

Working with Web Sockets

Request Response pattern

Request/response pattern

Client can send a request to the server anytime it wants, server can't do the same thing back to the client.

The server can't keep track of all the different IP addresses of all the different clients that are accessing it at any given time. That would be very expensive. sending a get request, sending a post request, that's one-way sort of communication where the client can send a request to the server and the server can send a response. However, the server can't then contact the client afterwards. The server can't initiate communication with the client.

Sub/Pub Pattern

Web-sockets allow real-time,two-way communication between the front-end and back-end.

WebSockets work is the client will open a persistent connection with the server and this is called a WebSocket and both parties in that situation will be able to send messages back and forth to each other. And this allows us to do things like real-time communication.

The socket.io library

- Npm install socket.io socket.io-client
server
`import { Server } from 'socket.io';`
`socket = new Server(httpServer)`
client
`import { io } from 'socket.client';`
`socket = io(serverURI)`
- `Socket.on('myEvent', callback);`
- `Socket.emit('myEvent', data);`