
twitch-observer Documentation

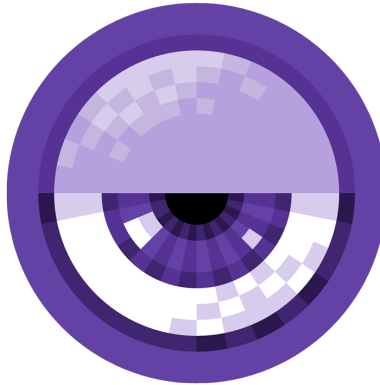
Release 0.9.0

Joshua Skelton, Felix Siebeneicker

Jul 10, 2018

Contents:

1	Features	3
2	TwitchChatObserver	5
3	TwitchChatEvent	9
4	BadTwitchChatEvent	11
5	TwitchChatEventType	13
6	TwitchChatColor	15
7	Aliases	17
	Python Module Index	19



Turn Twitch chatter into Python events.

twitchobserver makes interacting with Twitch chat super easy. It is built and tuned for realtime applications. You can make chatbots chat. You can build Twitch Plays video games.

CHAPTER 1

Features

- *Pure Python*: No extra dependencies. Just plain and simple Python.
- *Small API*: With a few classes and a handful of methods, you can learn it over a coffee break.
- *Event Based*: Makes writing apps easy and straightforward.
- *Context Manager*: Further simplifies working with observers.

TwitchChatObserver

class twitchobserver.twitchobserver.**TwitchChatObserver** (*nickname, password*)

Class for watching a Twitch channel. Creates events for various chat messages.

Parameters

- **nickname** – The user nickname to connect to the channel as
- **password** – The OAuth token to authenticate with

add_moderator (*channel, nickname*)

Adds a user to the list of chat moderators.

Parameters

- **channel** – The channel name
- **nickname** – The nickname of the user

ban_user (*user, channel*)

Bans a user from a channel.

Parameters

- **user** – The user name
- **channel** – The channel name

change_name_color (*color*)

Changes the color of your name in chat.

Parameters **color** – The new color of the name. Can be a hex value like #000000 or one of the colors defined in *colors.Color*

clear_chat_history (*channel*)

Clears the chat history of a channel.

Parameters **channel** – The channel name

emoteonly_mode (*channel, enable=True*)

Manages the emote-only mode of the chat.

Parameters

- **channel** – The channel name
- **enable** – Boolean to decide whether to enable or disable the emote-only mode (defaults to True)

followers_only_mode (*channel, enable=True*)

Manages the followers-only mode of the chat.

Parameters

- **channel** – The channel name
- **enable** – Boolean to decide whether to enable or disable the followers-only mode (defaults to True)

get_events ()

Returns a sequence of events since the last time called.

Returns A sequence of TwitchChatEvents

ignore_user (*nickname, channel*)

Ignores a particular user by not showing his messages in chat.

Parameters

- **nickname** – The nickname of the user
- **channel** – The channel name

join_channel (*channel*)

Joins a channel.

Parameters **channel** – The channel name

leave_channel (*channel*)

Leaves a channel.

Parameters **channel** – The channel name

list_moderators (*channel*)

Lists all moderators of a given channel.

Parameters **channel** – The channel name

on_event (*event_type*)

Decorator for event handlers based on the event type.

The handler needs to take an event as argument.

Parameters **event_type** – A ChatEventType to listen for

Usage:

```
>>> @observer.on_event(ChatEventType.TWITCHCHATJOIN)
>>> def handle_join_event(event):
>>>     print(event.nickname + " joined")
```

r9k_mode (*channel, enable=True*)

Manages the r9k mode of the chat where messages with more than nine characters are checked for uniqueness.

Parameters

- **channel** – The channel name

- **enable** – Boolean to decide whether to enable or disable the r9k mode (defaults to True)

remove_moderator (*channel, nickname*)

Removes a user from the list of chat moderators.

Parameters

- **channel** – The channel name
- **nickname** – The nickname of the user

send_message (*message, channel*)

Sends a message to a channel.

Parameters

- **message** – The message to be send
- **channel** – The channel name

send_whisper (*user, message*)

Sends a whisper (private message) to a user.

Parameters

- **user** – The user name
- **message** – The message to be whispered

set_marker (*channel*)

Sets a stream marker for the Twitch Highlight feature.

Parameters **channel** – The channel name

slow_mode (*channel, duration=10, enable=True*)

Manages the slow mode of the chat, i.e. each message of a user needs to have *duration* of seconds in between them.

Parameters

- **channel** – The channel name
- **duration** – The duration of forced time between two messages (in seconds, defaults to 10)
- **enable** – Boolean to decide whether to enable or disable the slow mode (defaults to True)

start ()

Starts the observer.

Attempt to connect to the Twitch channel with the given nickname and password. If successful a worker thread will be started to handle socket communication.

Raises RuntimeError – If authentication fails

stop (*force_stop=False*)

Stops the observer.

Stop watching the channel, shutdown the socket, and stop the worker threads once all of the outbound events have been sent.

Parameters **force_stop** – Immediately stop and do not wait for remaining outbound events to be sent.

subscribe (*callback*)

Receive events from watched channel.

Parameters **callback** – A function that takes a single TwitchChatEvent argument

subscribers_only_mode (*channel, enable=True*)

Manages the subscribers-only mode of the chat where only the streamer, moderators and subscribers can write messages.

Parameters

- **channel** – The channel name
- **enable** – Boolean to decide whether to enable or disable the subscribers-only mode (defaults to True)

timeout_user (*nickname, channel, duration=600*)

Timeouts a user by deleting all previously send messages and letting him not be able to send messages for ten minutes (default) or for *duration* of seconds.

Parameters

- **nickname** – The nickname of the user
- **channel** – The channel name
- **duration** – The duration to ban the user from chat (in seconds). Defaults to ten minutes

unban_user (*user, channel*)

Unbans a user from a channel.

Parameters

- **user** – The user name
- **channel** – The channel name

unignore_user (*nickname, channel*)

Unignores a particular user by showing his messages in chat again.

Parameters

- **nickname** – The nickname of the user
- **channel** – The channel name

unsubscribe (*callback*)

Unsubscribe a callback from the observer.

Parameters **callback** – A function that has been subscribed prior

TwitchChatEvent

```
class twitchobserver.twitchobserver.TwitchChatEvent (channel=None, command=None, message="")
```

A class for representing Twitch chat events.

Parameters

- **type** – Type of the event. Always will be ‘TWITCHCHATEVENT’
- **nickname** – The nickname of the user
- **channel** – The name of the channel
- **command** – The command sent. Will be one of the following: JOIN, PART or PRIVMSG
- **message** – The message sent by the user

dumps ()

Dump the event to a readable string for the socket commutation.

CHAPTER 4

BadTwitchChatEvent

```
class twitchobserver.twitchobserver.BadTwitchChatEvent  
    A class that is raised if a bad Twitch chat event occurred.
```

TwitchChatEventType

```
class twitchobserver.twitchobserver.TwitchChatEventType
```

```
    A collection of twitch chat event types.
```

```
    TWITCHCHATCLEARCHAT = 'TWITCHCHATCLEARCHAT'
```

```
    TWITCHCHATCOMMAND = 'TWITCHCHATCOMMAND'
```

```
    TWITCHCHATHOSTTARGET = 'TWITCHCHATHOSTTARGET'
```

```
    TWITCHCHATJOIN = 'TWITCHCHATJOIN'
```

```
    TWITCHCHATLEAVE = 'TWITCHCHATLEAVE'
```

```
    TWITCHCHATMESSAGE = 'TWITCHCHATMESSAGE'
```

```
    TWITCHCHATMODE = 'TWITCHCHATMODE'
```

```
    TWITCHCHATNOTICE = 'TWITCHCHATNOTICE'
```

```
    TWITCHCHATRECONNECT = 'TWITCHCHATRECONNECT'
```

```
    TWITCHCHATROOMSTATE = 'TWITCHCHATROOMSTATE'
```

```
    TWITCHCHATUSERNOTICE = 'TWITCHCHATUSERNOTICE'
```

```
    TWITCHCHATUSERSTATE = 'TWITCHCHATUSERSTATE'
```

```
    TWITCHCHATWHISPER = 'TWITCHCHATWHISPER'
```


