



# HW5 Tutorial

NodeJS, NPM, socket.io

# NodeJS安裝

- 官網下載Node.js LTS版本 <https://nodejs.org/en/>
- node -v
- npm -v

```
C:\Users\Gama>node -v  
v20.13.0  
  
C:\Users\Gama>npm -v  
10.2.5
```



2

Web Server



Browser

輸入網址，發出request請求

request head 資訊

response 回傳資料

response head 資訊



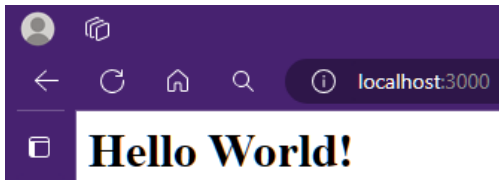
Web Server  
(Node.js)

- ❑ `createServer()` 建立server
- ❑ `listen()` 監聽port
- ❑ “node filename.js” 啟動server
- ❑ 127.0.0.1:3000或localhost:3000瀏覽

回應請求

```
1 var http = require('node:http');
2 /* 開啟Web Server */
3 http.createServer(function(request,response){
4     response.writeHead(200,{"content-type":"text/html"});
5     response.write('<h1>Hello World!</h1>');
6     response.end();
7 }).listen(3000);
```

```
D:\test>node server.js
```



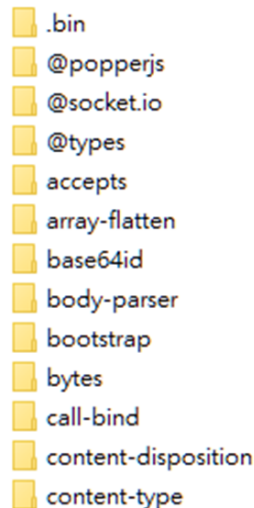
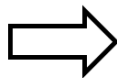
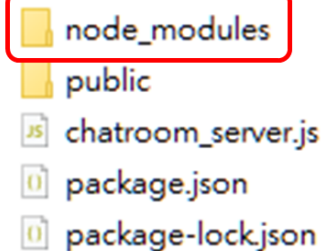
3

NPM



- Node Package Manager
- Node.js 套件管理工具
- `npm install <模組名稱>`
- `npm uninstall <模組名稱>`

名稱



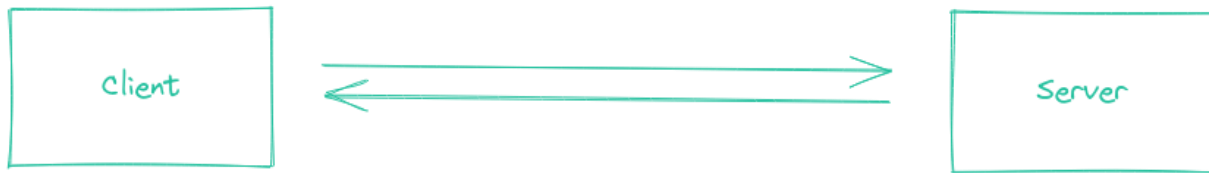


4

Socket.io



- JavaScript library
- 即時雙向通訊
- <https://socket.io/>
- socket.emit - 對一個特定的 socket 傳訊息
- socket.on - 對特定 event 的運行結果進行接收
- io.emit - 對所有線上 socket 傳訊息



