



# Weather\_API using Python

I want to express something more different way which i done this project previously i done extracting raw data set from Kaggle doing cleaning and EDA part in excel , Python, SSMS but this time doing different one which is API fetching weather data in python by using API (Application program interface) API acts as a bridge between two applications when there is need

This is the public weather API which i fetched the data why i choose this API is because it has open API key which can access the data most API are locked. kindly click on below link to see the website

I will explain the steps how could i fetched the data point 1 visit the website

[One Call API 3.0 - OpenWeatherMap](#)

point 2 kindly sign up with your credentials and create your account to get the API key

kindly read the API doc to know more about the website and what type of code to write to fetch the data

I tried so many queries for the best output, i will give you this one please try from your side and update me it works or not

```
from requests import Request, Session
from requests.exceptions import ConnectionError, Timeout, Too
ManyRedirects
import json
```

```

url = 'https://api.openweathermap.org/data/2.5/weather?q=LOCATION&appid=API_KEY'

parameters = {
    'start': '1',
    'limit': '50',
    'convert': 'USD'
}

session = Session()
session.headers.update(headers)

try:
    response = session.get(url, params=parameters)
    data = json.loads(response.text)
    #print(data)
except (ConnectionError, Timeout, TooManyRedirects) as e:
    print(e)
    print(data)
    -----output-----
    -----

    {'coord': {'lon': -0.1257, 'lat': 51.5085}, 'weather': [{'id': 803, 'main': 'Clouds', 'description': 'broken clouds', 'icon': '04n'}], 'base': 'stations', 'main': {'temp': 276.99, 'feels_like': 274.61, 'temp_min': 274.87, 'temp_max': 278.01, 'pressure': 1013, 'humidity': 92, 'sea_level': 1013, 'grnd_level': 1008}, 'visibility': 10000, 'wind': {'speed': 2.57, 'deg': 320}, 'clouds': {'all': 75}, 'dt': 1735787101, 'sys': {'type': 2, 'id': 2075535, 'country': 'GB', 'sunrise': 1735805158, 'sunset': 1735833770}, 'timezone': 0, 'id': 2643743, 'name': 'London', 'cod': 200}

```

To get the data in table format use below query and above one in JSON format

```
import pandas as pd

# Extracting and normalizing relevant parts of the data
df = pd.json_normalize(data, sep='_') # This flattens the nested structure

# Display the DataFrame
print(df)
```

-----output-----  
---

The data we get in separate way to understand easily

```

                                weather      bas
e  visibility \
0  [{'id': 804, 'main': 'Clouds', 'description': ... station
s      10000

           dt  timezone      id      name  cod  coord_lon  c
oord_lat  ... \
0  1737238426      10800  149155  Usa River  200      36.85
-3.3667  ...

      main_humidity  main_sea_level  main_grnd_level  wind_speed
wind_deg \
0              71              1013              883      0.78
301

      wind_gust  clouds_all  sys_country  sys_sunrise  sys_sunset
0          1.57          90          TZ  1737257675  173730190
0
```

now i extracted data just by entering location name , USA manually

rather than entering location names manually “n number” of times i used ‘for loop’ for cities which can load all cities.

here i think in a different way like getting data is good but storing same kind of data in excel makes me easy to import into SSMS

```
import pandas as pd
import os
import requests
from datetime import datetime

# List of 25 state names (or cities if needed)
states = [
    "California", "Texas", "Florida", "New York", "Illinois",
    "Pennsylvania", "Ohio", "Georgia", "North Carolina", "Michigan",
    "New Jersey", "Virginia", "Washington", "Arizona", "Massachusetts",
    "Tennessee", "Indiana", "Missouri", "Maryland", "Wisconsin",
    "Colorado", "Minnesota", "South Carolina", "Alabama", "Louisiana", "Kentucky"
]

# Assuming the weather data API setup is already correct
user_api = os.environ.get('current_weather_data', 'API_KEY')
# Replace with your API key
file_path = r"FILE PATH"

# Loop through each state
for state in states:
    # Build the complete API link
    complete_api_link =
    f"https://api.openweathermap.org/data/2.5/weather?q={state}&appid={user_api}"
    api_link = requests.get(complete_api_link)
```

```

api_data = api_link.json()

# Check if the response is valid (status code 200)
if api_data['cod'] == 200:
    # Extracting and converting the data
    latitude = api_data['coord']['lat']
    longitude = api_data['coord']['lon']
    weather_desc = api_data['weather'][0]['description']
    temp_kelvin = api_data['main']['temp']
    temp_city_celsius = temp_kelvin - 273.15
    temp_city_fahrenheit = (temp_city_celsius * 9/5) + 32
    humidity = api_data['main']['humidity']
    wind_speed = api_data['wind']['speed']
    cloud_coverage = api_data['clouds']['all']
    pressure = api_data['main']['pressure']
    timestamp = datetime.now().strftime("%Y-%m-%d %H:%M:%S")

# Creating the data for the new row
data = {
    'Location': [state],
    'Latitude': [latitude],
    'Longitude': [longitude],
    'Weather Description': [weather_desc],
    'Temperature (°C)': [temp_city_celsius],
    'Temperature (°F)': [temp_city_fahrenheit],
    'Humidity (%)': [humidity],
    'Wind Speed (km/h)': [wind_speed],
    'Cloud Coverage (%)': [cloud_coverage],
    'Pressure (hPa)': [pressure],
    'Date & Time': [timestamp]
}

df = pd.DataFrame(data)

# Check if the file already exists to decide whether

