



Tutorialr.com

# Spotify for Developers Player

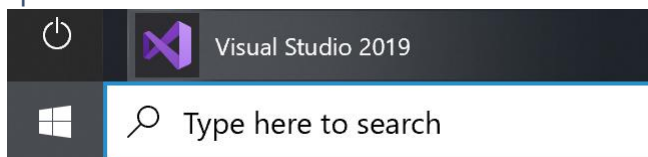
## Add an Item to the User's Playback Queue

Add an item to the end of the user's current playback queue.

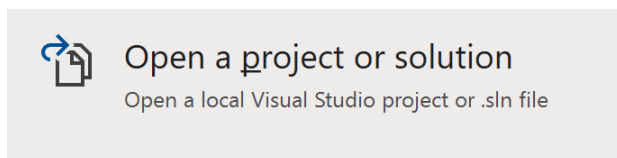
POST <a href="https://api.spotify.com/v1/me/player/queue">https://api.spotify.com/v1/me/player/queue</a>	
Header	
<b>Authorization</b>	User Token from Spotify Accounts service with <b>user-modify-playback-state</b> scope
Query Parameter	
<b>uri</b>	Uri of the item to add to the queue. Must be a track or an episode uri
<b>device_id</b>	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

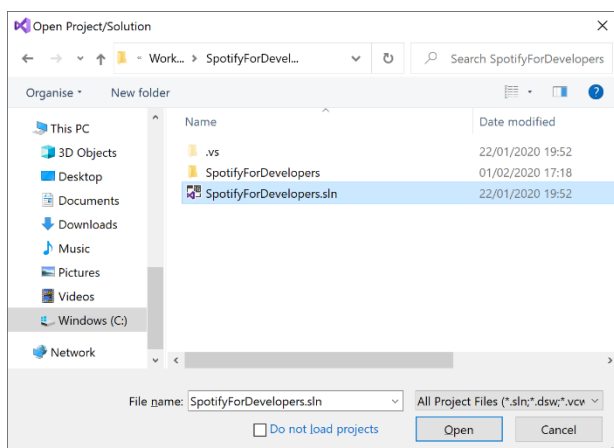
### Step 1



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

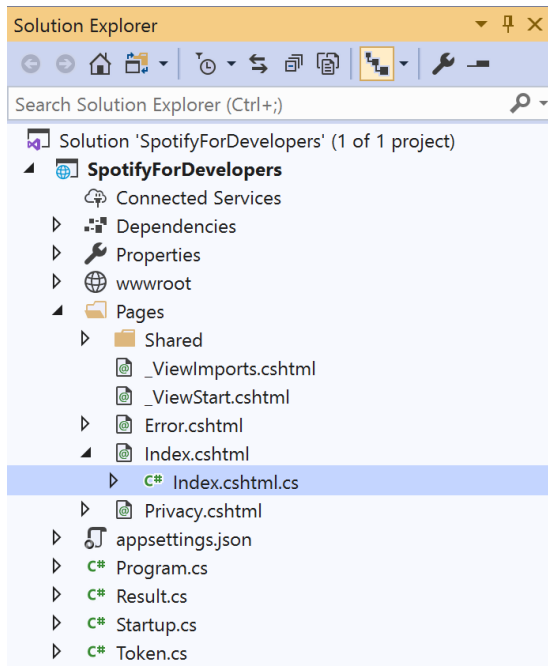


Once done, from the **Get started** screen for **Visual Studio 2019** select **Open a project or solution**



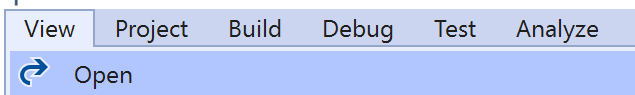
Then locate and select **SpotifyForDevelopers.sln** and select **Open** if you don't have this file already then please follow the previous parts of the workshop including **Getting Started, Authorisation Guide, Search & Browse, Playlists & Artists, Albums & Tracks, Episodes & Shows, Follow, Playlists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

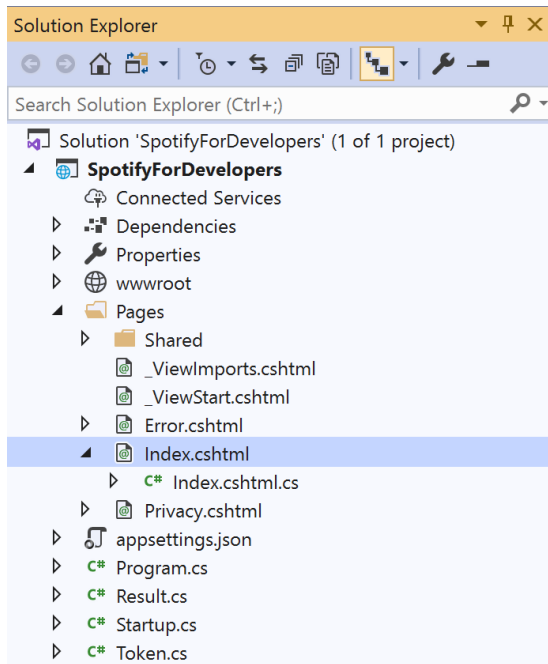
## Step 4

In the **Code View** for **Index.cshtml.cs** below the **method** for `public async Task<IActionResult> 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();
}
```

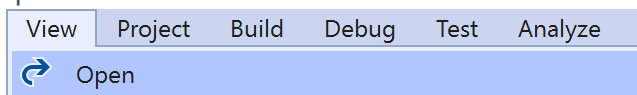
The **method** for `OnPostUserPlaybackAddToQueueAsync` **adds** the **track** or **episode** to the **playback queue** with Value of the **Item Id** of the **Track Id** or **Episode Id** and the Option of **Track** or **Episode** and populate the **property** for Results of the **success** of the operation 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**

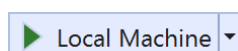
## Step 7

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**.

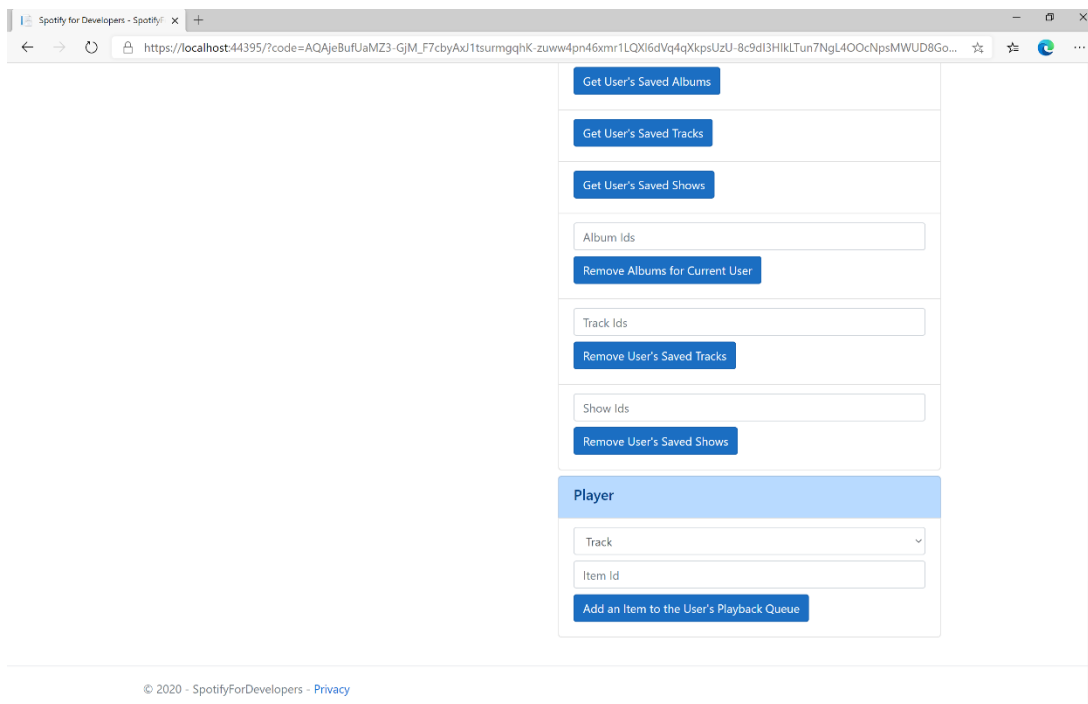
## Step 8



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

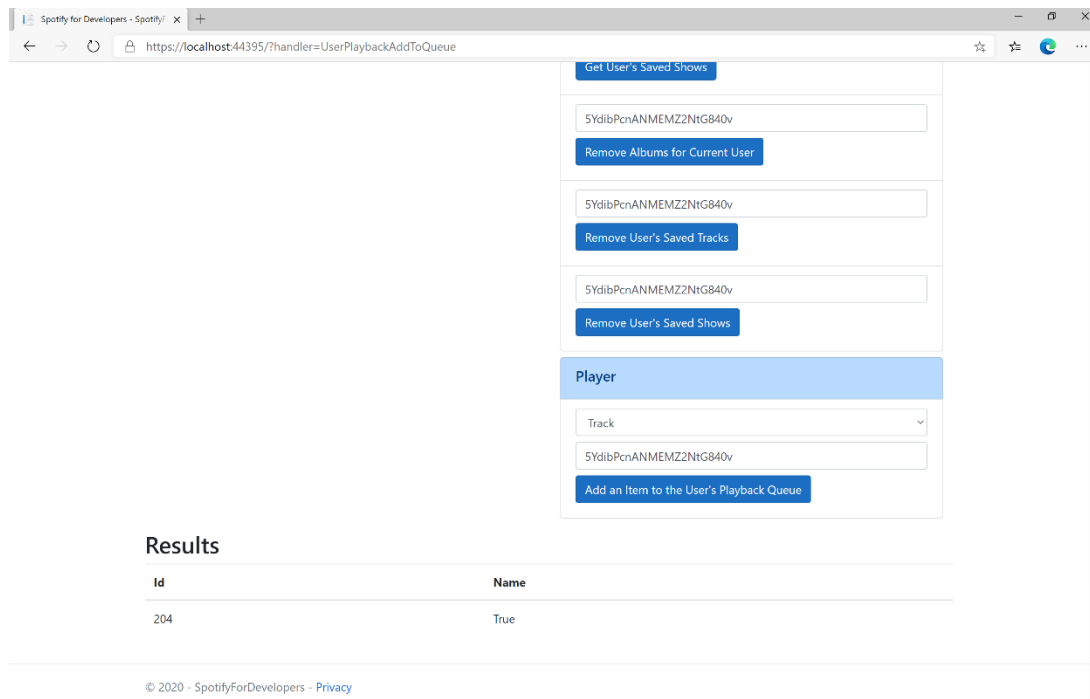
## Step 9

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:



## 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:



## 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

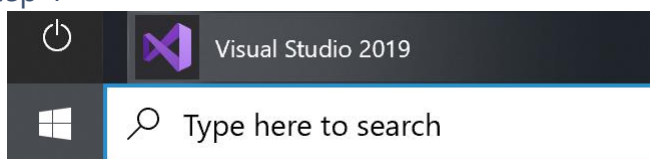
## Start/Resume a User's Playback

Start a new context or resume current playback on the user's active device.

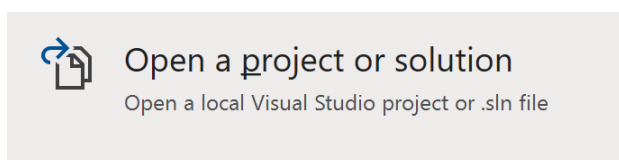
PUT <a href="https://api.spotify.com/v1/me/player/play">https://api.spotify.com/v1/me/player/play</a>	
Header	
<b>Authorization</b>	User Token from Spotify Accounts service with <b>user-modify-playback-state</b> scope
Query Parameter	
device_id	Device Id to target, if not supplied the user's current 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

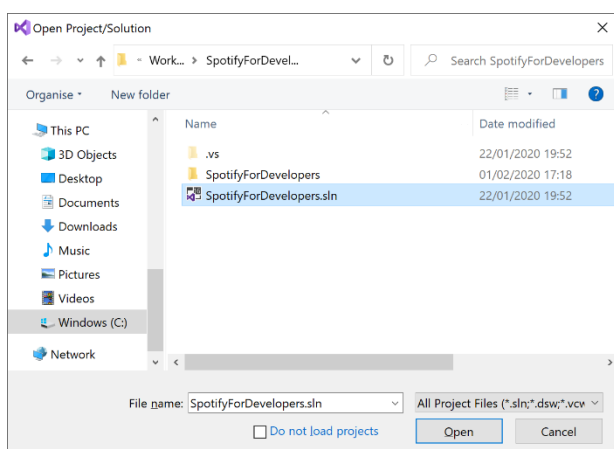
### Step 1



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**

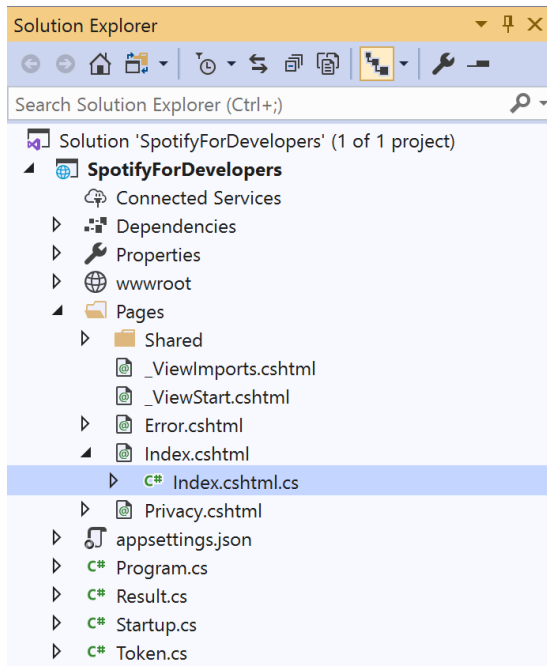


Once done, from the **Get started** screen for **Visual Studio 2019** select **Open a project or solution**



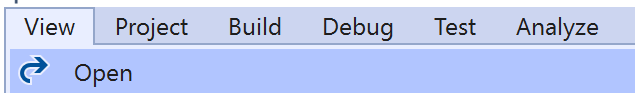
Then locate and select **SpotifyForDevelopers.sln** and select **Open** if you don't have this file already then please follow the previous parts of the workshop including **Getting Started, Authorisation Guide, Search & Browse, Playlists & Artists, Albums & Tracks, Episodes & Shows, Follow, Playlists,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

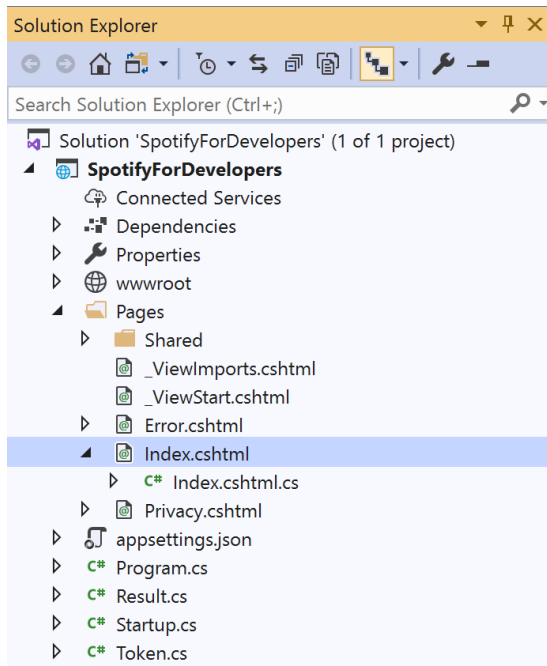
## Step 4

In the **Code View** for **Index.cshtml.cs** below the **method** for `public async Task<IActionResult> 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();
}
```

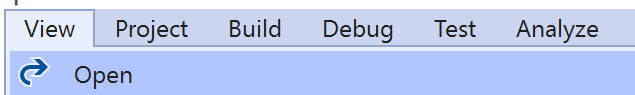
The **method** for `OnPostUserPlaybackStartResumeAsync` will **start** the **playback** with the Value of the **Item Id** of the Option of an **Album**, **Track**, **Artist** or **Playlist** and populate the **property** for Results of the **success** of the operation accordingly.

