

資工二乙 Theo

Node.js & Socket.IO

Node.js 簡介

- ▶ Node.js是一個開放原始碼、跨平台的、可用於伺服器端和網路應用的執行環境
- ▶ 採用V8引擎來執行程式碼
- ▶ 內建許多module，大部分module是由JavaScript寫成

Node.js & JavaScript?

- ▶ 之前寫Pokemon GO地圖的JavaScript是「前端（瀏覽器端）程式碼」，搭配HTML和CSS。
- ▶ Node.js 可以說是「後端（伺服器端）的JavaScript」，就像是PHP這種後端程式語言。

今天的重點
"Socket.IO"

What is Socket.IO?

Socket.IO是一個用於建立即時性通訊網頁應用程式（realtime web applications）的跨平台 JavaScript 函式庫。

可以廣義的把它視為是WebSocket的包裝工具，簡化了原生WebSocket API的操作。



socket.io

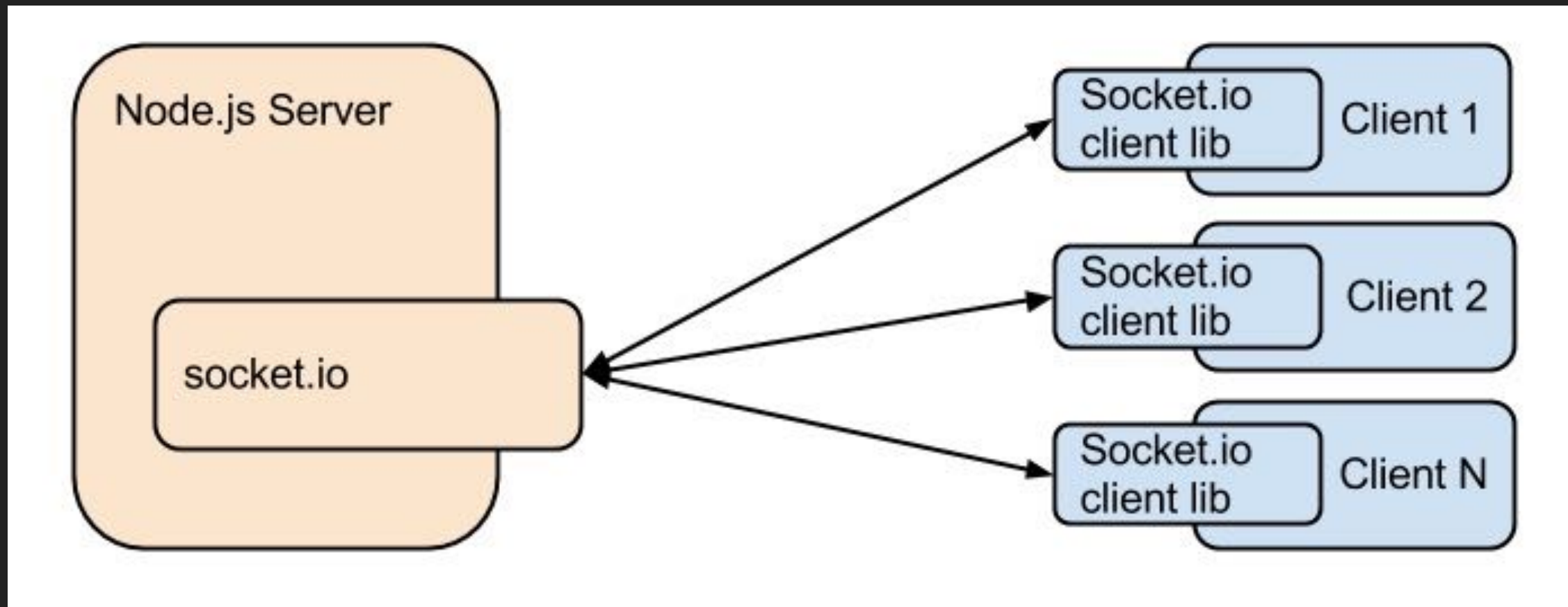
What is Socket.IO?

- ▶ WebSocket
- ▶ FlashSockets
- ▶ xhr-polling
- ▶ xhr-multipart
- ▶ JSONP-polling
- ▶ htmlfile

What is Socket.IO?

Socket.IO 包含瀏覽器端函式庫（client-side library，運行於瀏覽器中）與伺服器端函式庫（server-side library，運行於 Node.js 環境），兩者提供的 API 幾乎一樣。

連線方式



圖片來源：<http://blog.lightstreamer.com/2013/05/benchmarking-socketio-vs-lightstreamer.html>



講了那麼多文言文
還是來動手操作好了

goo.gl/FvSMWd

package.json

- ▶ 記錄有關此專案的所有資訊
- ▶ 把所有需要的module統一管理
- ▶ 根據package.json安裝module

> cd /path/to/your/project/folder

(例如 C:\Users\CSPC645\Desktop\DIY)

> npm install

最基本的Socket.IO

兩個event：

1. on - 接收server的資料
2. emit - 發送資料給server

使用方法

- ▶ `.on(事件名稱, 資料)`
- ▶ `.emit(事件名稱, 資料)`
- ▶ 資料可以是單一變數、陣列 `[]`、甚至是物件 `{}`。

Client端要做的事

▶ `<script src="/socket.io/socket.io.js"></script>`

```
<script>
  var socket = io();
  socket.on('server message', function(msg) {
    alert(msg);
  });
</script>
```

Server端要做的事

▶ `var io = require('socket.io');`

```
io.on('connection', function (socket) {  
  io.emit('server message', "A user connect!!");  
  socket.on('disconnect', function () {  
    io.emit('server message', "A user disconnect!!");  
  });  
});
```

執行server.js

- ▶ `cd /path/to/your/project/folder`
- ▶ `node server.js`
- ▶ 如果看到「Server is now listening on port 3000」，打開瀏覽器到「localhost:3000」看看成果。

Socket.IO

即時匿名聊天室

128.199.136.31:8080

goo.gl/Vksey9

Missing Code

- ▶ 閱讀一下程式碼和註解
- ▶ 嘗試完成server.js第69行 & public/index.html第105行。
- ▶ Hint: 都是剛剛教過的東西~





XSS



回想之前XSS教過的東西
玩壞旁邊同學的聊天室吧 ~

問題出在哪？

server.js

```
79 ▼ function check(type, msg) {  
80     var s = msg;  
81 ▼     if (type == "liver" || type == "yee") {  
82         s = "";  
83 ▼         for (var i = 0; i < msg.length; i++) {  
84             if (type == "liver")  
85                 s += "...";  
86             else if (type == "yee")  
87                 s += "YEE~";  
88         }  
89     }  
90     return s;  
91 }
```


永遠不要相信
使用者輸入的內容

Example

```
72 ▼ function check(type, msg) {  
73     var s = msg;  
74 ▼     if (type == "liver" || type == "yee") {  
75         s = "";  
76 ▼         for (var i = 0; i < msg.length; i++) {  
77             if (type == "liver")  
78                 s += "...";  
79             else if (type == "yee")  
80                 s += "YEE~";  
81         }  
82 ▼     } else {  
83         // escape htmlspecialchars  
84 ▼         var map = {  
85             '&': '&amp;',  
86             '<': '&lt;',  
87             '>': '&gt;',  
88             '"': '&quot;',  
89             "'": '&#039;';  
90         };  
91 ▼         return s.replace(/([<>"']/g, function (m) {  
92             return map[m];  
93         });  
94     }  
95     return s;  
96 }
```



今日課程結束～
