



Tutorialr.com

Spotify for Developers Artists, Albums & Tracks

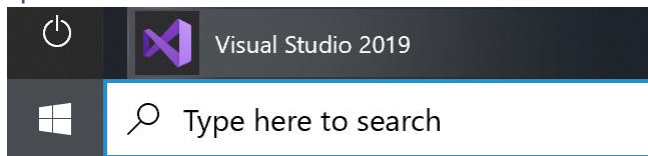
Get Multiple Artists

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

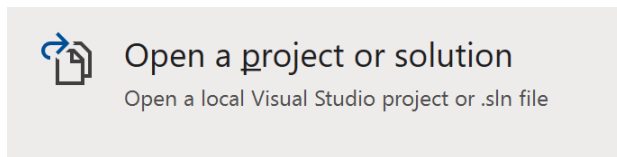
GET https://api.spotify.com/v1/artists	
Header	
Authorization	Valid Access Token from Spotify Accounts service
Query Parameter	
ids	Maximum of 50 comma-separated list of artist Spotify Ids

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

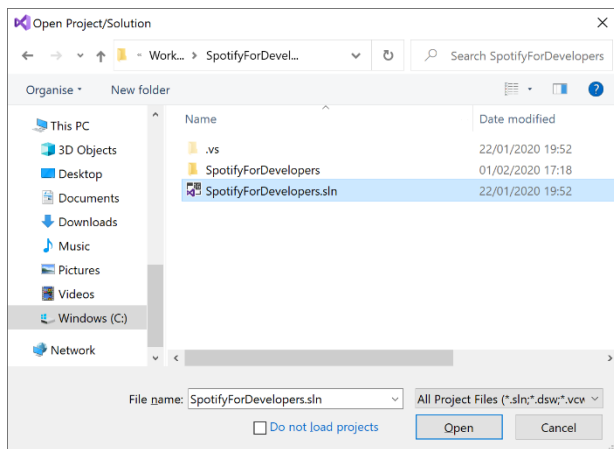
Step 1



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

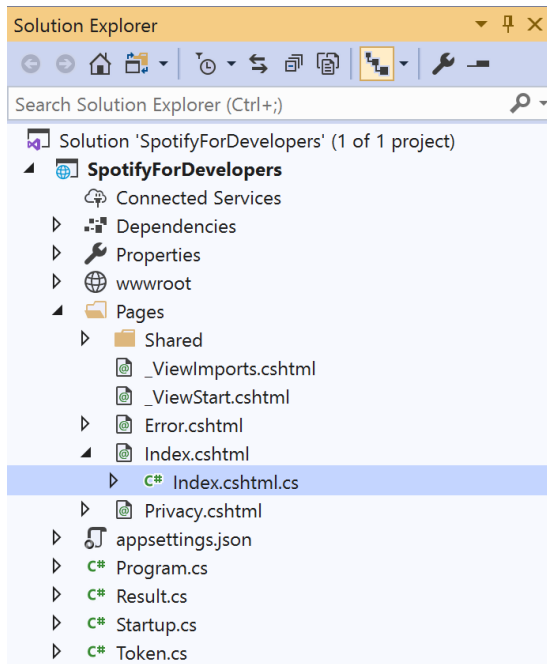


Once done, from the **Get started** screen for **Visual Studio 2019** select **Open a project or solution**



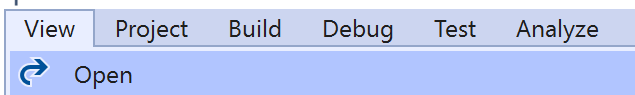
Then locate and select **SpotifyForDevelopers.sln** and select **Open** if you don't have this file already then please follow the previous parts of the workshop including **Getting Started**, **Authorisation Guide**, **Search & Browse**

Step 2



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**

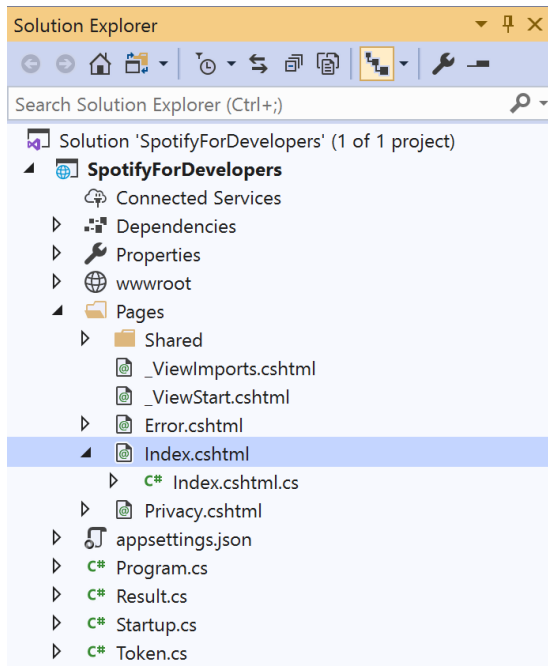
Step 4

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();
}
```

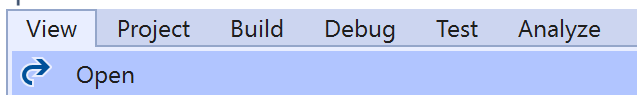
The **method** for `OnPostGetMultipleArtistsAsync` is used to get **artists** by multiple **Artist Id** on Spotify with the value and populate the **property** for `Results` accordingly.

Step 5



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

Step 6



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

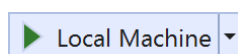
Step 7

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**.

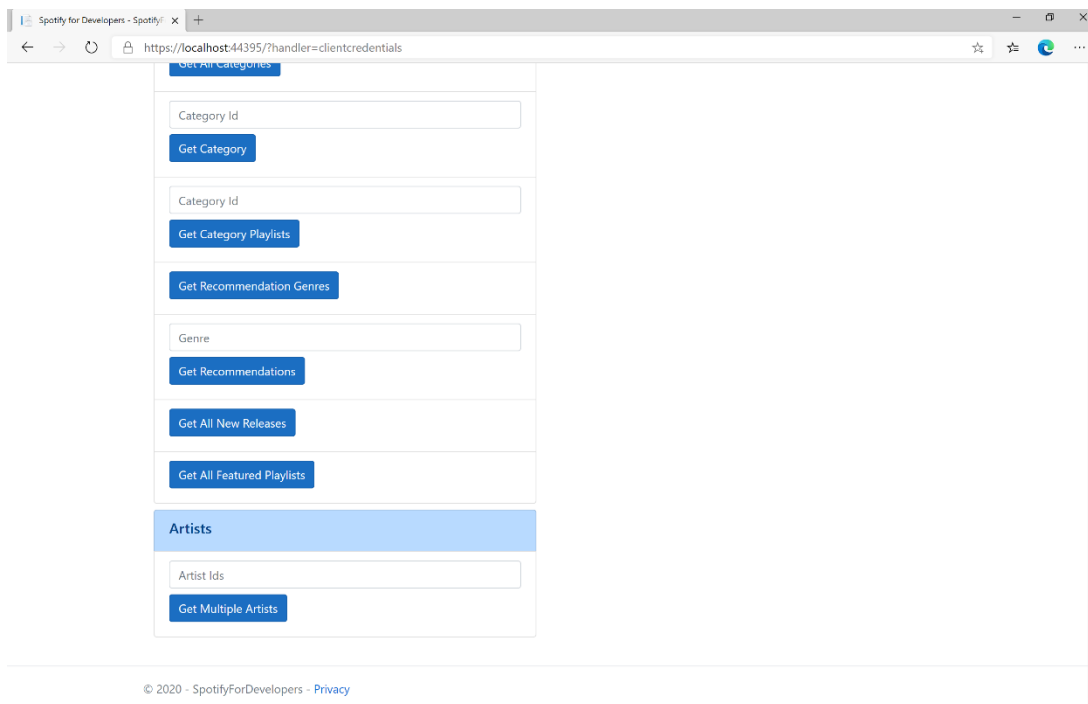
Step 8



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

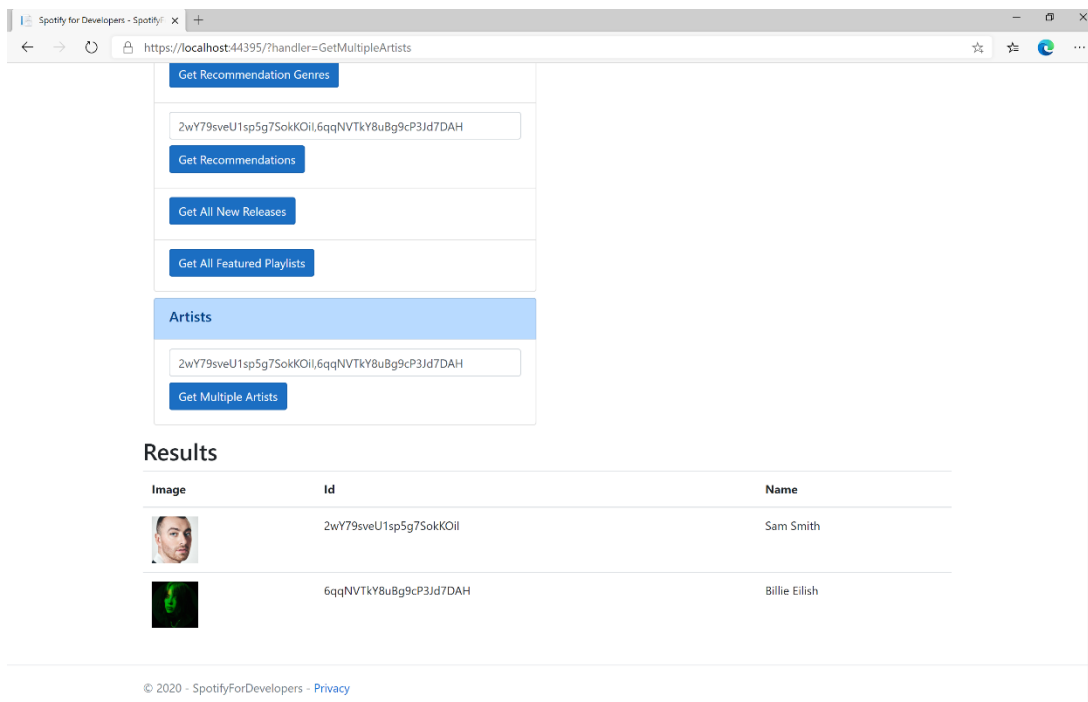
Step 9

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



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:



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

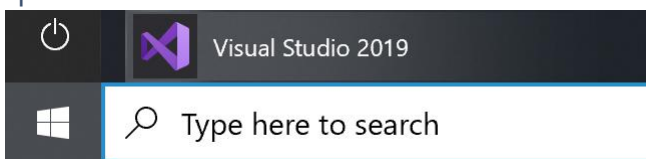
Get an Artist

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

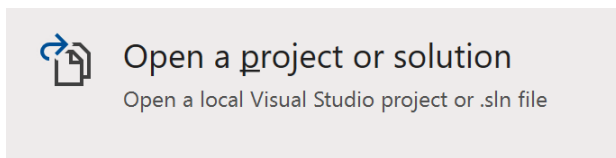
GET https://api.spotify.com/v1/artists/{id}	
Header	
Authorization	Valid Access Token from Spotify Accounts service
Path Parameter	
id	Spotify Id of the artist

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

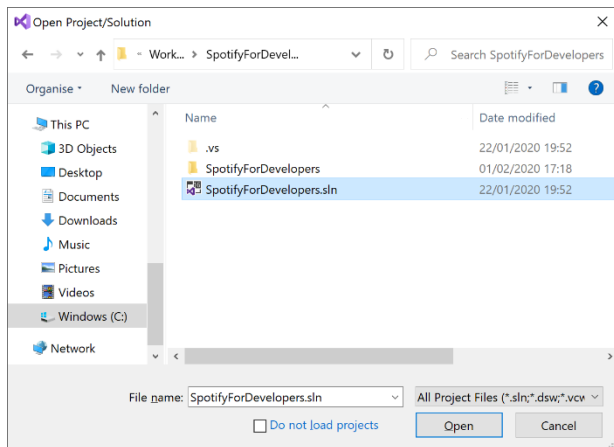
Step 1



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**

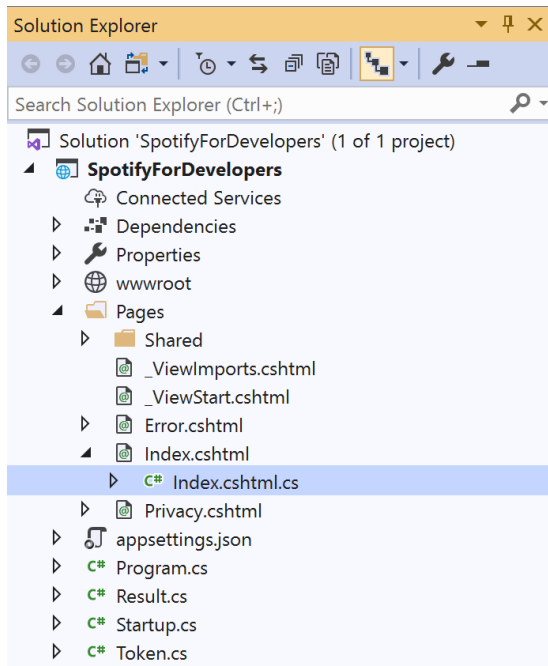


Once done, from the **Get started** screen for **Visual Studio 2019** select **Open a project or solution**



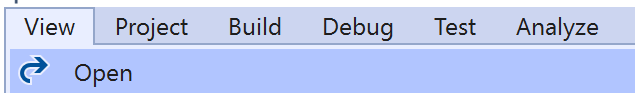
Then locate and select **SpotifyForDevelopers.sln** and select **Open** if you don't have this file already then please follow the previous parts of the workshop including **Getting Started**, **Authorisation Guide**, **Search & Browse** and **Artists, Albums & Tracks**

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**

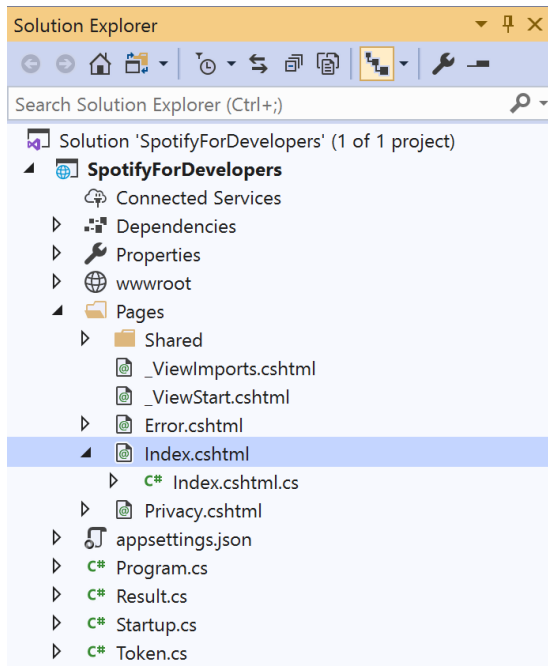
Step 4

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

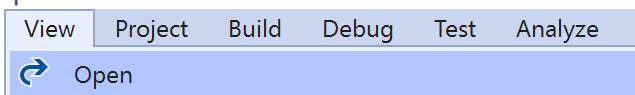
The **method** for `OnPostGetArtistAsync` is used to get an **artist** by **Artist Id** on Spotify with the value and populate the **property** for `Results` accordingly.

Step 5



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

Step 6



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

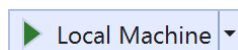
Step 7

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**.

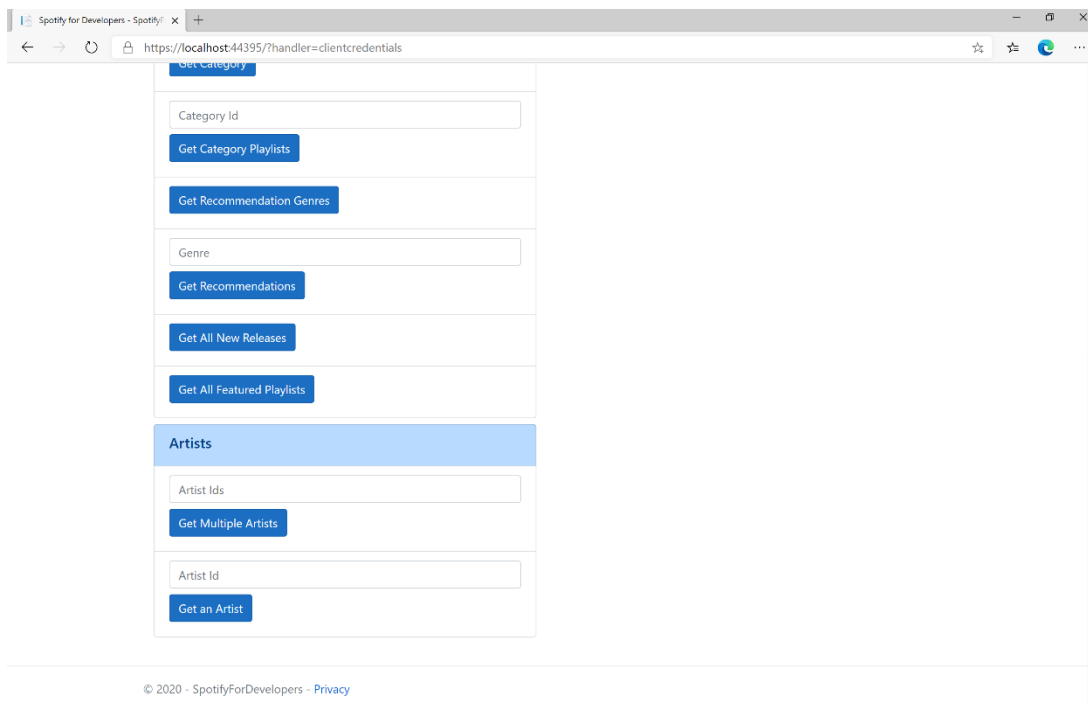
Step 8



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

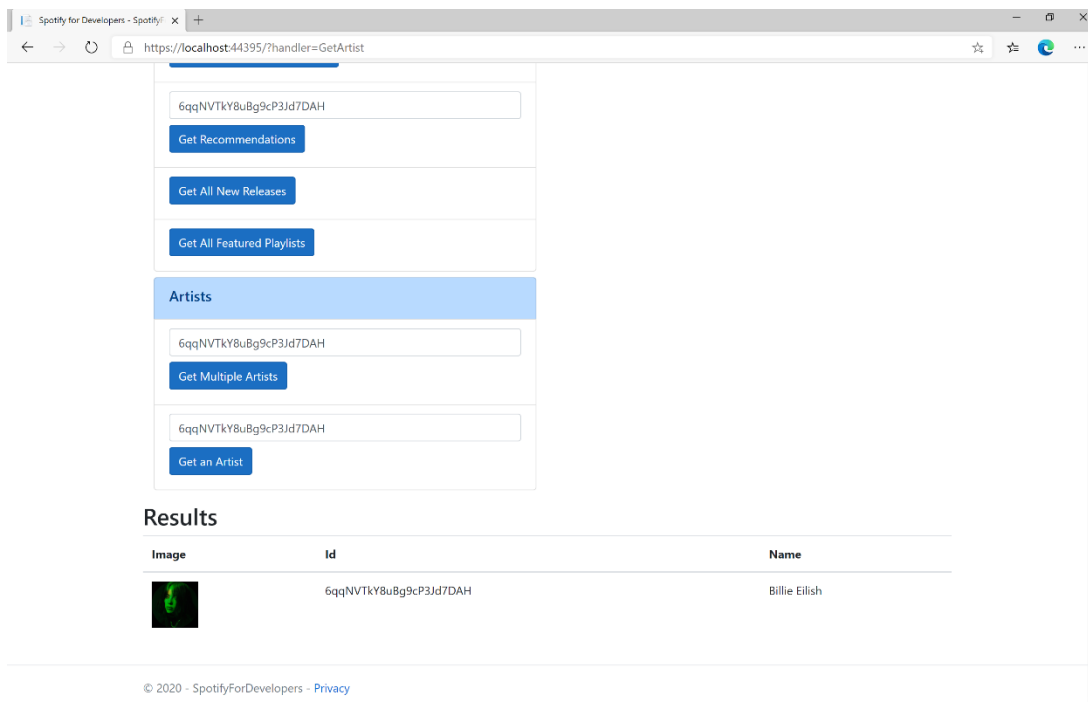
Step 9

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



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:



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

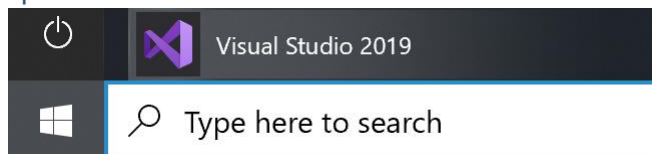
Get an Artist's Albums

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

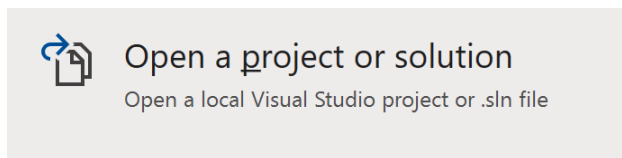
GET https://api.spotify.com/v1/artists/{id}/albums	
Header	
Authorization	Valid Access Token from Spotify Accounts service
Path Parameter	
id	Spotify ID of the artist
Query Parameters	
include_groups	Comma-separated list of keywords that will be used to filter the response from: album, single, appears_on 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

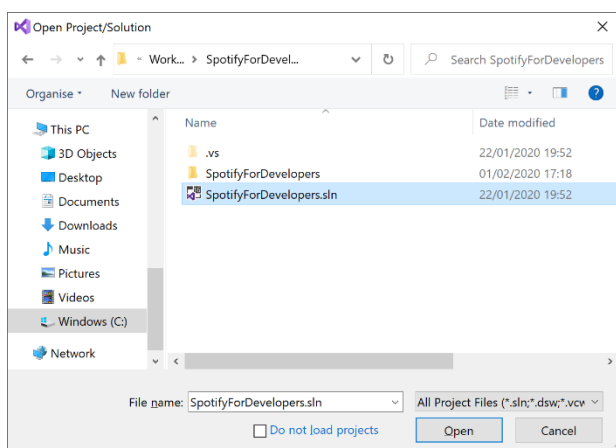
Step 1



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**

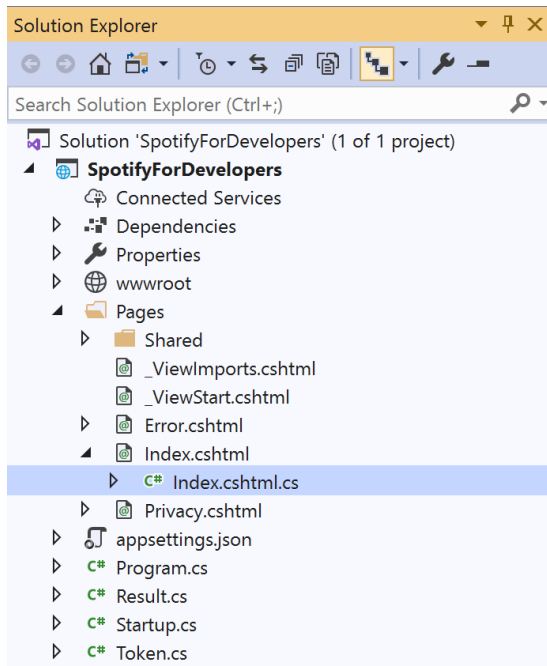


Once done, from the **Get started** screen for **Visual Studio 2019** select **Open a project or solution**



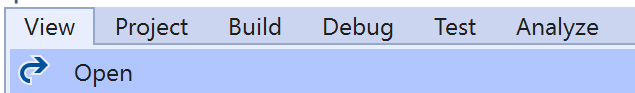
Then locate and select **SpotifyForDevelopers.sln** and select **Open** if you don't have this file already then please follow the previous parts of the workshop including **Getting Started**, **Authorisation Guide**, **Search & Browse** and **Artists, Albums & Tracks**

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**

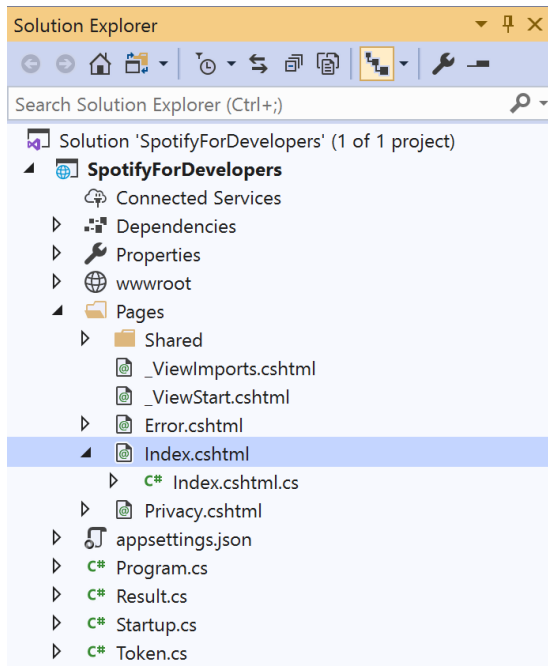
Step 4

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

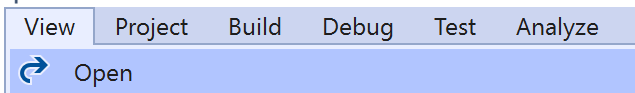
The **method** for `OnPostGetArtistAlbumsAsync` is used to get the **albums** of an **artist** by **Artist Id** on Spotify with the value and populate the **property** for `Results` accordingly.

Step 5



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

Step 6



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

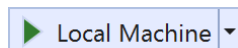
Step 7

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**.

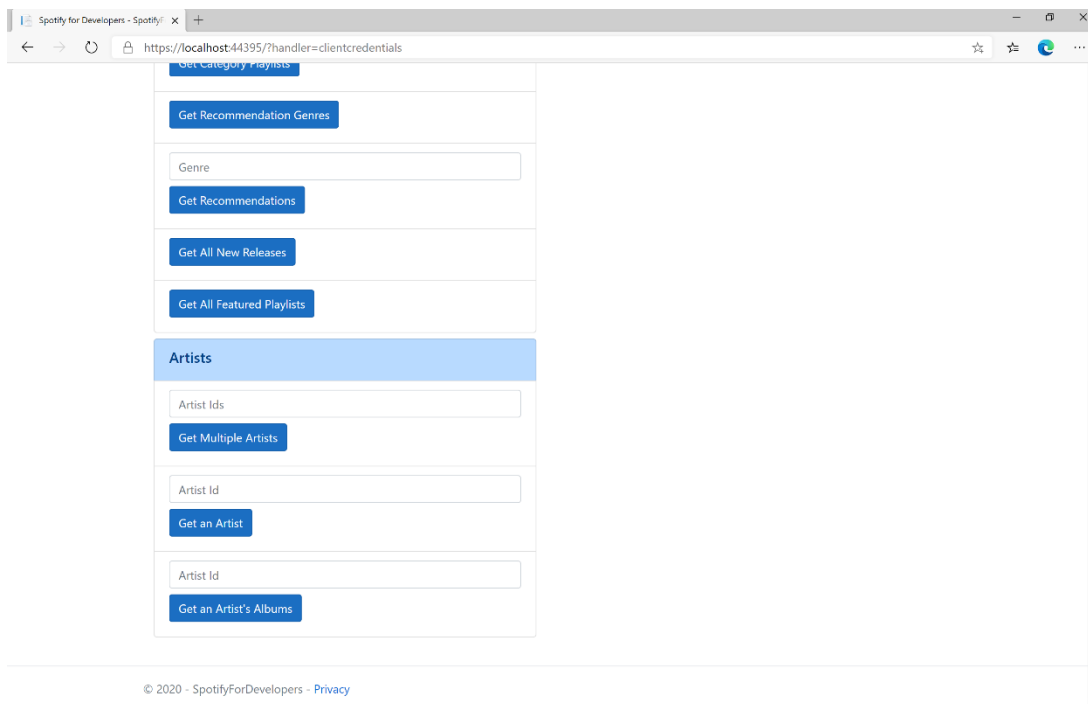
Step 8



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

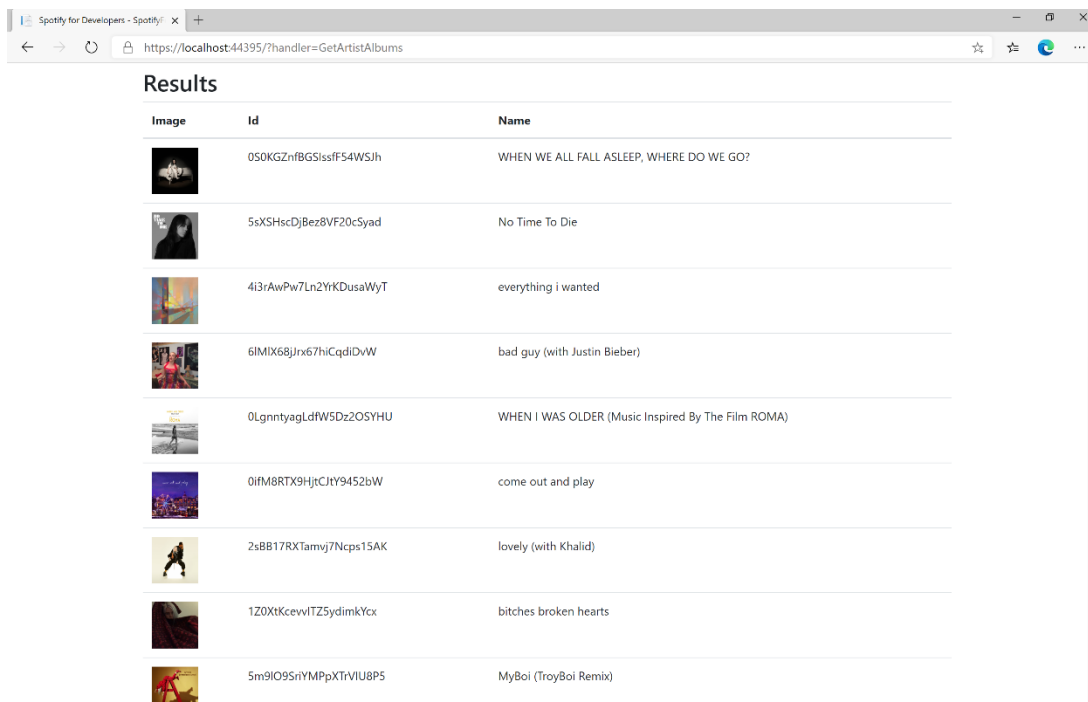
Step 9

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



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:



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

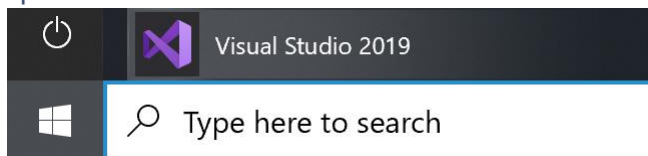
Get an Artist's Top Tracks

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

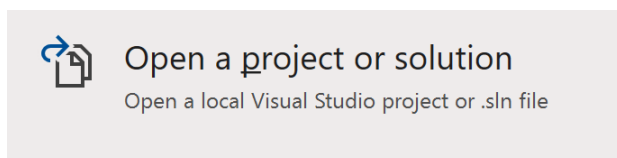
GET https://api.spotify.com/v1/artists/{id}/top-tracks	
Header	
Authorization	Valid Access Token from Spotify Accounts service
Path Parameter	
id	Spotify Id of the artist
Query Parameters	
market	ISO 3166-1 alpha-2 country code e.g. "GB" or "from_token"

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

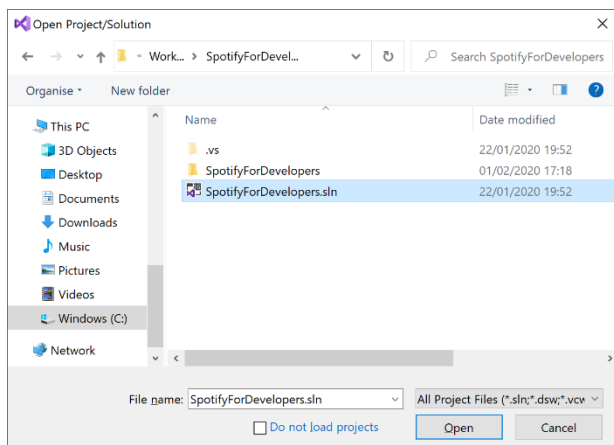
Step 1



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**

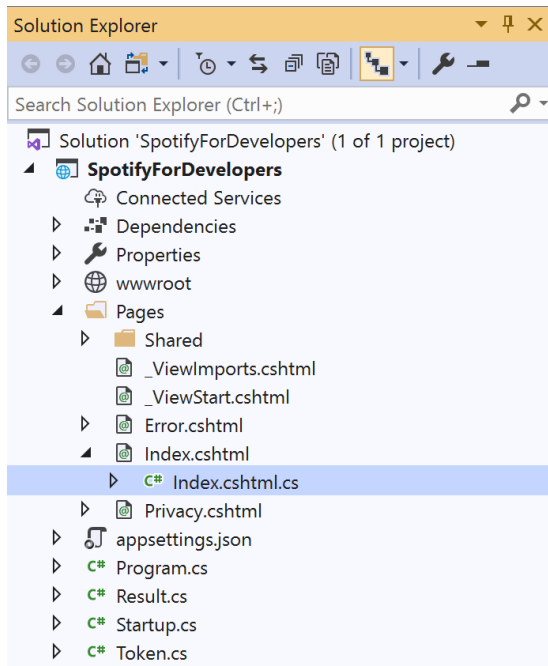


Once done, from the **Get started** screen for **Visual Studio 2019** select **Open a project or solution**



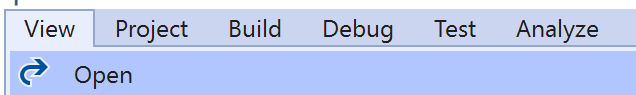
Then locate and select **SpotifyForDevelopers.sln** and select **Open** if you don't have this file already then please follow the previous parts of the workshop including **Getting Started**, **Authorisation Guide**, **Search & Browse** and **Artists, Albums & Tracks**

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**

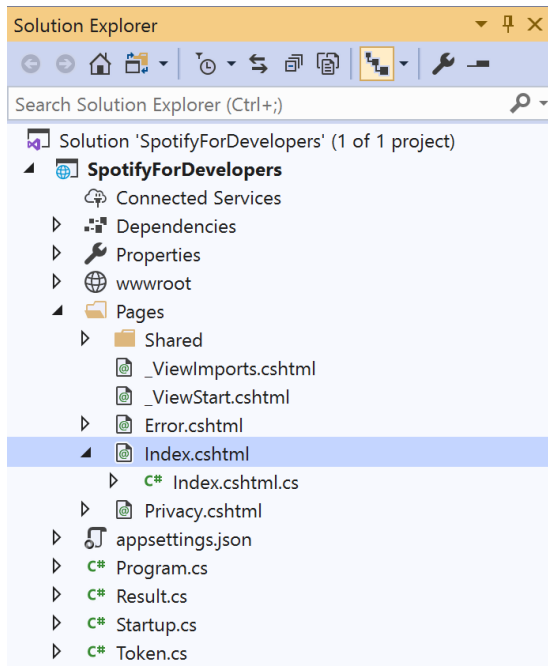
Step 4

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

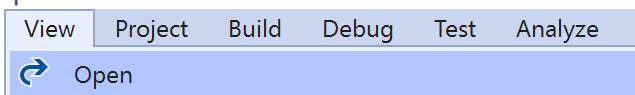
The **method** for `OnPostGetArtistTopTracksAsync` is used to get the top **tracks** of an **artist** by **Artist Id** on Spotify with the value and populate the **property** for `Results` accordingly.

Step 5



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

Step 6



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

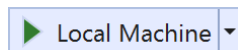
Step 7

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**.

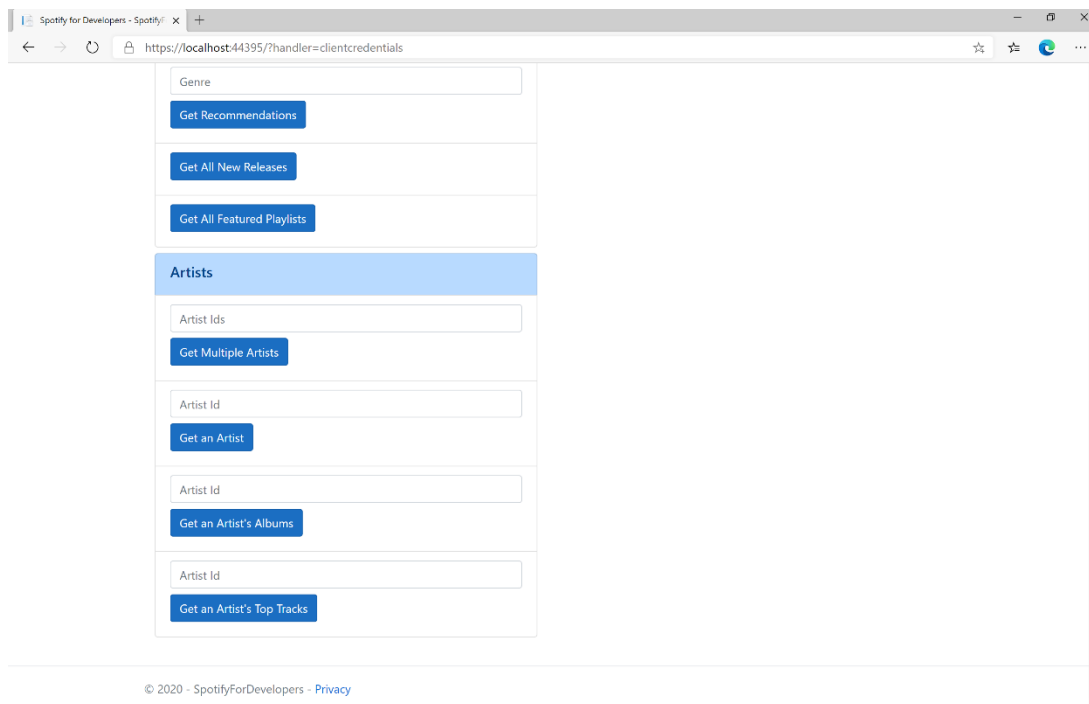
Step 8



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

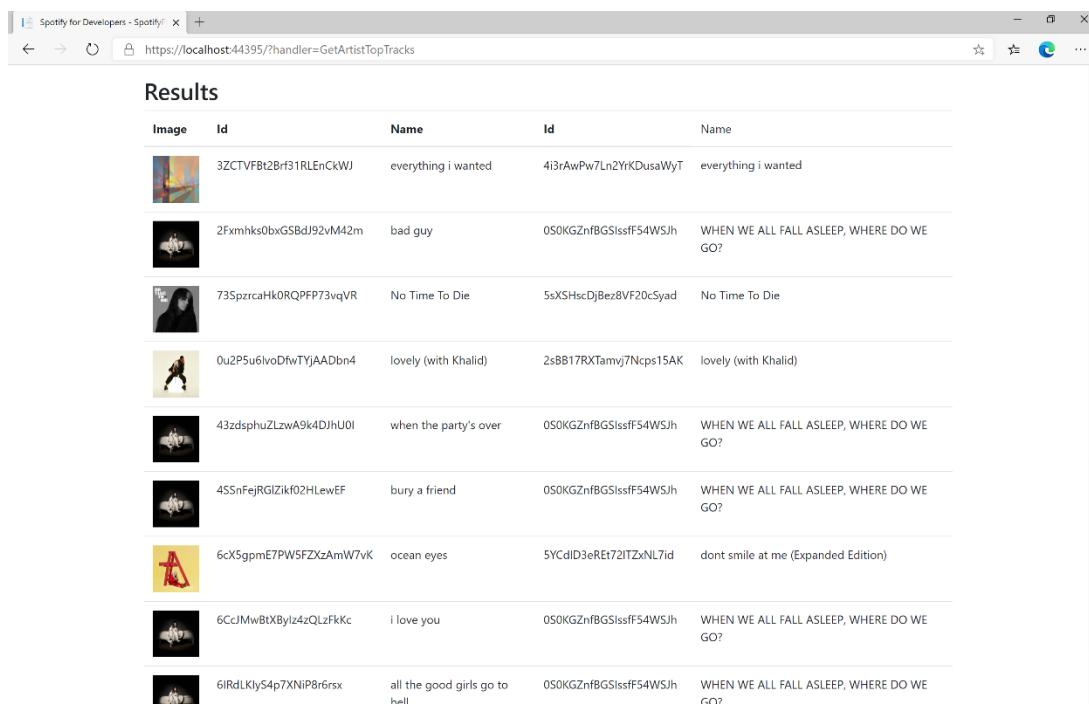
Step 9

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



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:



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

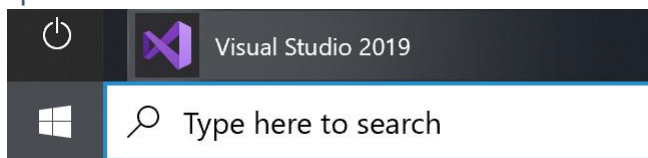
Get an Artist's Related Artists

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

