Hemant Sathian

Professor Segal

Object-Oriented Programming 04:547:202 Section C7

19 July 2021

New York Times Final Project

Summary: The cryptocurrency Dogecoin has seen a massive rise in value in 2021. Since January 2021, the cryptocurrency has shot up in value by over 12,0000%. A prevalent supporter of Dogecoin is Elon Musk, famous entrepreneur and CEO of Tesla Motors and SpaceX. Elon Musk's tweets about Dogecoin in 2021 have helped the cryptocurrency reach a massive audience. Many have believed Musk's support of Dogecoin to be "jumping on the bandwagon". However, dogecoin developers recently confirmed that Elon Musk had been working with them since 2019 to improve the cryptocurrency. I was curious to see if Elon Musk had been mentioned with any New York Times articles regarding Dogecoin since the cryptocurrency's introduction in 2013 to the end of 2020. I also wanted to obtain all articles in the year 2021 that mentioned both Elon Musk and Dogecoin. I would use the New York Times Article Search API to extract my data by first obtaining an API key. Next, I would create two parameters. The 'params' parameter would filter the data for all articles mentioning Dogecoin between the introduction of the cryptocurrency in 2013 and 2020. The 'params2' parameter would filter the data for all articles mentioning Dogecoin in 2021. Using a for loop to iterate through the data filtered by each parameter, I would create two dictionary objects. The first dictionary object (Article_13_20_list) would contain the publication date, headline, website url, and keyword of all articles mentioning Dogecoin and Elon Musk between 2013 and 2020. The second dictionary object (Article_21_list) would contain the publication date, headline, website url, and keyword of all articles mentioning Dogecoin and Elon Musk in the year 2021. Only one dictionary object (Article_21_list) contained data pertaining to articles that mentioned both Dogecoin and Elon Musk, so this was the dictionary object I would use to create a csv file. I would make a csv file using the pathlib and csv modules in python. The csv file allowed for the processed data (the publication date, headline, website url, and keyword of all NY Times Articles mentioning both Dogecoin and Elon Musk) to be clearly presented.

Code Snippet 1:

```
1 import requests, pathlib, pprint, csv, json, time
2
3 cwd = pathlib.Path.cwd()/"oop_final_proj_files"
4 cwd.mkdir(exist_ok = True)
```

Description: Firstly, I imported the requests, pathlib, pprint, csv, json, and time modules. The requests module allows me to send HTTP requests. The pathlib module allows me to create new directories and files (In this instance, I used pathlib to create the folder "oop_final_proj_files" in my current working directory). The pprint module allows for content to be easily read. The csv module allows me to write tabular data (spreadsheet) in a csv file format using python. The json module allows me to save a file that I would need to reference later. The time module allows me to delay an iteration process when I am accessing data through an API.

Code Snippet 2:

```python
url = 'https://api.nytimes.com/svc/search/v2/articlesearch.json'
API_KEY = 'J8ufcl0Buuf4ZZZxBxxNcCsHFYXL2EZB'
params = {'query': 'dogecoin', 'api-key': API_KEY, 'page': 0, 'fq': 'document_type:("article")',
'begin_date': '20130101', 'end_date': '20201231'}
params2 ={'query': 'dogecoin', 'api-key': API_KEY, 'page': 0, 'fq': 'document_type:("article")',
'begin_date': '20210101', 'end_date': '20211231'}
```

Description: Next, I created my two parameters. The 'params' parameter would filter the data for all articles mentioning Dogecoin between the introduction of the cryptocurrency in 2013 and 2020. The 'params2' parameter would filter the data for all articles mentioning Dogecoin in 2021.

Code Snippet 3:

```python
def api_updated_param(parameters):
    """
    This function will return a number of hits
    based on the supplied parameters.
    args - [dict] parameters in utilizing API
    """
    response = requests.get(url, params = parameters)
    content = response.json()
    return(content['response']['meta']['hits'])

print(f"The number of articles written about dogecoin from its introduction in 2013 to 2020 was
{api_updated_param(parameters = params)}")
print(f"The number of articles written about dogecoin in 2021 was {api_updated_param(parameters =
params2)}")
```

Description: I would define a function called api_update_param. This function allows me to return the number of hits I recieve from the API based on the parameters I supply it. I would use this function to print out both the number of articles mentioning Dogecoin between 2013 and 2020 (14 articles) and the number of articles mentioning Dogecoin in 2021 (52 articles). I was suprised that the New York Times had written nearly 4 times more articles mentioning Dogecoin in 2021 than they had in over 7 years.

