Twitch Chat Connect - Documentation v1.1.0

Table of contents

Main feature

Twitch Chat Connect is a client that can be used to connect to any Twitch chat using IRC protocol. It has an easy usage by using callbacks to know whenever a chat message is sent.

Setup

OAuth Token

The first thing to do is to get an OAuth token to be able to connect to the Twitch Chat. In order to do that, go to https://twitchapps.com/tmi/ and log-in with your username/password of Twitch and then you will get your token. It's recommended to use a secondary account.

Configuration file

Now it's necessary to create a config.json file with the following structure:

```
{
        "username": "<your_username>",
        "userToken": "<oauth:your_token_here>",
        "channelName": "<your_channel_name>"
}
```

username: Username which was used to generate the OAuth token.
userToken: OAuth token generated in the previous step.
channelName: Name of the channel you want to connect to.

Dependencies

The following dependencies are necessary. It's possible to download them in Unity's Asset Store.
- JSON .NET For Unity

Usage

Adding the client

Add an empty object in your scene and add the component **TwitchChatClient**. The component is a singleton and also it's not destroyed when loading a new

scene. It's only necessary to add it once in your whole game. It could be added in multiple scenes without any problem.

The component has the following fields that can be modified in the *Inspector*.

Name	Description	Default value
Configuration Path	Configuration path where the config.json file is located.	Application.persistentDataPath
Command Prefix	All messages starting with this prefix will be recognized as commands. All other messages will be ignored.	**!**

Available methods

TwitchChatClient.instance.Init

This method is necessary to connect to Twitch's chat. It receives two parameters which are callbacks **onSuccess()** and **onError(string)**.

Receiving messages

TwitchChatClient.instance.onChatMessageReceived

In order to receive messages, It's necessary to subscribe to *onChatMessageReceived* event. The method used as the event subscription needs to receive a parameter of type *TwitchChatMessage*

TwitchChatMessage is a POCO class that contains the following public attributes.

Name	Description	Type
sender	Username who sent the message.	string
command	Command sent with the prefix included.	string
parameters	Array of strings as result of splitting the message with a space, excluding the command.	string[]

Example

```
public class Example : MonoBehaviour
{
    void Start()
    {
        TwitchChatClient.instance.Init(() =>
        {
            TwitchChatClient.instance.onChatMessageReceived += ShowMessage;
        }, message =>
        {
            Debug.LogError(message);
        });
    }

    void ShowMessage(TwitchChatMessage chatMessage)
    {
        // chatMessage contains all the necessary data of the message.
    }
}
```

For a full working example in Unity, open the scene **SampleScene**
in the directory *Example*.