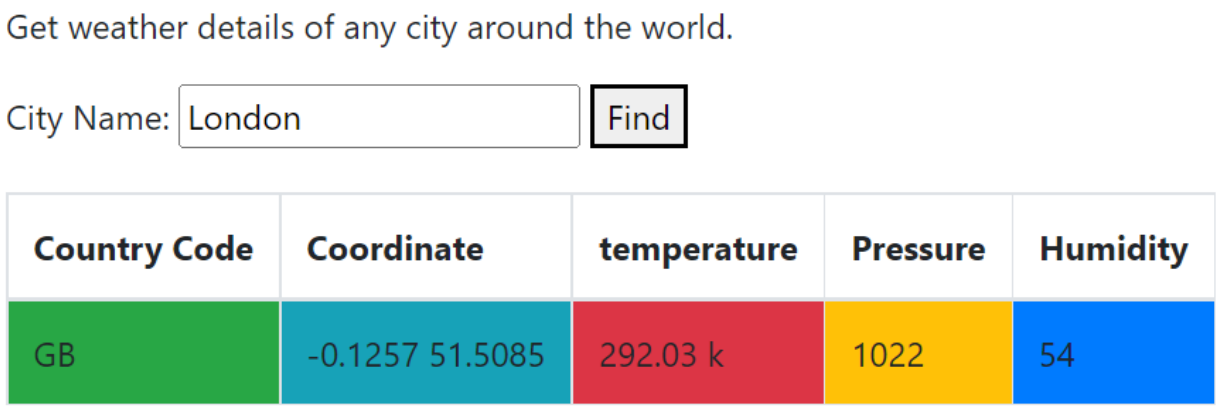
Develop Weather App using Flask in Python

webdamn.com/develop-weather-app-using-flask-in-python

webdamn May 21, 2022

In our previous Python tutorial, we have explained how to **create your first web app in Python**. In this tutorial, we will explain how to develop **Weather App in Python**.

The weather application are created to provides details of weather of any location. Here we will create a weather app using **Python and Flask** to provide current weather details of any city along with temperature.



In this tutorial, we will use **Open Weather Map API** to get weather details.

So let's proceed with developing Weather App in Python.

Modules Required

- **Flask**: It is a lightweight WSGI web application framework that used to create web applications using Python. It can be installed using the below command:

  ```
  pip install Flask
  ```

- **Requests**: We need to install this module. The module allows to make HTTP requests using Python. It can be installed using the below command:

  ```
  pip install requests
  ```

Get API Key

As we will use third party Weather API to get weather details, so we need to visit OpenWeatherMap to get API key to use in this application.

Steps to generate an API key:

- Login to Open Weather Map
- Then go to the ***My API Keys*** section. Then click ***API Keys***section and there you will get **default API key**. You can also create new API keys.
- Then go to ***API*** section and get the API link to use in your script ***api.openweathermap.org/data/2.5/weather?q={city name}&appid={API key}***

Develop Weather App

After installing required modules and getting API Keys, we will create `app.py` Python file.

Then we will import `Flask` and `requests` modules. We will also use Flask's helper `request` and `render_template` for posts data and rendering template.

```
from flask import Flask, request, render_template
import requests
```

Then we will create an instance of the flask application.

```
app = Flask(__name__)
```

Then we will create route. We will handle here both `Get` and `POST` requests.

```
@app.route('/', methods =["GET", "POST"])
def index():
```

Now we will make a GET HTTP request to openweathermap API to get weather JSON data. Here we are getting city name from form post as we will allow users to enter city to find weather details.

```
if request.method == "POST":
        cityName = request.form.get("cityName")
        weatherApiKey = 'Your API Key Goes Here'
        url = "https://api.openweathermap.org/data/2.5/weather?
q="+cityName+"&appid=" + weatherApiKey
        weatherData = requests.get(url).json()
```

Now we call `render_template` function to call template file and passed weather JSON data to render.

```
return render_template('index.html', data = weatherData, cityName = cityName,
error = error)
```

Here is complete `app.py` file.

```
from flask import Flask, request, render_template
import requests

app = Flask(__name__)

@app.route('/', methods =["GET", "POST"])
def index():
    weatherData = ''
    error = 0
    cityName = ''
    if request.method == "POST":
        cityName = request.form.get("cityName")
        if cityName:
            weatherApiKey = 'Your API Key Goes Here'
            url = "https://api.openweathermap.org/data/2.5/weather?
q="+cityName+"&appid=" + weatherApiKey
            weatherData = requests.get(url).json()
        else:
            error = 1
    return render_template('index.html', data = weatherData, cityName = cityName,
error = error)
if __name__ == "__main__":
    app.run()
```

As we are using HTML Form for city weather search and displaying details in template file. So we will create a folder `templates` in project directory and create `index.html` file.

Then we will create following HTML to create FORM and also render weather data passed after API call.

Here is complete `index.html` file.

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-
fit=no">
<title>Weather App using Flask in Python</title>
 <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.1/dist/css/bootstrap.min.css">
</head>
<body>
        <div class="container">
            <br><br><br>
                <div class="row"><h2>Weather App using Flask in Python</h2></div>
                <br>
                <div class="row">
                        <p>Get weather details of any city around the world.</p>
                </div>
                <div class="row">
                        {% block content %}
                                <form action="{{ url_for("index")}}"
method="post">
                                <div class="form-group">
                                <label for="cityName">City Name:</label>
                                <input type="text" id="cityName" name="cityName"
value="{{cityName}}" placeholder="City Name">
                                <button class="submit">Find</button>
                                {% if error is defined and error %}
                                        <br><br><span class="alert alert-
danger">Error: Please enter valid city name.</span></br>
                                {% endif %}
                                </div>
                        {% endblock %}
                        {% if data is defined and data %}
                        <table class="table table-bordered">
                                <thead>
                                        <tr>
                                                <th>Country Code</th>
                                                <th>Coordinate</th>
                                                <th>temperature</th>
                                                <th>Pressure</th>
                                                <th>Humidity</th>
                                        </tr>
                                </thead>
                                <tbody>
                                        <tr>
                                                <td class="bg-success">{{
data.sys.country }}</td>
                                                <td class="bg-info">
{{data.coord.lon }} {{data.coord.lat}}</td>
                                                <td class="bg-danger">
{{data.main.temp }} k</td>
                                                <td class="bg-warning">
{{data.main.pressure}}</td>
                                                <td class="bg-primary">
{{data.main.humidity}}</td>
```

```
                                        </tr>
                                    </tbody>
                                </table>
                                {% endif %}
                            </div>
                    </div>
</body>
</html>
```

Output: