

▸ [Home](#)

· [Setup &](#)

[Authentication](#) · [Main](#)

Classes:

- [SpotifyService: Service Component](#)
- [SpotifyUIBase UI base class](#)
- [Event Manager](#)
- [Analysis: Spotify4Unity's Optional Logging](#)

---

· [Examples](#)

· [ExamplePlayerController:](#)

Simple Player control

· [ExamplePlaylistController:](#)

Displays all playlists

· [ExampleTracksControl:](#)

Displays all saved tracks

· [ExampleUserController:](#)

Displays all User Information

· [ExampleDevicesController:](#)

---

Displays all Spotify Devices

· [UI Theming](#)

---

## Home

## Welcome to the Spotify4Unity wiki!

Check out the full Wiki index for relevant pages. Also check the F.A.Q page for any problems you might have. **IMPORTANT NOTE; Users require Spotify Premium to be able to set and control their playback. If the user is not premium, you can still access their data like playlists, songs, profile info, etc.**

**For the latest information, to report bugs or issues, please use the GitHub repository [github.com/JoshLmao/Spotify4Unity](https://github.com/JoshLmao/Spotify4Unity)**

**Also, for more detailed Wiki, check here [github.com/JoshLmao/Spotify4Unity/wiki](https://github.com/JoshLmao/Spotify4Unity/wiki)**

## Quick Setup:

---

1. Add [SpotifyAPI-NET's latest release](#) to your Unity project (If you're not sure where, inside the "/Plugins/" folder)
  - Last checked working version (19/09/18): **v3.0.2**
  - If the latest release doesn't have the needed .dll files, clone the repo and create them
2. Add the Spotify4Unity UnityPackage into your project
3. Drag the "SpotifyService" script onto any Game Object in your scene
  - **Insert your client id, secret id & port number into the Inspector for the SpotifyService script**
  - Check out the [Setup & Authentication](#) page for how to get your ID's & port
4. Drag into the scene the "ExampleSmallPlayerControl.prefab" from inside "Assets/examples/all\_controls/prefabs/"

5. Press Play. A browser window will appear asking for you to log in with your Spotify credentials.
6. On return to the program, the player will update with your currently playing song

## Setup Helpful Pages

Also check out these useful pages relevant to the Quick Setup process

- [Setup And Authentication](#)
- [Spotify Service](#)
- [UI Prefabs:](#)
  - [ExamplePlayerController](#)

## Setup & Authentication Authentication

Before adding any components and scripts, you should create a Client ID needed to use the Web API side of the plugin. Visit <https://developer.spotify.com/dashboard/applications> and register an ID.

Once created, you only need to add "<http://localhost:{portNumber}>" to the "Redirect URIs" section of the settings, where {portNumber} is the port used on the "SpotifyService" component added later on

### EDIT SETTINGS

Application name

**Spotify4Unity Example Plugin**

Application description

Spotify API integration with Unity

Website

Add a website

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

Redirect URIs

<http://localhost:8000> Remove

<https://example.com/callback/>

ADD

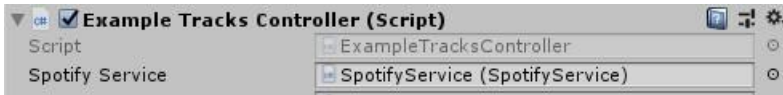
White-listed addresses to redirect to after authentication success OR failure (e.g. <http://mysite.com/callback/>)

## Unity Setup

After adding all the assets needed, the only requirement of these scripts is for a constant game object with the "SpotifyService" script attached

After that, you can create any UI and obtain all the information you need by inheriting any script from the "SpotifyUIBase" class.

Any script inheriting from the base will have a property asking for the "SpotifyService" script. If left blank, the base script will try to find the service script itself

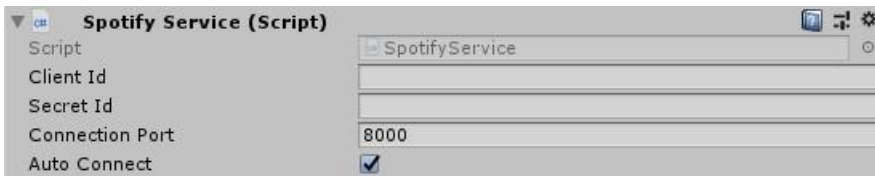


## SpotifyService: Service Component Purpose

The service is mainly used to talk between the SpotifyUIBase component & [JohnnyCrazy's SpotifyAPI-.NET](#). The service needs to be attached to **any game object and be within the scene** for the SpotifyUIBase to work.

You'll need to supply:

- Client ID - The client ID from the [Setup & Authentication](#) setup page
- Secret ID - The secret ID from the [Setup & Authentication](#) setup page
- Connection Port - The port number from the [Setup & Authentication](#) setup page
- Auto Connect - Should the service try to connect to the WebAPI on start?



## SpotifyUIBase UI base class Purpose

This UI Base Class is the main class to inherit from if you wish to create any UI component using Spotify features. It contains callbacks on the main thread which can be overridden for you to populate your own UI.

## Callbacks/Events

*ToDo: more information*

**OnPlayStatusChanged** - When the currently playing track's play status changes. Contains if the current track is playing or not

**OnTrackTimeChanged** - When the current track's position has changed (in s). Contains the current position (in s) and the total time (in s). Also contains TimeSpan options for both

**OnAlbumArtLoaded** - Contains a loaded sprite of the currently playing song's album art

**OnSavedTracksLoaded** - When all saved tracks have been loaded, a new one gets added or one gets removed

**OnPlaylistsChanged** - When a new playlist has been added or one has been removed

**OnUserInformationLoaded** - When the current user's profile information is loaded. Contains Username, DisplayName, if they are a premium user, how many followers, the UserId, the user's profile picture url, their country (as country code, eg UK), the user's birthday

**OnUserProfilePictureLoaded** - Contains a loaded sprite of the user's profile picture

**OnShuffleChanged** - When the shuffle state has been changed. Contains the new shuffle state

**OnRepeatChanged** - When repeat has been changed. Contains the new state

**OnMuteChanged** - When Spotify gets muted or unmuted. Note: this is different from when the volume is changed

**OnVolumeChanged** - When the volume of Spotify gets changed. Contains the current volume and the max volume

# Event Manager Purpose

To make the workflow easier, the plugin comes with an EventManager class, [by Ben D'Angelo](#). This helps move events from the SpotifyService to the main thread for you to add UI.

The EventManager class is mainly used inside [SpotifyUIBase](#) and comes with event classes to handle all available events from the service.

## Events

All events are inherited from GameEventBase which is required to pass through all events. The current events are:

- ConnectedChanged
- DevicesChanged
- MuteChanged
- PlaylistsChanged
- PlayStatusChanged
- RepeatChanged
- SavedTracksLoaded
- ShuffleChanged
- TrackChanged
- TrackTimeChanged
- UserInfoLoaded
- VolumeChanged

## Analysis: Spotify4Unity's Optional Logging Spotify4Unity's Optional Logging

### Purpose

---

This class contains optional logging information about the plugin. Like which song is currently playing, which song it changes to, if songs are skipped, etc.

### Include

---

To include the logs, add "S4U\_LOGS" to Unity's **Scripting Define Symbols** inside **Player Settings**

If you don't want these logs, simply remove "S4U\_LOGS" from the Scripting Define Symbols

## Examples

Check out the example UI controls. All these are examples of all the code you should need to write yourself to make the UI work

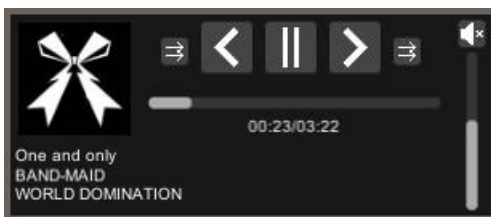
- [ExamplePlayerController](#)
- [ExamplePlaylistController](#)
- [ExampleTracksController](#)
- [ExampleUserController](#)
- [ExampleDevicesController](#)

## ExamplePlayerController: Purpose

The ExamplePlayerController shows how you can create a simple player using the SpotifyUIBase. You can also use this control and change the prefab to however you like.

The control has all the functionality you expect:

- Play/Pause
- Skip
- Previous
- Volume control
- Mute/unmute
- Current track & Total track time
- Setting track position
- Current song title, album name and artist name
- Current album art
- Toggle shuffle
- Toggle repeat

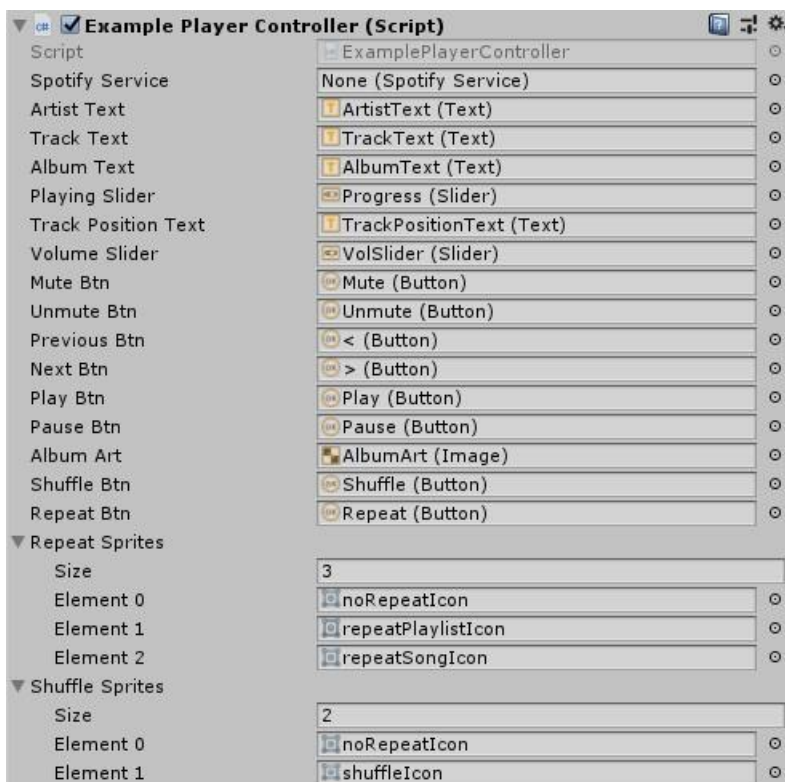


## Customization

The script can be used on any prefab, however the script requires a few things to function and work. **The script doesn't require you to have all of these, if you want only some of the functionality.**

- Artist Text - Text to show the current artist
- Track Text - Text to show the current track
- Album Text - Text to show the current album
- Playing Slider - Slider used to show the current song's progress. Can allow the user to change track position using the slider
- Track Position Text - Text to show the current position as "{currentPositionMinutes}:{CurrentPositionSeconds}/{totalTimeMinutes}:{totalTimeSeconds}"

- Volume Slider - Slider used to show and change the current volume of Spotify
- MuteBtn - Button to mute the volume on clicked
- Unmute Btn - Button to unmute the volume on clicked
- Previous Btn - Change the current track to the previous
- Next Btn - Change the current track to the next in queue
- Play Btn - Play the current track, if paused. Will be hidden if track is playing
- Pause Btn - Pauses the current track, if playing. Will be hidden if track is paused
- Album Art - Image of the current playing track's album art
- Shuffle Btn - Button to change the current shuffle state. **REQUIRES: A child object with an Image component for setting the current state**
- Repeat Btn - Button to change the current repeat state. **REQUIRES: A child object with an Image component for setting the current state**
- Repeat Sprites - An array of sprites that will be shown in the Repeat Btn. Make sure the length is the same amount of states inside the Repeat enumerator.
- inside the Shuffle enumerator.



Shuffle Sprites - An array of sprites that will be

shown in the Shuffle Btn. Make sure the length is the same amount of states

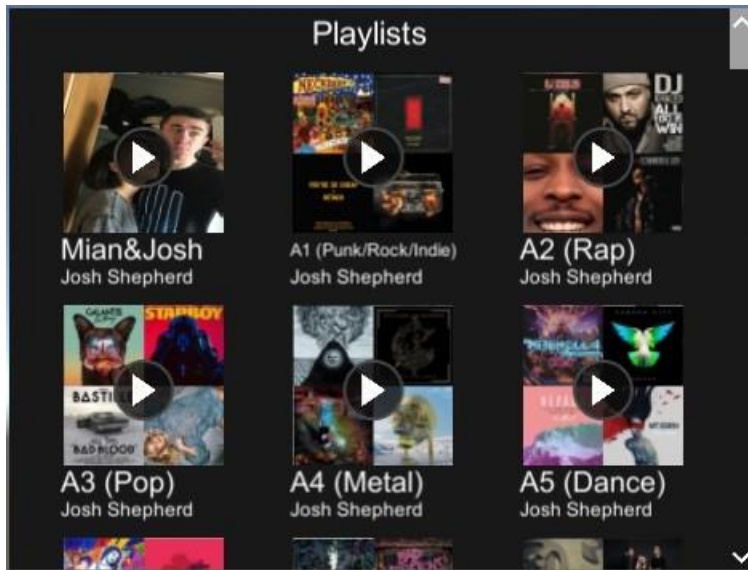
## ExamplePlaylistController:

### Purpose

The ExamplePlaylistController display all playlists the user is following or has created.

The control can:

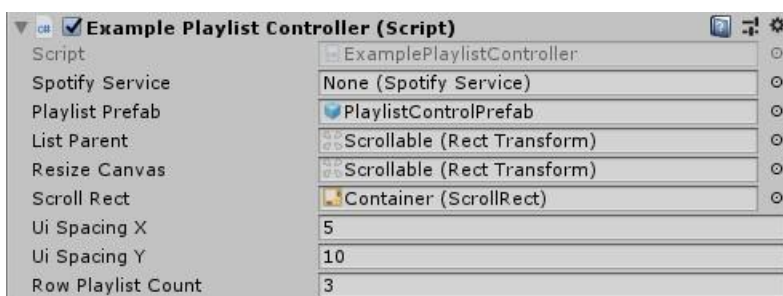
- Display playlist name & playlist creator
- Play a playlist



## Customization

Like all scripts, this can be customisation or you can create your own and use this script. The elements in the inspector are what can be currently shown. **The script doesn't require you to have all of these, if you want only some of the functionality.**

- Playlist Prefab - A square prefab. **REQUIRED: An image, playlist name text, playlist creator text and a button to play**
- List Parent - The parent to hold all of the instantiated prefabs
- Resize Canvas - The canvas that will be resized to hold all prefabs. Usually the same as the "List Parent" *//May change to remove this element and use the ListParent in the future*
- Scroll Rect - The rect that holds the parent canvas for all elements.
- Ui Spacing X - The amount of pixels between each prefab element in a row
- Ui Spacing Y - The amount of pixels between each list's row
- Row Playlist Count - The amount of playlist prefabs on a single row



## ExampleTracksControl:

### Purpose

The ExampleTracksControl displays all the saved tracks the user has in Spotify.

