Workshop 6 – Node and Sockets

This workshop is worth 2.5%

You must demonstrate your completed workshop to your tutor in your lab to receive the marks.

In this lab we will be using sockets with node.js to create a live chat feature.

Task 1 – Create a new project

This lab we will be creating code to generate a simple live Chat application that will allow multiple clients to broadcast text messages to the group.

This will be created using Node for the server, Angular for the front end and utilise sockets.io for the real-time chat functionality.

You will have two projects going at once.

Front end

- Create your angular project first. **ng new chat**
- Add any NPM dependencies that the front end code will need. (Bootstrap, socket.io)
- Use **ng serve –open** to host your angular front end application at http://localhost:4200.

Server

- Create a directory called “server” at the root level of the Angular website (same level as the “src” directory)
- Initialize this as a new NPM project with **npm init**.
- Add the NPM dependencies that the server side code is going to require
    1. Express.js
    2. Socket.io
    3. Cors
- Create the server code as three files.
    1. server.js // main file
    2. sockets.js //module to contain socket implementation
    3. listen.js // module to start node server listening for requests on port 3000
- Use **node server.js** or **nodemon server.js** to host your node server

Task 2 – Create a Chat Component

Create a component for a chat interface. This will include a HTML form with a text input element and a button.

Chat messages will be displayed in an unordered list below the form.

Add a route for this component so that it appears in the <router-outlet> when the app starts.

Clicking on the chat button will take the value currently in the text input and send (emit) it to the server via a service that holds the implementation of the required socket.io code.

Task 3 – Server side code for sockets.

Implement code (See lecture sides 6.5) that will accept a sockets connection and emit text messages to all connected clients as they arrive. (No express routes are quired for this to work)

Running the finished Application
Start the server listening at localhost:3000
Using a browser run a version of the client at http://localhost:4200
Open a second browser and run a version of the client at http://localhost:4200

Submit messages from each browser and watch them appear in each of the other browser.