|  |
| --- |
|  |
| Spotify for Developers  Episodes & Shows |
| |  |  |  | | --- | --- | --- | |  |  |  | |

## Get an Episode

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

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

|  |  |
| --- | --- |
| Header | Response |
| Success | |
| HTTP Status 200 OK | Episode Object |
| Error | |
| Error Code | Error Object |

### Step 1

|  |  |
| --- | --- |
| A screenshot of a cell phone  Description automatically generated | 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** |
| A screenshot of a social media post  Description automatically generated | Then locate and select **SpotifyForDevelopers.sln** and select **Open** if you don’t have this file already then please follow the previous parts of the workshop including **Getting Started**, **Authorisation Guide**, **Search** **&** **Browse** and **Artists, Albums & Tracks** |

### Step 2

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

### Step 3

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

### Step 4

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

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

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

### Step 5

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

### Step 6

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

### Step 7

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

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

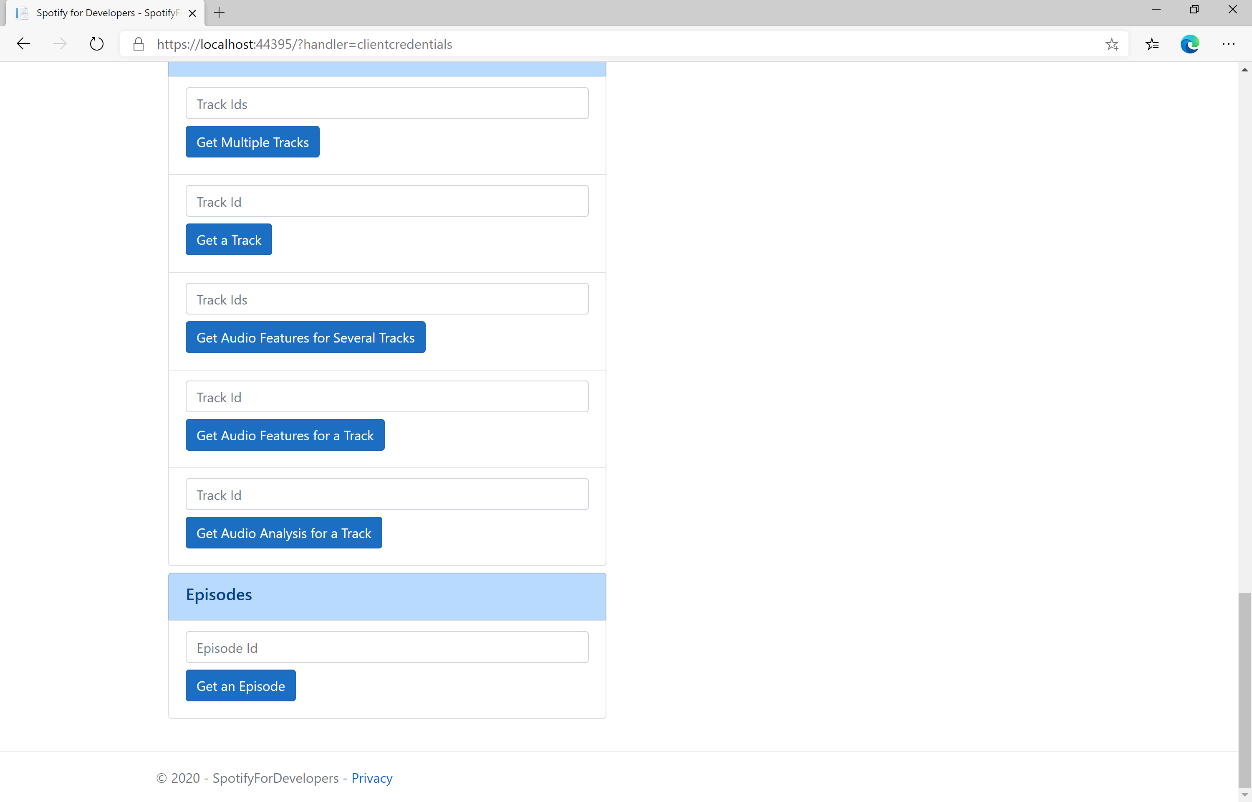
This form will **post** to the **method** for GetEpisode with the Value as **Episode Id** and will output to the **Results**.

### Step 8

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

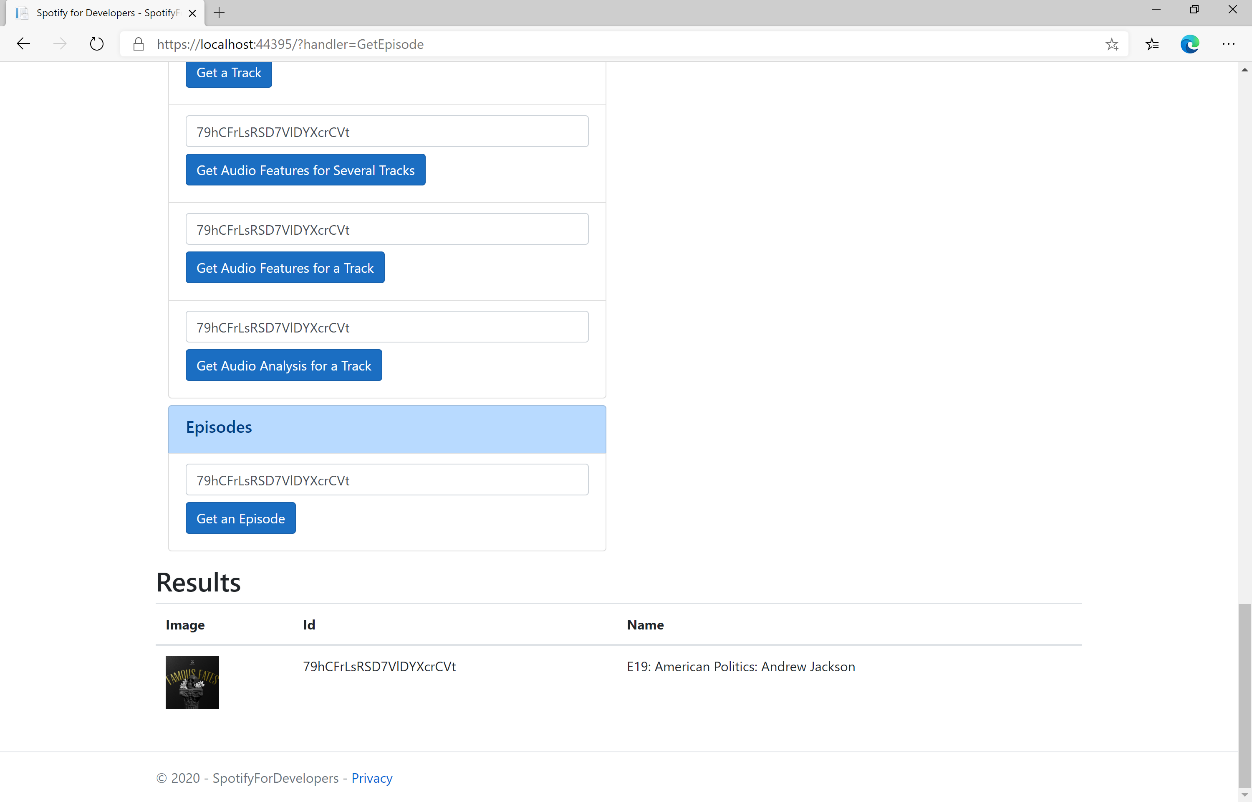
### Step 9

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



### Step 10

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



### 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 thispart of the workshop |

## Get Multiple Episodes

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

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

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

### Step 1

|  |  |
| --- | --- |
| A screenshot of a cell phone  Description automatically generated | 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** |
| A screenshot of a social media post  Description automatically generated | Then locate and select **SpotifyForDevelopers.sln** and select **Open** if you don’t have this file already then please follow the previous parts of the workshop including **Getting Started**, **Authorisation Guide**, **Search** **&** **Browse**, **Artists, Albums & Tracks** and **Episodes & Shows** |

### Step 2

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

### Step 3

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

### Step 4

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

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

The **method** for OnPostGetMultipleEpisodesAsync is used to get **episodes** by multiple **Episode Ids** on Spotify with the value and populate the **property** for Results accordingly.

### Step 5

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

### Step 6

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

### Step 7

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

|  |
| --- |
| <li class="list-group-item">  <form **asp-page-handler**="GetMultipleEpisodes" method="post">  <input **asp-for**="Value" placeholder="Episode Ids" class="form-control mb-2" />  <button class="btn btn-primary mb-2">  Get Multiple Episodes  </button>  </form>  </li> |

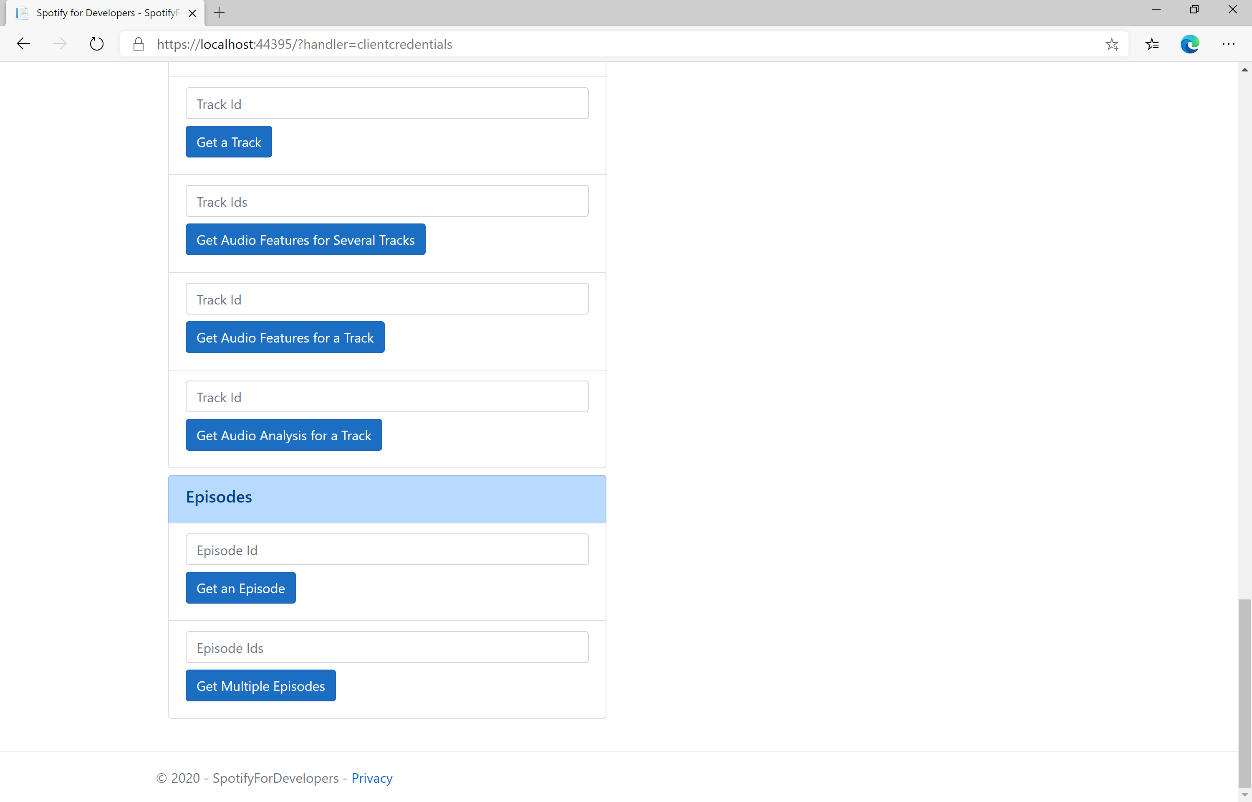
This form will **post** to the **method** for GetMultipleEpisodes with the Value as **Episode Ids** and will output to the **Results**.

### Step 8

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

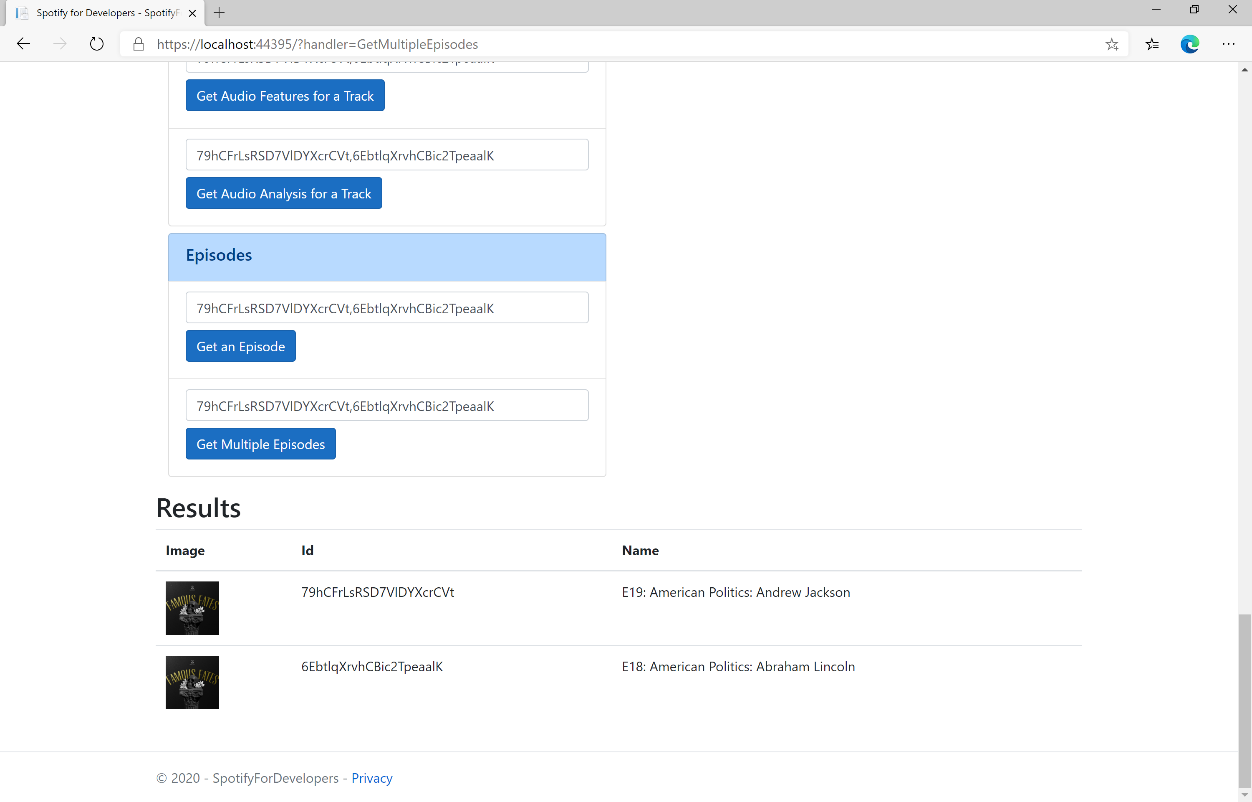
### Step 9

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



### Step 10

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



### 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 thispart of the workshop |

## Get a Show

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

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

|  |  |
| --- | --- |
| Header | Response |
| Success | |
| HTTP Status 200 OK | Show Object |
| Error | |
| Error Code | Error Object |

### Step 1

|  |  |
| --- | --- |
| A screenshot of a cell phone  Description automatically generated | 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** |
| A screenshot of a social media post  Description automatically generated | Then locate and select **SpotifyForDevelopers.sln** and select **Open** if you don’t have this file already then please follow the previous parts of the workshop including **Getting Started**, **Authorisation Guide**, **Search** **&** **Browse**, **Artists, Albums & Tracks** and **Episodes & Shows** |

### Step 2

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

### Step 3

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

### Step 4

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

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

The **method** for OnPostGetShowAsync is used to get a **show** by **Show Id** on Spotify with the value and populate the **property** for Results accordingly.

### Step 5

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

### Step 6

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

### Step 7

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

|  |
| --- |
| <ul class="list-group mb-2">  <li class="list-group-item list-group-item-primary">  <h5 class="list-group-item-heading">Shows</h5>  </li>  <li class="list-group-item">  <form **asp-page-handler**="GetShow" method="post">  <input **asp-for**="Value" placeholder="Show Id" class="form-control mb-2" />  <button class="btn btn-primary mb-2">  Get a Show  </button>  </form>  </li>  <!-- Shows -->  </ul> |

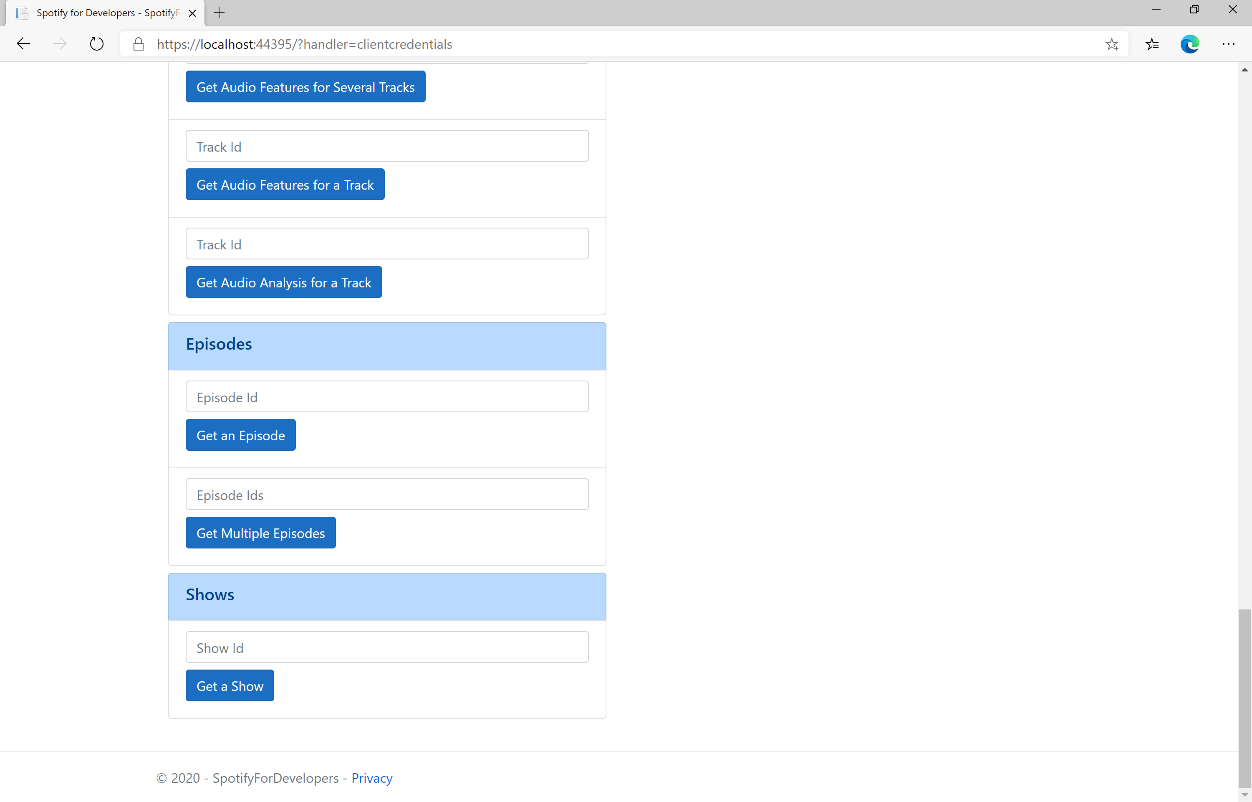
This form will **post** to the **method** for GetShow with the Value as **Show Id** and will output to the **Results**.

### Step 8

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

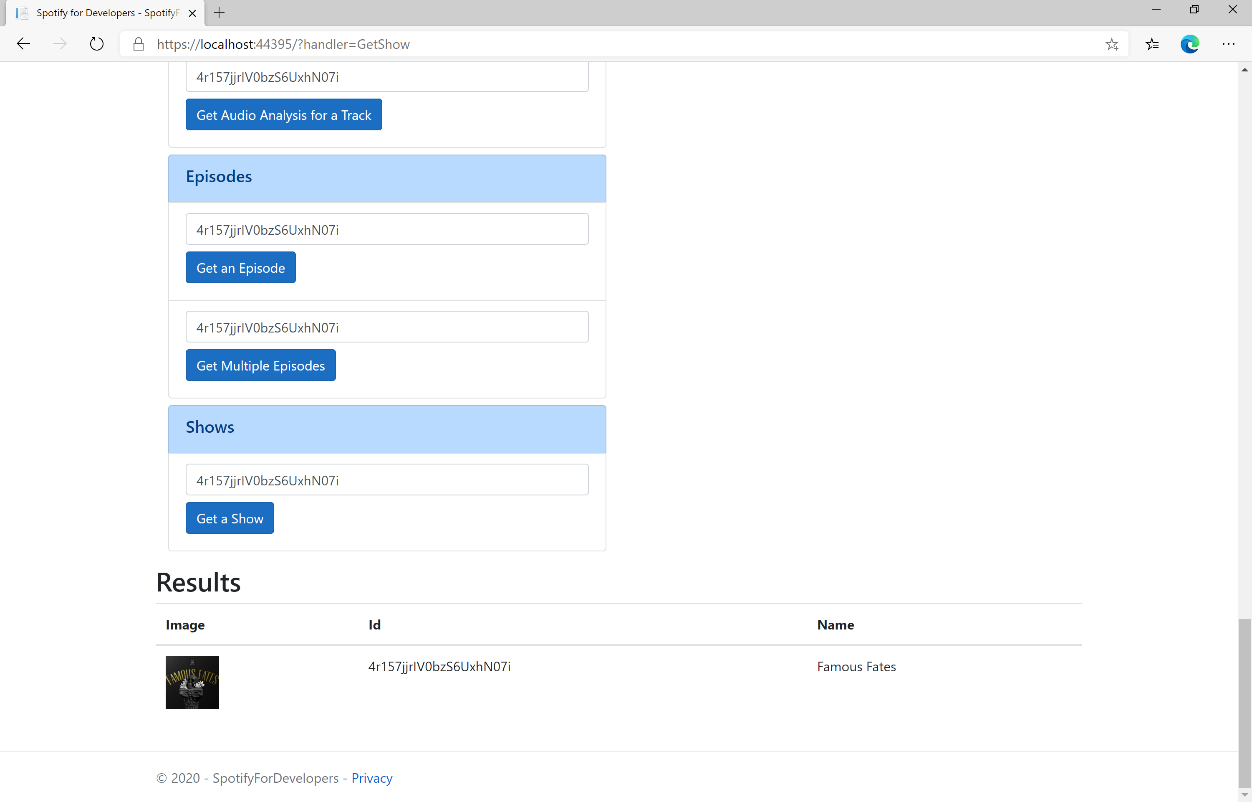
### Step 9

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



### Step 10

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



### 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 thispart of the workshop |

## Get Multiple Shows

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

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

|  |  |
| --- | --- |
| Header | Response |
| Success | |
| HTTP Status 200 OK | Shows as Array of Show Object |
| Error | |
| Error Code | Error Object |

### Step 1

|  |  |
| --- | --- |
| A screenshot of a cell phone  Description automatically generated | 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** |
| A screenshot of a social media post  Description automatically generated | Then locate and select **SpotifyForDevelopers.sln** and select **Open** if you don’t have this file already then please follow the previous parts of the workshop including **Getting Started**, **Authorisation Guide**, **Search** **&** **Browse**, **Artists, Albums & Tracks** and **Episodes & Shows** |

### Step 2

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

### Step 3

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

### Step 4

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

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

The **method** for OnPostGetMultipleShowsAsync is used to get **shows** by multiple **Show Ids** on Spotify with the value and populate the **property** for Results accordingly.

### Step 5

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

### Step 6

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

### Step 7

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

|  |
| --- |
| <li class="list-group-item">  <form **asp-page-handler**="GetMultipleShows" method="post">  <input **asp-for**="Value" placeholder="Show Ids" class="form-control mb-2" />  <button class="btn btn-primary mb-2">  Get Multiple Shows  </button>  </form>  </li> |