```

```

to write the header or not
    if os.path.exists(file_path):
        # Append the data without writing the header
        df.to_csv(file_path, mode='a', header=False, index=False)
    else:
        # Write the data with the header
        df.to_csv(file_path, mode='w', header=True, index=False)

    print(f"Data for {state} saved successfully!")
else:
    print(f"Failed to get data for {state}.")

-----
-----

output look like below image with all columns and cities which we provide

```

	A	B	C	D	E	F	G	H	I	J	K
1	Location	Latitude	Longitude	Weather	Temperature	Temperature	Humidity (%)	Wind Speed	Cloud Cover	Pressure (hPa)	Date & Time
2	California	38.3004	-76.5074	clear sky	-1.17	29.894	62	2.06	0	1014	1/11/2025 21:34
3	Texas	31.2504	-99.2506	few clouds	4.97	40.946	48	3.6	20	1011	1/11/2025 21:34
4	Florida	28.7505	-82.5001	clear sky	5.84	42.512	81	2.06	0	1019	1/11/2025 21:34
5	New York	40.7143	-74.006	clear sky	-0.48	31.136	60	7.72	0	1007	1/11/2025 21:34
6	Illinois	40.0003	-89.2504	overcast clouds	-2.23	27.986	89	4.12	100	1013	1/11/2025 21:34
7	Pennsylvania	40.2724	-76.9057	overcast clouds	-0.19	31.658	60	4.12	100	1013	1/11/2025 21:34
8	Ohio	40.2503	-83.0002	clear sky	-8.55	16.61	92	0	0	1018	1/11/2025 21:34