5

## HW5 tutorial

- 打開終端機cmd進入專案資料夾 Ex: cd D:\test
- 輸入 “npm init” 初始化專案
- package name: 專案名稱
- version: 版本
- description: 專案描述
- entry point: 選擇主要執行的檔案
- test command: 是否有測試程式碼
- git repository: 是否有其他git 數據庫
- keywords: 設定關鍵字讓其他人找到你的專案
- author: 專案的作者名稱
- license: 是否開源
- **()內為預設值，如果不需設定直接按enter就能套用預設值**

```
D:\test>npm init
This utility will walk you through creating a package.json file.
It only covers the most common items, and tries to guess sensible defaults.

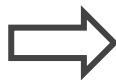
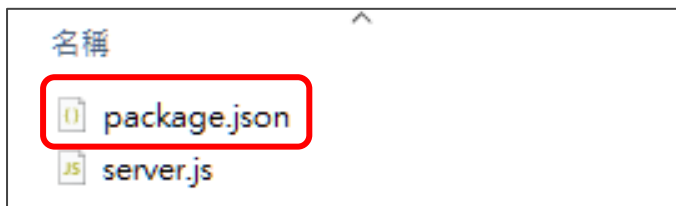
See `npm help init` for definitive documentation on these fields
and exactly what they do.

Use `npm install <pkg>` afterwards to install a package and
save it as a dependency in the package.json file.

Press ^C at any time to quit.
package name: (test) hw5test
version: (1.0.0)
description: hw5test
entry point: (server.js)
test command:
git repository:
keywords:
author: Daniel
license: (ISC)
About to write to D:\test\package.json:

{
  "name": "hw5test",
  "version": "1.0.0",
  "description": "hw5test",
  "main": "server.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1",
    "start": "node server.js"
  },
  "author": "Daniel",
  "license": "ISC"
}

Is this OK? (yes) yes
D:\test>
```



```
JS server.js  {} package.json ×
{} package.json > ...
1  {
2    "name": "hw5test",
3    "version": "1.0.0",
4    "description": "hw5test",
5    "main": "server.js",
6    > 偵錯
7    "scripts": {
8      "test": "echo \"Error: no test specified\" && exit 1",
9      "start": "node server.js"
10   },
11   "author": "Daniel",
12   "license": "ISC"
13 }
```

- npm install express --save 安裝express
- npm install socket.io --save 安裝socket.io
- --save:安裝NPM模組並寫入package.json的"dependencies"
- 其他套件視情況自行安裝

```
D:\test>npm install express --save
added 64 packages, and audited 65 packages in 3s

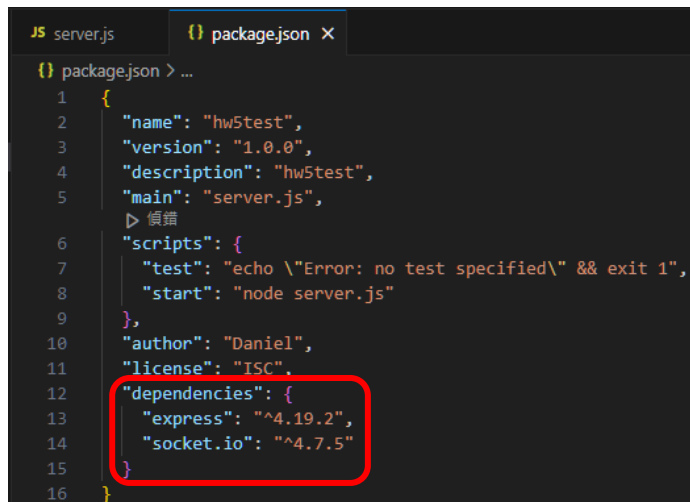
12 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities

D:\test>npm install socket.io --save
added 23 packages, and audited 88 packages in 3s

12 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```



```
JS server.js  {} package.json x
() package.json > ...
1  {
2    "name": "hw5Test",
3    "version": "1.0.0",
4    "description": "hw5Test",
5    "main": "server.js",
6    "scripts": {
7      "test": "echo \"Error: no test specified\" && exit 1",
8      "start": "node server.js"
9    },
10   "author": "Daniel",
11   "license": "ISC",
12   "dependencies": {
13     "express": "^4.19.2",
14     "socket.io": "^4.7.5"
15   }
16 }
```

名稱	修改日期	類型	大小
node_modules	2024/5/10 下午 03:34	檔案資料夾	
public	2024/5/10 下午 03:45	檔案資料夾	
package.json	2024/5/10 下午 05:14	JSON 來源檔案	1 KB
package-lock.json	2024/5/10 下午 03:34	JSON 來源檔案	34 KB
server.js	2024/5/10 下午 05:13	JavaScript 來源檔...	2 KB

你輸入的訊息:

對方傳送的訊息:

send

## server.js

