Sidi Mohamed Ben Abdallah University of Fez - Polydisciplinary Faculty of Taza Department of Mathematics, Physics and Computer Science

Master: Intelligent and mobile system « SIM »

Report

Extract Data Twitter

Realized by :

El aboui Hafida & Zouitni Chaimae

Proposed by :

Pr. Akharraz Ismail

Academic Year: 2019 /2020

General Introduction

Social media platforms have been a major part of our daily lives. But with the freedom of expression there is no way one can check whether the posts/tweets/expressions are classified on which polarity. Since Twitter is one of the biggest social platforms for microblogging, hence the experiment was done on this platform. There are several topics that are popular over the internet like sports, politics, finance, technology are chosen as the source of the experiment.Every tweet can be divided into three categories based on sentiment analysis, positive, negative or neutral. In the process of analyzing the sentiment, Natural Language Processing is widely used for data processing like removing stopwords, lemmatization, tokenization and POS tagging. In this work, focus is on the detection and prediction of sentiments based on tweets, associated with different topics.There are several ways to carry out the analysis using libraries, APIs, classifiers and tools. The use of data mining techniques namely data extraction, data cleaning, data storage.Also, Twitter is one of the biggest platforms of expressing a person’s feeling on a social media. These set of information can be used in several ways as data to analyze or deduce something.For example, if a new amendment law has been passed it can have its own pros or cons depending on the set of people and how they are affected. I wanted to therefore analyze the sentiment of each tweets and find out whether they are inclined more towards positive or negative polarity. This in future would help in determining whether the political situation is getting better or worse. This kind of analysis would set an alarm in the world and politicians would get a feedback of whether their practices are creating a negative or positive impact to the world.

In this project, Twitter was chosen as the platform for analysis. This is a social media website where people are able to express with the help of tweets that are basically a string of words.Millions of people are using this platform to express their views on any matter or current affairs. But there is an immense risk of determining how authentic these posts were. So with the help of certain useful parameters like accuracy, false positive tweets, false negative tweets, recall and precision efficiency was measured on the analysis.

In short, the main objective of this project is to implement machine learning based data analysis for finding sentiment or polarity for a particular tweet.In order to get more appropriate results, the data goes through several steps of data was collected over a large amount of time over many vivid topics. Then a proper cleaning of data was performed to normalize the piece of information received.

Chapter 1: General Information on Data Twitter

❖ Definition

Twitter is an online news and social networking service where users post and interact with messages, “tweets,” restricted to 140 characters. Registered users can post tweets, but those who are unregistered can only read them.

Many people use Twitter to discuss relevant topics. These topics may be related anything of interest to those posting on twitter and may including: science, data science, computing, sports, politics, weather, news, media and more.

❖ Why Use Twitter?

There are many reasons why Twitter is used as a source for information associated with a disturbance including:

1. **Data from mixed sources:** Anyone can use Twitter and thus the sources of information can include media, individuals, official and others. Mixed sources of information provides a more well-rounded perspective of the impacts of the particular event and the actions being taken to deal with that event.
2. **Embedded content:** Twitter allows users to embed pictures, videos and more to capture various elements of a disturbance both visually and quantitatively.
3. **Instantaneous coverage:** Twitter allows users to communicate directly in real time. Thus, reports on what is going on during an event can happen as the incident unfolds.

Structure of a Tweet

There are various components of a tweet that you can use to extract information:

- **User Name:** This is how each unique user is identified.
- **Time Stamp:** When the tweet was sent.
- **Tweet Text:** The body of the tweet - needs to be 140 characters of less!

- **Hashtags:** Always proceeded by a # symbol. A hashtag is often describes a particular event or can be related to a particular topic. It is a way for users to communicate with a particular group of people on twitter - for instance those attending a conference #agu2016 or those using r #rstats.
- **Links:** Links can be embedded within a tweet. Links are a way that users share information.
- **Embedded Media:** tweets can contain pictures and videos. The most popular tweets often contain pictures.
- **Replies:** When someone posts a tweet, another user can reply directly to that user - similar to a text message except the message is visible to the public.
- **Retweets:** a retweet is when someone shares a tweet with their followers.
- **Favorites:** You can “like” a tweet to keep a history of content that you like in your account.
- **Latitude/Longitude:** about 1% of all tweets contains coordinate information.

❖ Access Tweets and Using the Twitter REST API

Twitter has an API which allows us to access everyone’s tweets. The API has certain limitations including:

1. **You can only access tweets from the last 6-9 days:** This means that you need to think ahead if you want to collect tweets for a particular event.
2. **You can only request 18,000 tweets in one call:** You can stream tweets and collect them using ongoing protocols however there are limitations to how much data you can collect!

Chapter 2: Data Twitter Extraction Steps

❖ Introduction

To get started,we'll need to do the following things:

- Set up a Twitter account if you don't have one already.
- Using your Twitter account, you will need to apply for Developer Access and then create an application that will generate the API credentials that you will use to access Twitter from Python.
- Import the tweepy package.

Once you've done these things, you are ready to begin querying Twitter's API to see what you can learn about tweets.

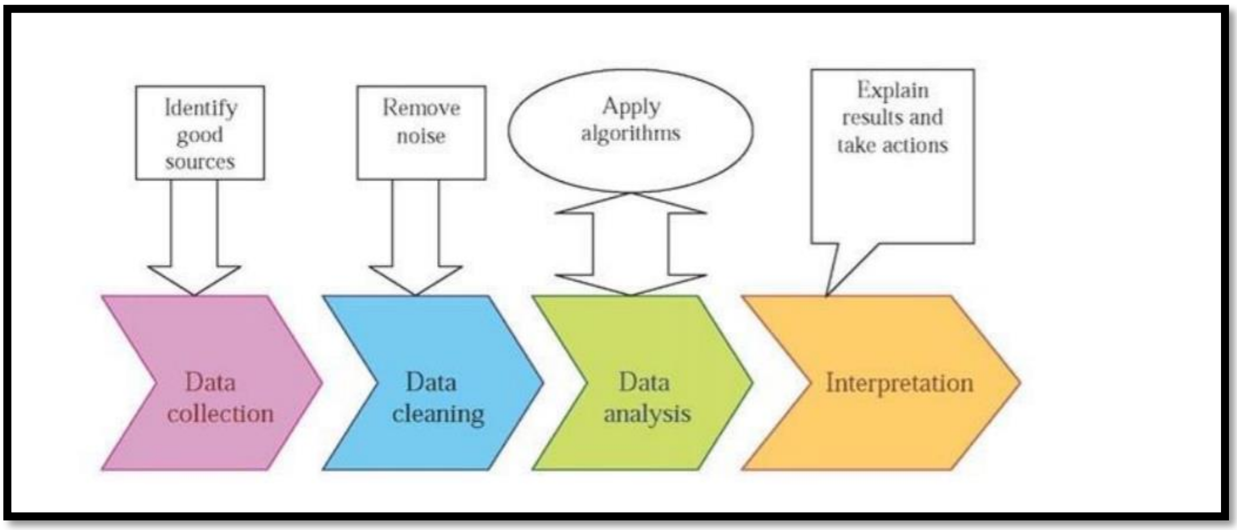
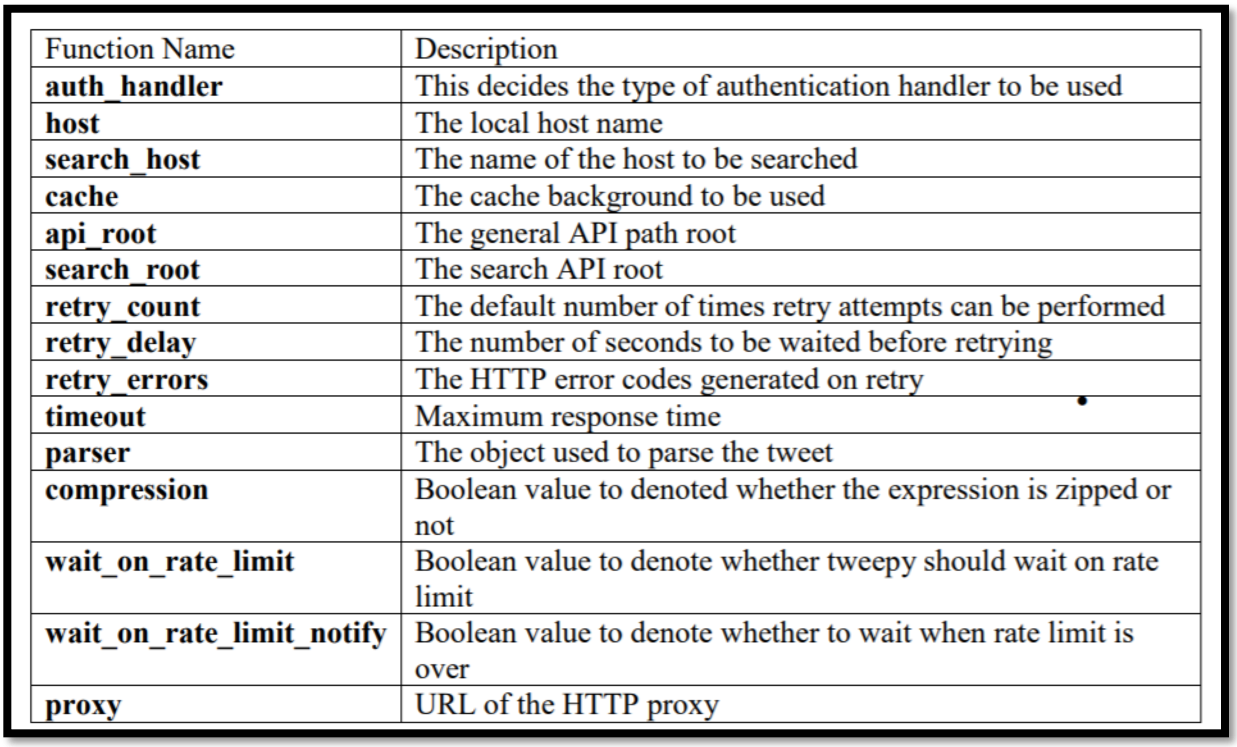


Figure 1: all data extraction steps

❖ Tweepy library

This provides a wrapper for the API as provided by Twitter to easily used by Python language. The functions provided in this class are listed below.



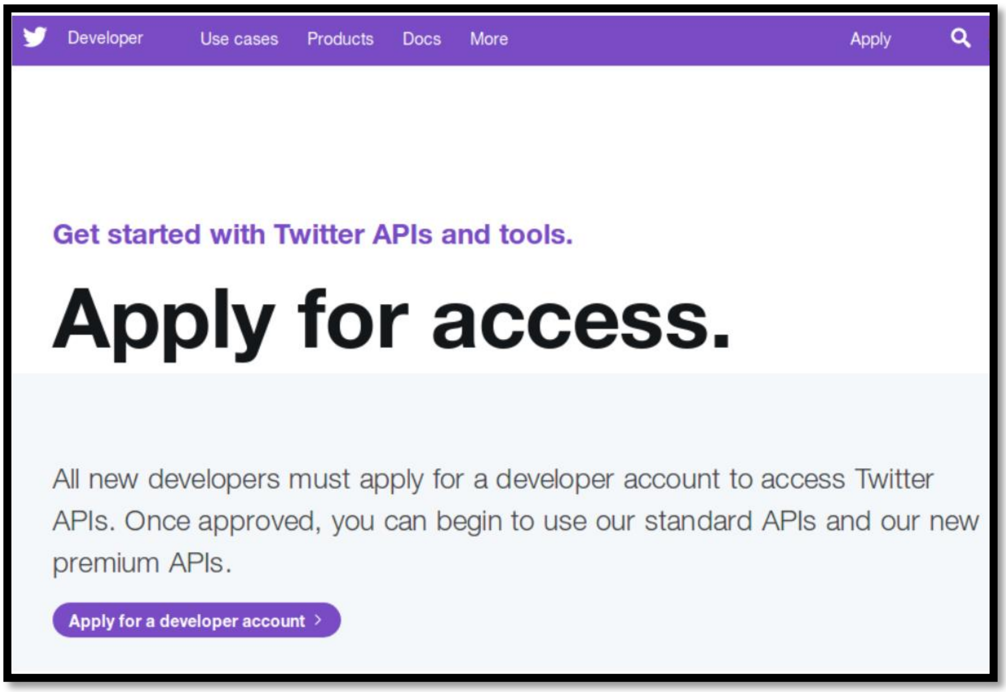
Function Name	Description
auth_handler	This decides the type of authentication handler to be used
host	The local host name
search_host	The name of the host to be searched
cache	The cache background to be used
api_root	The general API path root
search_root	The search API root
retry_count	The default number of times retry attempts can be performed
retry_delay	The number of seconds to be waited before retrying
retry_errors	The HTTP error codes generated on retry
timeout	Maximum response time
parser	The object used to parse the tweet
compression	Boolean value to denoted whether the expression is zipped or not
wait_on_rate_limit	Boolean value to denote whether tweepy should wait on rate limit
wait_on_rate_limit_notify	Boolean value to denote whether to wait when rate limit is over
proxy	URL of the HTTP proxy

Figure 2: all tweepy libraries

❖ Data access steps

➢ Step 1 : create an account on twitter developer

After you have applied for Developer Access, you can create an application in Twitter that you can use to access tweets. Make sure you already have a Twitter account.Apply for a devloper Account: If you don't have any apps, log in at « https://developer.twitter.com/ » with your Twitter username and password. Go to *Apply for a developer account*.

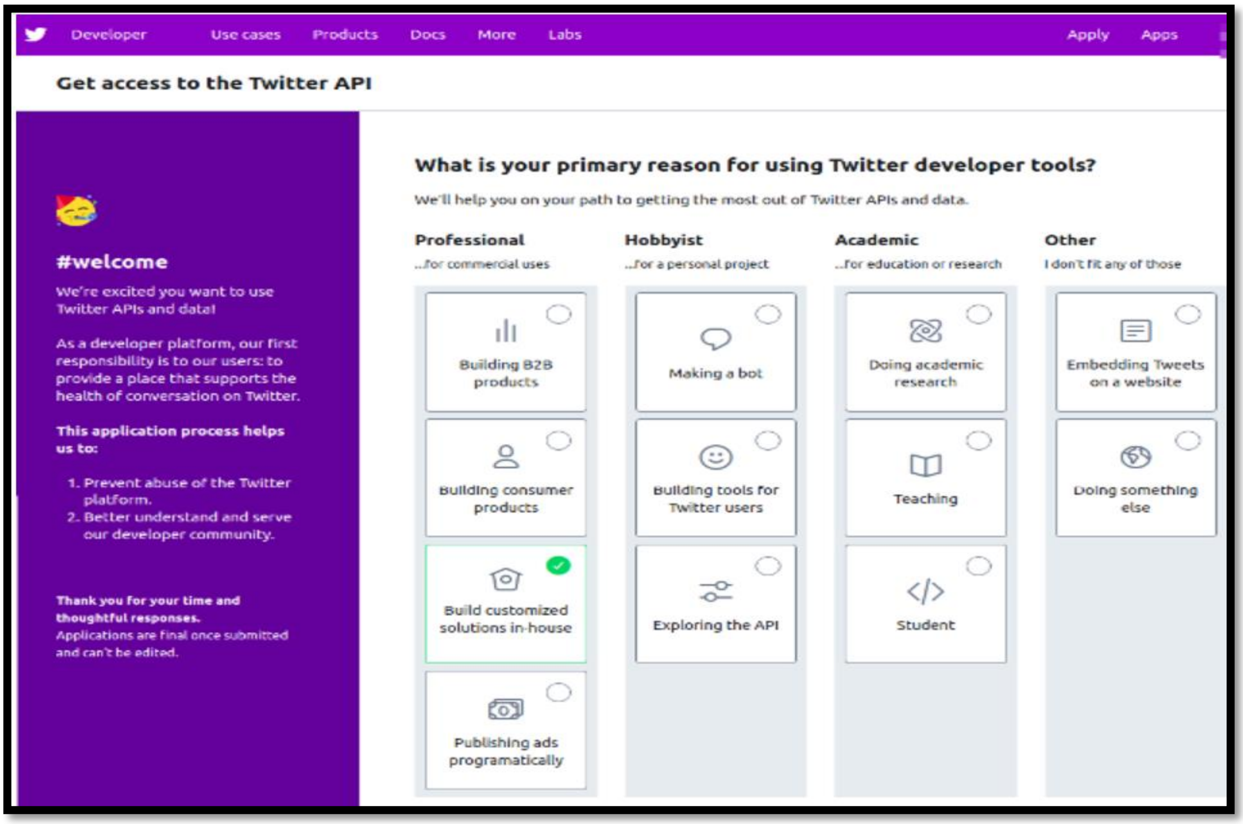


Get started with Twitter APIs and tools.

Apply for access.

All new developers must apply for a developer account to access Twitter APIs. Once approved, you can begin to use our standard APIs and our new premium APIs.

- If you already have apps, log in at « https://apps.twitter.com/ » with your Twitter username and password. Go to *Apply for a developer account*.
- Select your *Primary reason for using Twitter Developer Tools*. For auto post to Twitter select *Build customized solutions in-house*



Get access to the Twitter API

#welcome

We're excited you want to use Twitter APIs and data!

As a developer platform, our first responsibility is to our users: to provide a place that supports the health of conversation on Twitter.

This application process helps us to:

1. Prevent abuse of the Twitter platform.
2. Better understand and serve our developer community.

Thank you for your time and thoughtful responses.
Applications are final once submitted and can't be edited.

What is your primary reason for using Twitter developer tools?

We'll help you on your path to getting the most out of Twitter APIs and data.

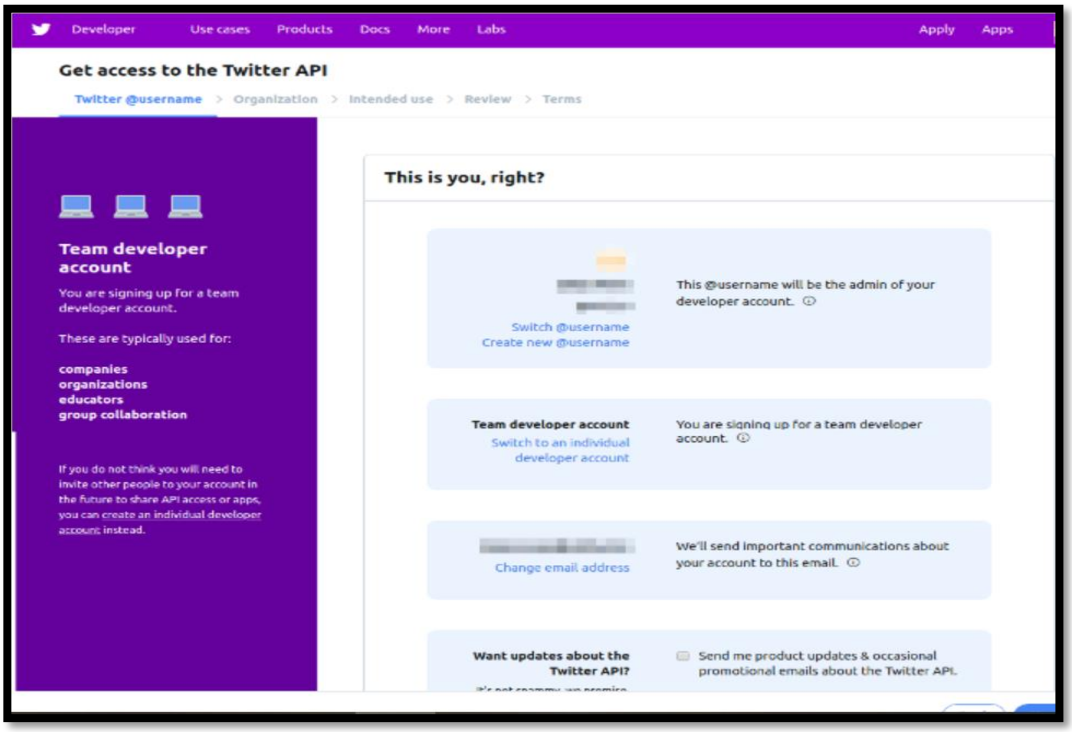
Professional ...for commercial uses
- ○ Building B2B products
- ○ Building consumer products
- ✓ Build customized solutions in-house
- ○ Publishing ads programatically

Hobbyist ...for a personal project
- ○ Making a bot
- ○ Building tools for Twitter users
- ○ Exploring the API

Academic ...for education or research
- ○ Doing academic research
- ○ Teaching
- ○ Student

Other I don't fit any of those
- ○ Embedding Tweets on a website
- ○ Doing something else

- Verify the *Twitter Username* details associated to the developer account.



Get access to the Twitter API

Twitter @username > Organization > Intended use > Review > Terms

Team developer account

You are signing up for a team developer account.

These are typically used for:

companies
organizations
educators
group collaboration

If you do not think you will need to invite other people to your account in the future to share API access or apps, you can create an individual developer account instead.

This is you, right?

[redacted]
Switch @username
Create new @username

This @username will be the admin of your developer account. ⓘ

Team developer account
Switch to an individual developer account

You are signing up for a team developer account. ⓘ

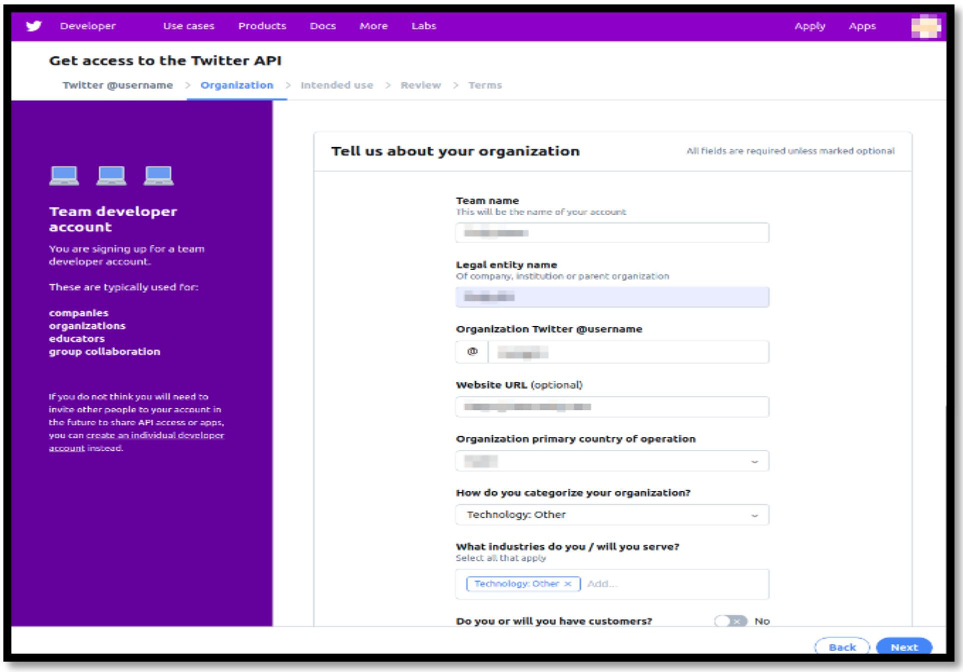
[redacted]
Change email address

We'll send important communications about your account to this email. ⓘ

Want updates about the Twitter API?

□ Send me product updates & occasional promotional emails about the Twitter API.

- Fill all the data required about your organization.



Developer Use cases Products Docs More Labs Apply Apps

Get access to the Twitter API

Twitter @username > Organization > Intended use > Review > Terms

Team developer account

You are signing up for a team developer account.

These are typically used for:

companies
organizations
educators
group collaboration

If you do not think you will need to invite other people to your account in the future to share API access or apps, you can create an individual developer account instead.

Tell us about your organization

All fields are required unless marked optional

Team name
This will be the name of your account

[redacted]

Legal entity name
Of company, institution or parent organization

[redacted]

Organization Twitter @username

@ [redacted]

Website URL (optional)

[redacted]

Organization primary country of operation

[redacted]

How do you categorize your organization?

Technology: Other

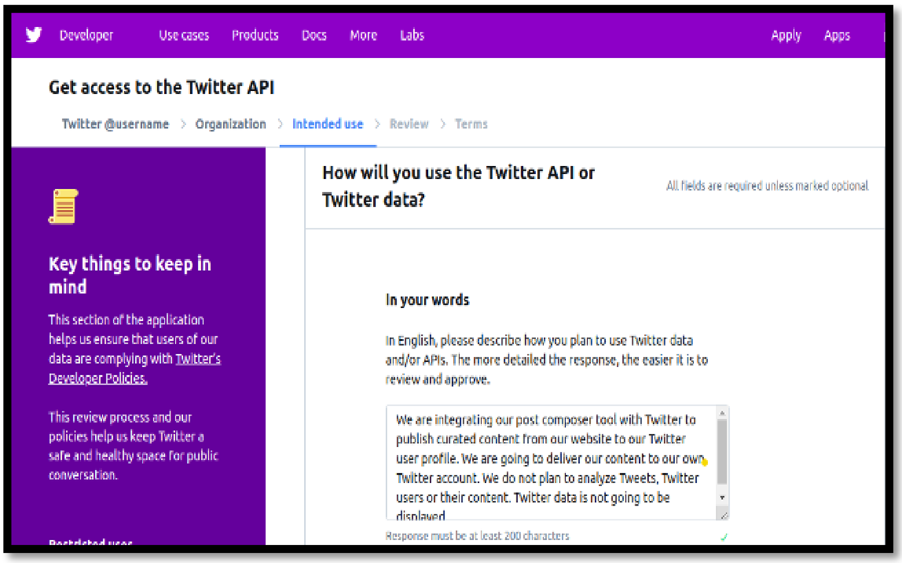
What industries do you / will you serve?
Select all that apply

Technology: Other ✕ Add...

Do you or will you have customers? No

Back Next

- Describe your *Intended Use* of the Twitter API



Developer Use cases Products Docs More Labs Apply Apps

Get access to the Twitter API

Twitter @username > Organization > Intended use > Review > Terms

Key things to keep in mind

This section of the application helps us ensure that users of our data are complying with Twitter's Developer Policies.

This review process and our policies help us keep Twitter a safe and healthy space for public conversation.

Restricted uses

How will you use the Twitter API or Twitter data?

All fields are required unless marked optional

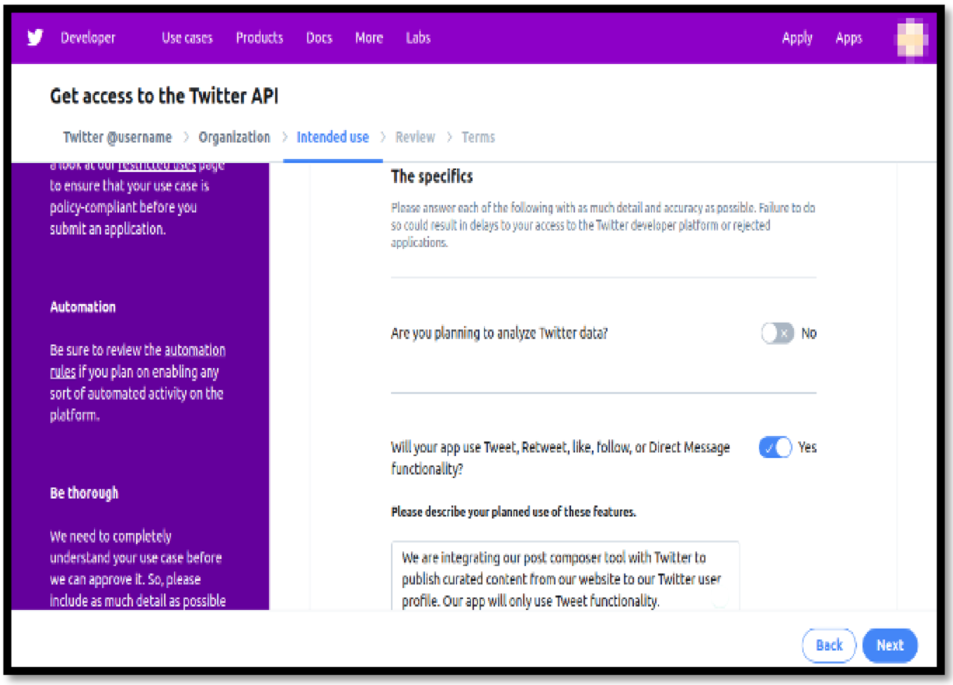
In your words

In English, please describe how you plan to use Twitter data and/or APIs. The more detailed the response, the easier it is to review and approve.

We are integrating our post composer tool with Twitter to publish curated content from our website to our Twitter user profile. We are going to deliver our content to our own Twitter account. We do not plan to analyze Tweets, Twitter users or their content. Twitter data is not going to be displayed

Response must be at least 200 characters ✓

- Describe in your words how you plan to use the Twitter API.



Get access to the Twitter API

Twitter @username › Organization › Intended use › Review › Terms

a look at our restricted uses page to ensure that your use case is policy-compliant before you submit an application.

Automation

Be sure to review the automation rules if you plan on enabling any sort of automated activity on the platform.

Be thorough

We need to completely understand your use case before we can approve it. So, please include as much detail as possible

The specifics

Please answer each of the following with as much detail and accuracy as possible. Failure to do so could result in delays to your access to the Twitter developer platform or rejected applications.

