



**COLLECTING TWEETS USING TWITTER
STREAMING API'S**

**SUBJECT: PRINCIPLES OF THE BIG DATA MANAGEMENT
PROJECT PHASE1**

SESHA SAI SRIKAR PAVAN KUMAR CHONGALA	(16273361)
Gowtham Varma Vegnesna	(16286291)
Goutham Gandreddi	(16286424)

INSTRUCTOR

Dr. PRAVEEN RAO, Ph.D.

ABSTRACT

This project collects tweets from Twitter using Twitter API. The collected tweets in JSON format are pushed into HDFS and the hashtags and URLs are extracted using python code. The word count program is then run on the extracted hashtags and URL's document both in Hadoop and spark.

DISCRIPTION OF THE PROBLEM:

Nowadays, data is growing and accumulating faster than ever before. Currently, around 90% of all data generated in our world was generated only in the last two years. Due to this staggering growth rate, big data platforms had to adopt radical solutions to maintain such huge volumes of data.

Data collection from social networking sites can help analyzing easily. As data collected is unstructured. Hadoop, Spark and python are the best big data solutions.

HADOOP DISTRIBUTED FILE SYSTEM (HDFS):

Big data uses Hadoop Distributed File System (HDFS) which can store Petabytes of data. It can store both structured and unstructured data. It uses social media for survey. The proposed system is totally an automatic system. It uses the social media data to conduct a survey and generates the same result as it has done manually. It has the following advantages:

- Data is available in real time and at high frequency.
- Low-cost source of valuable information.
- No wastage of manpower and time.
- It can answer the questions that would have known through surveys in advance.
- Social media offer a distinctive window into economic activity

System Configuration:

- Operating System:

Mac OS - 10.14.3

- Tools used:

Hadoop 3.1.2

Spark 2.4

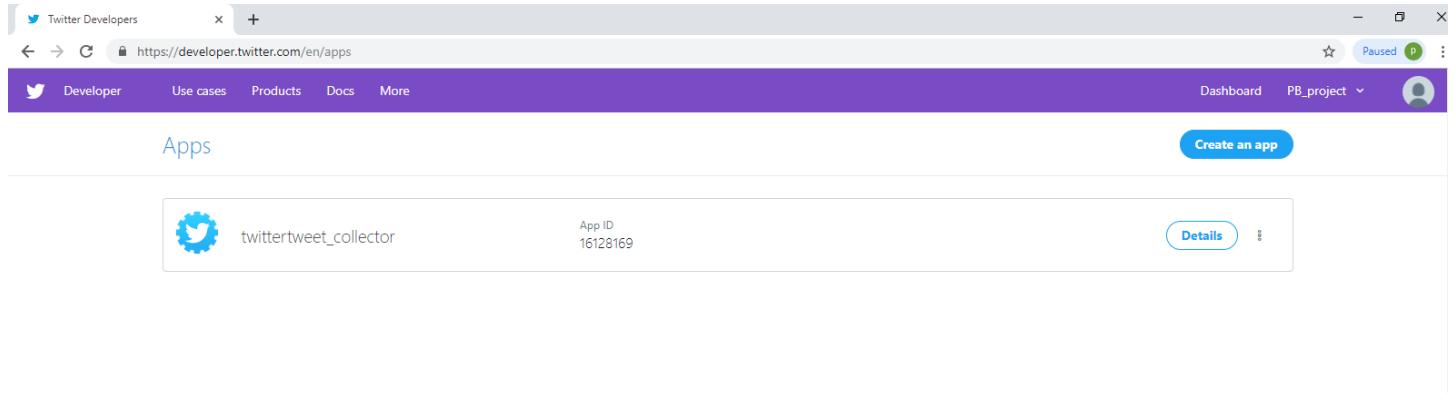
- Language:

Python 3

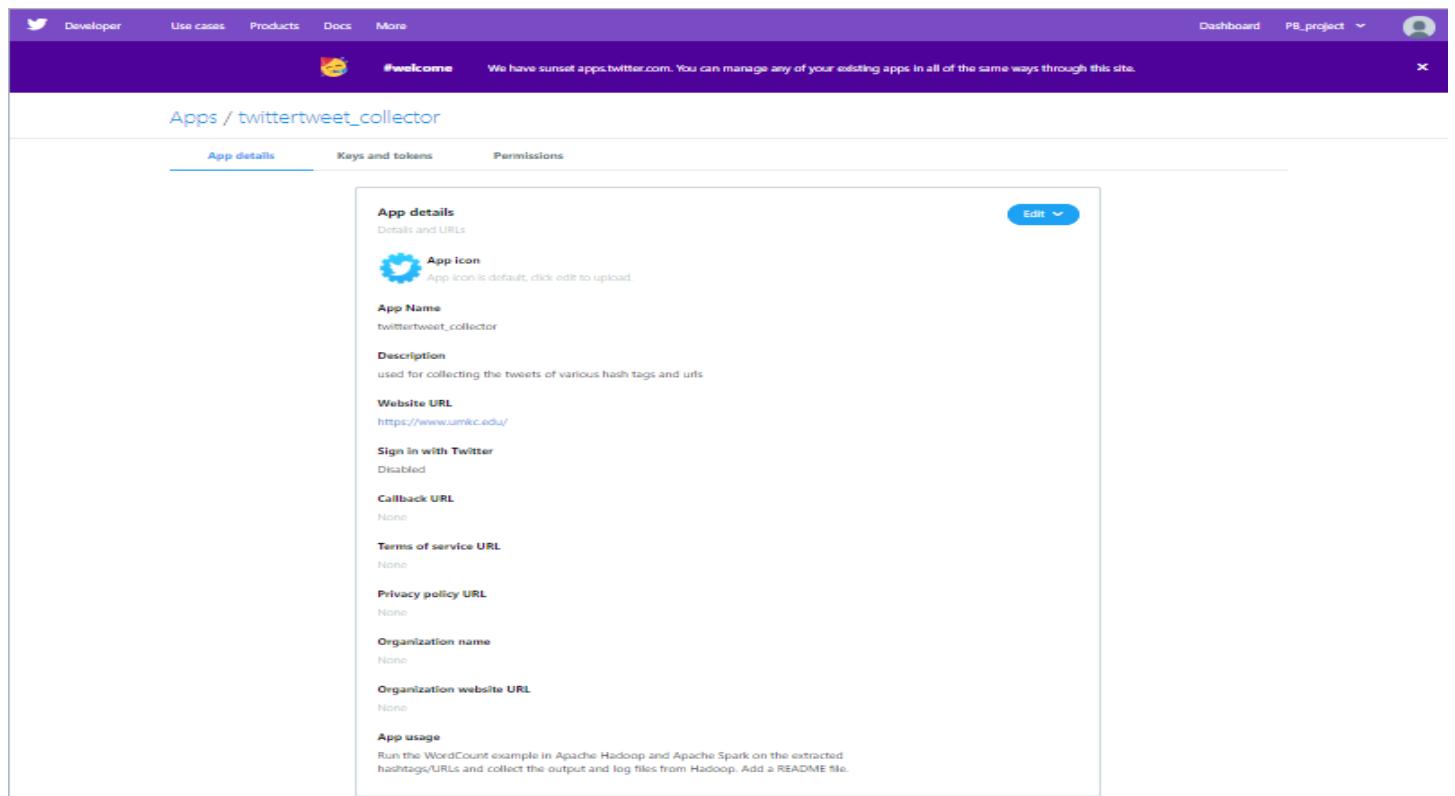
Creating Credentials for Twitter APIs

Before collection of tweets, it is important to sign in with twitter and signup for a development account with the following link: <https://dev.twitter.com/resources/signup>

Create a Twitter Application “twittertweet_collector” with these we get 4 keys – Consumer key, Consumer secret key, Access token and Access token secret which are later used for the collection of tweets



The screenshot shows the Twitter Developers website with the URL https://developer.twitter.com/en/apps. The page displays a list of applications under the heading 'Apps'. One application is listed: 'twittertweet_collector' with App ID 16128169. A 'Details' button is visible next to the app entry.



The screenshot shows the 'App details' tab for the 'twittertweet_collector' application. The page includes fields for App Name (twittertweet_collector), Description (used for collecting the tweets of various hash tags and urls), Website URL (https://www.umkc.edu/), Sign in with Twitter (Disabled), Callback URL (None), Terms of service URL (None), Privacy policy URL (None), Organization name (None), Organization website URL (None), and App usage (Run the WordCount example in Apache Hadoop and Apache Spark on the extracted hashtags/URLs and collect the output and log files from Hadoop. Add a README file.). An 'Edit' button is located at the top right of the form.

Keys and tokens:

Consumer key, Consumer secret, Access token and Access token secret keys which are used for fetching the tweets from the twitter

Apps / twittertweet_collector

The screenshot shows the 'Keys and tokens' tab selected in the Twitter Developers interface. It displays two sets of API keys: Consumer API keys (40zanMEonze4XGwH7h6Tbirpa and de7U5lpto5pyawqULryJnhqltFkMe1qhp26yj4sTx4SqyzbNYR) and Access token & access token secret (1098697660947726338-ryCGw3N7cu3F9AcxVOSalarJv540qu and Klugkdl6Ktr7qaoOlCyaSkwTgZqa7zyPIMJGFUrdrkYm). Buttons for Regenerate and Revoke are present for each set.