## Step 5



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

## Step 6



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

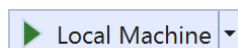
## Step 7

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

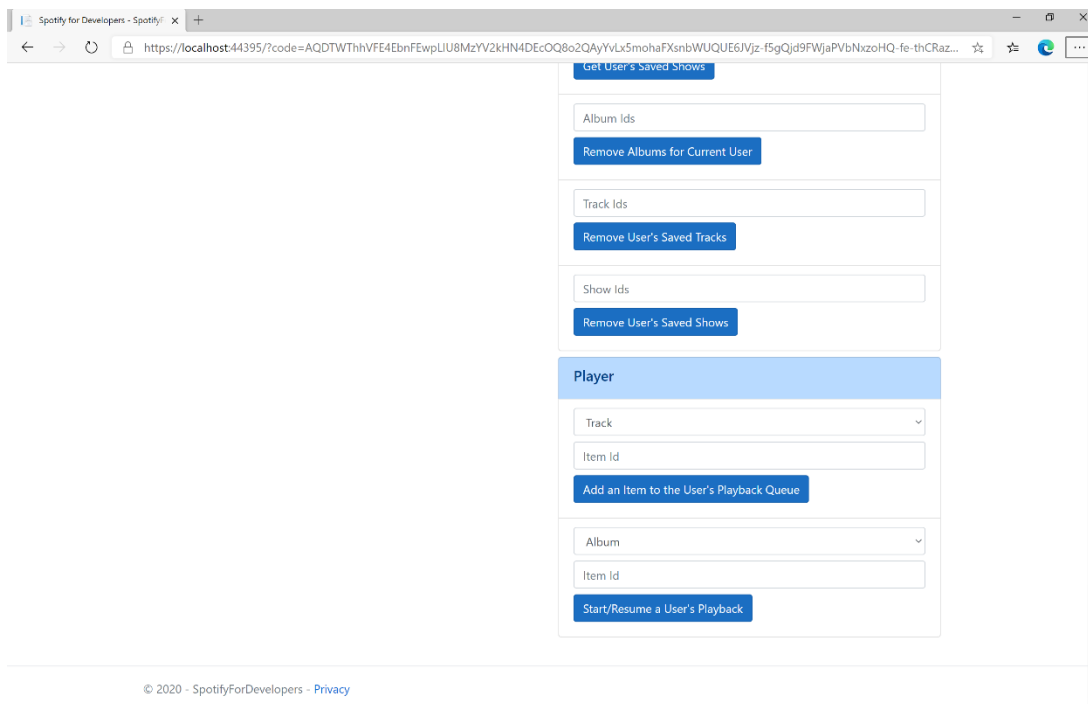


Finally, in **Visual Studio 2019** select **IIS Application** to run the **Web Application**



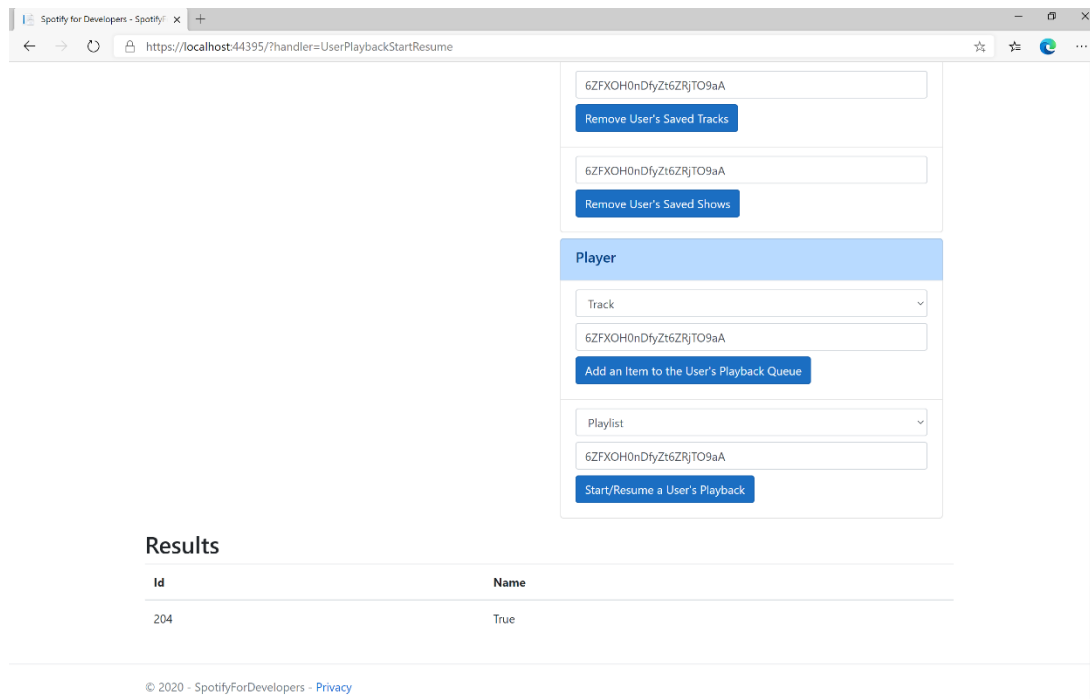
## Step 9

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:



## 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:



## 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

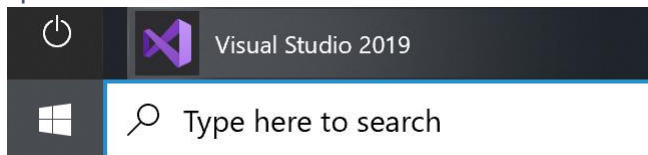
## Pause a User's Playback

Pause playback on the user's account.

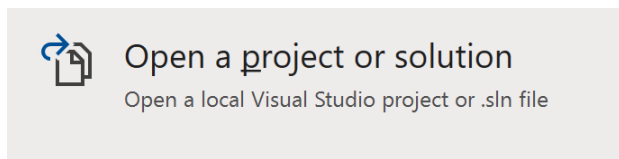
PUT <a href="https://api.spotify.com/v1/me/player/pause">https://api.spotify.com/v1/me/player/pause</a>	
Header	
<b>Authorization</b>	User Token from Spotify Accounts service with <b>user-modify-playback-state</b> scope
Query Parameter	
device_id	Device Id to target, if not supplied the user's current 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

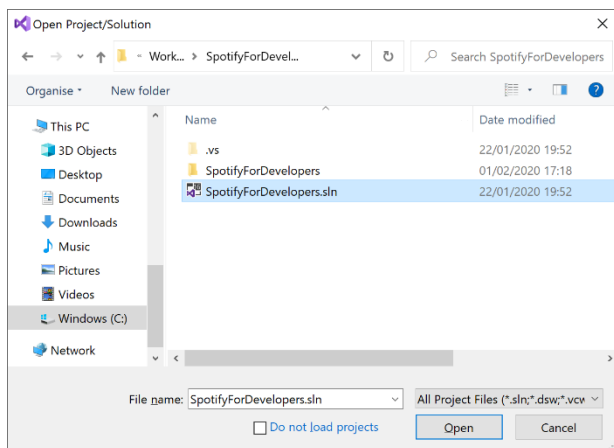
### Step 1



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**

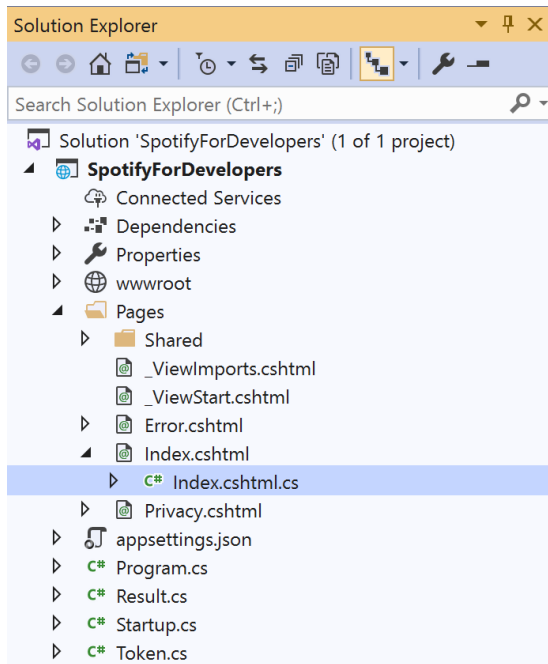


Once done, from the **Get started** screen for **Visual Studio 2019** select **Open a project or solution**



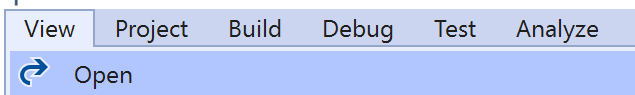
Then locate and select **SpotifyForDevelopers.sln** and select **Open** if you don't have this file already then please follow the previous parts of the workshop including **Getting Started, Authorisation Guide, Search & Browse, Playlists & Artists, Albums & Tracks, Episodes & Shows, Follow, Playlists, 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> 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();
}
```

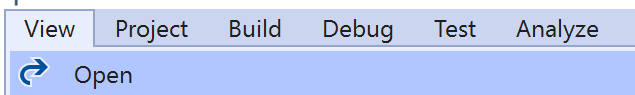
The **method** for `OnPostUserPlaybackPauseAsync` will **pause** any **playback** and populate the **property** for `Results` of the **success** of the operation accordingly.

## Step 5



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

## Step 6



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

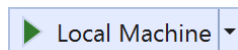
## Step 7

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**.

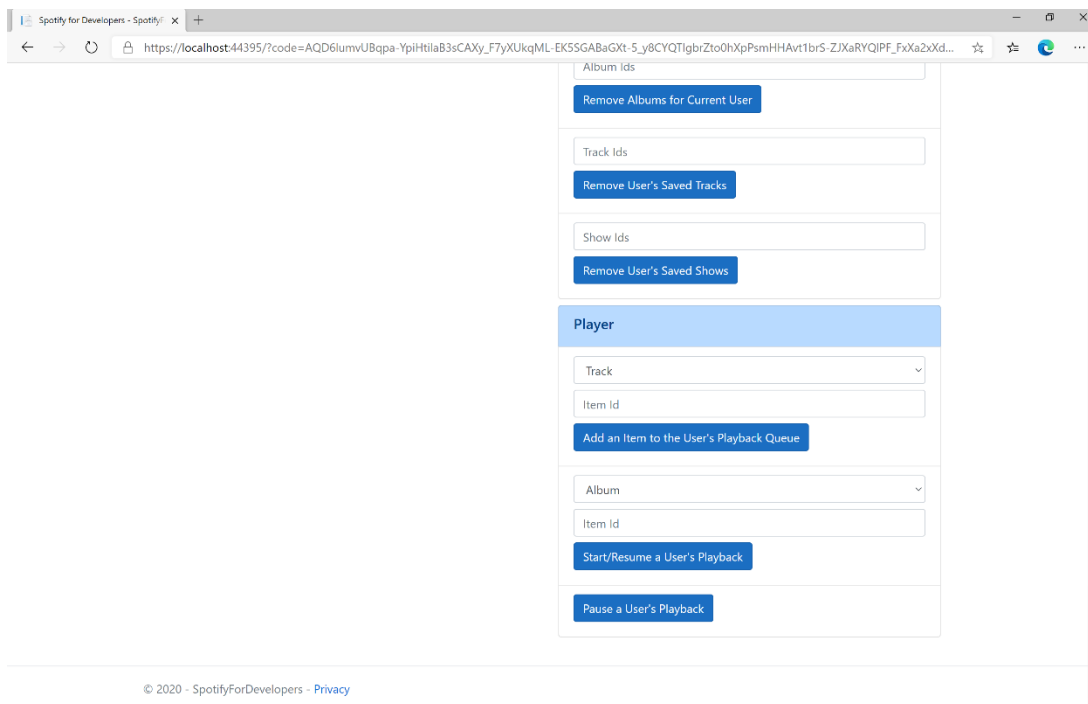
## Step 8



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

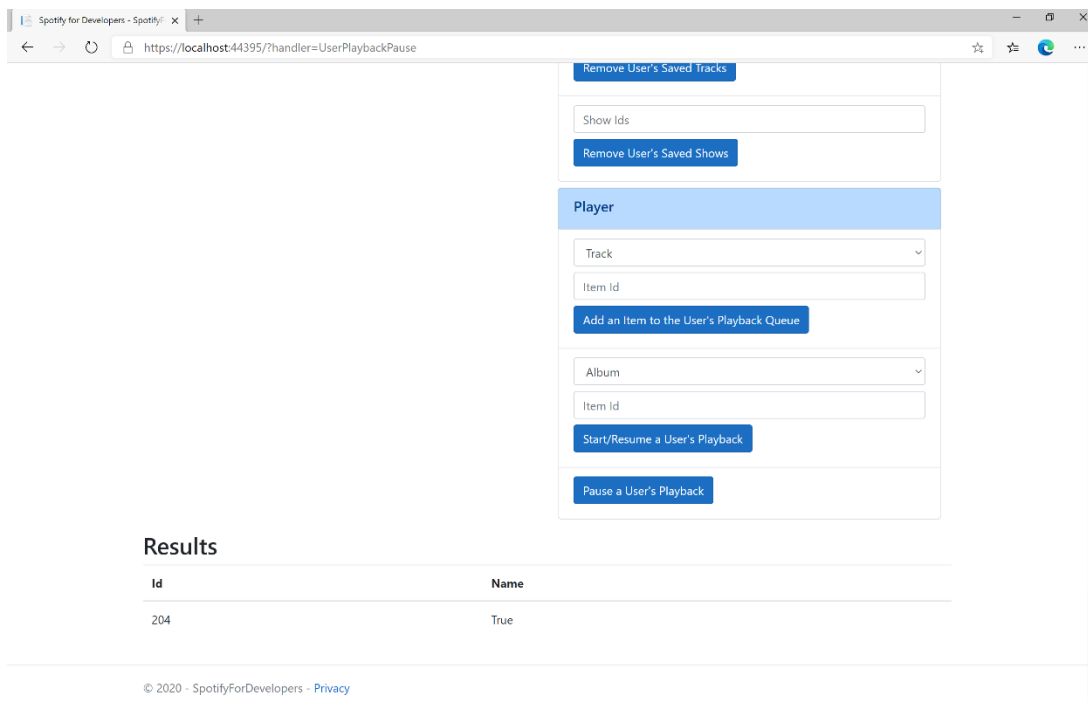
## Step 9

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:



## 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:



## 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

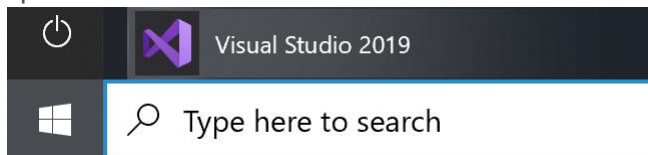
## Skip User's Playback to Next Track

Skips to next track in the user's queue.

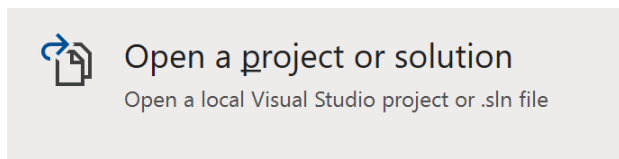
POST <a href="https://api.spotify.com/v1/me/player/next">https://api.spotify.com/v1/me/player/next</a>	
Header	
<b>Authorization</b>	User Token from Spotify Accounts service with <b>user-modify-playback-state</b> scope
Query Parameter	
device_id	Device Id to target, if not supplied the user's current 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

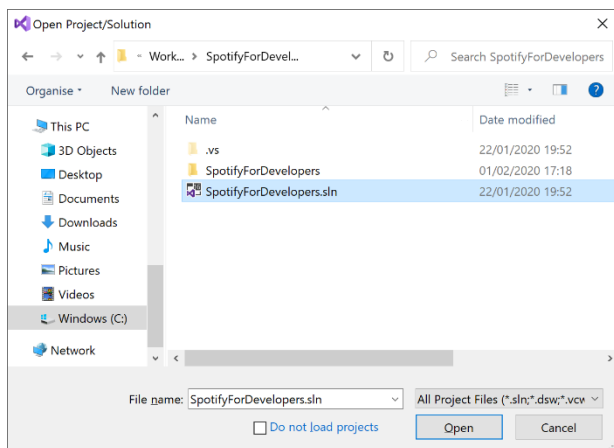
### Step 1



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**

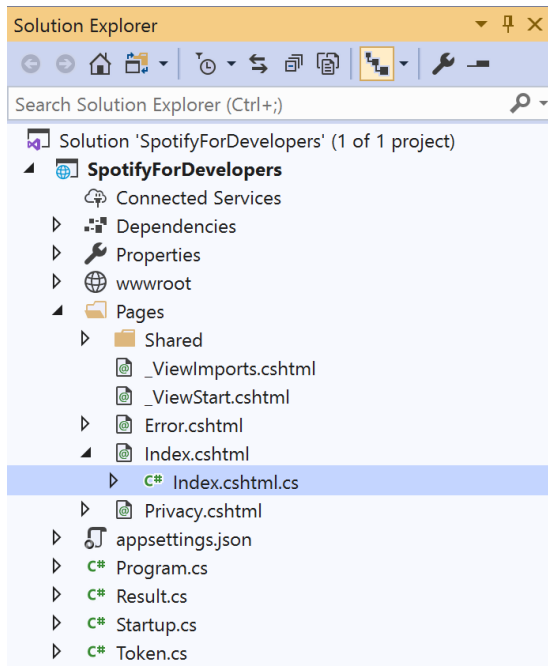


Once done, from the **Get started** screen for **Visual Studio 2019** select **Open a project or solution**



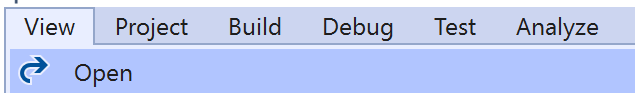
Then locate and select **SpotifyForDevelopers.sln** and select **Open** if you don't have this file already then please follow the previous parts of the workshop including **Getting Started, Authorisation Guide, Search & Browse, Playlists & Artists, Albums & Tracks, Episodes & Shows, Follow, Playlists,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

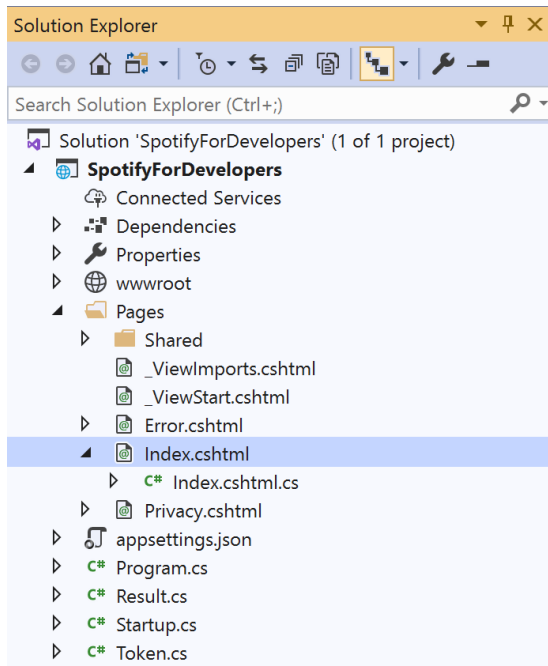
## Step 4

In the **Code View** for **Index.cshtml.cs** below the **method** for `public async Task<IActionResult> 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**

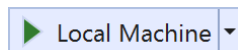
## Step 7

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

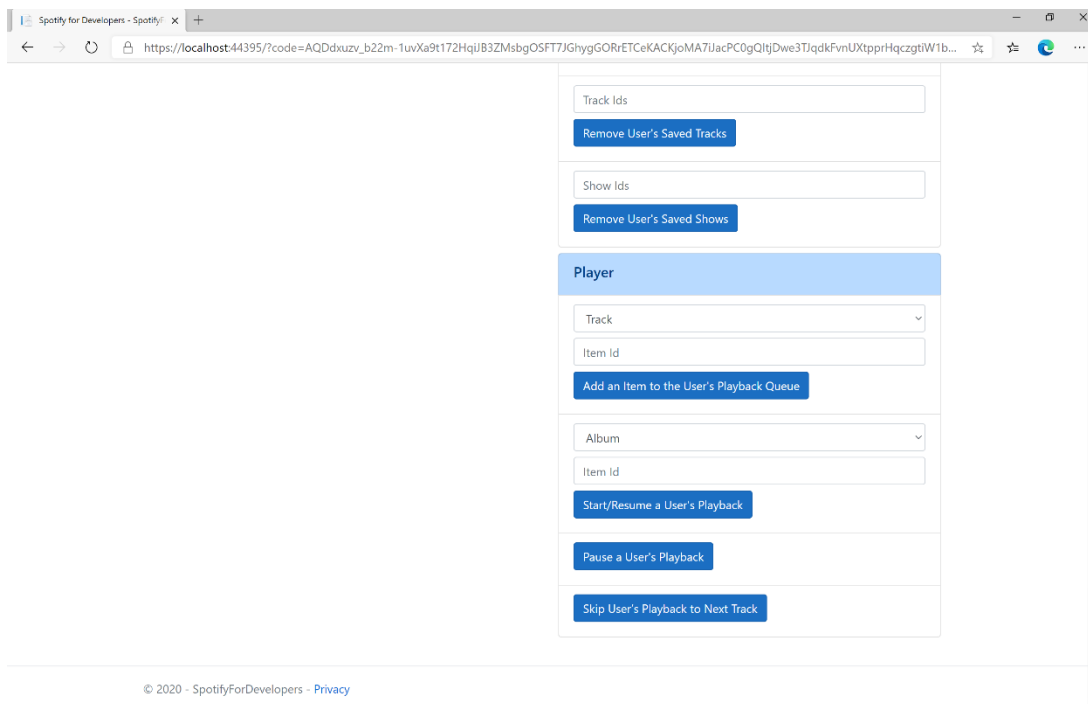


Finally, in **Visual Studio 2019** select **IIS Application** to run the **Web Application**



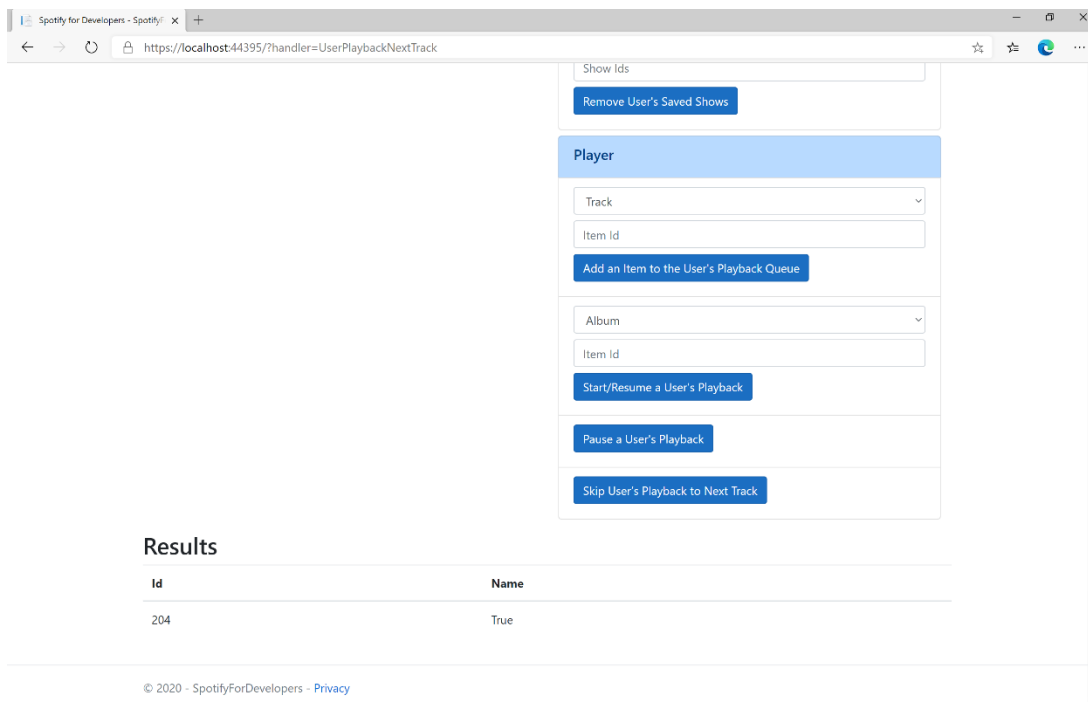
## Step 9

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:



## 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:



## 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

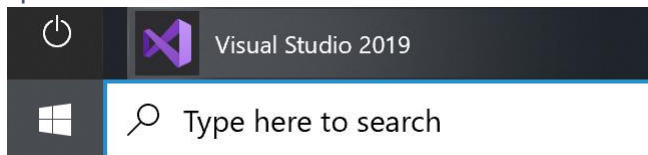
## Skip User's Playback to Previous Track

Skips to previous track in the user's queue.

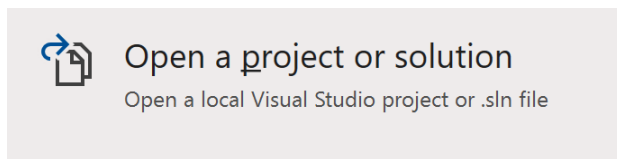
POST <a href="https://api.spotify.com/v1/me/player/previous">https://api.spotify.com/v1/me/player/previous</a>	
Header	
<b>Authorization</b>	User Token from Spotify Accounts service with <b>user-modify-playback-state</b> scope
Query Parameter	
device_id	Device Id to target, if not supplied the user's 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

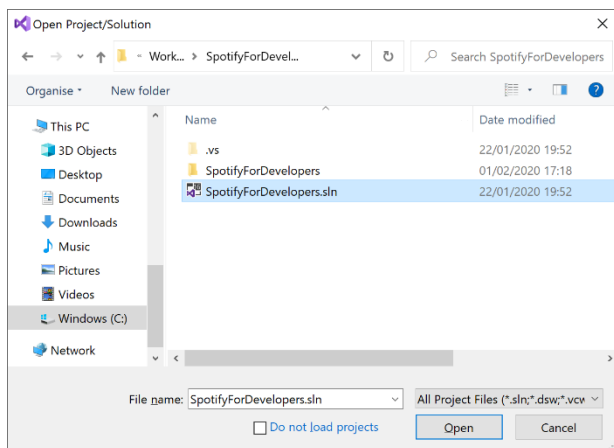
### Step 1



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**

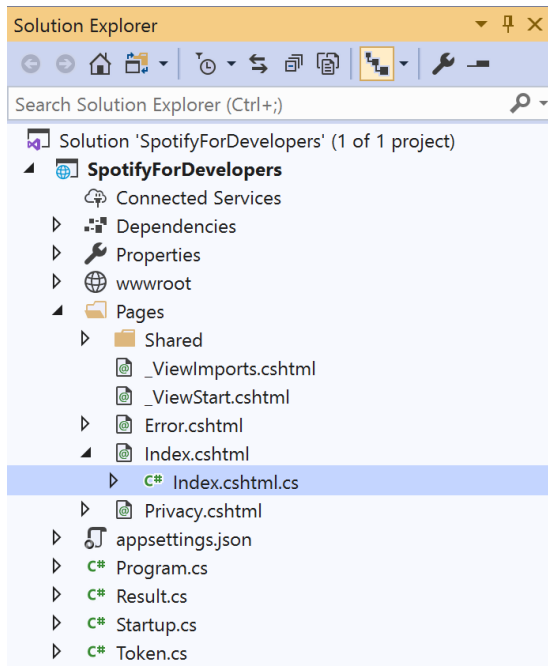


Once done, from the **Get started** screen for **Visual Studio 2019** select **Open a project or solution**



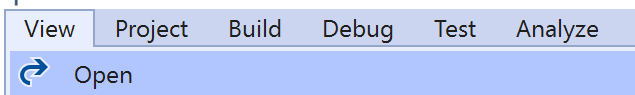
Then locate and select **SpotifyForDevelopers.sln** and select **Open** if you don't have this file already then please follow the previous parts of the workshop including **Getting Started, Authorisation Guide, Search & Browse, Playlists & Artists, Albums & Tracks, Episodes & Shows, Follow, Playlists, 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> 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();
}
```

