

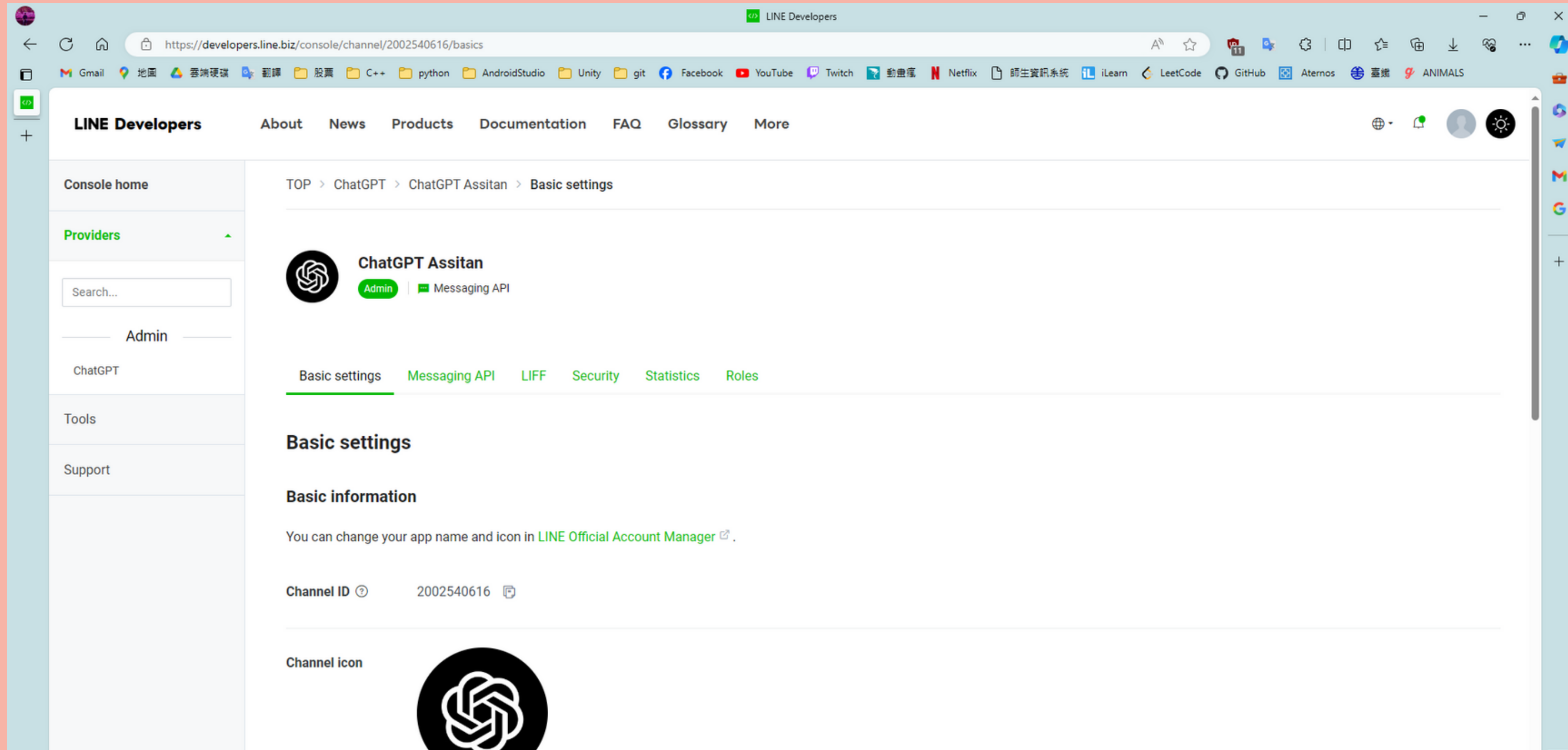


Line Bot連結 ChatGPT

S09350356 呂崇維

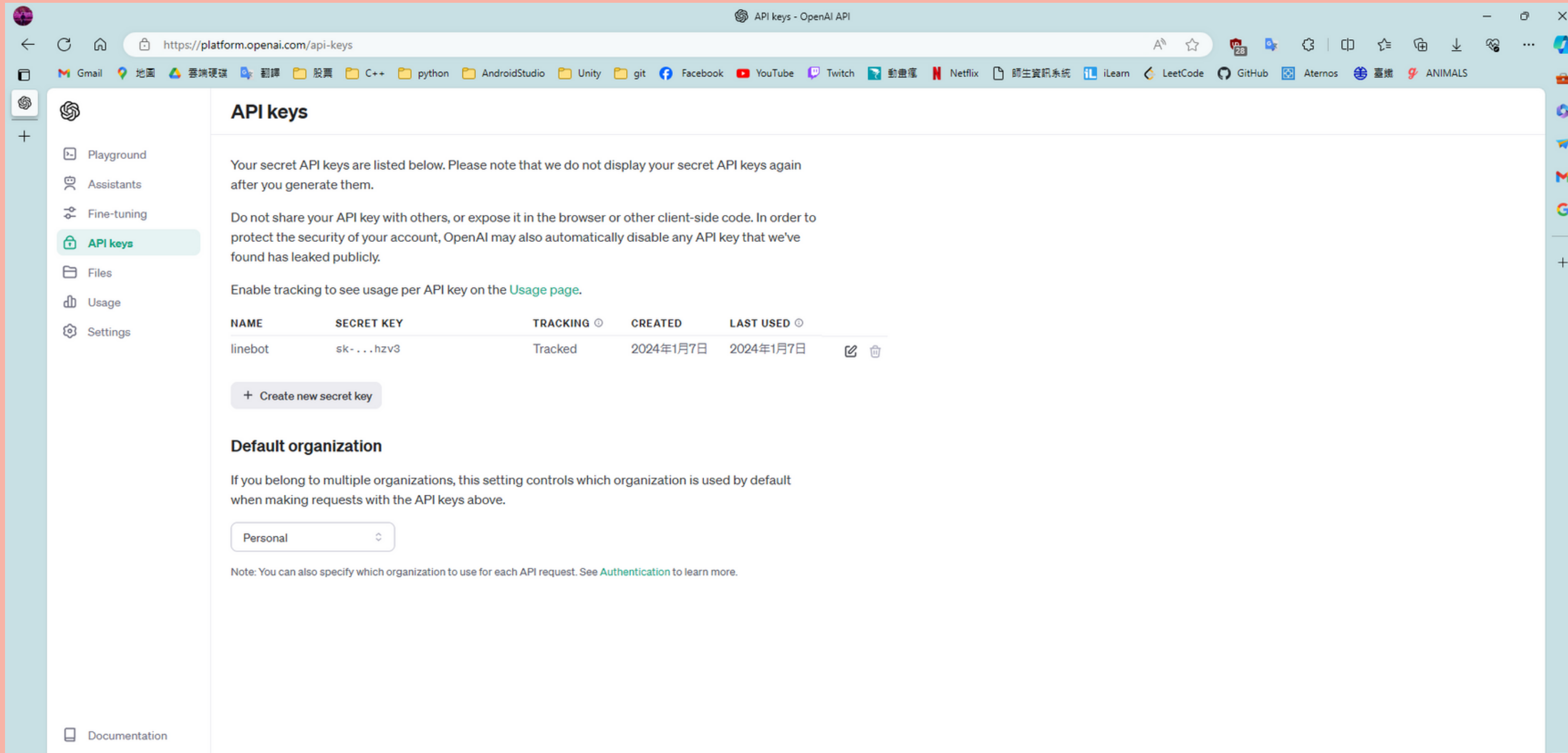
註冊LINE Bot

利用LINE Developers註冊，並取得Channel access token和Channel secret



取得OpenAI的API Key

search



The screenshot shows the OpenAI API keys management interface in a web browser. The browser's address bar displays the URL `https://platform.openai.com/api-keys`. The page title is "API keys - OpenAI API".

API keys

Your secret API keys are listed below. Please note that we do not display your secret API keys again after you generate them.

