

Chats con Node (websocket)

Paloma
PML



se envia p a todos

bin

www → servidor

public

↳ stylesheets

↳ javascripts → chat.js → cliente

↳ prueba.html

Routes

↳ chat.js → ruta

↳ index.js

↳ users.js

views

↳ index.ejs

↳ users.ejs

↳ chat.ejs

```
<!DOCTYPE html>
<html>
  <head>
    <title><%= title %></title>
    <link rel='stylesheet' href='/stylesheets/style.css' />
    <script src="/socket.io/socket.io.js" defer></script>
    <script src="/javascripts/chat.js" defer></script>
  </head>
  <body>
    <ul id="messages"></ul>
    <form id="form">
      <input id="input">
      <button>Send</button>
    </form>
  </body>
</html>
```

```
const socket = io();

const form = document.getElementById('form');
const input = document.getElementById('input');
const messages = document.getElementById('messages');

form.addEventListener('submit', (e) => {
  e.preventDefault();
  if(input.value){
    socket.emit('chat', input.value);
    input.value = '';
  }
});

socket.on('chat', (msg) => {
  console.log("Mensaje recibido");
  const item = document.createElement("li");
  item.textContent = msg;
  messages.appendChild(item);
});
```

```
var express = require('express');
var router = express.Router();

/* GET home page. */
router.get('/', function(req, res, next) {
  res.render('chat', { title: 'SW1' });
});

module.exports = router;
```

```
var app = require('..../app');
var debug = require('debug')('ejemploexpressgenerator:ser');
var http = require('http');
const { Server } = require("socket.io");

/**
 * Get port from environment and store in Express.
 */

var port = normalizePort(process.env.PORT || '3000');
app.set('port', port);

/**
 * Create HTTP server.
 */

var server = http.createServer(app);
const io = new Server(server);
io.on('connection', (socket) => {
  console.log('User connected');
  socket.on('chat', (msg) => {
    console.log("Mensaje recibido del cliente " + msg);
    io.emit('chat', msg);
  });
  socket.on('disconnect', () => {
    console.log('Usuario desconectado');
  });
});
```

↓ es respo como está

cliente

Enviar

`socket.emit('chat',);`

Enviar

Mensaje q enviamos al servidor

`'Mensaje: ' + input.value`

`' + $user.textContent + $input.value`

Paloma: Hola!!

ESCUCHAR

`socket.on('chat', (msg) => {`

msg ya en el cliente

`});`

imprimirlo por pantalla

`const item = document.createElement("li");`

`item.textContent = msg;`

`messages.appendChild(item);`

`<ul id="messages">`

``

`const messages = getElementById('messages');`

views/chat.js

```
<div class="container m1-5">
  <h1 class="title">Servidor Express</h1>
  <p class="welcome-text">Bienvenido, <span id="username"><%= username %></span>, al chat</p>

  <form id="form" class="form">
    <input id="input" class="input-field" placeholder="Escribe tu mensaje aquí..." autofocus>
    <button type="submit" class="submit-button">Enviar</button>
  </form>
  <ul id="messages" class="message-list"></ul>
</div>
```

servidor

1. `io.on('connection', (socket) => {`

2. `io.emit('chat', (msg) => {`

`});`

Recibo mensaje

msg

Para todas

`io.emit('chat', msg);`

disconnect

`console.log("Usuario se desconecta")`

se envia username + p

bin/www

```
// Conexión (hay que escuchar)
io.on('connection', (socket) => {
  console.log('Usuario conectado');

  // Escuchar mensajes cliente
  socket.on('chat', (msg) => {
    // console.log(`${username} dice: ${msg}`);
    console.log('Desde el servidor recibimos: ${msg}');

    // Respuesta --> ENVIAR
    io.emit('chat', {msg});
  });

  socket.on('disconnect', () => {
    console.log('Usuario desconectado');
  });
});
```

public/chat.js

```
const socket = io();

const form = document.getElementById('form');
const input = document.getElementById('input');
const messages = document.getElementById('messages');
const username = document.getElementById('username');

form.addEventListener('submit', (e) => {
  e.preventDefault();
  if(input.value){
    console.log('soy ' + username.textContent);
    socket.emit('chat', {username: username.textContent, msg: input.value});
    input.value = '';
  }
});

// ESCUCHAR
socket.on('chat', (msg) => { // imp: ({msg}) si (json), (msg) NO!!
  //Recibo mensaje --> lo imprimo
  console.log('Mensaje recibido desde frontend: ' + msg);
  const item = document.createElement("li");
  // item.textContent = `${data.username}: ${data.msg}`;
  item.textContent = msg;
  messages.appendChild(item);
});
```