

Pràctica 2

Meriem Abjil Bajja

Universitat de Girona
Xarxes troncales i serveis públics de dades

28 de gener de 2019

Índex

- 1 Introducció
- 2 Entorn de treball
- 3 Creació del Chat amb el Socket IO
- 4 Resultats
- 5 Demo

Índex

- 1 Introducció
- 2 Entorn de treball
- 3 Creació del Chat amb el Socket IO
- 4 Resultats
- 5 Demo

Índex

- 1 Introducció
- 2 Entorn de treball
- 3 Creació del Chat amb el Socket IO
- 4 Resultats
- 5 Demo

Índex

- 1 Introducció
- 2 Entorn de treball
- 3 Creació del Chat amb el Socket IO
- 4 Resultats
- 5 Demo

Índex

- 1 Introducció
- 2 Entorn de treball
- 3 Creació del Chat amb el Socket IO
- 4 Resultats
- 5 Demo

Índex

- 1 Introducció
- 2 Entorn de treball
- 3 Creació del Chat amb el Socket IO
- 4 Resultats
- 5 Demo

S'ha de realitzar una aplicació de xat, Client-Servidor (CS), basada en el navegador, amb **Node.JS** i **socket.io**.

Els requisits de realitzar l'aplicació són:

- el client de xat s'ha de fer amb **HTML**, **JS** i **socket.io**
- el servidor web (on s'allotja la pàgina web amb el client de xat) s'ha de fer amb **Node.JS** (i **Express**)
- el servidor de xat s'ha de fer amb **Node.JS** i **socket.io**
- s'ha de fer un sol script, que fa alhora de servidor web i de servidor de xat

A part, s'han de realitzar dues millores entre les següents:

- 1 que el servidor no reenvii el missatge al client que l'envia sinó que això ho faci el mateix client
- 2 **que hi hagi el nickname de l'usuari al davant de cada línia**
- 3 **que quan un usuari entri/surti del xat es difongui a tots els usuaris un missatge informant-ho**
- 4 que hi hagi "sales" de xat

Les millores aplicades són la 2 i la 3.

Índex

- 1 Introducció
- 2 Entorn de treball**
- 3 Creació del Chat amb el Socket IO
- 4 Resultats
- 5 Demo

Cal el següent:

- una 1a màquina amb el servidor web i el servidor de xat (VM)
 - una 2a i 3a (i potser més) màquines amb un navegador *
- (Màquina real i mòbil)

Les tres (o més) màquines unides per una xarxa.

Índex

- 1 Introducció
- 2 Entorn de treball
- 3 Creació del Chat amb el Socket IO**
- 4 Resultats
- 5 Demo

Els Sockets permeten que el servidor pot enviar missatges als clients. Cada vegada que s'escriu un missatge de xat, la idea és que el servidor l'obtingui i l'empeny a tots els altres clients connectats.

Primer anem a crear un fitxer **package.json** que descriu el nostre projecte.

Codi del *package.json*

```
1 {  
2   "name": "nodechat",  
3   "version": "0.1.0",  
4   "description": "Chat Socket IO",  
5   "main": "index.js",  
6   "repository": "",  
7   "author": "Meriem Abjil Bajja",  
8   "dependencies": {  
9     "express": "^3.2.4",  
10    "socket.io": "^0.9.14"  
11  }  
12 }
```

Ara que s'ha descrit el projecte, podem crear un arxiu **index.js** que configurarà la nostra aplicació.

Codi *index.js*

```
1 /var express = require('express'),  
2   app = express(),  
3   server = require('http').createServer(app),  
4   io = require("socket.io").listen(server),  
5   nicknames = {};  
6  
7 //S'escoltarà pel port 8000  
8 server.listen(8000);  
9
```

Codi *index.js*

```
10 //Al carregar l'arrel ('/'), redirecciona l'enllaç del directori + l'index.html
11 app.get('/', function(req, res) {
12     res.sendFile(__dirname + '/index.html');
13 });
14
15 //Sempre que s'utilitzi l'arrel ('/'), carrega el contingut del directori actual
16 app.use('/', express.static(__dirname));
17
18 //S'activa la funció cada vegada que un usuari s'ha connectat.
19 //La funció rep per parametre el socket de la connexió
20 io.sockets.on('connection', function(socket) {
21     //Quan rep un missatge "send message", envia a tots els sockets el missatge
22     //i el nickname del socket
23     socket.on('send message', function(data) {
24         io.sockets.emit('new message', {msg: data, nick: socket.nickname});
25     });
26 });
```


