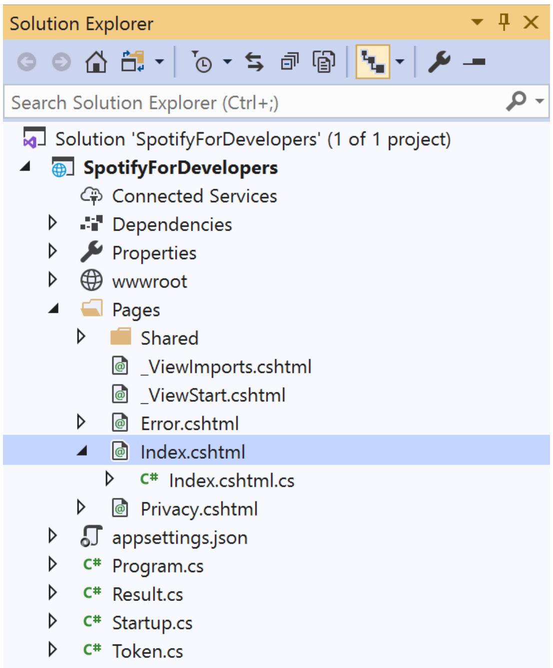
Step 4

In the **Code View** for **Index.cshtml.cs** below the **method** for `public async Task<IActionResult> OnPostUserPlaybackPauseAsync() { ... }` enter the following **method**:

```
public async Task<IActionResult> OnPostUserPlaybackNextTrackAsync()
{
    LoadToken();
    var result = await Api.UserPlaybackNextTrackAsync();
    if (result != null)
    {
        Results = new List<Result>() { new Result()
        {
            Id = result.Code.ToString(),
            Name = result.Success.ToString()
        }};
    }
    return Page();
}
```

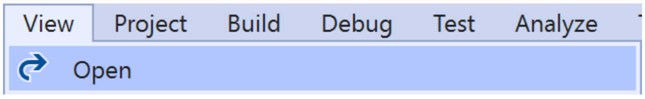
The **method** for `OnPostUserPlaybackNextTrackAsync` will **skip** to the **next track** in **playback** and populate the **property** for `Results` of the **success** of the operation accordingly.

Step 5



In the **Solution Explorer** in the **Pages** section select **Index.cshtml**

Step 6



Then from the **Menu** choose **View** and then **Open**

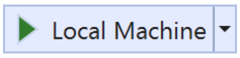
Step 7

Once in the **Code View** for **Index.cshtml** above `<!-- Player -->` enter the following:

```
<li class="list-group-item">
    <form asp-page-handler="UserPlaybackNextTrack" method="post">
        <button class="btn btn-primary mb-2">
            Skip User’s Playback to Next Track
        </button>
    </form>
</li>
```

This `form` will **post** to the **method** for `UserPlaybackNextTrack` and will output to the **Results**.

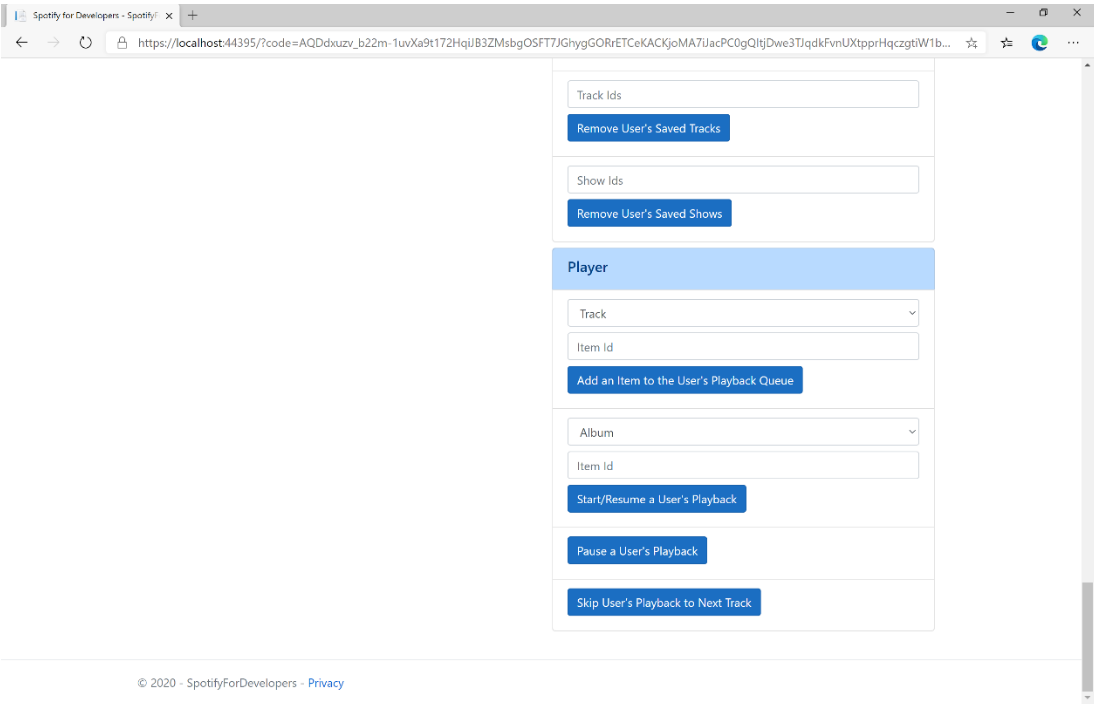
Step 8



Finally, in **Visual Studio 2019** select **IIS Application** to run the **Web Application**

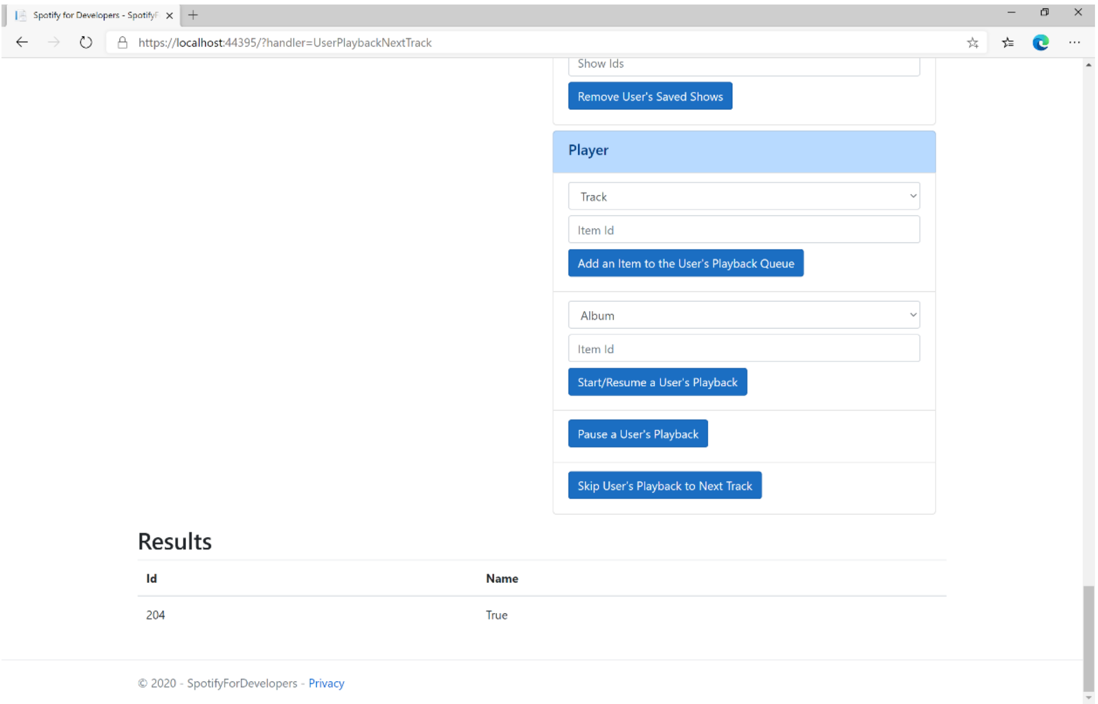
Step 9

Once the **Web Application** is running and you select **Authorisation Code Flow Login** or **Implicit Grant Flow Login** and scroll down you should see something like the following:


Spotify for Developers - SpotifyF
https://localhost:44395/?code=AQDdxuzv_b22m-1uvXa9t172HqiJB3ZMsbgOSFT7JGhygGORrETCeKACKjoMA7iJacPC0gQltjDwe3TJqdkFvnUXtpprHqczgtiW1b...
Track Ids
Remove User's Saved Tracks
Show Ids
Remove User's Saved Shows
Player
Track
Item Id
Add an Item to the User's Playback Queue
Album
Item Id
Start/Resume a User's Playback
Pause a User's Playback
Skip User's Playback to Next Track
© 2020 - SpotifyForDevelopers - Privacy

Step 10

Make sure **Spotify** is running then you can select **Skip User's Playback to Next Track** and then scroll down to view **Results** like the following:


Spotify for Developers - SpotifyF
https://localhost:44395/?handler=UserPlaybackNextTrack
Show Ids
Remove User's Saved Shows
Player
Track
Item Id
Add an Item to the User's Playback Queue
Album
Item Id
Start/Resume a User's Playback
Pause a User's Playback
Skip User's Playback to Next Track
© 2020 - SpotifyForDevelopers - Privacy

Results

Id	Name
204	True

Step 11

You can stop the **web application** in **Visual Studio 2019** by selecting the **Stop debugging** button

Step 12

You can choose to exit **Visual Studio 2019** by selecting the **Close** button in the top right of the **application** as that completes this part of the workshop

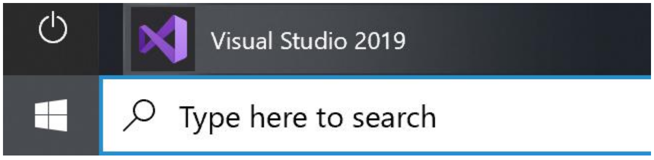
Skip User’s Playback to Previous Track

Skips to previous track in the user’s queue.

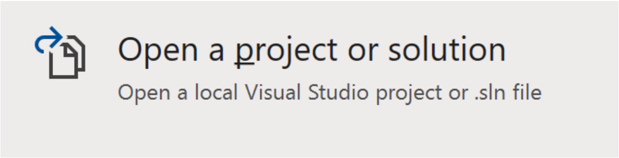
POST https://api.spotify.com/v1/me/player/previous	
Header	
Authorization	User Token from Spotify Accounts service with **user-modify-playback-state** scope
Query Parameter	
device_id	Device Id to target, if not supplied the user’s current active device is the target

Header	Response
Success	
HTTP Status 204 No Content	
Error	
HTTP Status 404 Not Found	NO_ACTIVE_DEVICE - Device was not found
HTTP Status 403 Forbidden	PREMIUM_REQUIRED - Returned when account non-premium
Error Code	Error Object

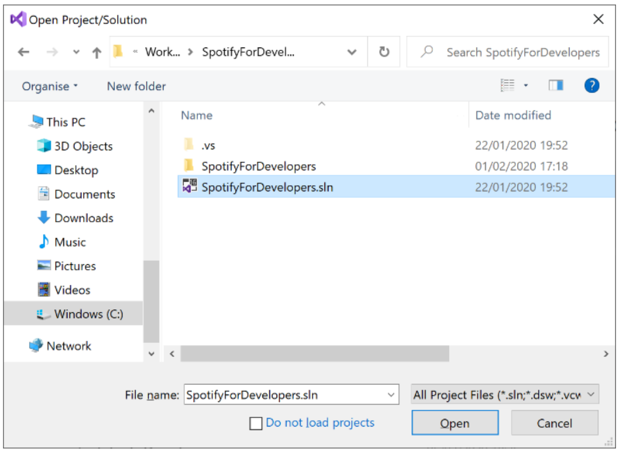
Step 1



If you chose to close **Visual Studio 2019** previously, in **Windows 10** choose **Start**, and then from the **Start Menu** find and select **Visual Studio 2019**


Open a project or solution
Open a local Visual Studio project or .sln file

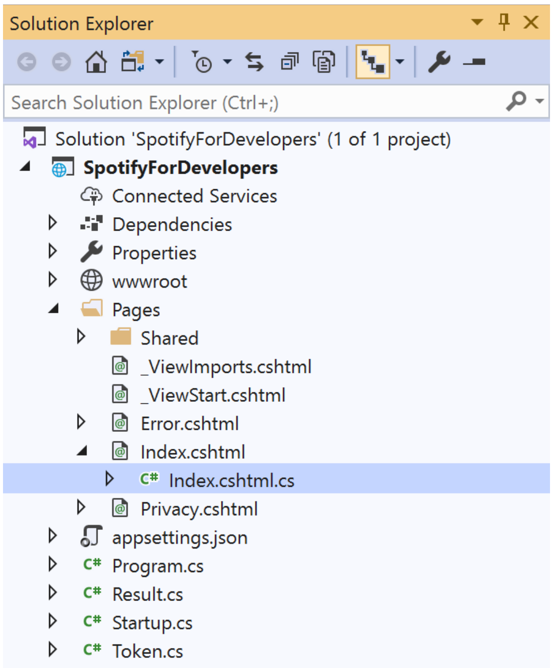
Once done, from the **Get started** screen for **Visual Studio 2019** select **Open a project or solution**



File name: SpotifyForDevelopers.sln

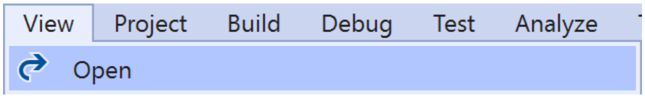
Then locate and select **SpotifyForDevelopers.sln** and select **Open** if you don’t have this file already then please follow the previous parts of the workshop including **Getting Started**, **Authorisation Guide**, **Search & Browse**, **Playlists & Artists**, **Albums & Tracks**, **Episodes & Shows**, **Follow**, **Playlists**, **Library** and **Player**

Step 2



Once opened, in the **Solution Explorer** open the **Pages** section, then open the **Index.cshtml** section and select **Index.cshtml.cs**

Step 3



Then from the **Menu** choose **View** and then **Open**

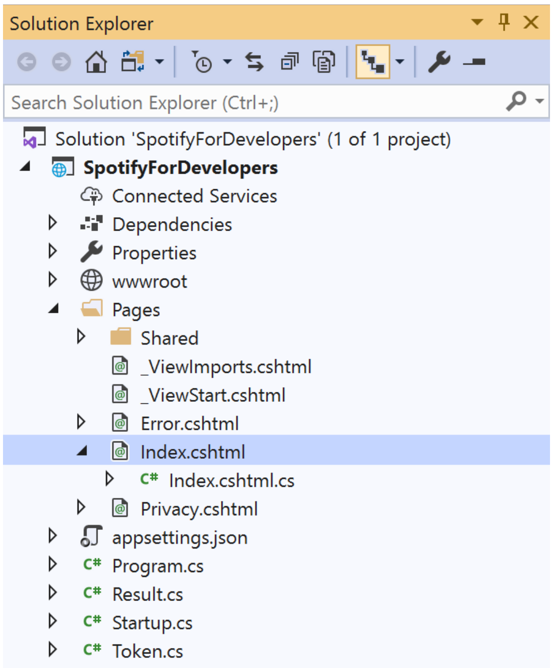
Step 4

In the **Code View** for **Index.cshtml.cs** below the **method** for `public async Task<IActionResult> OnPostUserPlaybackNextTrackAsync() { ... }` enter the following **method**:

```
public async Task<IActionResult> OnPostUserPlaybackPreviousTrackAsync()
{
    LoadToken();
    var result = await Api.UserPlaybackPreviousTrackAsync();
    if (result != null)
    {
        Results = new List<Result>() { new Result()
        {
            Id = result.Code.ToString(),
            Name = result.Success.ToString()
        }};
    }
    return Page();
}
```

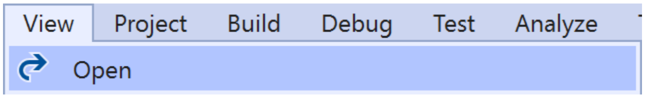
The **method** for `OnPostUserPlaybackPreviousTrackAsync` will **skip** to the **previous track** in **playback** and populate the **property** for `Results` of the **success** of the operation accordingly.

Step 5



In the **Solution Explorer** in the **Pages** section select **Index.cshtml**

Step 6



Then from the **Menu** choose **View** and then **Open**

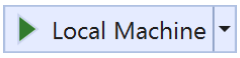
Step 7

Once in the **Code View** for **Index.cshtml** above `<!-- Player -->` enter the following:

```
<li class="list-group-item">
    <form asp-page-handler="UserPlaybackNextTrack" method="post">
        <button class="btn btn-primary mb-2">
            Skip User’s Playback to Previous Track
        </button>
    </form>
</li>
```

This `form` will **post** to the **method** for `UserPlaybackNextTrack` and will output to the **Results**.

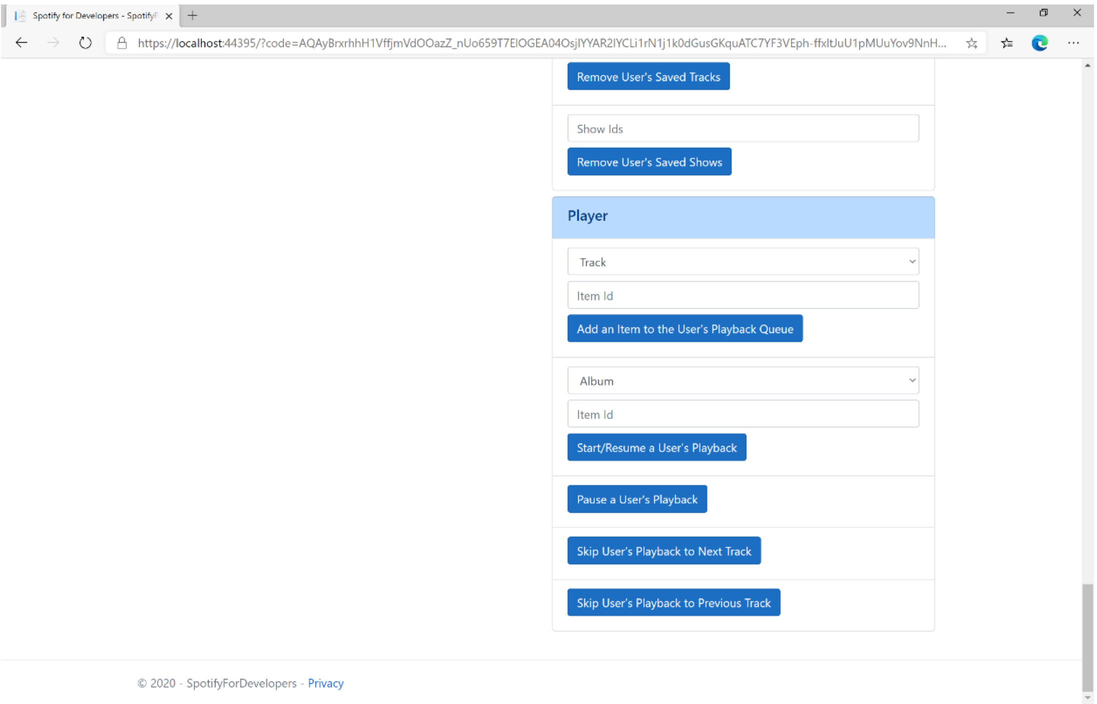
Step 8



Finally, in **Visual Studio 2019** select **IIS Application** to run the **Web Application**

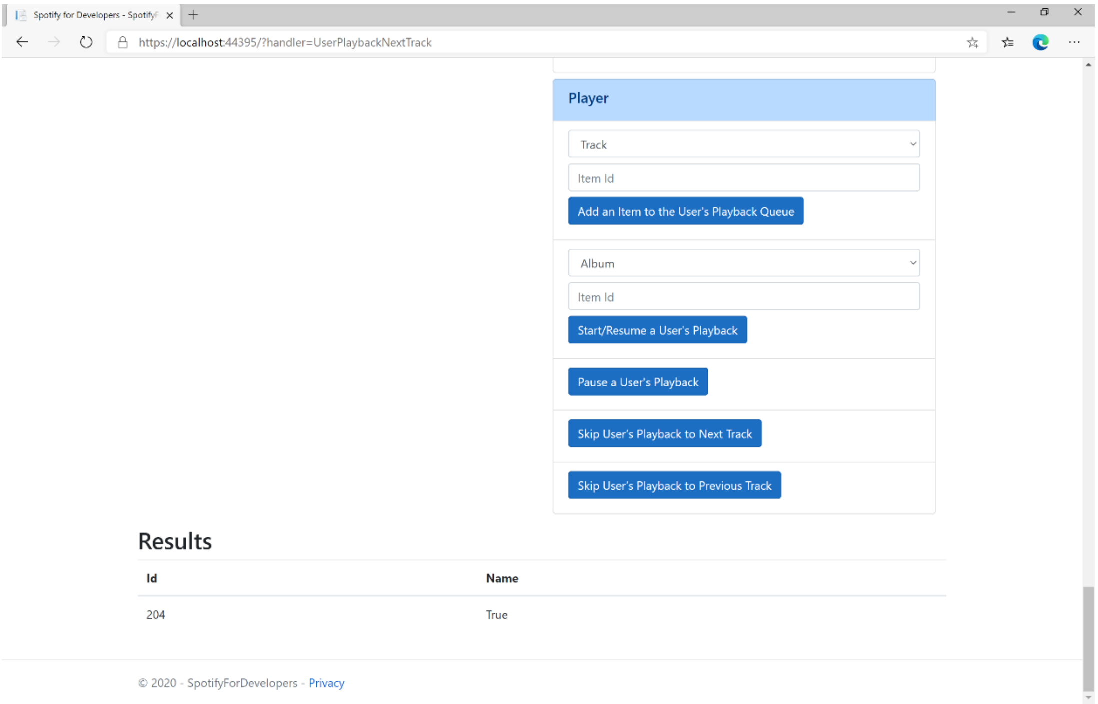
Step 9

Once the **Web Application** is running and you select **Authorisation Code Flow Login** or **Implicit Grant Flow Login** and scroll down you should see something like the following:


Spotify for Developers - SpotifyF
https://localhost:44395/?code=AQAyBrxrhhH1VffjmVdOOazZ_nUo659T7ElOGEA04OsjlYYAR2lYCLi1rN1j1k0dGusGKquATC7YF3VEph-ffxltJuU1pMUuYov9NnH...
Remove User's Saved Tracks
Show Ids
Remove User's Saved Shows
Player
Track
Item Id
Add an Item to the User's Playback Queue
Album
Item Id
Start/Resume a User's Playback
Pause a User's Playback
Skip User's Playback to Next Track
Skip User's Playback to Previous Track
© 2020 - SpotifyForDevelopers - Privacy

Step 10

Make sure **Spotify** is running then you can select **Skip User's Playback to Previous Track** and then scroll down to view **Results** like the following:


Spotify for Developers - SpotifyF
https://localhost:44395/?handler=UserPlaybackNextTrack
Player
Track
Item Id
Add an Item to the User's Playback Queue
Album
Item Id
Start/Resume a User's Playback
Pause a User's Playback
Skip User's Playback to Next Track
Skip User's Playback to Previous Track

Results

Id	Name
204	True

© 2020 - SpotifyForDevelopers - Privacy

Step 11

■

You can stop the **web application** in **Visual Studio 2019** by selecting the **Stop debugging** button

Step 12

✕

You can choose to exit **Visual Studio 2019** by selecting the **Close** button in the top right of the **application** as that completes this part of the workshop

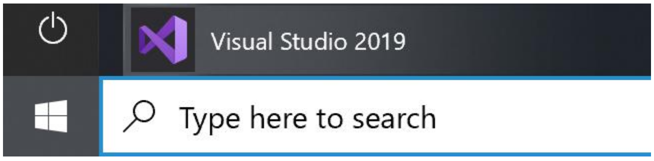
Seek to Position in Currently Playing Track

Seeks to the given position in the user's currently playing track.