CHAPTER 6

TwitchChatColor

```
class twitchobserver.twitchobserver.TwitchChatColor
    A collection of colors for the chat name.

    BLUE = 'Blue'
    BLUEVIOLET = 'BlueViolet'
    CADETBLUE = 'CadetBlue'
    CHOCOLATE = 'Chocolate'
    CORAL = 'Coral'
    DODGERBLUE = 'DodgerBlue'
    FIREBRICK = 'Firebrick'
    GOLDENROD = 'GoldenRod'
    GREEN = 'Green'
    HOTPINK = 'HotPink'
    ORANGERED = 'OrangeRed'
    RED = 'Red'
    SEAGREEN = 'SeaGreen'
    SPRINGGREEN = 'SpringGreen'
    YELLOWGREEN = 'YellowGreen'
```


There exist the following aliases:

`twitchobserver.Observer`
alias of `twitchobserver.twitchobserver.TwitchChatObserver`

`twitchobserver.ChatEvent`
alias of `twitchobserver.twitchobserver.TwitchChatEvent`

`twitchobserver.ChatEventType`
alias of `twitchobserver.twitchobserver.TwitchChatEventType`

`twitchobserver.ChatColor`
alias of `twitchobserver.twitchobserver.TwitchChatColor`

t

`twitchobserver`, [17](#)

`twitchobserver.twitchobserver`, [3](#)

A

`add_moderator()` (twitchobserver.twitchobserver.TwitchChatObserver method), 5

B

`BadTwitchChatEvent` (class in twitchobserver.twitchobserver), 11

`ban_user()` (twitchobserver.twitchobserver.TwitchChatObserver method), 5

`BLUE` (twitchobserver.twitchobserver.TwitchChatColor attribute), 15

`BLUEVIOLET` (twitchobserver.twitchobserver.TwitchChatColor attribute), 15

C

`CADETBBLUE` (twitchobserver.twitchobserver.TwitchChatColor attribute), 15

`change_name_color()` (twitchobserver.twitchobserver.TwitchChatObserver method), 5

`ChatColor` (in module twitchobserver), 17

`ChatEvent` (in module twitchobserver), 17

`ChatEventType` (in module twitchobserver), 17

`CHOCOLATE` (twitchobserver.twitchobserver.TwitchChatColor attribute), 15

`clear_chat_history()` (twitchobserver.twitchobserver.TwitchChatObserver method), 5

`CORAL` (twitchobserver.twitchobserver.TwitchChatColor attribute), 15

D

`DODGERBLUE` (twitchobserver.twitchobserver.TwitchChatColor attribute), 15

`dumps()` (twitchobserver.twitchobserver.TwitchChatEvent method), 9

E

`emoteonly_mode()` (twitchobserver.twitchobserver.TwitchChatObserver method), 5

F

`FIREBRICK` (twitchobserver.twitchobserver.TwitchChatColor attribute), 15

`followers_only_mode()` (twitchobserver.twitchobserver.TwitchChatObserver method), 6