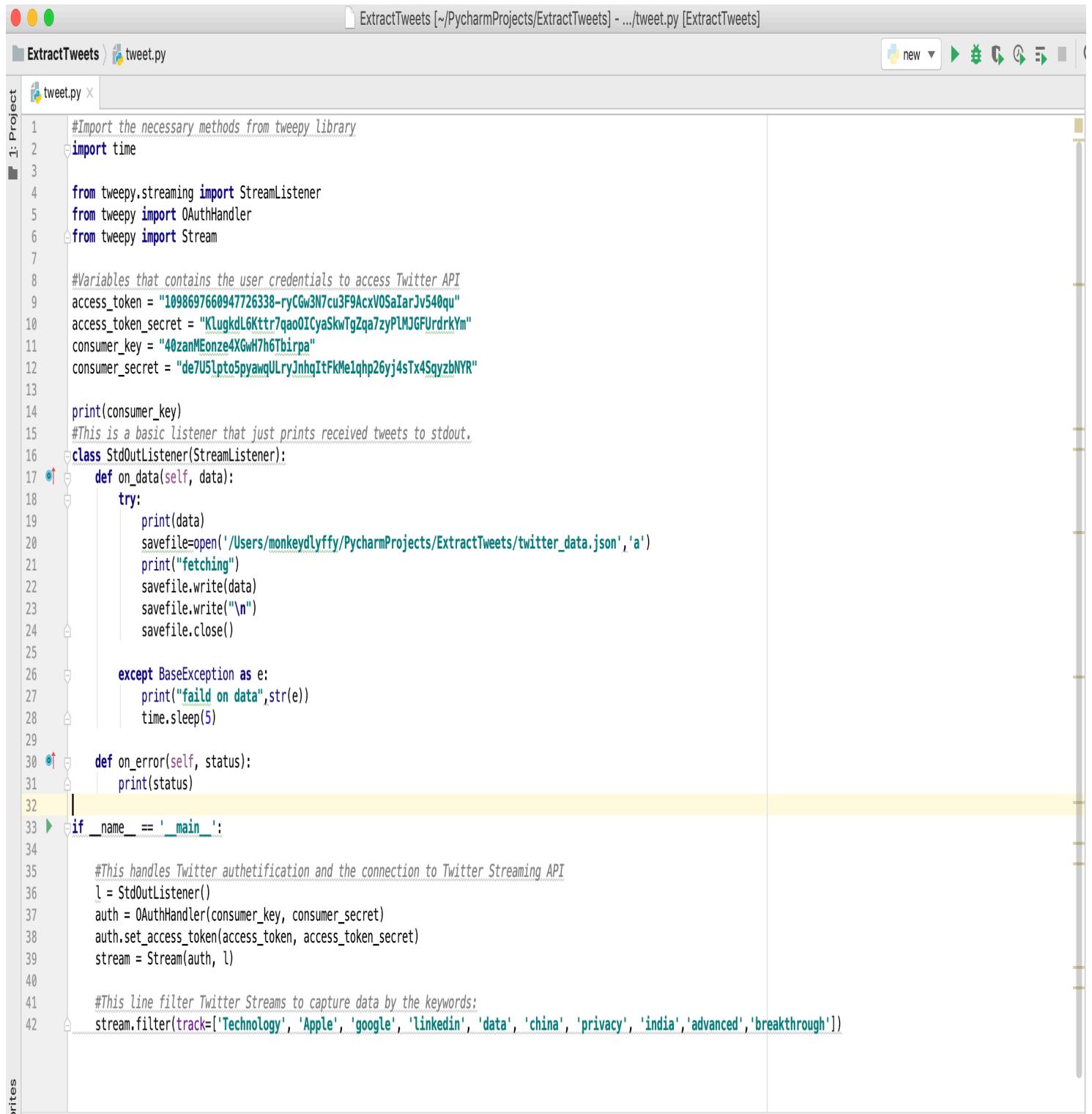
The screenshot shows the Twitter Developers dashboard with a purple header. A message at the top states: '#welcome We have sunset apps.twitter.com. You can manage any of your existing apps in all of the same ways through this site.' Below the header, the 'Dashboard' and 'PB_project' buttons are visible.

The screenshot shows the 'Permissions' tab selected in the Twitter Developers interface. It displays the current permissions for the app, including an 'Access permission' (Read and write) and an 'Additional permissions' section (None). An 'Edit' button is located in the top right corner of the permissions box.

Twitter API:

Twitter streaming API is used for downloading large volume of tweets from the twitter using python 3

Code for collecting tweets



The screenshot shows the PyCharm IDE interface with the following details:

- Title Bar:** ExtractTweets [~/PycharmProjects/ExtractTweets] - .../tweet.py [ExtractTweets]
- Toolbars:** Standard PyCharm toolbars for file operations, search, and navigation.
- Project View:** Shows the "ExtractTweets" project with "tweet.py" selected.
- Code Editor:** Displays the Python script "tweet.py".
- Code Content:** The script uses the tweepy library to interact with the Twitter Streaming API. It defines a class `StdOutListener` that implements the `StreamListener` interface. The `on_data` method handles incoming tweets by saving them to a JSON file named `twitter_data.json`. The `on_error` method handles errors and sleeps for 5 seconds. The script also authenticates with the Twitter API using consumer keys and access tokens. Finally, it filters tweets based on specific keywords like 'Technology', 'Apple', 'google', 'linkedin', 'data', 'china', 'privacy', 'india', 'advanced', and 'breakthrough'.

```
#Import the necessary methods from tweepy library
import time

from tweepy.streaming import StreamListener
from tweepy import OAuthHandler
from tweepy import Stream

#Variables that contains the user credentials to access Twitter API
access_token = "1098697660947726338-ryCGw3N7cu3F9AcxV0SaIarJv540qu"
access_token_secret = "KlugkdL6Kttr7qao0ICyaSkwTgZqa7zyPLMJGFUrdrkYm"
consumer_key = "40zanMEonze4XGwH7h6Tbirpa"
consumer_secret = "de7U5Lpto5pyawqULryJnhqItFkMe1qhp26yj4sTx4SgyzbNYR"

print(consumer_key)
#This is a basic listener that just prints received tweets to stdout.
class StdOutListener(StreamListener):
    def on_data(self, data):
        try:
            print(data)
            savefile=open('/Users/monkeydlyffy/PycharmProjects/ExtractTweets/twitter_data.json','a')
            print("fetching")
            savefile.write(data)
            savefile.write("\n")
            savefile.close()

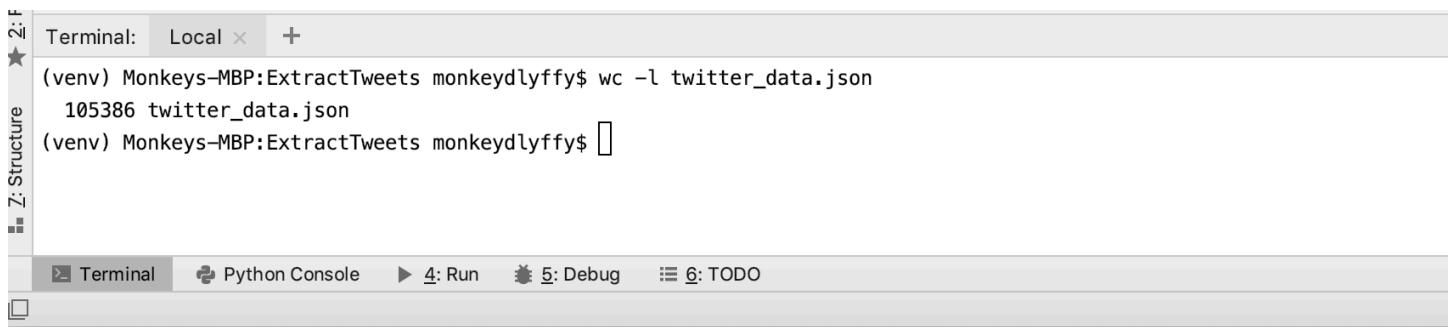
        except BaseException as e:
            print("faild on data",str(e))
            time.sleep(5)

    def on_error(self, status):
        print(status)

if __name__ == '__main__':
    #This handles Twitter authetification and the connection to Twitter Streaming API
    l = StdOutListener()
    auth = OAuthHandler(consumer_key, consumer_secret)
    auth.set_access_token(access_token, access_token_secret)
    stream = Stream(auth, l)

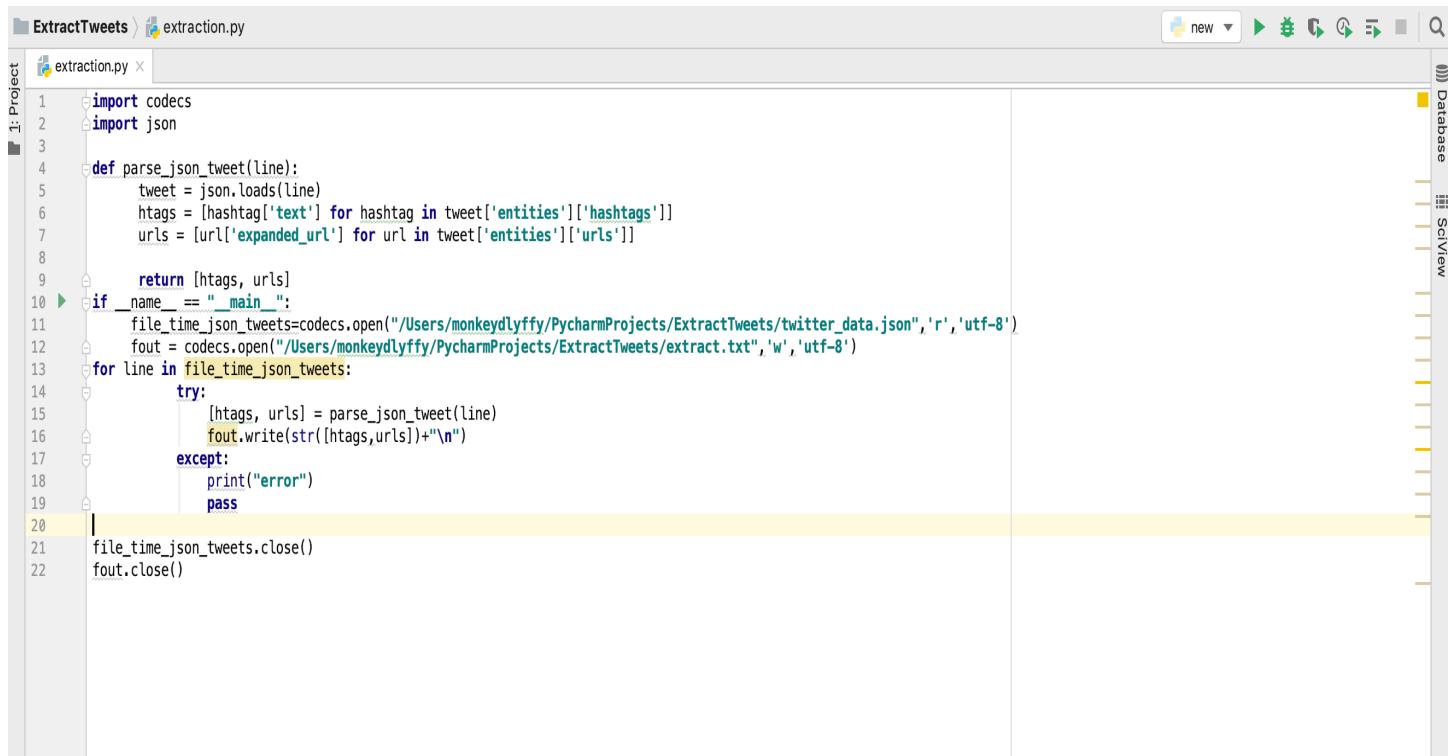
    #This line filter Twitter Streams to capture data by the keywords:
    stream.filter(track=['Technology', 'Apple', 'google', 'linkedin', 'data', 'china', 'privacy', 'india', 'advanced', 'breakthrough'])
```

Number of tweets downloaded to twitter_data.json file



The screenshot shows the PyCharm IDE interface. At the top, there's a terminal window titled 'Terminal: Local' with the command 'wc -l twitter_data.json' run, resulting in the output '105386'. Below the terminal is a navigation bar with tabs for 'Terminal', 'Python Console', 'Run', 'Debug', and 'TODO'.