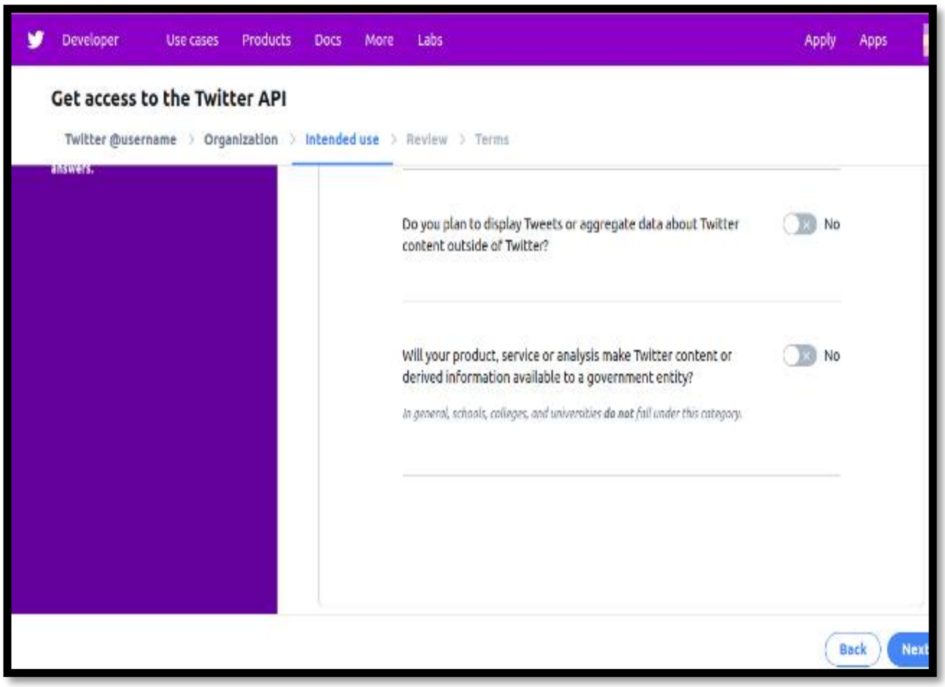
Are you planning to analyze Twitter data? ○ No

Will your app use Tweet, Retweet, like, follow, or Direct Message functionality? ✓ Yes

Please describe your planned use of these features.

We are integrating our post composer tool with Twitter to publish curated content from our website to our Twitter user profile. Our app will only use Tweet functionality.

- Describe your planned use of these feaures.



Get access to the Twitter API

Twitter @username › Organization › Intended use › Review › Terms

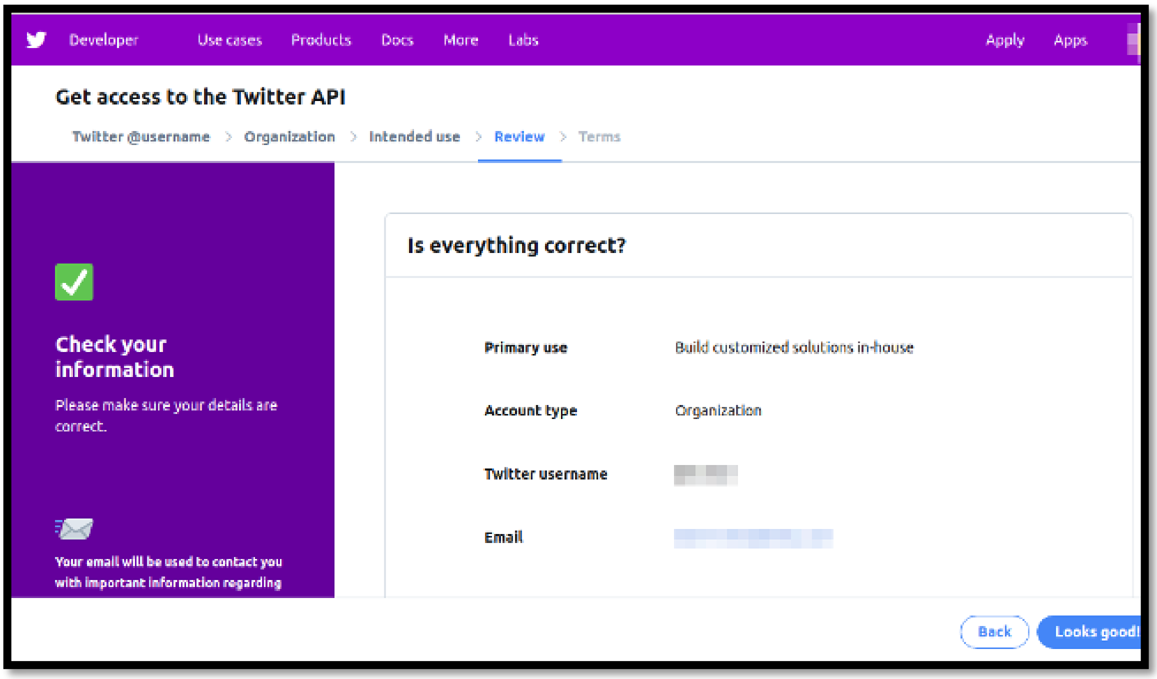
answers.

Do you plan to display Tweets or aggregate data about Twitter content outside of Twitter? ○ No

Will your product, service or analysis make Twitter content or derived information available to a government entity? ○ No

*In general, schools, colleges, and universities **do not** fall under this category.*

- Check your information



Get access to the Twitter API

Twitter @username > Organization > Intended use > Review > Terms

☑ Check your information

Please make sure your details are correct.

Your email will be used to contact you with important information regarding

Is everything correct?

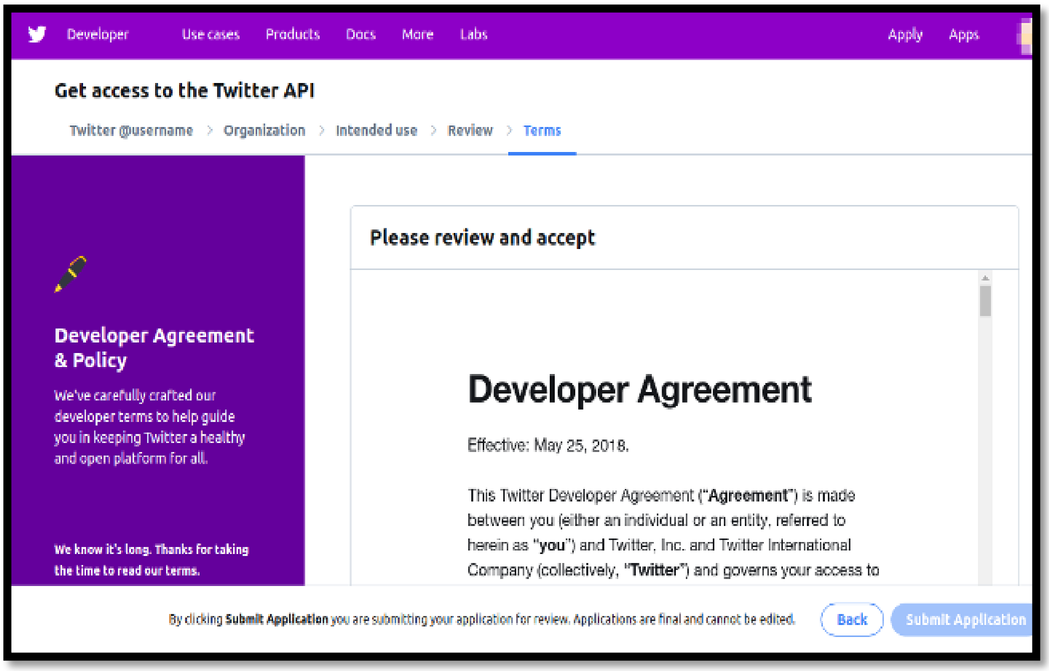
Primary use: Build customized solutions in-house

Account type: Organization

Twitter username: [redacted]

Email: [redacted]

- Accept the *Developer Agreement* and verify your email account.



Get access to the Twitter API

Twitter @username > Organization > Intended use > Review > Terms

Developer Agreement & Policy

We've carefully crafted our developer terms to help guide you in keeping Twitter a healthy and open platform for all.

We know it's long. Thanks for taking the time to read our terms.

Please review and accept

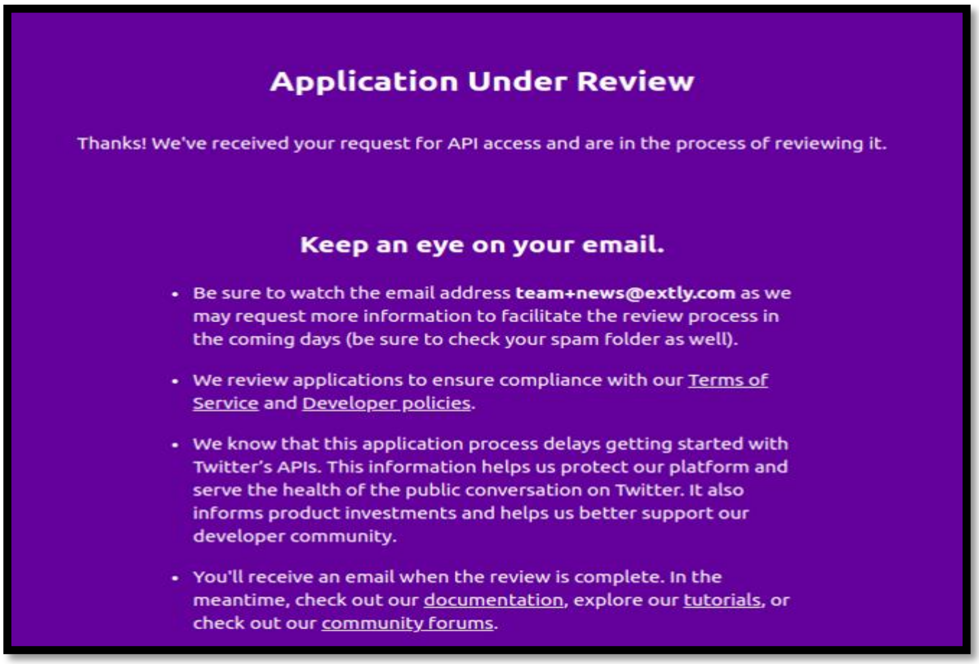
Developer Agreement

Effective: May 25, 2018.

This Twitter Developer Agreement (“**Agreement**”) is made between you (either an individual or an entity, referred to herein as “**you**”) and Twitter, Inc. and Twitter International Company (collectively, “**Twitter**”) and governs your access to

By clicking **Submit Application** you are submitting your application for review. Applications are final and cannot be edited.

- Your application is under review, and you will receive a notification with the result.



Application Under Review

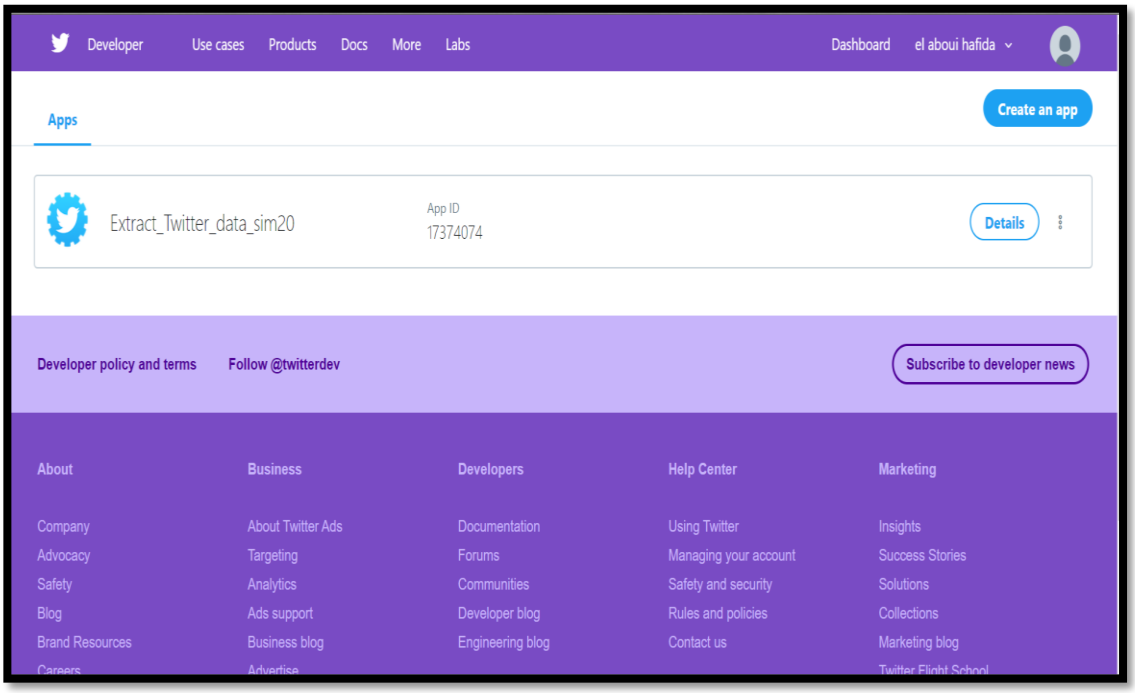
Thanks! We've received your request for API access and are in the process of reviewing it.

Keep an eye on your email.

- Be sure to watch the email address **team+news@extly.com** as we may request more information to facilitate the review process in the coming days (be sure to check your spam folder as well).
- We review applications to ensure compliance with our Terms of Service and Developer policies.
- We know that this application process delays getting started with Twitter's APIs. This information helps us protect our platform and serve the health of the public conversation on Twitter. It also informs product investments and helps us better support our developer community.
- You'll receive an email when the review is complete. In the meantime, check out our documentation, explore our tutorials, or check out our community forums.

➢ Step 2 : create a Twitter application

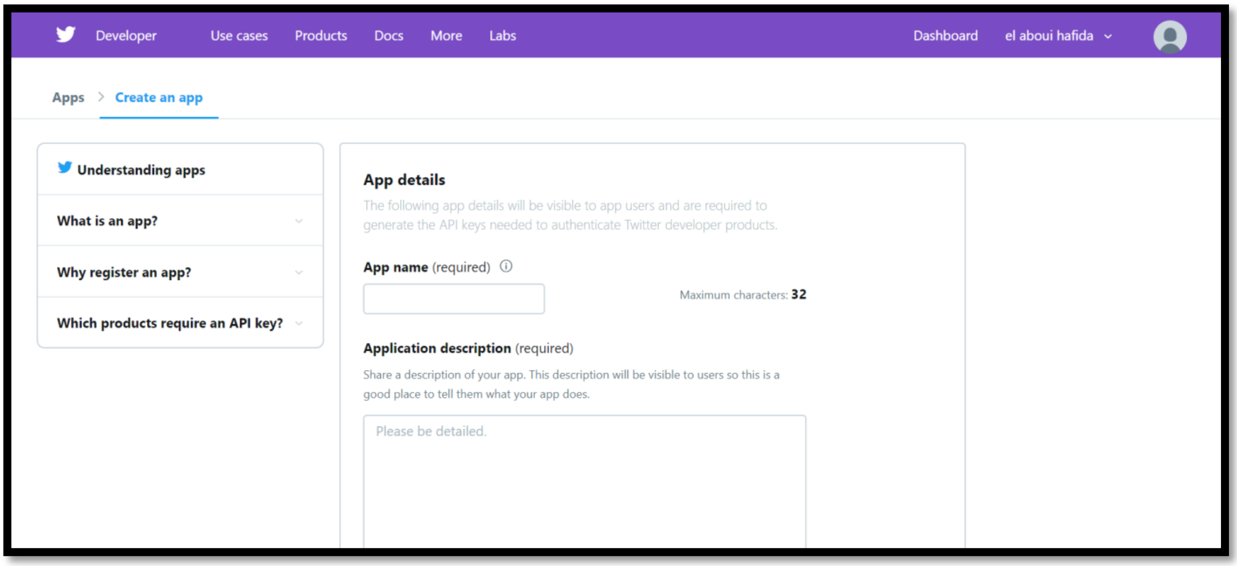
1. Visit the Twitter Developers Site



The first thing you need to do is head on down to « dev.twitter.com ». In order to create an account, all you need to do is click on the “Sign In” link at the top right.

3. Go to « crete an app »

If you are new to the Developers site you won't see any applications registered. Either way, it's time to create our first application. To do this, click on the big "Create a new application" button.



App details

The following app details will be visible to app users and are required to generate the API keys needed to authenticate Twitter developer products.

App name (required)

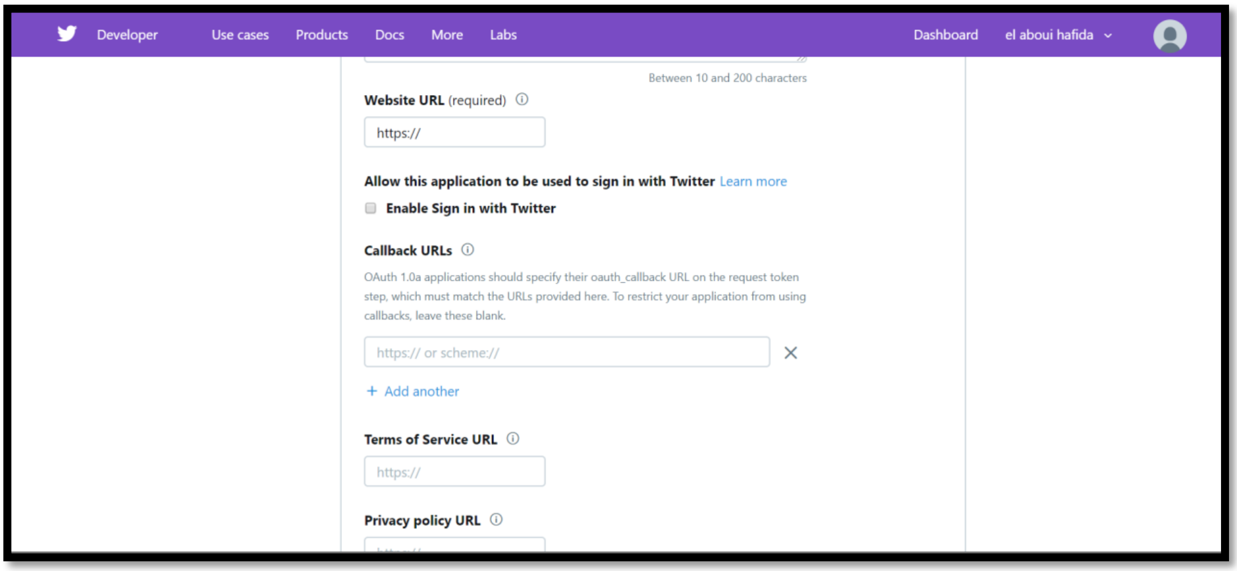
Maximum characters: 32

Application description (required)

Share a description of your app. This description will be visible to users so this is a good place to tell them what your app does.

Please be detailed.

- Put **your website** in the website field- don't worry that it isn't (as Twitter ask) your application's publicly accessible home page. However, this website will be where your app is hosted.
- For now **ignore the Callback URL field**. If you are allowing users to log into your app to authenticate themselves, you'd enter the URL where they would be returned after they've given permission to Twitter to use your app.
- Once you've done this, make sure you've read the "Developer Rules Of The Road" blurb, check the "Yes, I agree" box, fill in the CAPTCHA (don't you just love them) and click the "create your Twitter Application" button. Hurrah!



Between 10 and 200 characters

Website URL (required)

https://

Allow this application to be used to sign in with Twitter Learn more

□ **Enable Sign in with Twitter**

Callback URLs

OAuth 1.0a applications should specify their oauth_callback URL on the request token step, which must match the URLs provided here. To restrict your application from using callbacks, leave these blank.

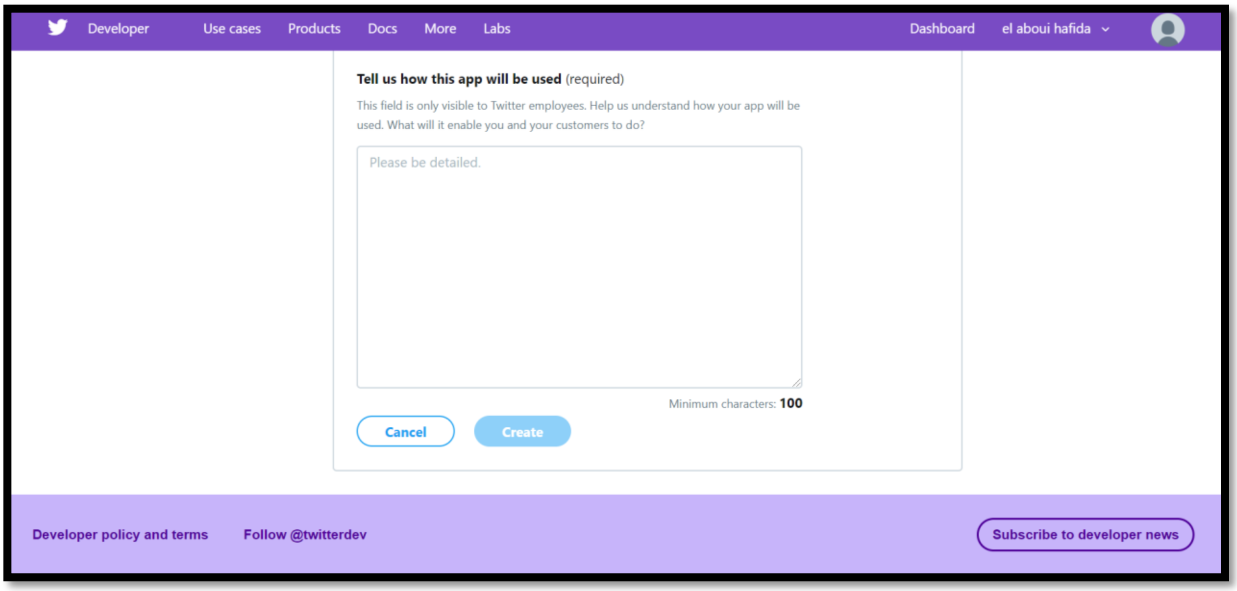
https:// or scheme://

+ Add another

Terms of Service URL

https://

Privacy policy URL



Tell us how this app will be used (required)

This field is only visible to Twitter employees. Help us understand how your app will be used. What will it enable you and your customers to do?

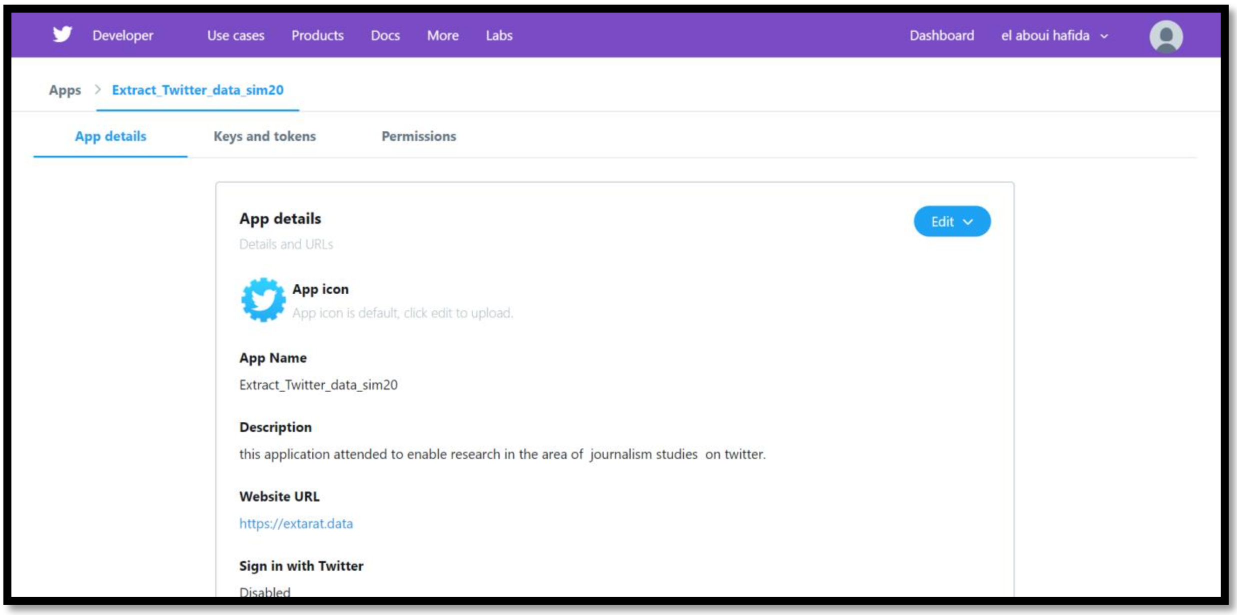
Please be detailed.

Minimum characters: **100**

Once you’ve done this, make a note of your OAuth settings.

- Consumer Key
- Consumer Secret
- OAuth Access Token
- OAuth Access Token Secret

It goes without saying that you should keep these secret. If anyone was to get these keys, they could effectively access your Twitter account.



App details

Details and URLs

App icon

App icon is default, click edit to upload.

App Name

Extract_Twitter_data_sim20

Description

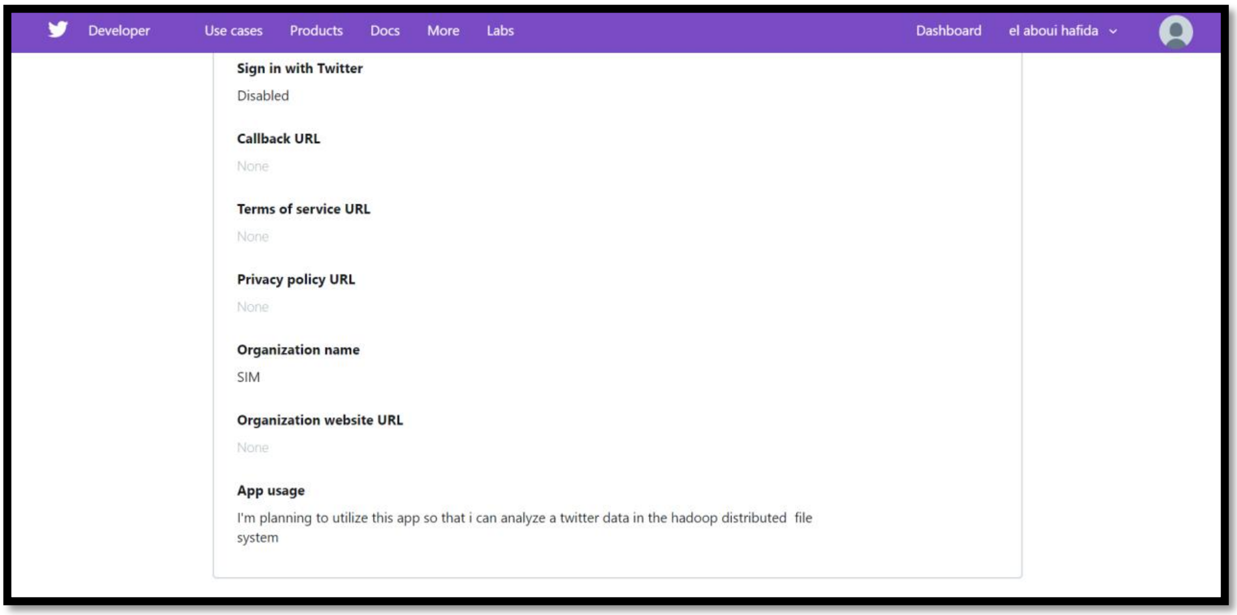
this application attended to enable research in the area of journalism studies on twitter.

Website URL

https://extarat.data

Sign in with Twitter

Disabled



Sign in with Twitter

Disabled

Callback URL

None

Terms of service URL

None

Privacy policy URL

None

Organization name

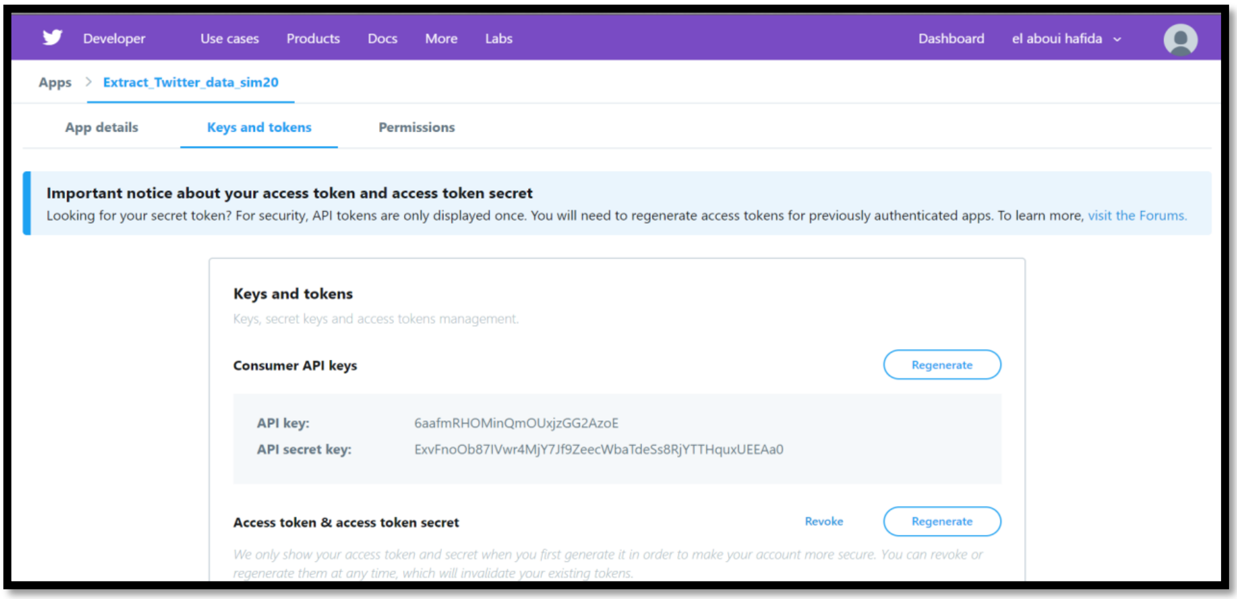
SIM

Organization website URL

None

App usage

I'm planning to utilize this app so that i can analyze a twitter data in the hadoop distributed file system



Important notice about your access token and access token secret

Looking for your secret token? For security, API tokens are only displayed once. You will need to regenerate access tokens for previously authenticated apps. To learn more, visit the Forums.

