

# GitHub 專案資訊視覺化系統

專題編號：111-CSIE-S010

執行期限：111 年第 1 學期至 112 年第 1 學期

指導教授：劉建宏

專題參與人員： 109590012 吳定賢  
109590048 劉濬賡  
109590008 莊喆安  
109590057 羅焜銘  
108590009 蔡秉宏

## 一、摘要

此專題是一個針對 Github 此類提供線上軟體原始碼代管服務平台的資料視覺化系統。目的是希望能讓各位開發人員能夠方便並直覺的一覽 Github Repository 的各項資訊。在本專題中，我們使用到了 Java 語言進行後端的開發，透過 Github 提供的 REST API 取得資料並進行整理。前端使用 Vue.js、TypeScript 並透過 Node.js 的 npm 與 vue/cli 套件來整合工具並完成架設網站的工作。最後再透過 Vue.js 提供的模塊，將取得的資料視覺化呈現給用戶。

**關鍵詞：**REST API，Vue.js，資料視覺化

## 二、緣由與目的

因為 Github 原先的介面文字元素較多，配置較複雜。對於只想快速的管理、了解專案資訊的人來說較為不方便。這個系統提供了簡單直覺的介面，讓用戶可以更容易地掌握 Github Repository 的狀況。透過將資料進行視覺化的方式，可以讓用戶更快速、直覺的方式一覽整個專案。而藉由透過本次專案，我們可以學習到如何自行透過網站提供的 API 獲取資料，再另外使用前端網頁的套件來將我們的資料使用圖表的方式展示出來。

## 三、使用技術工具

### (一) Node.js

Node.js 是一個能夠在伺服器端運行 JavaScript 的開放原始碼執行環境。由於 JavaScript 本身是用於使用者端的程式設計語言使用，Node.js 的出現讓 JavaScript 可以在伺服器端執行。而 Node.js 本身也有一系列的內建模組，使得程式可以脫離其他網頁伺服器架設軟體也可以運行。

### (二) TypeScript

TypeScript 是 JavaScript 的超集，實際上我們這次在本專案使用的是 TypeScript 而不是 JavaScript。因為 TypeScript 可以支援型態定義，相對來說物件的型態會更明確。而 TypeScript 本身也對 JavaScript 完全兼容，JavaScript 編寫的程式(.js)可以直接命名為(.ts)直接使用。

### (三) 容器化技術

容器化技術是一個輕量級的虛擬化技術。可以將應用程式的程式碼與該應用程式所需的程式庫、環境設定、資料庫、網路配置等打包成一個容器。可以讓一個應用程式的可攜性大幅提高，也簡化了環境的搭建過程。使得各類應用程式可以更快速的部署、修補及擴展。

## 四、工具說明

(一) IntelliJ IDEA：後端開發環境的 IDE，進行 Java 的環境配置以及負責運行後端 Java 軟體的工作。

(二)SQLite：包含在一個相對小的 C 程式庫中。用來儲存本地端的使用者資訊。

(三)Vue Cli：是一個整合 webpack 以及 webpack-dev-server 的套件，讓我們可以快速架設一套立即可用的前端開發、執行環境。

(四)Docker：容器化技術的管理軟體，我們透過 Docker 來架設、使用容器。

## 五、研究成果

使用者目前可以直接透過 Github 帳號登入。利用 Github 提供的 REST API 取得 JWT token 的回傳以後就可以登入系統。這會儲存在後端的資料庫中並加入在 API 請求的 Header 裡。然後使用者就可以在介面中管理自己的專案。

一開始使用者可以先建立自己的專案，然後就可以進入到專案裡面匯入自己想要關注的 Repository。一個專案底下可以有多个 Repository。進去以後會以圖表的方式顯示 Commit、Contributor 等資訊。

而網頁左上角也提供了 Compare 的選項，可以將兩個 Repository 的資訊並列，讓使用者可以一覽兩個專案的差異。

## 六、結論與未來發展

我們在這次的專題中學習到了如何進行網頁前後端的開發，包含了 3 種程式語言的使用 (Java、TypeScript、Vue)。並且能夠利用網站提供的 API 提取資訊並整理輸出。最後還學到了能夠讓專案可攜性更高、架設更快速的容器化技術。未來這個專案可以再擴展到其他的服務上，例如 SonarCloud 可以提供 Github Repository 的程式碼品質檢測。將這個功能整合進來也可以大大的提升本專案的實用性。

## 七、參考文獻

- [1] [Github REST API documentation](#)
- [2] [Vue.js v2 Document](#)
- [3] [Docker Container Documentation](#)