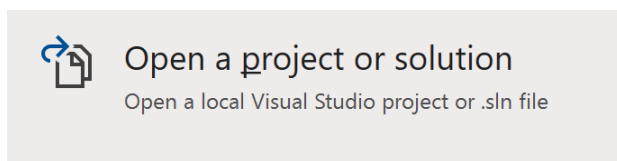
GET https://api.spotify.com/v1/artists/{id}/top-tracks	
Header	
Authorization	Valid Access Token from Spotify Accounts service
Path Parameter	
id	Spotify Id of the artist

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

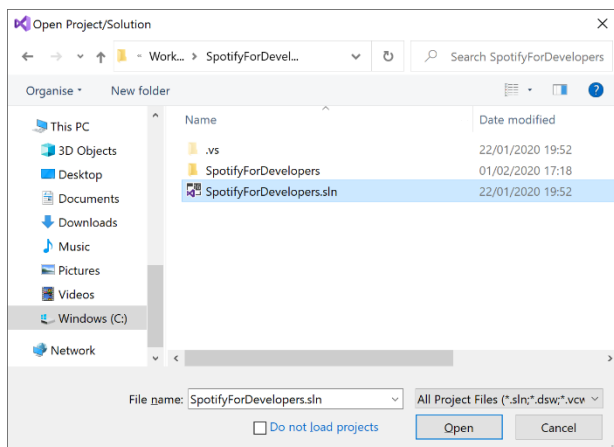
Step 1



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**

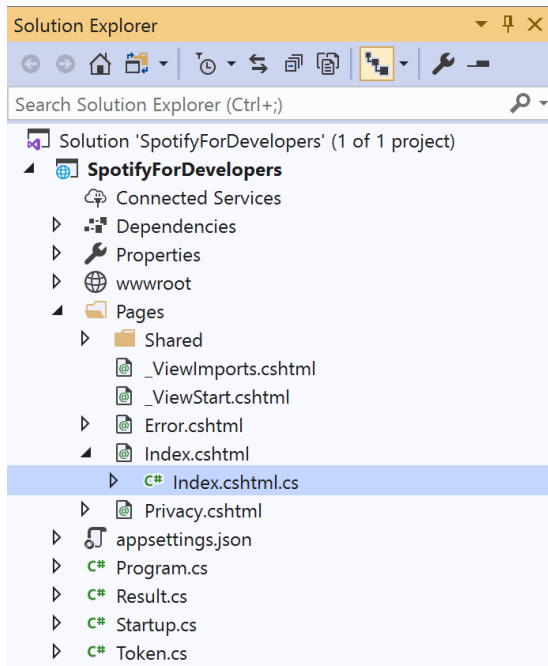


Once done, from the **Get started** screen for **Visual Studio 2019** select **Open a project or solution**



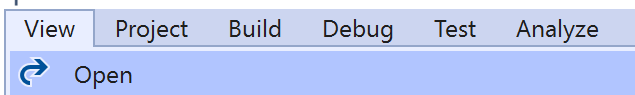
Then locate and select **SpotifyForDevelopers.sln** and select **Open** if you don't have this file already then please follow the previous parts of the workshop including **Getting Started**, **Authorisation Guide**, **Search & Browse** and **Artists, Albums & Tracks**

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**

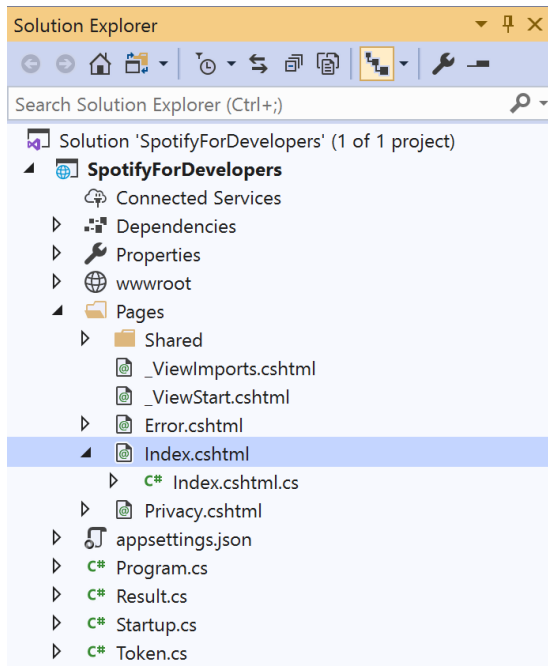
Step 4

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

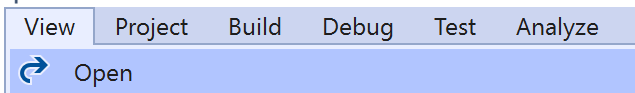
The **method** for `OnPostGetArtistRelatedArtistsAsync` is used to get the related **artists** of an **artist** by **Artist Id** on Spotify with the value and populate the **property** for Results accordingly.

Step 5



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

Step 6



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

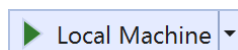
Step 7

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**.

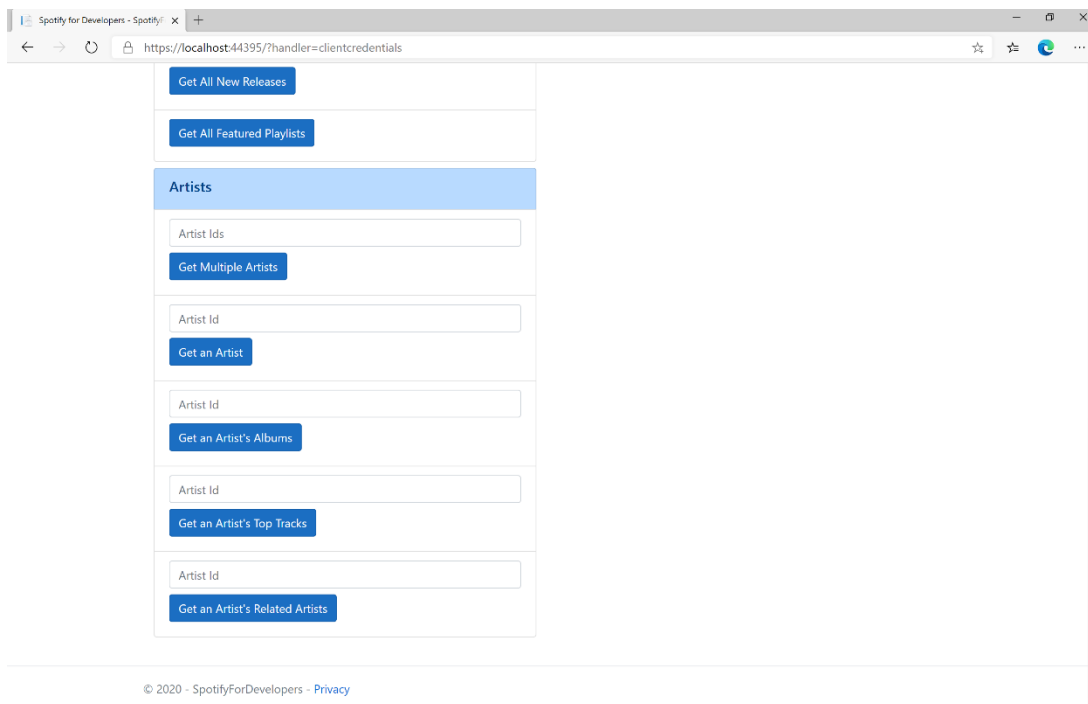
Step 8



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

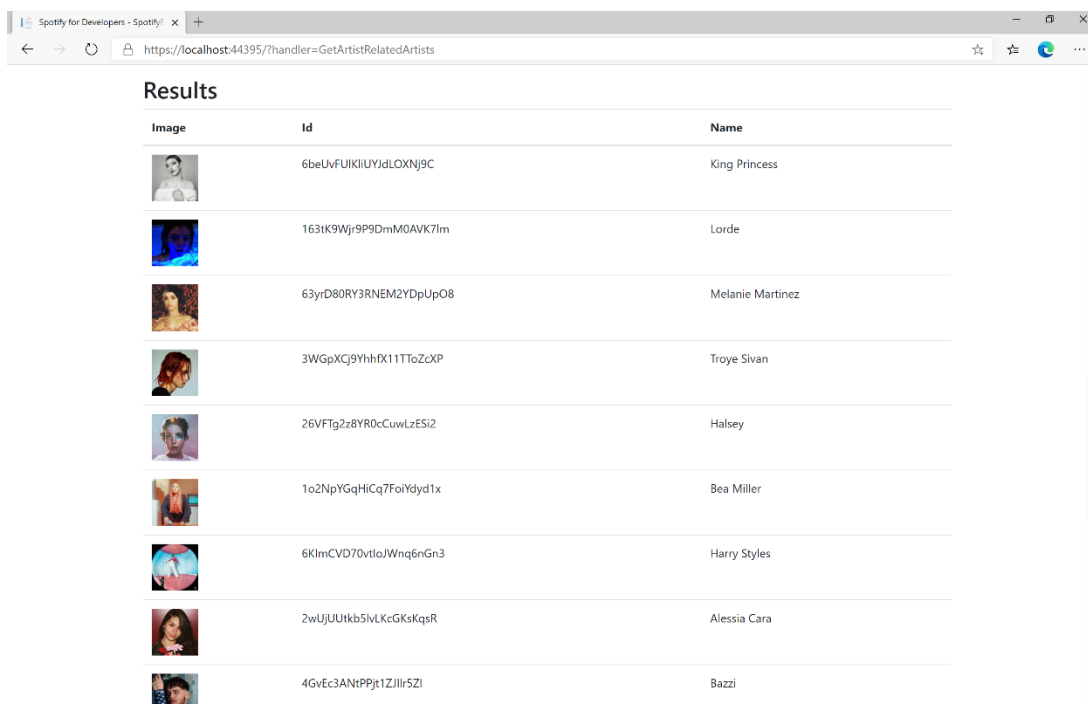
Step 9

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



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:



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

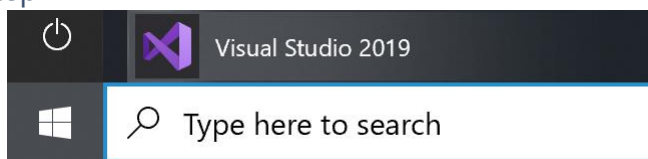
Get Multiple Albums

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

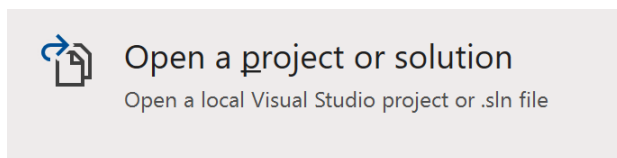
GET https://api.spotify.com/v1/albums	
Header	
Authorization	Valid Access Token from Spotify Accounts service
Query Parameter	
ids	Maximum of 20 comma-separated list of album Spotify Ids
market	ISO 3166-1 alpha-2 country code or "from_token" Provide to apply Track Relinking.

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

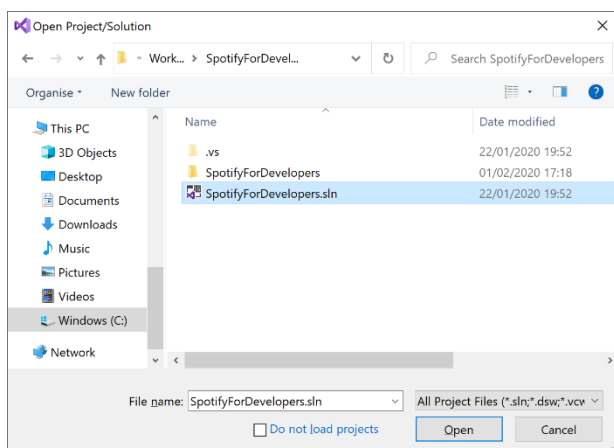
Step 1



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**

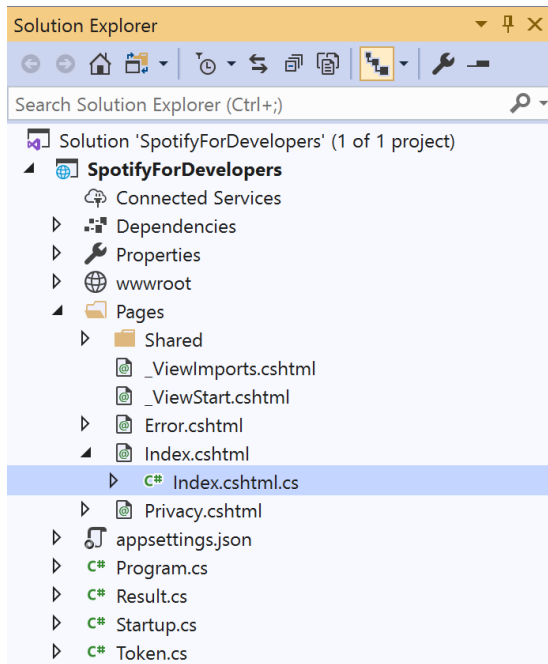


Once done, from the **Get started** screen for **Visual Studio 2019** select **Open a project or solution**



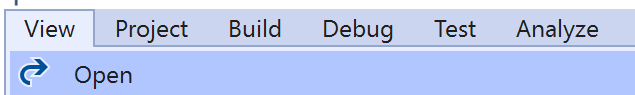
Then locate and select **SpotifyForDevelopers.sln** and select **Open** if you don't have this file already then please follow the previous parts of the workshop including **Getting Started**, **Authorisation Guide**, **Search & Browse** and **Artists, Albums & Tracks**

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**

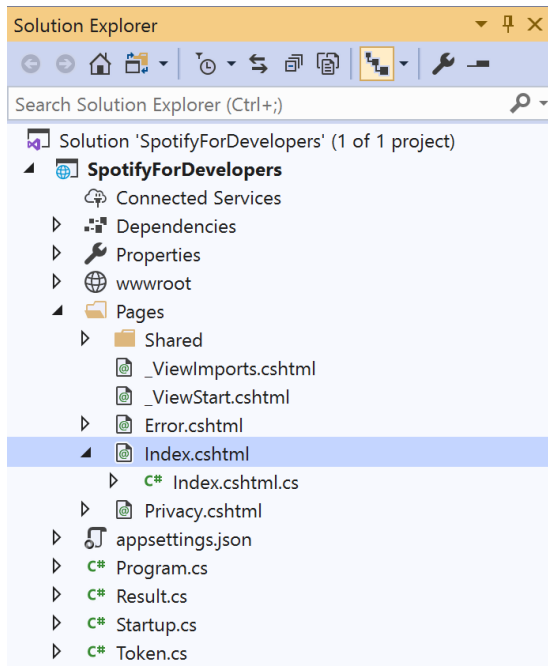
Step 4

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

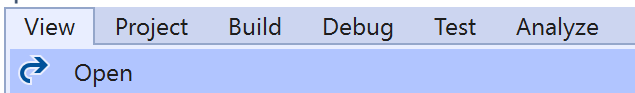
The **method** for `OnPostGetMultipleAlbumsAsync` is used to get **albums** by multiple **Album Id** on Spotify with the value and populate the **property** for Results accordingly.

Step 5



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

Step 6



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

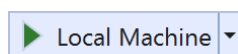
Step 7

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**.

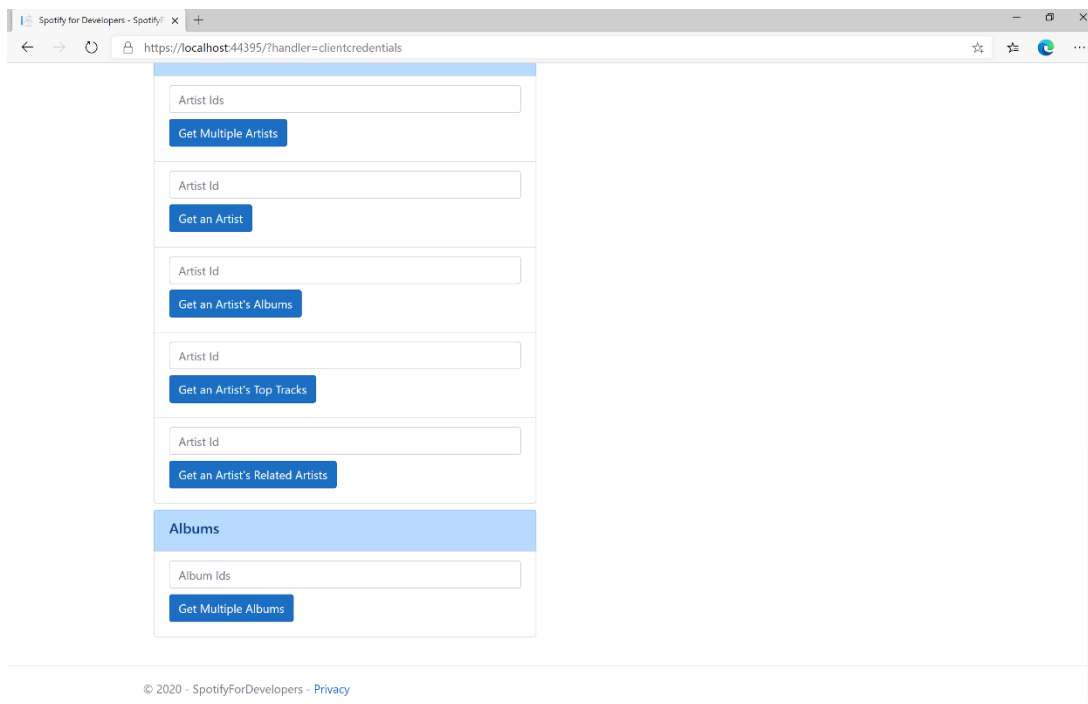
Step 8



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

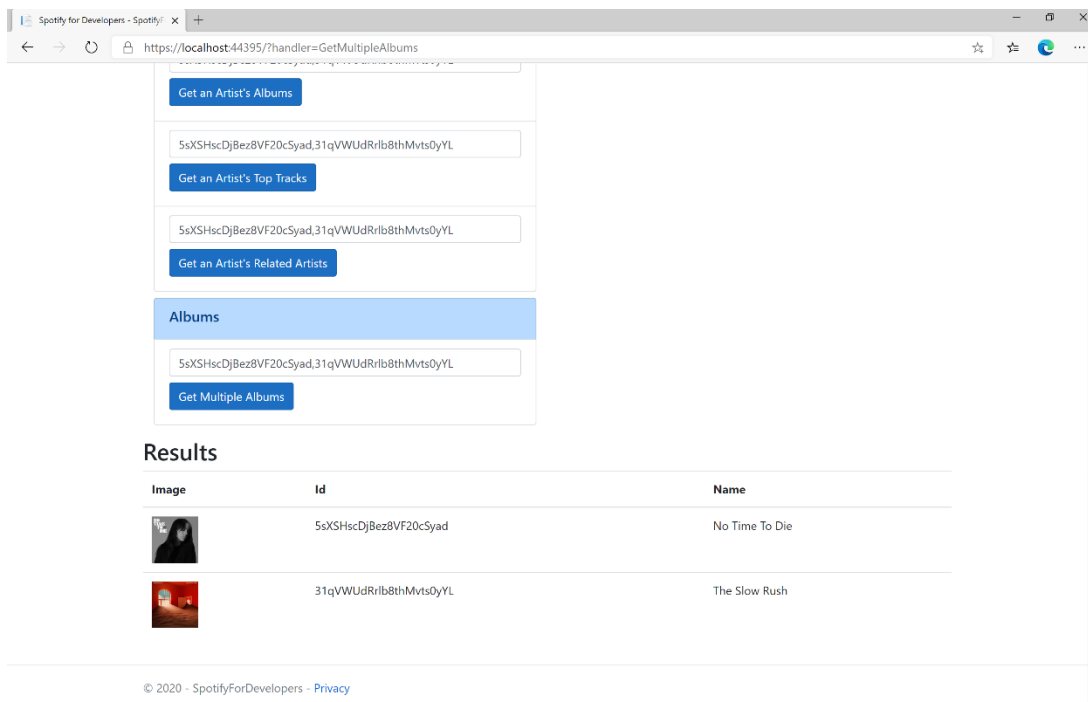
Step 9

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



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:



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

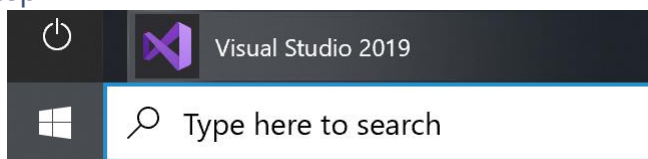
Get an Album

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

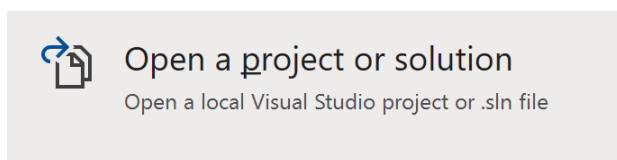
GET https://api.spotify.com/v1/albums/{id}	
Header	
Authorization	Valid Access Token from Spotify Accounts service
Path Parameter	
id	Spotify Id of the album
market	ISO 3166-1 alpha-2 country code e.g. "GB"

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

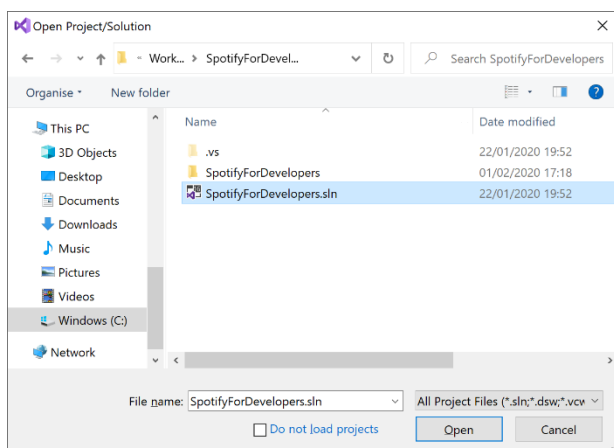
Step 1



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**

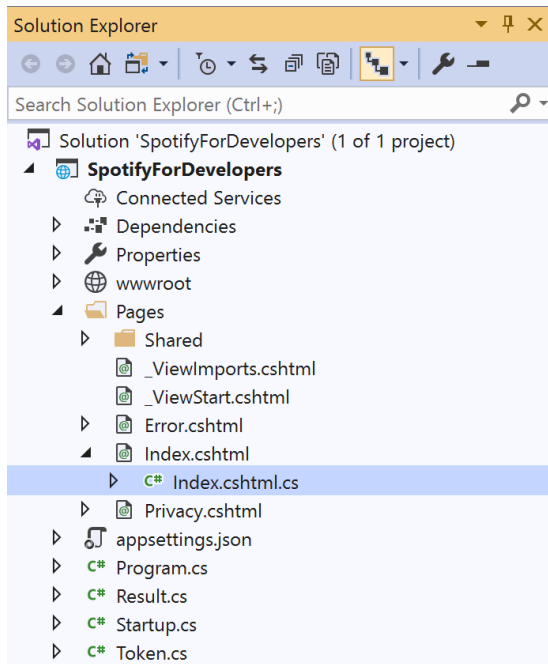


Once done, from the **Get started** screen for **Visual Studio 2019** select **Open a project or solution**



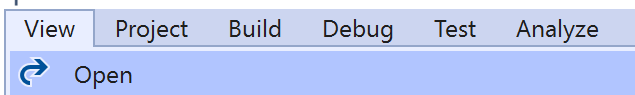
Then locate and select **SpotifyForDevelopers.sln** and select **Open** if you don't have this file already then please follow the previous parts of the workshop including **Getting Started**, **Authorisation Guide**, **Search & Browse** and **Artists, Albums & Tracks**

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**

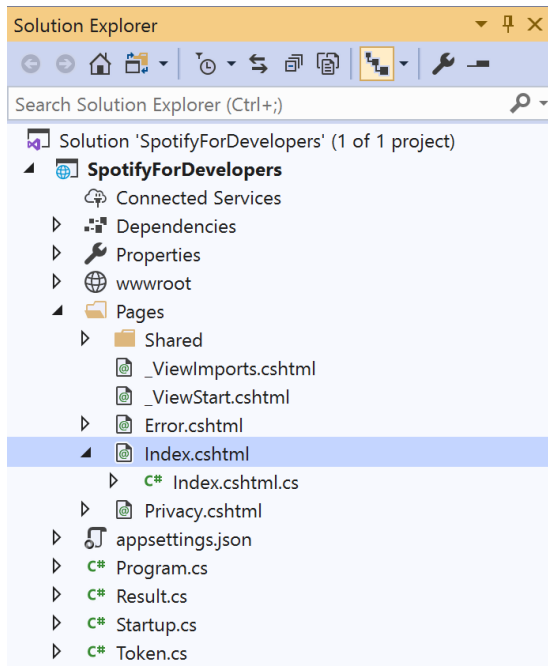
Step 4

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

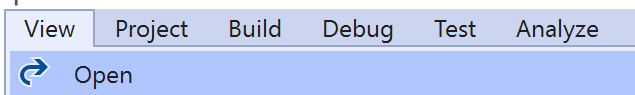
The **method** for `OnPostGetAlbumAsync` is used to get an **album** by **Album Id** on Spotify with the value and populate the **property** for `Results` accordingly.

Step 5



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

Step 6



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

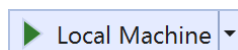
Step 7

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**.

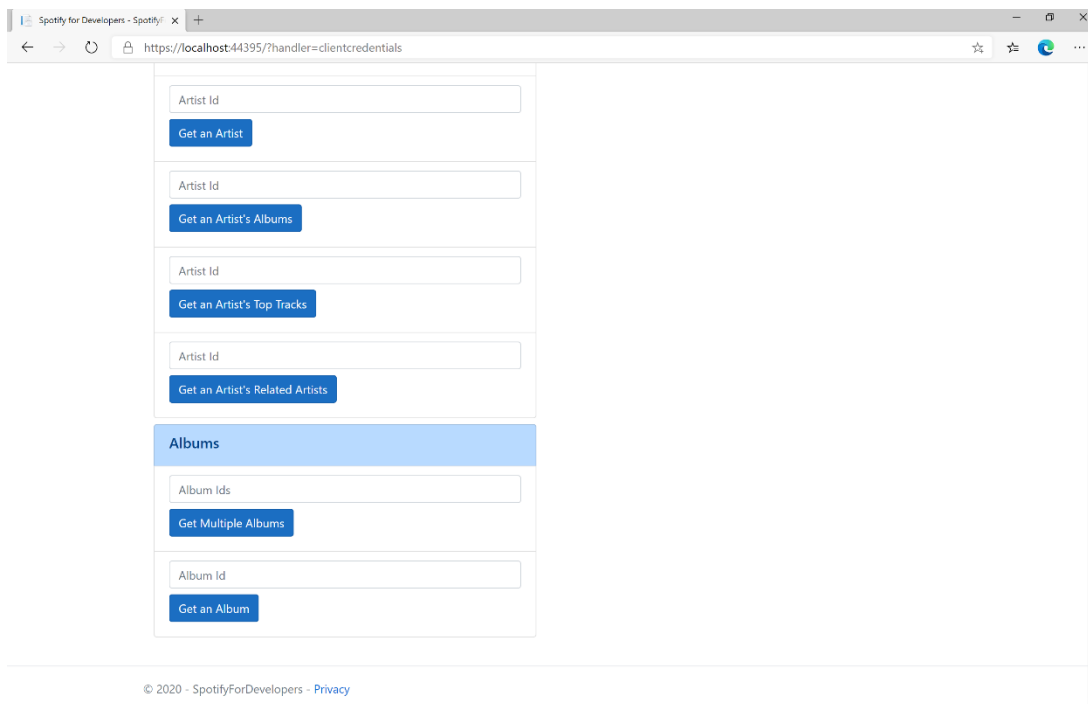
Step 8



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

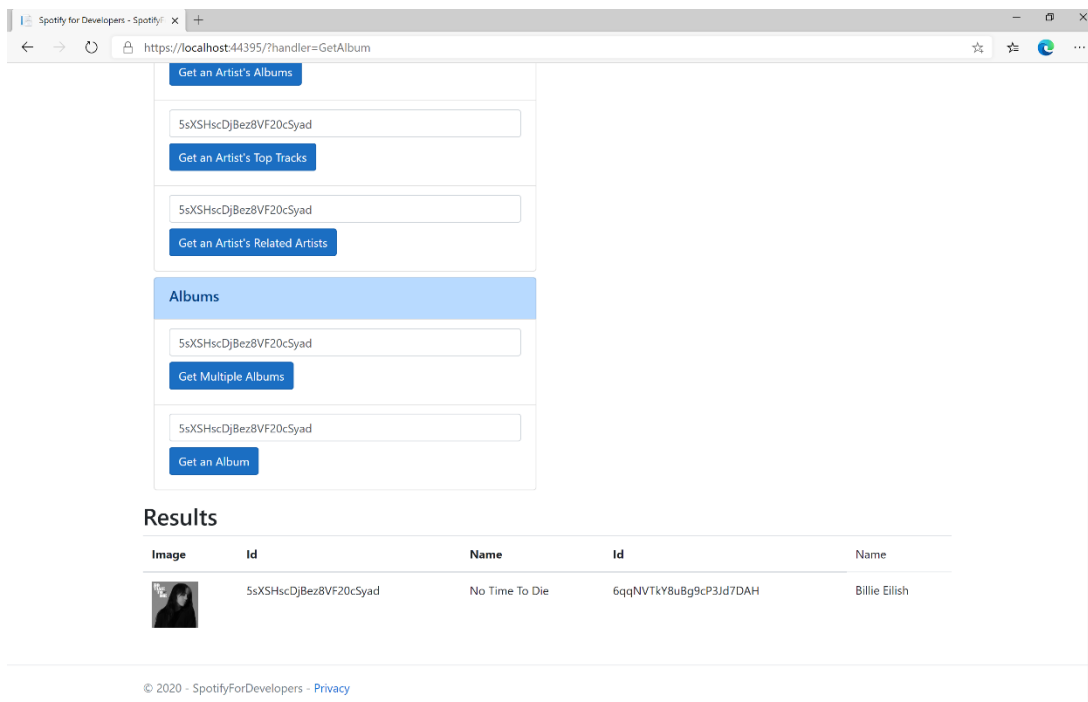
Step 9

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



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:



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

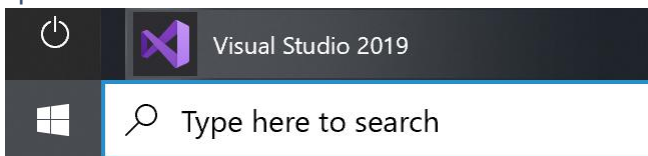
Get an Album's Tracks

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

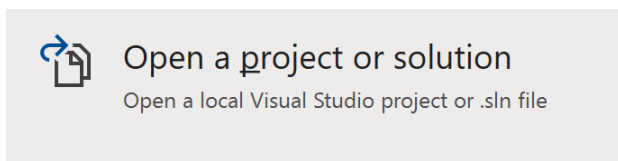
GET https://api.spotify.com/v1/albums/{id}/tracks	
Header	
Authorization	Valid Access Token from Spotify Accounts service
Path Parameter	
id	Spotify Id of the album
Query Parameters	
limit	Maximum number of results to return
offset	Index of first result to return
market	ISO 3166-1 alpha-2 country code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

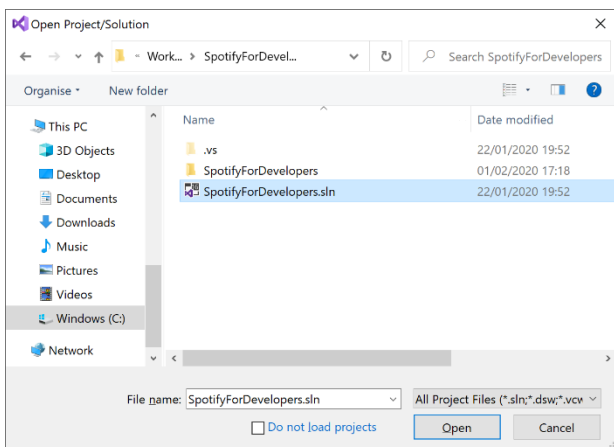
Step 1



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**

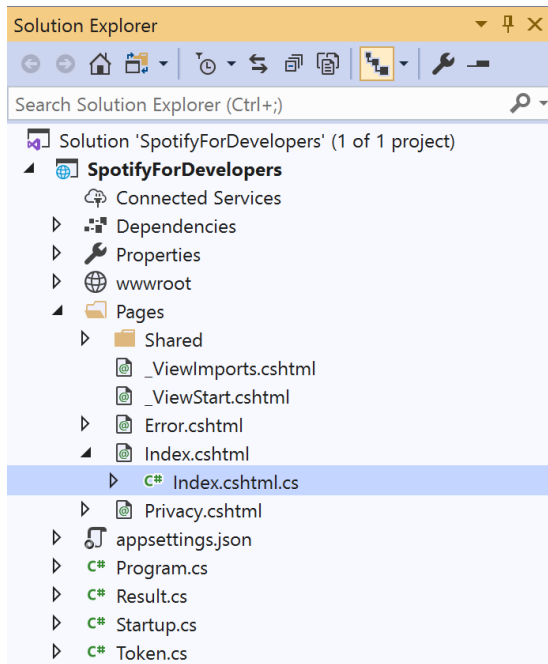


Once done, from the **Get started** screen for **Visual Studio 2019** select **Open a project or solution**



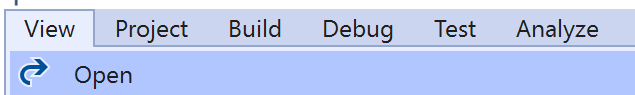
Then locate and select **SpotifyForDevelopers.sln** and select **Open** if you don't have this file already then please follow the previous parts of the workshop including **Getting Started**, **Authorisation Guide**, **Search & Browse** and **Artists, Albums & Tracks**

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**

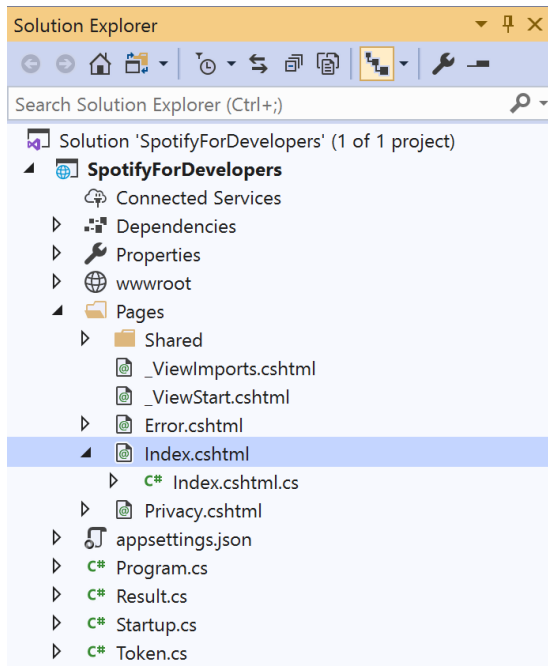
Step 4

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

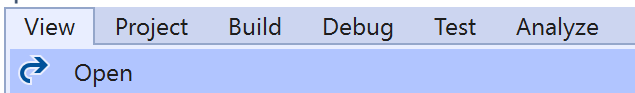