Do not share your API key with others, or expose it in the browser or other client-side code. In order to protect the security of your account, OpenAI may also automatically disable any API key that we've found has leaked publicly.

Enable tracking to see usage per API key on the [Usage page](#).

NAME	SECRET KEY	TRACKING ⓘ	CREATED	LAST USED ⓘ	
linebot	sk-...hzv3	Tracked	2024年1月7日	2024年1月7日	✎ 🗑

[+ Create new secret key](#)

Default organization

If you belong to multiple organizations, this setting controls which organization is used by default when making requests with the API keys above.

Personal ⓘ

Note: You can also specify which organization to use for each API request. See [Authentication](#) to learn more.

[Documentation](#)

利用Flask建立伺服器端

設定 LINE Bot 和 ChatGPT 的相關連接參數

```
# Line Bot的Channel Secret和Channel Access Token
line_bot_api = LineBotApi('')
handler = WebhookHandler('')

# ChatGPT的API key和API URL
chatgpt_api_key = ''
chatgpt_api_url = 'https://api.openai.com/v1/chat/completions'
```



利用Flask建立伺服器端

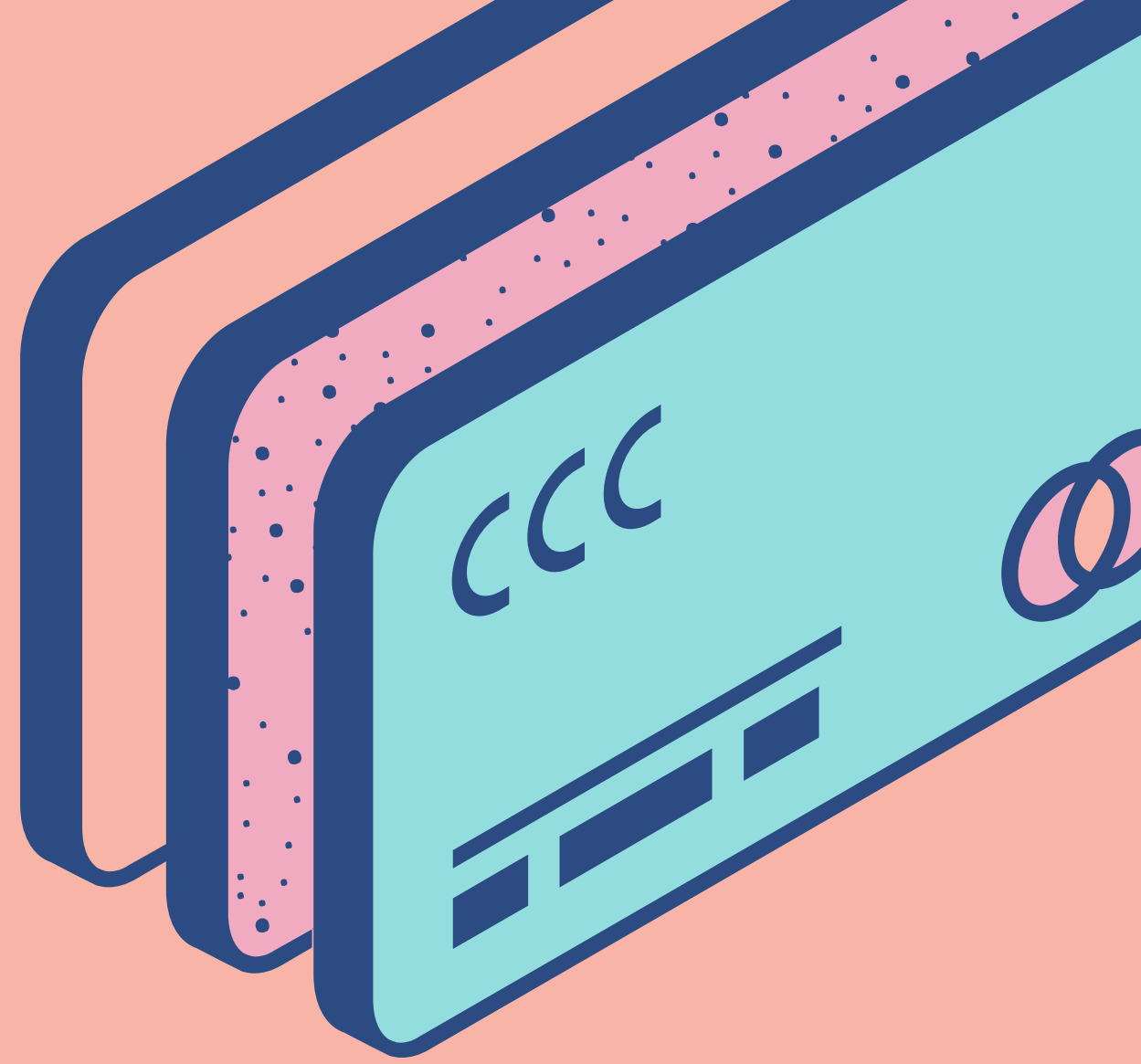
建立route接收傳送給LINE Bot的訊息

```
@app.route("/callback", methods=['POST'])
def callback():
    signature = request.headers['X-Line-Signature']

    body = request.get_data(as_text=True)
    app.logger.info("Request body: " + body)

    try:
        handler.handle(body, signature)
    except InvalidSignatureError:
        abort(400)

    return 'OK'
```



