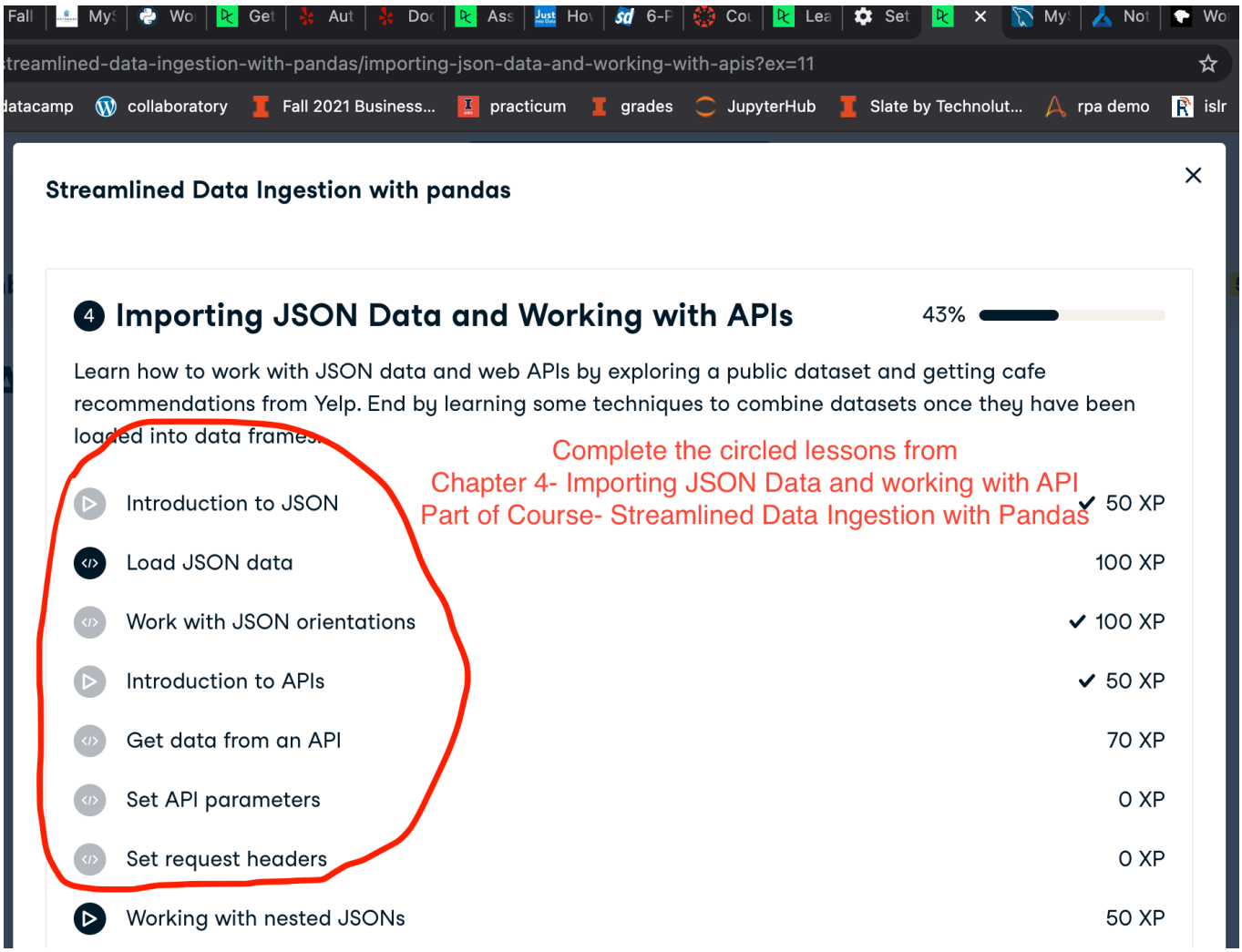
**This notebook gives you starter code to use the Yelp FUSION API to extract data about businesses, reviews, users and more. You should review the datacamp chapter as indicated in the image below, before starting this exercise. This counts for some class participation credit.