After storing data in excel sheet imported those data to SSMS to load data in table form for cleaning and EDA part.

```

first create DataBase >> then right click on to get the TASK
in that select import flat file to load excel sheet it will create
table format under the database

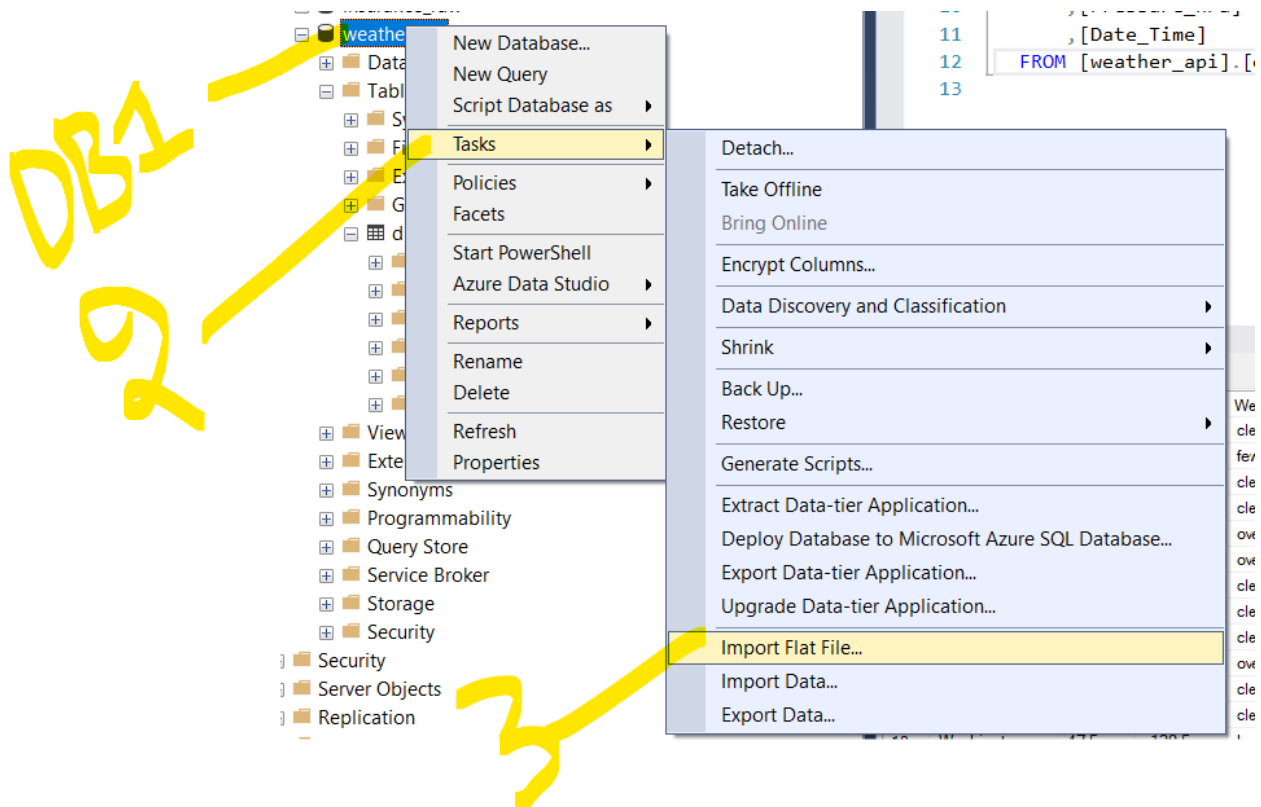
```

for cleaning part i used this one

```
SELECT Location, ROUND(Latitude, 2) AS Latitude, Longitude
FROM [weather_api].[dbo].[weather_data];
```

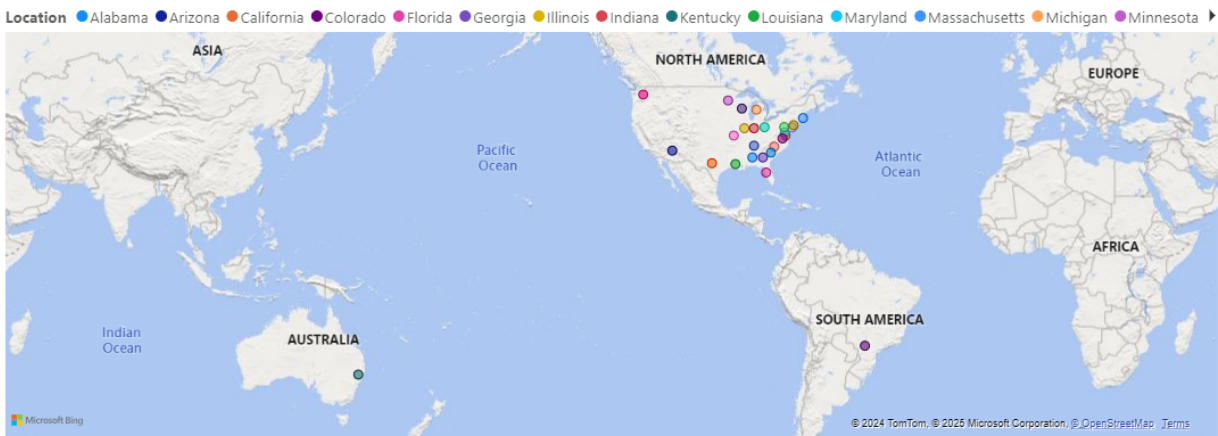
```
SELECT Location, ROUND(Longitude, 2) AS Longitude
FROM [weather_api].[dbo].[weather_data];
```

```
UPDATE [weather_api].[dbo].[weather_data]
SET Wind_Speed_kmh = ROUND(longitude, 2)
```



for visual those data for better insights to business people i used power Bi tool which i feel better for visualization and easy to use

