

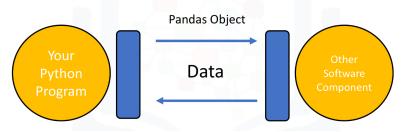
import pandas as pd import matplotlib.pyplot as plt

You create a dictionary, this is just data.

[]: dict\_={'a':[11,21,31],'b':[12,22,32]}

When you create a Pandas object with the Dataframe constructor in API lingo, this is an "instance". The data in the dictionary is passed along to the pandas API. You then use the dataframe to communicate with the API.

[ ]: df=pd.DataFrame(dict\_)
type(df)



When you call the method head the dataframe communicates with the API displaying the first few rows of the dataframe.

[ ]: df.head()

When you call the method mean, the API will calculate the mean and return the value.

[ ]: df.mean()

## **REST APIs**

Rest API's function by sending a request, the request is communicated via HTTP message. The HTTP message usually contains a JSON file. This contains instructions for what operation we would like the service or resource to perform. In a similar manner, API returns a response, via an HTTP message, this response is usually contained within a JSON.

In this lab, we will use the NBA API to determine how well the Golden State Warriors performed against the Toronto Raptors. We will use the API do the determined number of points the

Golden State Warriors won or lost by for each game. So if the value is three, the Golden State Warriors won by three points. Similarly it the Golden State Warriors lost by two points the result will be negative two. The API is relatively will handle a lot of the details such a Endpoints and Authentication

In the nba api to make a request for a specific team, it's quite simple, we don't require a JSON all we require is an id. This information is stored locally in the API we import the module teams

[ ]: from nba\_api.stats.static import teams import matplotlib.pyplot as plt

[ ]: #https://pypi.org/project/nba-api/

The method get\_teams() returns a list of dictionaries the dictionary key id has a unique identifier for each team as a value

[ ]: nba\_teams = teams.get\_teams()

The dictionary key id has a unique identifier for each team as a value, let's look at the first three elements of the list:

[ ]: nba\_teams[0:3]

To make things easier, we can convert the dictionary to a table. First, we use the function one dict, to create a dictionary. We use the common keys for each team as the keys, the value is a list; each element of the list corresponds to the values for each team. We then convert the dictionary to a dataframe, each row contains the information for a different team.

[]: dict\_nba\_team=one\_dict(nba\_teams)
 df\_teams=pd.DataFrame(dict\_nba\_team)
 df teams.head()

Will use the team's nickname to find the unique id, we can see the row that contains the warriors by using the column nickname as follows:

[ ]: df\_warriors=df\_teams[df\_teams['nickname']=='Warriors']
df\_warriors

we can use the following line of code to access the first column of the dataframe:

[]: id\_warriors=df\_warriors[['id']].values[0][0]
#we\_now\_have\_an\_integer\_that\_can\_be\_used\_\_\_to\_request\_the\_Warriors\_information\_

The function "League Game Finder " will make an API call, its in the module stats.endpoints

[ ]: from nba\_api.stats.endpoints import leaguegamefinder

The parameter team\_id\_nullable is the unique ID for the warriors. Under the hood, the NBA API is making a HTTP request. The information requested is provided and is transmitted via an HTTP response this is assigned to the object gamefinder.

[]: # Since https://stats.nba.com does lot allow api calls from Cloud IPs and Skills Network labs uses a Cloud IP.
# The following code is comment out, you can run it on jupyter labs on your own computer..
# gamefinder = leaguegamefinder.leagueGamefinder(team\_id\_nullable=id\_warriors)

we can see the json file by running the following line of code

[]: # Since https://stats.nba.com does lot allow api calls from Cloud IPs and Skill<u>s Network Labs.uses a Cloud IP.</u> # The following code is comment out, you can run it on jupyter Labs on your own.computer. # gamefinder.get\_ison()

The game finder object has a method <code>get\_data\_frames()</code>, that returns a dataframe. If we view the dataframe, we can see it contains information about all the games the Warriors played.

The <code>PLUS\_MINUS</code> column contains information on the score, if the value is negative the Warriors lost by that many points, if the value is positive, the warriors one by that amount of points. The column <code>MATCHUP</code> had the team the Warriors were playing, GSW stands for Golden State Warriors and TOR means Toronto Raptors; <code>vs</code> signifies it was a home game and the <code>@ symbol</code> means an away game.

[]: # Since https://stats.nba.com does lot allow api calls from Cloud IPs and Skill<u>s Network Labs uses a Cloud IP</u>.

# The following code is comment out, you can run it on jupyter Labs on your own computer.

# games = gamefinder.get\_data\_frames()[0]

# games.head()

you can download the dataframe from the API call for Golden State and run the rest like a video.

[]: |\_wget\_https://s2-api.us-geo.objectstorage.softlayer.net/cf-courses-data/CognitiveClass/PY0101EN/Chapter%205/Labs/Golden\_State.pkl

[ ]: file\_name = "Golden\_State.pk1"
 games = pd.read\_pickle(file\_name)
 games.head()

We can create two dataframes, one for the games that the Warriors faced the raptors at home and the second for away games.

[]: games\_home=games\_[games\_['MATCHUP']=='GSW\_vs.\_TOR']
games\_away=games\_[games\_['MATCHUP']=='GSW\_@\_TOR']

We can calculate the mean for the column PLUS\_MINUS for the dataframes games\_home and games\_away :

[ ]: games\_home.mean()['PLUS\_MINUS']

[ ]: games\_away.mean()['PLUS\_MINUS']

We can plot out the PLUS MINUS column for for the dataframes games\_home and games\_away . We see the warriors played better at home.

[]: fig, ax = plt.subplots()

games\_away.plot(x='GAME\_DATE',y='PLUS\_MINUS',\_ax=ax)
games\_home.plot(x='GAME\_DATE',y='PLUS\_MINUS',\_ax=ax)
ax.legend(["away", "home"])
plt.show()



Learn

Get started or get better with built-in learning.



Create
Use the best of open source tooling with IBM innovation



Work smarter using community, work faster with yo

Sign Up For a Free Trial

## Joseph Santarcangelo

Joseph Santarcangelo has a PhD in Electrical Engineering, his research focused on using machine learning, signal processing, and computer vision to determine how videos impact human cognition. Joseph has been working for IBM since he completed his PhD.

## **Change Log**

Date (YYYY-MM-DD)	Version	Changed By	Change Description
2020-09-09	2.1	Malika Singla	Spell Check
2020-08-26	2.0	Lavanya	Moved lab to course repo in GitLab

© IBM Corporation 2020. All rights reserved.

**Did you know?** IBM Watson Studio lets you build and deploy an Al solution, using the best of open source and IBM software and giving your team a single environment to work in. Learn more here.









Mode: Command 😵 Ln 1, Col 1 English (American) PY0101EN-5.1\_Intro\_API.ipynb