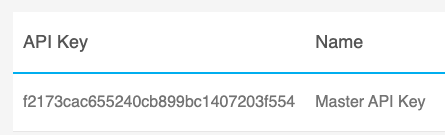
**API Automation Test (Postman)**

**Test steps:**

1. **Sign up to the Weather API for Free to Get the API Key:**- Go to <https://www.weatherbit.io/account/create> to sign up for the free version (21-day trial).

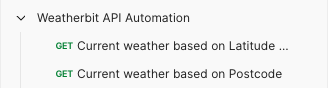
- You will receive a confirmation email; click **Activate Your Account** to access your dashboard and see your **API Key**.

Example:



1. **Create a new collection in Postman "Weatherbit API Automation":**- Open Postman, In the left sidebar, click on **Collections**.

- Click **New Collection** and name it, for example, "Weatherbit API Automation".  
- Save the collection.

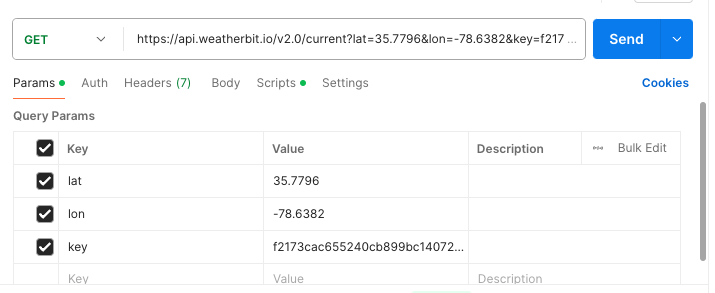


1. **Create the Lat/Lon Request, and run the request manually to validate the tests:**- In your new collection, click **Add Request**.  
   - Name the request Current weather based on Latitude and Longitude.  
   - Set the method to **GET**.  
   - Enter the following URL (replace the “YOUR\_API\_KEY” with your actual API key):  
   <https://api.weatherbit.io/v2.0/current?lat=35.7796&lon=-78.6382&key=YOUR_API_KEY>

Note:

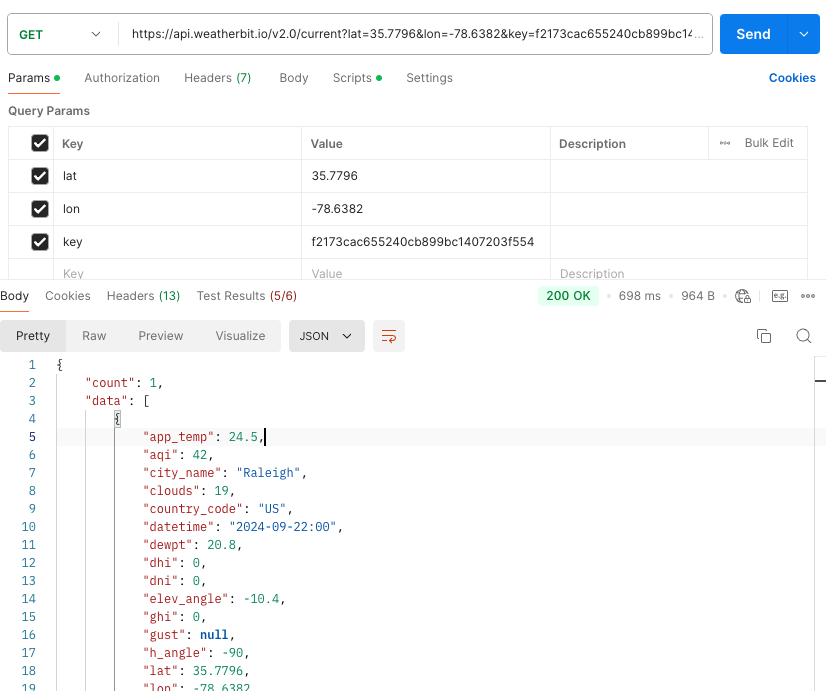
* lat: Latitude (35.7796)
* lon: Longitude (-78.6382)
* key: Your API key from Weatherbit.

- Click **save**.



- Click **Send** to manually run the API requests.

- Check the **Test Results** at the bottom of the Postman window to inspect the **response**.



1. **Create the Postcode Request, and run the request manually to validate the Tests:**

- In the same collection, click **Add Request**.

- Name the request Current weather based on Postcode.

- Set the method to **GET**.

