



Node.js 超級入門

OneJar

onejar99@gmail.com

Last Updated: 2018/05/13



Outline

- 開發工具：VSCode
- Hello Node.js：Node.js 是什麼？為何學？
- 環境安裝：Node.js
- 套件引用
- 套件管理工具：npm
- 開發輔助工具套件：nodemon
- CLI Tool 類型專題實作：景點天氣查詢 App
- Web API 類型專題實作：賣場商品 RESTful API (Express & MongoDB)
- Appendix：VSCode Extensions



開發工具：VSCode

Visual Studio Code (VSCode)



- 官網下載安裝: <https://code.visualstudio.com/>
- On Mac: '\$ brew cask visual-studio-code'

➤ 特色：



IntelliSense



Debugging



Built-in Git



Extensions

- Open source，社群活躍
- Debugging
- 預設 Terminal 介面整合

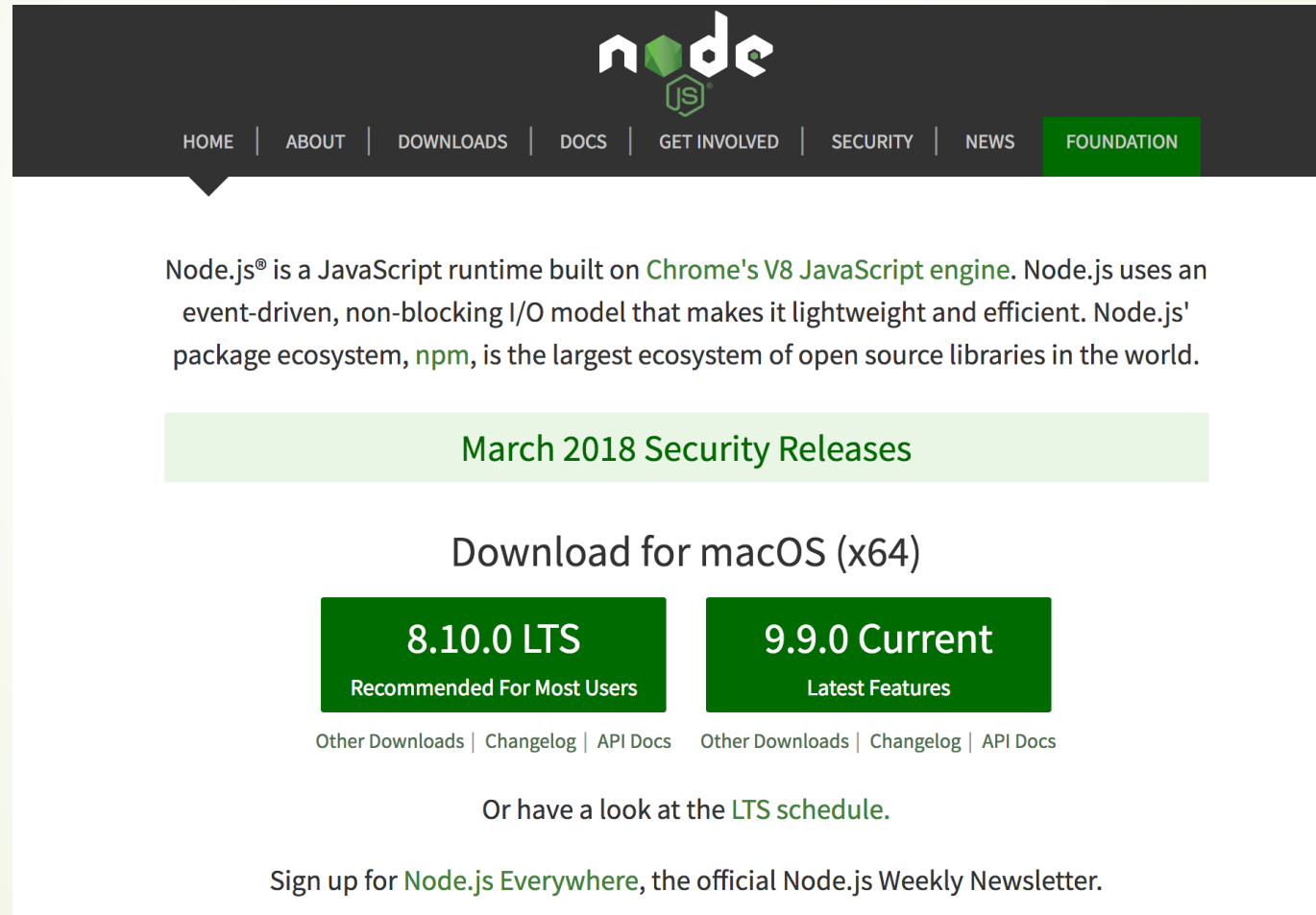


Hello Node.js

Node.js 是什麼？為何學？

What's Node.js

➡ <https://nodejs.org/>



The screenshot shows the Node.js website homepage. At the top is a dark navigation bar with the Node.js logo and links for HOME, ABOUT, DOWNLOADS, DOCS, GET INVOLVED, SECURITY, NEWS, and FOUNDATION. The FOUNDATION link is highlighted in green. Below the navigation bar, a paragraph describes Node.js as a JavaScript runtime built on Chrome's V8 JavaScript engine, using an event-driven, non-blocking I/O model, and mentions its package ecosystem, npm. A green banner highlights 'March 2018 Security Releases'. Below this, the section 'Download for macOS (x64)' features two green buttons: '8.10.0 LTS Recommended For Most Users' and '9.9.0 Current Latest Features'. Links for 'Other Downloads', 'Changelog', and 'API Docs' are provided for both versions. A link to the 'LTS schedule' is also present. At the bottom, there is a sign-up link for 'Node.js Everywhere', the official Node.js Weekly Newsletter.

node

HOME | ABOUT | DOWNLOADS | DOCS | GET INVOLVED | SECURITY | NEWS | FOUNDATION

Node.js® is a JavaScript runtime built on [Chrome's V8 JavaScript engine](#). Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient. Node.js' package ecosystem, [npm](#), is the largest ecosystem of open source libraries in the world.

March 2018 Security Releases

Download for macOS (x64)

8.10.0 LTS
Recommended For Most Users

9.9.0 Current
Latest Features

[Other Downloads](#) | [Changelog](#) | [API Docs](#) [Other Downloads](#) | [Changelog](#) | [API Docs](#)

Or have a look at the [LTS schedule](#).

Sign up for [Node.js Everywhere](#), the official Node.js Weekly Newsletter.



What's & Why Node.js

- Node.js 官網首頁的三句簡介涵蓋了 Node.js 主要特性和優點：
- A JavaScript runtime
 - built on Chrome's V8 JavaScript engine
- Lightweight and efficient
 - event-driven
 - non-blocking I/O model
- The largest package ecosystem of open source libraries
 - npm (Node.js' package ecosystem)

Node.js 自我介紹之懶人包

- 我是一個可以讓 JavaScript 在 Browser 以外運行的環境
- 我的執行效能很好
- 我有世界上數一數二豐富的開源套件朋友圈
- 學一種語言，前後端通吃



選我選我～



環境安裝：Node.js

Node Version Manager (nvm)

- <https://github.com/creationix/nvm>
 - Simple bash script to manage multiple active node.js versions
- On Mac:
 - 安裝指令:
 - ``# curl -o- https://raw.githubusercontent.com/creationix/nvm/v0.33.6/install.sh | bash``
 - PS: 不要使用 homebrew 安裝
 - 官方文檔 : Homebrew installation is not supported.
- On Windows
 - 參考:
 - <https://oranwind.org/nvm-windows/>
 - <http://trunk-studio.com/blog/nvm-for-windows/>

NVM 常用指令

Command	Description
\$ nvm ls	列出 node.js 版本安裝狀態與當前使用版本。
\$ nvm ls-remote	列出有哪些 node.js 版本可下載。
\$ nvm install <version>	安裝指定版本的 node.js。(若沒特殊需求，原則上安裝 LTS 版本) e.g., ` \$ nvm install v6.11.2`
\$ nvm use <version>	指定當前使用版本。 e.g., ` \$ nvm use v8.2.1`
\$ nvm alias default <version>	設定default版本，開新的shell session就不用每次都要設定`nvm use <version>`。 e.g., nvm alias default v6.11.2
\$ nvm --version	顯示 nvm 的版本

