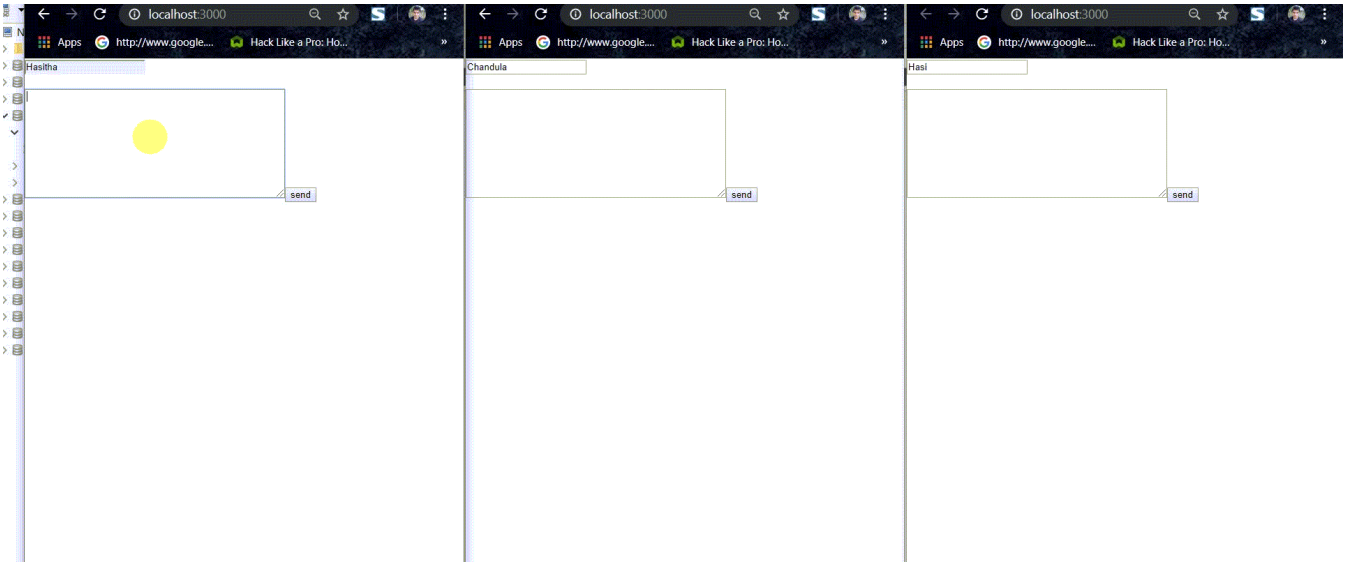
Real time chat app with React, Nodejs, MongoDb and Socket.io (Part 1 Serverside)

Hasitha Chandula
Nov 19, 2019 · 4 min read

. . .



I wanted to Build a real-time chat app using React and Nodejs and MongoDB. To do that the best solution is to use socket.io. After few hours Browsing internet but I didn't find the expected solution correctly. There are many tutorials for creating chat app with react-socket.io , react-nodejs, nodejs-socket.io but i didn't find real solution for (Real time chat app with React, Nodejs, MongoDB and Socket.io) this…

So After few Hours I understand how socket.io works with react and nodejs then i build this real time demo chat application.

first of all I create a node-express backend server by using git-bash command-prompt and vs code as text editor..

```
Hasitha@MSI MINGW64 ~/Desktop/chat-app
$ npm init
```

And i put default values (Press Enter to Everything) and created package.json file

```
package name: (chat-app)
version: (1.0.0)
description:
entry point: (index.js)
test command:
git repository:
keywords:
author:
license: (ISC)
About to write to C:\Users\Hasitha\Desktop\chat-app\package.json:

{
  "name": "chat-app",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "",
  "license": "ISC"
}


Is this OK? (yes) yes
```

Then I create my index.js file (Which is the main js file in server side)

To get express mongoose and other staff I need to install them in to my app folder. I install these dependencies to my app using npm i — save command.

```
Hasitha@MSI MINGW64 ~/Desktop/chat-app
$ npm i express mongoose socket.io concurrently nodemon --save
```

In this app I use these dependencies — express as nodejs framework mongooge to connect mongoDB, socket.io to make my app real time update, concurrently to run nodejs server and react app using one command and finally nodemon for restart my server when I save the js file (without nodemon we need to manually restart my server every time we make changes to our code)… We need to install some other dependencies to our React app but we will do the later (After we Create the React app).

```
+ mongoose@5.6.9
added 39 packages from 33 contributors, updated 4 packages and audited 2614 packages in 64.821s
found [92m0[0m vulnerabilities
```

After Installed all dependencies we get the messages like above. And When we look at our package.json file it will looks like this,

package.json > ...

```json
{
  "name": "chat-app",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "",
  "license": "ISC",
  "dependencies": {
    "concurrently": "^4.1.1",
    "express": "^4.17.1",
    "mongoose": "^5.6.9",
    "nodemon": "^1.19.1",
    "socket.io": "^2.2.0"
  }
}
```

(I forgot one dependency called config to install it we can simply do npm i config — save)

Now we can Start with Our backend. Here is my Code in index.js page

```
const express = require("express");
const socketIo = require("socket.io");
const connectDB = require("./config/db");

//Connect database
connectDB();
const app = express();

// INIT MIDDLEWARE
app.use(
```

```
  express.json({
    extended: true
  })
);

// Models
require('./models/Chat');



app.use(express.static(__dirname + "/public"));

// Port to Listen
const port = process.env.PORT || 5000;

const expressServer = app.listen(port, () => {
  console.log(`App Started at port ${port}`);
});

// Listen Socketio to our express app
const io = socketIo(expressServer);


// Variables
const Home = require("./routes/Home");

// GET Requests
// Pass app and io to used them in Home.js Page
Home(app, io);
```

My Database connection file code,

```
const mongoose = require("mongoose");
const config = require("config");
const db = config.get("mongoURI");

const connectDB = async () => {
  try {
    await mongoose.connect(db, {
        useNewUrlParser: true,
        useCreateIndex: true,
        useFindAndModify: false
    });
    console.log('Connected to the Database');
```

```
    console.error(err.message);
    process.exit(1);
  }
};

module.exports = connectDB;
```

In Here we use config dependency because we need to protect our database details if we use some clientIds and clientSecrects so i put all of them in a json file and with the help of config I can access them in any where in my app like I did in Here.. All we need to do is config.get('variable-name').. i will show my json file then you can get the point.

```
{
    "mongoURI": "mongodb://localhost:27017/chatapp",
    "jwtSecret":"[redacted]",
    "githubClientId" : "[redacted]",
    "githubSecret" : "[redacted]"
}
```

To Build this app we don't need jwtSecret, or github cliendId or secret. But Using config we can get all these files. And Also we need to install mongoDB local to your pc or you can do using mongoDB Online I use local mongoDB.

And To save data in MongoDB We need to create models for that. In here we only Need one model i called it chats model and here is it.

```
const mongoose = require('mongoose');

const Schema = mongoose.Schema;

const ChatSchema = new Schema({
    username: {
        type: String,
        required: true
    },
    message: {
        type: String,
        required: true
    },
    date: {
        type: Date,
        default: Date.now
    }
});

module.exports = Chat = mongoose.model('chats', ChatSchema);
```

We Only Store username name and message and the message date.. In Here first i create a Schema Using Mongoose and export it so we can access it anywhere in our

app..

And We need to create Home page Which in the Index page as a Variable,

```
// get user model
const mongoose = require("mongoose");
const Chat = mongoose.model("chats");

const Home = (app, io) => {
  // normal nodejs
  app.get("/api", async (req, res) => {
    const chatList = await Chat.find()
      .sort({ date: -1 })
      .limit(4);
    return res.json({ chats: chatList });
  });

  //   Socket io Part
  io.of("/").on("connect", async socket => {
    console.log("connected");

    socket.on("typing", async msg => {
        console.log(msg);
        socket.broadcast.emit("typing", { msg: msg.name });
      });

    // msg submit
    try {
      socket.on("msg", async msg => {
        // When the Request come save then
        const chatList = await Chat.find()
          .sort({ date: -1 })
          .limit(4);
        io.emit("msg", { chats: chatList });

        const chat = new Chat({
          username: msg.name,
          message: msg.msg
        });
        await chat.save();
        const chats = await Chat.find()
          .sort({ date: -1 })
          .limit(4);
        io.emit("msg", { chats: chats });
      });
    } catch (err) {
      console.error(err.message);
    }

    socket.on("typing", name => {
      io.emit("typing", { name: `${name.name}` });
    });
    socket.on("disconnect", () => {
      console.log("Disconnected");
    });
  });


};

module.exports = Home;
```

Here is My Home page Code..

Using React app first we load latest 4 messages in the database to do that we use axios in our react app when the axios request “http://localhost:5000/api” our backend server send the latest 4 message to the react app as json format.

After that the socket.io magic happens.. In socket.on means get data and socket.emit means send data io.emit means send data to all users and socket.emit means send data

to the one specific user which is who send the data to the server.

Now Our Serverside part is done. And Part2 I will Implement the Client side Using React.

React Nodejs Mongodb Express Socketio