技術筆記

Node.js

筆記更新紀錄

|  |  |  |
| --- | --- | --- |
| 日期 yyyy/MM/dd | 更新人員 | 更新備註 |
| 2019/07/23 | Miles | 建立文件 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

目錄

[1 關於 Node.js 基本講解與使用 3](#_Toc14781049)

[2 關於 Web框架 5](#_Toc14781050)

## 1 關於 Node.js 基本講解與使用

|  |
| --- |
| Node |
| 簡介: |
| 1. Node.js 是一個server-side javascript 執行環境。 2. Node.js強調自己提供一種non-blocking(非阻塞)，event-drivent(事件驅動)的程式開發模式，適合用來處理data-intensive real-time applications(數據密集型實時應用程序)。 3. Node.js是作為網站程式的後端框架。 |
| 第一個範例: |
| 1. 首先要先安裝Node.js(請自行google查詢) 2. 進入vscode(請自行安裝vscode) 3. 建立一個目錄(該目錄為這個練習的專案。      1. 回到vscode建立一個index.js，並且輸入以下內容      1. 到控制台輸入以下語法     node index.js 可以啟動一個port 3000的web server   1. 進入瀏覽器 輸入localhost:3000即可以看到結果 |
| 首先了解npm |
| 1. npm全名: Node Package Manager。 2. 是Node.js的套件(package)管理工具。 3. 加快軟體開發的效率與使用第三方線上的套件。 4. 例如，我們要使用Google 的Firbase第三方函示庫，我們可以直接使用語法npm install firbase即可安裝。 |
| 初始化專案(npm init) |
| 1. 初始化專案會建立一個package.json 的json檔案。 2. 該檔案是一種CommonJS規定用來描述包的文件。 3. 裡面可以定義相依的相關套件以及應用程式的資訊。 |
| 建立package.json |
| 1. 進入專案目錄後到終端機建立npm init     會詢問您 package name ，注意package name並非檔案名稱，而是存放在這個pacakge.json當中的name 屬姓名。   1. 然後詢問版本、desc等等。 2. 建立完成後專案會多產生一個package.json檔案。 3. 未來下載第三方資訊可以直接在終端機輸入npm install語法。 |
| 關於package.json |
| 1. package.json範例如下: |
| 修改第一個練習 |
| 1. 根據不同URL回傳不同資訊 |
| 使用reuqest 模組 |
| 1. 需要使用request 模組 需要先install |

## 2 關於 Web框架

|  |
| --- |
| 建構一個專案框架 |
| 簡介: |
| 1. web框架百百款，如果我們遵循上一個章節的方式一個個建立js需要花很多時間，所以我們必須使用前端框架。 2. 首要推薦也是網路範例最多的為Express.js框架。 3. Express.js具備了MVC架構。 4. Express 可以說是 Node.js 底下的一個前端 + 後端的框架。 5. 具備MVC Framework。 View 就是前端畫面呈現 HTML + CSS，Model、Controller 為後端 API 搭建資料庫與控制流程。 |
| 第一個範例(安裝Express.js): |
| 1. 首先至終端機輸入npm install -g express-generator     建立完成後，可以輸入express -v檢查版本，也可以代表安裝成功 |
| 建立一個express 專案 |
| 1. 終端機輸入如下: express -f 自定義專案名稱      1. 完成後會產生一個專案      1. 然後可以對這個專案進行初始化 使用npm install進行初始化。 |

## 3 結構

|  |
| --- |
| Node |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

## 模板(複製用)

|  |
| --- |
| Node |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |