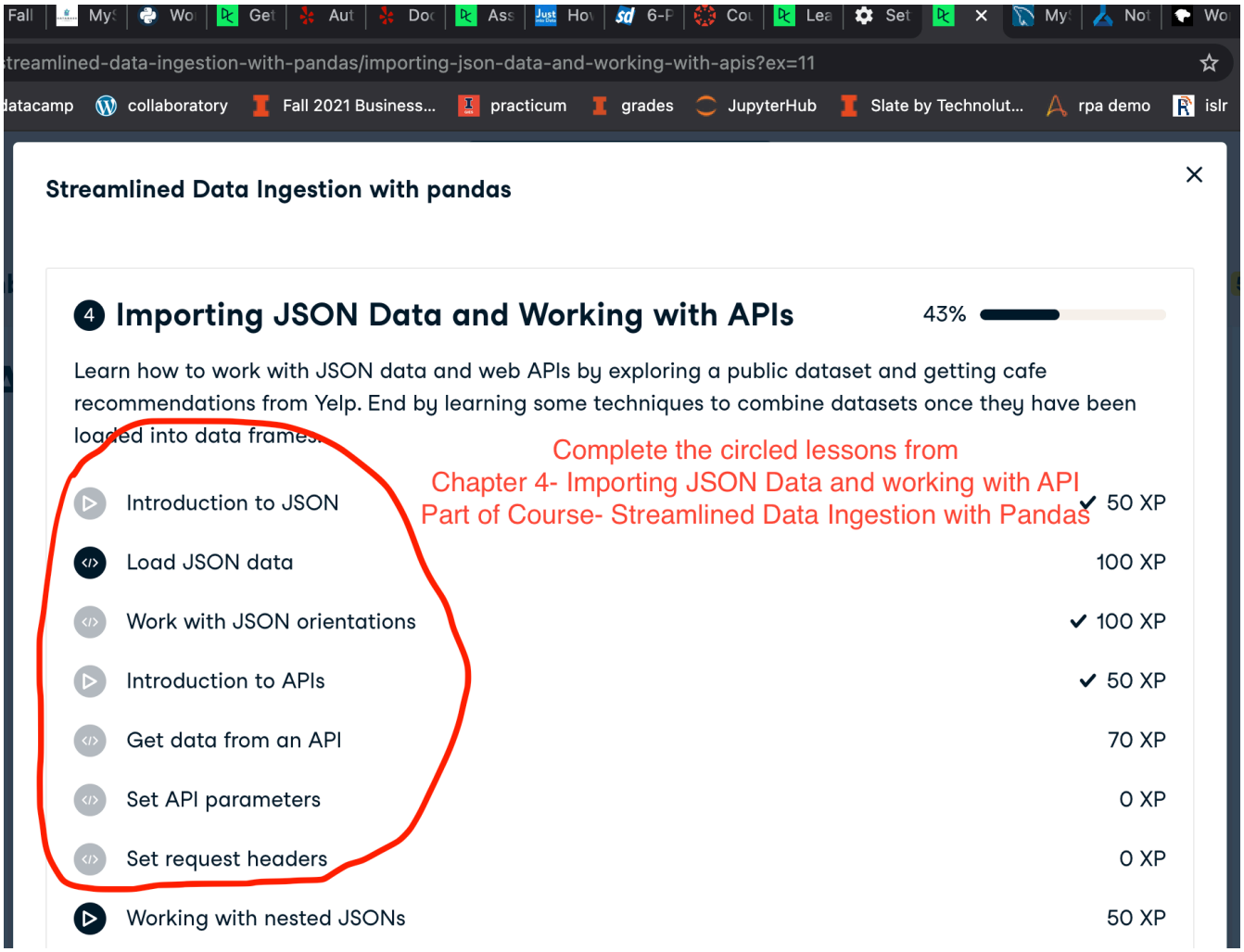
**This notebook gives you starter code to use the Yelp FUSION API to extract data about businesses, reviews, users and more. You should review the datacamp chapter as indicated in the image below, before starting this exercise. This counts for some class participation credit.

Create a copy of this notebook and follow the steps below



To understand what is an API, please refer to https://www.mulesoft.com/resources/api/what-is-an-api

For an interesting read on how API calls can be used to scrape data from the web (includes basic knowledge of JSON and Python) refer to https://towardsdatascience.com/json-and-apis-with-python-fba329ef6ef0

Let's start with importing some libraries to extract data from Yelp Fusion API

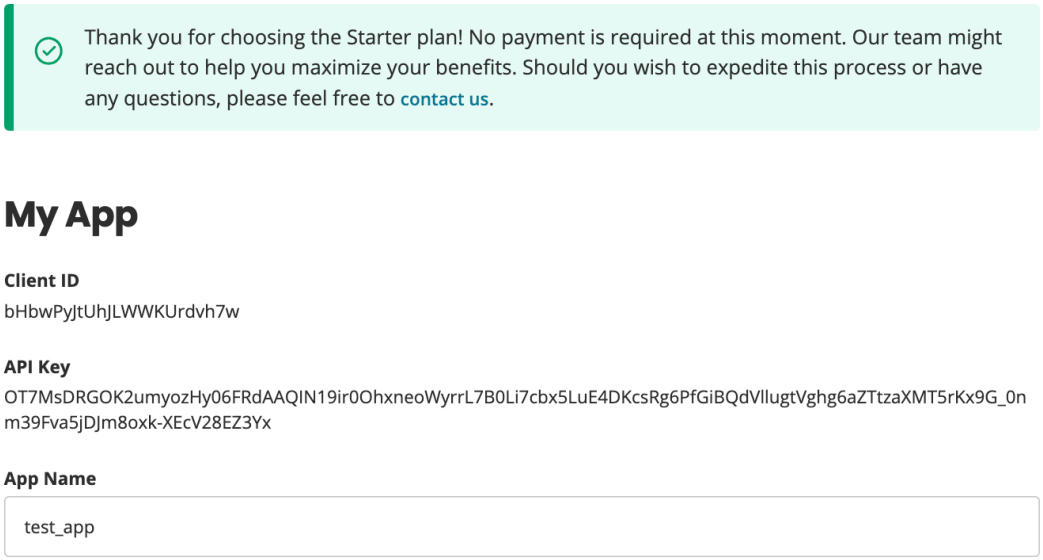
ChatGPT

```
# standard Python library for handling HTTP is reque
import requests
```

```
#import pandas
import pandas as pd
```

Make sure you have followed the steps given on canvas to generte the Yelp API key

Once you see the key you need to copy this for the next code cell.Do not try and copy the key shown here




```
# paste your api key in this variable

api_key = 'Your-api-key-here'
```

In order to **GET** data from the API, we **request** it at a particular **search_api_url**. Before using the **GET** method, let's collect all the information into separate variables that we will combine and send to the API. So a call to an API must have the following information

1. A URL from where it gets information.
2. There are separate endpoints for different kinds of information. This is usually just a part of the URL
3. The API Key (usually transmitted in the header)
4. The parameters for the search(params)

You can read about the different ways you can search and get information at https://docs.developer.yelp.com/docs/fusion-intro

Lets decide a business question. Lets try and a list of 10 businesses with the term "coffee" located in '61820', the main zip code for Champaign.

```
# headers contain the api key defined in the previous cell
headers = {'Authorization': 'Bearer {}'.format(api_key)}

# the api endpoint url. We are working with the 'businesses- Search' endpoint. There are other Business endpoints as we
search_api_url = 'https://api.yelp.com/v3/businesses/search'

# Lets search for ten businesses with the term 'coffee'located in zip code '61820'
#Review all parameters and documentation at https://docs.developer.yelp.com/reference/v3_business_search
params = {'term': 'coffee',
          'location': '61820',
          'limit': 10}
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-3-f4251c19dc25> in <cell line: 2>()
      1 # headers contain the api key defined in the previous cell
----> 2 headers = {'Authorization': 'Bearer {}'.format(api_key)}
      3
      4 # the api endpoint url. We are working with the 'businesses- Search' endpoint. There are other Business
endpoints as well
      5 search_api_url = 'https://api.yelp.com/v3/businesses/search'

NameError: name 'api_key' is not defined
```

```
# we can feed these variables into the "get"function
# we also set timeout = 5 to stop Requests from waiting for a response after 5 seconds.
response = requests.get(search_api_url, headers=headers, params=params, timeout=5)

# extract JSON data from the response and print it .
# Read what the response.json() function does at https://www.geeksforgeeks.org/response-json-python-requests/#
data = response.json()
# when the data prints out below it  pay attention to the reference you have to use to access the data
# always think about the data type of the output. In this case you are getting a python 'dictionary'
print(data)

# Load data to a data frame. Note that the reference to 'businesses' is used as that is the 'Key'
# The function below picks the 'values' referenced by the 'key' - 'businesses' and assigns to df

#df = pd.DataFrame(data['businesses'])

# display the top rows. Default value is 5. Lets view all 50.
# Documentation for attributes available at https://docs.developer.yelp.com/reference/v3_business_search
# Spend some time understanding all the attributes.
#df.head(50)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-2-d69fdaa1b22b> in <cell line: 3>()
      1 # we can feed these variables into the "get"function
      2 # we also set timeout = 5 to stop Requests from waiting for a response after 5 seconds.
----> 3 response = requests.get(search_api_url, headers=headers, params=params, timeout=5)
      4
      5 # extract JSON data from the response and print it .

NameError: name 'search_api_url' is not defined
```

Double-click (or enter) to edit

For the lab you need to do the following

1. Try out the Businesses Search Endpoint with three different types of queries. Change the 'params' and write the code in three separate cells. Convert the JSON output to Dataframes and show the top 5 rows.
2. For every query that you write, ensure you have a new variable for params i.e. params1, params2, params3, and so on. This would also require response variables to be different, to store the responses of each of the params, such as response1, response2, response3, and so on. Similarly, ensure you have different queries stored in different dataframes such as df1, df2, df3 and so on. This is to ensure that each of your queries can be identified uniquely using a combination of responses and params.
3. Try out one more endpoint(besides 'Businesses-Search'but within 'Business') from the list at https://docs.developer.yelp.com/reference and create one query . Convert the JSON output to Dataframes and show the top 5 rows. 4.Using the Yelp API's Transaction Search endpoint (https://api.yelp.com/v3/transactions/delivery/search), make a request to find businesses that support food delivery in a specified geographic area. You can either use a location string (e.g., "New York City") or provide latitude and longitude coordinates for the search. Customize the query by including a term (e.g., "pizza" or "burgers") and optionally add a price filter (e.g., "1, 2" for $ or $$). After making the API call, convert the JSON response into a pandas DataFrame and display the top 5 rows of your results
4. Submit in the class participation assignment.

New Section

For Advanced users (not needed for participation credit, but you can do this one query, rather than the scope above)

1. Create a query where you need to get data from two end points and combine the results, using a join. You can do the join using python/pandas. One example could be to get a list of top 5 bakeries in a particular zip code, and list the 3 highest rating reviews and the 3 lowest rating reviews.