```
/*----- 當server收到connection代表有人透過瀏覽器訪問 -----*/
io.on('connection', (socket) => {
  console.log('a user connected');
  socket.on('sendmsg', (data) => {
    //socket.on接收client端傳送過來的資料
    //第一個參數代表這筆資料是什麼名稱
    //第二個參數是call back function, 用來處理接收到的資料
    io.emit('receivmsg', {msg: data.msg, id: data.id}); //向所有client端發送"receivmsg"這筆資料
  });
});
```

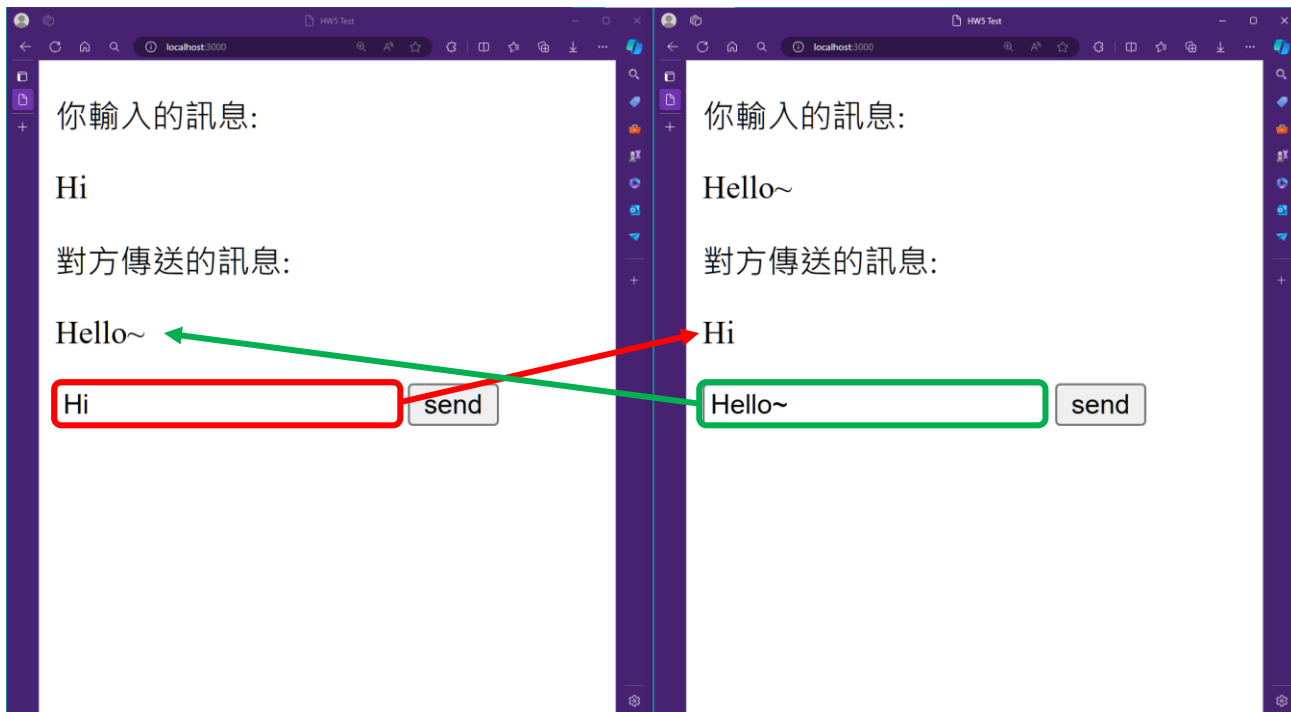
data:  
{key: value}

## client.js

```
/*-----socket事件-----*/
socket.on('receivmsg', (data) => {
  //socket.on接收server端傳送過來的資料
  //第一個參數代表這筆資料是什麼名稱
  //第二個參數是call back function, 用來處理接收到的資料
  if(data.id !== id){
    document.getElementById('receivmsg').innerText = data.msg; //顯示收到的訊息
  }
});

function sendMsg(){
  let msg = document.getElementById('msg').value;
  document.getElementById('sendmsg').innerText = msg;
  socket.emit('sendmsg', {msg: msg, id: id}); //socket.emit(資料名稱, 資料)傳送訊息給server端
}
```





**THANKS!**

**Any questions?**