

**Estimated time needed**: 30 minutes

**What you will learn**

In this lab, you will integrate an external API OpenWeatherMap into a web application using JavaScript. You will learn how to fetch data asynchronously, parse JSON responses, and dynamically update the webpage based on the received information. You will understand the core concepts of API integration, asynchronous operations, DOM manipulation, and user interaction, providing a foundational understanding of web development practices for utilizing external data sources in a simple interface.

**Learning objective**

After completing this lab, you will be able to:

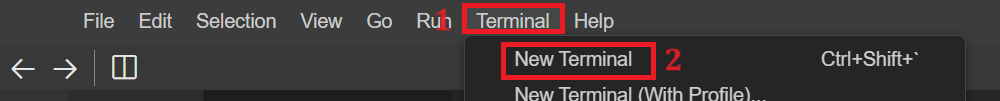
* User-friendly weather retrieval: Enable user input for city names, facilitating the retrieval of real-time weather information via an intuitive web interface.
* API integration for weather data: Utilize the OpenWeatherMap API to fetch precise weather data based on user-entered cities, dynamically displaying temperature and weather descriptions on the webpage.
* HTML form submission handling and JS event implementation: Manage form submissions within HTML and implement event listeners in JavaScript, ensuring smooth user interactions and data retrieval processes.
* Demonstration of asynchronous requests and dynamic DOM updates: Showcase the practical application of asynchronous requests using fetch(), parsing JSON responses, and dynamically updating the DOM to display fetched weather details seamlessly, eliminating the need for page refreshes.

**Prerequisites**

* Basic Knowledge of HTML.
* Web browser with a console (Chrome DevTools, Firefox Console, and so on).

**Step 1: Setting up the environment**

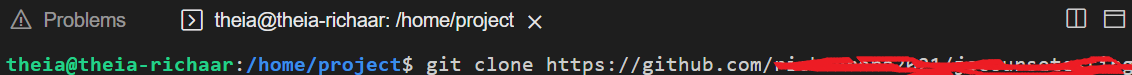
1. Firstly, you need to clone your main repository in the **Skills Network Environmemnt** which you have created in the first lab and where you have pushed all of your previous labs files and folders. Follow given steps:
   * For this click on terminal on the top-right window pane and then select **New Terminal**.



* + Perform git clone command by writing given command in the terminal.
  + 1
  + git clone <github-repository-url>

Copied!

***Note:****Put your own GitHub repository link instead of <github-repository-url>.*



* + Above step will clone folder for your GitHub repository under project folder in explorer. You will also see multiple folders inside cloned folder.
  + Now you need to navigate inside the cloned folder. For this write given command in the terminal:
  + 1
  + cd <repository-folder-name>

Copied!

A black background with red and blue text

Description automatically generated

***Note:****Write your cloned folder name instead of <repository-folder-name>. Perform git clone if you have logged out of****Skills Network Environment****and you cannot see any files or folder after you logged in.*

1. Now select **cloned Folder Name** folder, right-click on it and click on **New Folder**. Enter folder name as **weatherReport**. It will create the folder for you. Then select **weatherReport** folder, right-click and select **New File**. Enter file named as **weather\_report.html** and click OK. It will create your HTML file.
2. Now select **weatherReport** folder again, right click, and select **New File**. Enter file named as **weather\_report.js** and click OK. It will create your javascript file.
3. To fetch data from API you need to have your personal **API Key** of that particular website from where you will generate request to fetch data.
4. For this you need to open given website “[https://openweathermap.org/"](https://openweathermap.org/%22) and Signup for this website.
5. After signing up it will redirect you to the dashboard page in which you need to click on **API Keys** menu item.

A screenshot of a phone

Description automatically generated

* + Note: If it will not redirect you to above said page, then you can also access it by clicking on your profile manu item in navigation bar and then select **My API Keys** as shown in given screenshot.

A screenshot of a computer

Description automatically generated

1. After accessing page for API keys, you will navigate to page where the website has already generated a personal key for you. This key will play a vital role while fetching data from specific URL.

A screenshot of a computer

Description automatically generated

Note: Persoanl key act as password for your URL from where you will get the desired data related to weather.

1. Now you need the URL as well with the help of which data will be fetched for particular city. For this you need to click on **Services** as shown in given screenshot.  
   A group of words on a white background

   Description automatically generated
2. Then you need to click on **view**.

A group of words on a white background

Description automatically generated

1. As soon as you will click on view, it will open new window and navigate to new page. You need to scroll down little in the current page where you will see option to use API for free. From there you need to click on **Current Weather**.  
   A screenshot of a computer program

   Description automatically generated
2. Again it will open new window and navigate you to the new page. Then you need to scroll down again, where you will multiple URL links provided to you which you can use to fetch data from.
3. From the current page you need to select URL as highlited in red color. You will also use this URl in coming instructions to fetch weather related data.

