Streamlabs Chatbot
Documentation

©Streamlabs.com

©Streamlabs.com

How to setup Streamlabs Chatbot?!

Youtube Streamer

Step 1: Go to Connections

Simply navigate to the bottom left corner of the screen and click on [icon] which will open the Connections window and then click on "*Youtube Streamer*".
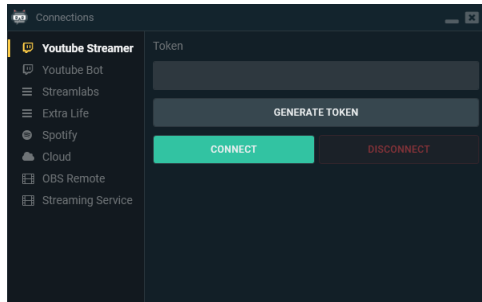
Step 3: Generating a Token

Click on Generate Oauth-Token , this will open the Authorization page on the bot.
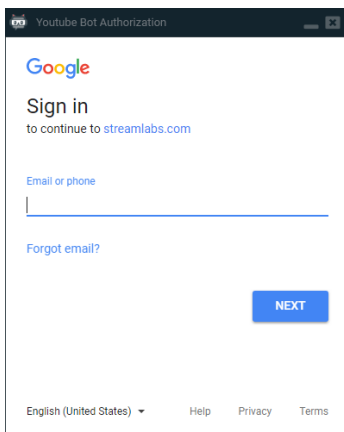
Step 4: Authorize

Login to your Google Account and then choose your account and click Allow, this will immediately fill in the token field with a new token after which you can click connect.

Step 5: Click Connect

If you've done everything correctly your account will be connected to chat. In case it fails you probably didn't activate live streaming for your account yet. This can be done by going to https://www.youtube.com/features .

Youtube Bot

Step 1: Make sure you've made a Youtube account for the bot

Go to Youtube and create a new account for the bot to use and make sure to also create the bot's channel by going to the picture in the top right and going to my channel and confirming the popup afterwards.

Step 2: Go to Connections

Simply navigate to the bottom left corner of the screen and click on which will open the Connections window and then click on "*Youtube Bot*".

## Step 4: Generating a Token

Click on Generate Oauth-Token , this will open the Authorization page on the bot. You will have to log in to your bot's account.

Step 5: Click Authorize

Login to your Google Account and then choose your bot's account and click Allow, this will immediately fill in the token field with a new token after which you can click connect.

Step 6: Click Connect

If you've done everything correctly your account will be connected to chat.

©Streamlabs.com
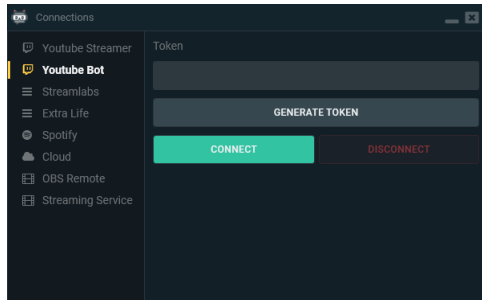
Streamlabs

Step 1: Go to Connections

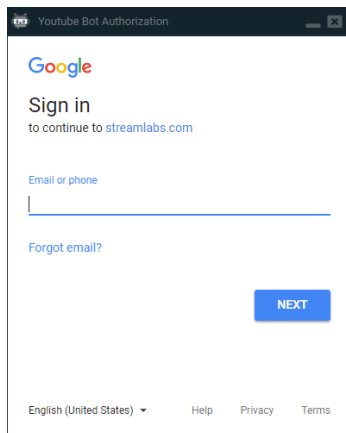Simply navigate to the bottom left corner of the screen and click on [icon] which will open the Connections window and then click on "*Streamlabs*".
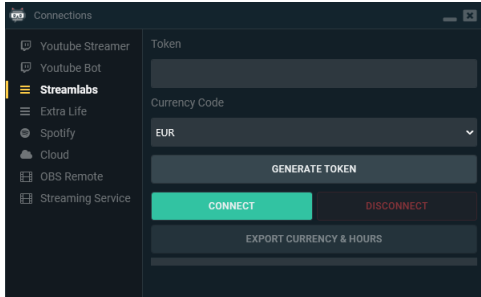
Step 2: Generating a Token

Click on "*Generate Token*" this will open the Authorization page in on the bot.
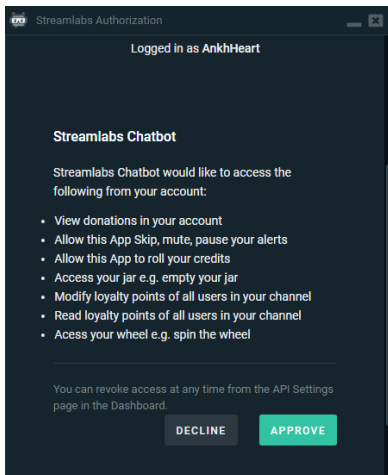
Step 3: Click Authorize

Click "*Approve*" and this will automatically fill in the token in to the token field.

Step 4: Click Connect

Finally click "*Connect*" and if everything went well then your Streamlabs will be connected. Now you can set how much someone gains for every $/€/... someone donates under the currency system.

If you wish for the bot to post an in chat notification then go to Notifications and enable the Streamlabs Donate Notification.

Step 5: Enable Chatbot pages on your Streamlabs Tipping Page

Go to Settings -> General -> Scroll to Show Chatbot Tab on Donations Page and enable it. Afterwards pick which tabs you want to show. Next go to Connections -> Streamlabs and click Sync to sync your Commands & Quotes. The Playlist, Songlist and Queue auto sync every 2.5 minutes so no need to manually sync those.

©Streamlabs.com

Cloud

Step 1: Disclaimer

In order to use this you will either have to have Dropbox, Google Drive or another similar Cloud service's client installed on your system.

These services usually come with a dedicated server on your Computer from where data will automatically be synced to the cloud.

You'll have to set these up on your own seeing as there are more than enough tutorials on youtube.

Dropbox: https://www.dropbox.com/install
Google Drive: https://www.google.com/drive/download/

Step 2: Go to Connections

Simply navigate to the bottom left corner of the screen and click on [icon] which will open the Connections window and then click on "*Cloud*".
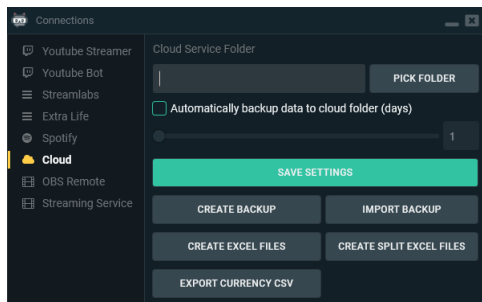
Step 3: Picking your Cloud Folder Path

Click on "*Pick Folder*" and Navigate to Cloud Service of Choice's Local folder and click "*Save*". This is where the bot will be able to output Automated Backups and Excel files which you can share with the stream.

In case you want the bot to create automated backups check the box and set the interval of the backups. Do mind though that the bot does not delete older backups so this is your responsibility. Once in a while be sure to delete some of the older ones so your Cloud data doesn't get capped out.

Click "*Save Settings*" to finish the process.

Step 4: Sharing a Link to Excel Files

Click "Create Excel Files" this will generate 3 excel files based on your data. The Data.xlxs will contain your Commands, Timers, SFX, Events, Points, Ranks, … As for the Songlist.xlsx this will contain your Songlist. The Queue.xlsx will contain your Queue.

Now that those files exist navigate to your Cloud folder, right click on the file for which you want a link.

In case you're using Dropbox click on "*Copy Dropbox Link*" this will have a link copied to your clipboard.

In case you're using Google Drive click on "*Google Drive*" -> "*Share*" -> "*Get Shareable Link*" and copy the link.

Now you can either a short link using http://tinyurl.com or use the full link in your commands.

The Songlist & Queue excel files get updated every 2.5 minutes. The Data files only get created whenever you click "*Create Read Only Excel Files*".
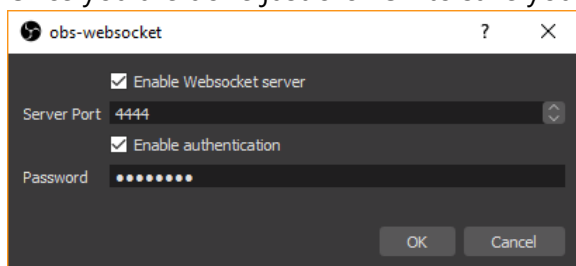
OBS Remote

Step 1: Open your Browser

Next up make sure you have the OBS Remote Plugin installed if you do not then go here:
https://obsproject.com/forum/resources/obs-websocket-remote-control-of-obs-studio-made-easy.466/ and install the plugin.

Step 2: Open OBS

Go to Tools -> Websocket Server Settings inside of OBS and it will pop up a small window that allows you to configure the OBS Remote Plugin. Set which port you wish to use there and Enable Authentication. After doing so plug in a password that you wish to use so only authorized clients may connect to your OBS.

Once you are done just click OK to save your settings.

Step 3: Go to Connections

Simply navigate to the bottom left corner of the screen and click on which will open the Connections window and then click on "*OBS Remote*".
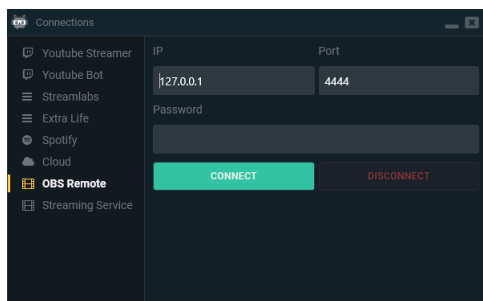
Step 4: Fill in the fields

If Streamlabs Chatbot is running on the same PC as your OBS then simply leave the IP to 127.0.0.1 if you are running a two PC setup and OBS is on another system the fill in the IP of your second PC which is running OBS.

If you changed the port in the OBS Websocket Server Settings then change it here as well. If you are using Authentication then fill in your password.

Step 5: Click Connect

Finally click connect and the bot will connect to your OBS allowing you to create commands and scripts which hide/show specific sources, Unmute your mic when you're being a dummy, Stop your stream when you pass out directly from chat so people can't watch you snooze away,...

©Streamlabs.com

Importing Data from another Bot

Simply click on the ? in the top Right -> Open External Bot Importer

FussBot
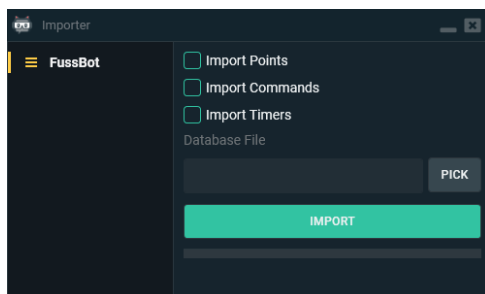
In order to Import Data from FussBot you need to navigate to your fussbot .db file. Once you have done that select which data you want to import and afterwards start the import by clicking on Import Data.

©Streamlabs.com

Features
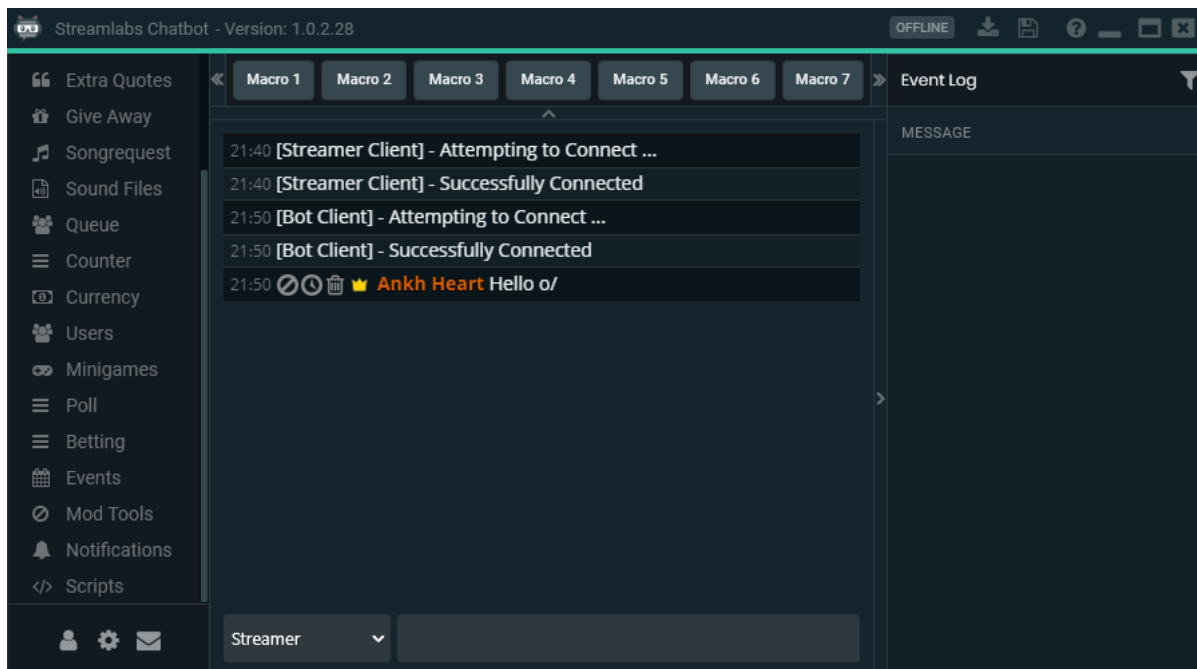
Console

On the console you will see all the incoming chat messages. On the right side you have a menu list of events such as subs, sponsors, etc.. There are other options which can be enabled by simply clicking on the filter icon.

Aside from this at the top of the console you have access to Macro buttons which you can bind commands to. Further in the document this will be explained in more detail.

Commands
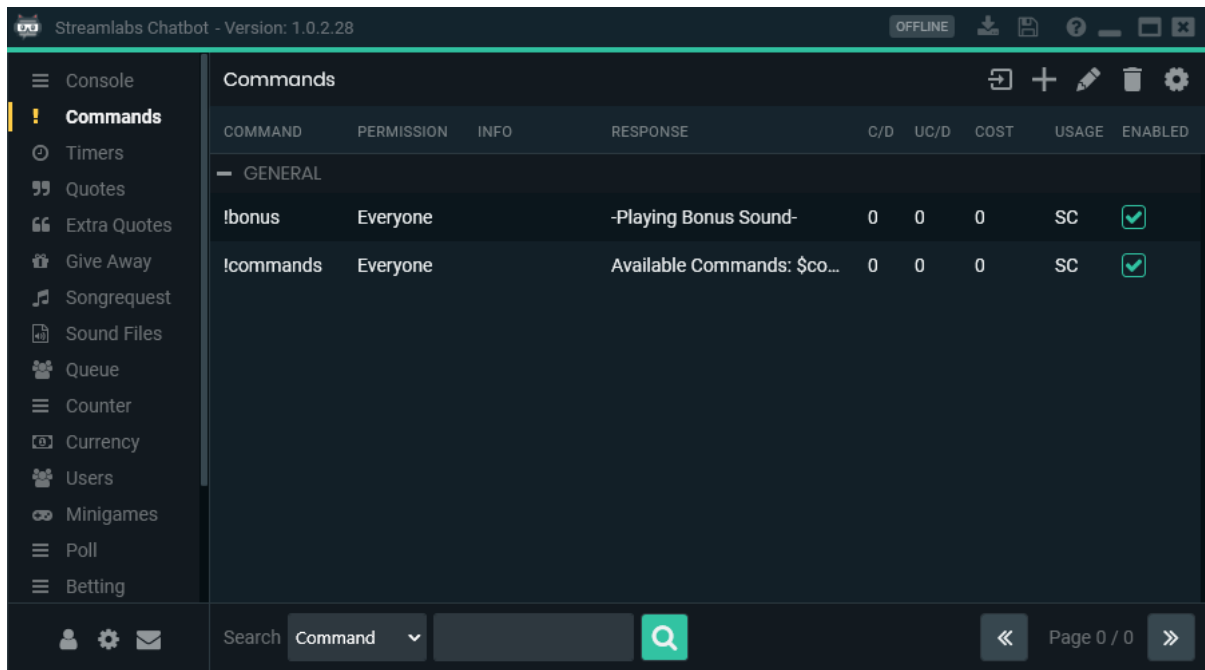
This is where you would start off if you want to create Commands. There are $parameters that you can use in the commands to achieve various result. More information on these parameters can be found on page XYZ.

$Parameters & Permission levels can be found further in to the documentation.

!Command Add (command) (permlvl) (response)	**[EDITOR]**
Example	!Command Add !Cookie +r All your cookies belong to me!
Response	{user} --> Successfully added !Cookie. Permission: Regular - Message: All your cookies belong to me!

!Command Edit (command) (permlvl) and/or (response)	**[EDITOR]**
Example	!Command Edit !Cookie +a /me ate $count cookies!
Response	{user} --> Successfully edited !Cookie. Permission: Everyone. Message: /me ate $count cookies!

| **!Command Remove (command)**	**[EDITOR]**
Example	!Command Remove !Cookie
Response	{user} --> Successfully removed !Cookie.

!Command Count (command) (num)	**[EDITOR]**
Example	!Command Count !cookie 10
Response	{user} --> Successfully set the count for !cookie to 10.

!Command Usage (command) (usage ex: SC)	**[EDITOR]**
Example	!Command Usage !cookie SC
Response	{user} --> Successfully set the usage of $command to $value.

©Streamlabs.com

!Enable (command) (true/false)   **[EDITOR]**

Example	!Enable !cookie true
Response	{user} --> Succesfully enabled !Cookie

!Command Cooldown (command) (minutes)   **[EDITOR]**

Example	!Command Cooldown !cookie 2
Response	{user} --> Successfully set the cooldown of !cookie to 2.

!Command UserCooldown (command) (minutes)   **[EDITOR]**

Example	!Command Cooldown !cookie 5
Response	{user} --> Successfully set the user cooldown of !cookie to 5.

Sharing Commands

If you wish to share commands with your fellow streamer you can export them as .abcom (Streamlabs Chatbot Command) or .abcomg (Streamlabs Chatbot Command Group) by right clicking on a command. You have two options Export Command to export the single command or Export Group to Export all commands in that specific Group.

Importing Commands

Importing a script is simple. Simply click the Import Button in   the Command Tab, Navigate to the Zip File and Open it. Afterwards the bot will import the script for you and reload your scripts so it's ready to go.

©Streamlabs.com
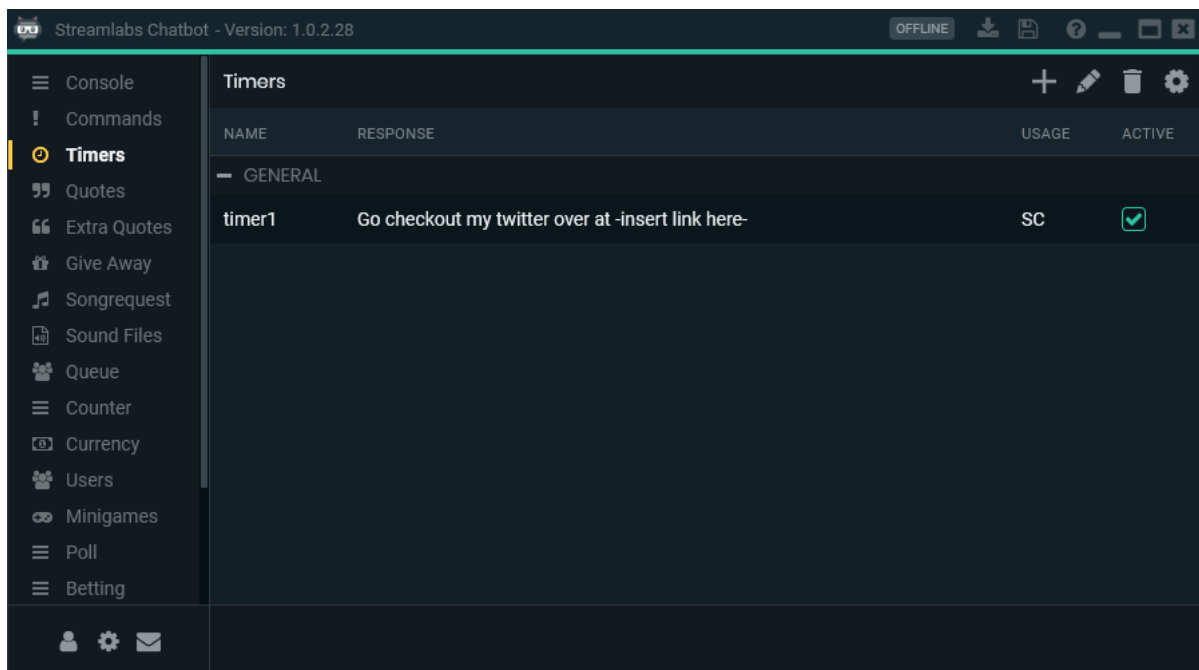
Timers

This is where you will create your own Timers. These are messages that the bot will automatically post into chat after an interval of X minutes. The interval is completely based on the Setting at the top.

All the timers will follow this same interval so this means the bot will post the first timer after the interval passes. Then it will start timing again, once the interval passes again it will post the second timer and so on eventually going through all of them and then starting back at the top.

!Timer Add (name) (response) [EDITOR]

Example	!Timer Add !Meow /me meows at $randusername
Response	{user} --> Successfully added !meow. Permission: Everyone - Message: /me meows at $randusername

!Timer Edit (name) (response) [EDITOR]

Example	!Timer Edit !Meow /me growls at $randusername
Response	{user} --> Successfully edited !Meow. Message: /me growls at $randusername!timer remove (name) / Doesn't remove command [Ed

!Timer Remove (name) [EDITOR]

Example	!Timer Remove !Meow
Response	{user} --> Successfully removed !Meow.

!Activate (name) (true/false) [EDITOR]

Example	!Activate !Meow false
Response	{user} --> Succesfully de-activated !Cookie

Quotes
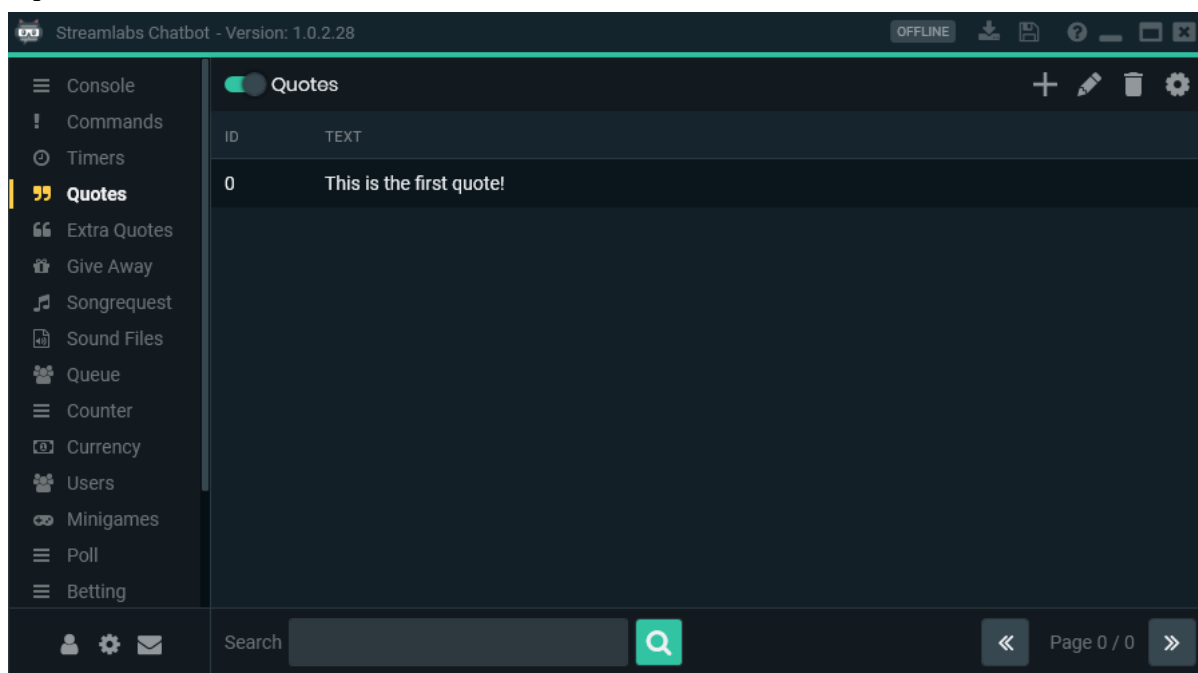
This is where things you've said on stream can be stored. You can change the permission on who can request a random quote and who can add them for you through chat.

You can also set the Cooldown and the Date Format. Every quote that gets added will automatically contain the Game & Date when the quote was created. So whenever someone calls upon the random quote they'll see when it happened and what you were playing at the time.

!Quote Add (text)	**[ADD PERMISSION]**
Example	!Quote Add "I am a cat!" - AnkhHeart
Response	{user} --> Succesfully added Quote #0: "I am a cat!" – AnkhHeart [Thief] [01/01/2015]

!Quote Edit (id) (text)	**[EDITOR]**
Example	!Quote Edit 0 "I am not a cat!" – AnkhHeart [Thief] [02/01/2015]
Response	{user} --> Successfully edited Quote #0: "I am not a cat!" – AnkhHeart [Thief] [02/01/2015]

!Quote Remove (id)	**[EDITOR]**
Example	!Quote Remove 0
Response	{user} --> Successfully deleted Quote #0

!Quote	**[VIEW PERMISSION]**
Example	!Quote
Response	Quote #2: "Duct tape solves all problems!" - AnkhHeart

!Quote (id)	**[VIEW PERMISSION]**
Example	!Quote 0
Response	Quote #0: "I am not a cat!" - AnkhHeart

Extra Quotes
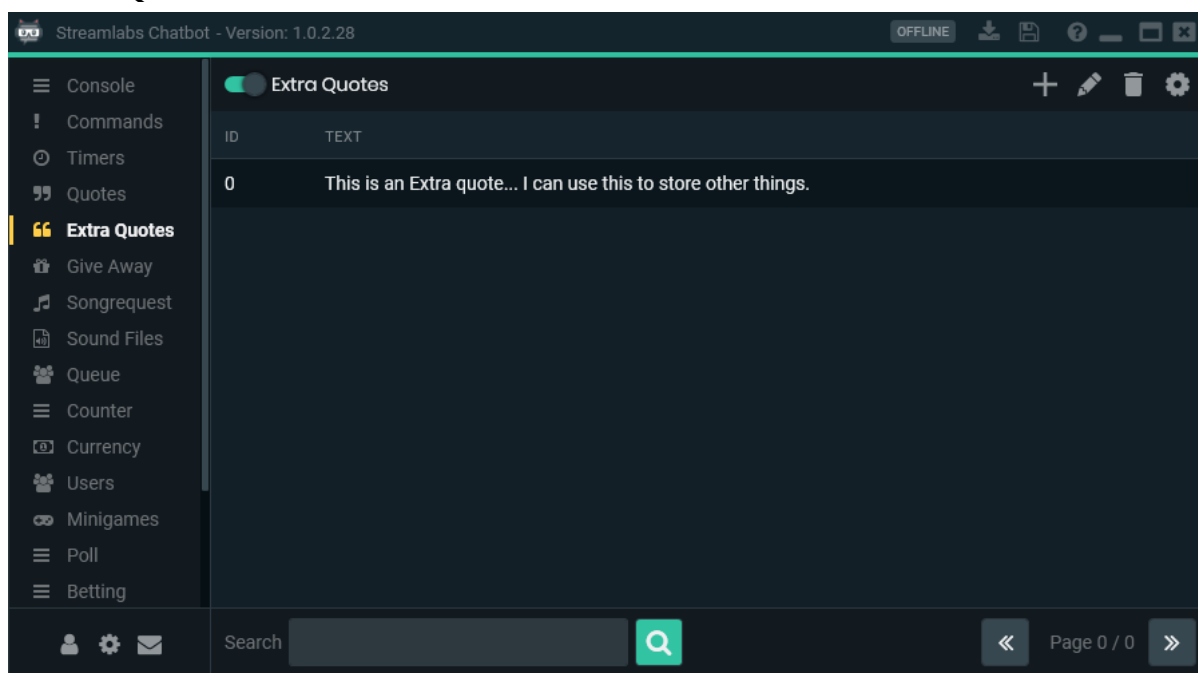
Using the Extra Quotes you can create your own version of the Quote System to store things that aren't specifically quotes. You can change the command, decide whether you want the Game & Date to show or not, change the Permissions and Response.

