

# Python 機器人

## LINE篇

Created by 戴嘉熙 【小孫學堂】

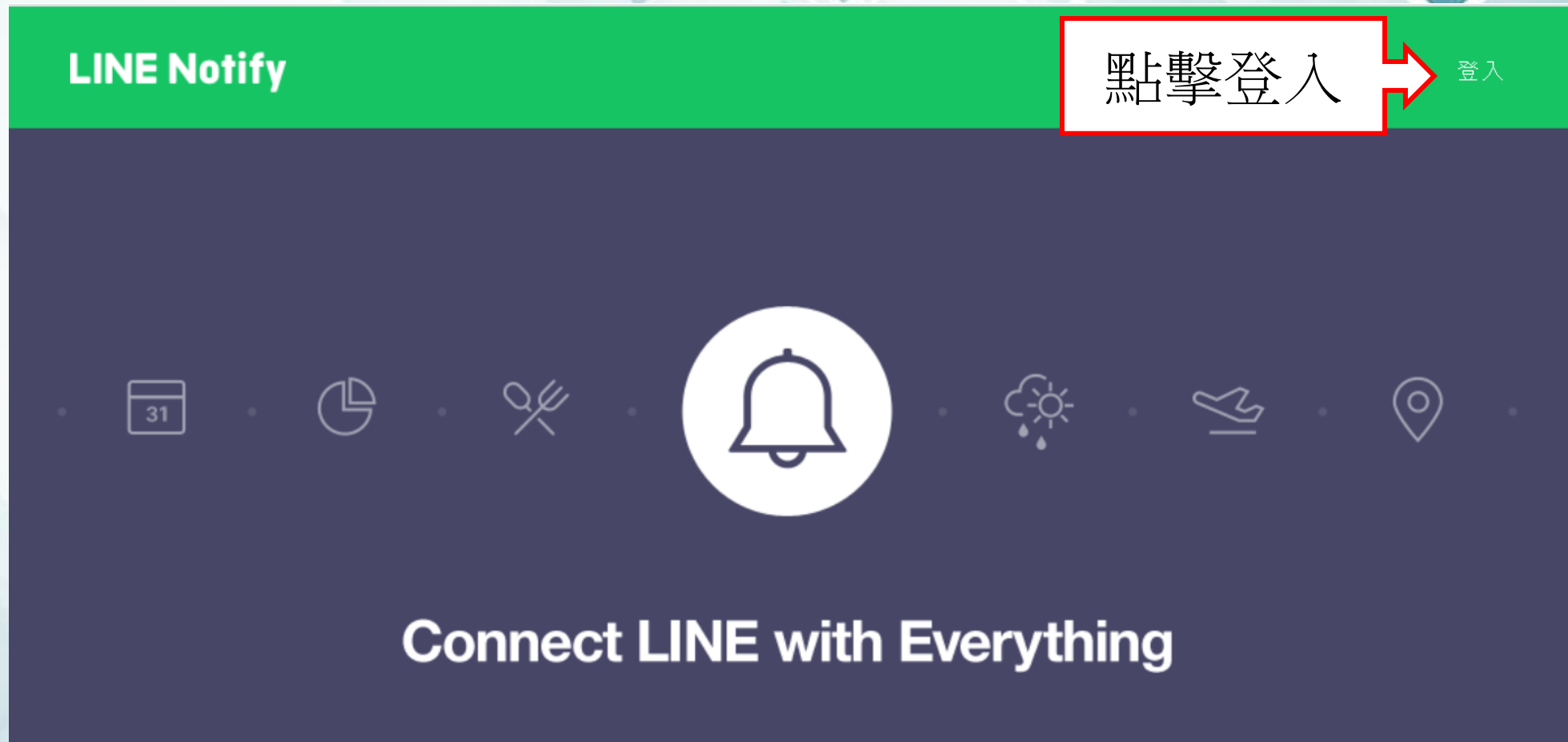
LINE機器人

Notify

Created by 戴嘉熙

# 建立LINE Notify

[https://notify-bot.line.me/zh\\_TW/](https://notify-bot.line.me/zh_TW/)



# 發行權杖

1. 點擊個人頁面

Alan\_Tai 小戴 ▾

個人頁面

管理登錄服務

登出

## 發行存取權杖(開發人員用)

若使用個人存取權杖, 不須登錄網站服務, 即可設定通知。

發行權杖

2. 點擊發行權杖

LINE Notify API Document



# 設定窗口

## 發行權杖

請填寫權杖名稱(將於傳送提醒時顯示)

Python\_Notify

1.設定Notify名稱

請選擇您要接收通知的聊天室。

Search by group name



透過1對1聊天接收LINE Notify的通知



※若公開個人存取權杖, 第三者將能取得您所連動的聊天室名稱及個人資料上的姓名。

發行

2.選取接收通知的聊天室

3.點選發行

已發行的權杖如下。

STWxusgFWBM5rRcdJb3API9hltRWXlmyBixg7i

若離開此頁面, 將不會再顯示新發行的權杖。離開頁面前, 請先複製權杖。

4.複製權杖

複製

關閉

!!!注意!!!  
權杖只會出現一次  
建議另外儲存

# 設定TOKEN

```
1 import requests
2
3 # LINE Notify 權杖
4 token = '貼上TOKEN'
5
6 # 要發送的訊息
7 message = '發送訊息'
8
9 # 要傳送的圖片檔案
10 image = open("C:/Users/justw/Desktop/123.png", 'rb')
11 files = { 'imageFile': image }
12
13 # HTTP 標頭參數與資料
14 headers = { "Authorization": "Bearer " + token }
15 data = { 'message': message }
16
17 # 以 requests 發送 POST 請求
18 requests.post("https://notify-api.line.me/api/notify",
19               headers = headers, data = data, files = files)
```

1.貼上Token

2.設定文字訊息

3.設定圖片訊息

確認完畢  
開始運行

LINE機器人

創立Channel

Created by 戴嘉熙

# 步驟流程

**1.建立LINE developers APP**

[\*\*https://developers.line.biz/zh-hant/\*\*](https://developers.line.biz/zh-hant/)

**2.使用ngrok建立外網通道**

[\*\*https://ngrok.com/\*\*](https://ngrok.com/)

**3.使用Python架設LINE Bot**

**4.完成LINE Webhooks串聯**



# 建立LINE developers APP

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

The screenshot shows the LINE Developers website interface. At the top, the header includes 'LINE Developers' and a dropdown menu labeled '產品' (Products). A red box with the text '1. 點擊產品' (Click Product) and an arrow points to the '產品' dropdown. Below the header, a sidebar lists various products: 'Products top', 'LINE Login', 'Messaging API', 'LINE MINI App', 'Bot Designer', 'LINE Things', 'LINE Pay', and 'Blockchain Service'. A second red box with the text '2. 點擊 Messging API' (Click Messaging API) and an arrow points to the 'Messaging API' link in the sidebar. The main content area features three large cards: 'LINE Login' (with a laptop icon), 'Messaging API' (with a smartphone icon), and 'LINE MINI App' (with a 'MINI' logo icon). Each card contains a brief description, a '更多資訊' (More Information) button, and a '技術文件' (Technical Documentation) link.

**LINE Developers** 產品 ▾ 1. 點擊產品 其他 ▾

**Products top**

- LINE Login
- Messaging API
- LINE MINI App
- Bot Designer
- LINE Things
- LINE Pay
- Blockchain Service

**2. 點擊 Messging API**

**LINE Login**  
讓 LINE 用戶輕鬆登入並連接開發者的服務。  
更多資訊  
技術文件

**Messaging API**  
開啟與 LINE 用戶的雙向溝通。  
更多資訊  
技術文件

**LINE MINI App**  
Connect your service and users with LINE MINI App.  
更多資訊

# 設定APP資料1

## Create a new channel

Channel type

Messaging API

✓ Don't leave this empty

Provider

戴嘉熙

✓ Don't leave this empty

Company or  
owner's country or  
region <sup>?</sup>

Taiwan

Corporations should select the

✓ Don't leave this empty

Channel name

LINE機器人

Note: The channel name can't be changed for seven days.

- ✓ Don't leave this empty
- ✓ Don't use special characters (4-byte Unicode)
- ✓ Enter no more than 20 characters

Channel description

測試LINE 機器人

- ✓ Don't leave this empty
- ✓ Don't use special characters (4-byte Unicode)
- ✓ Enter no more than 500 characters

Category

廣播

✓ Don't leave this empty

# 設定APP資料2

Subcategory

廣播 ( 其他 )

✓ Don't leave this empty

Email address ?