A screenshot of a computer

Description automatically generated

1. An API key is like a secret passcode, and a URL is the web address where we get the information.

**Create HTML Structure**

1. Create the basic template structure of HTML file by adding content the provided content.
2. 1
3. 2
4. 3
5. 4
6. 5
7. 6
8. 7
9. 8
10. 9
11. 10
12. 11
13. 12
14. 13
15. 14
16. 15
17. 16
18. 17
19. 18
20. <!DOCTYPE html>
21. <html>
22. <head>
23. <title>Weather Report</title>
24. </head>
25. <body>
26. <h1>Weather Report</h1>
27. <form id="weatherForm">
28. <label for="city">Enter City:</label>
29. <input type="text" id="city" name="city">
30. <button type="submit">Get Weather</button>
31. </form>
32. <div id="weatherInfo"></div>
33. <script src="./weather\_report.js"></script>
34. </body>
35. </html>

Copied!

**Note:** When you have pasted the code, save your file.

1. Above code has given content and elements:

* <form> Tag: The <form> tag creates a section that contains input elements and a submit button. It has an id attribute set to “weatherForm”.
* <label> Tag: This tag creates a label for the input field with text “Enter City:” to be displayed by the label.
* <input> Tag: It is used to take the entered city name by user.
* <button> Tag: This button indicates that this button submits the form when clicked.
* <div> Tag: id=”weatherInfo”: An empty <div> element with an identifier. This is where the weather information will be displayed.
* <script> Tag: This references an external JavaScript file named “weather\_report.js” to be included in current “weather\_report.html” file

**Step 3: Check the output**

1. To view how your HTML page, right-click the **weather\_report.html** file after selecting this file, then select “Open with Live Server”.
2. The server should start on port 5500, indicated by a notification on the bottom right side.

A black rectangular object with white text

Description automatically generated

1. Click the Skills Network button on the left at number 1. It will open the "Skills Network Toolbox". Next, click on "Other" and then select "Launch Application" at number 2. From there, you enter port number 5500 at number 3 and click this button .

A screenshot of a computer

Description automatically generated

1. It will open your default browser where you will see **cloned-folder-name** folder name. Click on that **cloned-folder-name** folder name and then click on **weatherReport** folder name. You will see files related to this folder where you will click again on **weather\_report.html** file as shown below.



1. It will open the front page and you will see the output as shown below.

A close up of a report

Description automatically generated

1. Enter any city name in input box and then click on **Get Weather** button and then you will see the result.

A screenshot of a weather report

Description automatically generated

1. You can also include catch method to catch error if user enters wrong citry name. Include below code before end of function.
   1. 1
   2. 2
   3. 3
   4. 4
   5. 5
   6. .catch(error => {
   7. console.error('Error fetching weather:', error);
   8. const weatherInfo = document.getElementById('weatherInfo');
   9. weatherInfo.innerHTML = `<p>Failed to fetch weather. Please try again.</p>`;
   10. });

Copied!

***Note:****After pasting the code, save your file. You can use any output method for saving. If you make further edits to your code, simply refresh your browser running on port number 5500. This eliminates the need to relaunch the application repeatedly.*

**Step 4: Perform Git commands**

1. Perform git add to add latest files and folder by writing given command in terminal in git environment.
   1. 1
   2. git add --a

Copied!

Make sure terminal should have path as follows:

A black background with red and blue text

Description automatically generated

1. Then perform git commit in the terminal. While performing git commit, terminal can show message to set up your git config --global for user. name and user.email. If yes, then you need to perform git config command as well for user.name and user.email as given.
   1. 1
   2. git config --global user.email "[you@example.com](mailto:you@example.com)"

Copied!

* 1. 1
  2. git config --global user.name "Your Name"

Copied!

***Note:****Replace data within qoutes with your own details.*

Then perform commit command as given:

* 1. 1
  2. git commit -m "message"

Copied!

1. Then perform git push just by writing given command in terminal.
   1. 1
   2. git push origin

Copied!

* 1. After the push command, the system will prompt you to enter your username and password. Enter the username for your GitHub account and the password that you created in the first lab. After entering the credentials, all of your latest folders and files will be pushed to your GitHub repository.

**Practice Task**

1. You will use another URL to access details weather reports. For example you need to use URL as shown in given screenshot by accessing the same page from where you use URL to be used in the lab instructions.

A screenshot of a computer

Description automatically generated

Note: Instead of city name, now you will gather deatils for lattitude and longitude from user.

1. For this you need to take two <input> fields, one to access lattitude and to get lngitude value from user.

A screenshot of a computer

Description automatically generated  
3. You need to create one submit button after clicking on which it will display the results.

1. Then apply fetch api method to get data related to weather for the user input and create javaScript code for the same.

**Summary**

1. **Weather retrieval form:** Constructed an HTML page with a form enabling users to input a city name and fetch weather data from OpenWeatherMap upon submission.
2. **JavaScript functionality:** Defineed a JavaScript function **showweatherDetails** to handle form submission, preventing default behavior, extracting the entered city, and generating a URL to fetch weather details using the OpenWeatherMap API.
3. **Dynamic weather display:** Utilized JavaScript's fetch() to retrieve weather data, updating the webpage dynamically with the fetched weather information, including city name, temperature in Celsius, and weather description, ensuring a seamless display of weather details without page reload.