Consuming an API

Overview

The power of the internet is the ability to share information. The web is a collection of documents that are linked together through hyperlinks. The web is also a collection of **APIs** (Application Programming Interfaces) that allow you to access and use data from other applications and services.

The purpose of this activity is to introduce an external weather source which can be used to serve up live, weather information based on location and other parameters.

Prepare

This activity uses the third-party **OpenWeatherMap API** which will require you to obtain a **free account** in order to consume the data. You will only need to submit some basic information to obtain your free account. You do **not** need to use a credit card.

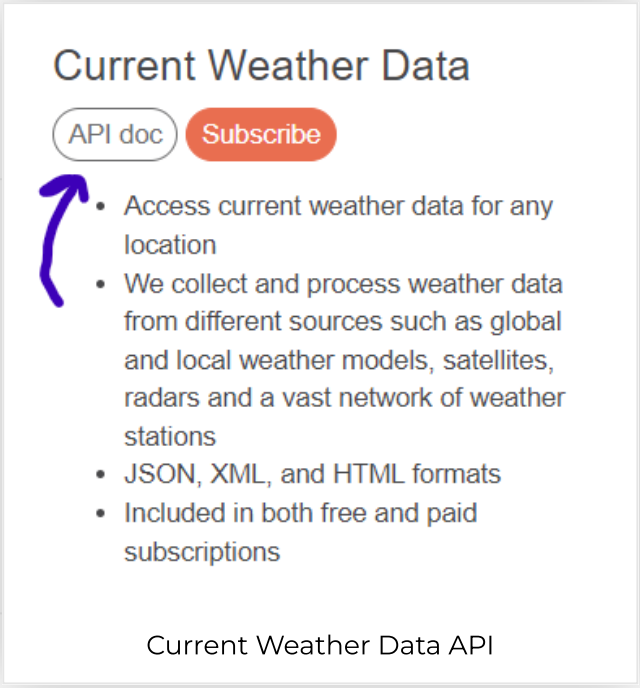
Step 1: OpenWeatherMap

1. Navigate to OpenWeatherMap and find the "Search city" input box below the page's hero image.
2. Enter your city name in the input box provided and click **Search**.
3. Click on the appropriate search result for your city/location.
4. Review the weather data that is provided. You may want to switch to **metric** (C) or **imperial** (F) based results using the buttons in the upper-right corner of the search interface.
5. Now, navigate to the menu of provided **APIs** using OpenWeatherMap: Weather API.
6. Under the **Current & Forecast weather data collection**, the accounts that you will be using can **freely** use the
    - **Current Weather Data** collection, and the
    - **5 Day / 3 Hour Forecast** collection.

> It is important that you understand that these are the collections that you can use with the Free Account. The others require a chargeable account. There are always limitations regardless of the account so that is why it is important to reference the documentation on any API that you are using.

7. Scroll down to the **Current Weather Data** API section, and click on the **API doc** button.

> The Current Weather Data page explains how to call the current weather data for one location. Most APIs provide useful documentation to help you use the data. You will find how to make an API call documentation in this document along with examples of responses.



Current Weather Data API

8. In the **Current Weather Data** API documentation, find the *Call current weather data – How to make an API call* section and study the API parameters and examples provided.
9. **Bookmark** this documentation for future reference.

Step 2: Get an Account

1. Navigate to the OpenWeatherMap: Pricing page.
2. Scroll down to the *Current weather and forecasts collection* and find the **Free** account column. This account type will be sufficient.

> This service is provided under Creative Commons Attribution-ShareAlike 4.0 International license (CC BY-SA 4.0). The data is open and licensed by the Open Data Commons Open Database License (ODbL).

3. Click on the Get API Key button under the **Free** column.
4. Proceed through the account sign up directions to get an API key which you will need to store and keep secure for your use only. **This unique key is required in the data requests that you make to this service**.

> Do **not** use other people's API keys. Doing so is a sign that you may not be learning through proven patterns of digesting material and then producing

original content, which is essential to your learning.

5. Record this **API key (appid)** in a safe place. You will need it for this activity and for your assignments.

Activity Instructions

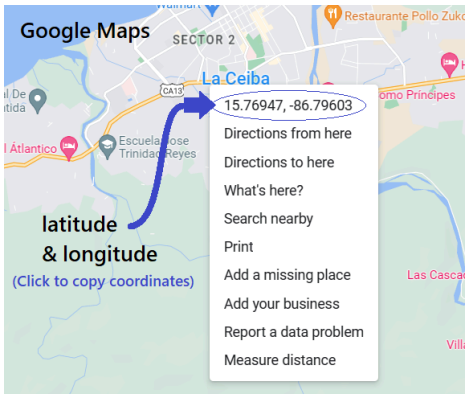
In this practice activity, you will create a simple page that displays some current weather conditions for a particular location in the world, **Trier, Germany**.

Step 1: Location

1. Using Google Maps, find the **latitude** and **longitude** coordinates of Trier, Germany.
    1. First locate the city on the map.
    2. On the map, right-click on the city name.
    3. Click on the latitude and longitude coordinates given. This will copy the coordinates to your clipboard.

Trier, Germany – Porta Nigra

> You will not need to be so specific with so many significant digits after the decimal on the coordinates to use the closest weather station. Use two (2) digits after the decimal.



Location Information

▼ **Check Your Coordinates**

Trier, Germany is located at 49.75° N latitude and 6.64° E longitude.

Step 2: HTML

1. In VS Code, create a new HTML page in your **wdd231** folder.

2. Use the following **template** for the body of your HTML document:

```
<h1>OpenWeatherMap.org API Test</h1>
<main>
  <p>The current temperature in Trier, Germany is <span id="current-temp
  <figure>
    <img id="weather-icon" src="" alt="">
    <figcaption></figcaption>
  </figure>
</main>
<footer>
  [Enter Your Name Here] | OpenWeatherMap.org | CC BY-SA 4.0
</footer>
```

Step 3: JavaScript

1. Create a new JavaScript file and source reference that file in your HTML file. Make sure you place this file in an appropriate location given the course file and folder standards.
2. In the JavaScript file, first select all of the HTML elements that will need to be manipulated and assign them to **`const`** variables.

▼ **Check Your Understanding – Example**

```
// select HTML elements in the document
const currentTemp = document.querySelector('#current-temp');
const weatherIcon = document.querySelector('#weather-icon');
const captionDesc = document.querySelector('figcaption');
```

3. Declare a **`const`** variable named "**`url`**" and assign it a valid URL string as given in the openweathermap api documentation that was presented above and bookmarked.

```
const url = 'https://api.openweathermap.org/data/2.5/___________';
```

    1. Use the **Current Weather API** named '**`weather`**'.
    2. Start a query string with the "**`?`**" character as shown in the examples.
    3. Use a **`&`** between each key/value pair in the query string in these next steps.
    4. Specify the latitude and longitude of Trier, Germany using the information you have gathered and the examples provided.
    5. Set the **units** to imperial: "**`units=imperial`**" or to metric: "**`units=metric`**"
    6. Provide your **API key**: "**`appid=[enter your key here]`**"

4. Define an asynchronous function named "**apiFetch()**" that uses a try block to handle errors.
    1. Store the results of the URL **fetch** into a variable named "**`response`**".
    2. If the response is OK, then store the result of **`response.json()`** conversion in a variable named "**data**", and
    3. Output the results to the console for testing.
    4. Else, throw an **Error** using the **`response.text()`.**
    5. Finish off the catch block by outputting any try error to the console.
5. Remember to invoke the apiFetch() function with a call somewhere in your script.

▼ **Check Your Understanding**

```
async function apiFetch() {
  try {
    const response = await fetch(url);
    if (response.ok) {
      const data = await response.json();
      console.log(data); // testing only
      // displayResults(data); // uncomment when ready
    } else {
        throw Error(await response.text());
    }
  } catch (error) {
      console.log(error);
  }
}

apiFetch();
```

6. Run the page locally and view the console output. Find the current temperature (**`temp`**) and the weather event description (**`weather.description`**), and image icon reference (**`weather[0].icon`** – 3 characters) in the data.

The weather array indicates that there can be more than one current weather *event*. You only need to focus on the first weather event if there is more than one.

The icon is just a preset code name that corresponds to OpenWeatherMap's library of images which is found at the base addresses of:

- **`https://openweathermap.org/img/w/`**

Example 10d.png: Rain  https://openweathermap.org/img/w/10d.png

- `https://openweathermap.org/img/wn` (This version allows sizing using @.)

Example 10d@2x: Rain

https://openweathermap.org/img/wn/10d@2x.png

All these images are .png file types. Here is a link to the documentation about Weather icons.

7. Build the **displayResults** function to output to the given HTML document.

▼ **Check Your Understanding – *The blanks are intentional***

```
function displayResults(data) {
  currentTemp.innerHTML = `${data._____}&deg;F`;
  const iconsrc = `https://openweathermap.org/img/w/${_____}.___`
  let desc = data.weather[0]._____;
  weatherIcon.setAttribute('___', _____);
  weatherIcon.setAttribute('___', _____);
  captionDesc.textContent = `${desc}`;
}
```

These steps are demonstrated in the following **video series**.

Current Weather Part 1

Overview

Current Weather Part 2

Fetching the data

Current Weather Part 3

Adding JSON to the web page

Current Weather Part 4

Adding some CSS

Step 4: Test

1. Run this page locally and test the results making sure there are no JavaScript errors and that the current weather data is displayed accurately.

2. Debug your JavaScript as needed.

3. Share any issues that you have with your group on **Microsoft Teams**.

Optional Resources

What are third-party APIs? – MDN

Back