# USA States (Ajax, jQuery and JSON)

## Using external data

In this assignment you will write JavaScript that retrieves JSON data from an API endpoint, found here:

<http://misdemo.temple.edu/states>

It is important to note that there is nothing particularly magical about the misdemo server here. The endpoint URL could be a machine anywhere on the internet!

**ADVISORY!**  There’s quite a lot in the start file that can be copy / pasted in order to answer the “On your own” questions. This is not recommended. Students are encouraged to type the $.getJSON() syntax themselves in each question. This is the best way to learn the syntax and reinforce what it is doing.

## Getting started (Together as a class)

1. Visit the API endpoint by typing <http://misdemo.temple.edu/states> into a browser.
2. Retrieve assignment18\_states.zip provided by your instructor.
3. Extract the code into your mis2402workspace and open the index.html file in Visual Studio Code.
4. Notice that the click event handlers for buttons 2, 3, 4, and 5 have not been provided in the start file. You need to provide these. Do so now.
5. Next, take note of the basic structure of the getJSON method.

|  |
| --- |
|  $.getJSON("",function(result){ // the variable named result has  // block level scope inside the callback function. // it is a good practice to display result in console.log console.log(result); }); |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

1. Review the code found in the displayStatecodes function. Discuss: What needs to replace the “what goes here???” string?
2. Review the code found in the displayEastmost function. Notice that this function writes the answer to an HTML tag. This step is located **inside** the callback function.

This is the technique you should follow in the rest of this assignment!

SIDE NOTE: There’s a technical reason for this! It’s not just a random decision! JavaScript executes the getJSON command ***asynchronously.*** This makes sense, because there’s no telling if the url (in this case, misdemo.temple.edu/states) is going to respond in a timely manner. The only way we know that the call to the URL has come back with response is … the callback function. The callback function defines what to do when the call, comes back.

1. As a class, complete the displaySouthmost function. Notice that this button is to respond with both the latitude and the US State abbreviation.

You will need to create a new variable in addition to answer (for example, answertext ) to hold the text.

On your own

1. Complete the displayWestmost function. It should retrieve the data, determine the westmost state using longitude, and place that single value into the div tag with the id of textDisplayed4.
2. Complete the displayAvgLongitude function. It should retrieve the data, sum the values of all the longitudes, divide by the number of elements in the array, and place that single value into the div tag with the id of textDisplayed5.
3. Publish your work. Be sure that you can find your work on the class server by typing in its URL in the browser. Test your work \*again\* on the class server.

For example:
http://misdemo.temple.edu/tux99999/assignment18\_states

How will this assignment be graded?

* If your work is not found at the expected locations on misdemo, you will get a score of **zero**.
* If your work does not generate any output, you will get a score of **zero**.
* If your work generates **all output** correctly, you will get a score of 100%.
* If your work generates **all but one output** correctly, you will get a score of 80%
* If your work generates **only one output** correctly, you will get a score of 40%
* All other assignments will receive a score of 60%.