**What are web sockets?**

In a normal client server communication, a client sends a request to the server and the server responds to that request. So basically a client PULL information from the server.

Sometimes we have a different requirements. We want the server to notify the client about something happening on the server - say chat application.

We use WebSockets instead of HTTP. In HTTP a client sends a request and the server sends a response. A WebSockets is built on the top of HTTP - established via HTTP - they use HTTP handshake to upgrade the protocol to the WebSockets protocols that simply talk about how the data is exchanged.

**WebSockets solution**

There is plethora of packages for node express WebSockets but we are going to use *socket.io.* It needs to be configured on both - frontend and backend.

***npm install --save socket.io***

Backend*-*

***…***

mongoose

  .connect(MONGODB\_URI)

  .then(result => {

    const server = app.listen(3000);

    const io = require('socket.io')(server);

    // everytime a new client connects

    io.on('connection', socket => {

      console.log('Clinet Connected!')

    });

  })

  .catch(err => {

    console.log(err);

  });

Frontend-

<script src="/socket.io/socket.io.js"></script>

<script>

  var socket = io();

</script>

But here the problem with the backend current set up is - there is no anyway to share the socket io instances. For that -

// socket.js file

let io;

module.exports = {

    init: httpServer => {

        io = require('socket.io')(httpServer);

        return io;

    },

    getIO: () => {

        if(!io) {

            throw new Error('Socket.io not initialized');

        }

        return io;

    }

}

// app.js

...

mongoose

  .connect(MONGODB\_URI)

  .then(result => {

    const server = app.listen(3000);

    const io = require('./socket').init(server);

    // everytime a new client connects

    io.on('connection', socket => {

      console.log('Clinet Connected!')

    });

  })

  .catch(err => {

    console.log(err);

  });

// feed.ejs (backend)

const io = require('../socket');

io.getIO().emit('notification', {

    action: create,

    msg: 'A product created by Abhinav'

})

// notifications

<script src="/socket.io/socket.io.js"></script>

<script>

    var socket = io();

    socket.on('notification', data => {

        if(data.action === 'create') {

// add to notification div

            console.log(data.msg);

        }

    })

</script>