Technical Documentation

Meals on Wheels

Software Engineering

2017

# Introduction

## Overview

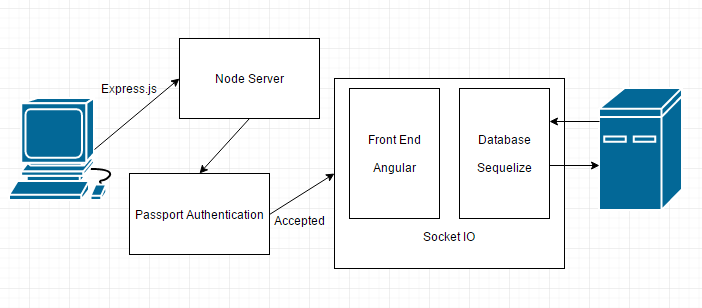
This document contains information about the project constructed by Otago Polytechnic Software Engineering (Semester-One-2017) class. Our task was to create a web based application for Meals on Wheels Company to handle & manage their order details as their current traditional paper based system has not adjusted well to growth.

## Objective

Our application had to run:

* Database – that correctly ordered and kept data valid & updated
* Front end web site – secure and easy to use interface for each employee to access
* Unique users & tasks – each user with a different login and personalized tasks
* Reliable server – making sure our website was up kept

## Design Diagram



## Technical Requirements

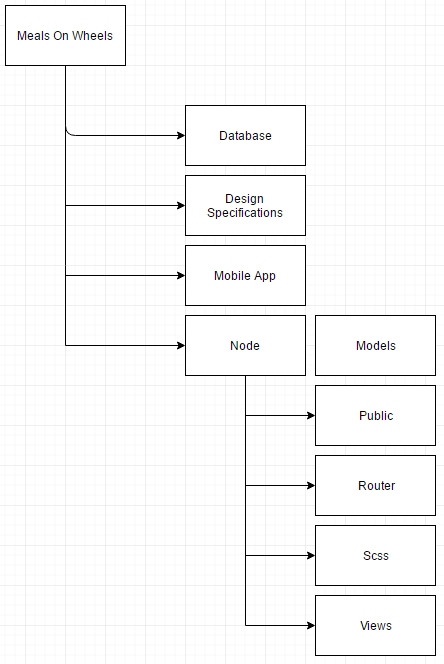
Node JS installed - <https://nodejs.org/en/download/>

Ruby - <https://www.ruby-lang.org/en/documentation/installation/>

Sass - <http://sass-lang.com/install>

MySQL - <https://dev.mysql.com/doc/workbench/en/wb-installing-windows.html>

## File Structure



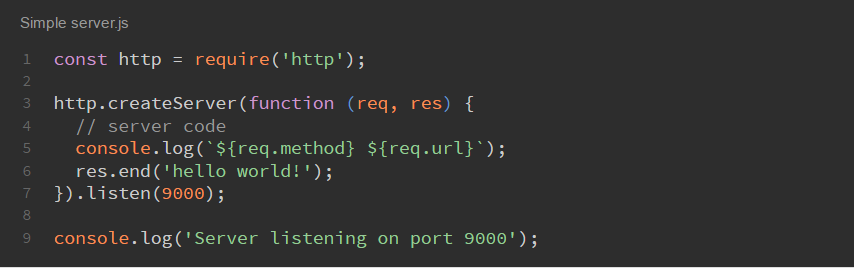
# Sections

## Node.js

*“Node.js is a platform built on Chrome's JavaScript runtime for easily building fast and scalable network applications. Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient, perfect for data-intensive real-time applications that run across distributed devices.”*

### Simple Node Server

Here is an example of a simple Node.js server.js folder. You would run this file to initialize your server and run it on your localhost domain loopback.



For this example; Open the browser and on <http://localhost:9000> and you will see the “hello world!” message.

http.createServer(function(req, res) – this function contains a call back that every time the client connects to the server the two parameters request and response request the clients information and send back to the client the data needed to run/display the website.

.listen(9000); - This allows you to set which port you want your server to run on.

## Express

*“Express is a minimal and flexible Node.js web application framework that provides a robust set of features to develop web and mobile applications. It facilitates the rapid development of Node based Web applications. Following are some of the core features of Express framework”*

### Key Features

* *“Allows to set up middleware to respond to HTTP Requests.”*
* *“Defines a routing table which is used to perform different actions based on HTTP Method and URL.”*
* *“Allows to dynamically render HTML Pages based on passing arguments to templates.”*

## Server Setup Steps

* Download Node onto the machine
* Open CMD in the server file location
* Run command: node server.js

## Dependency setup steps

Run commands:

* sudo apt-get install node.js
* sudo apt-get install npm
* sudo apt-get install node-legacy
* sudo apt-get update
* sudo apt-get upgrade
* sudo apt-get intstall mysql-server
* sudo apt-get mysql –u root –p 1234

## Angular

“AngularJS is a structural framework for dynamic web apps. It lets you use HTML as your template language and lets you extend HTML's syntax to express your application's components clearly and succinctly. AngularJS's data binding and dependency injection eliminate much of the code you would otherwise have to write. And it all happens within the browser, making it an ideal partner with any server technology.”

To install angular first open a command prompt window and type:

npm install angular

After this add the following to your index.html:

<script src="/node\_modules/angular/angular.js"></script>

In the addCustomers.ejs file angular is employed to gather values important to the “customer” element at run time. An example of angular being used is:

$scope.saveCustomer = {};

The $scope part of this line keeps all the variables inside scope of the function and stops from accessing variables that are outside of the function or out of scope. Both controllers and directives have reference to the scope, but not to each other. This arrangement isolates the controller from the directive as well as from the DOM.

## Node Passport

*“Passport is authentication middleware for Node. It is designed to serve a singular purpose: authenticate requests. When writing modules, encapsulation is a virtue, so Passport delegates all other functionality to the application. This separation of concerns keeps code clean and maintainable, and makes Passport extremely easy to integrate into an application.”*

Passport is a versatile add on for Node JS and has many different strategies

## Sequelize: MySQL, DAO (Data access object)

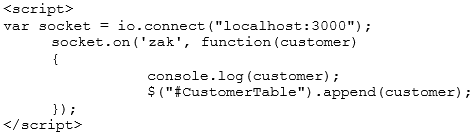
*“Sequelize is a promise-based ORM for Node.js v4 and up. It supports the dialects PostgreSQL, MySQL, SQLite and MSSQL and features solid transaction support, relations, read replication and more.”*

## Socket.IO

“Socket.IO is a JavaScript library for real-time web applications. It enables real-time, bi-directional communication between web clients and servers. It has two parts: a client-side library that runs in the browser, and a server-side library for Node.js. Both components have a nearly identical API. Like Node.js, it is event-driven.”

“Socket.IO primarily uses the WebSocket protocol with polling as a fallback option, while providing the same interface. Although it can be used as simply a wrapper for WebSocket, it provides many more features, including broadcasting to multiple sockets, storing data associated with each client, and asynchronous I/O.”

To install socket.io you must add it to your dependencies in the package.json file. After this you are ready to use socket.io in your product.

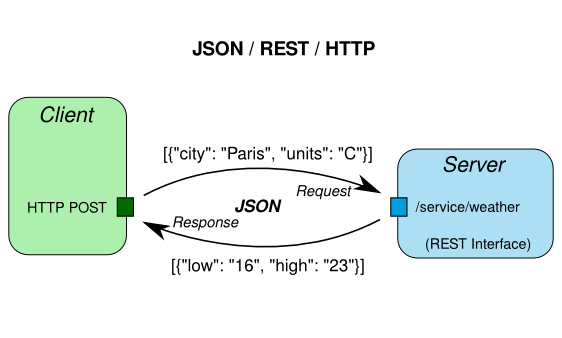


This section of socket.io code is used to connect to the database and update the website with changes that have been made on the fly. The first line var socket = io.connect(“localhost:3000”); sets the connection to the database as a variable that is called later with the socket.on line. Next 'zak' is a key given to the function, and that function when called logs the customer value (which is what the customer entered) then appends it to the table on the show customer’s page. The customer.log line is just to check the output for testing purposes.

## JSON parsing

### How it works?

The URLConnection class contains numerous techniques that let you speak with the URL over the system. URLConnection is a HTTP-driven class; that is, a hefty portion of its techniques are valuable just when you are working with HTTP URLs. Nonetheless, most URL conventions enable you to peruse from and keep in touch with the association. urlconnection getinputstream utilization.



After we make a case of JSONParser, we make a JSONObject by parsing the FileReader of our .json document. This JSONObject contains a gathering of key-esteem sets, from which we can get each estimation of the json document. To recover primitive articles, get() technique for the JSONObject's case is called, characterizing the predetermined key as a contention. It is critical to add the reasonable thrown to the strategy. For exhibit sorts in json document, JSONArray is utilized that speaks to a requested succession of qualities. As you can see in the code, an Iterator ought to be utilized as a part of request to take each estimation of the json exhibit. A structure in the json document, signs the production of another JSONObject to recover the qualities.