

The Twitter API and Authentication

INTERMEDIATE IMPORTING DATA IN PYTHON



Hugo Bowne-Anderson
Data Scientist at DataCamp

Herein, you'll learn

- How to stream data from the Twitter API
- How to filter incoming tweets for keywords
- About API Authentication and OAuth



Herein, you'll learn

- How to stream data from the Twitter API
- How to filter incoming tweets for keywords
- About API Authentication and OAuth
- How to use the Tweepy Python package



2. Herein, you'll learn

As a final deep dive, you're going to stream data from the Twitter API. You'll learn how to filter incoming tweets for keywords, you'll learn about the principles of API authentication and OAuth. You'll also learn the basics of the package

Access the Twitter API

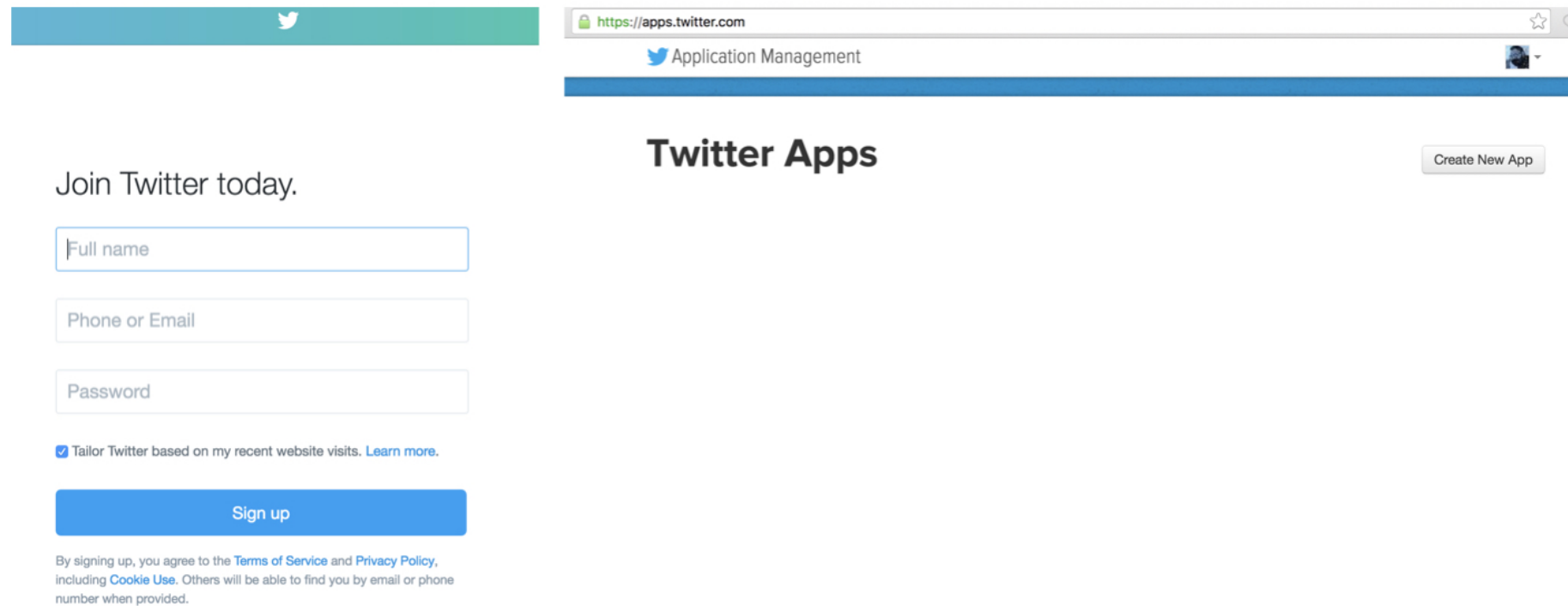


Join Twitter today.

☒ Tailor Twitter based on my recent website visits. [Learn more.](#)

By signing up, you agree to the [Terms of Service](#) and [Privacy Policy](#), including [Cookie Use](#). Others will be able to find you by email or phone number when provided.

Access the Twitter API



The screenshot shows the Twitter Apps management interface. On the left, there's a sign-up form with fields for 'Full name', 'Phone or Email', and 'Password'. Below these is a checkbox for 'Tailor Twitter based on my recent website visits' and a 'Sign up' button. A small disclaimer at the bottom states: 'By signing up, you agree to the Terms of Service and Privacy Policy, including Cookie Use. Others will be able to find you by email or phone number when provided.' On the right, the 'Twitter Apps' section is visible, featuring a 'Create New App' button. The browser's address bar shows 'https://apps.twitter.com'.

4. Access the Twitter API

One of the first major differences between the Twitter API and all the APIs you have seen so far is that you were able to access all the others anonymously and Twitter requires that you have an account. In order gain access to the Twitter API, one needs to create a twitter account if you don't already have one,

Access the Twitter API

Hugo Bowne-Anderson

[Details](#)[Settings](#)[Keys and Access Tokens](#)[Permissions](#)

Application Settings

Keep the "Consumer Secret" a secret. This key should never be human-readable in your application.

Consumer Key (API Key)

[REDACTED]

Consumer Secret (API Secret)

[REDACTED]

Access Level

Read-only ([modify app permissions](#))

Owner

hugobowne

Owner ID

[REDACTED]

Access the Twitter API

Your Access Token

This access token can be used to make API requests on your own account's behalf. Do not share your access token secret with anyone.

Access Token

[REDACTED]

Access Token Secret

[REDACTED]

Access Level

Read-only

Owner

hugobowne

Owner ID

[REDACTED]

Twitter has a number of APIs

REST APIs

The [REST APIs](#) provide programmatic access to read and write Twitter data. Author a new Tweet, read author profile and follower data, and more. The REST API identifies Twitter applications and users using [OAuth](#); responses are available in JSON.

If your intention is to monitor or process Tweets in real-time, consider using the [Streaming API](#) instead.

8. Twitter has a number of APIs

It is now important to mention that Twitter has a number of APIs. Firstly, they have a REST API; we won't go into the gory details of REST APIs here but I'll say two things - one: REST is short for Representational State Transfer; two: Twitter's REST API allows the user to "read and write Twitter data"; In order "monitor or process Tweets in real-time",

Twitter has a number of APIs

The Streaming APIs

Overview

The Streaming APIs give developers low latency access to Twitter's global stream of Tweet data. A proper implementation of a streaming client will be pushed messages indicating Tweets and other events have occurred, without any of the overhead associated with polling a REST endpoint.

If your intention is to conduct singular searches, read user profile information, or post Tweets, consider using the [REST APIs](#) instead.

Twitter offers several streaming endpoints, each customized to certain use cases.

Public streams	Streams of the public data flowing through Twitter. Suitable for following specific users or topics, and data mining.
User streams	Single-user streams, containing roughly all of the data corresponding with a single user's view of Twitter.
Site streams	The multi-user version of user streams. Site streams are intended for servers which must connect to Twitter on behalf of many users.

9. Twitter has a number of APIs that is, to stream Twitter data, however, we'll want to use Twitter's Streaming API. In particular,

Twitter has a number of APIs

The Streaming APIs

Overview

The Streaming APIs give developers low latency access to Twitter's global stream of Tweet data. A proper implementation of a streaming client will be pushed messages indicating Tweets and other events have occurred, without any of the overhead associated with polling a REST endpoint.

If your intention is to conduct singular searches, read user profile information, or post Tweets, consider using the [REST APIs](#) instead.

Twitter offers several streaming endpoints, each customized to certain use cases.

Public streams	Streams of the public data flowing through Twitter. Suitable for following specific users or topics, and data mining.
User streams	Single-user streams, containing roughly all of the data corresponding with a single user's view of Twitter.
Site streams	The multi-user version of user streams. Site streams are intended for servers which must connect to Twitter on behalf of many users.

10. Twitter has a number of APIs we'll use the public stream, which Twitter's API documentation states "Streams of the public data flowing through Twitter.". The Public Stream itself contains a number of options. As we want to read and process tweets,

Twitter has a number of APIs

GET statuses/sample

Returns a small random sample of all public statuses. The Tweets returned by the default access level are the same, so if two different clients connect to this endpoint, they will see the same Tweets.

Resource URL

```
https://stream.twitter.com/1.1/statuses/sample.json
```

11. Twitter has a number of APIs
we'll want to use the GET statuses/sample API, which "Returns a small random sample of all public statuses."

