

嵌入式系統 Lab 1：聊天室實作

組員：B04901157 鄭宇崴、B04901166 陳培鳴

一、功能介紹



這個聊天室可以同時讓多人聊天。中間的畫面是聊天記錄，右上角顯示的是目前連線的人數。當使用者加入聊天室時，須先在左下角的方框內輸入暱稱，接下來就可以在正下方的方框內輸入聊天內容，並按下「送出」發表。在中途可以不離線即改變暱稱。另外，由於我們沒有將聊天內容存到資料庫，使用者無法看到加入前的聊天紀錄。

二、系統架構

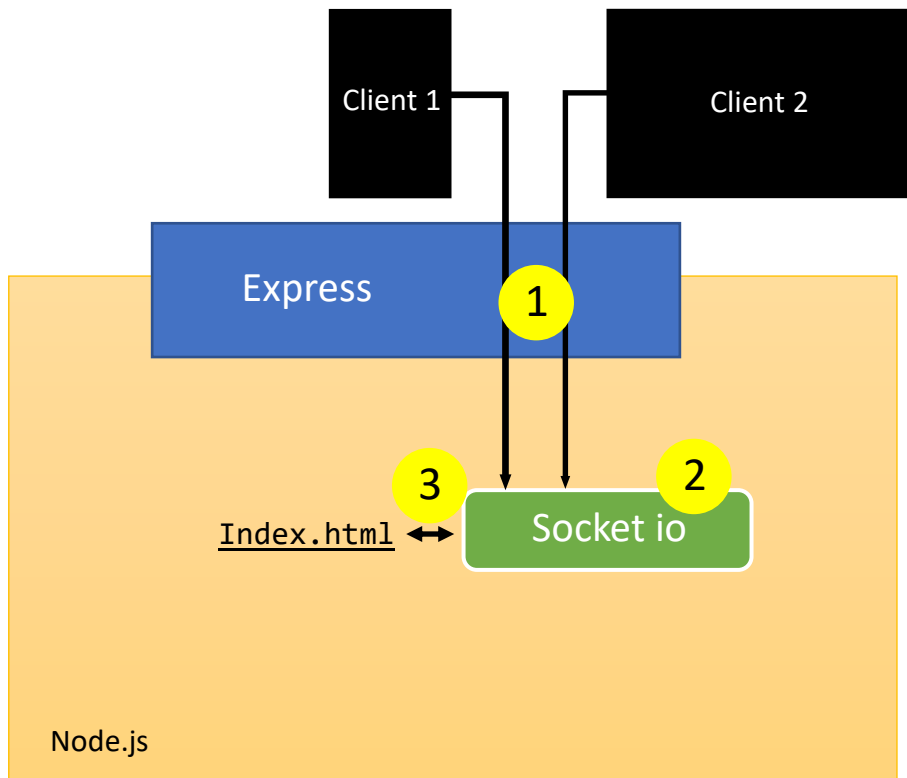
在這個系統中我們使用了兩個 **node.js** 的套件：

1. **express**：node.js 之下的網頁應用程式架構，可以接收並回應 **http** 請求
2. **socket.io**：用於即時網頁通訊的 **javascript** 函式庫

以及兩個程式檔案

1. **index.js**：聊天室的 **node.js** 程式碼
2. **index.html**：聊天室的 **html** 樣板

系統架構圖詳見下頁。



- (1) 每當一個新的 client 要連線時，client 會發送 http 請求給 express，express 會將 Index.html 回傳給 client，並開啟 socket io 與該 client 的雙向通訊
- (2) 當雙向通訊建立後，socket io 會監控是否有發生使用者定義的事件，並據此生成並回傳最新的 html 給 client。以本次聊天室為例，有以下幾個事件：
 - (a) connect：若有新 client 連線，將計數器加 1
 - (b) disconnect：若有 client 離線，將計數器減 1
 - (c) greet：新的 client 輸入暱稱
 - (d) send：有 client 按下「送出」
- (3) 一旦有上述的事件發生，socket io 會透過 html 檔裡面的 EventListener 修改 html 檔中的對應訊息，例如連線人數、暱稱、訊息等等，並且回傳給 client

三、實作心得

我們兩個雖然之前都有接觸過 web 相關的程式開發，但一個使用的語言是 python，一個用的是 Ruby，用 Javascript 對兩人來說都是第一次，也因此我們決定從零開始學習 JS 這個語言。一開始我們花了約兩個禮拜自學 JS 相關語法，之後選了網路上類似聊天室 app 的版型作為模板進行開發(對於 css 的設定實在是不太有概念...)。在這裡我們碰到一個很大的問題：如何即時的更新資料庫並在前端畫面 re-render 出來，這時候我們上網找到了 socket.io 這個函式庫，並嘗試從官方文件與其他例子裡學習其架構與語法；在此同時，我們又查到可以用

`express.js` 來簡化之前寫的 `node.js code`，就這樣我們終於拼拼湊湊地寫出了一個堪用的聊天室 app，原本有意再行擴增功能，但在嘗試 `deploy` 到 `heroku` 上的時候又發生了 `port` 設定問題(找了很久才發現 Orz)，解決完後才發現沒有什麼時間再寫其他東西了。總結這次寫 LAB1 的心得，雖然過程跌跌撞撞，但也知道了很多新的東西，碰到問題時也了解有哪些線上資源可以利用，整體來說儘管沒有達到最初的標準，但還是學到了蠻多東西。