

RoboSparkles Chatbot Documentation

Byrix

December 27, 2020

Contents

1	Introduction	2
2	Using this Bot	2
2.1	Installing	2
2.1.1	Streamlabs Chatbot	2
2.1.2	Scripts	3
2.2	Uninstalling	3
2.2.1	Uninstall Scripts	3
2.2.2	Uninstall Chatbot	3
3	Scripts	3
3.1	Shout-out Command	3
3.1.1	Use	4
3.1.2	Settings	4
3.1.3	Config Files	4
3.2	Wholesome Vibes	5
3.2.1	Use	5
3.2.2	Settings	5
3.2.3	Config Files	6
3.3	Quotes System	6
3.3.1	Use	6
3.3.2	Settings	6
3.4	Link Protection	7
3.4.1	Use	7
3.4.2	Settings	7

1 Introduction

RoboSparkles is a chatbot developed for Twitch streamer BioSparkles. It's written in Python 2.7 and utilises the in-built scripting functionality of Streamlabs Chatbot. For more information on Streamlabs Chatbot please consult their personal documentation [here](#). It is an important note that because this uses Streamlabs Chatbot the limitations of Streamlabs Chatbot apply, most notably: this is only possible on a Window device, the Streamlabs Chatbot application must be running and connected for your chatbot to function, and everything is locally hosted - cloud used for backups only.

A public version of all scripts is available on GitHub [here](#), for information on what the different scripts do and how they work please see Scripts section. Please note, while you are free to use and change any code taken from the GitHub repository for your personal use I ask that you give credit back to my original work.

2 Using this Bot

The following section will outline the adding and removing of these scripts for use on your channel. If you are using this bot I ask that you give credit to me in your about section

2.1 Installing

2.1.1 Streamlabs Chatbot

These scripts run-off the Streamlabs Chatbot (known as "Chatbot" from here on) in-built scripting features, therefore you must download and use Chatbot to be able to use them. This document will only outline the general idea of what steps are required for you to complete for the scripts to function, but will not go into specific detail of how to do it. For full detailed information of installing and setting up Chatbot please consult the official Streamlabs documentation found [here](#).

1. Complete the basic setup of the Chatbot
Download and install the Chatbot, complete the initial setup of connecting your streamer and bot accounts.
2. Install Python 2.7
To run the scripts Chatbot uses Python 2.7, so you will need to download [Python 2.7.13](#) from the Python website [here](#).
3. Navigate to the "Scripts" tab
Please note the "Scripts" tab will only appear on the list on the left if both the streamer and bot accounts are actively connected to Chatbot. To connect them click the person icon in the lower-left and press the "Connect" button in both the "Twitch Bot" and "Twitch Streamer". In the Chatbot settings - accessed by clicking the gear icon in the lower-left - you can set both accounts to connect automatically on launch
4. Configure Chatbot for Python
In the scripts tab click on the gear icon found in the top-right, here you will see a textbox asking for your Python 2.7.13 directory, simply paste the location of the "Lib" folder within your Python installation. By default this will be "C:\Python27\Lib"

2.1.2 Scripts

This section will walk you through the installation of new scripts into Chatbot.

1. Download the scripts
Download the scripts from the public GitHub repository found [here](#). They should download in a .zip file
2. Install the scripts
Navigate to the Chatbot scripts folder (default: "C:\Users\USER\AppData\Roaming\Streamlabs\Streamlabs Chatbot\Services\Scripts") and copy the download zip folder in. Extract the contents of the zip file into this folder
3. Refresh Streamlabs Chatbot
If the Chatbot is currently running: navigate to the Scripts tab from the left-hand side menu and click the refresh button in the top-right to update the scripts.
If the chatbot is not running, then simply opening it will refresh the scripts
In either case, you should see all the scripts listed now.

2.2 Uninstalling

2.2.1 Uninstall Scripts

To simply stop a script running on your chatbot, navigate to the Scripts tab in the Streamlabs Chatbot interface and uncheck the tick box of the desired script to disable it.

If you wish to remove a script entirely, navigate to the Scripts folder of the Chatbot files (default: "C:\Users\USER\AppData\Roaming\Streamlabs\Streamlabs Chatbot\Services\Scripts") and delete the folder corresponding to the script you wish to remove.

2.2.2 Uninstall Chatbot

To uninstall the entire chatbot simply follow the relevant instructions for removing applications from your computer for your operating system.

3 Scripts

In this section you will find an outline of all currently available scripts and how to use them. All instructions below assume the default permissions and commands are in use, if you change any of these settings please remember to use the updated versions

3.1 Shout-out Command

You can use the shout-out command (default: !so) to have your bot give shout-outs in the chat. This system supports the use of personalised shout-outs for your friends; you can specify various users and define a shout-out response specifically for them which is automatically detected and used by the bot.

Credit to Reecon for much of the inspiration and systems used in this script. Check out their original work over on Github [here](#).

3.1.1 Use

To use this command mods, editors, and the broadcaster can type `!so @target` or `!so target` where 'target' is the username of the person you wish to shout-out.

3.1.2 Settings

Clicking on the script in the 'Scripts' tab of Streamlabs Chatbot will bring up the settings panel for this system. Here you will find the various options you can change to configure this script to your liking

- **Command:** Defines the command used to active a shout-out, defaults to `!so`
- **Alternative Command:** Allows you to specify alternate commands to trigger a shout-out. For example, if you added `!s` to this list then you could create a shout-out by using `!so target` *or* `!s target` to the same effect. To specify multiple commands separate them with a space, e.g. `"!s !shoutout"`
- **Cooldown (second):** Defines the cooldown time between calls of the command
- **Permission:** The minimum permission levels required to use the command - selecting 'moderator' means that anyone with moderator *or above* can use this command (editors and broadcaster)
One option for permission is 'user_specific' which allows you to define the specific users who are able to use this command. Anyone specified in this way will be able to use the command irrespective of their roles on your channel. Use the 'Allowed Users' section to provide the list of users (more below)
- **Shoutout Prefix:** A prefix for all shout-outs to start with, irregardless of if they're personalised or default
- **Shoutout Suffix:** A suffix for all shout-outs to end with, irregardless of if they're personalised or default
- **Auto-Shoutout Raiders:** If selected, a shout-out will automatically be given to any incoming raiders after a slight delay
- **Delay on raid auto-shoutout:** How long to wait after a raid before shouting-out the raider - added as a way of avoided the shout-out being lost in raid messages
- **Allowed Users:** A list of users allowed to use this command. Only use this is you're using the "user_specific" permission level. To specify multiple users separate their names with spaces: e.g. `"userOne userTwo userThree"` allows userOne, userTwo and userThree to use the command
- **Open SO Config:** Opens the config file that allows you to specify the personalised shout-out commands, for more information see the Config section below

3.1.3 Config Files

In the settings panel you will also find a button to open the shout-out config file which allows you to define any personalised shout-outs as well as edit your default. The first few lines will define how to format your shout-outs as well as any variables you might want to use.

The required format for shout-outs is: username "shoutout". You can use the provided default shout-out as a template to define your own

The currently supported variables are:

- `$targetname`: will substitute itself for the name of the targeted user
- `$url`: will replace itself with the URL for the target's Twitch profile: "twitch.tv/targetName"

3.2 Wholesome Vibes

This script gives your bot two new commands to spread wholesome vibes in your chat. Users can use either the hug command (default: `!hug`) to have the bot print a hug message in chat, or use the compliment command (default: `!compliment`) to compliment a fellow user.

3.2.1 Use

Any user in chat can use `!hug target` or `!hug @target` to have a randomised hug message printed in chat, where the user who used the command hugs the target. The list of the hug messages that can be printed are specified by a file set by you

Any user in chat can use `!compliment target` or `!compliment @target` to have a random compliment printed in chat, targeted at the user 'target'. The compliments are pulled from the Complimentr API (found [here](#)) developed by grant.codes.

