***- Fazendo exemplo de juntando canais de comunicação***

Para fazer um teste simulando salas diferentes, ou seja canais de socket diferente, vamos fazer uma adaptação na view e no app.js

View > chat.ejs

        <script>

            // instancia do obj que se conecta com websocket

            var socket = io('http://localhost:3000');

            var conversation\_id = 1001;

            if($('#apelido').val() == 'paulo' || $('#apelido').val() == 'outro'){

                var conversation\_id = 1002;

            }

            socket.emit('subscribe', conversation\_id);

            // precisamos emitir a para o servidor a mensagem digitada pelo cliente

            $('#enviar\_mensagem').click(function() {

                socket.emit(

                    'msgParaServidor',

                    {room: conversation\_id, apelido: $('#apelido').val(), mensagem: $('#mensagem').val()}

                );

                $('#mensagem').val("");

            });

            // escutar msgParaCliente

            socket.on('msgParaCliente', function(data){

                var html = '';

                html += '<div class="dialogo">';

                    html += '<h4>' + data.apelido + '</h4>';

                    html += '<p>' + data.mensagem + '</p>';

                html += '</div>';

                $('#dialogos').append(html);

                // deixar a tela para mostrar a mensagem mais atualizada

                window.scrollTo(0, document.body.scrollHeight);

            });

        </script>

Na raiz e arquivo app.js

/\* criar a conexão por websocket \*/

// funcao ON vai estar escutando req lado do cliente e vice-versa

// estamos escutando eventos de connection, evento padrão do socket.io (propria conexao por parametro [socket])

io.on('connection', function (socket) {

console.log('Usuario conectou');

socket.on('disconnect', function () {

console.log('Usuario desconectou');

});

// vou fazer a junção no canal

socket.on('subscribe', function (room) {

console.log('joining room', room);

socket.join(room);

});

// escutando a mensagem que veio lá do cliente

socket.on('msgParaServidor', function (data) {

console.log("conversa na sala: " + data.room);

// emitir de volta para o cliente, passando apelido e mensagem

// envia para todos da sala, incluindo o remetente tbm

io.sockets.in(data.room).emit('msgParaCliente',

{ apelido: data.apelido, mensagem: data.mensagem }

);

// enviar para todos da sala via broadcast, deixa de fora o remetente

/\*socket.broadcast.to(data.room).emit('msgParaCliente',

{ apelido: data.apelido, mensagem: data.mensagem }

);\*/

});

});

/\* fazer o teste de comunicação entre usuário e outro privado, se estiverem no mesmo canal vai haver a conversa \*/