

## Assignment #5

# AJAX & JavaScript Libraries and Frameworks

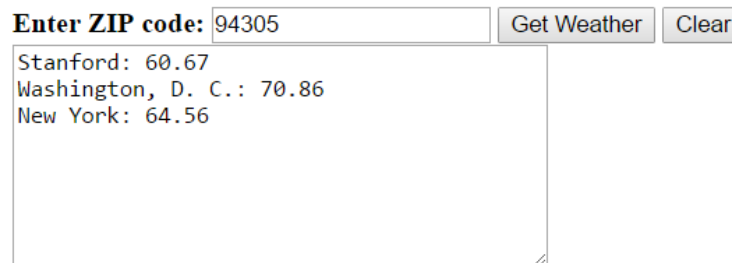
## CS193C Summer 2022, Young

For our last assignment we will experiment with AJAX and JavaScript Libraries and Frameworks. This assignment is due Thursday August 11<sup>th</sup> at 1:30pm. **In order to get all your assignments graded in time to make the end-of-quarter grades deadline, no assignments will be accepted after Thursday the 11<sup>h</sup> at 11:59pm.**

### AJAX

For the AJAX section of our assignment we'll retrieve weather information from openweathermap.org. Here is a screenshot showing the AJAX assignment in action:

## Open Weather via AJAX



Enter ZIP code:

Stanford: 60.67  
Washington, D. C.: 70.86  
New York: 64.56

The user enters in a ZIP code and clicks on “Get Weather”. The weather for that ZIP code is retrieved from openweathermap.org and added to a textarea. Weather information listed should include the City corresponding to the ZIP code and the temperature (which is provided to us by openweathermap.org). The textarea should list all requests made. The user can click on the “Clear” button to clear the textarea.

**This assignment is quite similar to the Fetch example we went over which can be found in the 112 Promises and Fetch.zip example file from Lecture 12.** We'll need to teach you a few things in order to get this assignment up and running.

### Creating an Account with OpenWeatherMap.org

Go to:

<http://openweathermap.org/>

Sign Up for an account (upper-right of website). Once you've created an account, go to your account summary and switch to the “API Keys” tab. You'll need to have an API Key to communicate with the server.

### Requesting Weather Reports

To make a request to the server for the weather at a particular zipcode the request format is:

`https://api.openweathermap.org/data/2.5/weather?zip=zipcodeDesired,us&units=imperial&APPID=yourAPIKey`

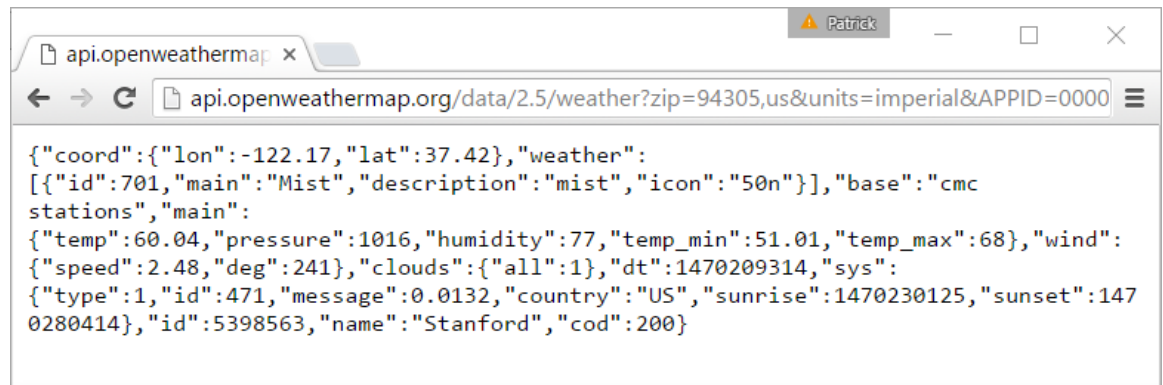
where *zipcodeDesired* is replaced by the zipcode you want and *yourAPIKey* is replaced by your account's API key.

Please note that openweather requires that you use the https secure version of http. It will block a non-secure http request so your URL should begin with https instead of http.

## Getting Weather Information

You'll pass your URL to OpenWeatherMap.org using a Fetch request. After a brief delay, your callback should run.

Take the result of the Fetch callback and call the json method on it. You will recall that this json method also uses Promises. When it returns its result, you should be able to pull the data for the zipcode out and place it on the webpage.



This is what's in the `responseText` or `responseXML` of your `XMLHttpRequestObject`. You will probably find JSON easier to use, but you may use either for this assignment.

**Don't forget Ajax with Fetch uses Promises because often a server will take a while before it responds.** You may have to wait quite a few seconds before your results show up.

## Reference

If you're still having trouble figuring Fetch out and going back the lecture 11 examples and lecture is not sufficient, you can also take a look at MDN's description here:

[https://developer.mozilla.org/en-US/docs/Web/API/Fetch\\_API/Using\\_Fetch](https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API/Using_Fetch)

You'll be doing something similar to what they show, however, you'll need to replace their

```
.then(function(myJson) {  
    console.log(JSON.stringify(myJson));  
});
```

with your code that will take the results and stick them on your webpage. They're just taking the JSON results, and restringifying them and printing them out on the console to demonstrate that their example works.

## JQuery

We'll keep the JQuery section simple and similar to the in-class examples. Our objective is to just give you just a bit of hands on experience so that you'll remember what you saw in lecture a bit better.

Start out with the `jquery-practice.html` file provided with this assignment's downloads. This file contains no JavaScript and no JQuery. It does contain HTML along with some CSS Styles. It also contains several buttons which you'll need to wire up to carry out various JQuery tasks.

### Get JQuery Loaded

First things first, you need to get JQuery loaded. I've provided a JQuery file with the assignment downloads. Load it in using a standard `<script>` tag.

### Turn Headings Red

Wire up the first button so that when the user clicks on it, all headings (h1, h2, and h3) turn red. Note that I've provided a style rule which you may find helpful for carrying out this task.

### Fading Items

Wire up the second button so that when the user clicks on it the heading "Speakers" fades out over a 1 second (1,000 millisecond) period. Fade just the `<h3>` tag, not the subsequent paragraph on speakers. Once the heading has completely faded out, it will be removed from the normal text flow and the subsequent paragraph will be bumped up. This is normal and not something you need to correct.

## Node

For our last problem, we're going to get a little bit of experience with Node. In lecture, I showed you how to write and execute a simple Node problem that accessed the file system (see the handout "h05 Basic Node Hands On").

Follow along with the handout and get Node installed on your computer. Experiment with using Node both as an interactive JavaScript environment, and by writing JavaScript files and executing them with Node.

Write a program named "getos.js" that is a very short program which retrieves and prints the type of OS you are using by using the "platform" method from the "os" module. See the documentation [here](https://nodejs.org/dist/latest-v8.x/docs/api/os.html).

<https://nodejs.org/dist/latest-v8.x/docs/api/os.html>

What is the value printed?

Turn in the "getos.js" file.