Consuming 3rd Party API and Designing RESTful API with CRUD Operations

# *Part 1: Consuming 3rd Party API and Data Validation*

## Sign Up and Get API Key from OpenWeather API

I Created a free account on <https://openweathermap.org/api> by using my email address then created a username and password. After that I received an email from OpenWeather that contains the initial api. On the cite I am able to create a new api to use for api calls. Save the key in .env file.

A screenshot of a computer

AI-generated content may be incorrect.

## Choose the Right API Endpoint For current weather data use

Visit:   
<https://openweathermap.org/current?utm_source=chatgpt.com#geo>

https://openweathermap.org/api/statistics-api#name

And copy API call from the Documentation

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

I replaced the city name London with Philadelphia and {API Key } with real API key.

## Test It in Browser Pheonix AZ’s Weather

A screenshot of a computer

AI-generated content may be incorrect.

Philadelphia PA’s Weather

A screenshot of a computer

AI-generated content may be incorrect.

## Test It in Postman

A screen shot of a computer code

AI-generated content may be incorrect.

A screenshot of a computer code

AI-generated content may be incorrect.

## 🔧 1. Install Dependencies

pip install requests pydantic pymongo python-dotenv  
A screenshot of a computer

AI-generated content may be incorrect.

A screen shot of a weather forecast

AI-generated content may be incorrect. 2. fetch\_weather.py — Fetch + Validate + Save

A screenshot of a computer

AI-generated content may be incorrect.

* It loads the API key from the environment variable OPENWEATHER\_API\_KEY using [os.getenv](vscode-file://vscode-app/c:/Users/tahir/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html).
* The API key is inserted into the [URL](vscode-file://vscode-app/c:/Users/tahir/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) string as a query parameter:

A screenshot of a computer

AI-generated content may be incorrect.

# Part 2: Designing and Implementing RESTful API with CRUD Operations

Build REST API (FastAPI + MongoDB)

Install Required Libraries  
pip install fastapi uvicorn

## Created a main.py file that will be your FastAPI server for the CRUD API.

Run the Server

uvicorn main:app –reload  
python --version

pip show uvicorn

Visit <http://127.0.0.1:8000/docs> to see the **Swagger UI** automatically generated by FastAPI.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.  
A screenshot of a computer program

AI-generated content may be incorrect.  
A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

6839f0bbf3fb6891f3a5bea2 Update the data under this ID

A screenshot of a computer code

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect. A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer code

AI-generated content may be incorrect.

A screenshot of a computer code

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

Under the Id I changed the data from San Francisco’ data to Miami’s data

Screens screenshot of a computer program

AI-generated content may be incorrect.

**✅ 2. Test GET /weather/{id} with Non-existent ID**

**❌ Case: Valid ID, but not in DB**

* Input ID: 60b8c0f6e1d1c8d1c8d1c8d1 (or any valid-looking but unused ID)
* Expected: 404 Not Found
* Message: "Weather data not found."

✅ **What to do:** Use a random valid ObjectId in Swagger or Postman.

A screenshot of a computer

AI-generated content may be incorrect.

**3. Test POST /weather with Missing Fields**

**❌ Case: Missing required fields**

A screenshot of a computer

AI-generated content may be incorrect.

**✅ 5. Test PUT /weather/{id} with Non-existent ID**

* Input: Valid ObjectId that’s not in DB
* Expected: 404 Not Found
* Message: "Weather data not found."

✅ Use the same test ID from step 2.

A screenshot of a computer program

AI-generated content may be incorrect.

Testing the URL  
 Non-existent but valid ID:  
A screenshot of a computer

AI-generated content may be incorrect.

Invalid ID format:  
A screenshot of a computer

AI-generated content may be incorrect.

Bad POST body:

A screenshot of a computer

AI-generated content may be incorrect.

How to Run the Project

# 1. Create virtual env (optional but recommended)

python -m venv venv

source venv/bin/activate # or venv\Scripts\activate on Windows

# 2. Install dependencies

pip install -r requirements.txt

# 3. Start MongoDB (locally or via Docker)

# 4. Run FastAPI

uvicorn app.main:app --reload

# 5. Run tests

pytest

"""fastapi = main web framework

uvicorn = ASGI server

pymongo = MongoDB driver, Connect and interact with MongoDB

bson = BSON data format support, Needed to work with ObjectId and BSON types manually

pytest = Testing framework, used for writing and running tests

pydantic = Data validation and settings management, Used for request and response models

httpx = HTTP client for making requests, FastAPI internally uses it; useful for client requests in tests

python"""

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer code

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

<https://code-maven.com/openweathermap-api-using-curl>

https://www.youtube.com/watch?v=vtaDaHVL\_is