- ✓ Don't leave this empty
- ✓ Enter a valid email address
- ✓ Enter no more than 100 characters

Privacy policy URL

optional

Enter privacy policy URL

- ✓ Enter a valid HTTPS URL
- ✓ Enter no more than 500 characters

兩項條款皆須勾選

- ☒ I have read and agree to the [LINE Official Account Terms of Use](#)
- ☒ I have read and agree to the [LINE Official Account API Terms of Use](#)

✓ Select the checkbox after reading the related document

Create

填寫完畢後點選創立

# 設定Token

TOP > 戴嘉熙 > AUTO\_BOT\_TEST > Messaging API



**AUTO\_BOT\_TEST**

Admin

Messaging API

Basic settings

Messaging API

1.點選Messaging API

## Messaging API settings

2.創建並取得token  
可點選Reissue更新

### Channel access token

Channel access token (long-lived) ?

[Redacted token value]



Reissue



# 更新程式碼Token

```
1 from flask import Flask, request
2 #https://github.com/line/line-bot-sdk-python
3 from linebot import LineBotApi, WebhookHandler
4 from linebot.models import TextMessage
5 token="貼上TOKEN" ← 貼上最新TOKEN
6 app = Flask(__name__)
7
8 # LINE 聊天機器人的基本資料
9 line_bot = LineBotApi(token)
10
11 @app.route("/", methods=['POST'])
12 def verify():
13     data = request.get_json()
14     print(data)
15     return 'OK', 200
16
17 if __name__ == "__main__":
18     app.run(port=3000)
```

# 註冊並登入ngrok

<https://ngrok.com/>

1.註冊並登入帳號

[How it works](#) [Pricing](#) [Enterprise solutions](#) [Docs](#) [Download ↓](#)

[Login](#)

[Sign up](#)

🔔 [Getting Started](#) ^

[Setup & Installation](#)

[Your Authtoken](#)

[Tutorials](#)

🌐 [Endpoints](#) v

↔ [Tunnel Agents](#) v

🛡 [TLS](#) v

⚡ [Events](#)

2.點擊進入設定及  
安裝頁面

# 下載ngrok

選擇與自己系統相容之版本  
並將檔案解壓縮於桌面

## Download ngrok

ngrok is easy to install. Download a single binary with zero run-time dependencies.

 [Download for Windows](#)



Mac OS  
Mac OS (ARM64)



Windows  
Windows (32-Bit)



Linux  
Linux (32-Bit)  
Linux (ARM)  
Linux (ARM64)



FreeBSD  
FreeBSD (32-Bit)



# 開啟ngrok，驗證Token

## 2. Connect your account

Running this command will add your authtoken to the default `ngrok.yml` configuration file. This will grant you access to more features and longer session times. Running tunnels will be listed on the [status page](#) of the dashboard.

```
$ ./ngrok authtoken 21h9dKusXXXXXXXXXXXXXXXXXXXXqdGkf5yDPJLJ
```

在ngrok輸入設定及安裝頁面第二點的Token

**ngrok authtoken XXXXXXXXXXXXXXXXXXXXXXXXXXXX**

```
C:\Users\justw\Desktop>ngrok authtoken 21h9dKus4X87hXXXXXXXXXXXXXXXXXXXXqdGkf5yDPJLJ
Authtoken saved to configuration file: C:\Users\justw/.ngrok2/ngrok.yml
```



# 建立通道

在ngrok輸入指令建立通道

ngrok http XXXX (XXXX可自行編號)

```
C:\Users\justw\Desktop\ngrok.exe - ngrok http 3000
ngrok by @inconshreveable

Session Status      online
Account             [REDACTED]
Version             2.3.40
Region              United States (us)
Web Interface        http://127.0.0.1:4040
Forwarding           http://c40e-123-194-247-253.ngrok.io -> http://localhost:3000
Forwarding           https://c40e-123-194-247-253.ngrok.io -> http://localhost:3000

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

# 測試連線

```
C:\Users\justw\Desktop\nngrok.exe - ngrok http 3000
ngrok by @inconshreveable

