



Part III

完善 ChatBot.py

回應基本訊息-建立新的Line Bot

LINE Developers

News Products Documentation FAQ Glossary Community

Console home

TOP > CCLAB > 回應機器人1 > Basic settings

Providers

Search...

Admin

CCLAB



回應機器人1

Admin | Messaging API

Basic settings

Messaging API

LIFF

Security

Statistics

Roles

Basic settings

Basic information

You can change your app name and icon in [LINE Official Account Manager](#).

Channel ID

1655796016

Channel icon



Channel name

回應機器人1

LINE帳號管理:

url: <https://manager.line.biz/>

功能: 建立linebot、圖形化介面功能、基本帳戶功能設定。

LINE Developers:

url: <https://developers.line.biz/>

功能: line開發者設定。

建立新的Line Bot



建立LINE官方帳號

停用自動發送歡迎及回覆訊息

啟用Messaging API

回應基本訊息-建立三方連接(本機網頁建立對外通道)

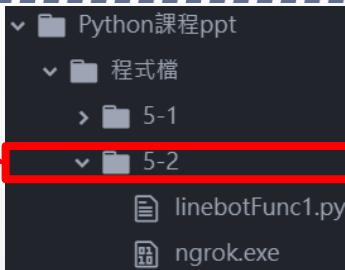
開啟命令提示字元(cmd)

#前往資料夾位置

➤ cd [path to program folder]

#啟動ngrok伺服器(Flask伺服器預設Port為5000)

➤ ngrok http 5000



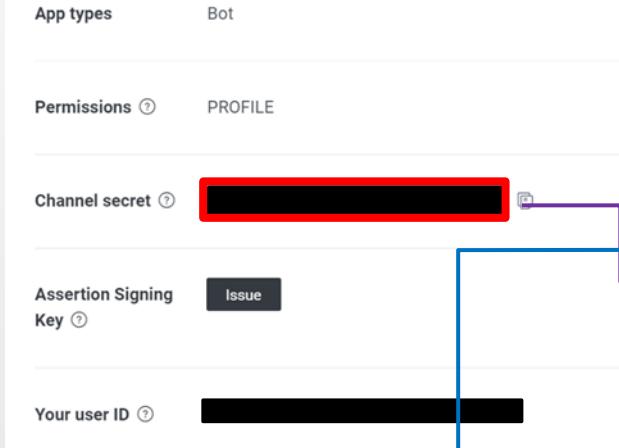
```
命令提示字元 - ngrok http 5000
ngrok by @inconshreveable
Session Status          online
Account                asd357jkl159@gmail.com (Plan: Free)
Version                2.3.35
Region                 United States (us)
Web Interface          http://127.0.0.1:4040
Forwarding             http://72e35f8443cd.ngrok.io -> http://localhost:5000
                           https://72e35f8443cd.ngrok.io -> http://localhost:5000
Connections            ttl     opn      r1      rt5      p50      p90
                           1       0       0.01    0.00    0.31    0.31
HTTP Requests
POST /callback          200 OK
```

- 代理服務器生成的網址是隨機的，每次執行得到不同網址(=>每次執行為Line Bot提供服務的外部網址會不同，需修改Line Bot設定)
- 在Line Bot提供服務期間不可中斷本機伺服器的對外通道，否則本機無法提供服務

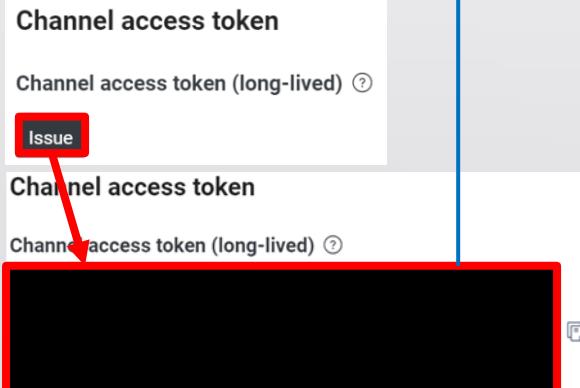
回應基本訊息-建立三方連接(本機處理對應Line Bot資料)1

<https://developers.line.biz/>

TOP > CCLAB > 回應機器人1 > Basic settings



TOP > CCLAB > 回應機器人1 > Messaging API



- 從Line Bot開發介面取得對應的Line Bot資料，修改程式碼指向提供服務的Line Bot對象。

```
1 from flask import Flask  
2 app = Flask(__name__)  
3  
4 from flask import request, abort  
5 from linebot import LineBotApi, WebhookHandler  
6 from linebot.exceptions import InvalidSignatureError  
7 from linebot.models import MessageEvent, TextMessage, TextSendMessage, ImageSendMessage  
8  
9  
10 line_bot_api = LineBotApi('你的 CHANNEL_ACCESS_TOKEN')  
11 handler = WebhookHandler('你的 CHANNEL_SECRET')  
12  
13 @app.route("/callback", methods=['POST'])  
14 def callback():  
15     signature = request.headers['X-Line-Signature']  
16     body = request.get_data(as_text=True)  
17     try:  
18         handler.handle(body, signature)  
19     except InvalidSignatureError:  
20         abort(400)  
21     return 'OK'  
22  
23 @handler.add(MessageEvent, message=TextMessage)  
24 def handle_message(event):  
25  
26 if __name__ == '__main__':  
27     app.run()
```

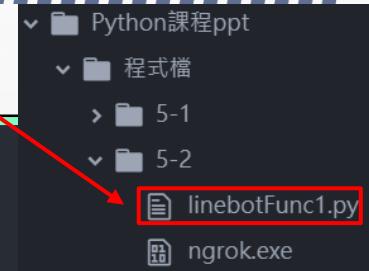
匯入模組

接收對應Line Bot資料

建立callback路由 . Post傳資料

接收使用者訊息，並回復。此處回覆各種基本訊息

呼叫程式時，執行app.run()



回應基本訊息-建立三方連接(本機處理對應Line Bot資料)2

#定義如何處理訊息

```
def handle_message(event):
```

#收到訊息事件觸發(mtext=收到的訊息內容)

```
mtext = event.message.text
```

```
if mtext == '@傳送文字':
```

```
    try:
```

```
        message = TextSendMessage(  
            text = "我是 Linebot , \n您好 ! ")
```

```
    except:
```

```
        line_bot_api.reply_message(event.reply_token,TextSendMessage(text='發生錯誤 ! '))
```

如果收到訊息”@傳送訊息”，則執行以下內容

定義回傳訊息內容及格式

回傳訊息

```
elif mtext == '@傳送圖片':
```

```
    try:
```

```
        message = ImageSendMessage(  
            original_content_url = "https://i.imgur.com/4QfKuz1.png",  
            preview_image_url = "https://i.imgur.com/4QfKuz1.png")
```

```
    line_bot_api.reply_message(event.reply_token,message)
```

```
except:
```

```
    line_bot_api.reply_message(event.re
```

```
elif mtext == '@傳送貼圖':
```

```
    try:
```

```
        message = StickerSendMessage(  
            package_id='1',  
            sticker_id='2')
```

```
    line_bot_api.reply_message(event.reply_token, message)
```

```
except:
```

```
    line_bot_api.reply_message(event.reply_token,TextSendMessage(text='發生錯誤 ! '))
```

STKPKGID : 1 (Moon James)			
STKID	STKPKGID	STKVER	IMG_sample
1	1	100	
2	2	100	

<https://developers.line.biz/en/docs/messaging-api/sticker-list/#sticker-definitions>



回應基本訊息-建立三方連接(本機處理對應Line Bot資料)3

```
elif mtext == '@多項傳送':  
    try:  
        message = [ #串列  
            StickerSendMessage( #傳送貼圖  
                package_id='1',  
                sticker_id='2'  
            ),  
            TextSendMessage( #傳送文字  
                text = "這是 Pizza 圖片！"  
            ),  
            ImageSendMessage( #傳送圖片  
                original_content_url = "https://i.imgur.com/4QfKuz1.png",  
                preview_image_url = "https://i.imgur.com/4QfKuz1.png"  
            )  
        ]  
        line_bot_api.reply_message(event.reply_token,message)  
    except:  
        line_bot_api.reply_message(event.reply_token,TextSendMessage(text='發生錯誤！'))
```

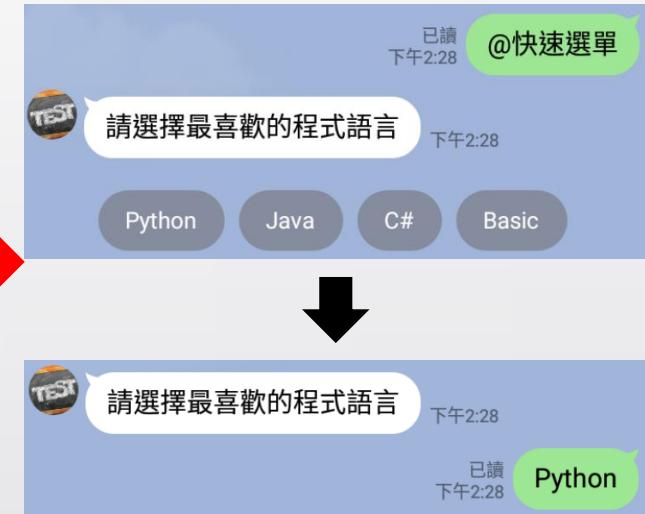


```
elif mtext == '@傳送位置':  
    try:  
        message = LocationSendMessage(  
            title='101大樓',  
            address='台北市信義路五段7號',  
            latitude=25.034207, #緯度  
            longitude=121.564590 #經度  
        )  
        line_bot_api.reply_message(event.reply_token, message)  
    except:  
        line_bot_api.reply_message(event.reply_token,TextSendMessage(text='發生錯誤！'))
```



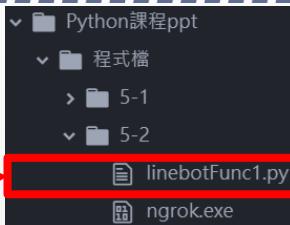
回應基本訊息-建立三方連接(本機處理對應Line Bot資料)4

```
if mtext == '@快速選單':  
    try:  
        message = TextSendMessage(  
            text='請選擇最喜歡的程式語言',  
            quick_reply=QuickReply(  
                items=[  
                    QuickReplyButton(  
                        action=MessageAction(label="Python", text="Python")  
                    ),  
                    QuickReplyButton(  
                        action=MessageAction(label="Java", text="Java")  
                    ),  
                    QuickReplyButton(  
                        action=MessageAction(label="C#", text="C#")  
                    ),  
                    QuickReplyButton(  
                        action=MessageAction(label="Basic", text="Basic")  
                    ),  
                ]  
            )  
        )  
        line_bot_api.reply_message(event.reply_token,message)  
    except:  
        line_bot_api.reply_message(event.reply_token,TextSendMessage(text='發生錯誤！'))
```



