CMPS 3680 Guided Lab 10

Spring 2024

This document is incomplete. I will try to provide more detailed instructions later.

Part 1 - Setup

1. For this assignment you will have multiple files, so I would highly recommend creating a lab10 folder.
2. Inside of the lab10 folder run the following commands from the terminal:

```
wget https://cs.csubak.edu/~paul/cs3680/labs/lab10.zip
unzip lab10.zip
```

   This will download and extract the required files you will use for this lab.
3. Review these files carefully, notice that I have already included jquery for you from the official CDN.
   Also notice I have already given all of the necessary elements IDs so you can access them in your JS code.
4. You will be using JSON RPC 2.0 to make API calls to the following URL:

```
https://paul.cs3680.com/labs/api
```

5. The documentation for the API can be found here:

```
https://paul.cs3680.com/labs/docs/labs.html
```

   Notice there are three remote methods available (**hello**, **add**, and **giphy**) and each one has different parameters that must be passed.
6. Review the specifications for JSON RPC 2.0 - pay specific attention to the provided examples:

```
https://www.jsonrpc.org/specification#examples
```

   All of the methods I have provided in the API are using **NAMED PARAMETERS** (key/value pairs)

Part 2 - script.js

1. For this lab you will be using the jquery $.ajax function to make remote procedure calls to the API I have provided. The parameters will be acquired from the input fields in lab9.html when the user clicks the corresponding button.
2. Remember that with JSON RPC all requests must be post requests, and the request body (aka payload) will always follow the same JSON format, for example:

```
{
    "method": "hello",
    "params": {
        "name": "BOB"
    },
    "id": "1",
    "jsonrpc": "2.0"
}
```

   HINT: You can convert a javascript object to a JSON string with the `JSON.stringify()` function.
3. If you receive a valid response from the server, you will update the corresponding response element for each call. For the **hello** and **add** calls you will modify the text inside the `<p>` elements. For the `giphy` call you will update the `src` attribute for the `<img>` element so the gif shows up on the page.
4. If you receive an error, update the provided error div with the error message received and remove the `hidden` class.

Example Image of Completed Lab: