

LINE Bot (聊天機器人)

- LINE Bot 和 LINE Notify 比較
 - 將 LINE Notify 加入群組後可**單向被動**接受訊息
 - 把 LINE Bot 加入好友可以**主動請求**訊息 (雙向溝通)
- 實現 (參考 <https://ithelp.ithome.com.tw/articles/10229943> 寫得很詳盡)
 - 建立 LINE Developers 帳號 (<https://developers.line.biz/>)
 - 新增專案並獲得 **Channel secret** 和 **Channel access token** 這兩個證照給網頁應用程式使用
 - 在威騰 98 號 NAS 上使用 Python 程式語言開發
 - **flask** 網頁框架 和 **line-bot-sdk** 開發模組
 - 網頁應用必須支援 https 協定並連上 internet (該網頁應用只能處理 http 協定，無法通過 LINE 認證)
 - 透過 **ngrok.com** 網站提供的 http 轉 https 服務只有兩小時期限，**localtunnel** 似乎也會斷線
 - 將程式移植到公有雲 **Heroku** 後似乎可長時間運轉 (Heroku 平台上跑的程式直接支援 https 協定)

測試對話



LINE
伺服器

https



NGROK
伺服器

http



WT98
伺服器

轉換協定
連上 internet

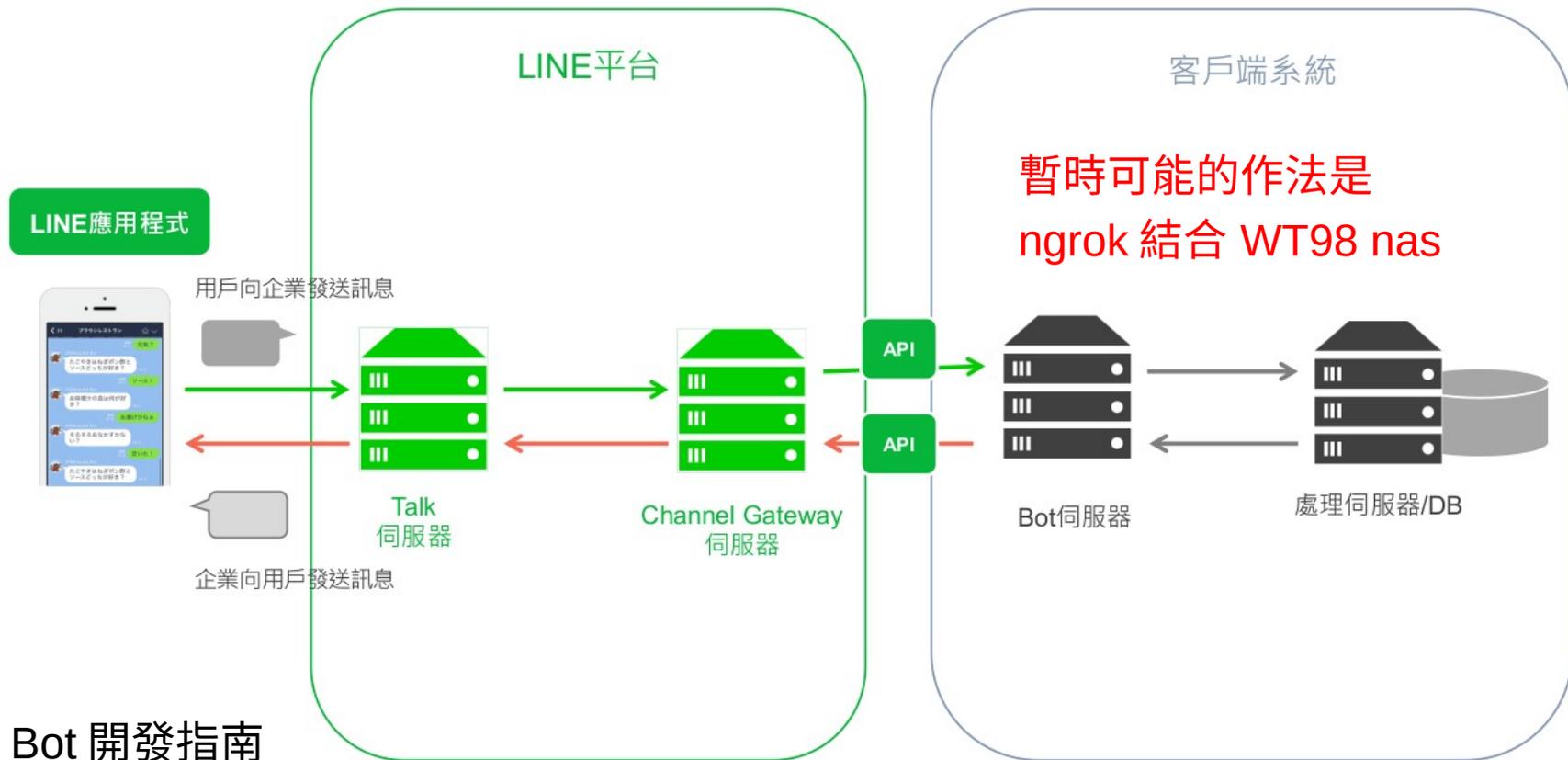
開發程式

或

取代 ngrok 和 wt98，改在 Heruko 公有雲上面跑程式

LINE Bot的機制

LINE Bot是一種透過Channel Gateway伺服器，在客戶端系統的Bot伺服器與LINE應用程式之間，發送和接收資訊的機制。使用發送和接收JSON格式數據的API，來發出請求。



程式

```
line_bot_0.py - /home/pydoc/python3_scripts/line_bot/line_bot_0.py (3.8.5) - [X]
File Edit Format Run Options Window Help
1 from __future__ import unicode_literals
2 import os
3 from flask import Flask, request, abort
4 from linebot import LineBotApi, WebhookHandler
5 from linebot.exceptions import InvalidSignatureError
6 from linebot.models import MessageEvent, TextMessage, TextSendMessage
7
8 import configparser
9
10 import random
11
12 app = Flask(__name__)
13
14 # LINE 聊天機器人的基本資料
15 config = configparser.ConfigParser()
16 config.read('config.ini')
17
18 line_bot_api = LineBotApi(config.get('line-bot', 'channel_access_token'))
19 handler = WebhookHandler(config.get('line-bot', 'channel_secret'))
20
21 # 接收 LINE 的資訊 給 LINE Bot 設定的部份網址 (不含 domain name)
22 @app.route("/callback", methods=['POST'])
23 def callback():
24     signature = request.headers['X-Line-Signature']
25
26     body = request.get_data(as_text=True)
27     app.logger.info("Request body: " + body)
28 |
```

```
29     try:
30         print(body, signature)
31         handler.handle(body, signature)
32
33     except InvalidSignatureError:
34         abort(400)
35
36     return 'OK'
37
38 @handler.add(MessageEvent, message=TextMessage) 設定機器人的對答
39 def handle_message(event):
40
41     if event.message.text == '1':
42
43         #line_bot_api.reply_message(
44             #event.reply_token,
45             #TextSendMessage(text=event.message.text))
46
47         line_bot_api.reply_message(
48             event.reply_token,
49             TextSendMessage(text='中央氣象局海象監測資料'))
50
51     elif event.message.text == '2':
52
53         line_bot_api.reply_message(
54             event.reply_token,
55             TextSendMessage(text='臺灣環境資訊網定點歷線圖'))
56     else:
57
58         line_bot_api.reply_message(
59             event.reply_token,
60             TextSendMessage(text='請輸入1或2 ?'))
61
62 if __name__ == "__main__":
63     app.run(host='0.0.0.0', port=5556)
64
```

透過容器執行 ngrok

test_ngrok

概況 程序 日誌 終端機

2A. 透過下面一道指令執行 ngrok

```
sudo docker run -d -it --name test_ngrok wernight/ngrok ngrok http http://60.249.144.60:5556
```

test_ngrok

```
(venv_linebot) Administrator@MTGroup:~/python3_scripts/line_bot$ export FLASK_APP=line_bot.py
(venv_linebot) Administrator@MTGroup:~/python3_scripts/line_bot$ flask run --host 0.0.0.0 --port 5556
* Serving Flask app 'line_bot.py' (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: off
* Running on all addresses.
  WARNING: This is a development server. Do not use it in a production deployment.
* Running on http://60.249.144.60:5556/ (Press CTRL+C to quit)
180.218.44.227 - - [01/Jun/2021 21:23:30] "GET / HTTP/1.1" 404 -
180.218.44.227 - - [01/Jun/2021 21:23:30] "GET /Favicon.ico HTTP/1.1" 404 -
180.218.44.227 - - [01/Jun/2021 21:23:42] "GET / HTTP/1.1" 404 -
```