The **method** for `OnPostGetAlbumTracksAsync` is used to get the **tracks** of an **album** by **Album Id** on Spotify with the value and populate the **property** for `Results` accordingly.

Step 5



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

Step 6



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

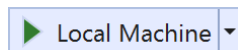
Step 7

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**.

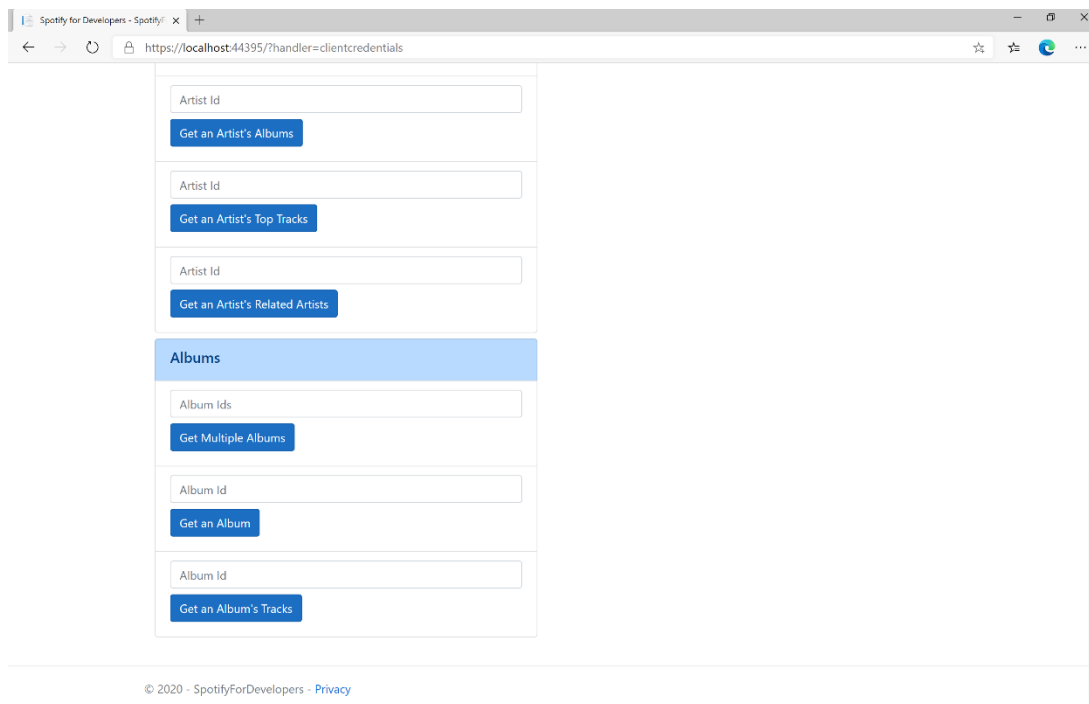
Step 8



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

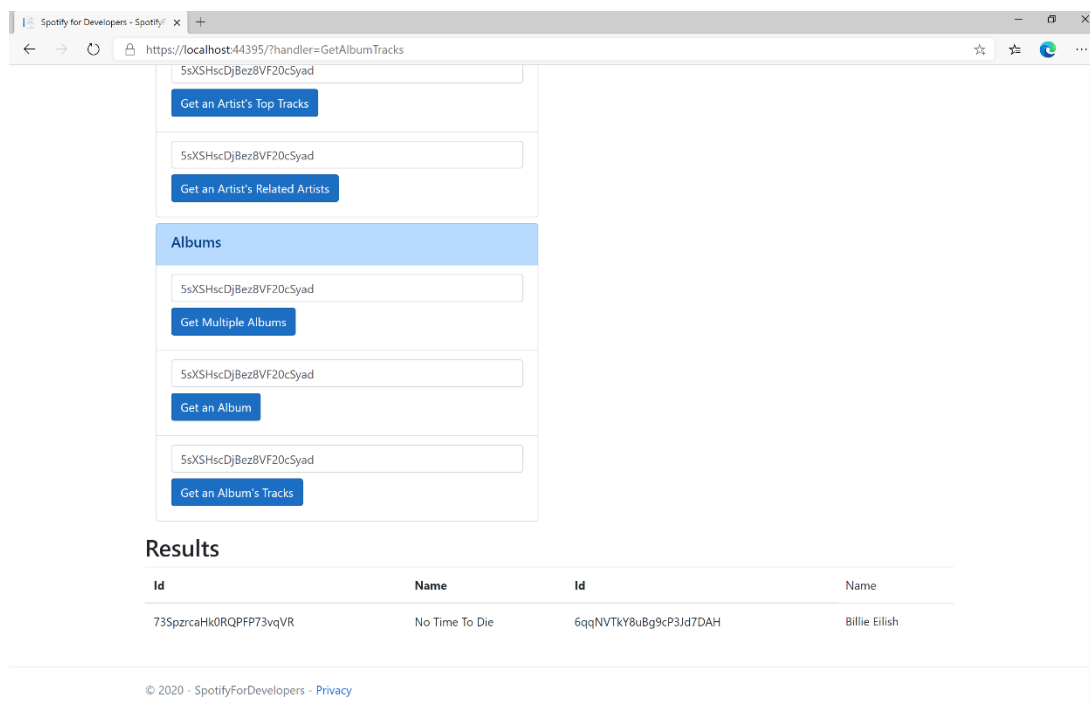
Step 9

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



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:



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

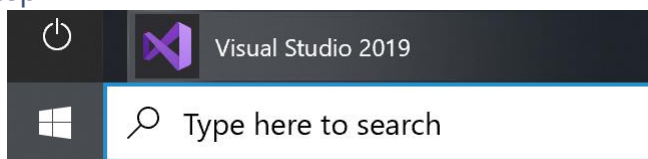
Get Multiple Tracks

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

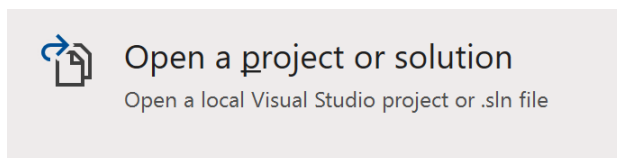
GET https://api.spotify.com/v1/tracks	
Header	
Authorization	Valid Access Token from Spotify Accounts service
Query Parameter	
ids	Maximum of 50 comma-separated list of track Spotify Ids
market	ISO 3166-1 alpha-2 country code e.g. "GB". Provide to apply Track Relinking

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

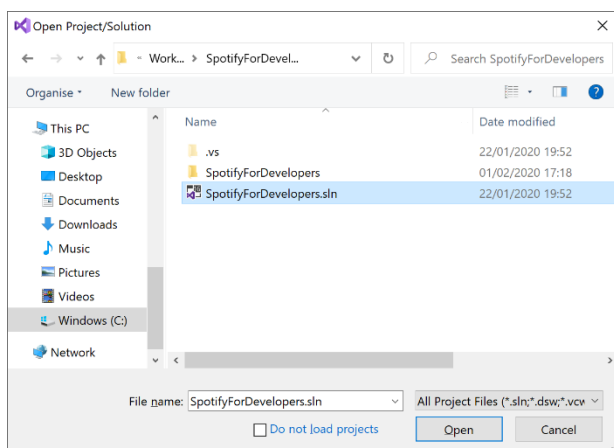
Step 1



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**

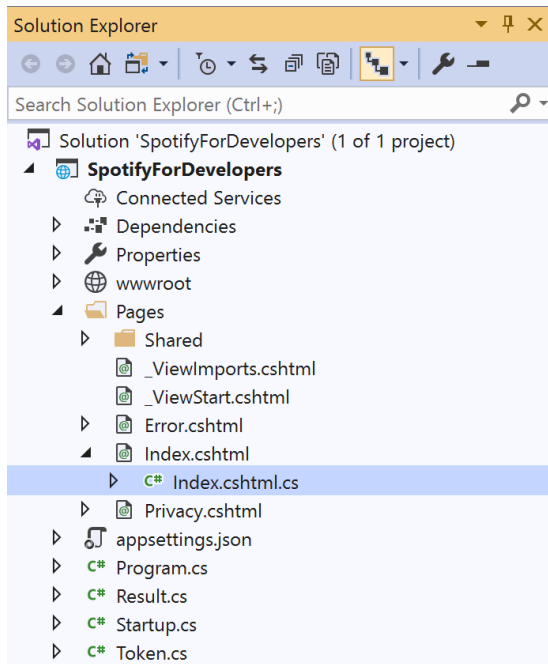


Once done, from the **Get started** screen for **Visual Studio 2019** select **Open a project or solution**



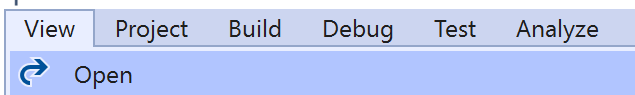
Then locate and select **SpotifyForDevelopers.sln** and select **Open** if you don't have this file already then please follow the previous parts of the workshop including **Getting Started**, **Authorisation Guide**, **Search & Browse** and **Artists, Albums & Tracks**

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**

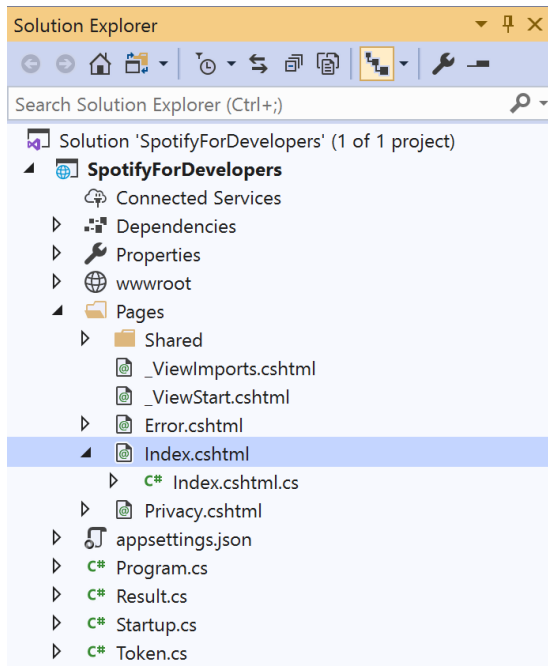
Step 4

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

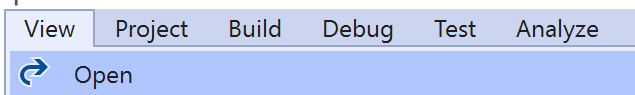
The **method** for `OnPostGetSeveralTracksAsync` is used to get **tracks** by multiple **Track Ids** on Spotify with the value and populate the **property** for `Results` accordingly.

Step 5



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

Step 6



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

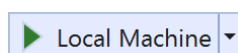
Step 7

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**.

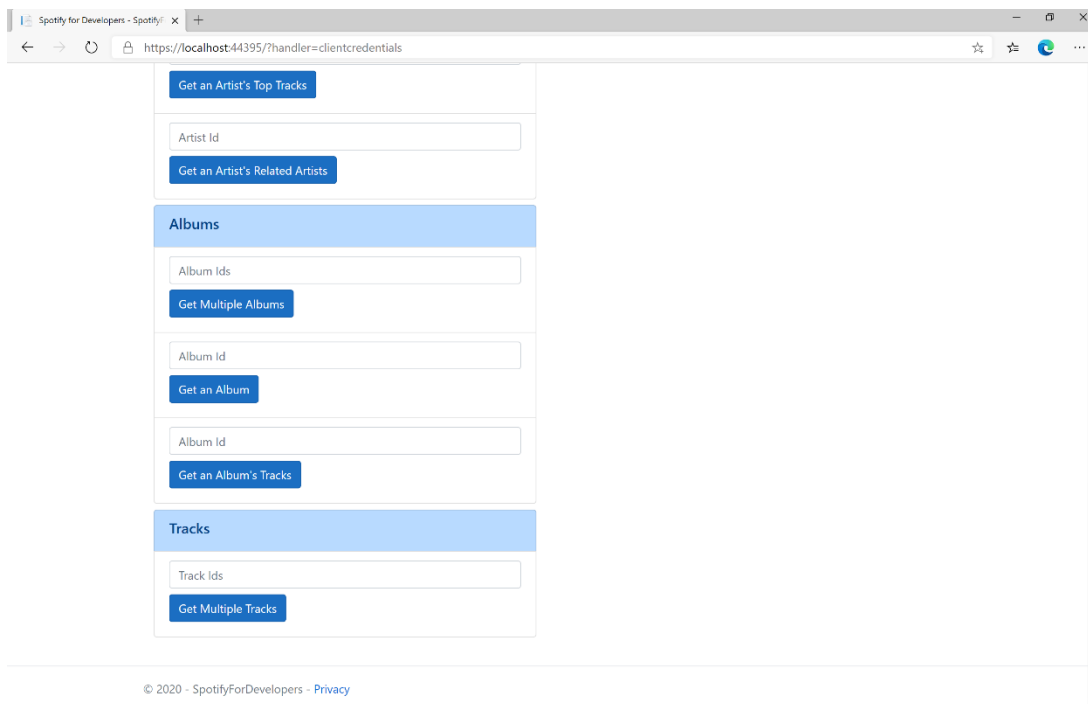
Step 8



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

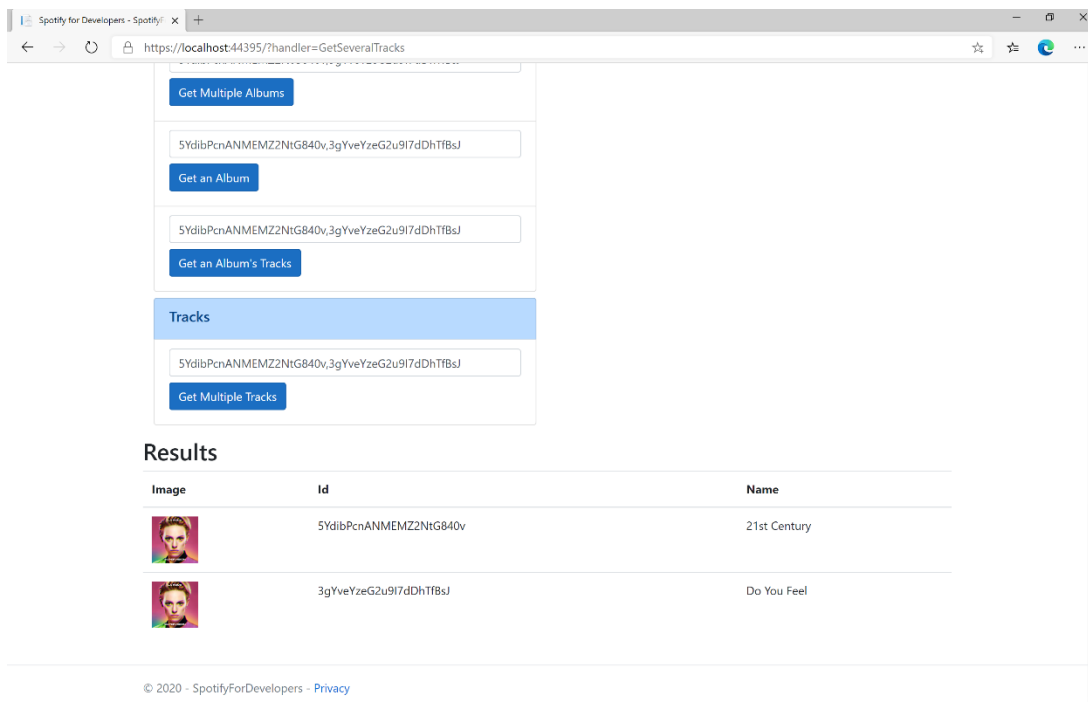
Step 9

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



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:



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

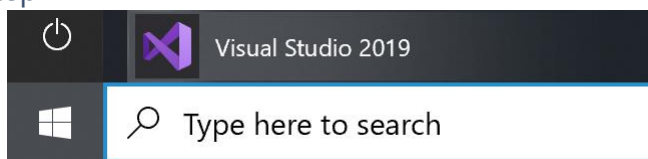
Get a Track

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

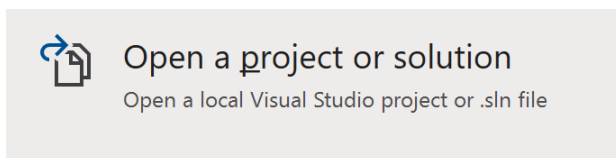
GET https://api.spotify.com/v1/tracks/{id}	
Header	
Authorization	Valid Access Token from Spotify Accounts service
Path Parameter	
id	Spotify Id of the track
market	ISO 3166-1 alpha-2 country code e.g. "GB". Provide to apply Track Relinking

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

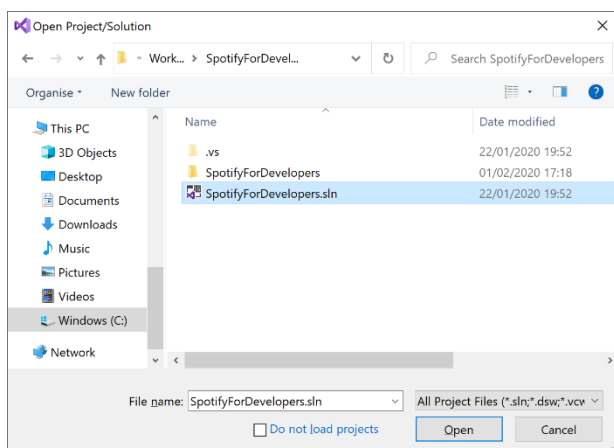
Step 1



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**

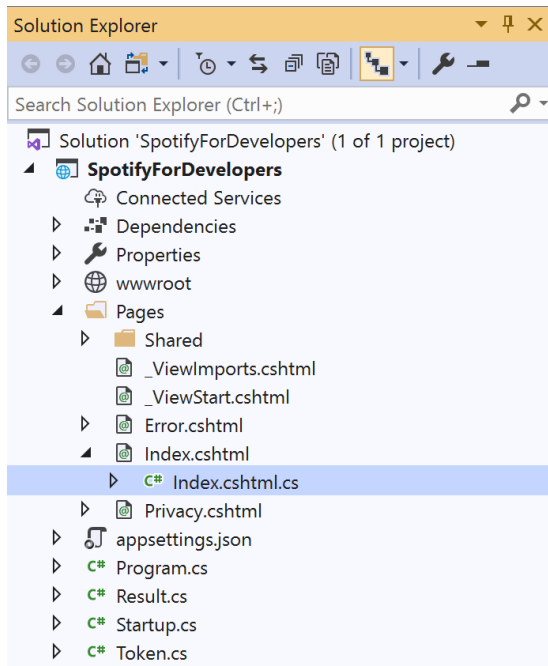


Once done, from the **Get started** screen for **Visual Studio 2019** select **Open a project or solution**



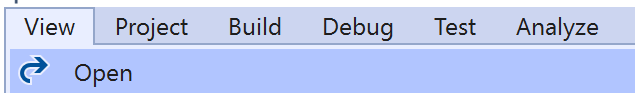
Then locate and select **SpotifyForDevelopers.sln** and select **Open** if you don't have this file already then please follow the previous parts of the workshop including **Getting Started**, **Authorisation Guide**, **Search & Browse** and **Artists, Albums & Tracks**

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**

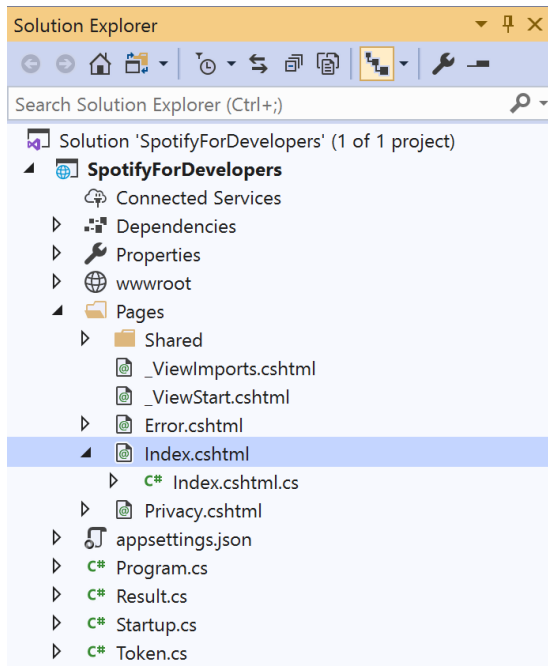
Step 4

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

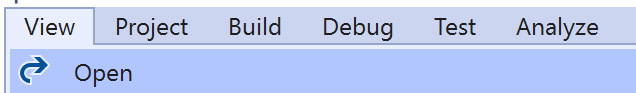
The **method** for `OnPostGetTrackAsync` is used to get a **track** by **Track Id** on Spotify with the value and populate the **property** for `Results` accordingly.

Step 5



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

Step 6



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

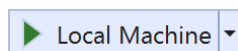
Step 7

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**.

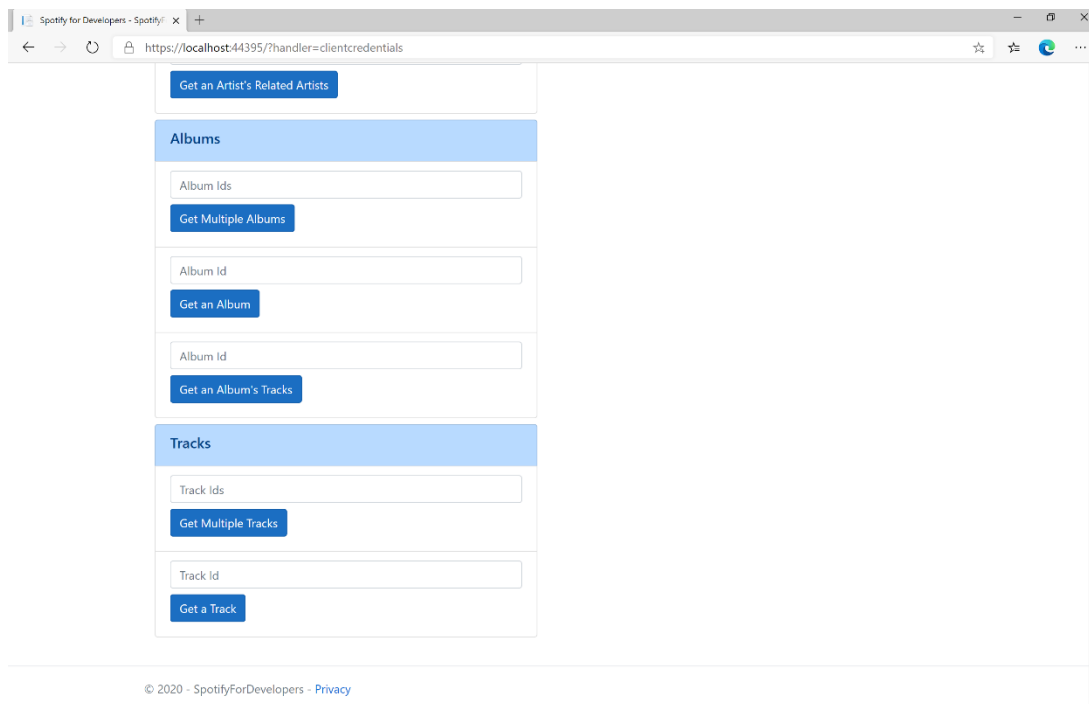
Step 8



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

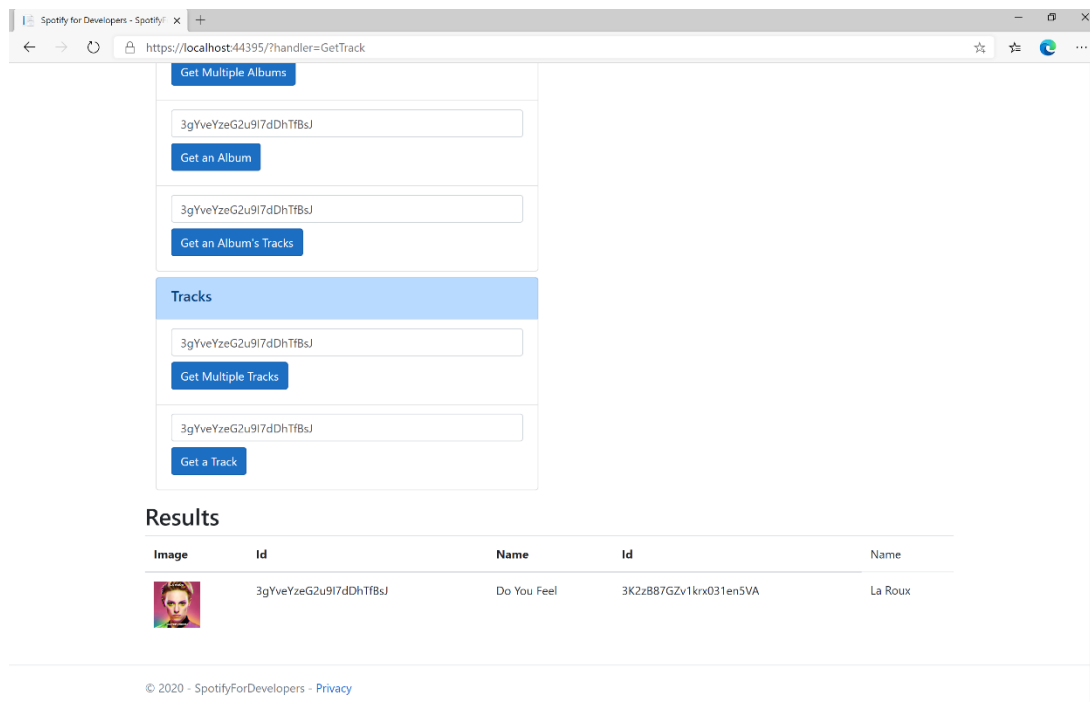
Step 9

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



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:



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

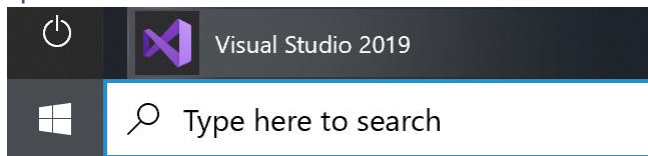
Get Audio Features for Several Tracks

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

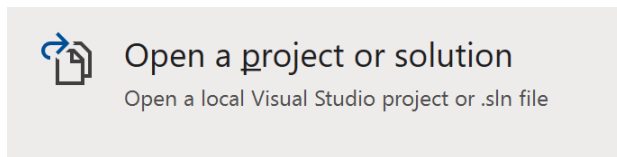
https://api.spotify.com/v1/audio-features	
Header	
Authorization	Valid Access Token from Spotify Accounts service
Path Parameter	
ids	Maximum of 100 comma-separated list of track Spotify Ids

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

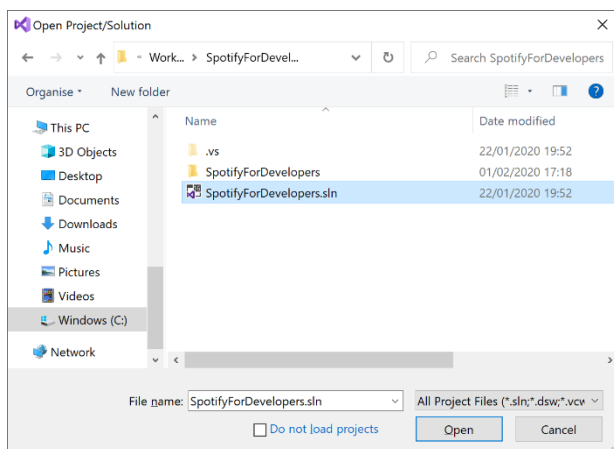
Step 1



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**

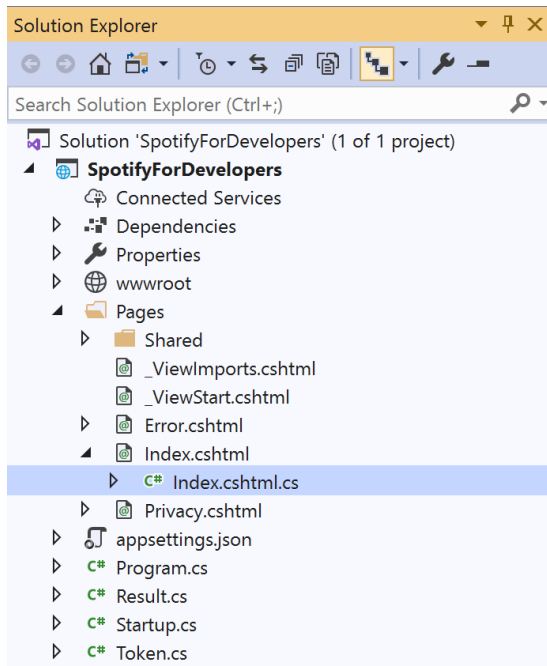


Once done, from the **Get started** screen for **Visual Studio 2019** select **Open a project or solution**



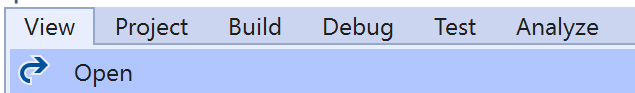
Then locate and select **SpotifyForDevelopers.sln** and select **Open** if you don't have this file already then please follow the previous parts of the workshop including **Getting Started**, **Authorisation Guide**, **Search & Browse** and **Artists, Albums & Tracks**

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**

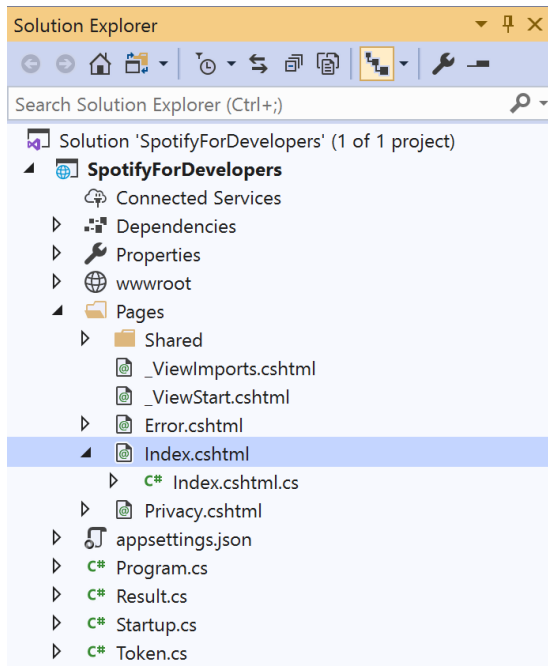
Step 4

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

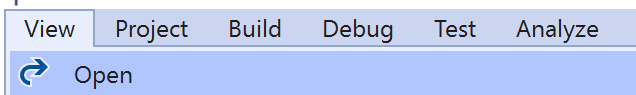
The **method** for `OnPostGetAudioFeaturesForSeveralTracksAsync` is used to get **audio features** for **tracks** by multiple **Track Ids** on Spotify with the value and populate the **property** for `Results` accordingly.

Step 5



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

Step 6



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

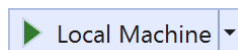
Step 7

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**.

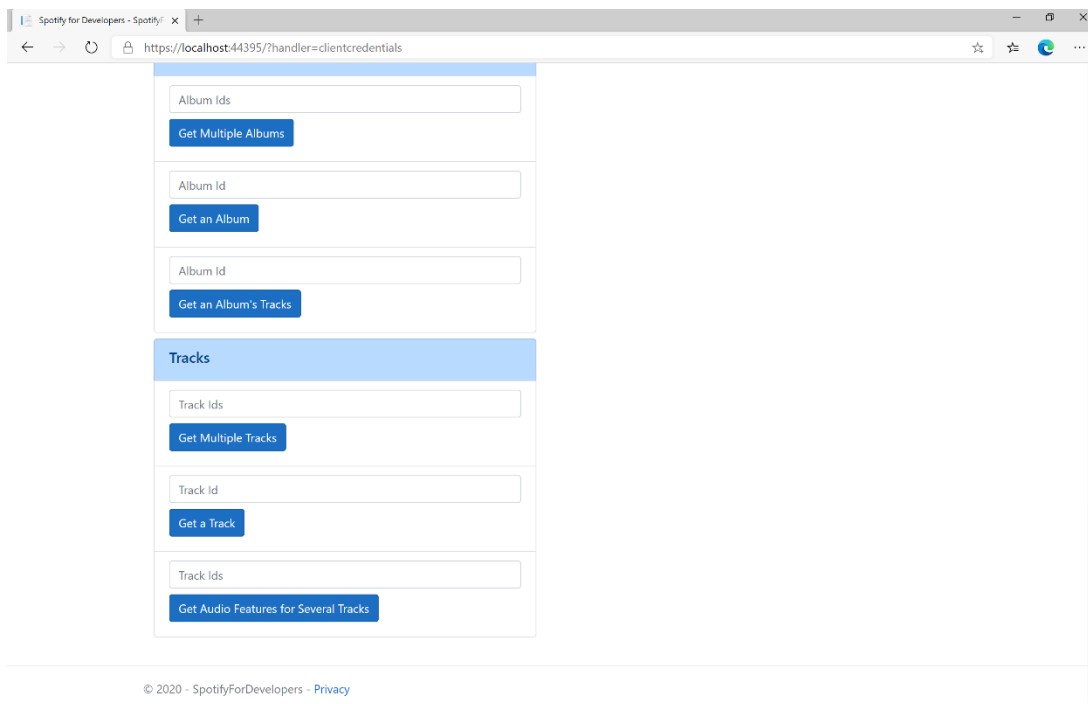
Step 8



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

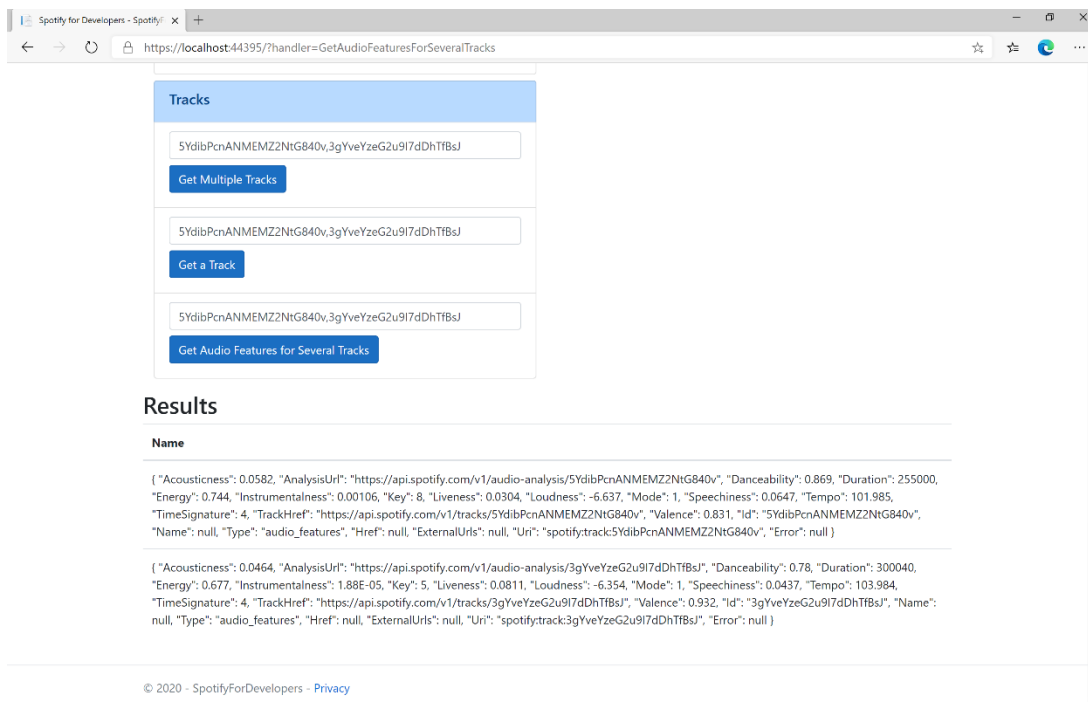
Step 9

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



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:



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

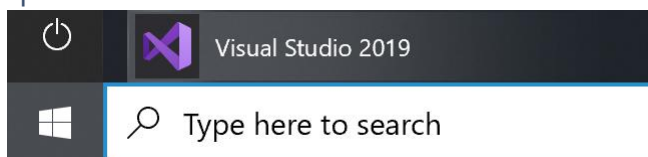
Get Audio Features for a Track

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

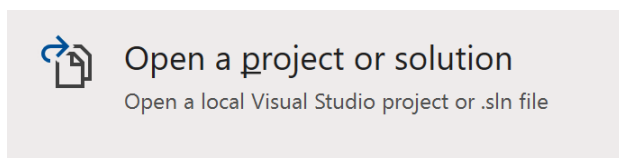
GET https://api.spotify.com/v1/audio-features/{id}	
Header	
Authorization	Valid Access Token from Spotify Accounts service
Path Parameter	
id	Spotify Id of the track

