



English



User Streams

# User streams

1. [Endpoint](#)
2. [Connections](#)
3. [User Stream messages](#)
  1. [Types of messages](#)
  2. [Data from accounts the user follows](#)
  3. [Replies](#)
  4. [Direct messages](#)
4. [Best Practices](#)
  1. [User streams and the REST API](#)
  2. [filtering Tweets for display](#)

## Overview

User Streams provide a stream of data and events specific to the authenticated user. Note that User Streams are not intended for server-to-server connections. If you need to make connections on behalf of multiple users from the same machine, consider using [site streams](#).

## Endpoint

- [GET user](#)

## Connections

Minimize the number of connections your application makes to User Streams. Each Twitter account is limited to only a few simultaneous User Streams connections per OAuth application, regardless of IP. Once the per-application limit is exceeded, the oldest connection will be terminated. An account logging in from too many instances of the same OAuth application will cycle connections as the application instances reconnect and disconnect each other.

Your application must handle the HTTP 420 error code that indicates that the account has been logging in too often. This response will be the indication that the account has exceeded the per-account-per-application connection limit described above. Indicate to the user that they have been automatically and temporarily banned from User Streams for an excessive login rate. The user should be advised to shut down extra copies of the application, perhaps instances running on another device, to restore streaming access.

Note that each application has its own allocation, so that logging in from App1 won't affect App2, and visa versa, but running too many copies of either App1 or App2 will cause problems. Also note that the number of simultaneous connections per IP address is still limited, regardless of application.

Applications that support simultaneous display of multiple Twitter accounts may open a connection per account.

## User streams messages

### Types of messages

There are some streaming messages which are not returned by the REST APIs. See [streaming message types](#) for information.

## Data from accounts the user follows

The `with` parameter controls the types of messages received. The default for User Streams is `with=followings`, which gives data about the user and about the user's followings (accounts which the authenticated user follows). The setting `with=user` sends only events about the user, and not events about their followings.

## Replies

By default @replies are only sent from mutual followings. All @replies by followings can be enabled with the `replies=all` parameter. For example, if Alice follows Bob, but Alice doesn't follow Carol, by default if Bob @replies Carol, Alice does not see the tweet. The default behavior mimics twitter.com and api.twitter.com behavior. If an application wishes to display all

@replies, or a filtered subset of @replies, use `replies=all`.

## Direct messages

Direct messages still require the proper permissions. See [the Application Permission Model](#).

## Best practices

### User streams and the REST API

An application will have the most versatility if it consumes both User Streams and the REST API. For example, a mobile app which switches from a cellular network to WiFi may choose to transition between polling the REST API for unstable connections, and connecting to a User Stream to improve performance. An application which connects to a User Stream may also backfill from the REST API to cover disconnected periods.

When transitioning from REST to a User Stream, perform a final REST API poll after a successful connection is established to ensure that there is no data loss.

In the event of a disconnect from the User Stream, attempt to reconnect for at least a minute or two before resuming REST API polling. This delay will help prevent a flood of REST API requests in the event of a minor hiccup on the User Streams API.

Do not attempt to use up your entire REST API rate limit while streaming – continue treat the REST API as a scarce resource.

### Filtering Tweets for display

Maintain a set of the user's followings to determine the contents of the home timeline. Apply follow social events to this set to keep the client set synchronized with the server side set. Expect retweets and mentions of the connected user. If you have DM access, expect direct messages from and to the connected user. All remaining tweets are the result of track and location filters, if specified.

Honor deletion messages. If a following deletes a tweet, you must remove the tweet from display and any application/server storage.

Consider obtaining a list of blocked users from the REST API regularly using [GET blocks / ids](#). Blocked accounts are filtered on the server-side in the home timeline. There is no need to fetch or apply the block list if only displaying the home timeline or the user timeline. Track results and @mentions, are not filtered for blocked accounts. Consider optionally grabbing the block

list to filter blocked accounts from track results. The block list should be cached locally - the change velocity is usually quite low. If possible, save the cache to disk to avoid polling for the block list upon startup. An update interval of 6 to 24 hours would be reasonable.

[SOLUTIONS](#)[Customer ServiceDigs](#)[Build Great Apps Sign in with](#)[Tell Great Stories Twitter](#)[Twitter Social](#)[Crashlytics](#)[Answers by](#)[Crashlytics](#)[Beta by](#)[Crashlytics](#)[MoPub](#)[FABRIC](#)[DATA & WEB](#)[Gnip](#)[REST APIs](#)[Streaming APIs](#)[Twitter for](#)[Websites](#)[RESOURCES](#)[Documentation](#)[Forums](#)[Blog](#)[Case Studies](#)[API Terms](#)[Policy Support](#)[PROGRAMS](#)[Official Partner](#)[Events](#)[Flight 2014](#)[TOOLS](#)[API Status](#)[API Console](#)[Manage Your](#)[Apps](#)[Cards Validator](#)[Subscribe to Our Newsletter](#)

Copyright © 2016 Twitter, Inc.

*All Rights Reserved*

---

[About](#) [Company](#) [Blog](#) [Help](#) [Status](#) [Jobs](#) [Terms](#) [Privacy](#) [Cookies](#) [Ads info](#) [Brand](#)  
[Advertise](#) [Businesses](#) [Developers](#)

© 2016 Twitter, Inc.