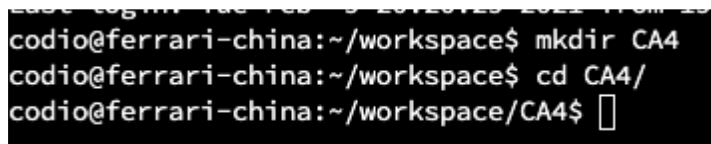


The following instructions are intended to help you get started with the sample code for the assessment with Node.js and Codio.

1. Access Codio via MyAberdeen.
2. Within Codio, create a new folder named **CA4** and navigate to that directory as shown below:

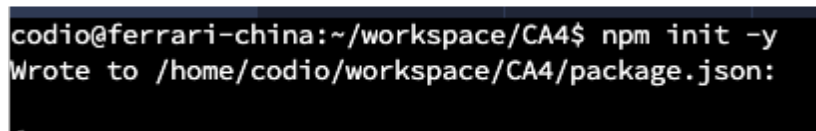
```
mkdir CA4
cd CA4
```



```
codio@ferrari-china:~/workspace$ mkdir CA4
codio@ferrari-china:~/workspace$ cd CA4/
codio@ferrari-china:~/workspace/CA4$
```

3. Within the CA4 folder enter the following command

```
npm init -y
```

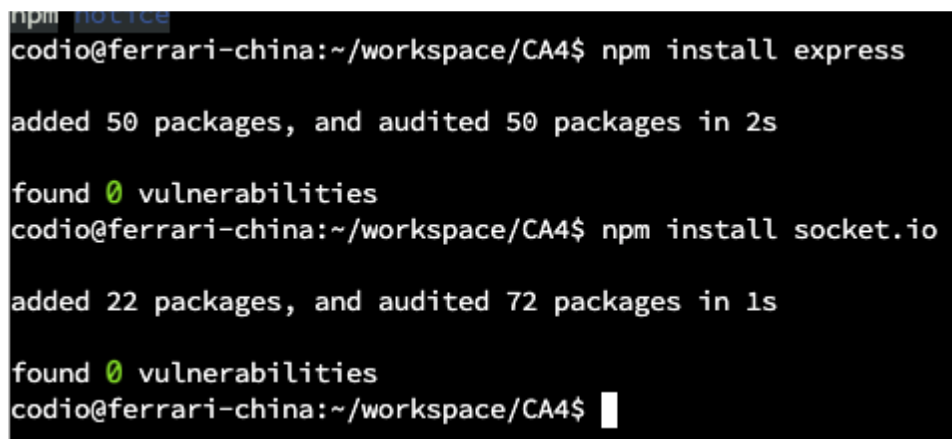


```
codio@ferrari-china:~/workspace/CA4$ npm init -y
Wrote to /home/codio/workspace/CA4/package.json:
```

This runs the node package manager and sets up an empty project for you. We will now use NPM to install some additional Node.js packages/modules which we will require for the chat application. Enter each of the following commands, one at a time.

```
npm install express --save
```

```
npm install socket.io --save
```



```
codio@ferrari-china:~/workspace/CA4$ npm install express
added 50 packages, and audited 50 packages in 2s

found 0 vulnerabilities
codio@ferrari-china:~/workspace/CA4$ npm install socket.io
added 22 packages, and audited 72 packages in 1s

found 0 vulnerabilities
codio@ferrari-china:~/workspace/CA4$
```

We have just installed the packages we require to develop the chat application.

4. Now, we will set up a few more things. Enter the following commands:

```
touch index.js

mkdir public

cd public

mkdir css

mkdir js

touch index.html

touch chat.html

cd css

touch demo.css

cd ..

cd js

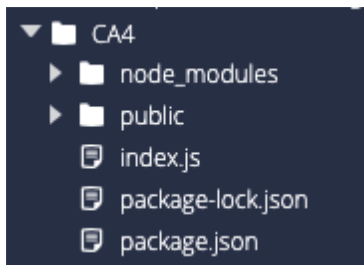
touch client.js

cd ..
```

A terminal window with a black background and white text. The prompt is 'codio@ferrari-china:~/workspace/CA4\$'. The commands and their outputs are: 'touch index.js', 'mkdir public', 'cd public/' (output: '/workspace/CA4/public\$'), 'mkdir css' (output: '/workspace/CA4/public/css\$'), 'mkdir js' (output: '/workspace/CA4/public/js\$'), 'touch chat.html', 'touch index.html', 'cd css' (output: '/workspace/CA4/public/css\$'), 'touch demo.css', 'cd ..' (output: '/workspace/CA4/public\$'), 'cd js' (output: '/workspace/CA4/public/js\$'), 'touch client.js', and 'cd ..' (output: '/workspace/CA4/public\$').

```
codio@ferrari-china:~/workspace/CA4$ touch index.js
codio@ferrari-china:~/workspace/CA4$ mkdir public
codio@ferrari-china:~/workspace/CA4$ cd public/
codio@ferrari-china:~/workspace/CA4/public$ mkdir css
codio@ferrari-china:~/workspace/CA4/public$ mkdir js
codio@ferrari-china:~/workspace/CA4/public$ touch chat.html
codio@ferrari-china:~/workspace/CA4/public$ touch index.html
codio@ferrari-china:~/workspace/CA4/public$ cd css
codio@ferrari-china:~/workspace/CA4/public/css$ touch demo.css
codio@ferrari-china:~/workspace/CA4/public/css$ cd ..
codio@ferrari-china:~/workspace/CA4/public$ cd js
codio@ferrari-china:~/workspace/CA4/public/js$ touch client.js
codio@ferrari-china:~/workspace/CA4/public/js$ cd ..
codio@ferrari-china:~/workspace/CA4/public$
```