Codi *index.js*

```
26    //Quan hi ha un nou client , li assigna un nickname al socket i incrementa el
      nombre de nicknames que hi ha.
27    socket.on('new user', function(data, callback) {
28        if (data in nicknames) {
29            callback(false);
30        } else {
31            callback(true);
32            socket.nickname = data;
33            nicknames[socket.nickname] = 1;
34            updateNickNames();
35        }
36    });
37
38    //Quan un usuari es desconnecta , s'elimina el nickname dels usuaris
39    socket.on('disconnect', function(data) {
40        if(!socket.nickname) return;
41        delete nicknames[socket.nickname];
42        nicknameDisconnect(socket.nickname);
43        updateNickNames();
```

Codi *index.js*

```
46    //Envia a tots els usuaris que el nickname "nicknameDisconnect" s'ha
      disconnectat
47    function nicknameDisconnect(nicknameDisconnect) {
48        io.sockets.emit('usernameDisconnect', {nick: nicknameDisconnect});
49    }
50
51    //Actualitza els nicknames dels usuaris
52    function updateNickNames() {
53        io.sockets.emit('usernames', nicknames);
54    }
55  });
```

Codi *enviament_missatges.js*

```

1 /jQuery(function($) {
2     //Creem un socket
3     var socket = io.connect();
4     //Formulari per enviar un missatge
5     var $messageForm = $('#send-message');
6     //El missatge a enviar
7     var $messageBox = $('#message');
8     //El panel de chat
9     var $chat = $('#chat');
10    //Els missatges del chat
11    var $messages_chat = $('#messages');
12    //El botó per el qual s'envia el missatge
13    var $buttonSend = $('#send');

```

Crearem un fitxer **index.html** que conté la interfície de la pàgina web.

Crearem un fitxer **enviament_missatges.js** i el servirem. El fitxer és el següent.

Codi *enviament_missatges.js*

```
15 //Formulari del nickname
16 var $nickForm = $('#setNick');
17 //El nickname a enviar
18 var $nickBox = $('#nickname');
19 //La llista d'usuaris que hi ha
20 var $users = $('#users');
21 //Creu de l'alerta de l'error
22 var $closeAlert = $('#closeAlert');
```

Codi *enviament_missatges.js*

```
26  $nickForm.click(function(e) {
27      e.preventDefault();
28      console.log("nickform");
29      //El socket emet el missatge de "new user" on envia el nickname i espera
        un valor retornat
30      //Si el valor és true, llavors amaga el formulari del nickname i mostra
        la sala de chat
31      //Si el valor és fals, llavors mostra el missatge d'error: "Aquest
        nickname ja està utilitzat. Prova-ho amb un altre."
32      socket.emit('new user', $nickBox.val(), function(data) {
33          console.log("new user");
34          if(data) {
35              console.log("new nickname is OK");
36              $('#nickWrap').hide();
```

Codi *enviament_missatges.js*

```
37         $('#linia_top_nick').hide();
38         $('#linia_bottom_nick').hide();
39         $('#contentWrap').show();
40     } else {
41         console.log("new nickname is WRONG");
42         $('#login - error').show();
43     }
44 });
45 //Eliminem el valor del nickname
46 $nickBox.val('');
47 });
```

Codi *enviament_missatges.js*

```
49 //Quan cliquem la creu de l'alerta "Aquest nickname ja està utilitzat. Prova-  
    ho amb un altre.", s'amagarà l'error  
50 $closeAlert.click(function(e) {  
51     $("#login-error").hide();  
52 });  
53  
54 //Quan enviem un missatge, s'envia pel socket "send message" amb el valor del  
    missatge.
```

