

Bài 1 : Làm quen với SocketIO

1. Tạo thư mục **SERVER**
2. Tạo Project: **npm init**
3. Download các thư viện **express**, **ejs** và **socketio** về: **npm install express ej**s
4. Tạo file **index.js** để viết server express trong file này
5. Tạo thư mục views để chứa source code từ client: **trangchu.ejs**
6. Cài đặt thêm socketio vào thư mục **SERVER** rồi sau đó chỉnh sửa file **index.js** để lắng nghe xem có yêu cầu kết nối từ client hay không?
7. Chỉnh sửa file **trangchu.ejs** để thực hiện kết nối với server
8. Chỉnh sửa file **index.js** để lắng nghe xem có yêu cầu ngắt kết nối từ client hay không?

Bài 2 : Socket io cơ bản

1. Chỉnh sửa file trangchu.ejs để test xem nếu kích vào nút thì nó có bắt được sự kiện hay không?
2. Thực hiện gửi dữ liệu từ client lên server (emit), và server nhận dữ liệu từ client gửi lên (on)
3. Server thực hiện gửi dữ liệu trả về cho tất cả các client thông qua phương thức **io.sockets.emit**
4. Server thực hiện gửi dữ liệu trả về cho client vừa emit thông qua phương thức **socket.emit**
5. Server thực hiện gửi dữ liệu trả về cho các client, trừ client vừa emit thông qua phương thức **socket.broadcast.emit**
6. Bài Tập 1
 - a. Khách hàng gửi hai giá trị a, b lên server bằng phương thức emit
 - b. Server lắng nghe dữ liệu từ khách hàng gửi lên, sau đó tính tổng a+b, gửi kết quả về cho tất cả các client
7. Bài tập 2
 - a. Tạo các nút Đỏ, Xanh, Vàng ở trang của khách hàng
 - b. Khi người dùng bấm vào nút màu nào thì đổi màu nền của tất cả các trang khác
8. Bài Tập 3: (Tự luyện) Xây dựng ứng dụng chat online

Bài 3: Xây dựng web chat online

I – Mô tả

1. Người dùng đăng ký tên để chat: Người dùng sẽ gửi username lên server, nếu username đã tồn tại thì server thông báo đăng ký thất bại cho người dùng này biết, nếu đăng ký thành công thì thông báo cho người dùng này biết, đồng thời cũng thông báo cho các người dùng khác biết có một người mới đăng ký thành công
2. Nếu đăng ký thành công thì chuyển sang màn hình chat, gồm cột bên trái hiển thị tất cả những người đang online, cột bên phải hiển thị khung để chat, nếu người dùng nhập vào nội dung chat và bấm gửi thì thông tin chat sẽ được hiển thị lên máy của tất cả các người dùng
3. Nếu có một người dùng nào đó thoát ra thì xóa tên người đó khỏi danh sách những người đang online ở cột bên trái
4. Nếu có người đang nhập văn bản thì hiển thị ra thông báo để các người dùng khác biết

II – Các bước thực hiện

1. Xây dựng giao diện cho phần mềm
2. Thực hiện client gửi username lên server, server nhận kết quả gửi lên từ client, sau đó kiểm tra:
 - a. Nếu username đã tồn tại thì thông báo về client đăng ký thất bại
 - b. Nếu đăng ký thành công thì ẩn form đăng ký đi, đồng thời hiện form chat ra với user vừa gửi form đăng ký
 - c. Cập nhật lại danh sách tất cả các user đang online trên trang của tất cả các người dùng
 - d. Nếu có người dùng thoát ra: hiện thị lại form login cho người ấy đồng thời xóa người này ra khỏi danh sách đang online
 - e. Nếu có người dùng mất kết nối với server: xóa người này ra khỏi danh sách đang online
3. Chức năng chat
 - a. Khi người dùng nhập nội dung vào khung chat và nhấn nút SEND: lấy nội dung trong khung chat và emit lên server
 - b. Sever nhận dữ liệu từ client gửi lên, sau đó xác định username của client vừa emit thông tin lên, sau đó gửi trả lại cho các client gồm username và nội dung chat
 - c. Client nhận thông tin từ server emit về và hiển thị ra khung chat
 - d. Nếu có ai đó đang nhập đoạn chat thì hiển thị cho các user khác được biết

Bài 4: Room & Namespace

I – Lý thuyết

1. Không có hàm tạo ROOM mà mỗi khi một client truy cập vào localhost, server sẽ tạo một socket cho client và được chứa trong 1 ROOM
2. Một ROOM có thể chứa nhiều socket, nếu muốn socket vào trong ROOM thì sử dụng phương thức join vào ROOM
3. Một socket có thể được chứa trong nhiều ROOM

II- Mô tả

1. Ban đầu, khi mỗi client truy cập localhost thì được tạo sẵn một ROOM
2. Khi client nhấn nút tạo ROOM, client sẽ gửi tên room lên server, server sẽ nhận tên room, sau đó đưa socket vào trong ROOM vừa tạo
3. Server tạo mảng để chứa tất cả các ROOM, sau đó emit về cho các client , đồng thời cũng emit tên ROOM về cho client vừa gửi tên ROOM lên!
4. Client nhận mảng các ROOM và hiển thị ra trình duyệt

III – Các bước thực hiện

1. Tạo giao diện cho trình duyệt
2. Client thực hiện gửi tên ROOM lên server
3. Server nhận tên ROOM từ client gửi lên, sau đó tạo ra một ROOM với tên của client gửi lên, và đưa socket vào trong ROOM này
4. Server tạo ra một mảng lưu trữ tất cả các ROOM, sau đó gửi mảng này về lại cho các client, đồng thời cũng gửi tên ROOM này về cho client vừa gửi tên ROOM lên
5. Client nhận kết quả từ server gửi về và hiển thị ra trình duyệt