

# MyWeb 專案靜態資源架構分析報告

本報告旨在分析 myweb 專案中，關於靜態資源（Static Resources）在 **Spring Boot** 與 **傳統 Java Web (WAR)** 架構下的目錄結構與實體對應關係。

## 1. 專案目錄結構概覽

目前專案的實體路徑如下：

- **專案根目錄:** d:\巨匠\Java Web全端開發實務班\Java\_code\MyWork\MyWeb
- **靜態資源主要分佈:**
  - src/main/resources (Spring Boot 預設)
  - src/main/webapp (傳統 Java Web 預設)

## 2. Spring Boot 靜態資源規則

在 Spring Boot 中，靜態資源預設由 `ResourceHttpRequestHandler` 處理。

### 2.1 預設搜尋路徑

Spring Boot 會自動從以下 Classpath 路徑尋找靜態資源：

1. `/static`
2. `/public`
3. `/resources`
4. `/META-INF/resources`

### 2.2 目前專案現況 (Spring Boot 部分)

- **實體路徑:** src/main/resources
- **對應關係:**
  - 目前該目錄下僅有 `application.properties`。
  - 若需放置 CSS/JS/Images，建議在此目錄下建立 `static` 資料夾。

- **優點:** 打包成 JAR 時會自動包含，無需額外配置。

## 3. Java Web (傳統 Servlet/JSP) 靜態資源規則

這是在使用原生 Servlet 或將專案打包為 WAR 時的傳統做法。

### 3.1 預設搜尋路徑

- Web 應用程式的根目錄（即 `src/main/webapp`）。
- 瀏覽器可以直接透過 URL 存取 `webapp` 目錄下（除 `WEB-INF` 與 `META-INF` 外）的所有內容。

### 3.2 目前專案現況 (Java Web 部分)

- **實體路徑:** `src/main/webapp`
- **目前目錄內容:**
  - `hello.html` : 位於根目錄，存取路徑為 `/hello.html`。
  - `images/` : 存放圖片資源。
  - `dist/` : 存放編譯後的靜態資源（如 CSS/JS）。
  - `WEB-INF/` : **受保護目錄**，瀏覽器無法直接存取。內含 `member/register.jsp`。
- **對應關係:**
  - `src/main/webapp/hello.html` -> `http://localhost:8080/hello.html`
  - `src/main/webapp/images/logo.png` -> `http://localhost:8080/images/logo.png`

## 4. 關鍵差異與建議

特性	Spring Boot (Classpath)	Java Web (Webapp)
實體對應	<code>src/main/resources/static</code>	<code>src/main/webapp</code>
封裝方式	封裝在 JAR 內的 <code>BOOT-INF/classes</code>	封裝在 WAR 的根目錄
安全性	預設開放，但可透過配置過濾	<code>WEB-INF</code> 外預設全開放
目前建議	既然已使用 Spring Boot， 建議將通用靜態資源移至 <code>resources/static</code>	目前專案已習慣使用 <code>webapp</code> ， 在開發 JSP 專案時較為直觀

# 總結

本專案目前採用**併行**模式：

1. **JSP 視圖**: 放在 `webapp/WEB-INF/` 受到保護，由 Servlet 轉發。
2. **靜態 HTML/Images**: 放在 `webapp` 根目錄直接提供存取。
3. **Spring 配置**: 放在 `resources` 下。

報告產生時間: 2026-01-14

檔案路徑: `src/main/webapp/docs/static_analysis.md`