

objects

objcor

obcor

Server

①

io server connected

- on → ① user - joined → user joined
- ② disconnect → user left
- ③ message → message received

②

user joined

- ① add user in ^{self} array
- ② emit message to all users (user joined)

③

user left

- ① put current user to null
- ② ~~update key array~~
- ② update key array
- ③ delete all socket list
- ④ emit user left with user broadcast

③ Message Received

if → to public →
↳ broadcast message with data

else
↳ emit to the receiver with message with data

</body>

</html>

6.2 JavaScript

6.2.1 App JS

```
var express = require('express');
```

```
var http = require('http');
```

```
var socketio = require('socket.io');
```

```
var app = express();
```

```
var httpServer = http.Server(app);
```

```
var ioServer = socketio(httpServer);
```

```
var allSockets = {};
```

```
app.use(express.static(__dirname + '/public'));
```

```
function httpServerConnected(){
```

```
    console.log('Http Server started');
```

```
}
```

```
function ioServerConnected(socket){
```

```
    console.log('A new socket connection');
```

```
    socket.on('user-joined', userJoined);
```

```
    socket.on('disconnect', userLeft);
```

```
    socket.on('message', messageReceived);
```

```
}
```

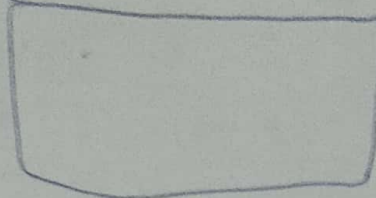
```
function userJoined(user){
```

html

cancel message

- ① read only
- ② disable whiteboard

release message



I want

① cancel

② user joined

③ user - left

④ message

⑤ join public user