/ Streaming APIs

English

Search

Overview

Streaming message types

Public stream messages

In addition to standard Tweet payloads, the following kinds of messages may be delivered on a stream. Note that this list may not be comprehensive—additional objects may be introduced into streams in the future. Ensure that your parser is tolerant of unexpected message formats.

When parsing Tweets, keep in mind that Retweets are streamed as a status with another status nested inside it. If you are matching Tweet fields using regular expressions, it is possible that you will match fields in the nested Tweet instead of the wrapper. As a rule of thumb it is better to use a JSON parser to extract information from message payloads.

Blank lines

On slow streams, some messages may be blank lines which serve as “keep-alive” signals to prevent clients and other network infrastructure from assuming the stream has stalled and closing the connection.

Status deletion notices (delete)

These messages indicate that a given Tweet has been deleted. Client code must honor these messages by clearing the referenced Tweet from memory and any storage or archive, even in the rare case where a deletion message arrives earlier in the stream that the Tweet it references.

```
{
 "delete":{
  "status":{
   "id":1234,
   "id_str":"1234",
   "user_id":3,
   "user_id_str":"3"
  }
 }
}
```

Location deletion notices (scrub_geo)

These messages indicate that geolocated data must be stripped from a range of Tweets. Clients must honor these messages by deleting geocoded data from Tweets which fall before the given status ID and belong to the specified user. These messages may also arrive before a Tweet which falls into the specified range, although this is rare.

```
{
 "scrub_geo":{
  "user_id":14090452,
  "user_id_str":"14090452",
  "up_to_status_id":23260136625,
  "up_to_status_id_str":"23260136625"
 }
}
```

Limit notices (limit)

These messages indicate that a filtered stream has matched more Tweets than its current rate limit allows to be delivered. Limit notices contain a total count of the number of undelivered Tweets since the connection was opened, making them useful for tracking counts of track terms, for example. Note that the counts do not specify which filter predicates undelivered messages matched.

```
{
 "limit":{
  "track":1234
 }
}
```

Withheld content notices (status_withheld, user_withheld)

These messages indicate that either the indicated tweet or indicated user has had their content withheld.See withheld content notices

status_withheld

These events contain an `id` field indicating the status ID, a `user_id` indicating the user, and a collection of `withheld_in_countries` uppercase two-letter country codes. This example illustrates a hypothetical tweet that has been withheld in Germany and Argentina.

```
{
 "status_withheld":{
   "id":1234567890,
   "user_id":123456,
   "withheld_in_countries":["DE", "AR"]
 }
}
```

user_withheld

These events contain an `id` field indicating the user ID and a collection of `withheld_in_countries` uppercase two-letter country codes. This example illustrates a hypothetical user who has been withheld in Germany and Argentina.

```
{
 "user_withheld":{
  "id":123456,
  "withheld_in_countries":["DE","AR"]
 }
}
```

Disconnect messages (disconnect)

Streams may be shut down for a variety of reasons. The streaming API will attempt to deliver a message indicating why a stream was closed. Note that if the disconnect was due to network issues or a client reading too slowly, it is possible that this message will not be received.

```
{
 "disconnect":{
  "code": 4,
  "stream_name":"",
  "reason":""
 }
}
```

The following table lists possible status codes and their meanings. Additional codes may be used without warning.

Code	Name	Description
1	Shutdown	The feed was shutdown (possibly a machine restart)
		The same endpoint was connected

2	Duplicate stream	too many times.
3	Control request	Control streams was used to close a stream (applies to sitestreams).
4	Stall	The client was reading too slowly and was disconnected by the server.
5	Normal	The client appeared to have initiated a disconnect.
6	Token revoked	An oauth token was revoked for a user (applies to site and userstreams).
7	Admin logout	The same credentials were used to connect a new stream and the oldest was disconnected.
8		Reserved for internal use. Will not be delivered to external clients.
9	Max message limit	The stream connected with a negative count parameter and was disconnected after all backfill was delivered.
10	Stream exception	An internal issue disconnected the stream.
11	Broker stall	An internal issue disconnected the stream.
12	Shed load	The host the stream was connected to became overloaded and streams were disconnected to balance load. Reconnect as usual.

Stall warnings (warning)

When connected to a stream using the `stall_warnings` parameter, you may receive status notices indicating the current health of the connection. See the stall_warnings documentation for more information.

```
{
 "warning":{
  "code":"FALLING_BEHIND",
  "message":"Your connection is falling behind and messages are being
queued for delivery to you. Your queue is now over 60% full. You will be
disconnected when the queue is full.",
  "percent_full": 60
 }
}
```

Note that in the case of Site Streams warning messages apply to the entire stream and will not be wrapped with a `for_user` envelope.

User update

Everytime a user updates their profile we broadcast a `user_update` event to indicate that an update has been made to the user profile. The source and target objects are identical in content.

```
{
  "created_at": "Tue Aug 06 02:23:21 +0000 2013",
  "source": {
    ...
  },
  "target": {
   ...
  },
  "event": "user_update"
}
```

User stream messages

Friends lists (friends)

Upon establishing a User Stream connection, Twitter will send a preamble before starting regular message delivery. This preamble contains a list of the user's friends. This is represented as an array of user ids, for example:

```
{
  "friends":[
   1497,
   169686021,
   790205,
   15211564,
   ...
  ]
}
```

This message will only be sent once per connection.

