Open in Colab

Using API Calls in Python with JSON and DataFrames - University Data Example

In this notebook, we will make API requests to the Hipolabs Universities API, parse the JSON response, and convert it into Pandas DataFrames for analysis and visualization.

Step 1: Making the API Call to Fetch University Data

In [45]:
```
import requests
import json
import pandas as pd

# API endpoint for fetching universities
url = "http://universities.hipolabs.com/search?country=United%20States"

# Make the GET request
response = requests.get(url)

# Check the status code
if response.status_code == 200:
    print("API request successful!")
else:
    print(f"Failed to retrieve data: {response.status_code}")
```

```
API request successful!
```

Step 2: Parsing the JSON Response

In [46]:
```
# Parse the JSON response
university_data = response.json()

# Pretty-print the first university
print(json.dumps(university_data[0], indent=2))
```


```
{
  "domains": [
    "marywood.edu"
  ],
  "alpha_two_code": "US",
  "web_pages": [
    "http://www.marywood.edu"
  ],
  "name": "Marywood University",
  "state-province": null,
  "country": "United States"
}
```

Step 3: Converting the JSON Data to a DataFrame

In [47]:
```
# Convert JSON data to a DataFrame
df = pd.DataFrame(university_data)

# Display the first few rows of the DataFrame
print(df.head())
df
```

```
           domains alpha_two_code                    web_pages  \
0    [marywood.edu]             US     [http://www.marywood.edu]
1  [lindenwood.edu]             US  [http://www.lindenwood.edu/]
2    [sullivan.edu]             US       [https://sullivan.edu/]
3        [fscj.edu]             US       [https://www.fscj.edu/]
4      [xavier.edu]             US     [https://www.xavier.edu/]

                                  name state-province        country
0                   Marywood University           None  United States
1                 Lindenwood University           None  United States
2                   Sullivan University           None  United States
3  Florida State College at Jacksonville           None  United States
4                     Xavier University           None  United States
```

Out[47]:

	domains	alpha_two_code	web_pages	name	pr
0	[marywood.edu]	US	[http://www.marywood.edu]	Marywood University	
1	[lindenwood.edu]	US	[http://www.lindenwood.edu/]	Lindenwood University	
2	[sullivan.edu]	US	[https://sullivan.edu/]	Sullivan University	
3	[fscj.edu]	US	[https://www.fscj.edu/]	Florida State College at Jacksonville	
4	[xavier.edu]	US	[https://www.xavier.edu/]	Xavier University	
...	...	...	...	...	
2329	[vermontlaw.edu]	US	[https://www.vermontlaw.edu/]	Vermont Law School	
2330	[wnc.edu]	US	[https://wnc.edu/]	Western Nevada College	
2331	[westernu.edu]	US	[https://www.westernu.edu/]	Western University of Health Sciences	
2332	[stmarytx.edu]	US	[https://www.stmarytx.edu/]	St. Mary's University	
2333	[southflorida.edu]	US	[https://www.southflorida.edu/]	South Florida State College	

2334 rows × 6 columns

Step 4: Basic Data Analysis

In [48]:
```
# Count the number of universities by state
state_counts = df['state-province'].value_counts()
print(state_counts)
```

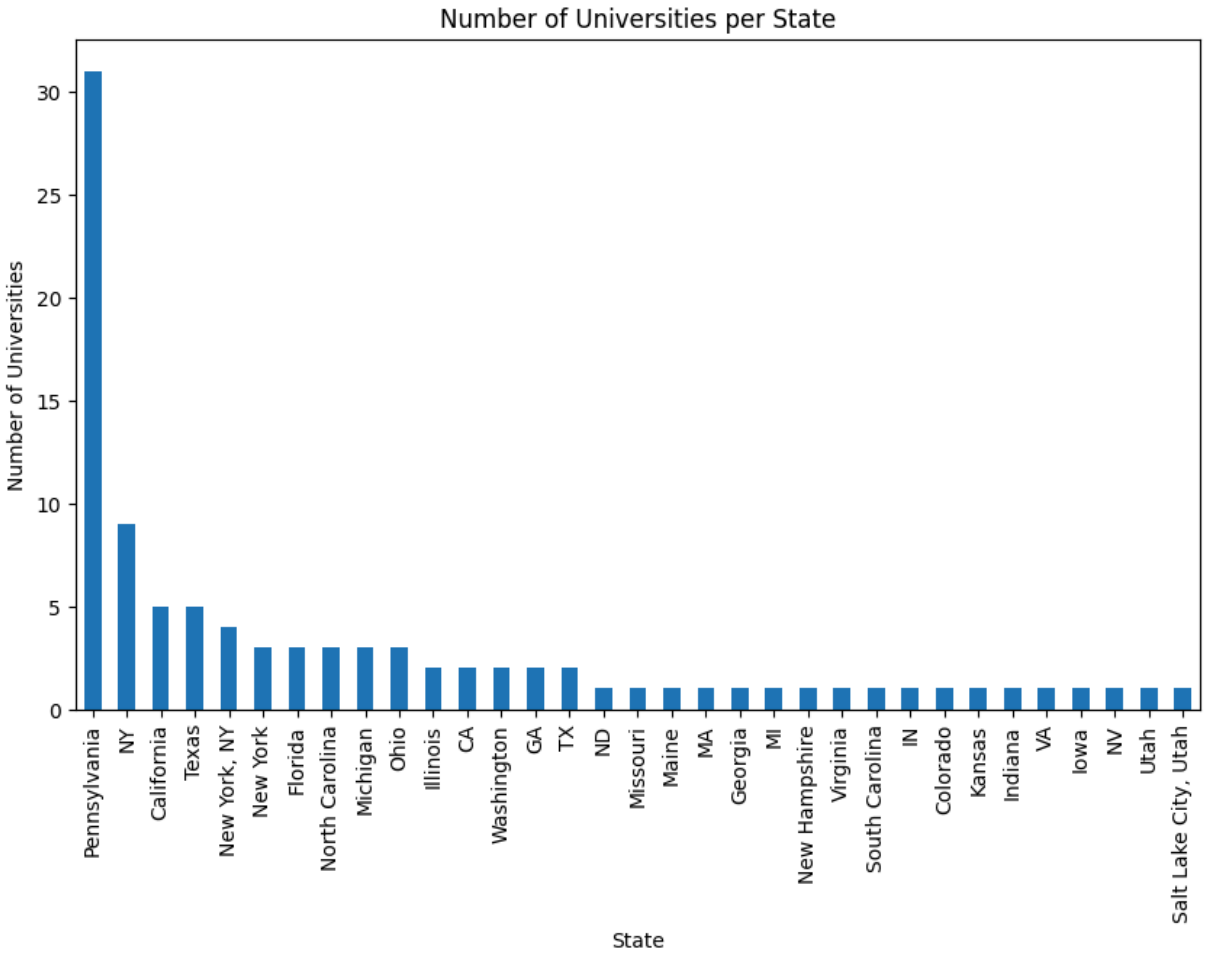
```
state-province
Pennsylvania            31
NY                       9
California               5
Texas                    5
New York, NY             4
New York                 3
Florida                  3
North Carolina           3
Michigan                 3
Ohio                     3
Illinois                 2
CA                       2
Washington               2
GA                       2
TX                       2
ND                       1
Missouri                 1
Maine                    1
MA                       1
Georgia                  1
MI                       1
New Hampshire            1
Virginia                 1
South Carolina           1
IN                       1
Colorado                 1
Kansas                   1
Indiana                  1
VA                       1
Iowa                     1
NV                       1
Utah                     1
Salt Lake City, Utah     1
Name: count, dtype: int64
```

Step 5: Data Visualization

In [49]:

```python
import matplotlib.pyplot as plt

# Plot the number of universities per state
state_counts.plot(kind='bar', figsize=(10, 6))
plt.title("Number of Universities per State")
plt.xlabel("State")
plt.ylabel("Number of Universities")
plt.show()
```



Step 6: Extending to Multiple Countries

In [50]:

```
countries = ["United States", "Canada", "Australia", "United Kingdom"]
university_list = []

for country in countries:
    response = requests.get(f"http://universities.hipolabs.com/search?countr
    data = response.json()

    for uni in data:
        uni['country'] = country
        university_list.append(uni)

# Convert to DataFrame
df_universities = pd.DataFrame(university_list)

# Display the first few rows
print(df_universities.head())
```

