
Marco Bonzanini

Python, Data Science, Text Analytics

≡ MENU

Mining Twitter Data with Python (Part 1: Collecting data)

March 2, 2015 · Marco

[Twitter](#) is a popular social network where users can share short SMS-like messages called *tweets*. Users share thoughts, links and pictures on Twitter, journalists comment on live events, companies promote products and engage with customers. The list of different ways to use Twitter could be really long, and with 500 millions of tweets per day, there's a lot of data to analyse and to play with.

This is the first in a series of articles dedicated to mining data on Twitter using Python. In this first part, we'll see different options to collect data from Twitter. Once we have built a data set, in the next episodes we'll discuss some interesting data applications.

Update July 2016: my new [book on data mining for Social Media](#) is out! Part of the content in this tutorial has been improved and expanded as part of the book, so please have a look. Chapter 2 about mining Twitter is available as a free sample from [the publisher's web site](#), and the companion code with many more examples is available [on my GitHub](#)

Table of Contents of this tutorial:

- Part 1: Collecting Data (this article)
- [Part 2: Text Pre-processing](#)
- [Part 3: Term Frequencies](#)
- [Part 4: Rugby and Term Co-Occurrences](#)
- [Part 5: Data Visualisation Basics](#)
- [Part 6: Sentiment Analysis Basics](#)
- [Part 7: Geolocation and Interactive Maps](#)

More updates: fixed version number of Tweepy to avoid problem with Python 3; fixed discussion on `_json` to get the JSON representation of a tweet; added example of `process_or_store()`.

Register Your App

In order to have access to Twitter data programmatically, we need to create an app that interacts with the Twitter API.

The first step is the registration of your app. In particular, you need to point your browser to <http://apps.twitter.com>, log-in to Twitter (if you're not already logged in) and register a new application. You can now choose a name and a description for your app (for example "Mining Demo" or similar). You will receive a *consumer key* and a *consumer secret*: these are application settings that should always be kept private. From the configuration page of your app, you can also require an access token and an access token secret. Similarly to the consumer keys, these strings must also be kept private: they provide the application access to Twitter on behalf of your account. The default permissions are read-only, which is all we need in our case, but if you decide to change your permission to provide writing features in your app, you must negotiate a new access token.

Important Note: there are rate limits in the use of the Twitter API, as well as limitations in case you want to provide a downloadable data-set, see:

- <https://dev.twitter.com/overview/terms/agreement-and-policy>
- <https://dev.twitter.com/rest/public/rate-limiting>

Accessing the Data

Twitter provides [REST APIs](#) you can use to interact with their service. There is also [a bunch of Python-based clients](#) out there that we can use without re-inventing the wheel. In particular, [Tweepy](#) is one of the most interesting and straightforward to use, so let's install it:

```
pip install tweepy==3.3.0
```

Update: the release 3.4.0 of Tweepy has introduced a problem with Python 3, currently fixed on [github](#) but not yet available with pip, for this reason we're using version 3.3.0 until a new release is available.

More Updates: the release 3.5.0 of Tweepy, already available via pip, seems to solve the problem with Python 3 mentioned above.

In order to authorise our app to access Twitter on our behalf, we need to use the OAuth interface:

```
1 import tweepy
2 from tweepy import OAuthHandler
3
4 consumer_key = 'YOUR-CONSUMER-KEY'
5 consumer_secret = 'YOUR-CONSUMER-SECRET'
6 access_token = 'YOUR-ACCESS-TOKEN'
7 access_secret = 'YOUR-ACCESS-SECRET'
8
9 auth = OAuthHandler(consumer_key, consumer_secret)
10 auth.set_access_token(access_token, access_secret)
11
12 api = tweepy.API(auth)
```

The `api` variable is now our entry point for most of the operations we can perform with Twitter.

For example, we can read our own timeline (i.e. our Twitter homepage) with:

```
1 for status in tweepy.Cursor(api.home_timeline).items(10):
2     # Process a single status
3     print(status.text)
```

Tweepy provides the convenient `Cursor` interface to iterate through different types of objects. In the example above we're using `10` to limit the number of tweets we're reading, but we can of course access more. The `status` variable is an instance of the `Status` () class, a nice wrapper to access the data. The JSON response from the Twitter API is available in the attribute `_json` (with a leading underscore), which is not the raw JSON string, but a dictionary.

