

使用 LINE Bot SDK

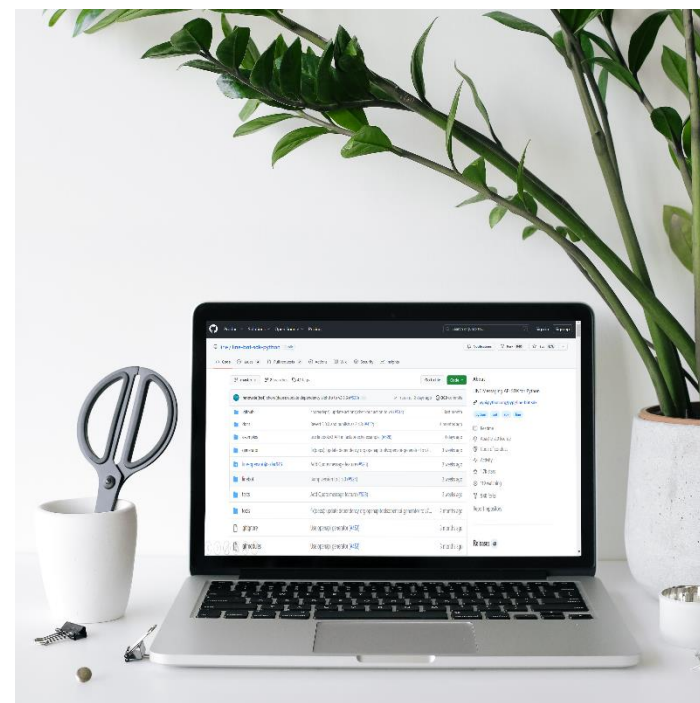
介紹與安裝 LINE Bot SDK





LINE Bot SDK

- ① 要在Python中，使用LINE Messaging API 的功能，一定要安裝LINE Bot SDK。
- ② LINE Bot SDK 是為開發者設計的Python 軟體開發工具。
- ③ 有簡單易用的函式庫，建立執行LINE Bot，與LINE平台進行溝通。





啟動「pip」(軟體包管理系統) 的程序



找到搜尋工具



輸入 cmd (命令提示字元)





命令提示字元畫面



命令提示字元



Microsoft Windows [版本 10.0.22621.2361]
(c) Microsoft Corporation. 著作權所有，並保留一切權利。

C:\Users\schoo>



安裝 Line-bot-SDK

命令提示字元

Microsoft Windows [版本 10.0.22621.2428]
(c) Microsoft Corporation. 著作權所有，並保留一切權利。

C:\Users\schoo>pip install line-bot-sdk 輸入指令「**pip install line-bot-sdk**」，按 Enter



安裝 Line-bot-SDK

命令提示字元

×

+

▼

—

□

×

Microsoft Windows [版本 10.0.22621.2361]

(c) Microsoft Corporation. 著作權所有，並保留一切權利。

C:\Users\schoo>pip install line-bot-sdk

Collecting line-bot-sdk

Obtaining dependency information for line-bot-sdk from https://files.pythonhosted.org/packages/a7/a2/1b2d2790183debe62063d7844b7399c4f36a931fb2bf65b7c1059498abe3/line_bot_sdk-3.5.0-py2.py3-none-any.whl.metadata

Using cached line_bot_sdk-3.5.0-py2.py3-none-any.whl.metadata (11 kB)

Collecting requests==2.31.0 (from line-bot-sdk)

Obtaining dependency information for requests==2.31.0 from <https://files.pythonhosted.org/packages/70/8e/0e2d847013cb52cd35b38c009bb167a1a26b2ce6cd6965bf26b47bc0bf44/requests-2.31.0-py3-none-any.whl.metadata>

Using cached requests-2.31.0-py3-none-any.whl.metadata (4.6 kB)

Collecting aiohttp==3.8.5 (from line-bot-sdk)

Using cached aiohttp-3.8.5.tar.gz (7.4 MB)

Installing build dependencies ... done

Getting requirements to build wheel ... done

Installing backend dependencies ... done

Preparing metadata (pyproject.toml) ... done

Collecting future (from line-bot-sdk)

Using cached future-0.18.3-py3-none-any.whl

Collecting pydantic<3,>=2.0.3 (from line-bot-sdk)

Obtaining dependency information for pydantic<3,>=2.0.3 from <https://files.pythonhosted.org/packages/73/66/0a72c9fcde42e5650c8d8d5c5c1873b9a3893018020c77ca8eb62708b923/pydantic-2.4.2-py3-none-any.whl.metadata>

Using cached pydantic-2.4.2-py3-none-any.whl.metadata (158 kB)

Collecting aenum>=3.1.11 (from line-bot-sdk)

Obtaining dependency information for aenum>=3.1.11 from <https://files.pythonhosted.org/packages/d0/fa/ca0c66b388624ba9dbbf35aab3a9f326bfd5e56a7237fe8f1b600da6864/aenum-3.1.15-py3-none-any.whl.metadata>

Using cached aenum-3.1.15-py3-none-any.whl.metadata (3.7 kB)

Collecting python-dateutil>=2.5.3 (from line-bot-sdk)

Using cached python_dateutil-2.8.2-py2.py3-none-any.whl (247 kB)



確認已經成功安裝 Line bot sdk

```
命令提示字元
Microsoft Windows [版本 10.0.22621.2361]
(c) Microsoft Corporation. 著作權所有，並保留一切權利。

C:\Users\schoo>pip list 輸入「pip list」
Package            Version
-----
aiohttp             3.8.4
aiosignal           1.3.1
async-timeout       4.0.2
attrs               22.2.0
blinker             1.6.2
certifi             2022.12.7
charset-normalizer  3.1.0
click               8.1.3
colorama            0.4.6
Flask               2.3.1
frozenlist          1.3.3
future              0.18.3
idna                3.4
itsdangerous        2.1.2
Jinja2              3.1.2
line-bot-sdk        2.4.2 找到line-bot-SDK表示成功
lxml                4.9.2
MarkupSafe          2.1.2
multidict           6.0.4
numpy               1.24.2
openai              0.27.4
pandas              2.0.0
pip                 23.0.1
python-dateutil     2.8.2
```


使用 LINE Bot SDK

設計 Python 串接程式





Python 串接程式

在瀏覽器中輸入網址：

<https://github.com/line/line-bot-sdk-python/tree/master/examples>

ff01-220-141-10-231.ngrok.io x | </> Getting started with LINE Logi x | line-bot-sdk-python/examples x | line-bot-sdk-python/examples x +

github.com/line/line-bot-sdk-python/tree/master/examples

Product Solutions Open Source Pricing

Search or jump to... / Sign in Sign up

line / line-bot-sdk-python Public line-bot-sdk 開發網站

Code Issues 4 Pull requests 2 Actions Wiki Security Insights

Files

master

Go to file

> .github

> docs

> examples

> aiohttp-echo

> fastapi-echo

> flask-echo

> flask-kitchensink

> rich-menu

line-bot-sdk-python / examples /

tokuhirom use linebot.v3 API in fastapi-echo example. (#528) ✓ 1b3638e · 4 days ago History

Name	Last commit message	Last commit date
..		
aiohttp-echo	Use openapi generator (#453)	
fastapi-echo	use linebot.v3 API in fastapi-echo example. (#528)	4 days ago
flask-echo	Use openapi generator (#453)	3 months ago
flask-kitchensink	Use openapi generator (#453)	3 months ago
rich-menu	Use openapi generator (#453)	3 months ago



Flask 串接 LINE Bot 的範例程式碼

<> Code Issues 4 Pull requests 2 Actions Wiki Security Insights

Files

master

Go to file

- > .github
- > docs
- ▼ examples
 - > aiohttp-echo
 - > fastapi-echo
 - ▼ flask-echo
 - README.md
 - app.py
 - app_with_handler.py
 - requirements.txt
 - > flask-kitchensink
 - > rich-menu
 - > simple-server-echo
 - > generator

line-bot-sdk-python / examples / flask-echo /

Yang-33 Use openapi generator (#453) ✓

c9aa518 · 3 months ago History

Name	Last commit message	Last commit date
..		
README.md	Fix documentation issues	7 years ago
app.py	Use openapi generator (#453)	
app_with_handler.py	Use openapi generator (#453)	3 months ago
requirements.txt	Rename package and modify related code. Rewrite README.md. Fix simple...	7 years ago



LINE Messaging API SDK for Python

在瀏覽器中輸入網址：
<https://github.com/line/line-bot-sdk-python>



Product ▾ Solutions ▾ Open Source ▾ Pricing

Sign in Sign up

line / line-bot-sdk-python Public

Notifications Fork 940 Star 1.7k

<> Code Issues 4 Pull requests 2 Actions Wiki Security Insights

master 2 branches 42 tags Go to file Code

renovate[bot]	chore(deps): update dependency aiohttp to v3.8.6 (#530)	4a16f1b 2 days ago 369 commits
.github	chore(deps): update actions/checkout action to v4 (#514)	last month
docs	Revert 3.0.0 and publish as 2.4.3 (#452)	4 months ago
examples	use linebot.v3 API in fastapi-echo example. (#528)	4 days ago
generator	fix(deps): update dependency org.openapitools:openapi-generator to v7...	3 weeks ago
line-openapi @ cdac946	Add Quote message feature (#523)	3 weeks ago
linebot	Bump version to 3.5.0 (#524)	3 weeks ago
tests	Add Quote message feature (#523)	3 weeks ago
tools	fix(deps): update dependency org.openapitools:openapi-generator to v7...	2 months ago
.gitignore	Use openapi generator (#453)	3 months ago
.gitmodules	Use openapi generator (#453)	3 months ago

About

LINE Messaging API SDK for Python

pypi.python.org/pypi/line-bot-sdk

python bot sdk line

- Readme
- Apache-2.0 license
- Code of conduct
- Activity
- 1.7k stars
- 119 watching
- 940 forks
- Report repository

Releases 40





LINE Messaging API SDK for Python 介紹

☰ README.rst

LINE Messaging API SDK for Python [↗](#)

pypi package 3.5.0

SDK of the LINE Messaging API for Python.

Introduction [↗](#) 簡介

The LINE Messaging API SDK for Python makes it easy to develop bots using LINE Messaging API, and you can create a sample bot within minutes.

Documentation [↗](#) 目前只有 英文 & 日文

See the official API documentation for more information

English: <https://developers.line.biz/en/docs/messaging-api/overview/>

Japanese: <https://developers.line.biz/ja/docs/messaging-api/overview/>

Requirements [↗](#)

- Python >= 3.8

Installation [↗](#)

Languages





LINE Messaging API SDK for Python 介紹

☰ README.rst

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

API [↗](#) 主要是串接 `line_bot_api.reply_message` 的項目

See [linebot/v3/messaging/docs](https://github.com/line/linebot/blob/master/docs/messaging/docs/*.md). Other docs are in `linebot/v3/<feature>/docs/*.md`.

Webhook [↗](#)

WebhookParser [↗](#)

※ You can use WebhookParser

`__init__(self, channel_secret)` [↗](#)

```
parser = linebot.v3.WebhookParser('YOUR_CHANNEL_SECRET')
```



`parse(self, body, signature, as_payload=False)` [↗](#)

Parses the webhook body, and returns a list of Event objects or a WebhookPayload object (depending on as_payload).
If the signature does NOT match, `InvalidSignatureError` is raised.

```
events = parser.parse(body, signature)
```





LINE Messaging API SDK for Python 介紹

☰ README.rst

```
handler.handle(body, signature)
```



`add(self, event, message=None)` [↗](#)

Add a **handler** method by using this decorator.

```
@handler.add(MessageEvent, message=TextMessage)
def handle_message(event):
    line_bot_api.reply_message(
        ReplyMessageRequest(
            reply_token=event.reply_token,
            messages=[TextMessage(text=event.message.text)]
        )
    )
```

API裡面會用到的程式



When the event is an instance of `MessageEvent` and `event.message` is an instance of `TextMessage`, this handler method is called.

```
@handler.add(MessageEvent)
def handle_message(event, destination):
    # do something
```



If the arity of the handler method is more than one, a `destination` property in a webhook request is passed to it as the second argument.



Python 串接程式需要用到的兩個檔案 (詳見教材檔案)

```
#!/usr/bin/env python
# coding: utf-8
# @Date: 2017-07-10 10:10:10
# @Author: 102461821@qq.com
# @Description: LINE Bot SDK 原始碼程式
```