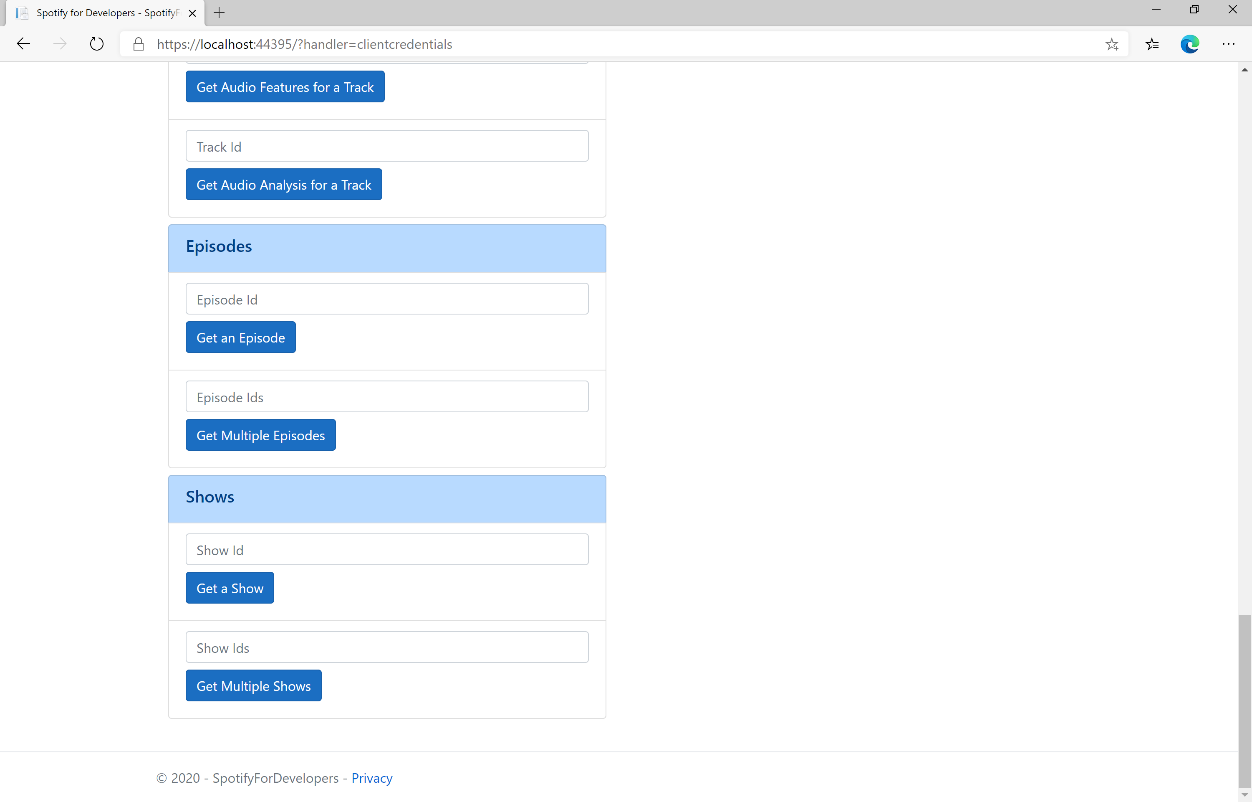
This form will **post** to the **method** for GetMultipleShows with the Value as **Show Ids** and will output to the **Results**.