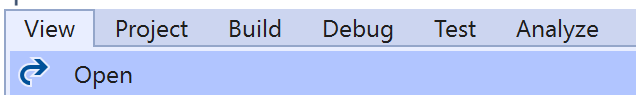
The **method** for `OnPostUserPlaybackPreviousTrackAsync` will **skip** to the **previous track** in **playback** and populate the **property** for **Results** of the **success** of the operation accordingly.

## Step 5



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

## Step 6



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

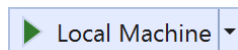
## Step 7

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**.

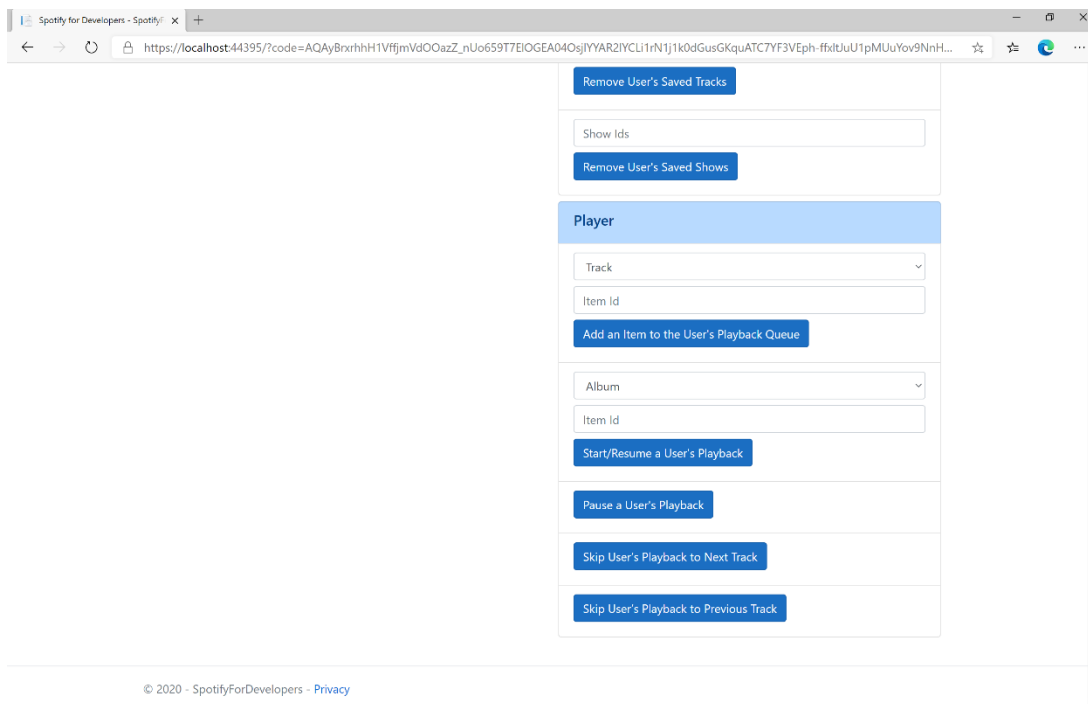
## Step 8



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

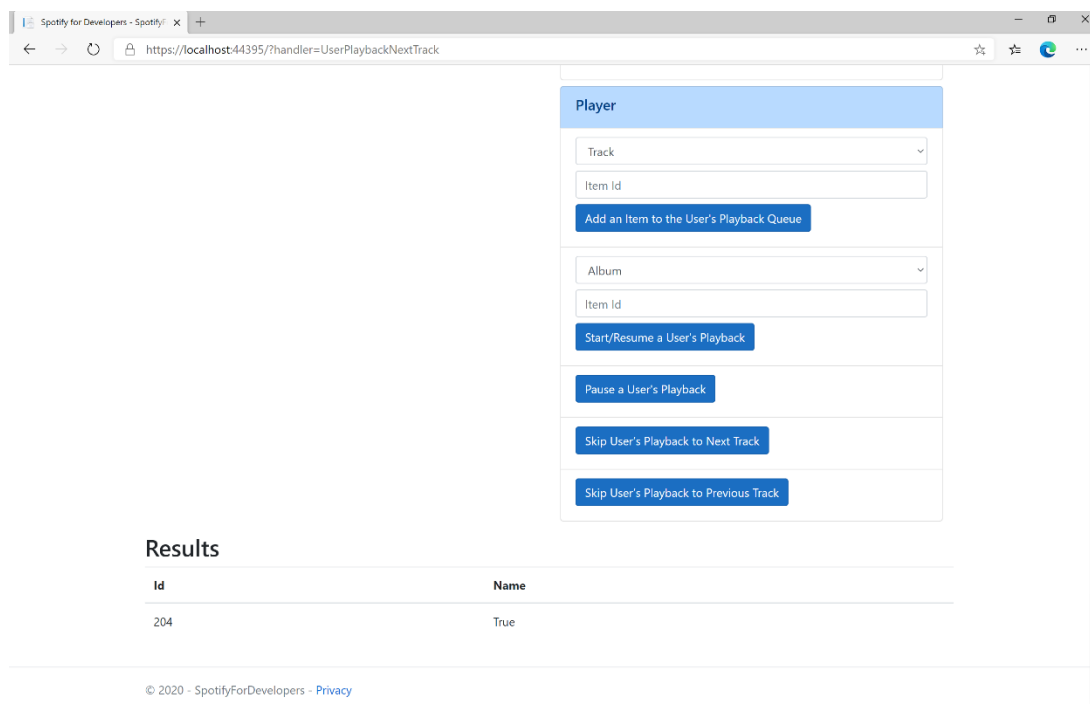
## Step 9

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:



## 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:



## 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

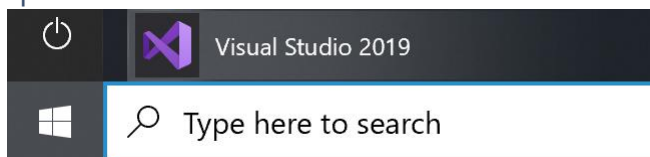
## Seek to Position in Currently Playing Track

Seeks to the given position in the user's currently playing track.