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

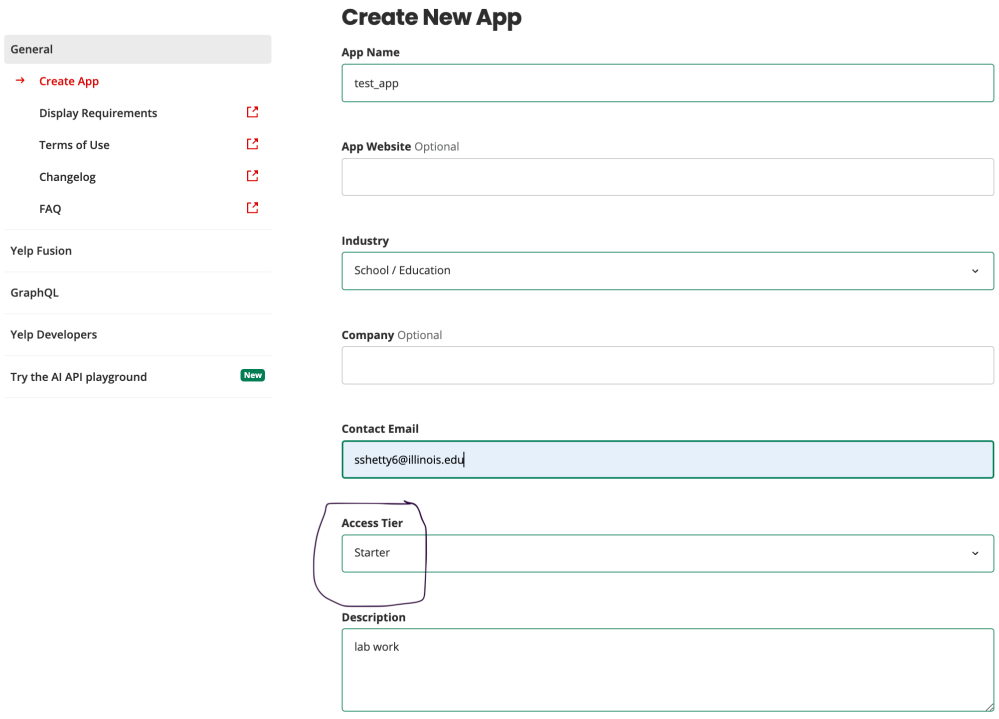
For an interesting read on how API calls can be used to scrape data from the web (includes basic knowledge of JSON and Python) refer to https://towardsdatascience.com/json-and-apis-with-python-fba329ef6ef0

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

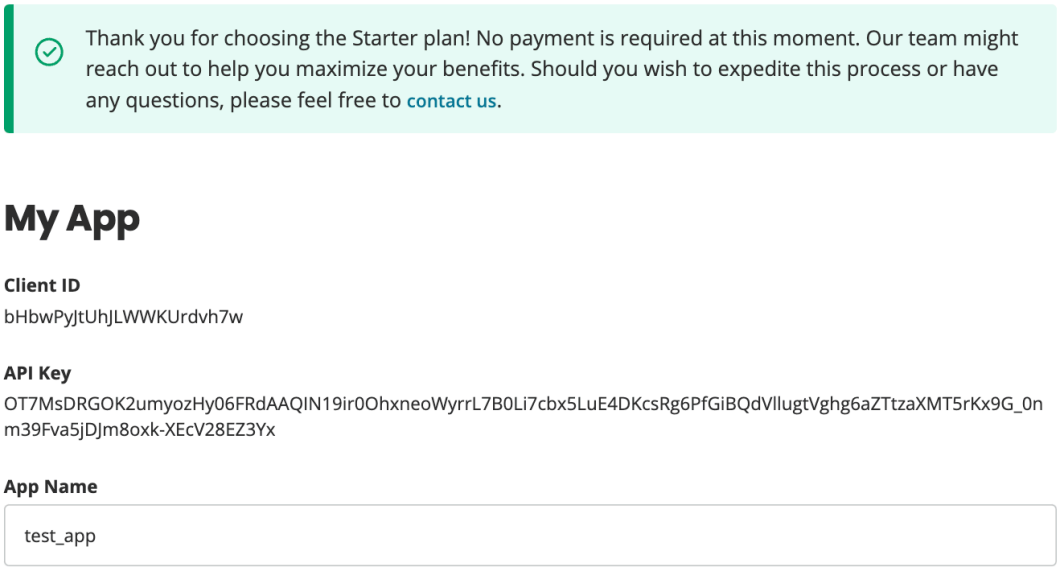
```
# standard Python library for handling HTTP is requests
import requests

#import pandas
import pandas as pd
```

Go to https://www.yelp.com/login?return_url=%2Fdevelopers%2Fv3%2Fmanage_app and Log In (You can use the 'Continue with Google' option), Put in your app details (choose "Starter" in the Access Tier)



