Cumulative Project: Content Creators Contracting

Complete your first project on your own computer.

Content Creators Contracting

Project Overview

In this project, you have been hired by a "professional" company, Content Creators Contracting, to write a function that signs their HTTP requests with the proper content types. Normally this behavior is handled by the browser, but they really mucked things up in thinking they could solve this problem better. Now, none of their files are loading properly.

They have managed to craft the code to get the file extension from a string containing a filename and store it in a variable called 'extension':

```
const extension = filename.match(/.*\.([^\.]*)$/)[1];
```

In this code, if `filename` were equal to 'index.html', the string 'html' would be stored in `extension`. However, they're struggling to write the rest of the function. (Note: this code uses a Regular Expression, which you're not expected to know how to use yet! The important thing to know is that the filename extension will be stored in the `extension` variable.)

It's up to you to save their site and their business.

You can view what the final version of this project should look like here.

The goal of this project is to get you running a testing suite and implementing the functionality laid out in the testing suite.

How To Begin

To start, download the starting code for this project here. After downloading the zip folder, double click it to uncompress it and access the contents of this project.

Implementation Details

To complete this project, your code will need to contain the following:

- A function called `getContentType`, which will take a string representing a filename and return the proper content-type extension. You will need to implement the functionality for determining content types for `'text/html'`, `'text/css'`, `'image/jpeg'`, and `'text/plain'`. For more information on this functionality, run the testing suite (detailed below).

All of your code should be written in **js/request-logic.js**. Use the descriptions above and the testing suite (discussed below) to guide implementation of all necessary functionality.

To demo the site, open **index.html** in your browser (by double clicking **index.html** in a file browser or dragging it into your Internet browser). You will be writing JavaScript code that uses new syntax (you will learn more about this later in the intensive), so you will need to use the most up-to-date version of Chrome to ensure your code runs correctly. If your version of Chrome is too old, correctly-written code may still not run as expected.

Testing

A testing suite has been provided for you, checking for all essential functionality and edge cases.

To run these tests, first open the root project directory in your terminal. Then run `npm install` to install all necessary testing dependencies (you will only need to do this step once). Finally, run `npm run test`. You will see a list of tests that ran with information about whether or not each test passed. After this list, you will see more specific output about why each failing test failed.

As you implement functionality, run the tests to ensure you are creating the correctly named function and that it returns the proper values. The tests will additionally help you identify edge cases that you may not have anticipated when first writing the function.

Solution

We'll provide solution code immediately after this project for you to check your work when you finish. You can also look ahead if you're feeling stuck or need a push in the right direction!