實際跑跑看 Node.js 程式 - 1. Hello World

1. \$ vim app.js

```
console.log( 'Hello Node.js' );
```

2. \$ node app.js

實際跑跑看 Node.js 程式 - 2. Hello Web

1. \$ vim app.js


```
var http = require('http');  
http.createServer(function (req, res) {  
  res.writeHead(200, {'Content-Type': 'text/plain'});  
  res.end('Hello World\n');  
}).listen(3000, '127.0.0.1');  
console.log('Server running at http://127.0.0.1:3000/');
```

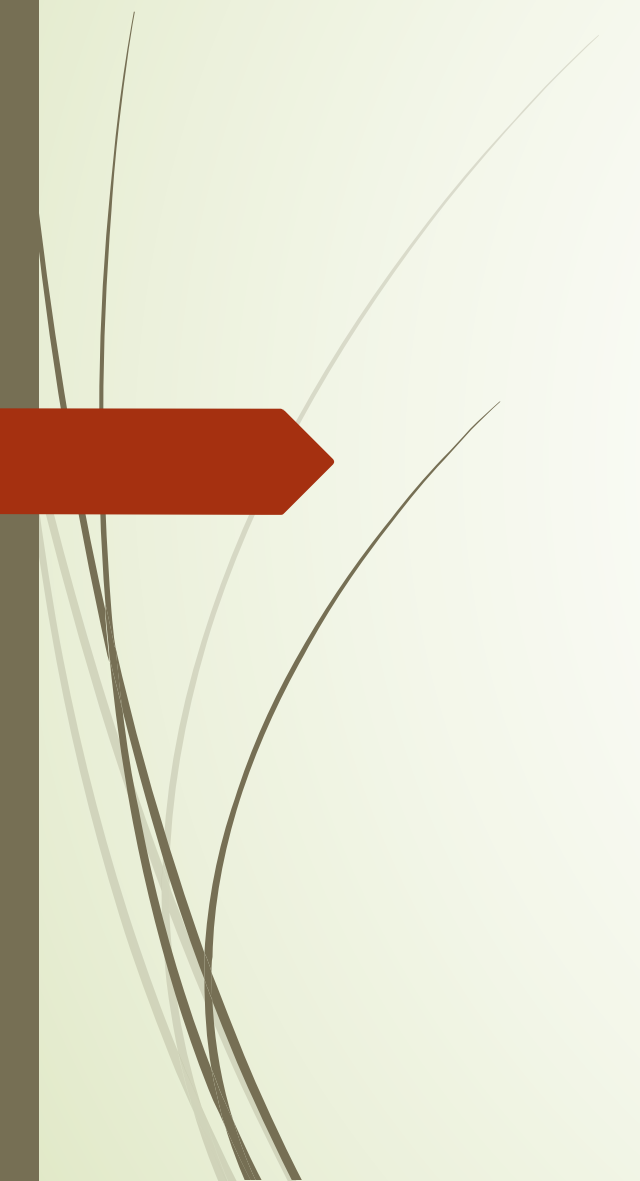
2. \$ node app.js

3. Visit <http://127.0.0.1:3000>



專題作品目標

- CLI Tool 類型：景點天氣查詢 App