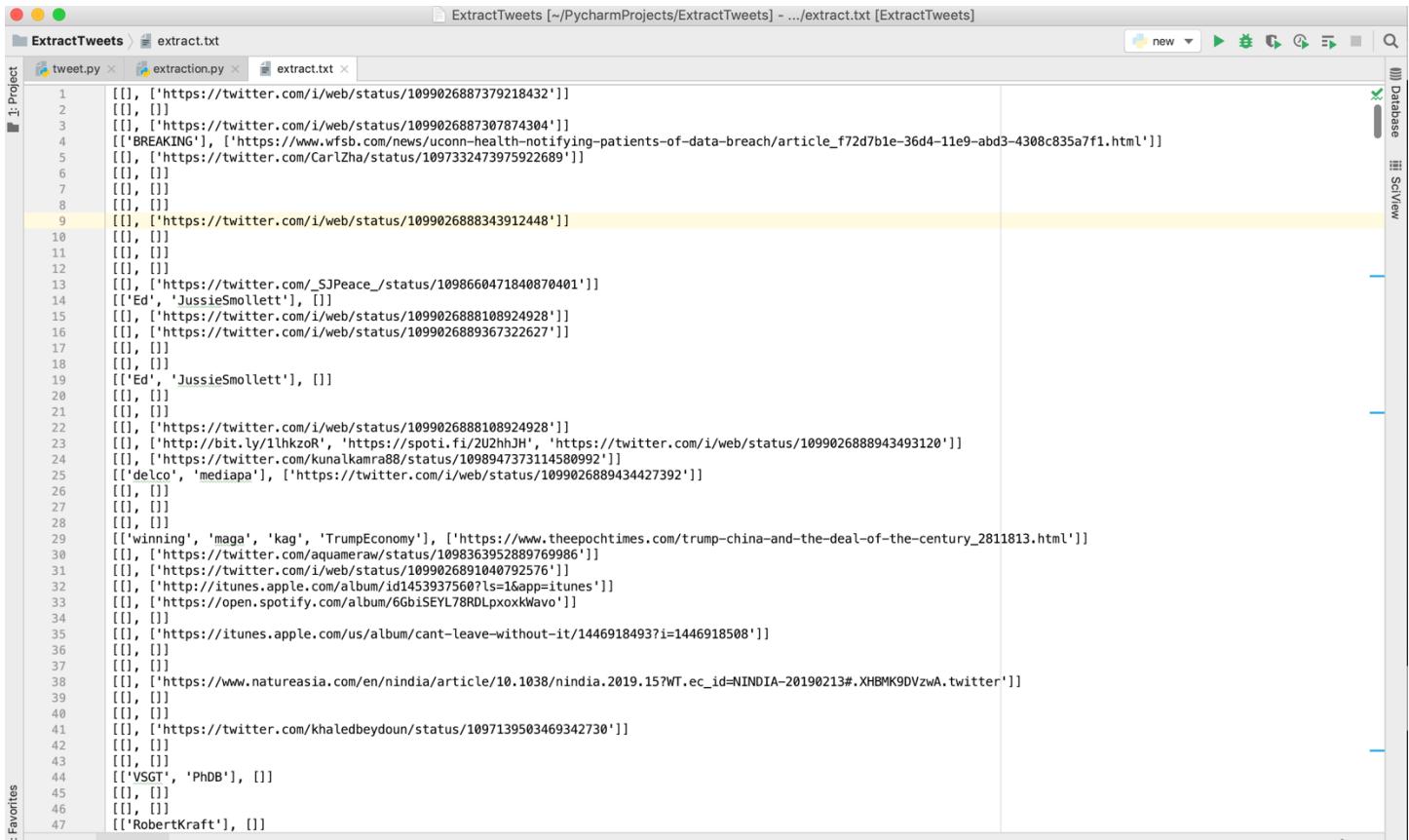
Fetching Code for extracting Hashtags and Url's from Tweets



The screenshot shows the PyCharm IDE interface with a code editor open for 'extraction.py'. The code defines a function 'parse_json_tweet' that takes a JSON line and returns a tuple of hashtags and URLs. It then checks if the script is being run directly ('__name__ == "__main__"'). If so, it reads from 'twitter_data.json', writes to 'extract.txt', and handles exceptions. The code editor has syntax highlighting and a vertical scrollbar.

```
1: import codecs
2: import json
3:
4: def parse_json_tweet(line):
5:     tweet = json.loads(line)
6:     hashtags = [hashtag['text'] for hashtag in tweet['entities']['hashtags']]
7:     urls = [url['expanded_url'] for url in tweet['entities']['urls']]
8:
9:     return [hashtags, urls]
10: if __name__ == "__main__":
11:     file_time_json_tweets=codecs.open("/Users/monkeydlyffy/PycharmProjects/ExtractTweets/twitter_data.json",'r','utf-8')
12:     fout = codecs.open("/Users/monkeydlyffy/PycharmProjects/ExtractTweets/extract.txt",'w','utf-8')
13:     for line in file_time_json_tweets:
14:         try:
15:             [hashtags, urls] = parse_json_tweet(line)
16:             fout.write(str([hashtags, urls])+"\n")
17:         except:
18:             print("error")
19:             pass
20:
21: file_time_json_tweets.close()
22: fout.close()
```

