1. Crear carpeta socket.
2. Npm init -y (creamos el paquete package.json en forma mas directa sin toda la configuracion)
3. Importamos los paquetes express
4. Creamos carpeta Server-main.js
5. Creamos el servidor
6. var server = require('http').Server(app);//http esta por defecto!

Utilizamos http como servidor web en vez de usar express, porque utilizaremos la libreria socket.io y es mas conveniente para que funcione bien express y socket io en conjunto.

App = aplicacion express.

Al objeto socket.io le pasamos el servidor http creado que contiene la aplicacion express.

1. Npm install socket.io –save
2. Vemos que el servidor esta corriendo
3. Cannot GET / al no tener ninguna ruta asignada sucede esto, sin embargo el servidor esta corriendo
4. npm i nodemon -D (paquete de dependencias de desarrollo)
5. start: “nodemon server/main.js” esto permite escribir simplemente npm start
6. Haciendolo con socket--- Para esto estamos como escuchando un mensaje que llega del navegdor o de otro server. Para esto precisamos una pagina html con codigo js.
7. Importamos socket io en el html porque se utilizara la libreria en ambos sitios. Agregamos Tambien main.js que sera la parte javascript del lado publico
8. Cuando el cliente se conecta, en el servidor se ejecuta el método io.on(‘conncection’), emit.., lado cliente io.on(‘message’)….
9. Quitamos action en el form para que no se recargue la pagina y sea un poco mas real time
10. Onclick 🡪 llama a la function addMessage() una vez se haga click en el boton.
11. Para guardar los mensajes usamos un array, podria ser una base de datos cualquiera pero a modo de ejemplo usamos un array. Esto obviamente no es escalable ni para produccion.