The control can:

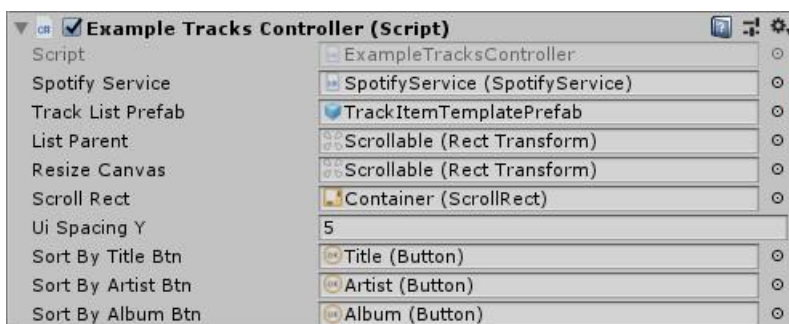
- Play any track using the ">" button
- Sort the tracks by "Title", "Artist" & "Album" (Noted by the up or down arrow next to the top column names)

Saved Tracks				
Title	Artist	Album		
New World ft. Vava	Krewella, Yellow Claw, Vava	New World ft. Vava	+	▶
PONPONPON	Kyary Pamyu Pamyu	KPP BEST	+	▶
Soredemoaruteru	Keyakizaka46	Kazenifukaretemo (Special Edition)	+	▶
I Have a Problem	Beartooth	Sick	+	▶
Disease	Beartooth	Disease / Bad Listene	+	▶
Manipulation	Beartooth	Manipulation	+	▶
Can't Help Falling in Love	Kina Grannis	Crazy Rich Asians (Original Motion	+	▶
Yellow	Katherine Ho	Crazy Rich Asians (Original Motion	+	▶
I Love It (& Lil Pump)	Kanye West, Lil Pump	I Love It	+	▶
Busy Eamin'	Jungle	Jungle	+	▶
Time	Jungle	Jungle	+	▶
キズナミ	Kyary Pamyu Pamyu	キズナミ	+	▶
Equal Rights	The Lonely Island, Pink	Popstar: Never Stop Never Stopping	+	▶
Finest Girl (Bin Laden Song)	The Lonely Island	Popstar: Never Stop Never Stopping	+	▶
The Lines	Beartooth	Disgusting (Deluxe Edition)	+	▶
I Have a Problem	Beartooth	Disgusting (Deluxe Edition)	+	▶
In Between	Beartooth	Disgusting (Deluxe Edition)	+	▶
Relapsing	Beartooth	Disgusting (Deluxe Edition)	+	▶
SLOW DANCING IN THE DARK	Joji	SLOW DANCING IN THE DARK	+	▶
It Goes Through	Chester Bennington, Dave Farrell, Joe	MALL (Music From The Motion Picture)	+	▶
White Noise	Chester Bennington, Dave Farrell, Joe	MALL (Music From The Motion Picture)	+	▶
The Last Line	Chester Bennington, Dave Farrell, Joe	MALL (Music From The Motion Picture)	+	▶

## Customization

Like all scripts, this can be customisation or you can create your own and use this script. The elements in the inspector are what can be currently shown. The script doesn't require you to have all of these, if you want only some of the functionality.

- Track List Prefab - Prefab of one element in the list
- List Parent - The parent canvas that will contain all prefabs
- Resize Canvas - The canvas that will be resized to hold all prefabs. Usually the same as the "List Parent" *//May change to remove this element and use the ListParent in the future*
- Scroll Rect - The rect that holds the parent canvas for all elements.
- Ui Spacing Y - The amount of pixels inbetween each element in the list
- Sort By Title Btn - Button to sort by song title. **REQUIRES: Child game object called "Icon" with an Image component**
- Sort By Artist Btn - Button to sort by song artist. **REQUIRES: Child game object called "Icon" with an Image component**
- **component**



album. **REQUIRES: Child game object called "Icon" with an Image**



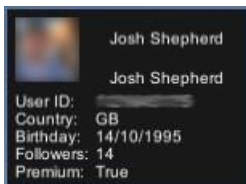
# ExampleUserController:

## Purpose:

The ExampleUserController is an example of all the currently available user information you are able to obtain through the SpotifyService.