After extracting the hashtags and url's



The screenshot shows the PyCharm IDE interface with the project 'ExtractTweets' open. The 'extract.txt' file is the active editor tab, displaying a list of tweet data. The code consists of a series of numbered lines (1 through 47) each containing a list of URLs. Lines 9 and 22 are highlighted in yellow, indicating specific tweets of interest.

```
1  [ [], ['https://twitter.com/i/web/status/1099026887379218432']]  
2  [ [], []]  
3  [ [], ['https://twitter.com/i/web/status/1099026887307874304']]  
4  [ ['BREAKING'], ['https://www.wfsb.com/news/uconn-health-notifying-patients-of-data-breach/article_f72d7b1e-36d4-11e9-abd3-4308c835a7f1.html']]  
5  [ [], ['https://twitter.com/CarlZha/status/1097332473975922689']]  
6  [ [], []]  
7  [ [], []]  
8  [ [], []]  
9  [ [], ['https://twitter.com/i/web/status/1099026888343912448']]  
10 [ [], []]  
11 [ [], []]  
12 [ [], []]  
13 [ [], ['https://twitter.com/_SJPeace_/status/1098660471840870401']]  
14 [ ['Ed', 'JussieSmollett'], []]  
15 [ [], ['https://twitter.com/i/web/status/1099026888108924928']]  
16 [ [], ['https://twitter.com/i/web/status/1099026889367322627']]  
17 [ [], []]  
18 [ [], []]  
19 [ ['Ed', 'JussieSmollett'], []]  
20 [ [], []]  
21 [ [], []]  
22 [ [], ['https://twitter.com/i/web/status/1099026888108924928']]  
23 [ [], ['http://bit.ly/1hkz0R', 'https://spoti.fi/2U2hhJH', 'https://twitter.com/i/web/status/109902688943493120']]  
24 [ [], ['https://twitter.com/kunalkamra88/status/1098947373114580992']]  
25 [ ['delco', 'mediapa'], ['https://twitter.com/i/web/status/1099026889434427392']]  
26 [ [], []]  
27 [ [], []]  
28 [ [], []]  
29 [ ['winning', 'maga', 'kag', 'TrumpEconomy'], ['https://www.theepochtimes.com/trump-china-and-the-deal-of-the-century_2811813.html']]  
30 [ [], ['https://twitter.com/aquameraw/status/1098363952889769986']]  
31 [ [], ['https://twitter.com/i/web/status/1099026891040792576']]  
32 [ [], ['http://itunes.apple.com/album/id1453937560?ls=1&app=itunes']]  
33 [ [], ['https://open.spotify.com/album/6Gb1SEYL78RDlxoxkkWavo']]  
34 [ [], []]  
35 [ [], ['https://itunes.apple.com/us/album/cant-leave-without-it/1446918493?i=1446918508']]  
36 [ [], []]  
37 [ [], []]  
38 [ [], ['https://www.natureasia.com/en/nindia/article/10.1038/nindia.2019.15?WT.ec_id=NINDIA-20190213#.XHBMK9DVzwA.twitter']]  
39 [ [], []]  
40 [ [], []]  
41 [ [], ['https://twitter.com/khaledbeydoun/status/1097139503469342730']]  
42 [ [], []]  
43 [ [], []]  
44 [ ['VSGT', 'PhDB'], []]  
45 [ [], []]  
46 [ [], []]  
47 [ ['RobertKraft'], []]
```

Runing The Word Count for the gathered tweets –

Code for word count –

```
1 import java.io.IOException;
2 import java.util.StringTokenizer;
3 import org.apache.hadoop.conf.Configuration;
4 import org.apache.hadoop.fs.Path;
5 import org.apache.hadoop.io.IntWritable;
6 import org.apache.hadoop.io.Text;
7 import org.apache.hadoop.mapreduce.Job;
8 import org.apache.hadoop.mapreduce.Mapper;
9 import org.apache.hadoop.mapreduce.Reducer;
10 import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
11 import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
12
13 public class WordCount {
14
15     public static class TokenizerMapper
16         extends Mapper<Object, Text, Text, IntWritable>{
17             private final static IntWritable one = new IntWritable(1);
18             private Text word = new Text();
19             public void map(Object key, Text value, Context context
20                 ) throws IOException, InterruptedException {
21                 StringTokenizer itr = new StringTokenizer(value.toString());
22                 while (itr.hasMoreTokens()) {
23                     word.set(itr.nextToken());
24                     context.write(word, one);
25                 }
26             }
27         }
28     public static class IntSumReducer
29         extends Reducer<Text,IntWritable,Text,IntWritable> {
30         private IntWritable result = new IntWritable();
31         public void reduce(Text key, Iterable<IntWritable> values,Context context) throws IOException, InterruptedException {
32             int sum = 0;
33             for (IntWritable val : values) {
34                 sum += val.get();
35             }
36             result.set(sum);
37             context.write(key, result);
38         }
39     }
40
41     public static void main(String[] args) throws Exception {
42         Configuration conf = new Configuration();
43         Job job = Job.getInstance(conf, "word count");
44         job.setJarByClass(WordCount.class);
45         job.setMapperClass(TokenizerMapper.class);
46         job.setCombinerClass(IntSumReducer.class);
47         job.setReducerClass(IntSumReducer.class);
48         job.setOutputKeyClass(Text.class);
49         job.setOutputValueClass(IntWritable.class);
50         FileInputFormat.addInputPath(job, new Path(args[0]));
51         FileOutputFormat.setOutputPath(job, new Path(args[1]));
52         System.exit(job.waitForCompletion(true) ? 0 : 1);
53     }
54 }
```

