Documentation

Muhammad Usman Saeed

W25-PAK-INP-AI-03

Weather Data Retrieval and Storage

Date: 22 December 2024

Syed Kumail Haider

Instructor

Overview

This Python script fetches real-time weather data for a given location using the OpenWeatherMap API. It provides weather statistics such as temperature, humidity, pressure, wind speed, and more. Additionally, it allows users to store this information in a text file for future reference.

Requirements

- **Python 3.x:** The script is written in Python 3.
- **Requests library:** Used to make HTTP requests to the OpenWeatherMap API.
- **OpenWeatherMap API Key:** An API key from OpenWeatherMap is required to access weather data.

Dependencies

- **requests:** Install via pip install requests.
- **os:** Built-in library used to access environment variables.
- **datetime:** Built-in library used to handle and format date and time.

Key Features

1. **User Input for City:** The user is prompted to input the name of the city for which weather data is needed.
2. **API Integration:** The script interacts with the OpenWeatherMap API to fetch real-time weather data based on the provided location.

3. Weather Information:

- Current Temperature (in Celsius)
- Feels Like Temperature (in Celsius)
- Weather Description (e.g., clear sky, rain, etc.)
- Humidity (in %)
- Pressure (in hPa)
- Visibility (in meters)
- Wind Speed (in km/h)
- Sunrise and Sunset Times (formatted in local time)

4. Error Handling: The script handles errors in case the API calls fails (e.g., invalid city name or connection issues).

5. Optional Data Saving: The user is asked if they want to save the retrieved weather data into a text file.

Instructions

Step 1: Set up API Key

1. Obtain an API key from OpenWeatherMap.
2. Store the API key as an environment variable named current_weather_data:

```
export current_weather_data="your_api_key_here"
```

Step 2: Run the Script

1. The script will prompt the user to enter the name of the city for which weather data is desired.
2. Once the city is entered, the script will make an API request to retrieve weather data.
3. If the request is successful, the weather data will be displayed on the console in a readable format.
4. The user will be asked if they wish to save the data:
    - If 'yes', the weather data is saved to a file named {city}_weather.txt.
    - If 'no', the data is not saved and the script finishes.

Step 3: View the Saved Data

1. If the user chose to save the data, they can open the file {city}_weather.txt in any text editor to view the weather details.
2. The file contains the following:
    - The city name and the date/time when the data was retrieved.
    - The full weather data, including temperature, humidity, pressure, wind speed, etc

Code Breakdown

1. Importing Libraries

```
import requests
import os
from datetime import datetime
```

The Libraries Used:

- **requests:** For making API requests to retrieve weather data.
- **os:** For accessing environment variables (specifically, the API key).
- **datetime:** For formatting and displaying date and time.

2. Fetching the API Key

```
user_api = os.environ.get('current_weather_data')
if not user_api:
    print("Incorrect API key set. The environment variable
'current_weather_data' needs to be set.")
    exit()
```

The script retrieves the API key from the environment variable. If the key is missing, an error is displayed and the script exits.

3. User Input for City Name

```
location = input("Enter the city name:")
```

The user is prompted to input the name of the city for which they want the weather data.

4. Making the API Request

```
complete_api_link =
f"https://api.openweathermap.org/data/2.5/weather?q={locati
on}&appid={user_api}"
api_link = requests.get(complete_api_link)
```

An API request is made to OpenWeatherMap to fetch weather data for the specified city.

5. Parsing and Displaying Weather Data

```
temp_city = api_data['main']['temp'] - 273.15
Feels_like = api_data['main']['feels_like'] - 273.15
Weather_desc = api_data['weather'][0]['description']
...
```

The script extracts relevant weather data from the API response, such as the current temperature, humidity, and wind speed, and converts temperatures from Kelvin to Celsius. The data is then displayed to the user.

6. Saving the Data to a File

```
save_data = input("Would you like to write down the weather
data? (yes/no): ").strip().lower()
if save_data == 'yes':
   with open(f"{location}_weather.txt", "w") as file:
      file.write("-----------------------------------------------------------
\n")
      file.write(f"Weather Stats for - {location.upper()} ||
{date_time}\n")
      ...
   print(f"Weather data saved to {location}_weather.txt.")
```

The user is given the option to save the weather data to a text file. If they choose 'yes', the data is written to a file named {city}_weather.txt. The file is saved in the same directory as the script.

Example Output

```
------------------------------------------------------------
Weather Stats for - NEW YORK || 22 Dec 2024 || 01:30:45 PM
------------------------------------------------------------
Current temperature : 12.34°C
Feels like temperature : 10.12°C
Current weather : clear sky
Humidity : 65%
Pressure : 1013 hPa
Visibility : 10000 meters
Wind speed : 5 km/h
Sunrise at : 07:30:15 AM
Sunset at : 04:45:22 PM
```

Conclusion

This Python script offers a simple method to retrieve and display real-time weather data. It also gives the option to store the data in a text file for future reference. It is a flexible tool that can be expanded with additional features like forecasts or data visualization as needed.