
Websockets+SocketIO

Gabriel Rodríguez Flores

February 2, 2023

- Despliegue de servidor websocket con NodeJS
- Uso con SocketIO

Contents

1	Teoría	3
1.1	Websockets	3
1.2	SocketIO	3
2	Ejemplos	3
2.1	Servidor	3
2.1.1	Nativo	3
2.1.2	Express	3
2.2	Ejemplo eventos servidor a cliente	4
3	Ejercicios	6
4	Entregables	6
4.1	En clase	6
4.2	Tarea	6

1 Teoría

1.1 Websockets

1.2 SocketIO

- Servidor
- Cliente
- Recibir eventos (on)
- Emitir eventos (emit)

2 Ejemplos

- Tutorial: Chat

2.1 Servidor

2.1.1 Nativo

```
1 import { createServer } from "http";
2 import { Server } from "socket.io";
3
4 const httpServer = createServer();
5 const io = new Server(httpServer, {
6   // options
7 });
8
9 io.on("connection", (socket) => {
10   // ...
11 });
12
13 httpServer.listen(3000);
```

2.1.2 Express

```
1 const express = require('express');
2 const socketio = require('socket.io');
3
4 const app = express();
```

```
5 const server = app.listen(3000, err => {
6   if(err){
7     console.log('Error');
8     return;
9   }
10  console.log('Servidor HTTP activo');
11 });
12
13 const io = socketio(server);
14
15 io.on('error', err => { console.log(err) });
16 io.on('connection', socket => {
17   socket.on('disconnect', () => {
18     console.log(` ${socket.handshake.address} has been disconnected`);
19   });
20   console.log(`A user connected from: : ${socket.handshake.address}`);
21 });
```

2.2 Ejemplo eventos servidor a cliente

- Cliente

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>Minimal working example</title>
6   <script src="https://cdnjs.cloudflare.com/ajax/libs/socket.io
7     /3.0.4/socket.io.js"></script>
8   <script>
9     window.onload = init;
10
11     function init() {
12       const ul = document.getElementById('events');
13
14       const newItem = content => {
15         const item = document.createElement('li');
16         item.innerText = content;
17         return item;
18       };
19       const socket = io('ws://127.0.0.1:3000', { withCredentials:
20         false});
21
22       socket.on('connect', () => {
23         ul.appendChild(newItem('connect'));
24       });
25     }
26   </script>
27 </head>
```

```
26 <body>
27   <ul id="events"></ul>
28 </body>
29 </html>
```

- Servidor

```
1  /* Módulo Express para montar el servidor Http */
2  const express = require('express');
3  const app = express();
4  const server = app.listen(3000, err => {
5    if(err){
6      console.log('Error');
7      return;
8    }
9    console.log('Servidor HTTP activo');
10 });
11 /* Exponer los ficheros estáticos */
12 app.use(express.static('public'));
13
14 /* Iniciar la escucha de sockets con el servidor Http */
15 const socketio = require('socket.io');
16 /* Necesario para CORS desde Socketv3 */
17 const socketioConfig = {
18   cors: {
19     origin: 'http://localhost:3000',
20     methods: ["GET", "POST"],
21   },
22 };
23 const io = socketio(server, socketioConfig);
24 /* Eventos sockets de escucha */
25 io.on('error', err => { console.log(err) });
26 io.on('connection', socket => {
27   socket.on('disconnect', () => {
28     console.log(` ${socket.handshake.address} has been disconnected`);
29   });
30   console.log(`A user connected from: : ${socket.handshake.address}`);
31 });
```

3 Ejercicios

4 Entregables

4.1 En clase

4.2 Tarea

- Realizar conexión entre cliente y servidor
- Intercambiar eventos