Pushing the extracted file & running the wordcount program

```
Monkeys-MBP:hadoop-2.8.1 monkeydlyffy$ bin/hdfs dfs -copyFromLocal /Users/monkeydlyffy/PycharmProjects/ExtractTweets/extract.txt /user/hadoop/
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by org.apache.hadoop.security.authentication.util.KerberosUtil (file:/Users/monkeydlyffy/hadoop-2.8.1/share/hadoop/common/lib/hadoop-auth-2.8.1.jar) to method sun.security.krb5.Config.getInstance()
WARNING: Please consider reporting this to the maintainers of org.apache.hadoop.security.authentication.util.KerberosUtil
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
19/02/22 19:57:29 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Monkeys-MBP:hadoop-2.8.1 monkeydlyffy$ 
Monkeys-MBP:hadoop-2.8.1 monkeydlyffy$ 
Monkeys-MBP:hadoop-2.8.1 monkeydlyffy$ 
Monkeys-MBP:hadoop-2.8.1 monkeydlyffy$ 
Monkeys-MBP:hadoop-2.8.1 monkeydlyffy$ 
Monkeys-MBP:hadoop-2.8.1 monkeydlyffy$ bin/hadoop jar /Users/monkeydlyffy/hadoop-2.8.1/wordcount.jar WordCount /user/hadoop/extract.txt /user/hadoop/output
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by org.apache.hadoop.security.authentication.util.KerberosUtil (file:/Users/monkeydlyffy/hadoop-2.8.1/share/hadoop/common/lib/hadoop-auth-2.8.1.jar) to method sun.security.krb5.Config.getInstance()
WARNING: Please consider reporting this to the maintainers of org.apache.hadoop.security.authentication.util.KerberosUtil
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
19/02/22 19:59:49 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
19/02/22 19:59:49 INFO client.RMProxy: Connecting to ResourceManager at localhost/127.0.0.1:8032
19/02/22 19:59:50 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with ToolRunner to remedy this.
19/02/22 19:59:50 INFO input.FileInputFormat: Total input files to process : 1
19/02/22 19:59:50 INFO mapreduce.JobSubmitter: number of splits:1
19/02/22 19:59:50 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1550886705704_0001
19/02/22 19:59:50 INFO impl.YarnClientImpl: Submitted application application_1550886705704_0001
19/02/22 19:59:50 INFO mapreduce.Job: The url to track the job: http://monkeys-mbp:8088/proxy/application_1550886705704_0001/
19/02/22 19:59:50 INFO mapreduce.Job: Running job: job_1550886705704_0001
19/02/22 19:59:56 INFO mapreduce.Job: Job job_1550886705704_0001 running in uber mode : false
19/02/22 19:59:56 INFO mapreduce.Job: map 0% reduce 0%
19/02/22 20:00:00 INFO mapreduce.Job: map 100% reduce 0%
19/02/22 20:00:05 INFO mapreduce.Job: map 100% reduce 100%
```

Copied the output folder to local and viewing the contents of part-r-0000 file:

```
● ● ● hadoop-2.8.1 — bash — 134x39
Monkeys-MBP:hadoop-2.8.1 monkeydlyffy$ ls -lrt
total 755312
drwxr-xr-x 30 monkeydlyffy staff      960 Jun  2 2017 sbin
drwxr-xr-x 12 monkeydlyffy staff      384 Jun  2 2017 libexec
drwxr-xr-x  3 monkeydlyffy staff       96 Jun  2 2017 lib
drwxr-xr-x  7 monkeydlyffy staff     224 Jun  2 2017 include
-rw-r--r--  1 monkeydlyffy staff    1366 Jun  2 2017 README.txt
-rw-r--r--  1 monkeydlyffy staff   15915 Jun  2 2017 NOTICE.txt
-rw-r--r--  1 monkeydlyffy staff   99253 Jun  2 2017 LICENSE.txt
drwxr-xr-x  5 monkeydlyffy staff    160 Feb 21 10:56 etc
drwxr-xr-x  5 monkeydlyffy staff    160 Feb 21 11:02 share
-rwxrwxrwx@ 1 monkeydlyffy staff   2286 Feb 21 11:10 test.txt
-rw-r--r--  1 monkeydlyffy staff  115358 Feb 22 17:39 extract.txt
drwxr-xr-x 15 monkeydlyffy staff    480 Feb 22 18:24 bin
-rw-r--r--  1 monkeydlyffy staff   4924 Feb 22 18:55 wordcount.jar
-rw-r--r--@ 1 monkeydlyffy staff 375119526 Feb 22 19:48 twitter_data.json
drwxr-xr-x  5 monkeydlyffy staff    160 Feb 22 19:51 name
drwx-----  4 monkeydlyffy staff   128 Feb 22 19:51 data
drwxr-xr-x 93 monkeydlyffy staff   2976 Feb 22 19:51 logs
drwxr-xr-x  4 monkeydlyffy staff   128 Feb 22 20:06 output
Monkeys-MBP:hadoop-2.8.1 monkeydlyffy$ Monkeys-MBP:hadoop-2.8.1 monkeydlyffy$ cat output/part-r-00000
"ht...://www.google.com/bookmarks/mark?op=edit&output=popup&bkmk=http://www.v2porno.com/une-eplucheuse-de-lentilles-au-buste-imposant-s-e-divertit-dans-un-broute-minette-avec-une-partenaire-complice.html&title=V2PORNO.com%20-%20une%20stagiaire%20qui%20s'est%20laissee%20seduire%20par%20sa%20directrice%20prend%20drolement%20gout%20a%20ces%20lechotteries%20feminines"]]" 1
'100pcERen2050'], 1
'11minutes', 1
'140years', 1
'1yrago', 1
'2019Elections', 1
'20EmpresasporMariano'], 1
'20РоківАвіаціїДСНС', 3
'25Degrees', 1
'2ndAmendment', 1
'30for30', 2
'33822回目', 1
'3D', 5
```

word count output:

```
hadoop-2.8.1 — bash — 134x46
[['求人', 1
[['猫の日'], 1
[['生活保護'], 1
[['福田飛銳'], 1
[['稼ぐ'], 1
[['自衛隊'], 1
[['藤崎八幡宮秋季例大祭'], 1
[['転職したい'], 1
[['転職活動アプリ'], 1
[['速報'], 1
[['過去記事'], 1
[['遺失物法'], 1
[['邓伦'], 1
[['配信'], 1
[['音楽'], 12
[['高収入'], 1
[['高収益'], 1
[['髪型自由'], 1
[['魔道祖师'], 1
[['갓 세븐'], 1
[['김태리'], 1
[['닉쿤'], 2
[['단추DATA'], 1
[['데 이식스'], 1
[['도경수'], 13
[['딴지'], 1
[['마마무'], 1
[['메이저놀이터총판'], 1
[['몬스타엑스'], 1
[['박지훈'], 1
[['백현'], 7
[['버스터즈'], 1
[['베리베리'], 1
[['샌드위치'], 1
[['아이유'], 4
[['에이프릴'], 1
[['원고키링'], 1
[['자동'], 1
[['자취남녀'], 1
[['재배소년'], 1
[['찬열'], 3
[['황민현'], 1
[['1つでも好きなのがあったらフォロミー']], 1
[], 43222
[], 31388
Monkeys-MBP:hadoop-2.8.1 monkeydlyffy$
```