Twitter has a number of APIs

Firehose

API Reference Documents

Streaming

[GET statuses/firehose](#)


This endpoint requires special permission to access.

Returns all public statuses. Few applications require this level of access. Creative use of a combination of other resources and various access levels can satisfy nearly every application use case.

12. Twitter has a number of APIs

If you wanted to access absolutely "All public statuses", you would need to use Twitter's Firehose API, which is not publicly available and would most likely cost you a pretty penny.

Tweets are returned as JSONs

 <https://dev.twitter.com/overview/api/tweets>

Field Guide

The actual UTF-8 text of the status update. See [twitter-text](#) for details on what is currently considered valid characters.

Example:

text

String


```
"text": "Tweet  
Button, Follow  
Button, and Web  
Intents  
javascript now  
support SSL  
http:\\\\t.co\\9f  
bA0oYy ^TS"
```

13. Tweets are returned as JSONs

One last point to note before we begin streaming tweets: tweets are returned to us as JSONs and they contain numerous possible fields.

Check out the Twitter tweet field guide [here](#). You can get tweet text, user, language, time of tweet,

Tweets are returned as JSONs

 <https://dev.twitter.com/overview/api/tweets>

Field Guide

lang

String

Nullable. When present, indicates a [BCP 47](#) language identifier corresponding to the machine-detected language of the Tweet text, or “und” if no language could be detected.

Example:

```
"lang": "en"
```

14. Tweets are returned as JSONs among many other fields. Lets see how to access and stream data from the Twitter API!. For first-time Python tweet-streamers, I usually recommend the package tweepy,

Using Tweepy: Authentication handler

tw_auth.py

```
import tweepy, json
access_token = "...
access_token_secret = "...
consumer_key = "...
consumer_secret = "...
auth = tweepy.OAuthHandler(consumer_key, consumer_secret)
auth.set_access_token(access_token, access_token_secret)
```

15. Using Tweepy: Authentication handler

which has a nice balance of usability and capability. Let's now use it to stream some tweets! First off, it has an OAuth handler which takes care of all of that nasty stuff for you: all you need to do is to pass the API Key and Secret to the handler and then to pass to access credentials using the `set_access_token` method.

Tweepy: define stream listener class

st_class.py

```
class MyStreamListener(tweepy.StreamListener):
    def __init__(self, api=None):
        super(MyStreamListener, self).__init__()
        self.num_tweets = 0
        self.file = open("tweets.txt", "w")
    def on_status(self, status):
        tweet = status._json
        self.file.write(json.dumps(tweet) + '\\n')
        tweet_list.append(status)
        self.num_tweets += 1
        if self.num_tweets < 100:
            return True
        else:
            return False
        self.file.close()
```

16. Tweepy: define stream listener class

After this, you'll need to define your Twitter stream listener Class. I wouldn't necessarily expect you to be able to do this yourself so I'm going to do that for you both here and in the interactive exercises that follow. Here I define a Tweet listener that creates a file called 'tweets dot txt', collects streaming tweets and writes them to the file 'tweets dot txt'; once 100 tweets have been streamed, the listener closes the file and stops listening.

Using Tweepy: stream tweets!!

tweets.py

```
# Create Streaming object and authenticate
l = MyStreamListener()
stream = tweepy.Stream(auth, l)
# This line filters Twitter Streams to capture data by keywords:
stream.filter(track=['apples', 'oranges'])
```

17. Using Tweepy: stream tweets!!

Now that we have written our Twitter Stream Listener Class, all you need to do is to create an instance of it and authenticate it. You can then stream tweets that containing keywords of choice by applying the filter method to the object stream! In the following exercises, you'll practice writing Python code to stream tweets and then you'll do some basic analysis of these tweets to see how often particular keywords are mentioned.

Let's practice!

INTERMEDIATE IMPORTING DATA IN PYTHON

Final Thoughts

INTERMEDIATE IMPORTING DATA IN PYTHON



Hugo Bowne-Anderson
Data Scientist at DataCamp

What you've learned:

- Importing text files and flat files
- Importing files in other formats
- Writing SQL queries
- Getting data from relational databases
- Pulling data from the web
- Pulling data from APIs

Let's practice!

INTERMEDIATE IMPORTING DATA IN PYTHON