**Testing Manual & Guide for Automation Test   
  
  
  
  
  
Test Target: OpenWeatherMap API**

**GitHub:** [***Click Here To redirect***](https://github.com/rahulmusunuri/OpenWeatherTest.git)(https://github.com/rahulmusunuri/OpenWeatherTest.git)

**Author: Rahul Musunuri**

**Designation: Junior QA Automation**

**Company Name: Twenty7tec**

**Phone: +1-(519)-(851)-6961**

**Address:** **Huron street London Ontario N5V2E6**

**Twenty7Tec Junior Automation QA Tester Tech Test**

**Brief**

This tech test consists of using the public OpenWeatherMap API, you need to build five scenarios to display the current weather data on a simple webpage. Please access the following links to access the OpenWeatherMap website: <https://openweathermap.org/current>

The **API key for Current weather** has already been prepared for you to use within your project: **9d50450a48809637b4862bdcb125927d**

The cities to query are as follows:

1. London, GB (city id: 2643743) – English language
2. Paris, FR (city id: 2988507) – French language
3. New York, US (city id: 5128581) – English language
4. Delhi ,IN (city id: 2650225) – Hindi Language
5. Tokyo, JP (city id: 1850147) – Japanese Language

#### The results returned when each of these cities is queried by ID needs to display and returned the following information:

* Temp
* Humidity
* Description
* Temp min
* Temp max

Note results from the API will be returned in **KELVIN** by default but we would like to see them converted and displayed in **CELSIUS**.

**Output**

1. Write a series of **manual** test cases that cover the five scenarios above. Make sure you give detailed instructions for each test case (pre conditions, steps, expected results). You can use any format you want.
2. Write scripts that would **automate** the manual test cases that you see fit to be included from step 1 above. Please use any programming language/test framework of choice.
3. Once the tech test is completed and ready for submission, please push the file containing the manual test cases and the automation files to your own publicly hosted project to GitHub (you can create an account for free here if you don’t already have an account: <https://github.com/signup>).

Once you have completed your submission please email the Github link to Beverly Bryan: [beverley.bryan@twenty7tec.com](mailto:beverley.bryan@twenty7tec.com). Good luck from all the team at Twenty7tec!

**Table of Contents**

[Summary 3](#_Toc90767940)

[Details Provided 3](#_Toc90767941)

[Description 3](#_Toc90767942)

[Test Cases : Basic Tests 3](#_Toc90767943)

[1. Scenario: The APIs are working as expected with response code 200 for right Values 3](#_Toc90767944)

[2. Scenario: The APIs are working as expected with status code 401 with wrong apiKey 4](#_Toc90767945)

[3. Scenario: The APIs are working as expected with status code 404 with wrong apiKey 6](#_Toc90767946)

[4. Scenario: The APIs are working as expected with status code 400 with wrong apiKey 6](#_Toc90767947)

[Test Case: Property Value Checks 7](#_Toc90767948)

[1. Scenario: Check if the API with City Name and API Key retrieves the right response. 7](#_Toc90767949)

[2. Scenario: Temperature is not null 8](#_Toc90767950)

[3. Scenario: Minimum Temperature is not null 9](#_Toc90767951)

[4. Scenario: Maximum Temperature is not null 9](#_Toc90767952)

[5. Scenario: Check for Maximum and Minimum range of Temperature 10](#_Toc90767953)

[6. Scenario: Humidity is not null 11](#_Toc90767954)

# Summary

The OpenWeatherMap API is created to fetch the weather conditions of different regions based on many parameters such as “City Name”, “State”, “City Id”, “Country”, “Language”, “Latitude”, “Longitude”, and etc., This request to the API is only possible with the API key. The API key is used as an Authorization token to listen the response to the request.