3.2.2 Settings

Clicking on the script in the 'Scripts' tab of Streamlabs Chatbot will bring up the settings panel for this system. Here you will find the various options you can change to configure this script to your liking

- **Compliment Command:** Defines the command used to active a compliment, defaults to `!compliment`
- **Cooldown (seconds):** Defines the cooldown time between calls of either command
- **Hug Command:** Defines the command used to activate a hug, defaults to `!hug`
- **Permission:** The minimum permission levels required to use the command - selecting 'moderator' means that anyone with moderator *or above* can use this command (editors and broadcaster)
One option for permission is 'user_specific' which allows you to define the specific users who are able to use this command. Anyone specified in this way will be able to use the command irrespective of their roles on your channel. Use the 'Allowed Users' section to provide the list of users (more below)
- **Allowed Users:** A list of users allowed to use this command. Only use this is you're using the "user_specific" permission level. To specify multiple users separate their names with spaces: e.g. "userOne userTwo userThree" allows userOne, userTwo and userThree to use the command
- **Edit Hug File:** Opens the config file that allows you to specify the list of hug responses. For more information see the Config section below

3.2.3 Config Files

In the settings panel you will also find a button to open the hugs file which allows you to define the list of hug responses.

The currently supported variables are:

- `$username`: will substitute itself with the name of the user calling the command
- `$targetname`: will substitute itself with the name of the targeted user

3.3 Quotes System

This script allows users to immortalise the things said on your stream. Users can add quotes and the system will record the quote, game you're currently playing, and the date. At any time then users can call up quotes again.

3.3.1 Use

- Subscribers and above can add a new command using `!addquote quote` where 'quote' is the quote you wish to add
- Users can call up a quote using `!quote num` or `!quote phrase`. If a number is given, then that number quote will be pulled up. If a phrase is given, then a random quote containing that phrase will be pulled up
- Moderators and above can edit and remove existing quotes using `!editquote num newQuote` where 'newQuote' is the updated version of the quote to be stored in quote #'num'; or `!removequote num` to delete quote #'num'.

3.3.2 Settings

- **Add Quote Command**: Defines the command used to add a new quote
- **Add Cooldown**: Defines the cooldown time between calls of the add quote command
- **Add Permission**: The minimum permission levels required to use the add quote command - selecting 'subscriber' means that anyone with moderator *or above* can use this command (moderators, editors, and broadcaster).
- **Call Command**: Defines the command used to call up a quote
- **Call Cooldown**: Defines the cooldown time between calls of the quote command
- **Call Permission**: The minimum permission levels required to use the quote command - selecting 'moderator' means that anyone with moderator *or above* can use this command (editors and broadcaster).
- **Show Date?**: If enabled, called up quotes will display the date that the quote as originally added
- **Show Game?**: If enabled, called up quotes will display the game that was being played when the quote was added
- **Edit Quote Command**: Defines the command used to edit a quote

- **Edit Cooldown:** Defines the cooldown time between calls of the edit quote command
- **Edit Permission:** The minimum permission levels required to use the edit quote command - selecting 'moderator' means that anyone with moderator *or above* can use this command (editors and broadcaster).
- **Remove Quote Command:** Defines the command used to remove a quote

3.4 Link Protection

Blocks links being posted in your chat - useful for stopping bots or trolls. Added functionality to allow links to be posted from trusted users.

3.4.1 Use

- This script stops links being posted by default, no action required on your behalf. Moderators and above can post links without being given explicit permission
- To allow a user to post a single link moderators and above can use the command `!permit target time` or `!permit @target time` where 'target' is the user you wish to allow and 'time' is the number of seconds they're allowed to post a link for. The specified user can post any and as many links as they wish within the time frame.
- To permanently allow your trusted users to post links moderators and above can use `!permit target` or `!permit @target` to give user 'target' permissions forever. Target user will then always be allowed to post links in your chat without being given permission again. This action can be undone in the same way by using `!unpermit target` or `!unpermit @target`.

3.4.2 Settings

- **Permit & Link Permission:** The minimum permission levels required to use the command - selecting 'moderator' means that anyone with moderator *or above* can use this command (editors and broadcaster). In this instance, these permissions also specifies who can post links without being given explicit permissions.
- **Permit Command:** Defines the command used to grant a user link posting permission
- **Unpermit Command:** Defines the command used to remove link posting permissions from a user