

[Get started](#)[Open in app](#)[Follow](#)

589K Followers



You have 2 free member-only stories left this month. [Sign up for Medium and get an extra one](#)

Foods Around Me: Google Maps Data Scraping with Python & Google Colab



Regita H. Zakia · Jan 18, 2020 · 6 min read ★

I have always been that person who reaches for my smartphone once a question struck my mind; faster when it relates to food. This is what I experienced when I went to Bali last December. It was on peak season — of course, wrong choice of time. As a foodie I am, food is the first thing I was thinking of when I arrived. We all know Bali for its Ayam Betutu, Sate Lilit, and Lawar — yet I didn't know where to get them, or to be exact, I didn't know what places are best to get them.

“If only I had the complete list of all restaurants nearby with its rating and address...” said the little voice in my head.

Once I came back from the vacay, I rushed to my laptop. Finding out how to make that “if only....” real. The answer is — of course — data scraping. Data scraping, or web scraping, is importing data from website into our local machine. The output is usually in form of spreadsheet. So, I could get the complete list of restaurants around me, with its address & ratings, in a simple spreadsheet! Sounds great, I think.

I wrote a simple Python script in my Notepad since I didn't have Python 3 installed in my (office) laptop just yet. It's not a secret that installing Python can be quite handy, especially for office laptop with so many security settings installed in it. My friend, Mahendri, who also helped me proofreading my script, suggested Google Colab to run the script. It allows me to run the Python script on the cloud, so I didn't need to install Python 3 on my laptop beforehand. I just need to write my code in Google Colab, press 'Run', and it's done.

Since my goal is to get the complete list of places, then Google Maps data is the answer. With Google Maps data, we get the place's name, address, coordinate, type of

place, rating, phone number, and other basic data. Of course, for the starter, we need to get the Places API. Once we have the API, we could proceed to scrape Places data.

The first step: What data would you like to get?

For me, I would like to get **restaurants** around me (chilling by the beach in **Sanur, Bali**) in radius of **1 km**. So, the parameters would be 'restaurant', 'Sanur Beach' (in coordinate), and '1 km'.

Translated into Python, it would be:

```
coordinates = ['-8.705833, 115.261377']
keywords = ['restaurant']
radius = '1000'
api_key = 'acbhsjbfQur2y8r' #insert your API key here
```

The 'keywords' let us get any places that are registered as restaurants OR whose name has 'restaurant' in it. So, I think it's better compared to using 'name' or 'type' of places because I can get complete list of places that the type and the name contain 'restaurant'. For example, I can get Sushi Tei and Se'i Sapi Restaurant at the same time. If we use 'name', we'll only get places whose name has 'restaurant' in it. If we use 'type', we'll only get places whose type is 'restaurant'. However, the downside of using 'keywords' is it will take a longer time to clean the data.

Second step: Prepare some libraries needed, such as:

```
import pandas as pd, numpy as np
import requests
import json
import time
from google.colab import files
```

Did you notice the "from google.colab import files"? Yes, using Google Colab requires us to use google.colab library to open or save data.

Third step: Craft a code that generates data based on your parameters in step 1.

```
for coordinate in coordinates:
    for keyword in keywords:

        url = 'https://maps.googleapis.com/maps/api/place/nearbysearch/json?
        location='+coordinate+'&radius='+str(radius)+'&keyword='+str(keyword)
       +'&key='+str(api_key)

        while True:
            print(url)
            respon = requests.get(url)
            jj = json.loads(respon.text)
```

```

results = jj['results']
for result in results:
    name = result['name']
    place_id = result ['place_id']
    lat = result['geometry']['location']['lat']
    lng = result['geometry']['location']['lng']
    rating = result['rating']
    types = result['types']
    vicinity = result['vicinity']

    data = [name, place_id, lat, lng, rating, types, vicinity]
    final_data.append(data)

time.sleep(5)

if 'next_page_token' not in jj:
    break
else:

    next_page_token = jj['next_page_token']

url = 'https://maps.googleapis.com/maps/api/place/nearbysearch/json?
key='+str(api_key)+'&pagetoken='+str(next_page_token)

labels = ['Place Name', 'Place ID', 'Latitude', 'Longitude', 'Types',
'Vicinity']

```

The code allows us to get the place's name, id, latitude-longitude, rating, type, and vicinity for every coordinate and keyword. Since Google only shows 20 points of data in each page, we have to insert 'next_page_token' to scrape the next page's data. Let's say there are 40 restaurants around Sanur, then Google will show the data in two pages. If there are 58, then it would be three pages.

The maximum data point we can extract is only 60 places. It's a rule from Google. An example, there are 140 restaurants around Sanur in the radius of 1 km from where I stand. It means that only 60 of 140 restaurants will be generated. So, to prevent discrepancies, we should control our radius and coordinate efficiently. Make sure the radius is not too wide, which leads to "only 60 points are generated meanwhile there are lots of them". Also, make sure the radius is not too small, which leads us to list down lots of coordinates. Both would not be efficient, hence we need to understand the context of the location beforehand.

