



技術棧與使用文件

1. 架構概觀

- a. 前端 (Static Site) : Pug → dist 靜態頁, Stylus → CSS。
- b. 後端 (Serverless) : Netlify Functions (tracking.js) 處理 routing (/api/tracking、/api/health、/api/monitoring/stats)。
- c. 配置 : netlify.toml、redirects。

2. 主要技術

- a. 前端 : Pug、Stylus、原生 JS、Chart.js (Admin 可選)。
- b. 後端 : Node 18、Netlify Functions、Airtable JS SDK。
- c. 平台 : Netlify Pro (自訂網域、部署、環境變數)。
- d. 版本控管 : GitHub。

3. 目錄結構 (重點)

- a. netlify/functions/tracking.js (主函式)
- b. index.html (前台查詢頁)
- c. backend/admin.html (監控頁, 可選)
- d. Templates/Styles → 編譯至 dist/
- e. _redirects、netlify.toml

4. 環境變數

- a. AIRTABLE_API_KEY、AIRTABLE_BASE_ID、AIRTABLE_SHIPMENTS_TABLE=Tracking
- b. (可選) API_KEY_xxx (若需要多把金鑰)
- c. NODE_VERSION=18 (netlify.toml)

5. 部署流程

- a. 本地 : npm ci && npm run build → 產出 dist/
- b. Git 推送 → Netlify 自動建置 (netlify.toml)
- c. 確認 Function 路徑 : /.netlify/functions/tracking
- d. 路由 :
 - i. /api/tracking → /.netlify/functions/tracking (200)
 - ii. /api/monitoring/stats → 同上 (200, 可選)
 - iii. /api/health → 同上 (200)
 - iv. /* → /index.html (SPA fallback)

6. API 規格(摘要)

a. GET /api/tracking

- i. query: orderNo(必)、trackingNo(必)
- ii. 200: { success, data: { orderNo, trackingNo, status, eta, lastUpdate, timeline[] } }
- iii. 404: { error: 'Not found' }
- iv. 429: { error: 'Too many requests' }
- v. 500: { error: 'Internal server error' }

b. GET /api/monitoring/stats(可選)

- i. 200: { today, thisMonth, tracking: { averageResponseTime, successRate, successfulQueries, totalQueries }, system: { totalRequests }, recentRequests: [{ time, method, path, status, responseTime, orderNo, trackingNo }] }

c. GET /api/health

- i. 200: { status: 'ok' }

7. 錯誤處理與日誌

- a. try/catch → 500; Airtable 失敗紀錄錯誤、回傳訊息。
- b. 404: 無資料。
- c. 監控: recordMonitoringEntry() 記錄 timestamp、path、success、durationMs、orderNo、trackingNo。
- d. Netlify Logs 觀測。

8. 安全與限流

- a. IP + API Key(可選) rate limit(預設匿名 100/hr、有 Key 50/hr 可調)。
- b. CORS 僅允許必要來源。
- c. 不回傳敏感欄位, Airtable 欄位白名單輸出。

9. 前端使用說明(track.tailormed-intl.com)

- a. URL 參數: ?orderNo=&trackingNo=(亦相容 ?order=&tracking=)(可選)。
- b. 查詢成功時以 history.pushState 更新 URL。
- c. RWD 與樣式: Stylus 編譯; 時間軸樣式依資料切換。

10. 本地開發

- a. npm run dev
- b. 修改後端: functions 熱更新; 前端: 直接預覽 dist/ 或開發模板編譯。