

LINE 通知中心 v2.3 (SQLite 版)

完整文字說明版使用手冊

系統架構 : LINE OA × Apache Reverse Proxy × Flask × SQLite

文件產生日期 : 2026-02-27

一、系統整體架構說明（文字說明）

本系統為一套 LINE 官方帳號整合通知中心平台。系統運作流程如下：使用者透過 LINE 官方帳號與系統互動，LINE 平台將事件透過 Webhook 傳送至 Apache 伺服器（HTTPS 443），Apache 再將請求反向代理至本機 127.0.0.1:5000 上的 Flask 應用程式。Flask 負責所有業務邏輯處理，並將資料寫入 SQLite 資料庫（line_center.db）。

架構層級可分為四層：第一層為使用者端（LINE App）；第二層為 Apache SSL 反向代理層；第三層為 Flask 應用程式邏輯層；第四層為 SQLite 資料儲存層。

二、使用者綁定完整流程（逐步說明）

- 步驟 1：使用者必須先對 LINE 官方帳號傳送任一訊息。此動作會觸發 Webhook，並在 identities 表中建立 userId 紀錄。
- 步驟 2：使用者開啟 <https://test.prof-jj.com/line/login>，輸入手機號碼。
- 步驟 3：系統將使用者導向 LINE Login 官方授權頁面。
- 步驟 4：使用者完成 LINE Login 驗證後，LINE 會呼叫 callback URL。
- 步驟 5：Flask 使用 LINE Login API 驗證 id_token，取得 userId。
- 步驟 6：系統將手機號碼與 userId 進行綁定，並產生會員 token。
- 步驟 7：系統自動導向 /member?t=token 頁面，讓使用者填寫姓名、Email、群組、備註。
- 步驟 8：資料儲存至 members 表，並同步更新 identities.note。

三、API 推播完整說明與範例

API 基本格式：<https://test.prof-jj.com/api/push?參數&msg=訊息內容&key=APIKEY>

- 範例 1：單一 userId 推播：[/api/push?id=Uxxxxxxxxx&msg=你好&key=APIKEY](https://api/push?id=Uxxxxxxxxx&msg=你好&key=APIKEY)
 - 範例 2：多手機推播：[/api/push?phone=0912345678,0922333444&msg=通知&key=APIKEY](https://api/push?phone=0912345678,0922333444&msg=通知&key=APIKEY)
 - 範例 3：群組模糊查詢：[/api/push?group=醫管三&msg=上課提醒&key=APIKEY](https://api/push?group=醫管三&msg=上課提醒&key=APIKEY)
 - 範例 4：姓名模糊查詢：[/api/push?name=王&msg=會議通知&key=APIKEY](https://api/push?name=王&msg=會議通知&key=APIKEY)
 - 範例 5：多姓名模糊查詢：[/api/push?name=王,陳&msg=公告&key=APIKEY](https://api/push?name=王,陳&msg=公告&key=APIKEY)
- 系統會自動去除重複 line_id。若無目標，回傳 404。若 API key 錯誤，回傳 403。成功則回傳 JSON 結果與發送統計。

四、Admin 後台完整說明

- 後台網址：<https://test.prof-jj.com/admin>
- 需使用 Basic Auth 帳號密碼登入。
- 功能包含：
 1. 列出 identities 清單（包含 id_type、line_id、phone、note、last_seen）。
 2. 對單一使用者發送推播訊息。
 3. 編輯會員資料（手機、姓名、Email、群組、備註）。
 4. 刪除 identities 與對應 members 資料。
 5. 匯出 CSV 檔案供行政用途。

五、SQLite 資料庫結構說明

- identities 表：儲存 LINE userId / groupId / roomId。欄位包含 id_type、line_id、phone、note、member_token。
- members 表：phone 為主鍵。欄位包含 name、email、group_name、remark、line_id。
- login_states 表：儲存 LINE Login 暫存 state 與有效期限。

六、LOG 機制與資料流程說明

目前系統於 API 推播時會回傳 JSON 結果。建議未來新增 logs 資料表，紀錄以下欄位：發送時間、發送人、目標數量、成功數量、失敗原因。資料流程為：API 請求進入 驗證 key 查詢目標 呼叫 LINE Push API 回傳結果（未來可寫入 logs）。

七、安全與最佳化建議

- 建議 Flask 不使用內建開發模式，改用 waitress 伺服器。
- API key 請使用高強度隨機字串。
- Apache ProxyTimeout 建議設定為 120 秒以上。
- LINE Login redirect_uri 必須與後台設定完全一致（HTTPS）。
- 建議定期備份 SQLite 資料庫。

文件版本：v2.3 文字完整版