**Extract Sentence from Website Related to the Stock**

############################################################################################

# #

# This code is written during team meet-up by team members as shown below ... #

# #

# HAFIFI BIN YAHYA - WQD170042 #

# NOR ASMIDAH BINTI MOHD ARSHAD - WQD180006 #

# MAS RAHAYU BINTI MOHAMAD - WQD180048 #

# LEE CHUN MUN - WQD180066 #

# JOJIE ANAK NANJU - WQD180029 #

# #

############################################################################################

import requests

import re

import pymysql

from bs4 import BeautifulSoup

# Retrieving list of main board companies from KLSE database, Main\_Listed\_Companies table.

connection = pymysql.connect(host='localhost', user='root', password='', db='KLSE')

cursor = connection.cursor()

query = "SELECT `stock\_code` FROM `Main\_Listed\_Companies`"

cursor.execute(query)

select\_result = cursor.fetchall()

companies\_codes = [list(i) for i in select\_result]

#print(companies\_codes) # For debugging purpose

for company\_code in companies\_codes:

company\_code = company\_code[0]

#print(company\_code) # For debugging purpose

# Retrieving all weblinks(news) those related to the company\_code or stock\_code.

connection = pymysql.connect(host='localhost', user='root', password='', db='KLSE')

cursor = connection.cursor()

query = "SELECT `weblink` FROM `News\_Announcement\_Links` where `stock\_code` = "

query = query + "'" + str(company\_code) + "'"

cursor.execute(query)

select\_result = cursor.fetchall()

weblinks = [list(i) for i in select\_result]

print(weblinks) # For debugging purpose

for weblink in weblinks:

weblink = weblink[0]

print(weblink)

main\_site = 'https://www.klsescreener.com'

URL = main\_site + weblink

list\_of\_searches\_in\_html = ['h1', 'h2', 'h3', 'p', 'a', 'ul', 'span', 'input']

website = requests.get(URL)

soup = BeautifulSoup(website.content, 'lxml')

tags = soup.find\_all(list)

connection = pymysql.connect(host='localhost', user='root', password='', db='KLSE')

cursor = connection.cursor()

query = "SELECT `known\_name` FROM `Main\_Listed\_Companies` where `stock\_code` ="

query = query + "'" + str(company\_code) + "'"

cursor.execute(query)

select\_result = cursor.fetchall()

known\_names = [list(i) for i in select\_result]

for known\_name in known\_names:

known\_name = known\_name[0]

known\_name = ''.join(known\_name.split()[:1])

regex = str('^\s\*') + str(known\_name) + str('.\*')

print(regex)

text = soup.find\_all(text=re.compile(regex, re.I))

print(company\_code)

print(text)

if text is not None:

text = str(text)

connection = pymysql.connect(host='localhost', user='root', password='', db='KLSE')

cursor = connection.cursor()

try:

insert\_to\_database = (company\_code, text, weblink)

sql = 'INSERT INTO News\_Announcement\_Extraction (stock\_code, news\_details, link) VALUES (%s,%s,%s)'

cursor.execute(sql, insert\_to\_database)

connection.commit()

except:

print('No Related News')

**News\_Announcement\_Links\_KLSEScreener**

# Importing related Python modules/packages.

from bs4 import BeautifulSoup

from urllib.request import Request, urlopen

import re

import pymysql

# Retrieving list of main board companies from KLSE database, Main\_Listed\_Companies table.

connection = pymysql.connect(host='localhost', user='root', password='', db='KLSE')

cursor = connection.cursor()

query = "SELECT `stock\_code` FROM `Main\_Listed\_Companies`"

cursor.execute(query)

select\_result = cursor.fetchall()

companies\_codes = [list(i) for i in select\_result]

#print(companies\_codes) # For debugging purpose

for company\_code in companies\_codes:

company\_code = company\_code[0]

#print(company\_code) # For debugging purpose

#company\_code = str(4723) # For debugging purpose

main\_site = 'https://www.klsescreener.com'

URL = 'https://www.klsescreener.com/v2/stocks/view/'

URL = URL + str(company\_code)

print(URL)

req = Request(URL)

html\_page = urlopen(req)

soup = BeautifulSoup(html\_page, "lxml")

#print(soup)

weblinks = []

for weblink in soup.findAll('a'):

weblinks.append(weblink.get('href'))

stock\_news\_links =[]

for weblink in weblinks:

if weblink != None:

stock\_news\_links.append(weblink)

#print(weblinks)

#print(stock\_news\_links)

for stock\_news\_link in stock\_news\_links:

print(stock\_news\_link)

x = re.findall("news|announcements", stock\_news\_link)

if (x):

# Storing scrapped data to News\_Announcement\_Links table in KLSE database.

connection = pymysql.connect(host='localhost', user='root', password='', db='KLSE')

cursor = connection.cursor()

insert\_to\_database = (company\_code, main\_site, stock\_news\_link)

sql = 'INSERT INTO News\_Announcement\_Links (stock\_code, main\_site, weblink) VALUES (%s,%s,%s)'

cursor.execute(sql, insert\_to\_database)

connection.commit()

print('Stock Code:'+ company\_code)

print('Website:'+ main\_site + stock\_news\_link)

print("Records were saved to database successfully\n\n")

else:

print("No match")

**News\_Announcement\_Links\_The\_Star**

# Importing related Python modules/packages.

from bs4 import BeautifulSoup

from urllib.request import Request, urlopen

import re

import pymysql

# Retrieving list of main board companies from KLSE database, Main\_Listed\_Companies table.

connection = pymysql.connect(host='localhost', user='root', password='', db='KLSE')

cursor = connection.cursor()

query = "SELECT `stock\_code` FROM `Main\_Listed\_Companies`"

cursor.execute(query)

select\_result = cursor.fetchall()

companies\_codes = [list(i) for i in select\_result]

#print(companies\_codes) # For debugging purpose

for company\_code in companies\_codes:

company\_code = company\_code[0]

#print(company\_code) # For debugging purpose

#company\_code = str(4723) # For debugging purpose

main\_site = 'https://www.thestar.com.my'

URL = 'https://www.thestar.com.my/business/marketwatch/stocks/?qcounter='

URL = URL + str(company\_code)

print(URL)

req = Request(URL)

html\_page = urlopen(req)

soup = BeautifulSoup(html\_page, "lxml")

#print(soup)

weblinks = []

for weblink in soup.findAll('a'):

weblinks.append(weblink.get('href'))

stock\_news\_links =[]

for weblink in weblinks:

if weblink != None:

stock\_news\_links.append(weblink)

#print(weblinks)

#print(stock\_news\_links)

for stock\_news\_link in stock\_news\_links:

print(stock\_news\_link)

x = re.findall("news|announcements", stock\_news\_link)

if (x):

# Storing scrapped data to News\_Announcement\_Links table in KLSE database.

connection = pymysql.connect(host='localhost', user='root', password='', db='KLSE')

cursor = connection.cursor()

insert\_to\_database = (company\_code, main\_site, stock\_news\_link)

sql = 'INSERT INTO News\_Announcement\_Links (stock\_code, main\_site, weblink) VALUES (%s,%s,%s)'

cursor.execute(sql, insert\_to\_database)

connection.commit()

print('Stock Code:'+ company\_code)

print('Website:'+ main\_site + stock\_news\_link)

print("Records were saved to database successfully\n\n")

else:

print("No match")

**Quarter\_Reports**

# Importing related Python modules/packages.

import lxml.html

import requests

import pymysql

# Retrieving list of main board companies from KLSE database, Main\_Listed\_Companies table.

connection = pymysql.connect(host='localhost', user='root', password='', db='KLSE')

cursor = connection.cursor()

query = "SELECT `stock\_code` FROM `Main\_Listed\_Companies`"

cursor.execute(query)

select\_result = cursor.fetchall()

companies\_codes = [list(i) for i in select\_result]

#print(companies\_codes) # For debugging purpose

# Using for loop to crawl on every company from klsescreener website.

# The basic URL is shown below.

# The URL is concatenate with the company or stock code.

# As the result, the URL can be used as html address to go to specific stock page of the respective company.

for company\_code in companies\_codes:

company\_code = company\_code[0]

#print(company\_code) # For debugging purpose

#company\_code = str(4723) # For debugging purpose

URL = 'https://www.klsescreener.com/v2/stocks/view/'

URL = URL + str(company\_code)

print(URL)

html = requests.get(URL)

doc = lxml.html.fromstring(html.content)

# QR = Quarter Report Tab Path.

# We scrape data from Quarter Report tab of the webpage.

QR = doc.xpath('//div[@id="quarter\_reports"]')[0]

# Most of the companies release Quarter Reports

# Means 4 reports for every year.

# But there are companies that not released the previous years reports such as for year 2016, 2017 etc.

i = [-6,-5,-4,-3,-2,-1]

q = 1

a = 6

# The code gives flexibility in scrapping Quarter Reports.

# If the user needs to scrap Quarter Report just for a year back, then user may use q < 5.

# If the user needs to scrap Quarter Report for 2 years back, then user may use q < 9.

# And so on for 3 years, 4 years back.

while q < 9:

if q <= 4:

Q\_Year = 2018

elif 5 <= q <= 8:

Q\_Year = 2017

elif 9 <= q <= 12:

Q\_Year = 2016

elif 13 <= q <= 16:

Q\_Year = 2015

elif 17 <= q <= 20:

Q\_Year = 2014

# Scrapped quarter financial data.

null = None

try:

EPS = QR.xpath('.//td[@class="number"]/text()')[i[0]+a]

except:

EPS = str(null)

try:

DPS = QR.xpath('.//td[@class="number"]/text()')[i[1]+a]

except:

DPS = str(null)

try:

NTA = QR.xpath('.//td[@class="number"]/text()')[i[2]+a]

except:

NTA = str(null)

try:

Revenue = QR.xpath('.//td[@class="number"]/text()')[i[3]+a]

except:

Revenue = str(null)

try:

PL = QR.xpath('.//td[@class="number"]/text()')[i[4]+a]

except:

PL = str(null)

try:

Quarter = QR.xpath('.//td[@class="number"]/text()')[i[5]+a]

except:

Quarter = str(null)

q = q + 1

a = a + 6

# Storing scrapped data to Quarter\_Reports table in KLSE database.

connection = pymysql.connect(host='localhost', user='root', password='', db='KLSE')

cursor = connection.cursor()

insert\_to\_database = (company\_code, EPS, DPS, NTA, Revenue, PL, Quarter, Q\_Year)

sql = 'INSERT INTO Quater\_Reports (stock\_code, EPS, DPS, NTA, Revenue, PL, Quarter, Q\_Year) VALUES (%s,%s,%s,%s,%s,%s,%s,%s)'

cursor.execute(sql, insert\_to\_database)

connection.commit()

# For debugging purpose

print('Stock Code:'+ company\_code)

print('EPS:'+ EPS)

print('DPS:'+ DPS)

print('NTA:'+ NTA)

print('Revenue:'+ Revenue)

print('PL:'+ PL)

print('Quarter:'+ Quarter)

print('Q Year:' + str(Q\_Year))

print("Records were saved to database successfully\n\n")

**Tweeter**

from \_\_future\_\_ import print\_function

import tweepy

import json

#import MySQLdb

#import mysqlclient

#from dateutil import parser

from dateutil.parser import \*

import pymysql

#WORDS = ['#bigdata', '#AI', '#datascience', '#machinelearning', '#ml', '#iot']

#WORDS = ['#news', '#BursaMalaysia', '#stocknews','#KLCI', '#BursaMalaysia', '#openingbell']

WORDS = ['#BursaMalaysia', '#stocknews','#KLCI', '#openingbell', '#stockinvestment', '#klsemarket', '#BursaMalaysiaTradeStatistics', '#1mdb', '#stockexchange',

'#Bursa', '#Market', '#Index', '#stock', '#stocknews', '#KLSEtraders', '#KLSEnews', '#updates','#klsetraders']

#IWD2019

#CONSUMER\_KEY = "KEY"

#CONSUMER\_SECRET = "SECRET"

#ACCESS\_TOKEN = "TOKEN"

#ACCESS\_TOKEN\_SECRET = "TOKEN\_SECRET"

ACCESS\_TOKEN = "69793210-cQfMYtjTriuL2dB0nCzftkTjYBsSB0z8YM4MWKo6n"

ACCESS\_TOKEN\_SECRET = "EtGiwIWFkYkFwCxUyvW3qOkbylgb4ONy5Np9MRTAEJQv2"

CONSUMER\_KEY = "DSTr85cSnpFSELigMDsPoKGYI"

CONSUMER\_SECRET = "ke9a5eXnRrVfNbeKU36UyRJegdIgi3QhUrBIsMC5OYBUlodm0X"

HOST = "localhost"

USER = "root"

PASSWD = ""

DATABASE = "KLSE"

# This function takes the 'created\_at', 'text', 'screen\_name' and 'tweet\_id' and stores it

# into a MySQL database

def store\_data(created\_at, text, screen\_name, tweet\_id):

db = pymysql.connect(host=HOST, user=USER, passwd=PASSWD, db=DATABASE, charset="utf8")

cursor = db.cursor()

insert\_query = "INSERT INTO twitter (tweet\_id, screen\_name, created\_at, text) VALUES (%s, %s, %s, %s)"

cursor.execute(insert\_query, (tweet\_id, screen\_name, created\_at, text))

db.commit()

cursor.close()

db.close()

return

class StreamListener(tweepy.StreamListener):

# This is a class provided by tweepy to access the Twitter Streaming API.

def on\_connect(self):

# Called initially to connect to the Streaming API

print("You are now connected to the streaming API.")

def on\_error(self, status\_code):

# On error - if an error occurs, display the error / status code

print('An Error has occured: ' + repr(status\_code))

return False

def on\_data(self, data):

# This is the meat of the script...it connects to your mongoDB and stores the tweet

try:

# Decode the JSON from Twitter

datajson = json.loads(data)

# grab the wanted data from the Tweet

text = datajson['text']

screen\_name = datajson['user']['screen\_name']

tweet\_id = datajson['id']

created\_at = parse(datajson['created\_at'])

# print out a message to the screen that we have collected a tweet

print("Tweet collected at " + str(created\_at))

# insert the data into the MySQL database

store\_data(created\_at, text, screen\_name, tweet\_id)

except Exception as e:

print(e)

auth = tweepy.OAuthHandler(CONSUMER\_KEY, CONSUMER\_SECRET)

auth.set\_access\_token(ACCESS\_TOKEN, ACCESS\_TOKEN\_SECRET)

# Set up the listener. The 'wait\_on\_rate\_limit=True' is needed to help with Twitter API rate limiting.

listener = StreamListener(api=tweepy.API(wait\_on\_rate\_limit=True))

streamer = tweepy.Stream(auth=auth, listener=listener)

print("Tracking: " + str(WORDS))

streamer.filter(track=WORDS)