After creating the App, you will get your personal API 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 = 'nIFqVG7vcoZTbq52XeHblZsaMlOlbV504-5tMXs-H2nuQ_ggD0ghWVn0cagQoE-VDC-oJ-BDivtZHKxV_6MUrjiYSw3TtTntVLVZGlnLrgXG3SOkqPW5-
```

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 well
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}


# 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'])
df.head(5)

# 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)
```

{'businesses': [{'id': 'MfJSKP5HRllfts13MiKs1g', 'alias': 'flying-machine-avionics-champaign', 'name': 'Flying Machine Avion

	id	alias	name	image_url	is_closed	url	review_cou
0	MfJSKP5HRllfts13MiKs1g	flying-machine-avionics-champaign	Flying Machine Avionics	https://s3-media4.fl.yelpcdn.com/bphoto/Hxt-FV...	False	https://www.yelp.com/biz/flying-machine-avioni...	
1	B9W60PTxn7XOZfk4BiTvyw	brewlab-coffee-champaign	BrewLab Coffee	https://s3-media3.fl.yelpcdn.com/bphoto/0uSIOy...	False	https://www.yelp.com/biz/brewlab-coffee-champa...	
2	dbCj5teyLVNxt605TlpPAQ	yummy-future-champaign-3	Yummy Future	https://s3-media3.fl.yelpcdn.com/bphoto/fJG0dI...	False	https://www.yelp.com/biz/yummy-future-champaig...	
3	gjDtBo28Qn4qJ5RxvKyW0Q	bakelab-urbana-2	Bakelab	https://s3-media2.fl.yelpcdn.com/bphoto/G7v161...	False	https://www.yelp.com/biz/bakelab-urbana-2?adju...	
4	4LuF3-1zxrjdEBbL7Vln7A	7-brew-coffee-champaign	7 Brew Coffee	https://s3-media1.fl.yelpcdn.com/bphoto/ec7F1z...	False	https://www.yelp.com/biz/7-brew-coffee-champai...	

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. Try out one new endpoint from https://docs.developer.yelp.com/reference/v3_business_reviews and write one query using a business id from the previous queries to find reviews for that business. Convert the JSON output to Dataframes and show the output
5. Submit in the class participation assignment.

```
params2 = {'term': 'rice',
          'location': '61801',
          'limit': 10}

response2 = requests.get(search_api_url, headers=headers, params=params2, timeout=5)
data2 = response2.json()
df2 = pd.DataFrame(data2['businesses'])
df2.head(5)
```

	id	alias	name	image_url	is_closed	url	review_
0	r0B-DJAA_XX2Dq_cwbPcaQ	sakura-japanese-restaurant-urbana	Sakura Japanese Restaurant	https://s3-media4.fl.yelpcdn.com/bphoto/Y2O9z_...	False	https://www.yelp.com/biz/sakura-japanese-resta...	
1	zyS0TeBajyPbXCb1zlk1cw	yatai-champaign	Yatai	https://s3-media1.fl.yelpcdn.com/bphoto/0I3QnW...	False	https://www.yelp.com/biz/yatai-champaign?adjus...	
2	QdganL5AW9E_fNoRVroLig	golden-wok-champaign	Golden Wok	https://s3-media3.fl.yelpcdn.com/bphoto/Q0AVTE...	False	https://www.yelp.com/biz/golden-wok-champaign?...	
3	5_cn7M-9swHrFbnqZtYGBQ	bangkok-thai-and-pho-911-champaign	Bangkok Thai and Pho 911	https://s3-media1.fl.yelpcdn.com/bphoto/AuYu7_...	False	https://www.yelp.com/biz/bangkok-thai-and-pho-...	
4	D1vTUM0rBlcU39A9emQW8Q	bab-plus-korean-restaurant-urbana	Bab Plus Korean Restaurant	https://s3-media2.fl.yelpcdn.com/bphoto/-CZMFW...	False	https://www.yelp.com/biz/bab-plus-korean-resta...	

```
params3 = {'term': 'sushi',
          'location': 'Waxahachie',
          'limit': 10}

response3 = requests.get(search_api_url, headers=headers, params=params3, timeout=5)
data3 = response3.json()
df3 = pd.DataFrame(data3['businesses'])
df3.head(5)
```

