**CRIAÇÃO DE UM CHAT UTILIZANDO O SOCKET IO**

**CONFIGURAÇÃO DO PROJETO**

1. Vamos utilizar a mesma base do código da sessão 25 e dar prosseguimento para a criação do chat.
2. No frontend no arquivo index.ejs, fazer as alterações como abaixo.
3. <!DOCTYPE html>
4. <html lang="en">
5. <head>
6. <meta charset="UTF-8">
7. <meta http-equiv="X-UA-Compatible" content="IE=edge">
8. <meta name="viewport" content="width=device-width, initial-scale=1.0">
9. <title>Document</title>
10. </head>
11. <body>
12. <h1> Chat do Guia </h1>
13. <div id="chat">
14. </div>
15. <input type="text" name="" id="msg"><br>
16. <input type="text" name="" id="username" placeholder="Seu nome de usuário">
17. <button>Enviar</button>
19. </body>
20. <script src="https://cdn.socket.io/4.4.1/socket.io.min.js"
21. integrity="sha384-fKnu0iswBIqkjxrhQCTZ7qlLHOFEgNkRmK2vaO/LbTZSXdJfAu6ewRBdwHPhBo/H"
22. crossorigin="anonymous">
23. </script>
24. <script>
25. *var* clienteFrontEnd = io("http://localhost:3000");
26. clienteFrontEnd.on("disconnect", ()*=>*{
27. console.log("Desconectado")
28. })
29. </script>
30. </html>
31. No backend faremos as alterações como abaixo;
32. *var* express = require("express");
33. *var* app = express();
34. //colocando a aplicação express para rodar dentro do servidor http nativo do node
35. *var* http = require("http").createServer(app);
36. *var* io = require("socket.io")(http);
37. //evento de conexão do socket.io
38. io.on("connection",(*clienteBackEnd*) *=>* {
39. *clienteBackEnd*.on("disconnect",() *=>* {
40. console.log("Cliente X desconectou "+ *clienteBackEnd*.id);
41. })


45. })
46. //configuando a view do projeto para utilizar o ejs
47. app.set("view engine","ejs");
48. //criando rota principal que irá renderizar o arquvio index.ejs
49. app.get("/", (*req*, *res*) *=>* {
50. *res*.render('index')
51. })
52. http.listen(3000, () *=>* {
53. console.log("Servidor rodando");
54. })

**ENVIANDO MENSAGEM**

1. No frontend criamos a função enviar que irá capturar os dados dos inputs e através do método emit do socket clientBackEnd vamos emitir esses dados em um json.
2. *function* enviar(){
3. *var* msgField = document.getElementById("msg");
4. *var* usernameField = document.getElementById("username");
5. *var* msg = msgField.value;
6. *var* username = usernameField.value;
7. clienteFrontEnd.emit("msg",{
8. msg: msg,
9. username: username
10. })
11. }
12. No botão enviar vamos chamar a função enviar no evento onclick.
13. <button onclick="enviar()">Enviar</button>

**RECEBENDO A MENSAGEM NO BACKEND**

1. No banckend utilizamos o método on do socket clienteBackEnd vamos ouvir os eventos enviados do frontEnd pelo método ‘msg’. vamos ainda configurar o backend para enviar um emit utilizando o método ‘showmsg’ que será configurado no frontend.
2. *clienteBackEnd*.on("msg",(*data*)*=>*{
3. *clienteBackEnd*.emit('showmsg',*data*);
4. console.log(*data*);
5. })
6. No frontend vamos novamente utilizar o método ‘on’ do socket clienteFrontEnd e vamos receber qualquer dados que seja enviado pelo método ‘showmsg’ enviado pelo backend e mostrála no console do frontend.
7. clienteFrontEnd.on('showmsg',(*data*) *=>* {
8. console.log(*data*);
9. })
10. Vamos agora pegar a mensagem do backend e inserir na página do frontend.
11. No frontEnd dentro do método que recebe os dados do backend pelo evento ‘showmsg’, vamos criar uma variável chamada ‘chat’, que irá capturar a div que tem a id chamada ‘chat’.
12. Vamos ainda criar uma variável que irá receber um parágrafo com o ‘document.createElement()’
13. Depois vamos passar os dados para o parágrafo utilizando o ‘.innerHTML’ do elemento criado
14. E adicionar este parágrafo dentro da div ‘chat’ com o ‘.append()’ no elemento que irá receber o novo item.
15. clienteFrontEnd.on('showmsg',(*data*) *=>* {
16. console.log(*data*);
17. *var* chat = document.getElementById("chat");
18. //criando um novo parágrafo
19. *var* p = document.createElement('p');
20. //adicionando os dados passados do backend para o parágrafo
21. p.innerHTML = 'Id: '+*data*.id+' Nome: '+ *data*.username +': '+*data*.msg;
22. //inserindo o parágrafo na página dentro da div
23. chat.append(p);
24. })

**EMISSÃO GLOBAL DAS MENSAGENS**

1. Vamos agora configurar a aplicação para que a mensagem seja enviada para todos os usuários da aplicação.
2. No backEnd vamos alterar o método ‘.on’ do socket clienteBackEnd que recebe os dados passados pelo evento ‘msg’ enviado pelo frontEnd. Iremos substituir o ‘clientBackEnd.emit’.
3. Temos a opção de substiruir por ‘io.emit’ ou ‘clienteFrontEnd.broadcast.emit’.
4. No clienteFrontEnd.broadcast.emit a mensagem será enviada para todos os usuáiro mas não para o mesmo usuário.
5. No ‘io.emit’ por que da forma anterior a mensagem somente é passada pelo mesmo cliente, já utilizando o ‘io.emit’ a mensagem será enviada para todos os usuário conectados no servidor ‘io’.
6. *clienteBackEnd*.on("msg",(*data*)*=>*{
7. io.emit('showmsg',*data*);
8. console.log(*data*);
9. })
10. Ou
11. *clienteBackEnd*.on("msg",(*data*)*=>*{
12. *clienteBackEnd*.broadcast.emit('showmsg',*data*);
13. console.log(*data*);
14. })

**MELHORANDO O ASPECTO DA APLICAÇÃO COM BOOTSTRAP**

1. Na página do bootstrap copie o cdn para o head da aplicação e os scripts para o fim do body.
2. <!DOCTYPE html>
3. <html lang="en">
4. <head>
5. <meta charset="UTF-8">
6. <meta http-equiv="X-UA-Compatible" content="IE=edge">
7. <meta name="viewport" content="width=device-width, initial-scale=1.0">
8. <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css" integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm" crossorigin="anonymous">
9. <title>Document</title>
10. </head>
11. <body>
12. <h1> Chat do Guia </h1>
13. <div id="chat">
14. </div>
15. <hr>
16. <input type="text" name="" id="msg"><br>
17. <input type="text" name="" id="username" placeholder="Seu nome de usuário">
18. <button onclick="enviar()">Enviar</button>
19. <!--Script para o bootstrap-->
20. <script src="https://code.jquery.com/jquery-3.2.1.slim.min.js" integrity="sha384-KJ3o2DKtIkvYIK3UENzmM7KCkRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN" crossorigin="anonymous"></script>
21. <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js" integrity="sha384-ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q" crossorigin="anonymous"></script>
22. <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js" integrity="sha384-JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PVCmYl" crossorigin="anonymous"></script>
23. </body>
24. Vamos melhorar a aparência dos inputs utilizando a classe ‘form-control’ e o overflow em uma div para que os inputs e botão não desçam para o fim da página.
25. Para isso vamos utilizar um style e configurar o id da div chat.
26. <style>
27. #chat{
28. *max-width*: 500px;
29. *max-height*: 500px;
30. *background-color*: aliceblue;
31. *overflow-y*: scroll;
32. }
33. </style>
34. <body>
35. <h1> Chat do Guia </h1>
36. <div id="chat">
37. </div><br>
38. <input class="form-control" type="text" name="" id="msg" placeholder="Mensagem..."><br>
39. <input class="form-control" type="text" name="" id="username" placeholder="Seu nome de usuário"><br>
40. <button type="button" class="btn btn-primary btn-lg btn-block" onclick="enviar()">Enviar</button>
41. </body>