Keys and tokens

Keys, secret keys and access tokens management.

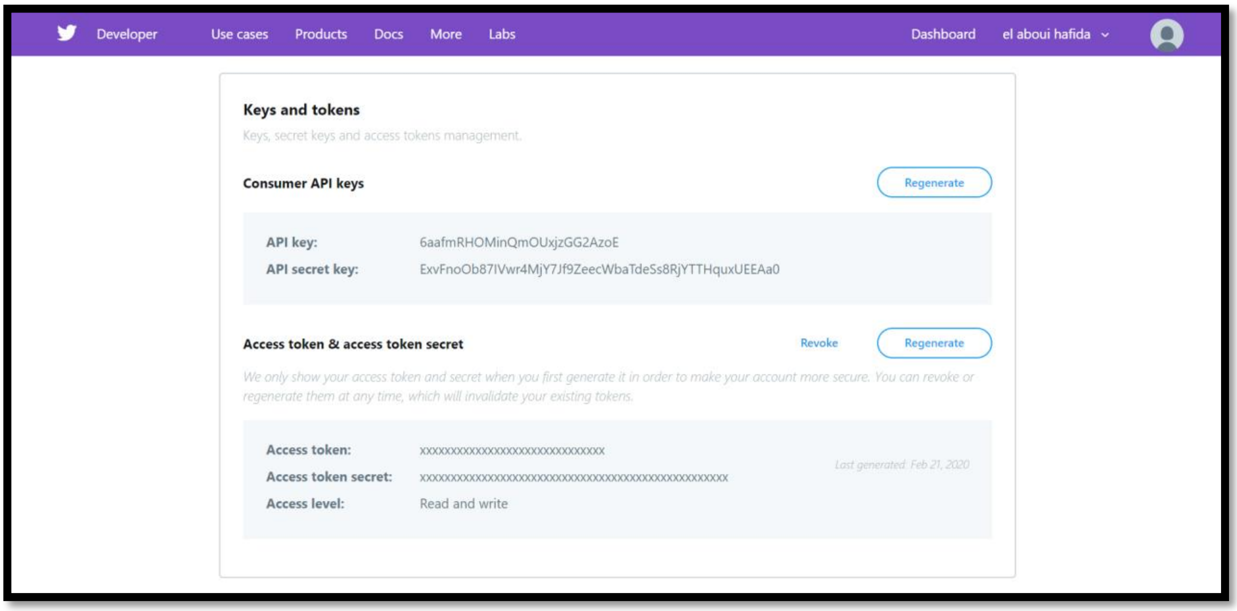
Consumer API keys

API key: 6aafmRHOMinQmOUxjzGG2AzoE

API secret key: ExvFnoOb87IVwr4MjY7Jf9ZeecWbaTdeSs8RjYTTHquxUEEAa0

Access token & access token secret

We only show your access token and secret when you first generate it in order to make your account more secure. You can revoke or regenerate them at any time, which will invalidate your existing tokens.



Keys and tokens

Keys, secret keys and access tokens management.

Consumer API keys

API key: 6aafmRHOMinQmOUxjzGG2AzoE

API secret key: ExvFnoOb87IVwr4MjY7Jf9ZeecWbaTdeSs8RjYTTHquxUEEAa0

Access token & access token secret

We only show your access token and secret when you first generate it in order to make your account more secure. You can revoke or regenerate them at any time, which will invalidate your existing tokens.

Access token: xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

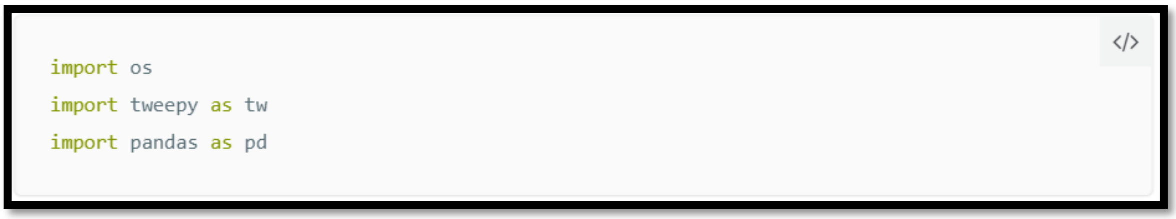
Access token secret: xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

Access level: Read and write

Last generated: Feb 21, 2020

➢ Step 3 : Access Twitter API in Python

- Once you have your Twitter app set-up, you are ready to access tweets in Python. Begin by importing the necessary Python libraries.



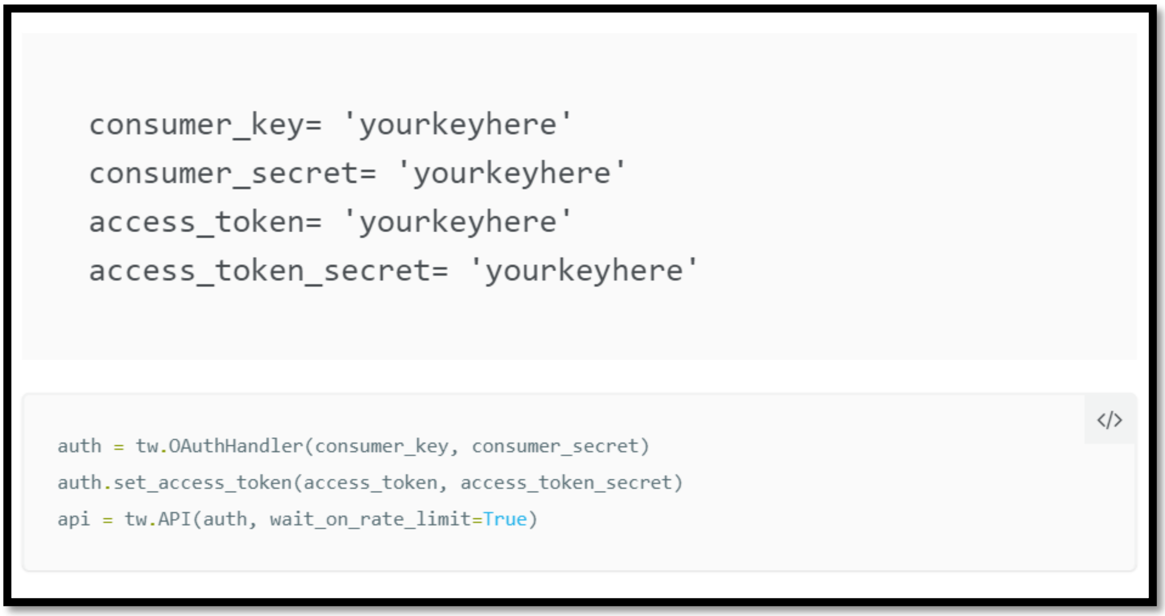
```
import os
import tweepy as tw
import pandas as pd
```

- To access the Twitter API, you will need 4 things from the your Twitter App page. These keys are located in your Twitter app settings in the Keys and Access Tokens tab.

  - consumer key
  - consumer seceret key
  - access token key
  - access token secret key

Do not share these with anyone else because these values are specific to your app.

- First you will need define your keys

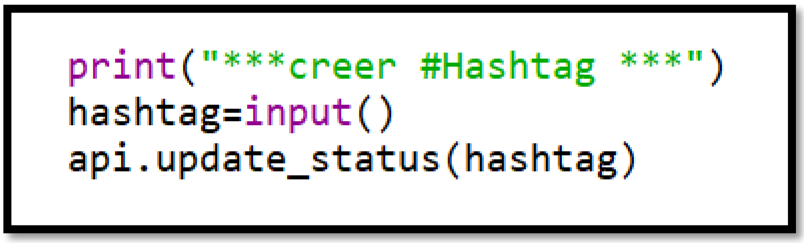


```
consumer_key= 'yourkeyhere'
consumer_secret= 'yourkeyhere'
access_token= 'yourkeyhere'
access_token_secret= 'yourkeyhere'
```

```
auth = tw.OAuthHandler(consumer_key, consumer_secret)
auth.set_access_token(access_token, access_token_secret)
api = tw.API(auth, wait_on_rate_limit=True)
```

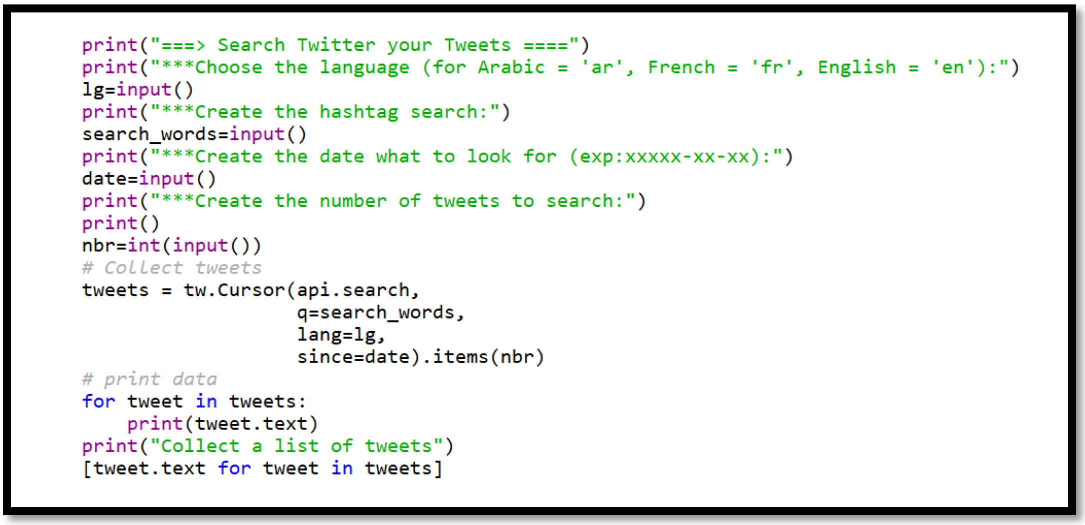
- **Send a Tweet**

You can send tweets using your API access. Note that your tweet needs to be 280 characters or less.



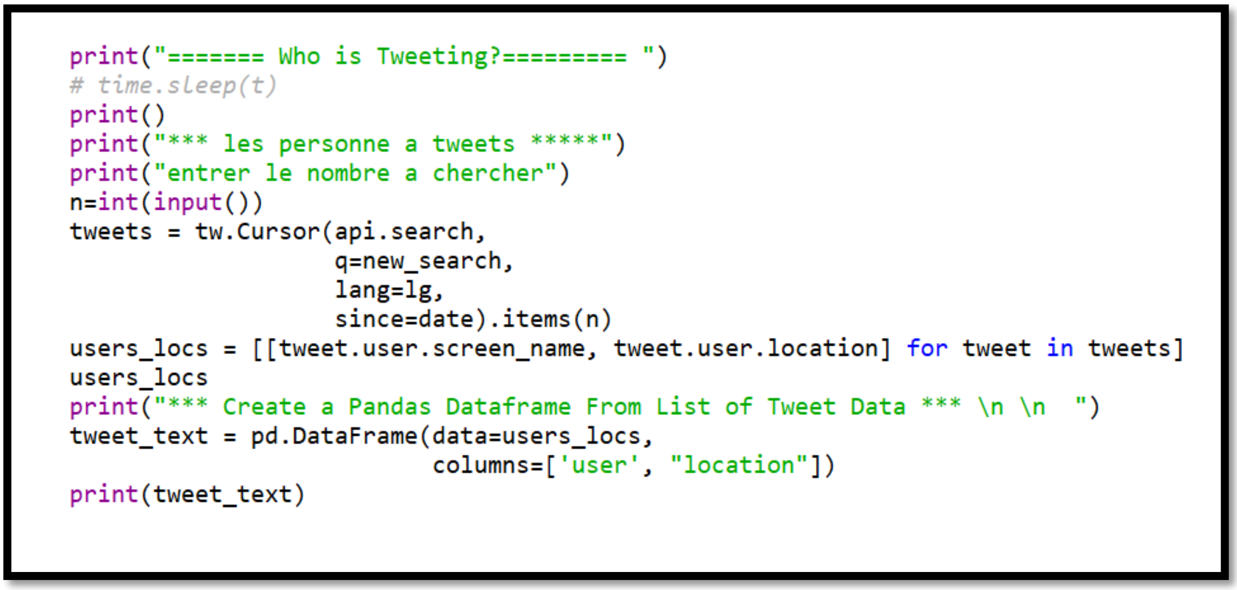
```
print("***creer #Hashtag ***")
hashtag=input()
api.update_status(hashtag)
```

- **Search Twitter for Tweets and crete a data**



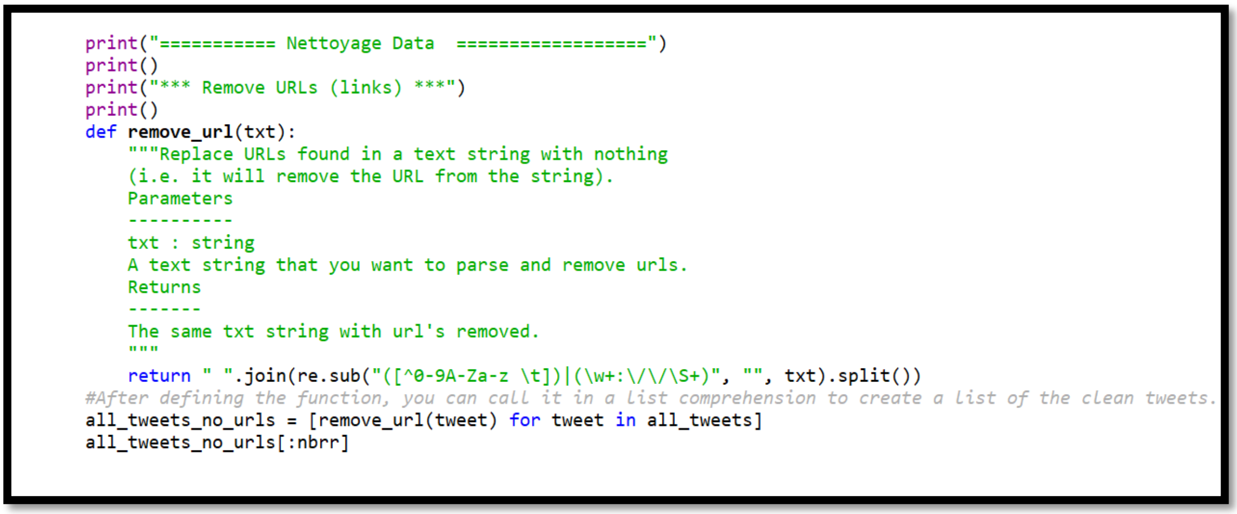
```
print("===> Search Twitter your Tweets ====")
print("***Choose the language (for Arabic = 'ar', French = 'fr', English = 'en'):")
lg=input()
print("***Create the hashtag search:")
search_words=input()
print("***Create the date what to look for (exp:xxxxx-xx-xx):")
date=input()
print("***Create the number of tweets to search:")
print()
nbr=int(input())
# Collect tweets
tweets = tw.Cursor(api.search,
                   q=search_words,
                   lang=lg,
                   since=date).items(nbr)
# print data
for tweet in tweets:
    print(tweet.text)
print("Collect a list of tweets")
[tweet.text for tweet in tweets]
```

○ Who is Tweeting



```
print("======= Who is Tweeting?========= ")
# time.sleep(t)
print()
print("*** les personne a tweets *****")
print("entrer le nombre a chercher")
n=int(input())
tweets = tw.Cursor(api.search,
                   q=new_search,
                   lang=lg,
                   since=date).items(n)
users_locs = [[tweet.user.screen_name, tweet.user.location] for tweet in tweets]
users_locs
print("*** Create a Pandas Dataframe From List of Tweet Data *** \n \n  ")
tweet_text = pd.DataFrame(data=users_locs,
                          columns=['user', "location"])
print(tweet_text)
```

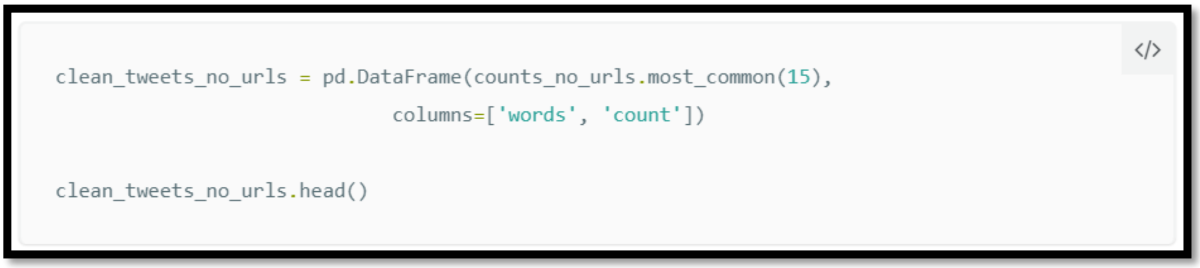
○ Remove URLs (links)



```
print("=========== Nettoyage Data  ==================")
print()
print("*** Remove URLs (links) ***")
print()
def remove_url(txt):
    """Replace URLs found in a text string with nothing
    (i.e. it will remove the URL from the string).
    Parameters
    ----------
    txt : string
    A text string that you want to parse and remove urls.
    Returns
    -------
    The same txt string with url's removed.
    """
    return " ".join(re.sub("([^0-9A-Za-z \t])|(\w+:\/\/\S+)", "", txt).split())
#After defining the function, you can call it in a list comprehension to create a list of the clean tweets.
all_tweets_no_urls = [remove_url(tweet) for tweet in all_tweets]
all_tweets_no_urls[:nbrr]
```

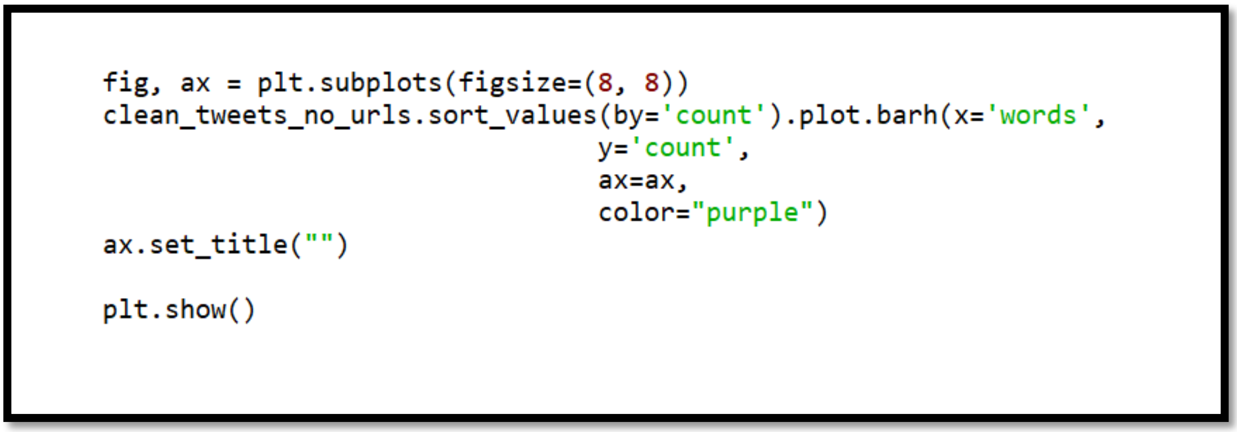
○ Calculate and Plot Word Frequency

Calculate a word frequency



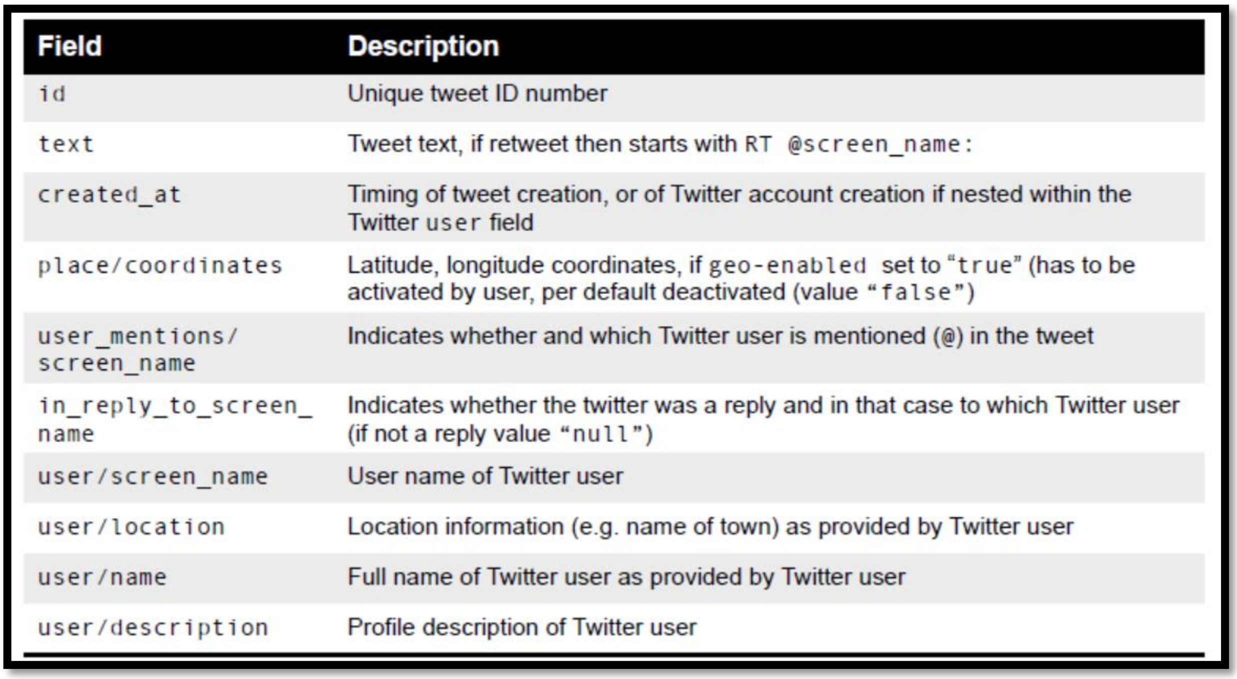
```
clean_tweets_no_urls = pd.DataFrame(counts_no_urls.most_common(15),
                                    columns=['words', 'count'])


clean_tweets_no_urls.head()
```

Plot a word frequency



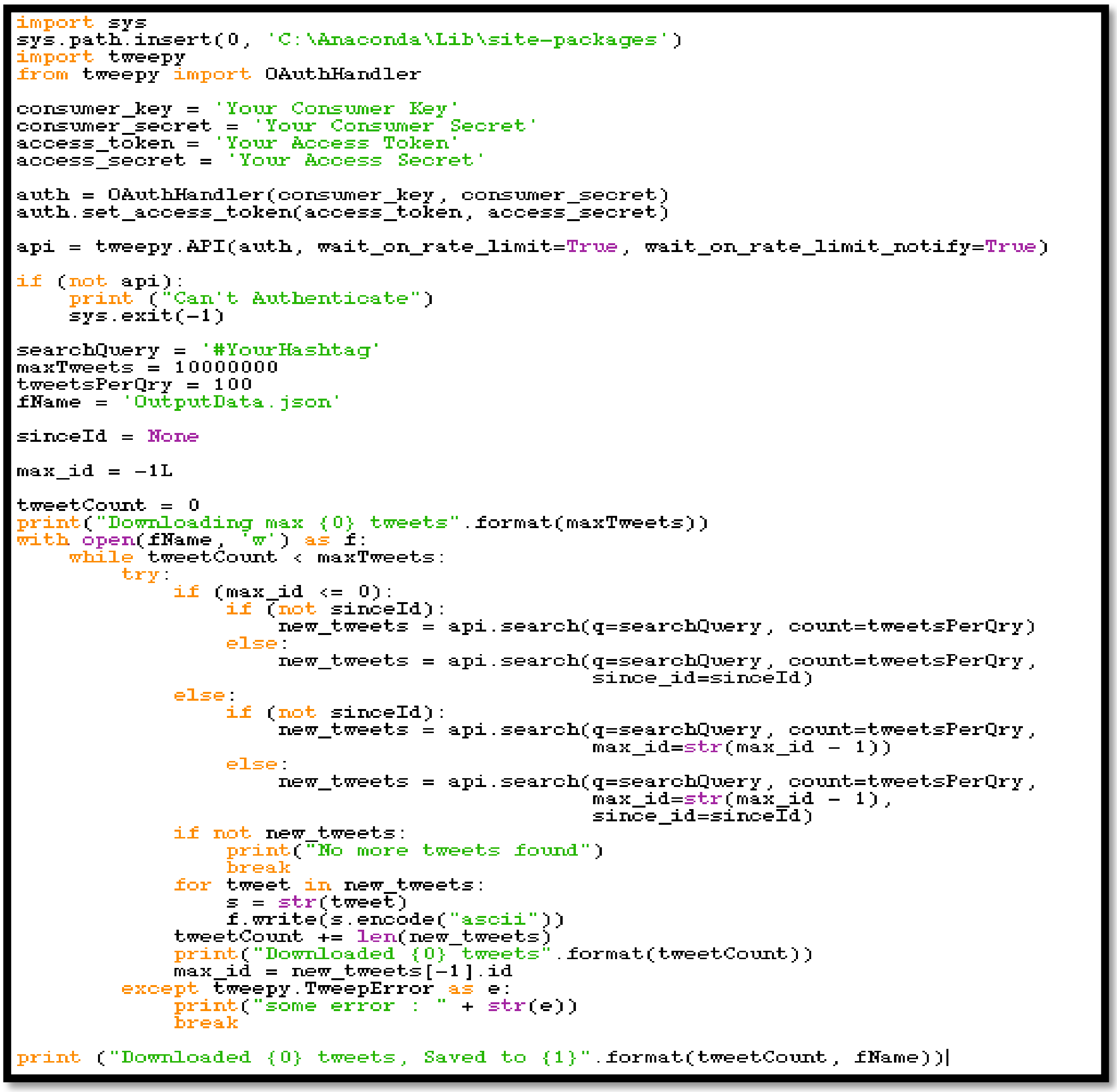
```
fig, ax = plt.subplots(figsize=(8, 8))
clean_tweets_no_urls.sort_values(by='count').plot.barh(x='words',
                                                       y='count',
                                                       ax=ax,
                                                       color="purple")
ax.set_title("")

plt.show()
```