So the code above can be re-written to process/store the JSON:

```
1 for status in tweepy.Cursor(api.home_timeline).items(10):
2     # Process a single status
3     process_or_store(status._json)
```

What if we want to have a list of all our followers? There you go:

```
1 for friend in tweepy.Cursor(api.friends).items():
2     process_or_store(friend._json)
```

And how about a list of all our tweets? Simple:

```
1 | for tweet in tweepy.Cursor(api.user_timeline).items():
2 |     process_or_store(tweet._json)
```

In this way we can easily collect tweets (and more) and store them in the original JSON format, fairly easy to convert into different data models depending on our storage (many NoSQL technologies provide some bulk import feature).

The function `process_or_store()` is a place-holder for your custom implementation. In the simplest form, you could just print out the JSON, one tweet per line:

```
1 | def process_or_store(tweet):
2 |     print(json.dumps(tweet))
```

Streaming

In case we want to “keep the connection open”, and gather all the upcoming tweets about a particular event, the streaming API is what we need. We need to extend the `StreamListener()` to customise the way we process the incoming data. A working example that gathers all the new tweets with the `#python` hashtag:

```
1 | from tweepy import Stream
2 | from tweepy.streaming import StreamListener
3 |
4 | class MyListener(StreamListener):
5 |
6 |     def on_data(self, data):
7 |         try:
8 |             with open('python.json', 'a') as f:
9 |                 f.write(data)
10 |            return True
11 |         except BaseException as e:
12 |             print("Error on_data: %s" % str(e))
13 |             return True
14 |
15 |     def on_error(self, status):
16 |         print(status)
17 |         return True
18 |
19 | twitter_stream = Stream(auth, MyListener())
20 | twitter_stream.filter(track=['#python'])
```

Depending on the search term, we can gather tons of tweets within a few minutes. This is especially true for live events with a world-wide coverage (World Cups, Super Bowls, Academy Awards, you name it), so keep an eye on the JSON file to understand how fast it grows and consider how many tweets you might need for your tests. The above script will save each tweet on a new line, so you can use the command `wc -l python.json` from a Unix shell to know how many tweets you’ve gathered.

You can see a minimal working example of the Twitter Stream API in the following Gist:

[twitter_stream_downloader.py](#)

Summary

We have introduced `tweepy` as a tool to access Twitter data in a fairly easy way with Python. There are different types of data we can collect, with the obvious focus on the “tweet” object.

Once we have collected some data, the possibilities in terms of analytics applications are endless. In the next episodes, we’ll discuss some options.

[@MarcoBonzanini](#)

Table of Contents of this tutorial:

- Part 1: Collecting Data (this article)
- [Part 2: Text Pre-processing](#)
- [Part 3: Term Frequencies](#)
- [Part 4: Rugby and Term Co-Occurrences](#)
- [Part 5: Data Visualisation Basics](#)
- [Part 6: Sentiment Analysis Basics](#)
- [Part 7: Geolocation and Interactive Maps](#)

Related

Mastering Social Media Mining with Python

In "API"

Mining Twitter Data with Python (Part 2: Text Pre-processing)

In "Data Mining"

Mining Twitter Data with Python (Part 3: Term Frequencies)

In "Data Mining"

Share this:



Like this:

Loading...



PREVIOUS POST

[Fuzzy String Matching in Python](#)

NEXT

[Mining Twitter Data with Python \(Part 2: Text Pre-processing\)](#)

156 thoughts on “Mining Twitter Data with Python (Part 1: Collecting data)”

[Older Comments](#)

Pingback: [Mining Twitter Data with Python \(Part 1: Collecting data\)](#) « — was einer so denkt —

Pingback: [DataCamp Data Import](#) « — was einer so denkt —



Sameque

F E B R U A R Y 1 0 , 2 0 1 7 A T 8 : 0 6 P M

Dear Marco.

How do I get twitters from other users that are not from my time line, that is, in general. It's possible? as? What is the limit of twitters I can get in python at a time.