利用Flask建立伺服器端

將接收到的訊息傳送到ChatGPT，並處理回傳訊息

```
def get_chatgpt_response(user_message):
    headers = {'Authorization': f'Bearer {chatgpt_api_key}'}
    data = {'messages': [{'role': 'system', 'content': '
You are a helpful assistant.'}, {
        'role': 'user', 'content': user_message}], 'model': '
gpt-3.5-turbo'}

    response = requests.post(chatgpt_api_url, json=data, headers=
headers)
    response_json = response.json()

    chatgpt_response = response_json['choices'][0]['message']['
content']

    return chatgpt_response
```



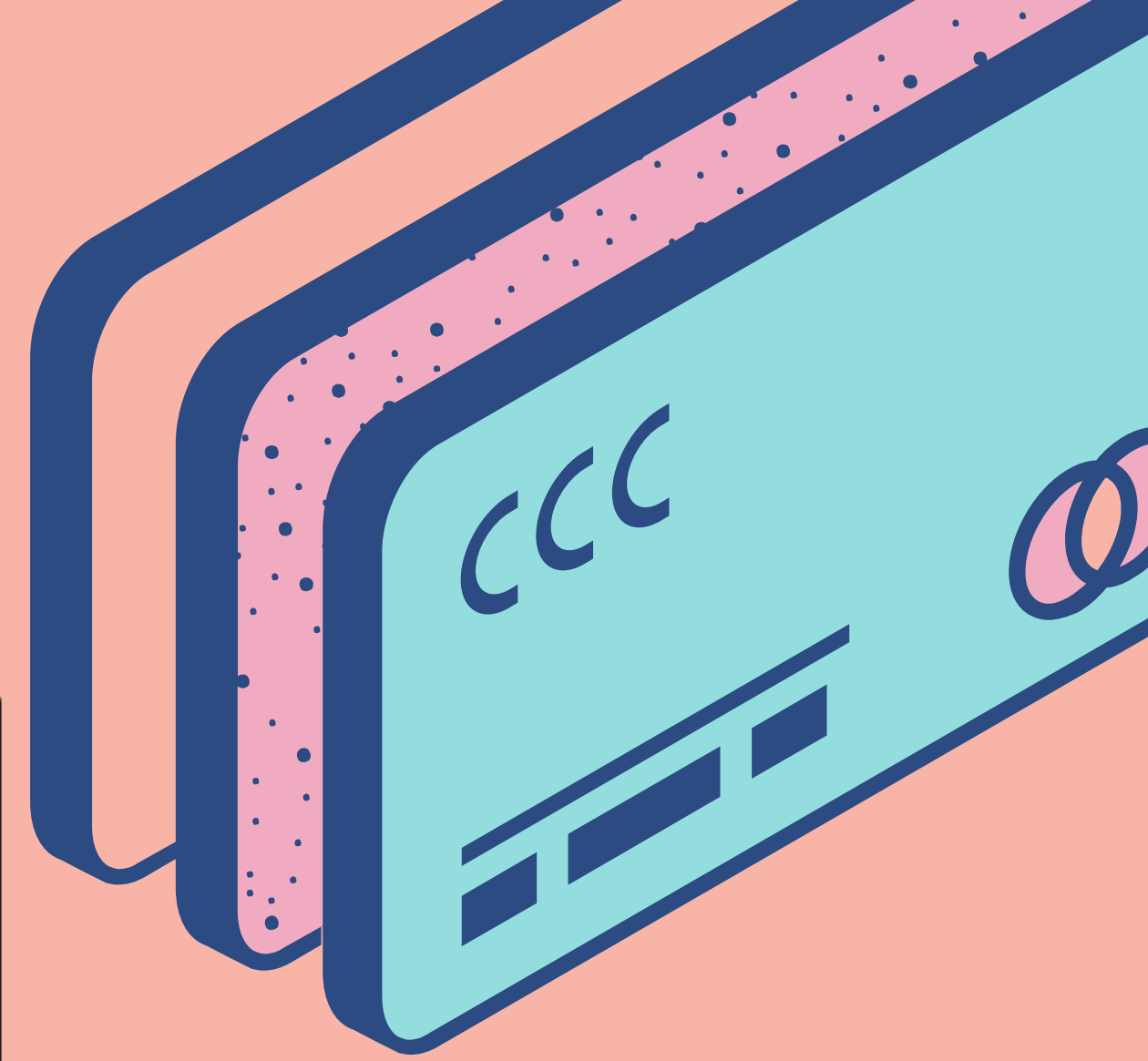
利用Flask建立伺服器端

將處理好的內容回傳給LINE Bot

```
@handler.add(MessageEvent, message=TextMessage)
def handle_message(event):
    user_message = event.message.text

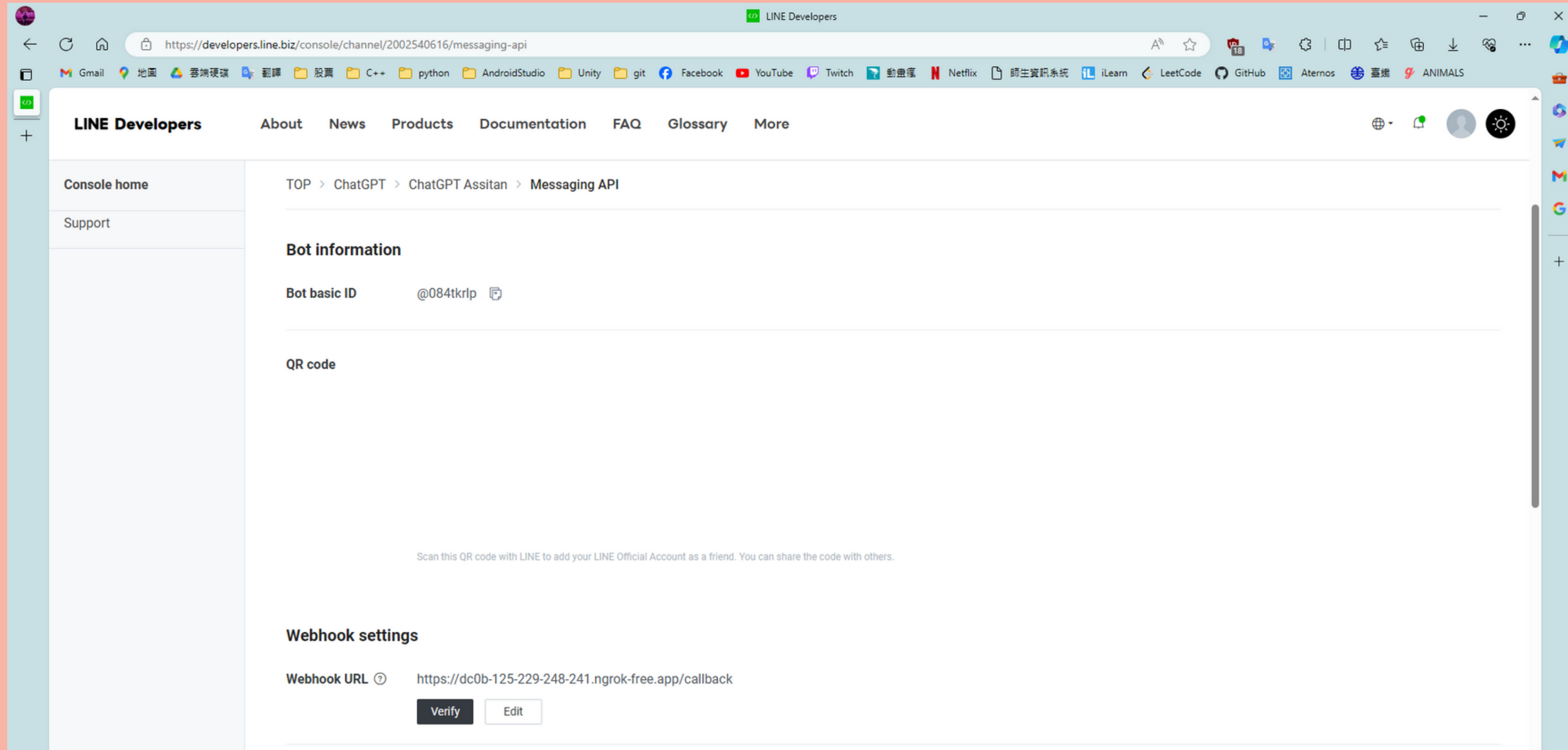
    # 使用ChatGPT API處理用戶消息
    chatgpt_response = get_chatgpt_response(user_message)

    # 發送ChatGPT的回覆到Line
    line_bot_api.reply_message(
        event.reply_token,
        TextSendMessage(text=chatgpt_response)
    )
```



將伺服器上傳到LINE Bot

將伺服器IP上傳到Webhook URL



透過NGROK轉為HTTPS

search

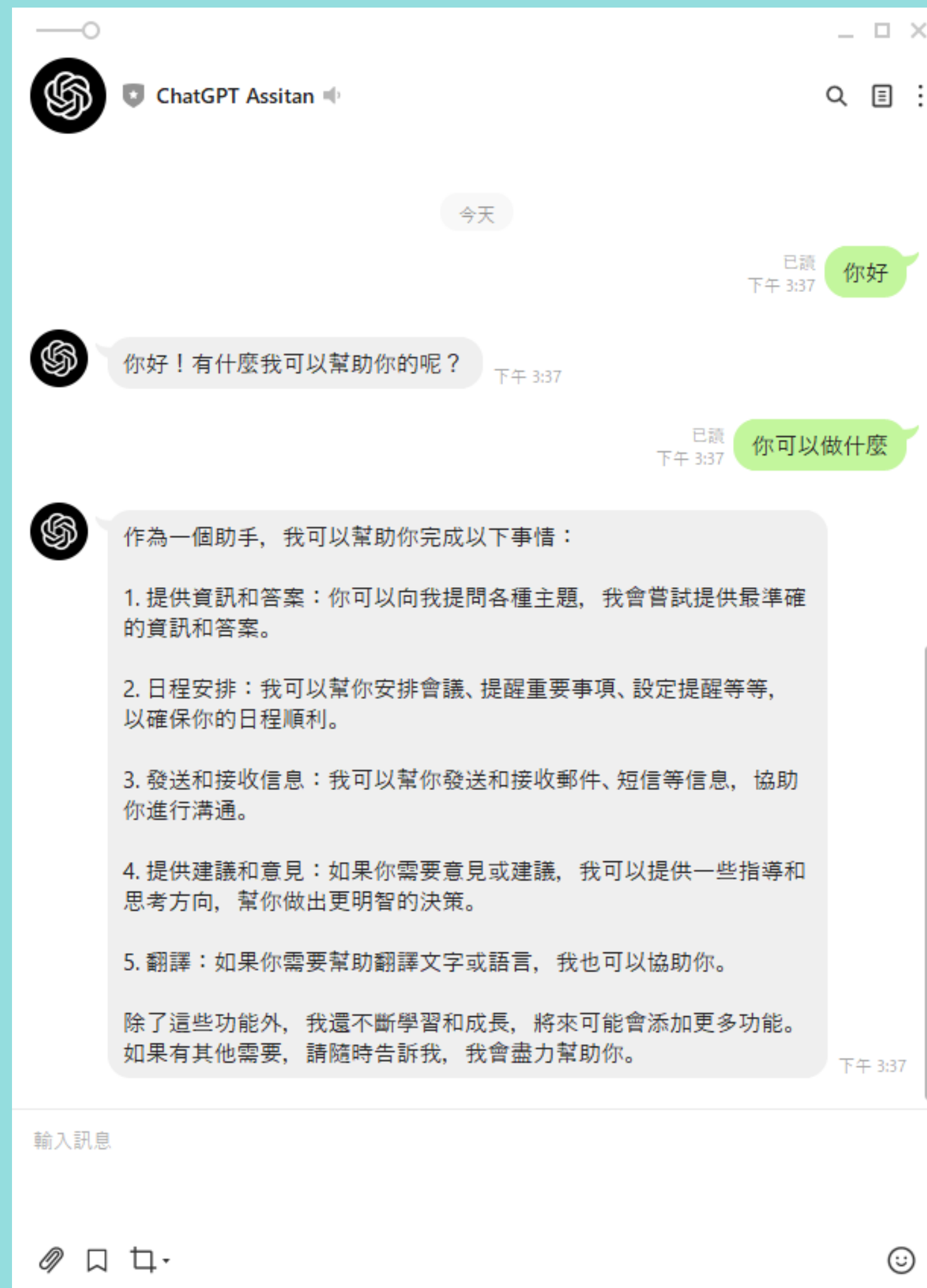
```
C:\Users\bucha\Downloads\ngrok-v3-stable-windows-amd64\ngrok.exe - ngrok http 80
ngrok (Ctrl+C to quit)
Build better APIs with ngrok. Early access: ngrok.com/early-access

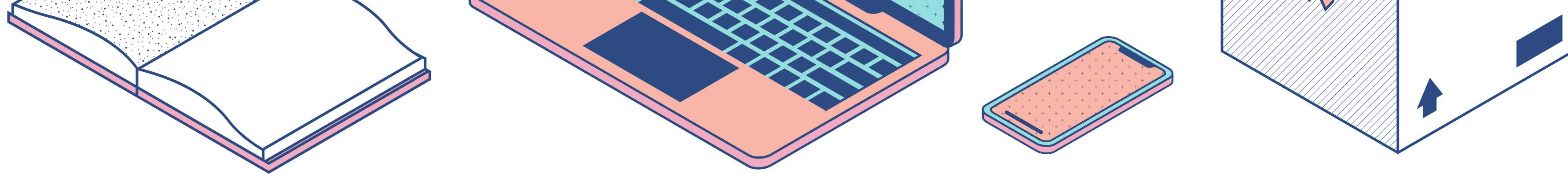
Session Status      online
Account             s09350356@go.thu.edu.tw (Plan: Free)
Version             3.5.0
Region              Japan (jp)
Latency             40ms
Web Interface       http://127.0.0.1:4040
Forwarding           https://dc0b-125-229-248-241.ngrok-free.app -> http://localhost:80

Connections          ttl    opn    rtl    rt5    p50    p90
                    5      0      0.00   0.00   2.00   2.01

HTTP Requests
-----
POST /callback
POST /callback
POST /callback
POST /callback      200 OK
POST /callback      200 OK
```

成果展示





報告結束