twitch.json

misinformation

Page ID: 99

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

['authentic']

PubSub | Twitch Developers Asset 20 Products Showcase Support Blog Docs Tutorials Log in with Twitch Contents Overview Twitch API EventSub Chat & Chatbots Authentication Organizations Drops Embedding Twitch Extensions Insights & Analytics Mobile Deep Links PubSub Guide Introduction Terminology Connection Management API Limits Topics Receiving Messages Video Broadcast Twitch CLI Changelog Product Lifecycle PubSub Guide Introduction PubSub enables you to subscribe to a topic, for updates (e.g. when a user cheers in a channel).

The Twitch PubSub system allows back-end services to broadcast realtime messages to clients.

Example applications include: An instant messaging service sending instant messages between friends.

A back-end video system pushing real-time viewer count updates to video players.

A presence system broadcasting users’ online status to all their friends.

These example applications share a common pattern: on application load, the application fetches a complete snapshot of its state and uses a persistent connection through PubSub to receive updates to this state.

The updates act as “diffs” to the initial state.

['authentic']

Malicious or careless applications that result in abnormally high server load may be denylisted from establishing connections.

If clients are slow to read messages from their connection and the server is simultaneous buffering more than 30 messages to an individual client, the client will be disconnected.

Topics Once a client establishes a connection, it can LISTEN on topics it cares about.

Clients can LISTEN on many topics at once and add new topics at any time by issuing new LISTEN commands.

Once a client no longer cares about a particular set of topics, it should issue UNLISTEN commands to stop receiving messages on those topics.

Authentication All topics require an OAuth token, but only some topics have a specific required scope (noted in the table below).

Available Topics All topics require an OAuth token, but only some topics have a specific required scope (noted in the table).

Feature Topic and Example

Required Scope You are notified when… Bits channel-bits-events-v1.<channel\_id> Example: channel-bits-events-v1.44322889 bits:read Anyone cheers in a specified channel.

Bits channel-bits-events-v2.<channel\_id>

Example: channel-bits-events-v2.4602499 bits:read Anyone cheers in a specified channel.

Bits Badge Notification channel-bits-badge-unlocks.<channel\_id> Example: channel-bits-badge-unlocks.44322889 bits:read Message sent when a user earns a new Bits badge in a particular channel, and chooses to share the notification with chat.

['authentic', 'trust']

Subgift subscription messages contain recipient information.

Chat automod-queue.<moderator\_id>.<channel\_id> Example: automod-queue.46024993.44322889 channel:moderate AutoMod flags a message as potentially inappropriate, and when a moderator takes action on a message.

Chat chat\_moderator\_actions.<user\_id>.<channel\_id>

Example: chat\_moderator\_actions.46024993.44322889 channel:moderate Supports moderators listening to the topic, as well as users listening to the topic to receive their own events.

Examples of moderator actions are bans, unbans, timeouts, deleting messages, changing chat mode (followers-only, subs-only), changing AutoMod levels, and adding a mod.

Chat low-trust-users.<channel\_id>.<suspicious\_user\_id>

Example: low-trust-users.12826.141981764 channel:moderate The broadcaster or a moderator updates the low trust status of a user, or a new message has been sent in chat by a potential ban evader or a bans shared user.

Chat user-moderation-notifications.<current\_user\_id>.<channel\_id> Example: user-moderation-notifications.46024993.44322889 chat:read A user’s message held by AutoMod has been approved or denied.

Whispers whispers.<user\_id>

Example: whispers.44322889 whispers:read Anyone whispers the specified user.

Where: <channel ID> or <user ID> is the id field returned by the Twitch API Get Users endpoint for the respective user(s) Note: channel-bitsevents is deprecated.

For Bits events, use channel-bits-events-v2 instead.

Note: channel-commerce-events-v1.<channel id> is deprecated.

Requests This is an example request for Bits events.

It listens to Bits events on channel 44322889.

The authorization scope is specified when you generate the OAuth token using our authorization flow; see the Apps & Authentication Guide .

//

Request from client to server { "type" : "LISTEN" , "nonce" : "44h1k13746815ab1r2" , "data" : { "topics" : [ "channel-bits-events-v1.44322889" ], "auth\_token" : "cfabdegwdoklmawdzdo98xt2fo512y" } } Request Parameter Type Description type string Valid values: LISTEN , UNLISTEN .

nonce string (Optional) Random string to identify the response associated with this request.

data JSON Wraps the topics and auth\_token fields.

topics array of strings List of topics to listen on.

Valid values are any of the topics listed in Available Topics .

['trust', 'false']

user\_name string Login name of the person who used the Bits.

version string Message version Example:

Bits Badge Notification Message { "type" : "MESSAGE" , "data" :{ "topic" : "channel-bits-badge-unlocks.401394874" , "message" : " { \" user\_id \" : \" 232889822 \" , \" user\_name \" : \" willowolf \" , \" channel\_id \" : \" 401394874 \" , \" channel\_name \" : \" fun\_test12345 \" , \" badge\_tier \" :1000, \" chat\_message \" : \" this should be received by the public pubsub listener \" , \" time \" : \" 2020-12-06T.71253159Z \" }" } } Field Type Description user\_id string ID of user who earned the new Bits badge user\_name string Login of user who earned the new Bits badge channel\_id string ID of channel where user earned the new Bits badge channel\_name string Login of channel where user earned the new Bits badge badge\_tier integer Value of Bits badge tier that was earned (1000, 10000, etc.)

chat\_message string [Optional] Custom message included with share time string Time when the new Bits badge was earned.

RFC 3339 format.

Example: Channel Points Event Message { "type" : "reward-redeemed" , "data" : { "timestamp" : "2019-11-12T.98329743Z" , "redemption" : { "id" : "9203c6f0-51b6-4d1d-a9ae-8eafdb0d6d47" , "user" : { "id" : "30515034" , "login" : "davethecust" , "display\_name" : "davethecust" }, "channel\_id" : "30515034" , "redeemed\_at" : "2019-12-11T.128421623Z" , "reward" : { "id" : "6ef17bb2-e5ae-432e-8b3f-5ac4dd774668" , "channel\_id" : "30515034" , "title" : "hit a gleesh walk on stream" , "prompt" : "cleanside's finest \n " , "cost" : 10 , "is\_user\_input\_required" : true , "is\_sub\_only" : false , "image" : { "url\_1x" : "https://static-cdn.jtvnw.net/custom-reward-images/30515034/6ef17bb2-e5ae-432e-8b3f-5ac4dd774668/7bcd9ca8-da17-42c9-800a-2f08832e5d4b/custom-1.png" , "url\_2x" : "https://static-cdn.jtvnw.net/custom-reward-images/30515034/6ef17bb2-e5ae-432e-8b3f-5ac4dd774668/7bcd9ca8-da17-42c9-800a-2f08832e5d4b/custom-2.png" , "url\_4x" : "https://static-cdn.jtvnw.net/custom-reward-images/30515034/6ef17bb2-e5ae-432e-8b3f-5ac4dd774668/7bcd9ca8-da17-42c9-800a-2f08832e5d4b/custom-4.png" }, "default\_image" : { "url\_1x" : "https://static-cdn.jtvnw.net/custom-reward-images/default-1.png" , "url\_2x" : "https://static-cdn.jtvnw.net/custom-reward-images/default-2.png" , "url\_4x" : "https://static-cdn.jtvnw.net/custom-reward-images/default-4.png" }, "background\_color" : "#00C7AC" , "is\_enabled" : true , "is\_paused" : false , "is\_in\_stock" : true , "max\_per\_stream" : { "is\_enabled" : false , "max\_per\_stream" : 0 }, "should\_redemptions\_skip\_request\_queue" : true }, "user\_input" : "yeooo" , "status" : "FULFILLED" } } } Field Type Description timestamp string Time the pubsub message was sent redemption object Data about the redemption, includes unique id and user that redeemed it channel\_id string ID of the channel in which the reward was redeemed.

redeemed\_at string Timestamp in which a reward was redeemed reward object Data about the reward that was redeemed user\_input (optional) string A string that the user entered if the reward requires input status string reward redemption status, will be FULFULLED if a user skips the reward queue, UNFULFILLED otherwise Example:

Channel Subscriptions Event Message

The data field is a JSON object that contains topic and message fields.

Typically, message is a JSON object that has been escaped (see “Example:

Bits Event Message”) and cast into a string.

Note:

The months field is deprecated.

We now have fields for cumulative-months and streak-months .

Following is an example of a sub/resub message: { "type" : "MESSAGE" , "data" : { "topic" : "channel-subscribe-events-v1.44322889" , "message" : { "user\_name" : "tww2" , "display\_name" : "TWW2" , "channel\_name" : "mr\_woodchuck" , "user\_id" : "13405587" , "channel\_id" : "89614178" , "time" : "2015-12-19T-08:00" , "sub\_plan" : "1000" , "sub\_plan\_name" : "Channel Subscription (mr\_woodchuck)" , "cumulative\_months" : 9 , "streak\_months" : 3 , "context" : "resub" , "is\_gift" : false , "sub\_message" : { "message" : "A Twitch baby is born!

KappaHD" , "emotes" : [ { "start" : 23 , "end" : 7 , "id" : 2867 } ] } } } } Here is an example of a subgift message, which contains recipient information.

{ "type" : "MESSAGE" , "data" : { "topic" : "channel-subscribe-events-v1.44322889" , "message" : { "user\_name" : "tww2" , "display\_name" : "TWW2" , "channel\_name" : "mr\_woodchuck" , "user\_id" : "13405587" , "channel\_id" : "89614178" , "time" : "2015-12-19T-08:00" , "sub\_plan" : "1000" , "sub\_plan\_name" : "Channel Subscription (mr\_woodchuck)" , "months" : 9 , "context" : "subgift" , "is\_gift" : true , "sub\_message" : { "message" : "" , "emotes" : null }, "recipient\_id" : "19571752" , "recipient\_user\_name" : "forstycup" , "recipient\_display\_name" : "forstycup" } } } Here is an example of a multi-month subgift message, which contains recipient information as well as the number of months gifted.

{ "type" : "MESSAGE" , "data" : { "topic" : "channel-subscribe-events-v1.44322889" , "message" : { "user\_name" : "tww2" , "display\_name" : "TWW2" , "channel\_name" : "mr\_woodchuck" , "user\_id" : "13405587" , "channel\_id" : "89614178" , "time" : "2015-12-19T-08:00" , "sub\_plan" : "1000" , "sub\_plan\_name" : "Channel Subscription (mr\_woodchuck)" , "months" : 9 , "context" : "subgift" , "is\_gift" : true , "sub\_message" : { "message" : "" , "emotes" : null }, "recipient\_id" : "19571752" , "recipient\_user\_name" : "forstycup" , "recipient\_display\_name" : "forstycup" , "multi\_month\_duration" : 6 } } } Here is an example of an anonsubgift message.

This is like subgift but contains no sender information, since the gifter is anonymous.

{ "type" : "MESSAGE" , "data" : { "topic" : "channel-subscribe-events-v1.44322889" , "message" : { "channel\_name" : "mr\_woodchuck" , "channel\_id" : "89614178" , "time" : "2015-12-19T-08:00" , "sub\_plan" : "1000" , "sub\_plan\_name" : "Channel Subscription (mr\_woodchuck)" , "months" : 9 , "context" : "anonsubgift" , "is\_gift" : true , "sub\_message" : { "message" : "" , "emotes" : null }, "recipient\_id" : "13405587" , "recipient\_user\_name" : "tww2" , "recipient\_display\_name" : "TWW2" } } } Field Type Description channel\_id string ID of the channel that has been subscribed or subgifted channel\_name string Name of the channel that has been subscribed or subgifted context string Event type associated with the subscription product, values: sub , resub , subgift , anonsubgift , resubgift , anonresubgift data JSON Wraps the topics and message fields user\_id string User ID of the person who subscribed or sent a gift subscription user\_name string Login name of the person who subscribed or sent a gift subscription display\_name string Display name of the person who subscribed or sent a gift subscription message string The body of the user-entered resub message.

Depending on the type of message, the message body contains different fields months int Cumulative number of months the gifter has giften in the channel (Deprecated) recipient\_id string User ID of the subscription gift recipient recipient\_user\_name string Login name of the subscription gift recipient recipient\_display\_name string Display name of the person who received the subscription gift sub\_plan string Subscription Plan ID, values: Prime , 1000 , 2000 , 3000 sub\_plan\_name string Channel Specific Subscription Plan Name time string Time when the subscription or gift was completed.

RFC 3339 format topic string The topic that the message pertains to type string Valid value: MESSAGE cumulative\_months string Cumulative number of tenure months of the subscription streak\_months string Denotes the user’s most recent (and contiguous) subscription tenure streak in the channel is\_gift bool

If this sub message was caused by a gift subscription multi\_month\_duration int Number of months gifted as part of a single, multi-month gift OR number of months purchased as part of a multi-month subscription Example: Multi-month Subscription Message Here is an example of a multi-month subscription message.

This message is published upon purchase, auto-renewal, or re-subscription for a multi-month, recurring subscription { "type" : "MESSAGE" , "data" : { "topic" : "channel-subscribe-events-v1.44322889" , "message" : { "user\_name" : "tww2" , "display\_name" : "TWW2" , "channel\_name" : "mr\_woodchuck" , "user\_id" : "13405587" , "channel\_id" : "89614178" , "time" : "2015-12-19T-08:00" , "sub\_plan" : "1000" , "sub\_plan\_name" : "Channel Subscription (mr\_woodchuck)" , "months" : 4 , "context" : "sub" , "is\_gift" : false , "sub\_message" : { "message" : "" , "emotes" : null }, "recipient\_id" : "19571752" , "recipient\_user\_name" : "forstycup" , "recipient\_display\_name" : "forstycup" , "multi\_month\_duration" : 6 } } } Example: AutoMod Queue Message { "type" : "MESSAGE" , "data" : { "topic" : "automod-queue.46024993.44322889" , "message" : { "type" : "automod\_caught\_message" , "data" : { "message" : { "id" : "some-automod-caught-message-id" , "content" : { "text" : "a bad word" , "fragments" : [ { "text" : "a bad word" , "automod" : { "topics" : { "swearing" : 6 } } } ] }, "sender" : { "user\_id" : "559969259" , "login" : "msongvo222" , "display\_name" : "msongvo222" , "chat\_color" : "#8A2BE2" }, "sent\_at" : "2021-05-24T.964Z" }, "content\_classification" : { "category" : "swearing" , "level" : 2 }, "status" : "PENDING" , "reason\_code" : "" , "resolver\_id" : "" , "resolver\_login" : "" } } } } Field Type Description id string Identifier of the message content object Object containing details about the message text string The text of the message fragments object Object defining the potentially inappropriate content of the message sender object Object representing the sender of the message user\_id integer User ID of the sender login string User login on the sender display\_name string User display name of the sender chat\_color string Chat color of the sender sent\_at string

Timestamp the message was sent in RFC3339 format.

content\_classification object Object defining the category and level that the content was classified as category string Category that the message was classified as level number The level that the message was classified as.

status string Current status of the message, can be “PENDING”, “ALLOWED”, “DENIED”, or “EXPIRED”.

reason\_code string Reserved for internal use resolver\_id string User ID of the moderator that resolved this message resolver\_login string User login of the moderator that resolved this message Example: Low Trust User Treatment Message { "low\_trust\_id" : "12826.141981764" , "channel\_id" : "12826" , "updated\_by" : { "id" : "12826" , "login" : "twitch" , "display\_name" : "Twitch" }, "updated\_at" : "2023-02-03TZ" , "target\_user\_id" : "141981764" , "target\_user" : "twitchdev" , "treatment" : "RESTRICTED" , "types" : [ "MANUALLY\_ADDED" ], "ban\_evasion\_evaluation" : "" , "evaluated\_at" : "2023-01-26TZ" } Field Type Description low\_trust\_id string An ID for the suspicious user entry, which is a combination of the channel ID where the treatment was updated and the user ID of the suspicious user.

channel\_id string ID of the channel where the suspicious user was present.

updated\_by object Information about the moderator who made any update for the suspicious user.

id string User ID of the moderator.

login string Login of the moderator.

display\_name string Display name of the moderator.

updated\_at string RFC3339 timestamp of when the treatment was updated for the suspicious user.

target\_user\_id string User ID of the suspicious user.

target\_user string Login of the suspicious user.

treatment string The treatment set for the suspicious user, can be “NO\_TREATMENT”, “ACTIVE\_MONITORING”, or “RESTRICTED” types array of strings User types (if any) that apply to the suspicious user, can be “UNKNOWN\_TYPE”, “MANUALLY\_ADDED”, “DETECTED\_BAN\_EVADER”, or “BANNED\_IN\_SHARED\_CHANNEL” ban\_evasion\_evaluation string A ban evasion likelihood value (if any) that as been applied to the user automatically by Twitch, can be “UNKNOWN\_EVADER”, “UNLIKELY\_EVADER”, “LIKELY\_EVADER”, or “POSSIBLE\_EVADER” evaluated\_at string If applicable, an RFC3339 timestamp for the first time the suspicious user was automatically evaluated by Twitch.

Example: Low Trust User Chat Message { "low\_trust\_user" : { "id" : "141981764" , "low\_trust\_id" : "NDU3NTcxMjM0LjgyNjA2NzQyNg" , "channel\_id" : "12826" , "sender" : { "user\_id" : "141981764" , "login" : "twitchdev" , "display\_name" : "TwitchDev" }, "evaluated\_at" : "2023-01-26TZ" , "updated\_at" : "2023-02-03TZ" , "ban\_evasion\_evaluation" : "UNKNOWN\_EVADER" , "treatment" : "RESTRICTED" , "updated\_by" : { "id" : "12826" , "login" : "twitch" , "display\_name" : "Twitch" }, "shared\_ban\_channel\_ids" : null , "types" : [ "MANUALLY\_ADDED" ] }, "message\_content" : { "text" : "hola HeyGuys" , "fragments" : [ { "text" : "hola " }, { "text" : "HeyGuys" , "emoticon" : { "emoticonID" : "30259" , "emoticonSetID" : "0" } } ] }, "message\_id" : "853093ce-17c5-4daa-b500-d1ce03115812" , "sent\_at" : "2023-02-03TZ" } Field Type Description low\_trust\_user object Information about the low trust message and suspicious user.

id string The user ID of the suspicious user.

low\_trust\_id string Unique ID for this low trust chat message.

channel\_id string ID of the channel where the suspicious user was present.

sender object Information about the suspicious user.

user\_id string

The user ID of the suspicious user.

login string The login of the suspicious user.

display\_name string The display name of the suspicious user.