

## **jQuery IP Locator Ajax Lab**

Sometimes you have an IP address in your server logs but wonder where in the world that surfer was when they visited your web site. In this lab we're going to be consuming data from a web server that will provide location data if you give it an IP address. We'll start by taking a look at the data.

1. Open a browser and navigate to <http://ip-api.com/json/72.247.244.88>. What you'll see in your browser is a string that looks like JSON data. This is the data we'll consume via jQuery's Ajax.
2. Examine that data and see what data are available in it. Make a plan to decide what you want to appear on your page and what you want to ignore.

### **Grabbing the movie title**

3. Edit index.html and add jQuery to the page.
4. Create a new JavaScript file where you will write your JavaScript/jQuery for the rest of this lab.
5. In it, add a ready event handler.
6. Wire up the button click to a function.
7. Get the value in the IP Address textbox. Store it in a variable.
8. Craft the url which will be used for your Ajax request eventually. Store it in a variable and `console.log()` it. It should be `"http://ip-api.com/json/"` and then the IP address.
9. Run and test. Make sure that when you click the button you can see a good URL including the IP address in the console.

### **Making the Ajax call itself**

Once we've ensured that we're successfully getting the IP address and formatting a navigable URL, let's actually make the Ajax call.

10. Make an Ajax call to the url from above. The method should be "GET".
11. You can validate that the call was successful by looking in developer tools under the Network (or Net) tab. Go ahead and do that.
12. Add `.then()` to handle the callback from the server. In it, register a success callback and an error callback. (Hint: pass two functions into `.then()`.)
13. In the success callback, you can just `console.log()` the parameters.
14. In the error callback, `console.error()` the parameters.
15. Run and test. Make sure you can see either the success message or the error message. If it is the error message, tune your call to get to a success.

### **Putting the response data on the page**

Alright! Now that you have some data coming back from the Ajax server, let's put it on the page and make it look nice.

16. Edit your success function to write some data -- any data; organization, city, latitude, longitude, whatever -- from the server's response into the outputDiv. (Hint: use jQuery's `html()` function).
17. Run and test. Make sure that as you change the IP address in the text box and click the button you're getting different data back from the service.
18. Cool! You have some data in the `<div>`. Your mission, should you choose to accept it, is to grab all of the best data from this service and display it in the output `<div>` in a nicely formatted way.

Once you have fetched location data being displayed on button click, you can be finished.