
Line Chatbot實作與部署

李秉鴻

課程大綱

1. 樣品展示
2. 流程演示
3. 下載教材與資料夾架構解析
4. 功能選單製作
5. heroku帳號創建
6. heroku專案建置
7. 本地教材與設定修改
8. 上傳至heroku, 觀察效果
9. 重上傳至heroku
10. 結論

Line Chatbot第一次演示與實作

李秉鴻

下載教材

```
git clone https://github.com/BingHongLi/line_chatbot_quick_template.git
```

```
binghonglis-MacBook-Pro:my_project binghongli$ git clone https://github.com/BingHongLi/line_chatbot_quick_template.git
Cloning into 'line_chatbot_quick_template'...
remote: Enumerating objects: 81, done.
remote: Counting objects: 100% (81/81), done.
remote: Compressing objects: 100% (44/44), done.
remote: Total 81 (delta 9), reused 81 (delta 9), pack-reused 0
Unpacking objects: 100% (81/81), done.
```

觀看資料夾結構 - ls line_chatbot_quick_template

Procfile - Heroku 需要的執行指令檔案

line_secret_key.template - Line需要的安全碼

app.py - 主程式邏輯

進入教材資料夾，並複製修改line_secret_key.template

```
cd line_chatbot_quick_template  
cp line_secret_key.template line_secret_key
```

```
[binghonglis-MacBook-Pro:my_project binghongli$ cd line_chatbot_quick_template  
[binghonglis-MacBook-Pro:line_chatbot_quick_template binghongli$ cp line_secret_key.template line_secret_key  
[binghonglis-MacBook-Pro:line_chatbot_quick_template binghongli$
```

觀看line bot designer 範本 - 文字回應的內容

觀察到用戶輸入了

想知道AI的戰略價值



觀察專案資料夾內的素材資料夾

發現也有個資料夾叫做

想知道AI的戰略價值

▼ 素材	Today at 1:07 AM	--	Folder
▶ rich_menu_0	Today at 9:23 AM	--	Folder
▶ rich_menu_1	Today at 12:27 PM	--	Folder
▶ rich_menu_2	Today at 12:31 PM	--	Folder
▶ rich_menu_3	Today at 12:31 PM	--	Folder
▶ 以假混真玉瘋巔	Today at 12:44 AM	--	Folder
▶ 冷香熱藏	Today at 1:06 AM	--	Folder
▶ 切換分頁	Today at 12:56 AM	--	Folder
▶ 匆忙幾度	Today at 12:59 AM	--	Folder
▶ 因果交織	Today at 12:58 AM	--	Folder
▶ 埋金掛玉	Today at 1:06 AM	--	Folder
▶ 天地靈石	Today at 12:57 AM	--	Folder
▶ 情切切良宵花語	Today at 12:45 AM	--	Folder
▶ 想知道AI的戰略價值	Today at 1:43 AM	--	Folder
▶ 想知道區塊鏈的戰略價值	Today at 12:50 AM	--	Folder
▶ 想知道雲服務的戰略價值	Today at 12:51 AM	--	Folder
▶ 我要看更多機器人成功案例	Today at 12:53 AM	--	Folder
▶ 才人贊善	Today at 1:08 AM	--	Folder
▶ 才德兩全	Today at 1:05 AM	--	Folder
▶ 收到反饋	Today at 12:49 AM	--	Folder

登入Line Messaging API



line messaging api



Google 搜尋

好手氣

line messaging api



全部

圖片

新聞

影片

地圖

更多

設定

工具

約有 35,000,000 項結果 (搜尋時間：0.86 秒)

【功能介紹】Messaging API：LINE@生活圈台灣官方Blog

at-blog.line.me/tw/messaging_api_intro ▼

2017年3月27日 - 本篇將介紹LINE於2016發表的全新Messaging API功能，點選以下連結可快速查詢該項目Messaging API是什麼？Messaging API三大特色如何 ...

【功能介紹】全新申請具有API ... · 現有LINE@帳號開啟API功能

Messaging API - LINE Developers

<https://developers.line.biz/messaging-api/overview> ▼ 翻譯這個網頁

Use the **Messaging API** to build bots that provide personalized experiences for your users on **LINE**.

You associate your bot with a channel on the **LINE** Platform.

[Messaging API SDKs](#) · [Getting started](#) · [Group chats](#)

您曾多次瀏覽這個網頁。上次瀏覽日期：2019/5/1

貼回Line Console的 channel_secret 與 channel_access_token

App icon



App name

李秉鴻-你的職涯好夥伴

App description

提供商業、資訊技術類的資訊整理

Edit

Channel ID ⓘ

1570680260

Channel secret ⓘ

Issue

Messaging settings

Channel access token (long-lived) ⓘ

Issue

```
vim line_secret_key
```

啟用伺服器, 取得連結

```
sh start.sh
```

```
[binghongli-MacBook-Pro:line_chatbot_quick_template binghongli$ sh start.sh
```

```
+ USERNAME=binghongli
```

```
+ IMAGE=line_bot_python
```

```
++ cat VERSION
```

```
+ version=0.0.1
```

```
++ docker ps -a
```

```
++ grep line-chat-bot-jupyter
```

```
++ wc -l
```

```
+ '[' 1 = 0 ']'
```

```
++ docker port line-ngrok 4040
```

```
+ curl 0.0.0.0:32772/api/tunnels
```

% Total	% Received	% Xferd	Average Dload	Average Upload	Speed	Time Total	Time Spent	Time Left	Current Speed
100	779	100	779	0	0	347k	0	--:--:--	380k

```
++ pwd
```

```
+ docker run -v /Users/binghongli/my_project/line_chatbot_quick_template/tunnels.json:/tmp/tunnels.json --rm realguess/jupyterline-chat-bot-jupyter /tmp/tunnels.json
```

```
"https://a8a6c973.ngrok.io"
```

將連結貼回Line Webhook url

Webhook URL Requires SSL ?

—

Edit

Webhook URL Requires SSL ?

https://

a8a6c973.ngrok.io

Max: 500 characters

Please enter a URL that supports HTTPS.

Update

Cancel

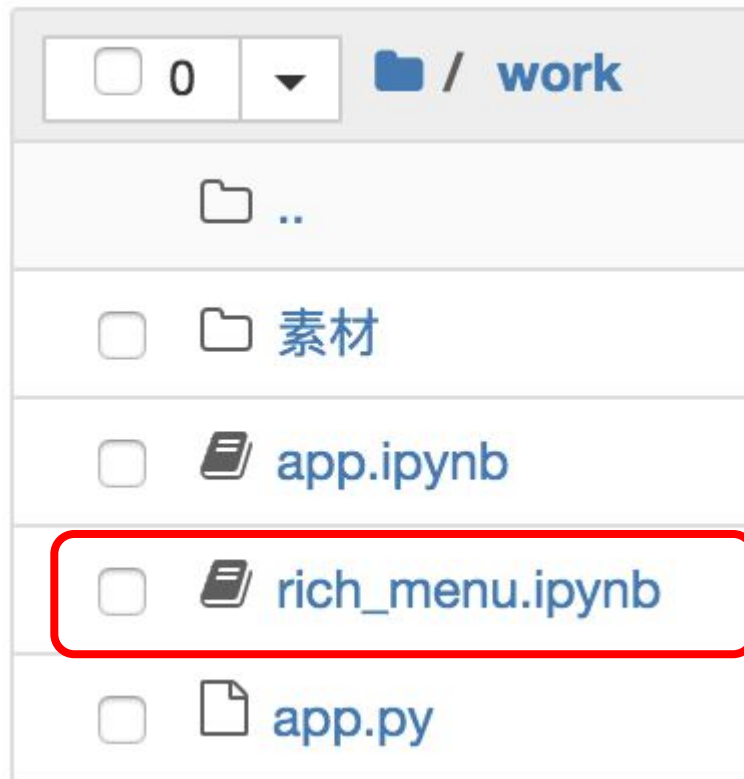
Webhook URL Requires SSL ?

https://a8a6c973.ngrok.io



製作功能選單，並上傳至Line

點擊rich_menu.ipynb



執行第一個資料框，進行選單製作

迴圈讀取本地列表，
上傳設定檔，取得id，並將id寫入檔案中，而後上傳圖片

```
'''
迴圈讀取本地列表，
上傳設定檔，取得id，並將id寫入檔案中，而後上傳圖片
'''

import json
# 載入安全設定檔
from linebot import LineBotApi

secretFileContentJson=json.load(open("./line_secret_key", 'r'))
line_bot_api = LineBotApi(secretFileContentJson.get("channel_access_token"))

'''
rich_menu的本地列表
'''
rich_menu_array=['rich_menu_0','rich_menu_1','rich_menu_2','rich_menu_3']

from linebot.models import RichMenu

for rich_menu_name in rich_menu_array:

    # 創建菜單，取得menuId
    lineRichMenuId = line_bot_api.create_rich_menu(rich_menu=RichMenu.new_from_json_dict(json.
    print("-設定檔上傳結果")
    print(lineRichMenuId)

    # id寫入本地端
    f = open("素材/"+rich_menu_name+"/"+rich_menu_id", "w")
    f.write(lineRichMenuId)
    f.close()

    # 上傳照片至該id
    set_image_response=''
    with open("素材/"+rich_menu_name+"/"+rich_menu.jpg', 'rb') as f:
        set_image_response=line_bot_api.set_rich_menu_image(lineRichMenuId, 'image/jpeg', f)

    print("-圖片上傳結果")
    print(set_image_response)
```

```
# 上傳照片至該id
set_image_response=''
with open("素材/"+rich_menu_name+"/"+rich_menu.jpg', 'rb') as f:
    set_image_response=line_bot_api.set_rich_menu_image(lineRichMenuId, 'image/jpeg', f)

print("-圖片上傳結果")
print(set_image_response)
```

```
-設定檔上傳結果
richmenu-50a793d3bcb78155934527e6aaf3df21
-圖片上傳結果
None
-設定檔上傳結果
richmenu-7be967b034d54aa211d838e874b4efb0
-圖片上傳結果
None
-設定檔上傳結果
richmenu-9fc4bd0dc686e5ba3e0c41d13500c190
-圖片上傳結果
None
-設定檔上傳結果
richmenu-d45139111147c49ea05be0e6465adcb4
-圖片上傳結果
None
```

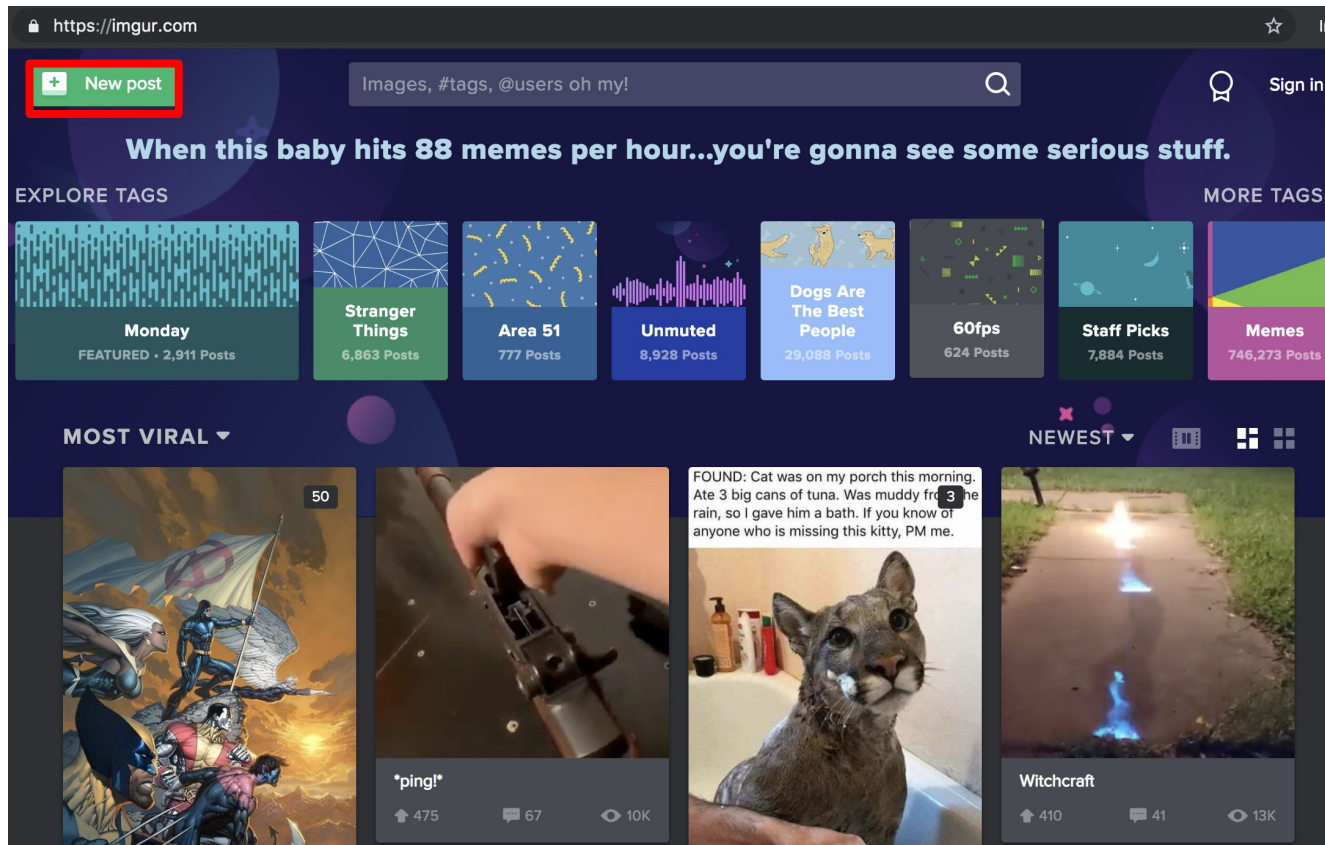
修改素材->關注->reply.json

有引用一張照片，將其連結改成現有連結位置

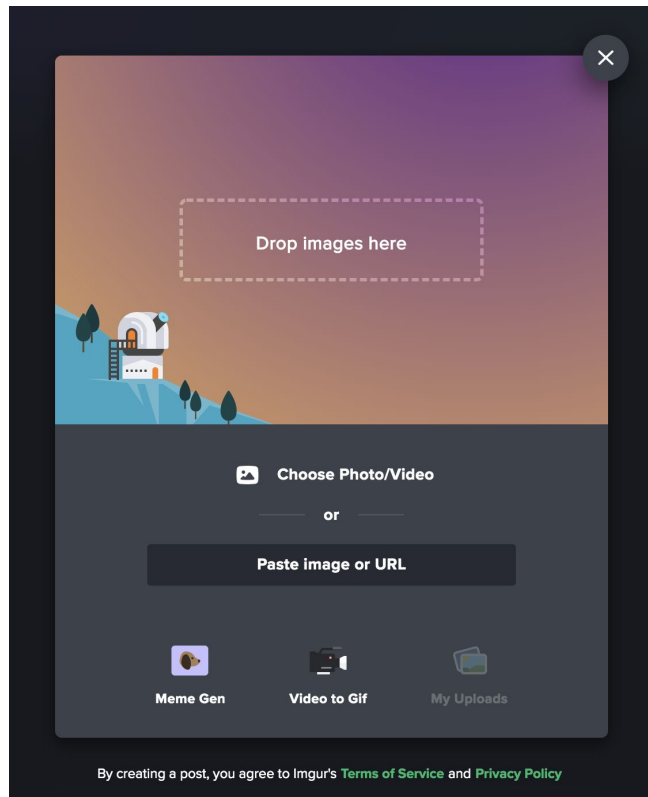


```
[
{
  "type": "text",
  "text": "其實也就是個定位在官方宣傳的內容機器人，但也不想顯得太無趣，
},
{
  "type": "image",
  "originalContentUrl": "PROVIDE_URL_FROM_YOUR_SERVER",
  "previewImageUrl": "PROVIDE_URL_FROM_YOUR_SERVER",
  "animated": true
},
}
```

連線至<https://imgur.com> 點擊左上角New post



圖片放到imgur空間，並生成連結



把連結貼回reply.json中

```
{  
  "type": "text",  
  "text": "其實也就是個定位在官方宣傳的內容機器人，但也不想顯得太無趣，耗了大家  
,  
{  
  "type": "image",  
  "originalContentUrl": "https://imgur.com/Vqlv8CB.jpg",  
  "previewImageUrl": "https://imgur.com/Vqlv8CB.jpg",  
  "animated": true  
,  
}
```

關注後，觀察效果

圖片出來了

未來可先把照片放到外部網站





把Line Chatbot放到長期免費空間

李秉鴻

搜尋Heroku



Heroku



全部

圖片

影片

新聞

地圖

更多

設定

工具

約有 7,120,000 項結果 (搜尋時間：0.32 秒)

Heroku: Cloud Application Platform

<https://www.heroku.com/> ▼ 翻譯這個網頁

Heroku is a platform as a service (PaaS) that enables developers to build, run, and operate applications entirely in the cloud.

[Pricing](#) · [Heroku platform](#) · [Products](#) · [Go at Heroku](#)

您曾多次瀏覽這個網頁。上次瀏覽日期：2019/3/26

登陸或創建新帳號

**HEROKU**

Products ▾

Marketplace ▾

Pricing

Documentation

Support

More ▾



Log in or

Sign up



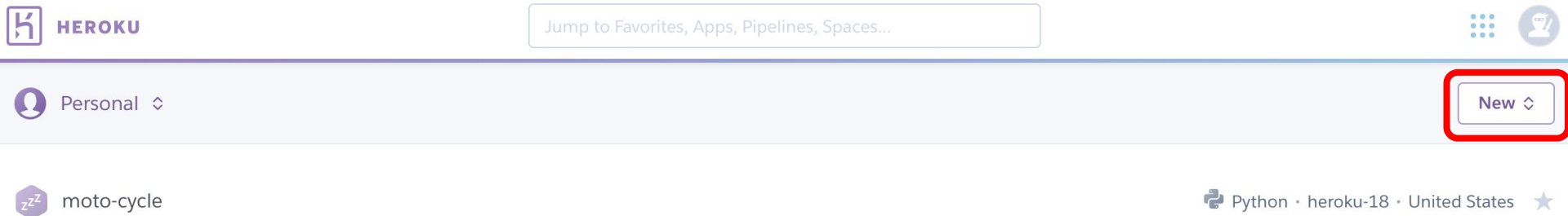
DEVELOPERS

Focus on your apps

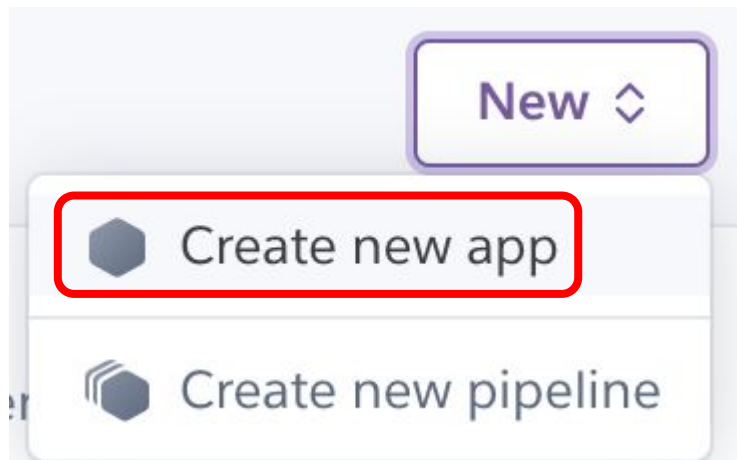
Invest in apps, not ops. Heroku handles the hard stuff — patching and upgrading, 24/7 ops and security, build systems, failovers, and more — so your developers can stay focused on building great apps.

[SIGN UP FOR FREE](#)[Explore the Heroku Platform](#)

介面一覽，點選New創建新的專案



點擊Create new app, 並輸入app name, 並create



Create New App

App name

personal-intro

personal-intro is available

Choose a region

United States

Add to pipeline...

Create app

創建後的介面

 Personal  >  personal-intro

[Open app](#)[More !\[\]\(e3f8612927870f2e0f9f5989e6dd3064_img.jpg\)](#)[Overview](#)[Resources](#)[Deploy](#)[Metrics](#)[Activity](#)[Access](#)[Settings](#)

Add this app to a pipeline

Create a new pipeline or choose an existing one and add this app to a stage in it.

Add this app to a stage in a pipeline to enable additional features



Pipelines let you connect multiple apps together and **promote code** between them. [Learn more.](#)



Pipelines connected to GitHub can enable **review apps**, and create apps for new pull requests. [Learn more.](#)

 Choose a pipeline 

Deployment method



Heroku Git
Use Heroku CLI



GitHub
Connect to GitHub



Container Registry
Use Heroku CLI

取得heroku給予的url位置



Personal



personal-intro

[Overview](#)[Resources](#)[Deploy](#)[Metrics](#)[Activity](#)[Access](#)[Settings](#)

點擊setting後，往下滑至Domain處

Buildpacks

Buildpacks are scripts that are run when your app is deployed. They are used to install dependencies for your app and configure your environment. [Find new buildpacks on Heroku Elements](#)

Buildpacks will appear here

Buildpacks are used to install dependencies for your app and configure your environment

Domains and certificates

Add your custom domains here then [point your DNS to Heroku](#).

Domain Your app can be found at <https://personal-intro.herokuapp.com/>

SSL

Upgrade to paid dynos to configure Heroku SSL

安裝heroku cli

Deployment method



Heroku Git
Use Heroku CLI



GitHub
Connect to GitHub



Container Registry
Use Heroku CLI

Deploy using Heroku Git

Use git in the command line or a GUI tool to deploy this app.

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.

選擇合適的作業系統進行安裝

Download and install



The Heroku CLI requires **Git**, the popular version control system. If you don't already have Git installed, complete the following before installing the CLI:

- [Git installation](#)
- [First-time Git setup](#)

macOS

Download the installer

Also available via Homebrew:

```
$ brew tap heroku/brew && brew install heroku
```

Windows

Download the appropriate installer for your Windows installation:

64-bit installer

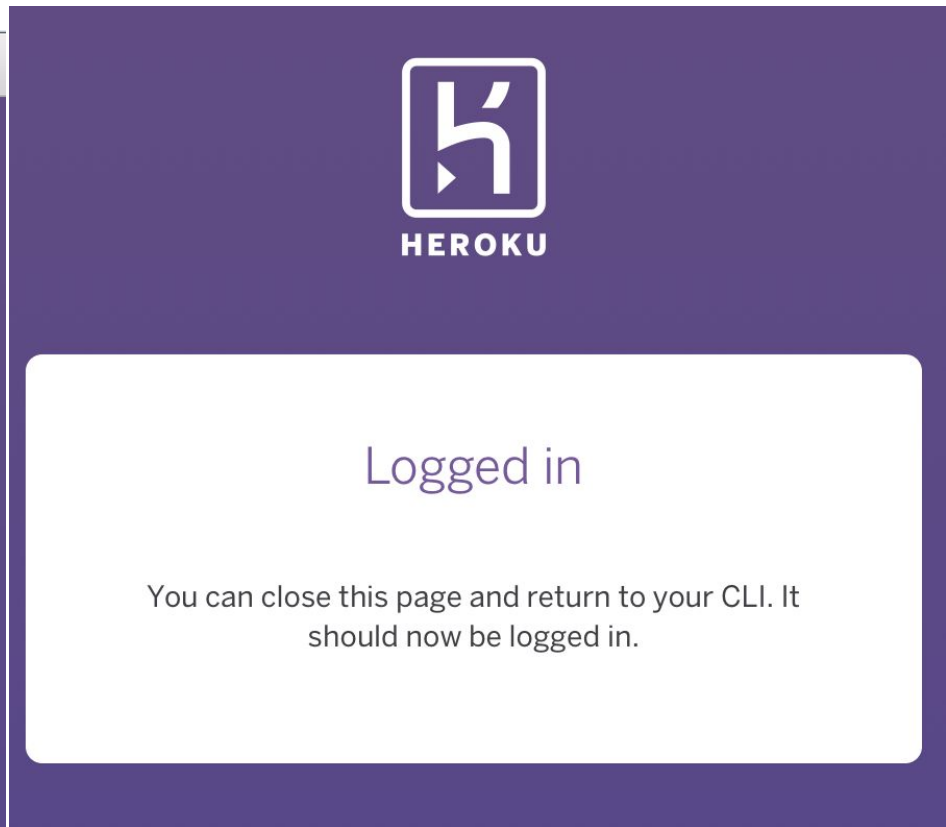
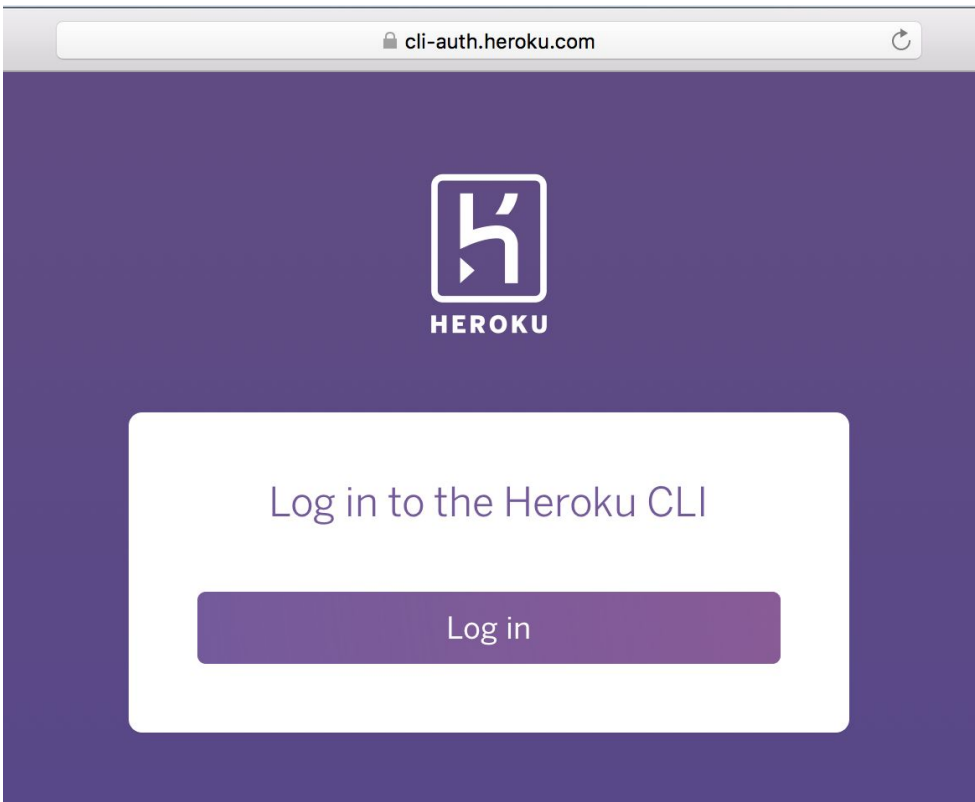
32-bit installer

開啟終端命令列，切換至該資料夾

```
[binghonglis-MacBook-Pro:line_chatbot_quick_template binghongli$ cd ..  
[binghonglis-MacBook-Pro:my_project binghongli$ cd line_chatbot_quick_template
```

登入

```
binghongli-MacBook-Pro:line_chatbot_quick_template binghongli$ heroku login  
heroku: Press any key to open up the browser to login or q to exit: █
```



添加進heroku的repo

git init

heroku git:remote -a {你的app name}

```
[binghonglis-MacBook-Pro:line_chatbot_quick_template binghongli$ git init
Reinitialized existing Git repository in /Users/binghongli/my_project/line_chatbot_quick_template/.git/
[binghonglis-MacBook-Pro:line_chatbot_quick_template binghongli$ heroku git:remote -a personal-intro
[heroku: Press any key to open up the browser to login or q to exit:
Opening browser to https://cli-auth.heroku.com/auth/browser/1a425bf4-7f86-4cf2-b30f-7a38280a635e...
Logging in... done
set git remote heroku to https://git.heroku.com/personal-intro.git
```


添加內容與指定執行環境為Python

```
git add .
```

```
git commit -am "make it better"
```

```
heroku buildpacks:set heroku/python
```

```
[binghonglis-MacBook-Pro:line_chatbot_quick_template binghongli$ git add .  
[binghonglis-MacBook-Pro:line_chatbot_quick_template binghongli$ git commit -am "make it better"  
[master 2cc1908] make it better  
8 files changed, 398 insertions(+), 1262 deletions(-)  
rewrite app.ipynb (81%)
```

```
binghonglis-MacBook-Pro:line_chatbot_quick_template binghongli$ heroku buildpacks:set heroku/python  
Buildpack set. Next release on personal-intro will use heroku/python.  
Run git push heroku master to create a new release using this buildpack.
```

將內容推上Heroku

```
[binghongli-MacBook-Pro:line_chatbot_quick_template binghongli$ git push heroku master  
Counting objects: 83, done.  
Delta compression using up to 4 threads.  
0 flask-1.0.2 future-0.17.1 idna-2.8 itsdangerous-1.1.0 line-bot-sdk-1.8.0 req  
remote:  
remote: ----> Discovering process types  
remote:       Procfile declares types -> web  
remote:  
remote: ----> Compressing...  
remote:       Done: 49.2M  
remote: ----> Launching...  
remote:       Released v3  
remote:       https://personal-intro.herokuapp.com/ deployed to Heroku  
remote:  
remote: Verifying deploy... done.  
To https://git.heroku.com/personal-intro.git  
* [new branch]      master -> master
```

觀看Heroku官網



personal-intro



Python · heroku-18 · United States ★

Latest activity

**b97607065@gmail.com:** Deployed `1e7171b3`

Today at 6:27 PM · v3

**b97607065@gmail.com:** Build succeededToday at 6:26 PM · [View build log](#)

將Line的Webhook改為heroku提供的網址

Webhook URL Requires SSL [?](#)

https://

personal-intro.herokuapp.com/

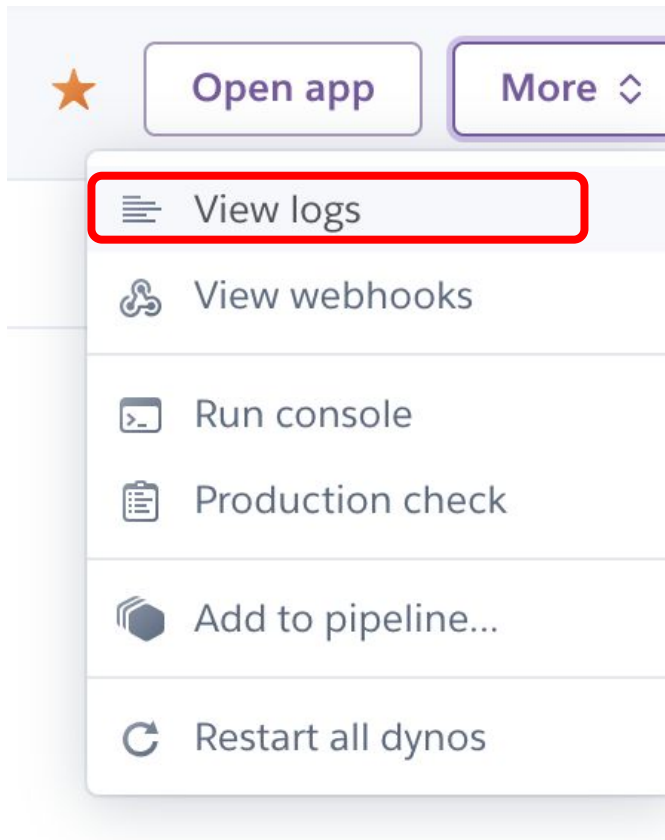
Max: 500 characters

Please enter a URL that supports HTTPS.

Update

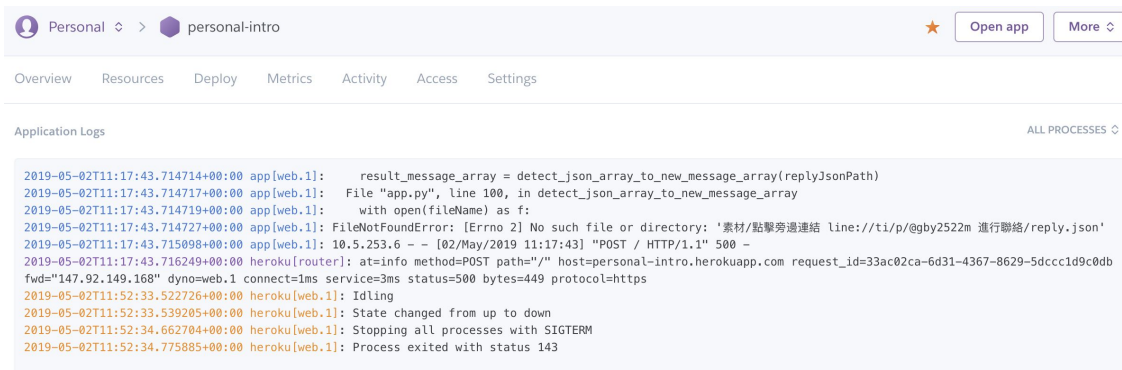
Cancel

觀測log



A screenshot of the Heroku dashboard for an application named 'personal-intro'. The 'View logs' option in the left-hand menu is highlighted with a red rectangle. The menu also includes 'View webhooks', 'Run console', 'Production check', 'Add to pipeline...', and 'Restart all dynos'.

- Open app
- More ▾
- View logs**
- View webhooks
- Run console
- Production check
- Add to pipeline...
- Restart all dynos

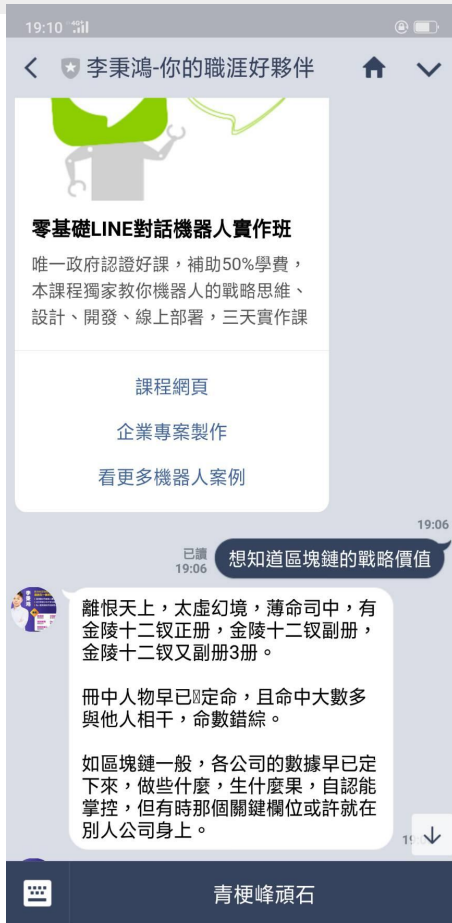


A screenshot of the Heroku Application Logs page for the 'personal-intro' application. The logs show a series of events, including a file not found error and a process exit.

Application Logs ALL PROCESSES ▾

```
2019-05-02T11:17:43.714714+00:00 app[web.1]: result_message_array = detect_json_array_to_new_message_array(replyJsonPath)
2019-05-02T11:17:43.714717+00:00 app[web.1]: File "app.py", line 100, in detect_json_array_to_new_message_array
2019-05-02T11:17:43.714719+00:00 app[web.1]:     with open(fileName) as f:
2019-05-02T11:17:43.714727+00:00 app[web.1]: FileNotFoundError: [Errno 2] No such file or directory: '素材/點擊旁邊連結 line://ti/p/@gby2522m 進行聯絡/reply.json'
2019-05-02T11:17:43.715098+00:00 app[web.1]: 10.5.253.6 - - [02/May/2019 11:17:43] "POST / HTTP/1.1" 500 -
2019-05-02T11:17:43.716249+00:00 heroku[router]: at=info method=POST path="/" host=personal-intro.herokuapp.com request_id=33ac02ca-6d31-4367-8629-5dccc1d9c0db fwd="147.92.149.168" dyno=web.1 connect=1ms service=3ms status=500 bytes=449 protocol=https
2019-05-02T11:52:33.522726+00:00 heroku[web.1]: Idling
2019-05-02T11:52:33.539205+00:00 heroku[web.1]: State changed from up to down
2019-05-02T11:52:34.662704+00:00 heroku[web.1]: Stopping all processes with SIGTERM
2019-05-02T11:52:34.775885+00:00 heroku[web.1]: Process exited with status 143
```

觀看Line是否正常運作





流程思路分析

沉浸式體驗

對話機器人故事線解析

資料夾內容對應解析

子資料夾內容解析

學員實作步驟拆解

下載資料夾

```
git clone https://github.com/BingHongLi/line_chatbot_quick_template.git
```

啟動環境

```
cd line_chatbot_quick_template
```

將自己準備好的內容放入素材資料夾中

rich_menu_0

rich_menu.jpg

rich_menu.json

rich_menu_id

用戶回應字詞

reply.json # 若有圖片引用, 使用imgur外部連結

開啟新的Line Messaging api 帳號

將先前隨機生成的url 貼入，作為webhook url

變更line_secret_key.template

將line_secret_key.template 改為 line_secret_key

並將裡面內容改為Line messaging api 提供內容

啟用app.ipynb 進行測試

啟用app.ipynb 點擊左上方的cell, 選擇run all

掃描Line QR Code, 進行用戶測試

確認無誤，以Windows訪問Heroku網站，會員申請

創建一個 app

將app 的url 貼回自己的圖像素材中

安裝heroku cli

在專案資料夾下執行一系列 heroku的指令

```
heroku login
```

```
git init
```

```
heroku git:remote -a {你的app name}
```

```
git add .
```

```
git commit -m "my teacher is lbh"
```

```
git push heroku master
```

觀察log狀態

在Line上進行測試