

🤖 AI Collaboration Guide & Rules

Project: HSUS Cloud Transformation (Hybrid Middleware) **Role:** Senior Backend Architect & DevOps Engineer

這份文件是給 AI (Cursor/Windsurf) 的核心指令集。請在開始任何 coding 之前，先讀取此文件。

1. 專案核心任務 (Mission)

我們要將一個基於 Google AppScript 的 Legacy System，依照「絞殺榕模式 (Strangler Fig Pattern)」遷移至 **Google Cloud Run + Supabase**。

- 前端：暫時保留 Google Sheets / AppScript (僅做 UI 與 Proxy)。
- 後端：所有邏輯移至 Cloud Run (Node.js/TypeScript)。
- 資料庫：所有資料移至 Supabase (PostgreSQL)。

2. 技術堆疊 (Tech Stack)

- **Runtime:** Node.js 20+ (TypeScript)
- **Framework:** Express.js
- **Database:** Supabase (PostgreSQL) - 使用 `supabase-js` client。
- **External APIs:** QuickBooks Online (OAuth2), PandaDoc (Webhook).
- **Tools:** Docker (for Cloud Run deployment).

3. 開發三大鐵律 (The 3 Golden Rules) - 嚴格遵守

🚫 Rule #1: AppScript 不准有邏輯 (The "No-Logic" Client)

- AppScript `.gs` 檔案只能做兩件事：
 1. 收集 Sheet 上的 User Input。
 2. 透過 `UrlFetchApp` 呼叫 Cloud Run API。
 3. 接收 API 回傳的 JSON，顯示給 User。
- 禁止在 AppScript 裡面寫複雜的 `if-else`、資料驗證或直接連線 QBO。

⚡ Rule #2: 效能優先 (Performance First)

- **API 端：**所有列表類 API 必須實作 **Pagination (分頁)**。
- **AppScript 端：**嚴禁在迴圈中使用 `sheet.appendRow()` 或 `sheet.getRange()`。
- **強制模式：**必須先構建好二維陣列 `data[][]`，然後使用 `range.setValues(data)` 一次性寫入。

🔒 Rule #3: 資料庫權威 (DB Authority)

- 所有資料 schema 變更必須先寫 SQL Migration (.sql file)，而不是在 TypeScript 裡面硬寫。
- orders 表是核心，order_number 是業務上的 key，但系統內部關聯必須使用 UUID 。

4. 檔案結構規範 (Directory Structure)

請依照以下結構建立檔案，不要隨意發明：

```
src/
├── config/          # env vars (Supabase URL, QBO Keys)
├── controllers/    # Req/Res handling (No business logic here)
├── services/        # Business Logic (Validation, Calculations)
├── repositories/   # DB queries (Supabase calls)
├── types/           # TypeScript interfaces (Sync with DB Schema)
├── utils/           # Logger, API Response Helper
└── routes/          # API Routes definitions
└── app.ts           # Entry point
```

5. 紿 AI 的 Prompt 範本 (Prompt Template)

當我要你寫功能時，我會這樣說：

"實作 [功能名稱]。請參考 System_Design_Spec_v2.1.md 的邏輯，並遵守 API_Interface_Reference.md 的格式。請先建立 SQL Migration，再寫 Service 層，最後寫 Controller。"

End of Context