 - Web API 類型：賣場商品 RESTful API
- 

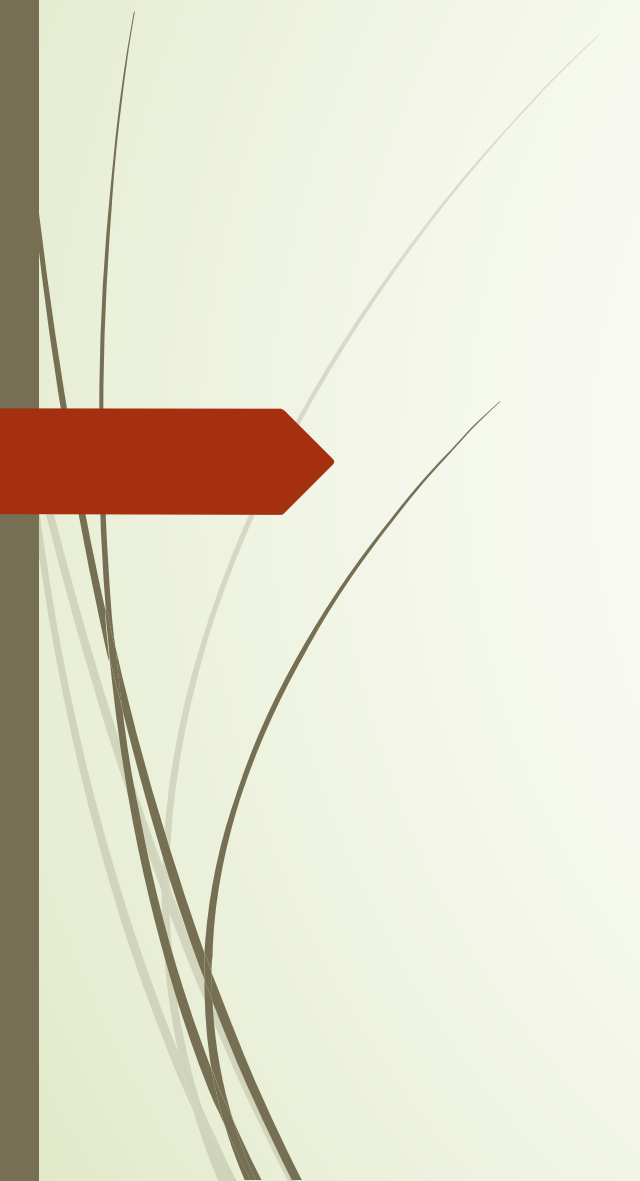


套件引用



Requiring Modules

- Use `'require()'`
- Built-in Modules
 - E.g., `'fs.appendFileSync('test.txt', 'Hello World');'`
- Self-defined Modules
- 3rd party Modules
 - E.g., Lodash (<https://lodash.com>)
 - install manually



套件管理工具：npm

Node Package Manager (npm)

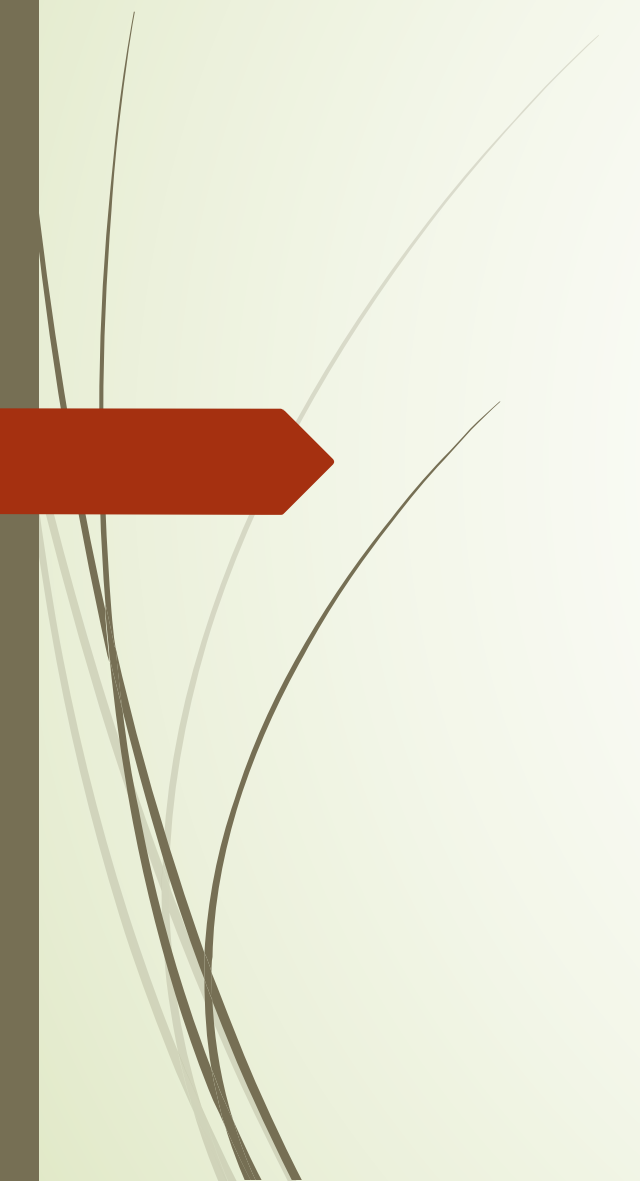
- www.npmjs.com
- 以標準化的方式來管理套件的相依性
 - 包含安裝、移除或更新
 - 'package.json'
- 安裝 Node.js 的同時也會安裝 npm
- npm 安裝會把編譯好的 Library 放到 node_modules 底下
- 補充：Yarn
 - <https://yarnpkg.com/>

