



動態文檔圖表生成系統

💡 問題：文檔圖表總是過時

傳統方式的問題

```
![架構圖](static/architecture.png) # 靜態圖片  
![進度圖](static/progress.png)    # 手動更新
```

- ✗ 圖表容易過時
- ✗ 需要手動維護
- ✗ 文檔與實際不同步

✨ LiveDoc 解決方案

動態方式

```
! [架構圖] (http://localhost:3000/architecture) # 即時生成  
! [進度圖] (http://localhost:3000/progress)      # 自動更新
```

- ✓ 圖表永遠最新
- ✓ 零維護成本
- ✓ 標準 Markdown 語法

核心理念

LiveDoc 是什麼？

一個讓 Markdown 文檔中的圖表永遠保持最新的系統。使用標準 Markdown 語法，實現圖表的自動更新。

一句話說明：讓文檔活起來，圖表不再過時。

核心價值：

1. 文檔永不過時 - 圖表即時生成
2. 零學習成本 - 標準 Markdown 語法
3. 零維護成本 - 自動更新

系統架構（精簡版）

基礎架構

Markdown 文檔 → LiveDoc Server → 動態圖表 → 顯示

核心組件

組件	功能
HTTP Server	接收圖片請求，返回動態生成的圖表
Generator	根據數據生成 PNG/GIF 圖表
Cache	智慧快取，平衡即時性與效能

💡 與 GitHub Badges 的對比

GitHub Badges (現有限制)


build passing

```
![build](https://img.shields.io/badge/build-passing-brightgreen)
```

-  簡單易用
-  樣式固定

LiveDoc (突破限制)

```
![系統架構](http://localhost:3000/architecture)
```

-  支援複雜圖表
-  完全客製化
-  標準 Markdown 相容

核心 Demo

Demo 1：基礎動態圖表

展示各種圖表的自動更新能力：

傳統方式

```
 # 靜態，會過時
```

LiveDoc 方式

```
 # 動態，永遠最新
```

可展示的圖表類型：

- 進度條（專案進度）
- 長條圖（API 回應時間）
- 狀態指示（系統健康度）

核心 Demo (2/2)

Demo 2：序列圖動態化（未來擴展）

基於實驗室 PlantUML 規範的創新應用：

- 將靜態序列圖轉為動態
- 顯示執行進度
- 標示當前步驟

註：此為概念展示，展現 LiveDoc 的擴展潛力

技術實現

技術棧

- Node.js + Express - 輕量級 HTTP 服務
- Canvas API + D3.js - 程式化繪製圖表



實施計劃

當前進度

-  核心概念驗證
-  基礎 Demo 完成
-  優化與精簡中

下一步

1. 完善基礎圖表類型
2. 撰寫使用文檔
3. 準備開源發布

總結

LiveDoc 專注於一個簡單但重要的問題：

讓技術文檔的圖表永遠保持最新。

不是要做監控系統，不是要取代 Dashboard，
而是讓文檔不會過時。

核心理念：「簡單、實用、可靠」

