DP0701EN-2-2-1-Foursquare-API-py-v1.0

April 6, 2020

Learning FourSquare API with Python

0.1 Introduction

In this lab, you will learn in details how to make calls to the Foursquare API for different purposes. You will learn how to construct a URL to send a request to the API to search for a specific type of venues, to explore a particular venue, to explore a Foursquare user, to explore a geographical location, and to get trending venues around a location. Also, you will learn how to use the visualization library, Folium, to visualize the results.

0.2 Table of Contents

0.2.1 Import necessary Libraries

[3]:
```
import requests # library to handle requests
import pandas as pd # library for data analsysis
import numpy as np # library to handle data in a vectorized manner
import random # library for random number generation

# !conda install -c conda-forge --no-deps altair --yes
# !conda install -c conda-forge --no-deps vincent --yes

# !conda install -c conda-forge geopy=1.19.0 --yes
from geopy.geocoders import Nominatim # module to convert an address into␣
 ↪latitude and longitude values

# libraries for displaying images
from IPython.display import Image
from IPython.core.display import HTML
```

```
# tranforming json file into a pandas dataframe library
from pandas.io.json import json_normalize

# !conda install -c conda-forge --no-deps folium=0.5.0 --yes
import folium # plotting library

print('Folium installed')
print('Libraries imported.')
```

```
Folium installed
Libraries imported.
```

0.2.2 Define Foursquare Credentials and Version

Make sure that you have created a Foursquare developer account and have your credentials handy

[4]:
```
CLIENT_ID = 'ERDVFRWHSSDZCO4NV1XRDYEDCZUTKO4XZNCDZLALI4WPAH2I' # your␣
↪Foursquare ID
CLIENT_SECRET = 'BPJ5DALEOQS4ZNXWUI43DDHY2BDJUOYKDJNWOMYUDU2M5CJZ' # your␣
↪Foursquare Secret
VERSION = '20180604'
LIMIT = 30
print('Your credentails:')
print('CLIENT_ID: ' + CLIENT_ID)
print('CLIENT_SECRET:' + CLIENT_SECRET)
```

```
Your credentails:
CLIENT_ID: ERDVFRWHSSDZCO4NV1XRDYEDCZUTKO4XZNCDZLALI4WPAH2I
CLIENT_SECRET:BPJ5DALEOQS4ZNXWUI43DDHY2BDJUOYKDJNWOMYUDU2M5CJZ
```

Let's again assume that you are staying at the Conrad hotel. So let's start by converting the Contrad Hotel's address to its latitude and longitude coordinates. In order to define an instance of the geocoder, we need to define a user_agent. We will name our agent foursquare_agent, as shown below.

[5]:
```
address = '102 North End Ave, New York, NY'

geolocator = Nominatim(user_agent="foursquare_agent")
location = geolocator.geocode(address)
latitude = location.latitude
longitude = location.longitude
print(latitude, longitude)
```

```
40.7149555 -74.0153365
```

0.3 1. Search for a specific venue category

https://api.foursquare.com/v2/venues/**search**?client_id=**CLIENT_ID**&client_secret=**CLIENT**

Now, let's assume that it is lunch time, and you are craving Italian food. So, let's define a query to search for Italian food that is within 500 metres from the Conrad Hotel.

[6]:
```
search_query = 'Italian'
radius = 500
print(search_query + ' .... OK!')
```

```
Italian … OK!
```

Define the corresponding URL

[7]:
```
url = 'https://api.foursquare.com/v2/venues/search?
↪client_id={}&client_secret={}&ll={},{}&v={}&query={}&radius={}&limit={}'.
↪format(CLIENT_ID, CLIENT_SECRET, latitude, longitude, VERSION, search_query,␣
↪radius, LIMIT)
url
```

[7]:
```
'https://api.foursquare.com/v2/venues/search?client_id=ERDVFRWHSSDZC04NV1XRDYEDC
ZUTKO4XZNCDZLALI4WPAH2I&client_secret=BPJ5DALE0QS4ZNXWUI43DDHY2BDJU0YKDJNWOMYUDU
2M5CJZ&ll=40.7149555,-74.0153365&v=20180604&query=Italian&radius=500&limit=30'
```

Send the GET Request and examine the results

[8]:
```
results = requests.get(url).json()
results
```

[8]:
```
{'meta': {'code': 200, 'requestId': '5e8ada4e923935001b98ca8b'},
 'response': {'venues': [{'id': '4fa862b3e4b0ebff2f749f06',
    'name': "Harry's Italian Pizza Bar",
    'location': {'address': '225 Murray St',
     'lat': 40.71521779064671,
     'lng': -74.01473940209351,
     'labeledLatLngs': [{'label': 'display',
       'lat': 40.71521779064671,
       'lng': -74.01473940209351}],
     'distance': 58,
     'postalCode': '10282',
     'cc': 'US',
     'city': 'New York',
     'state': 'NY',
     'country': 'United States',
     'formattedAddress': ['225 Murray St',
      'New York, NY 10282',
      'United States']},
    'categories': [{'id': '4bf58dd8d48988d1ca941735',
      'name': 'Pizza Place',
      'pluralName': 'Pizza Places',
      'shortName': 'Pizza',
```

```
      'icon': {'prefix': 'https://ss3.4sqi.net/img/categories_v2/food/pizza_',
       'suffix': '.png'},
      'primary': True}],
    'delivery': {'id': '294544',
     'url': 'https://www.seamless.com/menu/harrys-italian-pizza-bar-225-murray-
st-new-york/294544?affiliate=1131&utm_source=foursquare-affiliate-
network&utm_medium=affiliate&utm_campaign=1131&utm_content=294544',
     'provider': {'name': 'seamless',
      'icon': {'prefix': 'https://fastly.4sqi.net/img/general/cap/',
       'sizes': [40, 50],
       'name': '/delivery_provider_seamless_20180129.png'}}},
    'referralId': 'v-1586158265',
    'hasPerk': False},
   {'id': '4f3232e219836c91c7bfde94',
    'name': 'Conca Cucina Italian Restaurant',
    'location': {'address': '63 W Broadway',
     'lat': 40.714484000000006,
     'lng': -74.00980600000001,
     'labeledLatLngs': [{'label': 'display',
       'lat': 40.714484000000006,
       'lng': -74.00980600000001}],
     'distance': 469,
     'postalCode': '10007',
     'cc': 'US',
     'city': 'New York',
     'state': 'NY',
     'country': 'United States',
     'formattedAddress': ['63 W Broadway',
      'New York, NY 10007',
      'United States']},
    'categories': [{'id': '4d4b7105d754a06374d81259',
      'name': 'Food',
      'pluralName': 'Food',
      'shortName': 'Food',
      'icon': {'prefix': 'https://ss3.4sqi.net/img/categories_v2/food/default_',
       'suffix': '.png'},
      'primary': True}],
    'referralId': 'v-1586158265',
    'hasPerk': False},
   {'id': '3fd66200f964a520f4e41ee3',
    'name': 'Ecco',
    'location': {'address': '124 Chambers St',
     'crossStreet': 'btwn Church St & W Broadway',
     'lat': 40.71533713859952,
     'lng': -74.00884766217825,
     'labeledLatLngs': [{'label': 'display',
       'lat': 40.71533713859952,
```

```
        'lng': -74.00884766217825},
       {'label': '?', 'lat': 40.715202, 'lng': -74.008779}],
      'distance': 549,
      'postalCode': '10007',
      'cc': 'US',
      'city': 'New York',
      'state': 'NY',
      'country': 'United States',
      'formattedAddress': ['124 Chambers St (btwn Church St & W Broadway)',
       'New York, NY 10007',
       'United States']},
     'categories': [{'id': '4bf58dd8d48988d110941735',
       'name': 'Italian Restaurant',
       'pluralName': 'Italian Restaurants',
       'shortName': 'Italian',
       'icon': {'prefix': 'https://ss3.4sqi.net/img/categories_v2/food/italian_',
        'suffix': '.png'},
       'primary': True}],
     'referralId': 'v-1586158265',
     'hasPerk': False}]}}
```

Get relevant part of JSON and transform it into a *pandas* dataframe

[11]:
```
# assign relevant part of JSON to venues
venues = results['response']['venues']

# tranform venues into a dataframe
dataframe = json_normalize(venues)
dataframe.head()
```

```
/home/jupyterlab/conda/envs/python/lib/python3.6/site-
packages/ipykernel_launcher.py:5: FutureWarning: pandas.io.json.json_normalize
is deprecated, use pandas.json_normalize instead
  """
```

[11]:

	id	name	categories	referralId	hasPerk	location.address	location.lat	location.lng
0	4fa862b3e4b0ebff2f749f06	Harry's Italian Pizza Bar	[{'id': '4bf58dd8d48988d1ca941735', 'name': 'P…	v-1586158265	False	225 Murray St	40.715218	-74.014739
1	4f3232e219836c91c7bfde94	Conca Cucina Italian Restaurant	[{'id': '4d4b7105d754a06374d81259', 'name': 'F…	v-1586158265	False			
2	3fd66200f964a520f4e41ee3	Ecco	[{'id': '4bf58dd8d48988d110941735', 'name': 'I…	v-1586158265	False			

```
1    63 W Broadway     40.714484    -74.009806
2  124 Chambers St     40.715337    -74.008848

                            location.labeledLatLngs  location.distance  …  \
0  [{'label': 'display', 'lat': 40.71521779064671…                 58  …
1  [{'label': 'display', 'lat': 40.71448400000000…                469  …
2  [{'label': 'display', 'lat': 40.71533713859952…                549  …

  location.state location.country  \
0             NY    United States
1             NY    United States
2             NY    United States

                          location.formattedAddress delivery.id  \
0  [225 Murray St, New York, NY 10282, United Sta…      294544
1  [63 W Broadway, New York, NY 10007, United Sta…         NaN
2  [124 Chambers St (btwn Church St & W Broadway)…         NaN

                                       delivery.url delivery.provider.name  \
0  https://www.seamless.com/menu/harrys-italian-p…               seamless
1                                              NaN                    NaN
2                                              NaN                    NaN

             delivery.provider.icon.prefix delivery.provider.icon.sizes  \
0  https://fastly.4sqi.net/img/general/cap/                     [40, 50]
1                                      NaN                          NaN
2                                      NaN                          NaN

                delivery.provider.icon.name         location.crossStreet
0  /delivery_provider_seamless_20180129.png                          NaN
1                                      NaN                          NaN
2                                      NaN  btwn Church St & W Broadway

[3 rows x 23 columns]
```

Define information of interest and filter dataframe

[12]:
```
# keep only columns that include venue name, and anything that is associated
 ↪with location
filtered_columns = ['name', 'categories'] + [col for col in dataframe.columns
 ↪if col.startswith('location.')] + ['id']
dataframe_filtered = dataframe.loc[:, filtered_columns]

# function that extracts the category of the venue
def get_category_type(row):
    try:
        categories_list = row['categories']
```

```
    except:
        categories_list = row['venue.categories']

    if len(categories_list) == 0:
        return None
    else:
        return categories_list[0]['name']

# filter the category for each row
dataframe_filtered['categories'] = dataframe_filtered.apply(get_category_type,␣
↪axis=1)

# clean column names by keeping only last term
dataframe_filtered.columns = [column.split('.')[-1] for column in␣
↪dataframe_filtered.columns]

dataframe_filtered
```

[12]:

```
                            name          categories          address  \
0       Harry's Italian Pizza Bar         Pizza Place    225 Murray St
1  Conca Cucina Italian Restaurant                Food    63 W Broadway
2                            Ecco  Italian Restaurant  124 Chambers St

         lat        lng                                    labeledLatLngs  \
0  40.715218 -74.014739  [{'label': 'display', 'lat': 40.71521779064671…
1  40.714484 -74.009806  [{'label': 'display', 'lat': 40.71448400000000…
2  40.715337 -74.008848  [{'label': 'display', 'lat': 40.71533713859952…

   distance postalCode  cc      city state        country  \
0        58      10282  US  New York    NY  United States
1       469      10007  US  New York    NY  United States
2       549      10007  US  New York    NY  United States

                                formattedAddress  \
0  [225 Murray St, New York, NY 10282, United Sta…
1  [63 W Broadway, New York, NY 10007, United Sta…
2  [124 Chambers St (btwn Church St & W Broadway)…

                  crossStreet                        id
0                         NaN  4fa862b3e4b0ebff2f749f06
1                         NaN  4f3232e219836c91c7bfde94
2  btwn Church St & W Broadway  3fd66200f964a520f4e41ee3
```

Let's visualize the Italian restaurants that are nearby

[13]:

```
dataframe_filtered.name
```

[13]:
```
0          Harry's Italian Pizza Bar
1    Conca Cucina Italian Restaurant
2                              Ecco
Name: name, dtype: object
```

[15]:
```
venues_map = folium.Map(location=[latitude, longitude], zoom_start=13) #␣
↪generate map centred around the Conrad Hotel

# add a red circle marker to represent the Conrad Hotel
folium.features.CircleMarker(
    [latitude, longitude],
    radius=10,
    color='red',
    popup='Conrad Hotel',
    fill = True,
    fill_color = 'red',
    fill_opacity = 0.6
).add_to(venues_map)

# add the Italian restaurants as blue circle markers
for lat, lng, label in zip(dataframe_filtered.lat, dataframe_filtered.lng,␣
↪dataframe_filtered.categories):
    folium.features.CircleMarker(
        [lat, lng],
        radius=5,
        color='blue',
        popup=label,
        fill = True,
        fill_color='blue',
        fill_opacity=0.6
    ).add_to(venues_map)

# display map
venues_map
```

[15]:
```
<folium.folium.Map at 0x7fefba3f86d8>
```

0.4 2. Explore a Given Venue

https://api.foursquare.com/v2/venues/**VENUE_ID**?client_id=**CLIENT_ID**&client_secret=**C**

0.4.1 A. Let's explore the closest Italian restaurant – *Harry's Italian Pizza Bar*

[16]:
```
venue_id = '4fa862b3e4b0ebff2f749f06' # ID of Harry's Italian Pizza Bar
url = 'https://api.foursquare.com/v2/venues/{}?
↪client_id={}&client_secret={}&v={}'.format(venue_id, CLIENT_ID,␣
↪CLIENT_SECRET, VERSION)
```

```
url
```

[16]:
```
'https://api.foursquare.com/v2/venues/4fa862b3e4b0ebff2f749f06?client_id=ERDVFRW
HSSDZC04NV1XRDYEDCZUTKO4XZNCDZLALI4WPAH2I&client_secret=BPJ5DALEOQS4ZNXWUI43DDHY
2BDJUOYKDJNWOMYUDU2M5CJZ&v=20180604'
```

Send GET request for result

[17]:
```
result = requests.get(url).json()
print(result['response']['venue'].keys())
result['response']['venue']
```

```
dict_keys(['id', 'name', 'contact', 'location', 'canonicalUrl', 'categories',
'verified', 'stats', 'url', 'price', 'hasMenu', 'likes', 'dislike', 'ok',
'rating', 'ratingColor', 'ratingSignals', 'delivery', 'menu',
'allowMenuUrlEdit', 'beenHere', 'specials', 'photos', 'reasons', 'hereNow',
'createdAt', 'tips', 'shortUrl', 'timeZone', 'listed', 'hours', 'popular',
'seasonalHours', 'defaultHours', 'pageUpdates', 'inbox', 'attributes',
'bestPhoto', 'colors'])
```

[17]:
```
{'id': '4fa862b3e4b0ebff2f749f06',
 'name': "Harry's Italian Pizza Bar",
 'contact': {'phone': '2126081007', 'formattedPhone': '(212) 608-1007'},
 'location': {'address': '225 Murray St',
  'lat': 40.71521779064671,
  'lng': -74.01473940209351,
  'labeledLatLngs': [{'label': 'display',
    'lat': 40.71521779064671,
    'lng': -74.01473940209351}],
  'postalCode': '10282',
  'cc': 'US',
  'city': 'New York',
  'state': 'NY',
  'country': 'United States',
  'formattedAddress': ['225 Murray St',
   'New York, NY 10282',
   'United States']},
 'canonicalUrl': 'https://foursquare.com/v/harrys-italian-pizza-
bar/4fa862b3e4b0ebff2f749f06',
 'categories': [{'id': '4bf58dd8d48988d1ca941735',
   'name': 'Pizza Place',
   'pluralName': 'Pizza Places',
   'shortName': 'Pizza',
   'icon': {'prefix': 'https://ss3.4sqi.net/img/categories_v2/food/pizza_',
    'suffix': '.png'},
   'primary': True},
  {'id': '4bf58dd8d48988d110941735',
```

```
   'name': 'Italian Restaurant',
   'pluralName': 'Italian Restaurants',
   'shortName': 'Italian',
   'icon': {'prefix': 'https://ss3.4sqi.net/img/categories_v2/food/italian_',
    'suffix': '.png'}}],
 'verified': False,
 'stats': {'tipCount': 57},
 'url': 'http://harrysitalian.com',
 'price': {'tier': 2, 'message': 'Moderate', 'currency': '$'},
 'hasMenu': True,
 'likes': {'count': 120,
  'groups': [{'type': 'others', 'count': 120, 'items': []}],
  'summary': '120 Likes'},
 'dislike': False,
 'ok': False,
 'rating': 7.0,
 'ratingColor': 'C5DE35',
 'ratingSignals': 212,
 'delivery': {'id': '294544',
  'url': 'https://www.seamless.com/menu/harrys-italian-pizza-bar-225-murray-st-
new-york/294544?affiliate=1131&utm_source=foursquare-affiliate-
network&utm_medium=affiliate&utm_campaign=1131&utm_content=294544',
  'provider': {'name': 'seamless',
   'icon': {'prefix': 'https://fastly.4sqi.net/img/general/cap/',
    'sizes': [40, 50],
    'name': '/delivery_provider_seamless_20180129.png'}}},
 'menu': {'type': 'Menu',
  'label': 'Menu',
  'anchor': 'View Menu',
  'url': 'https://foursquare.com/v/harrys-italian-pizza-
bar/4fa862b3e4b0ebff2f749f06/menu',
  'mobileUrl': 'https://foursquare.com/v/4fa862b3e4b0ebff2f749f06/device_menu'},
 'allowMenuUrlEdit': True,
 'beenHere': {'count': 0,
  'unconfirmedCount': 0,
  'marked': False,
  'lastCheckinExpiredAt': 0},
 'specials': {'count': 0, 'items': []},
 'photos': {'count': 147,
  'groups': [{'type': 'venue',
    'name': 'Venue photos',
    'count': 147,
    'items': [{'id': '4fad980de4b091b4626c3633',
      'createdAt': 1336776717,
      'source': {'name': 'Foursquare for Android',
       'url': 'https://foursquare.com/download/#/android'},
      'prefix': 'https://fastly.4sqi.net/img/general/',
```

```
      'suffix': '/ya1iQFI7pLjuIJp1PGDKlrZS3OJdHCF7tpILMmjv_2w.jpg',
      'width': 480,
      'height': 640,
      'user': {'id': '13676709',
       'firstName': 'Leony',
       'lastName': 'N',
       'photo': {'prefix': 'https://fastly.4sqi.net/img/user/',
        'suffix': '/TOANFNGNMCHUDEUE.jpg'}},
      'visibility': 'public'}]}]},
 'reasons': {'count': 1,
  'items': [{'summary': 'Lots of people like this place',
    'type': 'general',
    'reasonName': 'rawLikesReason'}]},
 'hereNow': {'count': 0, 'summary': 'Nobody here', 'groups': []},
 'createdAt': 1336435379,
 'tips': {'count': 57,
  'groups': [{'type': 'others',
    'name': 'All tips',
    'count': 57,
    'items': [{'id': '53d27909498e0523841340b6',
      'createdAt': 1406302473,
      'text': "Harry's Italian Pizza bar is known for it's amazing pizza, but
did you know that the brunches here are amazing too? Try the Nutella French
toast and we know you'll be sold.",
      'type': 'user',
      'canonicalUrl': 'https://foursquare.com/item/53d27909498e0523841340b6',
      'lang': 'en',
      'likes': {'count': 4,
       'groups': [{'type': 'others',
         'count': 4,
         'items': [{'id': '369426',
           'firstName': 'P.',
           'lastName': 'M',
           'photo': {'prefix': 'https://fastly.4sqi.net/img/user/',
            'suffix': '/JPQYUWJKUTOH2OO4.jpg'}},
          {'id': '87587879',
           'firstName': 'Diane',
           'lastName': 'D',
           'photo': {'prefix': 'https://fastly.4sqi.net/img/user/',
            'suffix': '/87587879-ESLRSZLQ2CBE2P4W.jpg'}},
          {'id': '87591341',
           'firstName': 'Tim',
           'lastName': 'S',
           'photo': {'prefix': 'https://fastly.4sqi.net/img/user/',
            'suffix': '/-Z4YK4VKEOJSVXIY1.jpg'}},
          {'id': '87473404',
           'firstName': 'TenantKing.com',
```

```
            'photo': {'prefix': 'https://fastly.4sqi.net/img/user/',
             'suffix': '/87473404-HI5DTBTK0HX401CA.png'},
            'type': 'page'}]}],
        'summary': '4 likes'},
       'logView': True,
       'agreeCount': 4,
       'disagreeCount': 0,
       'todo': {'count': 0},
       'user': {'id': '87473404',
        'firstName': 'TenantKing.com',
        'photo': {'prefix': 'https://fastly.4sqi.net/img/user/',
         'suffix': '/87473404-HI5DTBTK0HX401CA.png'},
        'type': 'page'}}]}]},
 'shortUrl': 'http://4sq.com/JNblHV',
 'timeZone': 'America/New_York',
 'listed': {'count': 54,
  'groups': [{'type': 'others',
    'name': 'Lists from other people',
    'count': 54,
    'items': [{'id': '4fa32fd0e4b04193744746b1',
      'name': 'Manhattan Haunts',
      'description': '',
      'type': 'others',
      'user': {'id': '24592223',
       'firstName': 'Becca',
       'lastName': 'M',
       'photo': {'prefix': 'https://fastly.4sqi.net/img/user/',
        'suffix': '/24592223-RAW2UYM0GIB1U40K.jpg'}},
      'editable': False,
      'public': True,
      'collaborative': False,
      'url': '/becca_mcarthur/list/manhattan-haunts',
      'canonicalUrl': 'https://foursquare.com/becca_mcarthur/list/manhattan-
haunts',
      'createdAt': 1336094672,
      'updatedAt': 1380845377,
      'photo': {'id': '4e8cc9461081e3b3544e12e5',
       'createdAt': 1317849414,
       'prefix': 'https://fastly.4sqi.net/img/general/',
       'suffix': '/0NLVU2HC1JF4DXIMKWUFW3QBUT31DC11EFNYYHMJG3NDWAPS.jpg',
       'width': 492,
       'height': 330,
       'user': {'id': '742542',
        'firstName': 'Time Out New York',
        'photo': {'prefix': 'https://fastly.4sqi.net/img/user/',
         'suffix': '/XXHKCBSQHBORZNSR.jpg'},
        'type': 'page'},
```

```
       'visibility': 'public'},
