

---

# Streamlabs Chatbot

*Release Latest*

Jul 09, 2020



---

# Contents

---

<b>1</b>	<b>Documentation Contents</b>	<b>3</b>
1.1	Features . . . . .	3
1.2	Settings . . . . .	12
1.3	Permission Levels . . . . .	13
1.4	Usage Levels . . . . .	14
1.5	Basic Parameters . . . . .	14
1.6	Currency Parameters . . . . .	20
1.7	Twitch API Parameters . . . . .	21
1.8	File Reading Parameters . . . . .	22
1.9	Custom API Parameter . . . . .	22
1.10	File Saving Parameters . . . . .	22
1.11	Miscellaneous Parameters . . . . .	23
1.12	Generated Text Files . . . . .	23
1.13	Scripting . . . . .	23
<b>2</b>	<b>Frequently Asked Questions</b>	<b>27</b>
2.1	FAQ . . . . .	27
<b>3</b>	<b>Extra</b>	<b>31</b>



I (IAmTomahawkx) would like to alert you that this is *not* the Official Streamlabs Chatbot Documentation. I made this documentation so i could point people at a specific part of the documentation, instead of simply pointing at the documentation link and saying “figure it out”. Also because I figured the docs were due for an update. Any efforts to improve this documentation can be sent to me on discord, or emailed to me at [iamtomahawkx@gmail.com](mailto:iamtomahawkx@gmail.com) . Preferably don't add a pull request to the github repo, as i rarely look at github :P

Although these docs are mostly taken from the original documentation, if you would like the official documentation, it can be seen [Here](#).

Another note: this documentation is all designed with Twitch in mind. if you stream on Youtube or Mixer, you'll need to use the other docs until i finish those platforms (soon™). Sorry!



### 1.1 Features

#### 1.1.1 Console

On the console you will see all the incoming chat messages and the viewer list. In case you dislike seeing who's watching you can simply click the small button left of the viewer list to dock it to the side. 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.

#### 1.1.2 Dashboard

In order to access the full capability of the Dashboard you need to have your own Twitch account connected under Connections -> Twitch Streamer.

This should be done if you followed the entire setup guide in the beginning.

---

**Note:** You must be a Twitch Partner to use the commercials feature. Affiliates are unable to use it at this time.

---

This is where you will be able to change your Title & Game, run commercials if you're partnered with Twitch, have the bot automatically host streams of your choosing and where the bot will track news Followers, Subscribers, Raiders, Hosts and GameWisp Subscribers. We also have Raid Assist which is a system which allows you to reward viewers for joining you on a raid. The Session Event View which is located at the bottom of the Dashboard will have to be manually cleared before each stream by right clicking and Clearing the data. Otherwise new Hosts, Raids, Subs/Resubs won't be logged if they're done by the same person.

Usage	Example	Response
!Status (message)	!Status [24h] Charity Stream!	{user} -> Title has been updated: [24h] Charity Stream!
!Game (message)	!Game The Last of Us	{user} -> Game has been updated: The Last of Us
!StartHost	!StartHost	{user} -> Started Automated Hosting!
!StopHost	!StopHost	{user} -> Stopped Automated Hosting!

### 1.1.3 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.

There is also support for Command Grouping any group starting with [GAME] will only work when you're actually playing the game which is defined behind the tag

ex: [GAME] Pokemon Go

Usage	Example	Response
!Command Add (command) (permlvl) (response)	!Command Add !Cookie +r All your cookies belong to me!	{user} -> Successfully added !Cookie. Permission: Regular - Message: All your cookies belong to me!
!Command Edit (command) (permlvl) and/or (response)	!Command Edit !Cookie +a /me ate \$count cookies!	{user} -> Successfully edited !Cookie. Permission: Everyone. Message: /me ate \$count cookies!
!Command Remove (command)	!Command Remove !Cookie	{user} -> Successfully removed !Cookie.
!Command Count (command) (num)	!Command Count !cookie 10	{user} -> Successfully set the count for !cookie to 10
!Command Usage (command) (usage ex: SC,SW,SB,DC,DW,DB,CB,WB,A)	!Command Usage !cookie SC	{user} -> Successfully set the usage of \$command to \$value.
!Enable (command) (true/false)	!Enable !cookie true	{user} -> Successfully enabled !Cookie
!Command Cooldown (command) (minutes)	!Command Cooldown !cookie 2	{user} -> Successfully set the cooldown of !cookie to 2.
!Command UserCooldown (command) (minutes)	!Command Cooldown !cookie 5	{user} -> Successfully set the user cooldown of !cookie to 5 Regular - Message: All your cookies belong to me!

---

### 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.

### 1.1.4 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.

There is also support for Timer Grouping any group starting with [GAME] will only trigger when you're actually playing the game which is defined behind the tag ex: [GAME] Pokemon Go

Usage	Example	Response
!Timer Add (name) (response)	!Timer Add !Meow /me meows at \$randusername	{user} -> Successfully added !meow. Message: /me meows at \$randusername
!Timer Edit (name) (response)	!Timer Edit !Meow /me growls at \$randusername	{user} -> Successfully edited !Meow. Message: /me growls at \$randusername
!Timer Remove (name)	!Timer Remove !Meow	{user} -> Successfully removed !Meow.
!Activate (name) (true/false)	!Activate !Meow false	{user} -> Successfully de-activated !Cookie

### 1.1.5 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.

Usage	Example	Response
!Quote Add (text)	!Quote Add "I am a cat!" - AnkhHeart	{user} -> Successfully added Quote #0: "I am a cat!" - AnkhHeart [Thief] [01/01/2015]
!Quote Edit (id) (text)	!Quote Edit 0 "I am not a cat!" - AnkhHeart [Thief] [02/01/2015]	{user} -> Successfully edited Quote #0: "I am not a cat!" - AnkhHeart [Thief] [02/01/2015]
!Quote Remove (id)	!Quote Remove 0	{user} -> Successfully deleted Quote #0
!Quote	!Quote	Quote #2: "Duct tape solves all problems!" - AnkhHeart
!Quote (id)	!Quote 0	Quote #0: "I am not a cat!" - AnkhHeart

### 1.1.6 Extra Quote

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.

Usage	Example	Response
!Gif Add (text)	!Gif Add <a href="http://tinyurl.com/randomGif.gif">http://tinyurl.com/randomGif.gif</a>	{user} -> Successfully added Gif #0: <a href="http://tinyurl.com/randomGif.gif">http://tinyurl.com/randomGif.gif</a>
!Gif Edit (id) (text)	!Gif Edit 0 <a href="http://tinyurl.com/randomGif2.gif">http://tinyurl.com/randomGif2.gif</a>	{user} -> Successfully edited Gif #0: <a href="http://tinyurl.com/randomGif2.gif">http://tinyurl.com/randomGif2.gif</a>
!Gif Remove (id)	!Gif Remove 0	{user} -> Successfully deleted Gif #0
!Gif	!Gif	Gif #2: <a href="http://randomURL.com/randomGif15.gif">http://randomURL.com/randomGif15.gif</a>
!Gif (id)	!Gif 0	Gif #0: <a href="http://randomURL.com/randomGif2.gif">http://randomURL.com/randomGif2.gif</a>

### 1.1.7 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 -> Twitch -> 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.

Usage	Example	Response
!Death +	!Death +	[Increased] Deaths: 124
!Death -	!Death -	[Decreased] Deaths: 124
!Death (num)	!Death 10	[Set] Deaths: 10
!Death	!Death	Deaths: 10

### 1.1.8 Giveaway

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.

Usage	Example	Description
(see below)	(see below)	This starts a giveaway through chat with your own settings
(see below)	(see below)	starts a very simple give away without tickets and entry costs
!Giveaway Close	!Giveaway Close	Prevents anyone from entering past this point
!Giveaway Winner	!Giveaway Winner	Randomly picks the winner for the Give Away

### 1.1.9 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.

### 1.1.10 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.

The bot also supports Streamlabs currency. For this you need to connect Streamlabs and enable this functionality in your currency settings inside of the bot.

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.

---

**Note:** this is not the same as loyalty points. those are a separate thing.

---



---

**Note:** When using the extension currency, you cannot edit the hours of users from the chatbot.

---

There is also room for customizing your own Payout amounts and intervals when using the local currency. 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.

Usage	Example	Response	Per- mis- sion
!Points Add (name) (amount)	!Points Add AnkhHeart 10000	{user} -> Successfully given AnkhHeart 10000 Points	EDI- TOR
!Points Remove (name) (amount)	!Points Remove AnkhHeart 1234	{user} -> Successfully removed 1234 Points from AnkhHeart	EDI- TOR
!Points Add +View-ers/+active (amount)	!Points Add +viewers 100	{user} -> Done giving 100 Points to everyone in chat	EDI- TOR
!Points Remove +View-ers/+active (amount)	!Points Remove +viewers 100	{user} -> Done removing 100 Points from everyone in chat	EDI- TOR
!Points	!Points	AnkhHeart [Ninja Kitty] - Hours: 13 - Points: 1337	EV- ERY- ONE
!Transfer	!Transfer AnkhHeart MohammedBarax1	{user} -> Successfully transferred currency from AnkhHeart to MohammedBarax1	EDI- TOR

### 1.1.11 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.

Usage	Example	Description	Permission
!Bet (id) (amount)	!Bet 0 1000	[No response to prevent chat spam from the bot]	EVERYONE
!betting start (see below)	!Betting Start (see below)	This starts a custom betting session with custom settings	EDITOR
!Betting Start (see below)	!Points Start (see below)	<b>This starts a custom betting session that will use the settings that have been set in the UI</b>	EDITOR
!Betting Stop	!Betting Stop	Prevents anyone from betting once used	EDITOR
!Betting Abort	!Betting Abort	Cancels betting entirely and refunds anyone that has bet	EDITOR
!Betting Winner (id)	!Betting Winner 0	Picks the winning option and pay out points to everyone that bet on it	EDITOR

### 1.1.12 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.

Usage	Example	Description	Permission
!Vote (id)	!Vote 1	[No response to prevent chat spam from the bot]	EVERYONE
!Poll start (see below)	!Poll Start (see below)	This starts a custom poll with your own settings (overwrites UI settings)	EDITOR
!Poll Start (see below)	!poll start (see below)	This starts a custom poll that will use the settings that have been set in the UI	EDITOR
!Poll Stop	!Poll Stop	Ends the poll and posts the result in chat	EDITOR

### 1.1.13 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 on top 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.

Usage	Example	Response	Permission
!Heist (amount)	!Heist 123	<p><b>{user} is trying to get a team together in order to hit the nearest bank.</b></p> <ul style="list-style-type: none"> <li>• Everyone can Join!- In order to join type !Heist (amount).</li> </ul>	JOIN PERMISSION

### 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.

Usage	Example	Response	Permission
!challenge (name)	!Challenge ankhheart	{user} has challenged {target} to a fight! Type !challenge {user} to accept the challenge!	JOIN PERMISSION

### 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.

Usage	Example	Response	Permission
!ffa	!ffa	The arena is now open! Type !ffa to join!	JOIN PERMISSION

### 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.

- 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

Usage	Example	Response	Permission
!boss	!boss	{user} is trying to get a group of adventurers together to fight a boss! Type !boss to join him!	JOIN PERMISSION

### 1.1.14 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.

Join Events will perform its action when the person of your choice joins the channel. Then it will post its message and/or play its SFX.

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.

### 1.1.15 Song Requests

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.

Usage	Example	Response	Permission
!Songrequest (url/token)	!Songrequest TY9cSI0hqTk	{user} -> The song Amv - [MEP] o Long ntimnt has been added to the queue	REQUEST PERM
!Skip	!Skip	{user} -> Your vote to skip has been successfully registered!	SKIP PERM
!Veto	!Veto	Amv - [MEP] o Long ntimnt 720p has been successfully skipped!	VETO PERM
!Songblacklist add (id)	!songblacklist add dQw4w9WgXcQ	{user} -> dQw4w9WgXcQ has been successfully Blacklisted!	EDITOR
!Songblacklist remove (id)	!songblacklist remove dQw4w9WgXcQ	{user} -> dQw4w9WgXcQ has been successfully Un-Blacklisted!	EDITOR
!Wrongsong	!Wrongsong	{user}, Successfully removed the last song you requested.	EVERYONE
!Songlist	!Songlist	[Create this yourself and point it to <a href="https://streamlabs.com/&lt;your_name&gt;#/chatbot/songlist">https://streamlabs.com/&lt;your_name&gt;#/chatbot/songlist</a> ]	EVERYONE
!Volume (number)	!Volume 50	{user}, Volume set to 50	EDITOR

### 1.1.16 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.

Usage	Example	Response	Permission
!Join <note>	!Join AnkhHeart#4798	[None unless enabled under Settings -> Localization]	EVERYONE
!Queue Open <game>	!Queue Open Warframe	A queue has opened up for: Warframe - Cost: 0 points - Type !join (optional:Note) to join!	EDITOR
!Queue Close	!Queue Close	The queue has been closed! You can no longer enter!	EDITOR
!Queue Clear	!Queue Clear	The Queue has been cleared!	EDITOR
!Queue Pick <number>	!Queue Pick 3	Next up: AnkhHeart, Castorr91, Must13	EDITOR
!Queue Random <number>	!Queue Random 3	Next up: FurRiffic, WellBrained, Ocgineer	EDITOR
!Leave	!Leave	AnkhHeart has left the queue.	EVERYONE

### 1.1.17 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 Follow, Subscriber, Host & Streamlabs notifications require you to have your Streamlabs account connected.

The Cheer Notifications require you to have your Streamer Account connected.

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

### 1.1.18 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](https://en.wikipedia.org/wiki/Wildcard_character)

### 1.1.19 Discord

The Discord tab will allow you to activate specific functionality to work in Discord in regards to Timers, automatically assigning a role to everyone and even announcing when you go live. Note that you can only have one channel for interacting with the bot, and they must use the !linkdiscord command to be able to interact with the discord bot (more on that in the *faq*)

More on Discord can be seen in the *faq*

### 1.1.20 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, how many times they've raided you 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...

## 1.2 Settings

### 1.2.1 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.

Usage	Example	Response	Per- mis- sion
!Reg Add/Remove (Name)	!Reg Add AnkhHeart / !Reg Re- move AnkhHeart	AnkhHeart has become a Regular! / AnkhHeart is no longer a Regular.	EDI- TOR
!Sub Add/Remove (Name)	!Sub Add AnkhHeart / !Sub Re- move AnkhHeart	AnkhHeart has become a Subscriber! / AnkhHeart is no longer a subscriber	EDI- TOR
!Blacklist Add/Remove (Name)	!Blacklist Add AnkhHeart / !Blacklist Remove AnkhHeart	AnkhHeart has been Blacklisted! / AnkhHeart has been removed from the blacklist	EDI- TOR

### 1.2.2 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.

### 1.2.3 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.

### 1.2.4 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.

## 1.2.5 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

## 1.2.6 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.

## 1.2.7 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.

## 1.3 Permission Levels

basic permissions

Usage	Permission
+a	Everyone
+r	Regular
+s	Subscriber
+gw	GameWisp Subscriber
+m	Moderator
+e	Editor

Special Permissions

Usage	example	Permission
+u(name)	+u(ankheart)	User Specific
+r(MinRank)	+r(Lion)	Minimum Rank
+p(MinPoints)	+p(1000)	Minimum Points
+h(MinHours)	+h(13)	Minimum Hours

## 1.4 Usage Levels

Usage	Permission
SC	Stream Chat
SW	Stream Whisper
SB	Stream Both
DC	Discord Chat
DW	Discord Whisper
DB	Discord Both
CB	Chat Both
WB	Whisper Whisper
A	A

## 1.5 Basic Parameters

Parameter	Description	Example Command	Example Output
\$desc(description)	Special parameter that can be placed on the first line of a command to sync a custom description to the web	\$desc(This command does an API call somewhere!) \$readapi(https://randomapi.com/thing)	[whatever the api returns]
\$userid	Displays the user's id, in case of Twitch it's the user's name in lower case characters. Make sure to use \$userid when using \$addpoints, \$removepoints, \$givepoints parameters.	/me steals a cookie from \$userid	Bot steals a cookie from ankhheart
\$username	Make use of this parameter when you just want to output a good looking version of their name to chat.	/me steals a cookie from \$username	Bot steals a cookie from AnkhHeart
\$targetid	Displays the target's id, in case of Twitch it's the target's name in lower case characters. Make sure to use \$targetid when using \$addpoints, \$removepoints, \$givepoints parameters.	/me kicks \$targetid in the face!   !kick AnkhHeart	Bot kicks ankhheart in the face!

Continued on next page

Table 1 – continued from previous page

Parameter	Description	Example Command	Example Output
\$touserid	Displays the target's or user's id, in case of Twitch it's the target's or user's name in lower case characters. Make sure to use \$touserid when using \$addpoints, \$removepoints, \$givepoints parameters.	/me kicks \$touserid in the face! !kick Chair or !kick	Bot kicks chair in the face! or Bot kicks ankhheart in the face
\$tousername	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.	/me kicks \$tousername in the face! !kick Chair or !kick	Bot kicks chair in the face! or Bot kicks ankhheart in the face
\$botname	Displays the bot's name	Hello I am \$botname !name	Bot: Hello I am StreamlabsBot!
\$randuserid	Displays a random user that has spoken in chat recently. In case of Twitch it's the random user's name in lower case characters.	/me gives \$randuserid a hug! !hug	Bot gives ankhheart a hug!
\$randusername	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.	/me gives \$randusername a hug! !hug	Bot gives AnkhHeart a hug!
\$msg	Displays the text after the command	\$username rolls a \$randnum(1,21) for \$msg !msg I wish I had 9 lives!	Bot: AnkhHeart rolls a 18 for I wish I had 9 lives!
\$botname	Displays the bot's name	Hello I am \$botname !name	Bot: Hello I am StreamlabsBot!
\$mychannel	This will be replaced by the channel name where the bot is connected	Connected to \$mychannel !mychannel	Bot: Connected to AnkhHeart
\$dummyormsg	<b>This will get replaced by anything behind the command response message.</b>	<a href="http://api.com/">http://api.com/</a> <b>textdollar</b> dummyormsg !test Cats or !test	<b>If there is nothing to be cleared from the</b> or <a href="http://api.com">http://api.com</a>
\$dummy	This is a Dummy that will not post the message if there is nothing behind the command	\$dummy \$readrandline(C:UsersAnkhBlah.txt) !8ball Am I green?	Bot: Perhaps?!

Continued on next page

Table 1 – continued from previous page

Parameter	Description	Example Command	Example Output
\$arg1 to \$arg9	\$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.	<code>http://api.com/ \protect\T1\ textdollardummyormsg   !test Cats or !test</code>	<b>/me hugs says \$arg1 \$arg2!</b>   !argtest Hi Meow? Cookies?
\$arg11 to \$arg19	\$arg11 will give you the first word after the command and \$arg19 the ninth but all in lower case. If these parameters are in the command it expects them to be there. the command will not run without them	<b>/me hugs says \$arg1 \$arg2!</b>   !argtest Hi Meow? Cookies?	Bot: hi meow?!
\$num1 to \$num9	expects a valid integer	<b>/me hugs says \$arg1 \$arg2!</b>   !argtest Hi Meow? Cookies?	Bot hi meow?!
\$randnum(max) or \$randnum(min,max)	Displays a random number in a specified range	<code>/me rolls a \$randnum(1,7)!   !roll</code>	Bot rolls a 3!
\$randquote	Displays a random quote	<code>\$randextra   !randgif</code>	<b>Bot:</b> <a href="http://randomURL.com/gif12.gif">http://randomURL.com/gif12.gif</a>
\$quotes	Displays the amount of quotes	<code>There are \$quotes quotes.   !quotes</code>	Bot: There are 123 quotes.
\$randextra	Displays a random value from the extra quotes	<code>\$randextra   !randgif</code>	<b>Bot:</b> <a href="http://randomURL.com/gif12.gif">http://randomURL.com/gif12.gif</a>
\$maxquotes	Displays the highest number quote	<code>There are \$quotes quotes. Ranging from 0 to \$maxquotes.</code>	<b>Bot: There are 123 quotes.</b> Ranging from 0 to 122.
\$count	Counts amount of times a command has been used	<code>/me has \$count jars of salt.   !count</code>	Bot has 3 jars of salt.   Bot has 4 jars of salt.   Bot has 5 jars of salt   etc...
\$checkcount(command)	Displays the count of a specific command	<code>Cookie Count: \$checkcount(!cookie)   !check</code>	Bot: Cookie Count: 10
\$commands( NumCommandsPerPage)	Displays a list of all available commands for the user	<code>Commands: \$commands(3)   !commands or !commands (PageNumber)</code>	<b>Bot: Commands: !Cookie, !Slap, !Caster [Page 0/2]</b>

Continued on next page

Table 1 – continued from previous page

Parameter	Description	Example Command	Example Output
\$queuepos(targetid)	This will display the target's position in the queue	<b>\$username you are in Position 1</b>  !MyPos	<b>Bot: \$username is in Position 1</b>
\$queue(amount)	This will display the first X amount of people in the queue	Next Up in Queue: \$queue(3)   !NextUp	Bot: Next Up in Queue: #0 anaki, #1 ankhheart, #2 xailran
\$timers(NumTimersPerPage)	Displays a list of all available Timers	<b>Timers: \$timers(3)   !timers</b> (PageNumber)	<b>Bot: !Timers: !cct, !twitter, !youtube</b> [Page 0/1]
\$date	Displays the Date based on the format under Quote Settings	Currently it is \$date	Bot: Currently it is 08/09/2015
\$sfx(NumSFXPerPage)	Displays a list of all available SFX for the user	SFX: \$sfx(3)   !sfx or !sfx (PageNumber)	<b>Bot: SFX: !scream, !pika, !morph</b> [Page 0/0]
\$time	Displays the caster's time	<b>Currently it is \$time over a part of the world.</b>	<b>Bot: It is 10:20 PM over at AnkhHeart's part of the world.</b>
\$currencyname	Displays currencyname	<b>In this channel you can collect</b> !currency	<b>Bot: \$currency in this channel you can collect Cookies!</b>
\$nextsong and \$nextrequestedby	Return the current song that is being played through songrequest	<b>Current Song: \$currentsong</b> \$requestedby   !currentsong	<b>Bot: Requested By: ONE MORE FIGHT</b> – Requested By AnkhHeart
<b>\$nextsong and \$nextrequestedby</b>	Return the current song that is next in queue	<b>Next Song: \$nextsong – Requested By</b> \$nextrequestedby   !nextsong	<b>Bot: Next Song: ONE MORE FIGHT –</b> Requested By AnkhHeart
\$countdown(12:00 AM) or \$countdown(04/05/2015 12:00 AM)	Allows you to start a countdown from the current time to the set time/date	\$countdown(04/05/2015 12:00 AM)   !sleep	<b>Bot: 1 day 2 hours 48 minutes 36 seconds</b>
\$countup(12:00 AM) or \$countup(04/05/2015 12:00 AM)	Allows you to set a start date for when the bot should start counting	<b>\$countup(07/03/2016 12:00 AM)</b> Chatbot	<b>Bot: 1 day 2 hours 48 minutes 36 seconds</b>
\$math[MathFunction]	Allows you to perform math functions inside of Streamlabs Chatbot	\$math[10+5/2]   !Math	Bot: 12

Continued on next page

Table 1 – continued from previous page

Parameter	Description	Example Command	Example Output
\$extralifegoal	Grabs your Extra Life goal	\$extralifegoal   !goal	Bot: 5000
\$extralifeamount	Grabs the amount you currently raised for Extra Life	\$extralifeamount   !amount	Bot: 100



