

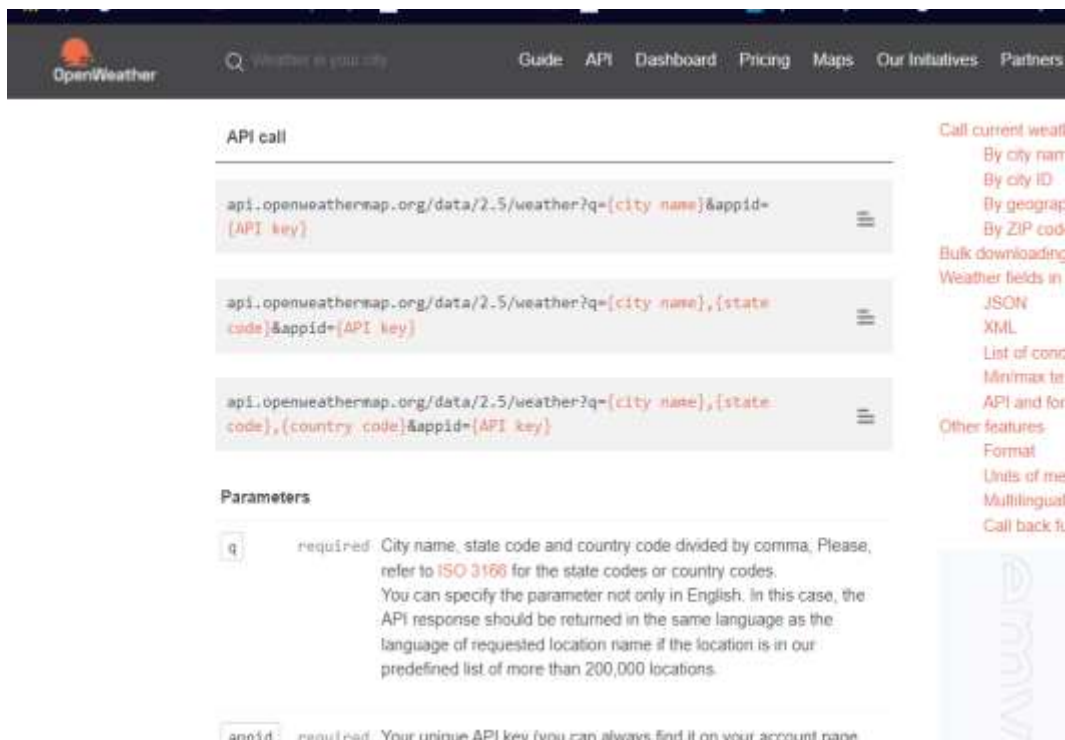
1.INTRODUCTION

In this Python project, you will learn to write a python app that will collect weather information such as current temperature, pressure, humidity, wind speed, weather description and many others, of any place on the earth, using OpenWeatherMap API. This project is going to be very short and simple. You need to invest 30 min to understand everything. You can find these weather information in real time by just typing the city name You will be able to get the data about only those places which are listed by OpenWeatherMap website. The inaccuracy of forecasting is due to the chaotic nature of the atmosphere, the massive computational power required to solve the equations that describe the atmosphere, the land, and the ocean, the error involved in measuring the initial conditions, and an incomplete understanding of atmospheric and related processes. Hence, forecasts become less accurate as the difference between current time and the time for which the forecast is being made (the range of the forecast) increases. The use of ensembles and model consensus help narrow the error and provide confidence level in the forecast.

2.WORKING

2.1 Importing Modules

- Creating account on OpenWeatherMap and API key
- Accessing confidential information from windows environment
- Creating API request from Python
- Data Interpretation
- Display the Result



The screenshot shows the OpenWeatherMap API documentation page. The header includes the OpenWeather logo, a search bar, and navigation links: Guide, API, Dashboard, Pricing, Maps, Our Initiatives, and Partners. The main content is divided into two columns. The left column, titled 'API call', shows three example API endpoints with expandable details (indicated by a hamburger menu icon):
1. `api.openweathermap.org/data/2.5/weather?q={city name}&appid={API key}`
2. `api.openweathermap.org/data/2.5/weather?q={city name},{state code}&appid={API key}`
3. `api.openweathermap.org/data/2.5/weather?q={city name},{state code},{country code}&appid={API key}`
Below these is the 'Parameters' section, which defines the 'q' parameter as required, representing city name, state code, and country code, and the 'appid' parameter as required, representing the unique API key. The right column contains a list of links for various API features: Call current weather, By city name, By city ID, By geographical coordinates, By ZIP code, Bulk downloading, Weather fields in JSON, XML, List of conditions, Min/max temperature, API and for other features, Format, Units of measurement, Multilingual, and Call back function.

2.2 Requirements

- Python version 3.6 and above (although lower version should also work)
- Python requests module should be installed
- Free account on OpenWeatherMap and api access token



