

# IFTTT

鄒仔

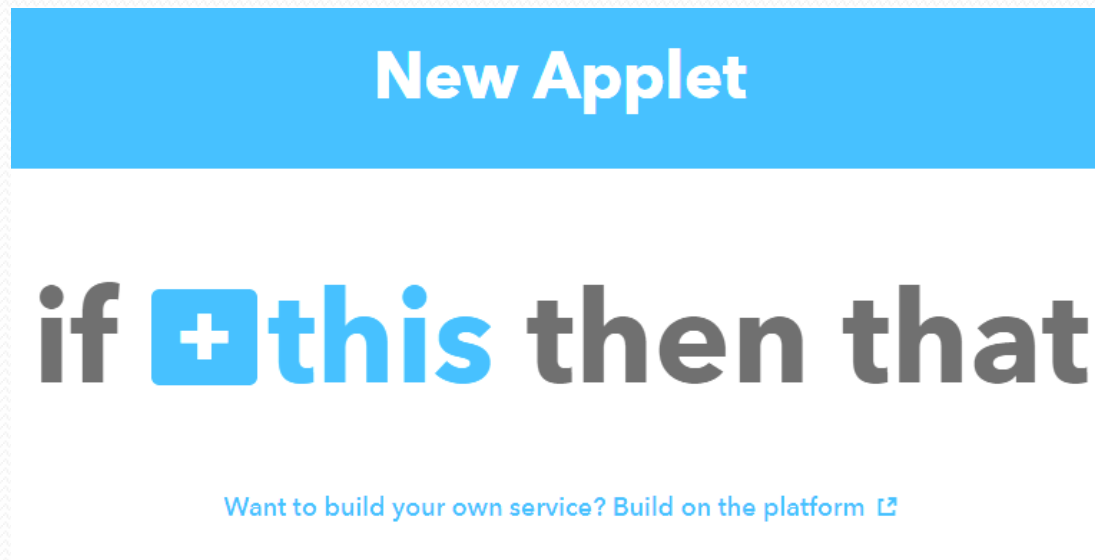
**New Applet**

**if**  **this** **then that**

Want to build your own service? Build on the platform [↗](#)

# IFTTT


- IFTTT，是一個新生的網絡服務平台，通過其他不同平台的條件來決定是否執行下一條命令。即對網絡服務通過其他網絡服務作出反應。IFTTT得名為其口號「if this then that」。




# Sign up 註冊

**IFTTT**Sign in

## Get started with IFTTT

 **Continue with Google**

 **Continue with Facebook**

Or use your password to [sign up](#) or [sign in](#)

# 綁定 Google

**IFTTT**

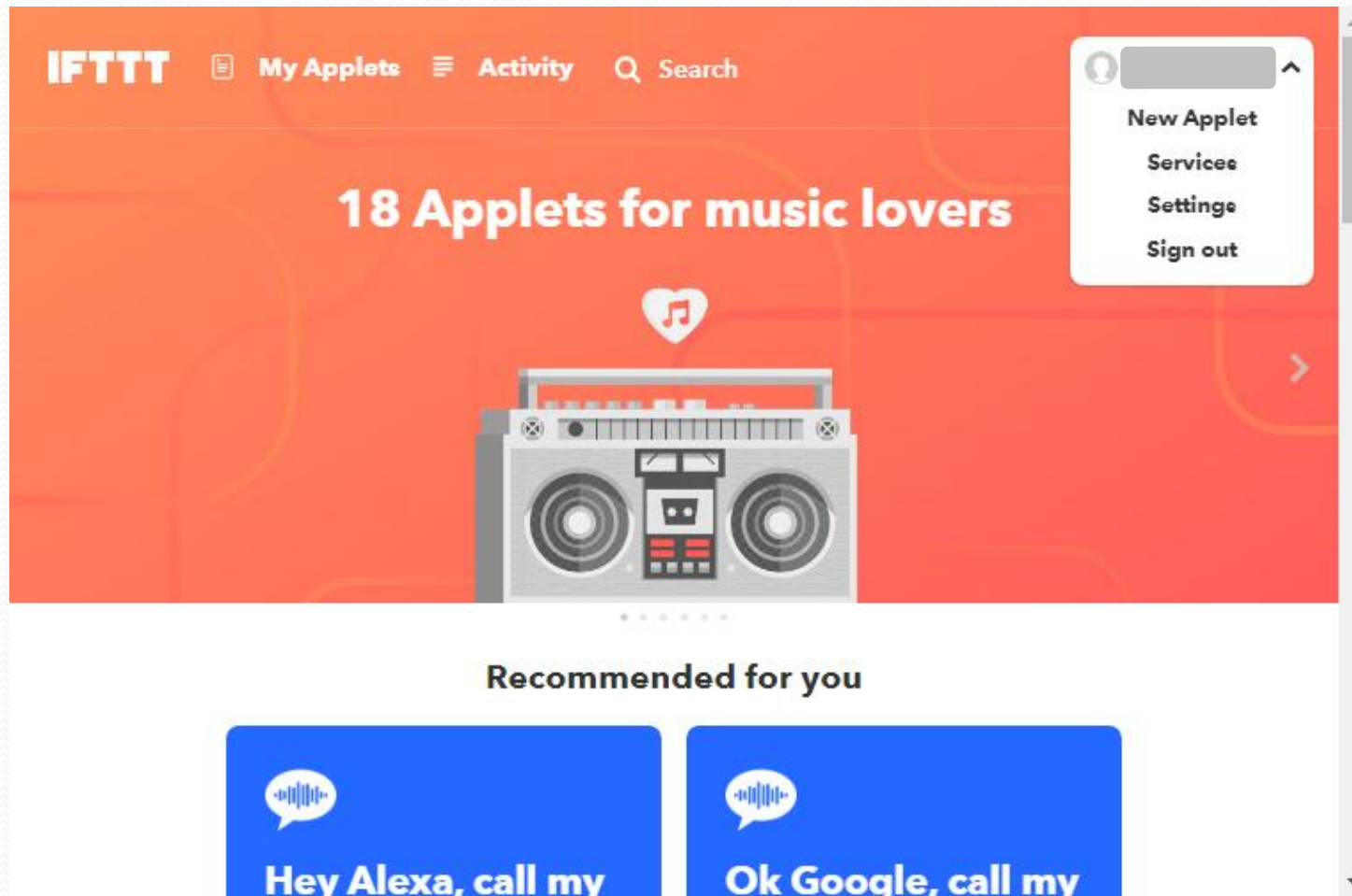
Sign in

## Action required

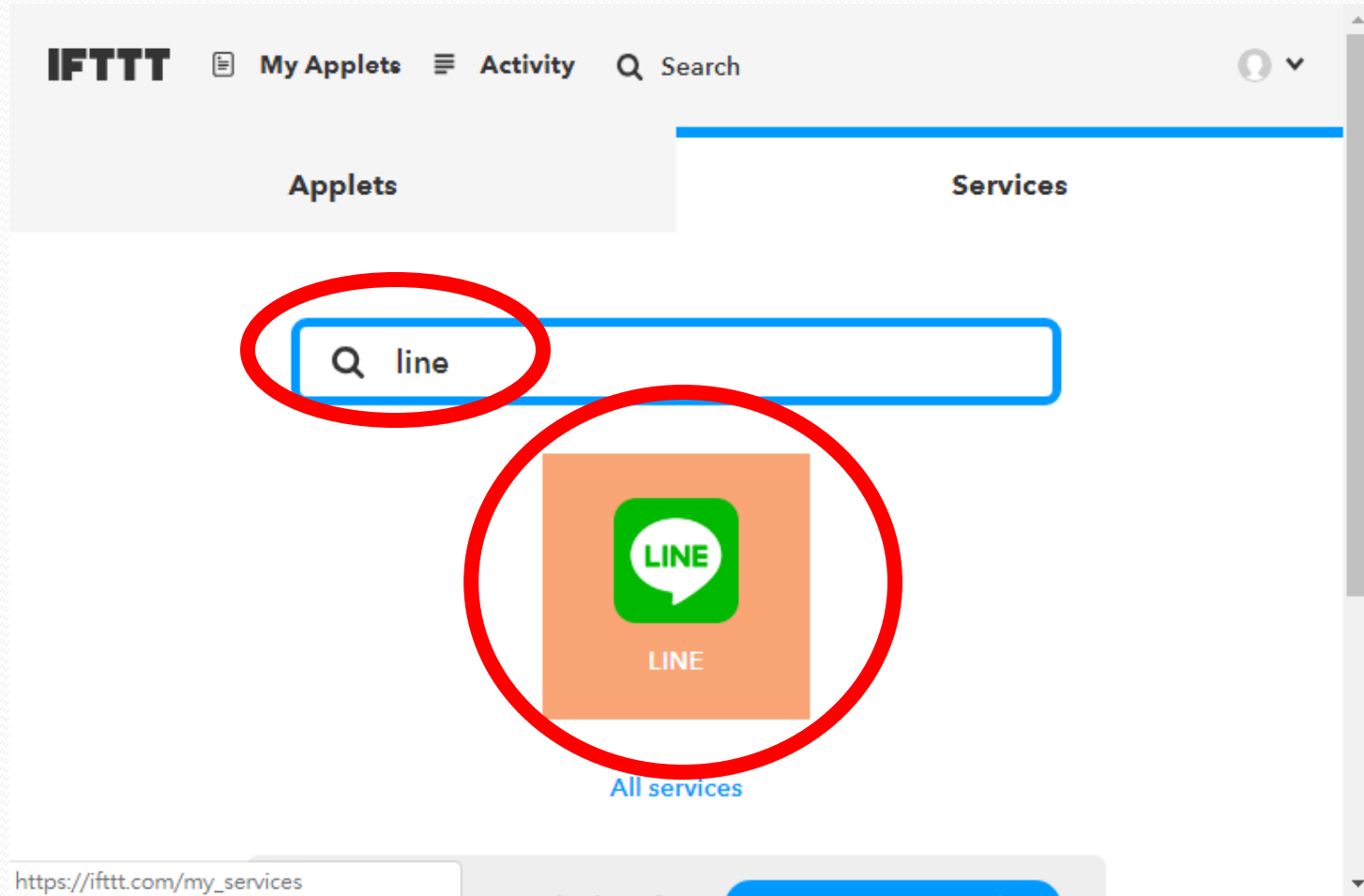
We found an account for [redacted]@gmail.com. Enter your IFTTT password to link your Google account.

[Forgot your password?](#)

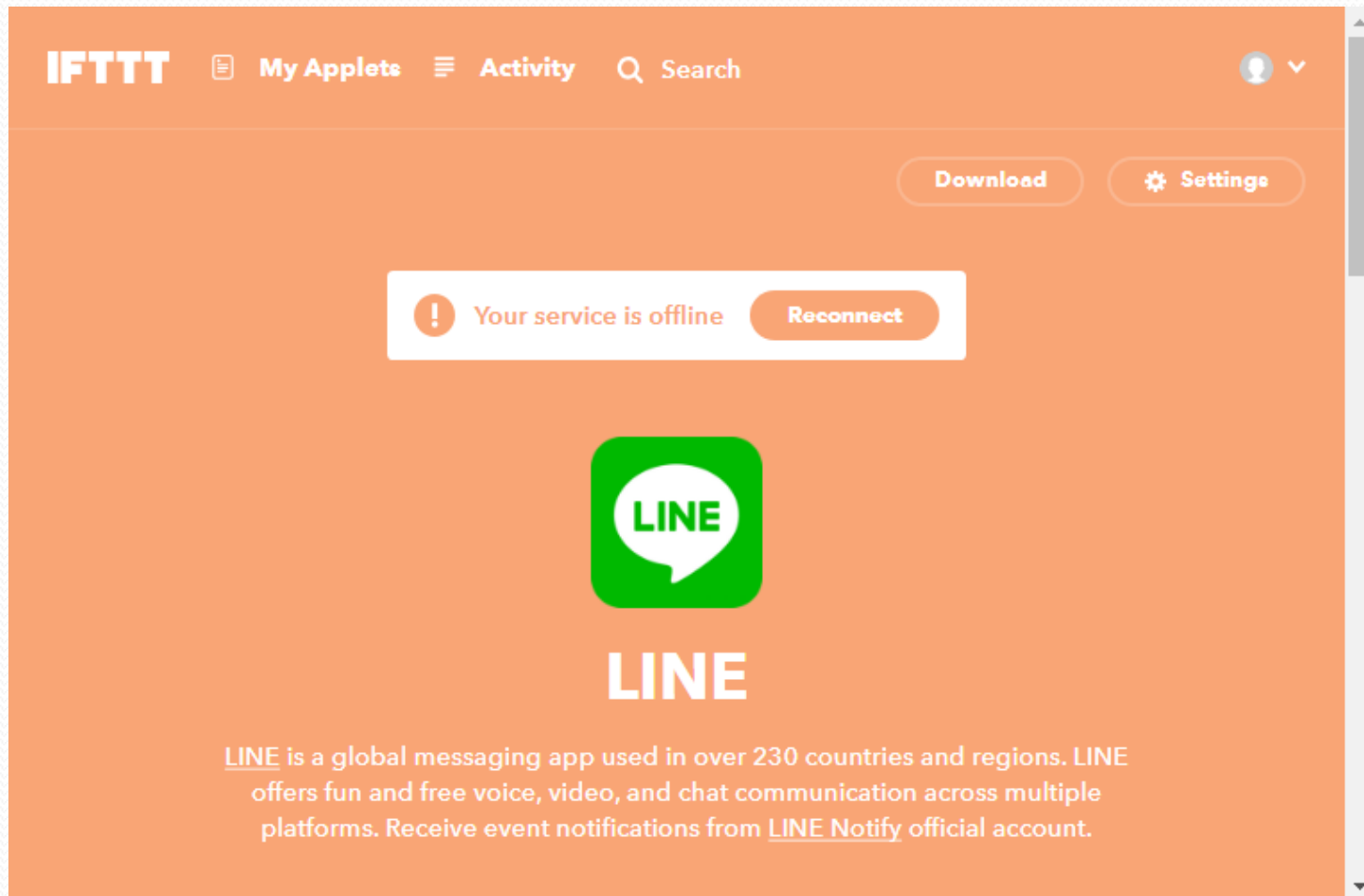
# Applet / Services



# 綁定 Line #1



# 綁定 Line #2








# 連動 Line notify






# 連動 Line notify

**IFTTT**  **My Applets**  **Activity**  Search  

**My Applets > LINE**



**LINE settings**  
[View activity log](#)

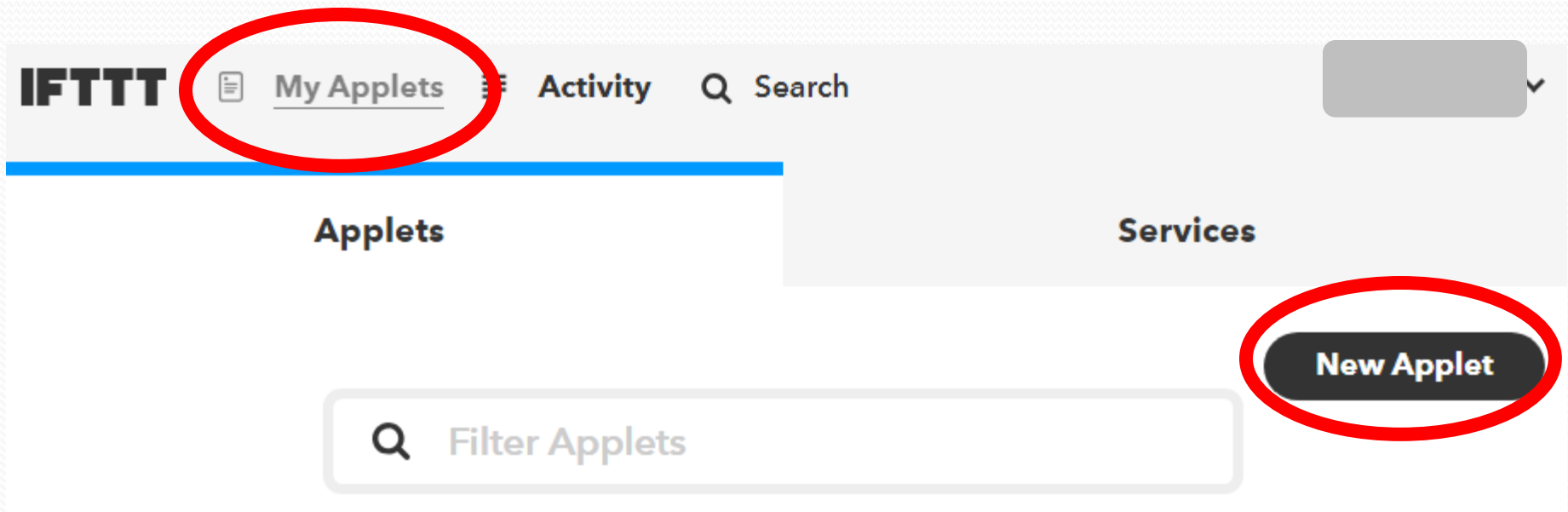
**Account Info**  
Connected as: Line.  
Status: **active**