The underlying chat commands function the same way except if you do change the command you will also have to adjust the commands. By default this is !Gif if you change it to !Pun then you will have to use the commands starting with !Pun instead of !Gif.

!Gif Add (text)	[ADD PERMISSION]
Example	!Gif Add http://tinyurl.com/randomGif.gif
Response	{user} --> Succesfully added Gif #0: http://tinyurl.com/randomGif.gif

!Gif Edit (id) (text)	[EDITOR]
Example	!Gif Edit 0 http://tinyurl.com/randomGif2.gif
Response	{user} --> Successfully edited Gif #0: http://tinyurl.com/randomGif2.gif

!Gif Remove (id)	[EDITOR]
Example	!Gif Remove 0
Response	{user} --> Successfully deleted Gif #0

!Gif	[VIEW PERMISSION]
Example	!Gif
Response	Gif #2: http://randomURL.com/randomGif15.gif

!Gif (id)	[VIEW PERMISSION]
Example	!Gif 0
Response	Gif #0: http://randomURL.com/randomGif2.gif

Counter
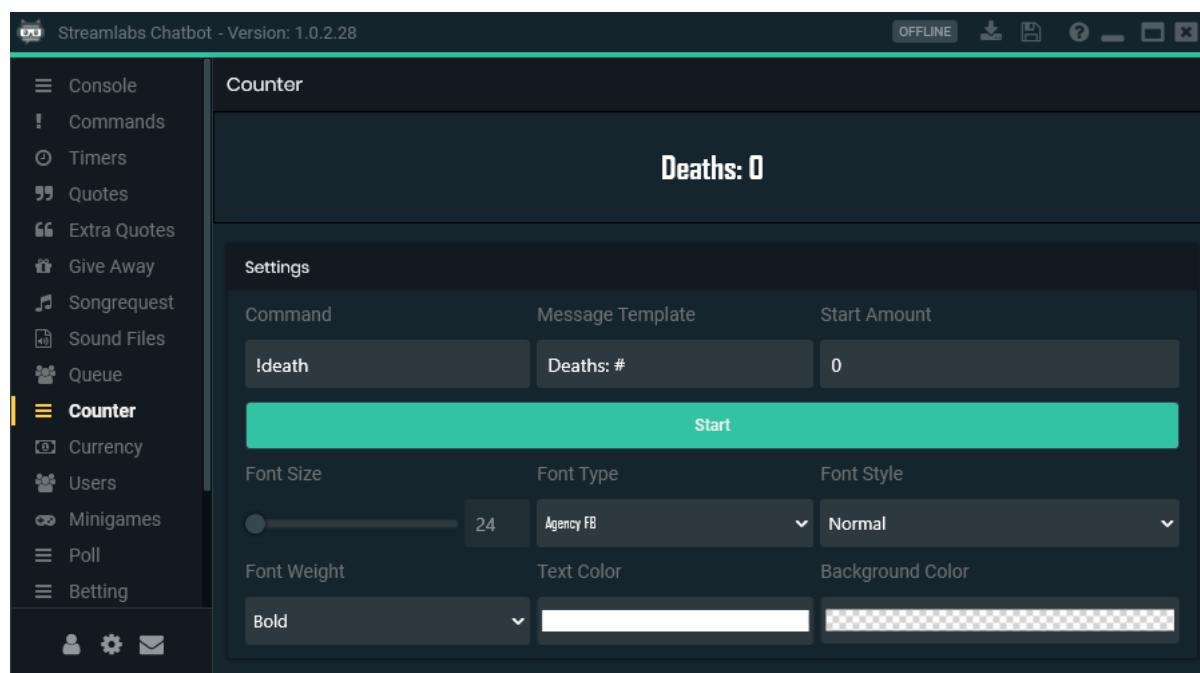
You can use the Counter to create a Death Counter, Hug Counter, Cookie Counter, etc.. It's used to count anything. You can change the settings to your liking just be sure to keep a # in the Msg Template since this will be replaced by the number.

In case you want to use the Counter but do not want to Capture the Display Area you can make use of a Death.txt file that is Located in the Bot's Install Directory -> Services -> Youtube -> Files Folder.

This file will be generated when you've added your first death. If you want to manually create this file then simply type !death 0 in chat. This will create the file with 0 Deaths inside. Do mind though if you changed the Command to something else you will have to use that instead.

!Death +   **[MOD]**

Example	!Death +
Response	[Increased] Deaths: 124

!Death -   **[MOD]**

Example	!Death -
Response	[Decreased] Deaths: 123

!Death (num)   **[MOD]**

Example	!Death 10
Response	[Set] Deaths: 10

!Death   **[EVERYONE]**

| *Example*	!Death
Response	Deaths: 10

Give Away
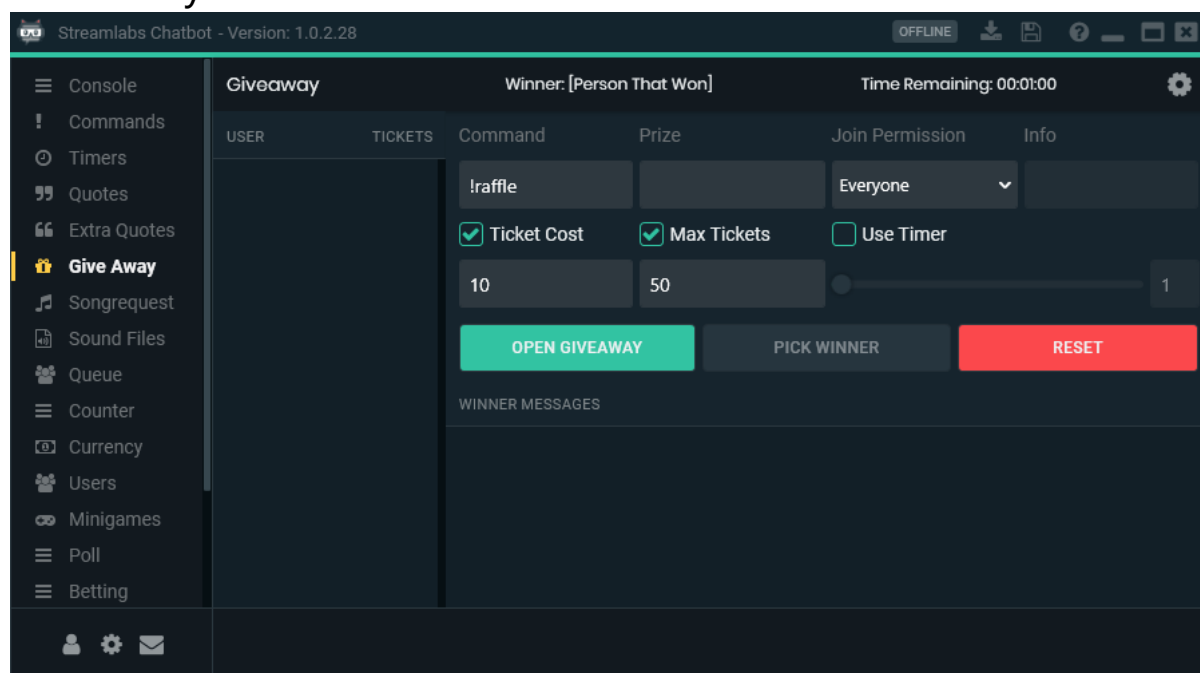
This is where you will be able to start Give Aways. You can either have people join the Give Away for free or have them pay a fee to enter or have them pay per ticket using in Channel Currency.

On the left side you will find all the people that are entered in the Give Away and how many tickets they possess. At the bottom of the window you will see all the messages posted by the Winner when one has been picked. That way you'll know if the user is active in case chat is moving really quickly.

!Giveaway Start Command|Prize|MaxEntriesPerUser|EntryCost|Permission	[EDITOR]
Example	!Giveaway Start !raffle|cookies|1|25|regular
Description	This starts a giveaway through chat with your own settings

!GiveAway Start Command|Prize|Permission	[EDITOR]
Example	!Giveaway Start !raffle|cookies|everyone
Description	This starts a very simple give away without tickets and entry costs

!GiveAway Close	[EDITOR]
Example	!Giveaway Close
Description	Prevents anyone from entering past this point

!GiveAway Winner	[EDITOR]
Example	!Giveaway Winner
Description	Randomly picks the winner for the Give Away

Sound Files
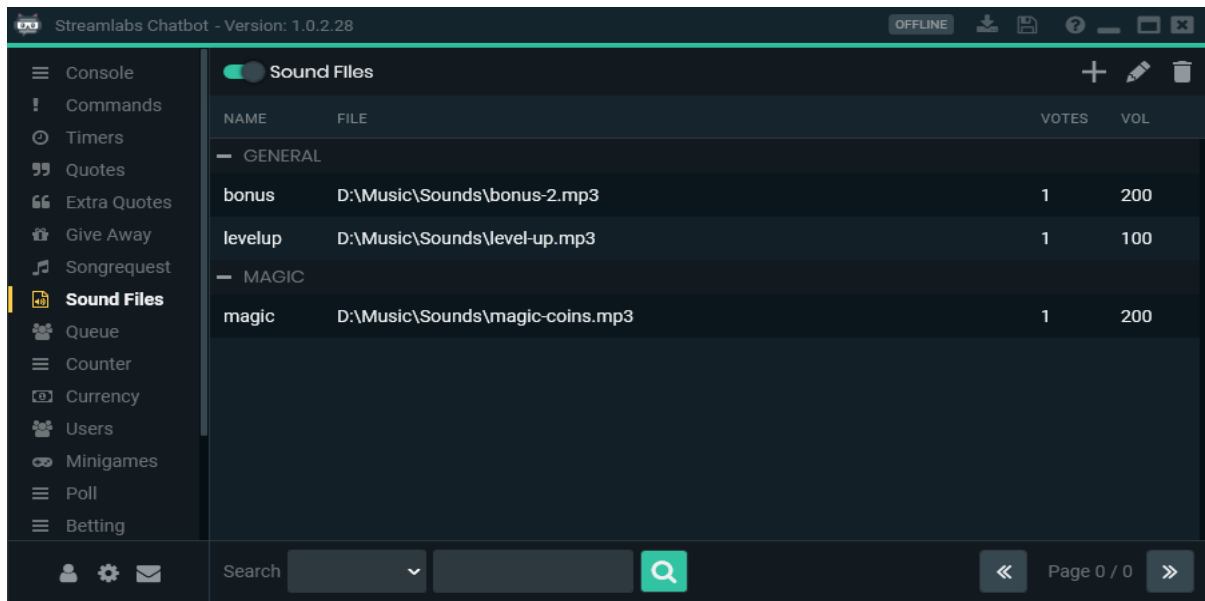
The Sound Files tab allows you to add sounds to the bot which you can attach to notifications and commands. From within this tab you are able to control the Volume and Votes. The votes option only applies to commands as it determines how many times a command has to be used before the sound goes off.

Currency

If the currency System is enabled everyone in your chat will start earning points based on your settings. These can be spent using the various other Systems in the bot such as Give Aways, SFX, Bet/Vote and enter Minigames.

You can create up to four Ranking Trees: One for Viewers, Subscribers, Mods and GameWisp Subscribers. Ranks are only assigned whenever the bot pays out points or when you use !points add +viewers 1 for example.

There is also room for customizing your own Payout amounts and intervals. This way you have full control over how many points people can accumulate in your stream.

If the Offline Payout amount is set to 0 the bot will not pay out any points with the stream is offline. Also replace !points with your own custom currency command.

!Points Add (name) (amount)	[EDITOR]
Example	!Points Add AnkhHeart 10000
Response	{user} --> Successfully given AnkhHeart 10000 Points

!Points Remove (name) (amount)	[EDITOR]
Example	!Points Remove AnkhHeart 1234
Response	{user} --> Successfully removed 1234 Points from AnkhHeart

!Points Add +Viewers / +active (amount)	[EDITOR]
Example	!Points Add +viewers 100
Response	{user} --> Done giving 100 Points to everyone in chat

!Points Remove +Viewers / +active (amount)	[EDITOR]
Example	!Points Remove +viewers 100
Response	{user} --> Done removing 50 Points from everyone in chat

!Points	[EVERYONE]
Example	!Points

Response	AnkhHeart [Ninja Kitty] - Hours: 13 - Points: 1337

!Transfer

|	
Example	!Transfer AnkhHeart MohammedBaraax1
Response	{user} --> Successfully transferred currency from AnkhHeart to MohammedBaraax1

©Streamlabs.com
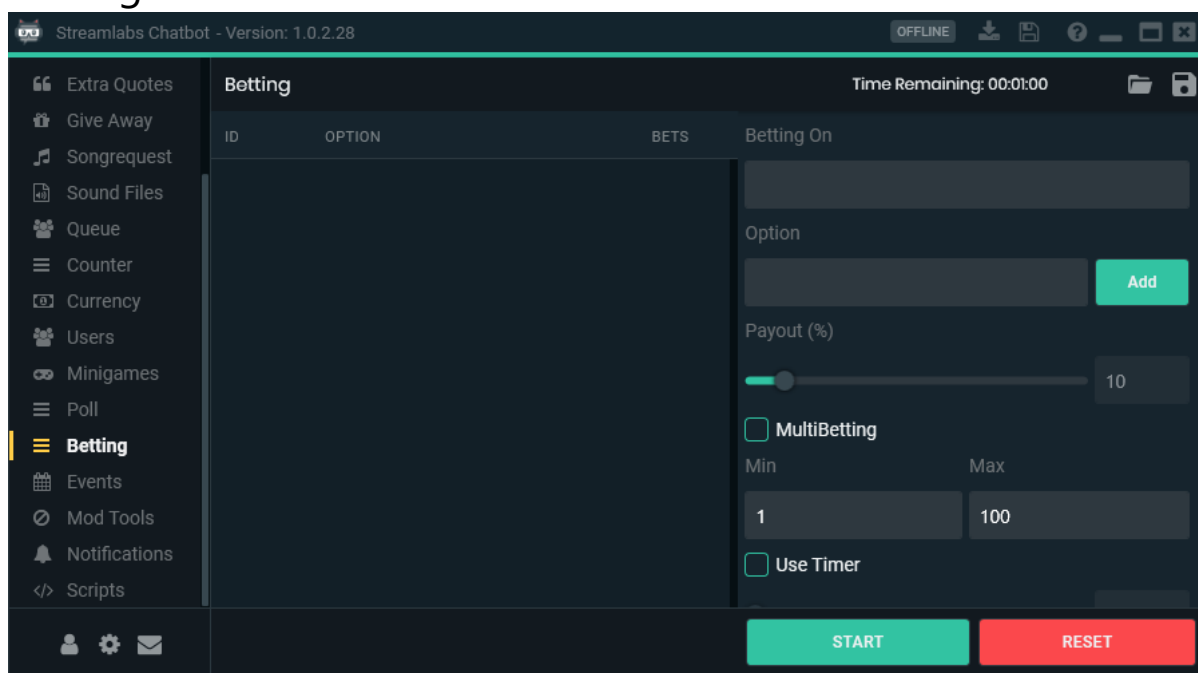
Betting

Using the Betting System you can open up the ability for Viewers to bet on the outcome of situations. These options can be saved into a present and loaded later in case you are playing the same game again.

If you wish to pick a winning option simply right click on the option and Pick it as the Winner. In case there are multiple correct Options this can be done for each of them.

!Bet (id) (amount)		[EVERYONE]
Example	!Bet 0 1000	
Response	[None to prevent chat spam from the bot]	

!Betting Start BettingOn|PayoutPercent|Min|Max|MultiBetting|Options		[EDITOR]
Example	!Betting Start Will Ankh Survive?|35|1|100|true|Yes|No|Maybe	
Description	This starts a custom betting session with custom settings	

!Betting Start BettingOn|Options		[EDITOR]
Example	!Betting Start Will Ankh Survive?|Yes|No|Maybe	
Description	This starts a custom betting session that will use the settings that have been set in the UI	

!Betting Stop		[EDITOR]
Example	!Betting Stop	
Description	Prevents anyone from betting once used	

!Betting Abort		[EDITOR]
Example	!Betting Abort	
Description	Cancels betting entirely and refunds anyone that has bet	

!Betting Winner (id)		[EDITOR]
Example	!Betting Winner 0	
Description	Picks the winning option and pay out points to everyone that bet on it	

Poll
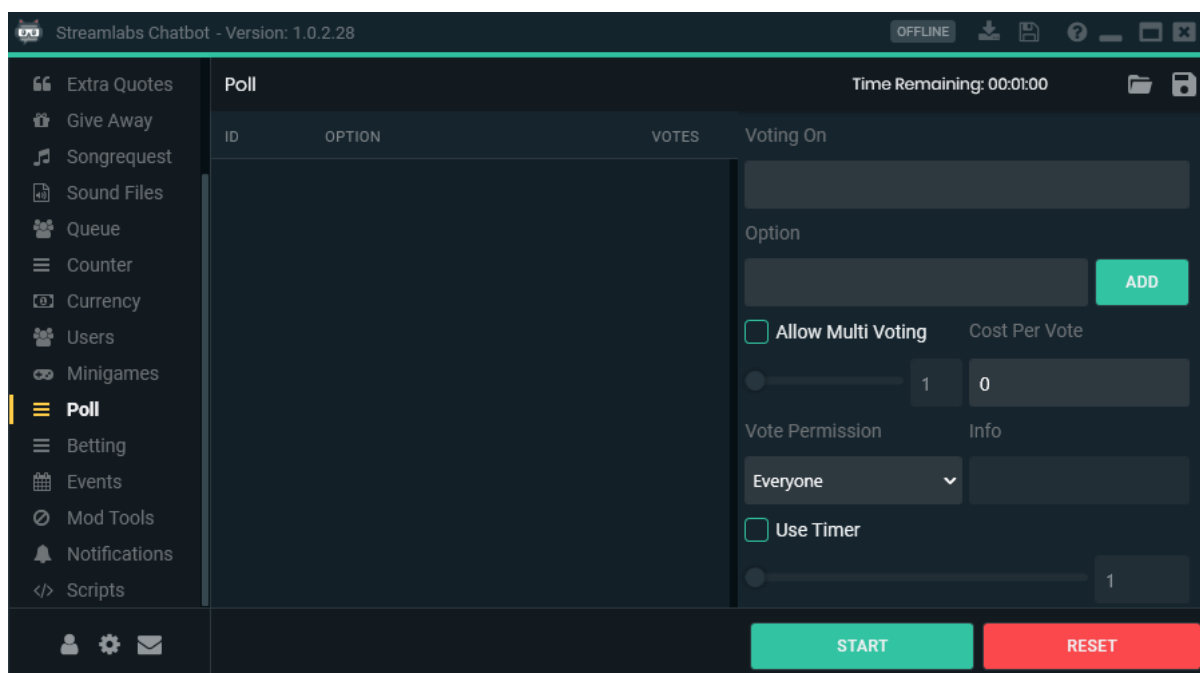
The Poll System allows you to start a poll in your channel and have your viewers vote. In case you want people to spend points for each vote they cast then you can enable this by checking Allow Multi Voting and increase the limit.

!Vote (id)	[VOTE PERMISSION]
Example	!Vote 1
Response	[None to prevent chat spam from the bot]

!Poll Start VotingOn|Cost|MaxVotes|MultiVoting|Options	[EDITOR]
Example	!poll start What Game should I play Next?|10|1|false|Witcher III|Pokemon
Description	This starts a custom poll with your own settings (overwrites UI settings)

!Poll Start VotingOn|Options	[EDITOR]
Example	!poll start What Game should I play Next?|Witcher III|Pokemon
Description	This starts a custom poll that will use the settings that have been set in the UI

!Poll Stop	[EDITOR]
Example	!Poll Stop
Description	Ends the poll and posts the result in chat

Minigames - Heist
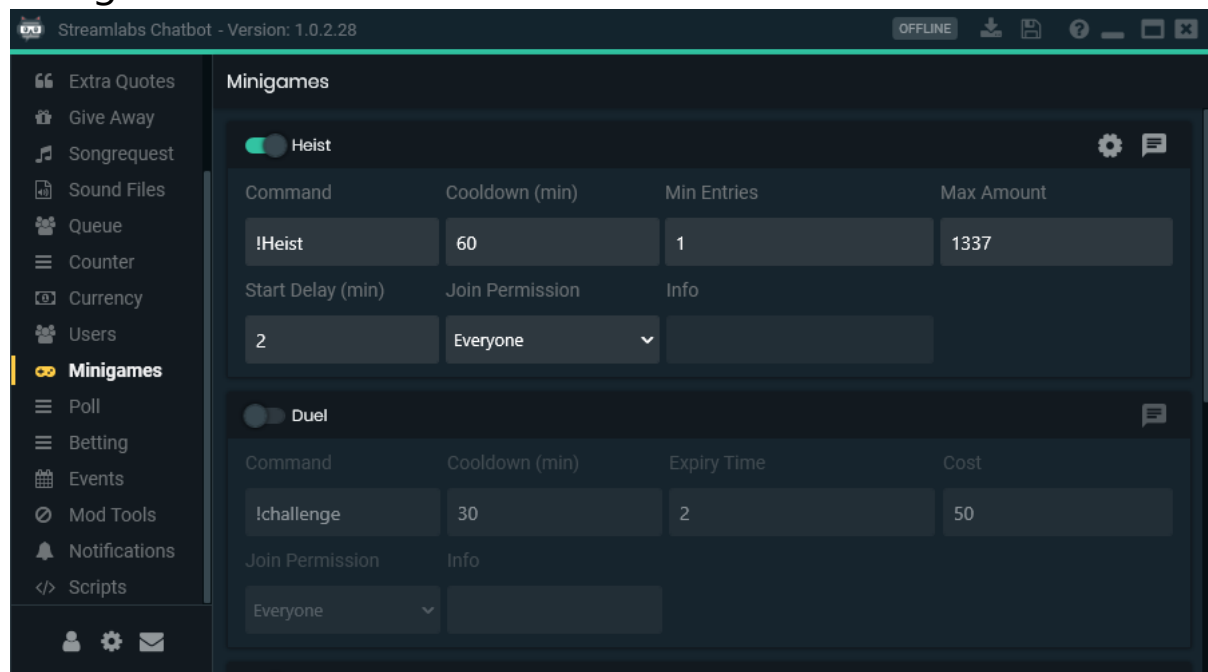
The Group Minigame allows you to create your own Minigame. You can start the customization by determining the Command that will be used, what the cooldown is, how many users have to enter before it starts, the Max amount someone can invest and who can Join.

Aside from all those options you can set the Probability for each usergroup. This determines how much chance people within that usergroup have to survive. The Payout can also be set that way you can choose how much someone gets ontop of the amount they invested in the minigame.

Finally you can fully customize all the messages that the bot will be posting in chat depending on the situation and how well/bad things are going for the ones that have joined. So if you wanted you could turn it into something completely different and not use the default Heist preset.

!Heist (amount)		[JOIN PERMISSION]
Example	!Heist 123	
Response(1x)	{user} is trying to get a team together in order to hit the nearest bank. - Everyone can Join!- In order to join type !Heist (amount).	

Minigames – Duel
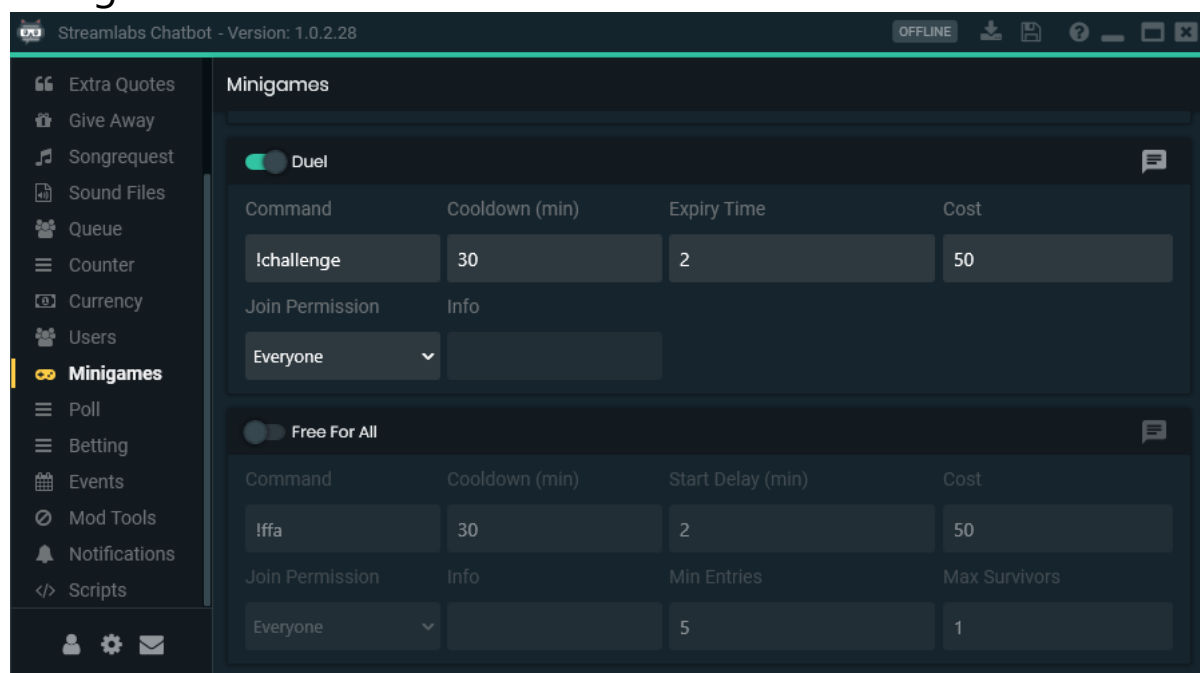
The Duel minigame allows viewers to challenge each other to a battle. The bot will process a secretive battle in the background, the winner will receive twice the cost. The loser will get nothing.

Aside from this both the challenger and challenged will go on cooldown once their fight concludes and can no longer challenge or be challenged till their cooldown expires.

!challenge (name)	[JOIN PERMISSION]
Example	!challenge ankhheart
Response(1x)	{user} has challenged {target} to a fight! Type !challenge {user} to accept the challenge!

Minigames – Free for All
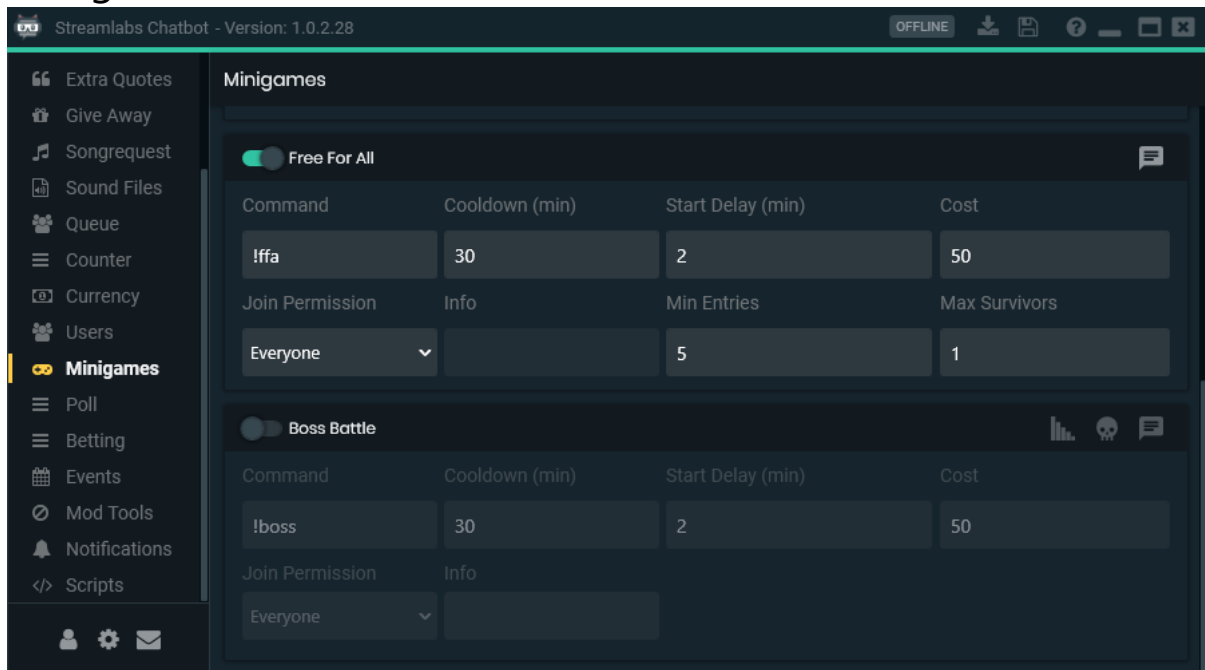
In the Free for All minigame multiple viewers can face off against one another. You can determine how many people end up surviving. The more people join the larger the prize pool becomes and the winner walks away with the pot. In more than one person can survive then it gets split amongst the survivors.