回應基本訊息-建立三方連接(本機處理對應Line Bot資料)3

打開Anaconda Prompt (anaconda3)
#使用line_env虛擬環境
➤ conda activate line_env
#執行程式
➤ python [path to .py file]



```
Anaconda Prompt (anaconda3) - python C:\Users\ccllab\Documents\專案助理\Python課程ppt\程式檔\5-2\linebotFunc1.py
(base) C:\Users\ccllab>conda activate line_env
(line_env) C:\Users\ccllab>python C:\Users\ccllab\Documents\專案助理\Python課程ppt\程式檔\5-1\
* Serving Flask app "linebotTest" (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 http://127.0.0.1:5000/ (Press CTRL+C to quit)
127.0.0.1 - - [25/Mar/2021 12:19:51] " [37mPOST /callback HTTP/1.1 [0m" 200 -
(line_env) C:\Users\ccllab>python C:\Users\ccllab\Documents\專案助理\Python課程ppt\程式檔\5-2\linebotFunc1.py
* Serving Flask app "linebotFunc1" (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 http://127.0.0.1:5000/ (Press CTRL+C to quit)
127.0.0.1 - - [25/Mar/2021 14:24:40] " [37mPOST /callback HTTP/1.1 [0m" 200 -
```

● 在Line Bot提供服務期間不可中斷本機伺服器的服務，否則無法給與Line Bot使用者回應。

回應基本訊息-建立三方連接(Line Bot透過代理伺服器連到本機服務)

修改Line Bot的伺服器連結，使Line Bot能夠將使用者訊息傳到本機處理並回應

TOP > CCLAB > 回應機器人1 > Messaging API

Webhook settings

Webhook URL ②

Edit

Webhook settings

Webhook URL ②

https://72e35f8443cd.ngrok.io/callback

Verify

Edit

Use webhook ②



● 可驗證Line Bot可
否透過代理伺服
器連到本機服務

```
命令提示字元 - ngrok http 5000
ngrok by @inconshreveable

Session Status
Account          online      asd357jkl159@gmail.com (Plan: Free)
Version          2.3.35
Region           United States (us)
Web Interface   http://127.0.0.1:4040
Forwarding       https://72e35f8443cd.ngrok.io -> http://localhost:5000
Forwarding       https://72e35f8443cd.ngrok.io -> http://localhost:5000

Connections
Connections      ttl     opn     rtt1    rt5     p50    p90
1                0       0       0.01   0.00   0.31   0.31
```

```
1  from flask import Flask
2  app = Flask(__name__)

3

4  from flask import request, abort
5  from linebot import LineBotApi, WebhookHandler
6  from linebot.exceptions import InvalidSignatureError
7  from linebot.models import MessageEvent, TextMessage, TextSendMessage, ImageSendMessage

8

9  line_bot_api = LineBotApi('你的 CHANNEL_ACCESS_TOKEN')
10 handler = WebhookHandler('你的 CHANNEL_SECRET')

11

12 @app.route('/callback', methods=['POST'])
13 def callback():
14     signature = request.headers['X-Line-Signature']
15     body = request.get_data(as_text=True)
16     try:
17         handler.handle(body, signature)
18     except InvalidSignatureError:
19         abort(400)
20     return 'OK'
21

22 @handler.add(MessageEvent, message=TextMessage)
23 def handle_message(event):
24     if __name__ == '__main__':
25         app.run()
```

回應基本訊息-圖文選單

<http://manager.line.biz/>

LINE Official Account Manager 回應機器人1 @941ecbtg 輕用量 1 回應模式：聊天機器人

主頁 提醒 分析 貼文串 聊天 基本檔案 口袋商店

群發訊息 漸進式訊息 NEW 自動回應訊息

訊息項目 推廣相關

聊天室相關
加入好友的歡迎訊息
圖文選單
資料管理

增加好友人數

圖文選單

您可建立具有視覺效果的互動型選單，於聊天室中提供優惠券、網址。本頁面僅可顯示透過本頁面所建立的圖文選單。

圖片	標題	動作	使用期間
	測試圖文選單	1 • 文字 - @傳送文字	2021/03/25 00:00 - 2021/03/31 23:59
		2 • 文字 - @傳送圖片	
		3 • 文字 - @傳送貼圖	
		4 • 文字 - @多項傳送	
		5 • 文字 - @傳送位置	
		6 • 文字 - @快速選單	

回應基本訊息-功能 Demo



回應多媒體訊息-建立新的Line Bot

developers.line.biz/console/channel/1655796219

LINE Developers News Products Documentation FAQ Glossary Community Blog

Console home TOP > CCLAB > 回應機器人2

Providers Search... Admin

CCLAB Basic settings Messaging API LIFF Security Statistics Roles

回應機器人2 Admin Messaging API

Basic settings

Basic information

You can change your app name and icon in [LINE Official Account Manager](#).

Channel ID 1655796219

Channel icon 

Channel name 回應機器人2

Channel description 回應多媒體訊息

Edit

LINE帳號管理:

url: <https://manager.line.biz/>

功能: 建立linebot、圖形化介面功能、基本帳戶功能設定。

LINE Developers:

url: <https://developers.line.biz/>

功能: line開發者設定。

建立新的Line Bot



建立LINE官方帳號

停用自動發送歡迎及回覆訊息

啟用Messaging API

回應多媒體訊息-建立三方連接(本機網頁建立對外通道)

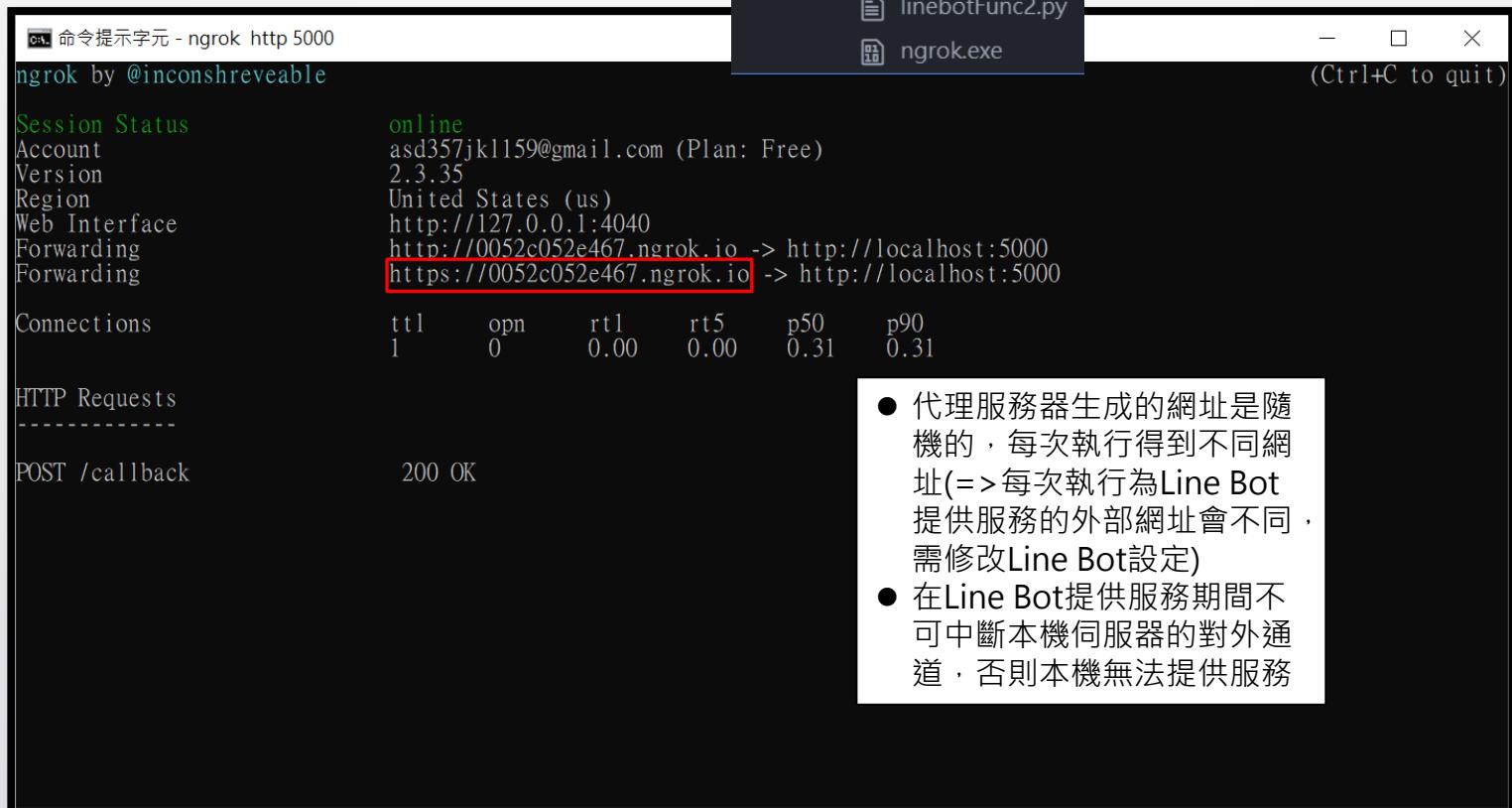
開啟命令提示字元(cmd)

#前往資料夾位置

► cd [path to program folder]

#啟動ngrok伺服器(Flask伺服器預設Port為5000)