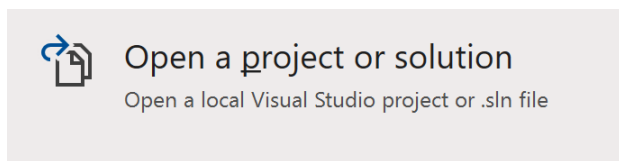
PUT <a href="https://api.spotify.com/v1/me/player/seek">https://api.spotify.com/v1/me/player/seek</a>	
Header	
<b>Authorization</b>	User Token from Spotify Accounts service with <b>user-modify-playback-state</b> scope
Query Parameter	
<b>position_ms</b>	Position in milliseconds to seek to, must be a positive number
<b>device_id</b>	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

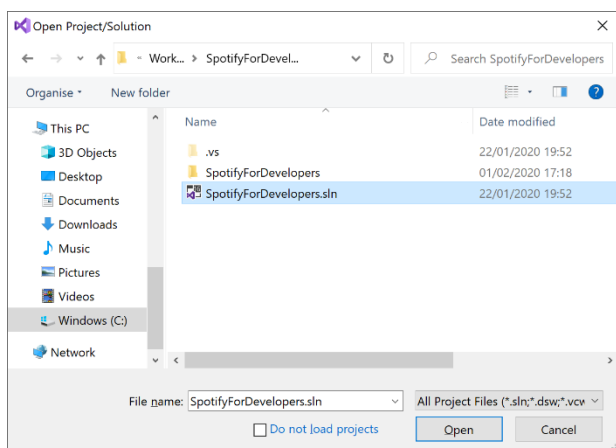
### Step 1



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**

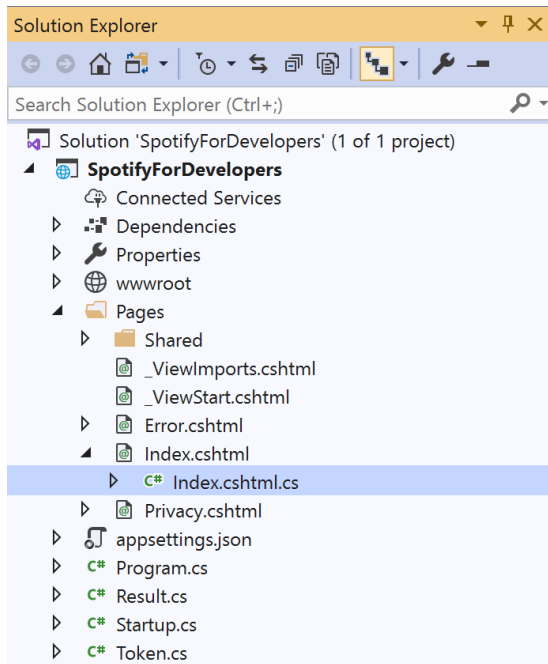


Once done, from the **Get started** screen for **Visual Studio 2019** select **Open a project or solution**



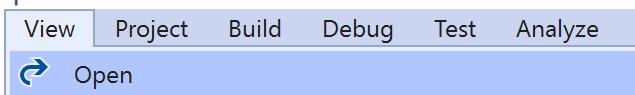
Then locate and select **SpotifyForDevelopers.sln** and select **Open** if you don't have this file already then please follow the previous parts of the workshop including **Getting Started, Authorisation Guide, Search & Browse, Playlists & Artists, Albums & Tracks, Episodes & Shows, Follow, Playlists,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

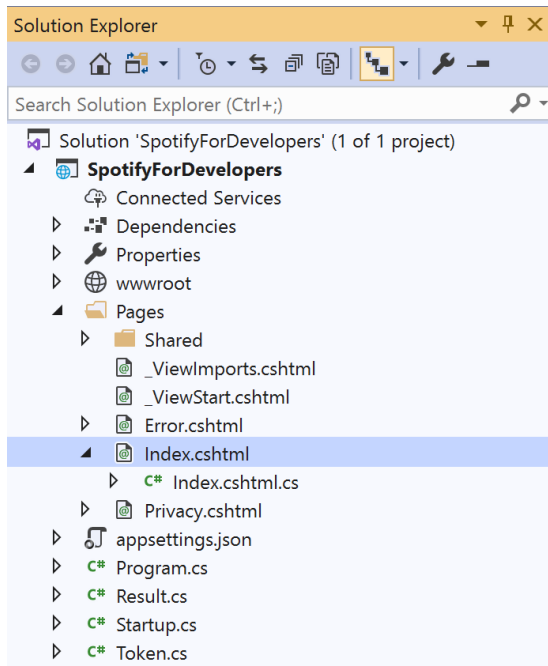
## Step 4

In the **Code View** for **Index.cshtml.cs** below the **method** for `public async Task<IActionResult> 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();
}
```

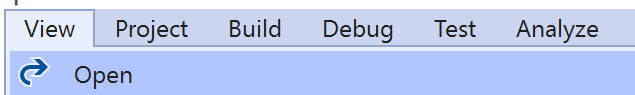
The **method** for `OnPostUserPlaybackSeekTrackAsync` is used to **seek** to the **position** of a currently playing **track** with the Option of the **Position Seconds** of the **track** and populate the **property** for Results of the **success** of the operation accordingly.

## Step 5



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

## Step 6



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

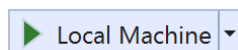
## Step 7

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

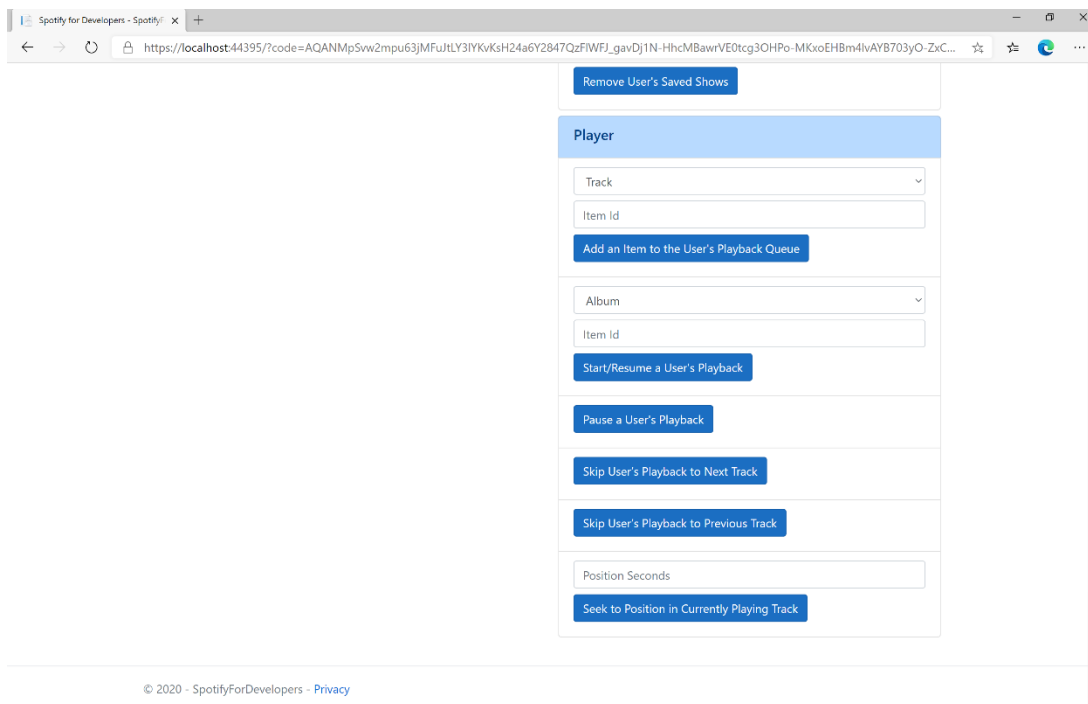


Finally, in **Visual Studio 2019** select **IIS Application** to run the **Web Application**



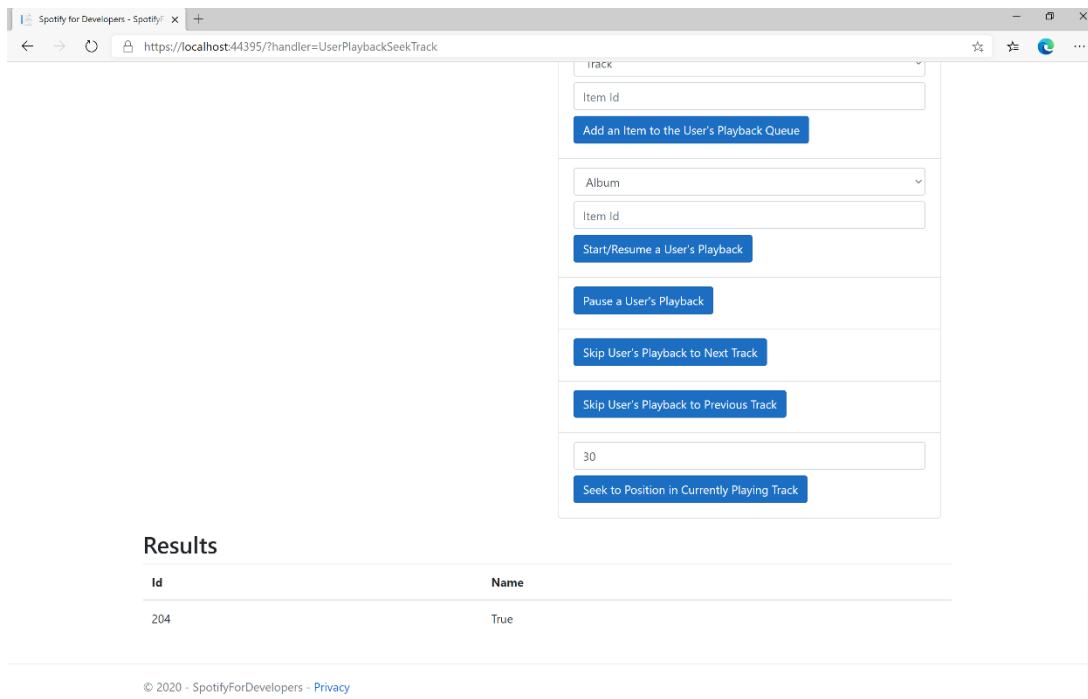
## Step 9

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:



## 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:



## 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

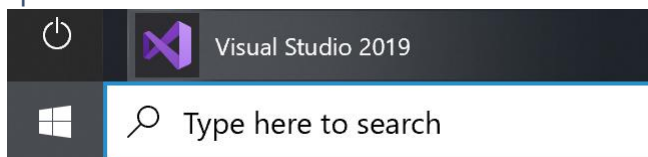
## Set Volume for User's Playback

Set the volume for the user's current playback device.

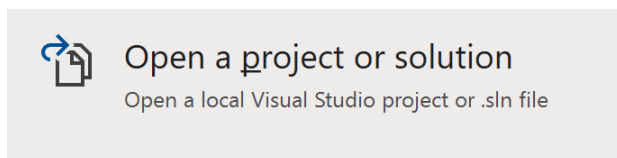
PUT <a href="https://api.spotify.com/v1/me/player/volume">https://api.spotify.com/v1/me/player/volume</a>	
Header	
<b>Authorization</b>	User Token from Spotify Accounts service with <b>user-modify-playback-state</b> scope
Query Parameter	
<b>volume_percent</b>	Volume to set - must be a value from 0 to 100
<b>device_id</b>	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

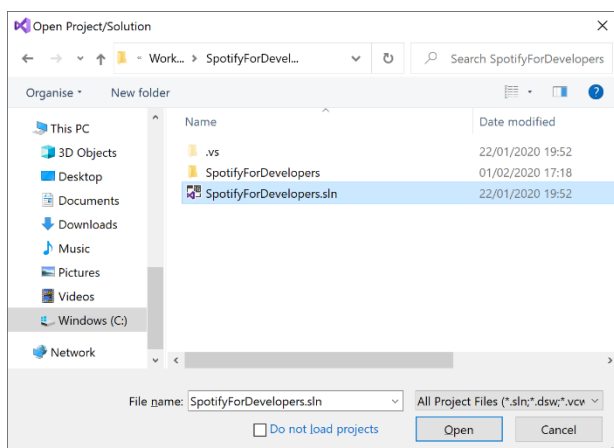
### Step 1



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**

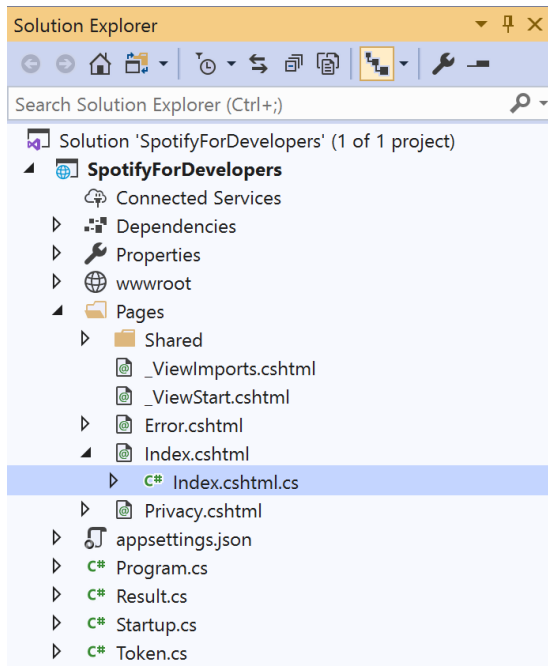


Once done, from the **Get started** screen for **Visual Studio 2019** select **Open a project or solution**



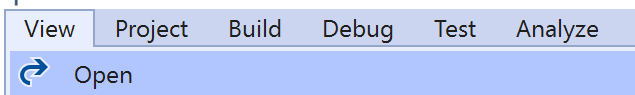
Then locate and select **SpotifyForDevelopers.sln** and select **Open** if you don't have this file already then please follow the previous parts of the workshop including **Getting Started, Authorisation Guide, Search & Browse, Playlists & Artists, Albums & Tracks, Episodes & Shows, Follow, Playlists, 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> 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();
}
```

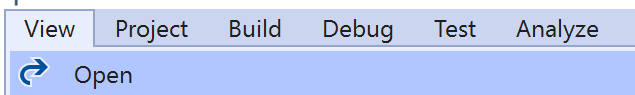
The **method** for `OnPostUserPlaybackSetVolumeAsync` is used set the **volume** of **playback** with the **Option** using a **Range** which by default is **0** to **100** and populate the **property** for **Results** of the **success** of the operation accordingly.

## Step 5



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

## Step 6



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

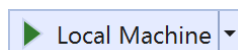
## Step 7

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**.

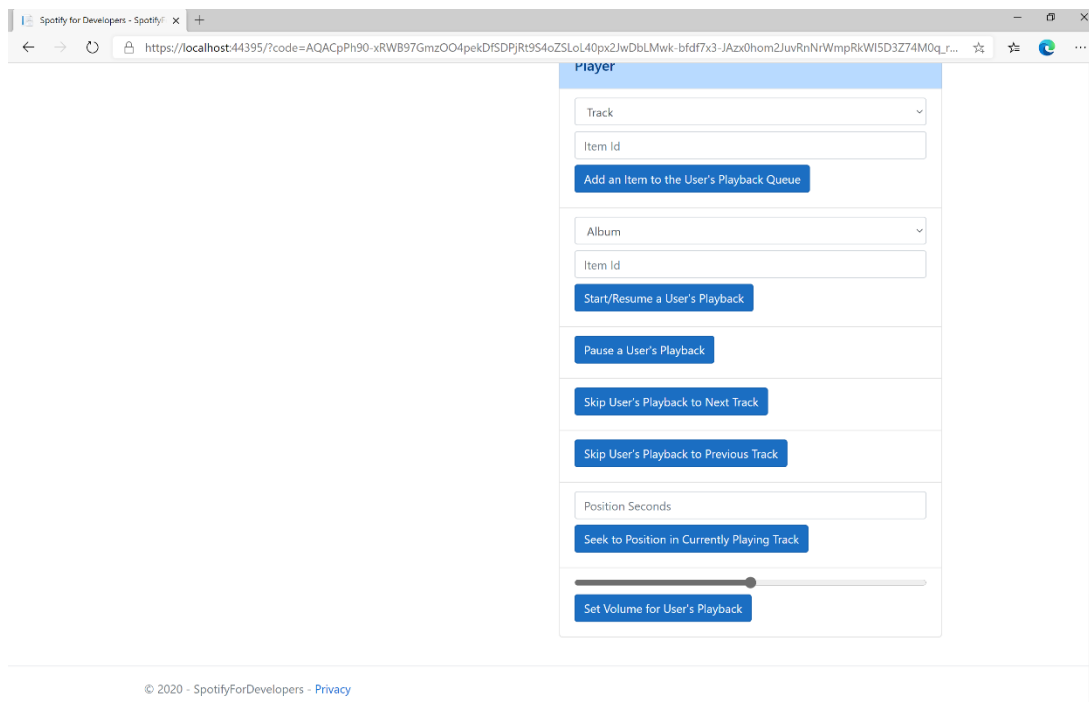
## Step 8



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

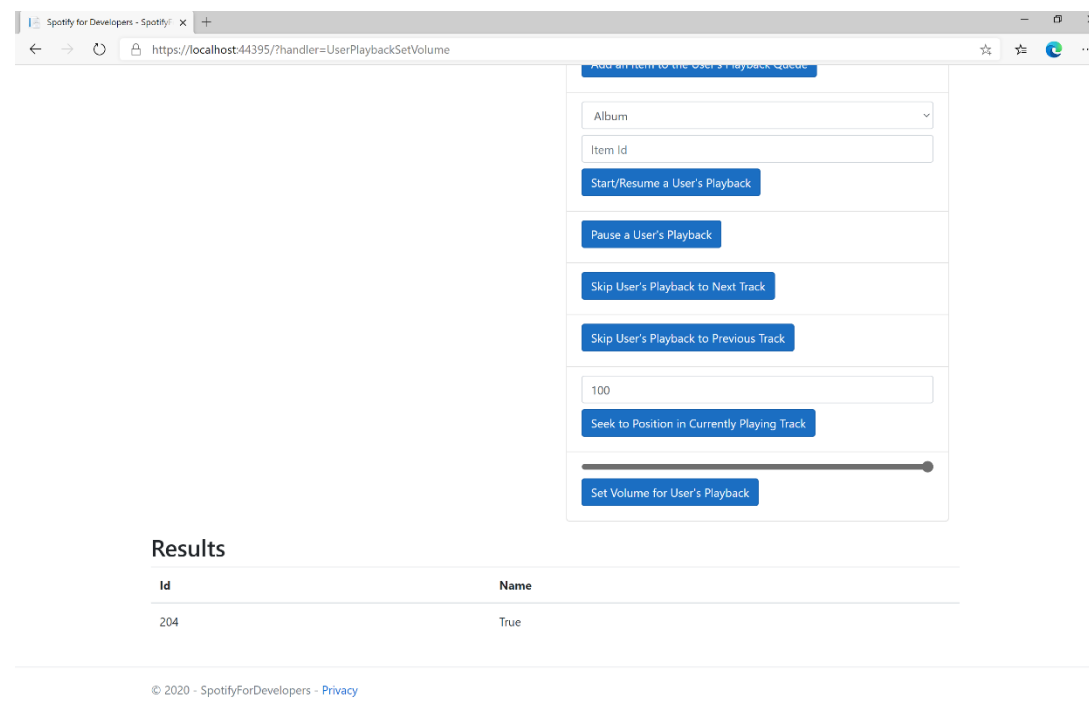
## Step 9

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:



## 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:



## 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

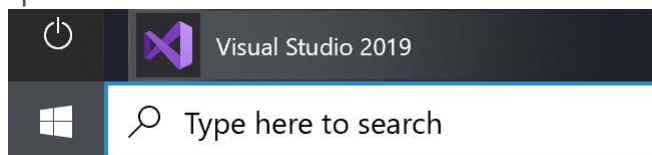
## Toggle Shuffle for User's Playback

Toggle shuffle on or off for user's playback.