If the `stringify_friend_id` parameter is sent, the friends list preamble will be returned as string objects (instead of integer objects). This is particularly valuable if your language or library has difficulty with 64-bit integers, because as the number of Twitter users grows, user ids will eventually exceed the 32-bit integer threshold. If the parameter is used, the `friends` array specified above will not be sent, and the `friends_str` array will be sent in its place. For example:

```
{
  "friends_str": [
   "1497",
   "169686021",
   "790205",
   "15211564",
   ...
  ]
}
```

Direct Messages

DM representations are streamed to both sender and recipient, but perspectival attributes (`following`, `follow_request_sent`, and `notifications`) always return false.

Events (event)

Notifications about non-Tweet events are also sent over a user stream. These generally have the form of:

```
{
  "event":"EVENT_NAME",
  "created_at": "Sat Sep 4 16:10:54 +0000 2010",
  "target": TARGET_USER,
  "source": SOURCE_USER,
  "target_object": TARGET_OBJECT
}
```

The values present will be different based on the type of event. The following types of events are streamed:

Description	Event Name	Source	Target	Target Object
User deauthorizes stream	`access_revoked` *	Deauthorizing user	App owner	`client_applicatio`
User blocks someone	`block`	Current user	Blocked user	Null
User removes a block	`unblock`	Current user	Unblocked user	Null
User likes a Tweet	`favorite`	Current user	Tweet author	Tweet
User's Tweet is liked	`favorite`	Liking user	Current user	Tweet
User unlikes a Tweet	`unfavorite`	Current user	Tweet author	Tweet
User's Tweet is unliked	`unfavorite`	Unliking user	Current user	Tweet
User follows someone	`follow`	Current user	Followed user	Null
User is followed	`follow`	Following user	Current user	Null
User unfollows someone	`unfollow`	Current user	Followed user	Null
User creates			Current	

a list	list_created	Current user	user	List
User deletes a list	list_destroyed	Current user	Current user	List
User edits a list	list_updated	Current user	Current user	List
User adds someone to a list	list_member_added	Current user	Added user	List
User is added to a list	list_member_added	Adding user	Current user	List
User removes someone from a list	list_member_removed	Current user	Removed user	List
User is removed from a list	list_member_removed	Removing user	Current user	List
User subscribes to a list	list_user_subscribed	Current user	List owner	List
User’s list is subscribed to	list_user_subscribed	Subscribing user	Current user	List
User unsubscribes from a list	list_user_unsubscribed	Current user	List owner	List
User’s list is unsubscribed from	list_user_unsubscribed	Unsubscribing user	Current user	List
User’s Tweet is quoted	quoted_tweet	quoting User	Current User	Tweet
User updates			Current	

their profile	`user_update` †	Current user	user	Null
User updates their protected status	`user_update` †	Current user	Current user	Null

* *For a user stream, if the user on behalf of whom the app is running deauthorizes the app, the stream will simply disconnect. For a site stream, if the deauthorizing user is the last remaining user on behalf of whom the app is running, the stream will send a disconnect message with code 6 and then close the connection.*

† *When a change triggering a* `user_update` *event in the stream is made by a user, the stream returns complete profile information for the user, not just the changed values. At present, no data is returned indicating what values have changed; therefore, if your application requires such updates, you are advised to cache previous user profile information and compare* `user_update` *data against it.*

Too many follows (warning)

User and Site Streams following graph size will be capped at 10,000 accounts for each connected user. If your application connects on behalf of a user who follows more than 10,000 accounts, the followings list for the connected user will be truncated and this message will be sent over the stream:

```
{
 "warning": {
  "code": "FOLLOWS_OVER_LIMIT",
  "message": "The requested user follows more accounts than the maximum supported by this streaming endpoint. Only a subset of 10000 followed accounts are included in this stream.",
  "user_id": <user_id>
 }
}
```

The connected user's Tweets, @replies, and social events for likes and retweets will always be streamed. However, the 10,000 accounts that will be included are a random subset of the accounts the connected user follows. Any with=followings connections will only stream content from users in the truncated list. The IDs delivered via the Control Streams friends/ids.json endpoint will also only include IDs from users in the truncated list. If your application requires a full list of followings, please resort to the REST API.

Note that in the case of Site Streams warning messages apply to the entire stream and will not be wrapped with a `for_user` envelope.

Site stream messages

Envelopes (for_user)

Site Streams are sent the same messages as User Streams (including friends lists in the preamble), but for multiple users instead of a single user. The same types of messages are streamed, but to identify the target of each message, an additional wrapper is placed around every message, except for blank keep-alive lines. The Site Streams messages for two friends lists would look like:

```
{
 "for_user":1888,
 "message":{"friends":[]}
}
{
 "for_user":9160152,
 "message":{"friends":[]}
}
```

If a message should be routed to multiple users, multiple wrapped messages will be sent, each with a different `for_user` value.

Note that warning messages apply to the entire stream and will not be wrapped with a `for_user` envelope.

By default the user id is given as an integer, but you can also get the user id as a string by giving stringify_friend_ids as a parameter when connecting to the site stream. With stringify_friend_ids set to true, the example above would look like:

```
{
 "for_user_str":"1888",
 "message":{"friends":[]}
}
{
 "for_user_str":"9160152",
 "message":{"friends":[]}
}
```

Control messages (control)

New Site Streams connections will receive a control message which may be used to modify the Site Streams connection without reconnecting. See Control Streams for Site Streams for details. Note that this message will not necessarily be the first message delivered on a Site Streams connection.

```
 {
  "control":{
   "control_uri":
 "/1.1/site/c/01_225167_334389048B872A533002B34D73F8C29FD09EFC50"
  }
 }
```

Advertise Businesses Developers