**Edit connection**

# 連動 Line notify 成功



# 時間觸發 → LINE Notify #1



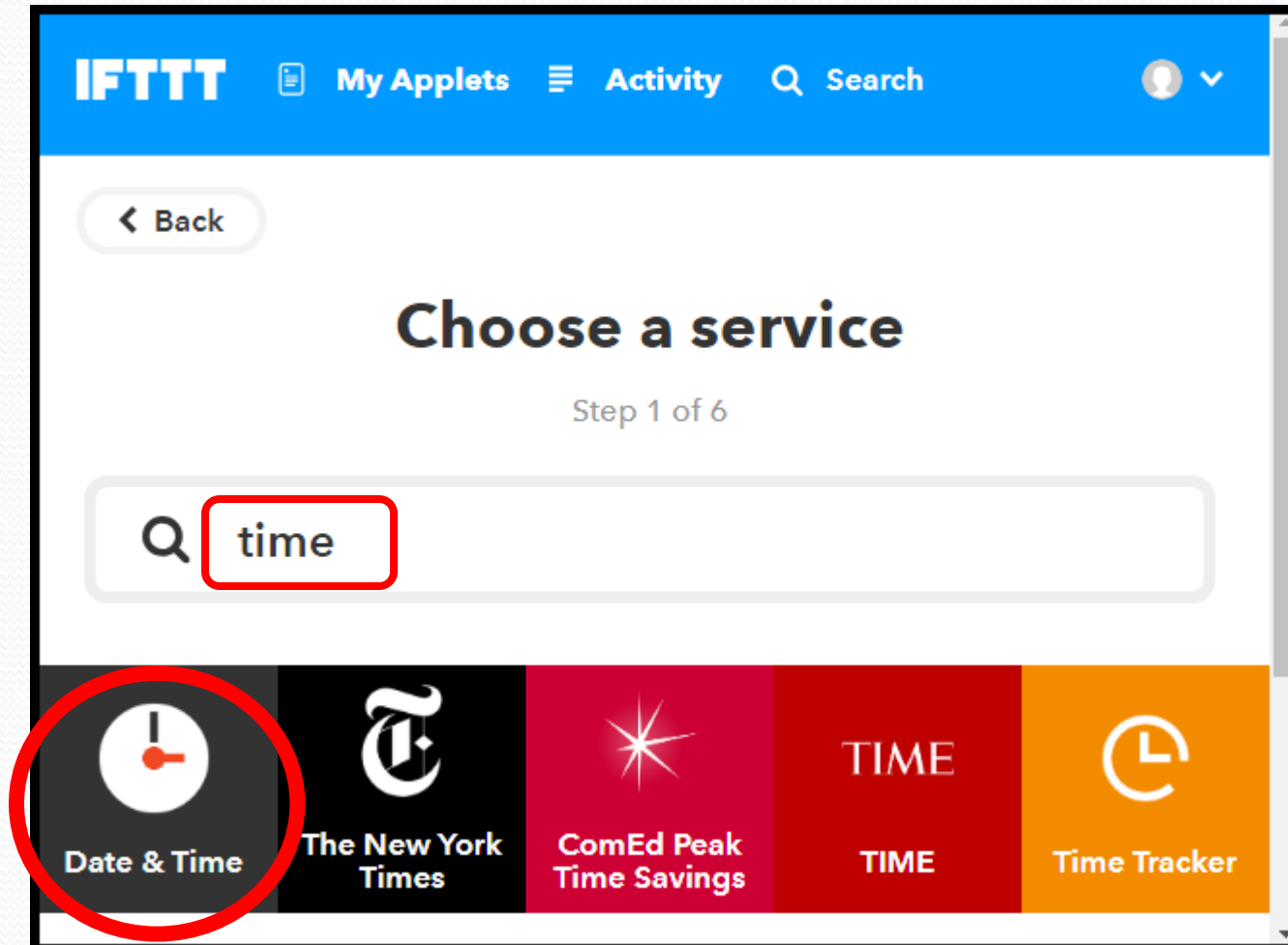
# 時間觸發 → LINE Notify #2

New Applet

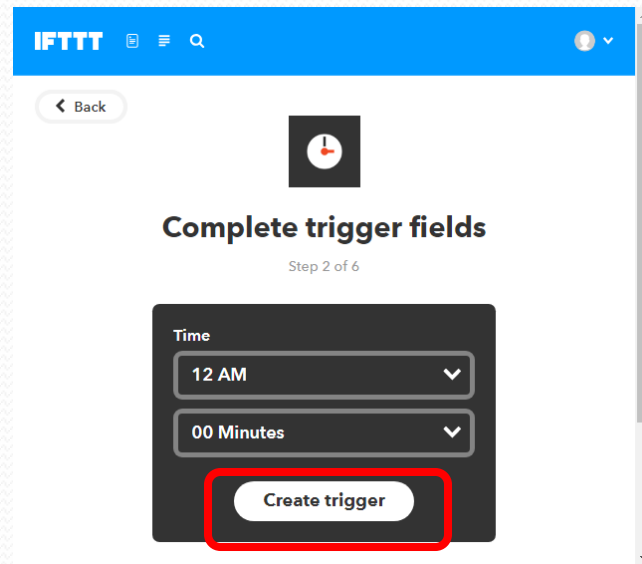
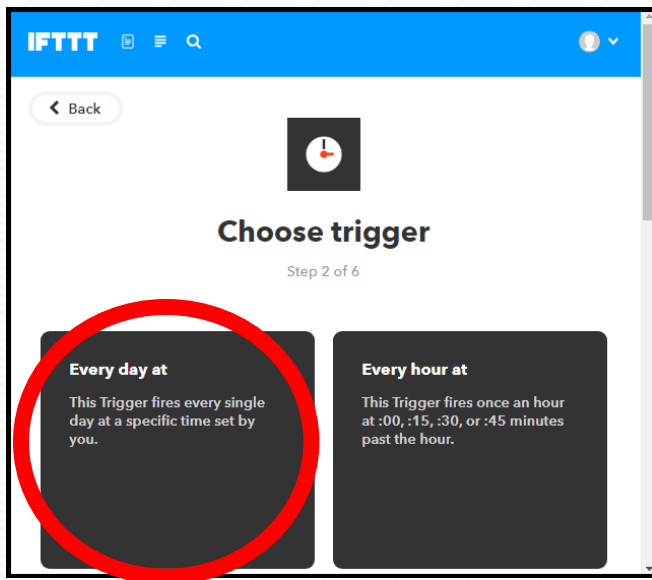
if  **this** then that

Want to build your own service? Build on the platform [🔗](#)

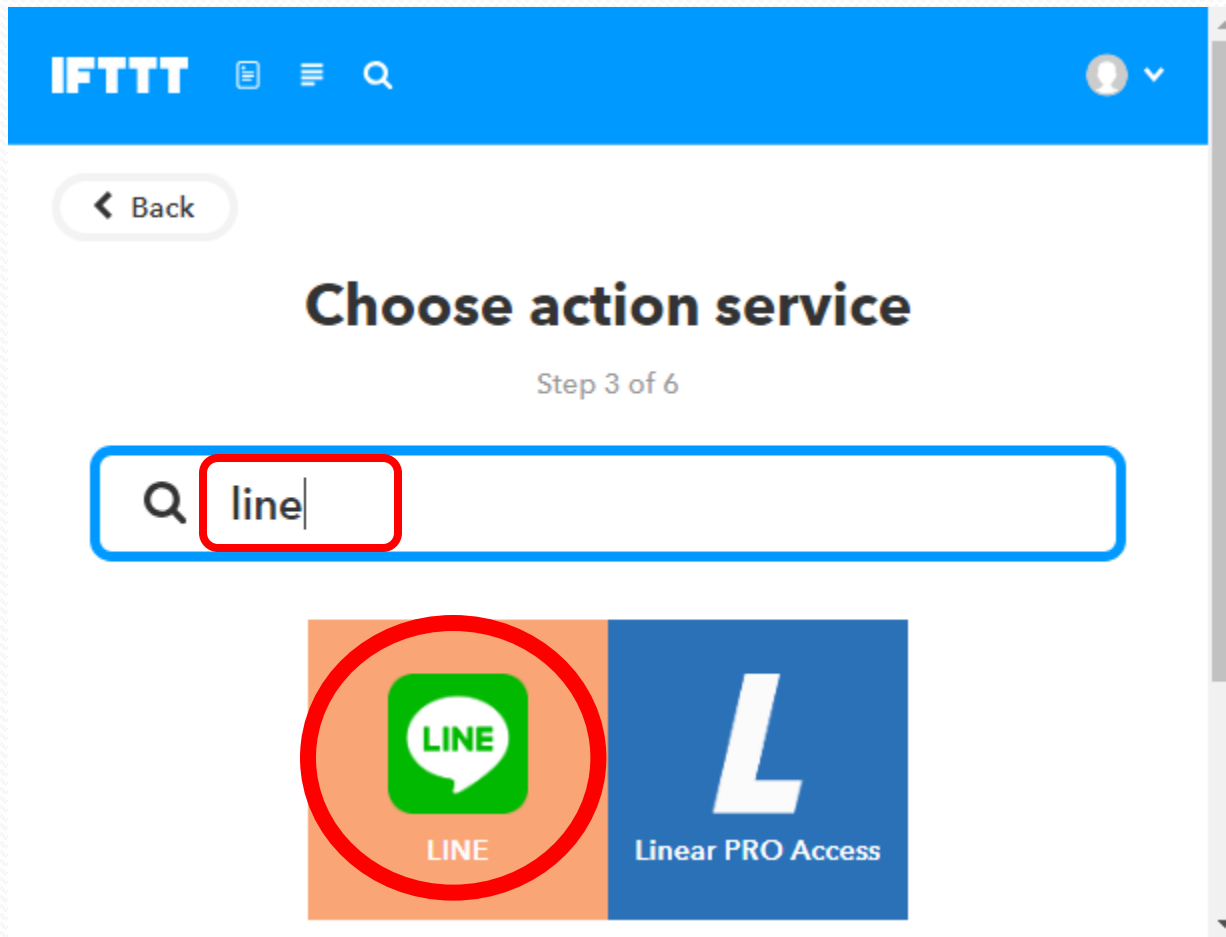
# 時間觸發 → LINE Notify #3



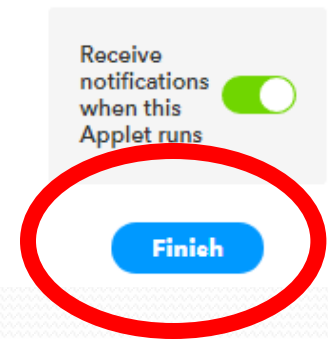
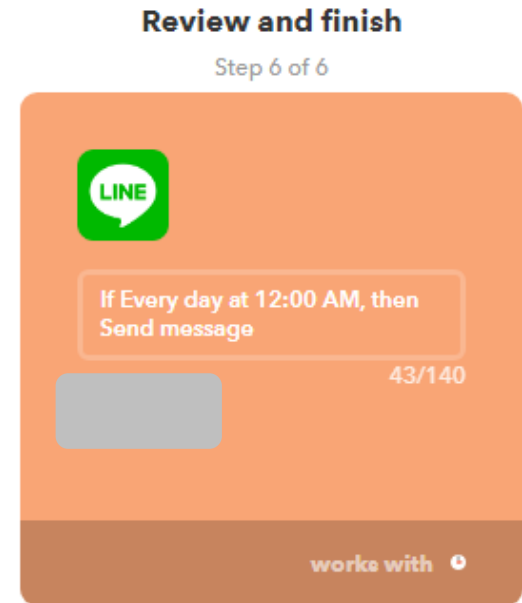
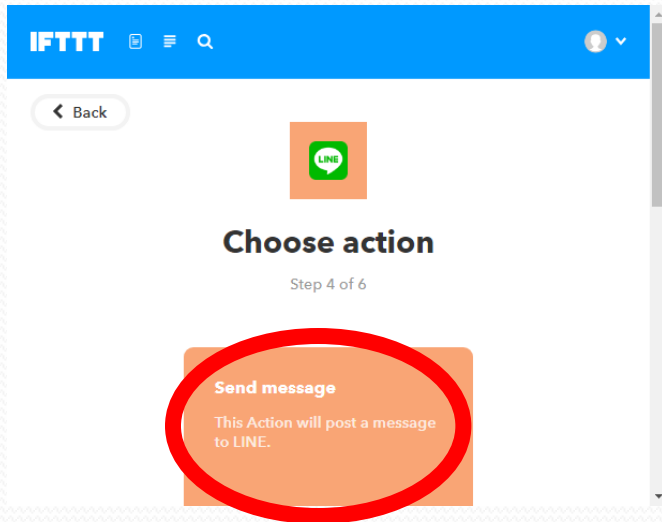
# 時間觸發 → LINE Notify #4



# 時間觸發 → LINE Notify #5

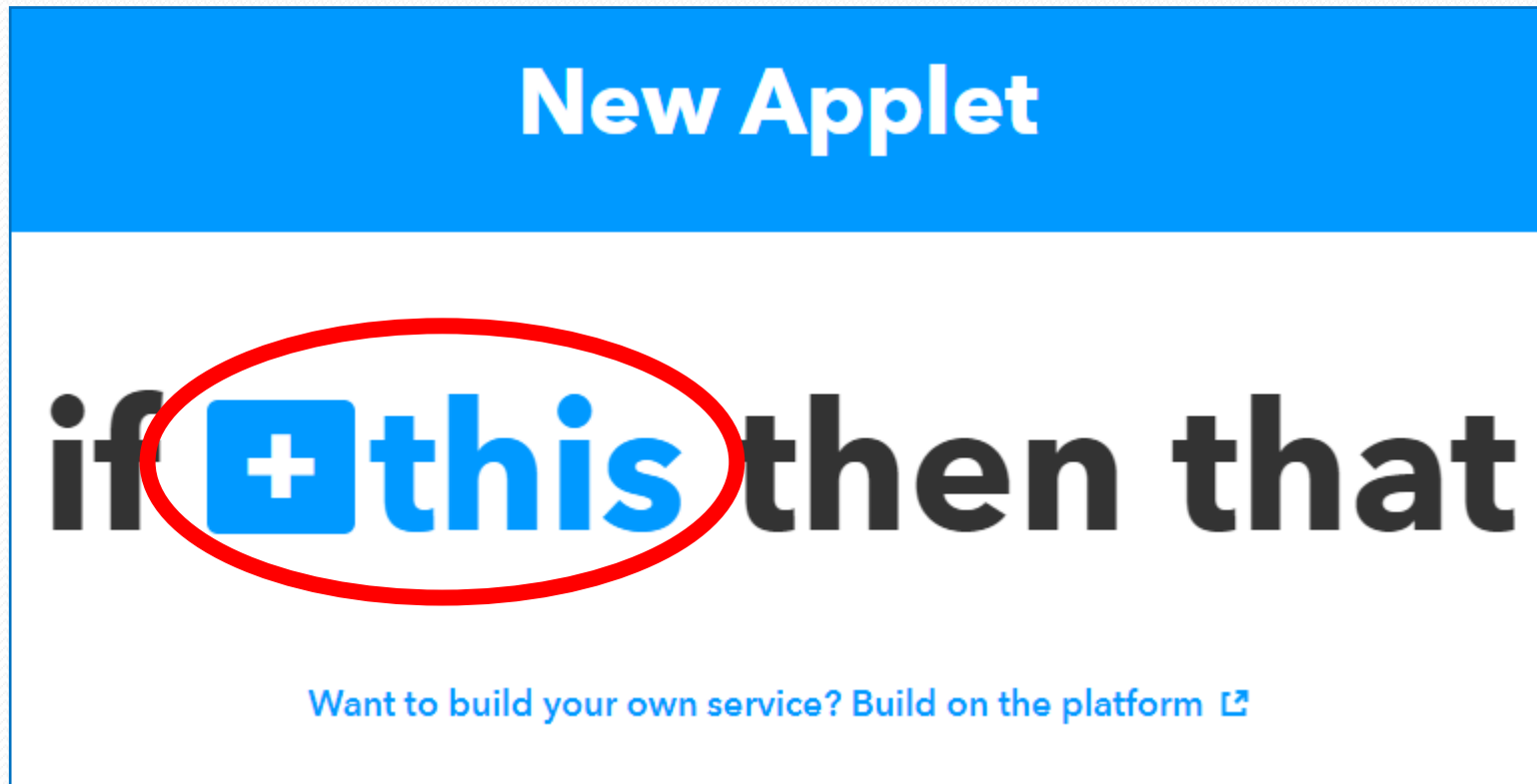


# 時間觸發 → LINE Notify #6

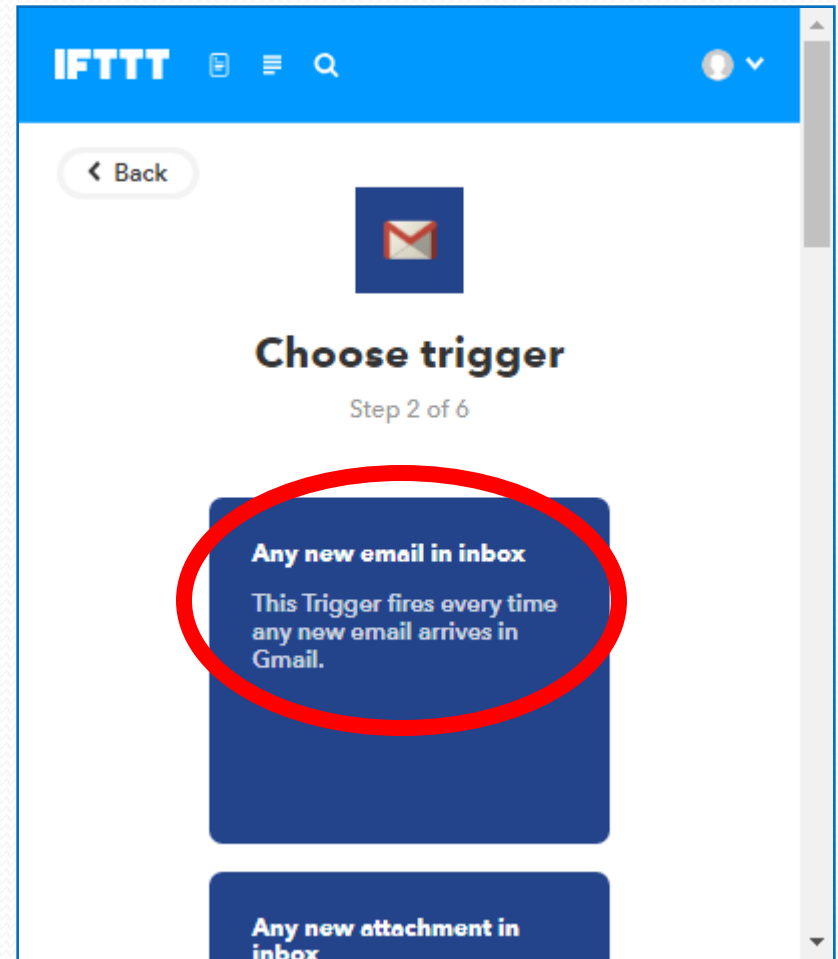
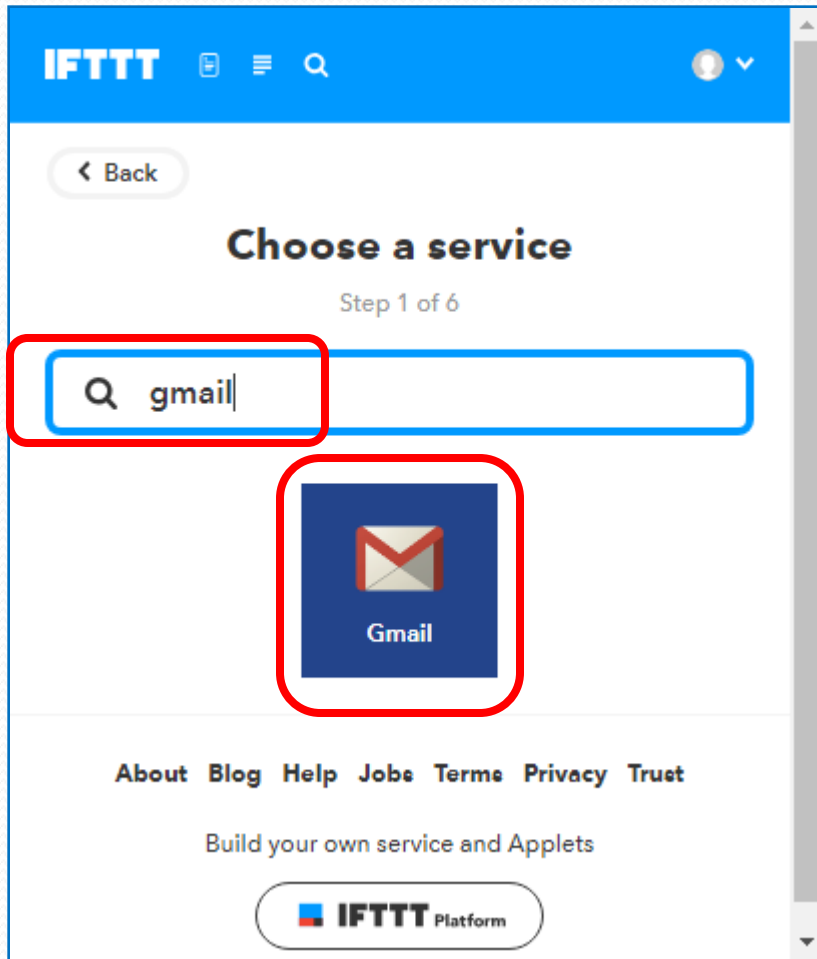




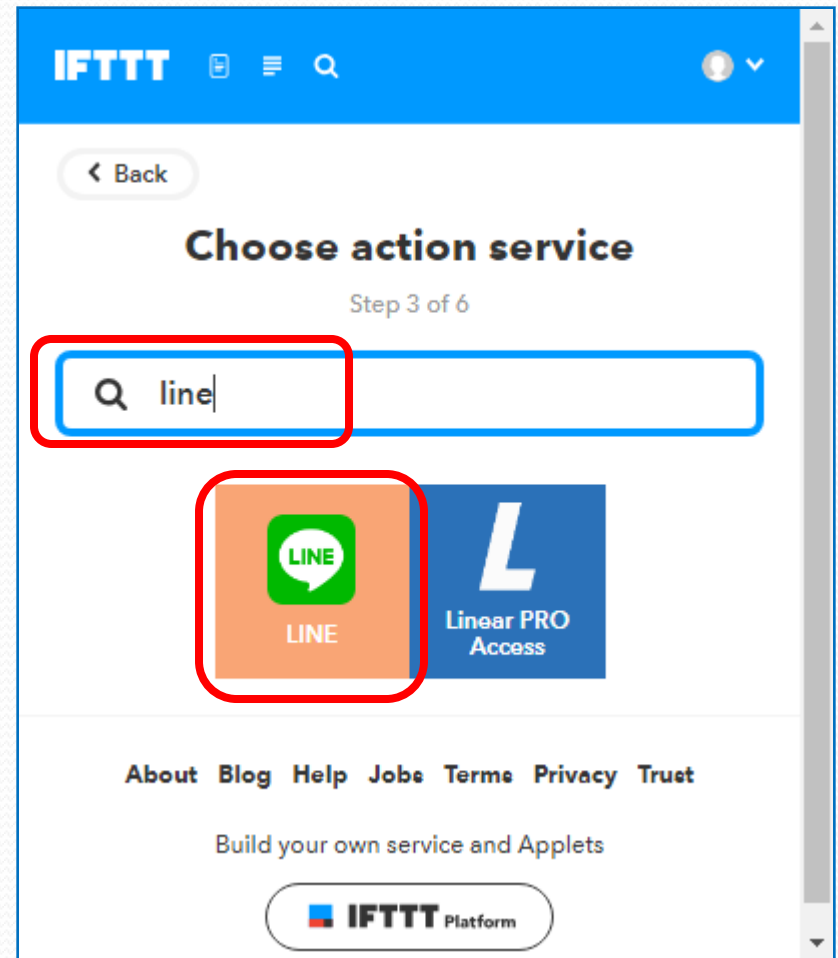
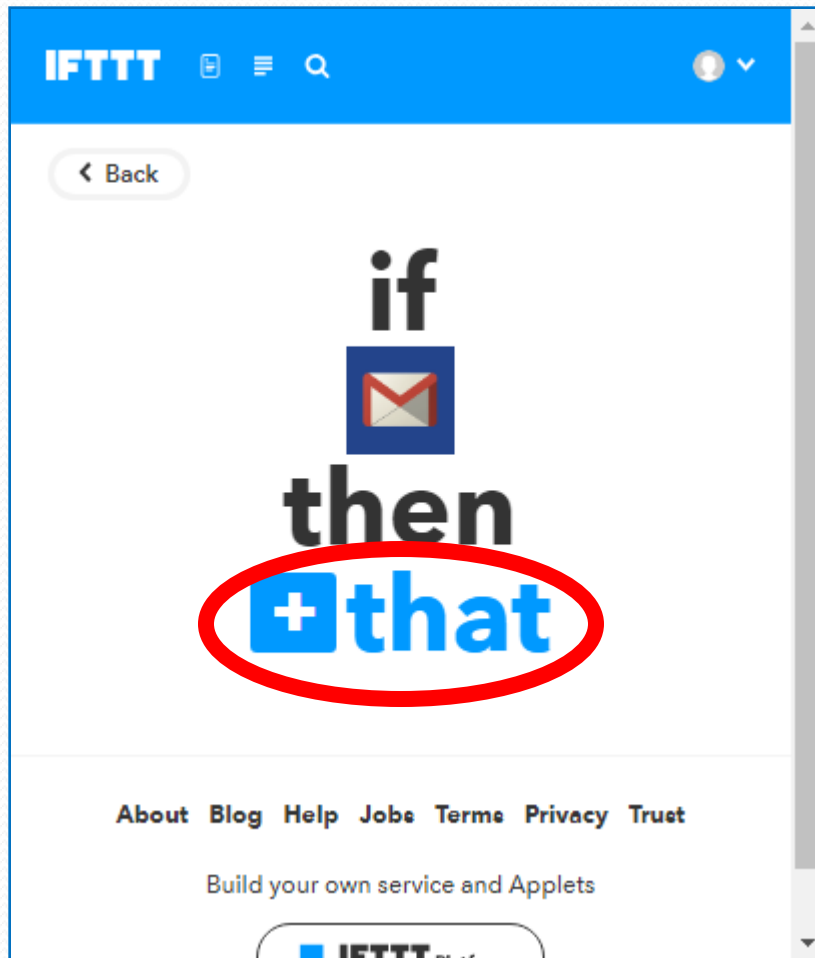
# Gmail to Line #1



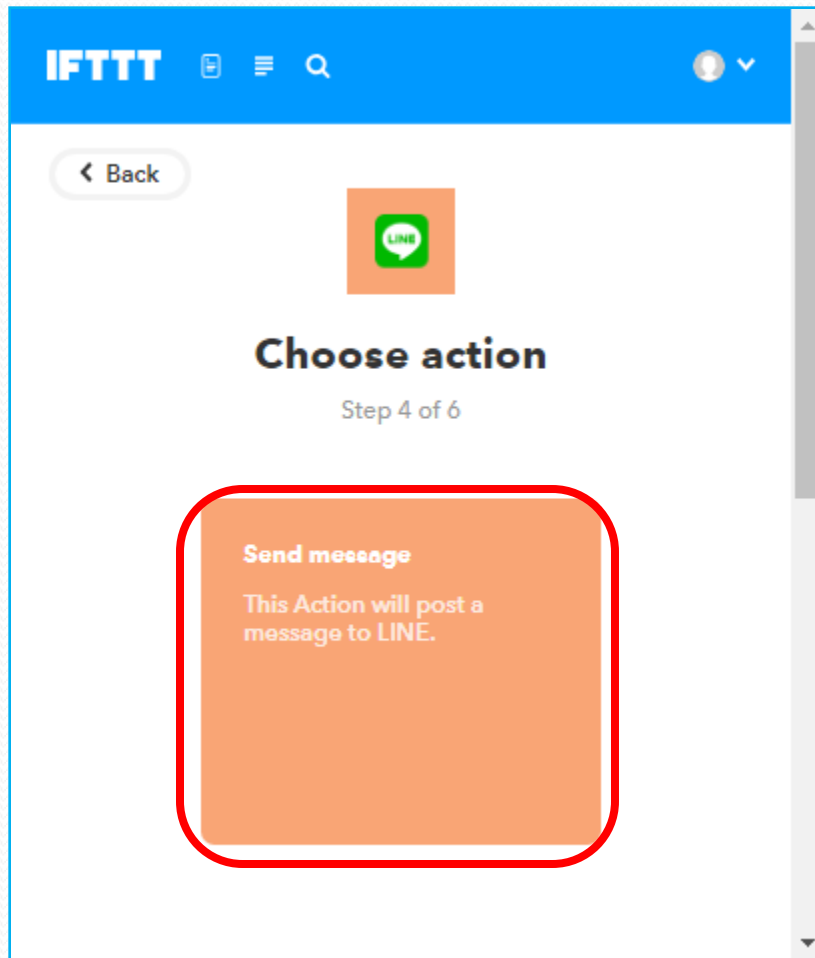
# Gmail to Line #2



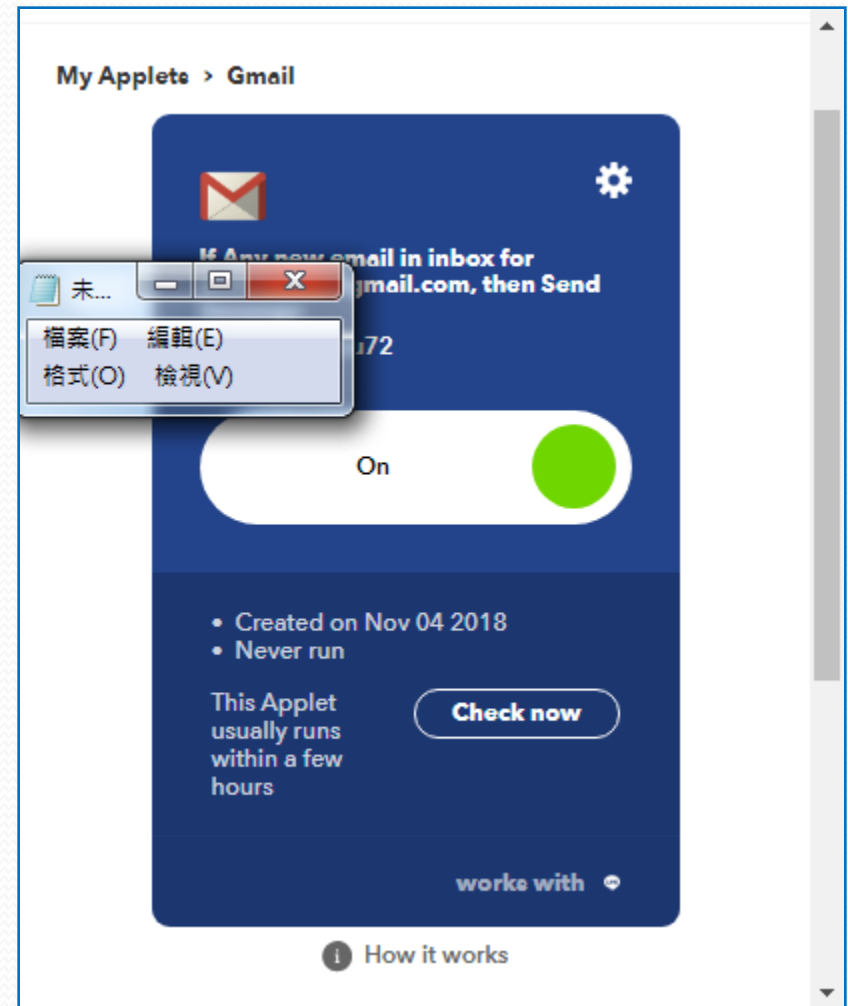
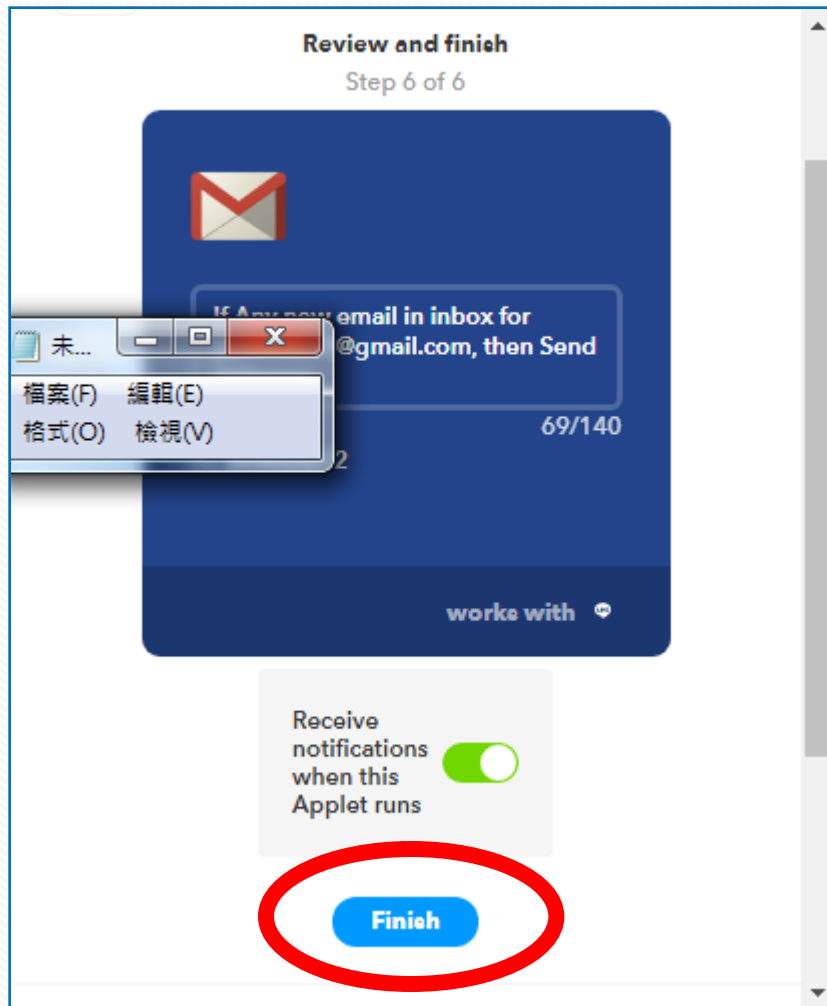
# Gmail to Line #3



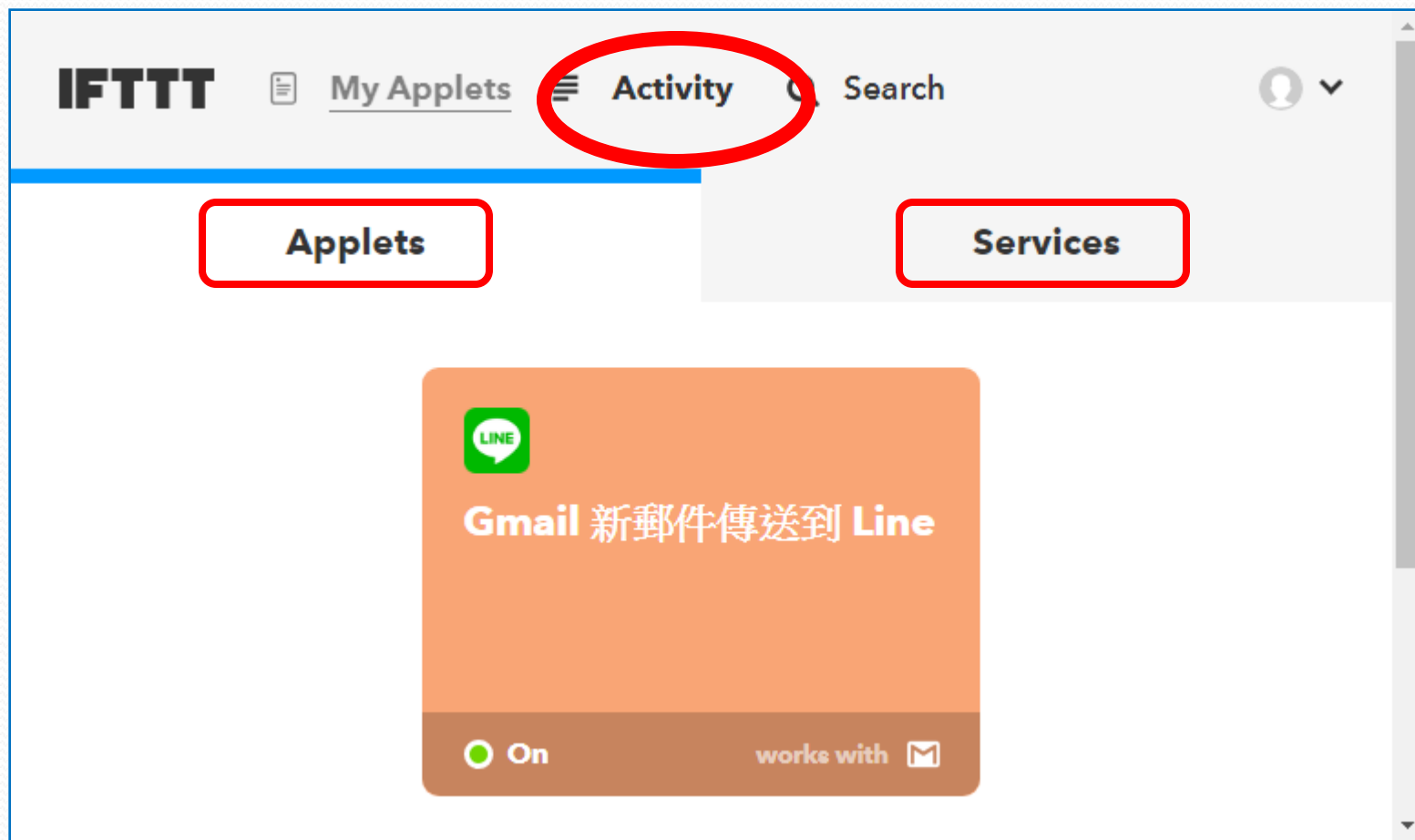
# Gmail to Line #4



# Gmail to Line #5

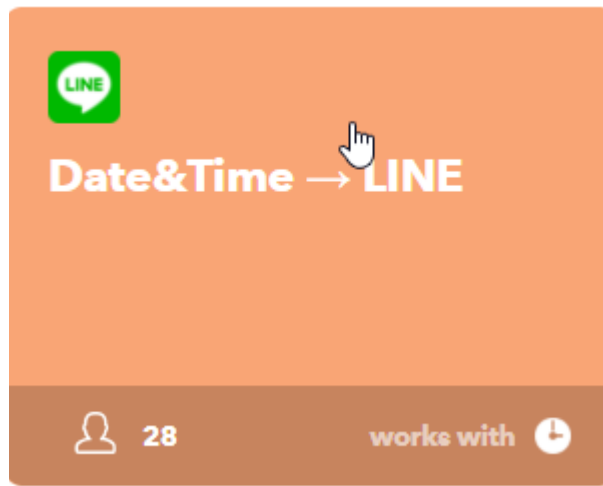
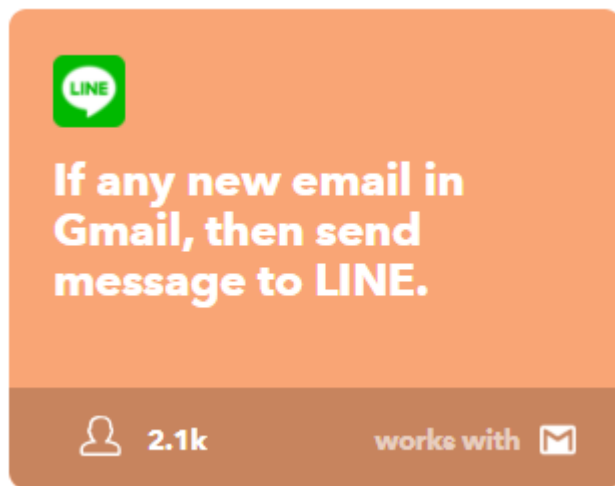


# 查看執行記錄檔

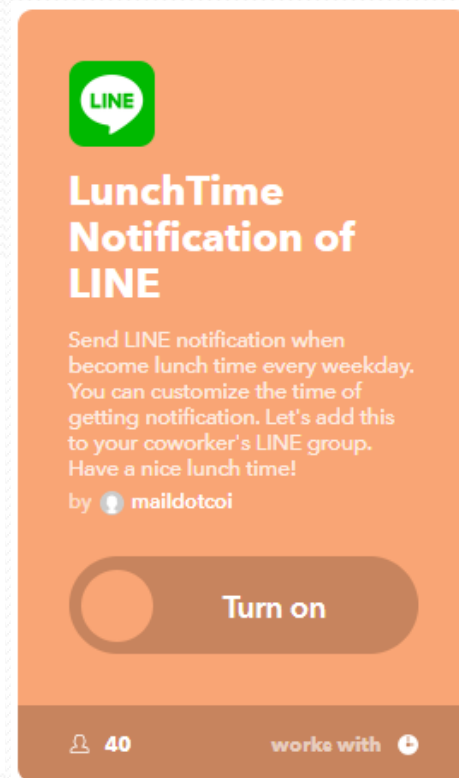


# 練習

- 基礎題
  - 時間觸發傳送訊息到 **Line** 的應用。  
(傳送到**Line**的資料，要出現時間變數和照片。)
  - ~~事件觸發Gmail即時收信→Line~~  
(要有圖片或照片，照片可固定在程序中。)
- 進階題
  - **Webhooks → LINE Notify**  
(類似創客，如家中的門窗被開啟或關閉的通知。)  
(其中一個參數傳照片網址並秀出。)



IFTTT



# Webhooks → LINE Notify

**New Applet**

**if  this then that**

Want to build your own service? Build on the platform [↗](#)



# Webhooks → LINE Notify



## Connect Webhooks

Step 1 of 6

Integrate other services on IFTTT with your DIY projects. You can create Applets that work with any device or app that can make or receive a web request. If you'd like to build your own service and Applets, check out the IFTTT platform.

**Connect**

# Webhooks → LINE Notify



## Choose trigger

Step 2 of 6

### Receive a web request

This trigger fires every time the Maker service receives a web request to notify it of an event. For information on triggering events, go to your Maker service settings and then the listed URL (web) or tap your username (mobile)

## Complete trigger fields

Step 2 of 6

### Event Name

WEB2LINE

The name of the event, like "button\_pressed" or "front\_door\_opened"

Create trigger

if



then



that

# Webhooks → LINE Notify

## Choose action service

Step 3 of 6

LINE



LINE



Linear PRO Access



## Choose action

Step 4 of 6

### Send message

This Action will post a message to LINE.

# Webhooks → LINE Notify

## Complete action fields

Step 5 of 6

Recipient

透過1對1聊天接收LINE Notify ▼

Message destination

Message

Value 1: Value1 <br>  
Value 2: Value2 <br>  
Value 3: Value3

Add ingredient

Photo URL

Add ingredient

Create action

## Review and finish

Step 6 of 6



If Maker Event "WEB2LINE",  
then Send message

by ilantzou

44/140

works with LINE

Receive notifications  
when this Applet runs



Finish

This Action will post a message to  
LINE.

Recipient

透過1對1聊天接收LINE Notify ▼

Message destination

修改

Message

<br>  
入侵偵測<br>  
位置: Value1 <br>  
狀況: Value2 <br>  
圖片網址: Value3

Add ingredient

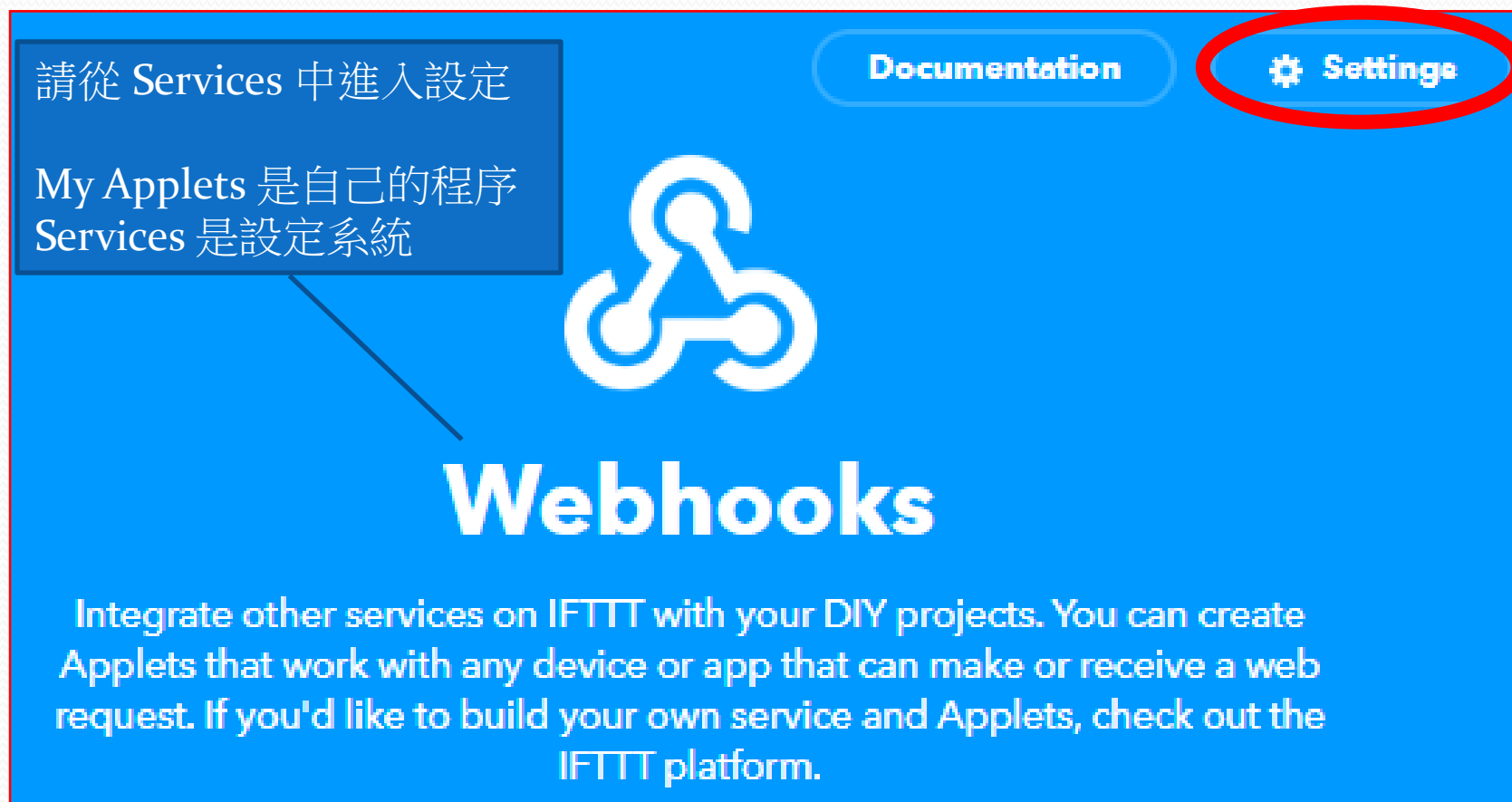
Photo URL (optional)

<http://img2.3png.com/bf765627615f4dc5073a64133f93ed28d243.png>

Add ingredient

Save

# Webhooks → LINE Notify



The screenshot shows the IFTTT Webhooks page. A blue box on the left contains the text: "請從 Services 中進入設定" (Please enter settings from Services) and "My Applets 是自己的程序" (My Applets are my own programs), with "Services 是設定系統" (Services are the settings system) below it. A line points from this box to the "Webhooks" title. In the top right, the "Settings" button (with a gear icon) is circled in red. The "Documentation" button is also visible. The main heading "Webhooks" is in large white text, followed by a paragraph explaining integration with IFTTT services and DIY projects.

請從 Services 中進入設定

My Applets 是自己的程序  
Services 是設定系統

Documentation Settings

## Webhooks

Integrate other services on IFTTT with your DIY projects. You can create Applets that work with any device or app that can make or receive a web request. If you'd like to build your own service and Applets, check out the IFTTT platform.

# 連結給予的參考網址

<https://maker.ifttt.com/use/bazJzbVqfIFeVuj3XTM3> (要用自己的)



## Webhooks settings

[View activity log](#)

### Account Info

Connected as:



URL: <https://maker.ifttt.com/use/bazJzbVqfIFeVuj3XTM3>

Status: **active**

**Edit connection**

會產生新的Key  
舊的Key仍有效  
但觸發會延遲

# 連結給予的參考網址

<https://maker.ifttt.com/use/bazJzbVqfIFeVuj3XTM3>(要用自己的)

1. 連線含權杖key的網址測試頁
2. 給定觸發的事件名稱
3. 輸入要傳送的資料
4. 會產生Curl範例 (JSON 資料格式)
5. 按 Test it 測試發送
6. 到 Line 檢查

Your key is: dGNKzr1\_pCDAqGye7Fyh40

◀ Back to service

## To trigger an Event

Make a POST or GET web request to:

[https://maker.ifttt.com/trigger/web2line/with/key/dGNKzr1\\_pCDAqGye7Fyh40](https://maker.ifttt.com/trigger/web2line/with/key/dGNKzr1_pCDAqGye7Fyh40)

With an optional JSON body of:

```
{ "value1" : "前門1", "value2" : "開啟", "value3" : "http://img2.3png.com/bf765627615f4dc5073a64133f93ed28d243.png" }
```

The data is completely optional, and you can also pass value1, value2, and value3 as query parameters or form variables. This content will be passed on to the Action in your Recipe.

You can also try it with curl from a command line.

```
curl -X POST -H "Content-Type: application/json" -d '{"value1": "前門1", "value2": "開啟", "value3": "http://img2.3png.com/bf765627615f4dc5073a64133f93ed28d243.png"}' https://maker.ifttt.com/trigger/web2line/with/key/dGNKzr1_pCDAqGye7Fyh40
```

Test it

```
curl -X POST -d "value1=Door&value2=Open&value3=Photo" https://maker.ifttt.com/trigger/web2line/with/key/dGNKzr1_pCDAqGye7Fyh40
```

# 傳送範例 Post

- Method: **POST**
- JSON : { "value1" : "Hello", "value2" : "Line", "value3" : "嘻嘻" }
- curl -X POST -H "Content-Type: application/json" -d '{"value1":"Hello","value2":"Line","value3":"嘻嘻"}' [觸發網址] (不一定支援)
- curl -X POST -d "value1=Hello&value2=Line&value3=嘻嘻" [觸發網址]



# 傳送範例 Get

- Method: **GET**
- URLcode : value1=**Door**&value2=**Open**&value3=Photo
- <https://maker.ifttt.com/trigger/事件名稱/with/key/Token?value1=Door&value2=Open&value3=Photo>
- curl -F "value1=Door" -F "value2=Open" -F "value3=Photo" "[觸發網址]"
- curl -d "value1=Door&value2=Open&value3=Photo" "[觸發網址]"

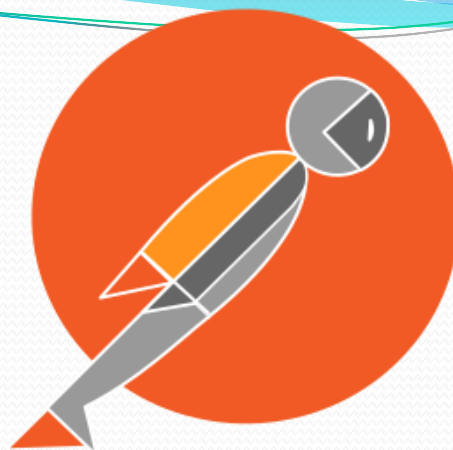
```
curl -X POST -d "value1=Door&value2=Open&value3=Photo" https://maker.ifttt.com/trigger/web2line/with/key/dGNKzr1_pCDAqGye7Fy自己的
```

# 傳送範例 表單Form Post 方式

```
<html><body><center>
<h2>請輸入資料傳送到 Line</h2>
<form action="https://maker.ifttt.com/trigger/事件名稱/with/key/Token"
method="POST">
Value1:<input type="text" name="value1"><br/>
Value2:<input type="text" name="value2"><br/>
Value3:<input type="text" name="value3"
value="https://www.cwb.gov.tw/V7/observe/satellite/Data/s3p/s3p.jpg"><br/>
<input type="submit" value="按一下傳送到 Line">
</form>
</center></body></html>
```

- 複製上方內容貼到記事本並存成 ifttt.htm
- 用瀏覽器開啟 ifttt.htm 後輸入資料送出

# 測試偵錯軟體



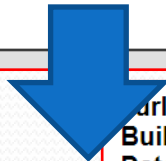
- Postman (Chrome)
- Curl
- iCurl (iPhone)
- iCurlHTTP (iPhone)
- Termux (Android) :  
pkg install curl 安裝。
- Terminal Emulator for Android :  
連線 SSL 請多加 -k 參數。



```
Window 1
--retry <num> Retry request <num> times if transient problems occur
--retry-delay <seconds> When retrying, wait this many seconds between each
--retry-max-time <seconds> Retry only within this period
-S/--show-error Show error. With -s, make curl show errors when they occur
-s/--silent Silent mode. Don't output anything
--socks4 <host[:port]> SOCKS4 proxy on given host + port
--socks4a <host[:port]> SOCKS4a proxy on given host + port
--socks5 <host[:port]> SOCKS5 proxy on given host + port
--socks5-hostname <host[:port]> SOCKS5 proxy, pass host name to proxy
-V/--speed-limit Stop transfer if below speed-limit for 'speed-time' secs
-y/--speed-time Time needed to trig speed-limit abort. Defaults to 30
--ssl Try SSL/TLS (FTP, IMAP, POP3, SMTP)
--ssl-reqd Require SSL/TLS (FTP, IMAP, POP3, SMTP)
-2/--sslv2 Use SSLv2 (SSL)
-3/--sslv3 Use SSLv3 (SSL)
--stderr <file> Where to redirect stderr, - means stdout
-t/--tcp-nodelay Use the TCP_NODELAY option
-t/--telnet-option <OPT=val> Set telnet option
--tftp-blksize <value> Set TFTP BLKSIZE option (must be >512)
-z/--time-cond <time> Transfer based on a time condition
-T/--tlsv1 Use TLSv1 (SSL)
--trace <file> Write a debug trace to the given file
--trace-ascii <file> Like --trace but without the hex output
--trace-time Add time stamps to trace/verbose output
-T/--upload-file <file> Transfer <file> to remote site
--url <URL> Set URL to work with
-B/--use-ascii Use ASCII/text transfer
-U/--user <user[:password]> Set server user and password
-A/--user-agent <string> User-Agent to send to server (H)
-v/--verbose Make the operation more talkative
-V/--version Show version number and quit
-w/--write-out <format> What to output after completion
--xattr Store metadata in extended file attributes
-q If used as the first parameter disables curl
s://notify-api.line.me/api/notify
```

# Curl 下載

Windows 32 bit			
Windows 32 bit	7.61.1	binary	the curl project
Windows 32 bit	7.61.1	binary	Chocolatey
Windows 32 bit	7.61.1	binary	Stefan Kanthak
Windows 32 bit	7.61.1	binary	Viktor Szakats
Windows 32 bit	7.61.0	binary	Dirk Paehl
Windows 32 bit	7.61.0	binary	Marc Hörsken



curl version: 7.61.1  
Build: 7.61.1\_5  
Date: 2018-09-05  
Changes: [7.61.1 changelog](#)

[Changelog](#)  
[Downloads](#)  
[FAQ](#)  
[License](#)  
[Manual](#)

## Packages



[curl for 64 bit](#)

Size: 3.1 MB  
sha256: d584f0e3a7fa7234d14fb85cfb57841ac123f77a256dc643a2493af9d6e4e7e8



[curl for 32 bit](#)

Size: 2.9 MB  
sha256: d80a7057c1a5f2794b29bd157e87a601d9c0c38b525467cd0b540ec76651c12

# Curl 測試與接收

**Recipient**

透過1對1聊天接收LINE ▼

Message destination

**Message**

<br>  
Value 1: Value1 <br>  
Value 2: Value2 <br>  
Value 3: Value3 <br>  
EventName

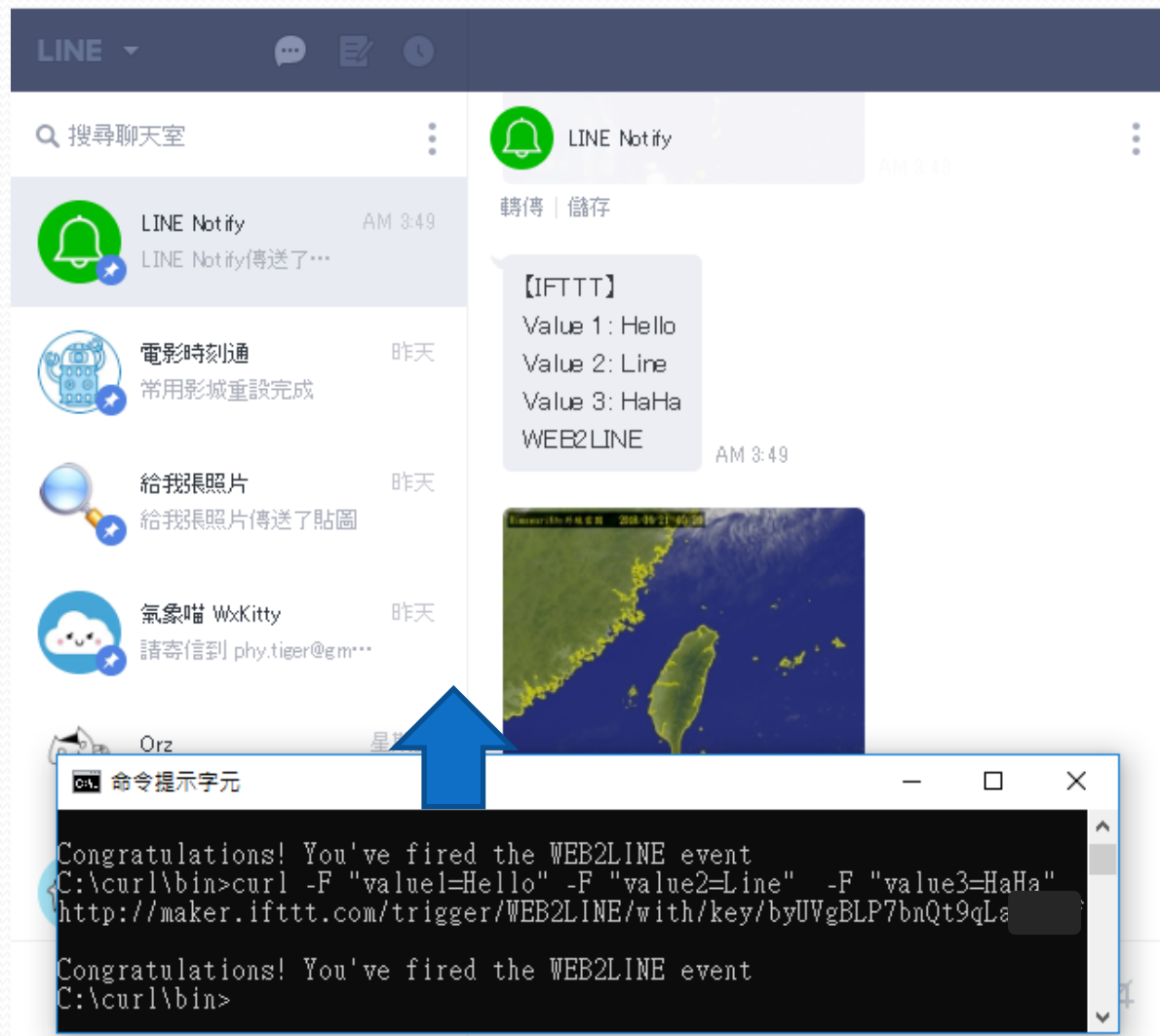
Add ingredient

**Photo URL (optional)**

<https://www.cwb.gov.tw/V7/observe/satellite/Data/s3p/...pg>

Add ingredient

Save



# 參考資料

- [https://notify-bot.line.me/zh\\_TW/](https://notify-bot.line.me/zh_TW/)
- <https://notify-bot.line.me/doc/en/>
- <https://engineering.linecorp.com/tw/blog/detail/88>
- <https://engineering.linecorp.com/tw/blog/detail/94>
- <https://engineering.linecorp.com/tw>
- <https://www.oxxostudio.tw/articles/201803/ifttt-line.html>
- <https://www.oxxostudio.tw/articles/201806/line-notify.html>

預告：

Google App Script => Line notify ◦

