

Fetching Data Using OpenWeatherMap API

Michael Borck

Table of contents

Today	2
Review Homework	2
Introduction to OpenWeatherMap API	3
Fetching Data	3
Example - Create Function	3
Example - Use function	3
What is JSON	4
Parsing JSON Data	4
Example - parse_weather_data	4
Saving Data to a CSV File	4
Example	5
Error Handling	5
Error Handling - Example	5
Breakout Room Activity	5
Q&A and Wrap-Up	6
Homework	6

***ELECTRONIC WARNING NOTICE FOR COPYRIGHT
STATUTORY LICENCES***

WARNING

This material has been reproduced and communicated to you by or on behalf of **Curtin University** in accordance with section 113P of the *Copyright Act 1968 (the Act)*

The material in this communication may be subject to copyright under the Act. Any further reproduction or communication of this material by you may be the subject of copyright protection under the Act.

Do not remove this notice.

Today

- Emphasize the importance of understanding APIs and data fetching in Python.
- Highlight the skills that will be learned in this session:
 - Fetching data using the OpenWeatherMap API.
 - Parsing JSON data.
 - Handling errors.

Review Homework

- Review the provided notebook on basics of Python

Introduction to OpenWeatherMap API

- The OpenWeatherMap API provides weather data for various locations. You need to sign up and get an API key to access the data.
- Steps to Get API Key:
 1. Go to [OpenWeatherMap](#) and sign up for an account.
 2. After signing up, navigate to the API keys section and generate a new API key.
 3. Keep this API key safe, as you will need it to make API requests.

Fetching Data

- Demonstrate how to use the `requests` library to fetch data from the OpenWeatherMap API.
- Introduce the concept of HTTP requests and responses.
- Show how to handle errors and exceptions in Python.

Example - Create Function

```
import requests

def fetch_weather_data(api_key, location):
    url = f"http://api.openweathermap.org/data/2.5/weather?q={location}&appid={api_key}"
    response = requests.get(url)
    if response.status_code == 200:
        data = response.json()
        return data
    else:
        print(f"Failed to fetch data: {response.status_code}")
        return None
```

Example - Use function

```
api_key = 'your_api_key_here'
location = 'London'
data = fetch_weather_data(api_key, location)
data
```

What is JSON

- JavaScript Object Notation,
- JSON is a text-based format
- JSON is easy to read and write
- JSON is widely used for data exchange

Parsing JSON Data

- JSON used to represent data.
- `import json` to serialise list, dict etc to JSON
- `response.json()` convert JSON to Python dict.
- Highlight the importance of data parsing and how it is used in the weather dashboard project.

Example - parse_weather_data

```
def parse_weather_data(data):
    if data:
        weather = {
            "Location": data["name"],
            "Temperature (K)": data["main"]["temp"],
            "Humidity (%)": data["main"]["humidity"],
            "Weather": data["weather"][0]["description"]
        }
        return weather
    else:
        return None

parsed_data = parse_weather_data(data)
parsed_data
```

Saving Data to a CSV File

- CSV another format for sharing data
- We'll save the fetched data to a CSV file using pandas.
- pandas open-source library for data manipulation and analysis

Example

```
import pandas as pd

def save_to_csv(data, filename):
    df = pd.DataFrame([data]). # convert dict to df
    df.to_csv(filename, index=False)

save_to_csv(parsed_data, 'data/raw/weather_data.csv')

df = pd.read_csv('data/raw/weather_data.csv')
df
```

Error Handling

- Introduce the concept of error handling in Python and its importance.
- Demonstrate how to use try-except blocks to handle errors in Python.
- Highlight the importance of error handling in the weather dashboard project.

Error Handling - Example

```
invalid_location = 'InvalidCity'
invalid_data = fetch_weather_data(api_key, invalid_location)
invalid_data
```

Breakout Room Activity

- Emphasise the importance of teamwork and collaboration in the project.
- Encourage students to work together to fetch and parse data for different cities.
- Encourage students to share their findings and discuss challenges.

Q&A and Wrap-Up

- In this session, we learned how to fetch weather data from the OpenWeatherMap API, parse the JSON response, and save the data to a CSV file. We also covered basic error handling to manage failed API requests.
- Any questions”
- The importance of practising and experimenting with the concepts learned in the session.
- Next session, we will focus on managing data with CSV and SQL.

Homework

- Experiment with fetching weather data for different cities.
- Explore the OpenWeatherMap API documentation to see what other data you can fetch.