✔ linebot_echo 串接 Python 的主要程式，有 Flask Web Service & LINE Bot SDK 原始碼程式



✔ config 設定 LINE Developers 開發者帳號使用金鑰資料檔的基本資訊

使用 LINE Bot SDK

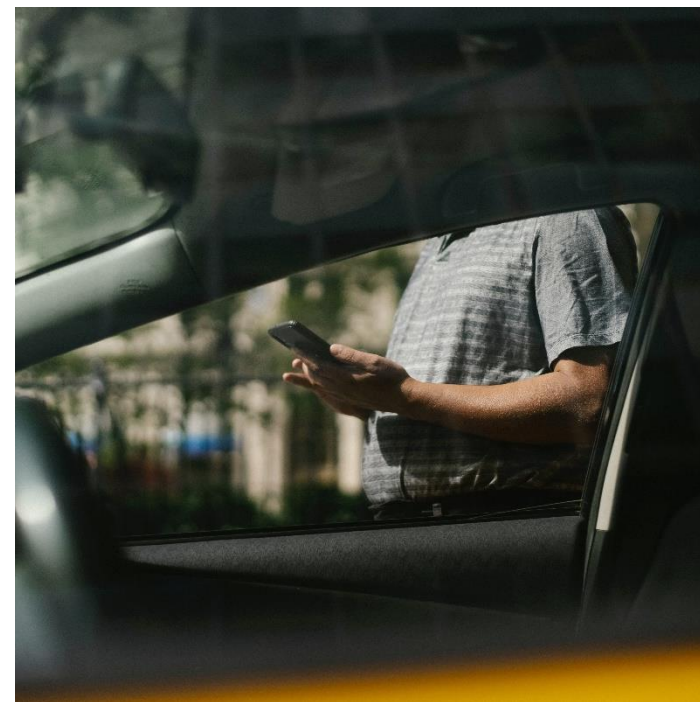
回話 LINE Bot 程式開發





常用的 LINE Bot SDK 程式 (1) linebot.LineBotApi

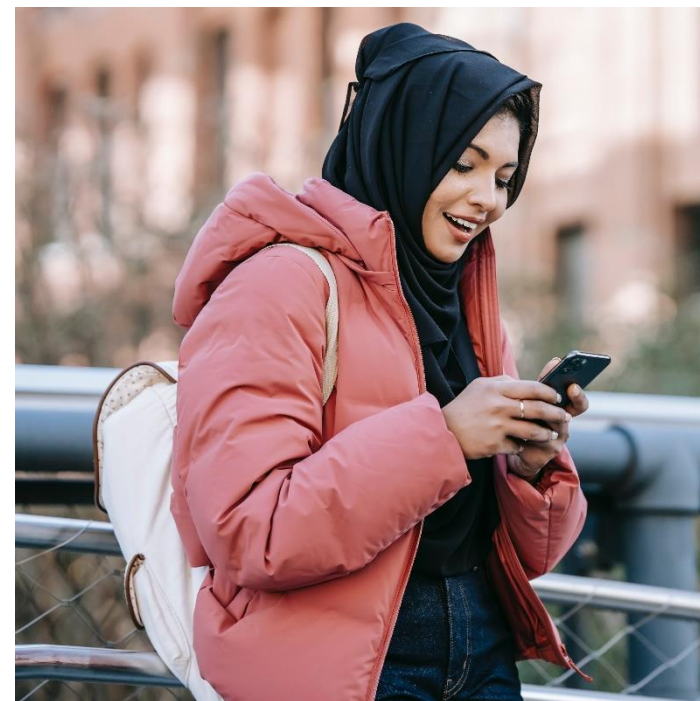
- ① 向LINE平台發送請求的函式功能
- ② 開發者可以向LINE平台發送各類型的請求
「文字訊息」，「圖片」，「影片」，「音訊」等。
- ③ 透過這個功能，開發者可以為用戶提供豐富的服務體驗。





常用的 LINE Bot SDK 程式 (2) linebot.WebhookHandler

- ① 處理 Line Bot 的 Webhook。
- ② Line Bot 收到用戶訊息時，平台會向 Webhook URL 發送一個 HTTP POST 請求。
- ③ linebot.WebhookHandler 解析請求之後，觸發相應的函式，Line Bot 做出相應的回應。





主要使用以下三個函式庫與相關功能

File Edit Format Run Options Window Help

```
# 使用系統和 config 檔案的函式庫功能
import os
import configparser

# 使用 Flask 的函式庫功能
from flask import Flask, request, abort

# 使用 LINE Bot SDK 的函式庫功能
from linebot import LineBotApi, WebhookHandler
from linebot.exceptions import InvalidSignatureError
from linebot.models import MessageEvent, TextMessage, TextSendMessage, FlexSendMessage
```

可參閱教材程式碼

```
# 設定讀入 config.ini 檔案
config = configparser.ConfigParser()
config.read('config.ini')

# Flask Web Service 啟用
app = Flask(__name__)

line_bot_api = LineBotApi(config.get('line-bot',
                                     'channel_access_token'))
handler = WebhookHandler(config.get('line-bot',
                                    'channel_secret'))

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

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

    # handle webhook body
    try:
        print(body, signature)
        handler.handle(body, signature)
```

Ln: 1 Col: 0

◆ 三項函式庫

- ① **os,configparser** : 讀 **config ini** 檔案
- ② **Flask** : 將**Flask Web Service** 功能放入程式中
- ③ **Line Bot SDK** : 傳送訊息 , **WebhookHandler** 等

① 使用系統和 **config** 檔案的函式庫功能

```
import os
import configparser
```

② 使用 **Flask** 的函式庫功能

```
from flask import Flask, request, abort
```

③ 使用 **LINE Bot SDK** 的函式庫功能

```
from linebot import LineBotApi, WebhookHandler
from linebot.exceptions import InvalidSignatureError
from linebot.models import MessageEvent, TextMessage,
TextSendMessage, FlexSendMessage
```



讀取 config.ini 資料需要的函式

File Edit Format Run Options Window Help

使用系統和 config 檔案的函式庫功能

```
import os
import configparser
```

使用 Flask 的函式庫功能

```
from flask import Flask, request, abort
```

使用 LINE Bot SDK 的函式庫功能

```
from linebot import LineBotApi, WebhookHandler
from linebot.exceptions import InvalidSignatureError
from linebot.models import MessageEvent, TextMessage, TextSendMessage, FlexS
```

設定讀入 config.ini 檔案

```
config = configparser.ConfigParser()
config.read('config.ini')
```

Flask Web Service 啟用

```
app = Flask(__name__)
```

```
line_bot_api = LineBotApi(config.get('line-bot',
                                     'channel_access_token'))
handler = WebhookHandler(config.get('line-bot',
                                     'channel_secret'))
```

```
@app.route("/callback", methods=['POST'])
```

```
def callback():
```

```
    # get X-Line-Signature header value
    signature = request.headers['X-Line-Signature']
```

```
    # get request body as text
```

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

```
    # handle webhook body
```

```
    try:
```

```
        print(body, signature)
        handler.handle(body, signature)
```

設定讀入 config.ini 檔案

```
config = configparser.ConfigParser()
config.read('config.ini') 要寫的檔案名稱
```

Flask Web Service 啟用

```
app = Flask(__name__)  User ID
```

```
line_bot_api = LineBotApi(config.get('line-bot',
                                     'channel_access_token'))
handler = WebhookHandler(config.get('line-bot',
                                     'channel_secret'))
```

提供 'channel_access_token' 啟動 Line Bot Api

Ln: 1 Col: 0



讀取 config.ini 資料

名稱	狀態	修改日期	類型	大小
linebot_echo	✓	2023/10/9 下午 05:16	Python File	2 KB
config			組態設定	1 KB



開啟檔案



記事本

config

檔案 編輯 檢視

```
[line-bot]
channel_access_token =
channel_secret =
line_user_id =
```