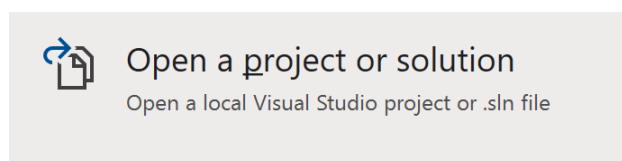
PUT https://api.spotify.com/v1/me/player/shuffle	
Header	
<b>Authorization</b>	User Token from Spotify Accounts service with <b>user-modify-playback-state</b> scope
Query Parameter	
<b>state</b>	If true, shuffle user's playback, if false do not shuffle user's playback
<b>device_id</b>	Device Id to target, 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

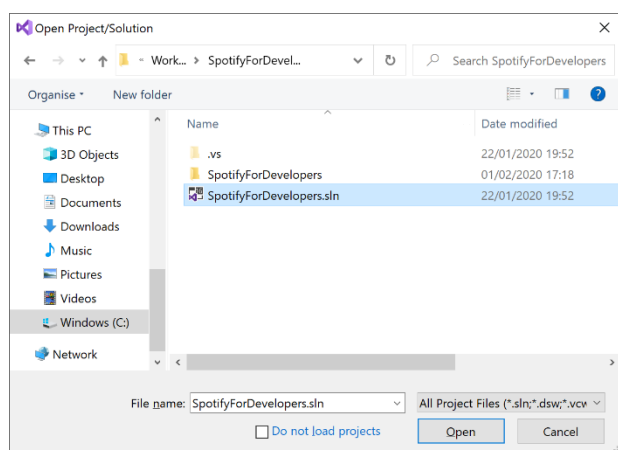
### Step 1



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**

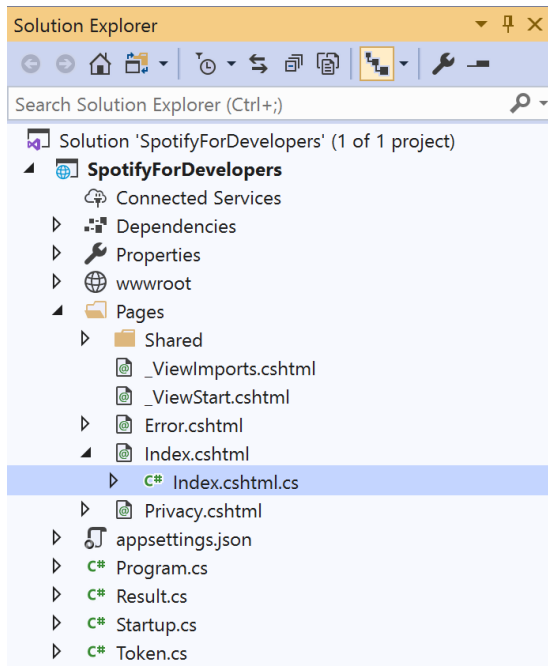


Once done, from the **Get started** screen for **Visual Studio 2019** select **Open a project or solution**



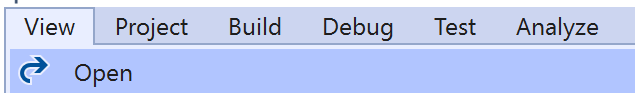
Then locate and select **SpotifyForDevelopers.sln** and select **Open** if you don't have this file already then please follow the previous parts of the workshop including **Getting Started, Authorisation Guide, Search & Browse, Playlists & Artists, Albums & Tracks, Episodes & Shows, Follow, Playlists,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

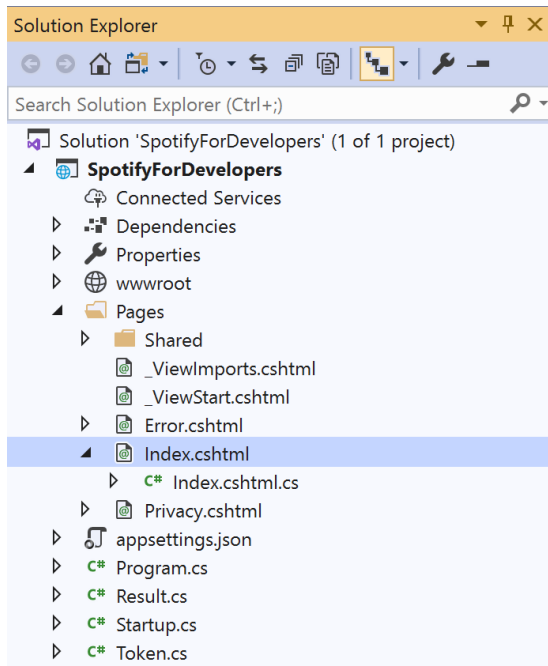
## Step 4

In the **Code View** for **Index.cshtml.cs** below the **method** for `public async Task<IActionResult> 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();
}
```

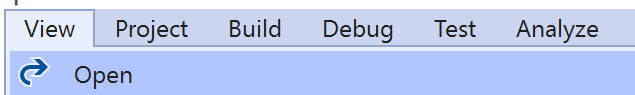
The **method** for `OnPostUserPlaybackToggleShuffleAsync` is used set the **shuffle** mode of the **playback** with the Flag of **true** or **false** and populate the **property** for Results of the **success** of the operation accordingly.

## Step 5



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

## Step 6



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

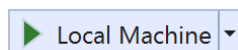
## Step 7

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

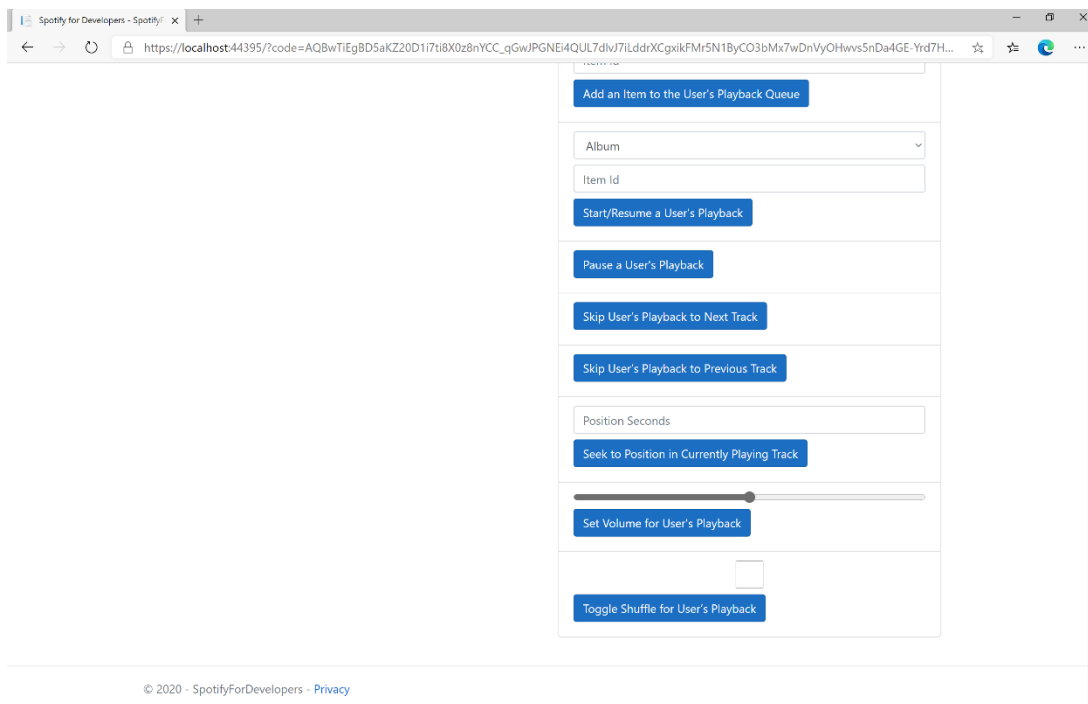


Finally, in **Visual Studio 2019** select **IIS Application** to run the **Web Application**



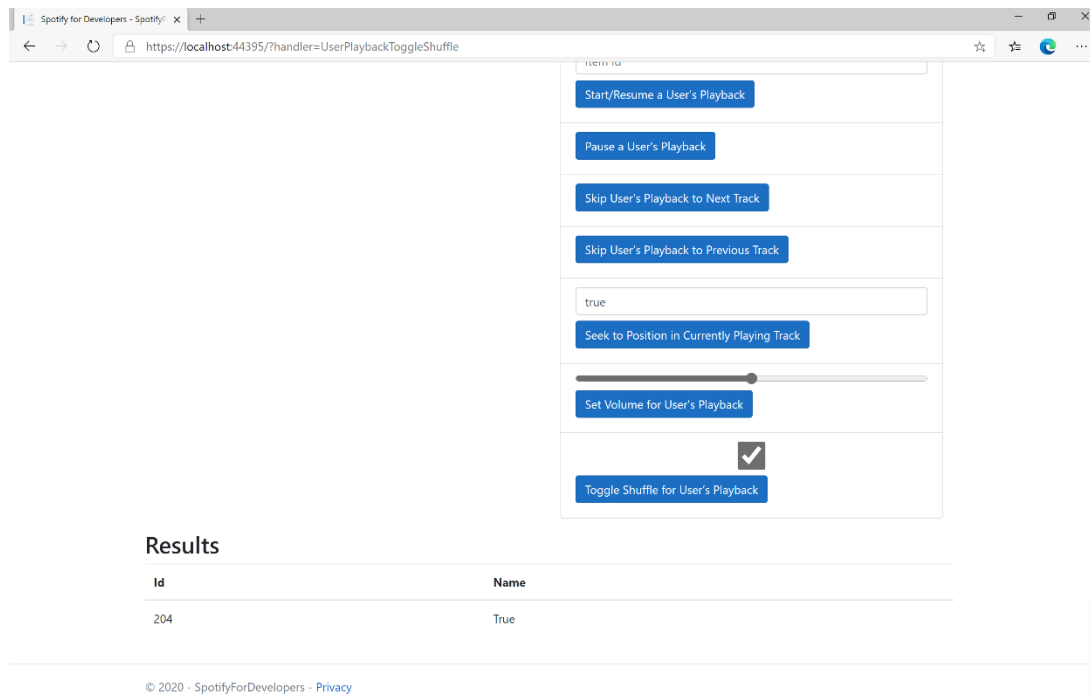
## Step 9

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:



## 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:



## 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

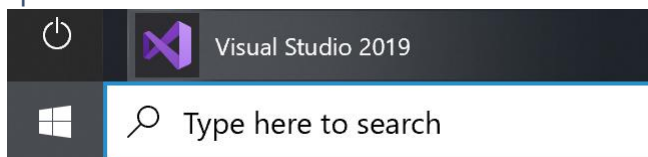
## Set Repeat Mode on User's Playback

Set the repeat mode for the user's playback. Options are repeat-track, repeat-context, and off.

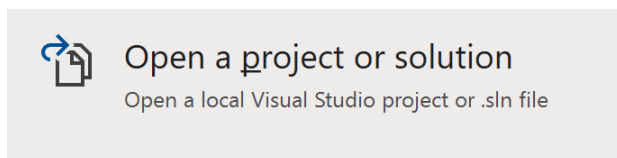
PUT <a href="https://api.spotify.com/v1/me/player/repeat">https://api.spotify.com/v1/me/player/repeat</a>	
Header	
<b>Authorization</b>	User Token from Spotify Accounts service with <b>user-modify-playback-state</b> scope
Query Parameter	
<b>state</b>	track will repeat current track, context will repeat current context and off will turn repeat off.
<b>device_id</b>	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

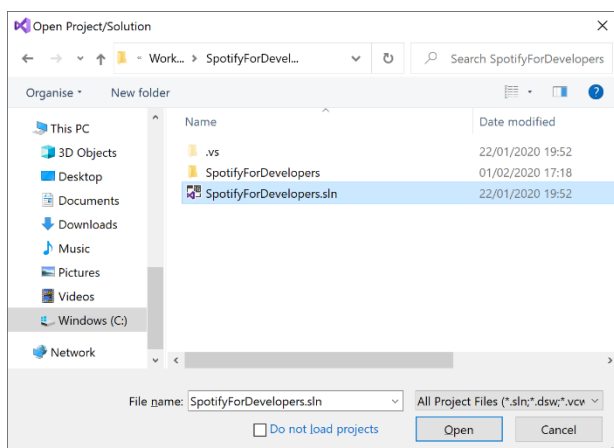
### Step 1



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**

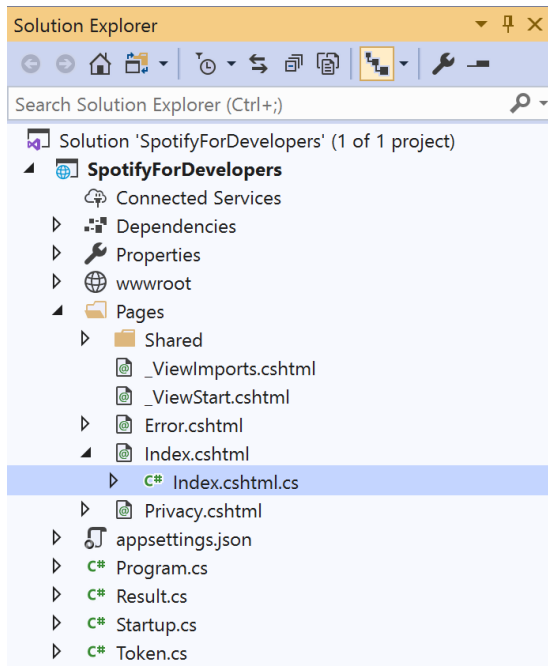


Once done, from the **Get started** screen for **Visual Studio 2019** select **Open a project or solution**



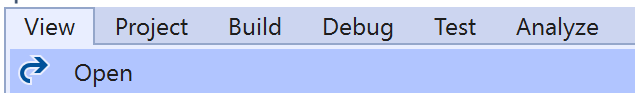
Then locate and select **SpotifyForDevelopers.sln** and select **Open** if you don't have this file already then please follow the previous parts of the workshop including **Getting Started, Authorisation Guide, Search & Browse, Playlists & Artists, Albums & Tracks, Episodes & Shows, Follow, Playlists, 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> 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();
}
```

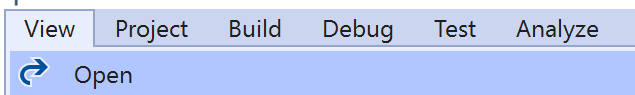
The **method** for `OnPostUserPlaybackSetRepeatModeAsync` is used set the **repeat** mode of the **playback** with the Flag of **true** or **false** and populate the **property** for Results of the **success** of the operation accordingly.

## Step 5



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

## Step 6



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

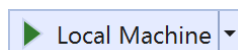
## Step 7

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**.