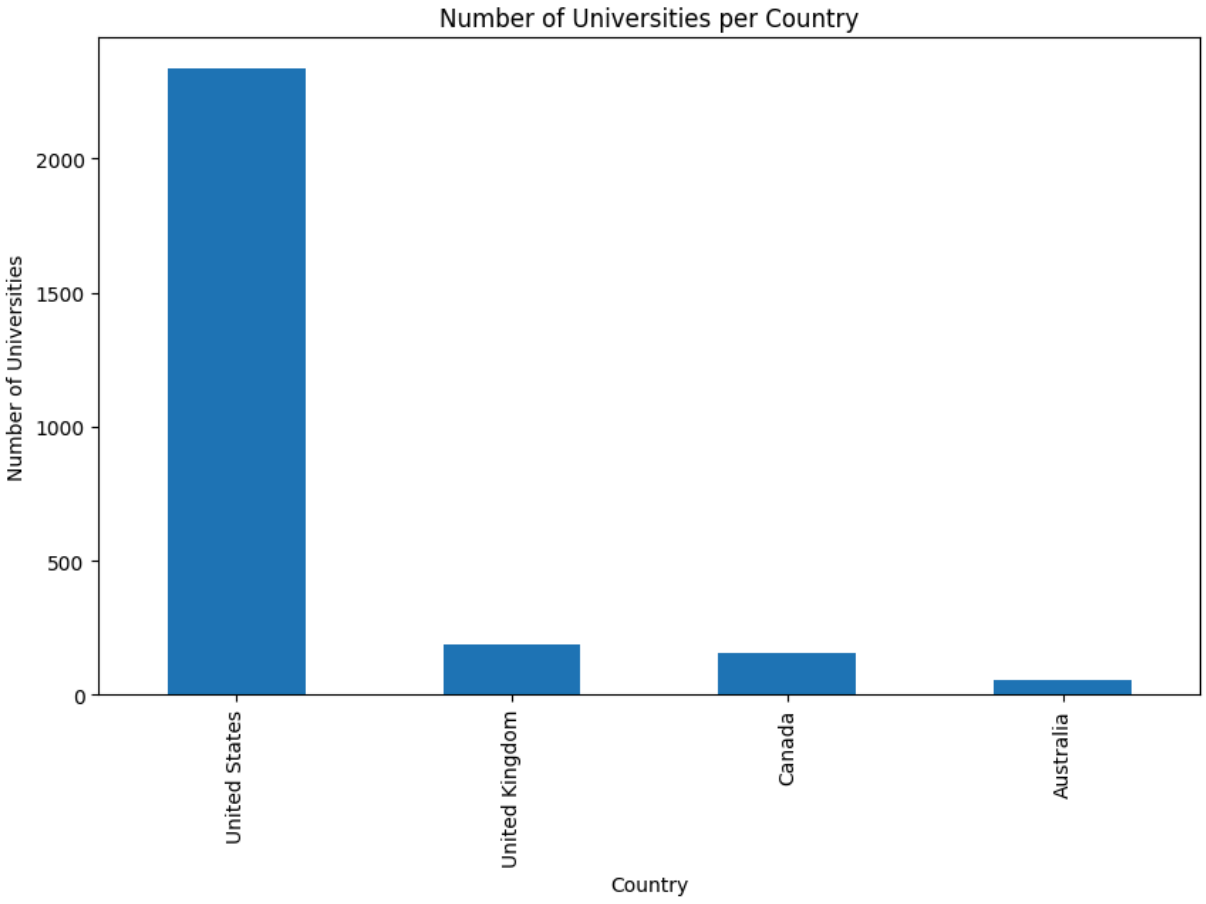
```
            domains alpha_two_code                      web_pages  \
0    [marywood.edu]             US     [http://www.marywood.edu]
1  [lindenwood.edu]             US  [http://www.lindenwood.edu/]
2    [sullivan.edu]             US       [https://sullivan.edu/]
3        [fscj.edu]             US       [https://www.fscj.edu/]
4      [xavier.edu]             US     [https://www.xavier.edu/]

                                name state-province        country
0                 Marywood University           None  United States
1               Lindenwood University           None  United States
2                 Sullivan University           None  United States
3  Florida State College at Jacksonville           None  United States
4                     Xavier University           None  United States
```

Step 7: Visualizing University Counts by Country

In [51]:
```
# Count the number of universities per country
country_counts = df_universities['country'].value_counts()

# Plot the data
country_counts.plot(kind='bar', figsize=(10, 6))
plt.title("Number of Universities per Country")
plt.xlabel("Country")
plt.ylabel("Number of Universities")
plt.show()
```



Step 8: Customizing the Workshop

You can experiment further by:

- Fetching data for specific regions.
- Analyzing other attributes such as university domains.
- Creating visualizations to compare universities in different regions.