3. IMPLEMENTATION

CODING

```
import tkinter as tk
```

```
import requests
```

```
import time
```

```
def getWeather(canvas):
```

```
    city = textField.get()
```

```
    api="https://api.openweathermap.org/data/2.5/weather?q="+city+"&appid=0  
6c921750b9a82d8f5d1294e1586276f"
```

```
    json_data = requests.get(api).json()
```

```
    condition = json_data['weather'][0]['main']
```

```
    temp = int(json_data['main']['temp'] - 273.15)
```

```
    min_temp = int(json_data['main']['temp_min'] - 273.15)
```

```
    max_temp = int(json_data['main']['temp_max'] - 273.15)
```

```
    pressure = json_data['main']['pressure']
```

```
    humidity = json_data['main']['humidity']
```

```
wind = json_data['wind']['speed']
```

```
sunrise = time.strftime('%I:%M:%S', time.gmtime(json_data['sys']['sunrise']  
- 21600))
```

```
sunset = time.strftime('%I:%M:%S', time.gmtime(json_data['sys']['sunset']  
- 21600))
```

```
final_info = condition + "\n" + str(temp) + "°C"
```

```
final_data = "\n" + "Min Temp: " + str(min_temp) + "°C" + "\n" +  
"MaxTemp: " + str(max_temp) + "°C" + "\n" + "Pressure: " + str(pressure) +  
"\n" + "Humidity: " + str(humidity) + "\n" + "Wind Speed: " + str(wind) + "\n" +  
"Sunrise: " + sunrise + "\n" + "Sunset: " + sunset
```

```
label1.config(text = final_info)
```

```
label2.config(text = final_data)
```

```
canvas = tk.Tk()
```

```
canvas.geometry("600x500")
```

```
canvas.title("Weather App")
```

```
f = ("poppins", 15, "bold")
```

```
t = ("poppins", 35, "bold")
```

```
textField = tk.Entry(canvas, justify='center', width = 20, font = t)
```

```
textField.pack(pady = 20)
```

```
textField.focus()
```

```
textField.bind('<Return>', getWeather)
```

```
label1 = tk.Label(canvas, font=t)
```

```
label1.pack()
```

```
label2 = tk.Label(canvas, font=f)
```

```
label2.pack()
```

```
canvas.mainloop()
```

XML

<current>

<city id="0" name="Mountain View">

<coord lon="-122.09" lat="37.39" />

<country>US</country>

<timezone>-28800</timezone>

<sun rise="2020-01-07T15:22:59" set="2020-01-08T01:05:37" />

</city>

<temperature value="278.07" min="273.15" max="282.59" unit="kelvin" />

<feels_like value="275.88" unit="kelvin" />

<humidity value="86" unit="%" />

<pressure value="1026" unit="hPa" />

<wind>

<speed value="0.93" unit="m/s" name="Calm" />

<gusts />

<direction value="23" code="NNE" name="North-northeast" />

</wind>

<clouds value="1" name="clear sky" />

<visibility value="16093" />

<precipitation mode="no" />

<weather number="800" value="clear sky" icon="01n" />

<lastupdate value="2020-01-07T11:33:40" />

</current>

Example of daily forecast weather API response

"dt":1406080800,

"temp":{

 "day":297.77, //daily averaged temperature

 "min":293.52, //daily min temperature

 "max":297.77, //daily max temperature

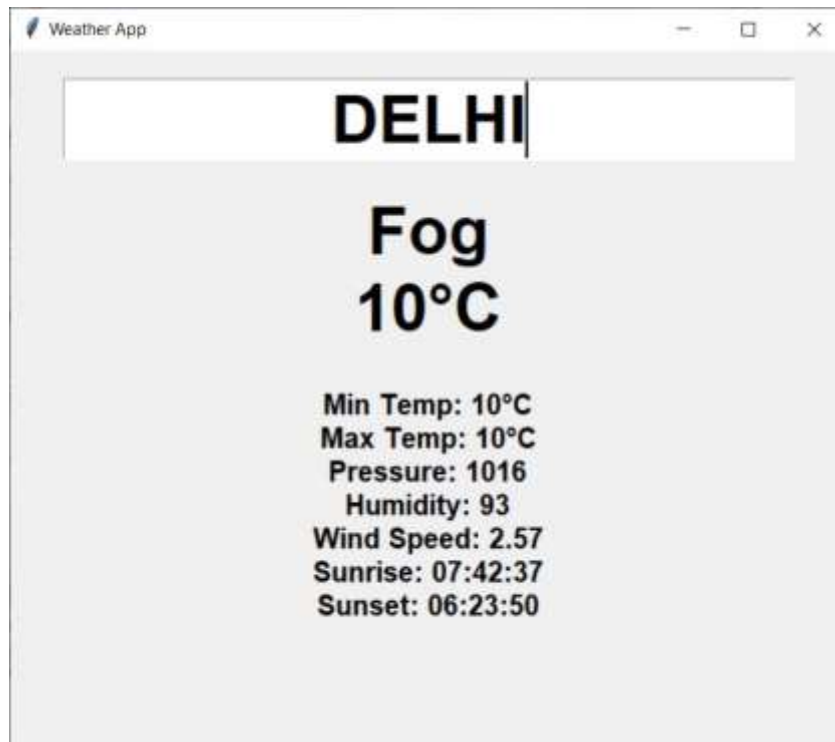
 "night":293.52, //night temperature

 "eve":297.77, //evening temperature

 "morn":297.77}, //morning temperature

4. RESULT





5. CONCLUSION

- Weather plays a major role in our daily life, and without the meteorologist and forecaster we would have difficulty planning our daily activities.
- Even with all these equipment, data, and observation tools, the weather continues to be a topic to study because it is constantly changing.
- Since outdoor activities are severely curtailed by heavy rain, snow and wind chill, forecasts can be used to plan activities around these events, and to plan ahead and survive them.
- There is a vast variety of end uses to weather forecasts. Weather warnings are important forecasts because they are used to protect life and property.

7. REFERENCES

<https://pypi.org/project/requests/>

<https://openweathermap.org/>

Dirmeyer, Paul A.; Schlosser, C. Adam; Brubaker, Kaye L. (February 1, 2009). "[Precipitation, Recycling, and Land Memory: An Integrated Analysis](#)" (PDF). *Journal of Hydrometeorology*. **10** (1):278288. [Bibcode:2009JHyMe..10..278D](#).

[University of California](#) Museum of Paleontology. "[Aristotle \(384-322 B.C.E.\) Archived](#) November 20, 2016, at the [Wayback Machine](#)". Retrieved January 12, 2008.