Getting Varied Responses from Server

This lesson will explain how to get a varied response from a server based on the input we send to it. This will help us build a more accurate dropdown menu in the end.

In the previous installment of our server stub, it doesn't respond to data. We are now at a point where we want to get different values depending on the menu item that's active, so we want the server to give varied responses based on the data passed.

We'll follow the same pattern as last time: start from a blank page so that we can focus on testing functionality.

For this example, we'll create an input, and whenever we click a button, the client will send a `GET` request to the server with whatever's been typed into the input as data. The client will then append the response to a `div`.

```
const HOST = 'server.com/';

const goElement = document.getElementsByClassName("go")[0];
goElement.onclick = function() {
  const inputElement = document.getElementsByClassName("test")[0];
  api.get(HOST + "menus", {menu: inputElement.value}, displayText);
}

function displayText(response) {
  const outputElement = document.getElementsByClassName("output")[0];
  outputElement.innerHTML += (response + "<br>");
}
```

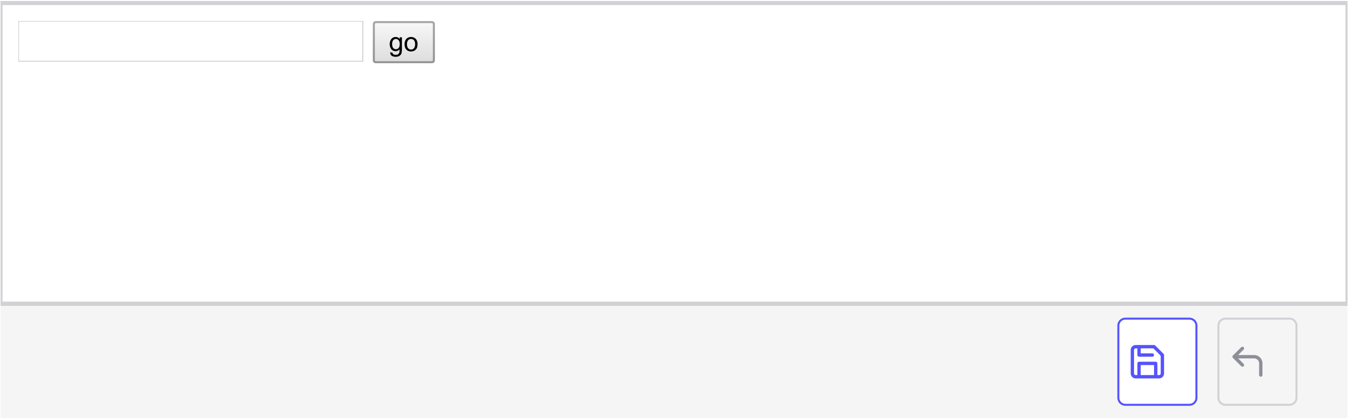
We should see this code working when our `api` and server is modified to handle various inputs,

Output

JavaScript

HTML

CSS (SCSS)



Try writing “a” into the input and click “go”. Now, try writing “b" into the input and click “go”. What happens if you type something else in the input and click “go”? You should see that the server returns something different. That’s really all we’re trying to achieve in this example- responding to an API call differently depending on the data passed in.

To test yourself that you really understand this, try this:

1. Extend the server code so that it responds to “c” with `I got a C` and
2. Extend the api library so that it calls the callback twice

Modify the code in the JavaScript tab below. When you’re ready, click the “Run Test” button. There’s some hidden JavaScript that’ll print depending on if you succeeded or failed!

Now that we’ve extended our server to handle data, let’s go back to our dropdown menu so that we can apply it there.