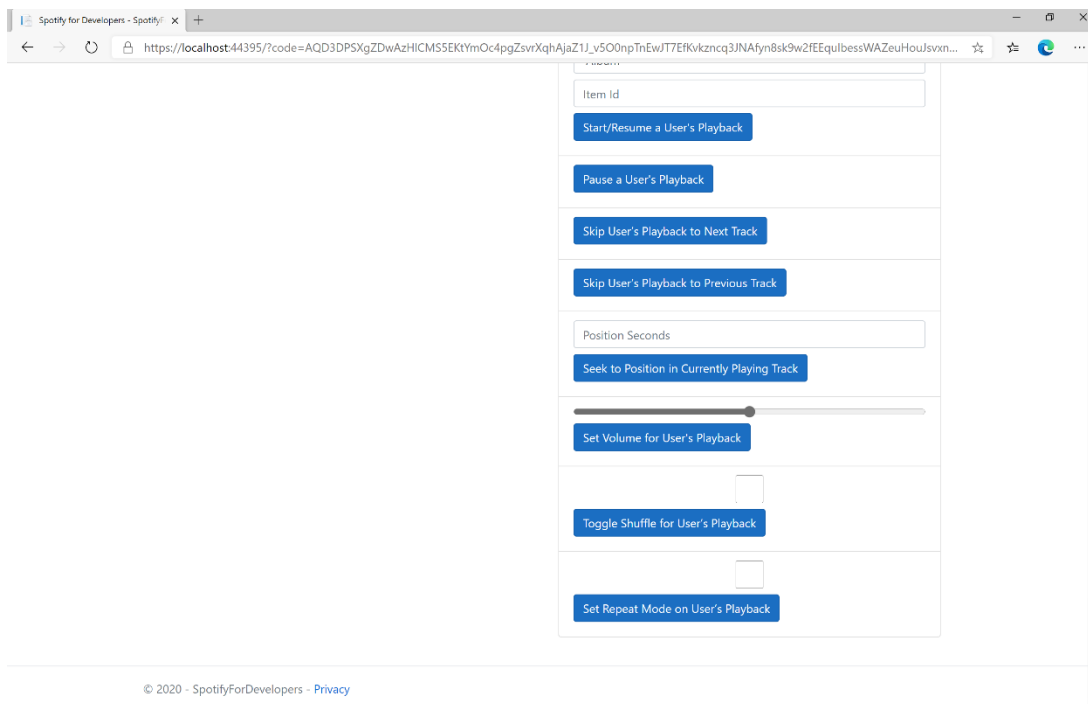
## Step 8



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

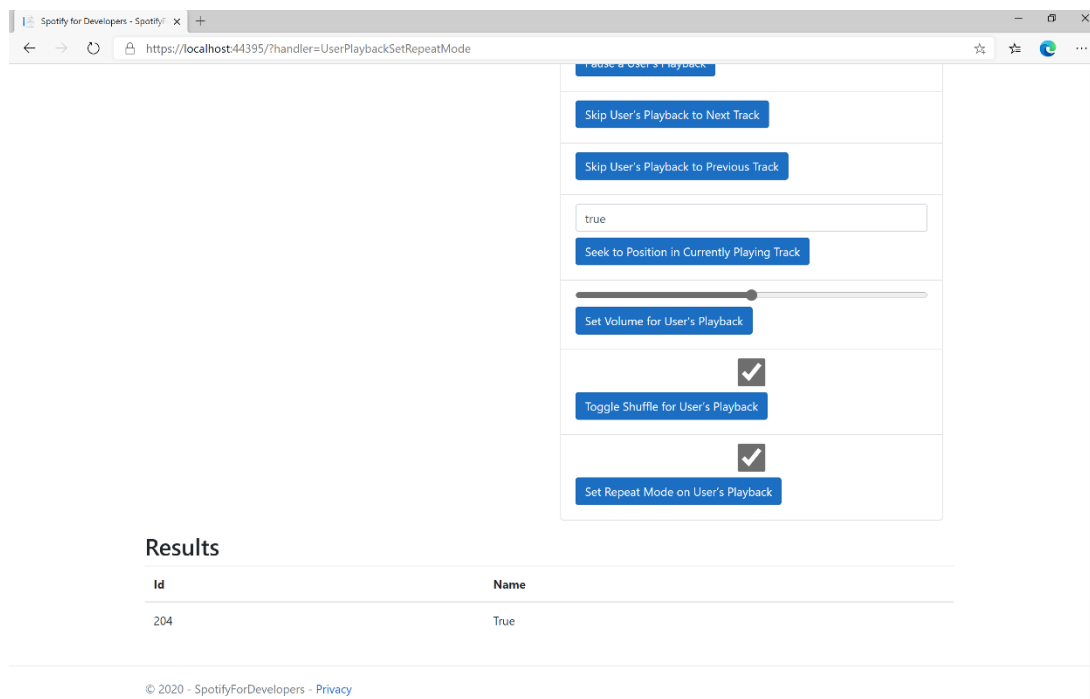
## Step 9

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:



## 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:



## 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

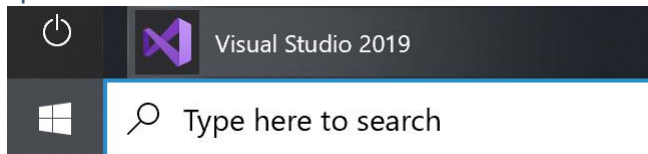
## Get Information about the User's Current Playback

Get information about the user's current playback state, including track, track progress, and active device.

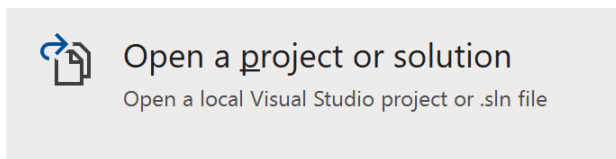
GET <a href="https://api.spotify.com/v1/me/player">https://api.spotify.com/v1/me/player</a>	
Header	
<b>Authorization</b>	User Token from Spotify Accounts service with <b>user-read-playback-state</b> scope
Query Parameter	
market	ISO 3166-1 alpha-2 country code. Provide to apply Track Relinking
additional_types	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

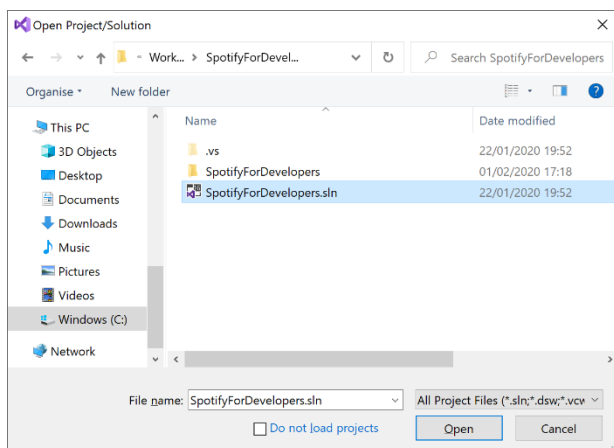
### Step 1



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**

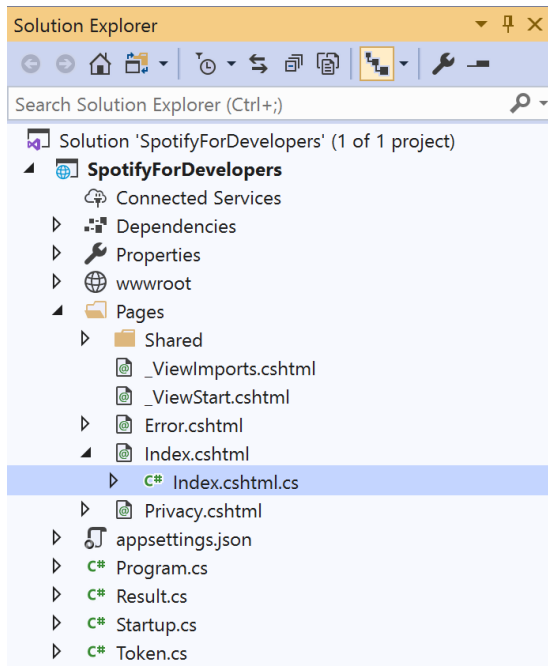


Once done, from the **Get started** screen for **Visual Studio 2019** select **Open a project or solution**



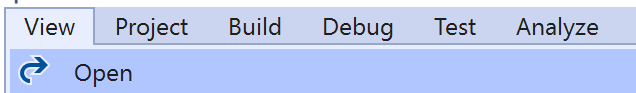
Then locate and select **SpotifyForDevelopers.sln** and select **Open** if you don't have this file already then please follow the previous parts of the workshop including **Getting Started, Authorisation Guide, Search & Browse, Playlists & Artists, Albums & Tracks, Episodes & Shows, Follow, Playlists,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

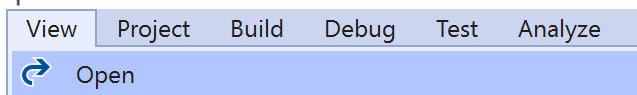


## Step 5



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

## Step 6



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

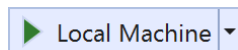
## Step 7

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**.

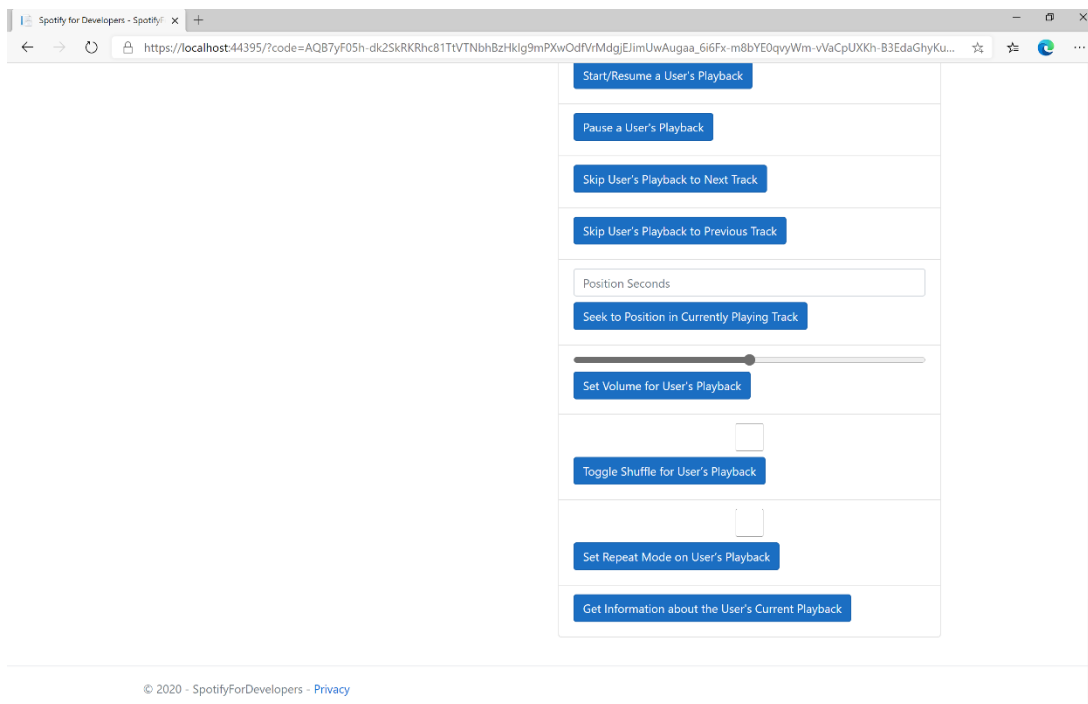
## Step 8



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

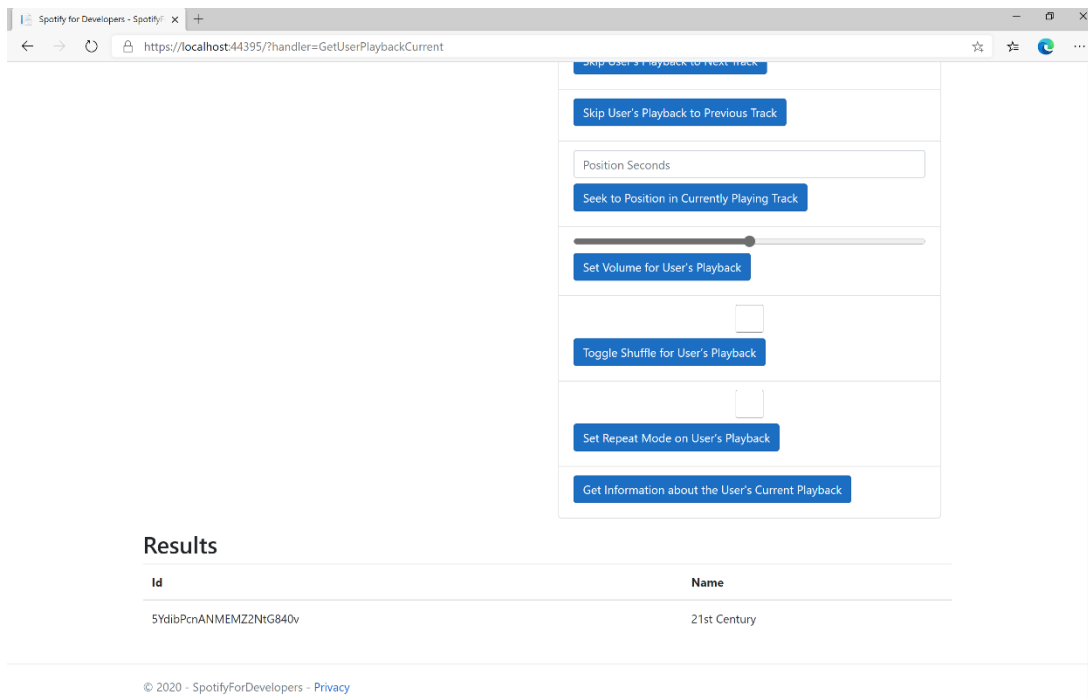
## Step 9

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:



## 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:



## 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

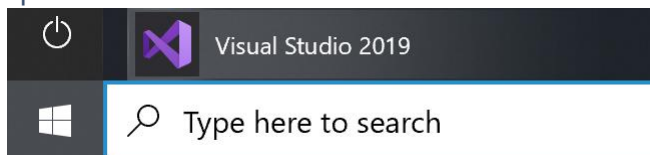
## Get the User's Currently Playing Track

Get the object currently being played on the user's Spotify account.

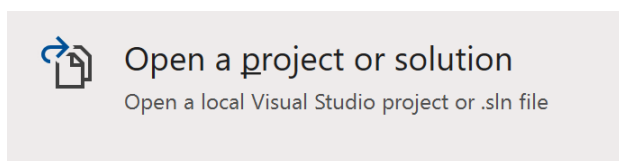
GET <a href="https://api.spotify.com/v1/me/player/currently-playing">https://api.spotify.com/v1/me/player/currently-playing</a>	
Header	
<b>Authorization</b>	User Token from Spotify Accounts service with <b>user-read-currently-playing</b> and/or <b>user-read-playback-state</b> scope
Query Parameter	
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

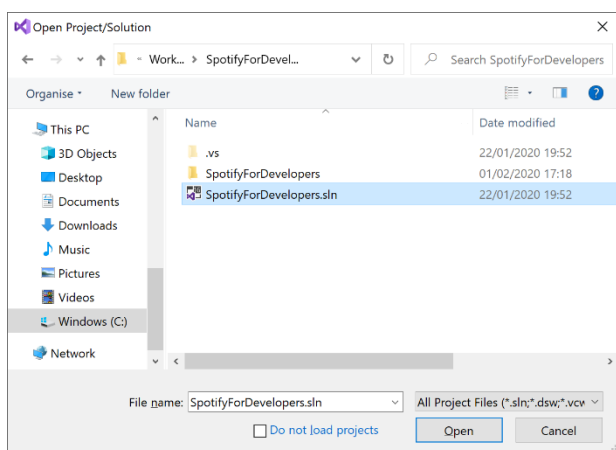
### Step 1



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**

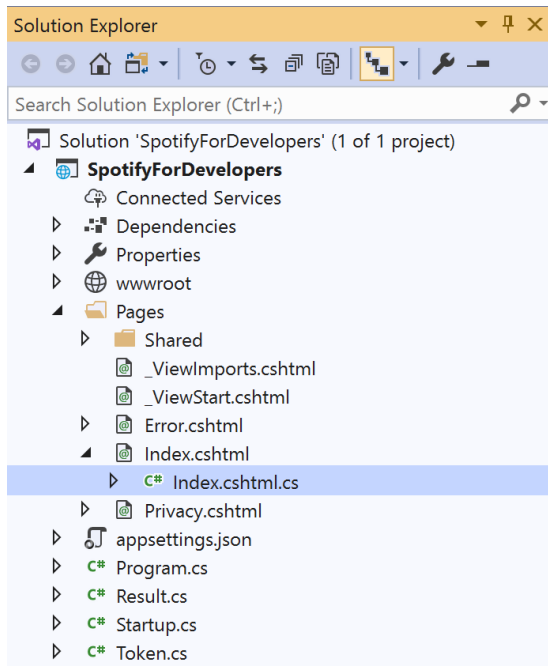


Once done, from the **Get started** screen for **Visual Studio 2019** select **Open a project or solution**



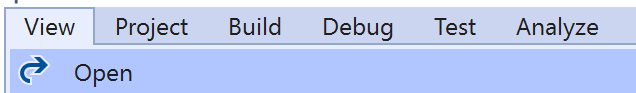
Then locate and select **SpotifyForDevelopers.sln** and select **Open** if you don't have this file already then please follow the previous parts of the workshop including **Getting Started, Authorisation Guide, Search & Browse, Playlists & Artists, Albums & Tracks, Episodes & Shows, Follow, Playlists,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