Session Status      online
Account
Version
Region
Web Interface      http://127.0.0.1:4040
Forwarding          http://c40e-123-194-247-253.ngrok.io -> http://localhost:3000
Forwarding          https://c40e-123-194-247-253.ngrok.io -> http://localhost:3000

Connections
  ttl    opn    rt1    rt5    p50    p90
    1      0    0.01   0.00   1.32   1.32

HTTP Requests
-----
GET /
```

於瀏覽器輸入此網址測試連線

502 Bad Gateway

成功取得連線的回應

# 瀏覽器網頁端



## 你要瀏覽的是詐騙網站

攻擊者可能會試圖透過 **c40e-123-194-247-253.ngrok.io** 誘使你做一些危險行為，例如安裝軟體或提供個人資料 (包括密碼、電話號碼或信用卡資料)。瞭解詳情

## 1. 點選詳細資訊

隱藏詳細資訊

Google 安全瀏覽功能最近在 **c40e-123-194-247-253.ngrok.io** 網站會偽裝成其他網站，藉此騙取你的資訊。

您可以回報偵測問題。或者在您瞭解安全性風險後，仍然可以前往這個不安全的網站。

## 2. 點選前往這個不安全的網站

3. 目前只有通道，網頁另一端尚未架設，所以此頁面是正常情況

## Failed to complete tunnel connection

The connection to **https://c40e-123-194-247-253.ngrok.io** was successfully tunneled to your ngrok client, but the client failed to establish a connection to the local address **localhost:3000**.

Make sure that a web service is running on **localhost:3000** and that it is a valid address.

The error encountered was: **dial tcp [::1]:3000: connectex: No connection could be made because the target machine actively refused it.**

# 確認通道

C:\Users\justw\Desktop\ngrok.exe - ngrok http 3000

ngrok by @inconshreveable

Session Status

online

Account

[REDACTED] (Plan: Free)

Update

update available (version 2.3.41, Ctrl-U to update)

Version

2.3.40

Region

United Sta

Web Interface

http://127.0.0.1:4040

Forwarding

http://95a7-123-194-247-253.ngrok-free.app -> http://localhost:3000

Forwarding

https://95a7-123-194-247-253.ngrok-free.app -> http://localhost:3000

Connections

ttl	opn	rt1	rt5	p50	p90
9	0	0.00	0.00	0.32	0.32

1.複製網址

2.確認Port編號



# 確認Port編號

```
1 from flask import Flask, request
2 #https://github.com/line/line-bot-sdk-python
3 from linebot import LineBotApi, WebhookHandler
4 from linebot.models import FlexSendMessage, TextMessage
5 token="貼上TOKEN"
6 app = Flask(__name__)
7
8 # LINE 聊天機器人的基本資料
9 line_bot = LineBotApi(token)
10
11 @app.route("/", methods=['POST'])
12 def verify():
13     data = request.get_json()
14     print(data)
15     return 'OK', 200
16
17 if __name__ == "__main__":
18     app.run(port=3000)
```

須與NGROK通道一致

rl-U to update)

ee.app -> http://localhost:3000  
ree.app -> http://localhost:3000

p90  
0.32

確認完畢  
開始運行

# 驗證Webhook

Basic settings

Messaging API

1.點選Messaging API

## Webhook settings

Webhook URL ?

https://95a7-123-194-247-253.ngrok-free.app

3.點選驗證

Verify

Edit

2.貼上NGROK通道網址

Use webhook ?



4.開啟功能

Success

OK

# 關閉自動回應1

Basic settings

Messaging API

1.點選Messaging API

## LINE Official Account features

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

Allow bot to join  
group chats (?)

Disabled

Edit

Auto-reply  
messages (?)

Disabled

Edit

2.將自動回應  
設定為Disabled

# 關閉自動回應2

## 回應功能

- 聊天 ☐   
可透過聊天與好友互動。
- 加入好友的歡迎訊息 ☒   
當用戶將本帳號加為好友時，可自動傳送訊息。  
[開啟「加入好友的歡迎訊息」設定畫面](#)
- Webhook ☒   
當用戶傳送訊息給本帳號或將本帳號加為好友時，可自動傳送訊息。  
[開啟Messaging API的設定畫面](#)
- 自動回應訊息 ☐   
可透過事先設定好的訊息內容進行自動回應。  
