

# Google Places API

This Notebook will show scripts of downloading jsons from Google Places. First, I will import the requests package and assign a variable to the place search (Wilson Library). Next I will do a "find place" with the variable. Finally, I will do a query to get to that same place. json downloads are included for each. I will know if it worked if I get a Place ID.

## Importing a Package and Creating a Variable

The requests package is the only package needed for the notebook. The place I will try to find is Wilson Library in West Bank. Since the find place and query have different separators. I put in the string of the library as a list.

```
In [1]: import requests
```

```
In [2]: library = ['O.', 'Meredith', 'Wilson', 'Library']  ##This is what I will try to find
```

```
In [3]: library # A list will allow me to use different separators.
```

```
Out[3]: ['O.', 'Meredith', 'Wilson', 'Library']
```

## Find Place

The main components of the find place are the input, input type, and key. The input will be Wilson Library (needs to be exact), the input type will be "text query", and the key will be from my google account.

```
In [4]: sep = "%20" ##This is the separator needed for find place
```

```
In [5]: inp = sep.join(library) ## creates the correct format.
```

```
In [6]: inp
```

```
Out[6]: 'O.%20Meredith%20Wilson%20Library'
```

```
In [7]: key = "AIzaSyArVS9-PEcAjocuZy7aUWApsD3_loo-5g0"  ##This is the key needed.

r = requests.get("https://maps.googleapis.com/maps/api/place/findplacefromtext/json?input=" + inp + "&inputtype=textquery&key=" + key)

#Base, input, input type, and key.

r.content

#The result is a json with the place ID.
```

```
Out[7]: b'{"candidates" : [{"place_id" : "ChIJS_44o0EtslIRrVGaAtTsTPA"}], "status" : "OK"}'
```

```
In [8]: open("Library.csv", "wb").write(r.content) ##This returns the json as a csv.
```

```
Out[8]: 117
```

It worked! I was able to get the place id from the json.

## Query Place

These next few cells will show a place query for Wilson Library. It is similar to find place except it does not need to be exact and therefore can get multiple results.

```
In [9]: sep2 = "+" ##Separator needed for query.
```

```
In [10]: inp2 = sep2.join(library)
```

```
In [11]: key = "AIzaSyArVS9-PEcAjocuZy7aUWApsD3_loo-5g0" ##next doing a textsearch.

r = requests.get("https://maps.googleapis.com/maps/api/place/textsearch/json?input=" + inp2 + "&key=" + key)

## Base, input, and key.

r.content

##The resulting json is more but the place id is the same.
```

```
Out[11]: b'{"html_attributions" : [], "results" : [{"business_status" : "OPERATIONAL", "formatted_address" : "309 19th Ave S, Minneapolis, MN 55455, United States", "geometry" : {"location" : {"lat" : 44.9709073, "lng" : -93.24349169999999}, "viewport" : {"northeast" : {"lat" : 44.97179177989273, "lng" : -93.24195402010729}, "southwest" : {"lat" : 44.96909212010728, "lng" : -93.24465367989272}}, "icon" : "https://maps.gstatic.com/mapfiles/place_api/icons/v1/png_71/library-71.png", "name" : "O. Meredith Wilson Library", "opening_hours" : {}, "photos" : [], "height" : 1366, "html_attributions" : [{"href" : "https://maps.google.com/maps/contrib/102856734615321129920", "text" : "O. Meredith Wilson Library"}], "photo_reference" : "ATtYBwKR4aAHf9E47FOepraLtj5YPBTiQMYG58TLJmfcVuM7PYb8PNUZanFw4HDSH40NlIg0ezBb2g-q5YI0NK3Zlgn8SVMNvibvTmbve7TCd4JK_GzllkmSiJGWrVdURgb8uNvefBGn7VeTukHbGLIGj65oQVZeyYWIpn doRggdEK7zt0N", "width" : 2048}, {"place_id" : "ChIJS_44o0Ets1IRrVGaAtTsTPA", "plus_code" : {"compound_code" : "XQC4+9J Minneapolis, Minnesota", "global_code" : "86P8XQC4+9J"}, "rating" : 4.6, "reference" : "ChIJS_44o0Ets1IRrVGaAtTsTPA", "types" : [ "library", "university", "point_of_interest", "establishment" ], "user_ratings_total" : 23}, {"status" : "OK"}]'
```

```
In [12]: open("Library2.csv", "wb").write(r.content) ##This retrieves the json as a csv.
```

```
Out[12]: 1808
```

It worked! I was able to get the same place id as before. This concludes the notebook.