### Step 8

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

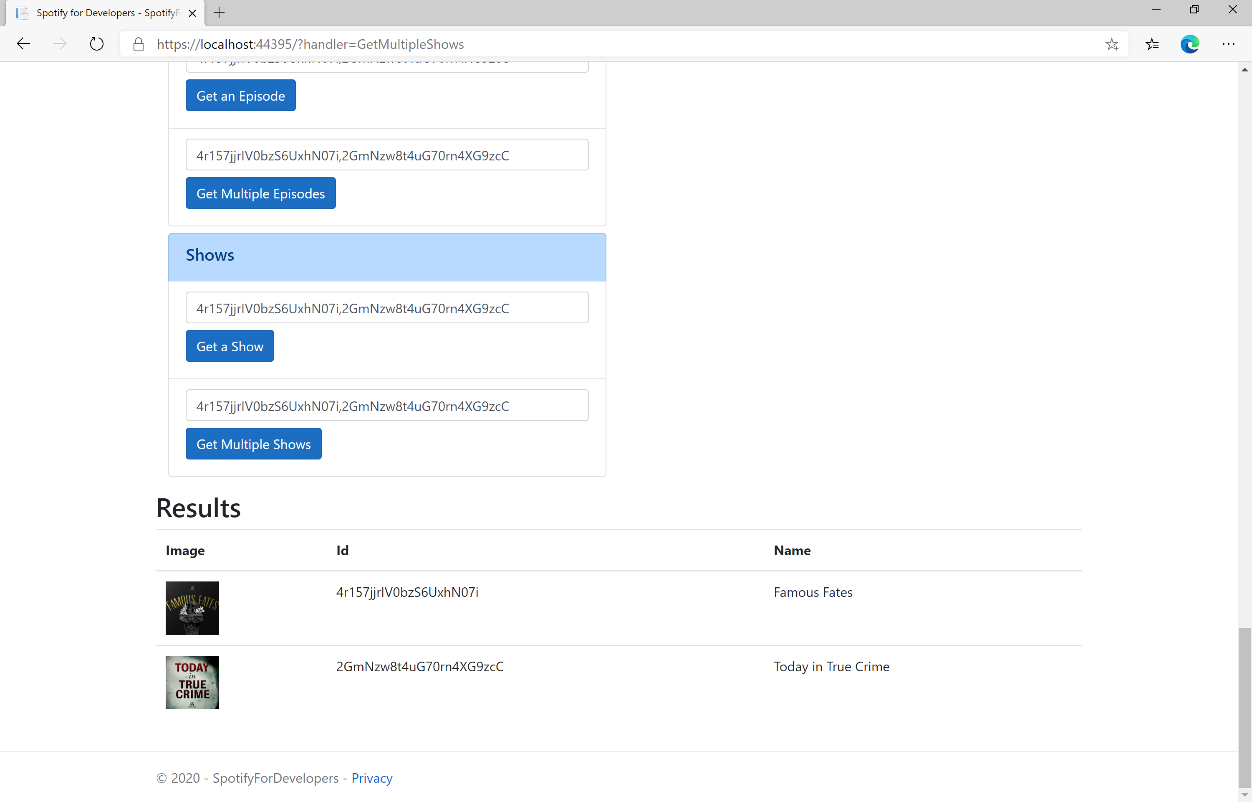
### Step 9

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



### Step 10

You can get some **Shows Ids** by using **Search for an Item** and pick **Shows** then into **Query** enter some **Podcast Names** then copy some **Shows Ids** to use and select **Get Multiple Shows** and scroll down to view **Results** like the following:



### 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 thispart of the workshop |

## Get a Show's Episodes

Get Spotify catalogue information about a show’s episodes. Optional parameters can be used to limit the number of episodes returned.

|  |  |
| --- | --- |
| GET https://api.spotify.com/v1/shows/{id}/episodes | |
| Header | |
| Authorization | Valid Access Token or User Token from Spotify Accounts service with **user-read-playback-position** scope to read user’s resume points on episode objects |
| Path Parameter | |
| id | The Spotify Id for the show |
| limit | The maximum number of episodes to return. Default: 20. Minimum: 1. Maximum: 50. |
| offset | The index of the first episode to return. Default: 0 (the first object). Use with limit to get the next set of episodes. |
| market | An ISO 3166-1 alpha-2 country code. If a country code is specified, only shows and episodes that are available in that market will be returned.  If a valid user access token is specified in the request header, the country associated with the user account will take priority over this parameter.  Note: If neither market nor user country are provided, the content is considered unavailable for the client.  Users can view the country that is associated with their account in the account settings. |

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

### Step 1

|  |  |
| --- | --- |
| A screenshot of a cell phone  Description automatically generated | 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** |
| A screenshot of a social media post  Description automatically generated | Then locate and select **SpotifyForDevelopers.sln** and select **Open** if you don’t have this file already then please follow the previous parts of the workshop including **Getting Started**, **Authorisation Guide**, **Search** **&** **Browse**, **Artists, Albums & Tracks** and **Episodes & Shows** |

### Step 2

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

### Step 3

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

### Step 4

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

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

The **method** for OnPostGetShowEpisodesAsync is used to get **episodes** by **Show Id** on Spotify with the value and populate the **property** for Results accordingly.

### Step 5

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

### Step 6

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

### Step 7

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

|  |
| --- |
| <li class="list-group-item">  <form **asp-page-handler**="GetShowEpisodes" method="post">  <input **asp-for**="Value" placeholder="Show Id" class="form-control mb-2" />  <button class="btn btn-primary mb-2">  Get a Show's Episodes  </button>  </form>  </li> |

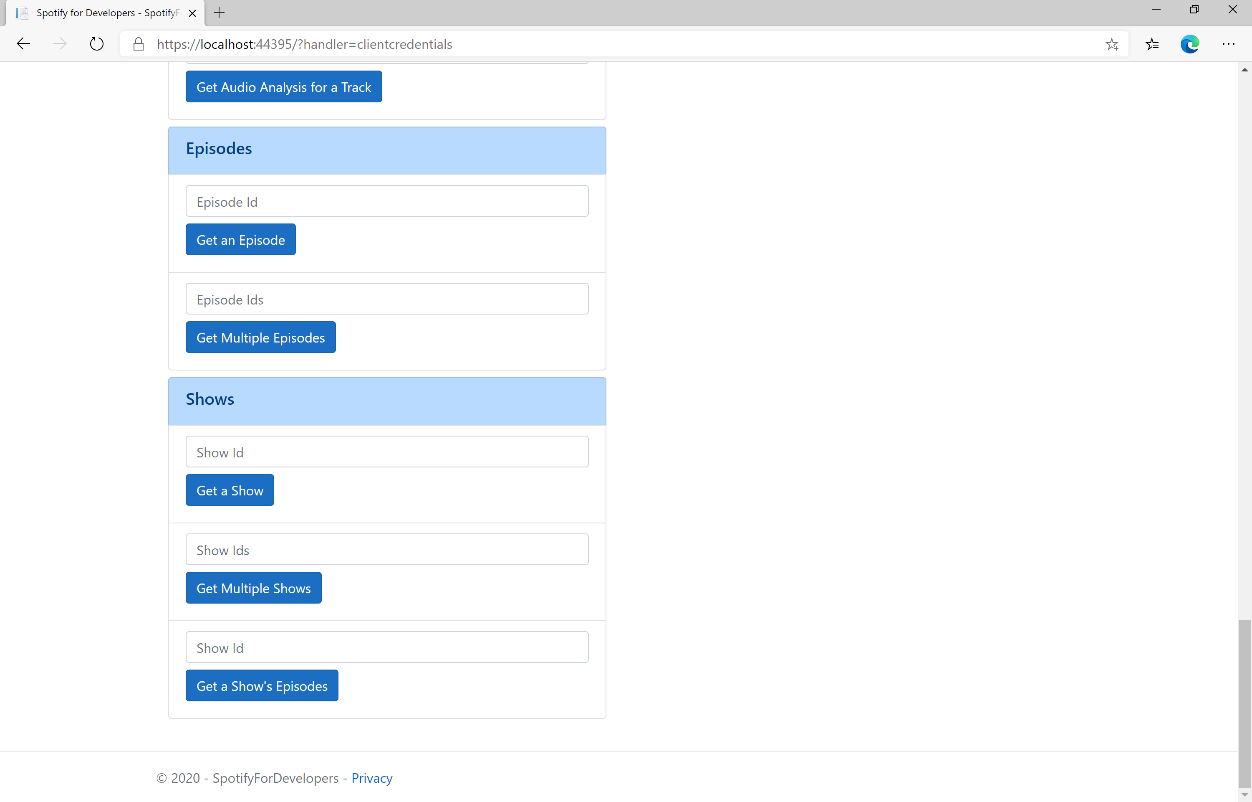
This form will **post** to the **method** for GetShowEpisodes with the Value as **Show Id** and will output to the **Results**.

### Step 8

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

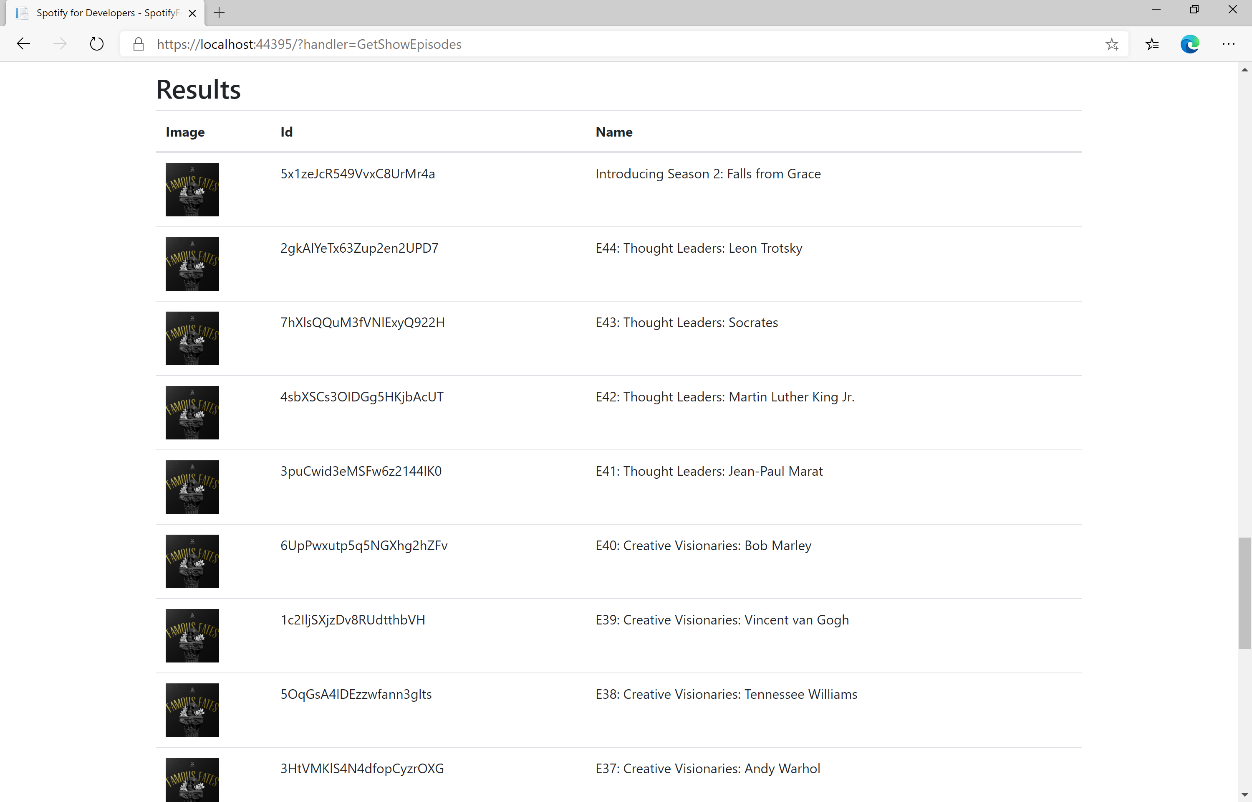
### Step 9

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



### Step 10

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



### 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 thispart of the workshop |