API Adventures: Conquering OpenWeatherMap

From API Requests to CSV Files in a Breeze

Michael Borck

Table of contents



# 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](https://home.openweathermap.org/users/sign_up) 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.