★ Like

[Reply](#)



Marco

F E B R U A R Y 2 8 , 2 0 1 7 A T 4 : 2 7 P M

Hi Sameque

you can check out an example to get the a user's timeline from my book:

https://github.com/bonzanini/Book-SocialMediaMiningPython/blob/master/Chap02-03/twitter_get_user_timeline.py

The rate limits of the Twitter API are described in their documentation:

<https://dev.twitter.com/rest/public/rate-limiting>

Cheers

Marco

★ Like

[Reply](#)



Rony Armon

M A R C H 1 0 , 2 0 1 8 A T 1 0 : 0 8 A M

Hi Marco, Thanks for the book and scripts

Can you please explain the use of this if loop in the script?

```
if __name__ == '__main__':
```

```
if len(sys.argv) != 2
```

★ Like



Kunal

F E B R U A R Y 1 9 , 2 0 1 8 A T 2 : 0 7 P M

I haven't found a way to get such data.

The official way is to use enterprise API, however, it is really expensive.

There also some services like this

https://www.fiverr.com/data_dealer/provide-all-tweets-of-a-particular-user-historical-data

but I haven't tested it yet

★ Like

[Reply](#)



Dharshini Jaiganesh

F E B R U A R Y 1 6 , 2 0 1 7 A T 5 : 2 1 A M

Very useful post. Can you suggest me how to store these retrieved tweets in MYSQL db?

★ Like

[Reply](#)



Marco

F E B R U A R Y 2 8 , 2 0 1 7 A T 4 : 3 0 P M

Hi, I haven't worked much on MySQL recently, but recent versions support a JSON data type so you just create a column of type JSON and dump the entire tweet in it. Another option is that you read the JSON and normalise the structure with only the fields that you need.

Cheers,

Marco

★ Like

[Reply](#)



HAMZA BOUFTAIH

F E B R U A R Y 2 0 , 2 0 1 7 A T 1 1 : 5 5 P M

please help !

i get this error :

File "C:/Users/Hamza-HP/Desktop/untitled/Tweepy.py", line 24, in
process_or_store(status._json)

File "C:/Users/Hamza-HP/Desktop/untitled/Tweepy.py", line 18, in process_or_store
print(_json.dumps(tweet))

NameError: name '_json' is not defined

but i define json as follow :

```
import simplejson as json
```

i don't really know what's wrong :(

★ Like

[Reply](#)



Marco

F E B R U A R Y 2 8 , 2 0 1 7 A T 4 : 3 2 P M

Hi

if you import simplejson as json, you'll need json.dumps(tweet) rather than _json (which is an attribute of status, but not defined in the main space)

Cheers

Marco

★ Like

[Reply](#)

Pingback: [Mining Twitter Data with Tweepy API – Ng Shun Mo Moses' Blog](#)

Pingback: [How You Talk To Your Coworkers Vs. How You Talk To Your Friends – Stuff I'm Doing](#)



Vijay Selvakumar

M A R C H 8 , 2 0 1 7 A T 9 : 3 1 A M

How can i see the output of live streaming and how can i save the tweets in hdfs for analysis?

★ Like

[Reply](#)



Marco

M A R C H 1 5 , 2 0 1 7 A T 8 : 1 9 A M

Hi,

in the MyListener.on_data() method you can print() the data instead of (or in addition to) writing to file.
For hdfs you'll need an hdfs client but I don't have any particular recommendation here.

Cheers,

Marco

★ Like

[Reply](#)

Pingback: [Tracking The Most Popular News Of Mobile World Congress 2017](#)

Pingback: [Build A Tweet Bot With Python | Webammer](#)



Nav

M A R C H 2 8 , 2 0 1 7 A T 2 : 5 3 A M

Hey Marco, I'm relatively new to coding and I was trying out your script to see 10 of my twitter feeds but its giving me a UnicodeEncodeError. What else do I need to add to the script that you provided to make this work?

★ Like

[Reply](#)



Marco

M A R C H 2 8 , 2 0 1 7 A T 2 : 0 2 P M

Hi Nav,

it depends on which line is throwing the error and what the exact error is. The examples are tested in Python 3.4+, so if you're using Python 2 please keep in mind that the string data type is different (unicode in Python 3, non-unicode in Python 2). If that's the case, please consider upgrading to Python 3. Also have a look at this one for more details:

<https://wiki.python.org/moin/UnicodeEncodeError>

Cheers

Marco

★ Like

[Reply](#)



Subhani

M A R C H 3 1 , 2 0 1 7 A T 3 : 2 1 P M

Hi,

I tried to the Streaming example given above. But unfortunately it gives a syntax error indicating the '&' symbol before 'quote', in "print("Error on_data: %s" % str(e))" line.

How can I get this error fixed ?

Thanks in advance !

★ Like

[Reply](#)



payungirta

J U L Y 1 9 , 2 0 1 7 A T 1 : 4 5 P M

hi, i got same error.

i just put commend on exception like this :

```
#print("Error on_data: %s" % str(e))
```

and put this code on the next line or replace it both working wonderfull :

```
print ('error', str(e))
```

★ Like

[Reply](#)



payungirta

J U L Y 1 9 , 2 0 1 7 A T 1 : 4 7 P M

pardon i put commend on #print("Error on_data: %s" % str(e))

★ Like



Marco

J U L Y 1 9 , 2 0 1 7 A T 2 : 1 2 P M

Hi,

wordpress keeps messing around with quotes and HTML symbols, I think this is fixed now, until it breaks the next time. The symbol should be a regular double quote like this “

★ Like



Kashf ul Huda

A P R I L 5 , 2 0 1 7 A T 5 : 4 3 A M

my code shows syntax error in this line:

```
print("Error on_data: %s" % str(e))
```

★ Like

[Reply](#)



Justin

A P R I L 9 , 2 0 1 7 A T 2 : 4 6 P M

Hi – can you recommend a hosting platform for a python listener such as this ? e.g. a host that will offer everything required and not charge the earth for an “always on” app ? Thanks !

★ Like

[Reply](#)



Marco

A P R I L 1 0 , 2 0 1 7 A T 1 0 : 0 9 A M

Hi Justin, in terms of hosting I've only used aws for this, not sure if it fits your requirements. I'm sure there are other options but I don't have a specific recommendation at the moment I'm afraid.

Cheers

Marco

★ Like

[Reply](#)



smile2nil

J U N E 1 5 , 2 0 1 7 A T 1 2 : 1 1 P M

Hi Marco,

I am interested in knowing more about how to host on aws. Where can i get details?

★ Like

Iacopo



A P R I L 1 5 , 2 0 1 7 A T 7 : 0 9 A M

Hello, these days I have saved several tweets with your script.

Today I was watching them and I seem all written by professionals and not by individual.

They are tweets that talk about the weather, news, promotion activities etc ...

it's normal? or am I doing something wrong?

For me it would be more useful to analyze the common people tweet, in order to have a vision of what people really think.

Here I have published some are in Italian, but your last name seems Italian: D

https://docs.google.com/document/d/1gYZVnFSpnAYqAEQBWeLJHC6QdpA2koY4i3_mF85Rf7E/edit?usp=sharing

★ Like

[Reply](#)



Jeremy

A P R I L 1 5 , 2 0 1 7 A T 7 : 4 5 A M

```
print("Error on_data: %s" % str(e))
```

^

SyntaxError: invalid syntax

I keep getting this error running in Andaconda. Could get it to run the 10 post but cant get it to run live stream.

★ Like

[Reply](#)



DaX

A P R I L 2 3 , 2 0 1 7 A T 9 : 0 0 P M

Tweepy doesn't work anymore.

★ Like

[Reply](#)



Marco

A P R I L 2 4 , 2 0 1 7 A T 7 : 0 2 P M

Tweepy 3.5.0 works just fine

★ Like

[Reply](#)



Data Factory (@datafactoryIN)

A P R I L 2 8 , 2 0 1 7 A T 8 : 5 8 P M

Hi Marco. I am running these codes on my mac. where do the json files get stored?

★ Like

[Reply](#)



Marco

M A Y 9 , 2 0 1 7 A T 7 : 5 1 A M

Hi

if you use the code as it is, it creates a "python.json" file in the same folder where you're running the script. If you check out the examples from my book you see how the filename is created dynamically.

Cheers,
Marco

★ Like

[Reply](#)



Rajan

A P R I L 3 0 , 2 0 1 7 A T 6 : 5 8 A M

Hi Marco, is it possible to save the json for each tweet in separate files not appending to python.json ?

regards

★ Like

[Reply](#)



Marco

M A Y 9 , 2 0 1 7 A T 7 : 5 3 A M

Hi

yes you can come up with the file name dynamically, using a different name every time, for example using the tweet id to ensure the names are unique. Maybe adding the timestamp as well so you can sort them "easily" (you'll end up with too many files)

Cheers
Marco

★ Like

[Reply](#)



Lewis

M A Y 3 , 2 0 1 7 A T 1 : 1 8 P M

hey Marco..how can i filter the tweets about political views in a certain country like for Zambia only..reply asap...