Code Snippet 4:

```python
response_dgmk1320 = requests.get(url, params = params)
content_dgmk1320 = response_dgmk1320.json()
Article_13_20_list = []
for b in content_dgmk1320['response']['docs']:
    pub_date = b['pub_date']
    web_url = b['web_url']
    headline = b['headline']['main']
    for k in b['keywords']:
        keyword = k['value']
        if keyword == 'Musk, Elon':
            print(f"date: {pub_date}, headline: {headline}, web_url: {web_url}")
            Article_13_20_list.append({'date':pub_date, 'headline':headline , 'website url':
web_url, 'keyword': keyword})
            time.sleep(2)
    else:
        pass

if Article_13_20_list == []:
    print(f"There was no NY Times article between 2013-2020 concerning Dogecoin and Elon Musk")
else:
    print(f"# of articles concerning Dogecoin and Elon Musk between 2013 and 2020:
{Article_13_20_list}")
```

Description: First I created a personalized response variable (response_dgmk1320) and content variable (content_dgmk1320) for the parameter params (params parameter accounts for all articles mentioning Dogecoin between the introduction of the cryptocurrency in 2013 and 2020). In the third line is an empty list (Article_13_20_list). Using the nested for loop, I'm appending the publication date, headline, website url, and keyword of all NY Times Articles from 2013-2020 that mention both Dogecoin and Elon Musk to the list Article_13_20_list. The sleep function of the time module in this application will wait 2 seconds after running the nested for loop before extracting more data from the API.

If Article_13_20_list returns an empty list, an if loop will print the string “There was no NY Times article between 2013-2020 concerning Dogecoin and Elon Musk”. If Article_13_20_list is not an empty list, the if loop would print the articles that mention both Dogecoin and Elon Musk. (In this instance, Article_13_20_list returned an empty list, indicating that none of the 14 articles written by the NY Times between 2013 and 2020 that mentioned Dogecoin also mentioned Elon Musk).

Code Snippet 5:

```
response_dgmk21 = requests.get(url, params = params2)
content_dgmk21 = response_dgmk21.json()
Article_21_list = []
for b in content_dgmk21['response']['docs']:
    pub_date = b['pub_date']
    web_url = b['web_url']
    headline = b['headline']['main']
    for k in b['keywords']:
        keyword = k['value']
        if keyword == 'Musk, Elon':
            print(f"date: {pub_date}, headline: {headline}, web_url: {web_url}")
            Article_21_list.append({'date':pub_date, 'headline':headline , 'website url': web_url, 'keyword': keyword})
            time.sleep(2)
    else:
        pass

if Article_21_list == []:
    print(f"There was no NY Times article in 2021 concerning Dogecoin and Elon Musk")
else:
    print(f"# of articles concerning Dogecoin and Elon Musk between 2013 and 2020: {Article_21_list}")
```

Description: For the second parameter params2 (params2 parameter accounts for all articles mentioning Dogecoin in 2021), I created a personalized response variable (response_dgmk21) and content variable (content_dgmk21). In the third line is an empty list (Article_21_list). Using the nested for loop, I’m appending the publication date, headline, website url, and keyword of all NY Times Articles written in 2021 that mention both Dogecoin and Elon Musk to the list Article_21_list. The sleep function of the time module in this application will wait 2 seconds after running the nested for loop before extracting more data from the API. If Article_21_list returns an empty list, an if loop will print the string “There was no NY Times article in 2021 concerning Dogecoin and Elon Musk”. If Article_21_list is not an empty list, the if loop would print the articles that mention both Dogecoin and Elon Musk. (In this instance, Article_21_list returned the publication date, headline, website url, and keyword “for all articles: Musk, Elon” of 7 articles that mentioned both Dogecoin and Elon Musk. This meant 7 of the 52 articles written by the NY Times in 2021 mentioned both Dogecoin and Elon Musk).

Code Snippet 6:

```
with open("oop_final_proj_files/Elon Musk and Dogecoin 2021 article(s).json", 'w') as j_file:
        json.dump(Article_21_list, j_file)


with open("oop_final_proj_files/Elon Musk and Dogecoin 2021 article(s).json",'r') as j_file:
    Article_21_load_file = json.load(j_file)

pprint.pprint(f"{Article_21_load_file}")
```

Description:

I wanted to showcase Data Persistence in this project by saving Article_21_list as a json file to be referenced later. I would use the dump method to save Article_21_list as a json file to the “oop_final_proj_files” directory I had created prior. I would use the load method to load the json file back as Article_21_load_file. Next, I would print the contents of the Article_21_load_file using the pprint function of the pprint module just to see if the data regarding the 7 articles I had extracted in the previous iterative process were still there.

