|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 🔑 **Essential** 🔑 | | | |  | 🏆 **Enhanced** 🏆 | | | |
|  |  |  |  |  |  |  |  |  |
| \_\_\_\_\_ | \_\_\_\_\_ | \_\_\_\_\_ | \_\_\_\_\_ |  | \_\_\_\_\_ | \_\_\_\_\_ | \_\_\_\_\_ | \_\_\_\_\_ |

**Score: \_\_\_\_\_**

**31 – Web APIs and AJAX**

**Activities**

COMP256 – Computing Abstractions

Dickinson College

Spring 2022

Prof. Grant Braught

**Name:**

**Introduction:**

Today’s class introduced the idea of a Web Application Programming Interface (API). These services were explained using ordering at a restaurant as a metaphor before concrete examples using the OpenWeather API were introduced. We saw how URLs with query strings are used to make API requests of web service, and how those requests can be sent in JavaScript using AJAX (Asynchronous JavaScript and XML). We saw that responses from API requests are often returned in JSON (JavaScript Object Notation) and how we to use JavaScript code to reference objects, arrays and fields in the response. Combining that with what we have learned previously about using JavaScript to modify HTML elements we were able to build a simple Weather App. The activities below will have you explore Web APIs a little further and then enhance the Weather App to include some additional features.

**Preliminary:**

🔑 1. Install a JSON Formatter extension for your web browser. The demos in class used Chrome with the JSON Formatter extension. Those listed below for Safari and Firefox are similar but will look and operate slightly differently.

* Chrome: <https://chrome.google.com/webstore/detail/json-formatter/bcjindcccaagfpapjjmafapmmgkkhgoa?hl=en>
* Safari: <https://apps.apple.com/us/app/json-peep-for-safari/id1458969831?mt=12>
* Firefox: <https://addons.mozilla.org/en-US/firefox/addon/json-formatter/>

There is nothing to turn in for this question. Just install the web browser extension described above. You will need it for later parts of this assignment.

**Web API Concept:**

🔑 2. Describe in a few sentences of your own words the role that a Web API plays in Web based applications.

🔑 3. In class we saw ordering at a restaurant as a metaphor to explain what a Web API does. Invent a metaphor of your own for explaining what a Web API does and explain it in a few sentences of your own words.

🔑 4. Go to <https://apilist.fun> and browse through the APIs that are listed there. Find at least two API’s that you find interesting. Give a link and a brief description of what types of information that each API provides.

**API Requests:**

🔑 5. Fill in the rows of the table below based on the following the URL that is a request to an API that provides news stories.

https://newsapi.org/v2/top-headlines?country=us&category=business

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  | |  |
|  | **Term** | **Value** | |  |
|  | API Endpoint |  | |  |
|  | Server |  | |  |
|  | Service |  | |  |
|  | Query String |  | |  |
|  | Query Parameters |  |  |  |
|  | Fields |  |  |  |
|  | Values |  |  |  |
|  |  |  | |  |

🔑 6. This question will ask you to write some new requests using the OpenWeather API that we saw as an example in class. You can try out your answers by entering them into the browser address bar as was shown in class. When doing so you can use the API Key (appid): 71600e803223b59828cade9d4fc4d46b. Or you may register for your own free OpenWeather API key here:

* <https://home.openweathermap.org/users/sign_up>

a. In class we saw an API request for the current weather in Carlisle using its zip code (17013). Give an API request for the current weather in the part of Harrisburg that contains the Pennsylvania state capital (zip code 17102).

b. The Current Weather API has a number of different ways to specify the location and to request information in different formats and even in different languages. Use the documentation for the Current Weather API endpoint that can be found here: <https://openweathermap.org/current> to answer the following questions.

i. It may not always be convenient to use the zip code of a location to request weather data. Use the section of the documentation labeled “Built-in API request by city name” write an API request for the current weather in Pittsburgh, Pennsylvania (pa) in the United States (us).

ii. The demo page we saw in class reported the temperature in degree Kelvin, which is a little inconvenient for most of us. Use the section of the documentation on “Units of measurement” to write an API request for the current weather in Carlisle with the temperature reported in Celsius.

🏆 iii. Dickinson is an international campus so there may be good reasons to get results back in different languages. Use the section on “Multilingual support” to write an API request that reports the current weather in Carlisle with the “description” of the weather appearing in a language of your choice (other than English).

**JSON Responses:**

The documentation for an API endpoint will (at least if it is good) describe the fields that appear in the responses that are returned for requests to that endpoint. For example, the “Weather fields in API responses” section of the Current Weather API endpoint documentation (<https://openweathermap.org/current>) provides an example response and describes each of the fields that it contains.

