# APA Hotel Woodbridge Instagram & Facebook 自動回覆機器人技術文件

## 目的

打造一個可自動回覆 IG 和 Facebook 訊息的多語言聊天機器人，服務 APA Hotel Woodbridge 粉絲與潛在顧客。

## 系統功能概述

* 支援平台：Instagram + Facebook Messenger
* 語言支援：中文、英文、日文（自動判斷）
* 功能內容：
  + 回覆訂房連結
  + 回覆飯店地址
  + 回覆入住/退房時間
  + 回覆一般感謝語

## 架構與技術

* 語言：Python
* Web 框架：Flask
* 伺服器部署：Render（推薦）
* 使用 Facebook Graph API
* 使用 Webhook 接收訊息事件

## 檔案結構

project/

├── webhook.py # 主程式

├── .env # 環境變數（不提交 GitHub）

└── requirements.txt # 套件清單（供 Render 使用）

## .env 環境變數格式

VERIFY\_TOKEN=your\_verify\_token\_here

PAGE\_ACCESS\_TOKEN=your\_page\_access\_token\_here

## webhook.py 程式功能解說

from flask import Flask, request

import os

import requests

from dotenv import load\_dotenv

load\_dotenv()

VERIFY\_TOKEN = os.getenv("VERIFY\_TOKEN")

PAGE\_ACCESS\_TOKEN = os.getenv("PAGE\_ACCESS\_TOKEN")

app = Flask(\_\_name\_\_)

REPLIES = {

"booking": {

"en": "Thank you for your interest in APA Hotel Woodbridge. You can make a reservation here: https://www.apahotelwoodbridge.com/",

"ja": "ご予約はこちら：https://www.apahotelwoodbridge.com/",

"zh": "感謝您聯絡 APA Hotel Woodbridge。訂房請點選：https://www.apahotelwoodbridge.com/"

},

"location": {

"en": "Our address is: 120 Wood Avenue South, Iselin, NJ 08830.",

"ja": "住所：120 Wood Avenue South, Iselin, NJ 08830。",

"zh": "地址：120 Wood Avenue South, Iselin, NJ 08830。"

},

"checkin": {

"en": "Check-in time: 3:00 PM / Check-out: 11:00 AM.",

"ja": "チェックイン：15:00 / チェックアウト：11:00。",

"zh": "入住時間：下午3點，退房時間：上午11點。"

}

}

# 偵測語言（簡化方式）

def detect\_lang(text):

if any(word in text.lower() for word in ["的", "請問", "您好"]):

return "zh"

elif any(word in text.lower() for word in ["です", "予約", "こんにちは"]):

return "ja"

else:

return "en"

@app.route("/webhook", methods=["GET", "POST"])

def webhook():

if request.method == "GET":

token = request.args.get("hub.verify\_token")

challenge = request.args.get("hub.challenge")

return challenge if token == VERIFY\_TOKEN else "Invalid token"

if request.method == "POST":

data = request.get\_json()

print("Incoming:", data)

for entry in data.get("entry", []):

for messaging in entry.get("messaging", []):

sender\_id = messaging["sender"]["id"]

message\_text = messaging.get("message", {}).get("text", "")

lang = detect\_lang(message\_text)

if "訂房" in message\_text or "book" in message\_text.lower():

reply\_text = REPLIES["booking"][lang]

elif "地址" in message\_text or "address" in message\_text.lower():

reply\_text = REPLIES["location"][lang]

elif "入住" in message\_text or "check-in" in message\_text.lower():

reply\_text = REPLIES["checkin"][lang]

else:

reply\_text = {

"en": "Thank you! We will get back to you shortly.",

"zh": "謝謝您的訊息，我們將盡快回覆。",

"ja": "ありがとうございます。すぐに返信いたします。"

}[lang]

send\_message(sender\_id, reply\_text)

return "OK", 200

def send\_message(recipient\_id, text):

url = "https://graph.facebook.com/v17.0/me/messages"

headers = {"Content-Type": "application/json"}

payload = {

"recipient": {"id": recipient\_id},

"message": {"text": text},

"messaging\_type": "RESPONSE",

"access\_token": PAGE\_ACCESS\_TOKEN

}

response = requests.post(url, headers=headers, json=payload)

print(f"[Bot] ➜ to {recipient\_id}: {text}")

print("[Graph API Response]", response.status\_code, response.text)

if \_\_name\_\_ == "\_\_main\_\_":

app.run(debug=True)

## 部署到 Render

1. 將程式 push 至 GitHub Repository
2. 到 [https://render.com](https://render.com/) 註冊帳號
3. 選擇：New + Web Service
4. 選擇語言：Python + 選擇 GitHub Repo
5. 設定欄位：
   * Start command：gunicorn webhook:app
   * Secret File：新增 .env 內容，包含 VERIFY\_TOKEN 和 PAGE\_ACCESS\_TOKEN
6. 點擊 Deploy
7. 複製 webhook URL，格式為：<https://xxx.onrender.com/webhook>

## IG / FB 串接設定步驟

**IG 必備條件**

* 將 IG 帳號升級為商業帳號（專業帳號）
* 綁定一個 Facebook 粉專

**Meta 開發者設定**

1. 前往 <https://developers.facebook.com/>
2. 建立應用程式（類型：企業）
3. 加入以下產品：
   * Instagram Graph API
   * Messenger
   * Webhooks
4. Webhook 訂閱：
   * 訂閱 messages 權限
   * 填入驗證網址 + verify\_token（與程式一致）
5. 設定權杖（使用 Graph API Explorer 取得 Page Access Token）

## 目前進度與未完成項目

### ✅ 已完成

* webhook.py 多語支援版本
* .env 設定
* Render 部署方式文件
* IG / FB 整體技術說明

### 🔜 未完成 / 可擴充

* 串接 send\_message 中的 Graph API 並測試
* 加入「常見問答資料庫」或 ChatGPT API
* 介面管理面板（Ex: 回覆記錄、客服模式切換）

## 常見回覆模版

| **類型** | **英文** | **中文** | **日文** |
| --- | --- | --- | --- |
| 訂房 | Thank you for your interest... | 感謝您聯絡... | ご予約はこちら... |
| 地址 | Our address is... | 地址是... | 住所は... |
| 入住 | Check-in: 3PM... | 入住時間是... | チェックインは... |
| 感謝 | Thank you... | 謝謝您... | ありがとうございます... |

## 已完成更新 已將技術文件調整為完整可部署的版本，包含：

* 多語系自動回覆設計（中、英、日）
* Flask 程式完整架構說明與語言偵測
* Render 部署說明
* Facebook / Instagram 串接教學
* 常見訊息回覆範本表格整理

## 下一步建議

1. **串接測試 send\_message 函式：**
   * 需提供粉專管理者權限與 Page Access Token。
2. **後續可加入：**
   * 用戶問題關鍵字資料庫
   * ChatGPT API 回覆更彈性內容
   * 客服切換開關（自動 / 人工）
   * 聊天紀錄儲存（如 SQLite / Google Sheet）