需要自己填入 Line Bot Channel 資訊

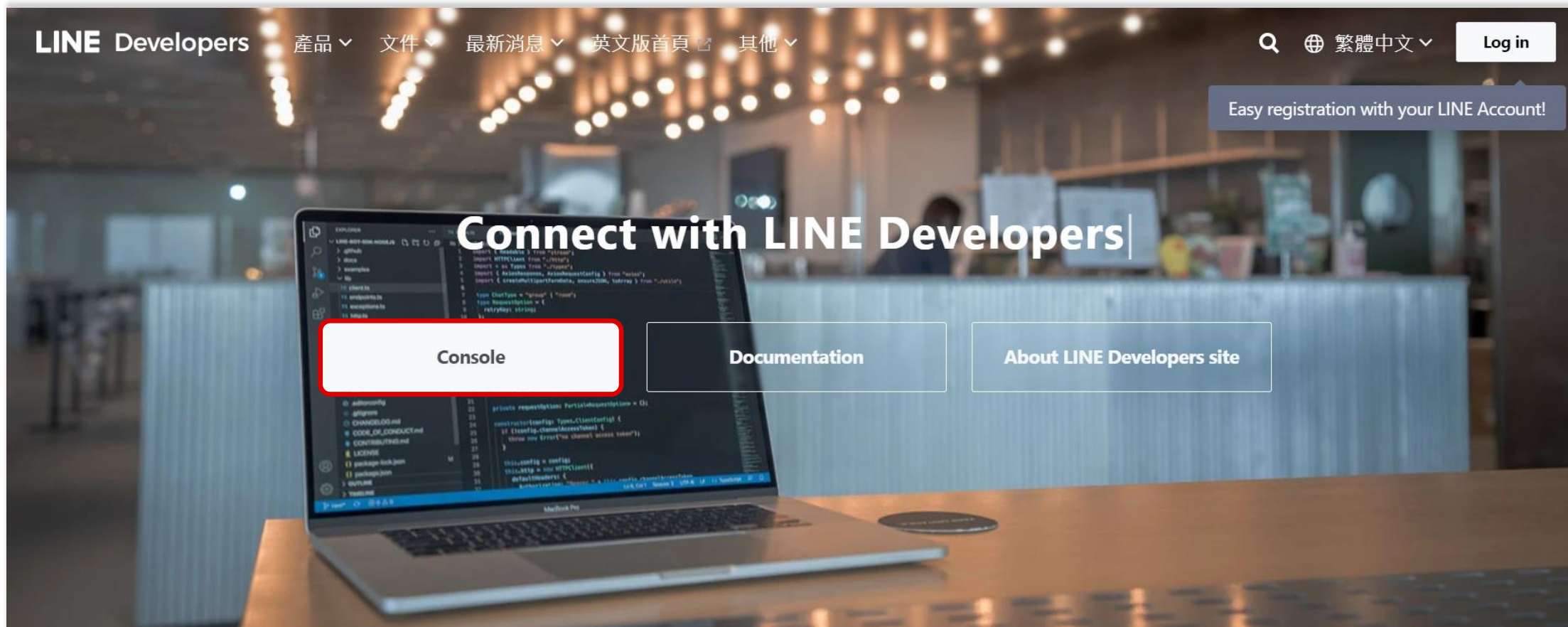
- ① channel_access_token =
- ② channel_secret =
- ③ line_user_id =

需要在 Line Developers 取得 Channel Messaging API 金鑰資料



登入 LINE Developers 開發者後台

登入網址：<https://developers.line.biz/zh-hant/>



Products



進入 LINE Developers 的帳號

LINE Developers

[News](#)

[Products](#)

[Documentation](#)

[FAQ](#)

[Glossary](#)

[Community](#)

[Blog](#)



Console home

Providers

Search...

Admin

Creator Training School

Tools

Support

TOP

Recently visited channels

Admin



創創小助手

 Messaging API

點進Channel

Providers (1)

Create





Basic settings

LINE Developers

[News](#)

[Products](#)

[Documentation](#)

[FAQ](#)

[Glossary](#)

[Community](#)

[Blog](#)



Console home

Providers

Search...

Admin

Creator Training School

Tools

Support

TOP > Creator Training School > 創創小助手 > Basic settings



創創小助手

Admin

Messaging API

Basic settings

Messaging API

LIFF

Security

Statistics

Roles

點進 Basic setting 設定

Basic settings

Basic information

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



channel_secret 之設定

LINE Developers

News Products Documentation FAQ Glossary

Console home

Providers

Search...

Admin

Creator Training School

Tools

Support

TOP > Creator Training School > 創創小助手

Permissions ?

PROFILE

Channel secret ?

ae9723e685e41c41e97527c74292978a



複製

Issue

Assertion Signing
Key ?

Register a public key

Delete this channel

Delete

```
[line-bot]
channel_access_token =
channel_secret = 
line_user_id =
```

需要自己填入 Line Bot Channel 資訊

① channel_access_token =

② channel_secret =

③ line_user_id =





line_user_id 之設定

LINE Developers

News Products Documentation FAQ Glossary Community

Console home

Providers

Search...