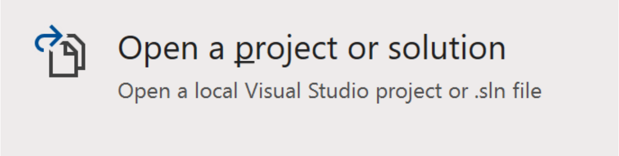
PUT https://api.spotify.com/v1/me/player/seek	
Header	
Authorization	User Token from Spotify Accounts service with **user-modify-playback-state** scope
Query Parameter	
position_ms	Position in milliseconds to seek to, must be a positive number
device_id	Device Id to target, if not supplied the user's current active device is the target

Header	**Response**
Success	
HTTP Status 204 No Content	
Error	
HTTP Status 404 Not Found	NO_ACTIVE_DEVICE - Device was not found
HTTP Status 403 Forbidden	PREMIUM_REQUIRED - Returned when account non-premium
Error Code	Error Object

Step 1



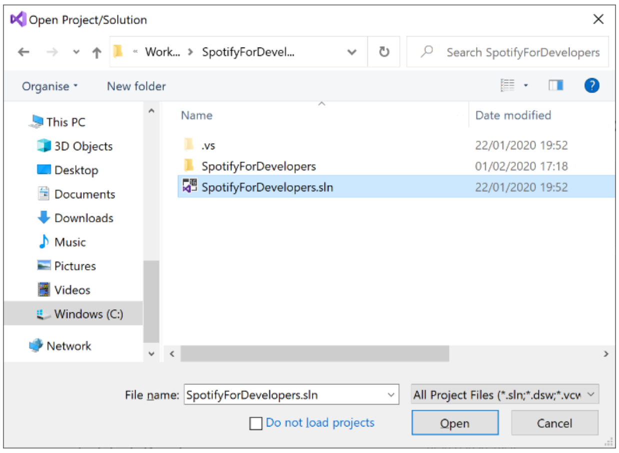
If you chose to close **Visual Studio 2019** previously, in **Windows 10** choose **Start**, and then from the **Start Menu** find and select **Visual Studio 2019**



Open a project or solution

Open a local Visual Studio project or .sln file

Once done, from the **Get started** screen for **Visual Studio 2019** select **Open a project or solution**

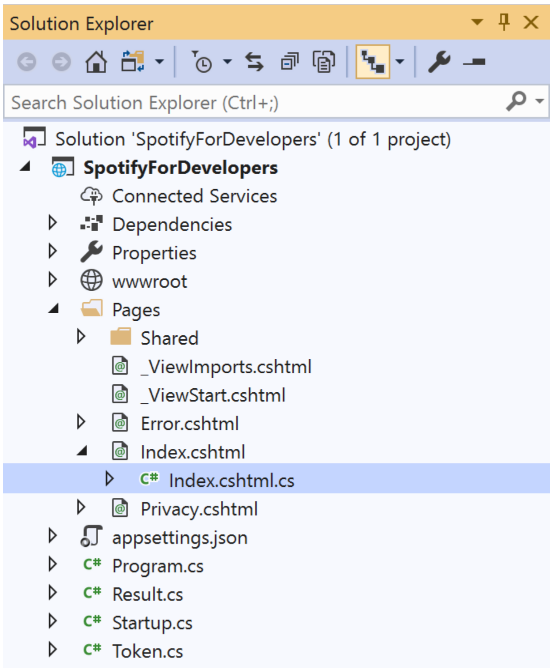


Name	Date modified
.vs	22/01/2020 19:52
SpotifyForDevelopers	01/02/2020 17:18
SpotifyForDevelopers.sln	22/01/2020 19:52

File name: SpotifyForDevelopers.sln
All Project Files (*.sln;*.dsw;*.vcw
□ Do not load projects

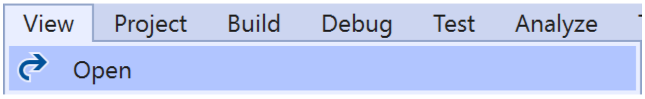
Then locate and select **SpotifyForDevelopers.sln** and select **Open** if you don't have this file already then please follow the previous parts of the workshop including **Getting Started**, **Authorisation Guide**, **Search & Browse**, **Playlists & Artists**, **Albums & Tracks**, **Episodes & Shows**, **Follow**, **Playlists**, **Library** and **Player**

Step 2



Once opened, in the **Solution Explorer** open the **Pages** section, then open the **Index.cshtml** section and select **Index.cshtml.cs**

Step 3



Then from the **Menu** choose **View** and then **Open**

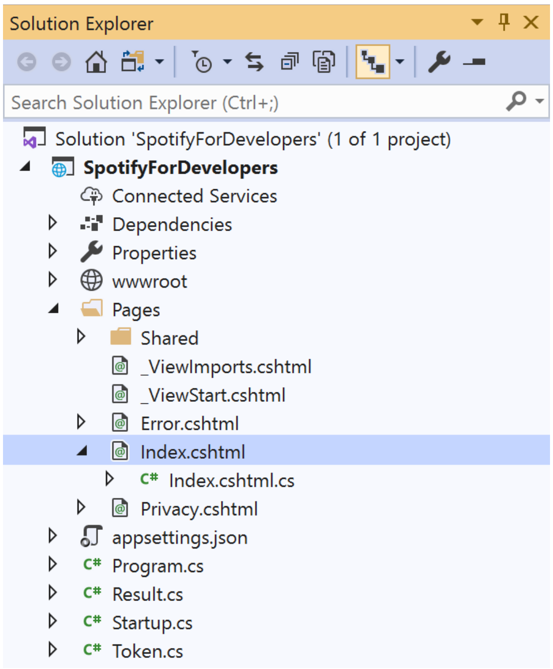
Step 4

In the **Code View** for **Index.cshtml.cs** below the **method** for `public async Task<IActionResult> OnPostUserPlaybackPreviousTrackAsync() { ... }` enter the following **method**:

```
public async Task<IActionResult> OnPostUserPlaybackSeekTrackAsync(string option)
{
    LoadToken();
    var position = (int)TimeSpan.FromSeconds(int.Parse(option)).TotalMilliseconds;
    var result = await Api.UserPlaybackSeekTrackAsync(position);
    if (result != null)
    {
        Results = new List<Result>() { new Result()
        {
            Id = result.Code.ToString(),
            Name = result.Success.ToString()
        }};
    }
    return Page();
}
```

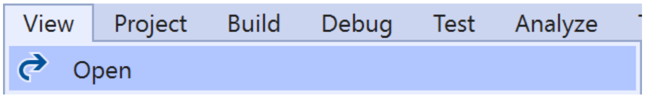
The **method** for `OnPostUserPlaybackSeekTrackAsync` is used to **seek** to the **position** of a currently playing **track** with the `Option` of the **Position Seconds** of the **track** and populate the **property** for `Results` of the **success** of the operation accordingly.

Step 5



In the **Solution Explorer** in the **Pages** section select **Index.cshtml**

Step 6



Then from the **Menu** choose **View** and then **Open**

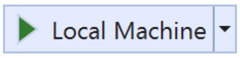
Step 7

Once in the **Code View** for **Index.cshtml** above `<!-- Player -->` enter the following:

```
<li class="list-group-item">
    <form asp-page-handler="UserPlaybackSeekTrack" method="post">
        <input asp-for="Option" placeholder="Position Seconds" class="form-control mb-2" />
        <button class="btn btn-primary mb-2">
            Seek to Position in Currently Playing Track
        </button>
    </form>
</li>
```

This `form` will **post** to the **method** for `UserPlaybackSeekTrack` with the `Option` of **Position Seconds** and will output to the **Results**.

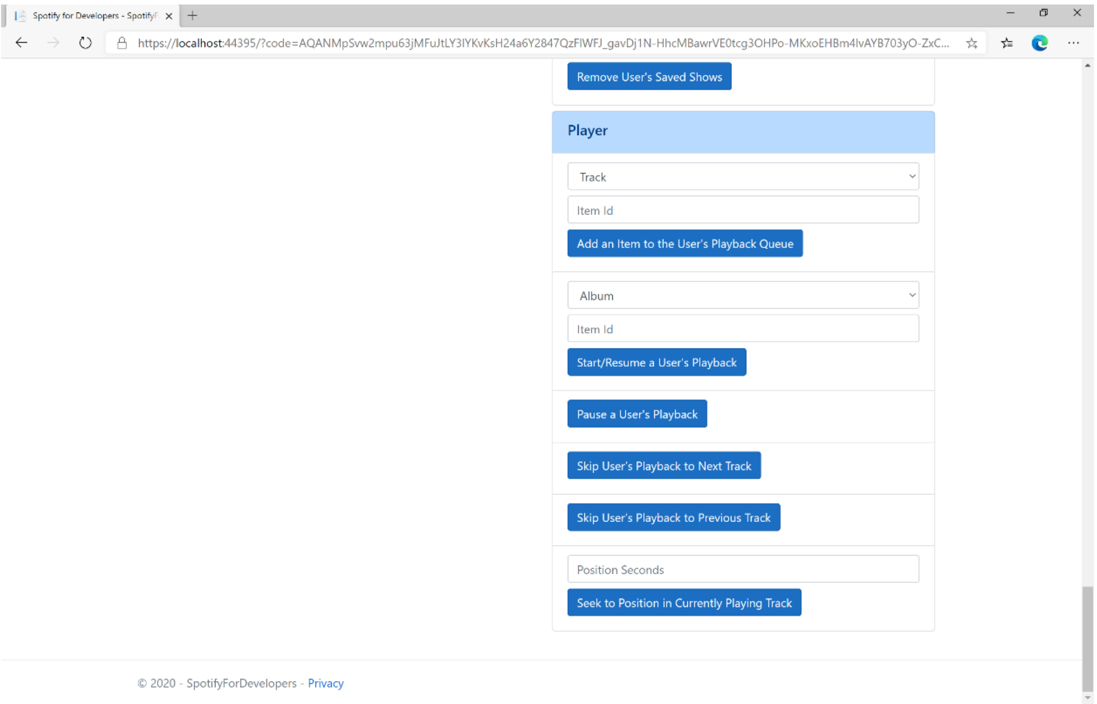
Step 8



Finally, in **Visual Studio 2019** select **IIS Application** to run the **Web Application**

Step 9

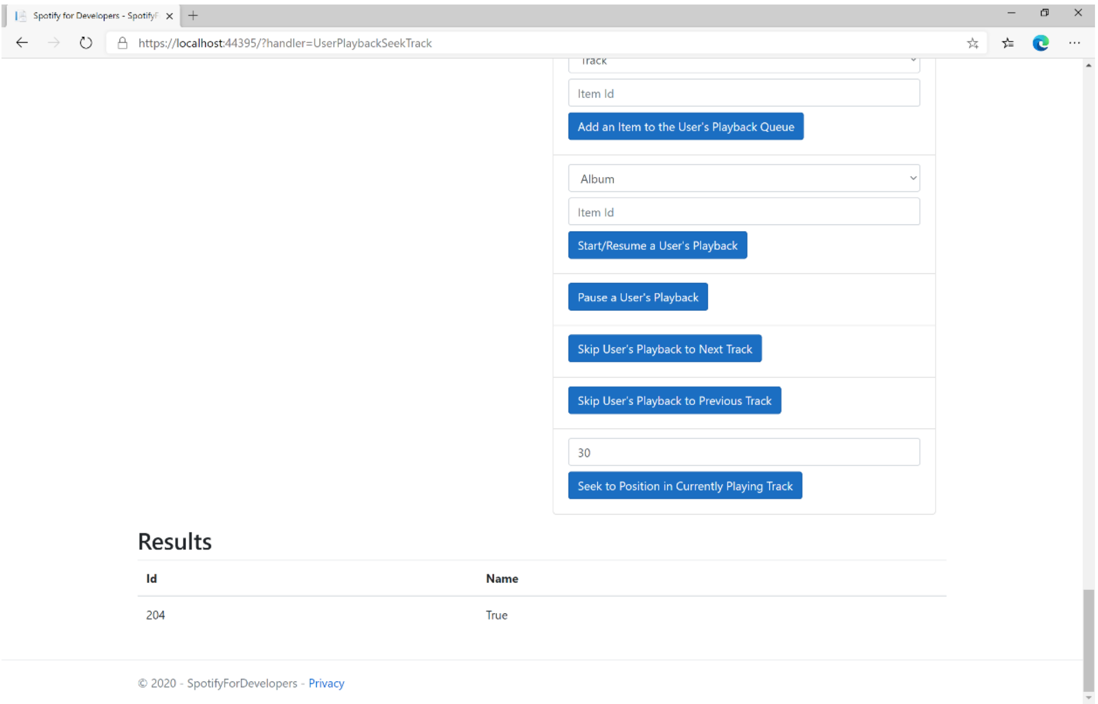
Once the **Web Application** is running and you select **Authorisation Code Flow Login** or **Implicit Grant Flow Login** and scroll down you should see something like the following:



Spotify for Developers - Spotify
https://localhost:44395/?code=AQANMpSvw2mpu63jMFuJtLY3lYKvKsH24a6Y2847QzFlWFJ_gavDj1N-HhcMBawrVE0tcg3OHPo-MKxoEHBm4lvAYB703yO-ZxC...
Remove User's Saved Shows
Player
Track
Item Id
Add an Item to the User's Playback Queue
Album
Item Id
Start/Resume a User's Playback
Pause a User's Playback
Skip User's Playback to Next Track
Skip User's Playback to Previous Track
Position Seconds
Seek to Position in Currently Playing Track
© 2020 - SpotifyForDevelopers - Privacy

Step 10

Make sure **Spotify** is running then you can then enter **Position Seconds** where you'd like **playback** to **seek** to above **Seek to Position in Currently Playing Track** and then select **Seek to Position in Currently Playing Track** and scroll down to view **Results** like the following:



Spotify for Developers - Spotify
https://localhost:44395/?handler=UserPlaybackSeekTrack
Track
Item Id
Add an Item to the User's Playback Queue
Album
Item Id
Start/Resume a User's Playback
Pause a User's Playback
Skip User's Playback to Next Track
Skip User's Playback to Previous Track
30
Seek to Position in Currently Playing Track

Results

Id	Name
204	True

© 2020 - SpotifyForDevelopers - Privacy

Step 11

■

You can stop the **web application** in **Visual Studio 2019** by selecting the **Stop debugging** button

Step 12

×

You can choose to exit **Visual Studio 2019** by selecting the **Close** button in the top right of the **application** as that completes this part of the workshop

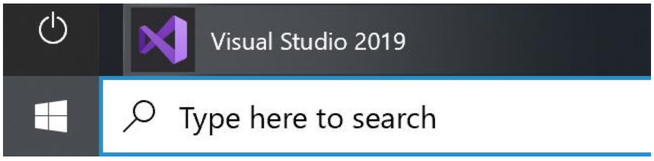
Set Volume for User's Playback

Set the volume for the user's current playback device.

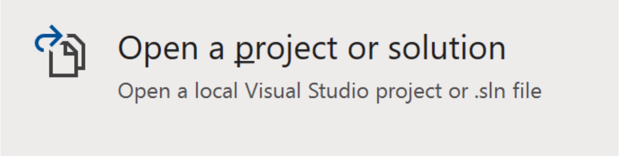
PUT https://api.spotify.com/v1/me/player/volume	
Header	
Authorization	User Token from Spotify Accounts service with **user-modify-playback-state** scope
Query Parameter	
volume_percent	Volume to set - must be a value from 0 to 100
device_id	Device Id to target, if not supplied the user's current active device is the target

Header	Response
Success	
HTTP Status 204 No Content	
Error	
HTTP Status 404 Not Found	NO_ACTIVE_DEVICE - Device was not found
HTTP Status 403 Forbidden	PREMIUM_REQUIRED - Returned when account non-premium
Error Code	Error Object

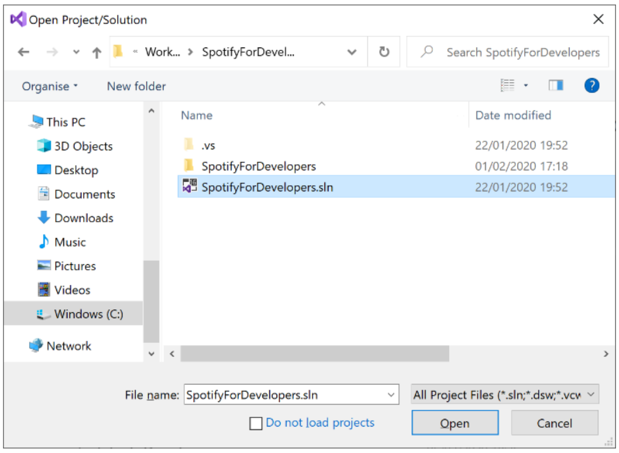
Step 1



If you chose to close **Visual Studio 2019** previously, in **Windows 10** choose **Start**, and then from the **Start Menu** find and select **Visual Studio 2019**


Open a project or solution
Open a local Visual Studio project or .sln file

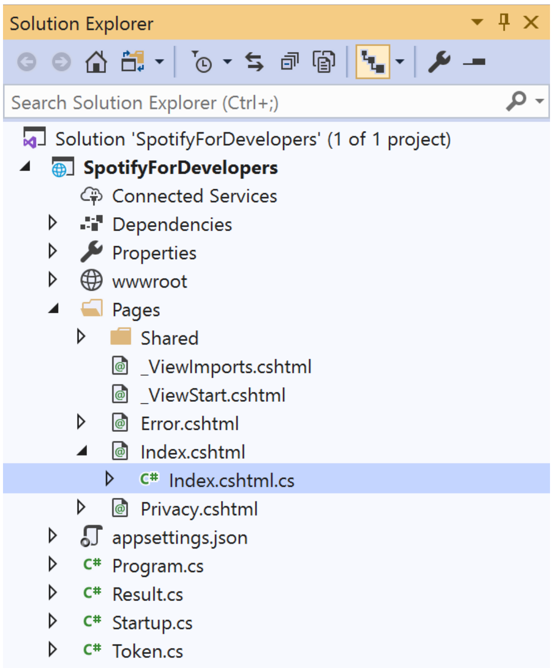
Once done, from the **Get started** screen for **Visual Studio 2019** select **Open a project or solution**



File name: SpotifyForDevelopers.sln

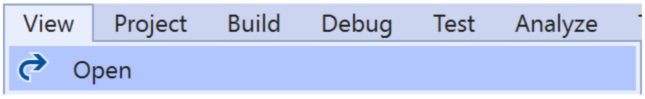
Then locate and select **SpotifyForDevelopers.sln** and select **Open** if you don't have this file already then please follow the previous parts of the workshop including **Getting Started**, **Authorisation Guide**, **Search & Browse**, **Playlists & Artists**, **Albums & Tracks**, **Episodes & Shows**, **Follow**, **Playlists**, **Library** and **Player**

Step 2



Once opened, in the **Solution Explorer** open the **Pages** section, then open the **Index.cshtml** section and select **Index.cshtml.cs**

Step 3



Then from the **Menu** choose **View** and then **Open**

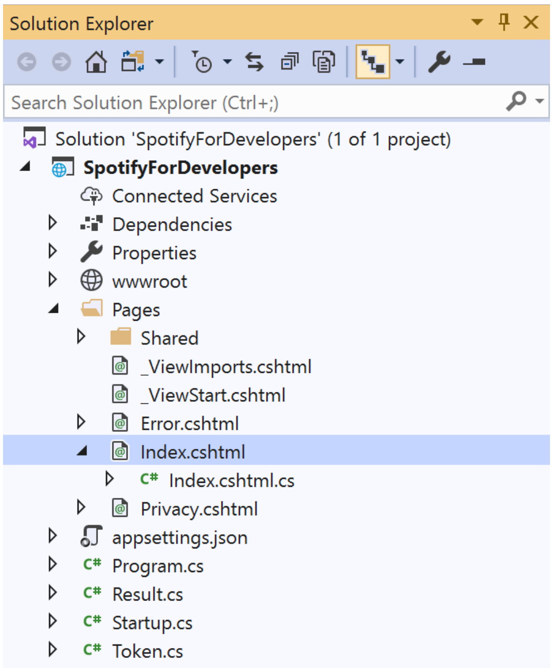
Step 4

In the **Code View** for **Index.cshtml.cs** below the **method** for `public async Task<IActionResult> OnPostUserPlaybackSeekTrackAsync(...) { ... }` enter the following **method**:

```
public async Task<IActionResult> OnPostUserPlaybackSetVolumeAsync(string option)
{
    LoadToken();
    var percent = int.Parse(option);
    var result = await Api.UserPlaybackSetVolumeAsync(percent);
    if (result != null)
    {
        Results = new List<Result>() { new Result()
        {
            Id = result.Code.ToString(),
            Name = result.Success.ToString()
        }};
    }
    return Page();
}
```

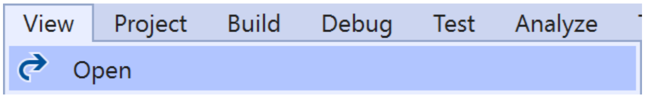
The **method** for `OnPostUserPlaybackSetVolumeAsync` is used set the **volume** of **playback** with the `Option` using a **Range** which by default is **0** to **100** and populate the **property** for `Results` of the **success** of the operation accordingly.

Step 5



In the **Solution Explorer** in the **Pages** section select **Index.cshtml**

Step 6



Then from the **Menu** choose **View** and then **Open**

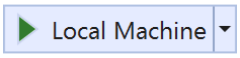
Step 7

Once in the **Code View** for **Index.cshtml** above `<!-- Player -->` enter the following:

```
<li class="list-group-item">
    <form asp-page-handler="UserPlaybackSetVolume" method="post">
        <input type="range" asp-for="Option" class="form-control-range mb-2" />
        <button class="btn btn-primary mb-2">
            Set Volume for User's Playback
        </button>
    </form>
</li>
```

This `form` will **post** to the **method** for `UserPlaybackSetVolume` with the `Option` of the **Range** selected and will output to the **Results**.

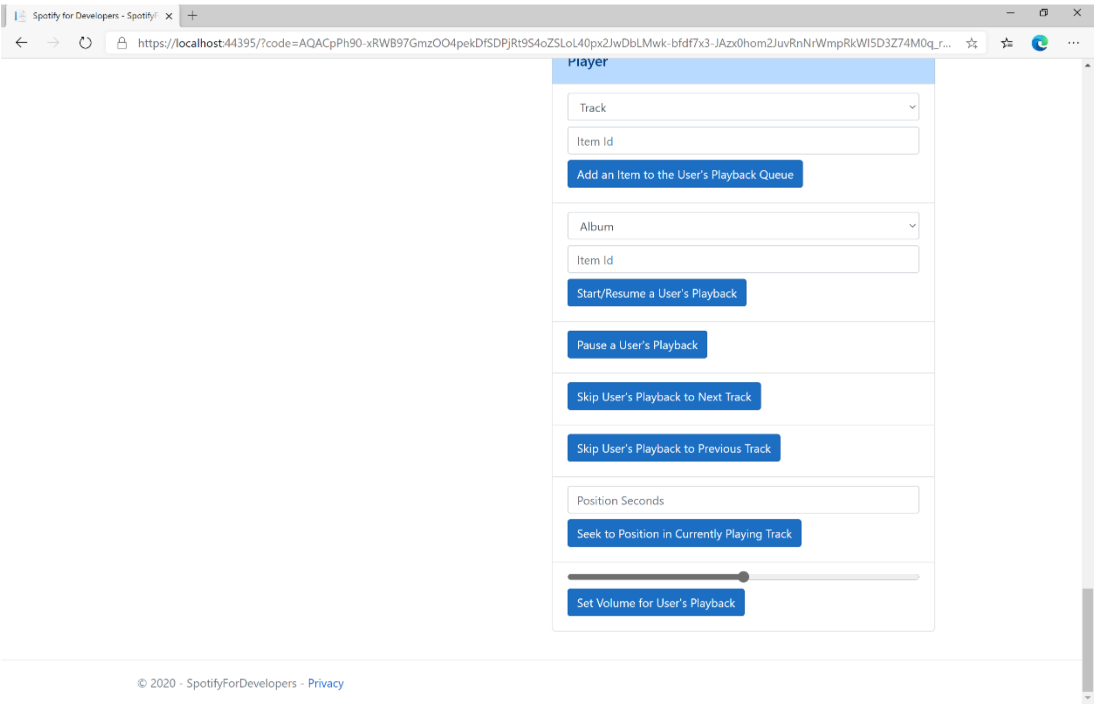
Step 8



Finally, in **Visual Studio 2019** select **IIS Application** to run the **Web Application**

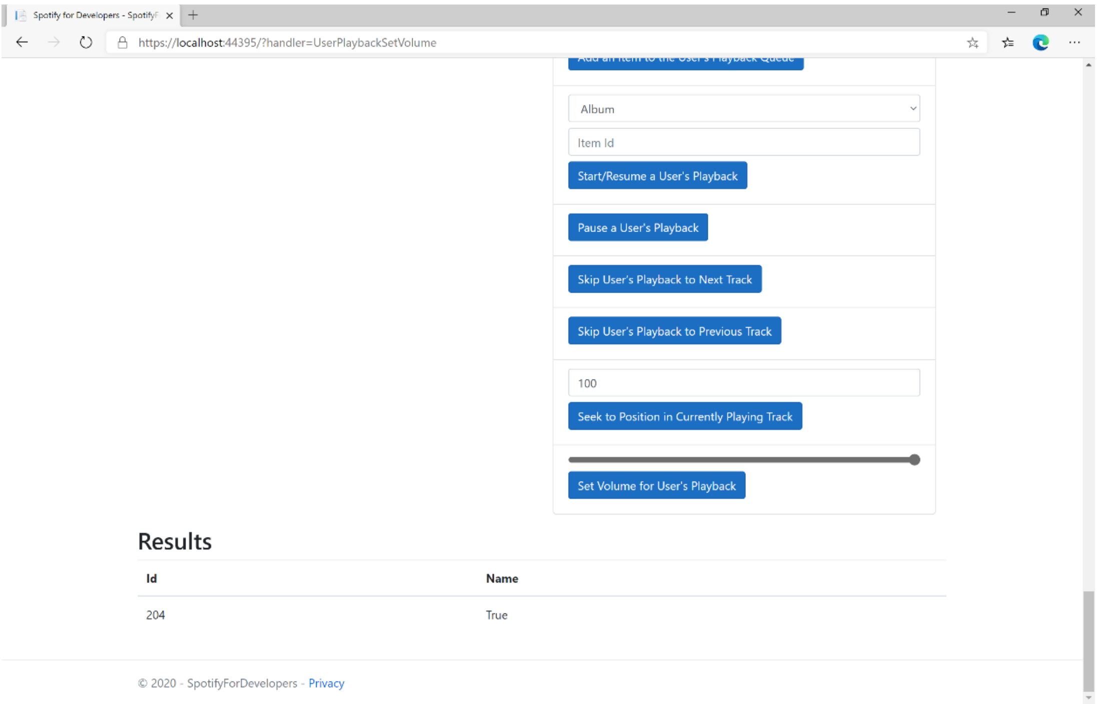
Step 9

Once the **Web Application** is running and you select **Authorisation Code Flow Login** or **Implicit Grant Flow Login** and scroll down you should see something like the following:


Spotify for Developers - Spotify
https://localhost:44395/?code=AQACpPh90-xRWB97GmzOO4pekDfSDPjRt9S4oZSLoL40px2JwDbLMwk-bfdf7x3-JAzx0hom2JuvRnNrWmpRkWI5D3Z74M0q_r...
Player
Track
Item Id
Add an Item to the User's Playback Queue
Album
Item Id
Start/Resume a User's Playback
Pause a User's Playback
Skip User's Playback to Next Track
Skip User's Playback to Previous Track
Position Seconds
Seek to Position in Currently Playing Track
Set Volume for User's Playback
© 2020 - SpotifyForDevelopers - Privacy

Step 10

Make sure **Spotify** is running then you can then use the **Range** to set the **volume** above **Set Volume for User's Playback** and then select **Set Volume for User's Playback** and scroll down to view **Results** like the following:


Spotify for Developers - Spotify
https://localhost:44395/?handler=UserPlaybackSetVolume
Add an Item to the User's Playback Queue
Album
Item Id
Start/Resume a User's Playback
Pause a User's Playback
Skip User's Playback to Next Track
Skip User's Playback to Previous Track
100
Seek to Position in Currently Playing Track
Set Volume for User's Playback
© 2020 - SpotifyForDevelopers - Privacy

Results

Id	Name
204	True

Step 11

You can stop the **web application** in **Visual Studio 2019** by selecting the **Stop debugging** button

Step 12

You can choose to exit **Visual Studio 2019** by selecting the **Close** button in the top right of the **application** as that completes this part of the workshop

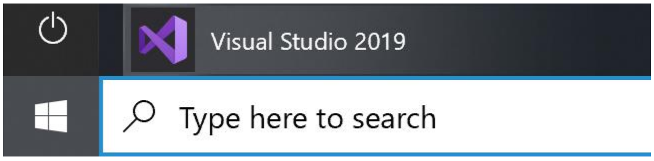
Toggle Shuffle for User's Playback

Toggle shuffle on or off for user's playback.

