

# Twitch Chat Connect - Documentation v1.2.0

## Table of contents

<b>Table of contents</b>	<b>1</b>
<b>Main feature</b>	<b>2</b>
<b>Setup</b>	<b>2</b>
OAuth Token	2
Configuration file	2
Dependencies	2
<b>Usage</b>	<b>2</b>
Adding the client	2
Available classes & methods	3
Receiving messages	3
<b>Example</b>	<b>5</b>

# 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
<b>Configuration Path</b>	Configuration path where the config.json file is located.	<a href="#">Application.persistentDataPath</a>
<b>Command Prefix</b>	All messages starting with this prefix will be recognized as commands. All other messages will be ignored.	!

## Available classes & methods

### ***TwitchChatClient.instance.Init(OnSuccess onSuccess, OnError onError)***

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

### ***TwitchUser TwitchUserManager.GetUser(string username)***

Receives an username and returns a ***TwitchUser*** with user information.

### ***bool TwitchUserManager.HasUser(string username)***

Receives an username and returns *true/false* if the user is in the chat.

### ***List<TwitchUser> TwitchUserManager.Users***

Returns a list of all connected users to the chat.

## Receiving messages

In order to receive messages, It's necessary to subscribe to different events.

### ***TwitchChatClient.instance.onChatCommandReceived(TwitchChatCommand)***

This is an event that will be triggered when a chat user sends a message started by the *command prefix* defined in the component ***TwitchChatClient***.

### ***TwitchChatClient.instance.onChatRewardReceived(TwitchChatReward)***

This is an event that will be triggered when a chat user unlocks a reward.

### ***TwitchChatClient.instance.onChatMessageReceived(TwitchChatMessage)***

This is an event that will be triggered when a chat user sends a normal message.

***TwitchChatMessage*** is a [POCO](#) class that contains twitch chat message information.

Name	Description	Type
<b>User</b>	User who sent the message.	<b><i>TwitchUser</i></b>
<b>Message</b>	Message sent	string

***TwitchChatCommand*** extends ***TwitchChatMessage*** and processes the message to provide information about the *command* and the *parameters* used.

Name	Description	Type
<b>command</b>	Command sent with the prefix included.	string
<b>parameters</b>	Array of strings as result of splitting the message with a space, excluding the command.	string[ ]

***TwitchChatReward*** extends ***TwitchChatMessage*** and adds the custom reward id.

Name	Description	Type
<b>CustomRewardId</b>	Custom reward id	string

***TwitchUser*** is a [POCO](#) class that contains the twitch user information and it has the following public attributes. The **Username** is always available but the rest of the data is only available if the user sent at least one message.

Name	Description	Type
<b>Username</b>	Twitch username.	string
<b>Id</b>	Twitch user ID	string
<b>IsSub</b>	Twitch subscription status	bool
<b>DisplayName</b>	Twitch displayname (returns <b>username</b> as default)	string

## Example

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

```
public class Example : MonoBehaviour
{
    void Start()
    {
        TwitchChatClient.instance.Init(() =>
        {
            TwitchChatClient.instance.onChatMessageReceived += ShowMessage;
            TwitchChatClient.instance.onChatCommandReceived += ShowCommand;
            TwitchChatClient.instance.onChatRewardReceived += ShowReward;

        }, message =>
        {
            Debug.LogError(message);
        });
    }

    void ShowCommand(TwitchChatCommand chatCommand)
    {
    }

    void ShowReward(TwitchChatReward chatReward)
    {
    }

    void ShowMessage(TwitchChatMessage chatMessage)
    {
    }
}
```