

建置 nodejs 環境

1. ~~下載並安裝對應的安裝檔(<https://nodejs.org/en/download/>)~~
* 於 www.quickcode.com.tw/posdemo 下載 NodeJs 安裝檔
2. 將安裝路徑設定在網頁底層(./htdocs)，以方便網頁調用
3. 進入 dos 輸入 node -v 與 npm -v 出現版號確認安裝完成
4. 配置 npm 的 global 與 cache 路徑
 - ~~npm config set prefix "安裝路徑/nodejs/node_global"~~
* 於 www.quickcode.com.tw/posdemo 選擇瀏覽檔案明細 64 位元，下載 nodejsprefix.bat，並將檔案放置安裝路徑底下後執行
 - ~~npm config set cache "安裝路徑/nodejs/node_cache"~~
* 於 www.quickcode.com.tw/posdemo 選擇瀏覽檔案明細 64 位元，下載 nodejscache.bat，並將檔案放置安裝路徑底下後執行
5. 進入我的電腦>內容>進階>環境變數
 - 使用者變數的 path 新增值 "安裝路徑/nodejs/node_global"
 - 系統變數新增"NODE_PATH"值填入 "安裝路徑/nodejs/node_global/node_modules"
6. ~~於安裝路徑底下新建資料夾"node_chat"，並於內部建立 node server 主程式 server.js(程式可於遠端取得)~~
* 於 www.quickcode.com.tw/posdemo 下載 node_chat，並於 nodejs 底下執行
7. 安裝三個模組：express、socket.io、cors
* 門市網路速度限制可能會 安裝失敗，換成使用自解檔安裝—於 www.quickcode.com.tw/posdemo 中下載 NodeJs 外掛自解檔 (node_modules.exe)
 - ~~npm install --save express~~
 - ~~npm install --save socket.io~~
 - ~~npm install --save cors~~
 - ~~npm install --save mysql(非必須)~~
8. 防火牆設定：允許的應用程式>變更設定>允許其他應用程式>"安裝路徑/node.exe"
9. ~~準備好 server.js 後，進入 dos 將路徑移動到程式所在路徑，執行 node server.js 開始運行~~
* 於 www.quickcode.com.tw/posdemo 選擇瀏覽檔案明細 64 位元，下載 nodejs.bat，並將檔案放置在 pos 底下，並設定排程。

server.js

```
var app = require('express')(); //引入 express 库
var cors = require('cors');
var http = require('http').Server(app); //将 express 注册到 http 中
var usocket = []; //全局变量
//new addition
var io = require('socket.io')(http);
//end

//允许 cross domain
const corsOptions = {
  origin: [
    'http://localhost',
    'http://127.0.0.1',
  ],
  methods: 'GET,HEAD,PUT,PATCH,POST,DELETE,OPTIONS',
  allowedHeaders: ['Content-Type', 'Authorization'],
};

app.use(cors(corsOptions));

//当访问根目录时，返回 Hello World
app.get('/', function(req, res){
  res.sendFile(__dirname + '/index.html');
});

app.get('/message', function(req, res){
  //res.sendFile(__dirname + '/index.html');
  res.send('get message.');
```

```
});

//new addition
io.on('connection', function(socket){
  console.log('a user connected');

  //监听 join 事件
  socket.on("join", function (name) {
    usocket[name] = socket
    io.emit("join", name) //服务器通过广播将新用户发送给全体群聊成员
    usocket[name].io.emit("join", name) //服务器通过广播将新用户发送给指定成员
  });

  //new addition
  socket.on("message", function (msg) {
    if(msg=='i am secview.'){
      io.emit("message", 'hello secview. i am server') //将新消息广播出去
    }
    else{
      io.emit("message", msg) //将新消息广播出去
    }
    console.log(msg);
  });
});

//end

//启动监听，监听 3000 端口
http.listen(3000, function(){
  console.log('listening on *:3000');
});
```

client.html(.php) *for jquery*

```
<script>
  $(document).ready(function(){
    var socket = io.connect('http://localhost:3000');
    socket.emit('message', 'i am secview.');
```

socket.on('message',function(msg){
 console.log(msg);
});
});
</script>