Codi *enviament_missatges.js*

```

55  $messageForm.submit(function(e) {
56      e.preventDefault();
57      console.log("messageForm submit");
58      if($messageBox.val()!='') {
59          console.log("send message"+$messageBox.val());
60          socket.emit('send message', $messageBox.val());
61      }
62      //Eliminem el valor del missatge
63      $messageBox.val('');
64  });
65
66  //Quan reps un missatge "new message", s'afageix en el chat, el nickname i el
    missatge
67  socket.on('new message', function(data) {
68      console.log('<b>'+data.nick+':</b> '+data.msg+"<br/>");
69      $messages_chat.append('<li><b>'+data.nick+':</b> '+data.msg+"</li>");
70  });

```


Codi *enviament_missatges.js*

```

72 //Quan reps un missatge "new message", s'afageix en el chat, el nickname i el
    missatge
73 socket.on('usernameDisconnect', function(data) {
74     console.log('Username Disconnect '+data.nick);
75     $messages_chat.append("<li>L'usuari <b>"+data.nick+"</b> s'ha desconnectat
        </li>");
76 });
77
78 //Quan reps el missatge "usernames", s'insereixen tots els nicknames dels
    usuaris en el div dels usuaris
79 socket.on('usernames', function(data) {
80     console.log("usernames");
81     var html = '';
82     for (var username in data) {
83         html += "<p>" + username + '</p>';
84     }
85     $users.html(html);
86 });

```

Índex

- 1 Introducció
- 2 Entorn de treball
- 3 Creació del Chat amb el Socket IO
- 4 Resultats**
- 5 Demo

A l'inici de l'aplicació, et demana un **nickname**.

Escriu el teu nickname

OK

Si no existeix el ***nickname***, mostra un **error** i et torna a demanar un ***nickname***.

Escriu el teu nickname

OK

Aquest nickname ja està utilitzat. Prova-ho amb un altre.

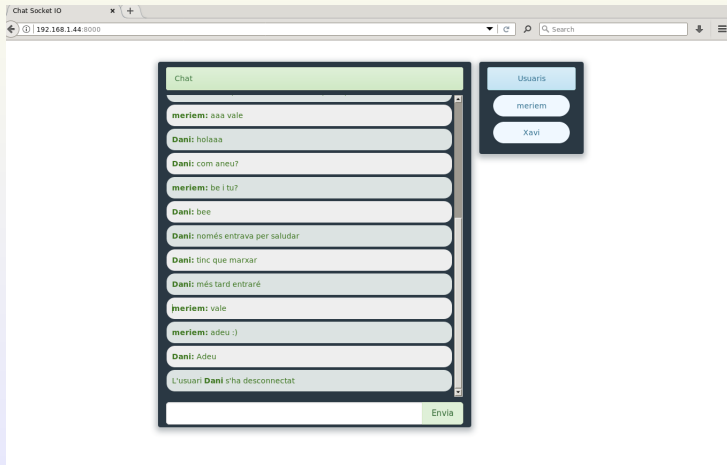
×

Si existeix el ***nickname***, entraràs al **chat**.

meriem

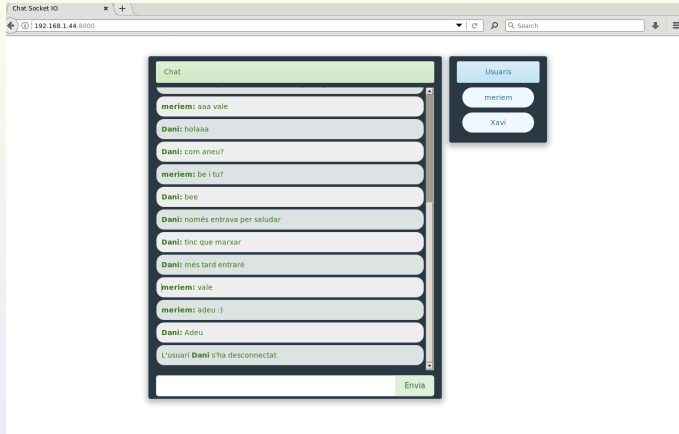
OK

Una vegada ha iniciat el **chat**, hi ha la comunicació entre els diferents usuaris. A la dreta hi ha el **chat** i a l'esquerra els diferents **usuaris**.



Índex

- 1 Introducció
- 2 Entorn de treball
- 3 Creació del Chat amb el Socket IO
- 4 Resultats
- 5 Demo**



Gràcies.
Alguna pregunta.