!ffa   **[JOIN PERMISSION]**

Example	!ffa
Response(1x)	The arena is now open! Type !ffa to join!

Minigames – Boss Battle
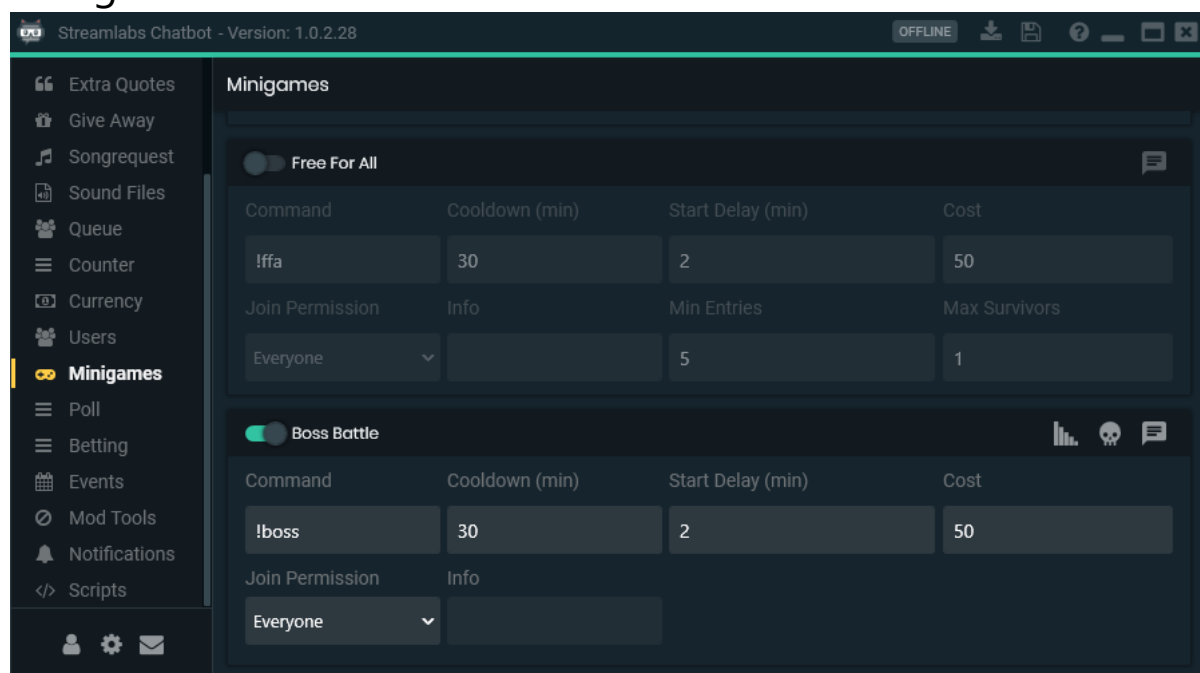
This allows you to create custom bosses for your viewers to fight based on how many people join. The difficulty / loot is completely up to you do mind that balancing it fairly is also your responsibility.

The Basics:

Balancing of the minigame is completely up to you so let's go down some of the basic concepts so you know how it functions in the background. That way you can determine what values would be best.

1) Players sign up for the battle and get a Stat sheet assigned based on their permission

2) Boss gets picked based on the group size (Between Min – Max Entries)

3) The fight starts against the boss

4) Damage Calculation: (User Attack – Target Defense) ex: 10 att – 5 def = 5 dmg that the target will receive

5) Attack order: The boss has to be attacked 3 times before it counters the last attacker. So let's say we have a group with Ankh, Momo and Gooru and Ankh attacks first and then Momo and then Gooru. After Gooru finishes his attack he would get countered by the boss and be the only person to receive damage. Now prior to every attack phase taking place the order of people attacking will be shuffled so it's not always the same person getting countered.

©Streamlabs.com

6) Make sure to keep the Boss's Defense lower than Player's Attack at all times so they at least have a chance to beat him

7) Balance the health based on the Min – Max Entries for this you will have to do a bit of math yourself based on the prior information given such as: Dmg Calculation and Attack Order

8) The Max Defense a player/boss can have is half their attack if this is higher than it will be capped out during calculation at 50%

9) Loot will get distributed evenly amongst all of the survivors at the end. In case no one survives then there is no loot to be distributed

!boss	[JOIN PERMISSION]
Example	!boss
Response(1x)	{user} is trying to get a group of adventurers together to fight a boss! Type !boss to join him!

©Streamlabs.com
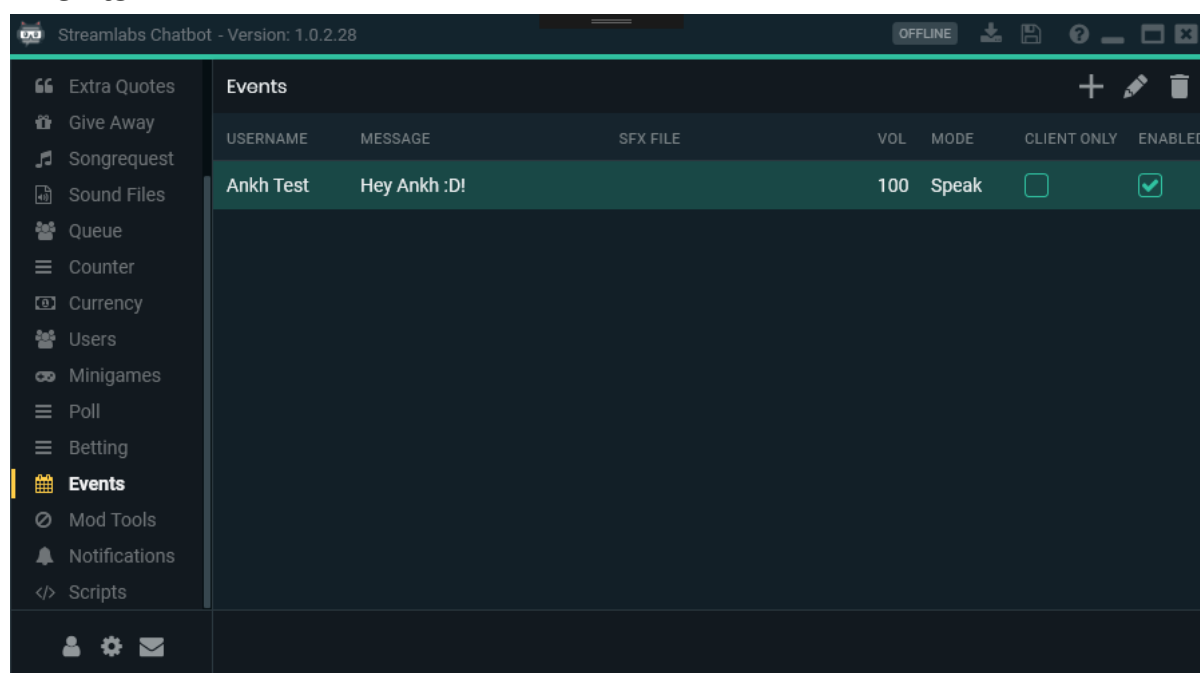
Events

The Event System will allow the bot to automatically Greet/Shoutout the person of your choice and play a SFX if you wish. The system consists of two modes Join events and Speak events.

Speak Events will perform its action when the person of your choice speaks in your channel for the first time. Then it will post its message and/or play its SFX.

In order for the bot to re-execute the events it has to be restarted. So the best thing is to restart it before a cast.

Songrequest
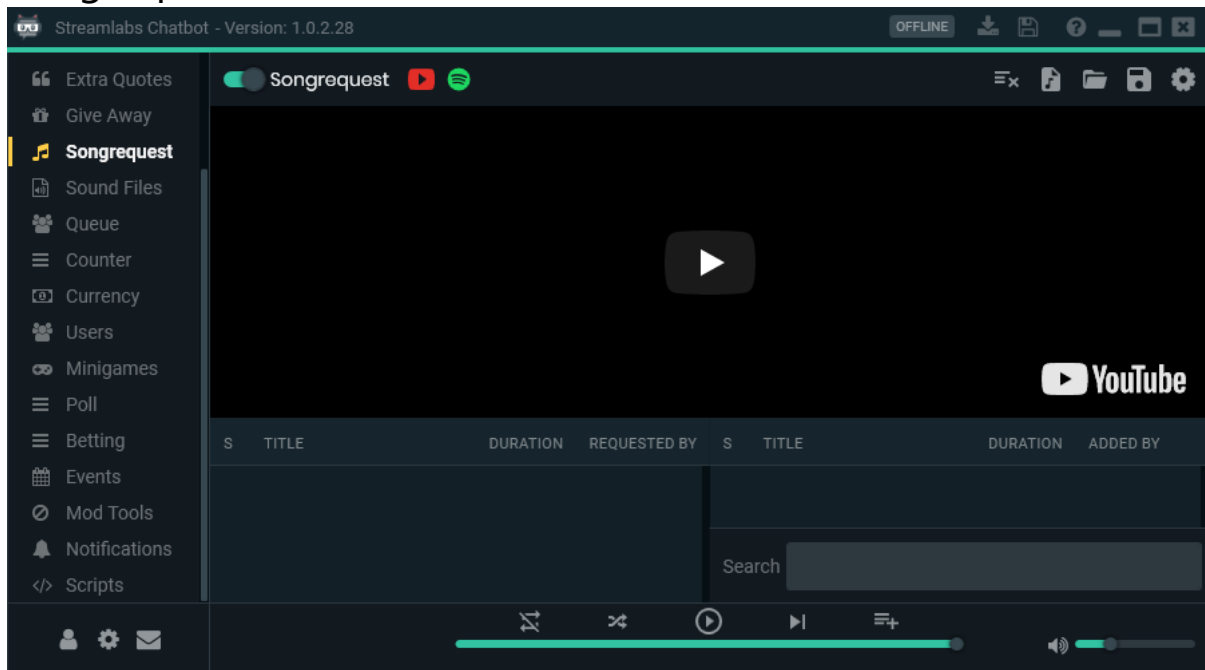
The Song Request System allow you to create your own youtube playlist through the bot have them play whenever you want. Aside from that your viewers can request songs and spend currency to do so.

!Songrequest (url/token) [REQUEST PERMISSION]

Example	!Songrequest TY9cSlOhqTk
Response	{user} --> The song Amv - [MEP] So Long Sentiment 720p has been added to the queue

!Skip [SKIP PERMISSION]

Example	!Skip
Response	{user} --> Your vote to skip has been successfully registered!

!Veto [VETO PERMISSION]

Example	!Veto
Response	Amv - [MEP] So Long Sentiment 720p has been successfully skipped!

!Songblacklist add (id) [EDITOR]

Example	!songblacklist add dQw4w9WgXcQ
Response	{user} --> dQw4w9WgXcQ has been successfully Blacklisted!

!Songblacklist remove (id) [EDITOR]

Example	!songblacklist remove dQw4w9WgXcQ
Response	{user} --> dQw4w9WgXcQ has been successfully Un-Blacklisted!

!WrongSong [EVERYONE]

Example	!WrongSong
Response	{user}, Successfully removed the last song you requested.

!Songlist

Example	!Songlist
Note	Create this yourself and point it to https://streamlabs.com/<your_name>#/chatbot/songlist

!Volume (number)   **[EDITOR]**

Example	!Volume 50
Response	{user}, Volume set to 50

Queue
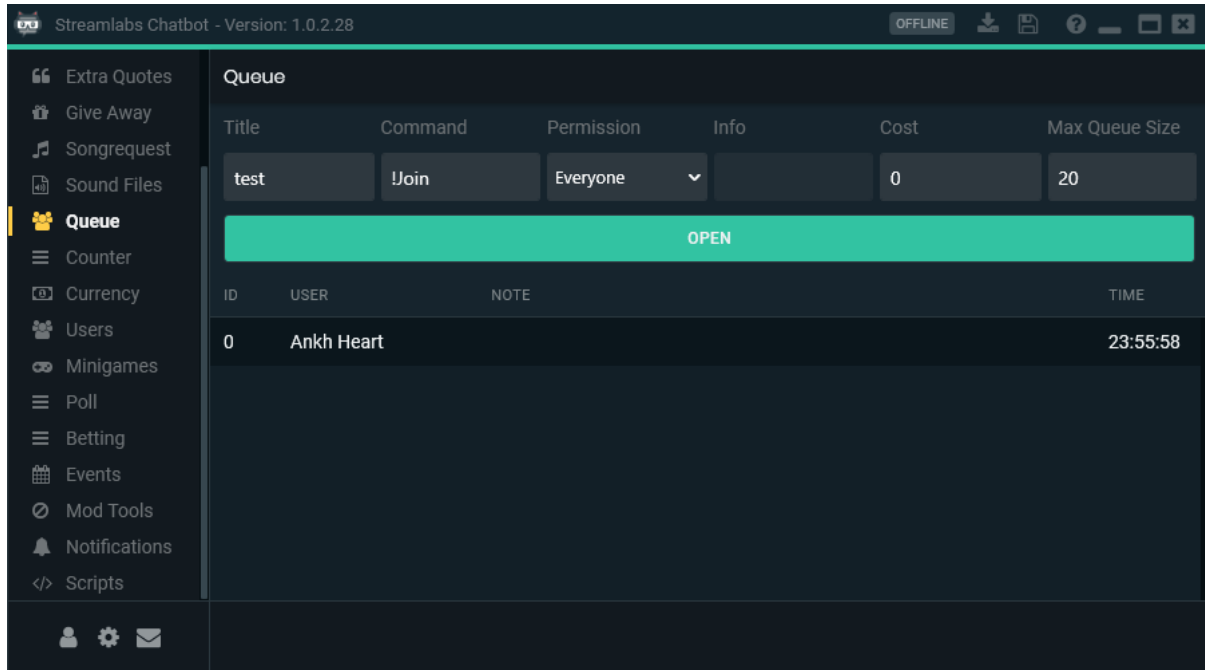
You can setup a Game Queue using this which allows your viewers to sign up to join you in a multiplayer game. You can have them spend currency to enter and you can even set it to Sub only in case you only want Subscribers to be able to sign up.

!Join <note>   **[EVERYONE]**

Example	!Join AnkhHeart#4798
Response	[None unless enabled under Settings -> Localization]

!Queuelist   **[EVERYONE]**

Example	!QueueList
Note	Create this yourself and point it to https://streamlabs.com/<your_name>#/chatbot/queue

!Queue Open <game>   **[EDITOR]**

Example	!Queue Open Warframe
Response	A queue has opened up for: Warframe - Cost: 0 points - Type !join (optional:Note) to join!

!Queue Close   **[EDITOR]**

Example	!Queue Close
Response	The queue has been closed! You can no longer enter!

!Queue Clear   **[EDITOR]**

Example	!Queue Clear
Response	The Queue has been cleared!

!Queue Pick <number>   **[EDITOR]**

Example	!Queue Pick 3
Response	Next up: AnkhHeart, Castorr91, Must13

©Streamlabs.com

!Queue Random <number>	[EDITOR]
Example	!Queue Random 3
Response	Next up: FurRiffic, WellBrained, Ocgineer

!Leave	[EDITOR]
Example	!Leave
Response	AnkhHeart has left the queue.

Notifications
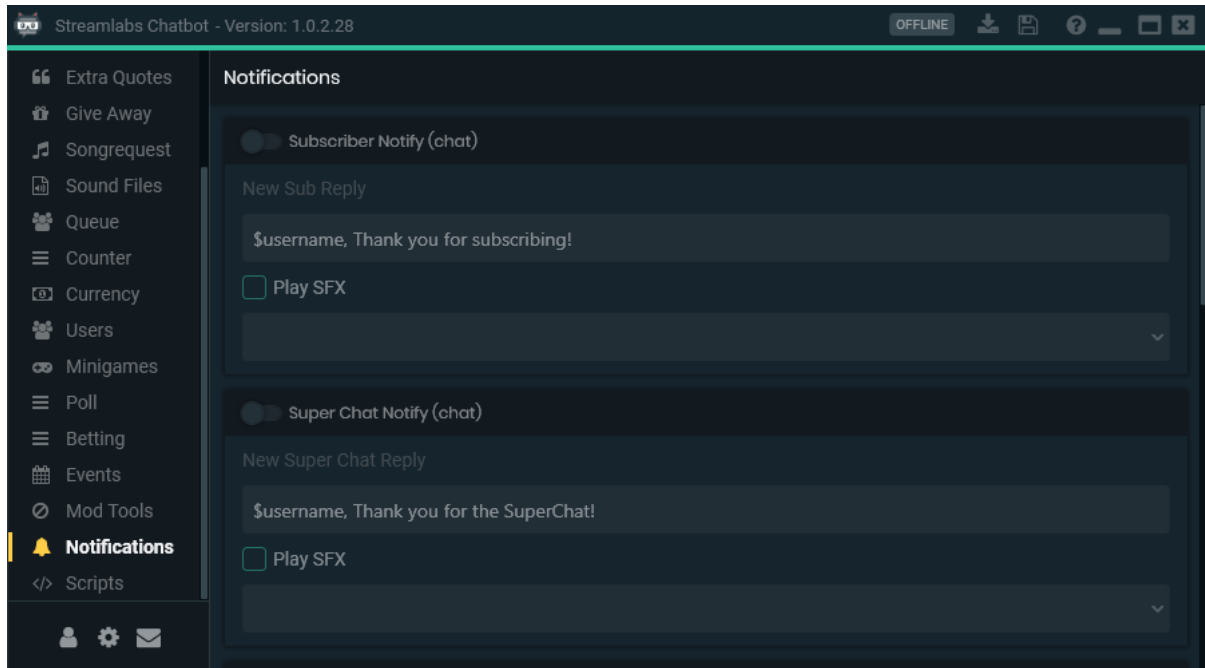


You will find various in Chat Notifications here ranging from Follower, Host, Subscriber Notifications to GameWisp Notifications. You can customize each of these to your liking.

The Subscriber, Superchat, Sponsor & Streamlabs notifications require you to have your Streamlabs account connected.

The Extra Life Notifications require you to connect your Extra Life Parcipant ID.

©Streamlabs.com
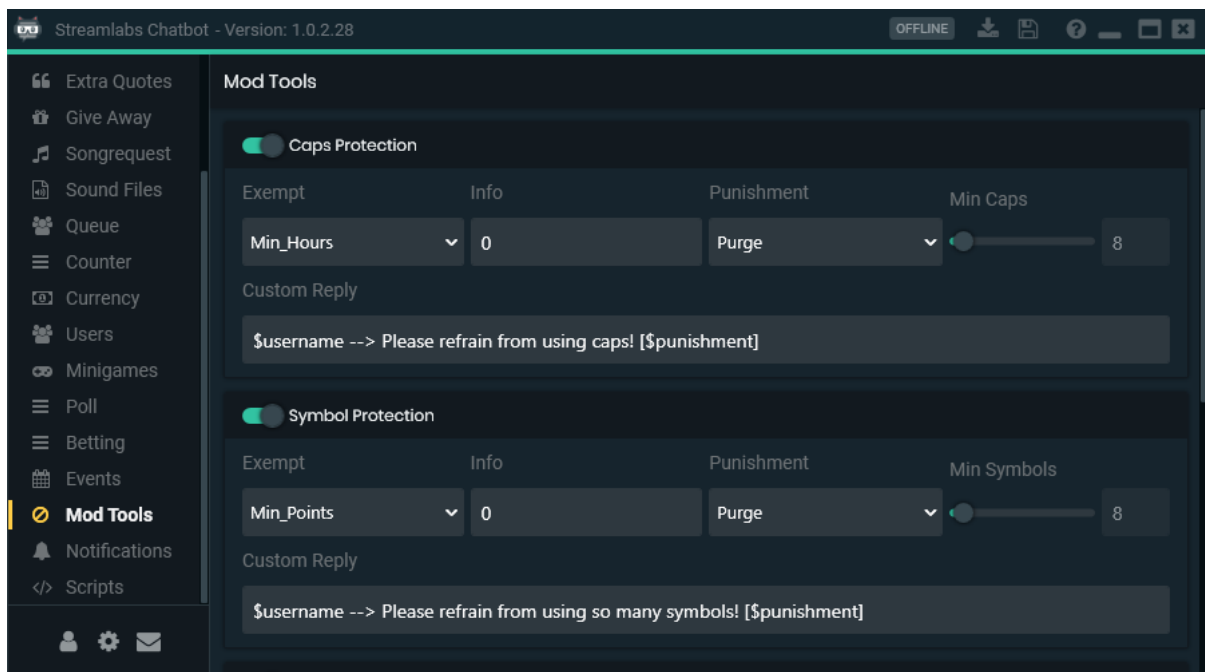
Mod Tools

Using the Mod Tools you can have the bot punish viewers that post Links without permission, Spam Caps/Symbols or very offensive words/sentences.

Each of these can be fully customized. When it comes to Link Protection you can exempt certain websites from being punished.

For the Word/Sentence Blacklist you can also make use wildcards such as * or ?. More information about Wildcards can be found on the internet ex: https://en.wikipedia.org/wiki/Wildcard_character

Users

In the Users tab you can see every user's id, name, whether they're an editor, external sub or regular. What they've been blacklisted from and when they were last in your channel. From here you are also able to assign the Editor status to users whom you trust. They are then able to use Editor based chat commands to add, edit, remove commands, currency, start giveaways, etc…

Settings

General
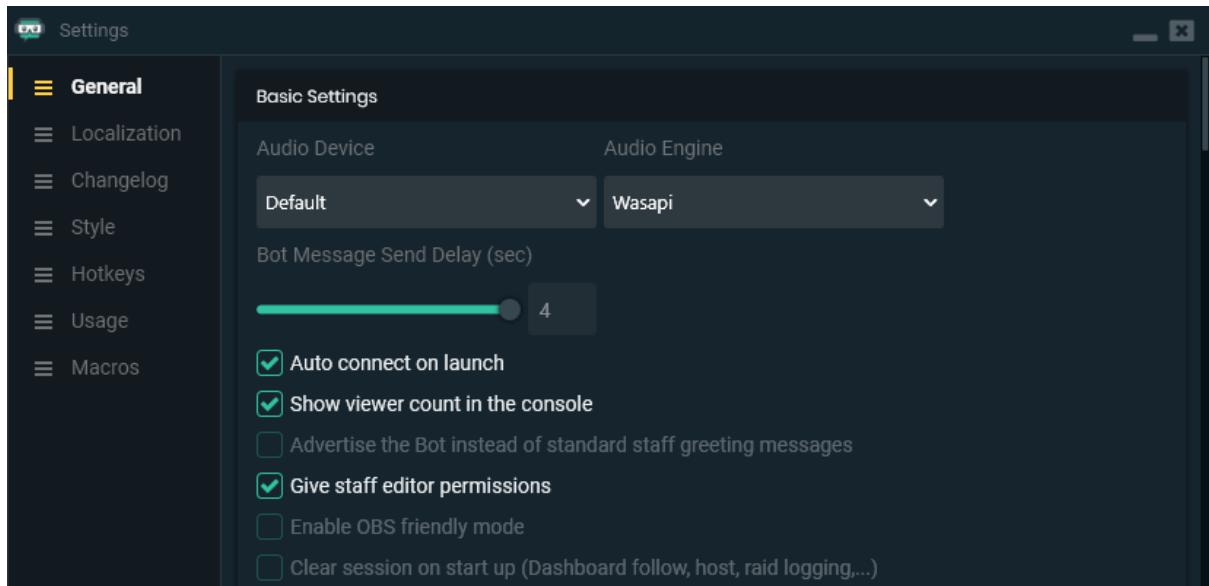
In the General Settings you will find some basic functionality such as the delay between bot messages, Auto connecting all your accounts when you launch it, etc...

Aside from this you can also setup a !raider command that can be used by Mods to save out a list of users that have raided your channel. These users will also get added to the Data.xlsx file in your Cloud folder if you have set that up.

!Reg Add/Remove (Name)	**[EDITOR]**
Example	!Reg Add AnkhHeart / !Reg Remove AnkhHeart
Response	AnkhHeart has become a Regular! / AnkhHeart is no longer a Regular.

!Sub Add/Remove (Name)	**[EDITOR]**
Example	!Sub Add AnkhHeart / !Sub Remove AnkhHeart
Response	AnkhHeart has become a Subscriber! / AnkhHeart is no longer a Subscriber.

!Blacklist Add/Remove (Name)	**[EDITOR]**
Example	!Blacklist Add AnkhHeart / !Blacklist Remove AnkhHeart
Response	AnkhHeart has been Blacklisted! / AnkhHeart has been removed from the Blacklist.

©Streamlabs.com
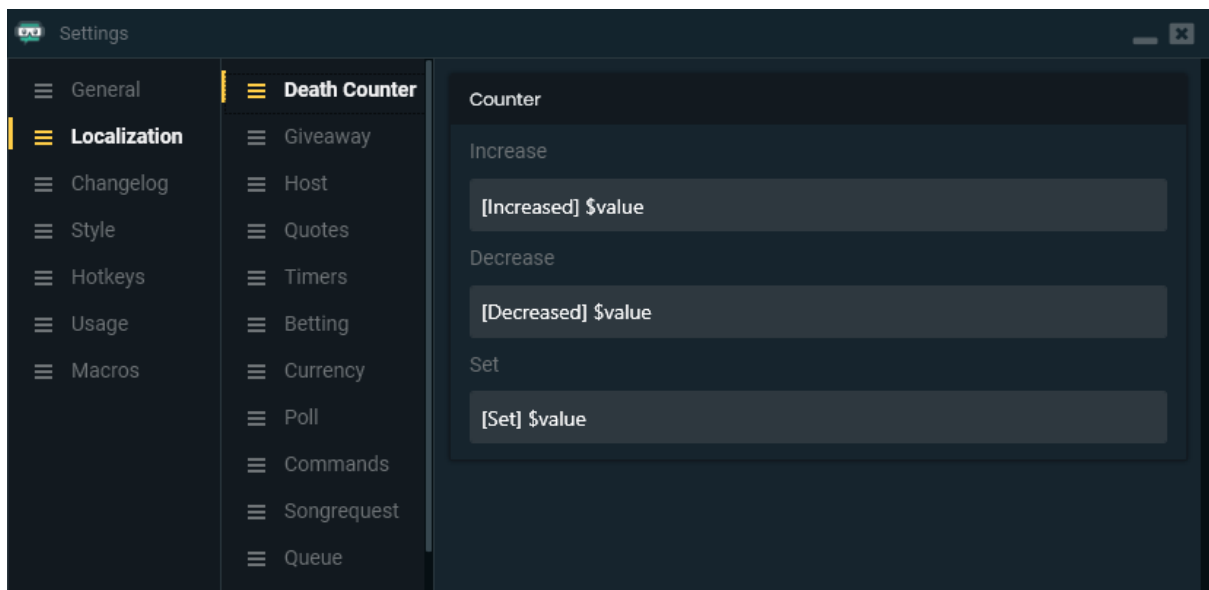
Localization

Within the Localization you have the ability to change any of the default responses though try to keep most of the $parameters unless you really don't want them to be there.

Usage
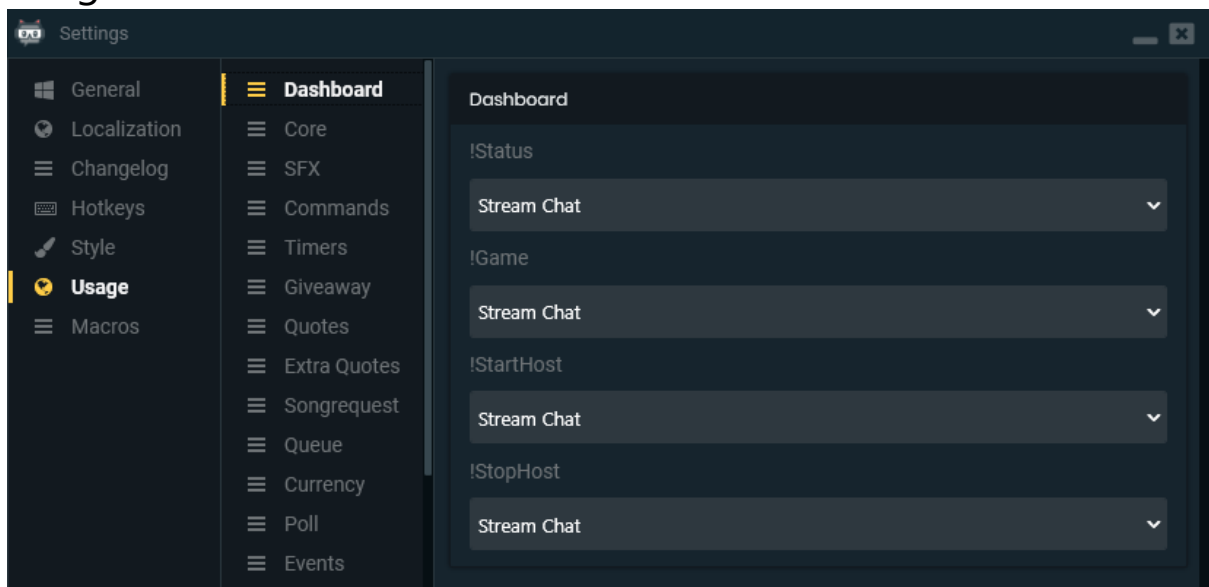
Within the Usage you can change where and how commands can be used. Whether it be Chat, Whispers or Discord or everywhere.

