WebSocket

Created By: Sajad Kamali

Advisor: Farid Karami

Course: React Native

Created At: 24 Jan 2020

WebSocket

- An internet communication protocol.
- It provides a full-duplex channel over a single TCP (Transmission Control Protocol) connection.
- A client and a server will be able to talk to each other in real time.
- **Socket.IO** is one of the most popular real-time engines for Node.js.

Socket.IO

- works by **Node.js events**
- listen for a connection event.
- emit a message.

Other Requirement libs

- **Express:** create server.
- **Axios:** make HTTP requests to the API:.

preparing the project _ Dark Sky API

- Create React native project.
- npm i axios express socket.io
- Create server.js with this codes:

```
const express = require("express");
const http = require("http");
const socketIo = require("socket.io");
const axios = require("axios");
const port = process.env.PORT || 8080;
const index = require("./routes/index");

const app = express();
app.use(index);
const server = http.createServer(app);
const io = socketIo(server); // < Interesting!
```

server.js

```
io.on("connection", socket => {
  console.log("New client connected"), setInterval(
    () => getApiAndEmit(socket),
    10000
  );
  socket.on("disconnect", () => console.log("Client disconnected"));
});
const getApiAndEmit = async socket => {
  try {
    const res = await axios.get(
      "https://api.darksky.net/forecast/1ed1ddbb4a7ab6edb9159c0a24db77ff/42.3601,-71.0589"
    ); // Getting the data from DarkSky
    socket.emit("FromAPI", res.data.currently.temperature); // Emitting a new message. It will be consumed by the
client
  } catch (error) {
    console.error(`Error: ${error.code}`);
  }
};
server.listen(port, () => console.log(`Listening on port ${port}`));
```

Routes/Index.js

- Create a file named **index.js** inside a **routes** directory:

```
const express = require("express");
const router = express.Router();

router.get("/", (req, res) => {
    res.send({ response: "I am alive" }).status(200);
});

module.exports = router;
```

App.js

```
import socketIOClient from 'socket.io-client';

class App extends Component {
constructor() {
      super();
      this.state = {
         response: false,
         endpoint: 'http://YOUR_SERVER_IP_ADDRESS:8080',
      };
   }
   componentDidMount() {
      const {endpoint} = this.state;
      const socket = socketIOClient(endpoint);
      socket.on('FromAPI', data => this.setState({response: data}));
   }
render() {
      const {response} = this.state;
      return (
         <Fragment>
            <View style={{textAlign: 'center'}}>
               {
                  response ? <Text>The temperature in Shiraz is: {response} °C</Text>
                  : <Text>Loading...</Text>
               }
            </View>
         </Fragment>
      );
   }
}
export default App;
```

Run project

- Node server.js
- React-native run-android

sources

- https://www.valentinog.com/blog/socket-react/