★ Like

[Reply](#)



Marco

M A Y , 2 0 1 7 A T 7 : 5 8 A M

The filter method also takes a “locations” argument. It has to follow the format as described here:

<https://dev.twitter.com/streaming/overview/request-parameters#locations>

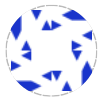
You can only pass the coordinates as a box so I don't think you can explicitly specify the country. Also, only a small number of tweets come with geolocation information.

Cheers

Marco

★ Like

[Reply](#)



Leo

J U L Y , 2 0 1 7 A T 7 : 3 8 P M

Hi Marco, I have read that one can only stream based on a keyword or location but not both. Does that sound right to you? thanks

★ Like



Marco

J U L Y , 2 0 1 7 A T 6 : 2 6 A M

Hi Leo,

you can pass keywords (track), user IDs (follow) and locations, but they will be put in logical OR, as described here

<https://dev.twitter.com/streaming/reference/post/statuses/filter>

★ Like



sanyam09

J U N E , 2 0 1 7 A T 7 : 1 1 P M

Hi

Actually I have to do sentiment analysis and for that purpose I need to collect some Twitter data so can you please tell how to we get consumer ID and consumer secret mentioned in your tutorials first part.

How the application will be set up?

★ Like

[Reply](#)

Pingback: [What Happened on Twitter During the Champions League Final | Open Data Science](#)

Pingback: [Học BigData \(BD\) – Ngày 8 – Anht1289](#)



Ryan Nazareth

J U L Y 1 2 , 2 0 1 7 A T 6 : 5 2 P M

Hi Marco, I really enjoy reading your blogs and attending meetups you speak at. I am trying to collect a year's worth twitter data from the current date based on selected keywords. I've used the twitter search API but it only seems to give me 12 days worth results (around 3000 tweets) for a keyword. Would I be able to get results from a longer period using tweepy (ideally I would like to specify the start and end date for my search) ? or would I need to subscribe to Twitter Firehouse ?

Best Wishes

Ryan

★ Like

[Reply](#)



Pat

J U L Y 3 0 , 2 0 1 7 A T 2 : 1 7 A M

Dear Marco,

i hope you can help me out.

I sucessfully set up tweepy and now i wanna get the latest trends for a specific place (set by woeid) as a list and not more than 10 trends. I wish i could do that myself but i'm a complete douche.

best regrets.

★ Like

[Reply](#)



Marco

A U G U S T 2 , 2 0 1 7 A T 3 : 3 5 P M

Hi Pat

I haven't used the twitter trend API, but this is the endpoint to check out

<https://dev.twitter.com/rest/reference/get/trends/place>

In Tweepy, this translated to a call to `api.trends_place(woeid)` — as far as I can see, you have to pass one woeid at a time

Cheers,

Marco

★ Like

[Reply](#)



Pat

A U G U S T 9 , 2 0 1 7 A T 1 1 : 1 6 A M

Hello Marco,

thank you for your reply. I was able to solve my wish by using snippets, examples and tutorials.

Cheers,
Pat

★ Like

Pingback: [RESOURCES - KARTHICK PHILLIP NAGARAJAN](#)



yousufmotiwalla

AUGUST 19, 2017 AT 3:49 PM

Hi!

i got this error, can you help me out

Traceback (most recent call last):

File "C:\Users\YousufM\Desktop\abc.py", line 17, in

process_or_store(status._json)

NameError: name 'process_or_store' is not defined

my code was this

for status in tweepy.Cursor(api.home_timeline).items(10):

Process a single status

process_or_store(status._json)

★ Like

[Reply](#)



Marco

AUGUST 29, 2017 AT 8:16 AM

Hi

you need to defined the function as described at the end of the paragraph (as an example, there's a simple implementation that prints out the json), you'll have to implement it depending on your needs (print it, store it on DB, etc.)