○ Twitter data, key variables



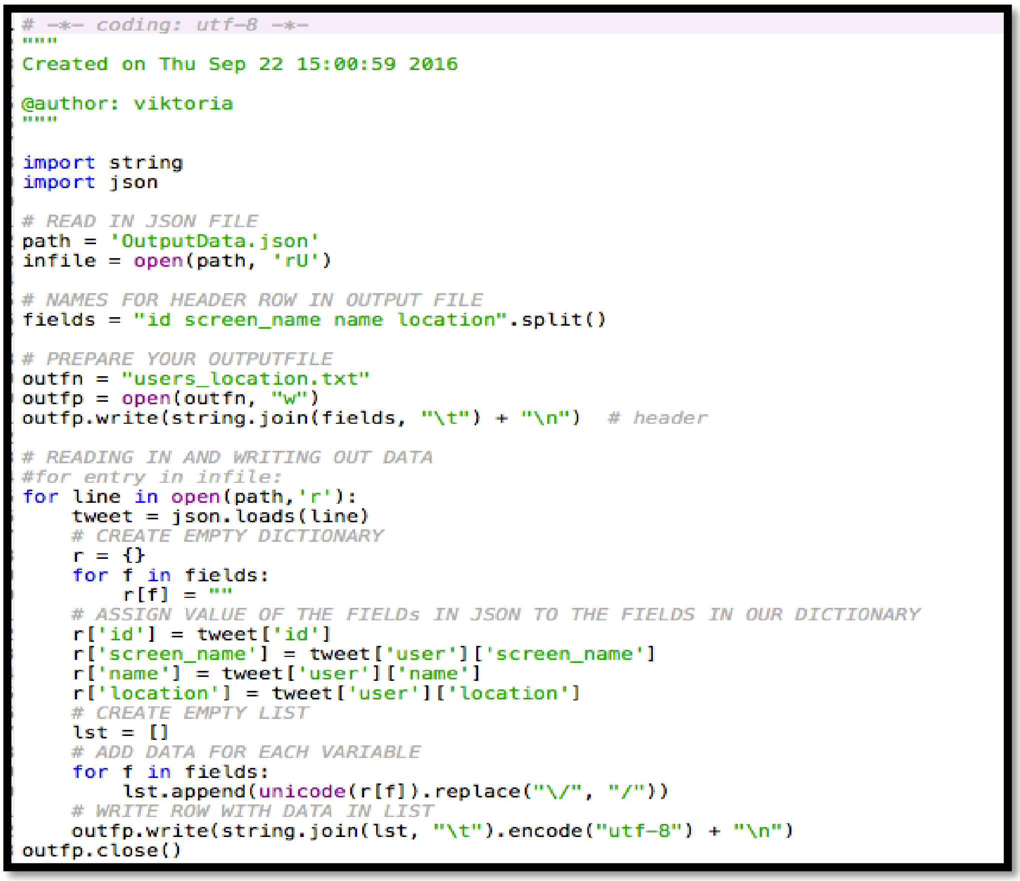
Field	Description
id	Unique tweet ID number
text	Tweet text, if retweet then starts with RT @screen_name:
created_at	Timing of tweet creation, or of Twitter account creation if nested within the Twitter user field
place/coordinates	Latitude, longitude coordinates, if geo-enabled set to "true" (has to be activated by user, per default deactivated (value "false")
user_mentions/ screen_name	Indicates whether and which Twitter user is mentioned (@) in the tweet
in_reply_to_screen_name	Indicates whether the twitter was a reply and in that case to which Twitter user (if not a reply value "null")
user/screen_name	User name of Twitter user
user/location	Location information (e.g. name of town) as provided by Twitter user
user/name	Full name of Twitter user as provided by Twitter user
user/description	Profile description of Twitter user

- ## Getting data from the Search API



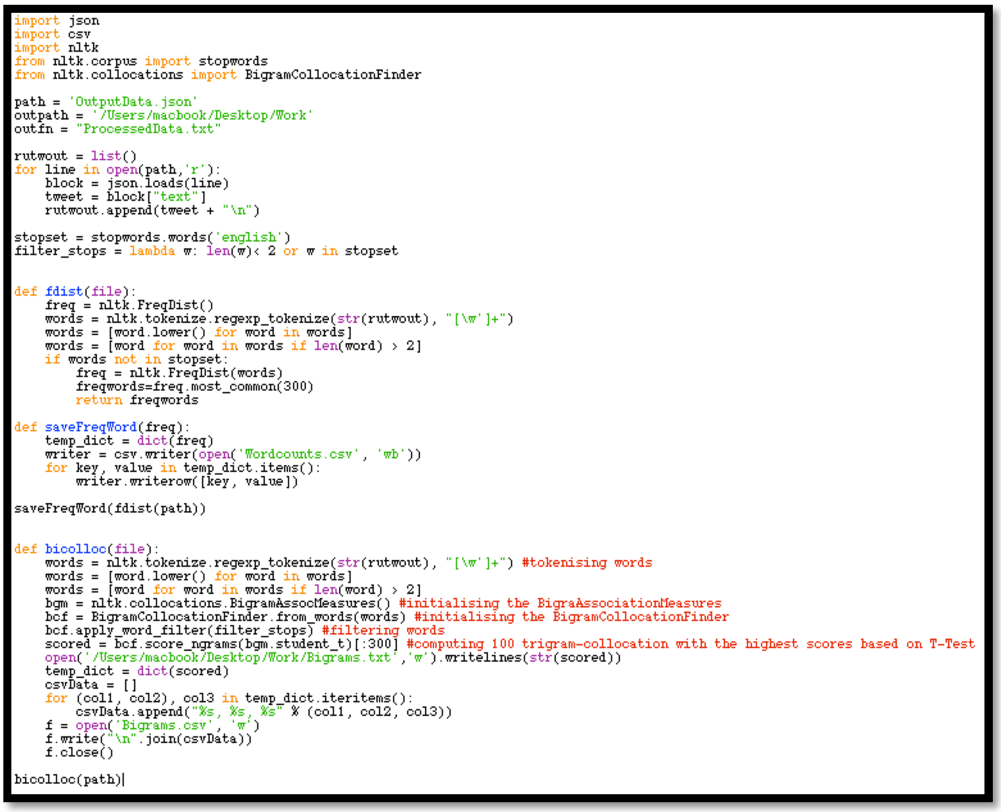
```
import sys
sys.path.insert(0, 'C:\Anaconda\Lib\site-packages')
import tweepy
from tweepy import OAuthHandler

consumer_key = 'Your Consumer Key'
consumer_secret = 'Your Consumer Secret'
access_token = 'Your Access Token'
access_secret = 'Your Access Secret'

auth = OAuthHandler(consumer_key, consumer_secret)
auth.set_access_token(access_token, access_secret)

api = tweepy.API(auth, wait_on_rate_limit=True, wait_on_rate_limit_notify=True)

if (not api):
    print ("Can't Authenticate")
    sys.exit(-1)

searchQuery = '#YourHashtag'
maxTweets = 10000000
tweetsPerQry = 100
fName = 'OutputData.json'

sinceId = None

max_id = -1L

tweetCount = 0
print("Downloading max {0} tweets".format(maxTweets))
with open(fName, 'w') as f:
    while tweetCount < maxTweets:
        try:
            if (max_id <= 0):
                if (not sinceId):
                    new_tweets = api.search(q=searchQuery, count=tweetsPerQry)
                else:
                    new_tweets = api.search(q=searchQuery, count=tweetsPerQry,
                                            since_id=sinceId)
            else:
                if (not sinceId):
                    new_tweets = api.search(q=searchQuery, count=tweetsPerQry,
                                            max_id=str(max_id - 1))
                else:
                    new_tweets = api.search(q=searchQuery, count=tweetsPerQry,
                                            max_id=str(max_id - 1),
                                            since_id=sinceId)
            if not new_tweets:
                print("No more tweets found")
                break
            for tweet in new_tweets:
                s = str(tweet)
                f.write(s.encode("ascii"))
            tweetCount += len(new_tweets)
            print("Downloaded {0} tweets".format(tweetCount))
            max_id = new_tweets[-1].id
        except tweepy.TweepError as e:
            print("some error : " + str(e))
            break

print ("Downloaded {0} tweets, Saved to {1}".format(tweetCount, fName))
```

- ## Processing JSON data



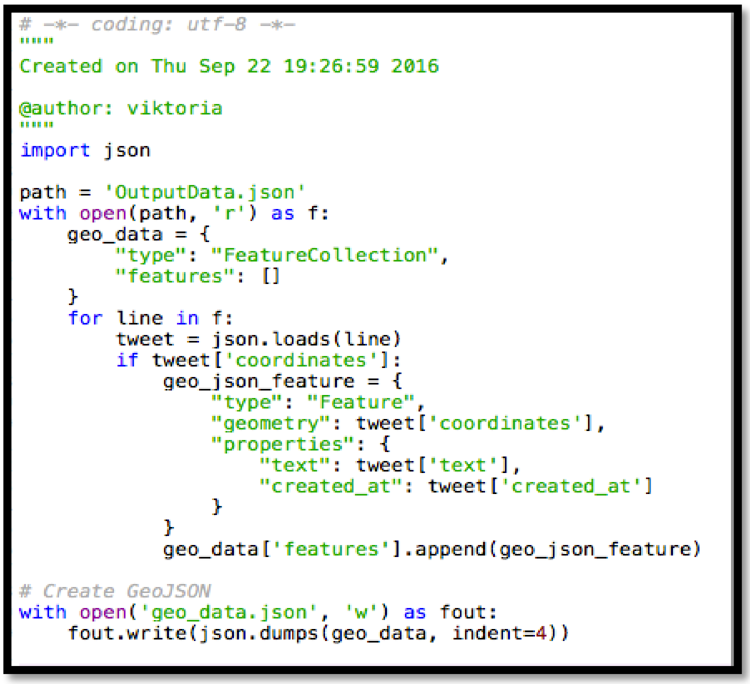
```
# -*- coding: utf-8 -*-
"""
Created on Thu Sep 22 15:00:59 2016

@author: viktoria
"""

import string
import json

# READ IN JSON FILE
path = 'OutputData.json'
infile = open(path, 'rU')

# NAMES FOR HEADER ROW IN OUTPUT FILE
fields = "id screen_name name location".split()

# PREPARE YOUR OUTPUTFILE
outfn = "users_location.txt"
outfp = open(outfn, "w")
outfp.write(string.join(fields, "\t") + "\n")  # header

# READING IN AND WRITING OUT DATA
#for entry in infile:
for line in open(path,'r'):
    tweet = json.loads(line)
    # CREATE EMPTY DICTIONARY
    r = {}
    for f in fields:
        r[f] = ""
    # ASSIGN VALUE OF THE FIELDs IN JSON TO THE FIELDS IN OUR DICTIONARY
    r['id'] = tweet['id']
    r['screen_name'] = tweet['user']['screen_name']
    r['name'] = tweet['user']['name']
    r['location'] = tweet['user']['location']
    # CREATE EMPTY LIST
    lst = []
    # ADD DATA FOR EACH VARIABLE
    for f in fields:
        lst.append(unicode(r[f]).replace("\/", "/"))
    # WRITE ROW WITH DATA IN LIST
    outfp.write(string.join(lst, "\t").encode("utf-8") + "\n")
outfp.close()
```

- ## Natural Language Processing



```
import json
import csv
import nltk
from nltk.corpus import stopwords
from nltk.collocations import BigramCollocationFinder

path = 'OutputData.json'
outpath = '/Users/macbook/Desktop/Work'
outfn = "ProcessedData.txt"

rutwout = list()
for line in open(path,'r'):
    block = json.loads(line)
    tweet = block["text"]
    rutwout.append(tweet + "\n")

stopset = stopwords.words('english')
filter_stops = lambda w: len(w)< 2 or w in stopset


def fdist(file):
    freq = nltk.FreqDist()
    words = nltk.tokenize.regexp_tokenize(str(rutwout), "[\w']+")
    words = [word.lower() for word in words]
    words = [word for word in words if len(word) > 2]
    if words not in stopset:
        freq = nltk.FreqDist(words)
        freqwords=freq.most_common(300)
        return freqwords

def saveFreqWord(freq):
    temp_dict = dict(freq)
    writer = csv.writer(open('Wordcounts.csv', 'wb'))
    for key, value in temp_dict.items():
        writer.writerow([key, value])

saveFreqWord(fdist(path))


def bicolloc(file):
    words = nltk.tokenize.regexp_tokenize(str(rutwout), "[\w']+") #tokenising words
    words = [word.lower() for word in words]
    words = [word for word in words if len(word) > 2]
    bgm = nltk.collocations.BigramAssocMeasures() #initialising the BigraAssociationMeasures
    bcf = BigramCollocationFinder.from_words(words) #initialising the BigramCollocationFinder
    bcf.apply_word_filter(filter_stops) #filtering words
    scored = bcf.score_ngrams(bgm.student_t)[:300] #computing 100 trigram-collocation with the highest scores based on T-Test
    open('/Users/macbook/Desktop/Work/Bigrams.txt','w').writelines(str(scored))
    temp_dict = dict(scored)
    csvData = []
    for (col1, col2), col3 in temp_dict.iteritems():
        csvData.append("%s, %s, %s" % (col1, col2, col3))
    f = open('Bigrams.csv', 'w')
    f.write("\n".join(csvData))
    f.close()

bicolloc(path)
```

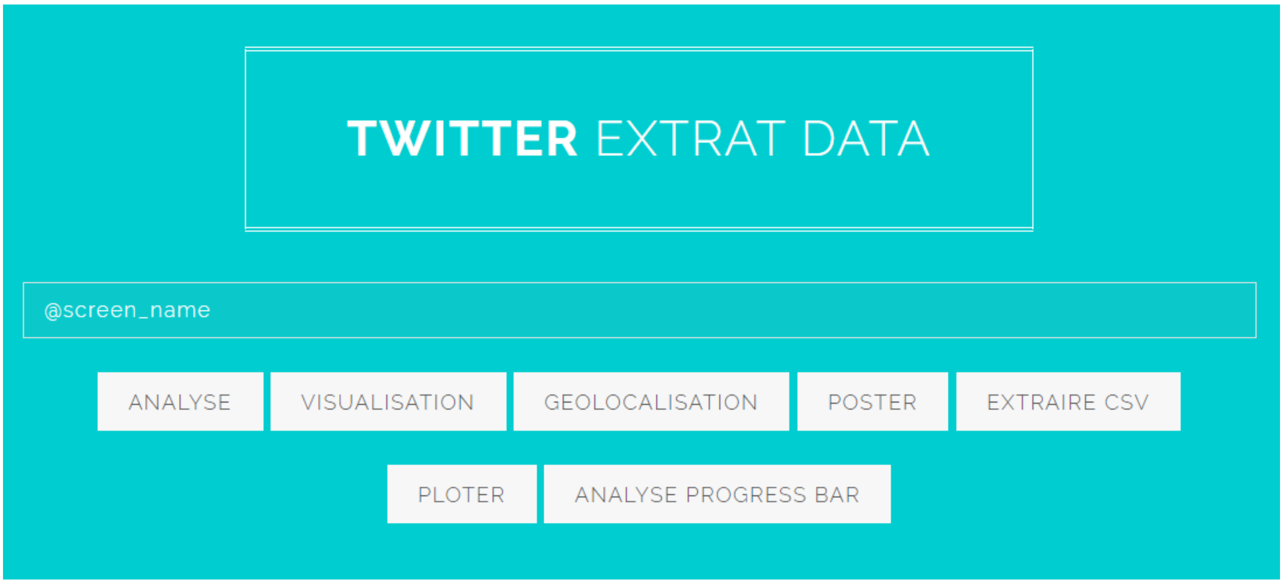
Geo-location Processing



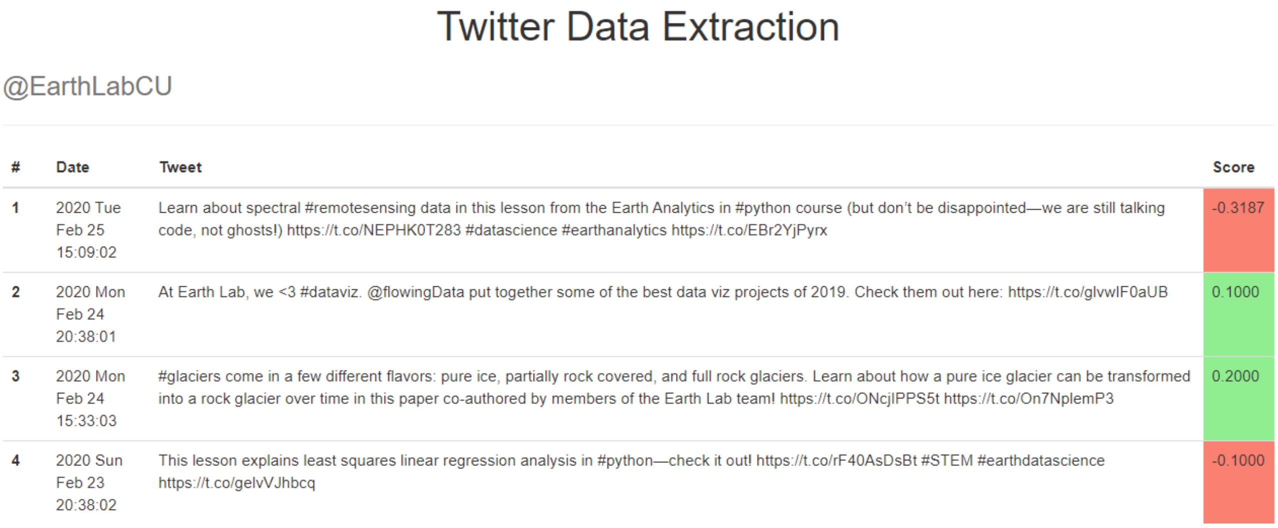
```
# -*- coding: utf-8 -*-
"""
Created on Thu Sep 22 19:26:59 2016

@author: viktoria
"""
import json

path = 'OutputData.json'
with open(path, 'r') as f:
    geo_data = {
        "type": "FeatureCollection",
        "features": []
    }
    for line in f:
        tweet = json.loads(line)
        if tweet['coordinates']:
            geo_json_feature = {
                "type": "Feature",
                "geometry": tweet['coordinates'],
                "properties": {
                    "text": tweet['text'],
                    "created_at": tweet['created_at']
                }
            }
            geo_data['features'].append(geo_json_feature)

# Create GeoJSON
with open('geo_data.json', 'w') as fout:
    fout.write(json.dumps(geo_data, indent=4))
```