	id	alias	name	image_url	is_closed	url	review_c
0	gOqiD1c1F7UwXaO5-thekA	ohana-waxahachie	Ohana	https://s3-media4.fl.yelpcdn.com/bphoto/p0isMP...	False	https://www.yelp.com/biz/ohana-waxahachie?adju...	
1	EhvVjQ71DJ_ckk6Cg2sGoQ	osuba-hibachi-sushi-and-bar-waxahachie	Osuba Hibachi Sushi and Bar	https://s3-media1.fl.yelpcdn.com/bphoto/H2kea4...	False	https://www.yelp.com/biz/osuba-hibachi-sushi-a...	
2	oD9hFDXqxRF21sXnXqtY2Q	hibachio-waxahachie	Hibachio	https://s3-media3.fl.yelpcdn.com/bphoto/hOGh72...	False	https://www.yelp.com/biz/hibachio-waxahachie?a...	
3	l5ttzdCBHVb-TgRuI0VzZg	asian-king-buffet-waxahachie	Asian King Buffet	https://s3-media2.fl.yelpcdn.com/bphoto/GR9dFz...	False	https://www.yelp.com/biz/asian-king-buffet-wax...	
4	44z2ya0e52DOBRFytyUZbw	adoriana-sushi-and-asian-grill-midlothian	Adoriana Sushi and Asian Grill	https://s3-media3.fl.yelpcdn.com/bphoto/p8U8Xm...	False	https://www.yelp.com/biz/adoriana-sushi-and-as...	

```
params4 = {'term': 'pounded yam',
          'location': 'Chicago',
          'limit': 10}

response4 = requests.get(search_api_url, headers=headers, params=params4, timeout=5)
data4 = response4.json()
df4 = pd.DataFrame(data4['businesses'])
df4.head(5)
```

	id	alias	name	image_url	is_closed	url	review_c
0	N7R0sZrXWjH1T-x1iti-sQ	nayos-african-cuisine-chicago	Nayo's African Cuisine	https://s3-media4.fl.yelpcdn.com/bphoto/rw-UzO...	False	https://www.yelp.com/biz/nayos-african-cuisine...	
1	HWm-6GvXYaEZbHOz7rXzAA	mama-osas-african-restaurant-chicago	Mama Osas African Restaurant	https://s3-media2.fl.yelpcdn.com/bphoto/wyh5nN...	False	https://www.yelp.com/biz/mama-osas-african-res...	
2	ZTc65Wcj89EUuC7z2lixZA	southside-african-restaurant-chicago	Southside African Restaurant	https://s3-media1.fl.yelpcdn.com/bphoto/S1D2gn...	False	https://www.yelp.com/biz/southside-african-res...	
3	c5aJ75MBiOifSUaPf78VPg	k-i-african-restaurant-chicago	K I African Restaurant	https://s3-media4.fl.yelpcdn.com/bphoto/ApRj3S...	False	https://www.yelp.com/biz/k-i-african-restauran...	
4	nbbwOgv57uzt5A9UfGtX3w	vee-vee-african-restaurant-chicago	Vee Vee African Restaurant	https://s3-media4.fl.yelpcdn.com/bphoto/bVPB2N...	False	https://www.yelp.com/biz/vee-vee-african-resta...	

Next steps: Generate code with df4 View recommended plots New interactive sheet

```
search_api_url2 = 'https://api.yelp.com/v3/businesses/search/phone'

params5 = {'phone': '+12173441400'
```

```
}

response5 = requests.get(search_api_url2, headers=headers, params=params5, timeout=5)
data5 = response5.json()
df5 = pd.DataFrame(data5['businesses'])
df5.head()
```

	id	alias	name	image_url	is_closed	url	review_count	categ
0	-zEpEmDfFQL-ph0N3BDlXA	sakanaya-champaign-2	Sakanaya	https://s3-media0.fl.yelpcdn.com/bphoto/cHogr6...	False	https://www.yelp.com/biz/sakanaya-champaign-2?...	650	[{'japa 'Japa

```
search_api_url3 = 'https://api.yelp.com/v3/categories'

params6 = {'locale': 'en_US'}

response6 = requests.get(search_api_url3, headers=headers, params=params6, timeout=5)
data6 = response6.json()
df6 = pd.json_normalize(data6['categories'])
df6.head()
```

	alias	title	parent_aliases	country_whitelist	country_blacklist
0	3dprinting	3D Printing	[localservices]	[]	[]
1	acaibowls	Acai Bowls	[food]	[]	[AR, CL, IT, MX, PL, TR]
2	accessories	Accessories	[fashion]	[]	[]
3	accountants	Accountants	[professional]	[]	[]
4	acnetreatment	Acne Treatment	[beautysvc]	[]	[]

Next steps: Generate code with df6 | View recommended plots | New interactive sheet

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.