PYTHON FOR DATA SCIENCE APIS & DATA COLLECTION

Ana-María Dobre
based on EDX Course
SEP 2023

1. SIMPLE APIs

An API lets two pieces of software talk to each other. Just like a function, you don't have to know how the API works, only its inputs and outputs
REST API = an essential type of API, that allows you to access resources via the internet

GOALS: In this lab, review the Pandas Library in the context of an API; also review a basic REST API

Pandas is actually a set of SW components, much of which is not even written in Python: (1)
Create a dictionary: (2) # just data

Note: 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 df to communicate with the API.

When you call the method **head,** the df communicates with the API displaying the first few rows of the df: (4)
When you call the method **mean**, the API will calculate the mean and return the value: (5)

Rest APIs:
- They function by sending a **request**
  - The request is communicated via HTTP message
  - The HTTP message usually contains a JSON file
  - The JSON contains instructions for what operation we would like the service or resource to perform.
- Similarly, the 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 (GSW) performed against the Toronto Raptors (TR).

- Use the API to get the number of points the GSW won or lost by for each game.
E.g. if the value is 3, the GSW won by 3 points. Similarly it the GSW lost by two points, the result will be -2.
- The API will handle a lot of the details, such a Endpoints and Authentication.

It's quite simple to use the **nba api** to make a request for a specific team. We don't require a JSON, all we require is an id. This information is stored locally in the API.

First, we import the module teams: (6)

```
(1)
import pandas as pd
import matplotlib.pyplot as plt

(2)
dict_={'a':[11,21,31],'b':[12,22,32]}

(3)
df=pd.DataFrame(dict_)
type(df)
# pandas.core.frame.DataFrame

(4)
df.head()

(5)
df.mean()

(6)
We import the module teams
!pip install nba_api

from nba_api.stats.static import teams
import matplotlib.pyplot as plt

#https://pypi.org/project/nba-api/
```

2. SIMPLE APIs (Part 2)

The method get_teams() returns a list of dictionaries: (7)
Get the first 3 elements of the list

To make things easier, we can convert the dictionary to a table.

First, use the function one_dict, to create a dictionary # first, define it
Then, convert the dictionary to a dataframe, each row contains the information for a different team.

Use the team's nickname to find the unique id
// we can see the row that contains the warriors by using the column nickname: (9)

Access the first column of the dataframe: (10)

The function "League Game Finder " will make an API call, it's in the module stats.endpoints: (11)

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 game finder.
Since https://stats.nba.com does not allow api calls from Cloud IPs and Skills Network Labs uses a Cloud IP.
The following code is commented out, you can run it on jupyter labs on your own computer: (12)

Since https://stats.nba.com does not allow api calls from Cloud IPs and Skills Network Labs uses a Cloud IP.
The following code is commented out, you can run it on jupyter labs on your own computer: (13)

The game finder object has a method get_data_frames(), that returns a df: (14)
If we view the df, we can see it contains information about all the games the Warriors played.
- The PLUS_MINUS 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 won by that amount of points.
- The column MATCHUP has the team the Warriors were playing, GSW stands for Golden State Warriors and TOR means Toronto Raptors. vs signifies it was a home game and the @ symbol means an away game.

you can download the dataframe from the API call for Golden State and run the rest like a video: (15)

We can create two dataframes, one for the games that the Warriors faced the raptors at home, and the second for away games: (16)

We can calculate the mean for the column PLUS_MINUS for the dataframes games_home and games_away: (17)

We can plot out the PLUS MINUS column for the dataframes games_home and games_away. We see the warriors played better at home. (18)

Calculate the mean for the column PTS for the dataframes games_home and games_away: (19)

```
(7)
nba_teams = teams.get_teams()
nba_teams[0:3]

(8)
def one_dict(list_dict):
    keys=list_dict[0].keys()
    out_dict={key:[] for key in keys}
    for dict_ in list_dict:
        for key, value in dict_.items():
            out_dict[key].append(value)
    return out_dict

dict_nba_team=one_dict(nba_teams)
df_teams=pd.DataFrame(dict_nba_team)
df_teams.head()

(9)
df_warriors=df_teams[df_teams['nickname']=='Warriors']
df_warriors

(10)
id_warriors=df_warriors[['id']].values[0][0]
# we now have an integer that can be used to request the Warriors information

(11) from nba_api.stats.endpoints import leaguegamefinder

(12)
# gamefinder =
leaguegamefinder.LeagueGameFinder(team_id_nullable=id_warriors)

(13) # gamefinder.get_json()

(14)  # Since https://stats.nba.com does not 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.
# games = gamefinder.get_data_frames()[0]
# games.head()

(15)
import requests
filename = "https://s3-api.us-geo.objectstorage.softlayer.net/cf-courses-
data/CognitiveClass/PY0101EN/Chapter%205/Labs/Golden_State.pkl"

def download(url, filename):
    response = requests.get(url)
    if response.status_code == 200:
        with open(filename, "wb") as f:
            f.write(response.content)
download(filename, "Golden_State.pkl")

file_name = "Golden_State.pkl"
games = pd.read_pickle(file_name)
games.head()

(16)
games_home=games[games['MATCHUP']=='GSW vs. TOR']
games_away=games[games['MATCHUP']=='GSW @ TOR']

(17)
games_home['PLUS_MINUS'].mean()
games_away['PLUS_MINUS'].mean()

(18)
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()

(19)  games_home['PTS'].mean()
games_away['PTS'].mean()
```