Location, Latitude and Longitude



Total weather data based on location in table format

Date_Time	Location	Humidity	Temperature_C	Pressure_hPa	Latitude	Cloud_Coverage	Temperature_F	Weather_Description	Wind_Speed_kmh	Longitude
1/11/2025 9:34:00 PM	Arizona	39	34.50	1018	34.50	0	-111.50	clear sky	-111.50	-111.50
1/11/2025 9:34:00 PM	Texas	48	31.25	1011	31.25	20	-99.25	few clouds	-99.25	-99.25
1/11/2025 9:34:00 PM	North Carolina	59	35.50	1018	35.50	0	-80.00	clear sky	-80.00	-80.00
1/11/2025 9:34:00 PM	Pennsylvania	60	40.27	1013	40.27	100	-76.91	overcast clouds	-76.91	-76.91
1/11/2025 9:34:00 PM	New York	60	40.71	1007	40.71	0	-74.01	clear sky	-74.01	-74.01
1/11/2025 9:34:00 PM	California	62	38.30	1014	38.30	0	-76.51	clear sky	-76.51	-76.51
1/11/2025 9:34:00 PM	New Jersey	67	40.17	1009	40.17	0	-74.50	clear sky	-74.50	-74.50
1/11/2025 9:34:00 PM	Georgia	77	32.75	1019	32.75	0	-83.50	clear sky	-83.50	-83.50
1/11/2025 9:34:00 PM	Virginia	77	37.55	1016	37.55	0	-77.45	clear sky	-77.45	-77.45
1/11/2025 9:34:00 PM	Washington	79	47.50	1031	47.50	75	-120.50	broken clouds	-120.50	-120.50
1/11/2025 9:34:00 PM	Florida	81	28.75	1019	28.75	0	-82.50	clear sky	-82.50	-82.50
1/11/2025 9:34:00 PM	Michigan	86	44.25	1013	44.25	100	-85.50	overcast clouds	-85.50	-85.50
1/11/2025 9:34:00 PM	Illinois	89	40.00	1013	40.00	100	-89.25	overcast clouds	-89.25	-89.25
1/11/2025 9:34:00 PM	Ohio	92	40.25	1018	40.25	0	-83.00	clear sky	-83.00	-83.00
1/11/2025 9:35:00 PM	Missouri	31	38.25	1012	38.25	75	-92.50	broken clouds	-92.50	-92.50
1/11/2025 9:35:00 PM	Kentucky	60	-30.77	1009	-30.77	100	151.45	light rain	151.45	151.45
1/11/2025 9:35:00 PM	Maryland	65	39.00	1013	39.00	100	-76.75	overcast clouds	-76.75	-76.75
1/11/2025 9:35:00 PM	Colorado	79	-22.84	1009	-22.84	8	-51.97	clear sky	-51.97	-51.97
1/11/2025 9:35:00 PM	South Carolina	79	34.00	1018	34.00	20	-81.00	few clouds	-81.00	-81.00
1/11/2025 9:35:00 PM	Indiana	80	40.00	1016	40.00	100	-86.25	overcast clouds	-86.25	-86.25
1/11/2025 9:35:00 PM	Massachusetts	82	42.37	1003	42.37	40	-71.11	scattered clouds	-71.11	-71.11
1/11/2025 9:35:00 PM	Wisconsin	83	44.50	1008	44.50	98	-90.00	overcast clouds	-90.00	-90.00
1/11/2025 9:35:00 PM	Alabama	90	32.75	1020	32.75	6	-86.75	clear sky	-86.75	-86.75
1/11/2025 9:35:00 PM	Minnesota	90	46.25	1005	46.25	100	-94.25	mist	-94.25	-94.25
1/11/2025 9:35:00 PM	Louisiana	91	31.00	1018	31.00	3	-92.00	clear sky	-92.00	-92.00
1/11/2025 9:35:00 PM	Tennessee	93	35.75	1019	35.75	15	-86.25	few clouds	-86.25	-86.25

In Power Bi based on selected data the remaining data will affect like below image if we select location remaining data will appear data



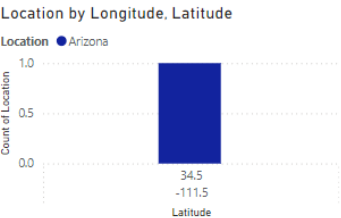
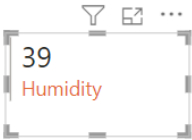
1/11/2025 9:34:00  
PM

weather

clear sky

- Location
- Alabama
  - Arizona
  - California
  - Colorado

wind speed  
-111.50



Temperature based on location

