

Developer Workshop

LIT Work Experience Programme

Front-End Web and API Development Overview



Copyright © 2014 Liberty Information Technology Ltd Northern Ireland. No part of this document may be reproduced, used in any form or by any means without the prior written permission of Liberty Information Technology.

Challenge Overview

- Over the next two days, a large portion of your time is going to be spent working on a coding challenge.
- This challenge includes front end and back end development in some of the latest technology and should provide an interesting look at web development.
- We will be working through the task together, so be prepared to ask questions and to talk to the rest of your team. While everyone will have their own solution, ideas can and should be shared.

Getting Setup

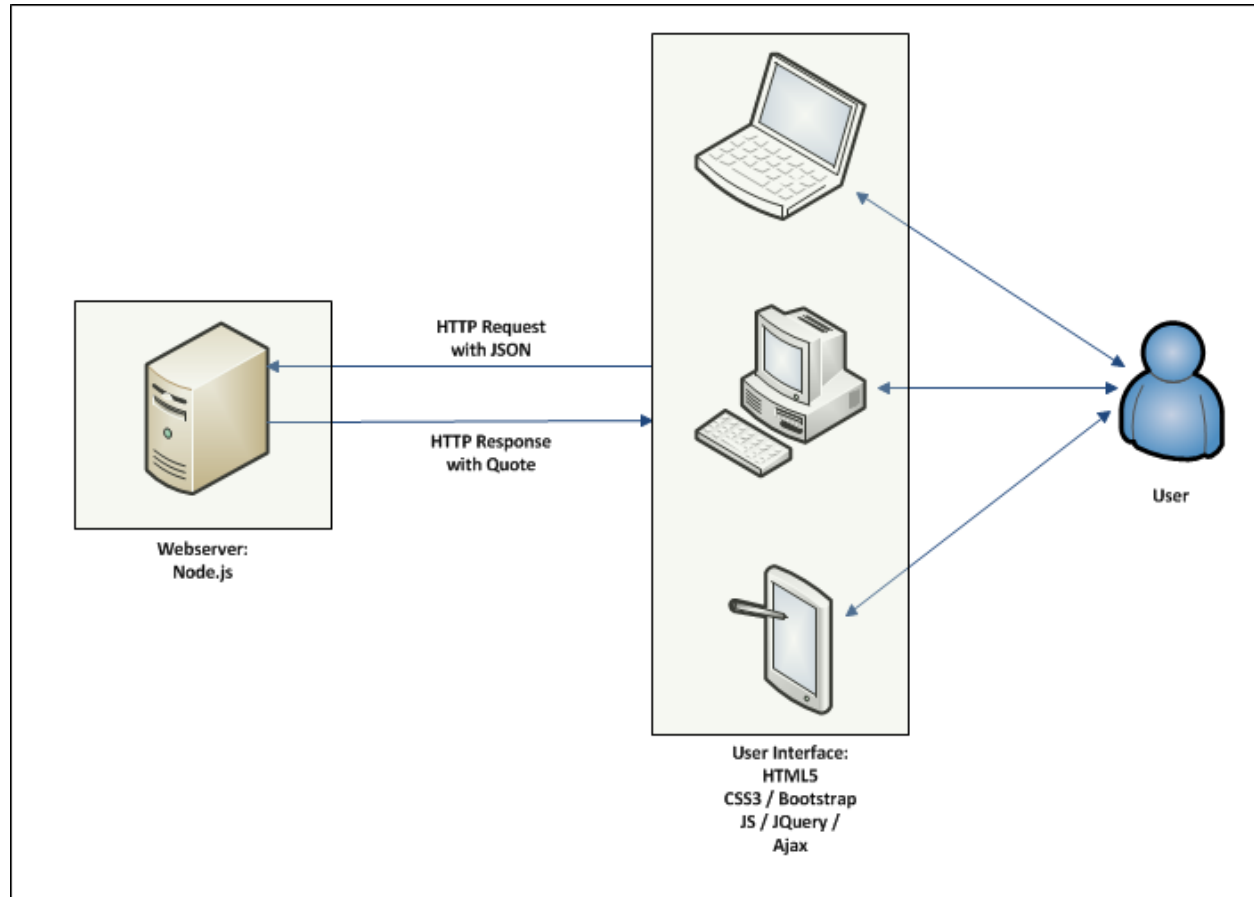
- For the project we are going to use GitHub as a means of source control
 - Source control is a way of maintaining files to ensure that any changes we make are tracked and can be reversed if issues occur.
- If you don't have a GitHub account setup, refer to the Work Experience Guide for the steps needed to create one.
- We need to setup a repository for your project
 - If you create multiple projects, you would want to create each in a separate repository
- The last step is to get the template project which we will be working on.

What's the challenge

- The challenge that we have for you today is to create a car insurance application to take in a user's details and return a quote for their vehicle.