## 1.6 Currency Parameters

Parameter	Description	Example Command	Example Output
\$points	Displays the num of points of the user or target	\$username has \$points \$currencyname   !cookies or !cookies ankhheart	Bot: AnkhHeart has 1234 Cookies!
\$pointstext	Displays the num of points of the user or target nicely formatted	\$username has \$pointstext \$currencyname   !cookies or !cookies ankhheart	Bot: AnkhHeart has 1,234 Cookies
\$raids	Displays amount of times the user or target has raided the channel	\$username has raided the channel \$raids time(s) so far!   !raids or !raids	Bot: AnkhHeart has raided the channel 3 time(s) so far!
\$rank	Displays the users rank	\$username is Rank: \$rank   !rank or !rank AnkhHeart	Bot: AnkhHeart is Rank: Ninja Kitty
\$hours	Displays amount of hours the user has been in the stream for	\$username spent \$hours in the stream!   !hrs	Bot: AnkhHeart spent 10.5 hrs in the stream!
\$level	Displays the users level	\$username is Level \$level!   !Lvl	Bot: AnkhHeart is Level 10
\$toppoints(num)	Displays top X amount of users based on points (Except Caster & Bots)	Top 3: \$toppoints(3)   !top3	Bot: Top 3: #1 Brain(1000), #2 Kruiser8(999),#3 EdeMonster(998)
\$stophours(num)	Displays top X amount of users based on hours(Except Caster & Bots)	Top 2: \$stophours(2)   !top2	Bot: Top 2: #1 Decius(123, Hrs) #2 Ocgi-neer(120 Hrs)
\$pointstpos	Displays the users position in the ranking based on amount of points	\$username is ranked # \$pointstpos   !mypos	Bot: AnkhHeart is ranked #1
\$hourspos	Displays the users position in the ranking based on amount of hours	\$username is ranked # \$hourspos   !hrspos	Bot: AnkhHeart is ranked #2
\$nxtrankreq	Displays the amount of points/hours the user requires for his next rank	\$username, You need \$nxtrankreq points to become a \$nxtrank!	Bot: AnkhHeart, You need 13 points to become a Ninja Kitty!
\$addpoints("targetid", "min", "max", "succeed", "fail")	Adds points to a certain user and sends a message upon succeeding / failing	\$addpoints("ankhheart", "10", "50", "AnkhHeart Got \$value points", "Failed to give points!")   !addpoints	Bot: AnkhHeart got 25 points
\$removepoints("targetid", "min", "max", "succeed", "fail", "forceremove true or false")	Removes points from a certain user and sends a message even if the user doesn't have enough.	\$removepoints("ankhheart", "10", "100", "AnkhHeart removed \$value points from ankhheart.", "Unable To remove \$value points from ankhheart!", "false")   !removepoints	Bot: AnkhHeart removed 100 points from ankhheart.
\$removepoints("targetid", "min", "max", "succeed", "fail", "forceremove true or false")	Removes points from a certain user and sends a message even if the user doesn't have enough.	\$removepoints("ankhheart", "10", "100", "AnkhHeart removed \$value points from ankhheart.", "Unable To remove \$value points from ankhheart!", "false")   !removepoints	Bot: AnkhHeart removed 100 points from ankhheart.
\$givepoints("fromid", "to", "value")	Gives points from one person to another.	\$givepoints("\$userid", "targetid", "value")   !givepoints	Bot: \$userid gave 50 points to Must13

## 1.7 Twitch API Parameters

Parameter	Description	Example Command	Example Output
\$userurl	Displays the user's twitch channel URL	\$user's twitch channel is: \$userurl   !userurl	Bot: AnkhHeart's twitch channel is: <a href="http://twitch.tv/AnkhHeart">http://twitch.tv/AnkhHeart</a>
\$usergame	Displays the user's last played/current game	\$user's was/is playing: \$usergame   !usergame	Bot: AnkhHeart was/is playing: Bloodborne
\$userstatus	Displays the user's stream title	\$user's Stream title is: \$userstatus   !userstatus	Bot: AnkhHeart's Stream title is: [720p] Fable: The Lost Chapters [PC]
\$url	Displays the target's twitch channel URL	\$targetname can be found streaming at: \$url   !url AnkhHeart	Bot: AnkhHeart can be found streaming at: <a href="http://twitch.tv/AnkhHeart">http://twitch.tv/AnkhHeart</a>
\$game	Displays the target's current/last played game	\$targetname has last played: \$game   !game AnkhHeart	Bot: AnkhHeart has last played: Bloodborne
\$status	Displays the target's stream title	\$targetname Stream title is: \$title   !title AnkhHeart	Bot: AnkhHeart Stream title is: [720p] Fable: The Lost Chapters
\$myurl	Displays the twitch channel URL for your stream	My channel is: \$myurl.   !myurl	Bot: My channel is: <a href="http://twitch.tv/AnkhHeart">http://twitch.tv/AnkhHeart</a>
\$mygame	Displays the game you are currently playing	I am playing: \$mygame   !currentgame	Bot: I am playing: Bloodborne
\$mystatus	Displays your stream title	Status: \$mystatus   !mystatus	Bot: Status: [720p] Fable: The Lost Chapters [PC]
\$uptime	Displays for how long the stream has been Live	The stream has been live for: \$uptime   !uptime	Bot: The stream has been live for: 1 hour 25 minutes 58 seconds
\$followercount	Displays your streams follower count	<b>AnkhHeart has \$followercount</b>  !followercount	<b>Bot: AnkhHeart has 1070</b> <b>followers!</b>
\$subcount	Displays your streams sub count	AnkhHeart has \$subcount subs!   !subcount	Bot: AnkhHeart has 0 subs!
\$currenthosts (NumHostsPerPage)	Displays everyone that is currently hosting the stream (Only works when you're live)	Current Hosts: \$currenthosts(2)	Bot: Current Hosts: EdeMonster, Promouse [Page 0/1]
\$setgame(game) and \$settitle(title)	Allows you to set the game & title through a command and create presets for certain games	\$setgame(Dungeon Defenders II) \$settitle(MMO Mornings)   !dd2	No response as the bot will simply update the game & title

## 1.8 File Reading Parameters

Parameter	Description	Example Command	Example Output
\$readline(FileLocation)	Reads the first line of the document	\$readline(C:test.txt)   !currentsong	Bot: Currently playing: Popskyy - Rize Up
\$readrandline (FileLocation)	Reads a random line from the file	/me slaps \$randusername with a \$readrandline(C:test2.txt)   !slap	Bot slaps AnkhHeart with a Tuna   slaps AnkhHeart with a Brick(etc)
\$readspecificline (FileLocation,LineNum)	Reads a specific line from the file (Starts from 0)	/me slaps \$randusername with a \$readspecificline(C:test2.txt,3)   !slap	<b>Bot slaps AnkhHeart with a Shovel!</b>
\$mygame	Displays the game you are currently playing	I am playing: \$mygame   !currentgame	Bot: I am playing: Bloodborne

## 1.9 Custom API Parameter

Parameter	Description	Example Command	Example Output
\$readapi(URL)	Displays the text on the URL's page. Max 500 characters	\$readapi( <a href="https://nightdev.com/hosted/followers.php?channel=ankhheart&amp;limit=5">https://nightdev.com/hosted/followers.php?channel=ankhheart&amp;limit=5</a> )	Bot: 1. BensGaming808, 2. Gamakuro, 3. GENERAL_XROS, 4. wulleybully, 5. NorQuel

## 1.10 File Saving Parameters

Parameter	Description	Example Command	Example Output
\$overwritefile("fileLocation ";"Text") And \$overwritefile("fileLocation ";"Text","Succeed","Fail")	Overwrites all the data in the .txt file with the added text	\$overwritefile("C:test.txt","\$msg","Succeeded :D","Failed! ")   !save I am a cat	Bot: Succeeded!

## 1.11 Miscellaneous Parameters

Parameter	Description	Example Command	Example Output
\$months	Only usable in the Twitch Resub Notification	\$username just subbed for \$months months in a row!	Bot: AnkhHeart just subbed for 3 months in a row!
\$bits / \$totalbits	Only usable in the Cheer Chat Notification	\$username just cheered \$bits bits for a total of \$totalbits bits!	Bot: AnkhHeart just cheered 100 bits for a total of 1234 bits!
\$donationmsg	Only usable in the Streamlabs Chat Notification	\$username just donated \$amount USD! Message: \$donationmsg	Bot: AnkhHeart just donated 10 USD! Message: Harro <3
\$donationmsg	Only usable in the Streamlabs Chat Notification	\$username just donated \$amount USD! Message: \$donationmsg	Bot: AnkhHeart just donated 10 USD! Message: Harro <3
\$viewers	Only usable in the Host Chat Notification	\$username just hosted you for \$viewers viewer(s)!	Bot: AnkhHeart just hosted you for 10 viewer(s)!

## 1.12 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 -> Twitch -> 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.

Filename	Description
AmountOfFollowers.txt	Follower Amount (per session)
Followers.txt	New Follower List (per session)
RecentFollower.txt	Last Follower
AmountOfHosts.txt	Host Amount (per session)
Hosts.txt	New Hosts (per session)
RecentHost.txt	Last Host
AmountOfSubs.txt	Twitch Sub Amount (per session)
Subs.txt	New Subs (per session)
RecentSub.txt	Last Sub
CurrentSong.txt	Current Song
RequestedBy.txt	Requested By
CurrentlyPlaying.txt	Complete Current Song + Requested By
Deaths.txt	Death Counter
Streamlabs_Recent_Donator.txt	Recent Donator
ExtraLife.txt	Extra Life (Raised / Goal)
ExtraLife_Team.txt	Extra Life Team (Raised / Goal)
ExtraLife_Donators.txt	Extra Life Donations (per Session)
ExtraLife_Recent_Donator.txt	Extra Life Last Donator

## 1.13 Scripting

Scripting is a great way to take the Streamlabs Chatbot further in what it can do.

### 1.13.1 Setup

To use scripts, you'll need two things.

- you'll need to install Python 2.7. You can download it [Here](#). After installing Python, you will need to open the settings in the scripts tab of your chatbot, select Folder, and set it to **C:\Python27\Lib**. 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.

**Warning:** Do **not** use *any* other version of Python. It will not work for the chatbot.

- you'll need scripts (*surprise!*). although any scripts you find online will (most likely) *work*, they could cause harm to your computer. Scripts do have access to all your files (as any program does) and you need to be cautious when installing them. On the Streamlabs Chatbot discord (note that it is separate from the Streamlabs discord), you will find scripts that have been checked by trusted staff, and are safe to use.