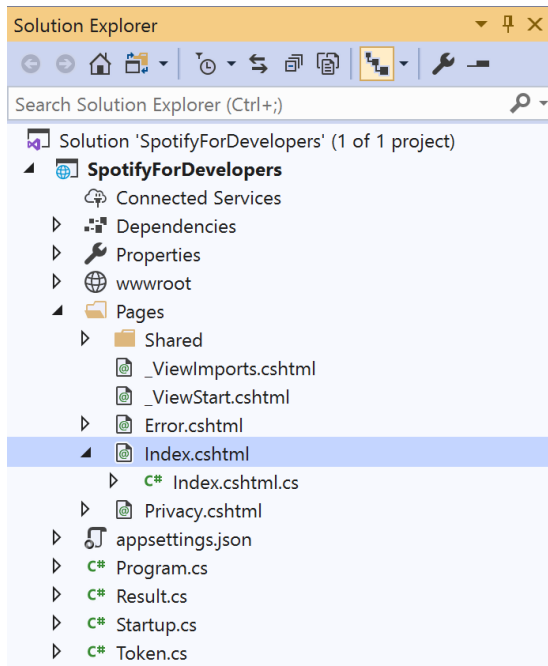
## Step 4

In the **Code View** for **Index.cshtml.cs** below the **method** for `public async Task<IActionResult> 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();
}
```

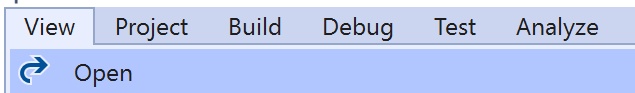
The **method** for `OnPostGetUserPlaybackCurrentTrackAsync` is used get **information** the currently playing **track** or **episode** and populate the **property** for `Results` accordingly.

## Step 5



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

## Step 6



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

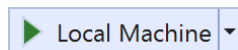
## Step 7

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**.

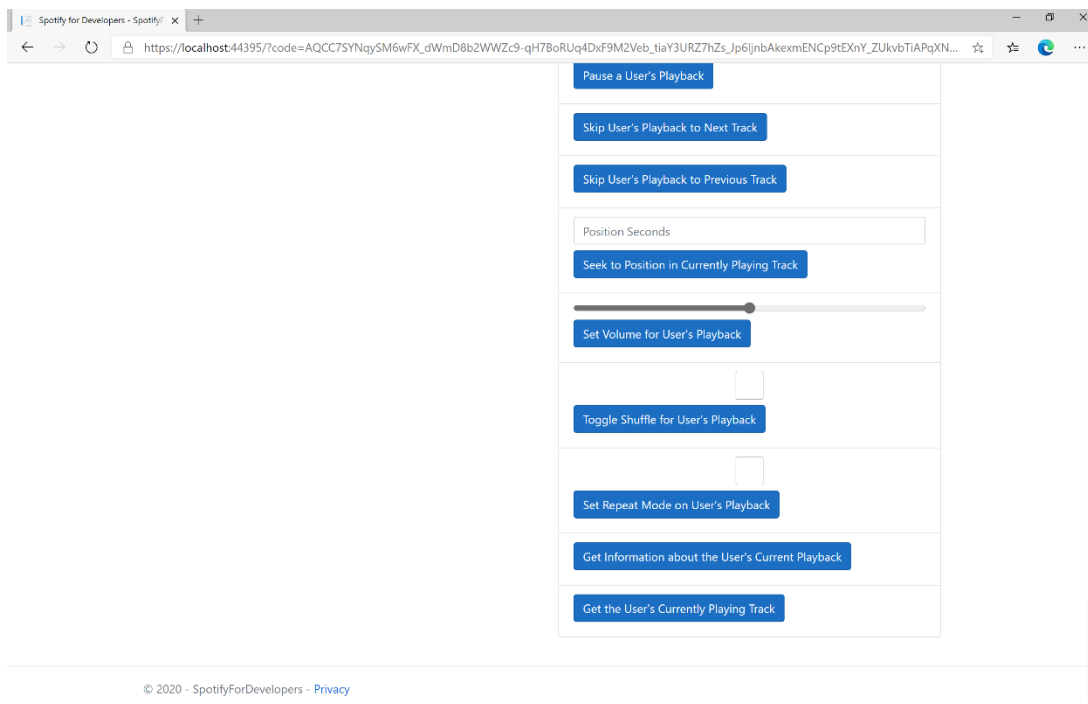
## Step 8



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

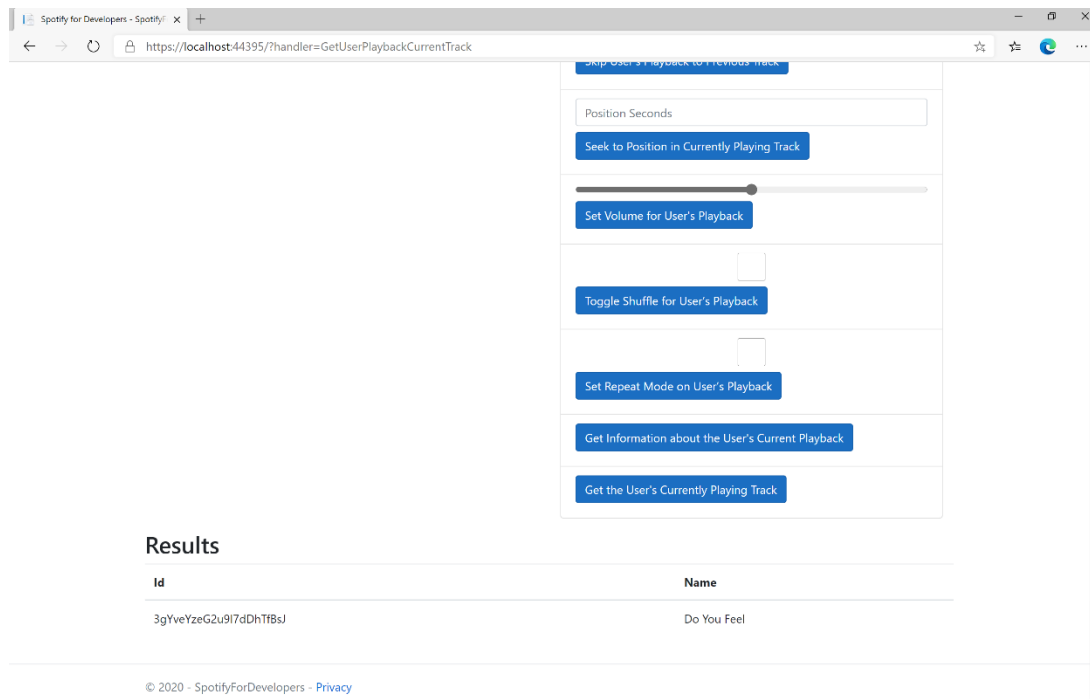
## Step 9

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:



## 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:



## 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

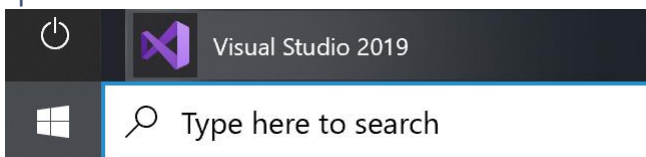
## Get Current User's Recently Played Tracks

Get tracks from the current user's recently played tracks.

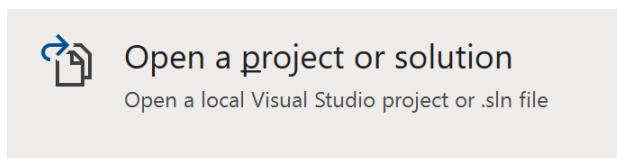
GET <a href="https://api.spotify.com/v1/me/player/recently-played">https://api.spotify.com/v1/me/player/recently-played</a>	
Header	
<b>Authorization</b>	User Token from Spotify Accounts service with <b>user-read-recently-played</b> scope
Query Parameter	
limit	Maximum number of items to return. Default: 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

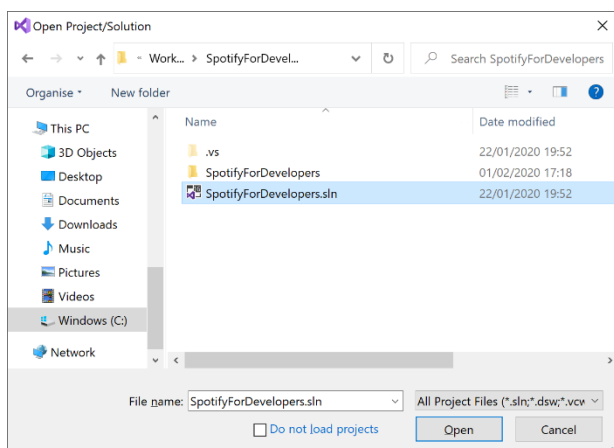
### Step 1



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**



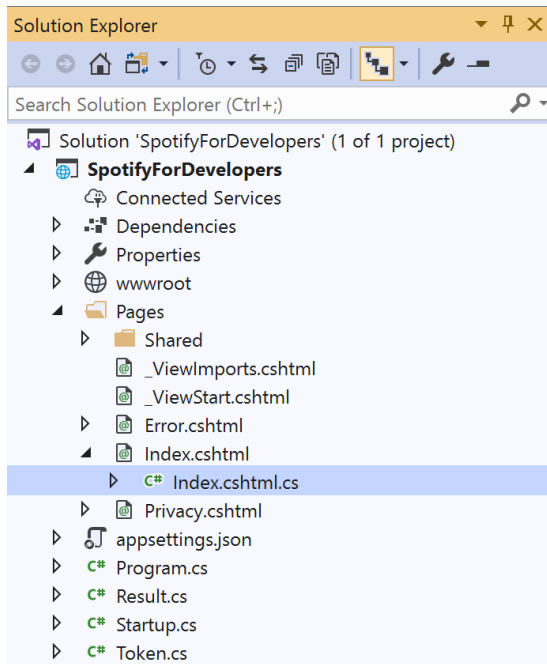
Once done, from the **Get started** screen for **Visual Studio 2019** select **Open a project or solution**



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**

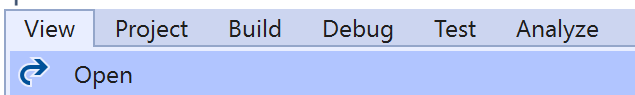


## Step 2



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**

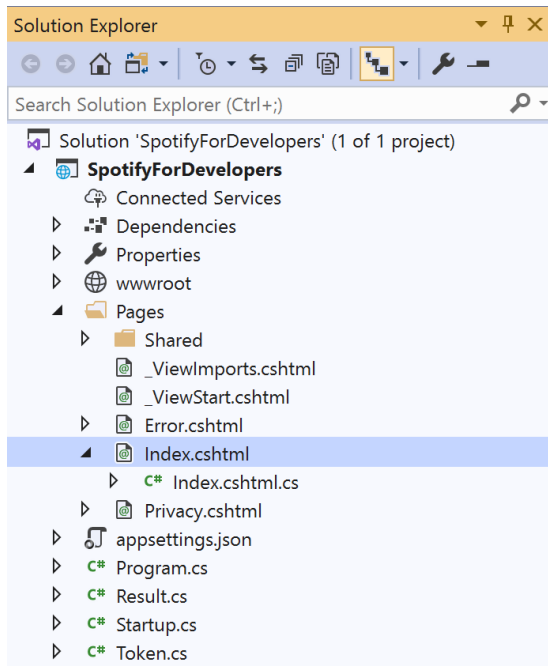
## Step 4

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();
}
```

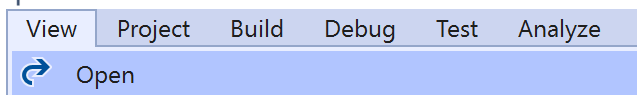
The **method** for `OnPostGetUserRecentlyPlayedTracksAsync` is used to get the **tracks** that have been played **recently** and populate the **property** for `Results` accordingly.

## Step 5



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

## Step 6



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

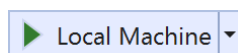
## Step 7

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**.

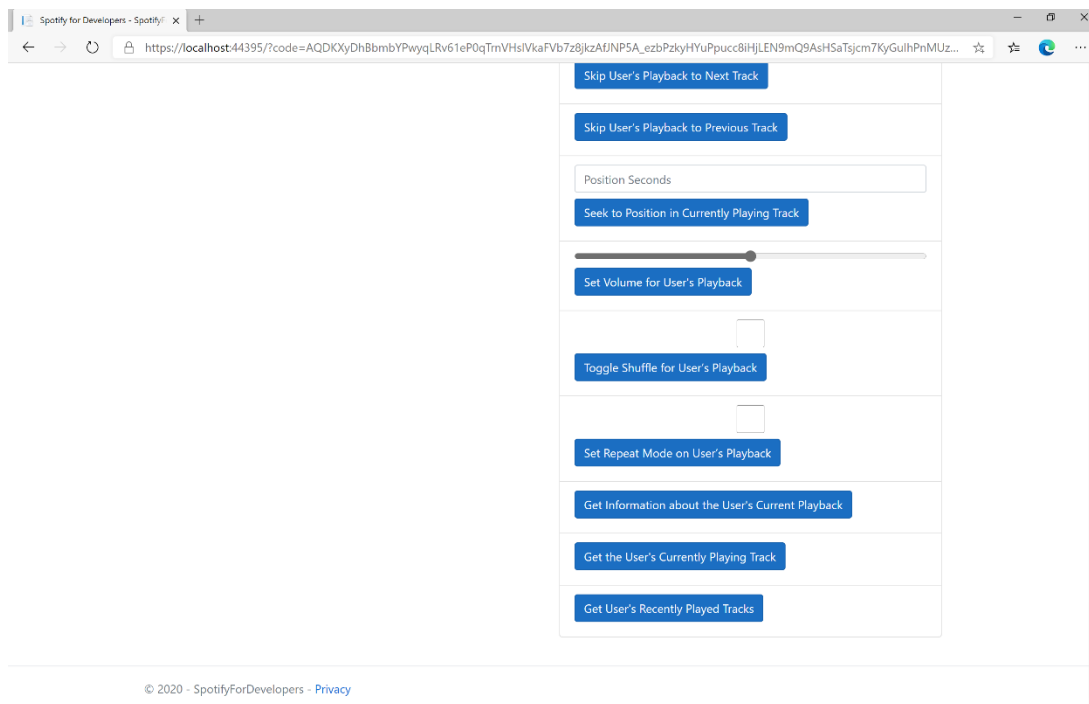
## Step 8



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

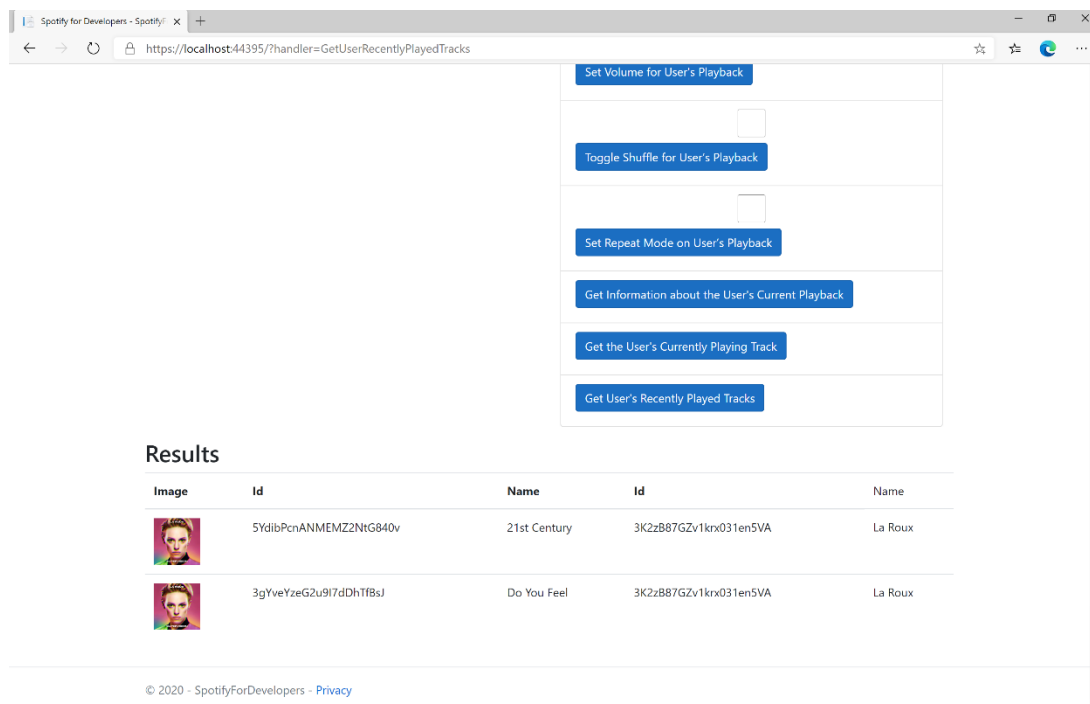
## Step 9

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:



## Step 10

You can select **Get User's Recently Played Tracks** and scroll down to view **Results** like the following:



## Step 11



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

## Step 12



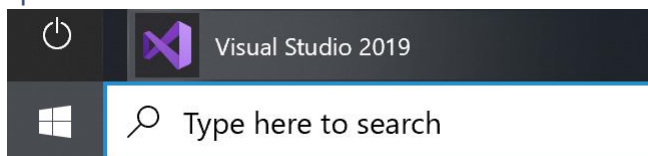
You can choose to exit **Visual Studio 2019** by selecting the **Close** button in the top right of the **application** as that completes this part of the workshop

## Get a User's Available Devices

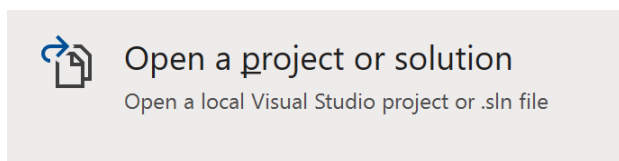
Get information about a user's available devices.