NPM 常用指令

Command	Description
\$ npm init	初始化專案的套件控管。
\$ npm install [<pkg-name>]	安裝套件。 沒有指定套件，就是依 package.json 的內容安裝。 e.g., ` \$ npm install lodash `
\$ npm install <pkg-name>@<version>	安裝指定套件且指定版本。 e.g., ` npm install yargs@4.7.1 `
\$ npm uninstall <pkg-name>	移除套件
\$ npm install <pkg-name> -g	全域安裝。 不針對特定哪個 Project，而是針對該版本的 Node.js。 因此不會安裝在該專案的 node_modules 裡，而會裝在目前使用的 Node.js 版本的 lib 路徑裡的 node_modules。
\$ npm install <pkg-name> --save-dev	將此 module 加進 package.json 的 devDependencies 資訊 (deploy 階段不需要的 dependency)。
\$ npm list	列出 node_modules 的相依性樹狀圖。 (如果不存在 package.json 裡，會標示 "extraneous")
\$ npm -v	顯示 npm 版本。

開發輔助工具套件：nodemon

- 用於開發階段，取代 node 指令
 - 會監聽檔案變更，一旦發現檔案變更，就會自動重新執行程式
 - 減少繁複的手動執行負擔
- 預設只會監聽 .js 檔，可以要求監聽其他副檔名類型
 - E.g., '\$ nodemon app.js -e js,html'
- 可以指定要執行的命令，不限定 node
 - E.g., '\$ nodemon --exec 'npm test' '



CLI Tool 類型專題實作

景點天氣查詢 App

需求規格

➡ 輸入執行：

- ➡ ``$ node app.js --addr 'Taipei 101'``
- ➡ ``$ node app.js -a 'Taipei 101'``

➡ 輸出結果：

景點名稱: Taipei 101

處理中...

完整地址: Taipei 101, No. 7, Section 5, Xinyi Road, Xinyi District, Taipei City, Taiwan 110

經緯度: (25.0339639, 121.5644722)

時區: Asia/Taipei

天氣概況: Partly Cloudy

溫度: 16.47 °C

體感溫度: 16.56 °C

利用第三方 API 服務

➤ Google Geo API


- E.g. <https://maps.googleapis.com/maps/api/geocode/json?address=Taipei101>

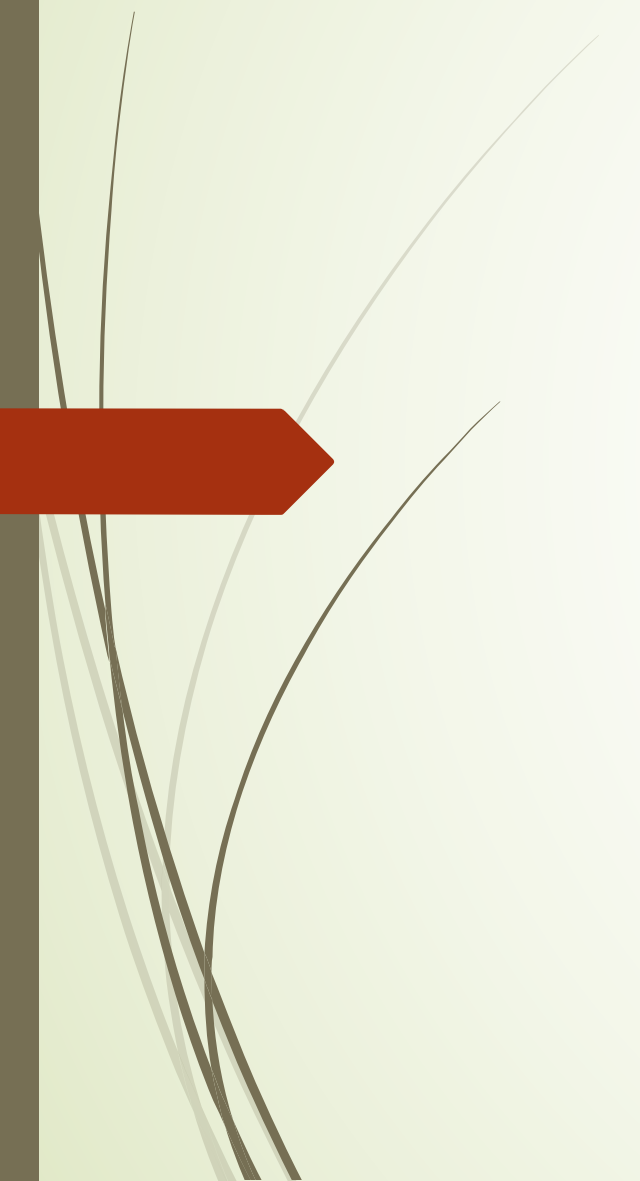
➤ [Weather API: Dark Sky\(forecast.io\)](#)

- [Dark Sky API – Developer](#)
- E.g. <https://api.darksky.net/forecast/{your-forecast-api-access-code}/37.8267,-122.4233>



Developing Keynote

- 1. Parsing command arguments
 - `process.argv`
 - Module: `yarn`
 - 2. Make HTTP calls
 - Module: `request`
- 



Web API 類型專題實作

賣場商品 RESTful API (Express & MongoDB)

需求規格

```
{  
  "name": "iPhone X",  
  "Description": "A revolution in recognition - Face ID.",  
  "price": 31999,  
  "isSoldOut": true  
}
```

- GET <http://localhost:3000/Products>
- GET <http://localhost:3000/Products/{id}>
 - E.g., /Products/5a3bdcf43b46d63b477f9105
- POST <http://localhost:3000/Products> w/ body
- DELETE <http://localhost:3000/Products/{id}>
- PATCH <http://localhost:3000/Products/{id}> w/ body
 - E.g., { "Description": "With iPhone X, the device is the display. " }

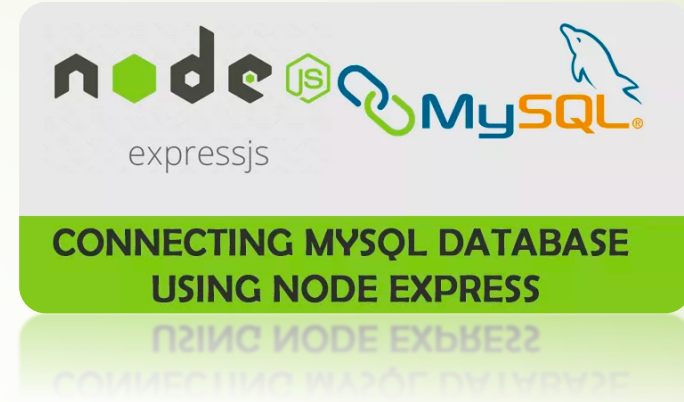
Developing Keynote

➤ 1. Web Service (HTTP APIs)

- Module: http
- Framework: Express

➤ 2. Persistent Data

- DB: MongoDB



MongoDB – Installation




➤ On Mac:

- 参考: [Install MongoDB Community Edition on macOS](#)
- \$ brew update
- \$ brew install mongod

➤ On Windows:

- 参考: [Install MongoDB Community Edition on Windows](#)



MongoDB – Running

- Step 1: Create a folder for dbpath
 - MongoDB 的 DB 資料會放這
- Step2: 啟動 MongoDB Server 服務，並指定 data directory
 - `$ mongod --dbpath <data directory>`

MongoDB – Testing (via Mongo Shell)

- Mongo Shell

- 用一種 JavaScript Shell 作為溝通語言

- Step 1: Enter mongo shell

- `$ mongo`

- Step2: 建立一筆資料

- `$> db.Products.insert({text:"this is my first mongo record"})`

- `$> db. Products.find()`



MongoDB – Operating (via GUI)

- Robo 3T (MongoDB 的 GUI 工具)
 - 官網下載安裝: <https://robomongo.org/>



Appendix

VSCode Extensions



VSCode Extensions

- Git History
 - '\$ git log'
- ESLint
 - Code Quality



Thank you