      'followers': {'count': 22},
      'listItems': {'count': 187,
       'items': [{'id': 'v4fa862b3e4b0ebff2f749f06',
         'createdAt': 1342934485}]}},
     {'id': '4fae817be4b085f6b2a74d19',
      'name': 'USA NYC MAN FiDi',
      'description': 'Where to go for decent eats in the restaurant wasteland of
Downtown NYC aka FiDi, along with Tribeca & Battery Park City.',
      'type': 'others',
      'user': {'id': '12113441',
       'firstName': 'Kino',
       'photo': {'prefix': 'https://fastly.4sqi.net/img/user/',
        'suffix': '/12113441-K5HTHFLU2MUCM0CM.jpg'}},
      'editable': False,
      'public': True,
      'collaborative': False,
      'url': '/kinosfault/list/usa-nyc-man-fidi',
      'canonicalUrl': 'https://foursquare.com/kinosfault/list/usa-nyc-man-fidi',
      'createdAt': 1336836475,
      'updatedAt': 1556754919,
      'photo': {'id': '55984992498e13ba75e353bb',
       'createdAt': 1436043666,
       'prefix': 'https://fastly.4sqi.net/img/general/',
       'suffix': '/12113441_iOa6Uh-Xi8bhj2-gpzkkw8MKiAIs7RmOcz_RM7m8ink.jpg',
       'width': 540,
       'height': 960,
       'user': {'id': '12113441',
        'firstName': 'Kino',
        'photo': {'prefix': 'https://fastly.4sqi.net/img/user/',
         'suffix': '/12113441-K5HTHFLU2MUCM0CM.jpg'}},
       'visibility': 'public'},
      'followers': {'count': 20},
      'listItems': {'count': 273,
       'items': [{'id': 'v4fa862b3e4b0ebff2f749f06',
         'createdAt': 1373909433}]}},
     {'id': '4fddeff0e4b0e078037ac0d3',
      'name': 'NYC Resturants',
      'description': '',
      'type': 'others',
      'user': {'id': '21563126',
       'firstName': 'Richard',
       'lastName': 'R',
       'photo': {'prefix': 'https://fastly.4sqi.net/img/user/',
        'suffix': '/21563126_v05J1KPw_SVj6Ehq9g8B9jeAGjFUMsU5QGl-
NZ8inUQ7pKQm5bKplW37EmR7jS2A7GYPBBAtl.jpg'}},
      'editable': False,
```

```
      'public': True,
      'collaborative': True,
      'url': '/rickr7/list/nyc-resturants',
      'canonicalUrl': 'https://foursquare.com/rickr7/list/nyc-resturants',
      'createdAt': 1339944944,
      'updatedAt': 1582232663,
      'photo': {'id': '5072dd13e4b09145cdf782d1',
       'createdAt': 1349704979,
       'prefix': 'https://fastly.4sqi.net/img/general/',
       'suffix': '/208205_fGh2OuAZ9qJ4agbAA5wMVNOSIm9kNUlRtNwj1N-adqg.jpg',
       'width': 800,
       'height': 800,
       'user': {'id': '208205',
        'firstName': 'Thalia',
        'lastName': 'K',
        'photo': {'prefix': 'https://fastly.4sqi.net/img/user/',
         'suffix': '/SNOOLCAW2AG04ZKD.jpg'}},
       'visibility': 'public'},
      'followers': {'count': 12},
      'listItems': {'count': 194,
       'items': [{'id': 'v4fa862b3e4b0ebff2f749f06',
         'createdAt': 1581655865}]}},
     {'id': '5266c68a498e7c667807fe09',
      'name': 'Foodie Love in NY - 02',
      'description': '',
      'type': 'others',
      'user': {'id': '547977',
       'firstName': 'WiLL',
       'photo': {'prefix': 'https://fastly.4sqi.net/img/user/',
        'suffix': '/-Q5NYGDMFDMOITQRR.jpg'}},
      'editable': False,
      'public': True,
      'collaborative': False,
      'url': '/sweetiewill/list/foodie-love-in-ny--02',
      'canonicalUrl': 'https://foursquare.com/sweetiewill/list/foodie-love-in-ny
--02',
      'createdAt': 1382467210,
      'updatedAt': 1391995585,
      'followers': {'count': 7},
      'listItems': {'count': 200,
       'items': [{'id': 'v4fa862b3e4b0ebff2f749f06',
         'createdAt': 1386809936}]}}]}],
 'hours': {'status': 'Closed until 11:30 AM',
  'richStatus': {'entities': [], 'text': 'Closed until 11:30 AM'},
  'isOpen': False,
  'isLocalHoliday': False,
  'dayData': [],
```

```
 'timeframes': [{'days': 'Mon-Wed, Sun',
   'includesToday': True,
   'open': [{'renderedTime': '11:30 AM-11:00 PM'}],
   'segments': []},
  {'days': 'Thu-Sat',
   'open': [{'renderedTime': '11:30 AM-Midnight'}],
   'segments': []}]},
'popular': {'isOpen': False,
 'isLocalHoliday': False,
 'timeframes': [{'days': 'Today',
   'includesToday': True,
   'open': [{'renderedTime': 'Noon-2:00 PM'},
    {'renderedTime': '6:00 PM-8:00 PM'}],
   'segments': []},
  {'days': 'Tue-Thu',
   'open': [{'renderedTime': 'Noon-2:00 PM'},
    {'renderedTime': '5:00 PM-10:00 PM'}],
   'segments': []},
  {'days': 'Fri',
   'open': [{'renderedTime': 'Noon-3:00 PM'},
    {'renderedTime': '5:00 PM-11:00 PM'}],
   'segments': []},
  {'days': 'Sat',
   'open': [{'renderedTime': 'Noon-11:00 PM'}],
   'segments': []},
  {'days': 'Sun',
   'open': [{'renderedTime': 'Noon-3:00 PM'},
    {'renderedTime': '5:00 PM-8:00 PM'}],
   'segments': []}]},
'seasonalHours': [],
'defaultHours': {'status': 'Closed until 11:30 AM',
 'richStatus': {'entities': [], 'text': 'Closed until 11:30 AM'},
 'isOpen': False,
 'isLocalHoliday': False,
 'dayData': [],
 'timeframes': [{'days': 'Mon-Wed, Sun',
   'includesToday': True,
   'open': [{'renderedTime': '11:30 AM-11:00 PM'}],
   'segments': []},
  {'days': 'Thu-Sat',
   'open': [{'renderedTime': '11:30 AM-Midnight'}],
   'segments': []}]},
'pageUpdates': {'count': 0, 'items': []},
'inbox': {'count': 0, 'items': []},
'attributes': {'groups': [{'type': 'price',
   'name': 'Price',
   'summary': '$$',
```

```
   'count': 1,
   'items': [{'displayName': 'Price', 'displayValue': '$$', 'priceTier': 2}]},
  {'type': 'payments',
   'name': 'Credit Cards',
   'summary': 'Credit Cards',
   'count': 7,
   'items': [{'displayName': 'Credit Cards',
     'displayValue': 'Yes (incl. American Express)'}]},
  {'type': 'outdoorSeating',
   'name': 'Outdoor Seating',
   'summary': 'Outdoor Seating',
   'count': 1,
   'items': [{'displayName': 'Outdoor Seating', 'displayValue': 'Yes'}]},
  {'type': 'serves',
   'name': 'Menus',
   'summary': 'Happy Hour, Brunch & more',
   'count': 8,
   'items': [{'displayName': 'Brunch', 'displayValue': 'Brunch'},
    {'displayName': 'Lunch', 'displayValue': 'Lunch'},
    {'displayName': 'Dinner', 'displayValue': 'Dinner'},
    {'displayName': 'Happy Hour', 'displayValue': 'Happy Hour'}]},
  {'type': 'drinks',
   'name': 'Drinks',
   'summary': 'Beer, Wine & Cocktails',
   'count': 5,
   'items': [{'displayName': 'Beer', 'displayValue': 'Beer'},
    {'displayName': 'Wine', 'displayValue': 'Wine'},
    {'displayName': 'Cocktails', 'displayValue': 'Cocktails'}]},
  {'type': 'diningOptions',
   'name': 'Dining Options',
   'summary': 'Delivery',
   'count': 5,
   'items': [{'displayName': 'Delivery', 'displayValue': 'Delivery'}]}]},
'bestPhoto': {'id': '4fad980de4b091b4626c3633',
 'createdAt': 1336776717,
 'source': {'name': 'Foursquare for Android',
  'url': 'https://foursquare.com/download/#/android'},
 'prefix': 'https://fastly.4sqi.net/img/general/',
 'suffix': '/ya1iQFI7pLjuIJp1PGDKlrZS3OJdHCF7tpILMmjv_2w.jpg',
 'width': 480,
 'height': 640,
 'visibility': 'public'},
'colors': {'highlightColor': {'photoId': '4fad980de4b091b4626c3633',
  'value': -13619152},
 'highlightTextColor': {'photoId': '4fad980de4b091b4626c3633', 'value': -1},
 'algoVersion': 3}}
```