We created the necessary files and folders for the assessment and returned to the main folder. We can use the Codio navigation panel to access these files and enter the demo code for the chat application.



5. We need to enter the demo code for the application. Open the index.js file by selecting it in the Codio navigation panel, as shown above. Enter the code for the index.js file you downloaded from MyAberdeen.

```
1  const express = require("express");
2  const socket = require("socket.io");
3
4  // App setup
5  const PORT = 5000;
6  const app = express();
7  const server = app.listen(PORT, function () {
8    console.log(`Listening on port ${PORT}`);
9    console.log(`http://localhost:${PORT}`);
10 });
11
12 // Static files
13 app.use(express.static("public"));
14
15 // Socket setup
16 const io = socket(server);
17
18 //we use a set to store users, sets objects are for unique values of any type
19 const activeUsers = new Set();
20
21 io.on("connection", function (socket) {
22   console.log("Made socket connection");
23
24   socket.on("new user", function (data) {
25     socket.userId = data;
26     activeUsers.add(data);
27     //we use the spread operator ... to pass the new user and the rest of
28     //active users in the emit
29     io.emit("new user", [...activeUsers]);
30   });
31
32   socket.on("disconnect", function () {
33     activeUsers.delete(socket.userId);
34     io.emit("user disconnected", socket.userId);
35   });
36
37   socket.on("chat message", function (data) {
38     io.emit("chat message", data);
39   });
40
41 });
```

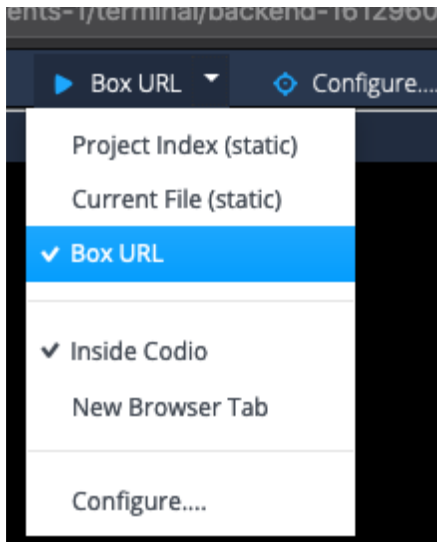
Do the same for each of the files you created in Step 4, using the files you downloaded from MyAberdeen.

6. It's time to test the application. Return to the terminal and enter the following command.

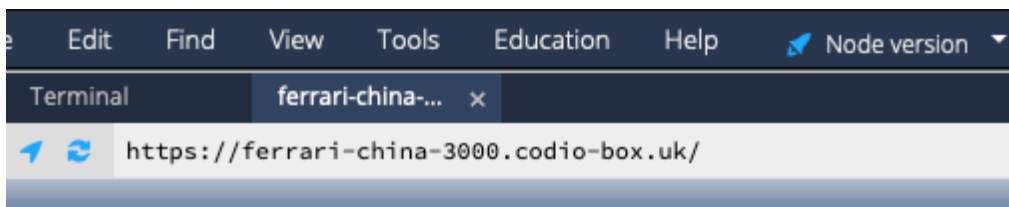
node index.js

```
Terminal x index.js
codio@ferrari-china:~/workspace/CA4$ node index.js
Listening on port 5000
http://localhost:5000
```

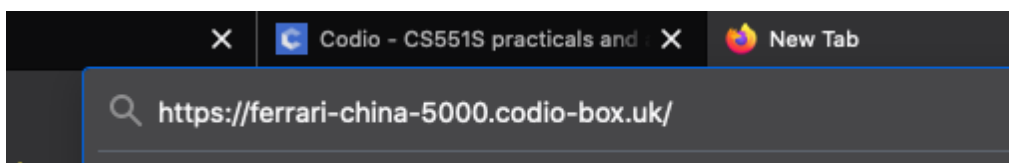
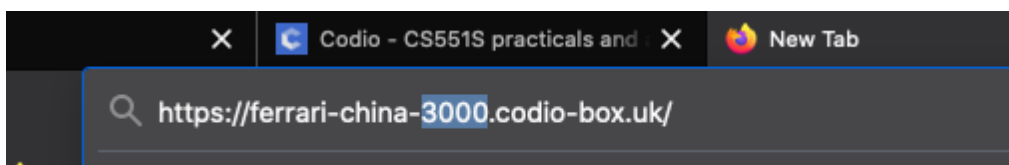
Find the domain name for your Codio box as shown below:



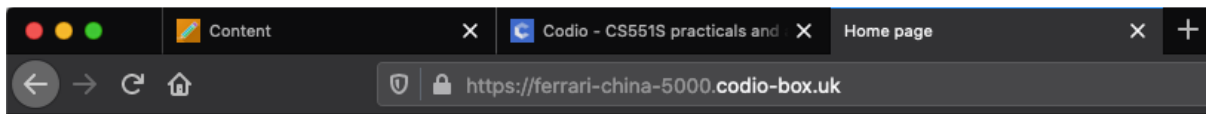
The internal browser in Codio opens with the domain name we need.



Open a new tab in your browser. Copy and paste the URL into the new tab. Change the port number to 5000 (instead of 3000).



This brings us to the index page of the application.

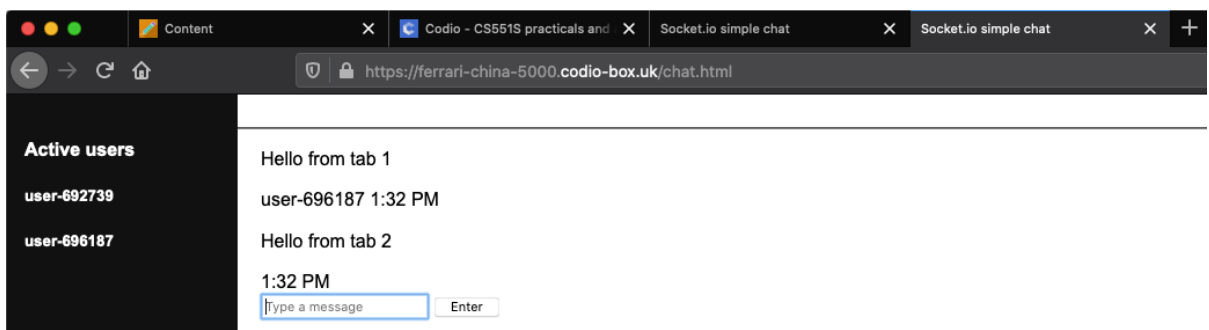
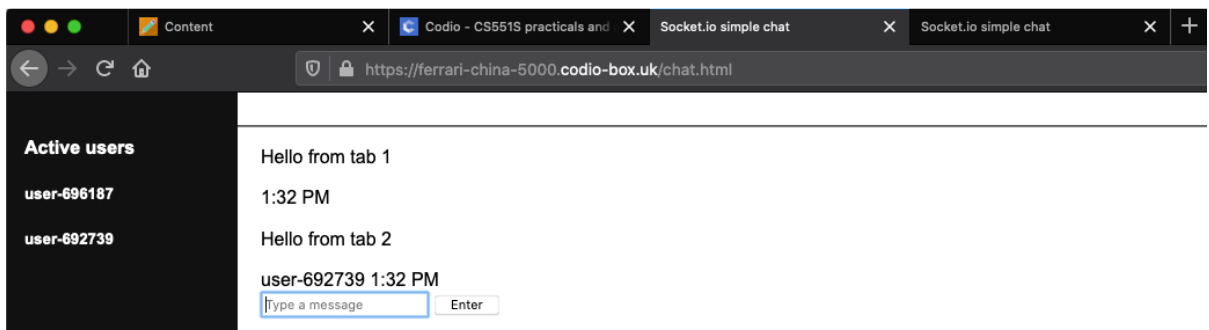


This is the home page

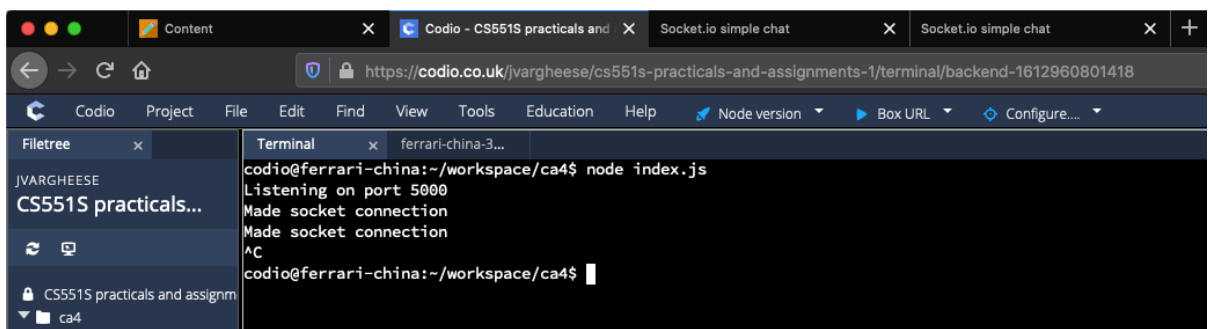
[Chat](#)

Open another tab with the same URL, so that we have two running in the browser. Check that the chat application is working as required.

In each tab, select the chat link which should bring up the chat application. Try sending messages from one tab to the other. You should see these displayed in both windows.



To stop Node.js return to the terminal and use [Control] + [C].



When you make changes, remember to stop the server and restart to view any updates you make.