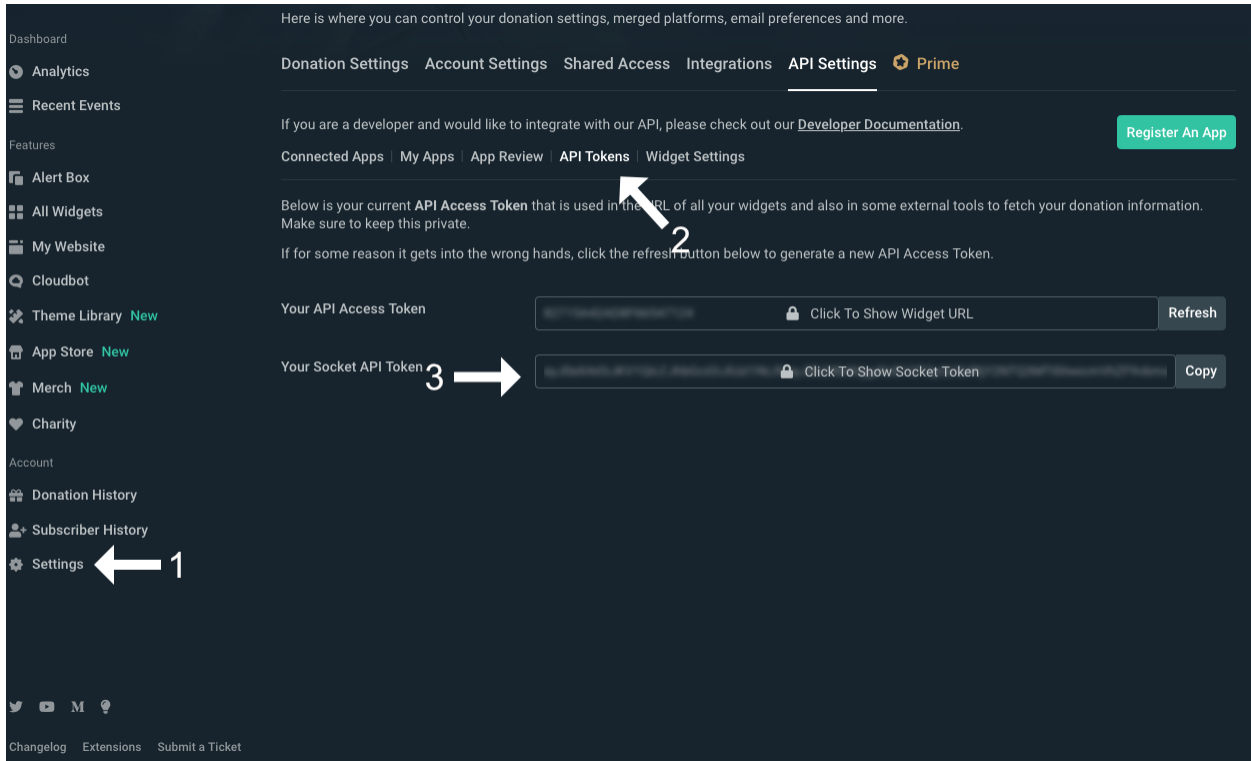
### 1.13.2 Importing Scripts

Importing a script is simple. Simply click the Import Button in 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.

### 1.13.3 Assigning API key to scripts

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.

If a script asks for the api key inside it's UI, here's one way you can get it:



### 1.13.4 Creating Scripts

*One of Us! One of Us!* If you're going to create scripts, check out the [boilerplate](#) and the [wiki](#)  
Also, join us in **Script-Devs** on the Streamlabs Chatbot [discord server!](#)



---

## Frequently Asked Questions

---

### 2.1 FAQ

#### 2.1.1 Where'd My Scripts Tab Go????

This is probably the most frequently asked question we get. The *scripts* tab only shows up if you have valid Streamer/Bot account tokens. If your *scripts* isn't showing up, it most likely means your token(s) are invalid. Go to Settings (gear icon, bottom left) > Streamer/Bot > Generate Token > Connect. That should solve the case of the disappearing tab

#### 2.1.2 Songrequest links

The default setting for *Song Requests* allows for only direct youtube links. you can change this by going into the *Song Requests* settings, changing the *mode* from **\$id** to either of the **\$readapi** options. Now you should be able to request songs by name.

#### 2.1.3 Bot not responding

Your tokens are probably expired. You need to You need to *regenerate them*

#### 2.1.4 Local and extension currency. What's the difference?

The local currency is stored directly in the bot, and allows for more customization. The extension currency is stored in the cloud, and syncs to the streamlabs currency. this does not allow as much customization.

#### 2.1.5 Notifications aren't working

Click the in the bottom left corner of the bot and go to streamlabs

- Disconnect from streamlabs
- Generate token (make sure it's using the streamer account)
- Connect to streamlabs

---

**Note:** you also need to have any widget open. Alert box/event list/ chat box/streamlabels/ Streamlabs OBS

---

### 2.1.6 Can't update game/tile through dashboard or command

You need to *regenerate your tokens*

### 2.1.7 Notifications aren't working

You need to *regenerate your tokens*

### 2.1.8 How do I regenerate my tokens?

You can regenerate your tokens by going to (bottom left corner), clicking on Streamer Account, pressing disconnect, then generate token, then connect. repeat this process for the Bot Account, and Streamlabs

### 2.1.9 Discord

#### How do I connect the bot to my Discord server?

We made two guides for that, one for the [Bot Owner](#) and one for [Everyone Else](#)

#### Can the bot connect to more than one channel in my Discord?

No not right now

#### Commands in Discord only works for me and nobody else?

Everyone that wants to use the bot in Discord must follow the steps in [this](#) guide.

### 2.1.10 Scripts

#### How do i get started with scripts?

It only takes a few minutes and have a guide for it [here](#)

#### How do I add an overlay from a script?

Follow the steps in [this](#) guide

### Where do I find documentation for scripts?

there will be new documentation coming as part of this documentation, but for now you can see the original documentation over at <https://github.com/AnkhHeart/Streamlabs-Chatbot-Python-Boilerplate/wiki>



## CHAPTER 3

---

### Extra

---

- [Developer Documentation](#)

can't find what you're looking for?

- [search](#)
- [Join the Streamlabs Chatbot Discord](#)