



Estimated time needed: **30** minutes

Objectives

After completing this lab, you will be able to:

- Collect job data using Jobs API
- Store the collected data into an excel spreadsheet.

Collecting Job Data Using APIs

Note: Before starting with the assignment make sure to read all the instructions and then move ahead with the coding part.

Instructions

To run the actual lab, firstly you need to click on the [Jobs_API](#) notebook link. The file contains flask code which is required to run the Jobs API data.


Now, to run the code in the file that opens up follow the below steps.

Step1: Download the file.


Step2: Upload the file into your current Jupyter environment using the upload button in your Jupyter interface. Ensure that the file is in the same folder as your working .ipynb file.


Step 2: If working in a local Jupyter environment, use the "Upload" button in your Jupyter interface to upload the Jobs_API notebook into the same folder as your current .ipynb file.


Skills Network Labs




+


















Filter files by name








 / labs / DA0101EN /


Name	Last Modified
 clean_df.csv	4 months ago
 data-wran...	3 months ago
 explorator...	2 months ago
 model-dev...	2 months ago
 model-eva...	2 months ago
 review-intr...	3 months ago

















Step3: Open the Jobs_API notebook, and run all the cells to start the Flask application. Once the server is running, you can access the API from the URL provided in the notebook.

If you want to learn more about flask, which is optional, you can click on this link [here](#).

Once you run the flask code, you can start with your assignment.

Dataset Used in this Assignment

The dataset used in this lab comes from the following source: <https://www.kaggle.com/promptcloud/jobs-on-naukricom> under the under a **Public Domain license**.

Note: We are using a modified subset of that dataset for the lab, so to follow the lab instructions successfully please use the dataset provided with the lab, rather than the dataset from the original source.

The original dataset is a csv. We have converted the csv to json as per the requirement of the lab.

Warm-Up Exercise

Before you attempt the actual lab, here is a fully solved warmup exercise that will help you to learn how to access an API.

Using an API, let us find out who currently are on the International Space Station (ISS).

The API at <http://api.open-notify.org/astros.json> gives us the information of astronauts currently on ISS in json format.

You can read more about this API at <http://open-notify.org/Open-Notify-API/People-In-Space/>

```
In [11]: import requests # you need this module to make an API call
import pandas as pd
```

```
In [12]: api_url = "http://api.open-notify.org/astros.json" # this url gives use the astronaut data
```

```
In [13]: response = requests.get(api_url) # Call the API using the get method and store the
# output of the API call in a variable called response.
```

```
In [14]: if response.ok:           # if all is well() no errors, no network timeouts
        data = response.json()    # store the result in json format in a variable called data
                                     # the variable data is of type dictionary.
```

```
In [15]: print(data)    # print the data just to check the output or for debugging
```

```
{'people': [{'craft': 'ISS', 'name': 'Oleg Kononenko'}, {'craft': 'ISS', 'name': 'Nikolai Chub'}, {'craft': 'ISS', 'name': 'Tracy Caldwell Dyson'}, {'craft': 'ISS', 'name': 'Matthew Dominick'}, {'craft': 'ISS', 'name': 'Michael Barratt'}, {'craft': 'ISS', 'name': 'Jeanette Epps'}, {'craft': 'ISS', 'name': 'Alexander Grebenkin'}, {'craft': 'ISS', 'name': 'Butch Wilmore'}, {'craft': 'ISS', 'name': 'Sunita Williams'}, {'craft': 'Tiangong', 'name': 'Li Guangsu'}, {'craft': 'Tiangong', 'name': 'Li Cong'}, {'craft': 'Tiangong', 'name': 'Ye Guangfu'}], 'number': 12, 'message': 'success'}
```

Print the number of astronauts currently on ISS.

```
In [16]: print(data.get('number'))
```

12

Print the names of the astronauts currently on ISS.

```
In [17]: astronauts = data.get('people')
print("There are {} astronauts on ISS".format(len(astronauts)))
print("And their names are :")
for astronaut in astronauts:
    print(astronaut.get('name'))
```

There are 12 astronauts on ISS

And their names are :

Oleg Kononenko

Nikolai Chub

Tracy Caldwell Dyson

Matthew Dominick

Michael Barratt

Jeanette Epps

Alexander Grebenkin

Butch Wilmore

Sunita Williams

Li Guangsu

Li Cong

Ye Guangfu

Hope the warmup was helpful. Good luck with your next lab!

Lab: Collect Jobs Data using Jobs API

Objective: Determine the number of jobs currently open for various technologies and for various locations

Collect the number of job postings for the following locations using the API:

- Los Angeles
- New York
- San Francisco
- Washington DC
- Seattle
- Austin
- Detroit

```
In [18]: #Import required libraries
import pandas as pd
import json
```

[https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DA0321EN-](https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DA0321EN-SkillsNetwork/labs/module%201/Accessing%20Data%20Using%20APIs/jobs.json####)

[SkillsNetwork/labs/module%201/Accessing%20Data%20Using%20APIs/jobs.json####](https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DA0321EN-SkillsNetwork/labs/module%201/Accessing%20Data%20Using%20APIs/jobs.json####) Write a function to get the number of jobs for the Python technology.

Note: While using the lab you need to pass the **payload** information for the **params** attribute in the form of **key value** pairs.

Refer the ungraded **rest api lab** in the course **Python for Data Science, AI & Development** [link](#)

The keys in the json are

- Job Title
- Job Experience Required

- Key Skills
- Role Category
- Location
- Functional Area
- Industry
- Role

You can also view the json file contents from the following [json](#) URL.

Calling the function for Python and checking if it works.

```
In [19]: api_url = "http://127.0.0.1:5000/data"
def get_number_of_jobs_T(technology):
    number_of_jobs = 0
    payload = {"Key Skills": technology}
    r = requests.get(api_url, params=payload)
    if r.ok:
        data = r.json()
        number_of_jobs += len(data)
    return technology, number_of_jobs
```

```
In [20]: get_number_of_jobs_T("Python")
```

```
Out[20]: ('Python', 1173)
```

Write a function to find number of jobs in US for a location of your choice

```
In [21]: def get_number_of_jobs_L(location):
    number_of_jobs=0
    payload={'Location':location}
    r=requests.get(api_url,payload)
    if r.ok:
        data=r.json()
```

```
number_of_jobs+=len(data)
return location,number_of_jobs
```

Call the function for Los Angeles and check if it is working.

```
In [22]: get_number_of_jobs_L("Los Angeles")
```

```
Out[22]: ('Los Angeles', 640)
```

Store the results in an excel file

Call the API for all the given technologies above and write the results in an excel spreadsheet.

If you do not know how create excel file using python, double click here for **hints**.

Create a python list of all technologies for which you need to find the number of jobs postings.

```
In [23]: #list of technologies
technologies_list = ["web analytics", "Oracle IDAM", "digital marketing", "networking", "JavaScript", "Python", "Scala"]
```

Import libraries required to create excel spreadsheet

```
In [24]: # your code goes here
!pip install openpyxl
import pandas as pd
from openpyxl import Workbook
```

Collecting openpyxl

Downloading openpyxl-3.1.3-py2.py3-none-any.whl (251 kB)

251.3/251.3 kB 23.9 MB/s eta 0:00:00

Collecting et-xmlfile (from openpyxl)

Downloading et_xmlfile-1.1.0-py3-none-any.whl (4.7 kB)

Installing collected packages: et-xmlfile, openpyxl

Successfully installed et-xmlfile-1.1.0 openpyxl-3.1.3

Create a workbook and select the active worksheet

```
In [25]: # your code goes here
wb = Workbook() # Create a new workbook
```



```
ws = wb.active          # Select the active worksheet
```

Find the number of jobs postings for each of the technology in the above list. Write the technology name and the number of jobs postings into the excel spreadsheet.

```
In [26]: #your code goes here
# Function to count jobs based on key skill
ws.append(['name of technology','number of job openings'])
for job in technologies_list:
    value = list(get_number_of_jobs_T(job))
    ws.append(value)
```

Save into an excel spreadsheet named **job-postings.xlsx**.

```
In [27]: # Convert to DataFrame and write to Excel
wb.save("job-postings.xlsx")
import pandas as pd
df=pd.read_excel("job-postings.xlsx")
df.head(7)
```

```
Out[27]:
```

	name of technology	number of job openings
0	web analytics	0
1	Oracle IDAM	0
2	digital marketing	0
3	networking	0
4	JavaScript	355
5	Python	1173
6	Scala	33

In the similar way, you can try for below given technologies and results can be stored in an excel sheet.

Collect the number of job postings for the following languages using the API:

- C
- C#
- C++
- Java
- JavaScript
- Python
- Scala
- Oracle
- SQL Server
- MySQL Server
- PostgreSQL
- MongoDB

```
In [28]: technologies=['C','C#','C++','Java','JavaScript','Python','Scala','Oracle','SQL Server','MySQL Server','PostgreSQL',
from openpyxl import Workbook
wb=Workbook()
ws2 = wb.create_sheet("Sheet_2")
ws2.title = "technologies"
ws2.append(['name of technology','number of job openings'])
for job in technologies:
    values=list(get_number_of_jobs_T(job))
    ws2.append(values)
wb.save("job-postings.xlsx")
import pandas as pd
df=pd.read_excel("job-postings.xlsx",sheet_name= "technologies")
df.head(12)
```

Out[28]:

	name of technology	number of job openings
0	C	13498
1	C#	333
2	C++	305
3	Java	2609
4	JavaScript	355
5	Python	1173
6	Scala	33
7	Oracle	784
8	SQL Server	250
9	MySQL Server	0
10	PostgreSQL	10
11	MongoDB	174

```
In [30]: import pandas as pd
import matplotlib.pyplot as plt

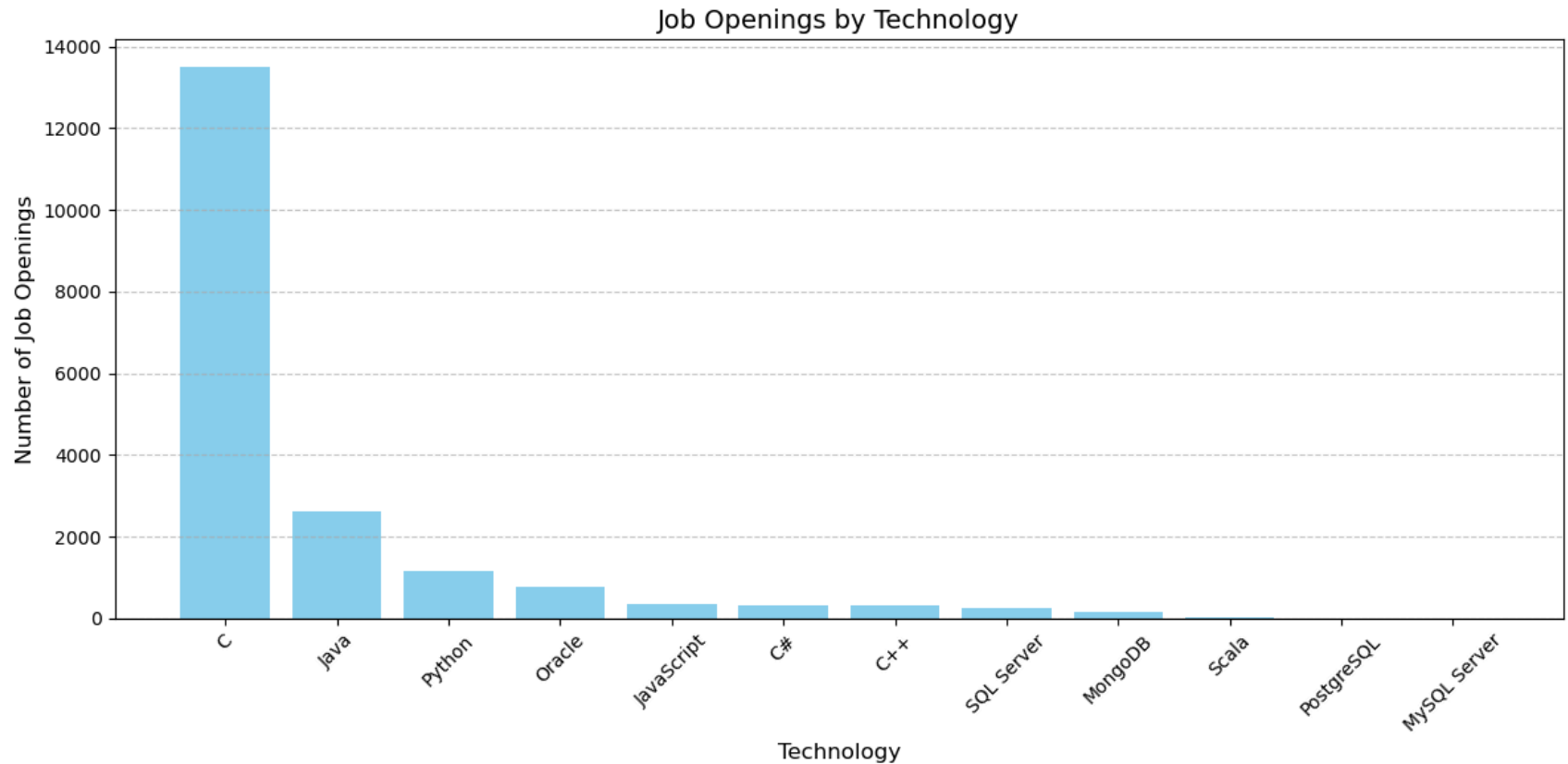
# Load your data
df = pd.read_excel("job-postings.xlsx", sheet_name="technologies")

# Sort the data by number of job openings in descending order
df_sorted = df.sort_values(by="number of job openings", ascending=False)

# Plotting the bar chart
plt.figure(figsize=(12, 6))
plt.bar(df_sorted["name of technology"], df_sorted["number of job openings"], color='skyblue')
plt.xlabel("Technology", fontsize=12)
plt.ylabel("Number of Job Openings", fontsize=12)
plt.title("Job Openings by Technology", fontsize=14)
plt.xticks(rotation=45)
plt.tight_layout()
```

```
plt.grid(axis='y', linestyle='--', alpha=0.7)

# Display the chart
plt.show()
```



```
In [22]: import pandas as pd
import matplotlib.pyplot as plt

# Define the data directly in Python
data = {
    "name of technology": [
        "C", "C#", "C++", "Java", "JavaScript", "Python",
        "Scala", "Oracle", "SQL Server", "MySQL Server",
        "PostgreSQL", "MongoDB"
    ],
    "number of job openings": [
```

```

        13498, 333, 305, 2609, 355, 1173,
        33, 784, 250, 0, 10, 174
    ]
}

# Create a DataFrame from the dictionary
df = pd.DataFrame(data)

# Sort the DataFrame in descending order by job openings
df_sorted = df.sort_values(by="number of job openings", ascending=False)

