

# 初始化 LineBot 專案

## 學習目標：

- 了解 Line Bot 聊天機器人專案所需使用的設定
- 了解每項開發工具的配合項目與使用方式

## 開始建立 Line Bot 專案

1. 在 Workspace 目錄下，建立一個新的專案資料夾：LineBot
2. 在 LineBot 資料匣下，先行建立 Python 主要程式檔：main.py

```
import os
from flask import Flask, request, abort
from linebot import LineBotApi, WebhookHandler
from linebot.exceptions import InvalidSignatureError

app = Flask(__name__)

# Line 聊天機器人的基本資料
line_bot_api = LineBotApi('Channel access token')
handler = WebhookHandler('Channel secret')

# 接收 line 平台送來的「通知」
@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)

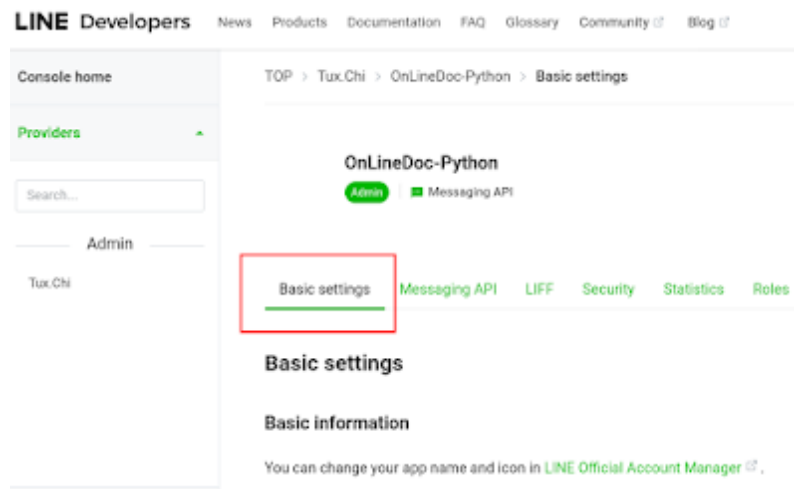
    try:
        handler.handle(body, signature)
    except InvalidSignatureError:
        abort(400)

    return 'OK'

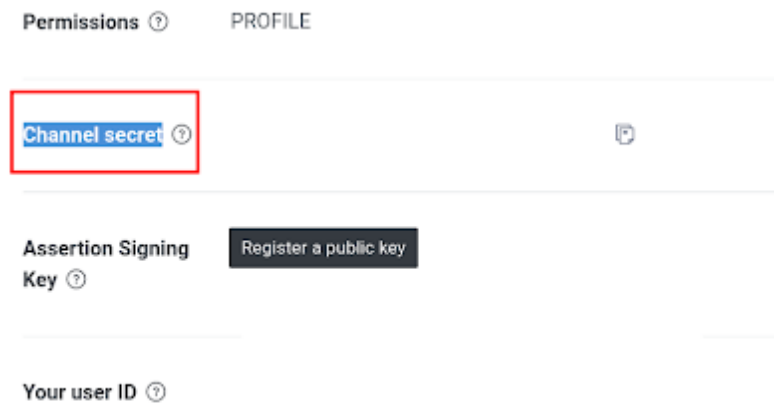
if __name__ == "__main__":
    app.run()
```

PS:先別急著執行測試該檔案

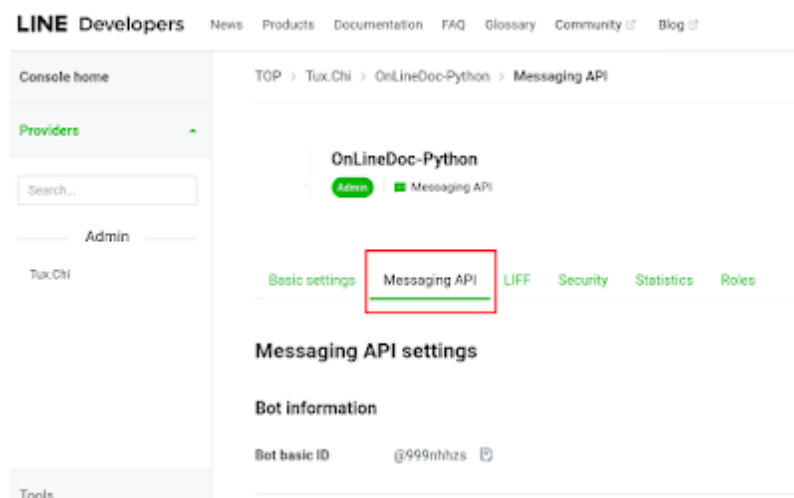
3. 到 Line Developers 申請 access token 以及 secret key:
  - 選擇 Basic setting



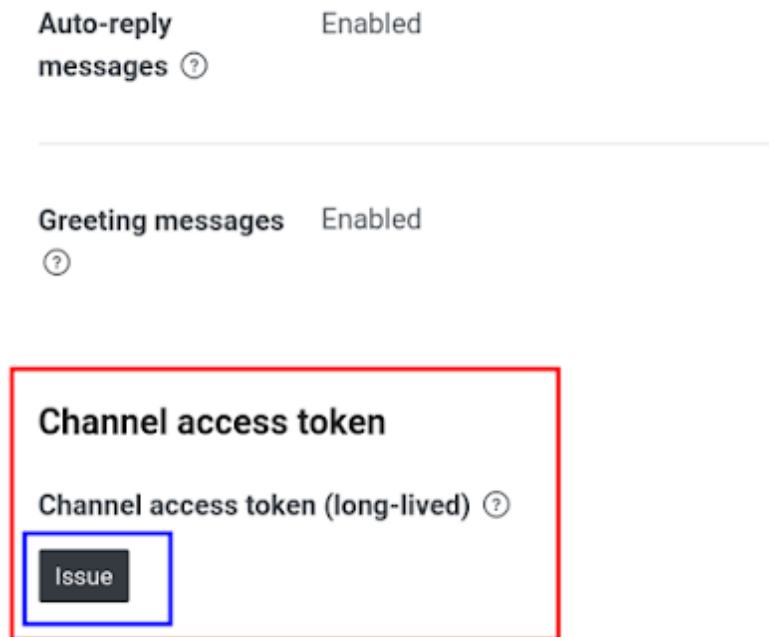
- 下接網頁至底，可以看見 Channel secret 項目！將資料複製下來，貼至程式 main.py 中，取代 Channel secret 字樣！



- 選擇 Messaging API :



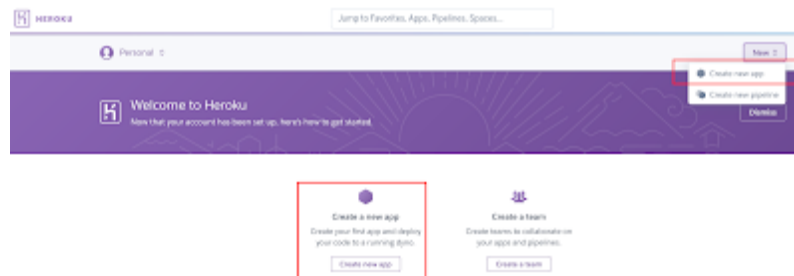
- 下接頁面至底，可以看到 Channel access token! 按下 issue，可以產生 access token !!



複製 token 後，取代程式 main.py 內的 Channel access token 字樣！


#### 4. 在 Heroku 中建立專案！

- 登入 Heroku 首頁中，選擇「Create new app」





- 5. 自行輸入專案名稱，只要是出現綠色框，表示 Heroku 同意建立！

App name

   
**mylinebottux is available**

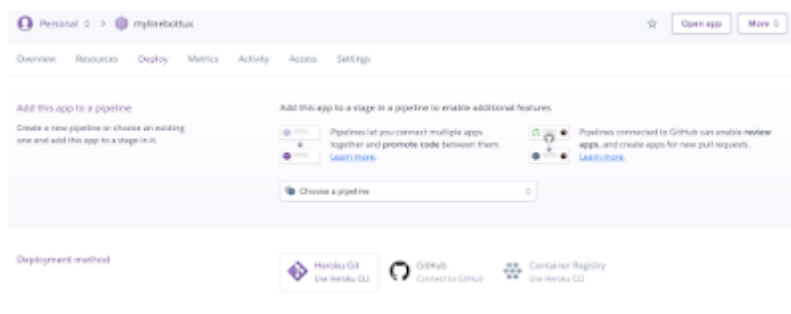
Choose a region

 United States 

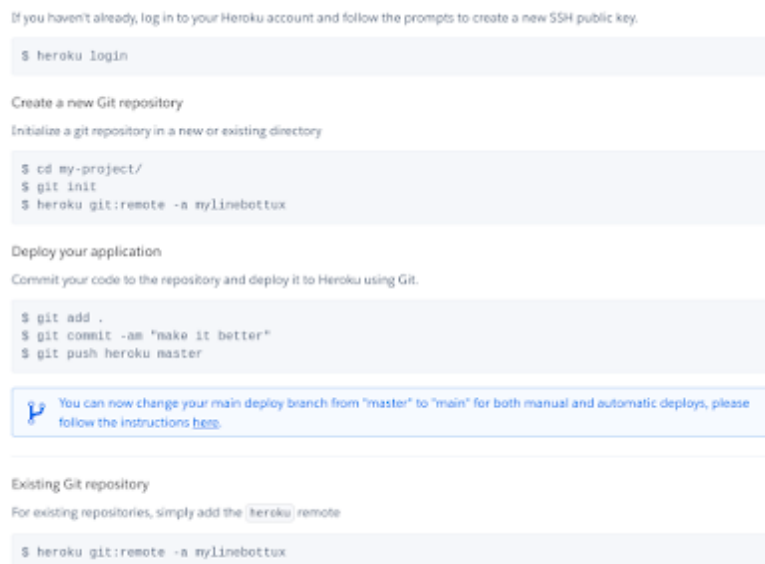
[Add to pipeline...](#)

[Create app](#)

6. 建立的主畫面如下：



7. 在主畫面往下拉，可看見 Heroku cli 的 git 教學



PS:等一下將會使用到該教學

8. 在專案目錄底下建立三個檔案，分別為 Procfile、requirements.txt、runtime.txt

- Procfile 檔案內容

```
web: gunicorn main:app --preload
```

- requirements.txt

```
Flask>=1.1.1  
gunicorn>=20.0.4  
line-bot-sdk>=1.15.0
```

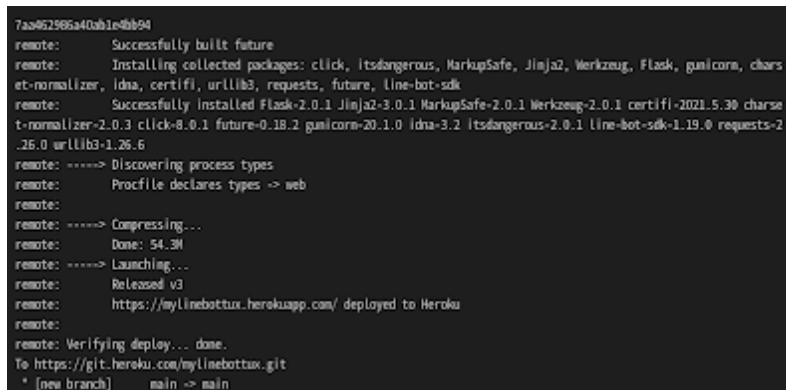
- runtime.txt

```
python-3.9.6
```

## 9. 將工作目錄切換至專案目錄內，將檔案推送至 Heroku 的專案內！

```
heroku login  
git init  
heroku git:remote -a 你的app名稱  
git add .  
git commit -m "Add new App"  
git branch  
git checkout -b main  
git branch -D master  
git push heroku main
```

PS:出現以下畫面，就算成功了！



```
7aa962986a40ab1e4bb94  
remote:      Successfully built future  
remote:      Installing collected packages: click, itsdangerous, MarkupSafe, Jinja2, Werkzeug, Flask, gunicorn, charse  
et-normalizer, idna, certifi, urllib3, requests, future, line-bot-sdk  
remote:      Successfully installed Flask-2.0.1 Jinja2-3.0.1 MarkupSafe-2.0.1 Werkzeug-2.0.1 certifi-2021.5.30 charse  
t-normalizer-2.0.3 click-8.0.1 future-0.18.2 gunicorn-20.1.0 idna-3.2 itsdangerous-2.0.1 line-bot-sdk-1.19.0 requests-2  
.26.0 urllib3-1.26.6  
remote: ----> Discovering process types  
remote:      Procfile declares types => web  
remote: ----> Compressing...  
remote:      Done: 54.3M  
remote: ----> Launching...  
remote:      Released v3  
remote:      https://mylinebottux.herokuapp.com/ deployed to Heroku  
remote:      Verifying deploy... done.  
To https://git.heroku.com/mylinebottux.git  
* [new branch]    main -> main
```