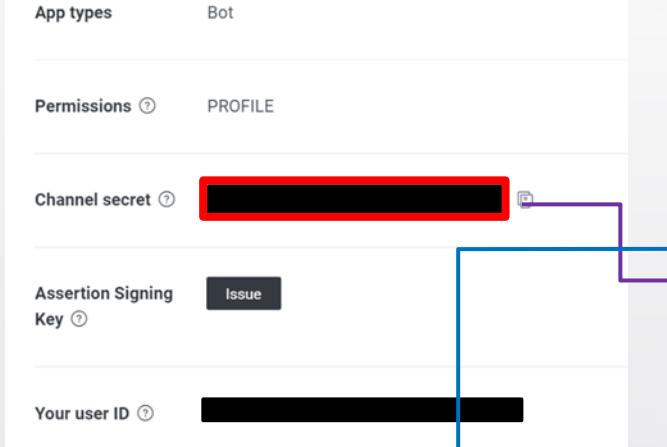
► ngrok http 5000



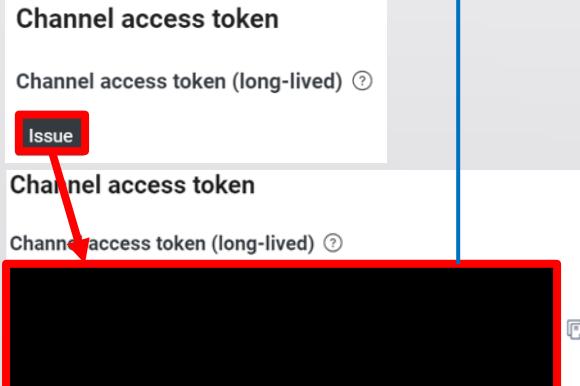
回應多媒體訊息-建立三方連接(本機處理對應Line Bot資料)1

<https://developers.line.biz/>

TOP > CCLAB > 回應機器人2 > Basic settings



TOP > CCLAB > 回應機器人2 > Messaging API



- 從Line Bot開發介面取得對應的Line Bot資料，修改程式碼指向提供服務的Line Bot對象。

```
1 from flask import Flask  
2 app = Flask(__name__)  
3  
4 from flask import request, abort  
5 from linebot import LineBotApi, WebhookHandler  
6 from linebot.exceptions import InvalidSignatureError  
7 from linebot.models import MessageEvent, TextMessage, Text
```

匯入模組

```
10 line_bot_api = LineBotApi('你的 CHANNEL_ACCESS_TOKEN')  
11 handler = WebhookHandler('你的 CHANNEL_SECRET')
```

接收對應Line Bot資料

```
12 @app.route("/callback", methods=['POST'])  
13 def callback():  
14     signature = request.headers['X-Line-Signature']  
15     body = request.get_data(as_text=True)  
16     try:  
17         handler.handle(body, signature)  
18     except InvalidSignatureError:  
19         abort(400)  
20     return 'OK'
```

建立callback路由。
Post傳資料

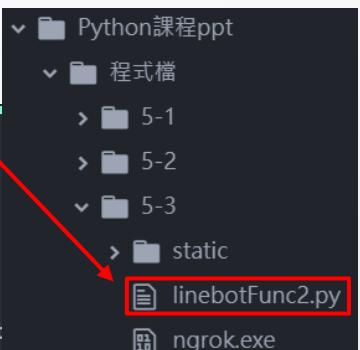
```
22 baseUrl = '你的NGROK網址/static/' #靜態檔案網址
```

接收使用者訊息，並回復。
此處回覆各種基本訊息

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

```
47 if __name__ == '__main__':  
48     app.run()
```

呼叫程式時，執行app.run()



回應多媒體訊息-建立三方連接(本機處理對應Line Bot資料)2

命令提示字元 - ngrok http 5000
ngrok by @inconsreveable

Session Status
Account asd357jk1159@gmail.com (Plan: Free)
Version 2.3.35
Region United States (us)
Web Interface http://127.0.0.1:4040
Forwarding http://0052c052e467.ngrok.io -> http://localhost:5000
Forwarding https://0052c052e467.ngrok.io -> http://localhost:5000

5-3
static
mario.m4a
robot.jpg
robot.mp4
linebotFunc2.py
ngrok.exe

```
baseurl = '你的NGROK網址/static/' #靜態檔案路徑

@handler.add(MessageEvent, message=TextMessage)
def handle_message(event):
    mtext = event.message.text
    if mtext == '@傳送聲音':
        try:
            message = AudioSendMessage(
                original_content_url=baseurl + 'mario.m4a', #聲音檔置於static資料夾
                duration=20000 #聲音長度20秒
            )
            line_bot_api.reply_message(event.reply_token, message)
        except:
            line_bot_api.reply_message(event.reply_token, TextSendMessage(text='發生錯誤！'))
    elif mtext == '@傳送影片':
        try:
            message = VideoSendMessage(
                original_content_url=baseurl + 'robot.mp4', #影片檔置於static資料夾
                preview_image_url=baseurl + 'robot.jpg'
            )
            line_bot_api.reply_message(event.reply_token, message)
        except:
            line_bot_api.reply_message(event.reply_token, TextSendMessage(text='發生錯誤！'))
```

如果收到訊息”@傳送聲音”，則執行以下內容

定義回傳訊息內容及格式

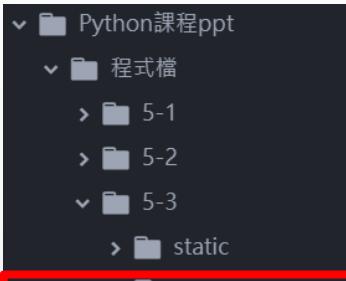
回傳訊息

已讀
下午3:30
@傳送聲音
0:20
音頻
下午3:30

已讀
下午3:30
@傳送影片
影片
下午3:30

回應多媒體訊息-建立三方連接(本機處理對應Line Bot資料)3

打開Anaconda Prompt (anaconda3)
#使用line_env虛擬環境
➤ conda activate line_env
#執行程式
➤ python [path to .py file]



```
Anaconda Prompt (anaconda3) - python C:\Users\ccllab\Documents\專案助理\Python課程ppt\程式檔\5-2\linebotFunc1.py
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
127.0.0.1 - - [25/Mar/2021 12:19:51] " [37mPOST /callback HTTP/1.1 [0m" 200 -
```

```
(line_env) C:\Users\ccllab>python C:\Users\ccllab\Documents\專案助理\Python課程ppt\程式檔\5-2\linebotFunc1.py
* Serving Flask app "linebotFunc1" (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 http://127.0.0.1:5000/ (Press CTRL+C to quit)
127.0.0.1 - - [25/Mar/2021 14:24:40] " [37mPOST /callback HTTP/1.1 [0m" 200 -
127.0.0.1 - - [25/Mar/2021 14:27:11] " [37mPOST /callback HTTP/1.1 [0m" 200 -
127.0.0.1 - - [25/Mar/2021 14:27:42] " [37mPOST /callback HTTP/1.1 [0m" 200 -
127.0.0.1 - - [25/Mar/2021 14:28:00] " [37mPOST /callback HTTP/1.1 [0m" 200 -
127.0.0.1 - - [25/Mar/2021 14:28:14] " [37mPOST /callback HTTP/1.1 [0m" 200 -
127.0.0.1 - - [25/Mar/2021 14:28:33] " [37mPOST /callback HTTP/1.1 [0m" 200 -
127.0.0.1 - - [25/Mar/2021 14:28:41] " [37mPOST /callback HTTP/1.1 [0m" 200 -
127.0.0.1 - - [25/Mar/2021 14:28:55] " [37mPOST /callback HTTP/1.1 [0m" 200 -
```

- 在Line Bot提供服務期間不可中斷本機伺服器的服務，否則無法給與Line Bot使用者回應。

```
(line_env) C:\Users\ccllab>python C:\Users\ccllab\Documents\專案助理\Python課程ppt\程式檔\5-3\linebotFunc2.py
* Serving Flask app "linebotFunc2" (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 http://127.0.0.1:5000/ (Press CTRL+C to quit)
127.0.0.1 - - [25/Mar/2021 15:26:20] " [37mPOST /callback HTTP/1.1 [0m" 200 -
```

回應多媒體訊息-建立三方連接(Line Bot透過代理伺服器連到本機服務)

修改Line Bot的伺服器連結，使Line Bot能夠將使用者訊息傳到本機處理並回應

TOP > CCLAB > 回應機器人2 > Messaging API

```
命令提示字元 - ngrok http 5000
ngrok by @inconshreveable

Session Status          online
Account                 asd357jkl159@gmail.com (Plan: Free)
Version                2.3.35
Region                 United States (us)
Web Interface          http://127.0.0.1:4040
Forwarding             http://0052c052e467.ngrok.io -> http://localhost:5000
Forwarding             https://0052c052e467.ngrok.io -> http://localhost:5000
```

Webhook settings

Webhook URL ⓘ

Edit

Webhook settings

Webhook URL ⓘ

https://0052c052e467.ngrok.io/callback

Verify

Edit

Use webhook ⓘ



● 可驗證Line Bot可
否透過代理伺服
器連到本機服務

```
1  from flask import Flask
2  app = Flask(__name__)

3
4  from flask import request, abort
5  from linebot import LineBotApi, WebhookHandler
6  from linebot.exceptions import InvalidSignatureError
7  from linebot.models import MessageEvent, TextMessage, TextSendMessage, AudioSendM

8
9  line_bot_api = LineBotApi('你的 CHANNEL_ACCESS_TOKEN')
10 handler = WebhookHandler('你的 CHANNEL_SECRET')

11
12 @app.route('/callback', methods=['POST'])
13 def callback():
14     signature = request.headers['X-Line-Signature']
15     body = request.get_data(as_text=True)
16     try:
17         handler.handle(body, signature)
18     except InvalidSignatureError:
19         abort(400)
20     return 'OK'
21
22 baseurl = '你的NGROK網址/static/' #靜態檔案網址
23
24 @handler.add(MessageEvent, message=TextMessage)
25 > def handle_message(event):
26
27     if __name__ == '__main__':
28         app.run()
```

回應多媒體訊息-圖文選單

<http://manager.line.biz/>

LINE Official Account Manager 回應機器人2 @556tiyxu 輕用量 0 回應模式：聊天機器人

主頁 提醒 分析 貼文串 聊天 基本檔案 口袋商店

群發訊息 漸進式訊息 NEW 自動回應訊息

訊息項目 推廣相關

聊天室相關
加入好友的歡迎訊息 圖文選單

資料管理

增加好友人數

收合選單

圖文選單

您可建立具有視覺效果的互動型選單，於聊天室中提供優惠券、網址連結及相關行銷資訊以進行宣傳。本頁面僅可顯示透過本頁面所建立的圖文選單。

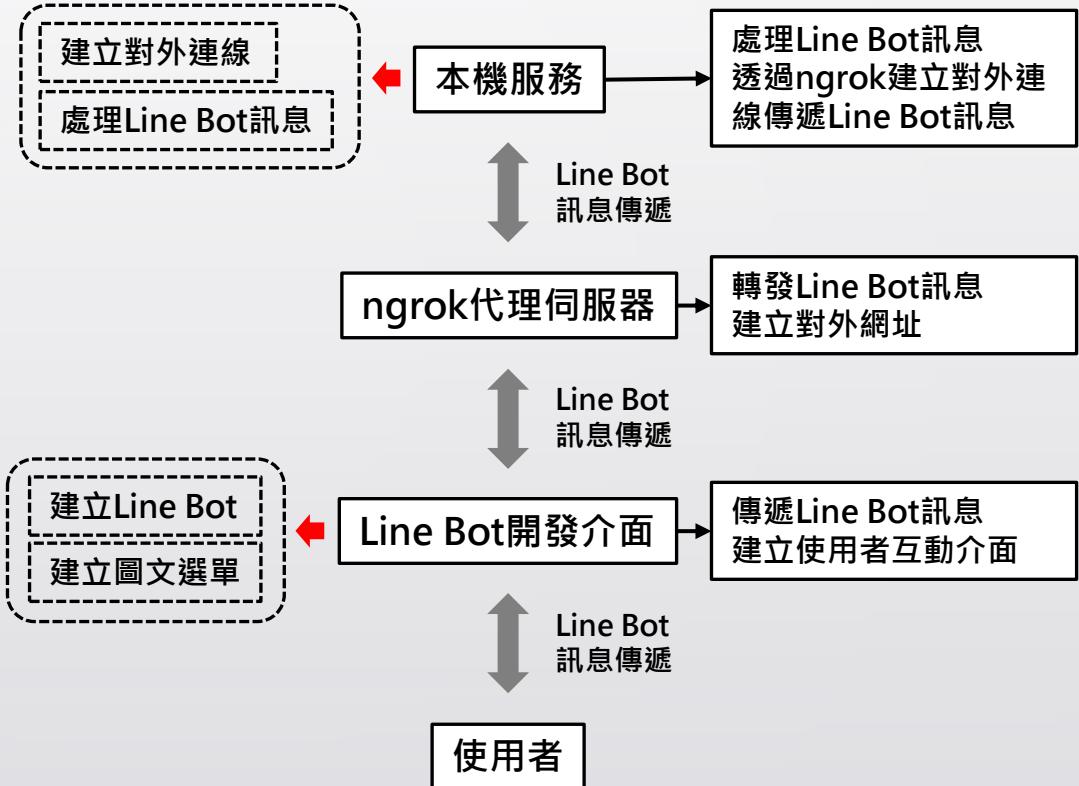
目前顯示的選單

此為顯示於聊天室供用戶點選的圖文選單。
※若透過Messaging API設定圖文選單，用戶實際看到的選單可能與此有所差異

標題	測試圖文選單
使用期間	2021/03/25 00:00 - 2021/03/31 23:59
動作	1 • 文字 - @傳送聲音 2 • 文字 - @傳送影片

預約 / 公開 待設定

回應多媒體訊息-功能 Demo





Part IV

ChatBot.py 進階API功能

回應樣板訊息-建立新的Line Bot

LINE Developers News Products Documentation FAQ Glossary Community Blog

Console home

Providers

Search...

Admin

CCLAB

TOP > CCLAB > 回應機器人3



回應機器人3

Admin

Messaging API

Basic settings

Messaging API

LIFF

Security

Statistics

Roles

Basic settings

Basic information

You can change your app name and icon in [LINE Official Account Manager](#).

Channel ID

1655796474

Channel icon



Channel name

回應機器人3

Channel description

回應樣板訊息

Edit

LINE帳號管理:

url: <https://manager.line.biz/>

功能: 建立linebot、圖形化介面功能、基本帳戶功能設定。

LINE Developers:

url: <https://developers.line.biz/>

功能: line開發者設定。

建立新的Line Bot



建立LINE官方帳號

停用自動發送歡迎及回覆訊息

啟用Messaging API

回應樣板訊息-建立三方連接(本機網頁建立對外通道)

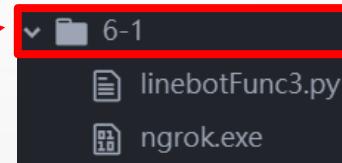
開啟命令提示字元(cmd)

#前往資料夾位置

► cd [path to program folder]

#啟動ngrok伺服器(Flask伺服器預設Port為5000)

► ngrok http 5000



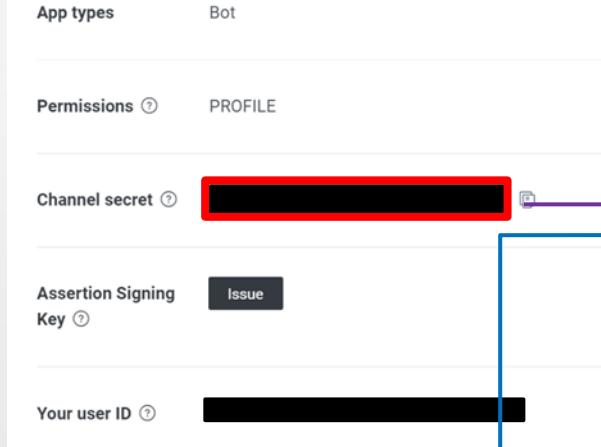
```
命令提示字元 - ngrok http 5000
ngrok by @inconshreveable
Session Status          online
Account                asd357jk1159@gmail.com (Plan: Free)
Version                2.3.35
Region                 United States (us)
Web Interface          http://127.0.0.1:4040
Forwarding             http://a3a789d63b0b.ngrok.io -> http://localhost:5000
                        https://a3a789d63b0b.ngrok.io -> http://localhost:5000
Connections            ttl     opn     rtt1    rt5     p50    p90
                        0       0      0.00   0.00   0.00   0.00
(Ctrl+C to quit)
```

- 代理服務器生成的網址是隨機的，每次執行得到不同網址(=>每次執行為Line Bot提供服務的外部網址會不同，需修改Line Bot設定)
- 在Line Bot提供服務期間不可中斷本機伺服器的對外通道，否則本機無法提供服務

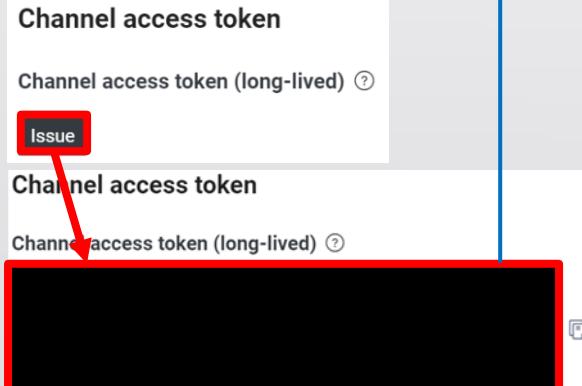
回應樣板訊息-建立三方連接(本機處理對應Line Bot資料)1

<https://developers.line.biz/>

TOP > CCLAB > 回應機器人3 > Basic settings



TOP > CCLAB > 回應機器人3 > Messaging API



- 從Line Bot開發介面取得對應的Line Bot資料，修改程式碼指向提供服務的Line Bot對象。

```
1 from flask import Flask
2 app = Flask(__name__)
3
4 from flask import request, abort
5 from linebot import LineBotApi, WebhookHandler
6 from linebot.exceptions import InvalidSignatureError
7 from linebot.models import MessageEvent, TextMessage, PostbackEvent, TextSendMessage
8 from urllib.parse import parse_qs
9
10
11 line_bot_api = LineBotApi('你的 CHANNEL_ACCESS_TOKEN')
12 handler = WebhookHandler('你的 CHANNEL_SECRET')
13
14 @app.route("/callback", methods=['POST'])
15 def callback():
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
```

匯入模組

接收對應Line Bot資料

建立callback路由，
Post傳資料

1.接收使用者訊息，並回覆。
2.接收模板觸發訊息，並回覆。

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

呼叫程式時，執行app.run()

回應樣板訊息-建立三方連接(本機處理對應Line Bot資料)2

```
@handler.add(MessageEvent, message=TextMessage)
def handle_message(event):
    mtext = event.message.text
    if mtext == '@按鈕樣板':
        sendButton(event)

    elif mtext == '@確認樣板':
        sendConfirm(event)

    elif mtext == '@轉盤樣板':
        sendCarousel(event)

    elif mtext == '@圖片轉盤':
        sendImgCarousel(event)

    elif mtext == '@購買披薩':
        sendPizza(event)

    elif mtext == '@yes':
        sendYes(event)
```



```
@handler.add(PostbackEvent) #PostbackTemplate
def handle_postback(event):
    backdata = dict(parse_qsl(event.postback.data)) #取得Postback資料
    if backdata.get('action') == 'buy':
        sendBack_buy(event, backdata)
    elif backdata.get('action') == 'sell':
        sendBack_sell(event, backdata)
```

回應樣板訊息-建立三方連接(本機處理對應Line Bot資料)3

```
def sendButton(event): #按鈕樣版
    try:
        message = TemplateSendMessage(
            alt_text='按鈕樣板',
            template=ButtonsTemplate(
                thumbnail_image_url='https://i.imgur.com/4QfKuz1.png', #顯示的圖片
                title='按鈕樣版示範', #主標題
                text='請選擇：', #副標題
                actions=[
                    MessageTemplateAction( #顯示文字訊息
                        label='文字訊息',
                        text='@購買披薩'
                    ),
                    URITemplateAction( #開啟網頁
                        label='連結網頁',
                        uri='http://www.e-happy.com.tw'
                    ),
                    PostbackTemplateAction( #執行Postback功能, 觸發Postback事件
                        label='回傳訊息', #按鈕文字
                        #text='@購買披薩', #顯示文字訊息
                        data='action=buy' #Postback資料
                    ),
                ]
            )
        @handler.add(PostbackEvent) #PostbackTemplateAction
        def handle_postback(event):
            backdata = dict(parse_qs(event.postback.data))
            if backdata.get('action') == 'buy':
                sendBack_buy(event, backdata)
            elif backdata.get('action') == 'sell':
                sendBack_sell(event, backdata)
    except:
```



```
def sendBack_buy(event, backdata): #處理Postback
    try:
        text1 = '感謝您購買披薩，我們將盡快為您製作。\\n(action 的值為 ' + backdata.get('action') + ')'
        text1 += '\\n(可將處理程式寫在此處。)'
        message = TextSendMessage( #傳送文字
            text = text1
        )
        line_bot_api.reply_message(event.reply_token, message)
    except:
        line_bot_api.reply_message(event.reply_token, TextSendMessage(text='發生錯誤！'))
```

9:29

100

< 16 **cloudsystem2022** ⚡

為您製作。

上午 9:27

已讀
上午 9:27

@按鈕樣板



按鈕樣版示範

請選擇：

文字訊息

連結網頁

回傳訊息

上午 9:27



感謝您購買披薩，我們將盡快
為您製作。

(action 的值為 buy)

(可將處理程式寫在此處。)

上午 9:27



回應樣板訊息-建立三方連接(本機處理對應Line Bot資料)4

```
def sendConfirm(event): #確認樣板
    try:
        message = TemplateSendMessage(
            alt_text='確認樣板',
            template=ConfirmTemplate(
                text='你確定要購買這項商品嗎？',
                actions=[
                    MessageTemplateAction( #按鈕選項
                        label='是',
                        text='@yes'
                    ),
                    MessageTemplateAction(
                        label='否',
                        text='@no'
                    )
                ]
            )
        )
    
```

```
@handler.add(MessageEvent, message=TextMessage)
except:
    def handle_message(event):
        mtext = event.message.text
        if mtext == '@按鈕樣板':
            ...
        elif mtext == '@確認樣板':
            ...
        elif mtext == '@轉盤樣板':
            ...
        elif mtext == '@圖片轉盤':
            ...
        elif mtext == '@購買披薩':
            ...
        elif mtext == '@yes':
            sendYes(event)
```

```
def sendYes(event):
    try:
        message = TextSendMessage(
            text='感謝您的購買，\n我們將盡快寄出商品。'
        )
        line_bot_api.reply_message(event.reply_token, message)
    except:
        line_bot_api.reply_message(event.reply_token, TextSendMessage(text='發生錯誤！'))
```



回應樣板訊息-建立三方連接(本機處理對應Line Bot資料)5

```
def sendCarousel(event): #轉盤樣板
    try:
        message = TemplateSendMessage(
            alt_text='轉盤樣板',
            template=CarouselTemplate(
                columns=[

                    CarouselColumn(
                        thumbnail_image_url='https://i.imgur.com/4QfKuz1.png',
                        title='這是樣板一',
                        text='第一個轉盤樣板',
                        actions=[

                            MessageTemplateAction(
                                label='文字訊息一',
                                text='賣披薩'
                            ),
                            URITemplateAction(
                                label='連結文淵閣網頁',
                                uri='http://www.e-happy.com.tw'
                            ),
                            PostbackTemplateAction(
                                label='回傳訊息一',
                                data='action=sell&item=披薩'
                            )
                        ]
                    ),
                    CarouselColumn(
                        thumbnail_image_url='https://i.imgur.com/oAdBkR.png',
                        title='這是樣板二',
                        text='第二個轉盤樣板',
                        actions=[

                            @handler.add(PostbackEvent) #PostbackTemplateAction觸發此事件
                            def handle_postback(event):
                                backdata = dict(parse_qsl(event.postback.data)) #取得Postback資料
                                if backdata.get('action') == 'buy':
                                    sendBack_buy(event, backdata)
                                elif backdata.get('action') == 'sell':
                                    sendBack_sell(event, backdata)

                            PostbackTemplateAction(
                                label='回傳訊息二',
                                data='action=sell&item=飲料'
                            )
                        ]
                    )
                ]
            )
        line_bot_api.reply_message(event.reply_token,message)
    except:
        line_bot_api.reply_message(event.reply_token,TextSendMessage(text='發生錯誤！'))
```



```
def sendBack_sell(event, backdata): #處理Postback
    try:
        message = TextSendMessage( #傳送文字
            text = '點選的是賣 ' + backdata.get('item')
        )
        line_bot_api.reply_message(event.reply_token, message)
    except:
        line_bot_api.reply_message(event.reply_token,TextSendMessage(text='發生錯誤！'))
```

回應樣板訊息-建立三方連接(本機處理對應Line Bot資料)6

```
def sendImgCarousel(event): #圖片轉盤
    try:
        message = TemplateSendMessage(
            alt_text='圖片轉盤樣板',
            template=ImageCarouselTemplate(
                columns=[  

                    ImageCarouselColumn(  

                        image_url='https://i.imgur.com/4QfKuz1.png',
                        action=MessageTemplateAction(  

                            label='文字訊息',
                            text='賣披薩'
                        )
                    ),
                    ImageCarouselColumn(  

                        image_url='https://i.imgur.com/qaAdBkR.png',
                        action=PostbackTemplateAction(  

                            label='回傳訊息',
                            data='action=sell&item=飲料'
                        )
                    )
                ]
            )
        line_bot_api.reply_message(event.reply_token, message)
    except:
        line_bot_api.reply_message(event.reply_token, TextSendMessage(text='發生錯誤！'))
```



```
@handler.add(PostbackEvent) #PostbackTemplateAction 觸發此事件
def handle_postback(event):
    backdata = dict(parse_qs(event.postback.data)) #取得Postback資料
    if backdata.get('action') == 'buy':
        sendBack_buy(event, backdata)
    elif backdata.get('action') == 'sell':
        sendBack_sell(event, backdata)

def sendBack_sell(event, backdata): #處理Postback
    try:
        message = TextSendMessage( #傳送文字
            text = '點選的是賣 ' + backdata.get('item')
        )
        line_bot_api.reply_message(event.reply_token, message)
    except:
        line_bot_api.reply_message(event.reply_token, TextSendMessage(text='發生錯誤！'))
```

回應樣板訊息-建立三方連接(本機處理對應Line Bot資料)3

打開Anaconda Prompt (anaconda3)

```
#使用line_env虛擬環境  
► conda activate line_env  
#執行程式  
► python [path to .py file]
```



Anaconda Prompt (anaconda3) - python C:\Users\ccllab\Documents\專案助理\Python課程ppt\程式檔\6-1\linebotFunc3.py

```
127.0.0.1 - - [25/Mar/2021 14:28:33] " [37mPOST /callback HTTP/1.1 [0m" 200 -  
127.0.0.1 - - [25/Mar/2021 14:28:41] " [37mPOST /callback HTTP/1.1 [0m" 200 -  
127.0.0.1 - - [25/Mar/2021 14:28:55] " [37mPOST /callback HTTP/1.1 [0m" 200 -
```

(line_env) C:\Users\ccllab>python C:\Users\ccllab\Documents\專案助理\Python課程ppt\程式檔\5-3\linebotFunc2.py

```
* Serving Flask app "linebotFunc2" (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 http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

```
127.0.0.1 - - [25/Mar/2021 15:26:20] " [37mPOST /callback HTTP/1.1 [0m" 200 -  
127.0.0.1 - - [25/Mar/2021 15:30:10] " [37mPOST /callback HTTP/1.1 [0m" 200 -  
127.0.0.1 - - [25/Mar/2021 15:30:21] " [37mPOST /callback HTTP/1.1 [0m" 200 -  
127.0.0.1 - - [25/Mar/2021 15:30:34] " [37mPOST /callback HTTP/1.1 [0m" 200 -  
127.0.0.1 - - [25/Mar/2021 15:30:37] " [37mGET /static/robot.jpg HTTP/1.1 [0m" 200 -  
127.0.0.1 - - [25/Mar/2021 15:30:40] " [37mGET /static/robot.mp4 HTTP/1.1 [0m" 200 -
```

- 在Line Bot提供服務期間不可中斷本機伺服器的服務，否則無法給與Line Bot使用者回應。

(line_env) C:\Users\ccllab>python C:\Users\ccllab\Documents\專案助理\Python課程ppt\程式檔\6-1\linebotFunc3.py

```
* Serving Flask app "linebotFunc3" (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 http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

回應樣板訊息-建立三方連接(Line Bot透過代理伺服器連到本機服務)

修改Line Bot的伺服器連結，使Line Bot能夠將使用者訊息傳到本機處理並回應

TOP > CCLAB > 回應機器人3 > Messaging API

Webhook settings

Webhook URL ②

Edit

Webhook settings

Webhook URL ②

https://a3a789d63b0b.ngrok.io/callback

Verify

Edit

Use webhook ②



● 可驗證Line Bot可
否透過代理伺服
器連到本機服務

```
c:\ 命令提示字元 - ngrok http 5000
ngrok by @inconshreveable

Session Status
Account          online
Version          asd357jk1159@gmail.com (Plan: Free)
Region           2.3.35
Web Interface   United States (us)
Forwarding       http://127.0.0.1:4040
Forwarding       https://a3a789d63b0b.ngrok.io-> http://localhost:5000
Forwarding       https://a3a789d63b0b.ngrok.io-> http://localhost:5000

Connections
          ttl     opn      rtt      rt5      p50      p90
          0        0     0.00     0.00    0.00    0.00
```

```
6  from linebot import LineBotApi
7  from linebot.models import MessageEvent, TextMessage, PostbackEvent, TextSendMessage
8  from urllib.parse import parse_qs

10 line_bot_api = LineBotApi('你的 CHANNEL_ACCESS_TOKEN')
11 handler = WebhookHandler('你的 CHANNEL_SECRET')

13 @app.route("/callback", methods=['POST'])
14 def callback():
22

23 @handler.add(MessageEvent, message=TextMessage)
24 def handle_message(event):
25

44 @handler.add(PostbackEvent) #PostbackTemplateAction觸發此事件
45 def handle_postback(event):
51

52 def sendButton(event): #按鈕樣板
53
54 def sendConfirm(event): #確認樣板
55
56 def sendCarousel(event): #轉盤樣板
57
58 def sendImgCarousel(event): #圖片轉盤
59
60
61 def sendYes(event):
62
63
64 def sendBack_buy(event, backdata): #處理Postback
65
66 def sendBack_sell(event, backdata): #處理Postback
67
68
69 if __name__ == '__main__':
70     app.run()
```

回應樣板訊息-圖文選單

<http://manager.line.biz/>

LINE Official Account Manager



回應機器人3 ▾

@162ysxvh

輕用量

1 人

回應模式：聊天機器人

主頁 提醒 分析 貼文串 聊天 基本檔案 口袋商店

群發訊息

漸進式訊息 NEW

自動回應訊息

訊息項目

推廣相關

聊天室相關

加入好友的歡迎訊息

圖文選單

資料管理

增加好友人數

收合選單

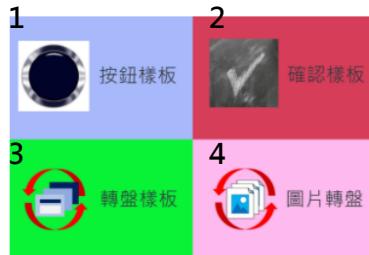
圖文選單

您可建立具有視覺效果的互動型選單，於聊天室中提供優惠券、網址連結及相關行銷資訊以進行宣傳。
本頁面僅可顯示透過本頁面所建立的圖文選單。

目前顯示的選單

此為顯示於聊天室供用戶點選的圖文選單。

※若透過Messaging API設定圖文選單，用戶實際看到的選單可能與此有所差異



標題 測試圖文選單

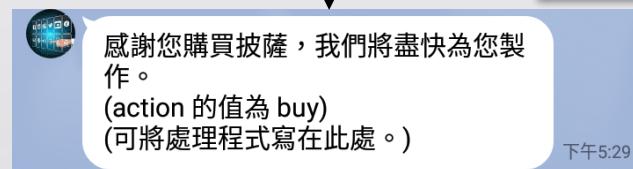
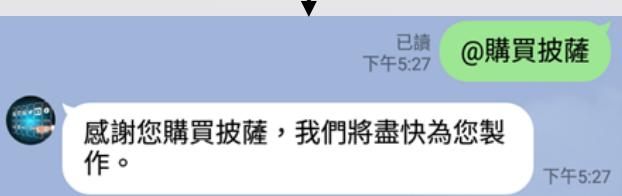
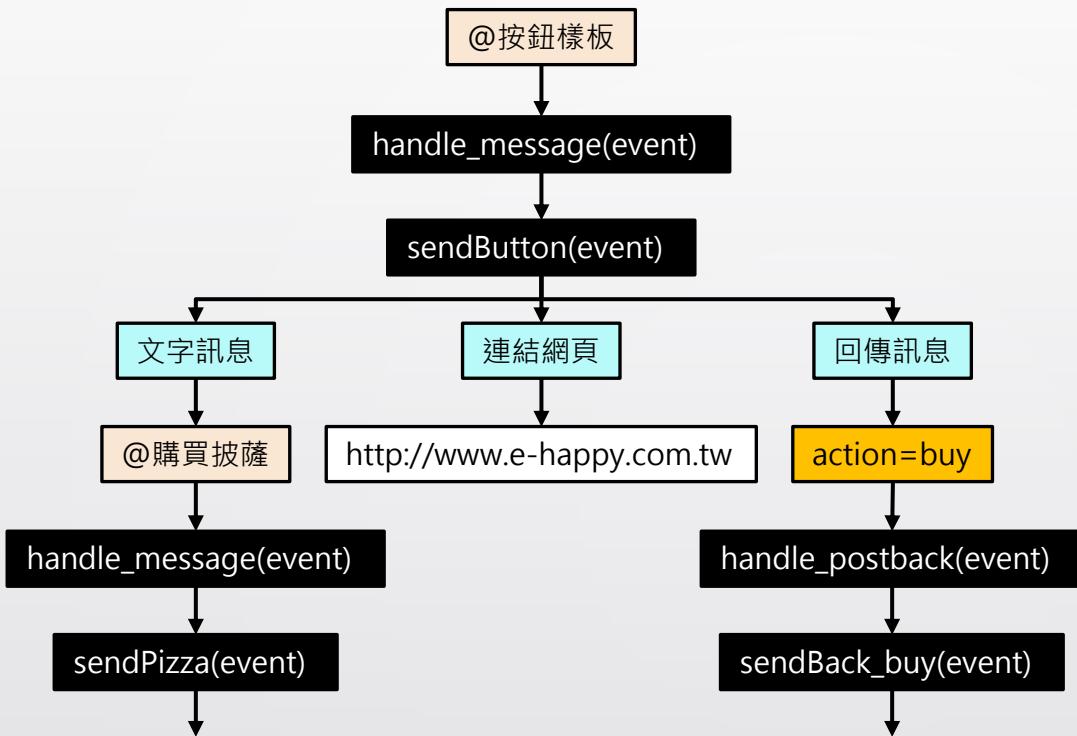
使用期間 2021/04/07 00:00 - 2021/04/30 23:59

- 動作
- 1 • 文字 - @按鈕樣板
 - 2 • 文字 - @確認樣板
 - 3 • 文字 - @轉盤樣板
 - 4 • 文字 - @圖片轉盤

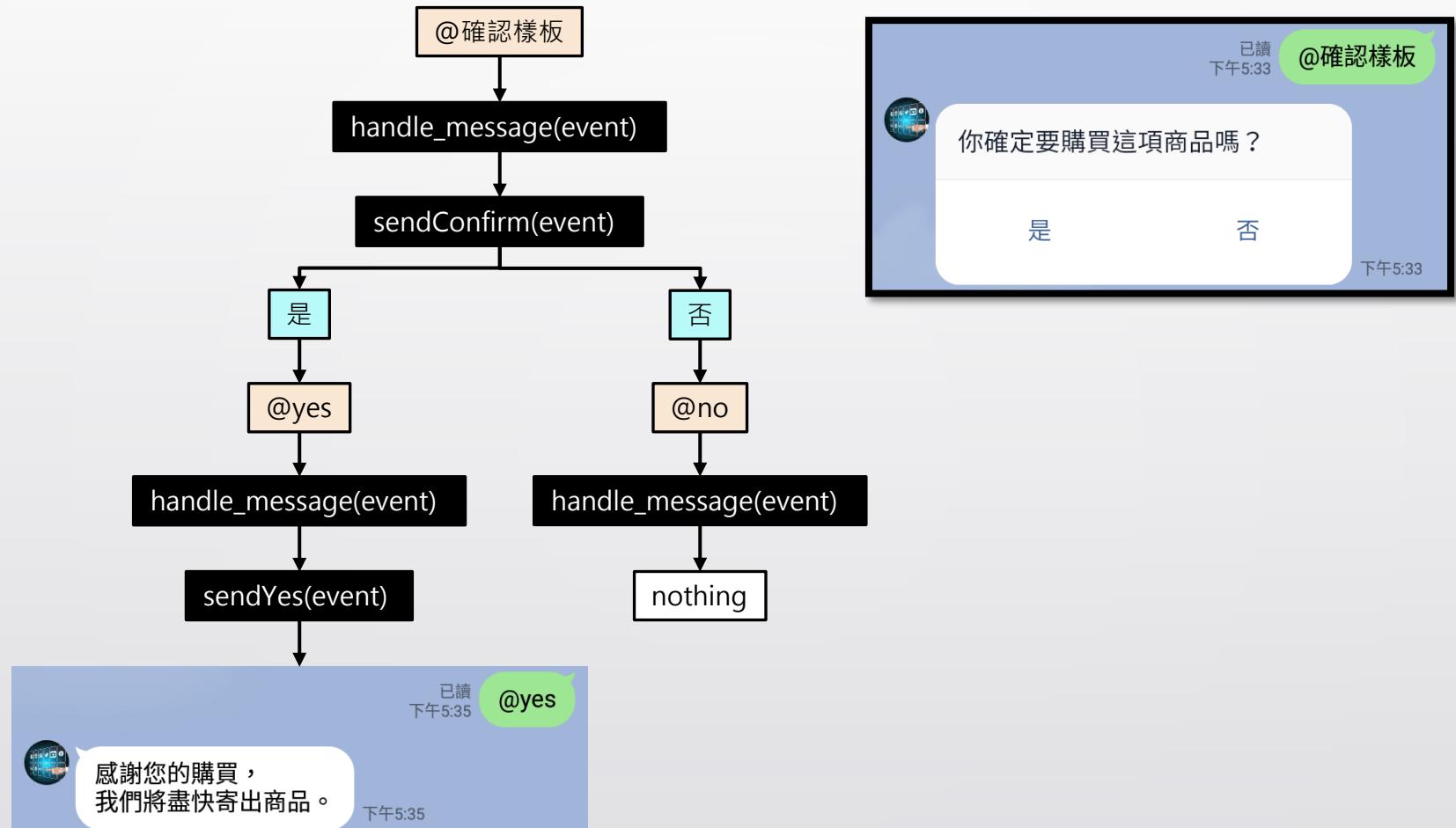
預約 / 公開

待設定

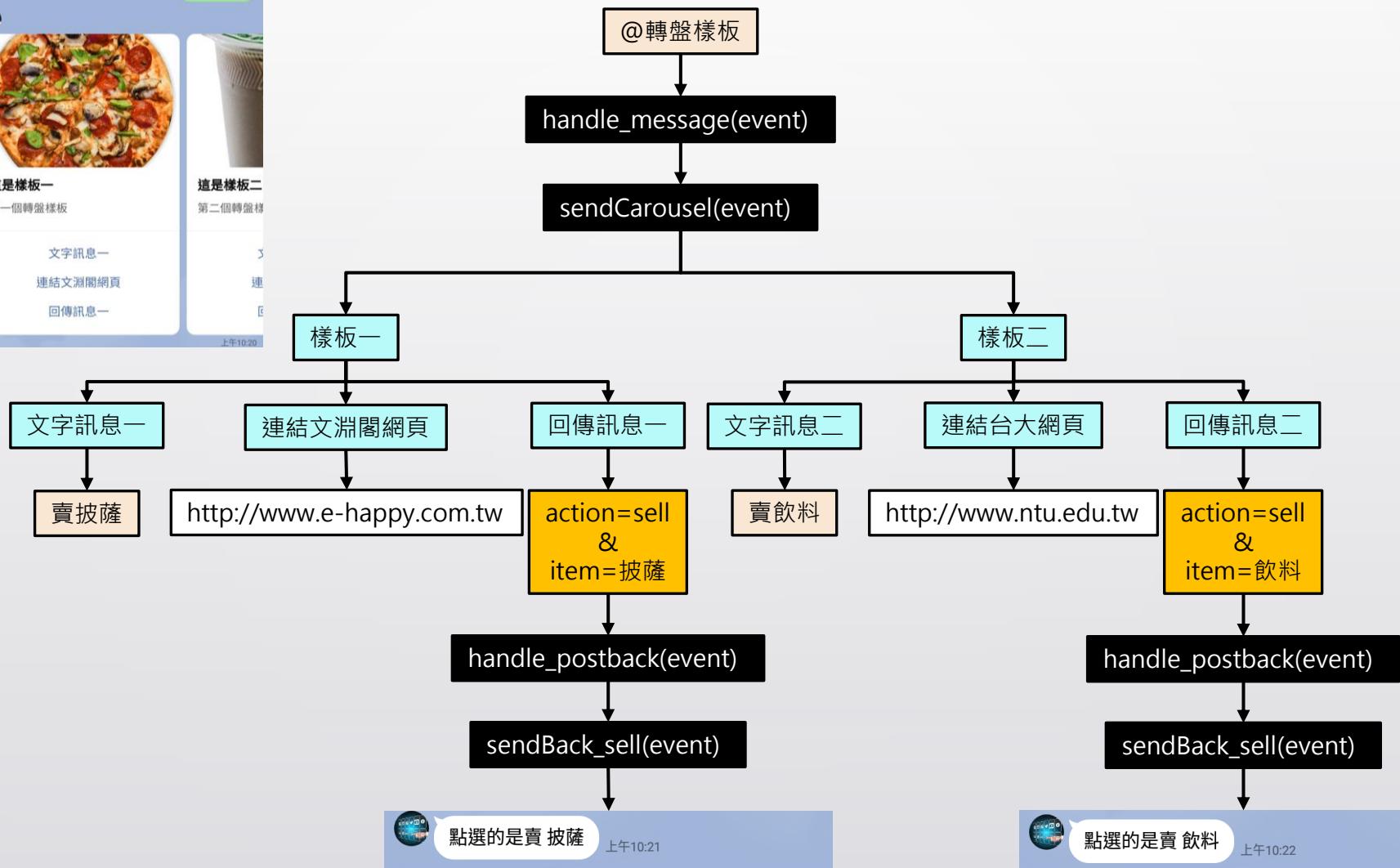
回應樣板訊息-功能 Demo 1



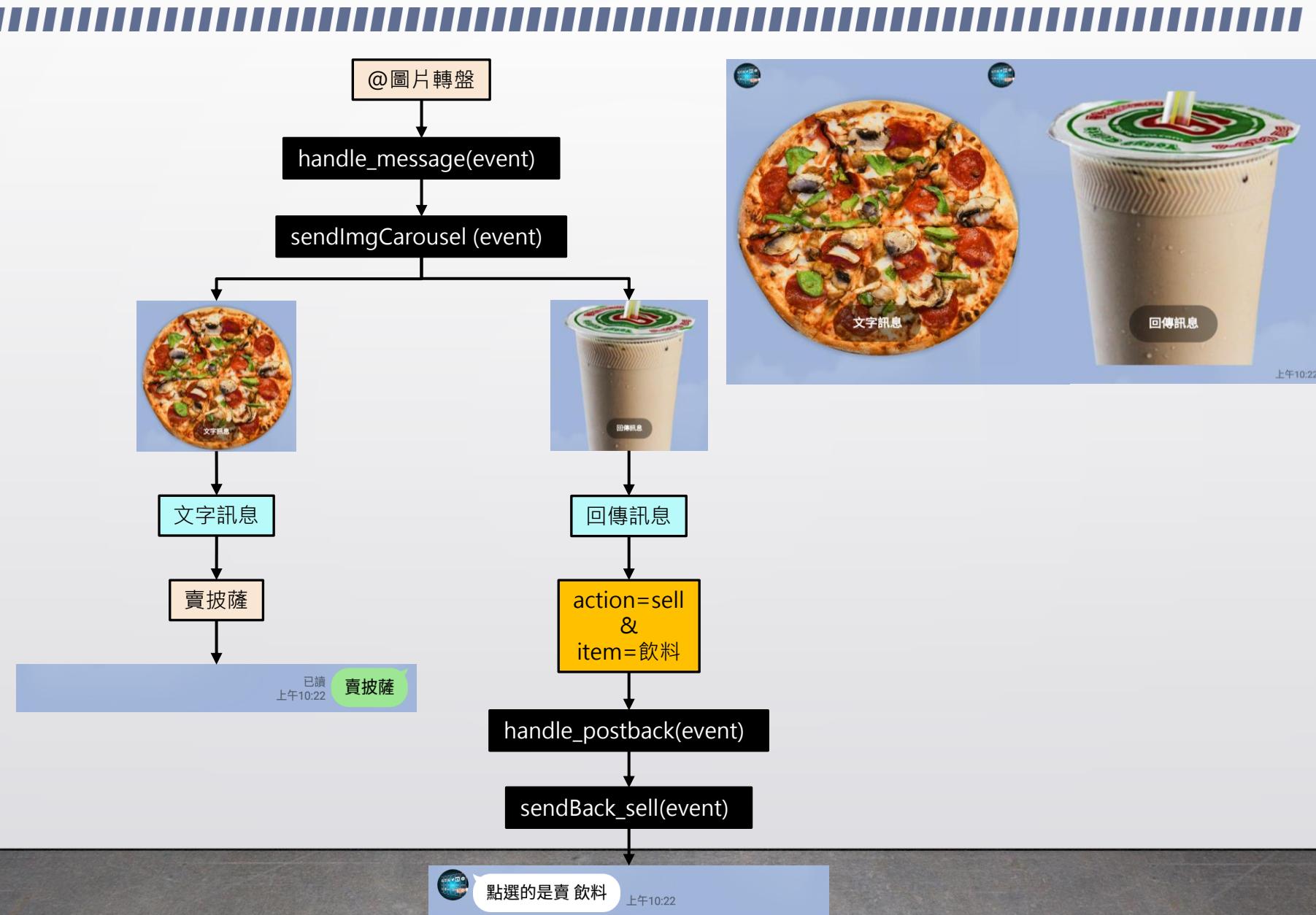
回應樣板訊息-功能 Demo 2



回應樣板訊息-功能 Demo 3



回應樣板訊息-功能 Demo 4



圖片地圖及日期時間-建立新的Line Bot

LINE Developers News Products Documentation FAQ Glossary Community Blog

Console home

Providers

Search...

Admin

CCLAB



回應機器人4

Admin

Messaging API

Basic settings

Messaging API

LIFF

Security

Statistics

Roles

Basic settings

Basic information

You can change your app name and icon in [LINE Official Account Manager](#).

Channel ID

1655797105

Channel icon



Channel name

回應機器人4

Channel description 圖片、時間圖形化界面

Edit

LINE帳號管理:

url: <https://manager.line.biz/>

功能: 建立linebot、圖形化介面功能、基本帳戶功能設定。

LINE Developers:

url: <https://developers.line.biz/>

功能: line開發者設定。

建立新的Line Bot



建立LINE官方帳號

停用自動發送歡迎及回覆訊息

啟用Messaging API

圖片地圖及日期時間-建立三方連接(本機網頁建立對外通道)

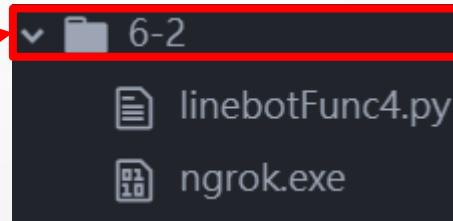
開啟命令提示字元(cmd)

#前往資料夾位置

> cd [path to program folder]

#啟動ngrok伺服器(Flask伺服器預設Port為5000)

> ngrok http 5000



```
命令提示字元 - ngrok http 5000
ngrok by @inconshreveable
(Ctrl+C to quit)

Session Status          online
Account                 asd357jkl159@gmail.com (Plan: Free)
Version                2.3.35
Region                 United States (us)
Web Interface          http://127.0.0.1:4040
Forwarding             http://bc757a5b0505.ngrok.io -> http://localhost:5000
                        https://bc757a5b0505.ngrok.io -> http://localhost:5000

Connections            ttl     opn     rtt      rt5     p50      p90
                        1       0      0.01    0.00    0.30    0.30

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

- 代理服務器生成的網址是隨機的，每次執行得到不同網址(=>每次執行為Line Bot提供服務的外部網址會不同，需修改Line Bot設定)
- 在Line Bot提供服務期間不可中斷本機伺服器的對外通道，否則本機無法提供服務

圖片地圖及日期時間-建立三方連接(本機處理對應Line Bot資料)1

<https://developers.line.biz/>

TOP > CCLAB > 回應機器人4 > Basic settings

App types: Bot
Permissions: PROFILE
Channel secret: [REDACTED]
Assertion Signing Key: Issue
Your user ID: [REDACTED]

TOP > CCLAB > 回應機器人4 > Messaging API

Channel access token
Channel access token (long-lived): Issue
Channel access token
Channel access token (long-lived): [REDACTED]

- 從Line Bot開發介面取得對應的Line Bot資料，修改程式碼指向提供服務的Line Bot對象。

```
1 from flask import Flask
2 app = Flask(__name__)
3
4 from flask import request, abort
5 from linebot import LineBotApi, WebhookHandler
6 from linebot.exceptions import InvalidSignatureError
7 from linebot.models import MessageEvent, TextMessage, PostbackEvent, TextSendMessage
8 from urllib.parse import parse_qs
9 import datetime
```

匯入模組

```
10
11 line_bot_api = LineBotApi('你的 CHANNEL_ACCESS_TOKEN')
12 handler = WebhookHandler('你的 CHANNEL_SECRET')
```

接收對應Line Bot資料

```
13
14 @app.route("/callback", methods=['POST'])
15 def callback():=
```

建立callback路由，
Post傳資料

```
16
17 @handler.add(MessageEvent, message=TextMessage)
18 def handle_message(event):=
```

```
19
20 @handler.add(PostbackEvent) #PostbackTemplateAction觸發事件
21 def handle_postback(event):=
```

```
22
23 def sendImgmap(event): #圖片地圖
24
25 def sendDatetime(event): #日期時間
26
27 def sendData_sell(event, backdata): #Postback,顯示日期時間
```

1.接收使用者訊息，並回覆。
2.接收模板觸發訊息，並回覆。

```
28
29 if __name__ == '__main__':
30     app.run()
```

呼叫程式時，執行app.run()

圖片地圖及日期時間-建立三方連接(本機處理對應Line Bot資料)2

```
@handler.add(MessageEvent, message=TextMessage)
def handle_message(event):
    mtext = event.message.text
    if mtext == '@圖片地圖':
        sendImgmap(event)
    elif mtext == '@日期時間':
        sendDatetime(event)
```

接收使用者訊息並回覆



```
@handler.add(PostbackEvent) #PostbackTemplateAction觸發此事件
def handle_postback(event):
    backdata = dict(parse_qs(event.postback.data)) #取得data資料
    if backdata.get('action') == 'sell':
        sendData_sell(event, backdata)
```

接收模板觸發訊息並回覆

圖片地圖及日期時間-建立三方連接(本機處理對應Line Bot資料)3

```
def sendImagemap(event): #圖片地圖
    try:
        image_url = 'https://i.imgur.com/Yz2yzve.jpg' #圖片位址
        imgwidth = 1040 #原始圖片寬度一定要1040
        imgheight = 300
        message = ImagemapSendMessage(
            base_url=image_url,
            alt_text="圖片地圖範例",
            base_size=BaseSize(height=imgheight, width=imgwidth), #圖片寬及高
            actions=[
                MessageImagemapAction( #顯示文字訊息
                    text='你點選了紅色區塊！',
                    area=ImagemapArea( #設定圖片範圍: 左方1/4區域
                        x=0,
                        y=0,
                        width=imgwidth*0.25,
                        height=imgheight
                    )
                ),
                URIImagemapAction( #開啟網頁
                    link_uri='http://www.e-happy.com.tw',
                    area=ImagemapArea( #右方1/4區域(藍色)
                        x=imgwidth*0.75,
                        y=0,
                        width=imgwidth*0.25,
                        height=imgheight
                    )
                )
            ]
        )
        line_bot_api.reply_message(event.reply_token, message)
    except:
        line_bot_api.reply_message(event.reply_token, TextSendMessage(text='發生錯誤！'))
```



9:39

回傳訊息

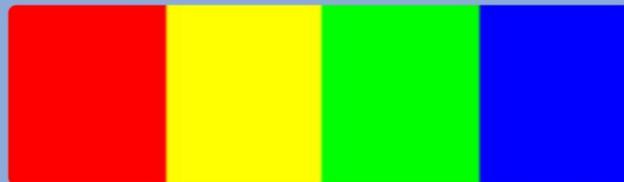
4G 99

< 20 ★ cloudsystem2022

上午 9:32

已讀
上午 9:39

@圖片地圖



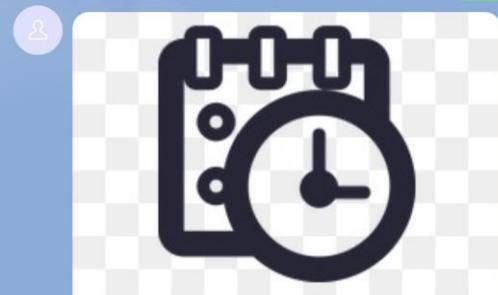
上午 9:39

已讀
上午 9:39

你點選了紅色區塊！

已讀
上午 9:39

@日期時間



日期時間示範

請選擇：

選取日期

選取時間

選取日期時間

上午 9:39



Aa



圖片地圖及日期時間-建立三方連接(本機處理對應Line Bot資料)4

```

def sendDatetime(event): #日期時間
    try:
        message = TemplateSendMessage(
            alt_text='日期時間範例',
            template=ButtonsTemplate(
                thumbnail_image_url='https://i.imgur.com/VxVB46z.jpg',
                title='日期時間示範',
                text='請選擇：',
                actions=[
                    DatetimePickerTemplateAction(
                        label="選取日期",
                        data="action=sell&mode=date", #觸發postback事件
                        mode="date", #選取日期
                        initial="2020-10-01", #顯示初始日期
                        min="2020-10-01", #最小日期
                        max="2021-12-31" #最大日期
                    ),
                    DatetimePickerTemplateAction(
                        label="選取時間",
                        data="action=sell&mode=time",
                        mode="time", #選取時間
                        initial="10:00",
                        min="00:00",
                        max="23:59"
                    ),
                    DatetimePickerTemplateAction(
                        label="選取日期時間",
                        data="action=sell&mode=datetime",
                        mode="datetime", #選取日期時間
                        initial="2020-10-01T10:00",
                        min="2020-10-01T00:00",
                        max="2021-12-31T23:59"
                    )
                ]
            )
        line_bot_api.reply_message(event.reply_token,message)
    except:
        line_bot_api.reply_message(event.reply_token,TextSendMessage(text='發生錯誤！'))

```



已讀 下午6:42 @日期時間

日期時間示範
請選擇：

選取日期
選取時間
選取日期時間

下午6:42

`@handler.add(PostbackEvent) #PostbackTemplateAction觸發此事件`

```

def handle_postback(event):
    backdata = dict(parse_qs(event.postback.data)) #取得data資料
    if backdata.get('action') == 'sell':
        sendData_sell(event, backdata)

```

`def sendData_sell(event, backdata): #Postback,顯示日期時間`

```

try:
    if backdata.get('mode') == 'date':
        dt = '日期為：' + event.postback.params.get('date')
    elif backdata.get('mode') == 'time':
        dt = '時間為：' + event.postback.params.get('time')
    elif backdata.get('mode') == 'datetime':
        dt = datetime.datetime.strptime(event.postback.params.get('datetime'), '%Y-%m-%dT%H:%M')
        dt = dt.strftime('{d}%Y-%m-%d, {t}%H:%M').format(d='日期為：', t='時間為：') #轉為字符串
    message = TextSendMessage(
        text=dt
    )
    line_bot_api.reply_message(event.reply_token,message)
except:
    line_bot_api.reply_message(event.reply_token,TextSendMessage(text='發生錯誤！'))

```

日期為：2020-10-01 下午6:43

時間為：10:00 下午6:43

日期為：2020-10-01, 時間為：10:00 下午6:44

圖片地圖及日期時間-建立三方連接(本機處理對應Line Bot資料)5

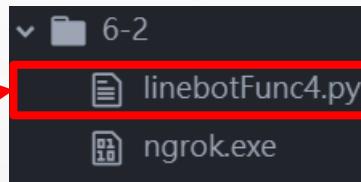
打開Anaconda Prompt (anaconda3)

#使用line_env虛擬環境

➤ conda activate line_env

#執行程式

➤ python [path to .py file]



```
Anaconda Prompt (anaconda3) - python C:\Users\ccllab\Documents\專案助理\Python課程ppt\程式檔\6-2\linebotFunc4.py
```

```
127.0.0.1 - - [25/Mar/2021 15:30:34] " [37mPOST /callback HTTP/1.1 [0m" 200 -
127.0.0.1 - - [25/Mar/2021 15:30:37] " [37mGET /static/robot.jpg HTTP/1.1 [0m" 200 -
127.0.0.1 - - [25/Mar/2021 15:30:40] " [37mGET /static/robot.mp4 HTTP/1.1 [0m" 200 -
```

```
(line_env) C:\Users\ccllab>python C:\Users\ccllab\Documents\專案助理\Python課程ppt\程式檔\6-1\linebotFunc3.py
```

```
* Serving Flask app "linebotFunc3" (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 http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

```
127.0.0.1 - - [25/Mar/2021 17:20:08] " [37mPOST /callback HTTP/1.1 [0m" 200 -
127.0.0.1 - - [25/Mar/2021 17:22:15] " [37mPOST /callback HTTP/1.1 [0m" 200 -
127.0.0.1 - - [25/Mar/2021 17:26:13] " [37mPOST /callback HTTP/1.1 [0m" 200 -
127.0.0.1 - - [25/Mar/2021 17:27:06] " [37mPOST /callback HTTP/1.1 [0m" 200 -
127.0.0.1 - - [25/Mar/2021 17:29:29] " [37mPOST /callback HTTP/1.1 [0m" 200 -
127.0.0.1 - - [25/Mar/2021 17:33:28] " [37mPOST /callback HTTP/1.1 [0m" 200 -
127.0.0.1 - - [25/Mar/2021 17:35:31] " [37mPOST /callback HTTP/1.1 [0m" 200 -
127.0.0.1 - - [25/Mar/2021 17:41:20] " [37mPOST /callback HTTP/1.1 [0m" 200 -
127.0.0.1 - - [25/Mar/2021 17:42:04] " [37mPOST /callback HTTP/1.1 [0m" 200 -
```

- 在Line Bot提供服務期間不可中斷本機伺服器的服務，否則無法給與Line Bot使用者回應。

```
(line_env) C:\Users\ccllab>python C:\Users\ccllab\Documents\專案助理\Python課程ppt\程式檔\6-2\linebotFunc4.py
```

```
* Serving Flask app "linebotFunc4" (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 http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

```
127.0.0.1 - - [25/Mar/2021 18:38:19] " [37mPOST /callback HTTP/1.1 [0m" 200 -
```

圖片地圖及日期時間-建立三方連接(Line Bot透過代理伺服器連到本機服務)

修改Line Bot的伺服器連結，使Line Bot能夠將使用者訊息傳到本機處理並回應

TOP > CCLAB > 回應機器人4 > Messaging API

```
c:\ 命令提示字元 - ngrok http 5000
ngrok by @inconshreveable

Session Status          online
Account                  asd357jk115@gmail.com (Plan: Free)
Version                 2.3.35
Region                  United States (us)
Web Interface           http://127.0.0.1:4040
Forwarding              http://bc757a5b0505.ngrok.io -> http://localhost:5000
Forwarding              https://bc757a5b0505.ngrok.io -> http://localhost:5000
```

Webhook settings

Webhook URL ⓘ

Edit

Webhook settings

Webhook URL ⓘ

https://bc757a5b0505.ngrok.io/callback

Verify

Edit

Use webhook ⓘ



● 可驗證Line Bot可
否透過代理伺服
器連到本機服務

```
1  from flask import Flask
2  app = Flask(__name__)

3
4  from flask import request, abort
5  from linebot import LineBotApi, WebhookHandler
6  from linebot.exceptions import InvalidSignatureError
7  from linebot.models import MessageEvent, TextMessage, PostbackEvent, TextSendMessage
8  from urllib.parse import parse_qs
9  import datetime

10
11 line_bot_api = LineBotApi('你的 CHANNEL_ACCESS_TOKEN')
12 handler = WebhookHandler('你的 CHANNEL_SECRET')

13
14 @app.route('/callback', methods=['POST'])
15 def callback():
16
17
18    @handler.add(MessageEvent, message=TextMessage)
19    def handle_message(event):
20
21
22    @handler.add(PostbackEvent) #PostbackTemplateAction觸發此事件
23    def handle_postback(event):
24
25
26    def sendImgmap(event): #圖片地圖
27
28    def sendDatetime(event): #日期時間
29
30
31    def sendData_sell(event, backdata): #Postback,顯示日期時間
32
33
34    if __name__ == '__main__':
35        app.run()
```

圖片地圖及日期時間-圖文選單

<http://manager.line.biz/>

LINE Official Account Manager 回應機器人4 @898oqhun 輕用量 1 人 回應模式：聊天機器人

主頁 提醒 分析 貼文串 聊天 基本檔案 口袋商店

群發訊息 漸進式訊息 (NEW) 自動回應訊息

訊息項目 推廣相關

聊天室相關
加入好友的歡迎訊息
圖文選單

資料管理

增加好友人數

收合選單

圖文選單

您可建立具有視覺效果的互動型選單，於聊天室中提供優惠券、網址連結及相關行銷資訊以進行宣傳。
本頁面僅可顯示透過本頁面所建立的圖文選單。

目前顯示的選單

此為顯示於聊天室供用戶點選的圖文選單。
※若透過Messaging API設定圖文選單，用戶實際看到的選單可能與此有所差異

標題 測試圖文選單
使用期間 2021/04/07 00:00 - 2021/04/30 23:59
動作 1• 文字 - @圖片地圖
2• 文字 - @日期時間

預約 / 公開 待設定

圖片地圖及日期時間-功能 Demo 1



圖片地圖及日期時間-功能 Demo 2