Chapitre 3 : Project Interface

➢ Data interface



➢ Data Analysis

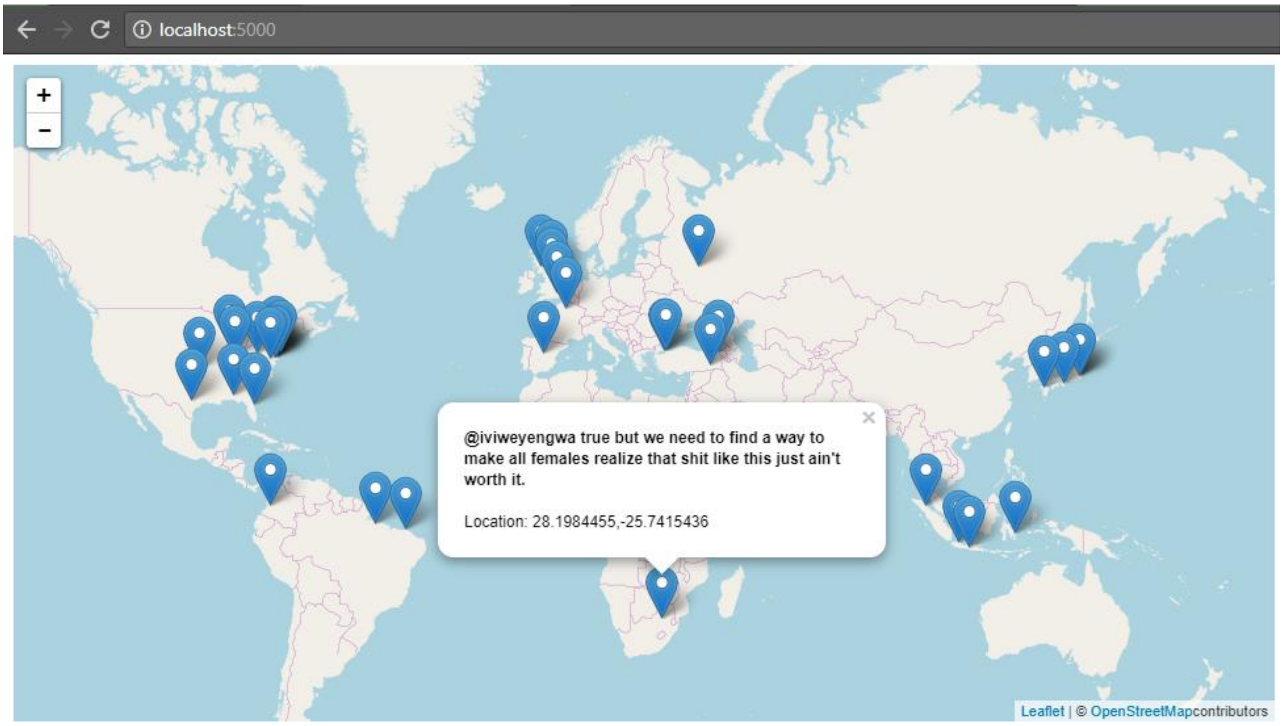


Twitter Data Extraction

@EarthLabCU

#	Date	Tweet	Score
1	2020 Tue Feb 25 15:09:02	Learn about spectral #remotesensing data in this lesson from the Earth Analytics in #python course (but don't be disappointed—we are still talking code, not ghosts!) https://t.co/NEPHK0T283 #datascience #earthanalytics https://t.co/EBr2YjPyrx	-0.3187
2	2020 Mon Feb 24 20:38:01	At Earth Lab, we <3 #dataviz. @flowingData put together some of the best data viz projects of 2019. Check them out here: https://t.co/glvwIF0aUB	0.1000
3	2020 Mon Feb 24 15:33:03	#glaciers come in a few different flavors: pure ice, partially rock covered, and full rock glaciers. Learn about how a pure ice glacier can be transformed into a rock glacier over time in this paper co-authored by members of the Earth Lab team! https://t.co/ONcjIPPS5t https://t.co/On7NplemP3	0.2000
4	2020 Sun Feb 23 20:38:02	This lesson explains least squares linear regression analysis in #python—check it out! https://t.co/rF40AsDsBt #STEM #earthdatascience https://t.co/gelvVJhbcq	-0.1000

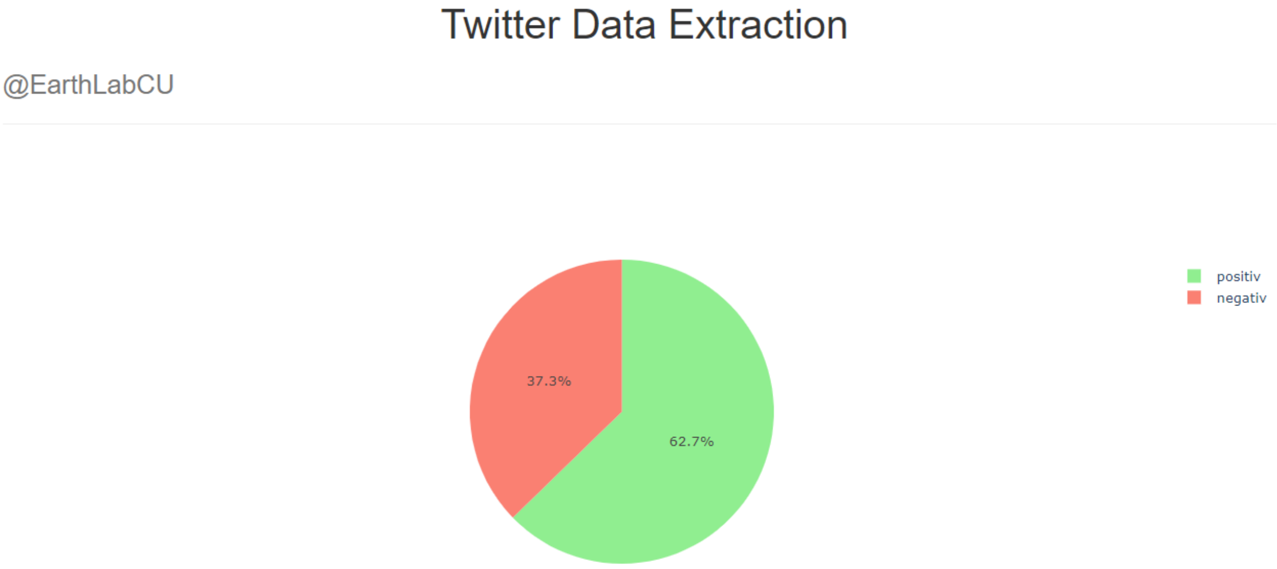
➢ Data Geolocation



@iviweyengwa true but we need to find a way to make all females realize that shit like this just ain't worth it.

Location: 28.1984455,-25.7415436

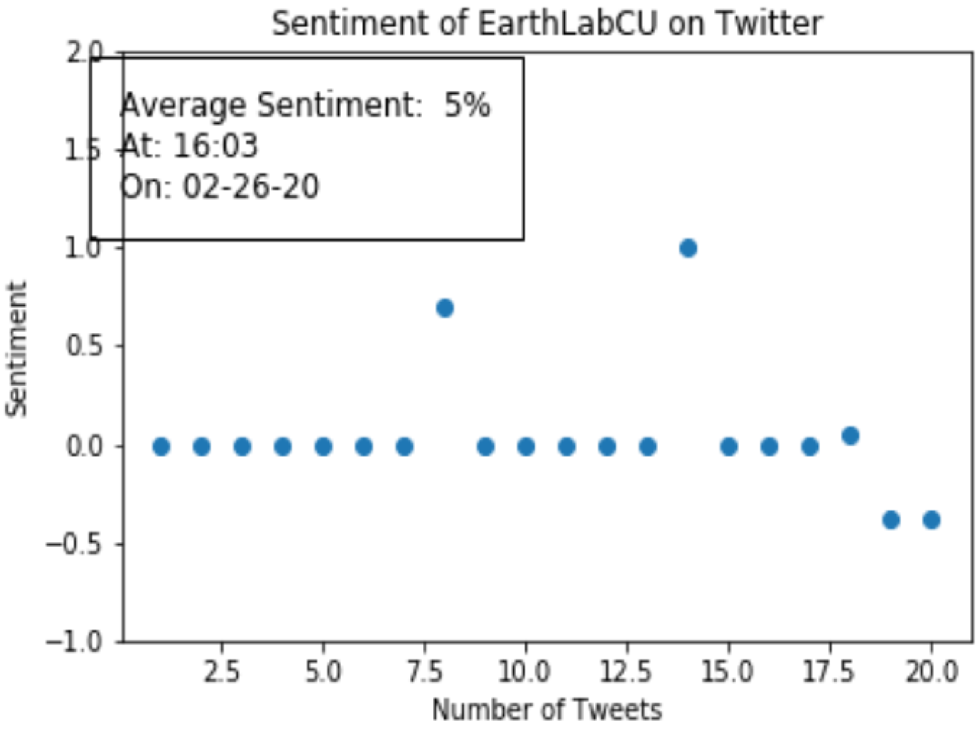
➢ Data Visualization



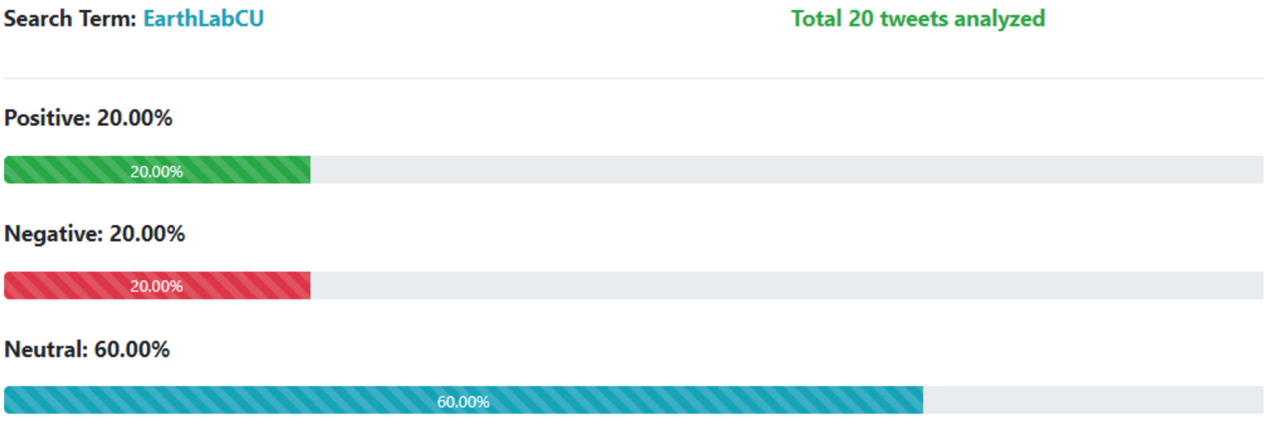
Twitter Data Extraction

@EarthLabCU

➢ Plot Data



- **Plot Data in progressbar form**



Search Term: EarthLabCU

Total 20 tweets analyzed

Positive: 20.00%

20.00%

Negative: 20.00%

20.00%

Neutral: 60.00%

60.00%

work tools

1. Flask :

Flask, is a web framework, or rather, a micro-framework. This “micro” simply means that Flask is not everything. It also means that to do more than what it allows, you will need to install extensions. Fortunately, these are numerous, of quality, and very well integrated, simply that Flask is a set of modules which will facilitate the programming of dynamic websites. In absolute terms, you could manage without a framework! Indeed, it is enough that your application follows the WSGI standard.

2. Spyder :

Spyder is an open source cross-platform integrated development environment (IDE) for scientific programming in the Python language. Spyder integrates with a number of prominent packages in the scientific Python stack, including NumPy, SciPy, Matplotlib, pandas, IPython, SymPy and Cython, as well as other open source software.

Initially created and developed by Pierre Raybaut in 2009, since 2012 Spyder has been maintained and continuously improved by a team of scientific Python developers and the community.

Conclusion

Twitter sentiment analysis comes under the category of text and opinion mining. It focuses on analyzing the sentiments of the tweets and feeding the data to a machine learning model to train it and then check its accuracy, so that we can use this model for future use according to the results. It comprises of steps like data collection, text preprocessing, sentiment detection, sentiment classification, training and testing the model. This research topic has evolved during the last decade with models reaching the efficiency of almost 85%-90%. But it still lacks the dimension of diversity in the data.

Along with this it has a lot of application issues with the slang used and the short forms of words. Many analyzers don’t perform well when the number of classes are increased. Also, it’s still not tested that how accurate the model will be for topics other than the

one in consideration. Hence sentiment analysis has a very bright scope of development in future.