Fourth step: Save the data to our local machine

```

export_dataframe_1_medium = pd.DataFrame.from_records(final_data,
columns=labels)
export_dataframe_1_medium.to_csv('export_dataframe_1_medium.csv')

```

Last step: Knitting all steps above into one complete code

```

import pandas as pd, numpy as np
import requests
import json
import time
final_data = []

```

```

# Parameters
coordinates = ['-8.705833, 115.261377']
keywords = ['restaurant']
radius = '1000'
api_key = 'acbhsjbfQur2y8r' #insert your Places API

for coordinate in coordinates:
    for keyword in keywords:

url = 'https://maps.googleapis.com/maps/api/place/nearbysearch/json?
location='+coordinate+'&radius='+str(radius)+'&keyword='+str(keyword)
+'&key='+str(api_key)

while True:
    print(url)
    respon = requests.get(url)
    jj = json.loads(respon.text)
    results = jj['results']
    for result in results:
        name = result['name']
        place_id = result['place_id']
        lat = result['geometry']['location']['lat']
        lng = result['geometry']['location']['lng']
        rating = result['rating']
        types = result['types']
        vicinity = result['vicinity']

data = [name, place_id, lat, lng, rating, types, vicinity]
final_data.append(data)

time.sleep(5)

if 'next_page_token' not in jj:
    break
else:

next_page_token = jj['next_page_token']

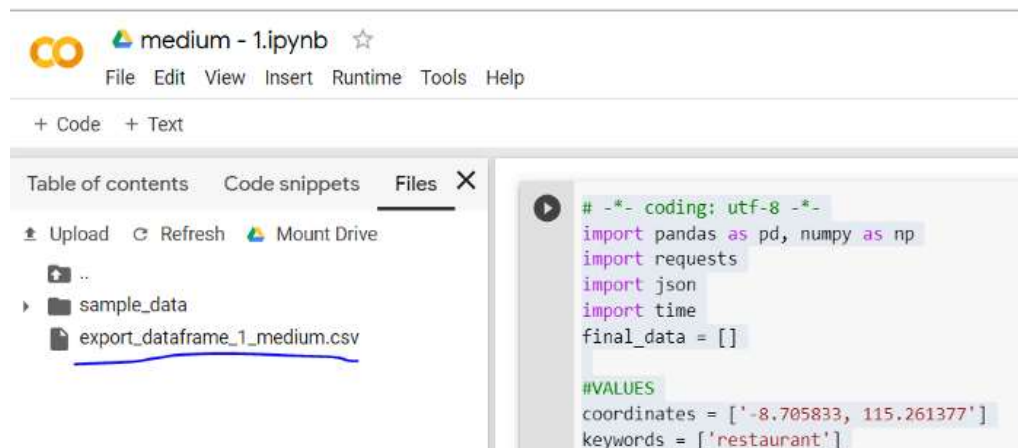
url = 'https://maps.googleapis.com/maps/api/place/nearbysearch/json?
key='+str(api_key)+'&pagetoken='+str(next_page_token)

labels = ['Place Name', 'Place ID', 'Latitude', 'Longitude', 'Types',
'Vicinity']

export_dataframe_1_medium = pd.DataFrame.from_records(final_data,
columns=labels)
export_dataframe_1_medium.to_csv('export_dataframe_1_medium.csv')

```

Now, easily download your data from the Google Colab files. Click the arrow button on your left-pane, click 'Files', then download your data! :)



```
radius = '1000'
```

Your data, which is saved in CSV format, can be visualized using any tools that you're familiar with! It can be R, Python, Tableau, etc. In this case, I visualized it using **Kepler.gl**; a data agnostic, WebGL empowered, high-performance web application for geospatial analytic visualizations. Oh, it's the description from the website, btw.

This is how my data looks like in spreadsheet:

A	B	C	D	E	F	G	H	I	J	K	L	M
	Place Name	Place ID	Latitude	Longitude	Rating	Types	Vicinity					
0	Beach House Sanur	ChIJodVYK	-8.70696	115.2626	4.3	['restaurant	Semawang Beach, Jl. Kusuma Sari No.11, Sanur, Kota Denpasar					
1	Gong Restaurant Sanur	ChIJ9Z5U5	-8.69933	115.2573	4.6	['restaurant	Jl. Tirta Akasa, Sanur Kauh, Kota Denpasar					
2	Massimo Italian Restaurant	ChIJpRptSc	-8.70467	115.2606	4.6	['restaurant	Jl. Danau Tamblingan No.228, Sanur, Kota Denpasar					
3	Dusk Blue Restaurant	ChIJZ61ZlC	-8.70309	115.2622	4.5	['restaurant	Jl. Duyung No.3, Sanur, Kota Denpasar					
4	Mezzanine Bar & Restaurant	ChIJexHj8r	-8.70784	115.2578	4.5	['restaurant	Jl. Cemara No.35, Sanur, Kota Denpasar					
5	Graze Bali	ChIJsYg2Ll	-8.6977	115.2633	4.6	['restaurant	Jl. Danau Tamblingan, Sanur, Kota Denpasar					
6	Canang Kitchen N Bar	ChIJJaVz-hs	-8.70486	115.2606	4.4	['restaurant	Jl. Danau Tamblingan No.212, Sanur, Kota Denpasar					
7	Warung Odah Oning	ChIJxSuDn	-8.70571	115.2607	4.3	['restaurant	Jl. Kusuma Sari No.4, Sanur, Kota Denpasar					
8	Little Mars	ChIJmR1n4	-8.70696	115.2588	4.6	['cafe', 'restaurant	Jl. Cemara No.27, Sanur, Kota Denpasar					
9	Charming Restaurant	ChIJWx9Gi	-8.70342	115.2613	4.4	['restaurant	Jl. Danau Tamblingan No.97, Sanur, Kota Denpasar					
10	Mermaid Bay Restaurant	ChIJrZ-Ay8	-8.70636	115.2632	4.4	['restaurant	Jalan Setapak, Sanur, South Denpasar, Sanur, Kota Denpasar					
11	Apa Kabar Restaurant	ChIJFxTdl8	-8.70363	115.2608	4.4	['restaurant	Jl. Danau Tamblingan No.190, Sanur, Kota Denpasar					
12	Yimo Korean Restaurant	ChIJeTH0il	-8.70483	115.2609	4.3	['restaurant	Jl. Danau Tamblingan No.204, Sanur, Kota Denpasar					
13	KOKOYA Japanese Restaurant	ChIJiTYoHi	-8.69853	115.2629	4.3	['restaurant	Jl. Danau Tamblingan No.178, Sanur, Kota Denpasar					
14	Basilico Italian Pacolion & Bar - P	ChIJZ_v7hj	-8.71084	115.256	4.4	['restaurant	Jl. Danau Tamblingan, Sanur Kauh, Kota Denpasar					
15	Warung Beach Breeze	ChIJdyBqJl	-8.70762	115.2562	4.5	['restaurant	Jl. Cemara No No.72BB, Sanur, Kota Denpasar					
16	Nelayan Sanur Bay	ChIJR6CPp	-8.70524	115.2638	4.4	['restaurant	Jl. Setapak, Sanur, Kota Denpasar					
17	Kanda Restaurant	ChIJT_UYil	-8.69731	115.2581	4.4	['restaurant	Jl. Kanda No.8A, Sanur, Kota Denpasar					
18	Malaika Secret Moksha	ChIJHwz2K	-8.7056	115.2557	4.4	['bakery', 'restaurant	Jl. Danau Poso No.68, Sanur Kauh, Kota Denpasar					
19	MD Warung	ChIJE2W3l	-8.69982	115.2584	4.6	['restaurant	Jl. Betngandang I No.3a, Sanur Kauh, Kota Denpasar					
20	Selada Restaurant	ChIJ09IDJl	-8.70724	115.2563	4.1	['restaurant	Jl. Sudamala No.17, Sanur Kauh, Kota Denpasar					
21	Cafe Jepun	ChIJrOfvfcI	-8.7051	115.2606	4.5	['restaurant	Jalan Danau Tamblingan No.212, Semawang, Sanur, Denpasar Sela					
22	Warung Baby Monkeys	ChIJi2OxJl	-8.70737	115.2563	4.6	['cafe', 'restaurant	Jl. Sudamala No.6, Sanur, Kota Denpasar					
23	Barramundi Restaurant	ChIJ86dBC	-8.70436	115.2642	4	['restaurant	Sanur, Denpasar City					
	export_dataframe_1_medium											

And, this is how it looks like in Kepler.gl map:



Yes, that's 59 restaurants around where I stand, chilling by the beach in Sanur. And yes, just add the name and rating in the map, then we're good to explore some foods around us!

This is my first published writing in Medium. Shout out to Swandika Gumilang for teaching me in Python & explaining some logics behind this script, Deo Dedianto & Aldo Leofiro for helping me proofreading my code, also to Mahendri M. D. for suggesting Google Colab to run the script & finding the bugs in it.

If you are willing to collaborate, need some assistance, or wanted to replicate this project, please kindly contact my [Linkedin](#) or [Upwork](#).

Sign up for The Variable

By Towards Data Science

Every Thursday, the Variable delivers the very best of Towards Data Science: from hands-on tutorials and cutting-edge research to original features you don't want to miss. [Take a look.](#)

Get this newsletter

[Google Cloud Platform](#)

[Google Maps](#)

[Python](#)

[Google Colab](#)

[Google Places Api](#)

[About](#) [Write](#) [Help](#) [Legal](#)

Get the Medium app

