

Socket.IO Programming using Swift for iOS/OSX Environment (3)

Project - Chat Room App

一. 實驗目的

利用前兩次實驗所學到的技巧，在 iOS 平台上實作出基本的聊天室 App。

二. Swift / JavaScript 技巧補充

1. Socket

- Server side
 - 強制斷線：`socket.disconnect();`
- Client side
 - 連線：`self.socket!.connect()`
 - 斷線：`self.socket!.disconnect()`

2. Int, String 轉換

- Int 轉 String
 - Swift:
`var strVar:String = String(intVar)`
 - JavaScript:
`var strVar = intVar.toString();`
- String 轉 Int
 - Swift
`var intVar = Int(strVar)`
 - JavaScript
`var intVar = parseInt(strVar ,10);`
`//strVar為10進制格式`

3. Array 使用

- 空 Array 宣告
 - Swift:
`var myArray = [String]() //需有Type, 第一個index為0`
 - JavaScript:
`var myArray = []; //第一個index為0`
- Array 宣告
 - Swift
`var myArray = ["A", "B", "C"]`

- JavaScript

```
var myArray = ["A", "B", "C"];
```
- 在 Array 尾端加入值
 - Swift

```
myArray.append("D")  
myArray = myArray + ["E"]
```
 - JavaScript

```
myArray.push("D");  
myArray[myArray.length] = "E"
```
- 在 Array Index i 的位置插入一個item，其他 item 往後推移
 - Swift

```
myArray.insert("F", at: i)
```
 - JavaScript

```
myArray.splice(i, 0, "F");  
// 在 i 的位置往後刪掉 0 個 item ,並且插入"F"
```
- 移除 Array Index 中第 i 個item，其他 item 往前遞補
 - Swift

```
myArray.remove(at: i)
```
 - JavaScript

```
myArray.splice(i, 1); //在 i 的位置往後刪掉1個 item
```
- 複製整個 Array
 - Swift

```
var duplicateArray = myArray
```
 - JavaScript

```
var duplicateArray = myArray.slice();
```

4. String 的分割與組合

- 把 Array of String 用某個 Separator 合併
 - Swift:

```
var joinString = myArray.joined(separator: ",")  
//separator: ", "  
//result:      "A,B,C"
```
 - JavaScript:

```
var joinString = nameArray.join(',');  
//separator: ", "  
//result:      "A,B,C"
```
- 把 String 用某個 Separator 分割成Array
 - Swift

```
var splitArray =  
joinString.components(separatedBy: "-")
```
 - JavaScript

```
var splitArray = joinString.split("-");
```

三. 實驗步驟

1. 讓 Server 記錄連入的 Socket

- (1) 打開 ChatRoomServer 中的 index.js 檔案
- (2) 加入一個 array 用來儲存連線中的 Socket 物件

```
var socketArray = [];
```

- (3) 再有新連線建立時，要把產生的 socket 存到 socketArray 裡。在 io.on('connection', ...) 的 callback function 中加入以下程式碼

```
//將新創造的 Socket 加到 socketArray 裡
socketArray[socketArray.length] = socket;
//在 server 的 console 中輸出現有的 socket 數量
console.log('socket count: ' + socketArray.length);
```

- (4) 註冊 socket 的 “disconnect” event，在 disconnect 時從 socketArray 裡移除斷線的 socket。在 io.on('connection', ...) 的 callback function 中加入以下程式碼

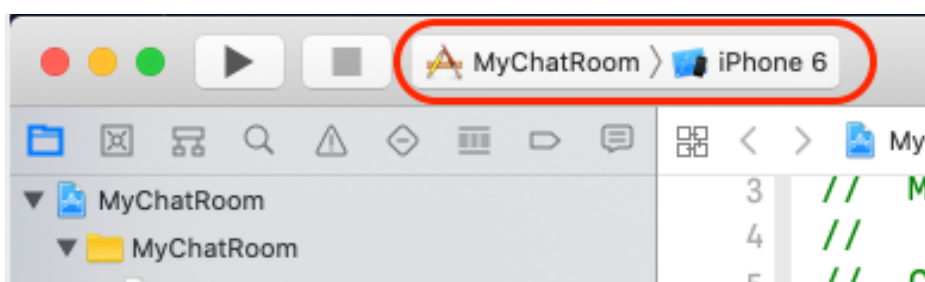
```
socket.on('disconnect',
function()
{
    //尋找要斷線的 socket 在 socketArray 中的 index
    var socketIndex;
    for(var i=0 ; i<socketArray.length ; i++)
    {
        if(socketArray[i] == socket)
            socketIndex = i;
    }

    //從 socketArray 中把斷線的 socket 移除
    socketArray.splice(socketIndex, 1);

    //在 server 的 console 中輸出現有的 socket 數量
    console.log('socket count: ' + socketArray.length);
});
```

