**Application Programming Interface (API) Users Using API testing tool Postman to Get Weather Report.**

**Git-Hub:https://github.com/olomansaas/Postman\_WetherReport.git**

**OpenWeatherAPI API testing tool Postman Collection:**  
The "OpenWeatherAPI" API testing tool Postman collection is a carefully designed suite designed to facilitate interactions with the OpenWeatherAPI. Its structure reflects a thoughtful consideration of diverse use cases for retrieving meteorological data.

**Here are the key components:**

**Basic Steps To Follow:**

1. Launch Postman: Open Postman on your computer.

2. Create a New Request:

- Click on the 'New' button.

- Select 'Request'.

- Name your request (e.g., 'Weather Report') and optionally add it to a collection.

3. Set Up the Request:

- In the request tab, select the 'GET' method from the dropdown.

- Enter the URL of the weather API you are using (e.g., `http://api.weatherapi.com/v1/current.json?key=YOUR\_API\_KEY&q=London`).

4. Send the Request and View Response:

- Click the 'Send' button.

- The response will be displayed in the lower section of Postman.

5. Check the Response Format:

- Ensure the response is in JSON format, which is typical for APIs.

- The JSON response should contain weather data like temperature, humidity, wind speed, etc.

6. Pretty Print the Response:

- Postman automatically 'pretty prints' JSON responses for readability.

- You can see the structured JSON in an easy-to-read format.

7. Save the Request:

- Click on 'Save' to save your request for future use.

8. (Optional) Add Tests or Scripts:

- You can add tests to validate the response.

- For example, you can write a test to check if the response status code is 200.

9. Document the API (Optional):

- You can add documentation to your request in Postman for reference purposes.

**Requests:**

**a. getcitybyname:**  
Executes a GET request to obtain weather data based on the city name.  
Utilizes pre-request scripts for dynamic setup of environment and global variables.  
**b. getcityId:**  
Initiates a GET request for meteorological data based on city ID.  
**c. getWeatherLatLong:**  
Engages in a GET request to fetch weather details using latitude and longitude.  
**d. getLanguage:**  
Executes a GET request, incorporating language specifications for weather data retrieval.

**Environment Variables:**

**a. Base\_URL:**  
  
A default variable for holding the base URL of the OpenWeatherAPI.

**b. apikey\_env:**  
  
A default variable acting as a placeholder for the API key in the environment.

**c. BaseURLDemo:**  
  
Another default variable serving as an additional placeholder for the base URL.

**d. city and City:**  
  
Variables of type "any" designed for flexibility in handling city names.

**e. global\_var1:**  
  
A variable of type "any" representing a global placeholder variable.

**Summary:**  
  
The Postman collection for testing the "OpenWeatherAPI" and its integration with the "QA\_Environment" configuration exemplifies a resilient and flexible testing framework. With precisely crafted requests, adept variable handling, and considerate pre-request scripts, this collection transcends a mere sequence of API interactions to become a versatile instrument adept at handling diverse testing scenarios across different environments. The careful organization of both the collection and environment underscores a dedication to optimizing efficiency, enhancing flexibility, and ensuring user-friendly experiences within the domain of OpenWeatherAPI testing.