twitch.json

misinformation

Page ID: 73

Source: https://dev.twitch.tv/docs/irc/

['authentic']

Chat & Chatbots | Twitch Developers Asset 20 Products Showcase Support Blog Docs Tutorials Log in with Twitch Contents Overview Twitch API EventSub Chat & Chatbots Overview Normal message flow Supported IRC messages Connecting to the Twitch IRC Server Keepalive messages Parsing messages Phone Verification Rate limits Verified bots

Next steps Getting Started Authenticating with the Server Requesting Twitch Capabilities Joining a Chat Room Send and Receive Messages Using Twitch Chat Commands Getting Emotes Twitch IRC Commands Membership Commands Twitch IRC Tags NOTICE Message IDs Example Chatbot Example Message Parser Authentication Organizations Drops Embedding Twitch Extensions Insights & Analytics Mobile Deep Links PubSub Video Broadcast Twitch CLI Changelog Product Lifecycle Twitch Chat & Chatbots Twitch provides an Internet Relay Chat (IRC) interface that lets chatbots connect to Twitch chat rooms using a WebSocket or TCP connection.

Once connected, bots can send and receive chat messages.

For example, bots can provide simple reminders like get up and move or hydrate , or they can perform Twitch actions like banning a user, or they can react to user input.

For a simple example to get you started quickly, see Getting Started .

Twitch’s IRC service is based on RFC1459 and IRCv3 Message Tag specification .

If you’re not already familiar with them, reading them may help you understand the Twitch IRC server.

While Twitch’s IRC server generally follows RFC1459, it doesn’t support all IRC messages.

For a list of supported messages, see Supported IRC messages .

Normal message flow After connecting to the server , the first messages that all bots must send are the PASS and NICK messages.

These messages are used to authenticate the user account that the bot is running under.

Once the server successfully authenticates your bot, the next step is to send a JOIN message to join the chat room that the bot runs in.

The messages your bot sends and receives depends on what your bot does and the Twitch-specific IRC capabilities it requests .

If your bot simply sends out get up and move reminders at specific intervals, your bot can mostly ignore all other messages from the server.

To send the reminder, your bot sends a PRIVMSG message (see Sending a message to the chat room ).

If your bot responds to messages that users post in the chat room, your bot will need to read and parse the PRIVMSG messages that the server sends; the server sends a PRIVMSG message for each message users post in the chat room.

For example, if your bot performs an action in response to a user command, it must parse the user’s posted message to see if it contains the command.

The Getting Started example does just this by looking for the !dice command, rolling the die, and sending a PRIVMSG message with the rolled number.

['authentic']

Supported IRC messages While Twitch’s IRC server generally follows RFC1459 , it doesn’t support all IRC messages.

The following is the list of IRC messages that Twitch supports; if it’s not listed here, Twitch doesn’t support it.

Message Message flow Description JOIN Send Your bot sends this message to join a channel.

See Joining a channel .

NICK

Send Your bot sends this message to specify the bot’s nickname when authenticating with the Twitch IRC server.

See Authenticating with the Twitch IRC server .

NOTICE Receive Your bot receives this message from the Twitch IRC server when your bot fails to authenticate with the server.

You can get NOTICE messages for other reasons if you request the commands capability .

PART Send Receive Your bot sends this message to leave a channel.

Your bot receives this message from the Twitch IRC server when a channel bans it.

PASS Send Your bot sends this message to specify the bot’s password when authenticating with the Twitch IRC server.

See Authenticating with the Twitch IRC server .

PING Receive Your bot receives this message from the Twitch IRC server when the server wants to ensure that your bot is still alive and able to respond to the server’s messages.

See Keepalive messages .

PONG Send Your bot sends this message in reply to the Twitch IRC server’s PING message.

See Keepalive messages .

PRIVMSG Send Receive Your bot sends this message to post a chat message in the channel’s chat room.

Your bot receives this message from the Twitch IRC server when a user posts a chat message in the chat room.

['authentic']

toString ()); }); client .

on ( 'connect' , function ( connection ) { console .

log ( 'WebSocket Client Connected' ); // Send CAP (optional), PASS, and NICK messages }); client .

connect ( 'ws://irc-ws.chat.twitch.tv:80' );

If the connection succeeds, the next step is to request Twitch-specific capabilities if you want to use Twitch’s optional capabilities.

Otherwise, the next step is to authenticate your bot with the Twitch IRC server.

See Authenticating with the Twitch IRC Server .

Reconnecting to the Twitch IRC server

If your connection is dropped, you should try reconnecting using an exponential backoff approach.

For example, try reconnecting immediately.

If you have no luck, try again in 1 second, 2 seconds, 4 seconds, 8 seconds and so on for the number of attempts you want to make.

But be aware if you’re making multiple connections that there are rate limits that apply (see Rate limits ).

Keepalive messages The Twitch IRC server sends PING messages to ensure that your bot is still alive and able to respond to the server’s messages.

['authentic']

Limit Description 20 messages per 30 seconds If the user isn’t the channel’s broadcaster or moderator, the bot may send a maximum of 20 messages per 30 seconds.

100 messages per 30 seconds If the user is the channel’s broadcaster or moderator, the bot may send a maximum of 100 messages per 30 seconds.

7,500 messages per 30 seconds The bot is limited to sending 7,500 messages per 30 seconds across all channels.

This means that the bot could send 10 messages per 30 seconds to 750 different channels.

However, it couldn’t, for example, send 7,500 messages per 30 seconds to a single channel because it would exceed the 20 messages per 30 second limit.

Authentication and join rate limits The following lists show the rate limits for the number of authentication and join attempts.

A bot sending a pair of PASS and NICK messages is considered an authentication attempt.

The following table shows the rate limits for a normal account.

20 authentication attempts per 10 seconds per user.

20 join attempts per 10 seconds per user.

The following table shows the rate limits for a verified account.

200 authentication attempts per 10 seconds per user.

2,000 join attempts per 10 seconds per user.

Verified bots

As a chatbot grows in popularity, it’s likely that it may approach or exceed the rate limits .

Chatbots that enhance the Twitch user experience and have reached these limits may apply for verified bot status, but note that verified bot status is rarely granted.

A bot with verified bot status enjoys: Higher chat message limits than regular Twitch accounts.

Higher authentication and join limits than regular Twitch accounts.

But they: Don’t have higher whisper limits than regular Twitch accounts.

Aren’t exempt from AutoMod mode.

AutoMod analyzes chat messages and flags potentially risky messages for a channel moderator, who can then allow them to or prevent them from appearing in chat.

Requesting Verified Bot status If a chatbot has reached the rate limits for messages, authentications, or joins; the bot’s developer may request verified bot status.

To request verified bot status, go to IRC Command and Message Rate and fill out the form.

After Twitch reviews the request, Twitch sends its determination to the requestor via email.

Next steps Try the Getting Started example to get your first chatbot running in a matter of minutes.

Learn how to authenticate your bot user with the Twitch IRC server (see Authenticating with the Twitch IRC Server ).

Get familiar with the Internet Relay Chat (IRC) interface by reading the RFC1459 and IRCv3 Message Tag specifications.

For help, post questions in the chat category on the Twitch developer forums or join the TwitchDev Discord server .

Provide feedback for this page Docs Status Support Showcase Blog

What We Do Stream Watch

Develop Advertise twitch.tv Company Jobs Merch Brand TwitchCon Meetups Newsroom News Press Products Bits Subs Turbo Prime Extensions Sings Resources Legal Help Center Security Connect Twitter Facebook Instagram Terms of service Privacy Policy Ad Choices Cookie Policy Partners Affiliates

© 2023 Twitch Interactive, Inc.