[開啟自動回應訊息設定畫面](#)

關閉自動回應

## 聊天的回應方式

啟用「聊天」功能後，

關閉回應時間

回應時間



可設定「回應時間」及「非回應時間」以

回應方式

☒ 手動聊天

- ☐ 手動聊天 + 自動回應訊息  
可使用自動回應訊息功能來自動回覆。
- ☐ 手動聊天 + AI自動回應訊息  
基本問題可交由AI自動回應訊息功能。
- ☐ 手動聊天 + 自動回應訊息 + AI自動回應訊息  
常見問題可透過已設定關鍵字的自動回應訊息。



# 加APP好友

Basic settings

Messaging API

1.點選Messaging API

## Bot information

Bot basic ID

@  

QR code



2.搜尋好友ID  
或掃描QR code

3.發送訊息  
給APP

# LINE 機器人 Messaging 小幫手


Created by 戴嘉熙

# FLEX MESSAGE SIMULATOR

<https://developers.line.biz/flex-simulator/>






## FLEX MESSAGE SIMULATOR

New Showcase Send... </> View as JSON



**Brown Cafe**  
★★★★☆ 4.0  
Place Miraina Tower, 4-1-6 Shinjuku, Tokyo  
Time 10:00 - 23:00  
CALL  
WEBSITE

+ ^ v



▼ bubble

▶ header

▶ hero

▶ body

▶ footer

### Bubble

type

bubble

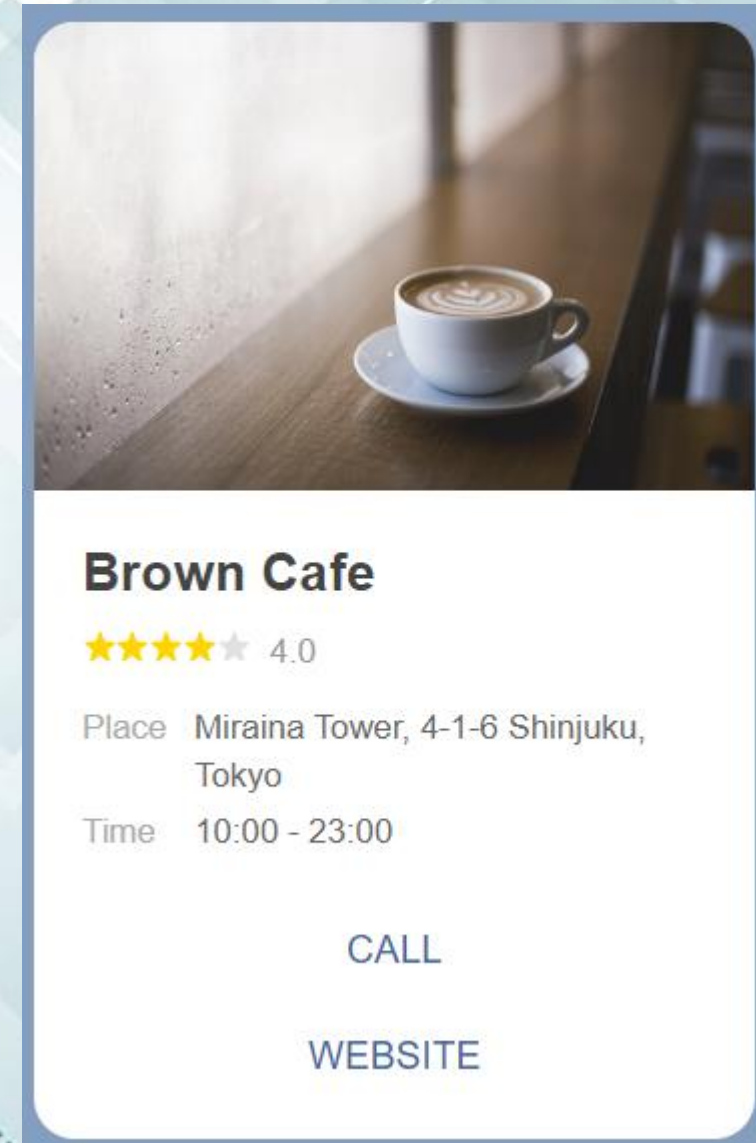
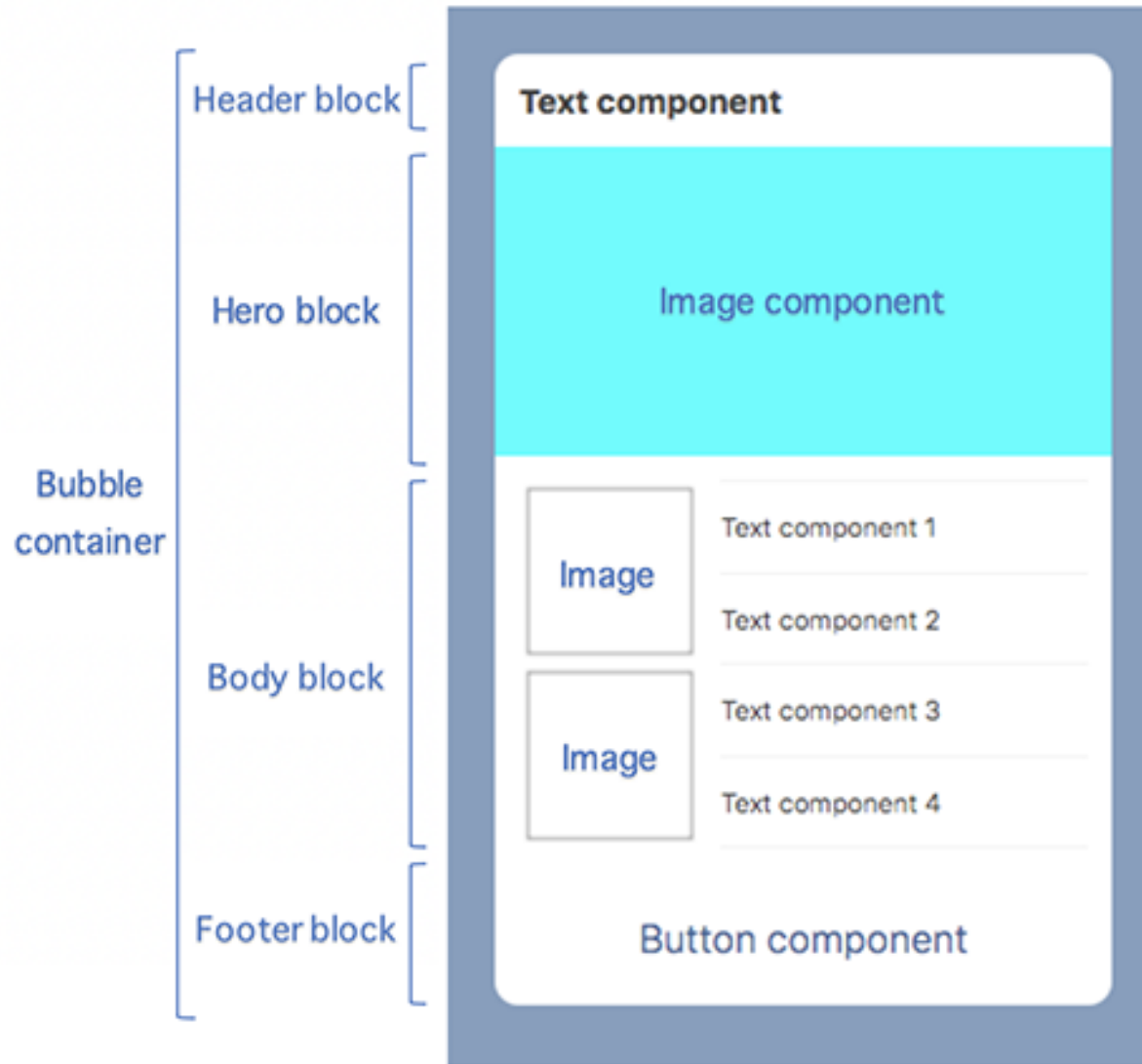
direction

size

### Action

type

# FLEX MESSAGE SIMULATOR 訊息架構





# LINE Bot Designer

<https://developers.line.biz/en/docs/messaging-api/download-bot-designer/>

## Bot Designer for macOS

Version number	Archive file	Release date
1.3.4	<a href="#">LINE Bot Designer-1.3.4.dmg</a>	2020-9-16

## Bot Designer for Windows

Version number	Archive file	Release date
1.3.4	<a href="#">LINE Bot Designer Setup 1.3.4.exe</a>	2020-9-16

# LINE Bot Designer

