## Import Python libraries ##

import http.client

import json

import numpy as np

import pandas as pd

from google.colab import files

## ------ Ignore this section ----- ##

# 2-letter ISO country codes:

#----- ae = United Arab Emirates, ar = Argentina, at = Austria, au = Australia, be = Belgium, bg = Bulgaria, br = Brazil, ca = Canada, ch = Switzerland, cn = China,

#----- co = Colombia, cu = Cuba, cz = Czech Republic, de = Germany, eg = Egypt, fr = France, gb = United Kingdom, gr = Greece, hk = Hong Kong, hu = Hungary, id = Indonesia,

#----- ie = Ireland, il = Israel, in = India, it = Italy, jp = Japan, kr = South Korea, lt = Lithuania, lv = Latvia, ma = Morocco, mx = Mexico, my = Malaysia, ng = Nigeria,

#----- nl = Netherlands, no = Norway, nz = New Zealand, ph = Philippines, pl = Poland, pt = Portugal, ro = Romania, rs = Serbia, ru = Russia, sa = Saudi Arabia, se = Sweden,

#----- sg = Singapore, si = Slovenia, sk = Slovakia, th = Thailand, tr = Turkey, tw = Taiwan, ua = Ukraine, us = USA, ve = Venezuela, za = South Africa

## -------------------------------- ##

## Initialize websites\_str, url\_query and dataFrame ##

websites\_str = ""

url\_query = ""

df\_main = pd.DataFrame([])

## Define news websites ##

websites = ['abs-cbn.com', 'afr.com', 'apnews.com', 'asiaone.com', 'axios.com',

'bangkokpost.com', 'bbc.co.uk', 'bloomberg.com', 'businessinsider.com', 'businesstimes.com.sg', 'business-standard.com', 'brisbanetimes.com.au',

'cbsnews.com', 'channelnewsasia.com', 'cnn.com', 'cnnphilippines.com',

'dailymail.co.uk', 'dailytelegraph.com.au', 'deccanchronicle.com',

'economist.com', 'elevenmyanmar.com', 'financialexpress.com', 'financialpost.com', 'freemalaysiatoday.com', 'ft.com', 'fortune.com',

'go.com', 'gmanetwork.com', 'gulf-times.com',

'huffingtonpost.com',

'independent.co.uk', 'indianexpress.com', 'indiatimes.com', 'indiatoday.in', 'inquirer.net', 'intellasia.net', 'irishtimes.com', 'irrawaddy.com',

'jakartaglobe.id',

'livemint.com',

'manilatimes.net','mb.com.ph', 'millenniumpost.in', 'mmtimes.com', 'msnbc.com',

'nationmultimedia.com', 'nbcnews.com', 'ndtv.com', 'news.com.au', 'news24.com', 'newsweek.com', 'nst.com.my', 'nymag.com', 'nytimes.com',

'philstar.com',

'reuters.com',

'smh.com.au', 'straitstimes.com', 'sunstar.com.ph',

'telegraph.co.uk', 'theaustralian.com.au', 'theedgemarkets.com', 'theglobeandmail.com', 'theguardian.com', 'thehindu.com', 'thejakartapost.com',

'themalaymailonline.com', 'thestar.com.my', 'thesundaily.my', 'theverge.com', 'time.com', 'timesnownews.com',

'vietnamnet.vn', 'vietnamnews.vn',

'washingtonpost.com', 'wired.com', 'wsj.com']

### Note: These websites are not found in newsapi: www.NNA.jp, www.bisnis.com, bworldonline.com, www.mmbiztoday.com, www.bnionline.net

## Join websites together into a long string ##

for news in range(len(websites)-1):

websites\_str = websites\_str + str(websites[news]) + ","

websites\_str = websites\_str + str(websites[len(websites)-1])

## Query companies and actions using keyword search ##

company = "sapporo" # In this example, I have used Sapporo

actions\_list = ['M&A', 'merge', 'acquire', 'invest', 'inject', 'venture', 'collaborate']

## Create url-encoded query for selected company and actions\_list ##

for action in range(len(actions\_list)-1):

url\_query = url\_query + actions\_list[action] + "%20OR%20"

url\_query = company + "%20AND%20" + "(" + url\_query + actions\_list[-1] + ")"

## Establish connection with newsapi.org ##

conn = http.client.HTTPSConnection("newsapi.org")

token = "apiKey=fa141b79c4164e43b77ec0ad15202106" #----- Default api token -----#

conn.request("GET", "/v2/everything?q=" + url\_query + "&" + "domains=" + websites\_str + "&pageSize=100&" + "sortby=publishedAt" + "&" + token)

res = conn.getresponse()

## Import data using json and pandas libraries ##

temp = res.read()

dict = json.loads(temp) # dict is a dictionary.

## Print error message if website is invalid or no feeds are found ##

if dict.get('articles') == []:

print ("Website is invalid or no feeds are found!")

## Create a Pandas Dataframe using 'articles' if website is valid ##

else:

df\_main = pd.DataFrame(dict['articles'])

## Update starting index from 0 to 1 using numpy library ##

df\_main.index = np.arange(1,len(df\_main)+1)

## Download data as News.csv in 'Download' folder of local directory ##

df\_main.to\_csv(company.upper() + 'News.csv')

files.download(company.upper() + 'News.csv')