🔑 7. Assume that the JavaScript variable resp refers to the example response in the “Weather fields in API responses” section of the Current Weather API endpoint documentation referenced above. Fill in expressions using dot and array notations that will reference the fields indicated in the table below.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  | **Field** | **Reference** |  |
|  | latitude | resp.coord.lat |  |
|  | barometric pressure |  |  |
|  | wind gust speed |  |  |
|  | icon for the weather |  |  |
|  | description of the weather |  |  |
|  | min temperature for day |  |  |
|  |  |  |  |

🔑 8. TheMealDB API (<https://www.themealdb.com/api.php>) provides access to a “crowd-sourced database of Recipes from around the world.” The request below to asks the TheMealDB API for all of the meals it knows that use carrot as a main ingredient:

<https://www.themealdb.com/api/json/v1/1/filter.php?i=carrot>

a. Use your browser to send the above request. Use your browser extension to parse the JSON response and paste a screen shot of the nicely formatted JSON here.

b. Assuming that the JavaScript variable resp refers to the response that you received in part a, give a reference to the name of:

i. The name (“strMeal” field) of first meal in the response.

ii. The URL of the thumbnail image of the last meal in the response.

**Enhancing the Weather App:**

The api1.html page in the development container you are using contains the weather page as was presented in class. You can render that page by clicking the “Weather: Use the OpenWeather API” link. You can see and modify the page by opening it in a text editor.

🔑 9. Open the page in your text editor and browser and confirm that clicking the button displays the temperature in degrees Kelvin, as we saw in class.

There is nothing to turn in for this question. Just confirm the page works and that you can see the source code and that it matches the code from class.

🔑 10. Reporting the temperature in degrees Kelvin is not that useful to most people. Modify the app so that it will report the temperature in degrees Fahrenheit or degrees Celsius (whichever you prefer). Paste the statement that you modified to change the API request here.

🔑 11. Let’s add some wind speed to the information reported by the app.

a. Add HTML elements to the page so that wind speed can be displayed. Use the elements for the Temperature as a guide. Give the code for the new HTML elements that you added here.

b. Modify the getWeather function so that that the app will now displays the wind speed in addition to the temperature when the button is clicked. Give the lines of code that you added to the getWeather function here.

🏆 12. Getting the weather in only one location isn’t that exciting. So, let’s modify the app so that we can get the weather from any zipcode in the US.

a. Add a text field to the page so that the user can enter a US zip code. Be sure to give your text field an id attribute so that you can select it in your JavaScript later. Give the HTML for the element that you added here.

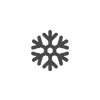
b. Modify the getWeather function so that the app uses the zip code provided by the user and reports the temperature at that location. You can test your change by comparing the temperature in Carlisle (17013) with what should be a colder temperature in Anchorage, Alaska (99501) and a warmer temperature in Orlando, Florida (32789). Give your complete modified getWeather function here.

c. What happens if you run the program without any value in the zip code field? It seems odd that we would get a temperature for nowhere. Research if/else statements in JavaScript and modify the code so that if the zip code field is empty when the button is clicked, the page puts 17013 into the zip code field and reports the weather for Carlisle. Give your complete modified getWeather function here.

🏆🏆 13. To help developers make their apps look a little nicer the OpenWeather site provides images for each of the weather conditions (<https://openweathermap.org/weather-conditions>) that it reports. For example:

Icon

Description automatically generated Graphical user interface

Description automatically generated 

Few Clouds Rain Snow

Let’s now enhance the app by adding images that represent the current weather conditions. This will work by having an img element in the page and then modifying its src attribute using the image (i.e. the value of the icon property) from the API response.

a. Add an img element to your page. Initially the src should be empty (“”), which will cause the image to display nothing. Recall from prior homework that in order to change the src attribute of the image you will need to be able to get the img element using its id attribute. So, be sure to include an id attribute in your img tag. Give the HTML for the img element that you added.

b. The value of the icon property (see question #7) in the response specifies the base filename of the URL for the image corresponding to the current weather conditions.

For example, the value of the icon property for “Few Clouds“ is “02d”. The images provided by OpenWeather are .png files. So, the full filename for the “Few Clouds” image is “02d.png”. Also, the OpenWeather service keeps the images at the URL

* + <http://openweathermap.org/img/wn/>

so the full URL for the “Few Clouds” image will be:

<http://openweathermap.org/img/wn/02d.png>

Add a line to the getWeather function that creates a new JavaScript variable named imgURL that holds a string containing the URL of the image for the current weather conditions.

Give the line of code you added.

Note: If you would like a larger image, you can request that by modifying the filename slightly. For example, the filename “02d@2x.png” or “02d@4x.png” will return images that are twice as large or four times as large.

c. Now that you have the URL for the image that should be displayed, you need to use JavaScript to change the src of the img tag to be that URL. Add statements to the getWeather function that set the src of your img element from part a so that it will display the image for the current weather.

d. Give the full HTML source code for your Weather App page here.

Optional: To help me improve and scope these activities for future semesters please consider providing the following feedback.

a. Approximately how much time did you spend on this activity outside of class time?

b. Please comment on any particular challenges you faced in completing this activity.