Cheers,

Marco

★ Like

[Reply](#)



Milton Teixeira

SEPTEMBER 5, 2017 AT 4:31 PM

How do we do that? oO

★ Like

Marco



S E P T E M B E R 8 , 2 0 1 7 A T 3 : 5 7 P M

Hi Milton, this is the simplest possible implementation as included in the article:

```
1 | def process_or_store(tweet):  
2 |     print(json.dumps(tweet))
```

This will simply print the JSON on the screen. If you want to store it on file you can re-use the example from my book: https://github.com/bonzanini/Book-SocialMediaMiningPython/blob/master/Chap02-03/twitter_get_user_timeline.py

Cheers,
Marco

★ Like



Dani

S E P T E M B E R 4 , 2 0 1 7 A T 9 : 1 8 A M

Hi Marco,

I am a student assisting a professor who wants to analyze twitter behavior of Dutch politicians (most of which have a public twitter profile) during the elections. To do this, I want to construct a database with the following information:

Per politician (about 300 in total), how many original tweets / retweets / replies they sent per day, for all days in the period 01/01/2017 – 03/31/2017. I have no experience with programming whatsoever (I do like math and working with Excel / statistic software).

Currently, I am doing this all manually (counting by scrolling through their timelines and putting all information in Excel). As you can imagine, this takes ages, and I suspect that it might be possible to do it so much quicker. Besides, learning (the basics of) programming has been on my wishlist for a long time.

However, I would like to know if it is likely that there is a 'programming solution' for this problem at all or that due to privacy/request limitations it will not be worth to devote all these hours to learning because it won't work anyway. With all your experience: Do you think this is possible? Or might I just as well continue counting?

Best,
Dani

★ Like

[Reply](#)



Marco

S E P T E M B E R 8 , 2 0 1 7 A T 3 : 5 2 P M

Hi Dani

there are several limitations imposed by the Twitter API, but there are definitely some workarounds. If you're tracking a specific account, you can retrieve up to 3,200 of its most recent tweets using this method (https://dev.twitter.com/rest/reference/get/statuses/user_timeline). An example of implementation using Python is in my book (https://github.com/bonzanini/Book-SocialMediaMiningPython/blob/master/Chap02-03/twitter_get_user_timeline.py). On top of the limitation given by the total number of tweets that you can retrieve with this approach, there is also a rate limit (described in the Twitter API link above), so retrieving a lot of data will likely

require some time just because you need to pause the requests (they don't let you hammer the API). If a user tweets a lot, you're unlikely to be able to capture a specific time window in the past because you can only retrieve the most recent 3,200 tweets.

On the other side, for upcoming tweets, you can keep the stream open and track the activity of specific accounts, using the Streaming API as described in this tutorial. The only difference is that you'd use the option "follow" to spell out the user names you want to include in the stream, rather than "track" that is used for keywords. This second approach will require an always-on server and possibly additional configuration to monitor the stream.

As a starting point you can use the user_timeline script from my book (link above) and try it with a few usernames.

Best wishes for your research, I hope this helps.

Cheers,
Marco

★ Like

[Reply](#)



Olawunmi Shyllon

O C T O B E R 2 8 , 2 0 1 7 A T 6 : 0 7 P M

Hello,

Thank you for the great job you have done. I tried the code and it works with a single term but when I tried using multiple terms for track term I got errors. How do I modify this code "python twitter_stream_download.py -q apple -d data" to account for the change of terms. Thank you

Ola

★ Like

[Reply](#)



Marco

N O V E M B E R 1 3 , 2 0 1 7 A T 2 : 3 1 P M

Hi Ola

you can use double quotes around your query, and all the query terms will be passed to the API, for example:

```
python twitter_stream_download.py -q "apple football" -d data
```

this will query the API for "apple AND football"

otherwise with:

```
python twitter_stream_download.py -q "apple, football" -d data
```

using the comma between keyword is equivalent to an OR query (in this case, apple OR football)

It's always worth checking out the official docs in case the API changes in the future

<https://developer.twitter.com/en.html>

Cheers
Marco

★ Like

[Reply](#)



Emre Yuksek

N O V E M B E R 1 6 , 2 0 1 7 A T 9 : 5 6 A M

Hi,
I tried the codes and it works very well but i want to store tweets using utf-8 encoding.
How can i store tweets to json file with using utf-8 encoding?
Thank you.

