

# Webpage Documentation

Michael Panunto  
MTechHub

August 30, 2018

# 1 Webpage Directory

- `package.json` contains the data containing each package that is used by this project.
- `webpack.config.js` configuration for webpack which is used when running the server.

## 1.1 node\_modules

`node_modules` is an automatically generated directory from `node.js`. It contains the files for all the libraries being used by the project, you shouldn't ever need to touch it. With `node` installed, open the command prompt and navigate to the webpage directory. Typing `npm install` will automatically install all the required libraries that are listed in `package.json`.

## 1.2 php

Contains all the `php` files that are required.

- `db_config.php`: configuration data for the SQL Server database connection.
- `query.php`: executes a `sql` query and returns the result.

## 1.3 react-client

### 1.3.1 dist

`dist` contains `bundle.js`, `index.html`, `styles.css`; these files shouldn't need to be changed. They contain data required for displaying the react components. **Note:** If a stylesheet is required via `html` (e.g. `bootstrap`), add it to the `index.html` file.

### 1.3.2 src

`src` is the directory where all the webpage code is located. `index.jsx` is the first file that will be called on start. It contains the lines:

```
<Router>  
  <App />  
</Router>
```

which means that it will render the `App.jsx` file in a `Router` block which is used for navigation.

#### 1.3.2.1 components

`components` contains the bulk of the `jsx` (react) files.

- `App.jsx`: contains

```
  <NavBar />  
  <Routes />
```

this renders the Navigation bar, and `Routes` which allows users to switch tabs.

- `Devices.jsx`: this class controls the devices tab.
- `HomePage.jsx`: this class controls the main tab (home). It just calls `Dashboard.jsx` from the dashboard directory.
- `Sensors.jsx`: this class controls the sensors tab.
- `dashboard`
  - `Container.jsx`: container class for the Dashboard.
  - `CustomFrame.jsx`: frame class for the Dashboard.
  - `Dashboard.jsx`: The main dashboard that gets called on the homepage.
  - `AddWidgetDialog.jsx`: The modal that pops up when the user wishes to add a new widget.
  - `EditBar.jsx`: The bar that appears above each section when 'edit' is selected.
  - `Widgets`: See description for this folder below.
- `header`
  - `NavBar.jsx`: contains the navigation bar that displays at the top of the webpage. The login dialog is also dealt with inside this class.
- `routes`
  - `Routes.jsx`: contains the data required to support navigation to different tabs whenever a button is clicked.
- `style`
  - this directory contains the different css styling classes. Note that some styling is also done at the bottom of jsx classes.
- `widgets`
  - this directory contains the chart widgets that will be displayed on the dashboard. The general format for making a chart is:

```
<Chart
  chartType="ScatterChart"
  rows={this.state.data}
  columns={this.state.columns}
  options={this.state.options}
  graph_id="ScatterChart"
  width='100%'
  legend_toggle
/>
```

where data is stored in the state.

## 1.4 server

`server` contains the file `index.js` which is how the webpage is hosted via `express.js`. The port can easily be changed here through the `const PORT` variable, or in a `.env` file using the variable `PORT`.