## 10. 設定 Line Developers 的 Webhook：

- 在 Line Developers 的 Messaging API 中，下拉網頁至 Webhook site

TOP > Tux.Chi > OnLineDoc-Python > Messaging API

Available APIs ⓘ

- REPLY\_MESSAGE
- PUSH\_MESSAGE

### Webhook settings

Webhook URL ⓘ

Edit

### LINE Official Account features

Edit the message text and other settings for these features in the LINE Official Account Manager

Allow bot to join  
group chats ⓘ

Disabled

- 在 Webhook URL 的欄位中，輸入 `https://你的app名稱.herokuapp.com/callback`

### Webhook settings

Webhook URL ⓘ

`https://mylinebottux.herokuapp.com/callback`

- ✓ Don't leave this empty
- ✓ Enter a valid HTTPS URL
- ✓ Enter no more than 500 characters

Update

Cancel

- 按下「Verify」可以驗證結果，並且要記得推開 Use webhook 的功能

### Webhook settings

Webhook URL ⓘ

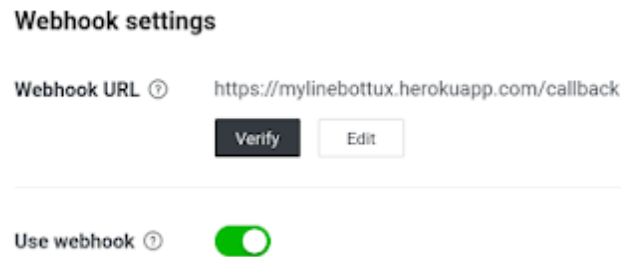
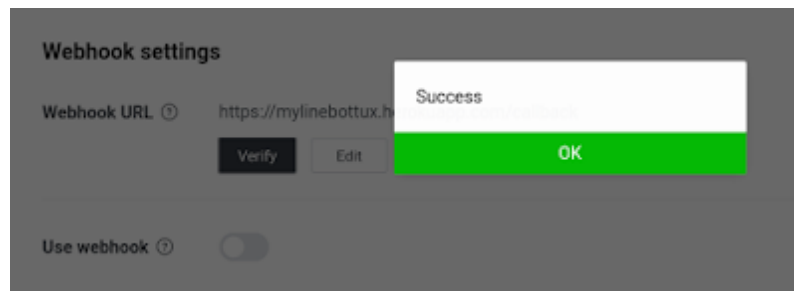
`https://mylinebottux.herokuapp.com/callback`

Verify

Edit

Use webhook ⓘ





11. 修改 main.py 檔案後，推送至 Heroku 專案，即可進行測試

```
import os
from typing import Text
from flask import Flask, request, abort
from linebot import LineBotApi, WebhookHandler
from linebot.exceptions import InvalidSignatureError

from linebot.models import MessageEvent, TextMessage, TextSendMessage

app = Flask(__name__)

# Line 聊天機器人的基本資料
line_bot_api = LineBotApi('Channel access token')
handler = WebhookHandler('Channel secret')

# 接收 line 平台送來的「通知」
@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'

# 鸚鵡學說話
@handler.add(MessageEvent, message=TextMessage)
def echo(event):
```

```
# user id 在 Line Developers / Basic Setting 下
if event.source.user_id == "user id":
    line_bot_api.reply_message(
        event.reply_token,
        TextSendMessage(text=event.message.text)
    )

if __name__ == "__main__":
    app.run()
```

參考文獻：

- [\[Python+LINE Bot教學\]快速掌握LINE Bot部署至Heroku雲端平台的重點](#)
- [Heroku + Flask + Python應用\(2\)：VSCode建置Line Bot，上傳Heroku佈署](#)
- [LINE Messaging API SDK for Python](#)
- [《Line Bot教學》用 Heroku、Python 建立自己的 Line Bot](#)
- [\[Python Linebot\]教你如何使用Python成功串接Linebot\(2020版\)](#)