

```
#####
# INSTITUTE TECHNOLOGY OF BANDUNG #
# School of Architecture, Planning, and Policy Development #
# Research on HBW Trip Production Modeling by #
# Doctoral Student: Rempu Sora Rayat #
# Script Name: Retrieving Twitter Historical/Archive Data #
# Purpose: Find Username, Timestamp, Location, Tweet and Save #
# Programmer: Richard Shiawase #
#####

import json
import time
from Sheet import Sheet
from Splitter import Splitter
from Row import Row
from Column import Column
from Profile import Profile
import requests
BEARER_TOKEN =
"AAAAAAAAAAAAAAAAAAC6W0wEAAAAA9yOzkGKPPE2fVkhWRFFveP7fPY%3DcbTLvaQFPlos
bZw3oa2y6JqioILgzHV9QiPz131bMxjq9IOkZ9"

listLong = []
listTweetCreated = []
listTweet = []
listLat = []

splitter = Splitter()
row = Row()
column = Column()
sheet = Sheet(row, column, splitter)

#define search twitter function
def search_twitter(query, tweet_fields, next_token, max_result, bearer_token =
BEARER_TOKEN):

    headers = {"Authorization": "Bearer {}".format(bearer_token)}
    url = "https://api.twitter.com/2/tweets/search/all?start_time=2018-01-
01T00%3A00%3A00Z&end_time=2021-04-
30T11%3A59%3A59Z&query={}&{}&{}".format(query, tweet_fields, next_token, ma
x_result)

    response = requests.request("GET", url, headers=headers)
    print(str(response.status_code)+" "+str(next_token))

    if response.status_code != 200:
        raise Exception(response.status_code)
    return response.json()

def search_twitter_init(query, tweet_fields, max_result, bearer_token =
BEARER_TOKEN):
    headers = {"Authorization": "Bearer {}".format(bearer_token)}

    url = "https://api.twitter.com/2/tweets/search/all?start_time=2018-01-
01T00%3A00%3A00Z&end_time=2021-04-
30T11%3A59%3A59Z&query={}&{}&{}".format(query, tweet_fields, max_result)
    response = requests.request("GET", url, headers=headers)

    print("init"+str(response.status_code))

    if response.status_code != 200:
```

```

        raise Exception(response.status_code)
    return response.json()

def getUsernameListFromSheet():
    sheet.getUsernameFromSheet()

def getListProfileFromSheet():
    return sheet.getListProfile()

def update_sheet(profile):
    sheet.update_sheet(profile)

def username_excel_name(name):
    sheet.read_file_name(name)

def output_name(name):
    sheet.xlsx_output_file_name(name)

def set_list_tweet_to_profile(profile, listTweet):
    sheet.setListTweet(profile, listTweet)

def set_list_latitude_to_profile(profile, listLat):
    sheet.setListLat(profile, listLat)

def set_list_longitude_to_profile(profile, listLongitude):
    sheet.setListLongitude(profile, listLongitude)

def set_list_created_at_to_profile(profile, listCreated):
    sheet.setListCreatedAt(profile, listTweetCreated)

def init_search():
    # twitter api call return value is dictionary
    json_response = search_twitter_init(query=query,
    tweet_fields=tweet_fields, max_result=max_result, bearer_token=BEARER_TOKEN)
    # pretty printing
    return json_response

def loop_and_store_init(json_response):
    listTweet.clear()
    listTweetCreated.clear()
    for json_response in json_response["data"]:
        tweet = json_response["text"]
        get_geo(json_response)
        created_at = json_response["created_at"]
        listTweetCreated.append(created_at)
        listTweet.append(tweet)

def loop_and_store(json_response):
    for json_response in json_response["data"]:
        print("loop and store")
        tweet = json_response["text"]
        get_geo(json_response)
        created_at = json_response["created_at"]
        listTweetCreated.append(created_at)
        listTweet.append(tweet)

def get_geo(json_response):
    # check for next_token key if exist
    try:
        coordinates = json_response["geo"]
        latitude = coordinates["coordinates"]["coordinates"][1]

```

```

        longitude = coordinates["coordinates"]["coordinates"][0]
        listLong.append(longitude)
        listLat.append(latitude)

    # return geo

except KeyError:
    listLat.append("-")
    listLong.append("-")

def get_next_token_from_init(json_response):
    # check for next_token key if exist
    try:
        token = json_response["meta"]["next_token"]
        return token

    except KeyError:
        return None

# username file to read
username_excel_name("*****.xlsx")

# set ouput file name
output_name("#####.xlsx")

# get username from sheet
getUsernameListFromSheet()

# get list profile
listProfile = getListProfileFromSheet()

# create excel and sheet
sheet.create_sheet(listProfile)

# twitter fields to be returned by api call
tweet_fields = "tweet.fields=text,author_id,created_at,geo"

# set max_result
max_result = "max_results=500"

x = 0
for Profile in listProfile:
    token_counter = 0
    no_data = True
    # set query name according to sheet name
    query = Profile.username

    try:
        # get json response
        json_response = init_search()

        # loop and store data to array
        no_data = False
        loop_and_store_init(json_response)

        # set tweets list to profile attribute
        set_list_tweet_to_profile(Profile, listTweet)

        # set tweets list to profile attribute
        set_list_latitude_to_profile(Profile, listLat)

```

```

# set tweet longitude to profile attribute
set_list_longitude_to_profile(Profile, listLong)

# set tweet created at list to profile attribute
set_list_created_at_to_profile(Profile, listTweetCreated)

# sleep to prevent blocked
time.sleep(1)

# get init token
token = get_next_token_from_init(json_response)

if token is not None:
    next_token = "next_token=" + token
    while token is not None:
        print("counter " + str(token_counter))
        next_token = "next_token=" + token
        meta = json_response["meta"]
        # time.sleep(1)
        try:
            json_response = search_twitter(query=query,
tweet_fields=tweet_fields, next_token=next_token,
max_result=max_result,
bearer_token=BEARER_TOKEN) # pretty printing
            print(json_response)

            # loop and store data to array
            loop_and_store(json_response)

            # set tweets list to profile attribute
            set_list_tweet_to_profile(Profile, listTweet)

            # set tweets list to profile attribute
            set_list_latitude_to_profile(Profile, listLat)

            # set tweet longitude to profile attribute
            set_list_longitude_to_profile(Profile, listLong)

            # set tweet created at list to profile attribute
            set_list_created_at_to_profile(Profile,
listTweetCreated)

            # check for next_token key if exist
            try:
                token = json_response["meta"]["next_token"]
                # print(meta.keys())
            except KeyError:
                print("ga ada")
                break

        except:
            print("Connection error")
            break
        token_counter += 1
        # if token_counter == 200:
        #     break
except Exception as e:
    print("No data"+str(e))
    if str(e) == "429":
        time.sleep(180)

```

```
no_data = True
time.sleep(1)
continue

if no_data is not True:
    update_sheet(Profile)

x+=1
print("Changing username...")
```