GET <a href="https://api.spotify.com/v1/me/player/devices">https://api.spotify.com/v1/me/player/devices</a>	
Header	
<b>Authorization</b>	User Token from Spotify Accounts service with <b>user-read-playback-state</b> scope
Response	
Success	
HTTP Status 200 OK	Array of Device Objects or Empty if none
Error	
Error Code	Error Object

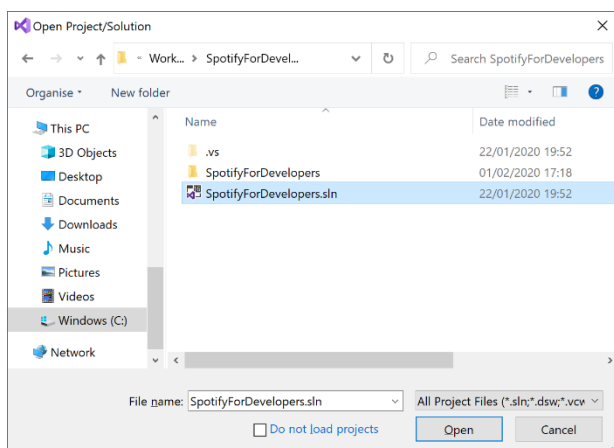
### Step 1



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**

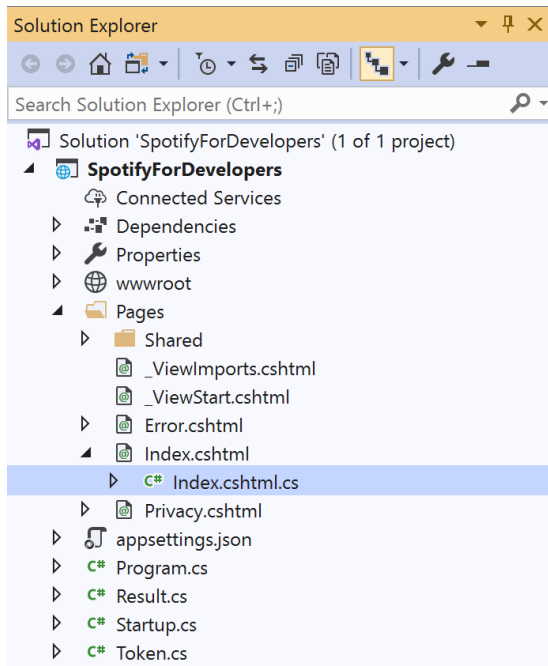


Once done, from the **Get started** screen for **Visual Studio 2019** select **Open a project or solution**



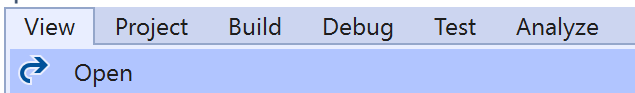
Then locate and select **SpotifyForDevelopers.sln** and select **Open** if you don't have this file already then please follow the previous parts of the workshop including **Getting Started, Authorisation Guide, Search & Browse, Playlists & Artists, Albums & Tracks, Episodes & Shows, Follow, Playlists,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



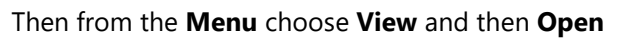
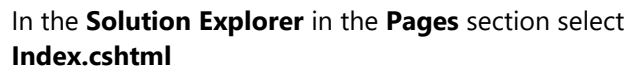
Then from the **Menu** choose **View** and then **Open**

## Step 4

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

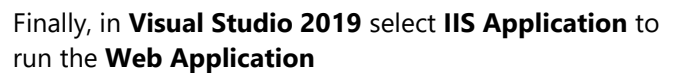
```
public async Task<IActionResult> OnPostGetUserPlaybackDevicesAsync()
{
    LoadToken();
    var results = await Api.GetUserPlaybackDevicesAsync();
    if (results?.Items != null)
    {
        Results = results.Items.Select(result => new Result()
        {
            Id = result.Id,
            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.



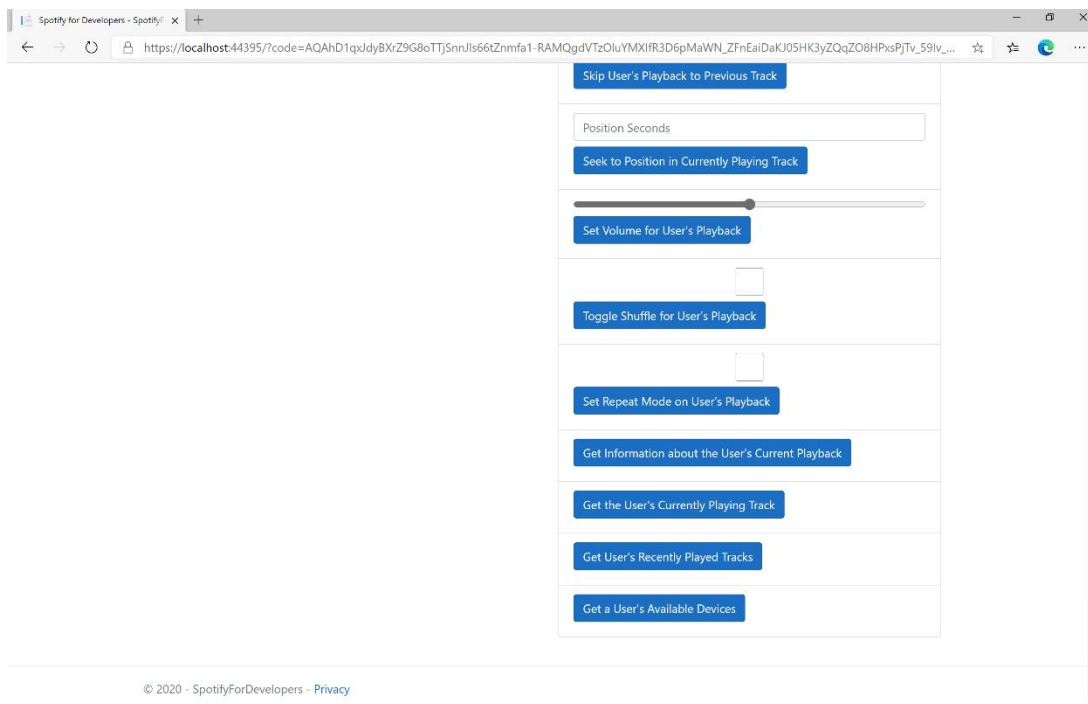
enter the following:

will output to the **Results**.



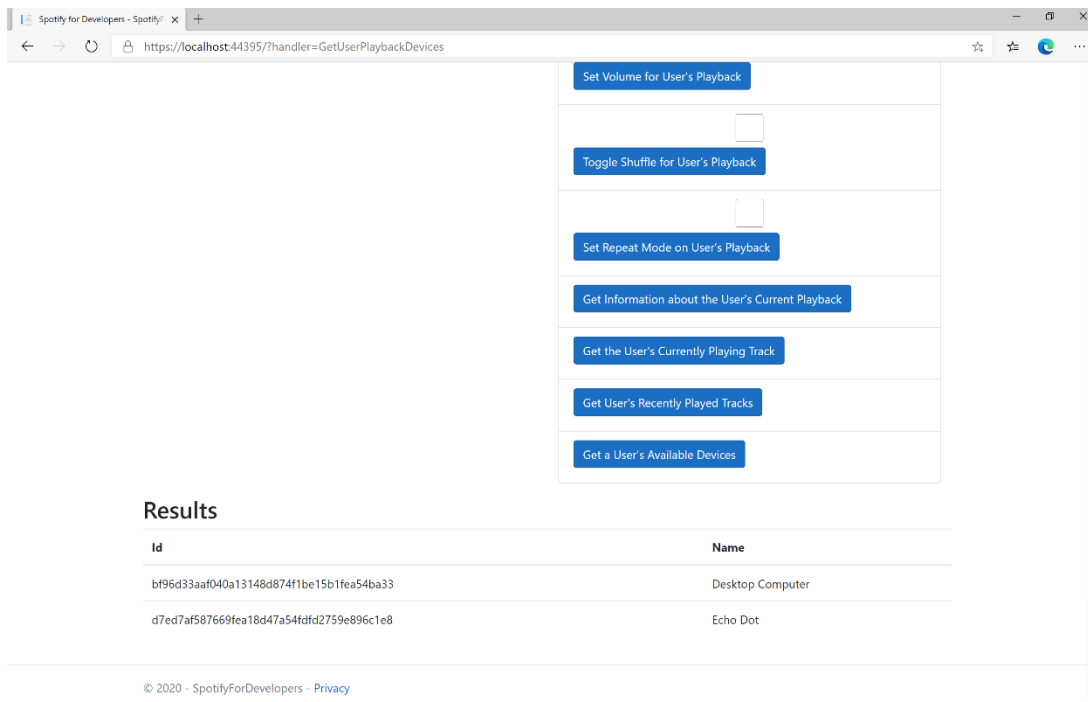
## Step 9

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 select **Get a User's Available Devices** and scroll down to view **Results** like the following:



## 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

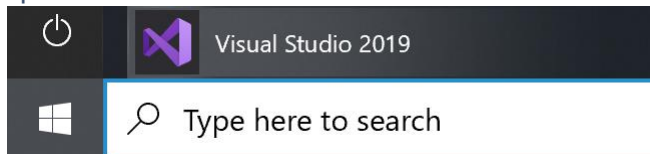
## Transfer a User's Playback

Transfer playback to a new device and determine if it should start playing.

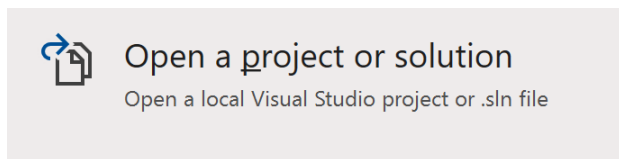
PUT <a href="https://api.spotify.com/v1/me/player">https://api.spotify.com/v1/me/player</a>	
Header	
<b>Authorization</b>	User Token from Spotify Accounts service with <b>user-modify-playback-state</b> scope
Body Parameter	
<b>device_ids</b>	Device Id on which playback should be started/transferred, only one Device Id is supported
<b>play</b>	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

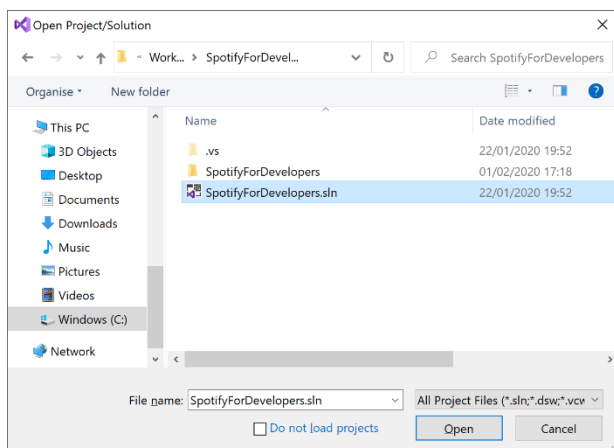
### Step 1



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**



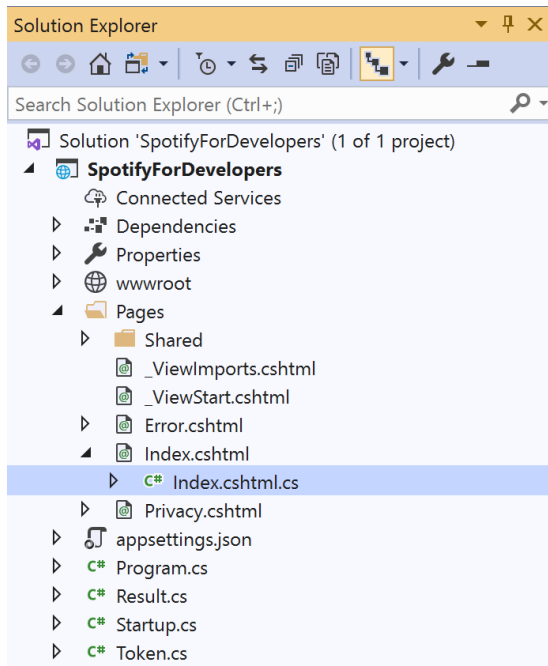
Once done, from the **Get started** screen for **Visual Studio 2019** select **Open a project or solution**



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**

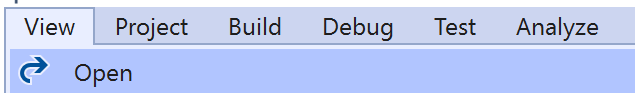


## Step 2



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**

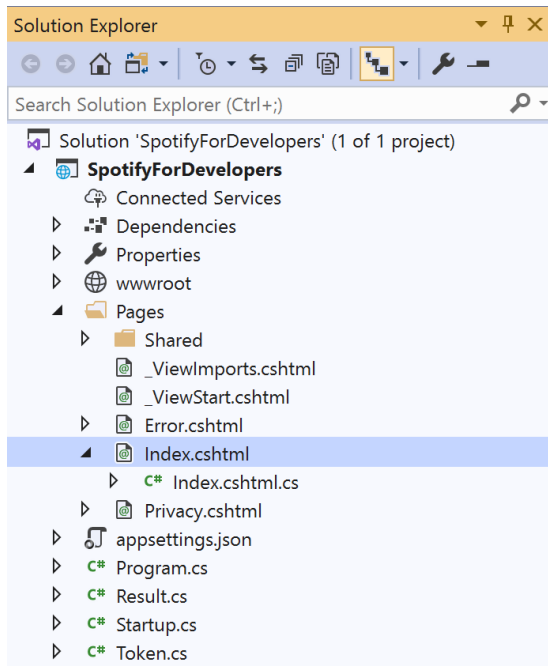
## Step 4

In the **Code View** for **Index.cshtml.cs** below the **method** for `public async Task<IActionResult> 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();
}
```

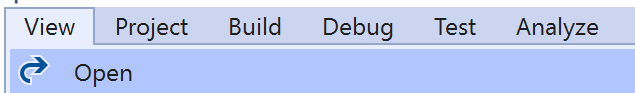
The **method** for `OnPostUserPlaybackTransferAsync` is used to set the **device** to use for **playback** with the Value of the **Device Id** and populate the **property** for Results of the **success** of the operation accordingly.

## Step 5



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

## Step 6



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

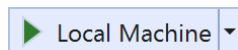
## Step 7

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**.

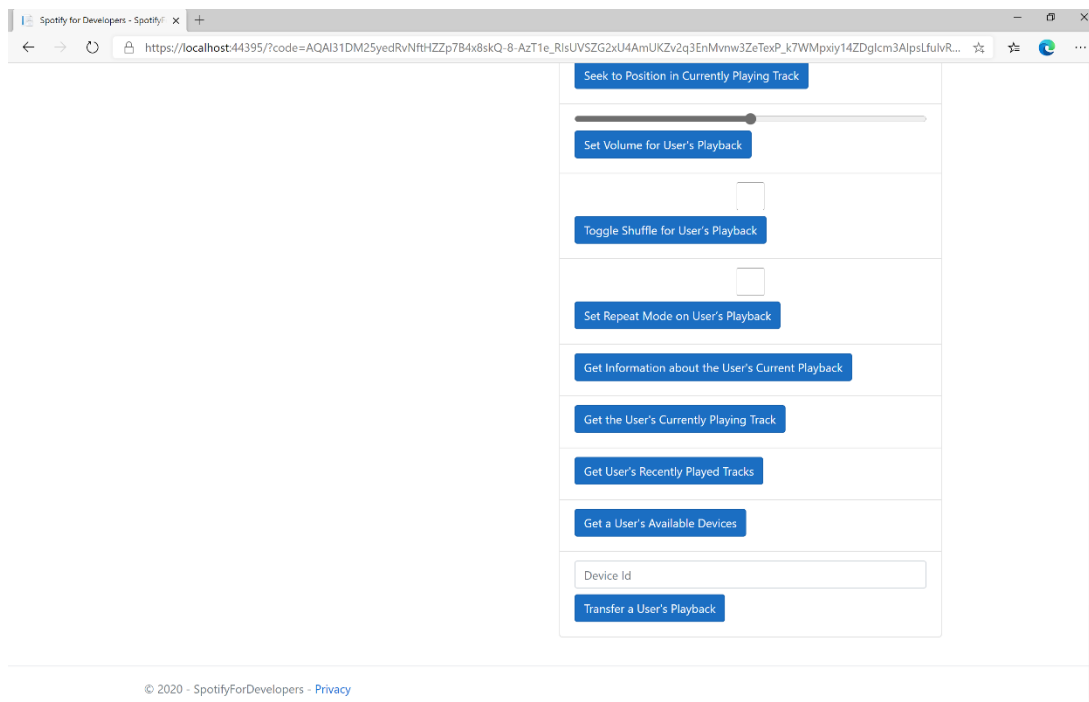
## Step 8



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

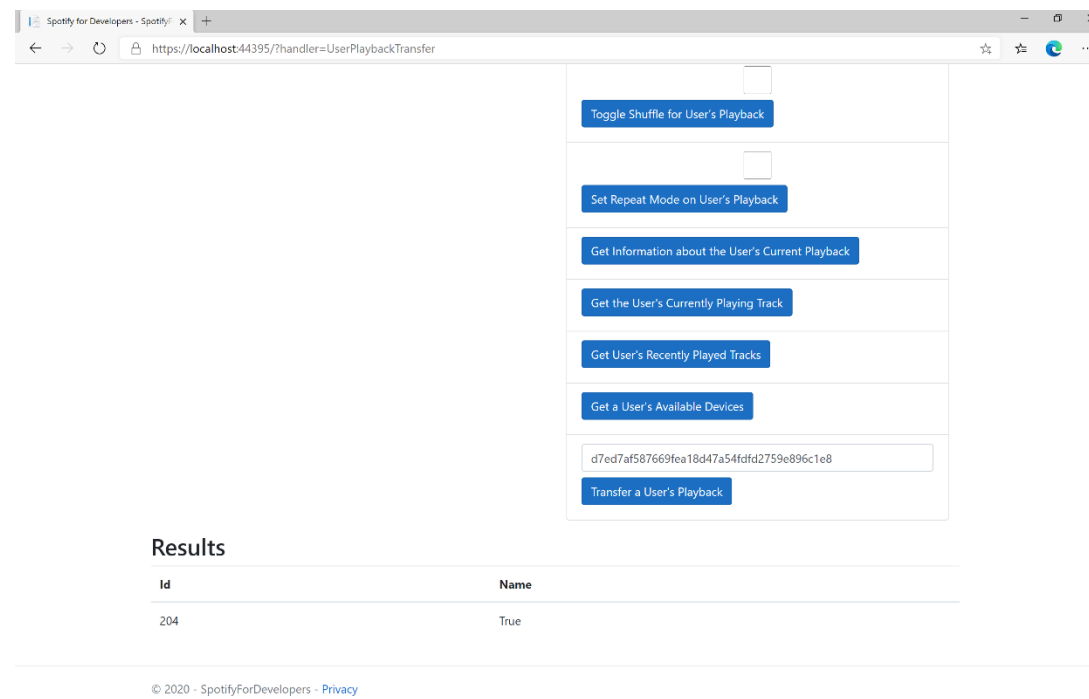
## Step 9

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:



## 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:



## 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