★ Liked by [1](#)

[person](#)

[Reply](#)



varunbike

N O V E M B E R 1 7 , 2 0 1 7 A T 4 : 4 5 A M

This is Truly a masterpiece .Very much explained and clear in meaning . I followed this and working fine .
Is there any way we can get rid of "tweepy.error.TweepError: Twitter error response: status code = 429" ? I searched this in google but not much help I got . I put this in tweepy github page but the response redirecting me to twitter Rate Limit page . I have seen many online application available which are making much more request than what I am trying to do .
Any help in this regard is highly appreciated .

Thank you .

★ Like

[Reply](#)



chrisra

J A N U A R Y 9 , 2 0 1 8 A T 1 : 2 8 P M

Dear Marco, thank you for this tutorial and your book, which is really great and addictive :)
I have collected quite a lot of tweets so far and tried different analyses.
Now would like to convert my json-Tweets into a csv-format but am struggling for some reasons. Could you give an example on how to do that based on your example here?
I am interested in the conversion of existing json-files with data collected from twitter, as well as in how to store twitter streaming data in csv as they directly come out of the stream.
I would appreciate your help very much!
Thank you

★ Like

[Reply](#)



Anvesh

F E B R U A R Y 7 , 2 0 1 8 A T 1 0 : 4 4 P M

I am getting this error.

"cannot import name 'OAuthHandler' "

★ Like

[Reply](#)



miladkordeh

F E B R U A R Y 1 6 , 2 0 1 8 A T 9 : 0 5 P M

Hi, Where do I need to run pip install tweepy==3.3.0 ????

★ Like

[Reply](#)



miladkordeh

F E B R U A R Y 1 6 , 2 0 1 8 A T 9 : 1 1 P M

Please ignore my question I got it: CMD

★ Like

[Reply](#)



anubhav

M A R C H 1 4 , 2 0 1 8 A T 3 : 5 8 P M

Hello,

I tried extracting historical tweets. Could you please provide any suggestion on this question:

<https://stackoverflow.com/questions/49119766/how-to-extract-historical-data-on-a-custom-query-from-tweeter-search-apis>

★ Like

[Reply](#)

Pingback: [How do i make a correct Json file format from this tweepy-script? | Question](#)

Older Comments

Leave a Reply

Enter your comment here...

Follow [@marcobonzanini](#)

 Book



 Video Course



 Video Course



Recent Posts

[Video Course: Practical Python Data Science Techniques](#) September 13, 2017

[Video Course: Data Analysis with Python](#) May 3, 2017

[PyCon Italy 2017 write-up](#) April 11, 2017

[PyCon UK 2016 write-up](#) September 25, 2016

[Mastering Social Media Mining with Python](#) August 2, 2016

[PyData London 2016 write-up](#) May 11, 2016

[PyCon Italia / PyData Italy 2016 Write-Up](#) April 19, 2016

[Retrocomputing and Python: import turtle](#) December 29, 2015

[Adding Slack Notifications to a Luigi Pipeline in Python](#) November 21, 2015

[Building Data Pipelines with Python and Luigi](#) October 24, 2015

Archives

[September 2017](#) (1)

[May 2017](#) (1)

[April 2017](#) (1)

[September 2016](#) (1)

[August 2016](#) (1)

[May 2016](#) (1)

[April 2016](#) (1)

[December 2015](#) (1)

[November 2015](#) (1)

[October 2015](#) (1)

[September 2015](#) (2)

[August 2015](#) (3)

[July 2015](#) (2)

[June 2015](#) (3)

May 2015 (2)

April 2015 (3)

March 2015 (4)

February 2015 (4)

January 2015 (4)

Categories

API (2)

Best Practices (1)

Big Data (4)

Books (1)

Conferences (4)

Data Mining (8)

data science (1)

Data Visualisation (2)

Elasticsearch (6)

Engineering (2)

Functional Programming (1)

Graph Databases (1)

Javascript (3)

London (3)

Machine Learning (1)

Maps (1)

Meet-ups (3)

MongoDB (1)

Neo4j (1)

NLP (9)

NoSQL (1)

Python (31)

Relevance (2)

Retrocomputing (1)

Search (7)

Sentiment Analysis (2)

Spark (1)

Text Analytics (3)

Text Mining (2)

Text Summarisation (1)

Uncategorized (1)



The content of this blog by [Marco Bonzanini](#) is licensed under a [Creative Commons Attribution 4.0 International License](#).

Blog Posts | About | Contact | Privacy Policy | Terms of Service

 Follow 