G

`get_events()` (twitchobserver.twitchobserver.TwitchChatObserver method), 6

`GOLDENROD` (twitchobserver.twitchobserver.TwitchChatColor attribute), 15

`GREEN` (twitchobserver.twitchobserver.TwitchChatColor attribute), 15

H

`HOTPINK` (twitchobserver.twitchobserver.TwitchChatColor attribute), 15

I

`ignore_user()` (twitchobserver.twitchobserver.TwitchChatObserver method), 6

J

`join_channel()` (twitchobserver.twitchobserver.TwitchChatObserver method), 6

L

`leave_channel()` (twitchobserver.twitchobserver.TwitchChatObserver method), 6

`list_moderators()` (twitchobserver.twitchobserver.TwitchChatObserver method), 6

O

`Observer` (in module twitchobserver), 17

`on_event()` (twitchobserver.twitchobserver.TwitchChatObserver method), 6

`ORANGERED` (twitchobserver.twitchobserver.TwitchChatColor attribute), 15

R

`r9k_mode()` (twitchobserver.twitchobserver.TwitchChatObserver method), 6

`RED` (twitchobserver.twitchobserver.TwitchChatColor attribute), 15

`remove_moderator()` (twitchobserver.twitchobserver.TwitchChatObserver method), 7

S

`SEAGREEN` (twitchobserver.twitchobserver.TwitchChatColor attribute), 15

`send_message()` (twitchobserver.twitchobserver.TwitchChatObserver method), 7

`send_whisper()` (twitchobserver.twitchobserver.TwitchChatObserver method), 7

`set_marker()` (twitchobserver.twitchobserver.TwitchChatObserver method), 7

`slow_mode()` (twitchobserver.twitchobserver.TwitchChatObserver method), 7

`SPRINGGREEN` (twitchobserver.twitchobserver.TwitchChatColor attribute), 15

`start()` (twitchobserver.twitchobserver.TwitchChatObserver method), 7

`stop()` (twitchobserver.twitchobserver.TwitchChatObserver method), 7

`subscribe()` (twitchobserver.twitchobserver.TwitchChatObserver method), 7

`subscribers_only_mode()` (twitchobserver.twitchobserver.TwitchChatObserver method), 8

T

`timeout_user()` (twitchobserver.twitchobserver.TwitchChatObserver method), 8

`TWITCHCHATCLEARCHAT` (twitchobserver.twitchobserver.TwitchChatEventType attribute), 13

`TwitchChatColor` (class in twitchobserver.twitchobserver), 15

`TWITCHCHATCOMMAND` (twitchobserver.twitchobserver.TwitchChatEventType attribute), 13

`TwitchChatEvent` (class in twitchobserver.twitchobserver), 9

`TwitchChatEventType` (class in twitchobserver.twitchobserver), 13

`TWITCHCHATHOSTTARGET` (twitchobserver.twitchobserver.TwitchChatEventType attribute), 13

`TWITCHCHATJOIN` (twitchobserver.twitchobserver.TwitchChatEventType attribute), 13

`TWITCHCHATLEAVE` (twitchobserver.twitchobserver.TwitchChatEventType attribute), 13

`TWITCHCHATMESSAGE` (twitchobserver.twitchobserver.TwitchChatEventType attribute), 13

`TWITCHCHATMODE` (twitchobserver.twitchobserver.TwitchChatEventType attribute), 13

`TWITCHCHATNOTICE` (twitchobserver.twitchobserver.TwitchChatEventType attribute), 13

`TwitchChatObserver` (class in twitchobserver.twitchobserver), 5

`TWITCHCHATRECONNECT` (twitchobserver.twitchobserver.TwitchChatEventType attribute), 13

`TWITCHCHATROOMSTATE` (twitchobserver.twitchobserver.TwitchChatEventType attribute), 13

`TWITCHCHATUSERNOTICE` (twitchobserver.twitchobserver.TwitchChatEventType attribute), 13

`TWITCHCHATUSERSTATE` (twitchobserver.twitchobserver.TwitchChatEventType attribute), 13

`TWITCHCHATWHISPER` (twitchobserver.twitchobserver.TwitchChatEventType attribute), 13

`twitchobserver` (module), 17

`twitchobserver.twitchobserver` (module), 3

U

`unban_user()` (twitchobserver.twitchobserver.TwitchChatObserver method), 8

`unignore_user()` (twitchobserver.twitchobserver.TwitchChatObserver method), 8

`unsubscribe()` (twitchobserver.twitchobserver.TwitchChatObserver method), 8

Y

`YELLOWGREEN` (twitchobserver.twitchobserver.TwitchChatColor attribute), 15