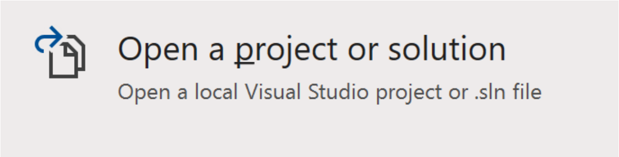
PUT https://api.spotify.com/v1/me/player/shuffle	
Header	
Authorization	User Token from Spotify Accounts service with **user-modify-playback-state** scope
Query Parameter	
state	If true, shuffle user's playback, if false do not shuffle user's playback
device_id	Device Id to target, if not supplied the user's current active device is the target

Header	Response
Success	
HTTP Status 204 No Content	
Error	
HTTP Status 404 Not Found	NO_ACTIVE_DEVICE - Device was not found
HTTP Status 403 Forbidden	PREMIUM_REQUIRED - Returned when account non-premium
Error Code	Error Object

Step 1



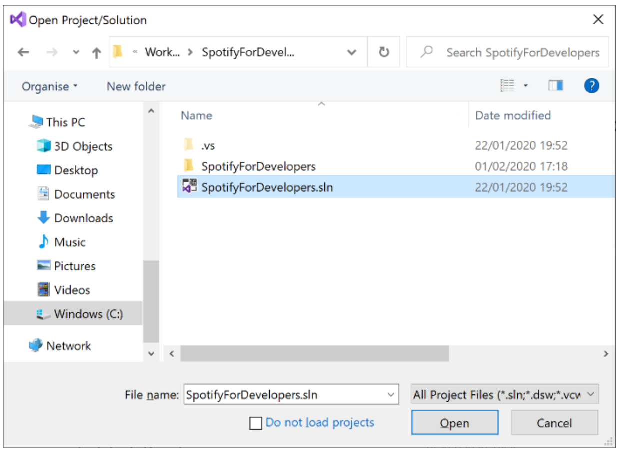
If you chose to close **Visual Studio 2019** previously, in **Windows 10** choose **Start**, and then from the **Start Menu** find and select **Visual Studio 2019**



Open a project or solution

Open a local Visual Studio project or .sln file

Once done, from the **Get started** screen for **Visual Studio 2019** select **Open a project or solution**



Name	Date modified
.vs	22/01/2020 19:52
SpotifyForDevelopers	01/02/2020 17:18
SpotifyForDevelopers.sln	22/01/2020 19:52

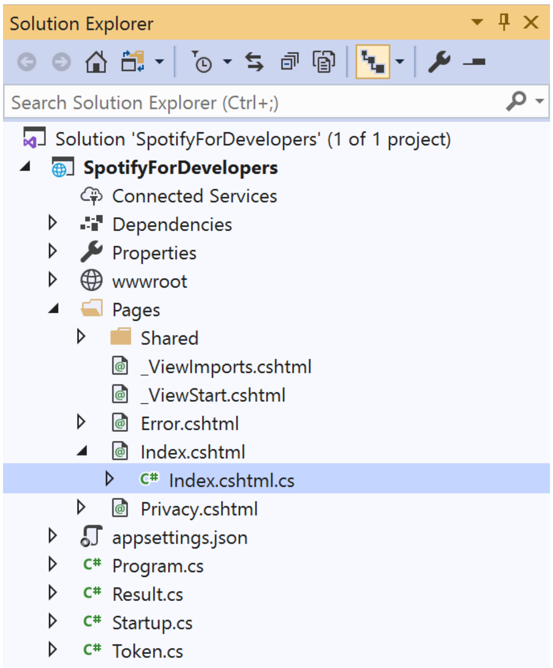
File name: SpotifyForDevelopers.sln

All Project Files (*.sln;*.dsw;*.vcw

☐ Do not load projects

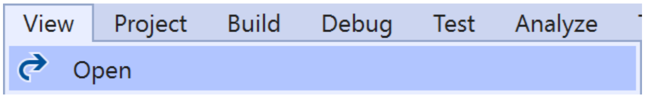
Then locate and select **SpotifyForDevelopers.sln** and select **Open** if you don't have this file already then please follow the previous parts of the workshop including **Getting Started**, **Authorisation Guide**, **Search & Browse**, **Playlists & Artists**, **Albums & Tracks**, **Episodes & Shows**, **Follow**, **Playlists**, **Library** and **Player**

Step 2



Once opened, in the **Solution Explorer** open the **Pages** section, then open the **Index.cshtml** section and select **Index.cshtml.cs**

Step 3



Then from the **Menu** choose **View** and then **Open**

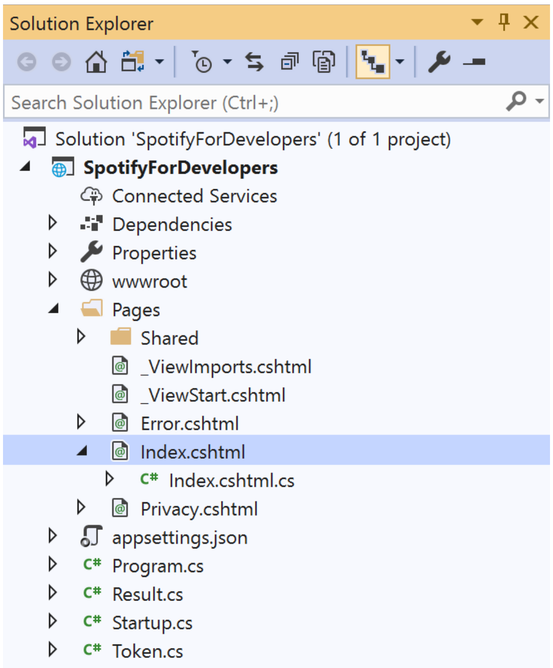
Step 4

In the **Code View** for **Index.cshtml.cs** below the **method** for `public async Task<IActionResult> OnPostUserPlaybackSetVolumeAsync(...) { ... }` enter the following **method**:

```
public async Task<IActionResult> OnPostUserPlaybackToggleShuffleAsync(bool flag)
{
    LoadToken();
    var result = await Api.UserPlaybackToggleShuffleAsync(flag);
    if (result != null)
    {
        Results = new List<Result>() { new Result()
        {
            Id = result.Code.ToString(),
            Name = result.Success.ToString()
        }};
    }
    return Page();
}
```

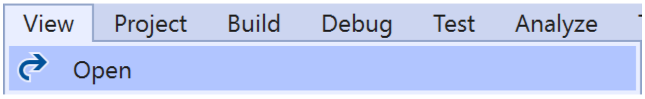
The **method** for `OnPostUserPlaybackToggleShuffleAsync` is used set the **shuffle** mode of the **playback** with the `Flag` of **true** or **false** and populate the **property** for `Results` of the **success** of the operation accordingly.

Step 5



In the **Solution Explorer** in the **Pages** section select **Index.cshtml**

Step 6



Then from the **Menu** choose **View** and then **Open**

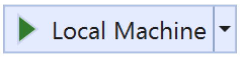
Step 7

Once in the **Code View** for **Index.cshtml** above `<!-- Player -->` enter the following:

```
<li class="list-group-item">
    <form asp-page-handler="UserPlaybackToggleShuffle" method="post">
        <input type="checkbox" asp-for="Flag" class="form-control mb-2" />
        <button class="btn btn-primary mb-2">
            Toggle Shuffle for User’s Playback
        </button>
    </form>
</li>
```

This `form` will **post** to the **method** for `UserPlaybackToggleShuffle` with the `Flag` of the **Checkbox** and will output to the **Results**.

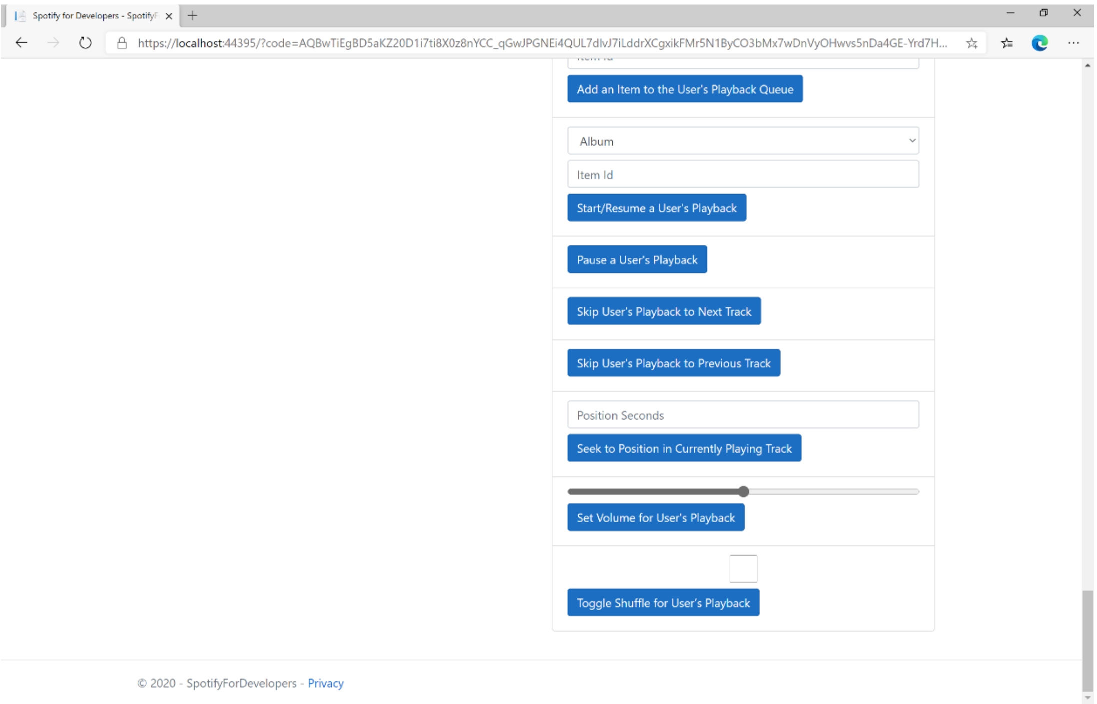
Step 8



Finally, in **Visual Studio 2019** select **IIS Application** to run the **Web Application**

Step 9

Once the **Web Application** is running and you select **Authorisation Code Flow Login** or **Implicit Grant Flow Login** and scroll down you should see something like the following:



Spotify for Developers - SpotifyF

https://localhost:44395/?code=AQBwTiEgBD5aKZ20D1i7ti8X0z8nYCC_qGwJPGNEi4QUL7dlvJ7iLddrXCgxikFMr5N1ByCO3bMx7wDnVyOHwvs5nDa4GE-Yrd7H...

Add an Item to the User's Playback Queue

Album

Item Id

Start/Resume a User's Playback

Pause a User's Playback

Skip User's Playback to Next Track

Skip User's Playback to Previous Track

Position Seconds

Seek to Position in Currently Playing Track

Set Volume for User's Playback

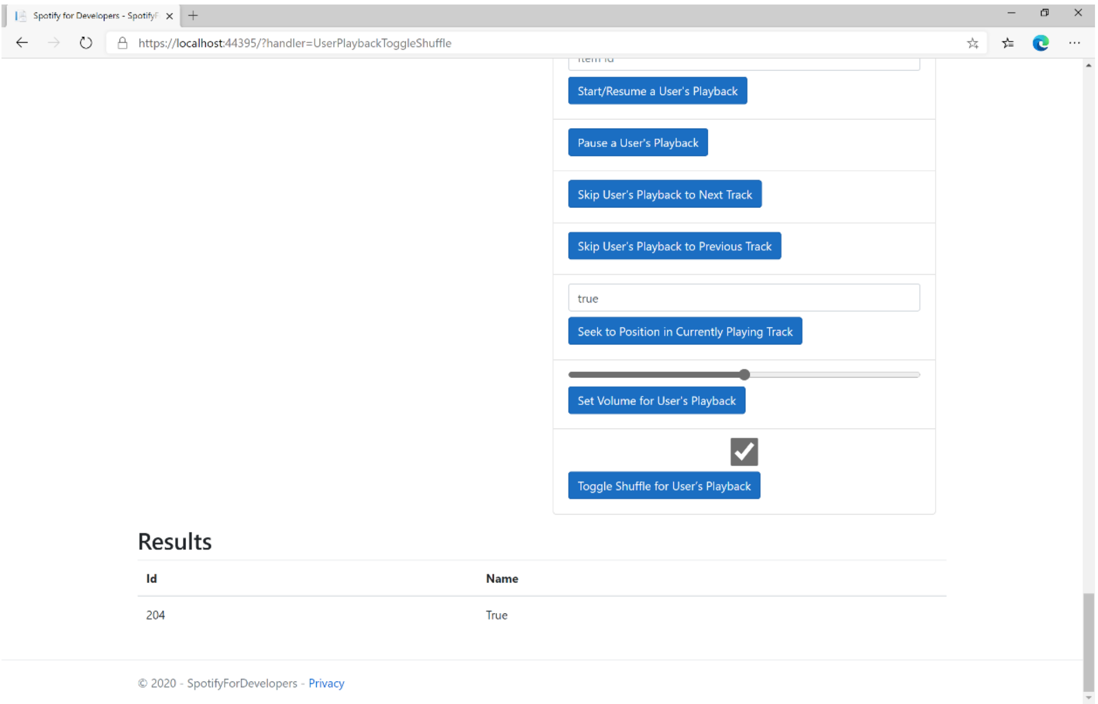
□

Toggle Shuffle for User's Playback

© 2020 - SpotifyForDevelopers - Privacy

Step 10

Make sure **Spotify** is running then you can then **tick** the **Checkbox** above **Toggle Shuffle for User's Playback** to enable **shuffle** and **untick** the **Checkbox** to disable **shuffle** and select **Toggle Shuffle for User's Playback** and scroll down to view **Results** like the following:



Spotify for Developers - SpotifyF

https://localhost:44395/?handler=UserPlaybackToggleShuffle

Start/Resume a User's Playback

Pause a User's Playback

Skip User's Playback to Next Track

Skip User's Playback to Previous Track

true

Seek to Position in Currently Playing Track

Set Volume for User's Playback

☑

Toggle Shuffle for User's Playback

Results

Id	Name
204	True

© 2020 - SpotifyForDevelopers - Privacy

Step 11

■

You can stop the **web application** in **Visual Studio 2019** by selecting the **Stop debugging** button

Step 12

×

You can choose to exit **Visual Studio 2019** by selecting the **Close** button in the top right of the **application** as that completes this part of the workshop

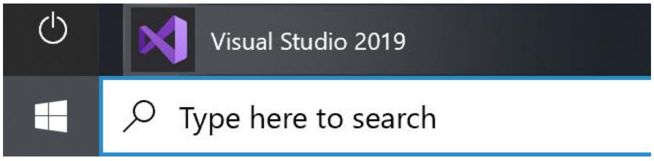
Set Repeat Mode on User’s Playback

Set the repeat mode for the user’s playback. Options are repeat-track, repeat-context, and off.

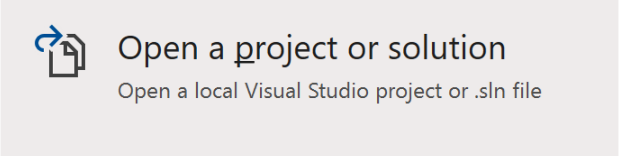
PUT https://api.spotify.com/v1/me/player/repeat	
Header	
Authorization	User Token from Spotify Accounts service with **user-modify-playback-state** scope
Query Parameter	
state	track will repeat current track, context will repeat current context and off will turn repeat off.
device_id	Device Id to target, if not supplied the user’s current active device is the target

Header	Response
Success	
HTTP Status 204 No Content	
Error	
HTTP Status 404 Not Found	NO_ACTIVE_DEVICE - Device was not found
HTTP Status 403 Forbidden	PREMIUM_REQUIRED - Returned when account non-premium
Error Code	Error Object

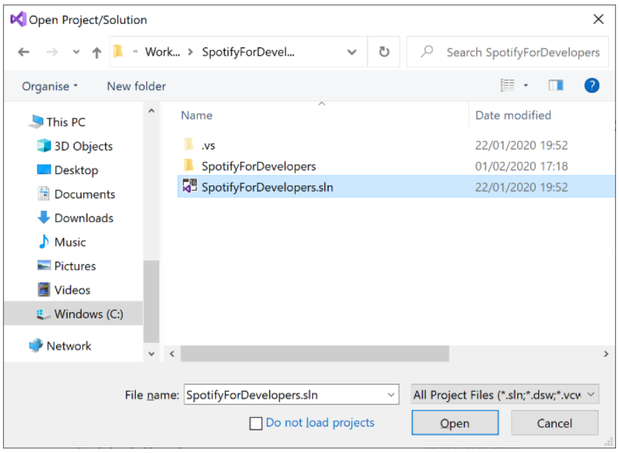
Step 1



If you chose to close **Visual Studio 2019** previously, in **Windows 10** choose **Start**, and then from the **Start Menu** find and select **Visual Studio 2019**


Open a project or solution
Open a local Visual Studio project or .sln file

Once done, from the **Get started** screen for **Visual Studio 2019** select **Open a project or solution**

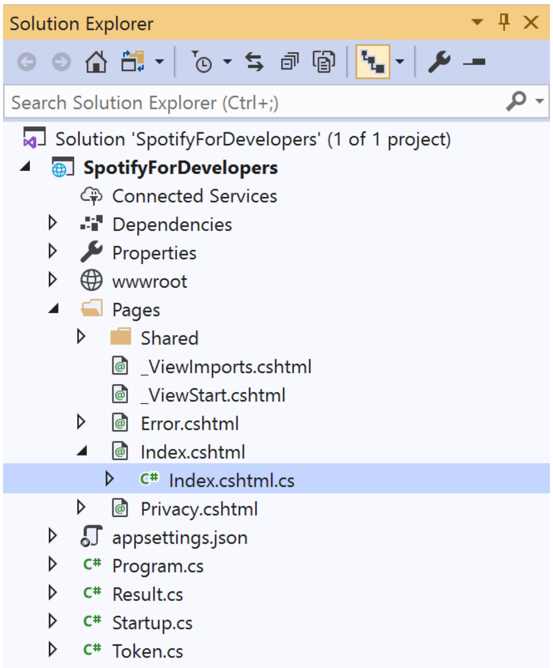


Name	Date modified
.vs	22/01/2020 19:52
SpotifyForDevelopers	01/02/2020 17:18
SpotifyForDevelopers.sln	22/01/2020 19:52

File name: SpotifyForDevelopers.sln

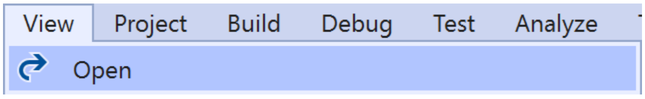
Then locate and select **SpotifyForDevelopers.sln** and select **Open** if you don’t have this file already then please follow the previous parts of the workshop including **Getting Started**, **Authorisation Guide**, **Search & Browse**, **Playlists & Artists**, **Albums & Tracks**, **Episodes & Shows**, **Follow**, **Playlists**, **Library** and **Player**

Step 2



Once opened, in the **Solution Explorer** open the **Pages** section, then open the **Index.cshtml** section and select **Index.cshtml.cs**

Step 3



Then from the **Menu** choose **View** and then **Open**

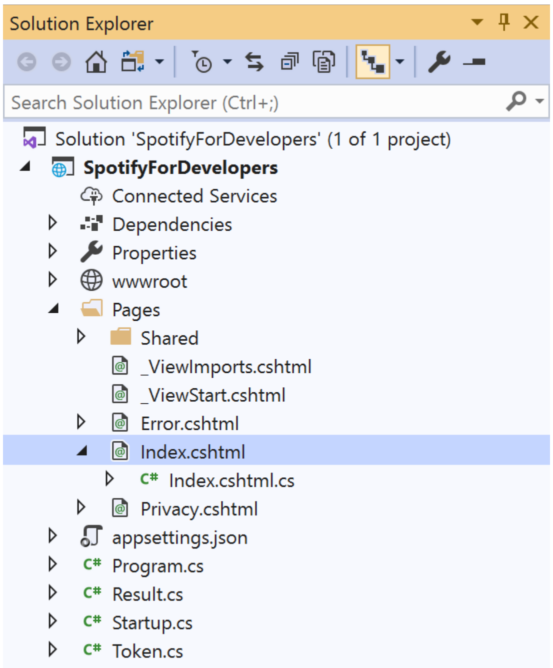
Step 4

In the **Code View** for **Index.cshtml.cs** below the **method** for `public async Task<IActionResult> OnPostUserPlaybackToggleShuffleAsync(...) { ... }` enter the following **method**:

```
public async Task<IActionResult> OnPostUserPlaybackSetRepeatModeAsync(bool flag)
{
    var state = flag ? RepeatState.Track : RepeatState.Off;
    var result = await Api.UserPlaybackSetRepeatModeAsync(state);
    if (result != null)
    {
        Results = new List<Result>() { new Result()
        {
            Id = result.Code.ToString(),
            Name = result.Success.ToString()
        }};
    }
    return Page();
}
```

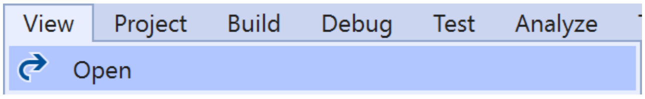
The **method** for `OnPostUserPlaybackSetRepeatModeAsync` is used set the **repeat** mode of the **playback** with the `Flag` of **true** or **false** and populate the **property** for `Results` of the **success** of the operation accordingly.

Step 5



In the **Solution Explorer** in the **Pages** section select **Index.cshtml**

Step 6



Then from the **Menu** choose **View** and then **Open**

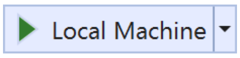
Step 7

Once in the **Code View** for **Index.cshtml** above `<!-- Player -->` enter the following:

```
<li class="list-group-item">
    <form asp-page-handler="UserPlaybackSetRepeatMode" method="post">
        <input type="checkbox" asp-for="Flag" class="form-control mb-2" />
        <button class="btn btn-primary mb-2">
            Set Repeat Mode on User’s Playback
        </button>
    </form>
</li>
```

This `form` will **post** to the **method** for `UserPlaybackSetRepeatMode` with the `Flag` of the **Checkbox** and will output to the **Results**.