Code Snippet 7:

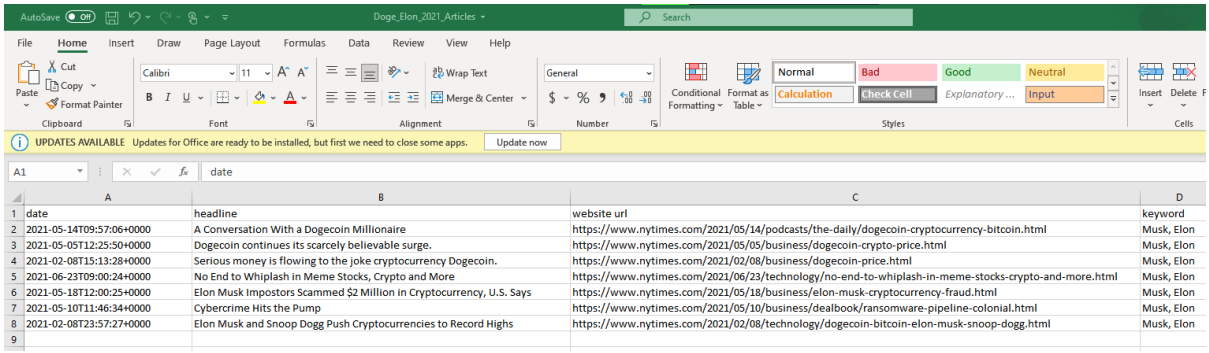
```python
Doge_Elon_2021_file = cwd/"Doge_Elon_2021_Articles.csv"
Doge_Elon_2021_file.touch()

with Doge_Elon_2021_file.open(mode = 'a', encoding = 'utf-8', newline = '') as Sathian_oop_final:
    write = csv.DictWriter(Sathian_oop_final, fieldnames = ["date", "headline", "website url",
"keyword"])
    write.writeheader()
    write.writerows(Article_21_load_file)
```

Description:

Now that I have extracted my data (Article_21_load_file), I am now going to place this data in a CSV file that could be viewed in Excel. Using the touch method, I created a Doge_Elon_2021_Articles.csv file in the "oop_final_proj_files" directory (cwd =pathlib.Path.cwd()/"oop_final_proj_files"). The fieldnames would extract data from Article_21_load_file into Doge_Elon_2021_Articles.csv. The dictwriters make the dictionaries within Article_21_load_file into rows in Doge_Elon_2021_Articles.csv. The writeheader() method writes the field names as column headers in the csv file. The writerows method writes the dictionary values of Article_21_load_file into the rows Doge_Elon_2021_Articles.csv.

CSV File:



	A	B	C	D
1	date	headline	website url	keyword
2	2021-05-14T09:57:06+0000	A Conversation With a Dogecoin Millionaire	https://www.nytimes.com/2021/05/14/podcasts/the-daily/dogecoin-cryptocurrency-bitcoin.html	Musk, Elon
3	2021-05-05T12:25:50+0000	Dogecoin continues its scarcely believable surge.	https://www.nytimes.com/2021/05/05/business/dogecoin-crypto-price.html	Musk, Elon
4	2021-02-08T15:13:28+0000	Serious money is flowing to the joke cryptocurrency Dogecoin.	https://www.nytimes.com/2021/02/08/business/dogecoin-price.html	Musk, Elon
5	2021-06-23T09:00:24+0000	No End to Whiplash in Meme Stocks, Crypto and More	https://www.nytimes.com/2021/06/23/technology/no-end-to-whiplash-in-meme-stocks-crypto-and-more.html	Musk, Elon
6	2021-05-18T12:00:25+0000	Elon Musk Impostors Scammed $2 Million in Cryptocurrency, U.S. Says	https://www.nytimes.com/2021/05/18/business/elon-musk-cryptocurrency-fraud.html	Musk, Elon
7	2021-05-10T11:46:34+0000	Cybercrime Hits the Pump	https://www.nytimes.com/2021/05/10/business/dealbook/ransomware-pipeline-colonial.html	Musk, Elon
8	2021-02-08T23:57:27+0000	Elon Musk and Snoop Dogg Push Cryptocurrencies to Record Highs	https://www.nytimes.com/2021/02/08/technology/dogecoin-bitcoin-elon-musk-snoop-dogg.html	Musk, Elon
9				

Description:

The CSV file displays the publication date, headline, website url, and keyword for the 7 NY Times articles that mentioned both Dogecoin and Elon Musk.

More Ideas: Regarding what I would do if I chose to do a follow-up project, I would like to expand further on the articles mentioning both Dogecoin and Elon Musk as to determine whether Elon Musk appears more so in Dogecoin articles that report a rise in value for the cryptocurrency, or does he occur more so in Dogecoin articles that report a drop in value for the cryptocurrency. I would use the same iterative process I had done for this project, albeit filtering for different keywords that either indicate a rise in value for the cryptocurrency or a drop in value. Another idea I'd be curious to analyze would be the most popular keywords associated to the articles I extracted in my first project. Elon Musk is a prevalent figure regarding the Dogecoin crypotcurrency, but it would be interesting to see any other popular keywords that are associated with articles that mention both Dogecoin and Elon Musk.