

Exercise 10: Promises and async/await

Weather Application

Upload a html file named USPnumber1.USPnumber2.html.

Attention: If you need to upload a file after the deadline use late.USPnumber1.USPnumber2.html

Activity: Build a Simple Weather Application (70 minutes)

Objective: The goal of this activity is to apply the knowledge of asynchronous JavaScript, Promises, and Fetch API to build a simple weather application that fetches data from a public API and displays it.

Instructions:

Open your preferred code editor or browser-based environments, such as JSFiddle or CodePen.

Create an HTML file and include the necessary tags for a basic HTML structure (<!DOCTYPE>, <html>, <head>, <body>). Add a <script> tag inside the <head> or at the end of the <body> to write your JavaScript code.

Create the following HTML structure for the weather application. If you want to use city names, call an API or save the latitude and longitude for, at least, 3 cities.

An input field for users to enter a latitude and longitude (or city name).

A button to fetch and display the weather data.

An area (div, p, etc.) to display the weather data.

Use the Fetch API to send a GET request to the Open-Meteo public API. The endpoint is http://api.open-meteo.com/v1/forecast?latitude={lat}&longitude={lon}¤t_weather=true.
API Documentation

Implement the following functionality:

When the user clicks the "Fetch Weather" button, use the entered latitude and longitude (or city name) to fetch the weather data from the API. Display the fetched data in the designated area. The displayed data should include latitude, longitude (or city name), temperature, weather condition (API Documentation), and an icon representing the weather condition.

For weather icons:

Use the icon map from GitHub (day time) OR

Use any 3 representative icons for Clear, Cloudy, Something Falling (Rain, Snow, Drizzle, etc)

Handle possible errors appropriately. If the fetch operation fails, display a user-friendly error message.

As you work on the activity, discuss the implementation with your partner or group members. Ask for help or clarification from the instructor or teaching assistant if needed.

Once the activity is completed, share your solution with the rest of the class and discuss any challenges you encountered or alternative approaches you discovered.

Remember, the goal of this activity is to practice working with `async/await`, Promises, and Fetch API. Enjoy and have fun!

(Note: For the sake of this activity, we don't consider the cross-origin (CORS) issues that might occur when making the API request. In a real-world scenario, you might need to handle this.)

Please note that the Open-Meteo API does not require an API key and is only used as an example. In real-world applications, always check the API's documentation and terms of use.