How are we going to do it



HTML



- Html stands for Hyper-Text Markup Language
- It is used in conjunction with CSS to create the visual components of the internet which a browser reads and displays in visual format

```
<div id="dvPersonalDetails">
  <div id="dvPersonalDetailsAlert" class="alert alert-warning personalDetailsAlert">Error Message</div>
  <h1>Please enter your details</h1>
  <p>In order to provide you with a quote, we need a few details. Please complete the form below. Thanks.</p>

  <div class="row">
    <form class="form-horizontal">

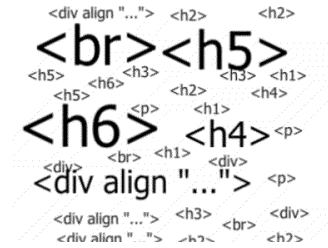
      <div class="form-group">
        <label for="nameInput" class="control-label col-md-2">Name</label>
        <div class="col-md-4">
          <input id="txtName" type="text" name="nameInput" class="form-control personalDetails" placeholder="Your Name" />
        </div>
      </div>

      <!-- The code for the remainder of the personal fields will be added here -->

      <div class="form-group">
        <div class="col-md-4">
          <input id="btnNext" value="Next" class="btn btn-default btn-sm btn-success" onclick="showCarDetails()" />
        </div>
      </div>

    </form>
  </div>
</div>
```

HTML



- Html is made up of a series of different tags which are opened and closed with `< >`. Each specify a different element.
- Some tags are purely for document structure
 - `<div>`
 - ``
- Other tags are used to add user controls to the page, with extra attributes to specify the details of that control
 - `<input id="textBox" type="text" />`
 - `<label>User Name:</label>`

[illegible]

- CSS (Cascading Style Sheet) is the language which is used to control how the page is laid out.
- It controls things such as page location, size, colour and many other options.
- These styles are applied to the html document in two main ways
 - By Id: `#textBoxId { width: 10px; }` would give the `textBoxId` element a width of 10 pixels.
 - By Class: `.textBox { width: 20px; }` would give all elements which have the `textBox` class on them, a width of 20 pixels.
- We are using a Bootstrap to remove the need to get too involved in CSS.

Twitter Bootstrap

- We are going to use Twitter's Bootstrap in our project. A bootstrap is a set of predefined styles which we can apply to our html to style our webpage with as little effort as possible.
- You can see in the template details page, that there are lots of classes applied to the html tags on the page. These are applying the styles in the bootstrap to our webpage elements.
- The benefits of using a bootstrap, apart from speeding up development, is that it also allows us to make our site responsive and faster. This bootstrap works on a grid which allows us to set page elements to be spread over a set portion of the grid.



JavaScript

- JavaScript is a sequential programming language. It runs in the client (aka the browser) and is local to the machine.
- Very powerful tool, with the ability to control the page elements and perform calculations, validations and many other functions.
- Code is added to a specific function which is called when a specific event has occurred such as a click on a button or the page loading.
- A function is written as:

```
function showCarDetails() {  
    // Hide the personal details section (dvPersonalDetails)  
    // Show the car details section (dvSectionDetails)  
}
```

Note the // symbol denotes a comment which will be ignored when **run**.

JavaScript

- As JavaScript is a language there are some extra concepts to know:
 - Variables
 - Conditional (If) statements
 - Loop structures
- Variables are like memory stores for pieces of data which we want to hold on too for later use.
- Conditional statements are useful when we only want an action to occur when some condition is true. Like saying I only want a new seat IF the one I have is broken.
- Loops are blocks of code (denoted by {...}) which we want to run multiple times over.

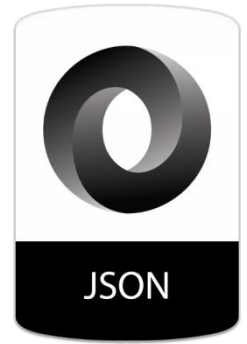
JQuery

- JQuery is a JavaScript library. In other words, it provides us with a set of tools for performing tasks within JavaScript, by providing an existing function to perform the task.
- For our project we will mostly be using it for selecting elements from our html document (DOM). It does this by using the “id”(ID Capitals?) and “class” selectors in our html.
 - `$("#textBoxId");` would return the element that has that ID (which is unique)
 - `$(".className");` would return the list of elements which has that class
- We can also use it to hide elements quickly:
 - `$("#element").hide();`
 - `$("#element").show();`
- Check out more JQuery functions at jquery.com.

What is Validation?

- Validation is a very important part of any application. It is the code that specifies that the user can only enter the desired information into the application.
- The reason that we need to validate inputs is for security. If we do not validate everything that comes into our server, then we run the risk that wrongful data could hang the server or worse, give back information private to you.
- If you do not validate your data, you leave your site open to hackers using SQL injection or cross-site scripting (XSS) attacks.
- To improve the user experience of the site we are going to be using JavaScript / JQuery to help us perform some basic validation.

JSON



- JSON stands for JavaScript Object Notation.
- It is the structure in which JavaScript holds an object, or simply a collection of data.
- It is a collection of “Key” : “Value” pairs, separated by a comma (,).
- JSON is a good way to pass information across a network as it is simple to create and can be strongly structured.
- An object can be made up of a single pair or a group of them, and the value of a pair can also represent an inner object.

AJAX

- AJAX stands for Asynchronous JavaScript and XML and is a client-side technology used to send and receive data to/from a server without disrupting the page flow.
- Asynchronous is the technical term to say that it allows other operations to occur while it is waiting for the server to reply.
- It creates a `HttpRequest` object which is sent to the server, which reads the request and replies with an `HttpResponse` object.
- Data can be sent through this request object, usually in the form of JSON, although it can be XML.
- We use it to send data to our server and get our quote back.

Node.js

- Node is a relatively new scripting language which has sprung up in popularity due to its ability to quickly create API's and servers.
- It allows commands to be written in JavaScript which run in Chrome's V8 engine and doesn't need to be run in a browser.
- Within node we create our server and define routes as functions, which run when that route is called by the client.
 - An example of a route is: `/api/calculateRates`
- It can differentiate between the different types of Requests (GET, POST, PUT, DELETE etc.)
- The server runs independently from the front end and can be called by different clients at the same time (it is asynchronous).

What's Left?



Come up with your own ideas to extend the API by adding in more features to the application. Or if you want to try and make the site look different you can start to play about with the CSS and try out different classes from the bootstrap. Maybe you could even add your own logo!



LibertyTM

**Information
Technology**