# Details Provided

The OpenWeatherMap has provided us with some values in ordered to access the API endpoint

Values In Hand

|  |  |
| --- | --- |
| **Values In Hand** | |
| **Key** | **Value** |
| API Key | 9d50450a48809637b4862bdcb125927d |

# Description

This entire testing manual is composed of testing steps for two end points majorly

1. [http://api.openweathermap.org/data/2.5/weather?q={cityName}&appid={apiKey}](http://api.openweathermap.org/data/2.5/weather?q=%7bcityName%7d&appid=%7bapiKey%7d)
2. [http://api.openweathermap.org/data/2.5/weather?id={cityId}&lang={langCode}&appid={apiKey}](http://api.openweathermap.org/data/2.5/weather?id=%7bcityId%7d&lang=%7blangCode%7d&appid=%7bapiKey%7d)

# Test Cases : Basic Tests

## Scenario: The APIs are working as expected with response code 200 for right Values

|  |  |
| --- | --- |
| **Case** | Test with API with cityName and APIKey |
| **Requirements** | **Request URL:** http://api.openweathermap.org/data/2.5/weather?q={cityName}&appid={apiKey}  **Request Parameter Details:**  [  {  "**cityName**": "London",  "**apiKey**": "9d50450a48809637b4862bdcb125927d"  }  ] |
| **Steps** | 1. Open postman and create a request with Get as the action. 2. Insert the request URL. 3. Click on params and provide the values to the keys of cityName and apiKey given. 4. Click on Send button, to trigger the request. |
| **Expectation** | The request call should throw the response with 200 as status. |
| **Automation Check** | Automation for this scenario is done in "**BasisTests**.**cs**"" file under "**StatusCodeCheck\_CityName**" Method, with TestCase 1 |
| **Status** | Tested manually, and the result of automation test run is also positive. |

|  |  |
| --- | --- |
| **Case** | Test with API with id, lang, and APIKey |
| **Requirements** | **Request URL:** http://api.openweathermap.org/data/2.5/weather?id={cityId}&lang={langCode}&appid={apikey}  **Request Parameter Details:**  [  {  "**id**": "5128581",  “**lang**”: “en”,  "**apiKey**": "9d50450a48809637b4862bdcb125927d"  }  ] |
| **Steps** | 1. Open postman and create a request with Get as the action. 2. Insert the request URL. 3. Click on params and provide the values to the keys of id, lang, and apiKey given. 4. Click on Send button, to trigger the request. |
| **Expectation** | The request call should throw the response with 200 as status. |
| **Automation Check** | Automation for this scenario is done in "**BasisTests**.**cs**"" file under " **StatusCodeCheck\_CityId\_LangCode** " Method, with TestCase 1 |
| **Status** | Tested manually, and the result of automation test run is also positive. |

## Scenario: The APIs are working as expected with status code 401 with wrong apiKey

|  |  |
| --- | --- |
| **Case** | Test with API with cityName and wrong APIKey |
| **Requirements** | **Request URL:** http://api.openweathermap.org/data/2.5/weather?q={cityName}&appid={apiKey}  **Request Parameter Details:**  [  {  "**cityName**": "London",  "**apiKey**": "9d50450a48809637b4862bdcb125927"  }  ] |
| **Steps** | 1. Open postman and create a request with Get as the action. 2. Insert the request URL. 3. Click on params and provide the values to the keys of cityName and apiKey given. 4. Click on Send button, to trigger the request. |
| **Expectation** | The request call should throw the response with 401 as status. |
| **Automation Check** | Automation for this scenario is done in "**BasisTests**.**cs**"" file under "**StatusCodeCheck\_CityName**" Method, with TestCase 2. |
| **Status** | Tested manually, and the result of automation test run is also positive. |

|  |  |
| --- | --- |
| **Case** | Test with API with id, lang, and APIKey |
| **Requirements** | **Request URL:** http://api.openweathermap.org/data/2.5/weather?id={cityId}&lang={langCode}&appid={apikey}  **Request Parameter Details:**  [  {  "**id**": "2650225",  “**lang**”: “hi”,  "**apiKey**": "9d50450a48809637b4862bdcb125927d"  }  ] |
| **Steps** | 1. Open postman and create a request with Get as the action. 2. Insert the request URL. 3. Click on params and provide the values to the keys of id, lang, and apiKey given. 4. Click on Send button, to trigger the request. |
| **Expectation** | The request call should throw the response with 401 as status. |
| **Automation Check** | Automation for this scenario is done in "**BasisTests**.**cs**"" file under " **StatusCodeCheck\_CityId\_LangCode**" Method, with TestCase 2. |
| **Status** | Tested manually, and the result of automation test run is also positive. |

## Scenario: The APIs are working as expected with status code 404 with wrong apiKey

|  |  |
| --- | --- |
| **Case** | Test with API with cityName and wrong APIKey |
| **Requirements** | **Request URL:** http://api.openweathermap.org/data/2.5/weather?q={cityName}&appid={apiKey}  **Request Parameter Details:**  [  {  "**cityName**": "Fersared",  "**apiKey**": "9d50450a48809637b4862bdcb125927s"  }  ] |
| **Steps** | 1. Open postman and create a request with Get as the action. 2. Insert the request URL. 3. Click on params and provide the values to the keys of cityName and apiKey given. 4. Click on Send button, to trigger the request. |
| **Expectation** | The request call should throw the response with 404 as status. |
| **Automation Check** | Automation for this scenario is done in "**BasisTests**.**cs**"" file under "**StatusCodeCheck\_CityName**" Method, with TestCase 3. |
| **Status** | Tested manually, and the result of automation test run is also positive. |

## Scenario: The APIs are working as expected with status code 400 with wrong apiKey

|  |  |
| --- | --- |
| **Case** | Test with API with id, lang, and APIKey |
| **Requirements** | **Request URL:** http://api.openweathermap.org/data/2.5/weather?id={cityId}&lang={langCode}&appid={apikey}  **Request Parameter Details:**  [  {  "**id**": "18501s47",  “**lang**”: “ja”,  "**apiKey**": "9d50450a48809637b4862bdcb125927d"  }  ] |
| **Steps** | 1. Open postman and create a request with Get as the action. 2. Insert the request URL. 3. Click on params and provide the values to the keys of id, lang, and apiKey given. 4. Click on Send button, to trigger the request. |
| **Expectation** | The request call should throw the response with 400 as status. |
| **Automation Check** | Automation for this scenario is done in "**BasisTests**.**cs**"" file under " **StatusCodeCheck\_CityId\_LangCode**" Method, with TestCase 3. |
| **Status** | Tested manually, and the result of automation test run is also positive. |

# Test Case: Property Value Checks

## Scenario: Check if the API with City Name and API Key retrieves the right response.

|  |  |
| --- | --- |
| **Case** | Test with API with cityName and right APIKey |
| **Requirements** | **Request URL:** http://api.openweathermap.org/data/2.5/weather?q={cityName}&appid={apiKey}  **Request Parameter Details:**  [  {  "**cityName**": "Mountain View",  "**apiKey**": "9d50450a48809637b4862bdcb125927d"  }  ] |
| **Steps** | 1. Open postman and create a request with Get as the action. 2. Insert the request URL. 3. Click on params and provide the values to the keys of cityName and apiKey given. 4. Click on Send button, to trigger the request. |
| **Expectation** | The request call should provide the response with **id** value as **5375480** |
| **Automation Check** | Automation for this scenario is done in "**TemperatureTest**.**cs**"" file under "**CityNameResponseCheck**” Method. |
| **Status** | Tested manually, and the result of automation test run is also positive. |

## Scenario: Temperature is not null

|  |  |
| --- | --- |
| **Case** | Test with API with cityName and right APIKey |
| **Requirements** | **Request URL:** http://api.openweathermap.org/data/2.5/weather?q={cityName}&appid={apiKey}  **Request Parameter Details:**  [  {  "**cityName**": "Mountain View",  "**apiKey**": "9d50450a48809637b4862bdcb125927d"  }  ] |
| **Steps** | 1. Open postman and create a request with Get as the action. 2. Insert the request URL. 3. Click on params and provide the values to the keys of cityName and apiKey given. 4. Click on Send button, to trigger the request. |
| **Expectation** | The request call should provide the response with **temp** value in the **Main** section of the response Node. Check the value is **not null.** |
| **Automation Check** | Automation for this scenario is done in "**TemperatureTest**.**cs**"" file under "**TemperatureNullCheck**” Method. |
| **Status** | Tested manually, and the result of automation test run is also positive. |

## Scenario: Minimum Temperature is not null

|  |  |
| --- | --- |
| **Case** | Test with API with cityName and right APIKey |
| **Requirements** | **Request URL:** http://api.openweathermap.org/data/2.5/weather?q={cityName}&appid={apiKey}  **Request Parameter Details:**  [  {  "**cityName**": "Mountain View",  "**apiKey**": "9d50450a48809637b4862bdcb125927d"  }  ] |
| **Steps** | 1. Open postman and create a request with Get as the action. 2. Insert the request URL. 3. Click on params and provide the values to the keys of cityName and apiKey given. 4. Click on Send button, to trigger the request. |
| **Expectation** | The request call should provide the response with **temp\_min** value in the **Main** section of the response Node. Check the value is **not null.** |
| **Automation Check** | Automation for this scenario is done in "**TemperatureTest**.**cs**"" file under "**TemperatureMinimumNullCheck**” Method. |
| **Status** | Tested manually, and the result of automation test run is also positive. |

## Scenario: Maximum Temperature is not null

|  |  |
| --- | --- |
| **Case** | Test with API with cityName and right APIKey |
| **Requirements** | **Request URL:** http://api.openweathermap.org/data/2.5/weather?q={cityName}&appid={apiKey}  **Request Parameter Details:**  [  {  "**cityName**": "Mountain View",  "**apiKey**": "9d50450a48809637b4862bdcb125927d"  }  ] |
| **Steps** | 1. Open postman and create a request with Get as the action. 2. Insert the request URL. 3. Click on params and provide the values to the keys of cityName and apiKey given. 4. Click on Send button, to trigger the request. |
| **Expectation** | The request call should provide the response with **temp\_max** value in the **Main** section of the response Node. Check the value is **not null.** |
| **Automation Check** | Automation for this scenario is done in "**TemperatureTest**.**cs**"" file under "**TemperatureMaximumNullCheck**” Method. |
| **Status** | Tested manually, and the result of automation test run is also positive. |

## Scenario: Check for Maximum and Minimum range of Temperature

|  |  |
| --- | --- |
| **Case** | Test with API with cityName and right APIKey |
| **Requirements** | **Request URL:** http://api.openweathermap.org/data/2.5/weather?q={cityName}&appid={apiKey}  **Request Parameter Details:**  [  {  "**cityName**": "Mountain View",  "**apiKey**": "9d50450a48809637b4862bdcb125927d"  }  ] |
| **Steps** | 1. Open postman and create a request with Get as the action. 2. Insert the request URL. 3. Click on params and provide the values to the keys of cityName and apiKey given. 4. Click on Send button, to trigger the request. |
| **Expectation** | The request call should provide the response with **temp\_min, temp,** and **temp\_max** value in the **Main** section of the response Node. Check the values are **not null** and the temp value falls in between temp\_min and temp\_max with the relation as (**temp\_min <= temp <= temp\_max**). |
| **Automation Check** | Automation for this scenario is done in "**TemperatureTest**.**cs**"" file under "**TemperatureRangeCheck**” Method. |
| **Status** | Tested manually, and the result of automation test run is also positive. |

## Scenario: Humidity is not null

|  |  |
| --- | --- |
| **Case** | Test with API with cityName and right APIKey |
| **Requirements** | **Request URL:** http://api.openweathermap.org/data/2.5/weather?q={cityName}&appid={apiKey}  **Request Parameter Details:**  [  {  "**cityName**": "Mountain View",  "**apiKey**": "9d50450a48809637b4862bdcb125927d"  }  ] |
| **Steps** | 1. Open postman and create a request with Get as the action. 2. Insert the request URL. 3. Click on params and provide the values to the keys of cityName and apiKey given. 4. Click on Send button, to trigger the request. |
| **Expectation** | The request call should provide the response with **humidity** value in the **Main** section of the response Node. Check the value is **not null.** |
| **Automation Check** | Automation for this scenario is done in "**TemperatureTest**.**cs**"" file under "**HumidityNullCheck**” Method. |
| **Status** | Tested manually, and the result of automation test run is also positive. |