In the Core you can determine if users have to be in your Channel to use Whisper commands with the bot or not.

Macros
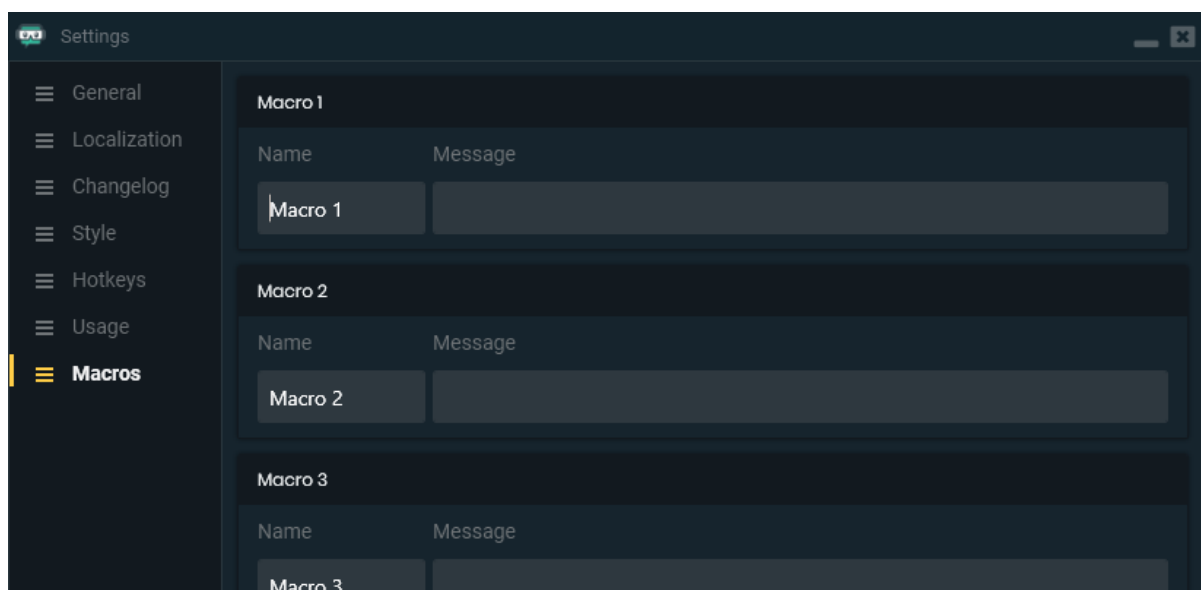
This is where you are able to setup Macros for the 7 buttons in the Console UI. It's a simple way to setup a few buttons to automatically post commands / messages in chat for you when you click them.

Hotkeys
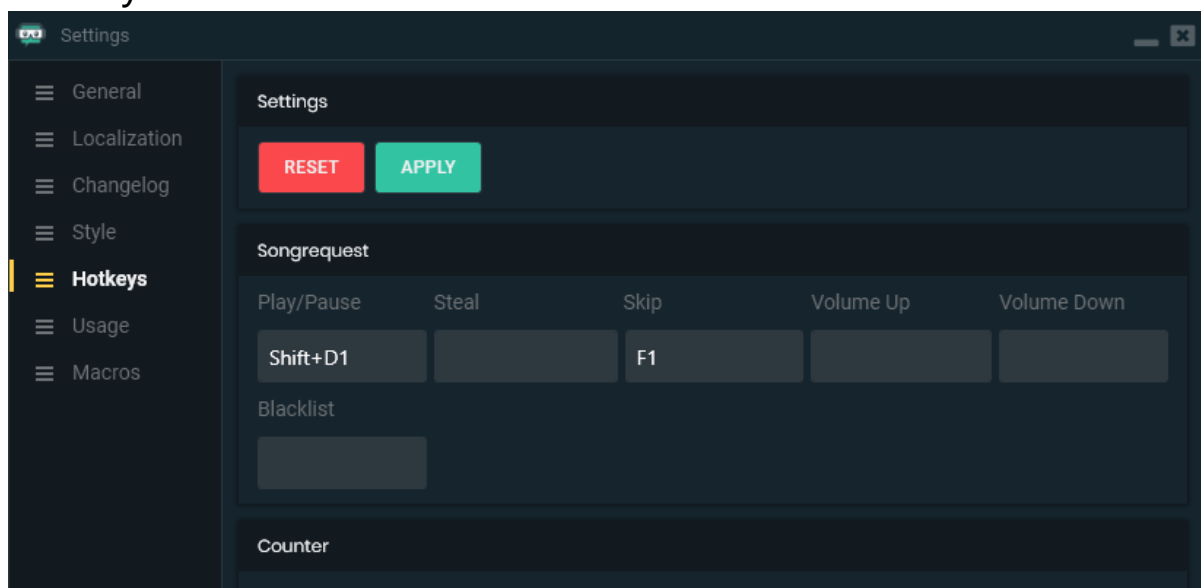
Using the Hotkeys you can set up Global Hotkeys for various actions within the bot such as Pausing a song, adding it to your playlist, skipping, increasing the volume,... Aside from this you're also able to setup Hotkeys for the previously Mentioned Macros so you don't have to click the buttons yourself. You can just hit your hotkey and it will execute them for you. Also don't forget to click Apply at the bottom to save your Hotkeys

Style
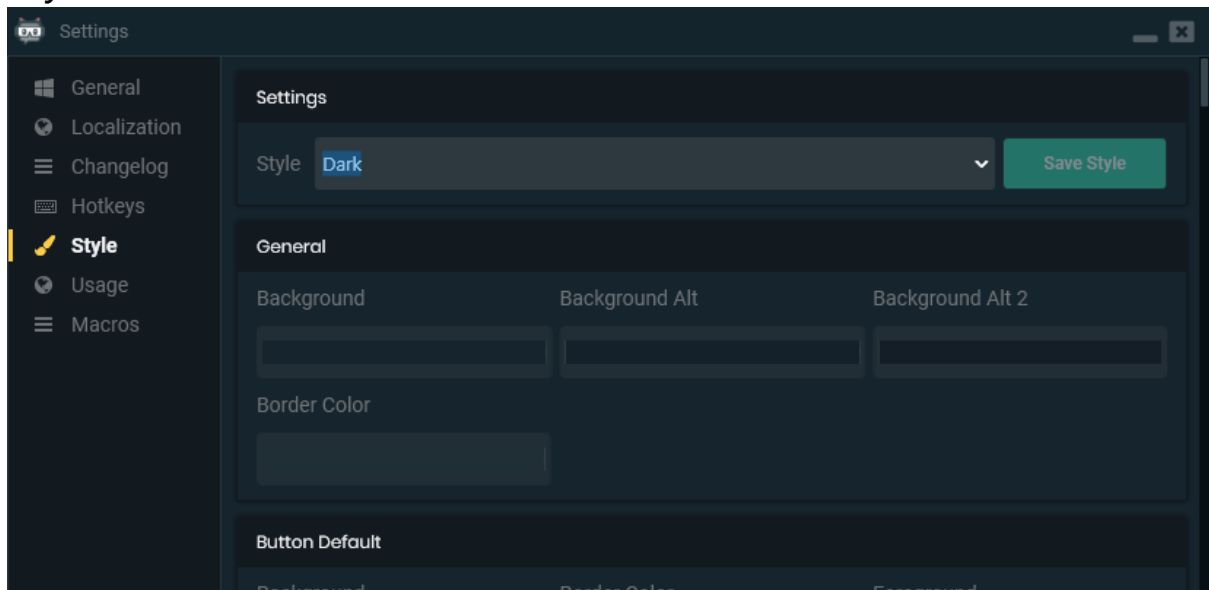
In case you're unhappy with the colours of Streamlabs Chatbot's UI then this is where you would go. You can change every single colour here though you have to set the Style to Custom. Once you're done with your changes and you like the result simply click on Save Custom Style and it will be saved.

ChangeLogs

This is where you will be to read up on the Changelogs in case you didn't read them prior to updating to a newer version.

Permission Levels

+a

Description	Everyone

+r

Description	Regular

+s

Description	Sponsor

+m

Description	Moderator

+e

Description	Editor

+i

Description	Invisible

+u(name) ex: +u(UCzWTPmaN9EUdM8Ro0Mb3FDg)

Description	User_Specific: UCzWTPmaN9EUdM8Ro0Mb3FDg

+r(MinRank) ex: +r(Lion)

Description	Min_Rank: Lion

+p(MinPoints) ex: +p(1000)

Description	Min_Points: 1000

+h(MinHours) ex: +h(13)

Description	Min_Hours: 13

Usage Levels

sc

Description	Stream Chat

Parameters

Basic Parameters

$desc(description)

Description	Special parameter that can be placed on the first line of a command to sync a custom description to the web
Example	$desc(This command does an API call somewhere!) $readapi(https://randomapi.com/thing)

$userid

Description	Displays the user's id, in case of Youtube it's the user's channel id. Make sure to use $userid when using $addpoints, $removepoints, $givepoints parameters.
Example	/me steals a cookie from $userid | !test
Response	Bot steals a cookie from ankhheart

$username

Description	Displays the user's display name. Make use of this parameter when you just want to output a good looking version of their name to chat.
Example	/me steals a cookie from $username | !test
Response	Bot steals a cookie from AnkhHeart

$targetid

Description	Displays the target's id, in case of Youtube it's the target's channel id Make sure to use $targetid when using $addpoints, $removepoints, $givepoints parameters.
Example	/me kicks $targetid in the face! | !kick AnkhHeart
Response	Bot kicks ankhheart in the face!

$targetname

Description	Displays the target's display name. Make use of this parameter when you just want to output a good looking version of their name to chat.
Example	/me kicks $targetid in the face! | !kick ankhheart
Response	Bot kicks AnkhHeart in the face!

$touserid

Description	Displays the target's or user's id, in case of Youtube it's the target's or user's channel id. Make sure to use $touserid when using $addpoints, $removepoints, $givepoints parameters.
Example	/me kicks $touserid in the face! | !kick Chair or !kick
Response	Bot kicks chair in the face! or Bot kicks ankhheart in the face

$tousername

Description	Displays the target's or user's display name. Make use of this parameter when you just want to output a good looking version of their name to chat.
Example	/me kicks $tousername in the face! | !kick thunderceptor11 or !kick
Response	Bot kicks Thunderceptor11 in the face! or Bot kicks AnkhHeart in the face!

$randuserid

Description	Displays a random user that has spoken in chat recently. In case of Youtube it's the random user's channel id. Make sure to use $randuserid when using $addpoints, $removepoints, $givepoints parameters.
Example	/me gives $randuserid a hug! | !hug
Response	Bot gives ankhheart a hug!

$randusername

Description	Displays a random user that has spoken in chat recently. Make use of this parameter when you just want to output a good looking version of their name to chat.
Example	/me gives $randusername a hug! | !hug
Response	Bot gives AnkhHeart a hug!

$msg

Description	Displays the text after the command
Example	$username rolls a $randnum(1,21) for $msg | !msg I wish I had 9 lives!
Response	Bot: AnkhHeart rolls a 18 for I wish I had 9 lives!

$dummyormsg

Description	This will get replaced by anything behind the command. If there is nothing it be cleared from the response message.
Example	http://api.com/$dummyormsg | !test Cats or !test
Response	Bot: http://api.com/Cats or http://api.com

$dummy

Description	This is a Dummy that will not post the message if there is nothing behind the command
Example	$dummy $readrandline(C:\Users\Ankh\Blah.txt) | !8ball Am I green?
Response	Bot: Perhaps?!

$arg1 to $arg9

Description	$arg1 will give you the first word after the command and $arg9 the ninth. If these parameters are in the command it expects them to be there if they are not entered the command will not post.
Example	/me hugs says $arg1 $arg2! | !argtest Hi Meow? Cookies?
Response	Bot Hi Meow?!

$argl1 to $argl9

Description	$argl1 will give you the first word after the command and $argl9 the ninth but all in lower case. If these parameters are in the command it expects them to be there if they are not entered the command will not post.
Example	/me hugs says $arg1 $arg2! | !argtest Hi Meow? Cookies?
Response	Bot hi meow?!

$num1 to $num9

Description	Expects a valid integer
Example	/me hugs $targetname $num2 times! | !hug ankhheart 10
Response	Bot hugs AnkhHeart 10 times!

$randnum(max) or $randnum(min,max)

Description	Displays a random number in a specified range
Example	/me rolls a $randnum(1,7)! | !roll
Response	Bot rolls a 3!

$randquote

Description	Displays a random quote
Example	$randquote | !randquote
Response	Bot: I am not a cat! – AnkhHeart [Thief] [01/01/2015] $randextra

$randextra

Description	Displays a random value from the extra quotes
Example	$randextra | !randgif
Response	Bot: http://randomURL.com/gif12.gif

$quotes

Description	Displays the amount of quotes
Example	There are $quotes quotes. | !quotes
Response	Bot: There are 123 quotes.

$maxquotes

Description	Displays the highest number quote
Example	There are $quotes quotes. Ranging from 0 to $maxquotes. | !quotes
Response	Bot: There are 123 quotes. Ranging from 0 to 122.

$count

Description	Counts amount of times a command has been used
Example	/me has $count jars of salt. | !count
Response	Bot has 3 jars of salt. | Bot has 4 jars of salt. | Bot has 5 jars of salt | etc...

$checkcount(command)

Description	Displays the count of a specific command
Example	Cookie Count: $checkcount(!cookie) | !check
Response	Bot: Cookie Count: 10

$commands(NumCommandsPerPage)

Description	Displays a list of all available commands for the user
Example	Commands: $commands(3) | !commands or !commands (PageNumber)
Response	Bot: Commands: !Cookie, !Slap, !Caster [Page 0/2]

$queuepos(targetid)

Description	This will display the target's position in the queue
Example	$username you are in Position $queuepos($userid) | !MyPos
Response	AnkhHeart you are in Position 1

$queue(amount)

Description	This will display the first X amount of people in the queue
Example	Next Up in Queue: $queue(3) | !NextUp
Response	Next Up in Queue: #0 mohammedbaraax1, #1 ankhheart, #2 gamegooru21

$timers(NumTimersPerPage)

Description	Displays a list of all available Timers
Example	Timers: $timers(3) | !timers or !timers (PageNumber)
Response	Bot: Timers: !ctt, !twitter, !youtube [Page 0/1]

$date

Description	Displays the Date based on the format under Quote Settings
Example	Currently it is $date
Response	Bot: Currently it is 08/09/2015

$sfx(NumSFXPerPage)

Description	Displays a list of all available SFX for the user

Example	SFX: $sfx(3) | !sfx or !sfx (PageNumber)
Response	Bot: SFX: !scream, !pika, !morph [Page 0/0]

$time

Description	Displays the caster's time
Example	Currently it is $time over at AnkhHeart's part of the world.
Response	Bot: Currently it is 10:20 PM over at AnkhHeart's part of the world.

$currencyname

Description	Displays currencyname
Example	In this channel you can collect $currencyname | !currency
Response	Bot: In this channel you can collect Cookies!

$currentsong and $requestedby

Description	Return the current song that is being played through songrequest
Example	Current Song: $currentsong – Requested By $requestedby| !currentsong
Response	Bot: Current song: ONE MORE FIGHT – Requested By AnkhHeart

$nextsong and $nextrequestedby

Description	Return the current song that is next in queue
Example	Next Song: $nextsong – Requested By $nextrequestedby| !nextsong
Response	Bot: Next song: ONE MORE FIGHT – Requested By AnkhHeart

$countdown(12:00 AM) or $countdown(04/05/2015 12:00 AM)

Description	Allows you to start a countdown from the current time to the set time/date
Example	$countdown(04/05/2015 12:00 AM) | !sleep
Response	Bot: 1 day 2 hours 48 minutes 36 seconds

$countup(12:00 AM) or $countup(04/05/2015 12:00 AM)

Description	Allows you to set a start date for when the bot should start counting
Example	$countup(07/03/2016 12:00 AM) | !UsingStreamlabs Chatbot
Response	Bot: 1 day 2 hours 48 minutes 36 seconds

$math[MathFunction]

Description	Allows you to perform math functions inside of Streamlabs Chatbot
Example	$math[10+5/2] | !Math
Response	Bot: 12

$extralifegoal

Description	Grabs your Extra Life goal
Example	$extralifegoal | !goal
Response	Bot: 5000

$extralifeamount

Description	Grabs the amount you currently raised for Extra Life
Example	$extralifeamount | !amount
Response	Bot: 100

Currency Parameters

$points

Description	Displays the num of points of the user or target
Example	$username has $points $currencyname | !cookies or !cookies ankhheart
Response	Bot: AnkhHeart has 1234 Cookies!

$pointstext

Description	Displays the num of points of the user or target nicely formatted
Example	$username has $pointstext $currencyname	!cookies or !cookies ankhheart
Response	Bot: AnkhHeart has 1,234 Cookies!

$raids

Description	Displays amount of times the user or target has raided the channel
Example	$username has raided the channel $raids time(s) so far!	!raids or !raids AnkhHeart
Response	Bot: AnkhHeart has raided the channel 3 time(s) so far!

$rank

Description	Displays the users rank
Example	$username is Rank: $rank	!rank or !rank AnkhHeart
Response	Bot: AnkhHeart is Rank: Ninja Kitty

$hours

Description	Displays amount of hours the user has been in the stream for
Example	$username spent $hours in the stream!	!hrs
Response	Bot: AnkhHeart spent 10.5 hrs in the stream!

$level

Description	Displays the users level
Example	$username is Level $level!	!Lvl
Response	Bot: AnkhHeart is Level 10

$toppoints(num)

Description	Displays top X amount of users based on points (Except Caster & Bots)
Example	Top 3: $toppoints(3) } !top3
Response	Bot: Top 3: #1 Promouse(10000), #2 Gamegooru21(9999), #3 EdeMonster(9998)

$tophours(num)

Description	Displays top X amount of users based on hours(Except Caster & Bots)
Example	Top 2: $tophours(2)	!top2
Response	Bot: Top 2: #1 KrystalRayne(123 Hrs), #2 Pixelmonkey (120 Hrs)

$pointspos

Description	Displays the users position in the ranking based on amount of points
Example	$username is ranked #$pointspos	!mypos
Response	Bot: AnkhHeart is ranked #1

$hourspos

Description	Displays the users position in the ranking based on amount of hours
Example	$username is ranked #$hourspos	!hrspos
Response	Bot: AnkhHeart is ranked #2

$nxtrankreq

Description	Displays the amount of points/hours the user requires for his next rank
Example	$username, You need $nxtrankreq points to become a $nxtrank!
Response	Bot: AnkhHeart, You need 13 points to become a Ninja Kitty!

$nxtrank

Description	Displays the next rank that the user can achieve

Example	$username, The next rank that you can achieve is $nxtrank!
Response	Bot: AnkhHeart, The next rank that you can achieve is Ninja Kitty!

$addpoints("targetid","min","max","succeed","fail")

Description	Adds points to a certain user and sends a message upon succeeding / failing
Example	$addpoints("ankhheart","10","50","ankhheart Got $value points","Failed to give points!")	!addpoints
Response	Bot: AnkhHeart got 25 points

$removepoints("targetid","min","max","succeed","fail","forceremove true or false")

Description	Removes points from a certain user and sends a message upon succeeding/failing. Force remove(true/false) removes points even if the user doesn't have enough.
Example	$removepoints("ankhheart","10","100","Removed $value points from ankhheart.","Unable To remove $value points from ankhheart!","false") |!removepoints
Response	Bot: Removed 85 points from ankhheart.

$givepoints("fromid","toid","num","succeed","fail","forcegive true or false")

Description	Gives points from one person to another.
Example	$givepoints("$userid","$targetid","50","$username gave $value points to $targetname","$username didn't have enough points to give to $targetname!","false")	!give gamegooru21
Response	Bot: AnkhHeart gave 50 points to Gamegooru21

$value [Only Works inside of $addpoints, $givepoints or $removepoints]

Description	Gets replaced with the random value between min & max
Example	$givepoints("$userid","$targetid","50","$username gave $value points to $targetname","$username didn't have enough points to give to $targetname!","false")| !give gamegooru21
Response	Bot: AnkhHeart gave 50 gamegooru21

$newbalance(targetid) [Only Works inside of $addpoints, $givepoints or $removepoints]

Description	Gets replaced with the remaining balance after a $removepoints, $addpoints or $givepoints transaction
Example	$givepoints("$userid","$targetid","50","$username gave $value points to $targetname. $targetname: $newbalance($targetid) points remaining.","fail","false")| !give gamegooru21
Response	Bot: AnkhHeart gave 50 gamegooru21. AnkhHeart 50 remaining

Sub Count

$subcount

Description	Displays your streams sub count
Example	AnkhHeart has $subcount subs! !subcount
Response	Bot: AnkhHeart has 0 subs!

File Reading Parameters

$readline(FileLocation)

Description	Reads the first line of the document
Example	$readline(C:\test.txt) | !currentsong
Response	Bot: Currently playing: Popskyy - Rize Up

$readrandline(FileLocation)

Description	Reads a random line from the file
Example	/me slaps $randusername with a $readrandline(C:\test2.txt)! | !slap
Response	Bot slaps AnkhHeart with a Tuna! | Bot slaps AnkhHeart with a Brick! | etc..

$readspecificline(FileLocation,LineNum)

Description	Reads a specific line from the file (Starts from 0)
Example	/me slaps $randusername with a $readspecificline(C:\test2.txt,3)! | !slap
Response	Bot slaps AnkhHeart with a Shovel!

Custom API Reading Parameter

$readapi(URL)

Description	Displays the text on the URL's page. Max 500 characters
Example	$readapi(https://nightdev.com/hosted/followers.php?channel=ankhheart&limit=5)
Response	Bot: 1. BensGaming808, 2. Gamakuro, 3. GENERAL_XROS, 4. wulleybully, 5. NorQuel

Save File Parameters

$savetofile("FileLocation","Text")
$savetofile("FileLocation","Text","SucceedMsg","FailMsg")

Description	Adds to the end of the file
Example	$savetofile("C:\test.txt","$msg","Succeeded :D","Failed! ") | !save I am a cat
Response	Bot: Succeeded!

$overwritefile("FileLocation","Text")
$overwritefile("FileLocation","Text","SucceedMsg","FailMsg")

Description	Overwrites all the data in the .txt file with the added text
Example	$overwritefile("C:\test.txt","$msg","Succeeded :D","Failed! ") | !save I am a cat
Response	Bot: Succeeded!

Miscellaneous Parameters

$months

Description	Only usable in the Sponsor Notifications
Example	$username just sponsored for $months months in a row!
Response	Bot: AnkhHeart just sponsored for 3 months in a row!

$donationmsg

Description	Only usable in the Streamlabs Chat Notification
Example	$username just donated $amount USD! Message: $donationmsg
Response	Bot: AnkhHeart just donated 10 USD! Message: Harro <3

Generated Text Files

The bot automatically generates text files that can be used to display information on stream. These files can be found in the Bot's Install Directory -> Services -> Youtube -> Files folder. If you don't remember where you installed the bot just Right click on its shortcut and select Open File Location. If this leads you to the Startup folder instead do it once more on the shortcut there and eventually you will end up in the Bot's install Directory.

Current Song

File	CurrentSong.txt

Requested By

File	RequestedBy.txt

Complete Current Song + Requested By

File	CurrentlyPlaying.txt

Death Counter

File	Deaths.txt

Recent Donator

File	Streamlabs_Recent_Donator.txt

Extra Life (Raised / Goal)

File	ExtraLife.txt

Extra Life Team (Raised / Goal)

File	ExtraLife_Team.txt

Extra Life Donations (per Session)

File	ExtraLife_Donators.txt

Extra Life Last Donator

File	ExtraLife_Recent_Donator.txt

Streamlabs Chatbot Python Scripting

Setup
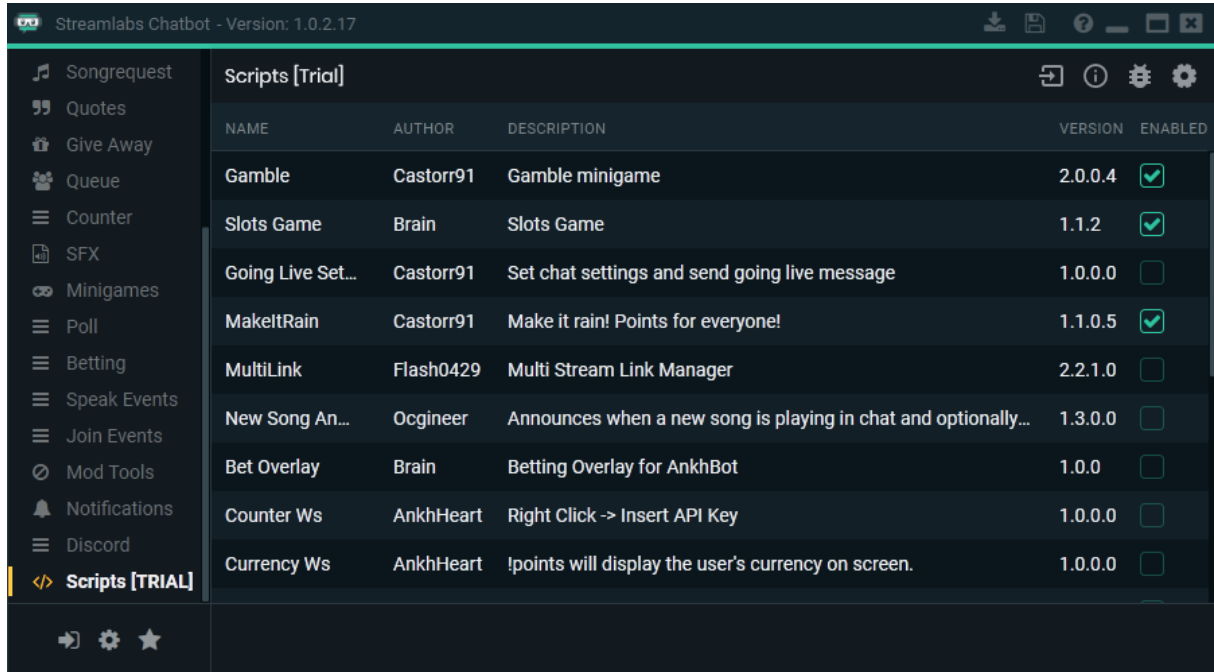
In order to properly use Python Scripts within Streamlabs Chatbot you need to have Python 2.7 installed: https://www.python.org/ftp/python/2.7.13/python-2.7.13.msi

Once you've done that click on the Settings Icon inside of the Scripts tab -> Pick Folder and navigate to your C:\Python27\Lib folder on your System. This can vary from the one that I am referring to since it all depends on where you installed Python. Once you've found the Lib folder select it and hit Save.

Afterwards you can right click within the Scripts View and reload any scripts. At the bottom of the page you will see Errors being logged from the Python Scripts that you're trying to load in case they are not considered valid or have bugs in them.

Importing
Importing a script is simple. Simply click the Import Button in [icon] the Scripts Tab, Navigate to the Zip File and Open it. Afterwards the bot will import the script for you and reload your scripts so it's ready to go.

Assigning API Key to Script

Whenever you right click on a script you're able to select `Insert Api Key` which will put a small .js file in the script's folder that will give it access to connect to the chatbot's websocket server.

The Api Key in question can be refreshed whenever you wish. Though in this case you will need to update each client.js file so it uses the appropriate key. This key is used as a password so only clients which you have granted access may connect to the server.

Creating your own script?

If you want to create your own scripts please checkout the Chatbot boilerplate over at:
Wiki: https://github.com/AnkhHeart/Streamlabs-Chatbot-Python-Boilerplate/wiki
Code: https://github.com/AnkhHeart/Streamlabs-Chatbot-Python-Boilerplate

FAQ

For the FAQ please check the website!

http://www.Streamlabs Chatbot.com/faq/