For the followig you will need to get your API key from https://financeapi.net/ Examine the demo page and sample data to be returned

In [52]:
```
import json
import pandas as pd
import json
import requests
```

In [53]:
```
#get stock from the user
stock=input()
```

In [54]:
```
print("The Stock we will research is:" + stock)
```

```
The Stock we will research is:AAPL
```

In [55]:
```
url = "https://yfapi.net/v6/finance/quote"

querystring = {"symbols":stock}

headers = {
    'x-api-key': apikey
    }

response = requests.request("GET", url, headers=headers, params=querystring)

print(response.text)
```

{"quoteResponse":{"result":[{"language":"en-US","region":"US","quoteType":"EQUITY","typeDisp":"Equity","quoteSourceName":"Nasdaq Real Time Price","triggerable":true,"customPriceAlertConfidence":"HIGH","priceHint":2,"postMarketChangePercent":-0.120722555,"postMarketTime":1727475898,"postMarketPrice":227.515,"postMarketChange":-0.2749939,"regularMarketChange":0.269989,"regularMarketTime":1727467204,"regularMarketDayHigh":229.52,"regularMarketDayRange":"227.3 - 229.52","currency":"USD","regularMarketDayLow":227.3,"regularMarketVolume":33706549,"regularMarketPreviousClose":227.52,"bid":227.59,"ask":240.09,"bidSize":5,"askSize":1,"fullExchangeName":"NasdaqGS","financialCurrency":"USD","regularMarketOpen":228.43,"averageDailyVolume3Month":55583992,"averageDailyVolume10Day":76355650,"fiftyTwoWeekLowChange":63.70999,"fiftyTwoWeekLowChangePercent":0.38828614,"fiftyTwoWeekRange":"164.08 - 237.23","fiftyTwoWeekHighChange":-9.440002,"fiftyTwoWeekHighChangePercent":-0.039792616,"fiftyTwoWeekLow":164.08,"fiftyTwoWeekHigh":237.23,"fiftyTwoWeekChangePercent":32.88944,"dividendDate":1723680000,"earningsTimestamp":1722544200,"earningsTimestampStart":1730372340,"earningsTimestampEnd":1730721600,"earningsCallTimestampStart":1722546000,"earningsCallTimestampEnd":1722546000,"isEarningsDateEstimate":true,"trailingAnnualDividendRate":0.97,"trailingPE":34.724083,"dividendRate":1.0,"trailingAnnualDividendYield":0.0042633615,"dividendYield":0.44,"epsTrailingTwelveMonths":6.56,"epsForward":7.48,"epsCurrentYear":6.7,"priceEpsCurrentYear":33.99851,"sharesOutstanding":15204100096,"bookValue":4.382,"fiftyDayAverage":222.0196,"fiftyDayAverageChange":5.7703857,"fiftyDayAverageChangePercent":0.025990432,"twoHundredDayAverage":197.008,"twoHundredDayAverageChange":30.781998,"twoHundredDayAverageChangePercent":0.15624745,"marketCap":3463341932544,"forwardPE":30.453207,"priceToBook":51.983112,"sourceInterval":15,"exchangeDataDelayedBy":0,"averageAnalystRating":"2.0 - Buy","regularMarketChangePercent":0.11866606,"regularMarketPrice":227.79,"marketState":"POST","exchange":"NMS","shortName":"Apple Inc.","longName":"Apple Inc.","messageBoardId":"finmb_24937","exchangeTimezoneName":"America/New_York","exchangeTimezoneShortName":"EDT","gmtOffSetMilliseconds":-14400000,"market":"us_market","esgPopulated":false,"tradeable":false,"cryptoTradeable":false,"hasPrePostMarketData":true,"firstTradeDateMilliseconds":345479400000,"displayName":"Apple","symbol":"AAPL"}],"error":null}}

In [56]:
```
#print the company name and price
stock_json = response.json()
print(stock_json['quoteResponse']['result'][0]["longName"] + " Price:$" + st
```

```
Apple Inc. Price:$227.79
```

Ask the user for a list of stocks and pass that answer back to the user

In [57]:
```
stocks = input("Please enter a list of stocks separated by commas: ")

print(f"You entered: {stocks}")

stock_list = stocks.split(',')

for stock in stock_list:
    stock = stock.strip()
    params = {'symbols': stock}
    response = requests.get(url, headers=headers, params=params)

    if response.status_code == 200:
        stock_json = response.json()
        if stock_json['quoteResponse']['result']:
```

```
            print(stock_json['quoteResponse']['result'][0]["longName"] + " P
        else:
            print(f"No data found for {stock}")
    else:
        print(f"Failed to fetch data for {stock}")
```

```
You entered: AAPL,TSLA
Apple Inc. Price:$227.79
Tesla, Inc. Price:$260.46
```