0.4.2 B. Get the venue's overall rating

[18]:
```
try:
    print(result['response']['venue']['rating'])
except:
    print('This venue has not been rated yet.')
```

```
7.0
```

That is not a very good rating. Let's check the rating of the second closest Italian restaurant.

[19]:
```
venue_id = '4f3232e219836c91c7bfde94' # ID of Conca Cucina Italian Restaurant
url = 'https://api.foursquare.com/v2/venues/{}?
 ↪client_id={}&client_secret={}&v={}'.format(venue_id, CLIENT_ID,␣
 ↪CLIENT_SECRET, VERSION)

result = requests.get(url).json()
try:
    print(result['response']['venue']['rating'])
except:
    print('This venue has not been rated yet.')
```

```
This venue has not been rated yet.
```

Since this restaurant has no ratings, let's check the third restaurant.

[20]:
```
venue_id = '3fd66200f964a520f4e41ee3' # ID of Ecco
url = 'https://api.foursquare.com/v2/venues/{}?
 ↪client_id={}&client_secret={}&v={}'.format(venue_id, CLIENT_ID,␣
 ↪CLIENT_SECRET, VERSION)

result = requests.get(url).json()
try:
    print(result['response']['venue']['rating'])
except:
    print('This venue has not been rated yet.')
```

```
7.4
```

Since this restaurant has a slightly better rating, let's explore it further.

0.4.3 C. Get the number of tips

[21]:
```
result['response']['venue']['tips']['count']
```

[21]:
```
19
```

0.4.4 D. Get the venue's tips

https://api.foursquare.com/v2/venues/**VENUE_ID**/tips?client_id=**CLIENT_ID**&client_secr

Create URL and send GET request. Make sure to set limit to get all tips

[22]:
```
## Ecco Tips
limit = 15 # set limit to be greater than or equal to the total number of tips
url = 'https://api.foursquare.com/v2/venues/{}/tips?
 ↪client_id={}&client_secret={}&v={}&limit={}'.format(venue_id, CLIENT_ID,␣
 ↪CLIENT_SECRET, VERSION, limit)

results = requests.get(url).json()
results
```

[22]:
```
{'meta': {'code': 200, 'requestId': '5e8adbca0f59680026ac284c'},
 'response': {'tips': {'count': 19,
   'items': [{'id': '5ab1cb46c9a517174651d3fe',
     'createdAt': 1521601350,
     'text': 'A+ Italian food! Trust me on this: my mom's side of the family is
100% Italian. I was born and bred to know good pasta when I see it, and Ecco is
one of my all-time NYC favorites',
     'type': 'user',
     'canonicalUrl': 'https://foursquare.com/item/5ab1cb46c9a517174651d3fe',
     'lang': 'en',
     'likes': {'count': 0, 'groups': []},
     'logView': True,
     'agreeCount': 4,
     'disagreeCount': 0,
     'todo': {'count': 0},
     'user': {'id': '484542633',
      'firstName': 'Nick',
      'lastName': 'E',
      'photo': {'prefix': 'https://fastly.4sqi.net/img/user/',
       'suffix': '/484542633_unymNUmw_FdPs3GjXHujmHcYnN4hf8kEPADlOZuIrdcdm97VX3t
FqL7fFNMNA_8Gl9NlU1GYg.jpg'}},
     'authorInteractionType': 'liked'}]}}}
```

Get tips and list of associated features

[26]:
```
tips = results['response']['tips']['items']

tip = results['response']['tips']['items'][0]
tip.keys()
```

[26]:
```
dict_keys(['id', 'createdAt', 'text', 'type', 'canonicalUrl', 'lang', 'likes',
'logView', 'agreeCount', 'disagreeCount', 'todo', 'user',
'authorInteractionType'])
```

Format column width and display all tips

[30]:
```
pd.set_option('display.max_colwidth', None)

tips_df = json_normalize(tips) # json normalize tips

# columns to keep
filtered_columns = ['text', 'agreeCount', 'disagreeCount', 'id', 'user.
 ↪firstName', 'user.lastName', 'user.id']
tips_filtered = tips_df.loc[:, filtered_columns]

# display tips
tips_filtered
```

```
/home/jupyterlab/conda/envs/python/lib/python3.6/site-
packages/ipykernel_launcher.py:3: FutureWarning: pandas.io.json.json_normalize
is deprecated, use pandas.json_normalize instead
  This is separate from the ipykernel package so we can avoid doing imports
until
```

[30]:
```
                  text  \
0  A+ Italian food! Trust me on this: my mom's side of the family is 100%
Italian. I was born and bred to know good pasta when I see it, and Ecco is one
of my all-time NYC favorites

   agreeCount  disagreeCount                        id user.firstName  \
0           4              0  5ab1cb46c9a517174651d3fe           Nick

  user.lastName    user.id
0             E  484542633
```

Now remember that because we are using a personal developer account, then we can access only 2 of the restaurant's tips, instead of all 15 tips.

0.5 3. Search a Foursquare User

https://api.foursquare.com/v2/users/**USER_ID**?client_id=**CLIENT_ID**&client_secret=**CLIE**

0.5.1 Define URL, send GET request and display features associated with user

[33]:
```
user_id = '484542633' # user ID with most agree counts and complete profile

ACCESS_TOKEN = '5C0D5A1PM5MJM4MHBPVDTSMHNJJBDZK4NP4WGDIIYMRONIXS'

url = 'https://api.foursquare.com/v2/users/{}?
 ↪client_id={}&client_secret={}&oauth_token={}&v={}'.format(user_id,␣
 ↪CLIENT_ID, CLIENT_SECRET, ACCESS_TOKEN, VERSION) # define URL

# send GET request
results = requests.get(url).json()
```

```
user_data = results['response']['user']

# display features associated with user
user_data.keys()
```

[33]:
```
dict_keys(['id', 'firstName', 'lastName', 'gender', 'canonicalUrl', 'photo',
'friends', 'tips', 'homeCity', 'bio', 'contact', 'photos', 'checkinPings',
'pings', 'type', 'mayorships', 'checkins', 'lists', 'blockedStatus', 'lenses'])
```

[34]:
```
print('First Name: ' + user_data['firstName'])
print('Last Name: ' + user_data['lastName'])
print('Home City: ' + user_data['homeCity'])
```

```
First Name: Nick
Last Name: El-Tawil
Home City: New York, NY
```

How many tips has this user submitted?

[35]:
```
user_data['tips']
```

[35]:
```
{'count': 99}
```

Wow! So it turns out that Nick is a very active Foursquare user, with more than 250 tips.

0.5.2 Get User's tips

[38]:
```
# define tips URL
url = 'https://api.foursquare.com/v2/users/{}/tips?
 ↪client_id={}&client_secret={}&v={}&limit={}'.format(user_id, CLIENT_ID,␣
 ↪CLIENT_SECRET, VERSION, limit)

# send GET request and get user's tips
results = requests.get(url).json()
tips = results['response']['tips']['items']

# format column width
pd.set_option('display.max_colwidth', None)

tips_df = json_normalize(tips)

# filter columns
filtered_columns = ['text', 'agreeCount', 'disagreeCount', 'id']
tips_filtered = tips_df.loc[:, filtered_columns]

# display user's tips
tips_filtered
```

```
/home/jupyterlab/conda/envs/python/lib/python3.6/site-
packages/ipykernel_launcher.py:11: FutureWarning: pandas.io.json.json_normalize
is deprecated, use pandas.json_normalize instead
  # This is added back by InteractiveShellApp.init_path()
```

[38]:

```
                                text  \
0
They serve coffee!!!!!!
1
The linguine with clams is on point
2
Quick, cheap lunch that tastes good! Way shorter line than Chipotle, too.
3
You’re not a real New Yorker until you’ve shame-ordered Insomnia Cookies for
delivery at 3am
4
Good for you yet still tasty! Clean green protein is my go-to after I hit the
gym
5
Burger game strong
6                                                                  Great burgers &
fries! Also, this place is exactly what it’s like when you go to a bar in the
Southwest. Source: I’m from Arizona.
7
Açaí bowl + peanut butter + whey protein =
8
Highly underrated and way less crowded than Central Park!
9   Way easier to navigate than the Met proper, plus the Met Breuer focuses on
modern art. If I only have a limited amount of time to spend in a museum, I
would rather go here than anywhere else!
10
Get the açaí bowl with peanut butter after your work out and thank me later
11
When you want a burger, this should be the first thing that comes to mind. A+!
12
Way less crowded than Central Park! People who live in the neighborhood rave
about Carl Schurz Park.
13
The best Mexican food in the Murray Hill / Kips Bay area!
14
Best coffee shop in the neighborhood!

    agreeCount  disagreeCount                        id
0            1              0  5accc98c0313204c9d7ec157
1            1              0  5accbe3a911fc423730f3ed3
2            2              0  5acbec70a0215b732e264fe8
3            1              0  5acbbd4eb1538e45373b07f5
```

```
4           2           0  5acbbcda01235808d5d6dc75
5           1           0  5ab575fb6bdee65f759da8c1
6           2           0  5ab5575d73fe2516ad8f363b
7           1           0  5ab42db53c858d64af2688a4
8           3           0  5ab42c396f706a29f53ad1a8
9           6           0  5ab42b987dc9e17930e5ff5b
10          1           0  5ab42aca2a7ab6333652b266
11          1           0  5ab42a28da5e5617d18e3a6a
12          3           0  5ab429db1ffe971b060083f5
13          1           0  5ab3f53f8496ca57542d5549
14          1           0  5ab3f428da5e5617d17d1475
```

Let's get the venue for the tip with the greatest number of agree counts

[39]:
```
tip_id = '5ab5575d73fe2516ad8f363b' # tip id

# define URL
url = 'http://api.foursquare.com/v2/tips/{}?client_id={}&client_secret={}&v={}'.
 ↪format(tip_id, CLIENT_ID, CLIENT_SECRET, VERSION)

# send GET Request and examine results
result = requests.get(url).json()
print(result['response']['tip']['venue']['name'])
print(result['response']['tip']['venue']['location'])
```

```
Cowgirl
{'address': '519 Hudson St', 'crossStreet': 'at W 10th St', 'lat':
40.73373338282062, 'lng': -74.0062998849649, 'labeledLatLngs': [{'label':
'display', 'lat': 40.73373338282062, 'lng': -74.0062998849649}], 'postalCode':
'10014', 'cc': 'US', 'city': 'New York', 'state': 'NY', 'country': 'United
States', 'formattedAddress': ['519 Hudson St (at W 10th St)', 'New York, NY
10014', 'United States']}
```

0.5.3 Get User's friends

[41]:
```
user_friends = json_normalize(user_data['friends']['groups'][0]['items'])
user_friends
```

```
/home/jupyterlab/conda/envs/python/lib/python3.6/site-
packages/ipykernel_launcher.py:1: FutureWarning: pandas.io.json.json_normalize
is deprecated, use pandas.json_normalize instead
  """Entry point for launching an IPython kernel.
```

[41]:
```
Empty DataFrame
Columns: []
Index: []
```

Interesting. Despite being very active, it turns out that Nick does not have any friends on

Foursquare. This might definitely change in the future.

0.5.4 Retrieve the User's Profile Image

[42]:
```
user_data
```

[42]:

```
{'id': '484542633',
 'firstName': 'Nick',
 'lastName': 'El-Tawil',
 'gender': 'male',
 'canonicalUrl': 'https://foursquare.com/user/484542633',
 'photo': {'prefix': 'https://fastly.4sqi.net/img/user/',
  'suffix': '/484542633_unymNUmw_FdPs3GjXHujmHcYnN4hf8kEPADlOZuIrdcdm97VX3tFqL7f
FNMNA_8Gl9NlU1GYg.jpg'},
 'friends': {'count': 0,
  'groups': [{'type': 'friends',
    'name': 'Mutual friends',
    'count': 0,
    'items': []},
   {'type': 'others', 'name': 'Other friends', 'count': 0, 'items': []}]},
 'tips': {'count': 99},
 'homeCity': 'New York, NY',
 'bio': 'https://www.tawil.team/nick-el-tawil/',
 'contact': {'twitter': 'nickeltawil'},
 'photos': {'count': 2,
  'items': [{'id': '5e41a7ce346b7c00085b0f70',
    'createdAt': 1581361102,
    'source': {'name': 'Foursquare for iOS',
     'url': 'https://foursquare.com/download/#/iphone'},
    'prefix': 'https://fastly.4sqi.net/img/general/',
    'suffix': '/484542633_cF12vc8JQJTd-hyznst-CsHHTdSoi4XwCGKhTuQXNfo.jpg',
    'width': 1440,
    'height': 1440,
    'visibility': 'public',
    'venue': {'id': '5ab194451ffed736e329b365',
     'name': 'Tawil & Team',
     'location': {'address': '413 W 14th St # 200',
      'crossStreet': '9th Avenue',
      'lat': 40.739614,
      'lng': -74.005877,
      'labeledLatLngs': [{'label': 'display',
        'lat': 40.739614,
        'lng': -74.005877}],
      'postalCode': '10014',
      'cc': 'US',
      'city': 'New York',
      'state': 'NY',
```

```
      'country': 'United States',
      'formattedAddress': ['413 W 14th St # 200 (9th Avenue)',
       'New York, NY 10014',
       'United States']},
     'categories': [{'id': '5032885091d4c4b30a586d66',
       'name': 'Real Estate Office',
       'pluralName': 'Real Estate Offices',
       'shortName': 'Real Estate',
       'icon': {'prefix':
'https://ss3.4sqi.net/img/categories_v2/shops/realestate_',
        'suffix': '.png'},
       'primary': True}],
     'like': False}}]},
 'checkinPings': 'off',
 'pings': False,
 'type': 'user',
 'mayorships': {'count': 0, 'items': []},
 'checkins': {'count': 1, 'items': []},
 'lists': {'count': 2,
  'groups': [{'type': 'created', 'count': 0, 'items': []},
   {'type': 'followed', 'count': 0, 'items': []},
   {'type': 'yours',
    'count': 2,
    'items': [{'id': '484542633/todos',
      'name': "Nick's Saved Places",
      'description': '',
      'type': 'todos',
      'editable': False,
      'public': True,
      'collaborative': False,
      'url': '/nickeltawil/list/todos',
      'canonicalUrl': 'https://foursquare.com/nickeltawil/list/todos',
      'listItems': {'count': 1}},
     {'id': '484542633/venuelikes',
      'name': 'Nick’s Liked Places',
      'description': '',
      'type': 'likes',
      'editable': False,
      'public': True,
      'collaborative': False,
      'url': '/nickeltawil/list/venuelikes',
      'canonicalUrl': 'https://foursquare.com/nickeltawil/list/venuelikes',
      'listItems': {'count': 0}}]}]},
 'blockedStatus': 'none',
 'lenses': []}
```

[43]:
```
# 1. grab prefix of photo
# 2. grab suffix of photo
# 3. concatenate them using the image size
Image(url='https://igx.4sqi.net/img/user/300x300/
 ↪484542633_mK2Yum7T_7Tn9fWpndidJsmw2Hof_6T5vJBKCHPLMK5OL-U5ZiJGj51iwBstcpDLYa3Zvhvis.
 ↪jpg')
```

[43]:
```
<IPython.core.display.Image object>
```

0.6 4. Explore a location

https://api.foursquare.com/v2/venues/explore?client_id=CLIENT_ID&client_secret=CLIEN

So, you just finished your gourmet dish at Ecco, and are just curious about the popular spots around the restaurant. In order to explore the area, let's start by getting the latitude and longitude values of Ecco Restaurant.

[44]:
```
latitude = 40.715337
longitude = -74.008848
```

Define URL

[45]:
```
url = 'https://api.foursquare.com/v2/venues/explore?
 ↪client_id={}&client_secret={}&ll={},{}&v={}&radius={}&limit={}'.
 ↪format(CLIENT_ID, CLIENT_SECRET, latitude, longitude, VERSION, radius, LIMIT)
url
```

[45]:
```
'https://api.foursquare.com/v2/venues/explore?client_id=ERDVFRWHSSDZC04NV1XRDYED
CZUTKO4XZNCDZLALI4WPAH2I&client_secret=BPJ5DALE0QS4ZNXWUI43DDHY2BDJU0YKDJNWOMYUD
U2M5CJZ&ll=40.715337,-74.008848&v=20180604&radius=500&limit=30'
```

Send GET request and examine results

[46]:
```
import requests
```

[47]:
```
results = requests.get(url).json()
'There are {} around Ecco restaurant.'.
 ↪format(len(results['response']['groups'][0]['items']))
```

[47]:
```
'There are 30 around Ecco restaurant.'
```

Get relevant part of JSON

[48]:
```
items = results['response']['groups'][0]['items']
items[0]
```

[48]:
```
{'reasons': {'count': 0,
  'items': [{'summary': 'This spot is popular',
    'type': 'general',
```

```
    'reasonName': 'globalInteractionReason'}]},
 'venue': {'id': '4af5d65ff964a52091fd21e3',
  'name': 'Korin',
  'location': {'address': '57 Warren St',
   'crossStreet': 'Church St',
   'lat': 40.71482437714839,
   'lng': -74.00940425461492,
   'labeledLatLngs': [{'label': 'display',
     'lat': 40.71482437714839,
     'lng': -74.00940425461492},
    {'label': '?', 'lat': 40.714727, 'lng': -74.009399}],
   'distance': 73,
   'postalCode': '10007',
   'cc': 'US',
   'neighborhood': 'Tribeca',
   'city': 'New York',
   'state': 'NY',
   'country': 'United States',
   'formattedAddress': ['57 Warren St (Church St)',
    'New York, NY 10007',
    'United States']},
  'categories': [{'id': '4bf58dd8d48988d1f8941735',
    'name': 'Furniture / Home Store',
    'pluralName': 'Furniture / Home Stores',
    'shortName': 'Furniture / Home',
    'icon': {'prefix':
'https://ss3.4sqi.net/img/categories_v2/shops/furniture_',
     'suffix': '.png'},
    'primary': True}],
  'photos': {'count': 0, 'groups': []},
  'venuePage': {'id': '33104775'}},
 'referralId': 'e-0-4af5d65ff964a52091fd21e3-0'}
```

Process JSON and convert it to a clean dataframe

[49]:
```
dataframe = json_normalize(items) # flatten JSON

# filter columns
filtered_columns = ['venue.name', 'venue.categories'] + [col for col in␣
 ↪dataframe.columns if col.startswith('venue.location.')] + ['venue.id']
dataframe_filtered = dataframe.loc[:, filtered_columns]

# filter the category for each row
dataframe_filtered['venue.categories'] = dataframe_filtered.
 ↪apply(get_category_type, axis=1)

# clean columns
```

```
dataframe_filtered.columns = [col.split('.')[-1] for col in dataframe_filtered.
 ↪columns]

dataframe_filtered.head(10)
```

```
/home/jupyterlab/conda/envs/python/lib/python3.6/site-
packages/ipykernel_launcher.py:1: FutureWarning: pandas.io.json.json_normalize
is deprecated, use pandas.json_normalize instead
  """Entry point for launching an IPython kernel.
```

[49]:

	name	categories	address	crossStreet	lat	lng	labeledLatLngs
0	Korin	Furniture / Home Store	57 Warren St	Church St	40.714824	-74.009404	[{'label': 'display', 'lat': 40.71482437714839, 'lng': -74.00940425461492}, {'label': '?', 'lat': 40.714727, 'lng': -74.009399}]
1	Los Tacos No. 1	Taco Place	136 Church St	NaN	40.714267	-74.008756	[{'label': 'display', 'lat': 40.714267, 'lng': -74.008756}]
2	Takahachi Bakery	Bakery	25 Murray St	at Church St	40.713653	-74.008804	[{'label': 'display', 'lat': 40.713652845301894, 'lng': -74.0088038953017}]
3	Chambers Street Wines	Wine Shop	148 Chambers St	btwn West Broadway & Hudson St	40.715773	-74.009718	[{'label': 'display', 'lat': 40.715773063928374, 'lng': -74.00971823312332}]
4	Juice Press	Vegetarian / Vegan Restaurant	83 Murray St	btwn Greenwich St & W Broadway	40.714788	-74.011132	[{'label': 'display', 'lat': 40.71478769908051, 'lng': -74.0111317502157}]
5	Philip Williams Posters	Antique Shop	122 Chambers St	NaN	40.715284	-74.008781	[{'label': 'display', 'lat': 40.71528423132827, 'lng': -74.00878093952018}]
6	Heyday	Spa	92 Reade St	NaN	40.715726	-74.007767	
7	Takahachi	Sushi Restaurant	145 Duane St	btwn W Broadway & Church St	40.716526	-74.008101	
8	Equinox Tribeca	Gym	54 Murray St	at W Broadway	40.714099	-74.009686	
9	Whole Foods Market	Grocery Store	270 Greenwich Street	at Warren St	40.715579	-74.011368	

```
6
[{'label': 'display', 'lat': 40.715726, 'lng': -74.007767}]
7                                                    [{'label': 'display',
'lat': 40.71652647412374, 'lng': -74.00810108466207}]
8                                                     [{'label': 'display',
'lat': 40.71409860726041, 'lng': -74.0096857179283}]
9                                                   [{'label': 'display',
'lat': 40.715579155420606, 'lng': -74.01136823958119}]

   distance postalCode  cc neighborhood      city state        country  \
0        73      10007  US      Tribeca  New York    NY  United States
1       119      10007  US          NaN  New York    NY  United States
2       187      10007  US          NaN  New York    NY  United States
3        88      10007  US          NaN  New York    NY  United States
4       202      10007  US          NaN  New York    NY  United States
5         8      10007  US          NaN  New York    NY  United States
6       100      10013  US          NaN  New York    NY  United States
7       146      10013  US          NaN  New York    NY  United States
8       154      10007  US          NaN  New York    NY  United States
9       214      10007  US      Tribeca  New York    NY  United States

formattedAddress  \
0                          [57 Warren St (Church St), New York, NY 10007, United
States]
1                                  [136 Church St, New York, NY 10007, United
States]
2                      [25 Murray St (at Church St), New York, NY 10007, United
States]
3  [148 Chambers St (btwn West Broadway & Hudson St), New York, NY 10007, United
States]
4     [83 Murray St (btwn Greenwich St & W Broadway), New York, NY 10007, United
States]
5                                  [122 Chambers St, New York, NY 10007, United
States]
6                                     [92 Reade St, New York, NY 10013, United
States]
7        [145 Duane St (btwn W Broadway & Church St), New York, NY 10013, United
States]
8                     [54 Murray St (at W Broadway), New York, NY 10007, United
States]
9               [270 Greenwich Street (at Warren St), New York, NY 10007, United
States]

                         id
0  4af5d65ff964a52091fd21e3
1  5d5f24ec09484500079aee00
2  4c154c9a77cea593c401d260
```

```
3  4adcf23cf964a520cc6221e3
4  54148bc6498ea7bb8c05b70a
5  4b747291f964a52042dd2de3
6  57ad129c498e05b086594d72
7  4a8f2f39f964a520471420e3
8  4a6e331af964a52031d41fe3
9  49bc3b0af964a52020541fe3
```

Let's visualize these items on the map around our location

[50]:
```
venues_map = folium.Map(location=[latitude, longitude], zoom_start=15) #␣
↪generate map centred around Ecco


# add Ecco as a red circle mark
folium.features.CircleMarker(
    [latitude, longitude],
    radius=10,
    popup='Ecco',
    fill=True,
    color='red',
    fill_color='red',
    fill_opacity=0.6
    ).add_to(venues_map)


# add popular spots to the map as blue circle markers
for lat, lng, label in zip(dataframe_filtered.lat, dataframe_filtered.lng,␣
↪dataframe_filtered.categories):
    folium.features.CircleMarker(
        [lat, lng],
        radius=5,
        popup=label,
        fill=True,
        color='blue',
        fill_color='blue',
        fill_opacity=0.6
        ).add_to(venues_map)

# display map
venues_map
```

[50]:
```
<folium.folium.Map at 0x7fefba2900f0>
```

0.7 5. Explore Trending Venues

`https://api.foursquare.com/v2/venues/trending?client_id=CLIENT_ID&client_secret=CLIE`

Now, instead of simply exploring the area around Ecco, you are interested in knowing the venues that are trending at the time you are done with your lunch, meaning the places with the highest foot traffic. So let's do that and get the trending venues around Ecco.

[51]:
```
# define URL
url = 'https://api.foursquare.com/v2/venues/trending?
 ↪client_id={}&client_secret={}&ll={},{}&v={}'.format(CLIENT_ID,␣
 ↪CLIENT_SECRET, latitude, longitude, VERSION)

# send GET request and get trending venues
results = requests.get(url).json()
results
```

[51]:
```
{'meta': {'code': 200, 'requestId': '5e8ae1839da7ee001bd95006'},
 'response': {'venues': []}}
```

0.7.1 Check if any venues are trending at this time

[52]:
```
if len(results['response']['venues']) == 0:
    trending_venues_df = 'No trending venues are available at the moment!'

else:
    trending_venues = results['response']['venues']
    trending_venues_df = json_normalize(trending_venues)

    # filter columns
    columns_filtered = ['name', 'categories'] + ['location.distance', 'location.
 ↪city', 'location.postalCode', 'location.state', 'location.country',␣
 ↪'location.lat', 'location.lng']
    trending_venues_df = trending_venues_df.loc[:, columns_filtered]

    # filter the category for each row
    trending_venues_df['categories'] = trending_venues_df.
 ↪apply(get_category_type, axis=1)
```

[53]:
```
# display trending venues
trending_venues_df
```

[53]:
```
'No trending venues are available at the moment!'
```

Now, depending on when you run the above code, you might get different venues since the venues with the highest foot traffic are fetched live.

0.7.2 Visualize trending venues

[54]:
```
if len(results['response']['venues']) == 0:
    venues_map = 'Cannot generate visual as no trending venues are available at␣
↪the moment!'

else:
    venues_map = folium.Map(location=[latitude, longitude], zoom_start=15) #␣
↪generate map centred around Ecco


    # add Ecco as a red circle mark
    folium.features.CircleMarker(
        [latitude, longitude],
        radius=10,
        popup='Ecco',
        fill=True,
        color='red',
        fill_color='red',
        fill_opacity=0.6
    ).add_to(venues_map)


    # add the trending venues as blue circle markers
    for lat, lng, label in zip(trending_venues_df['location.lat'],␣
↪trending_venues_df['location.lng'], trending_venues_df['name']):
        folium.features.CircleMarker(
            [lat, lng],
            radius=5,
            poup=label,
            fill=True,
            color='blue',
            fill_color='blue',
            fill_opacity=0.6
        ).add_to(venues_map)
```

[55]:
```
# display map
venues_map
```

[55]:
```
'Cannot generate visual as no trending venues are available at the moment!'
```

0.7.3 Thank you for completing this lab!

This notebook was created by Alex Aklson. I hope you found this lab interesting and educational. Feel free to contact me if you have any questions!

This notebook is part of a course on **Coursera** called *Applied Data Science Capstone*. If you accessed this notebook outside the course, you can take this course online by clicking here.