***The most effective method to Send Tweets with Google Scripts and Twitter API***

Needed to send tweets utilizing google content and twitter API yet don't have the foggiest idea how to?? This model shows how you can arrangement your own Twitter Client utilizing Google Apps Script and distribute tweets automatically from inside a Google Document, a spreadsheet or even a web application.

When working with the Twitter API, there are various reasons you might need to move a reaction payload to Google Sheets. Google sheets are incredible for fundamental information examination and sharing outcomes rapidly. This blog entry strolls you through how to utilize Python to cause a solicitation to the Twitter API v2 and send the reaction to a Google Sheets utilizing app script.

***WHAT IS TWITTER API???***

An API, short for Application Programming Interface, is fundamentally a rundown of strategies in which two programming applications can speak with each other. The Twitter API permits you to get to information and send information, for example, messages and media without opening the Twitter application.

**The key techniques are:**

GET: Method to recover some information.

POST: Method to make some information.

PUT: Method to update some information.

DELETE: Method to eliminate some information.

***Setting up***

Before you can utilize the Twitter API v2, you need an affirmed developer account. When you have an endorsed developer account, you need to make a Project, and an App within that Project. You additionally need to have to associate with the Google Drive API.

***PACKAGES***

To get set up, install pandas, gspread, demands, and oauth2client. You'll utilize pandas to change the JSON payload reaction from the Twitter API to an even organization, gspread to associate with Google Sheets, solicitations to interface with make a HTTP solicitation to the Twitter API, and oauth2client to assist with confirmation utilizing administration accounts.

import pandas as pd

import gspread

import os

import demands

from oauth2client.service\_account import ServiceAccountCredentials

***Set the environment variable for bearer token***

You get your bearer token in your App's keys and tokens area within the developer account . To guarantee you don't coincidentally share your bearer token you can set an environment variable. To do this, within your terminal run the accompanying and make certain to supplant where it says your-bearer token with your own bearer token .

export BEARER\_TOKEN='your-conveyor token '

Within your code editorial manager, make a function called connect\_to\_twitter where you can get your bearer token and return a header that you will pass into your HTTP request to the Twitter API.

def connect\_to\_twitter():

bearer\_token = os.environ.get('BEARER\_TOKEN')

return {"Authorization": "Conveyor {}".format(bearer\_token)}

***Make the Twitter API request***

Make a function called make\_request .Now, the URL will make a request to incorporate the fields you need returned and the Tweet IDs for the Tweets you are hoping to get more data about. Utilizing the request library, make a GET request to the URL variable, pass in your validation header, and determine that you need the request to restore a JSON payload.

def make\_request(headers):

url = "https://api.twitter.com/2/tweets"

params = { "tweet.fields": "author\_id,created\_at,lang",

"ids": "21,1293593516040269825,1334542969530183683", }

return requests.request("GET", url, headers=headers, params=params).json()

***Parsing your JSON payload***

The reaction that gets returned will be within a JSON payload.

To prepare it for lines and segments, make a function called make\_df which will transform your JSON payload into a pandas information outline, which is like a spreadsheet since they are both plain in nature.

def make\_df(response):

return pd.DataFrame(response["data"])

***Verifying to the Google Sheets API***

You'll be utilizing a service account to associate with the Google Sheets API.

To verify to the Google Sheets API, you will initially have to set the degree you will be working with, and pass in your service account credential information.

def authenticate\_to\_google():

scope = [ "https://spreadsheets.google.com/takes care of" ]

credentials = ServiceAccountCredentials.from\_json\_keyfile\_name(

"/way/to/your/file.json", scope

)

return credentialsdef make\_request(headers):

url = "https://api.twitter.com/2/tweets"

params = { "tweet.fields": "author\_id,created\_at,lang",

"ids": "21,1293593516040269825,1334542969530183683", }

return requests.request("GET", url, headers=headers, params=params).json()

def make\_df(response):

return pd.DataFrame(response["data"])

def authenticate\_to\_google():

scope = [ "https://spreadsheets.google.com/takes care of" ]

credentials = ServiceAccountCredentials.from\_json\_keyfile\_name(

"/way/to/your/file.json", scope

)

***Integrating it***

Presently, make a main function to call the entirety of the functions we've composed so far.

def principle():

headers = connect\_to\_twitter()

reaction = make\_request(headers)

df = make\_df(response)

qualifications = authenticate\_to\_google()

gc = gspread.authorize(credentials)

exercise manual = gc.open\_by\_key("spreadsheet\_id")

sheet = workbook.worksheet("Sheet1")

sheet.update("A1", [df.columns.values.tolist()] + df.values.tolist())

if \_\_name\_\_ == "\_\_main\_\_":

main()

TITLE :The most effective method to Send Tweets with Google

Scripts and Twitter API

KEYWORD : Tweets, Twitter API

AUTHOR : Shraddha Jain