2. 利用 Xcode 開啟多個 Simulator，模擬多個 iOS Client 之情況

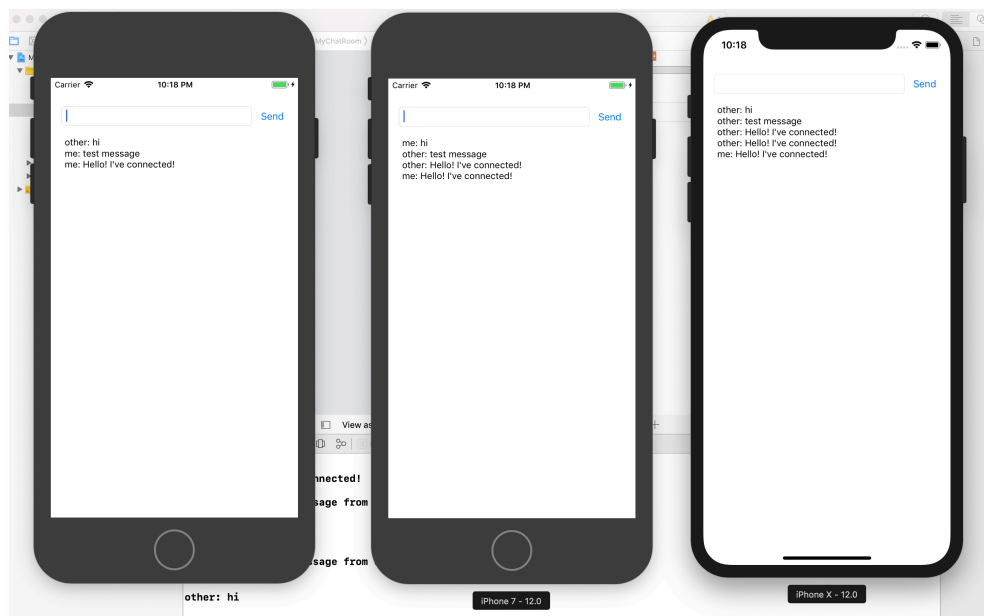
- (1) 執行一次 iOS MyChatRoom 專案，開啟 Simulator，執行成功後，按停止
- (2) 從左上方選擇不同的 iOS 裝置，執行



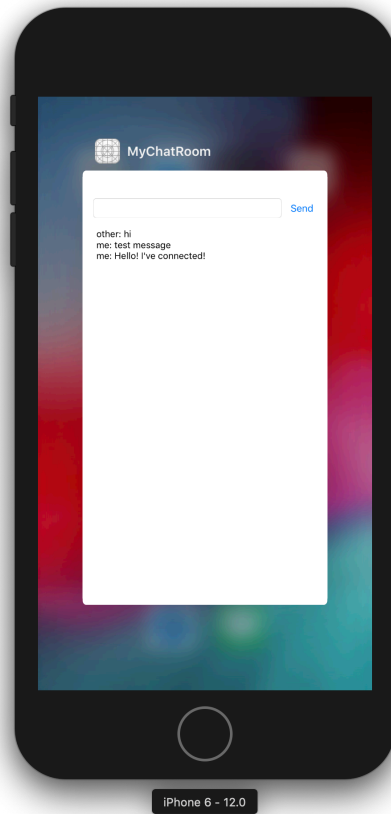
- (3) 把其他已停止 App 的 Simulator 手動點選 App 打開



(4) 重複 (2) 的操作，可開啟多個 Simulator測試



(5) 在 Simulator 介面中，利用 `command + shift + 兩下H` 切換置多工頁面，上滑可關閉 App

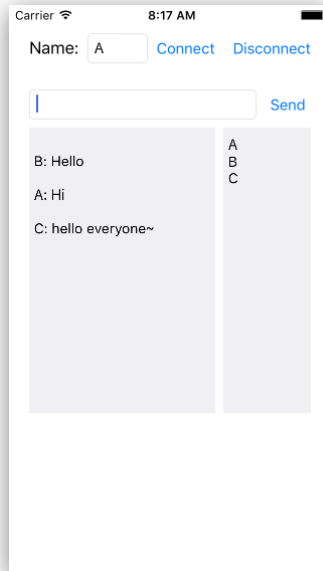


- (6) 若修改 Code 過後要使用多個 App 測試，請每個裝置都重新用 Xcode 執行過再測試
- (7) 開啟 Terminal.app，利用 “node index.js” 執行 Server 程式
- (8) 利用三個 App 之連線、終止，觀察 Terminal 中 server console 顯示的 Socket 數量正
不正確

四. Project

1. 基本題：

- 將 iOS client 改成利用按鈕控制連線與斷線
- 在 iOS App 上新增 name 輸入欄位以及 member list 區塊。當client連線成功後會自動告知 server 此 client 名稱，member list 則會即時顯示最新的聊天室成員名單
- 聊天畫面中，每則訊息會顯示發訊息者的名稱



※ 請在第四次 Lab 結束前 Demo 完基本題

2. 加分題（第四次Lab下課鐘響前截止）：

- Client 在顯示 member list 時會標出自己在 list 的哪個位置