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

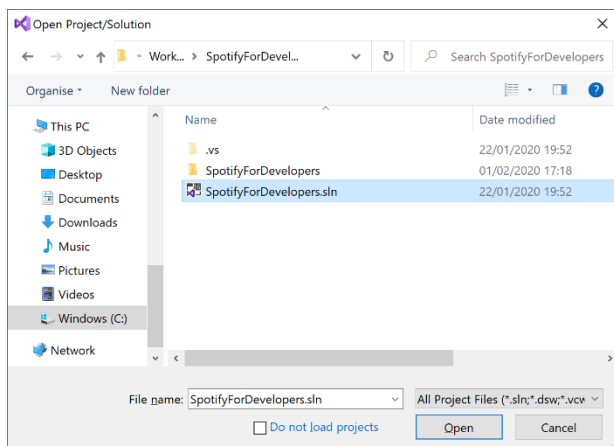
Step 1



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**

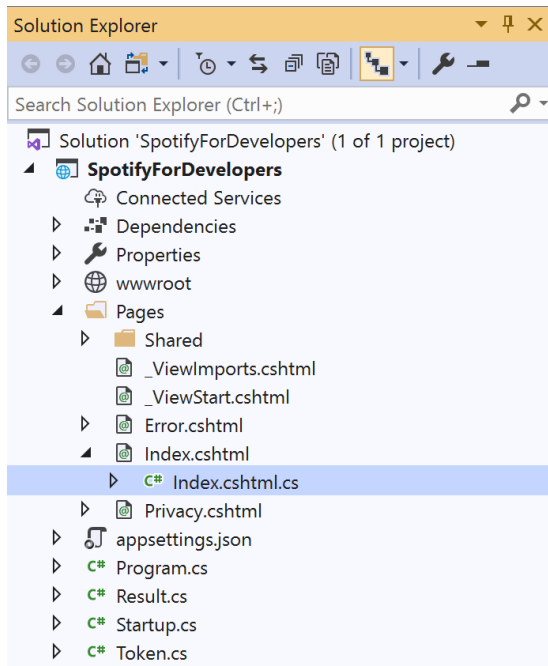


Once done, from the **Get started** screen for **Visual Studio 2019** select **Open a project or solution**



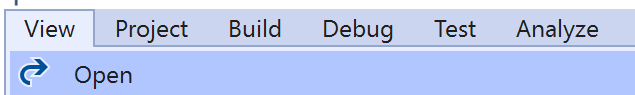
Then locate and select **SpotifyForDevelopers.sln** and select **Open** if you don't have this file already then please follow the previous parts of the workshop including **Getting Started**, **Authorisation Guide**, **Search & Browse** and **Artists, Albums & Tracks**

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**

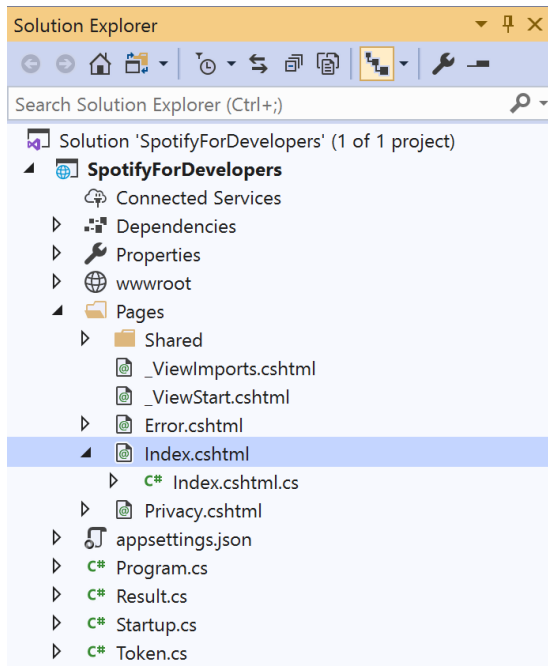
Step 4

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

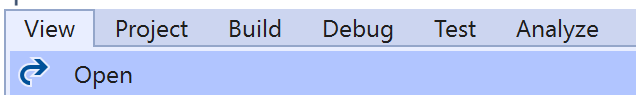
The **method** for `OnPostGetAudioFeaturesForTrackAsync` is used to get **audio features** for **track** by **Track Id** on Spotify with the value and populate the **property** for **Results** accordingly.

Step 5



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

Step 6



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

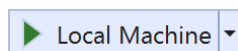
Step 7

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**.

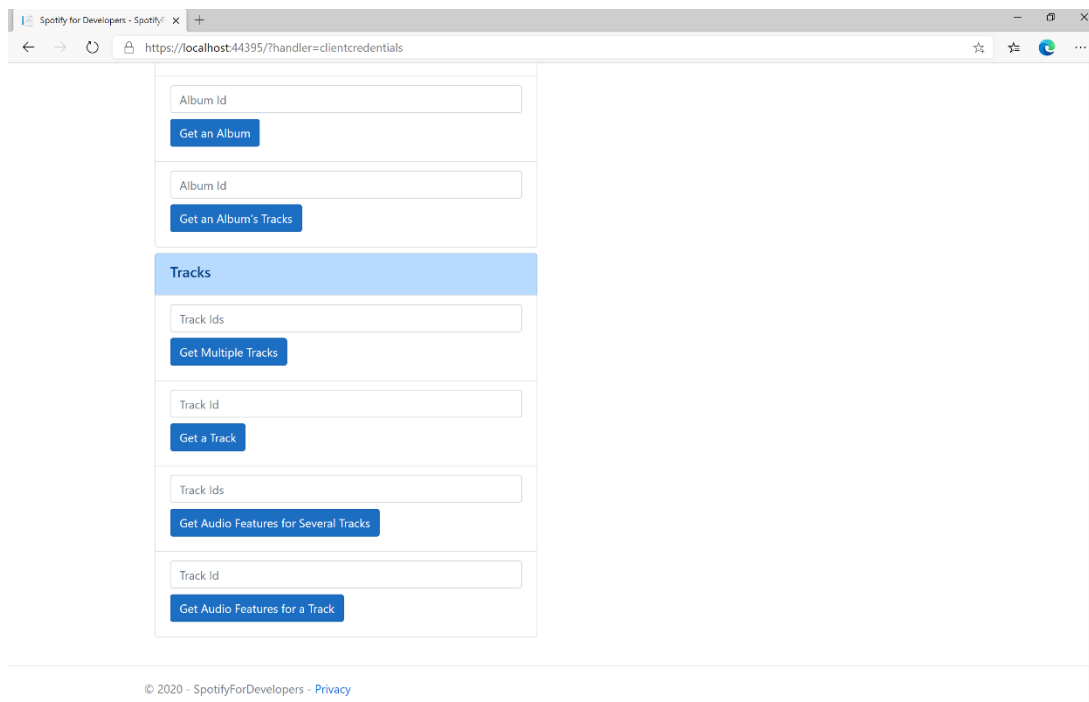
Step 8



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

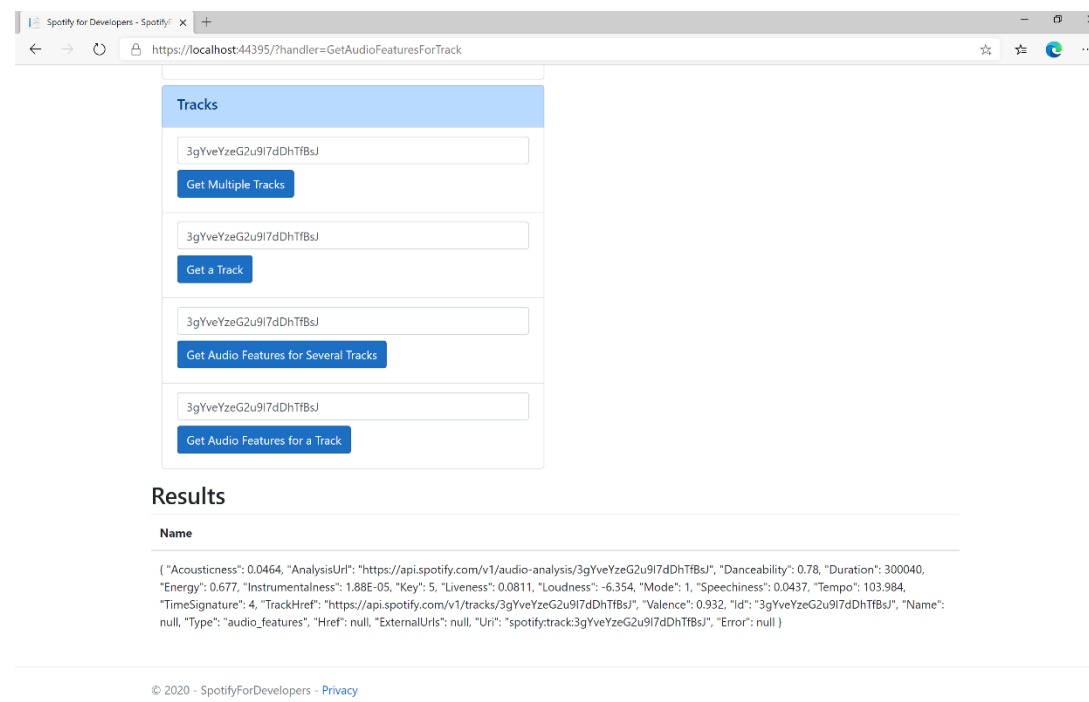
Step 9

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



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:



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

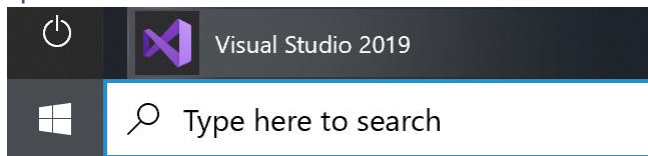
Get Audio Analysis for a Track

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

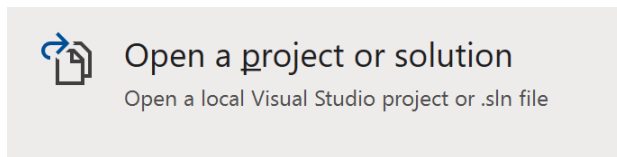
GET https://api.spotify.com/v1/audio-analysis/{id}	
Header	
Authorization	Valid Access Token from Spotify Accounts service
Path Parameter	
id	Spotify Id of the track

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

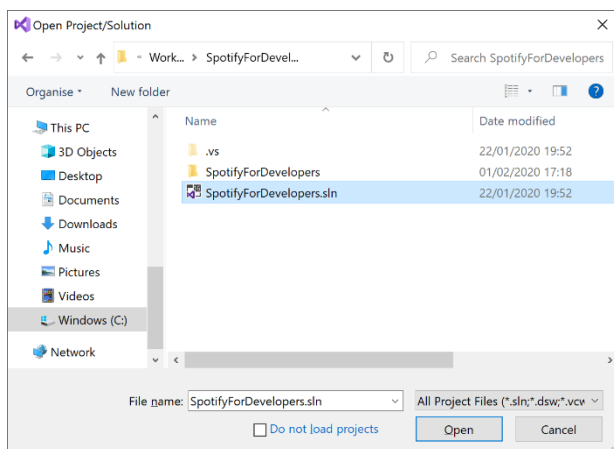
Step 1



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**

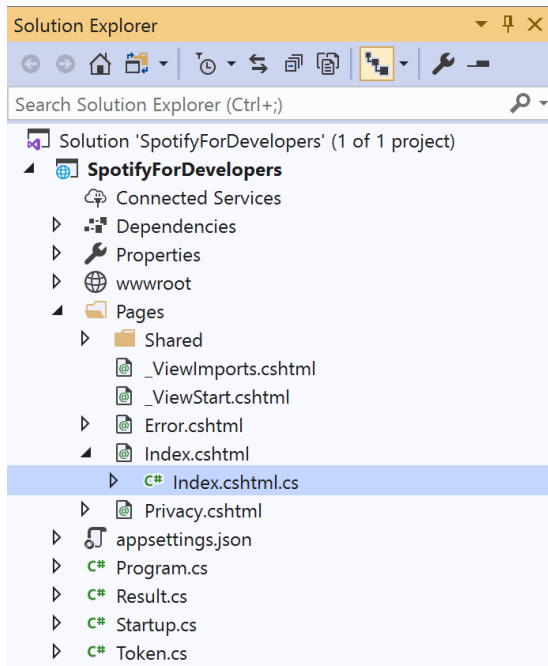


Once done, from the **Get started** screen for **Visual Studio 2019** select **Open a project or solution**



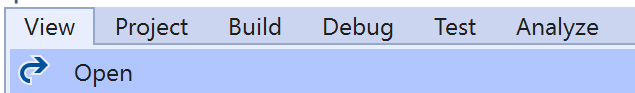
Then locate and select **SpotifyForDevelopers.sln** and select **Open** if you don't have this file already then please follow the previous parts of the workshop including **Getting Started**, **Authorisation Guide**, **Search & Browse** and **Artists, Albums & Tracks**

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**

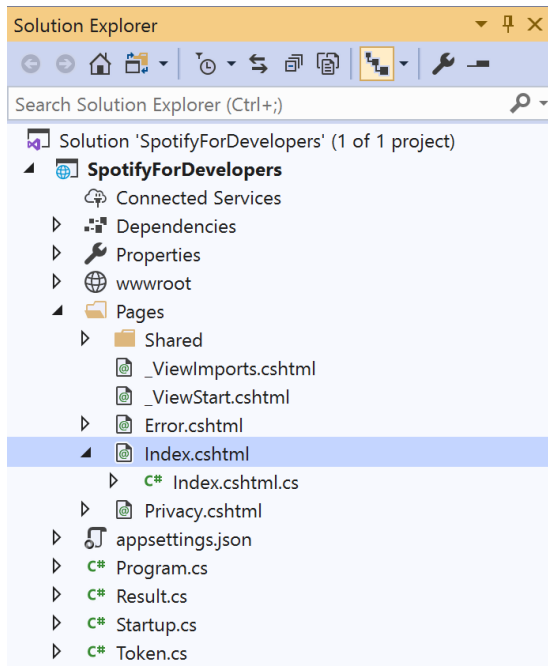
Step 4

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

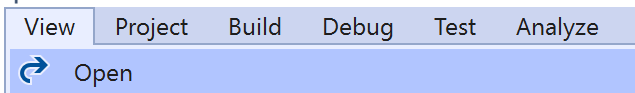
The **method** for `OnPostGetTrackAudioAnalysisAsync` is used to get **audio analysis** for **track** by **Track Id** on Spotify with the value and populate the **property** for `Results` accordingly.

Step 5



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

Step 6



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

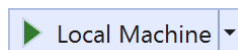
Step 7

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**.

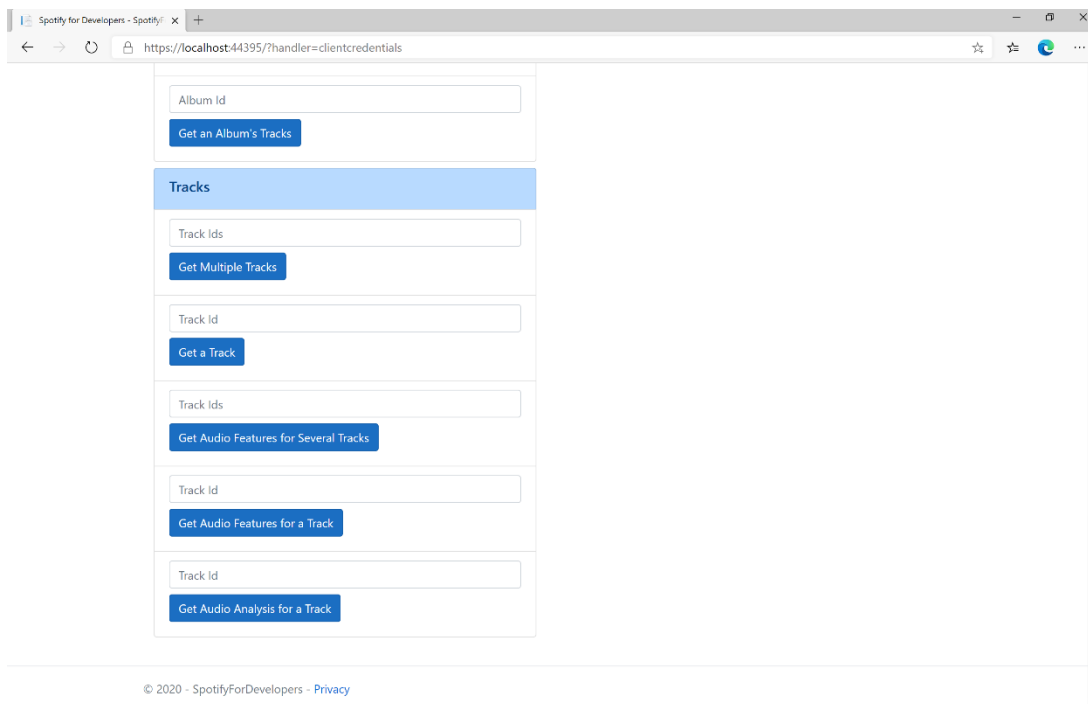
Step 8



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

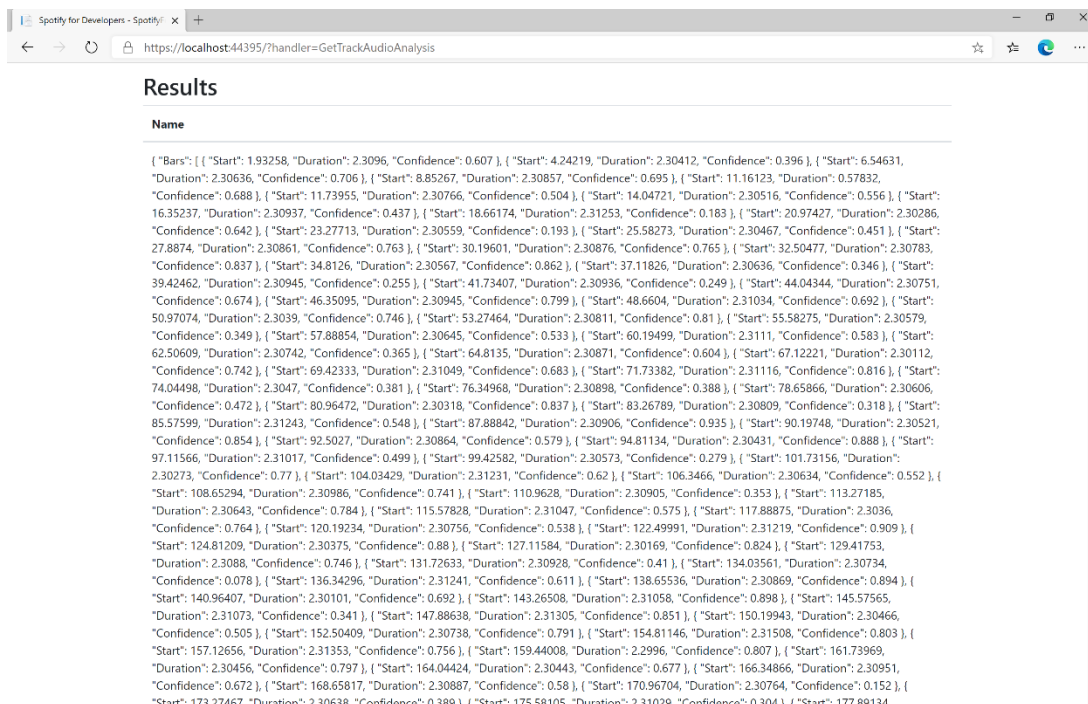
Step 9

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



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:



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

