

專業排版與設計 期中考題

LINE Messaging API 教學文件製作

座號: 017

班級: 資管 4A

學號: 411631277

姓名: 張誠恩

November 9, 2025

目錄

1	LINE Messaging API 簡介與應用場景	3
1.1	LINE Messaging API 概述	3
1.2	常見應用場景與發展趨勢	3
2	LINE Developers 帳號註冊申請與 Channel 建立流程	4
3	Channel Secret 及 Access Token 取得與設定	5
4	Webhook 的設定與測試方式（可使用 ngrok 或伺服器）	6
5	範例程式說明（接收及回覆 LINE 訊息）	7

圖目錄

2.1 圖片範例展示	4
----------------------	---

表目錄

程式碼目錄

5.1 Python app.py 範例	7
--------------------------------	---

第 1 章 LINE Messaging API 簡介與應用場景

1.1 LINE Messaging API 概述

LINE Messaging API 是由 LINE 公司提供的雲端服務介面，旨在讓開發者能以程式方式操作 LINE 聊天平台的互動功能。透過此 API，開發者可以建立能夠主動推播訊息、回覆使用者輸入、甚至與後端資料庫互動的自動化聊天系統。

Messaging API 採用 RESTful 架構，所有互動皆以 HTTPS 為基礎的 POST/GET 請求完成。開發者透過 LINE Developers 平台建立一個「Channel」，並使用「Channel Access Token」來進行身分驗證與授權。LINE 平台再透過 Webhook 機制，將使用者事件（例如訊息、加好友、按鈕互動等）傳送至開發者設定的伺服器。

此技術的誕生使得企業不再需要自行開發聊天系統，而能以 LINE 為前端通道，將客戶溝通、行銷與服務集中於單一平台。這大幅降低開發成本，同時擴大應用覆蓋範圍。

1.2 常見應用場景與發展趨勢

Messaging API 的應用非常廣泛，以下列出幾個具代表性的實際場景：

1. 智慧客服機器人: 許多企業利用 LINE Bot 回覆常見問題，整合 FAQ、訂單查詢、報修狀態查詢等功能，減少人力成本。
2. 行銷與推播通知: 透過 Broadcast Message 或 Multicast API，企業可向特定用戶群發送促銷訊息，並根據互動紀錄分析點擊率。
3. 服務整合平台: 結合 Google Sheets、Firebase 或 MySQL，可實現會員查詢、預約系統、積分兌換等功能。
4. 教育與社群應用: 學校可利用 LINE Bot 提供課程公告、作業提醒，社群團體亦能透過 Bot 管理活動報名。

展望未來，LINE Messaging API 將持續朝「智慧化」、「個人化」與「跨平台整合」方向發展。AI 對話與使用者行為分析將成為關鍵技術，使 Bot 不僅能回覆問題，更能主動理解與預測使用者需求。

第 2 章 LINE Developers 帳號註冊申請與 Channel 建立流程

2.1 LINE Developers 帳號建立

要使用 Messaging API，第一步是註冊 LINE Developers 帳號。開發者可前往 <https://developers.line.biz/> 網站，以現有的 LINE 帳號登入。進入後可建立一個或多個「Provider」——代表開發者或公司身分。

建立 Provider 後，平台會要求設定名稱、聯絡信箱與用途描述。Provider 相當於開發者群組，而每個 Provider 底下可以擁有多個 Channel，用以區分不同應用程式或專案。

在帳號註冊完成後，開發者即可在主控台中看到專案清單與狀態，並能管理權限、API 使用紀錄等資訊。



圖 2.1: 圖片範例展示

第 3 章 Channel Secret 及 Access Token 取得與設定

cccccc

第 4 章 Webhook 的設定與測試方式（可使用 ngrok 或伺服器）

dddddd

第 5 章 範例程式說明（接收及回覆 LINE 訊息）

程式碼 5.1: Python app.py 範例

```
1 from flask import Flask, request, abort
2 import os
3 from linebot.v3 import WebhookHandler
4 from linebot.v3.messaging import Configuration, ApiClient, MessagingApi,
    ReplyMessageRequest, TextMessage
5 from linebot.v3.exceptions import InvalidSignatureError
6 from linebot.v3.webhooks import MessageEvent, TextMessageContent
7 from dotenv import load_dotenv
8
9 # 載入環境變數
10 load_dotenv()
11
12 app = Flask(__name__)
13
14 # 讀取 LINE Credentials 從環境變數
15 CHANNEL_SECRET = os.getenv("LINE_CHANNEL_SECRET")
16 CHANNEL_ACCESS_TOKEN = os.getenv("LINE_CHANNEL_ACCESS_TOKEN")
```