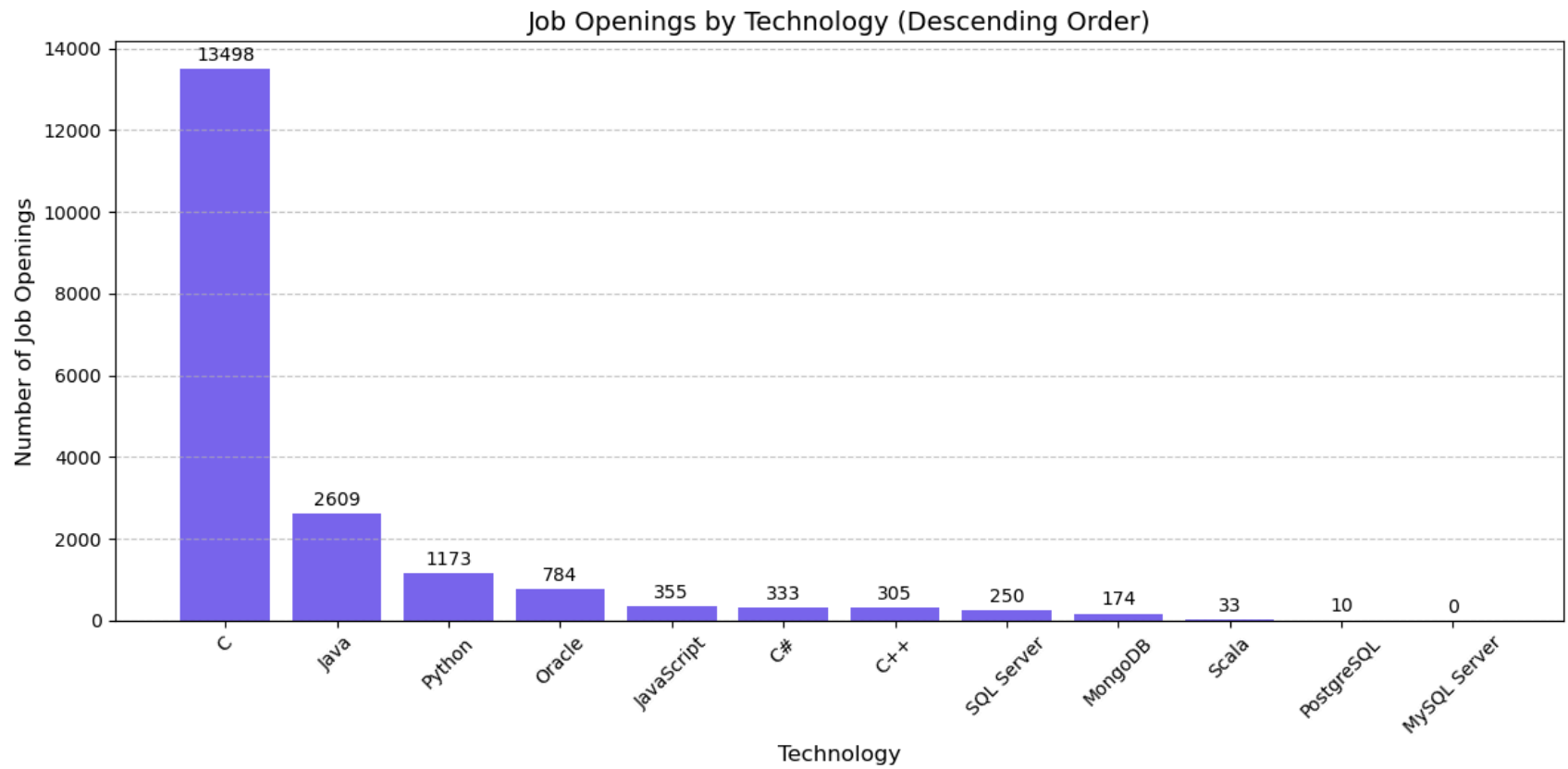
# Plot the bar chart
plt.figure(figsize=(12, 6))
bars = plt.bar(df_sorted["name of technology"], df_sorted["number of job openings"], color='mediumslateblue')

# Add value labels on top of each bar
for bar in bars:
    height = bar.get_height()
    plt.text(bar.get_x() + bar.get_width()/2, height + 100, str(height), ha='center', va='bottom', fontsize=10)

plt.xlabel("Technology", fontsize=12)
plt.ylabel("Number of Job Openings", fontsize=12)
plt.title("Job Openings by Technology (Descending Order)", fontsize=14)
plt.xticks(rotation=45)
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.tight_layout()

# Show the chart
plt.show()

```



Authors

Ayushi Jain

Other Contributors

Rav Ahuja

Lakshmi Holla

Malika

Copyright © IBM Corporation.

<!--## Change Log

| <!-- Date (YYYY-MM-DD) Version Changed By Change Description ----- ----- ----- ----- |
|---|
| ----- 2022-01-19 0.3 Lakshmi Holla Added changes in the markdown 2021-06-25 0.2 Malika Updated GitHub |
| job json link 2020-10-17 0.1 Ramesh Sannareddy Created initial version of the lab --!> |