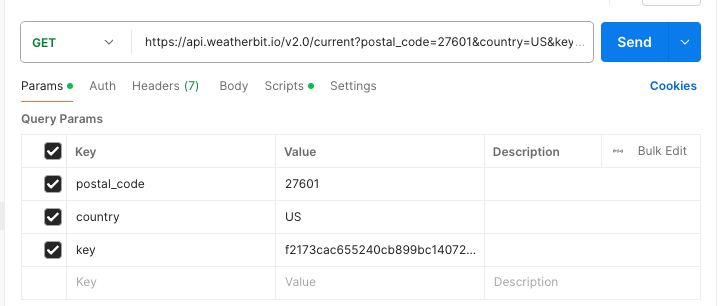
- Enter the following URL (replace the “YOUR\_API\_KEY” with your actual API key):

https://api.weatherbit.io/v2.0/current?postal\_code=27601&country=US&key=YOUR\_API\_KEY

Note:

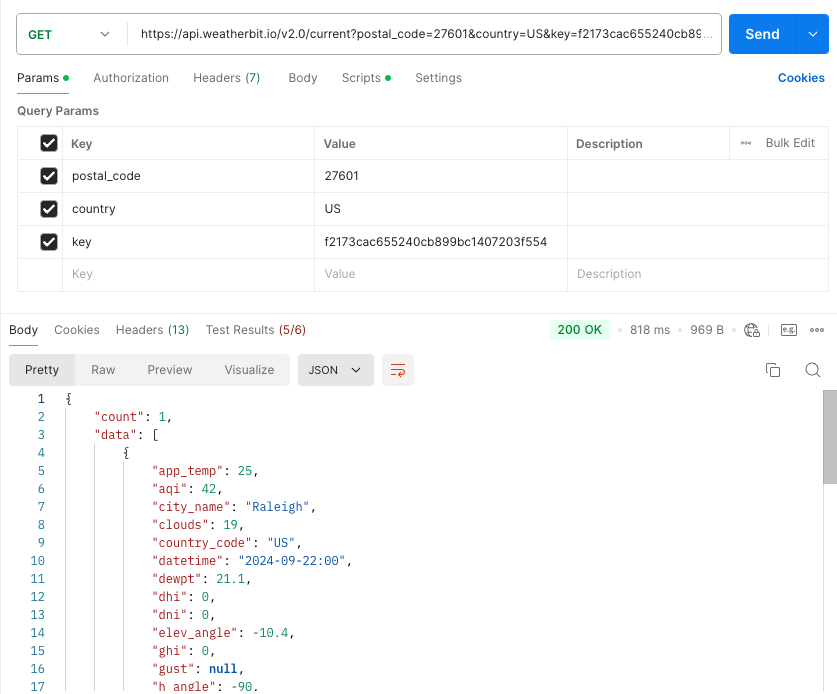
* postal\_code: 27601 (for Raleigh, NC)
* country: US
* key: Your API key from Weatherbit

- Click **Save**.



- Click **Send** to manually run the API request.

- Check the **Test Results** at the bottom of the Postman window to inspect the **response**.



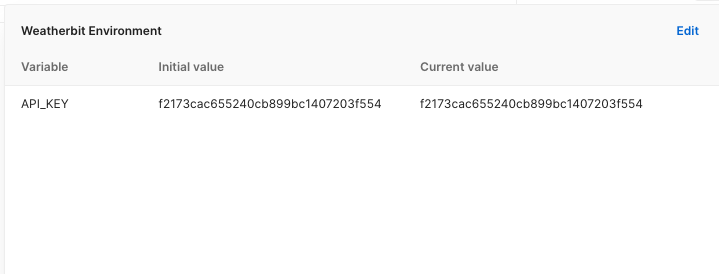
1. **Store the API Key in Postman Environment:**

- Go to **Manage Environments** in Postman.

- Click **Add** to create a new environment, name it (e.g., **Weatherbit Environment**).

- Add a variable called API\_KEY and set the value to your API key (e.g., f2173cac655240cb899bc1407203f554) and save.

- Select "Weatherbit Environment" in the environment dropdown at the top right of Postman.



1. **Add Tests for the Latitude/Longitude Request:**

- Go to the Tests tab for the Current weather based on Latitude and Longitude request.

- Update the requests to use variables for lat, lon: https://api.weatherbit.io/v2.0/current?lat={{lat}}&lon={{lon}}&key={{API\_KEY}}

- After running the requests in step 3, inspect the response to understand the structure.

- Click on **Scripts** tab.

- Write test code based on the **response** to automate the validation of the response:

Sample scripts:

// Test if the status code is 200

pm.test("Status code is 200", function () {

pm.response.to.have.status(200);

});

// Parse the response JSON

const jsonData = pm.response.json();

// Test if the city name is present

pm.test("City name is present", function () {

pm.expect(jsonData.data[0].city\_name).to.exist;

});

// Test if the temperature is a valid number

pm.test("Temperature is a valid number", function () {

pm.expect(jsonData.data[0].temp).to.be.a('number');

});

// Test if wind speed is valid

pm.test("Wind speed is valid", function () {

pm.expect(jsonData.data[0].wind\_spd).to.be.a('number');

});