1 網頁應用

ngrok by @inconshevable

```
Session Status      online
Session Expires    36 minutes
Update              update available (version 2.3.40, Ctrl-U to update)
Version             2.3.35
Region              United States (us)
Web Interface       http://0.0.0.0:4040
Forwarding          http://99abbdbef40b.ngrok.io -> http://60.249.144.60:5556
Forwarding          https://99abbdbef40b.ngrok.io -> http://60.249.144.60:5556
```

ngrok 提供的 https 網址

| Connections | tll | opn | rt1 | rt5 | p50 | p90 |
|-------------|-----|-----|------|------|------|------|
| | 14 | 0 | 0.00 | 0.00 | 0.13 | 0.16 |

網頁應用的位址

Webhook settings

3. 登入 LINE Developers 設定 ngrok 提供的網址並驗證

Webhook URL

Verify Edit

Use webhook

```
POST /callback 200 OK
```

2B. 從容器的終端機畫面看到 ngrok 執行結果

ngrok 的替代方案，localtunnel

1. 安裝 localtunnel

```
Administrator@WTGroup:~$ sudo npm install -g localtunnel
Password:
/usr/local/bin/lt -> /usr/local/lib/node_modules/localtunnel/bin/lt.js
+ localtunnel@2.0.1
added 22 packages from 22 contributors in 3.966s
Administrator@WTGroup:~$ lt --port 5556
your url is: https://odd-mouse-43.localtunnel.com
```

2. 指定轉接埠 5556

3. 獲得 localtunnel 提供的 https 網址

在 Heroku 公有雲上佈置程式 (1)

- 參考 <https://cruelshare.com/line-bot-second/>
- 使用 Heroku 前需要先安裝 (sudo apt install git) 並設定 git
- Heroku 使用流程
 - 帳號申請
 - 連上 heroku 網頁新建專案 (例如取名為 two-way-linebot)
 - 需要額外三個檔案來告訴專案如何執行

在 Heroku 公有雲上佈置程式 (2)

```
pydoc@pydoc:~/git_projects/heroku_line_bot$ tree
.
├── config.ini
├── line_bot_0.py
├── Procfile
├── requirements.txt
└── runtime.txt
```

1. 網頁程式的架構

0 directories, 5 files 多出 3 個檔案來指導專案如何執行

```
pydoc@pydoc:~/git_projects/heroku_line_bot$ cat Procfile
web: gunicorn line_bot_0:app -preload
```

```
pydoc@pydoc:~/git_projects/heroku_line_bot$ cat requirements.txt
Flask==1.1.1
gunicorn==19.9.0
line-bot-sdk==1.14.0
```

```
pydoc@pydoc:~/git_projects/heroku_line_bot$ cat runtime.txt
python-3.8.5
```

```
pydoc@pydoc:~/git_projects/heroku_line_bot$ git push heroku master
枚舉物件: 10, 完成.
物件計數中: 100% (10/10), 完成.
使用 4 個執行緒進行壓縮
壓縮物件中: 100% (7/7), 完成.
寫入物件中: 100% (10/10), 1.65 KiB | 1.65 MiB/s, 完成.
總共 10 (差異 1), 復用 0 (差異 0)
remote: Compressing source files... done.
remote: Building source:
.:
remote: -----> Compressing...
remote: Done: 52.7M
remote: -----> Launching...
remote: Released v3
remote: https://two-way-linebot.herokuapp.com/ deployed to Heroku
remote:
remote: Verifying deploy... done.
```

3. 佈置過程

最後生成的網址

2. 透過終端機的命令來佈置網頁程式

Install the Heroku CLI

Download and install the [Heroku CLI](#).

If you haven't already, log in to your Heroku account and follow the prompts to create a new SSH public key.

```
$ heroku login
```

Clone the repository

Use Git to clone two-way-linebot's source code to your local machine.

```
$ heroku git:clone -a two-way-linebot ← 沒用到，因為程式存在 local 系統，非透過 github
$ cd two-way-linebot (但稍後多了 heroku git:remote -a two-way-linebot)
```

Deploy your changes

Make some changes to the code you just cloned and deploy them to Heroku using Git.

```
$ git add .
$ git commit -am "make it better"
$ git push heroku master
```



You can now change your main deploy branch from "master" to "main" for both manual and automatic deploys, please follow the [instructions here](#).

加入我來聊天吧

TOP > wt_atrisk > two_way > Messaging API

Bot information

Bot basic ID @379wnwyx 

QR code