This control displays all the information below:

- Username
- Display Name
- User Id
- Country
- Birthday
- The amount of followers the user has
- If the user is premium
- The user's profile picture



## Customization

Like all scripts, this can be customised or you can create your own and use this script. The elements in the inspector are what can be currently shown. The script doesn't require you to have all of these, if you want only some of the functionality.

- Username - Text to display the current user's username
- Display Name - Text to display the current user's display name
- UserId - Text to display the current user's User Id. An internal ID used by Spotify
- Country - Text to display the current user's Country as a country code. Eg, GB for Great Britain.
- Birthday - Text to display the current user's Birthday (as Day/Month/Year)
- Followers Count - text to display how many followers the current user has
- Is Premium - Text to display true or false if the user is premium *//Redundant. Mainly used in example prefab*

□

# ExampleDevicesController:

## Purpose

The ExampleDevicesController is an example of how users can change their current active Spotify playback device

The control can:

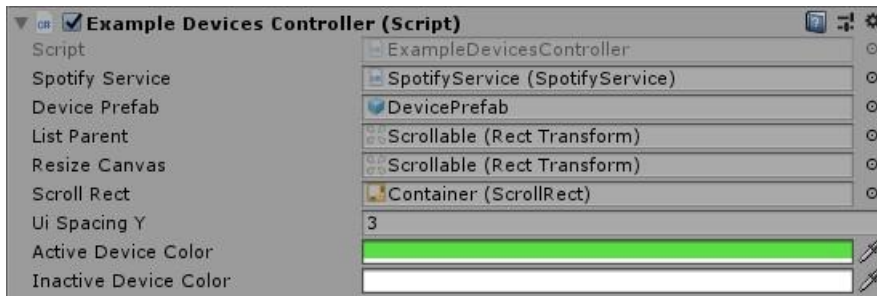
- Display all available Spotify devices in alphabetical order, showing the active one first
- Can change the active Spotify device

□

# Customization

The script can be used on any prefab, however the script requires a few things to function and work. **The script doesn't require you to have all of these, if you want only some of the functionality.**

- Device Prefab - Prefab of one element to represent the device
- Resize Canvas - The canvas that will be resized to hold all prefabs. Usually the same as the "List Parent" *//May change to remove this element and use the ListParent in the future*
- Scroll Rect - The rect that holds the parent canvas for all elements.
- Ui Spacing Y - The amount of pixels between each list's row
- Active Device Color - The active device text & icon foreground color
- Inactive Device Color - The foreground color of the other inactive Spotify devices



## UI Theming Colors

If you want to read more about Spotify's branding guidelines, check out [their post on their website](#). This post contains information about how the [example prefabs](#) are currently themed

## Materials

All prefabs colors are tied to materials to make customising the prefabs easy. All the materials are located inside "examples/all\_controls/materials/"

### General

- **uiAccentColor** - Spotify's main accent color **uiBackgroundColor** - Spotify's main background color,
- same color as Spotify UI (Used on all background)
- **uiBackgroundAccentColor** - Spotify's alternate color for the background, used to stand out against normal background color • **uiFontColor** - Spotify's main font color (Used with all text & icons)

### Buttons

- **btnBackgroundColor** - Main background color tint for all buttons **btnFontColor**-
- Main foreground color for contents of buttons

### ScrollView

- **scrollviewBackgroundColor\*** - Main background color of scroll view, same color as Spotify UI **scrollviewSlideColor**
- - Color of the handle in scroll view, same color as Spotify UI

## Slider

- sliderBackgroundColor - Back color of the slider component sliderForegroundColor - Color of the
- slider which shows the completed section, **without mouse over** sliderForegroundColor - Color of
- the slider which shows the completed section, **with mouse over** sliderHandleColor - Color of the handle
- used to drag the slider