// Test if the weather description is present

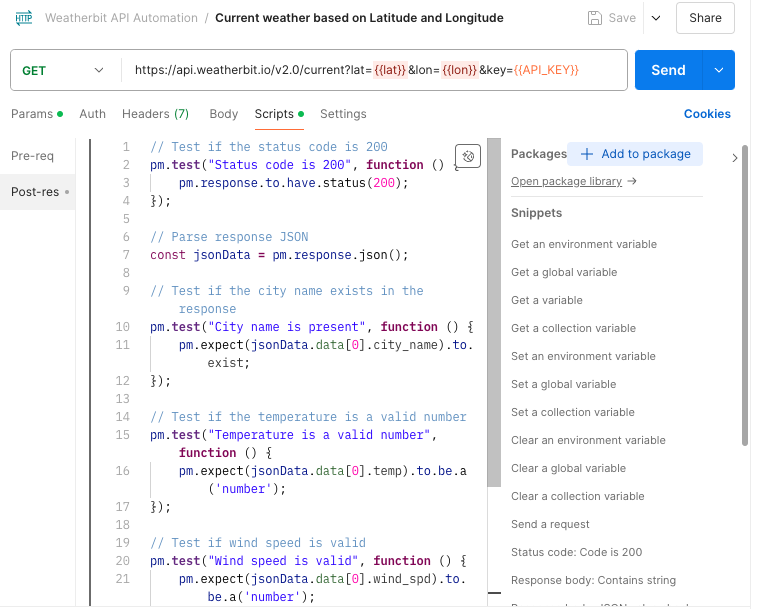
pm.test("Weather description is present", function () {

pm.expect(jsonData.data[0].weather.description).to.exist;

});

// Log the full response for debugging purposes

console.log("Full response:", jsonData);



1. **Add Tests for the Postal Code Request:**

- Go to the Tests tab for the Current weather based on Postcode request.

- Update the requests to use variables for post code:

<https://api.weatherbit.io/v2.0/current?postal_code={{postal_code}}&country=US&key={{API_KEY}}>

- After running the requests in step 4, inspect the response to understand the structure.

- Click on **Scripts** tab.

- Write test code based on the **response** to automate the validation of the response:

Sample scripts:

// Test if the status code is 200

pm.test("Status code is 200", function () {

pm.response.to.have.status(200);

});

// Parse the response JSON

const jsonData = pm.response.json();

// Test if the city name is present

pm.test("City name is present", function () {

pm.expect(jsonData.data[0].city\_name).to.exist;

});

// Test if the temperature is a valid number

pm.test("Temperature is a valid number", function () {

pm.expect(jsonData.data[0].temp).to.be.a('number');

});

// Test if wind speed is valid

pm.test("Wind speed is valid", function () {

pm.expect(jsonData.data[0].wind\_spd).to.be.a('number');

});

// Test if the weather description is present

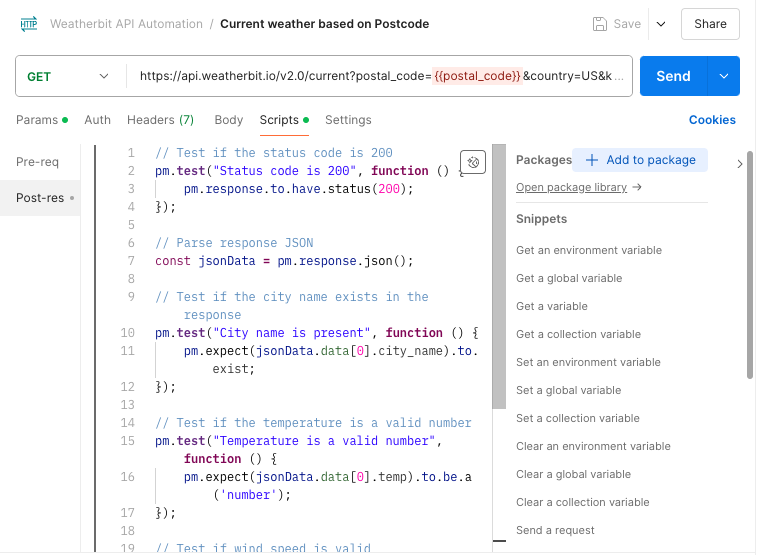
pm.test("Weather description is present", function () {

pm.expect(jsonData.data[0].weather.description).to.exist;

});

// Log the full response for debugging purposes

console.log("Full response:", jsonData);



1. **Automate the Tests for Multiple Locations:**

- Create a **CSV** file (e.g. locations.csv) with multiple lat/lon pairs and postal codes:

example:

lat,lon,postal\_code

35.7796,-78.6382,27601

34.0522,-118.2437,90001

40.7128,-74.0060,10001

-33.8688,151.2093,2000

48.8566,2.3522,75000

Note:

CSV File Mapping: Postman will replace the placeholders ({{lat}}, {{lon}}, {{postal\_code}}) in the requests with the corresponding values from the CSV file for each iteration.