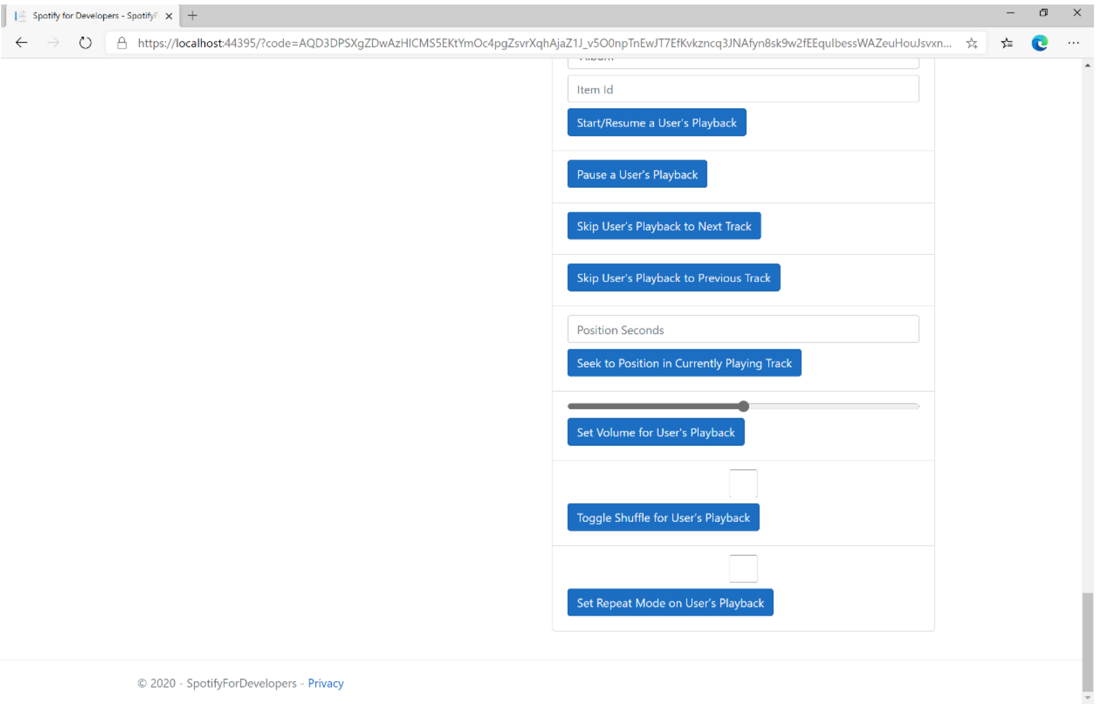
Step 8



Finally, in **Visual Studio 2019** select **IIS Application** to run the **Web Application**

Step 9

Once the **Web Application** is running and you select **Authorisation Code Flow Login** or **Implicit Grant Flow Login** and scroll down you should see something like the following:

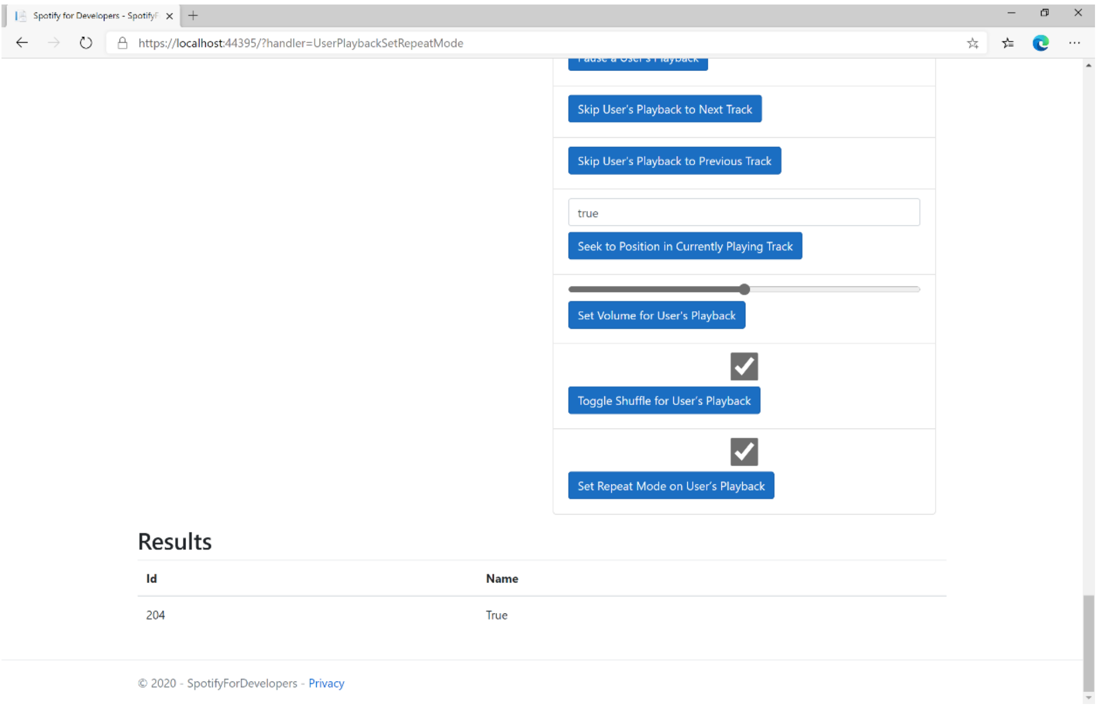


https://localhost:44395/?code=AQD3DPSXgZDwAzHICMS5EKtYmOc4pgZsvrXqhAjaZ1J_v5O0npTnEwJT7EfKvkzncq3JNAfyn8sk9w2fEEquIbessWAZeuHouJsvxn...

Spotify for Developers - SpotifyF
Item Id
Start/Resume a User's Playback
Pause a User's Playback
Skip User's Playback to Next Track
Skip User's Playback to Previous Track
Position Seconds
Seek to Position in Currently Playing Track
Set Volume for User's Playback
Toggle Shuffle for User's Playback
Set Repeat Mode on User's Playback
© 2020 - SpotifyForDevelopers - Privacy

Step 10

Make sure **Spotify** is running then you can then **tick** the **Checkbox** above **Set Repeat Mode on User's Playback** to enable **repeat** and **untick** the **Checkbox** to disable **repeat** and select **Set Repeat Mode on User's Playback** and scroll down to view **Results** like the following:



https://localhost:44395/?handler=UserPlaybackSetRepeatMode

Spotify for Developers - SpotifyF
Skip User's Playback to Next Track
Skip User's Playback to Previous Track
Seek to Position in Currently Playing Track
Set Volume for User's Playback
Toggle Shuffle for User's Playback
Set Repeat Mode on User's Playback
© 2020 - SpotifyForDevelopers - Privacy

true

☑ Toggle Shuffle for User's Playback

☑ Set Repeat Mode on User's Playback

Results

Id	Name
204	True

Step 11

■

You can stop the **web application** in **Visual Studio 2019** by selecting the **Stop debugging** button

Step 12

×

You can choose to exit **Visual Studio 2019** by selecting the **Close** button in the top right of the **application** as that completes this part of the workshop

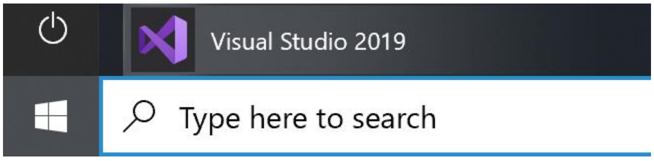
Get Information about the User's Current Playback

Get information about the user's current playback state, including track, track progress, and active device.

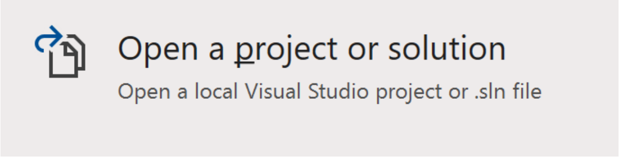
GET https://api.spotify.com/v1/me/player	
Header	
Authorization	User Token from Spotify Accounts service with **user-read-playback-state** scope
Query Parameter	
market	ISO 3166-1 alpha-2 country code. Provide to apply Track Relinking
additional_types	Comma-separated list of item types that your client supports besides the default track type. Valid types are track and episode. An unsupported type in the response is expected to be represented as null value in the item field

Header	**Response**
Success	
HTTP Status 200 OK	Currently Playing Object
Error	
Error Code	Error Object

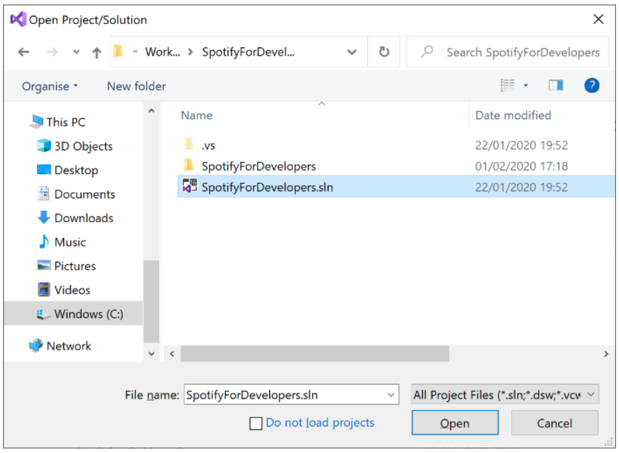
Step 1



If you chose to close **Visual Studio 2019** previously, in **Windows 10** choose **Start**, and then from the **Start Menu** find and select **Visual Studio 2019**


Open a project or solution
Open a local Visual Studio project or .sln file

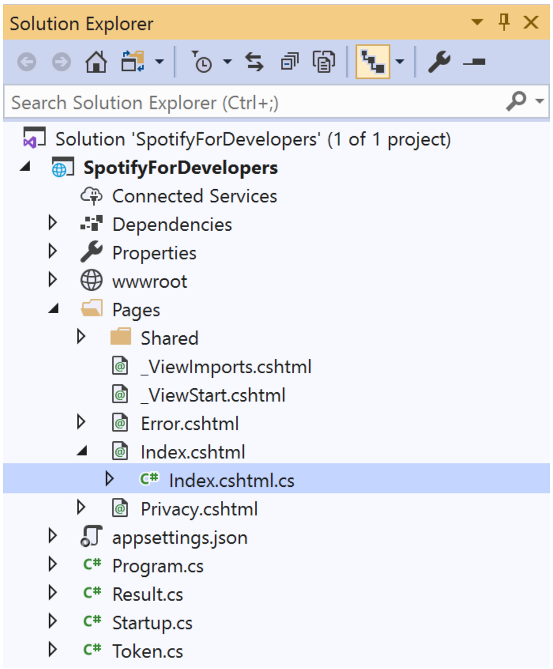
Once done, from the **Get started** screen for **Visual Studio 2019** select **Open a project or solution**



File name: SpotifyForDevelopers.sln

Then locate and select **SpotifyForDevelopers.sln** and select **Open** if you don't have this file already then please follow the previous parts of the workshop including **Getting Started**, **Authorisation Guide**, **Search & Browse**, **Playlists & Artists**, **Albums & Tracks**, **Episodes & Shows**, **Follow**, **Playlists**, **Library** and **Player**

Step 2



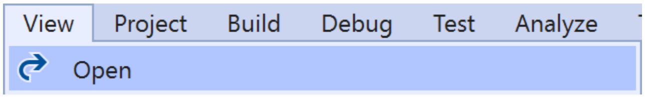
Solution Explorer

Search Solution Explorer (Ctrl+;)

- Solution 'SpotifyForDevelopers' (1 of 1 project)
  - **SpotifyForDevelopers**
    - Connected Services
    - Dependencies
    - Properties
    - wwwroot
    - Pages
      - Shared
      - _ViewImports.cshtml
      - _ViewStart.cshtml
      - Error.cshtml
      - Index.cshtml
        - Index.cshtml.cs
      - Privacy.cshtml
    - appsettings.json
    - Program.cs
    - Result.cs
    - Startup.cs
    - Token.cs

Once opened, in the **Solution Explorer** open the **Pages** section, then open the **Index.cshtml** section and select **Index.cshtml.cs**

Step 3



Then from the **Menu** choose **View** and then **Open**

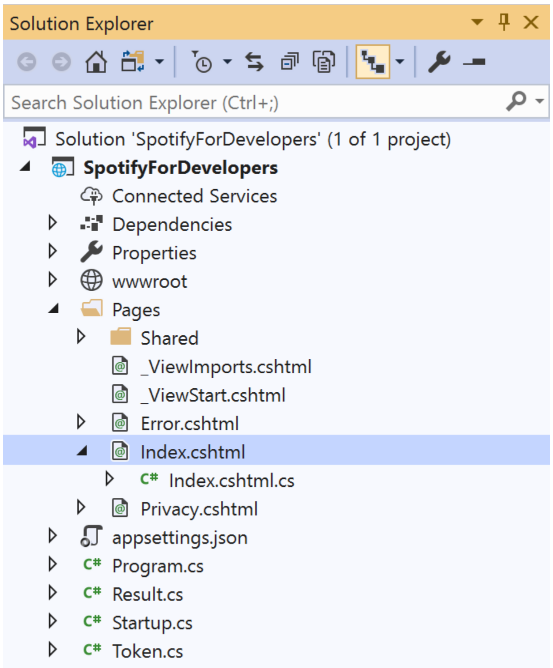
Step 4

In the **Code View** for **Index.cshtml.cs** below the **method** for `public async Task<IActionResult> OnPostUserPlaybackSetRepeatModeAsync(...) { ... }` enter the following **method**:

```
public async Task<IActionResult> OnPostGetUserPlaybackCurrentAsync()
{
    LoadToken();
    var result = await Api.GetUserPlaybackCurrentAsync(country,
        new List<string> { "track", "episode" });
    if (result != null)
    {
        if(result.Track != null)
        {
            Results = new List<Result> { new Result()
            {
                Id = result?.Track?.Id,
                Name = result?.Track?.Name,
                Image = result?.Track.Album?.Images?.FirstOrDefault()?.Url,
                Inner = new Result()
                {
                    Id = result?.Track?.Artists?.FirstOrDefault()?.Id,
                    Name = result?.Track?.Artists?.FirstOrDefault()?.Name
                }
            }};
        }
        if (result.Episode != null)
        {
            Results = new List<Result> { new Result()
            {
                Id = result?.Episode?.Id,
                Name = result?.Episode?.Name,
                Image = result?.Episode?.Images?.FirstOrDefault()?.Url,
                Inner = new Result()
                {
                    Id = result?.Episode?.Show?.Id,
                    Name = result?.Episode?.Show?.Name
                }
            }};
        }
    }
    return Page();
}
```

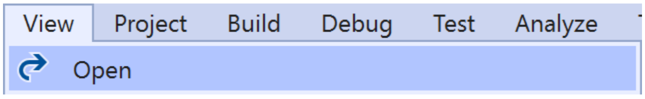
The **method** for `OnPostGetUserPlaybackCurrentAsync` is used get **information** the currently playing **track** or **episode** and populate the **property** for `Results` accordingly.

Step 5



In the **Solution Explorer** in the **Pages** section select **Index.cshtml**

Step 6



Then from the **Menu** choose **View** and then **Open**

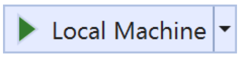
Step 7

Once in the **Code View** for **Index.cshtml** above `<!-- Player -->` enter the following:

```
<li class="list-group-item">
    <form asp-page-handler="GetUserPlaybackCurrent" method="post">
        <button class="btn btn-primary mb-2">
            Get Information about the User's Current Playback
        </button>
    </form>
</li>
```

This `form` will **post** to the **method** for `GetUserPlaybackCurrent` and will output to the **Results**.

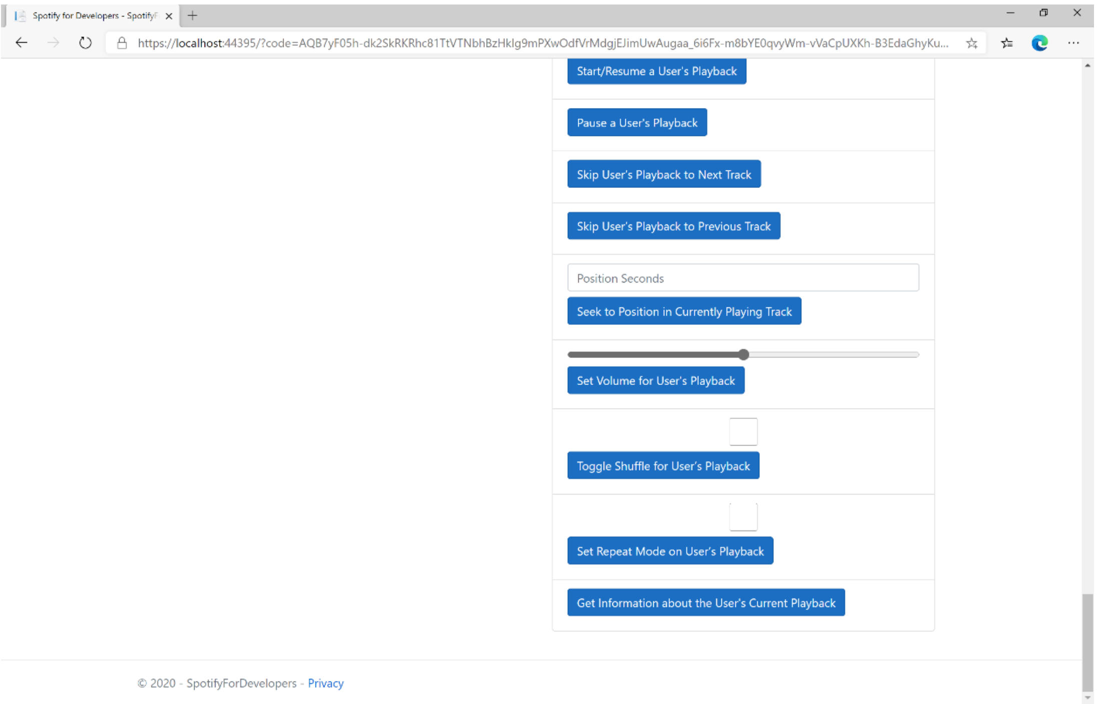
Step 8



Finally, in **Visual Studio 2019** select **IIS Application** to run the **Web Application**

Step 9

Once the **Web Application** is running and you select **Authorisation Code Flow Login** or **Implicit Grant Flow Login** and scroll down you should see something like the following:

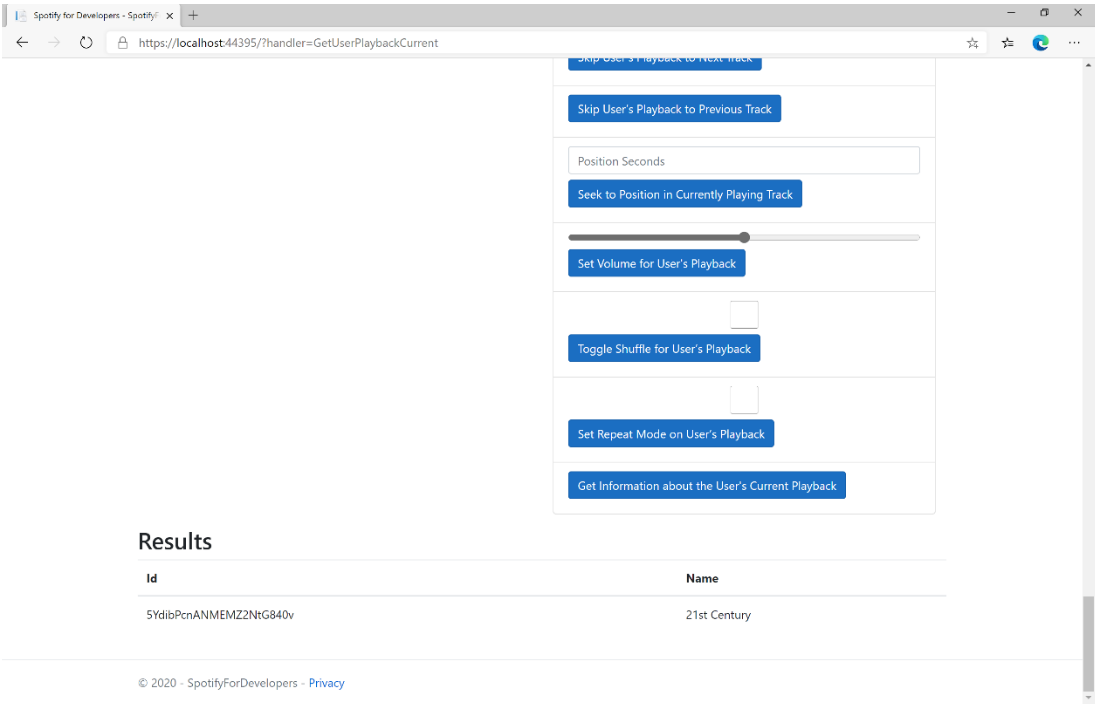

Spotify for Developers - SpotifyF
Start/Resume a User's Playback
Pause a User's Playback
Skip User's Playback to Next Track
Skip User's Playback to Previous Track
Position Seconds
Seek to Position in Currently Playing Track
Set Volume for User's Playback
Toggle Shuffle for User's Playback
Set Repeat Mode on User's Playback
Get Information about the User's Current Playback

https://localhost:44395/?code=AQB7yF05h-dk2SkRKRhc81TtVTNbhBzHklg9mPXwOdfVrMdgjEJimUwAugaa_6i6Fx-m8bYE0qvyWm-vVaCpUXKh-B3EdaGhyKu...

© 2020 - SpotifyForDevelopers - Privacy

Step 10

Make sure **Spotify** is running then you can then use **Start/Resume a User's Playback** with a **Track Id** or **Episode Id** from **Search For An Item** with option of **Track** or **Episode** then select **Get Information about the User's Current Playback** and scroll down to view **Results** like the following:


Spotify for Developers - SpotifyF
Skip User's Playback to Previous Track
Position Seconds
Seek to Position in Currently Playing Track
Set Volume for User's Playback
Toggle Shuffle for User's Playback
Set Repeat Mode on User's Playback
Get Information about the User's Current Playback

https://localhost:44395/?handler=GetUserPlaybackCurrent

Results

Id	Name
5YdibPcnANMEMZ2NtG840v	21st Century

© 2020 - SpotifyForDevelopers - Privacy

Step 11

You can stop the **web application** in **Visual Studio 2019** by selecting the **Stop debugging** button

Step 12

You can choose to exit **Visual Studio 2019** by selecting the **Close** button in the top right of the **application** as that completes this part of the workshop

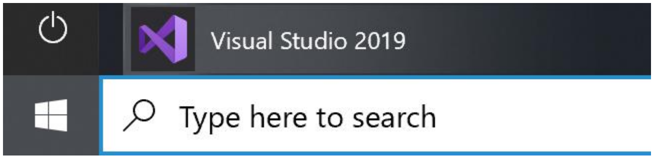
Get the User's Currently Playing Track

Get the object currently being played on the user's Spotify account.

