Daniel Collins

Student No. 20076240

Web App Development

Lecturer: D. Drohan  
The Internet of Things Yr. 3

Web App development

Assignment 2 Documentation

Contents

No table of contents entries found.

# Introduction

The first Assignment required for this module involved creating a Node.js "back-end" sever application based on application developed for keeping track of players, teams and available pitches. Three main models were created, "Players", "Teams", and "Pitches". As per the requirements of the assignment, CRUD functionality was implemented on these models, giving the user the ability to Create, Read, Update and Delete objects within each model.

This server was linked to MLab which enabled the persistence of the data. It was deployed on Heroku, making it “live”, which enabled anyone to access the server and routes. As we did not have a “Client” side of the application, testing was done with the use of a RESTfulAPI HTTP Web Server Client. The application “Postman” was used for this purpose. When all the core routes were working, some smaller, more cosmetic routes were added in.

For the second assignment in this Module, a Client side of the application is required to be linked to the server application produced for Assignment 1. This will provide a User-Friendly Interface for running the routes and provide some additional functionality such as presenting the data in an aesthetic manner. The application shall be written using Vue.js and Webstorm, a programme from Intellij. This application provides the user with multiple different “html” pages by utilising “components”. These are Vue.js elements which can hold the html code, css styling and javascript methods in the same file, easing the difficulty of having a large application with methods and stylesheets having to be referenced from separate pages. The following document will discuss some of the main functionalities of this application and shall provide an overview of how they were achieved.

# Assignment 1 – Recap of Functionality

The back end of an application is where most of the work is done. This will hold all the routes, the models for the components, and the practical and logical methods which eventually are referenced by the client server. This assignment contained four models from which the data objects were constructed. The four models used were;

* Pitches
* Players
* Teams
* Users

A model contains the schema for the objects. The list the required fields and their corresponding data types. As this application was linked to mLab, and the data was persisted, the models also contain the code for creating the collections within the database.

The routes are the connecting elements of the project. They specify what happens when an address is reached. Each model has a corresponding route object which hold the methods for each model. To delete a Player, for example, the delete function stored within the routes/players.js file would be called by the actual route element in the app.js file. The route in this file would specify the address required for the method to take place, followed by calling the method itself. If the address is incorrect or not reached, the method will not occur. A route in the app.js file follows the following template;

**app.get(‘/users’, users.findAll)**

This specific route calls a “get” HTTP method, which is used when retrieving and displaying data. The path specified is “/users”, so should that path not be reached, the method would not occur. The final section of the statement is telling the application what to do when it reaches this address. In this case, the application goes to the routes/users.js file, looks for the “findAll()” method and executes it. The application will then stay on this address until specified otherwise.