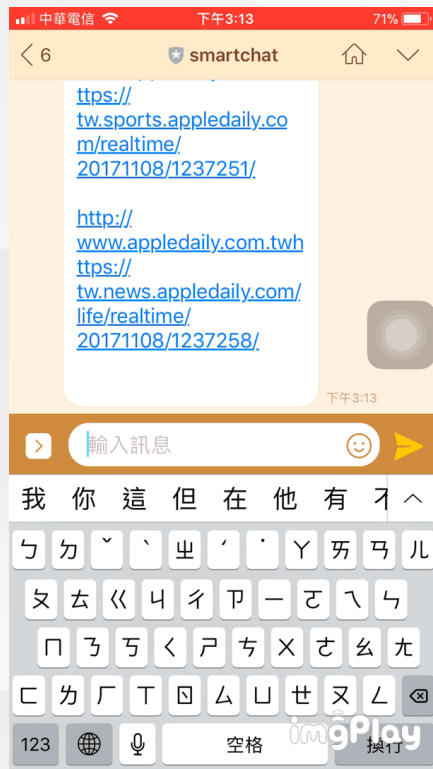


# What is a Chatbot?

# Chatbot

Chatbot 是一種服務，其透過 AI ( 人工智慧 ) 或自行定義的規則，讓使用者可以透過聊天的方式，與其進行互動。可以是各種類型的服務，例如 :CNN 用以提供新聞資訊、台灣房屋地產機器人、聯合報機器人、TaxiGO



# 台灣房屋地產機器人



# 聯合報機器人



# 醫療聊天機器人



# ChatBot 種類

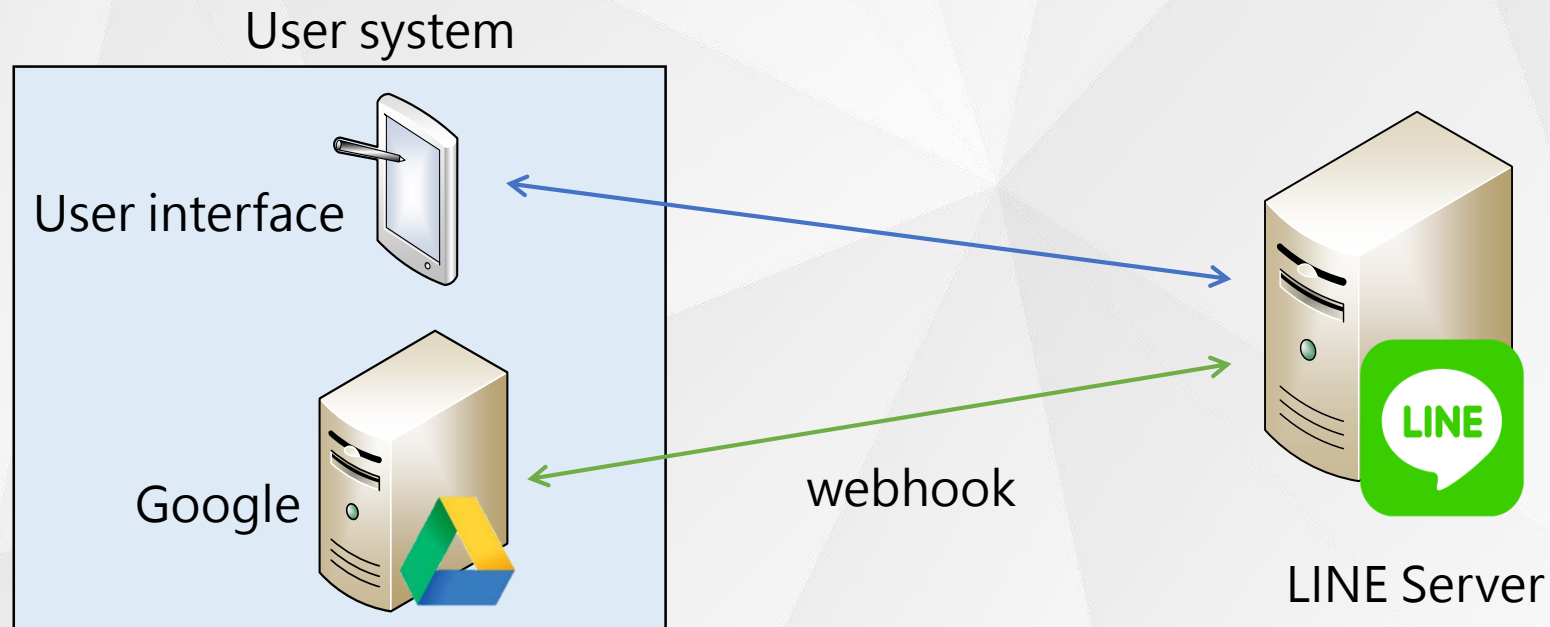


WeChat



slack

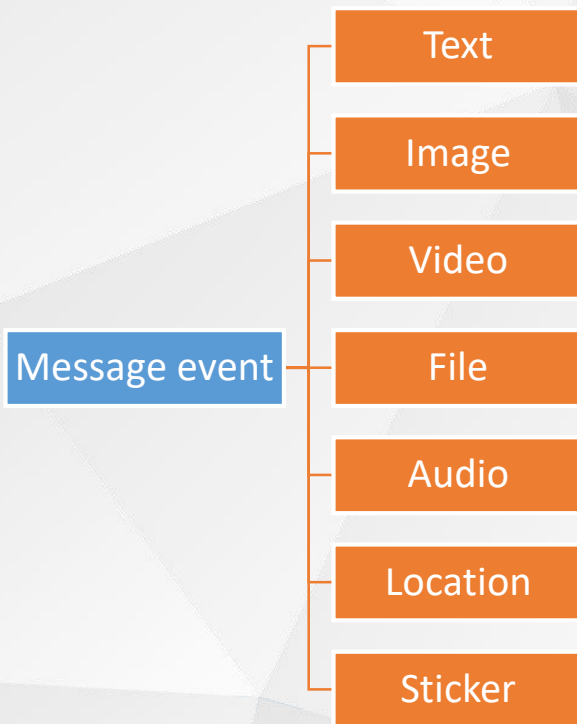
# ChatBot 簡單架構





# Webhook

LINE發送訊息給BOT的一些事件 例如:訊息、好友邀請、群組邀請  
以下舉例訊息事件

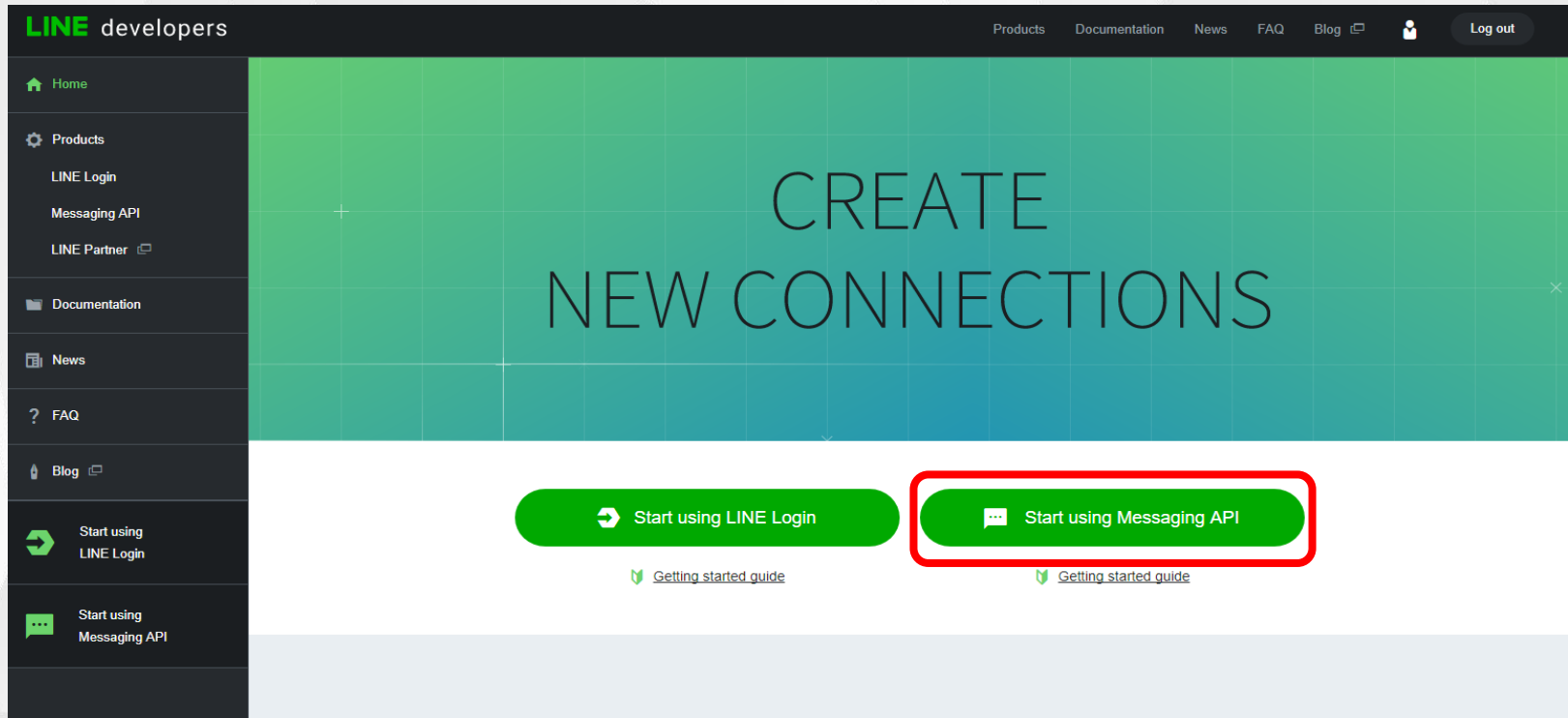




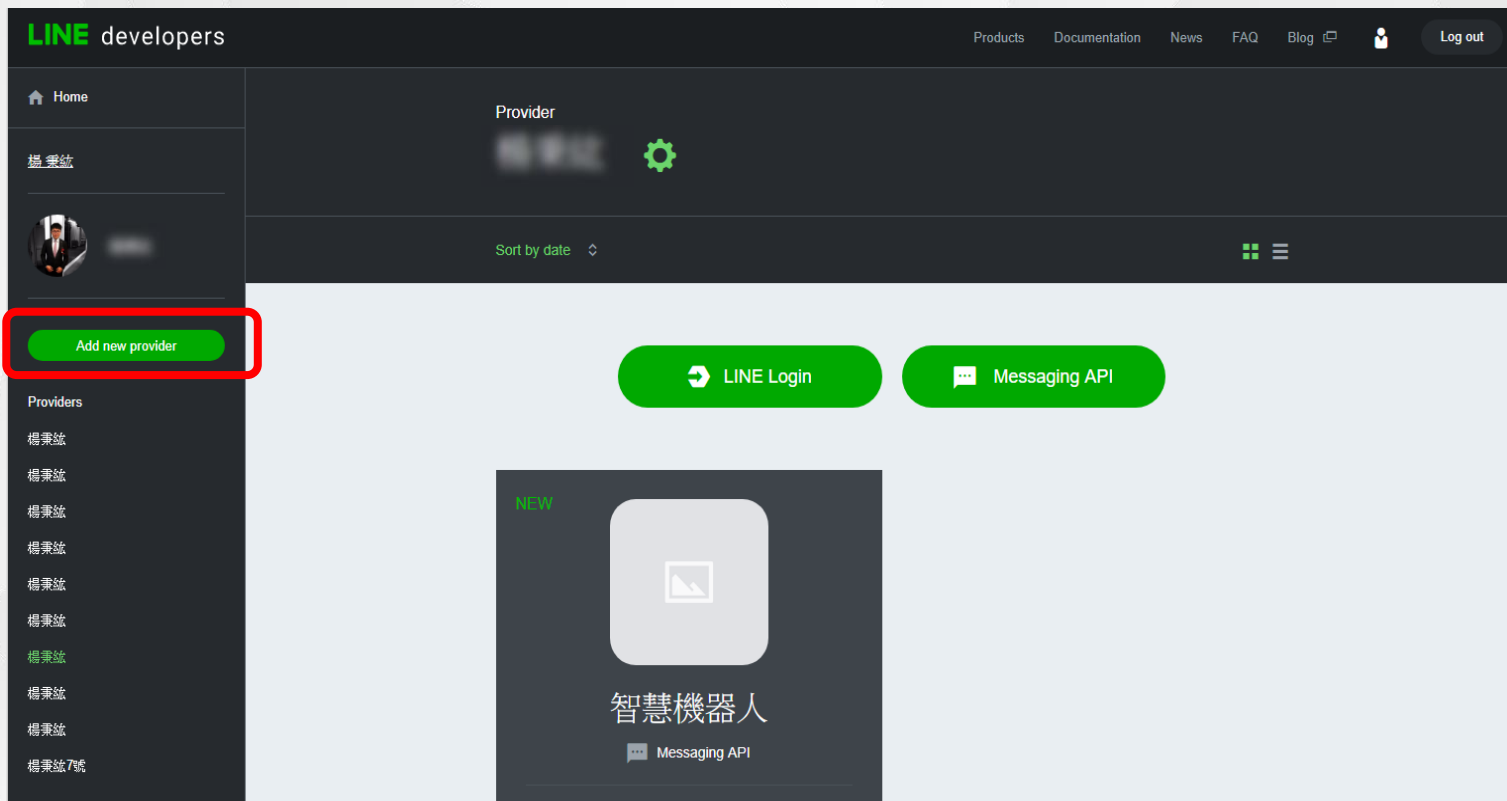
# 註冊流程

# 註冊流程

<https://developers.line.me/en/>



# 註冊流程



# 註冊流程

## Enter name of provider

The provider is the entity (individual or company) that offers the app.

**Provider name**

Max: 100 characters

Confirm

# 註冊流程



## STEP 1 Select a provider

The provider is the entity (individual or company) that offers the app.

### Select provider

Select a provider

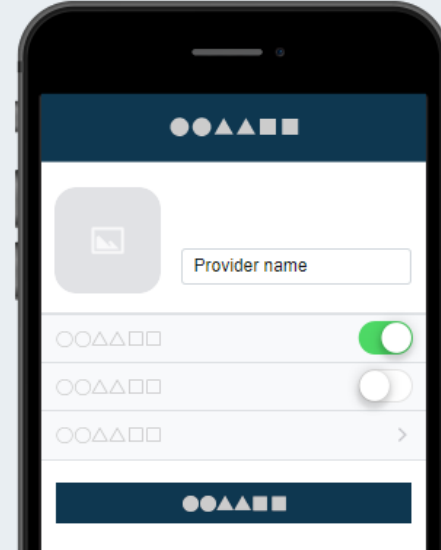
Name of new provider

Add

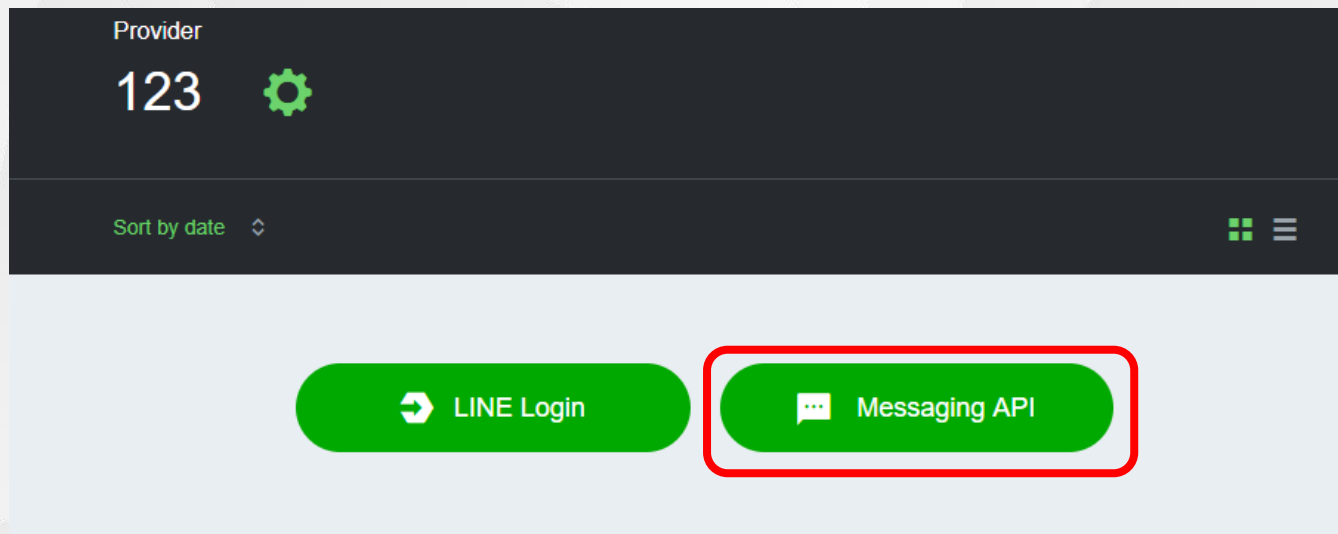
Max: 100 characters

—

Next page



# 註冊流程

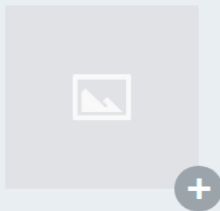


# 註冊流程



## STEP 2 Enter information for the Messaging API

### App icon



Under 3MB; JPEG/PNG/GIF/BMP

### App name

Enter app name

Max: 20 characters

### App description

Enter app description

Max: 500 characters





# 註冊流程

智慧機器人2

Max: 20 characters

## App description

Enter app description

Max: 500 characters

## Plan



### Developer Trial

A trial plan which lets you create a bot that can send push messages and have up to 50 friends.

Note: You cannot upgrade or buy a premium ID for a Developer Trial plan.



### Free

A plan which lets you create a bot with an unlimited number of friends. Push messages cannot be sent with this plan.

Note: You can upgrade this plan at any time.

## Category

Select category



## Subcategory

Select subcategory



Email address

example@line.me



For display purposes only.



選擇Developer Trial

# 註冊流程

プラン

Developer Trial

For display purposes only.

Category

公認人物、專業人士

Subcategory

公認人物、專業人士（其他）

Email address

n0912104416@gmial.com

☒ [LINE@ Terms of Use](#) ☐ I have read and agree to the Terms of Use.

☒ [Messaging API \(Developer Trial plan\) Terms of Use:](#) ☐ I have read and agree to the Terms of Use.

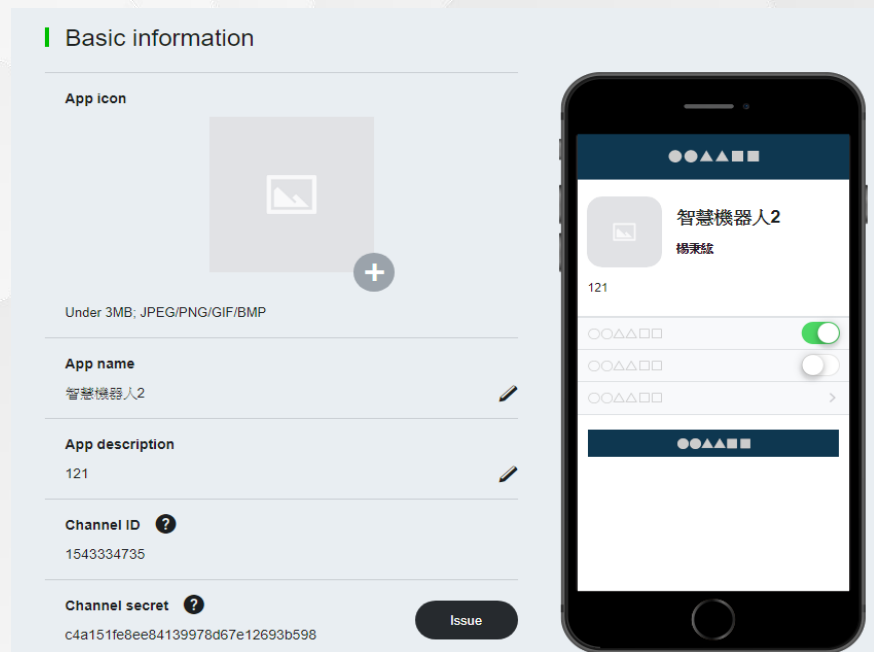
Edit

Create

填寫基本資料

建立機器人

# 註冊流程



# 註冊流程

## Messaging settings

Channel access token (long-lived) ?

Issue

Use webhooks ?

Disabled

Webhook URL Requires SSL ?

Allow bot to join group chats ?

Disabled

## Using LINE@ features

Message text for LINE@ features are set on the LINE@ Manager.

Auto-reply messages ?

Enabled

Greeting messages ?

Enabled

### Issue a new channel access token

Your current channel access token becomes invalid when a new token is issued. Set the amount of time (up to 24 hours) until the token becomes invalid and then click the "Issue" button.

Time until token becomes invalid 0 ▾ hours

Cancel

Issue

[Set message](#)

# 註冊流程

## Messaging settings

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

NhNkNgEA4TiguMnchF5dcjfvZyQ0NDcdFsoEQ1Q3iU9AAYGMvWw+mhEI9AfQAY18gOhICkac7  
O5dl8j/EZvb0DVwDMBVJ7Az+F6peH3PLcKtVOJeA64DVW9oGfaCDZN+L2w8D5sQtAYIsn35VaOx  
wdB04t89/1O/w1cDnyllFU=

Issue

成功產生access token

### Use webhooks ?

Disabled

### Webhook URL Requires SSL ?

—

### Allow bot to join group chats ?

Disabled

## Using LINE@ features

Message text for LINE@ features are set on the LINE@ Manager.

### Auto-reply messages ?

Enabled

[Set message](#)

### Greeting messages ?

# 註冊流程

Channel access token (long-lived) ?

[Redacted token]

Issue

啟用 webhooks

Use webhooks ?



Enabled



Disabled

Update

Cancel

Webhook URL Requires SSL ?

—





Allow bot to join group chats ?

Disabled

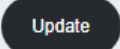
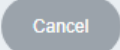




# 註冊流程

**Webhook URL** Requires SSL ?  
— 

**Allow bot to join group chats** ?  
Disabled 

**Using LINE@ features**  
Message text for LINE@ features are set on the LINE@ Manager.

**Auto-reply messages** ?  
☐ Enabled ☒ Disabled  

**Greeting messages** ?  
Enabled  

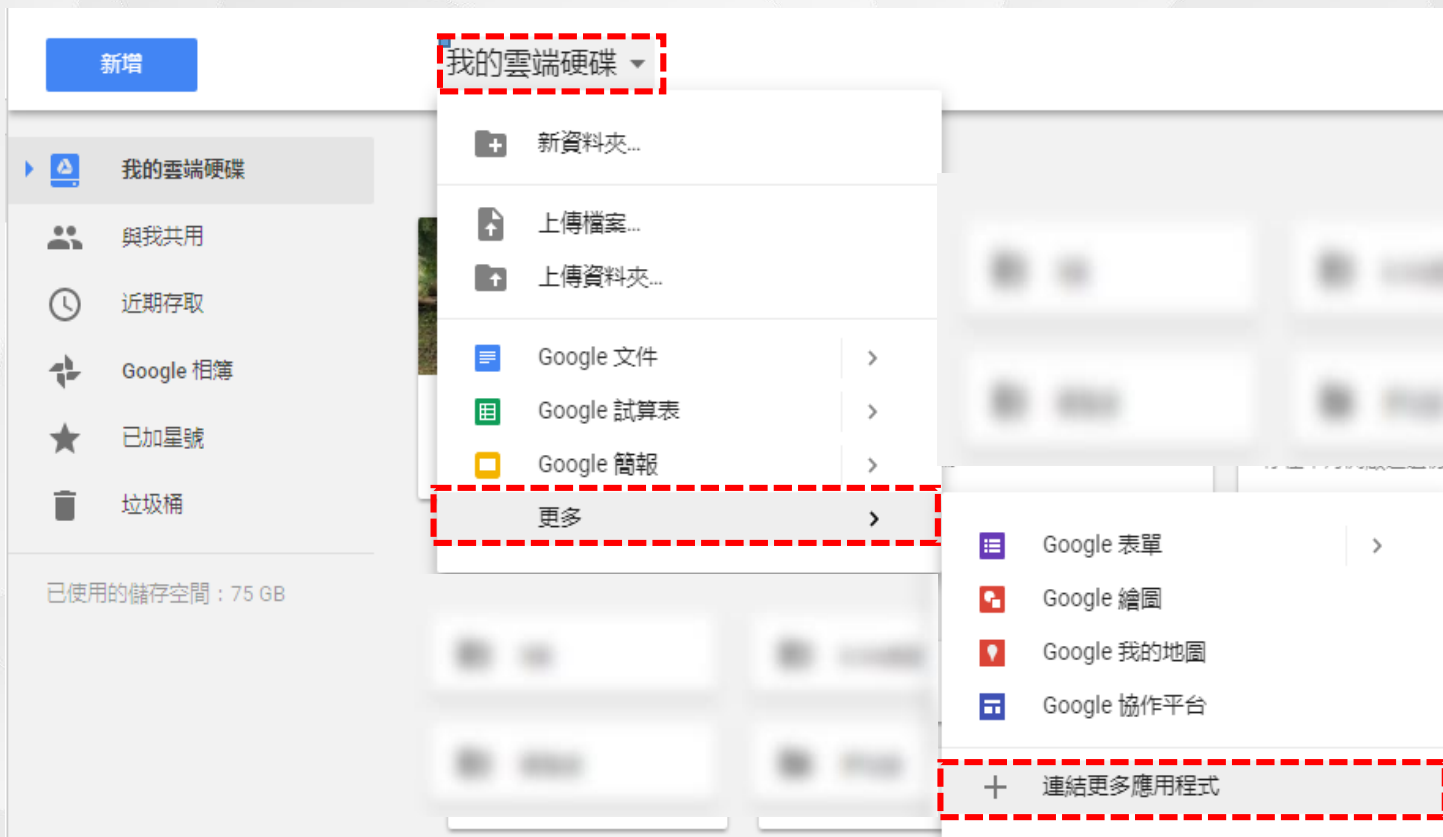
關閉 自動回話功能



# Google app script



# 雲端硬碟 新增 google app script




# 雲端硬碟 新增 google app script

將應用程式連接到雲端硬碟

全部

google app script




Google Apps Script

**Google Apps Script**  
來源網站: google.com  
Create and edit scripts

+ 連接

實用工具

★★★★★ (1001)




Google Apps Script Power Pack

**Apps Script Power Pack, Cloud Zip**  
來源網站: appscriptpack.youfiles.net  
Power tools with Google Apps Script, mostly for automating your Zip, Google Drive, Docs, Gmail. Upgrade your productivity.

+ 連接

公用程式

★★★★★ (65)



RSS Service for Google Drive

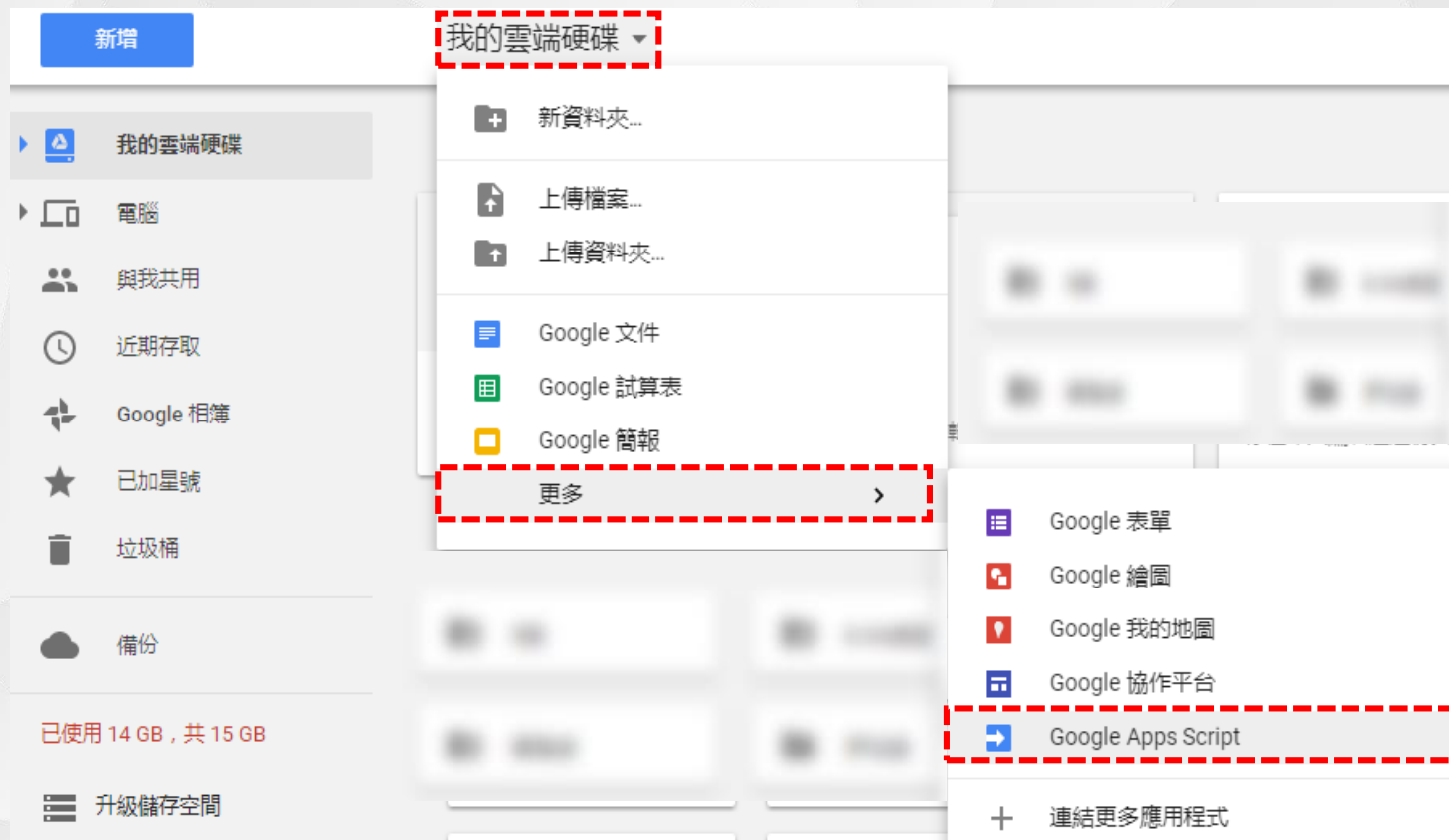
**Drive RSS, RSS Service for Google Drive**  
來源網站: driverss.nsspot.net  
RSS service for Google Drive share folders using Google Apps Script. Provide RSS/Notification on file update and recent list.

+ 連接

實用工具

★★★★★ (19)

# 開啟 google app script



# 專案命名



# REPLAY API 製作基本回話機器人

# REPLY API





# REPLY API

## HTTP request

POST `https://api.line.me/v2/bot/message/reply`

## Request headers

Request header	Description
Content-Type	application/json
Authorization	Bearer <code>{Channel Access Token}</code>

## Request body

Include an array of `send message objects` for your message.

Field	Type	Required	Description
replyToken	String	Yes	<code>replyToken</code> received via webhook
messages	Array of <code>send message objects</code>	Yes	Messages Max: 5

# 訊息json格式

```
"events": [  
  {  
    "replyToken": "nHuyWiB7yP5Zw52FIkcQobQuGDXCTA",  
    "type": "message",  
    "timestamp": 1462629479859,  
    "source": {  
      "type": "user",  
      "userId": "U206d25c2ea6bd87c17655609a1c37cb8"  
    },  
    "message": {  
      "id": "325708",  
      "type": "text",  
      "text": "Hello, world"  
    }  
  },  
],
```

# 程式撰寫

```
var CHANNEL_ACCESS_TOKEN = 'your token';
```

```
//抓取IP位置
```

```
function doPost(e) {
```

```
    var events = JSON.parse(e.postData.contents).events[0]; //event事件
```

```
    var reply_token = events.replyToken;
```

```
    var json = JSON.parse(e.postData.contents);
```

```
    switch(json.events[0].type){
```

```
        case 'message':
```

```
            messages = [ {'type': 'text', 'text': events.message.text}];
```

```
            break;
```

```
    }
```

# 程式撰寫

```
if (typeof reply_token === 'undefined')  
    return;  
var url = 'https://api.line.me/v2/bot/message/reply';  
var header = {  
    'Content-Type': 'application/json; charset=UTF-8',  
    'Authorization': 'Bearer ' + CHANNEL_ACCESS_TOKEN,  
}  
var payload = {  
    'replyToken': reply_token,  
    'messages' : messages  
}
```

# 程式撰寫

```
var options = {  
  'headers': header,  
  'method': 'post',  
  'payload': JSON.stringify(payload)  
}
```

```
UrlFetchApp.fetch(url, options);
```

```
return ContentService.createTextOutput(JSON.stringify(  
  {'content': 'post ok'})).setMimeType(ContentService.MimeType.JSON);  
}
```

# 更改access token



# 部屬應用程式





# 部屬應用程式

部署為網路應用程式

專案版本：

新增 ▼

請描述這個版本有哪些變更...

將應用程式執行為：

我 (n0912104416@gmail.com) ▼

您必須在發佈網址前授權指令碼。

具有應用程式存取權的使用者：

只有我自己 ▼

只有我自己

任何人

任何人，甚至是匿名使用者

部署

取消

說明

部署為網路應用程式

專案版本：

新增 ▼

請描述這個版本有哪些變更...

將應用程式執行為：

我 (n0912104416@gmail.com) ▼

您必須在發佈網址前授權指令碼。

具有應用程式存取權的使用者：

任何人，甚至是匿名使用者 ▼

部署

取消

說明

# 取得 應用程式網址(webhooks)

部署為網路應用程式

專案版本：  
1

需要授權

智慧機器人 needs your permission to access your data on Google.

核對權限

取消

部署

取消

說明

部署為網路應用程式


本專案現已部署為網路應用程式。

目前的網路應用程式網址：  
<https://script.google.com/macros/s/AKfycbyY2FeoXK90cWx2>

針對最新的程式碼測試網路應用程式。


確定

# 更改 Webhook URL

Use webhooks 

☒ Enabled ☐ Disabled Update Cancel


將網址貼至Webhook URL

Webhook URL Requires SSL 

https://

Max: 500 characters Update Cancel

Please enter a URL that supports HTTPS.

Allow bot to join group chats 

Disabled

部署為網路應用程式

本專案現已部署為網路應用程式。

目前的網路應用程式網址：  
`https://script.google.com/macros/s/AKfycbyY2FeoXK90cWx2`  
針對最新的程式碼測試網路應用程式。

確定

# 驗證 Webhook URL

Use webhooks ?



Enabled



Disabled

Update

Cancel

Webhook URL Requires SSL ?

`https://script.google.com/macros/s/AKfycbyY2FeoXK90cWx2te61-6rsAXPmnjmJUOKLsWKg7YHKO50mbDW/exec`

Verify



Success

驗證成功

Allow bot to join group chats ?

Disabled

# 加入機器人好友

QR code of your bot

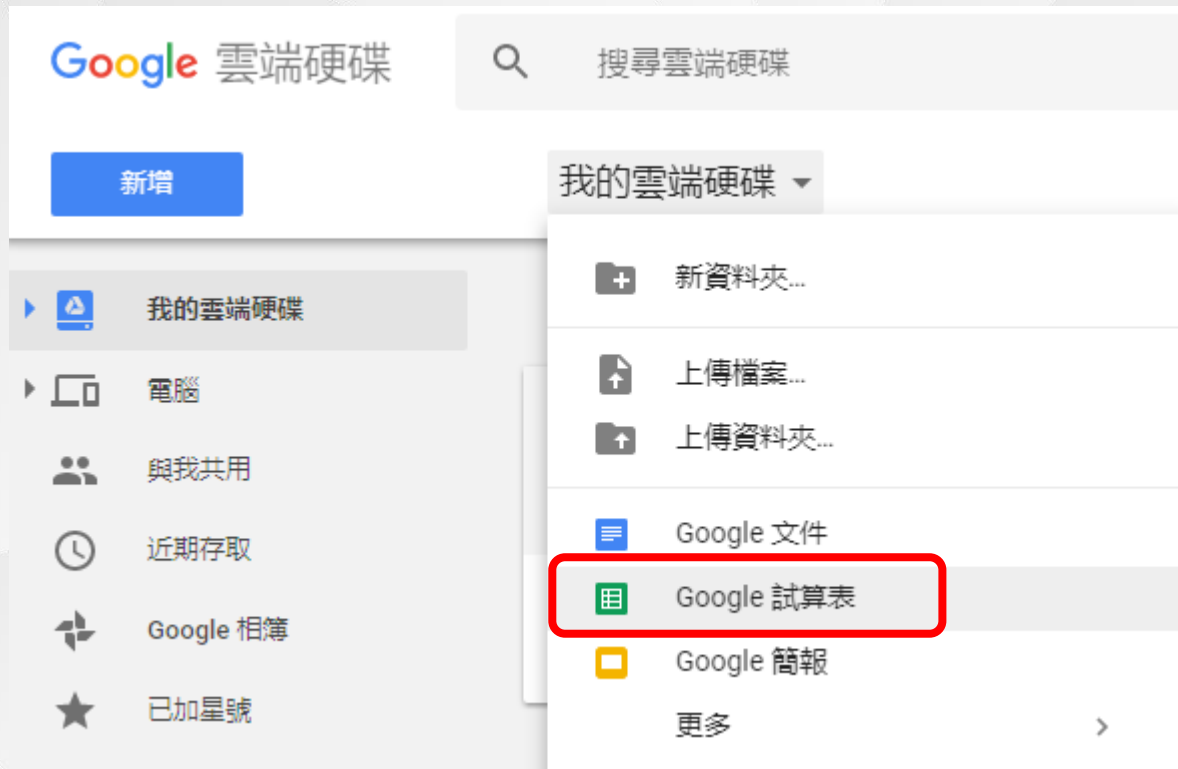
QR code



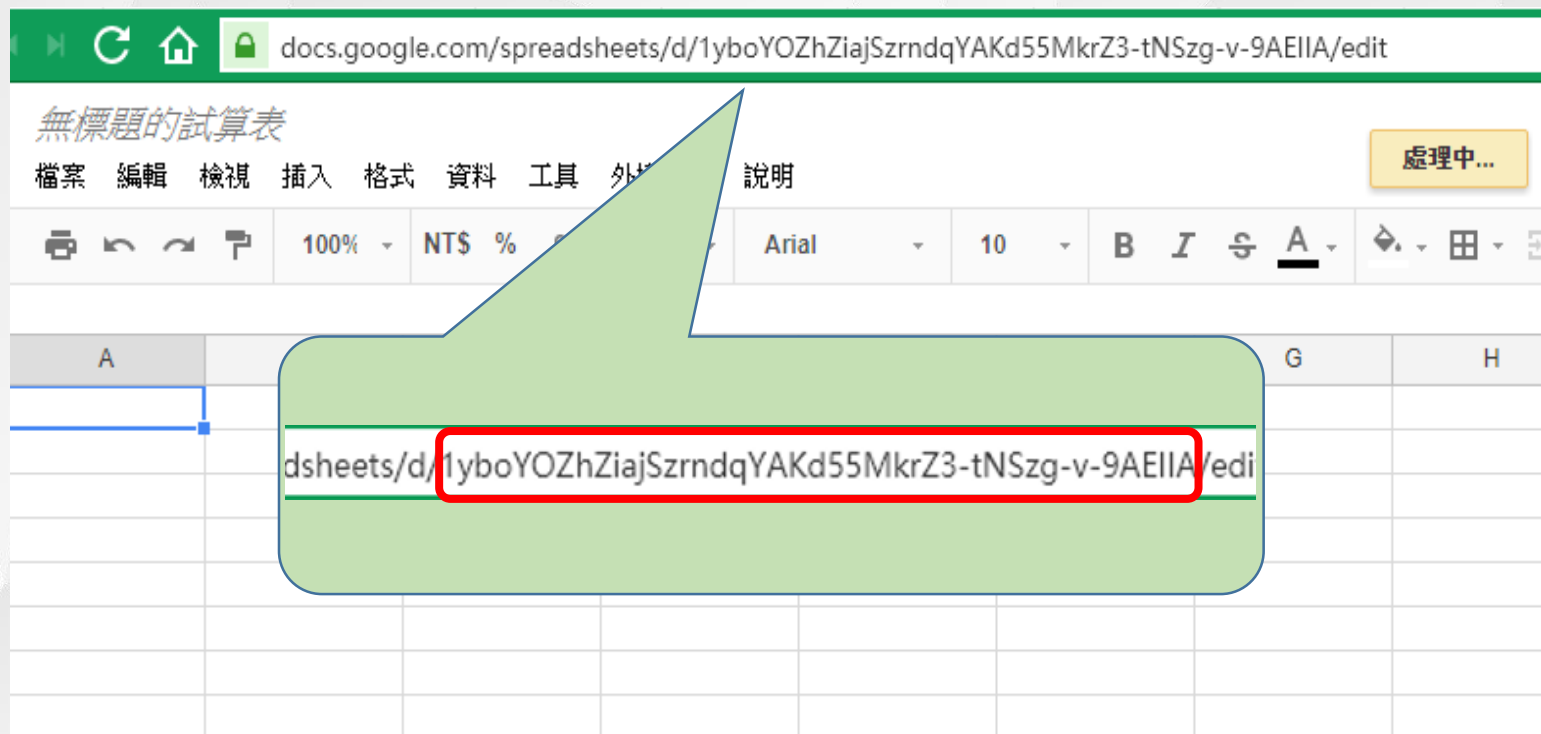
# 製作資料庫 GOOGLE 試算表寫入



# 建立 Google 試算表



# 取得表單 openid





# Google sheet 基本功能

- (1) SpreadsheetApp.openById("sheet 網址")
- (2) Spreadsheet.getSheetByName("工作表名稱")
- (3) Sheet.getLastRow()
- (4) Sheet.getRange( LastRow+1,1).setValue(資料)

# 程式撰寫

```
var CHANNEL_ACCESS_TOKEN = 'YOUR TOKEN';
```

```
function doPost(e) {  
  var events = JSON.parse(e.postData.contents).events[0]; //event事件  
  var reply_token = events.replyToken;  
  var json = JSON.parse(e.postData.contents);  
  var SpreadSheet = SpreadsheetApp.openById("sheet 網址"); //sheet 網址  
  var Sheet1 = SpreadSheet.getSheetByName("工作表1");  
  var LastRow1 = Sheet1.getLastRow();
```

# 程式撰寫

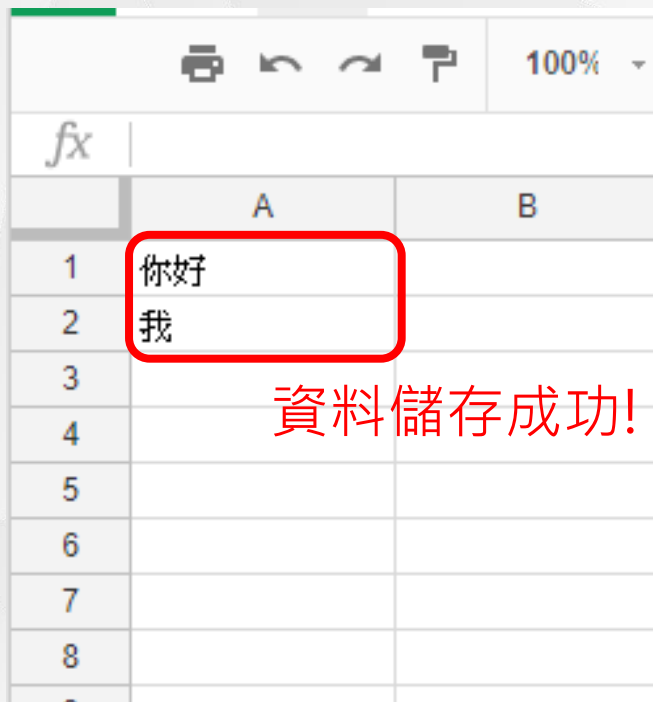
```
switch(json.events[0].type){
  case 'message':
    var data1 =events.message.text;
    Sheet1.getRange( LastRow1+1,1).setValue(data1);
    messages = [ {'type': 'text', 'text': "資料儲存成功!"}];
    break;
}
if (typeof reply_token === 'undefined')
return;
var url = 'https://api.line.me/v2/bot/message/reply';
var header = {
  'Content-Type': 'application/json; charset=UTF-8',
  'Authorization': 'Bearer ' + CHANNEL_ACCESS_TOKEN,
}
```

# 程式撰寫

```
var payload = {
  'replyToken': reply_token,
  'messages' : messages
}
var options = {
  'headers': header,
  'method': 'post',
  'payload': JSON.stringify(payload)
}

UrlFetchApp.fetch(url, options);
return ContentService.createTextOutput(JSON.stringify(
  {'content': 'post ok'})).setMimeType(ContentService.MimeType.JSON);
}
```

# Google sheet 資料儲存



A screenshot of a Google Sheet interface. The top toolbar shows icons for print, undo, redo, and a zoom level of 100%. The formula bar shows 'fx'. The spreadsheet has two columns, A and B, and rows numbered 1 to 8. In row 1, column A, the text '你好' (Hello) is entered. In row 2, column A, the text '我' (I) is entered. A red rectangle highlights the cell containing '你好'.

	A	B
1	你好	
2	我	
3		
4		
5		
6		
7		
8		

資料儲存成功!



# GOOGLE 試算表 存取USERID



# 如何取得Userid

## | Other

Your user ID



U5efa14cae71b0a8d40f5de0862f93d44

# Follow event

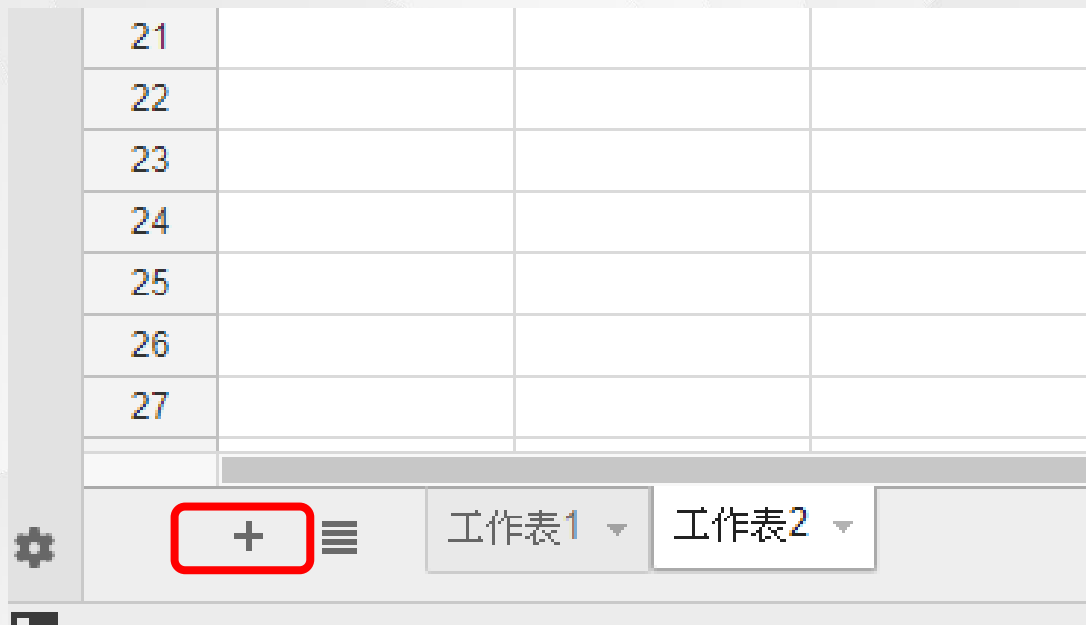
## Follow event example

```
{
  "replyToken": "nHuyWiB7yP5Zw52FIkcQobQuGDXCTA",
  "type": "follow",
  "timestamp": 1462629479859,
  "source": {
    "type": "user",
    "userId": "U206d25c2ea6bd87c17655609a1c37cb8"
  }
}
```

`userid=events.source.userId`



# 新增工作表



# 程式撰寫

```
var CHANNEL_ACCESS_TOKEN = 'YOUR TOKEN';
```

```
function doPost(e) {
```

```
    var events = JSON.parse(e.postData.contents).events[0]; //event事件
```

```
    var reply_token = events.replyToken;
```

```
    var json = JSON.parse(e.postData.contents);
```

```
    var Spreadsheet = SpreadsheetApp.openById("sheet 網址");
```

```
    var Sheet1 = Spreadsheet.getSheetByName("工作表1");
```

```
    var Sheet2 = Spreadsheet.getSheetByName("工作表2");
```

```
    var LastRow1 = Sheet1.getLastRow();
```

```
    var LastRow2 = Sheet2.getLastRow();
```

```
    var userid=events.source.userId;
```

# 程式撰寫

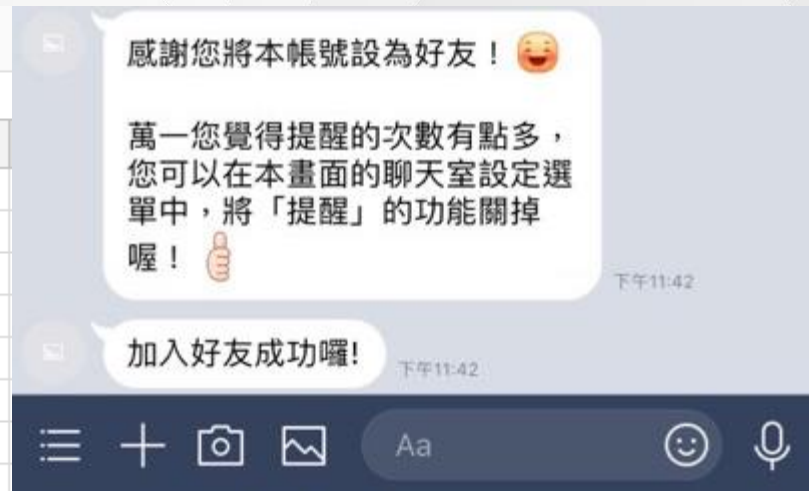
```
switch(json.events[0].type){
  case 'message':
    var data1 =events.message.text;
    Sheet1.getRange( LastRow1+1,1).setValue(data1);
    messages = [ {'type': 'text', 'text': "資料儲存成功!"}];
    break;
  case 'follow':
    Sheet2.getRange( LastRow2+1,1).setValue(userid);
    messages = [ {'type': 'text', 'text': "加入好友成功囉!"}];
    break;
}
if (typeof reply_token === 'undefined')
  return;
var url = 'https://api.line.me/v2/bot/message/reply';
```

# 程式撰寫

```
var header = {  
  'Content-Type': 'application/json; charset=UTF-8',  
  'Authorization': 'Bearer ' + CHANNEL_ACCESS_TOKEN, }  
var payload = {  
  'replyToken': reply_token,  
  'messages' : messages  }  
var options = {  
  'headers': header,  
  'method': 'post',  
  'payload': JSON.stringify(payload)  }  
UrlFetchApp.fetch(url, options);  
return ContentService.createTextOutput( JSON.stringify(  
  {'content': 'post ok'})).setMimeType(ContentService.MimeType.JSON);  
}
```

# 儲存userid

100% NT\$ % .0 .00 123		
fx	成功獲得 userid	
	A	B
1	U5efa14cae71b0a8d40f5de0862f93d44	
2	U8bbcbd880765d3c836ec4553cfde98e1	
3		
4		
5		
6		
7		
8		



# PUSH API

# PUSH API



# PUSH API

## HTTP request

POST `https://api.line.me/v2/bot/message/push`

## Request headers

Request header	Description
Content-Type	application/json
Authorization	Bearer {Channel Access Token}

## Request body

Include an array of [send message objects](#) for your message.

Field	Type	Required	Description
to	String	Yes	ID of the receiver
messages	Array of <a href="#">send message objects</a>	Yes	Messages Max: 5



# 程式撰寫

```
var CHANNEL_ACCESS_TOKEN = 'YOUR TOKEN';
```

```
function doPost(e) {  
  var events = JSON.parse(e.postData.contents).events[0]; //event事件  
  var reply_token = events.replyToken;  
  var json = JSON.parse(e.postData.contents);  
  var Spreadsheet = SpreadsheetApp.openById("sheet 網址"); //sheet 網址  
  var Sheet1 = Spreadsheet.getSheetByName("工作表1");  
  var Sheet2 = Spreadsheet.getSheetByName("工作表2"); //已經下的訂單  
  var LastRow1 = Sheet1.getLastRow();  
  var LastRow2 = Sheet2.getLastRow();  
  var payload;  var userid=events.source.userId;
```

# 程式撰寫

```
switch(json.events[0].type){
  case 'message':
    var data1 =events.message.text;
    if(data1=="小明"){
      messages = [ {'type': 'text', 'text': "小明你好~"}];
      payload = {
        'to': 'friends id',
        'messages': messages } }   break;
  case 'follow':
    Sheet2.getRange( LastRow2+1,1).setValue(userid);
    messages = [ {'type': 'text', 'text': "加入好友成功囉!" }]; break; }
if (typeof reply_token === 'undefined')  return;
var url = 'https://api.line.me/v2/bot/message/push';
```

# 程式撰寫

```
var header = {  
  'Content-Type': 'application/json; charset=UTF-8',  
  'Authorization': 'Bearer ' + CHANNEL_ACCESS_TOKEN, }  
var payload = {  
  'to': 'U8bbcbd880765d3c836ec4553cfde98e1',  
  'messages': messages }  
var options = {  
  'headers': header,  
  'method': 'post',  
  'payload': JSON.stringify(payload) }  
UrlFetchApp.fetch(url, options);  
return ContentService.createTextOutput( JSON.stringify(  
  {'content': 'post ok'})).setMimeType(ContentService.MimeType.JSON);  
}
```

# 使用PUSH API 傳訊息給隔壁好友