Screenshots from HDFS:

Before running the wordcount:

The screenshot shows the HDFS Web UI interface. At the top, there's a navigation bar with tabs for Hadoop, Overview, Datanodes, Datanode Volume Failures, Snapshot, Startup Progress, and Utilities. Below the navigation bar, the main area is titled "Browse Directory" and shows the path "/user/hadoop". A search bar at the top right has "Search:" and a "Go!" button. Below the search bar, there's a "Show 25 entries" dropdown and a "Search:" input field. The main content area is a table listing directory entries:

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
-rw-r--r--	monkeydlyffy	supergroup	2.04 MB	Feb 22 19:57	1	128 MB	extract.txt
drwxr-xr-x	monkeydlyffy	supergroup	0 B	Feb 22 20:00	0	0 B	output

At the bottom of the table, it says "Showing 1 to 2 of 2 entries". There are "Previous" and "Next" buttons at the bottom right.

After the wordcount - new directory output creation & its contents:

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
-rw-r--r--	monkeydlyffy	supergroup	0 B	Feb 22 20:00	1	128 MB	_SUCCESS
-rw-r--r--	monkeydlyffy	supergroup	1.1 MB	Feb 22 20:00	1	128 MB	part-r-00000

Code for running the word count in spark

```
monkeydlyffy — java - spark-shell — 170x19
scala> val inputfile=sc.textFile("hdfs://localhost:9000/user/hadoop/extract.txt")
inputfile: org.apache.spark.rdd.RDD[String] = hdfs://localhost:9000/user/hadoop/extract.txt MapPartitionsRDD[1] at textFile at <console>:24
scala> val counts = inputfile.flatMap(line => line.split(" ")).map(word => (word,1)).reduceByKey(_ + _)
counts: org.apache.spark.rdd.RDD[(String, Int)] = ShuffledRDD[4] at reduceByKey at <console>:25
scala> counts.take(5).foreach(println)
(['https://twitter.com/i/web/status/1099031836586471426'],1)
(['https://twitter.com/i/web/status/1099032456882200576'],1)
(['ELITE8'],1)
(['https://twitter.com/i/web/status/109903465287077889'],1)
(['https://zinchenkosergey.livejournal.com/7457609.html'],4)
scala> counts.collect().foreach(println)
(['https://twitter.com/i/web/status/1099031836586471426'],1)
(['https://twitter.com/i/web/status/1099032456882200576'],1)
(['ELITE8'],1)
(['https://twitter.com/i/web/status/109903465287077889'],1)
(['https://zinchenkosergey.livejournal.com/7457609.html'],4)
```

New output directory for spark wordcount:

```
monkeydlyffy — java - spark-shell — 161x5
... 49 elided
scala> counts.saveAsTextFile("hdfs://localhost:9000/user/hadoop/sparkwordcount")
scala>
```

Screenshots from HDFS for spark:

localhost:50070/explorer.html#/user/hadoop/sparkwordcount

Hadoop Overview Datanodes Datanode Volume Failures Snapshot Startup Progress Utilities

Browse Directory

/user/hadoop/sparkwordcount

Show 25 entries Search:

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
-rw-r--r--	monkeydlyffy	supergroup	0 B	Feb 22 21:41	3	128 MB	_SUCCESS
-rw-r--r--	monkeydlyffy	supergroup	576.52 KB	Feb 22 21:41	3	128 MB	part-00000
-rw-r--r--	monkeydlyffy	supergroup	593.31 KB	Feb 22 21:41	3	128 MB	part-00001

Showing 1 to 3 of 3 entries

Previous 1 Next

Hadoop, 2017.

localhost:4040/jobs/job/?id=3

Apache Spark 2.4.0 Jobs Stages Storage Environment Executors Spark shell application UI

Details for Job 3

Status: SUCCEEDED

Completed Stages: 1

Skipped Stages: 1

[Event Timeline](#)

[DAG Visualization](#)

Completed Stages (1)

Stage Id	Description	Submitted	Duration	Tasks: Succeeded/Total	Input	Output	Shuffle Read	Shuffle Write
7	runJob at SparkHadoopWriter.scala:78	+details 2019/02/22 21:41:12	0.6 s	2/2	1169.8 KB	656.1 KB		

Skipped Stages (1)

Stage Id	Description	Submitted	Duration	Tasks: Succeeded/Total	Input	Output	Shuffle Read	Shuffle Write
6	map at <console>:25	+details Unknown	Unknown	0/2				

The log files and the readme file can be found in the following location:

<https://drive.google.com/drive/u/0/folders/149xQxhQwCcDJThNJnwyjc85qtM8AKj02>

References:

- 1) <https://www.alexkras.com/how-to-get-user-feed-with-twitter-api-and-python/>
- 2) <http://socialmedia-class.org/twittertutorial.html>
- 3) <https://spark.apache.org/examples.html>