-Websocket hoạt động không dựa trên giao thức Http mà nó hoạt động dựa trên giao thức TCP/IP.

-Cách hoạt động:

+)HTTP hoạt động theo kiểu hang đợi là khi có nhiều request đến cùng 1 lúc thì nó sẽ xử lý từng thằng một.

+)Websocket hoạt động dựa trên cho tất cả vào 1 phòng và khi có thay đổi trong đó thì tất cả mọi người đều có thể thấy được.

Font-end:

\*onConnect

-let Sock=new Sock(“http:://localhost:8080/ws”) -> tạo ra 1 handshake với back-end và ws là 1 endpoint được tạo ra bởi config của websocket ở back-end.

-stompClient=over(Sock)

-stompClient.connect({},onConnected,onError); -> nhận vào 3 tham số.

+)Tham số đầu tiền là tuy chọn cấu hình cho qua trình kết nối.

+)Tham số thứ 2 sẽ là thường thường là 1 hàm xử lý gì đó khi quá trình kết nối thành công.

+)Tham số thứ 3 sẽ là hàm xử lý khi kết nối không thành công.

-stopmClient.subcribe(“/chatroom/public”,onMessageReceive) -> có 2 tham số là:

+)Địa chỉ Broker mà người đó join vào.

+) onMessageReceive sẽ chạy khi nhận đươc một thông điệp từ kênh đã đăng ký.

-stopmClient.send(“/app/message”,{},JSON.stringify(chatMessage)) -> nhận vào 3 tham số:

+)Tham số đầu tiên là destination mà đã được cấu hình trong websocket config nó sẽ áp dụng cho tất cả các api nào muốn áp dụng websocket còn “message” phía sau nó sẽ là @MessageMapping được định nghĩa ở controller thành cho các requestMapping.

+) Tham số thứ 2 là cấu hình của quá trình gửi.

+) Tham số thứ 3 là đối tượng gửi tinh nhắn gôm có người gửi , nội dung gửi.

-BE Config

+)ta sẽ đánh dấu @EnableWebSocketMessageBroker -> kích hoạt websocket

+)thực hiện implement interface WebSocketMessageBrokerConfigure

+)thực hiện ghi đi 2 phương thức registerStompEndpoint và configureMessageBroker

+)trong phương thức registerStompEndpoint ta sẽ đăng ký 1 đường dẫn đích -> đường dẫn này là để ta thực hiện bước đầu tiền của kết nối Websocket là handshake.

+)Trong phương thức configureMessageBroker ta sẽ thực hiện enableSimpleBroker -> định nghĩa đích đến của nhà môi giới -> thì ta sẽ dùng cái này để thực hiện công việc subcribe.Thực hiện setApplicationDestinationPrefix -> thực hiện đĩnh nghĩa là 1 đường dẫn và đường dẫn này sẽ áp dụng cho tất cả các method ở controller mà nó được sử dụng @MessageMapping khi mà ở font-end thực hiện method send.