**API Manual Testing**

You are given an API endpoint for a weather service that provides current weather information for a given city.

The API endpoint is GET /api/weather/{cityName}.

The task is to write at least 3 detailed test cases to verify the correctness of the API response.

Answer:

1. **Valid City Name - Successful Response:**
   * **Test Case Description**: Send a GET request to the endpoint with a valid city name for which weather information is available.
   * **Test Steps**:
     1. Send a GET request to the endpoint **/api/weather/{cityName}** with a valid city name (e.g., "New York").
     2. Receive the API response.
   * **Expected Behavior**:
     1. The API should return a successful response with status code 200 OK.
     2. The response body should contain valid weather information for the specified city.
     3. Verify that the response contains necessary fields like temperature, humidity, weather description, etc.
   * **Pass Criteria**:
     1. Status code is 200 OK.
     2. Response body contains valid weather information.
2. **Invalid City Name - Error Handling:**
   * **Test Case Description**: Send a GET request to the endpoint with an invalid city name for which weather information is not available.
   * **Test Steps**:
     1. Send a GET request to the endpoint **/api/weather/{invalidCityName}** with an invalid city name (e.g., "InvalidCityName").
     2. Receive the API response.
   * **Expected Behavior**:
     1. The API should return an error response with an appropriate status code (e.g., 404 Not Found or 400 Bad Request).
     2. The response body should contain an error message indicating that the city name is invalid or weather information is not available.
   * **Pass Criteria**:
     1. Status code is an appropriate error code (e.g., 404 or 400).
     2. Response body contains an error message indicating the reason for failure.
3. **Temperature in Celsius for Paris**:
   * **Test Case Description**: Request the weather information for Paris with the temperature unit specified as Celsius.
   * **Test Steps**:
     1. Send a GET request to the endpoint **/api/weather/Paris?unit=metric**.
     2. Receive the API response.
   * **Expected Behavior**:
     1. The API should return a successful response with status code 200 OK.
     2. The response body should contain weather information for Paris, and temperatures should be in Celsius.
   * **Pass Criteria**:
     1. Status code is 200 OK.
     2. Response body contains weather information with temperatures in Celsius.
4. **Historical Weather Data for Dubai**:
   * **Test Case Description**: Request historical weather data for the city of Paris for a specific date.
   * **Test Steps**:
     1. Send a GET request to the endpoint **/api/weather/Dubai/history?date=2024-03-01**.
     2. Receive the API response.
   * **Expected Behavior**:
     1. The API should return a successful response with status code 200 OK.
     2. The response body should contain historical weather data for Paris on the specified date, including details such as maximum and minimum temperatures.
   * **Pass Criteria**:
     1. Status code is 200 OK.
     2. Response body contains valid historical weather data for Paris on the specified date.
5. **Paris Weather in French Language**:

* **Test Case Description**: Send a GET request to the endpoint **/api/weather/Paris** and specify the language parameter as French (e.g., **/api/weather/Paris?lang=fr**).
* **Test Steps**:
  1. Send a GET request to the endpoint **/api/weather/Paris** with the language parameter.
  2. Receive the API response.
* **Expected Behavior**:
  1. The API should return a successful response with status code 200 OK.
  2. The response body should contain weather information for Paris in the French language.
  3. Verify that the response language matches the specified language parameter.
* **Pass Criteria**:
  1. Status code is 200 OK.
  2. Response body contains weather information for Paris in French.

1. **Empty City Name - Input Validation**:
   * **Test Case Description**: Send a GET request to the endpoint with an empty city name.
   * **Test Steps**:
     1. Send a GET request to the endpoint **/api/weather/**
     2. Receive the API response.
   * **Expected Behavior**:
     1. The API should return an error response with an appropriate status code (e.g., 400 Bad Request).
     2. The response body should contain an error message indicating that the city name is required.
   * **Pass Criteria**:
     1. Status code is 400 Bad Request.
     2. Response body contains an error message indicating the missing city name.