New York Times API with `Parsinig JSON` data (ง'-')ง

Mr Fugu Data Science

(◕‿◕✿)

Purpose & Outcome:

+

In [ ]:
```
import requests # get connection
import pandas as pd
from bs4 import BeautifulSoup as bsopa # webscrape
import json
```

Getting Started with the API:

GO TO: https://developer.nytimes.com/get-started (https://developer.nytimes.com/get-started)

You will have easy access to this api, just follow the directions.

```
Today we will be working with the [Most Popular API] from the New York Times
```

The general layout will look something like this:

```
https://api.nytimes.com/svc/mostpopular/v2/viewed/1.json?api-key=yourkey
```

First: we have to get an Idea of what we can use

For instance `/viewed/7.json` this will infer that we are looking at what was most popular and viewed in the last 7 days.

You have 3 options: emailed, shared, viewed

and if you were using a different API, you would change the part that says `mostpopular`

ex.)

https://api.nytimes.com/svc/mostpopular/v2/ (https://api.nytimes.com/svc/mostpopular/v2/)viewed/7.json?api-key=yourkey

the other portion is your `API key` , this should be hidden for obvious use `(think of) your password. You don't want some pirate taking your booty` .

(*But, if you are to use other API's make sure to use the check box so you will hae permission when you register api usage*)

In [ ]:
```
YOUR_API_KEY='Your Key Here'
base_url = 'https://api.nytimes.com/svc/mostpopular/v2/viewed/7.json?api-key='+YOUR_API_KEY
requests.get(base_url).json()
```

In [ ]:
```
# find the outer keys:

list(requests.get(base_url).json())
```

In [ ]:
```
# I want everything within [results]
dta_=[]
for i in requests.get(base_url).json()['results']:
    dta_.append(i)
```

In [21]:
```
# Extract Everything and place into a dataframe:
df_=pd.DataFrame(dta_)
df_.head()
```

Out[21]:

	uri	url	id	
0	nyt://article/1207a023-e98b-5dfd-a379-fb1461e5...	https://www.nytimes.com/2020/08/27/us/hurrican...	100000007309705	10000(
1	nyt://article/fa102828-c20a-5f7c-8685-de98792a...	https://www.nytimes.com/2020/08/27/us/kyle-rit...	100000007309185	10000(
2	nyt://article/607123ea-14ba-5f9c-ab43-7d8b6c7a...	https://www.nytimes.com/2020/08/28/movies/chad...	100000007314593	10000(
3	nyt://article/6bff4972-07cc-5b20-bd16-39f9cf19...	https://www.nytimes.com/2020/08/30/us/portland...	100000007315198	10000(
4	nyt://article/0487a919-ec10-5bf5-8f65-449c7a78...	https://www.nytimes.com/2020/08/29/health/coro...	100000007294406	10000(

5 rows × 22 columns

What if I had a clue of exact columns to take?

Let's instead systematically extract these columns and place them into a dataframe or csv file

- There is one more field I want to use as a nested form to enter: `['media'][0]['caption']` which may or may not exist. If it exists you will have a string, else NA.

In [ ]:
```
my_data={'source':[],'published_date':[],'adx_keywords':[],'byline':[],
         'title':[],'abstract':[],'des_facet':[],'per_facet':[],'media':
[]}
```

In [ ]:
```
pop_articles=requests.get(base_url).json()['results']
```

Check What is in `Media` so we can parse it

In [23]:
```
# dta_=[]
# for i in requests.get(base_url).json()['results']:
#     dta_.append(i['media'])
# dta_
```

In [17]:
```
h=[] # store our media which is a list of dictionaries

for news_stuff in pop_articles:
    if 'source' in news_stuff:
        my_data['source'].append(news_stuff['source'])
    if 'published_date' in news_stuff:
        my_data['published_date'].append(news_stuff['published_date'])

    if 'adx_keywords' in news_stuff:
        my_data['adx_keywords'].append(news_stuff['adx_keywords'])
    if 'byline' in news_stuff:
        my_data['byline'].append(news_stuff['byline'])
    else:
        my_data['byline'].append(None)
    if 'title' in news_stuff:
        my_data['title'].append(news_stuff['title'])
    if 'abstract' in news_stuff:
        my_data['abstract'].append(news_stuff['abstract'])

    if 'des_facet' in news_stuff:
        my_data['des_facet'].append(news_stuff['des_facet'])
    else:
        my_data['des_facet'].append(None)

    if 'per_facet' in news_stuff:
        my_data['per_facet'].append(news_stuff['per_facet'])
    else:
        my_data['per_facet'].append(None)

    if 'media' in news_stuff:
        h.append(news_stuff['media'])
    else:
        my_data['media'].append(None)
```

In [18]:
```
# Enter Nested Portion of Data:
a=[] # store the captions which are strings I want
for i in h:
    if i==[]:
        a.append(None)
    else:
        for j in i:
            if 'caption' in j:
                a.append(j['caption'])
            else:
                  a.append(None)
```

In [19]:
```
# Iterate through new data for captions:
for i in a:
    if i =='':
        my_data['media'].append(None)
    else:
        my_data['media'].append(i)
```

In [20]:
```
# dataframe with everything I want for later:
df_fin=pd.DataFrame(my_data)
df_fin.head()
```

Out[20]:

	source	published_date	adx_keywords	byline	title	abstract	
0	New York Times	2020-08-27	Hurricane Laura (2020);Deaths (Fatalities);Pow...		Hurricane Laura Kills at Least 6 People in Lou...	After landfall overnight with 150 m.p.h. winds...	[Hurr (2((F
1	New York Times	2020-08-27	Murders, Attempted Murders and Homicides;Demon...	By Haley Willis, Muyi Xiao, Christiaan Trieber...	Tracking the Suspect in the Fatal Kenosha Shoo...	Footage appears to show a teenager shooting th...	N Homici
2	New York Times	2020-08-28	Deaths (Obituaries);Movies;Actors and Actresse...	By Reggie Ugwu and Michael Levenson	‘Black Panther’ Star Chadwick Boseman Dies of ...	The actor also played groundbreaking figures l...	(Mo
3	New York Times	2020-08-30	George Floyd Protests (2020);Demonstrations, P...	By Mike Baker	One Person Dead in Portland After Clashes Betw...	A man affiliated with a right-wing group was s...	[G Prot Demor
4	New York Times	2020-08-29	Coronavirus (2019-nCoV);Tests (Medical);Contac...	By Apoorva Mandavilli	Your Coronavirus Test Is Positive. Maybe It Sh...	The usual diagnostic tests may simply be too s...	[((2 Test

What next?

How about Sentiment Analysis

LIKE, SHARE &

SUBscribe

Citations & Help:

https://dlab.berkeley.edu/blog/scraping-new-york-times-articles-python-tutorial (https://dlab.berkeley.edu/blog/scraping-new-york-times-articles-python-tutorial)

https://towardsdatascience.com/collecting-data-from-the-new-york-times-over-any-period-of-time-3e365504004 (https://towardsdatascience.com/collecting-data-from-the-new-york-times-over-any-period-of-time-3e365504004)

https://nycdatascience.com/blog/student-works/sentiment-analysis-of-media-coverage-of-presidential-candidates/ (https://nycdatascience.com/blog/student-works/sentiment-analysis-of-media-coverage-of-presidential-candidates/)

https://www.storybench.org/working-with-the-new-york-times-api-in-r/ (https://www.storybench.org/working-with-the-new-york-times-api-in-r/) (using R)

https://code.tutsplus.com/tutorials/using-the-new-york-times-api-to-scrape-metadata--cms-27894 (https://code.tutsplus.com/tutorials/using-the-new-york-times-api-to-scrape-metadata--cms-27894)

https://rstudio-pubs-static.s3.amazonaws.com/543546_05a9719b4e71483ea28f56c601ca4c3d.html (https://rstudio-pubs-static.s3.amazonaws.com/543546_05a9719b4e71483ea28f56c601ca4c3d.html) (R example)

```
Sentiment Analysis
```

https://towardsdatascience.com/https-towardsdatascience-com-algorithmic-trading-using-sentiment-analysis-on-news-articles-83db77966704 (https://towardsdatascience.com/https-towardsdatascience-com-algorithmic-trading-using-sentiment-analysis-on-news-articles-83db77966704)

https://www.kdnuggets.com/2018/08/emotion-sentiment-analysis-practitioners-guide-nlp-5.html (https://www.kdnuggets.com/2018/08/emotion-sentiment-analysis-practitioners-guide-nlp-5.html)

https://www.kaggle.com/mmmarchetti/sentiment-analysis-on-financial-news (https://www.kaggle.com/mmmarchetti/sentiment-analysis-on-financial-news)