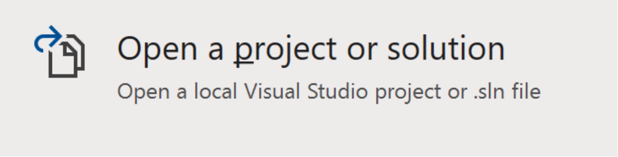
GET https://api.spotify.com/v1/me/player/currently-playing	
Header	
Authorization	User Token from Spotify Accounts service with **user-read-currently-playing** and/or **user-read-playback-state** scope
Query Parameter	
market	ISO 3166-1 alpha-2 country code. Provide to apply Track Relinking
additional_types	Comma-separated list of item types that your client supports besides the default track type. Valid types are track and episode. An unsupported type in the response is expected to be represented as null value in the item field

Header	**Response**
Success	
HTTP Status 200 OK	Currently Playing Track or Episode and Context
HTTP Status 204 No Content	No Payload if nothing playing or if private session an empty payload
Error	
Error Code	Error Object

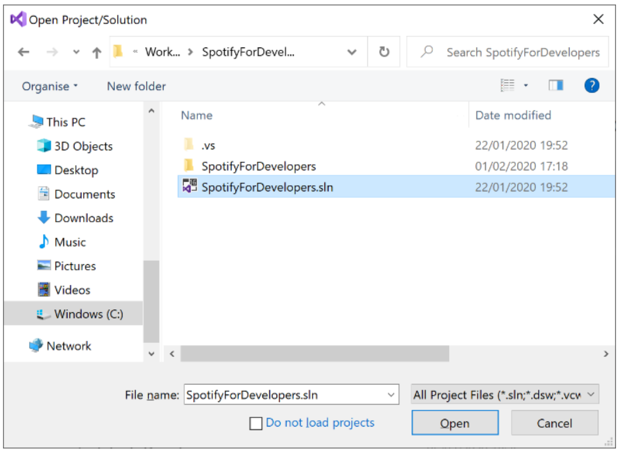
Step 1



If you chose to close **Visual Studio 2019** previously, in **Windows 10** choose **Start**, and then from the **Start Menu** find and select **Visual Studio 2019**


Open a project or solution
Open a local Visual Studio project or .sln file

Once done, from the **Get started** screen for **Visual Studio 2019** select **Open a project or solution**



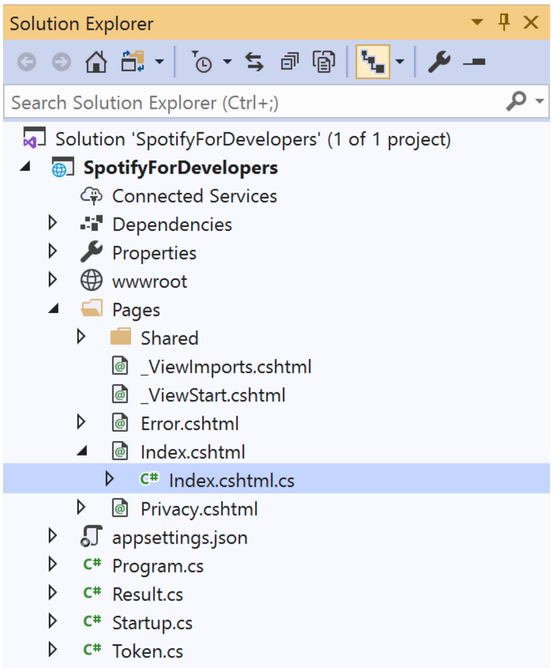
Name	Date modified
.vs	22/01/2020 19:52
SpotifyForDevelopers	01/02/2020 17:18
SpotifyForDevelopers.sln	22/01/2020 19:52

File name: SpotifyForDevelopers.sln

All Project Files (*.sln;*.dsw;*.vcw

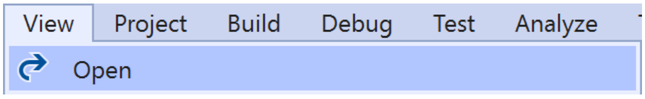
Then locate and select **SpotifyForDevelopers.sln** and select **Open** if you don't have this file already then please follow the previous parts of the workshop including **Getting Started**, **Authorisation Guide**, **Search & Browse**, **Playlists & Artists**, **Albums & Tracks**, **Episodes & Shows**, **Follow**, **Playlists**, **Library** and **Player**

Step 2



Once opened, in the **Solution Explorer** open the **Pages** section, then open the **Index.cshtml** section and select **Index.cshtml.cs**

Step 3



Then from the **Menu** choose **View** and then **Open**

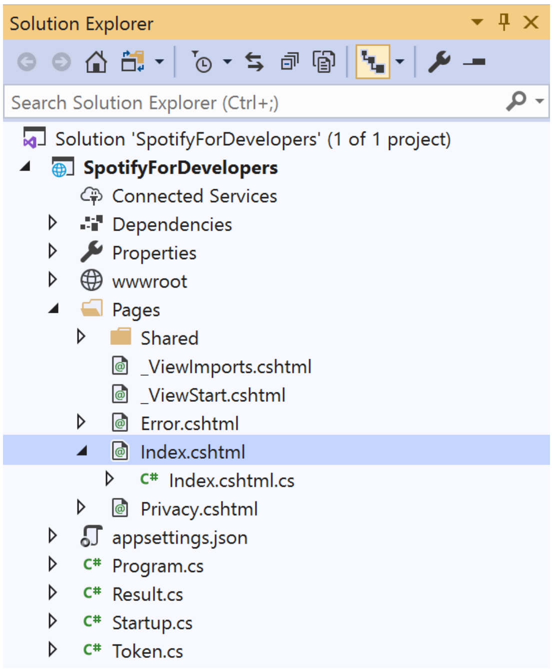
Step 4

In the **Code View** for **Index.cshtml.cs** below the **method** for `public async Task<IActionResult> OnPostGetUserPlaybackCurrentAsync() { ... }` enter the following **method**:

```
public async Task<IActionResult> OnPostGetUserPlaybackCurrentTrackAsync()
{
    LoadToken();
    var result = await Api.GetUserPlaybackCurrentTrackAsync(country,
        new List<string> { "track", "episode" });
    if (result != null)
    {
        if (result.Track != null)
        {
            Results = new List<Result> { new Result()
            {
                Id = result?.Track?.Id,
                Name = result?.Track?.Name,
                Image = result?.Track.Album?.Images?.FirstOrDefault()?.Url,
                Inner = new Result()
                {
                    Id = result?.Track?.Artists?.FirstOrDefault()?.Id,
                    Name = result?.Track?.Artists?.FirstOrDefault()?.Name
                }
            }};
        }
        if (result.Episode != null)
        {
            Results = new List<Result> { new Result()
            {
                Id = result?.Episode?.Id,
                Name = result?.Episode?.Name,
                Image = result?.Episode?.Images?.FirstOrDefault()?.Url,
                Inner = new Result()
                {
                    Id = result?.Episode?.Show?.Id,
                    Name = result?.Episode?.Show?.Name
                }
            }};
        }
    }
    return Page();
}
```

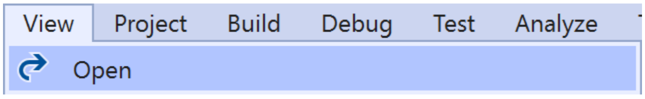
The **method** for `OnPostGetUserPlaybackCurrentTrackAsync` is used get **information** the currently playing **track** or **episode** and populate the **property** for `Results` accordingly.

Step 5



In the **Solution Explorer** in the **Pages** section select **Index.cshtml**

Step 6



Then from the **Menu** choose **View** and then **Open**

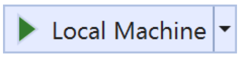
Step 7

Once in the **Code View** for **Index.cshtml** above `<!-- Player -->` enter the following:

```
<li class="list-group-item">
    <form asp-page-handler="GetUserPlaybackCurrentTrack" method="post">
        <button class="btn btn-primary mb-2">
            Get the User's Currently Playing Track
        </button>
    </form>
</li>
```

This `form` will **post** to the **method** for `GetUserPlaybackCurrentTrack` and will output to the **Results**.

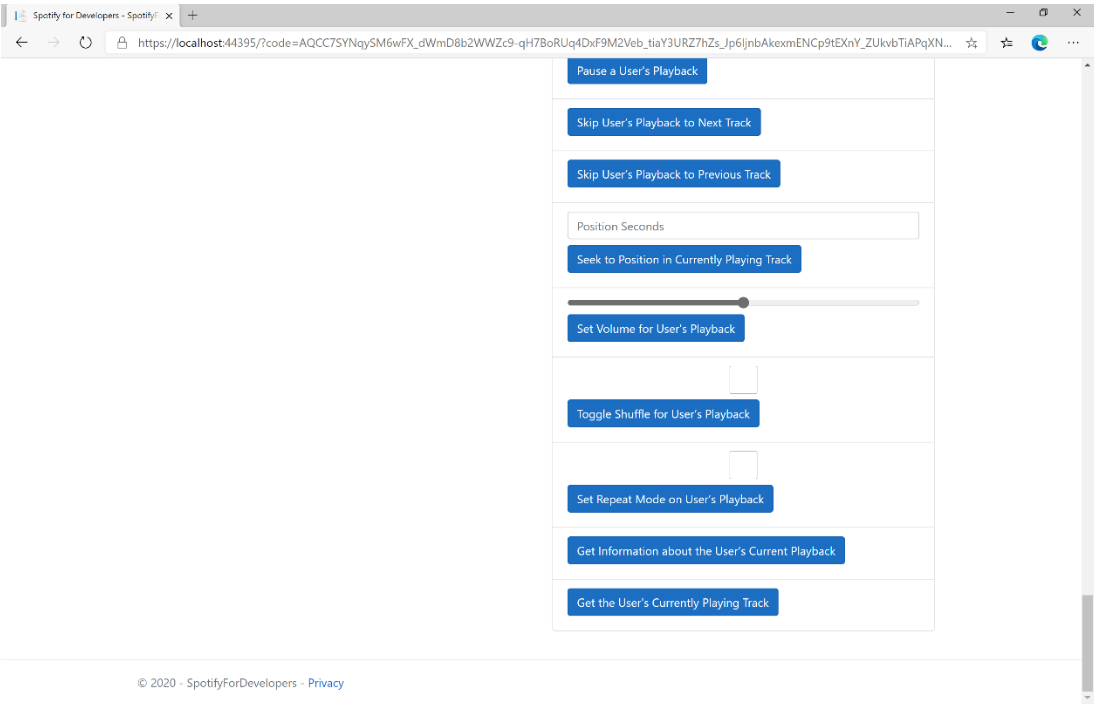
Step 8



Finally, in **Visual Studio 2019** select **IIS Application** to run the **Web Application**

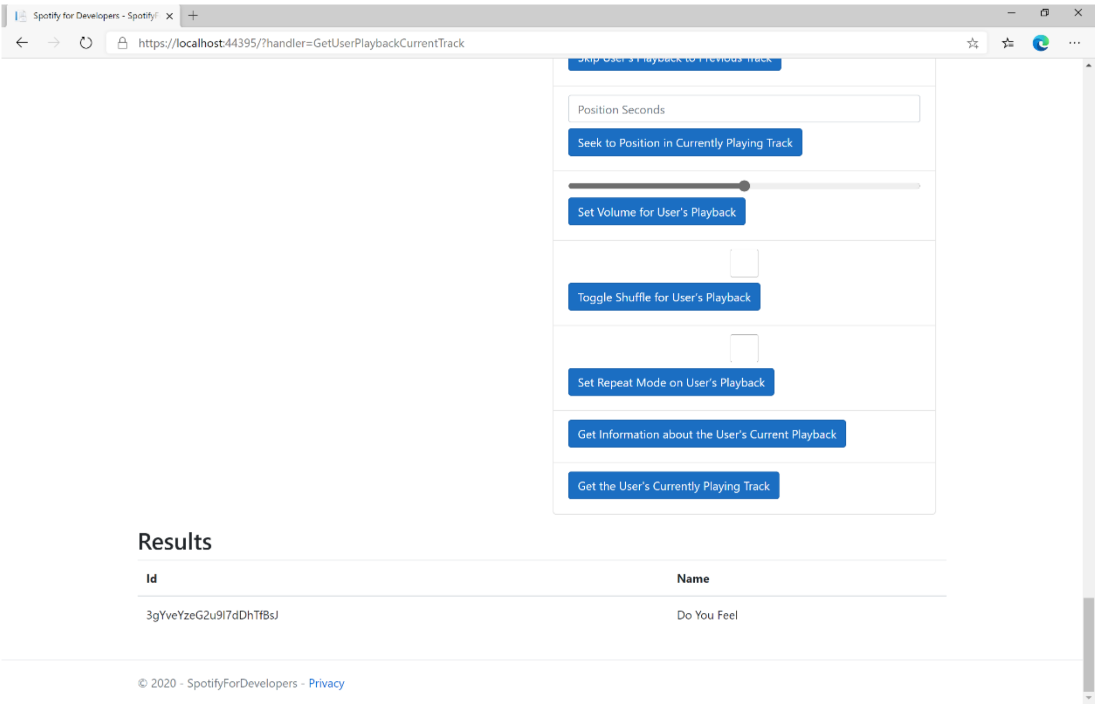
Step 9

Once the **Web Application** is running and you select **Authorisation Code Flow Login** or **Implicit Grant Flow Login** and scroll down you should see something like the following:


Spotify for Developers - Spotify
https://localhost:44395/?code=AQCC7SYNqySM6wFX_dWmD8b2WWZc9-qH7BoRUq4DxF9M2Veb_tiaY3URZ7hZs_Jp6ljnbAkexmENCp9tEXnY_ZUkvbTiAPqXN...
Pause a User's Playback
Skip User's Playback to Next Track
Skip User's Playback to Previous Track
Position Seconds
Seek to Position in Currently Playing Track
Set Volume for User's Playback
Toggle Shuffle for User's Playback
Set Repeat Mode on User's Playback
Get Information about the User's Current Playback
Get the User's Currently Playing Track
© 2020 - SpotifyForDevelopers - Privacy

Step 10

Make sure **Spotify** is running then you can then use **Start/Resume a User's Playback** with a **Track Id** or **Episode Id** from **Search For An Item** with option of **Track** or **Episode** then select **Get the User's Currently Playing Track** and scroll down to view **Results** like the following:


Spotify for Developers - Spotify
https://localhost:44395/?handler=GetUserPlaybackCurrentTrack
Skip User's Playback to Previous Track
Position Seconds
Seek to Position in Currently Playing Track
Set Volume for User's Playback
Toggle Shuffle for User's Playback
Set Repeat Mode on User's Playback
Get Information about the User's Current Playback
Get the User's Currently Playing Track
© 2020 - SpotifyForDevelopers - Privacy

Results

Id	Name
3gYveYzeG2u9I7dDhTfBsJ	Do You Feel

Step 11

■

You can stop the **web application** in **Visual Studio 2019** by selecting the **Stop debugging** button

Step 12

×

You can choose to exit **Visual Studio 2019** by selecting the **Close** button in the top right of the **application** as that completes this part of the workshop

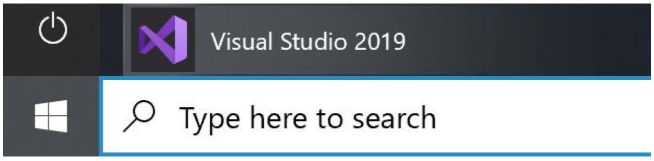
Get Current User's Recently Played Tracks

Get tracks from the current user’s recently played tracks.

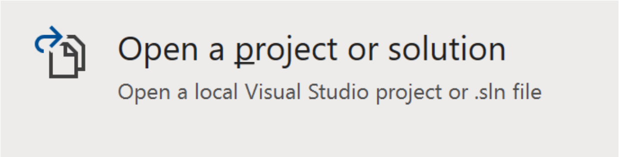
GET https://api.spotify.com/v1/me/player/recently-played	
Header	
Authorization	User Token from Spotify Accounts service with **user-read-recently-played** scope
Query Parameter	
limit	Maximum number of items to return. Default: 20. Minimum: 1. Maximum: 50.
after	Unix timestamp in milliseconds. Returns all items after but not including this cursor position. If after is specified, before must not be specified.
before	Unix timestamp in milliseconds. Returns all items before but not including this cursor position. If before is specified, after must not be specified.

Header	Response
Success	
HTTP Status 200 OK	Array of Play History Objects wrapped in a Cursor Paging Object
HTTP Status 204 No Content	If private session an empty payload
Error	
Error Code	Error Object

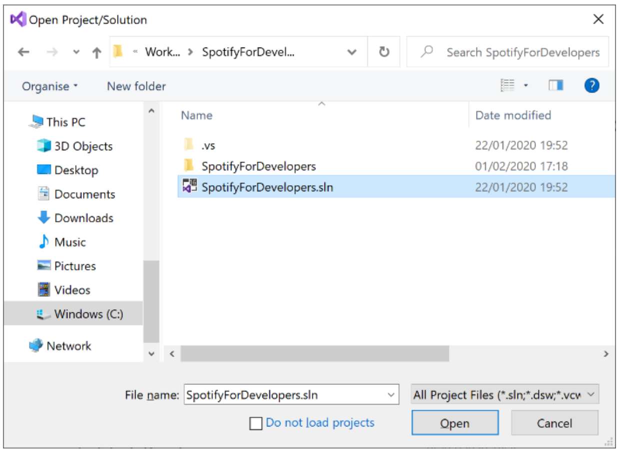
Step 1



If you chose to close **Visual Studio 2019** previously, in **Windows 10** choose **Start**, and then from the **Start Menu** find and select **Visual Studio 2019**

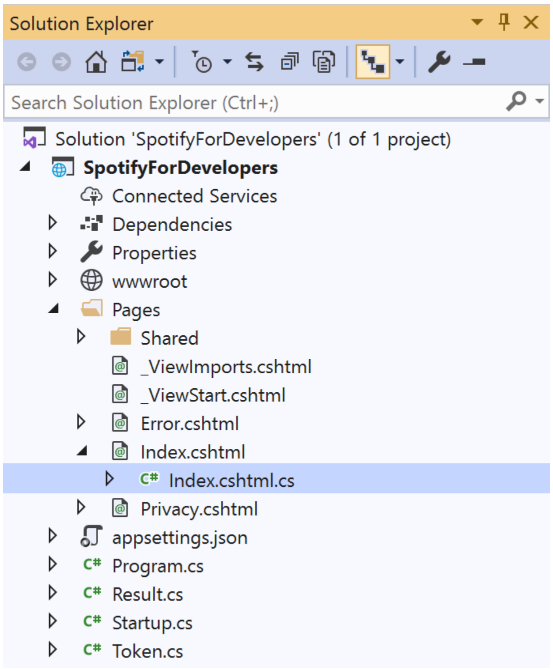

Open a project or solution
Open a local Visual Studio project or .sln file

Once done, from the **Get started** screen for **Visual Studio 2019** select **Open a project or solution**



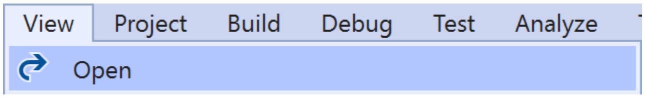
Then locate and select **SpotifyForDevelopers.sln** and select **Open** if you don’t have this file already then please follow the previous parts of the workshop including **Getting Started**, **Authorisation Guide**, **Search & Browse**, **Playlists & Artists**, **Albums & Tracks**, **Episodes & Shows**, **Follow**, **Playlists**, **Library** and **Player**

Step 2



Once opened, in the **Solution Explorer** open the **Pages** section, then open the **Index.cshtml** section and select **Index.cshtml.cs**

Step 3



Then from the **Menu** choose **View** and then **Open**

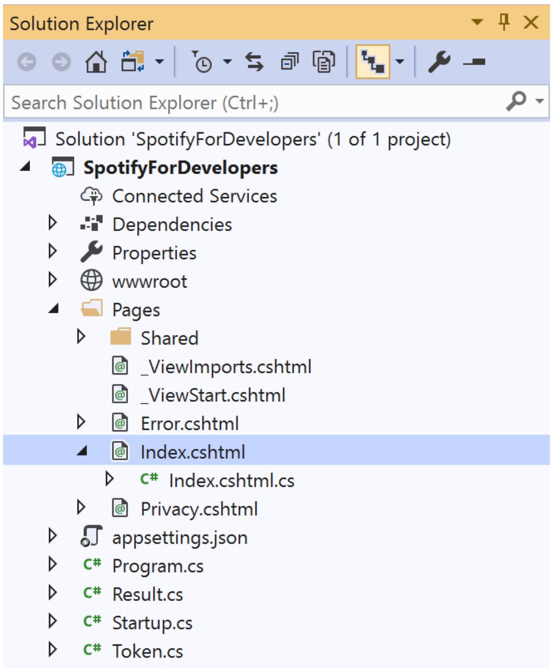
Step 4

In the **Code View** for **Index.cshtml.cs** below the **method** for `public async Task<IActionResult> OnPostGetUserPlaybackCurrentTrackAsync() { ... }` enter the following **method**:

```
public async Task<IActionResult> OnPostGetUserRecentlyPlayedTracksAsync()
{
    LoadToken();
    var results = await Api.GetUserRecentlyPlayedTracksAsync();
    if (results?.Items != null)
    {
        Results = results.Items.Select(result => new Result()
        {
            Id = result.Track.Id,
            Name = result.Track.Name,
            Image = result?.Track.Album?.Images?.FirstOrDefault()?.Url,
            Inner = new Result()
            {
                Id = result?.Track?.Artists?.FirstOrDefault()?.Id,
                Name = result?.Track?.Artists?.FirstOrDefault()?.Name
            }
        });
    }
    return Page();
}
```

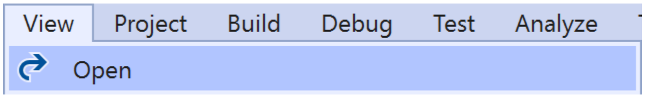
The **method** for `OnPostGetUserRecentlyPlayedTracksAsync` is used to get the **tracks** that have been played **recently** and populate the **property** for `Results` accordingly.

Step 5



In the **Solution Explorer** in the **Pages** section select **Index.cshtml**

Step 6



Then from the **Menu** choose **View** and then **Open**

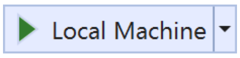
Step 7

Once in the **Code View** for **Index.cshtml** above `<!-- Player -->` enter the following:

```
<li class="list-group-item">
    <form asp-page-handler="GetUserRecentlyPlayedTracks" method="post">
        <button class="btn btn-primary mb-2">
            Get User's Recently Played Tracks
        </button>
    </form>
</li>
```

This `form` will **post** to the **method** for `GetUserRecentlyPlayedTracks` and will output to the **Results**.

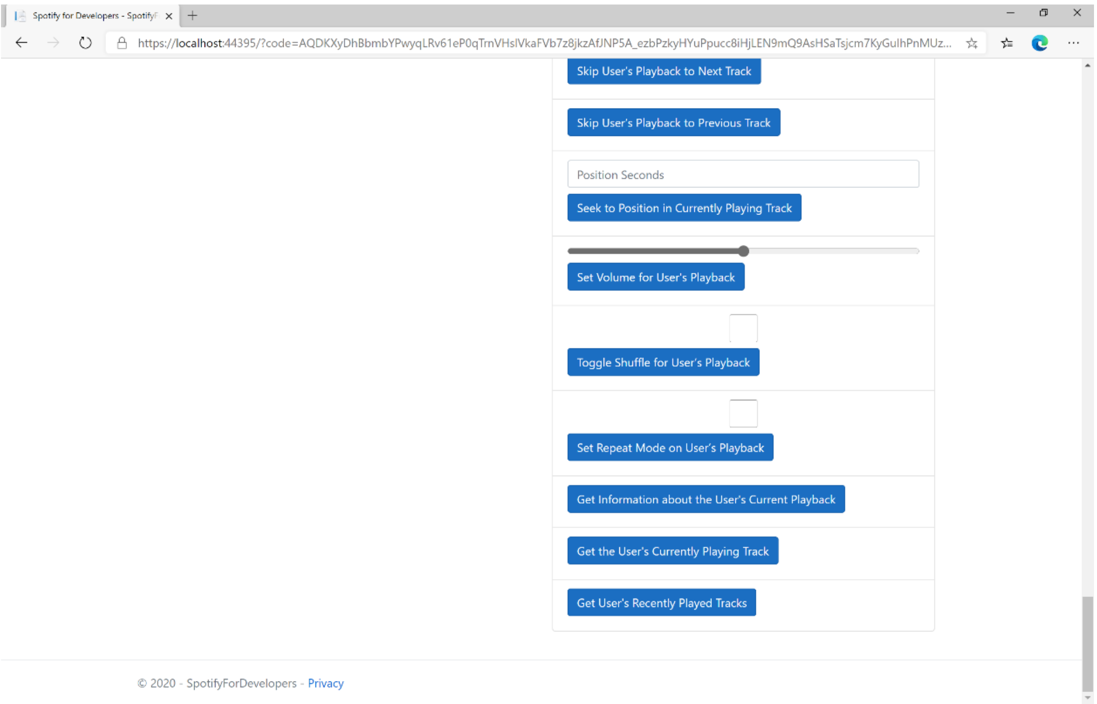
Step 8



Finally, in **Visual Studio 2019** select **IIS Application** to run the **Web Application**

Step 9

Once the **Web Application** is running and you select **Authorisation Code Flow Login** or **Implicit Grant Flow Login** and scroll down you should see something like the following:

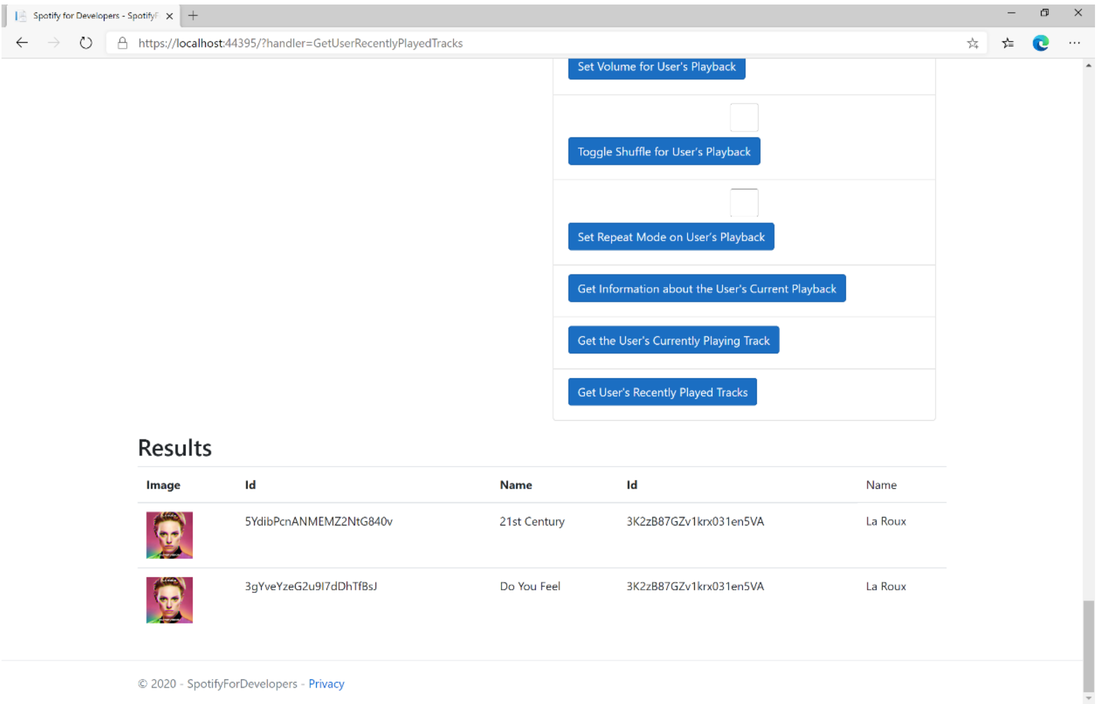


Spotify for Developers - SpotifyF
https://localhost:44395/?code=AQDKXyDhBbmbYPwyqLRv61eP0qTrnVHslVkaFVb7z8jkzAfJNP5A_ezbPzkyHYuPpucc8iHjLEN9mQ9AsHSaTsjcm7KyGulhPnMUz...
Skip User's Playback to Next Track
Skip User's Playback to Previous Track
Position Seconds
Seek to Position in Currently Playing Track
Set Volume for User's Playback
Toggle Shuffle for User's Playback
Set Repeat Mode on User's Playback
Get Information about the User's Current Playback
Get the User's Currently Playing Track
Get User's Recently Played Tracks

© 2020 - SpotifyForDevelopers - Privacy

Step 10

You can select **Get User's Recently Played Tracks** and scroll down to view **Results** like the following:



Spotify for Developers - SpotifyF
https://localhost:44395/?handler=GetUserRecentlyPlayedTracks
Set Volume for User's Playback
Toggle Shuffle for User's Playback
Set Repeat Mode on User's Playback
Get Information about the User's Current Playback
Get the User's Currently Playing Track
Get User's Recently Played Tracks

Results

Image	Id	Name	Id	Name
	5YdibPcnANMEMZ2NtG840v	21st Century	3K2zB87GZv1krx031en5VA	La Roux
	3gYveYzeG2u9I7dDhTfBsJ	Do You Feel	3K2zB87GZv1krx031en5VA	La Roux

© 2020 - SpotifyForDevelopers - Privacy

Step 11

■

You can stop the **web application** in **Visual Studio 2019** by selecting the **Stop debugging** button

Step 12

×

You can choose to exit **Visual Studio 2019** by selecting the **Close** button in the top right of the **application** as that completes this part of the workshop

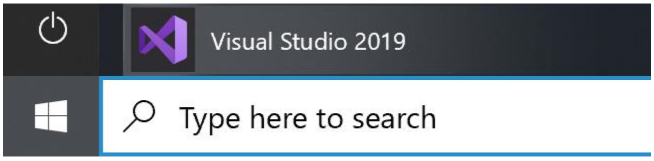
Get a User's Available Devices

Get information about a user's available devices.