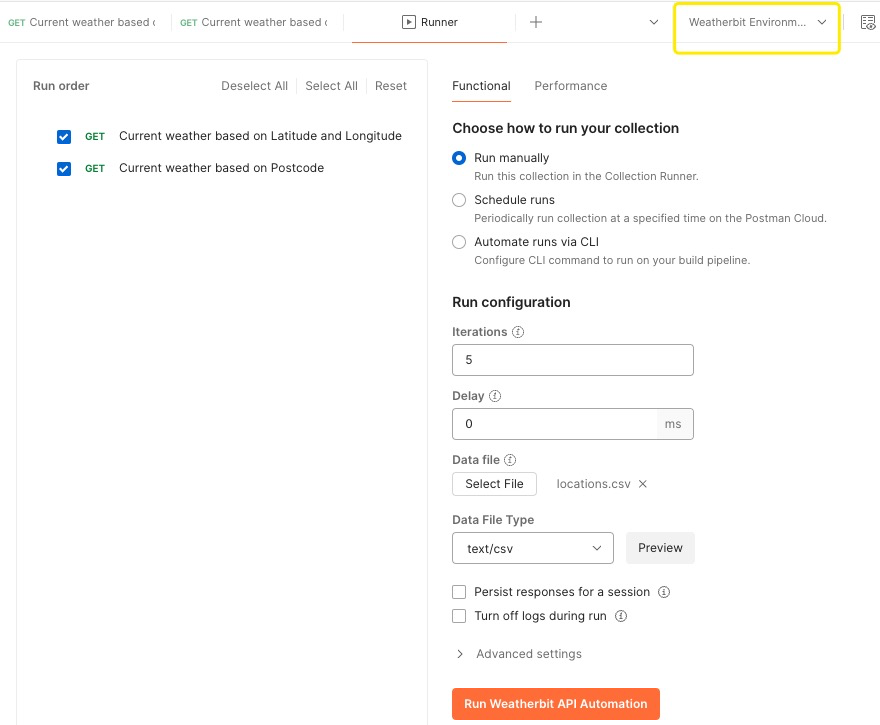
- Go to **Collection Runner** in Postman.

- Drag your Weatherbit API Automation collection into **Run Order**.

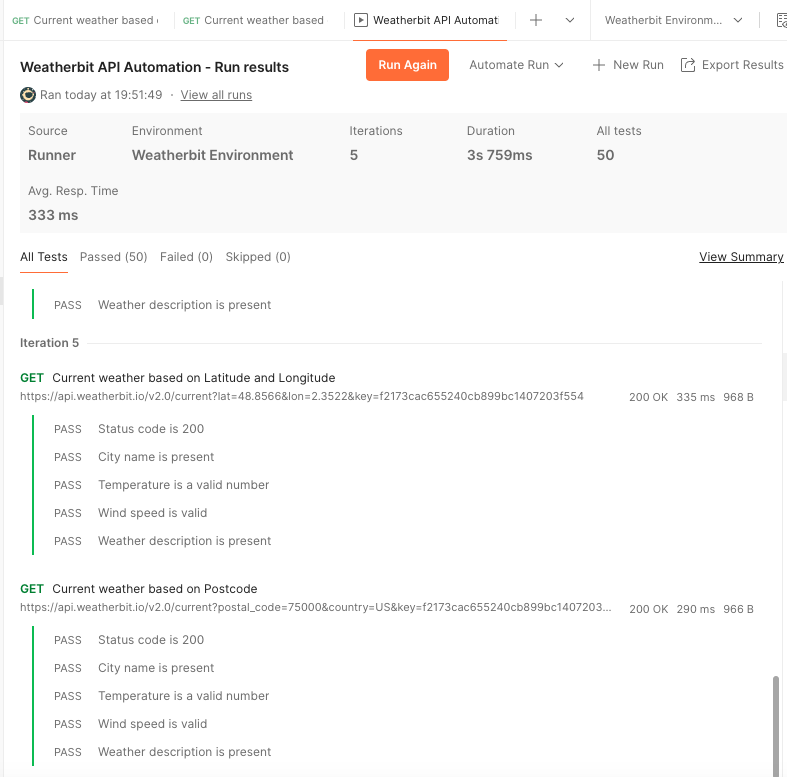
- Select both Current weather based on Latitude and Longitude and Current weather based on Postal Code files.

- Select "**Weatherbit Environment**" in the environment dropdown at the top right of Postman.

- Upload the CSV file with multiple locations.



- Click **Run** to execute the tests for each location.



1. **Generate cURL Commands:**

- For each request, click on the Code button (</>) next to the request URL.

- Select cURL from the dropdown.

- Copy the generated cURL command to use in the terminal or other tools.