Admin

Creator Training School

Tools

Support

TOP > Creator Training School > 創創小助手 > Messaging



創創小助手

Admin

Messaging API

Basic settings

Messaging API

LIFF

Security

Messaging API settings

Bot information

Bot basic ID

@972wr



QR code



檔案 編輯 檢視

config



```
[line-bot]
channel_access_token =
channel_secret =
line_user_id = 
```

需要自己填入 Line Bot Channel 資訊

① channel_access_token =

② channel_secret =

③ line_user_id =



Family sites



English





Channel Access Token 之產生

LINE Developers

[News](#) [Products](#) [Documentation](#) [FAQ](#) [Glossary](#) [Community](#) [Blog](#)



Console home

Providers

Admin

Creator Training School

Tools

Support

TOP > Creator Training School > 創創小助手 > Messaging API

Allow bot to join group chats Disabled [Edit](#)

Auto-reply messages Enabled [Edit](#)

Greeting messages Enabled [Edit](#)

Channel access token

Channel access token (long-lived)

[Issue](#) 點擊「Issue」產生加密金鑰，用以正式串接「Python」程式。



channel_access_token 之設定

Products Documentation FAQ Glossary Community [Blog](#)

TOP > Creator Training School > 創創小助手 > Messaging API

Allow bot to join group chats [?](#) Disabled

Auto-reply messages [?](#) Enabled

Greeting messages [?](#) Enabled

Channel access token

Channel access token (long-lived) [?](#)

FwhQe0WAp5jBfgshgEfEmSxF7ewufgrLYxh+23vvLU9eTJBtUhmtla2bCFazUXB+WHSNEH4Th/lhSokU3yqgD93IG59hssu37pVh0Gi/dzMwouv4wiSz4Kc
arRoRIAPnpD4GSQRN3pVVwYIGwD5oRwdB04t89/1O/w1cDnyilFU=



Reissue

config

檔案 編輯 檢視

```
[line-bot]
channel_access_token = 
channel_secret = 
line_user_id =
```

需要自己填入 Line Bot Channel 資訊

- ① channel_access_token =
- ② channel_secret =
- ③ line_user_id =

複製此加密金鑰



Line Webhook 的主要程式項目

```
File Edit Format Run Options Window Help
# 使用系統和 config 檔案的函式庫功能
import os
import configparser

# 使用 Flask 的函式庫功能
from flask import Flask, request, abort

# 使用 LINE Bot SDK 的函式庫功能
from linebot import LineBotApi, WebhookHandler
from linebot.exceptions import InvalidSignatureError
from linebot.models import MessageEvent, TextMessage, TextSendMessage, FlexSendMessage

# 設定讀入 config.ini 檔案
config = configparser.ConfigParser()
config.read('config.ini')

# Flask Web Service 啟用
app = Flask(__name__)

line_bot_api = LineBotApi(config.get('line-bot',
                                     'channel_access_token'))
handler = WebhookHandler(config.get('line-bot',
                                    'channel_secret'))

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

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

    # handle webhook body
    try:
        print(body, signature)
        handler.handle(body, signature)
```

Webhook 的主要程式，在此處是設定好，固定的程式，不需要修改或更動。



接收，回覆到Channel的程式

```
File Edit Format Run Options Window Help

line_bot_api = LineBotApi(config.get('line-bot',
                                     'channel_access_token'))
handler = WebhookHandler(config.get('line-bot',
                                    'channel_secret'))

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

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

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

    return 'OK'

@handler.add(MessageEvent, message=TextMessage)
def handle_message(event):
    # 從 Line 傳入的訊息
    msg = event.message.text

    # 回傳相同文字內容
    line_bot_api.reply_message(event.reply_token, TextSendMessage(msg))

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

Ln: 1 Col: 0

回話機器人程式用 **LINE Bot SDK** 中的 **TextMessage** 文字訊息，傳遞文字資料。

- ① 回應的訊息，寫在 **handle_message** 的 **event** 裡面。
- ② **handle_message** 會讀取 **event** 的資料。
- ③ 擷取出來的 **msg** 資料，就是 **event.message.text**。
- ④ 把 **Python** 裡面的文字，透過 **TextSendMessage(msg)**，變成 **Line Bot** 裡面的文字。



後續繼續試做 LINE BOT 聊天機器人的預計成果



Flask Web Service 和 LINE Bot SDK 串接後，在 LINE BOT 機器人中，使用者輸入文字，程式會回覆相同的文字。

使用 LINE Bot SDK

課後練習





課後練習

- ① 請安裝 Line Bot SDK，上傳安裝成功截圖。
- ② 請回答 Flask Web Service 啟用時，`config.ini` 資料需要的程式。
- ③ 請練習進行 `config.ini`
`channel_access_token =`
`channel_secret =`
`line_user_id =`
之設定。

