

# Weather API

## Manuella tester

**1. Verify that the API returns the correct HTTP status code (e.g., 200 OK) for a successful GET request.**

**Result:** Passed

**Description:** The API returned a HTTP status code of 200 OK for a successful get request.

**2. Check if the API returns the expected data format (e.g., JSON, XML) in the response.**

**Result:** Passed

**Description:** The API returned data in the expected format which was JSON.

**3. Ensure that the API returns the correct HTTP status code (e.g., 400 Bad Request) for an invalid request.**

**Result:** Passed

**Description:** The API returned a HTTP status code of 400 bad request for an invalid request.

**4. Test if the API returns the correct data when querying with specific filters or search criteria.**

**Result:** Passed

**Description:** The API returned correct data when queried with specific filters or search criteria.

**5. Verify that the API returns paginated results when a large number of records are requested.**

**Result:** Passed

**Description:** The API returned paginated results when a large number of records were requested.

**6. Check if the API handles special characters and non-English text correctly in input data and returned responses.**

**Result:** Passed

**Description:** The API handled special characters and non-English text correctly in input data and returned responses.

**7. Test the API's response when sending concurrent requests to ensure that it can handle multiple users and maintain data consistency**

**Result:** Passed.

**Description:** The API was able to handle multiple users and maintain data consistency when receiving concurrent requests.

**8. Test if the API correctly handles different HTTP methods (GET, POST, PUT, DELETE) for each endpoint and returns appropriate status codes and responses for each method.**

**Result:** Passed

**Description:** The API correctly handled different HTTP methods for each endpoint and returned correct status codes and responses for each method.

**9. Check if the API correctly handles updates to existing records, ensuring that changes are saved and reflected in subsequent requests.**

**Result:** passed

**Description:** The API correctly handled updates to existing records, ensuring that changes were saved and reflected in subsequent requests.

**10. Test the API's performance under heavy load, simulating a large number of users making requests simultaneously.**

**Result:** Passed

**Description:** The API performed well on the heavy load, simulating a large number of users making requests simultaneously.

**11. Verify that the API can recover gracefully from failures, such as database connection issues without compromising data integrity.**

**Result:** Passed

**Description:** The API was able to recover gracefully from failures, such as database connection issues, without compromising data integrity.

**12. Test the API's ability to handle edge cases, such as requests with missing or invalid parameters, and ensure that appropriate error messages are returned.**

**Result:** Passed

**Description:** The API correctly handled edge cases, such as requests with missing or invalid parameters, and returned appropriate error messages.

**13. Verify that the API correctly implements rate limiting or throttling mechanisms to prevent abuse or excessive use of resources.**

**Result:** Passed

**Description:** The API correctly implemented rate limiting or throttling mechanisms to prevent abuse or excessive use of resources.

## Automatiserade tester

**1. Validate that the API returns the correct HTTP status code (e.g., 200 OK) for a successful GET request.**

**Result:** Passed

**Description:** The API returned a HTTP status code of 200 OK for a successful GET request.

**2. Verify that the API returns the expected data format (e.g., JSON, XML) in the response.**

**Result:** Passed

**Description:** The API returned data in the expected format, which was JSON.

**3. Ensure that the API returns the correct HTTP status code (e.g., 400 Bad Request) for an invalid request.**

**Result:** Passed

**Description:** The API returned a HTTP status code of 400 bad requests for an invalid request.

**4. Create an automated test that sends a request with specific filters or search criteria and checks if the API returns the correct data.**

**Result:** Passed

**Description:** An automated test was created and successfully sent a request with specific filters or search criteria and checked if the API returned the correct data.

**5. Write an automated test to verify that the API returns paginated results when a large number of records are requested.**

**Result:** Passed

**Description:** An automated test was developed and successfully verified that the API returned paginated results when a large number of records were requested.

**6. Test if the API handles special characters and non-English text correctly in input data and returns responses using an automated testing tool.**

**Result:** Passed

**Description:** An automated test was created using a texting tool and successfully verified that the API handled the special characters and non-English text correctly in input data and returned responses.

**7. Develop an automated test that sends concurrent requests to the API to ensure that it can handle multiple users and maintain data consistency.**

**Result:** Passed

**Description:** An automated test was developed and successfully sent concurrent requests to the API to ensure that it can handle multiple users and maintain data consistency.

**8. Create an automated test and test if the API correctly handles different HTTP methods (GET, POST, PUT, DELETE) for each endpoint and returns appropriate status codes and responses for each method.**

**Result:** Passed

**Description:** An automated test was created and successfully tested that the API correctly handled different HTTP methods (GETS, POST, PUT, DELETE) for each endpoint and returned appropriate status codes and responses for each method.

**9. Write an automated test to check if the API correctly handles updates to existing records, ensuring that changes are saved and reflected in subsequent requests.**

**Result:** Passed

**Description:** An automated test was created and successfully verified that the API correctly handled updates to existing records, ensuring that changes were saved and reflected in subsequent requests.

**10. Design an automated performance test that simulates a large number of users making requests simultaneously to check the API's performance under heavy load.**

**Result:** Passed

**Description:** An automated performance test was designed and successfully simulated a large number of users making requests simultaneously to check the API's performance under heavy load.

**11. Create an automated test that verifies the API can recover gracefully from failures, such as database connection issues or third-party service outages, without compromising data integrity.**

**Result:** Passed

**Description:** An automated test was created and successfully verified that the API can recover gracefully from failures, such as database connection issues or third-party service outages without compromising data integrity.

**12. Develop an automated test to handle edge cases, such as requests with missing or invalid parameters, and ensure that appropriate error messages are returned.**

**Result:** Passed

**Description:** An automated test was created and successfully tested the API's handling of edge cases, such as requests with missing or invalid parameters, and ensured that appropriate error messages were returned.

**13. Write an automated test to verify that the API correctly implements any rate limiting or throttling mechanisms to prevent abuse or excessive use of resources.**

**Result:** Passed

**Description:** An automated test was created and successfully verified that the API correctly implements rate limiting or throttling mechanisms to prevent abuse or excessive use of resources.