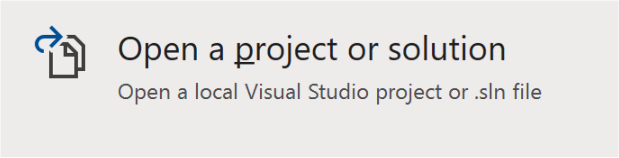
GET https://api.spotify.com/v1/me/player/devices	
Header	
Authorization	User Token from Spotify Accounts service with **user-read-playback-state** scope

Header	Response
Success	
HTTP Status 200 OK	Array of Device Objects or Empty if none
Error	
Error Code	Error Object

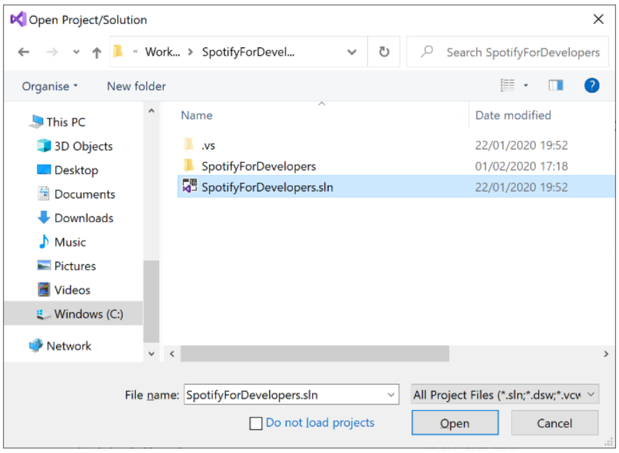
Step 1



If you chose to close **Visual Studio 2019** previously, in **Windows 10** choose **Start**, and then from the **Start Menu** find and select **Visual Studio 2019**


Open a project or solution
Open a local Visual Studio project or .sln file

Once done, from the **Get started** screen for **Visual Studio 2019** select **Open a project or solution**



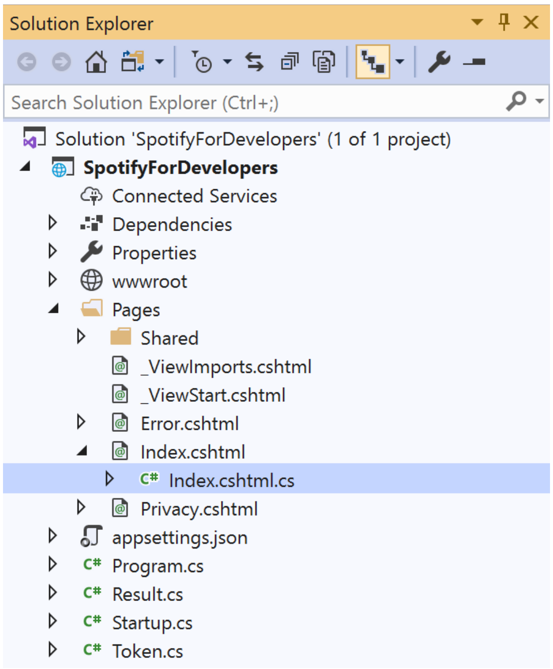
Name	Date modified
.vs	22/01/2020 19:52
SpotifyForDevelopers	01/02/2020 17:18
SpotifyForDevelopers.sln	22/01/2020 19:52

File name: SpotifyForDevelopers.sln

All Project Files (*.sln;*.dsw;*.vcw

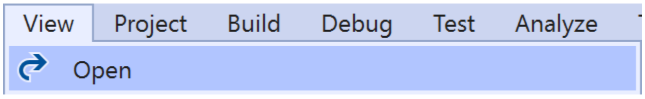
Then locate and select **SpotifyForDevelopers.sln** and select **Open** if you don't have this file already then please follow the previous parts of the workshop including **Getting Started**, **Authorisation Guide**, **Search & Browse**, **Playlists & Artists**, **Albums & Tracks**, **Episodes & Shows**, **Follow**, **Playlists**, **Library** and **Player**

Step 2



Once opened, in the **Solution Explorer** open the **Pages** section, then open the **Index.cshtml** section and select **Index.cshtml.cs**

Step 3



Then from the **Menu** choose **View** and then **Open**

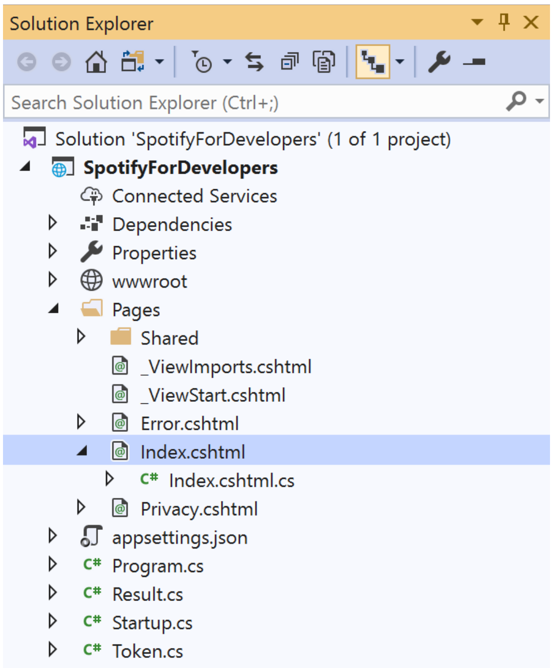
Step 4

In the **Code View** for **Index.cshtml.cs** below the **method** for `public async Task<IActionResult> OnPostGetUserRecentlyPlayedTracksAsync() { ... }` enter the following **method**:

```
public async Task<IActionResult> OnPostGetUserPlaybackDevicesAsync()
{
    LoadToken();
    var results = await Api.GetUserPlaybackDevicesAsync();
    if (results?.Items != null)
    {
        Results = results.Items.Select(result => new Result()
        {
            Id = result.Id,
            Name = result.Name,
        });
    }
    return Page();
}
```

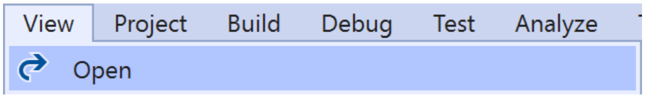
The **method** for `OnPostGetUserPlaybackDevicesAsync` is used to get the **devices** that are **available** for a **user** and populate the **property** for `Results` accordingly.

Step 5



In the **Solution Explorer** in the **Pages** section select **Index.cshtml**

Step 6



Then from the **Menu** choose **View** and then **Open**

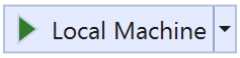
Step 7

Once in the **Code View** for **Index.cshtml** above `<!-- Player -->` enter the following:

```
<li class="list-group-item">
    <form asp-page-handler="GetUserPlaybackDevices" method="post">
        <button class="btn btn-primary mb-2">
            Get a User's Available Devices
        </button>
    </form>
</li>
```

This `form` will **post** to the **method** for `GetUserPlaybackDevices` and will output to the **Results**.

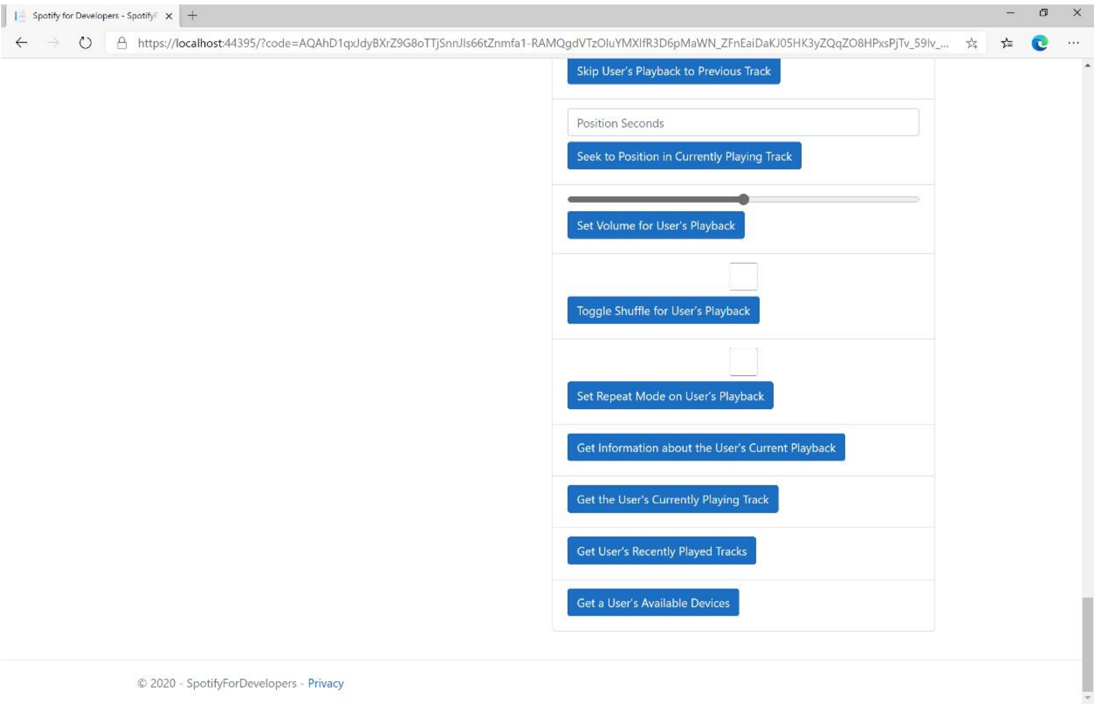
Step 8



Finally, in **Visual Studio 2019** select **IIS Application** to run the **Web Application**

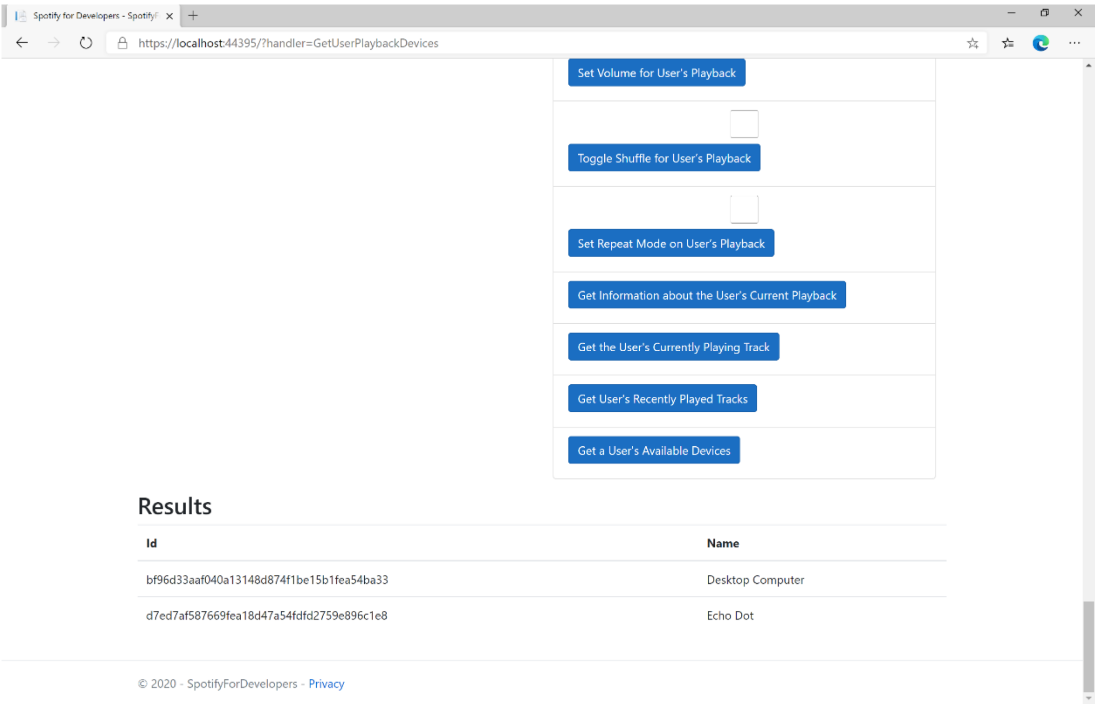
Step 9

Once the **Web Application** is running and you select **Authorisation Code Flow Login** or **Implicit Grant Flow Login** and scroll down you should see something like the following:


Spotify for Developers - SpotifyF
https://localhost:44395/?code=AQAhD1qxJdyBXrZ9G8oTTjSnnJls66tZnmfa1-RAMQgdVTzOluYMXIfR3D6pMaWN_ZFnEaiDaKJ05HK3yZQqZO8HPxsPjTv_59lv_...
Skip User's Playback to Previous Track
Position Seconds
Seek to Position in Currently Playing Track
Set Volume for User's Playback
Toggle Shuffle for User's Playback
Set Repeat Mode on User's Playback
Get Information about the User's Current Playback
Get the User's Currently Playing Track
Get User's Recently Played Tracks
Get a User's Available Devices
© 2020 - SpotifyForDevelopers - Privacy

Step 10

You can select **Get a User's Available Devices** and scroll down to view **Results** like the following:


Spotify for Developers - SpotifyF
https://localhost:44395/?handler=GetUserPlaybackDevices
Set Volume for User's Playback
Toggle Shuffle for User's Playback
Set Repeat Mode on User's Playback
Get Information about the User's Current Playback
Get the User's Currently Playing Track
Get User's Recently Played Tracks
Get a User's Available Devices
© 2020 - SpotifyForDevelopers - Privacy

Results

Id	Name
bf96d33aaf040a13148d874f1be15b1fea54ba33	Desktop Computer
d7ed7af587669fea18d47a54fdfd2759e896c1e8	Echo Dot

Step 11

You can stop the **web application** in **Visual Studio 2019** by selecting the **Stop debugging** button

Step 12

You can choose to exit **Visual Studio 2019** by selecting the **Close** button in the top right of the **application** as that completes this part of the workshop

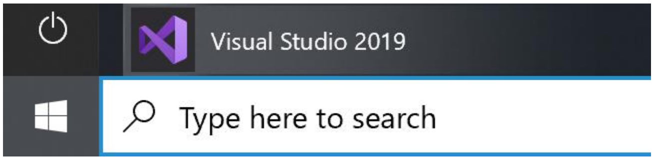
Transfer a User's Playback

Transfer playback to a new device and determine if it should start playing.

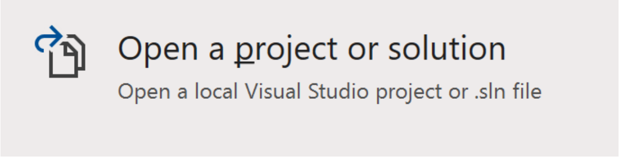
PUT https://api.spotify.com/v1/me/player	
Header	
Authorization	User Token from Spotify Accounts service with **user-modify-playback-state** scope
Body Parameter	
device_ids	Device Id on which playback should be started/transferred, only one Device Id is supported
play	If true, ensure playback happens on new device. If false, or not provided keep the current playback state.

Header	**Response**
Success	
HTTP Status 204 No Content	
Error	
Error Code	Error Object

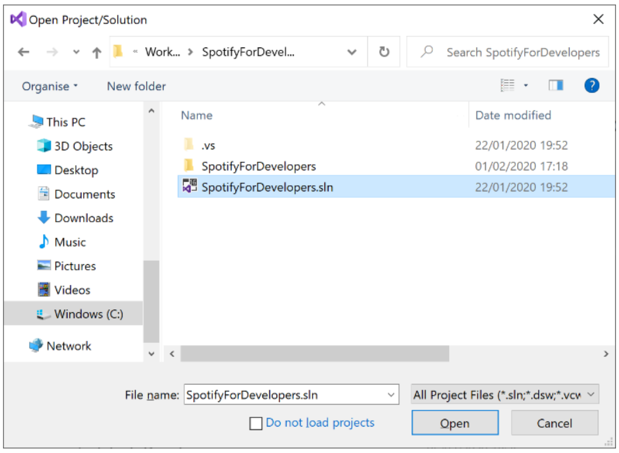
Step 1



If you chose to close **Visual Studio 2019** previously, in **Windows 10** choose **Start**, and then from the **Start Menu** find and select **Visual Studio 2019**


Open a project or solution
Open a local Visual Studio project or .sln file

Once done, from the **Get started** screen for **Visual Studio 2019** select **Open a project or solution**



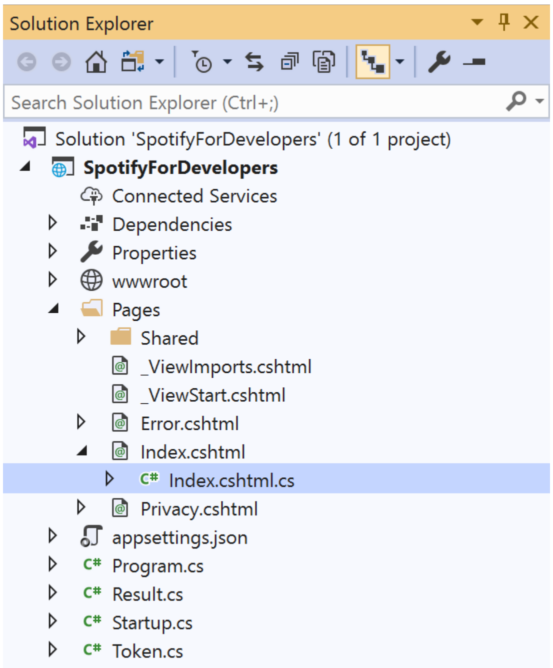
Name	Date modified
.vs	22/01/2020 19:52
SpotifyForDevelopers	01/02/2020 17:18
SpotifyForDevelopers.sln	22/01/2020 19:52

File name: SpotifyForDevelopers.sln

All Project Files (*.sln;*.dsw;*.vcw

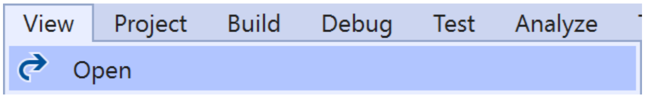
Then locate and select **SpotifyForDevelopers.sln** and select **Open** if you don't have this file already then please follow the previous parts of the workshop including **Getting Started**, **Authorisation Guide**, **Search & Browse**, **Playlists & Artists**, **Albums & Tracks**, **Episodes & Shows**, **Follow**, **Playlists**, **Library** and **Player**

Step 2



Once opened, in the **Solution Explorer** open the **Pages** section, then open the **Index.cshtml** section and select **Index.cshtml.cs**

Step 3



Then from the **Menu** choose **View** and then **Open**

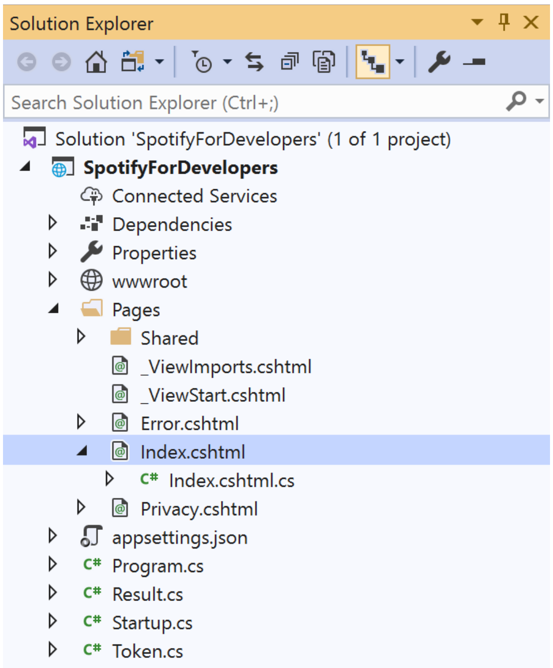
Step 4

In the **Code View** for **Index.cshtml.cs** below the **method** for `public async Task<IActionResult> OnPostGetUserPlaybackDevicesAsync() { ... }` enter the following **method**:

```
public async Task<IActionResult> OnPostUserPlaybackTransferAsync(string value)
{
    LoadToken();
    var deviceIds = new List<string> { value };
    var result = await Api.UserPlaybackTransferAsync(deviceIds, true);
    if (result != null)
    {
        Results = new List<Result>() { new Result()
        {
            Id = result.Code.ToString(),
            Name = result.Success.ToString()
        }};
    }
    return Page();
}
```

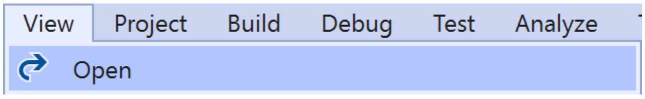
The **method** for `OnPostUserPlaybackTransferAsync` is used to set the **device** to use for **playback** with the `Value` of the **Device Id** and populate the **property** for `Results` of the **success** of the operation accordingly.

Step 5



In the **Solution Explorer** in the **Pages** section select **Index.cshtml**

Step 6



Then from the **Menu** choose **View** and then **Open**

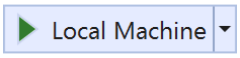
Step 7

Once in the **Code View** for **Index.cshtml** above `<!-- Player -->` enter the following:

```
<li class="list-group-item">
    <form asp-page-handler="UserPlaybackTransfer" method="post">
        <input asp-for="Value" placeholder="Device Id" class="form-control mb-2" />
        <button class="btn btn-primary mb-2">
            Transfer a User's Playback
        </button>
    </form>
</li>
```

This `form` will **post** to the **method** for `UserPlaybackTransfer` with the `Value` of **Device Id** and will output to the **Results**.

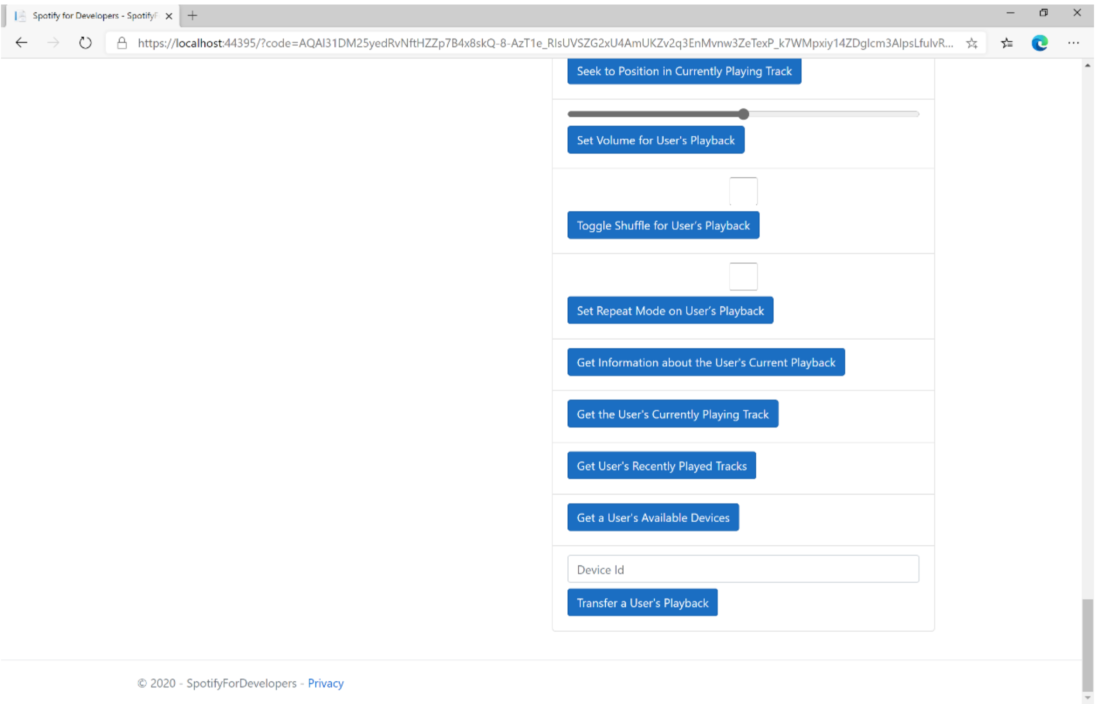
Step 8



Finally, in **Visual Studio 2019** select **IIS Application** to run the **Web Application**

Step 9

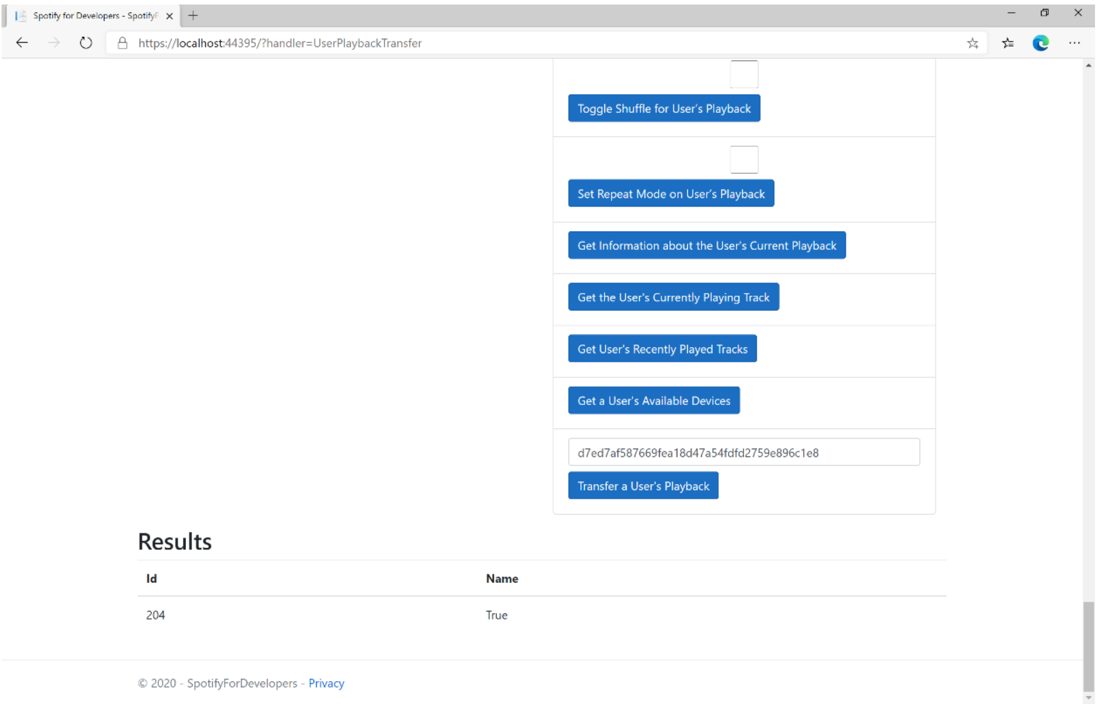
Once the **Web Application** is running and you select **Authorisation Code Flow Login** or **Implicit Grant Flow Login** and scroll down you should see something like the following:



Spotify for Developers - SpotifyF
https://localhost:44395/?code=AQAI31DM25yedRvNftHZZp7B4x8skQ-8-AzT1e_RlsUVSZG2xU4AmUKZv2q3EnMvnw3ZeTexP_k7WMpxiy14ZDglcm3AlpsLfulvR...
Seek to Position in Currently Playing Track
Set Volume for User's Playback
Toggle Shuffle for User's Playback
Set Repeat Mode on User's Playback
Get Information about the User's Current Playback
Get the User's Currently Playing Track
Get User's Recently Played Tracks
Get a User's Available Devices
Device Id
Transfer a User's Playback
© 2020 - SpotifyForDevelopers - Privacy

Step 10

Make sure **Spotify** is running then you can then enter a **Device Id** from **Get a User's Available Devices** and enter this in **Device Id** above **Transfer a User's Playback** and then select **Transfer a User's Playback** and scroll down to view **Results** like the following:



Spotify for Developers - SpotifyF
https://localhost:44395/?handler=UserPlaybackTransfer
Toggle Shuffle for User's Playback
Set Repeat Mode on User's Playback
Get Information about the User's Current Playback
Get the User's Currently Playing Track
Get User's Recently Played Tracks
Get a User's Available Devices
Transfer a User's Playback
© 2020 - SpotifyForDevelopers - Privacy

d7ed7af587669fea18d47a54fdfd2759e896c1e8

Results

Id	Name
204	True

Step 11

■

You can stop the **web application** in **Visual Studio 2019** by selecting the **Stop debugging** button

Step 12

×

You can exit **Visual Studio 2019** by selecting the **Close** button in the top right of the **application** as that completes this part of the workshop

Personalisation & User Profile

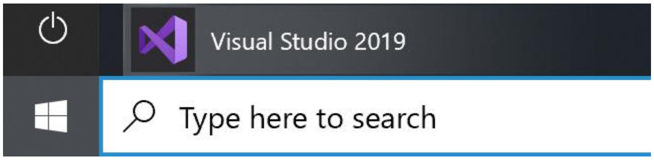
Get a User's Top Artists and Tracks

Get the current user's top artists based on calculated affinity.

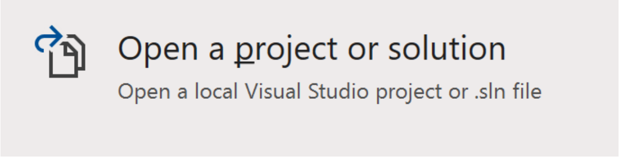
GET https://api.spotify.com/v1/me/top/	
Header	
Authorization	User Token from Spotify Accounts service with **user-top-read** scope
Path Parameter	
type	The type of entity to return. Valid values: artists or tracks
Query Parameter	
limit	Number of entities to return. Default: 20. Minimum: 1. Maximum: 50
offset	Index of the first entity to return. Default: 0
time_range	Over what time frame the affinities are computed. long_term - calculated from several years of data and including all new data as it becomes available, medium_term - approximately last 6 months and short_term - approximately last 4 weeks. Default: medium_term

Header	Response
Success	
HTTP Status 200 OK	List of Artists or Tracks wrapped in a Paging Object
Error	
Error Code	Error Object

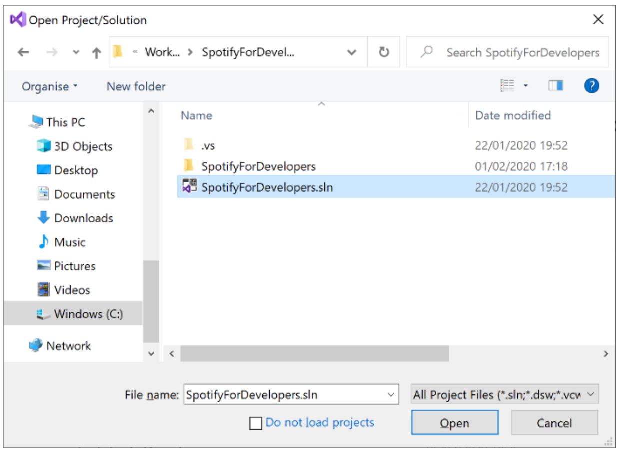
Step 1



In **Windows 10** choose **Start**, and then from the **Start Menu** find and select **Visual Studio 2019**


Open a project or solution
Open a local Visual Studio project or .sln file

Once done, from the **Get started** screen for **Visual Studio 2019** select **Open a project or solution**



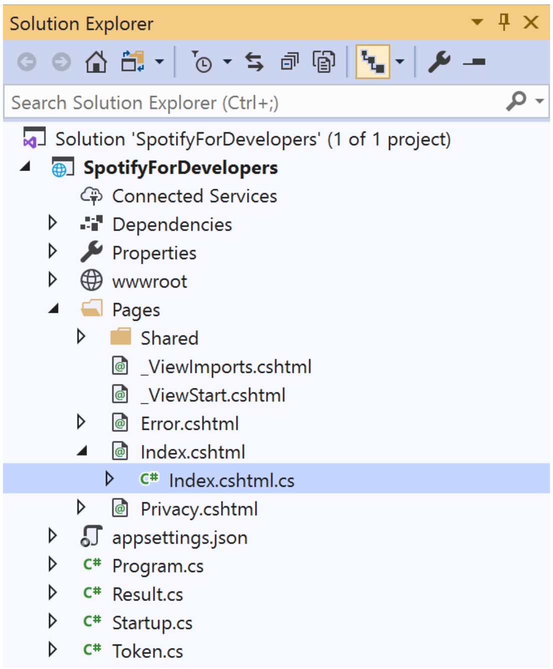
Name	Date modified
.vs	22/01/2020 19:52
SpotifyForDevelopers	01/02/2020 17:18
SpotifyForDevelopers.sln	22/01/2020 19:52

File name: SpotifyForDevelopers.sln

All Project Files (*.sln;*.dsw;*.vcw

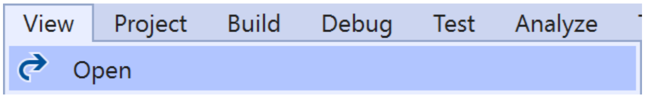
Then locate and select **SpotifyForDevelopers.sln** and select **Open** if you don't have this file already then please follow the previous parts of the workshop including **Getting Started**, **Authorisation Guide**, **Search & Browse**, **Playlists & Artists**, **Albums & Tracks**, **Episodes & Shows**, **Follow**, **Playlists**, **Library** and **Player**

Step 2



Once opened, in the **Solution Explorer** open the **Pages** section, then open the **Index.cshtml** section and select **Index.cshtml.cs**

Step 3



Then from the **Menu** choose **View** and then **Open**

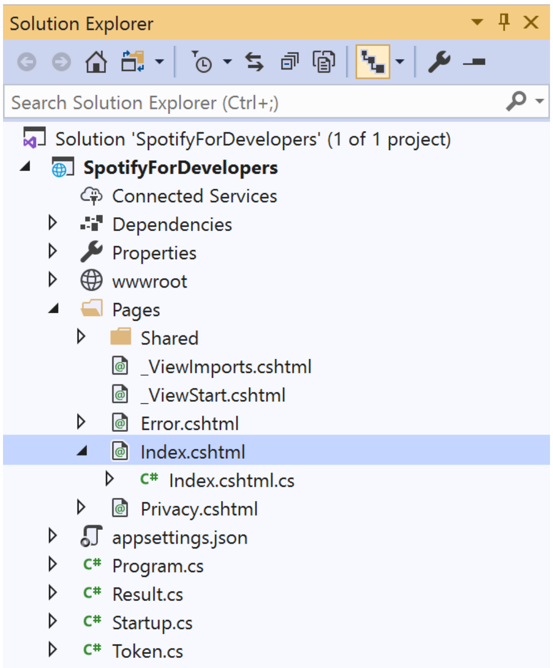
Step 4

In the **Code View** for **Index.cshtml.cs** below the **method** for `public async Task<IActionResult> OnPostUserPlaybackTransferAsync() { ... }` enter the following **method**:

```
public async Task<IActionResult> OnPostGetUserTopArtistsAndTracksAsync(string option)
{
    LoadToken();
    if(option.Equals("Artists"))
    {
        var results = await Api.GetUserTopArtistsAsync();
        if (results?.Items != null)
        {
            Results = results.Items.Select(result => new Result()
            {
                Id = result.Id,
                Name = result.Name,
                Image = result?.Images?.FirstOrDefault()?.Url
            });
        }
    }
    else
    {
        var results = await Api.GetUserTopTracksAsync();
        if (results?.Items != null)
        {
            Results = results.Items.Select(result => new Result()
            {
                Id = result.Id,
                Name = result.Name,
                Image = result?.Album?.Images?.FirstOrDefault()?.Url,
                Inner = new Result()
                {
                    Id = result?.Album?.Artists?.FirstOrDefault()?.Id,
                    Name = result?.Album?.Artists?.FirstOrDefault()?.Name
                }
            });
        }
    }
    return Page();
}
```

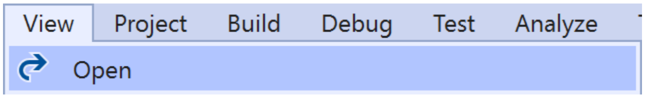
The **method** for `OnPostGetUserTopArtistsAndTracksAsync` is used to get the **top** played **artists** or **tracks** using the `Option` of the **Artists** and **Tracks** and populate the **property** for `Results` accordingly.

Step 5



In the **Solution Explorer** in the **Pages** section select **Index.cshtml**

Step 6



Then from the **Menu** choose **View** and then **Open**

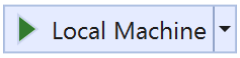
Step 7

Once in the **Code View** for **Index.cshtml** above `<!-- Spotify Web API User Authorisation -->` enter the following:

```
<ul class="list-group mb-2">
    <li class="list-group-item list-group-item-primary">
        <h5 class="list-group-item-heading">Personalisation</h5>
    </li>
    <li class="list-group-item">
        <form asp-page-handler="GetUserTopArtistsAndTracks" method="post">
            <select asp-for="Option" class="form-control mb-2">
                <option>Artists</option>
                <option>Tracks</option>
            </select>
            <button class="btn btn-primary mb-2">
                Get a User's Top Artists and Tracks
            </button>
        </form>
    </li>
</ul>
```

This `form` will **post** to the **method** for `GetUserTopArtistsAndTracks` with the `Option` of **Artists** and **Tracks** and will output to the **Results**.

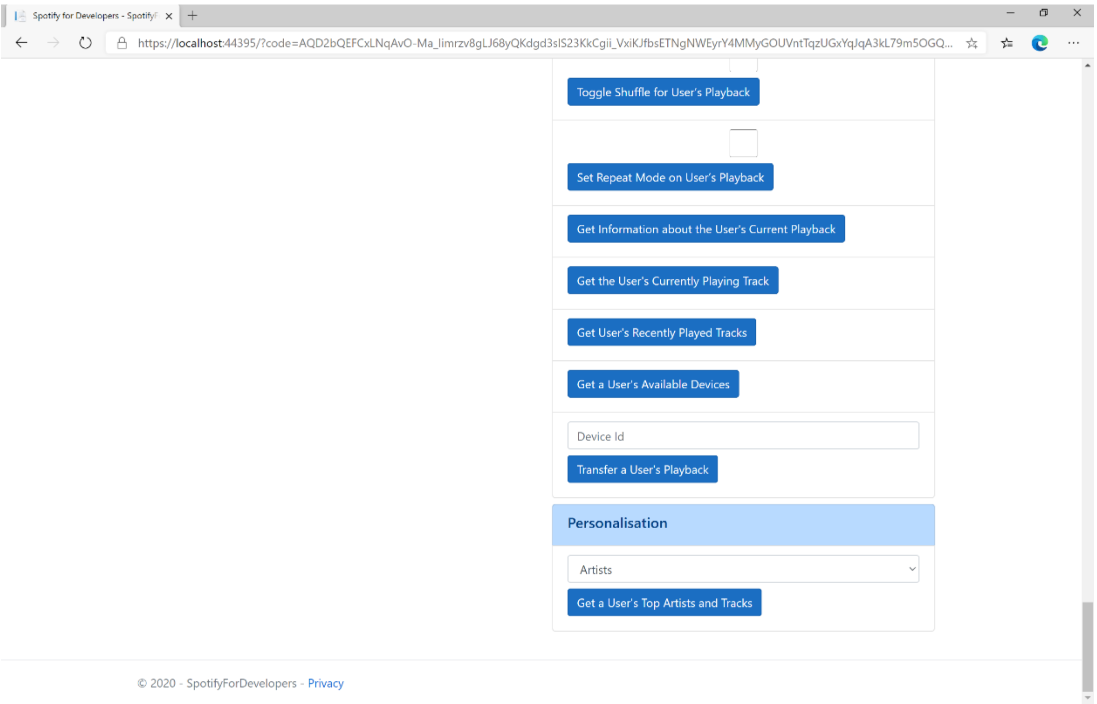
Step 8



Finally, in **Visual Studio 2019** select **IIS Application** to run the **Web Application**

Step 9

Once the **Web Application** is running and you select **Authorisation Code Flow Login** or **Implicit Grant Flow Login** and scroll down you should see something like the following:

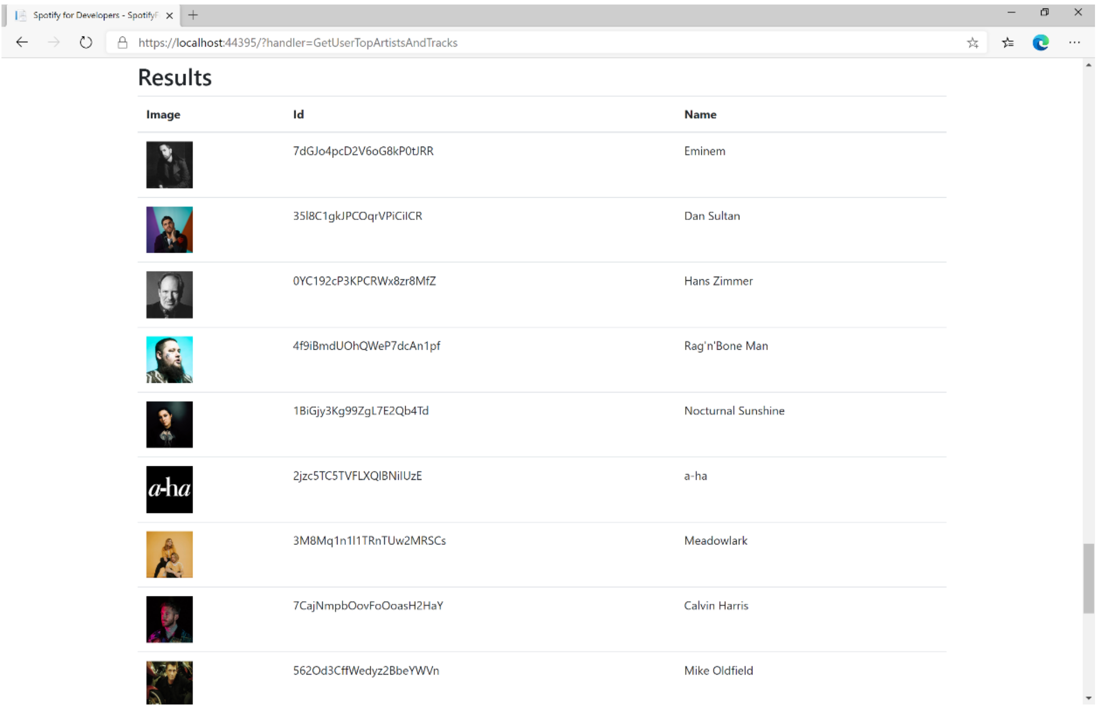


https://localhost:44395/?code=AQD2bQEFCxLNqAvO-Ma_limrzv8gLJ68yQKdgd3slS23KkCgii_VxiKJfbsETNgNWEyrY4MMyGOUVntTqzUGxYqJqA3kL79m5OGQ...

Spotify for Developers - SpotifyF
Toggle Shuffle for User's Playback
Set Repeat Mode on User's Playback
Get Information about the User's Current Playback
Get the User's Currently Playing Track
Get User's Recently Played Tracks
Get a User's Available Devices
Device Id
Transfer a User's Playback
Personalisation
Artists
Get a User's Top Artists and Tracks
© 2020 - SpotifyForDevelopers - Privacy

Step 10

You can then select either **Artists** or **Tracks** above **Get a User's Top Artists and Tracks** and then select **Get a User's Top Artists and Tracks** and scroll down to view **Results** like the following:



https://localhost:44395/?handler=GetUserTopArtistsAndTracks

Results

Image	Id	Name
	7dGJo4pcD2V6oG8kP0tJRR	Eminem
	35l8C1gkJPCOqrVPiCiICR	Dan Sultan
	0YC192cP3KPCRWx8zr8MfZ	Hans Zimmer
	4f9iBmdUOhQWeP7dcAn1pf	Rag'n'Bone Man
	1BiGjy3Kg99ZgL7E2Qb4Td	Nocturnal Sunshine
a-ha	2jzc5TC5TVFLXQIBNiIUzE	a-ha
	3M8Mq1n1l1TRnTUw2MRSCs	Meadowlark
	7CajNmpbOovFoOoasH2HaY	Calvin Harris
	562Od3CffWedyz2BbeYWVn	Mike Oldfield

Step 11

You can stop the **web application** in **Visual Studio 2019** by selecting the **Stop debugging** button

Step 12

You can choose to exit **Visual Studio 2019** by selecting the **Close** button in the top right of the **application** as that completes this part of the workshop

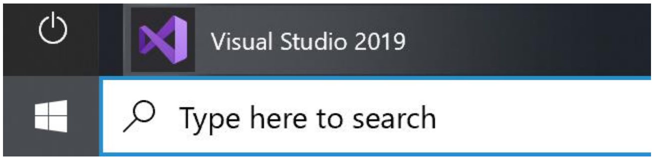
Get a User's Profile

Get public profile information about a Spotify user.

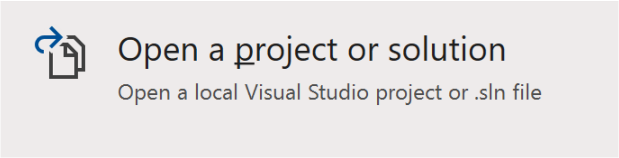
GET https://api.spotify.com/v1/users/{user_id}	
Header	
Authorization	User Token from Spotify Accounts service
Path Parameter	
user_id	The user’s Spotify User Id

Header	Response
Success	
HTTP Status 200 OK	List of Tracks wrapped in a Paging Object
Error	
HTTP Status 404 Not Found	User does not exist
Error Code	Error Object

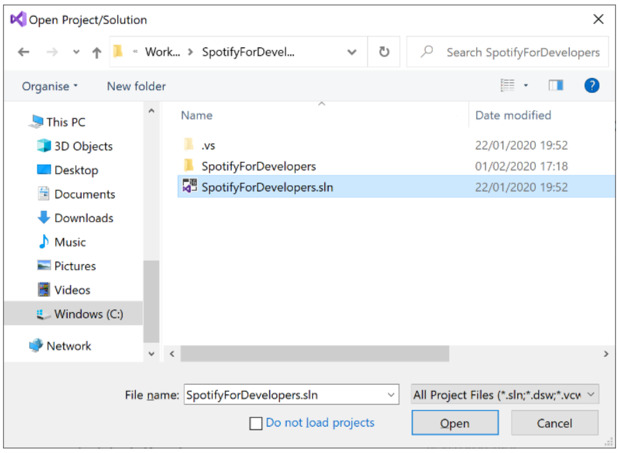
Step 1



If you chose to close **Visual Studio 2019** previously, in **Windows 10** choose **Start**, and then from the **Start Menu** find and select **Visual Studio 2019**


Open a project or solution
Open a local Visual Studio project or .sln file

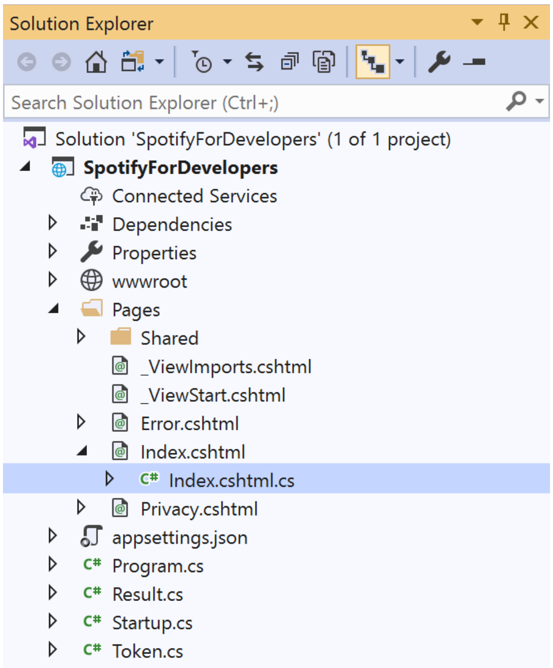
Once done, from the **Get started** screen for **Visual Studio 2019** select **Open a project or solution**



File name: SpotifyForDevelopers.sln

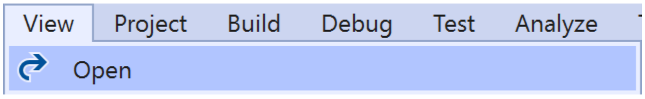
Then locate and select **SpotifyForDevelopers.sln** and select **Open** if you don’t have this file already then please follow the previous parts of the workshop including **Getting Started**, **Authorisation Guide**, **Search & Browse**, **Playlists & Artists**, **Albums & Tracks**, **Episodes & Shows**, **Follow**, **Playlists**, **Library**, **Player** and **Personalisation & User Profile**

Step 2



Once opened, in the **Solution Explorer** open the **Pages** section, then open the **Index.cshtml** section and select **Index.cshtml.cs**

Step 3



Then from the **Menu** choose **View** and then **Open**

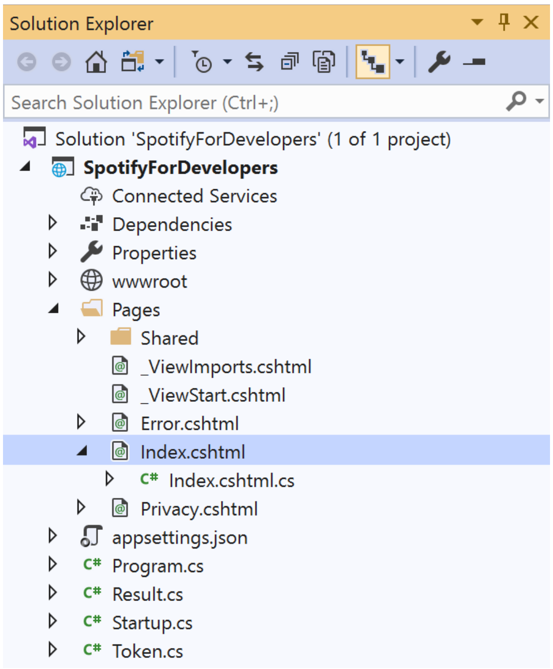
Step 4

In the **Code View** for **Index.cshtml.cs** below the **method** for `public async Task<IActionResult> OnPostGetUserTopArtistsAndTracksAsync(...) { ... }` enter the following **method**:

```
public async Task<IActionResult> OnPostGetUserProfileAsync(string value)
{
    LoadToken();
    var result = await Api.GetUserProfileAsync(value);
    if (result != null)
    {
        Results = new List<Result> { new Result()
        {
            Id = result.Id,
            Name = result.DisplayName,
            Image = result?.Images?.FirstOrDefault()?.Url
        }};
    }
    return Page();
}
```

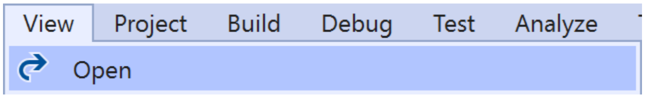
The **method** for `OnPostGetUserProfileAsync` is used to get the **profile** of a **user** using the `Value` of the **User Id** and populate the **property** for `Results` accordingly.

Step 5



In the **Solution Explorer** in the **Pages** section select **Index.cshtml**

Step 6



Then from the **Menu** choose **View** and then **Open**

Step 7

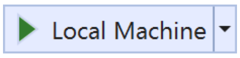
Once in the **Code View** for **Index.cshtml** above `<!-- Spotify Web API User Authorisation -->` enter the following:

```
<ul class="list-group mb-2">
    <li class="list-group-item list-group-item-primary">
        <h5 class="list-group-item-heading">User Profile</h5>
    </li>
    <li class="list-group-item">
        <form asp-page-handler="GetUserProfile" method="post">
            <input asp-for="Value" placeholder="User Id" class="form-control mb-2" />
            <button class="btn btn-primary mb-2">
                Get a User's Profile
            </button>
        </form>
    </li>

    <!-- User Profile -->
</ul>
```

This `form` will **post** to the **method** for `GetUserProfile` with the `Value` of **User Id** and will output to the **Results**.

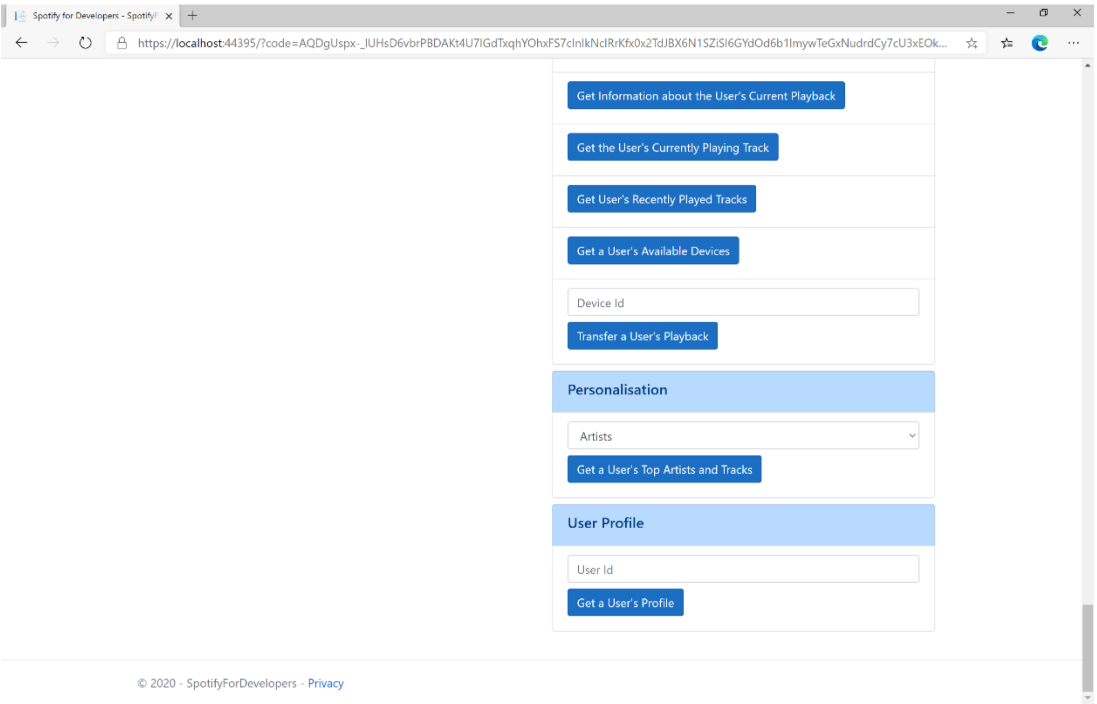
Step 8



Finally, in **Visual Studio 2019** select **IIS Application** to run the **Web Application**

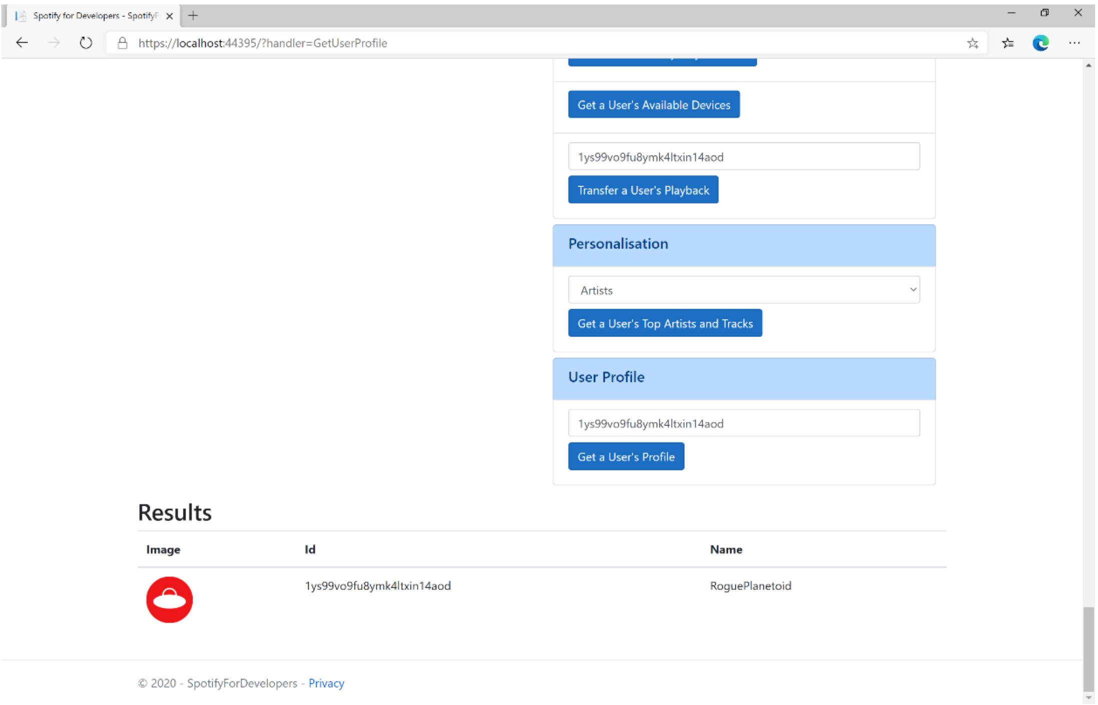
Step 9

Once the **Web Application** is running and you select **Authorisation Code Flow Login** or **Implicit Grant Flow Login** and scroll down you should see something like the following:


Spotify for Developers - SpotifyF
https://localhost:44395/?code=AQDgUspx-_lUHsD6vbrPBDAKt4U7lGdTxqhYOhxFS7cInlkNclRrKfx0x2TdJBX6N1SZiSI6GYdOd6b1lmywTeGxNudrdCy7cU3xEOk...
Get Information about the User's Current Playback
Get the User's Currently Playing Track
Get User's Recently Played Tracks
Get a User's Available Devices
Device Id
Transfer a User's Playback
Personalisation
Artists
Get a User's Top Artists and Tracks
User Profile
User Id
Get a User's Profile
© 2020 - SpotifyForDevelopers - Privacy

Step 10

You can then enter above **Get a User's Profile** your **User Id**, to get this in **Spotify** select your **Username** then **...** and **Copy Spotify URI** your **User Id** will be after **spotify:user:** and then select **Get a User's Profile** and scroll down to view **Results** like the following:


Spotify for Developers - SpotifyF
https://localhost:44395/?handler=GetUserProfile
Get a User's Available Devices
1ys99vo9fu8ymk4ltxin14aod
Transfer a User's Playback
Personalisation
Artists
Get a User's Top Artists and Tracks
User Profile
1ys99vo9fu8ymk4ltxin14aod
Get a User's Profile

Results

Image	Id	Name
	1ys99vo9fu8ymk4ltxin14aod	RoguePlanetoid

© 2020 - SpotifyForDevelopers - Privacy

Step 11

■

You can stop the **web application** in **Visual Studio 2019** by selecting the **Stop debugging** button

Step 12

✕

You can choose to exit **Visual Studio 2019** by selecting the **Close** button in the top right of the **application** as that completes this part of the workshop

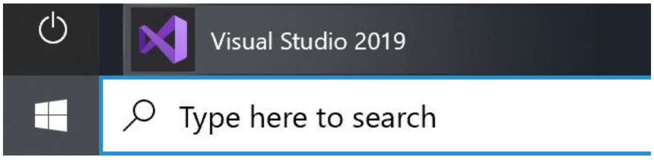
Get Current User's Profile

Get detailed profile information about the current user including the current user’s username.

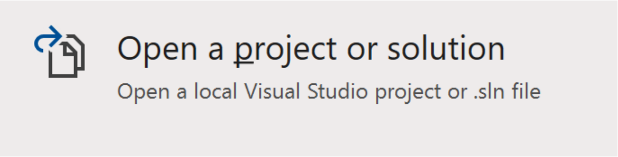
GET https://api.spotify.com/v1/me	
Header	
Authorization	User Token from Spotify Accounts service. Reading the user’s email address requires the **user-read-email** scope or reading country and product subscription level requires the **user-read-private scope**

Header	Response
Success	
HTTP Status 200 OK	List of Tracks wrapped in a Paging Object
Error	
HTTP Status 403 Forbidden	When requesting fields that you don’t have the user’s authorisation to access
Error Code	Error Object

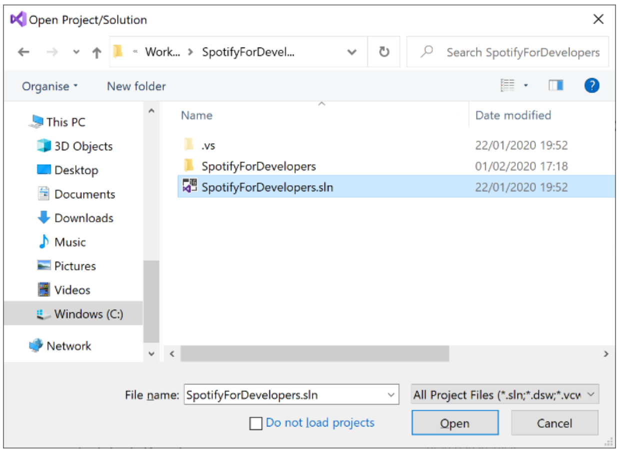
Step 1



If you chose to close **Visual Studio 2019** previously, in **Windows 10** choose **Start**, and then from the **Start Menu** find and select **Visual Studio 2019**


Open a project or solution
Open a local Visual Studio project or .sln file

Once done, from the **Get started** screen for **Visual Studio 2019** select **Open a project or solution**



Name	Date modified
.vs	22/01/2020 19:52
SpotifyForDevelopers	01/02/2020 17:18
SpotifyForDevelopers.sln	22/01/2020 19:52

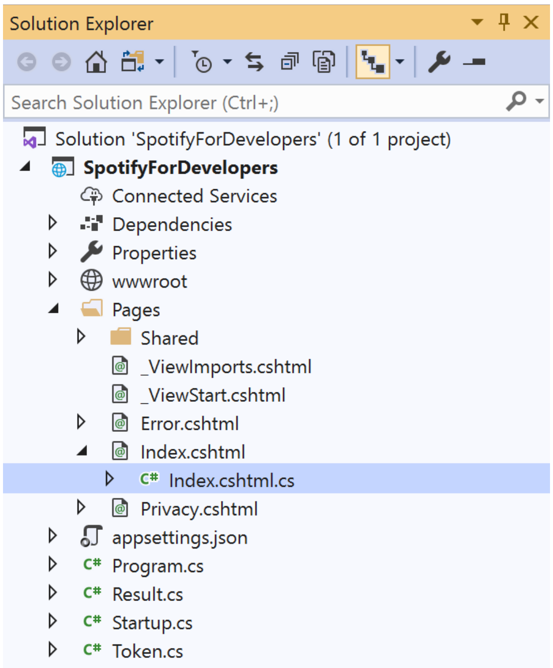
File name: SpotifyForDevelopers.sln

All Project Files (*.sln;*.dsw;*.vcw

□ Do not load projects

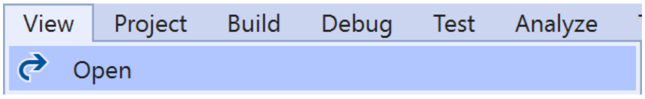
Then locate and select **SpotifyForDevelopers.sln** and select **Open** if you don’t have this file already then please follow the previous parts of the workshop including **Getting Started**, **Authorisation Guide**, **Search & Browse**, **Playlists & Artists**, **Albums & Tracks**, **Episodes & Shows**, **Follow**, **Playlists**, **Library**, **Player** and **Personalisation & User Profile**

Step 2



Once opened, in the **Solution Explorer** open the **Pages** section, then open the **Index.cshtml** section and select **Index.cshtml.cs**

Step 3



Then from the **Menu** choose **View** and then **Open**

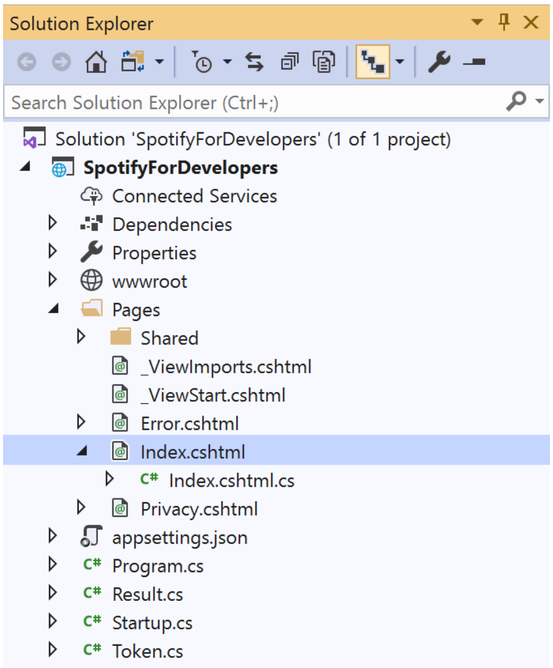
Step 4

In the **Code View** for **Index.cshtml.cs** below the **method** for `public async Task<IActionResult> OnPostGetUserProfileAsync(...) { ... }` enter the following **method**:

```
public async Task<IActionResult> OnPostGetCurrentUserProfileAsync()
{
    LoadToken();
    var result = await Api.GetUserProfileAsync();
    if (result != null)
    {
        Results = new List<Result> { new Result()
        {
            Id = result.Id,
            Name = result.DisplayName,
            Image = result?.Images?.FirstOrDefault()?.Url
        }};
    }
    return Page();
}
```

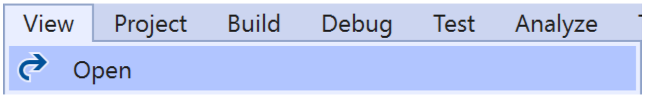
The **method** for `OnPostGetCurrentUserProfileAsync` is used to get the **profile** of the **current user** and populate the **property** for `Results` accordingly.

Step 5



In the **Solution Explorer** in the **Pages** section select **Index.cshtml**

Step 6



Then from the **Menu** choose **View** and then **Open**

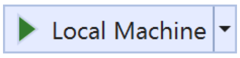
Step 7

Once in the **Code View** for **Index.cshtml** above `<!-- User Profile -->` enter the following:

```
<li class="list-group-item">
    <form asp-page-handler="GetCurrentUserProfile" method="post">
        <button class="btn btn-primary mb-2">
            Get Current User's Profile
        </button>
    </form>
</li>
```

This `form` will **post** to the **method** for `GetCurrentUserProfile` and will output to the **Results**.

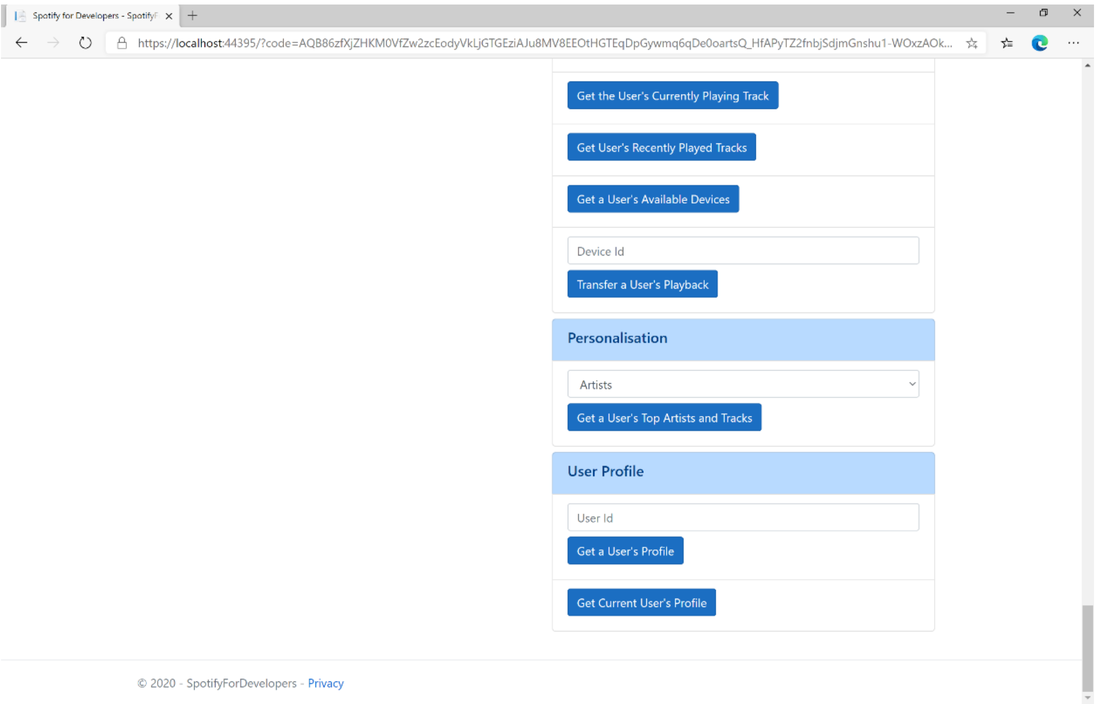
Step 8



Finally, in **Visual Studio 2019** select **IIS Application** to run the **Web Application**

Step 9

Once the **Web Application** is running and you select **Authorisation Code Flow Login** or **Implicit Grant Flow Login** and scroll down you should see something like the following:

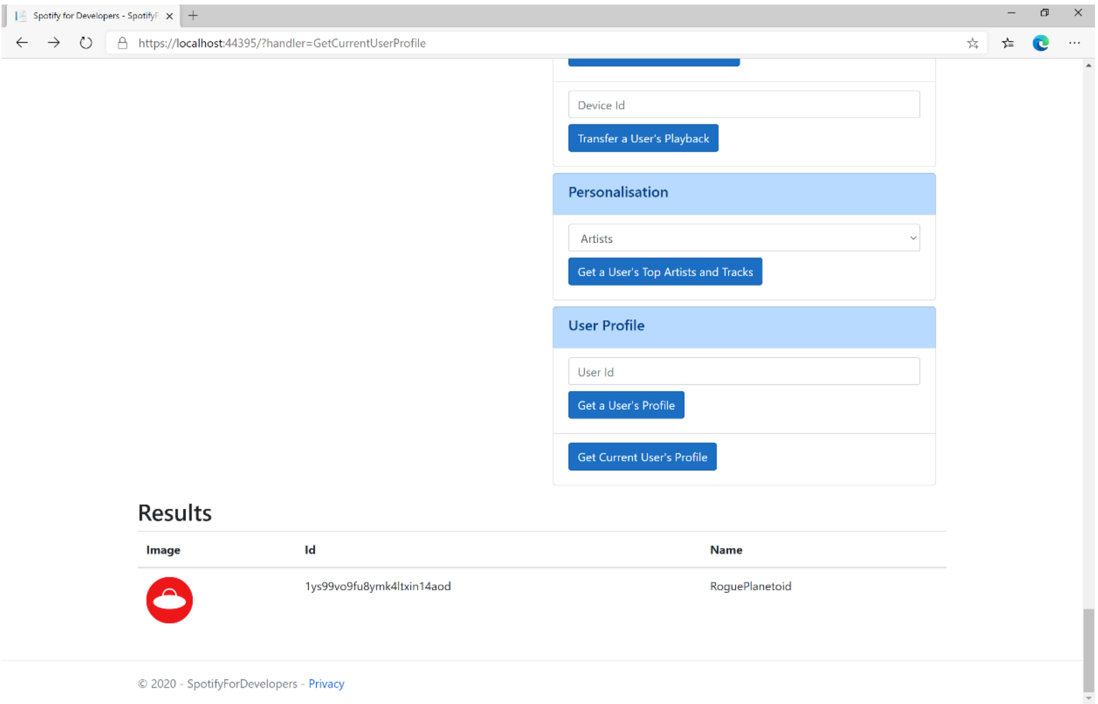

Spotify for Developers - SpotifyF
Get the User's Currently Playing Track
Get User's Recently Played Tracks
Get a User's Available Devices
Device Id
Transfer a User's Playback
Personalisation
Artists
Get a User's Top Artists and Tracks
User Profile
User Id
Get a User's Profile
Get Current User's Profile

https://localhost:44395/?code=AQB86zfXjZHKM0VfZw2zcEodyVkLjGTGEziAJu8MV8EEOtHGTEqDpGywmq6qDe0oartsQ_HfAPyTZ2fnbjSdjmGnshu1-WOxzAOk...

© 2020 - SpotifyForDevelopers - Privacy

Step 10

You can then select **Get Current User's Profile** and scroll down to view **Results** like the following:


Spotify for Developers - SpotifyF
Device Id
Transfer a User's Playback
Personalisation
Artists
Get a User's Top Artists and Tracks
User Profile
User Id
Get a User's Profile
Get Current User's Profile

https://localhost:44395/?handler=GetCurrentUserProfile

Results

Image	Id	Name
	1ys99vo9fu8ymk4ltxin14aod	RoguePlanetoid

© 2020 - SpotifyForDevelopers - Privacy

Step 11

■

You can stop the **web application** in **Visual Studio 2019** by selecting the **Stop debugging** button

Step 12

✕

You can exit **Visual Studio 2019** by selecting the **Close** button in the top right of the **application** as that completes this workshop

Requests & Responses

Spotify API allows applications to retrieve Spotify content such as album data, playlists and more in UTF-8 format with endpoints using standard HTTPS requests based on simple REST principles using the appropriate HTTP verbs for each action and return JSON formatted metadata about artists, albums and tracks. It also provides access to user related data, like playlists and music the user saves in their library.

Request Verbs

Request Method	Description
GET	Retrieves Resources
POST	Creates Resources
PUT	Changes and/or replaces Resources or Collections
DELETE	Deletes Resources

Response Codes

Response Codes	Description
200	**OK** - Request has succeeded. The client can read the result of the request in the body and the headers of the response.
201	**Created** - Request has been fulfilled and resulted in a new resource being created.
202	**Accepted** - Request has been accepted for processing, but processing has not been completed.
204	**No Content** - Request has succeeded but returns no message body.
304	**Not Modified** - If response has not changed means a cached version is still good
400	**Bad Request** - Request could not be understood by the server due to malformed syntax. The message body will contain more information.
401	**Unauthorized** - Request requires user authentication or, if the request included authorisation credentials, authorisation has been refused for those credentials.
403	**Forbidden** - The server understood the request but is refusing to fulfil it.
404	**Not Found** - The requested resource could not be found. This error can be due to a temporary or permanent condition.
429	**Too Many Requests** - Rate Limiting has been applied.
500	**Internal Server Error** - Should never receive this error as Spotify should catch them
502	**Bad Gateway** - Server was acting as a gateway or proxy and received an invalid response from upstream server.
503	**Service Unavailable** - Server is currently unable to handle request due to a temporary condition which will be alleviated after some delay. You can choose to resend the request again.