



(Frontend Test)

INSTRUCTIONS:

- You should aim to spend between 3 to 4 hours on the exercise
- Return the solution or make it available to us on completion
- Please don't mention BeamTrail on your solution
- Please use **React.js** Library when creating your solution

TASK:

Your task is to create a Front-end UI displaying various IOT sensors based on data provided from the WebSocket endpoint using React.js library. User should be able to Connect and Disconnect sensors from the UI.

To help you we've included a **wireframe** (beamtrail_fe.zip/client) which should give you an idea of the UI structure – colors, shadows, fonts and others are up to you, make sure the UI is as appealing as possible.

WebSocket Server

Running

You are provided with a tiny backend server exposing WebSocket endpoint on the port **5000**. In order to run it, unzip the provided **beamtrail_fe.zip** and from the **/server** directory in your Terminal/CMD execute:
npm install && npm start

API

On the connection event (triggered automatically when client connects to the provided WebSocket server) the server will stream back to the client current state of the app of type **Sensor[]** where **Sensor** is defined as **{ id: string; name: string; connected: boolean; unit: string; value: string }**

The server also accepts a WebSocket messages of type **{ command: "disconnect" | "connect"; id: string }** where **id** is the Sensor's id – which changes the Sensor's state and sends the updates back to the client if the given Sensor is connected.

Hint

For the simplicity we recommend using [browser's native WebSocket implementation](#) to connect with the Backend server.

BONUS POINTS

- Add React component Test(s)
- If we were receiving messages from the Websocket Server in much higher rate, what optimization techniques could be used on the client side to improves the website's performance and limit the amount of React re-renders? Include your answer in the reflections.

Checklist

- Establish connection with the WebSocket Server
- Display sensors received from the WebSocket Server
- Connect/Disconnect the Sensor(s)

REFLECTION:

- What aspect of this exercise did you find most interesting?
- What did you find most cumbersome?