

台股爆發預警系統 (TSBS) 開發技術說明書

1. 專案願景與目標

構建一個輕量化、高性能的 Web App，專門監測台股中具備「短期內爆發性成長」特徵的股票。系統需自動篩選「量能激增」、「均線糾結突破」與「法人佈局」三大核心指標，並透過 PWA 技術在 iOS 設備上提供類似原生 App 的使用體驗。

2. 技術棧 (Technical Stack)

- 框架: Next.js 14+ (App Router)
- 語言: TypeScript (嚴格類型定義)
- 樣式: Tailwind CSS + Shadcn UI
- 狀態與數據: TanStack Query (React Query) 用於頻繁的 API 緩存管理
- 圖表: Lightweight Charts (TradingView) 或 Recharts
- 部署: GitHub 存儲庫 + Netlify
- PWA: next-pwa (支持 Offline 緩存與 iOS 桌面圖標)

3. 數據源與 API 整合 (FinMind API)

參考 <https://api.finmindtrade.com/api/v4/data>，系統需實作以下接口封裝：

A. 核心數據接口

個股日行情 (TaiwanStockPrice): 獲取收盤價、成交量、最高/最低價。

三大法人持股 (TaiwanStockHoldingSharesPer): 篩選 Investment_Trust (投信) 買賣超。

月營收 (TaiwanStockMonthRevenue): 驗證基本面是否同步走強。

4. 核心算法: 爆發評分引擎 (The Detection Engine)

系統需計算每支股票的 爆發潛力分 (Breakout Score):

A. 數學公式

$$Score = (V_{ratio} \times 0.4) + (MA_{alignment} \times 0.3) + (Inst_{trend} \times 0.3)$$

B. 指標定義

量能比 (V_{ratio}):

$$V_{ratio} = \frac{\text{當日成交量}}{\text{過去 20 日平均成交量}}$$

觸發條件: $V_{ratio} \geq 3.0$ 。

均線擠壓 ($MA_{alignment}$):

計算 $MA5, MA10, MA20$ 。

判斷「糾結度」: $\frac{\text{Max}(MA) - \text{Min}(MA)}{\text{Min}(MA)} < 0.03$ 。

判斷「突破」: $\text{Price}_{close} > \text{Max}(MA5, MA10, MA20)$ 。

籌碼動向 ($Inst_{trend}$):

判斷投信是否連續 3 日買超。

5. UI/UX 與自適應需求

A. 響應式佈局 (Adaptive Layout)

豎屏 (Portrait): 以 Card 形式顯示預警清單，顯示代號、名稱、目前漲幅、爆發分數。

橫屏 (Landscape): 自動切換為「技術分析模式」。隱藏導航欄，K 線圖填充 100% 視窗高度。

B. PWA 與 iOS 優化

Manifest: 配置 icons, theme_color, background_color。

Apple Support: 必須包含 `<meta name="apple-mobile-web-app-capable" content="yes">` 以支援 iOS「加入主畫面」。

旋轉觸發: 監聽 `orientationchange` 事件, 調整圖表 `height` 與 `width`。

6. 開發里程碑 (Milestones for Agent)

第一階段: 基礎設施與數據

[] 初始化 Next.js 並安裝 Shadcn UI。

[] 完成 `finmindClient` 封裝。

[] 實作伺服器端 (Server Action) 獲取全台股清單。

第二階段: 引擎開發

[] 實作 `calculateTechnicalIndicators` 函數。

[] 實作 `breakoutScanner` 邏輯, 能過濾出當日符合條件的 Top 20 標的。

第三階段: 前端展示

[] 構建 Dashboard 頁面。

[] 整合 `Lightweight Charts` 繪製 K 線圖與均線。

[] 實現「橫屏全屏圖表」功能。

第四階段: PWA 與部署

[] 配置 `next-pwa` 與 `manifest.json`。

[] 設置 `Netlify` 環境變量並完成自動部署。

7. 部署說明 (Deployment)

環境變數: 在 `Netlify` 中設置 `NEXT_PUBLIC_FINMIND_TOKEN`。

構建命令: `npm run build`。

自動化: 只要 `GitHub` 提交新代碼, `Netlify` 自動執行部署並更新 PWA