

## Fiche Pratique

### Bibliothèque **socketio.js**

Objectif :

Permet d'assurer une communication événementielle en temps réel entre deux composantes; client ( navigateur ) et le serveur

**Install socketio.js:**

Npm install socketio --save

Install express:

Npm install express --save

Import socketio.js:

```
const { Server } = require("socket.io");
```

Setup a web server:

```
const express = require('express');
const app = express();
const http = require('http');
const server = http.createServer(app);
```

setup a route to home page(index.html)

```
app.get('/', (req, res) => {
  res.sendFile(__dirname + '/index.html');
});
```

Initialize a socketio object :

```
const io = new Server(server);
```

listen on the `simple` event (connection or disconnect):

```
io.on('connection', (socket) => {
  console.log('a user connected');
  socket.on('disconnect', () => {
    console.log('user disconnected');
  });
});
```

Connection , disconnect event are by default fire when initializing io() on local js

```
var socket = io();
```

create an event (chat message with value “message”) on local js:

```
socket.emit('chat message', 'message');
```

take the message and print it on server terminal:

```
io.on('connection', (socket) => {  
  socket.on('chat message', (msg) => {  
    console.log('message: ' + msg);  
  });  
});
```

send multiple messages : (client side)

```
io.emit('some event', value1, value2);
```

server side:

```
io.on('connection', (socket) => {  
  socket.on('chat message', (arg1,arg2) => {  
    io.emit('chat message', arg1+'and'+arg2);  
  });  
});
```

Exemple: if value1 = “me” and value2= “my girlfriend” the final message will be “me and my girlfriend”

Same thing if we want to print the messeage on user browser.

```
socket.on('chat message', function(arg1,arg2) {  
  let text1 = document.getElementById('text1');  
  let text2 = document.getElementById('text2');  
  
  text.value=arg1;  
  text.value=arg2;  
  
});
```

Practical application but made so fast (bugs) :

Server.js

```
const express = require('express');
const app = express();
const http = require('http');
const server = http.createServer(app);
const { Server } = require("socket.io");
const io = new Server(server);

app.get('/', (req, res) => {
  res.sendFile(__dirname + '/index.html');
  var ip = req.ip
  io.on('connection', (socket) => {
    console.log('a user connected');

    socket.on('disconnect', () => {
      console.log('user disconnected');
    });
  });
  io.on('connection', (socket) => {
    socket.on('chat message', (arg1) => {
      io.emit('chat message', ip + " : "+arg1);
    });
  });
});

server.listen(3000, () => {
  console.log('listening on *:3000');
});
```

Index.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>Socket.IO chat</title>
    <style>
      body { margin: 0; padding-bottom: 3rem; font-family: -apple-
system, BlinkMacSystemFont, "Segoe UI", Roboto, Helvetica, Arial, sans-serif; }

      #form { background: rgba(0, 0, 0, 0.15); padding: 0.25rem; position: fixed;
bottom: 0; left: 0; right: 0; display: flex; height: 3rem; box-sizing: border-
box; backdrop-filter: blur(10px); }
      #input { border: none; padding: 0 1rem; flex-grow: 1; border-
radius: 2rem; margin: 0.25rem; }
      #input:focus { outline: none; }
      #form > button { background: #333; border: none; padding: 0 1rem; margin: 0
.25rem; border-radius: 3px; outline: none; color: #fff; }

      #messages { list-style-type: none; margin: 0; padding: 0; }
      #messages > li { padding: 0.5rem 1rem; }
      #messages > li:nth-child(odd) { background: #efefef; }
    </style>
  </head>
  <body>
    <ul id="messages"></ul>
    <form id="form" action="">
      <input id="input" autocomplete="off" /><button>Send</button>
    </form>
    <script src="/socket.io/socket.io.js"></script>
    <script>
      var socket = io();
```

```

var messages = document.getElementById('messages');
var form = document.getElementById('form');
var input = document.getElementById('input');
var input2 = document.getElementById('input2');

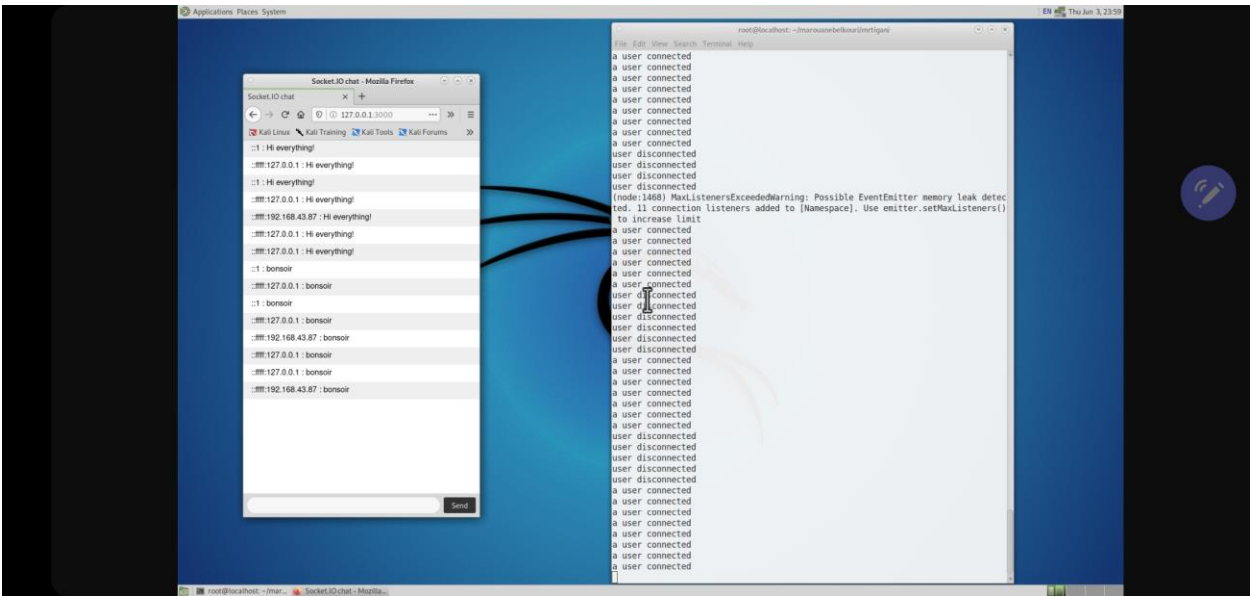
form.addEventListener('submit', function(e) {
    e.preventDefault();
    if (input.value) {
        socket.emit('chat message', input.value);

        input.value = '';
    }
});

socket.on('chat message', function(msg) {
    // var text = document.getElementById('text1');
    var item = document.createElement('li');
    // text.value=msg;
    item.textContent = msg;
    messages.appendChild(item);
    window.scrollTo(0, document.body.scrollHeight);
});
</script>
</body>
</html>

```

Results :



Prepared By Marouane Belkouri , Imane El Omary, Mohammed Abdellaoui