修平科技大學

資訊網路技術系實務專題

智慧環境應用系統的開發

LINE Bot 與 Google 行事曆統計系統之實作

指導老師：蔡篤校

學生：蔣耀庭 BN105016

陳柏翰 BN105080

張倉茂 BN105015

中華民國 109 年 06 月 04 日

修平科技大學

資訊網路技術系實務專題

智慧環境應用系統的開發

LINE Bot 與 Google 行事曆統計

系統之實作

學生： BN105016 蔣耀庭

BN105080 陳柏翰

BN105015 張倉茂

指導老師：_____老師

評審老師:_____老師

_____老師

_____老師

中華民國 109 年 06 月 04 日

目錄

圖目錄

摘要

Line 在台灣已經是家喻戶曉的社群軟體，大家每天都在使用 Line，並且透過 Line 與其他使用者聊天。我們想利用這一點，來設計一套可以在 Line 上統計 Google 日曆的聊天機器人，簡單與方便為設計導向，使用者可以使用 Google 語音翻譯來說出想要執行的動作，而我們則會針對翻譯出來的句子，在進行分析並且回答，讓在操作我們所開發的聊天機器人的使用者，只需要使用語音搜尋關鍵字便能得到結果，不需要打字。

第一章 前言

1-1 動機與目的

現代科技蓬勃發展，已有許多聊天機器人出現在各大知名網站裡，取代了傳統的客服服務，方便現代人的生活。在不斷的使用其他人製作的聊天機器人後，我們也想打造屬於自己的一個機器人，因此透過此專題研究，結合大家常用的 Google 行事曆，不但可以利用我們的聊天機器人去使用 Google 行事曆，更進一步的探討聊天機器人背後運作的模式。我們以 Google apps script 程式與 Line 結合建構出智慧化互動聊天機器人，解決 Google 行事曆沒有的統計功能。Google apps script 是由 Google 推出以 JavaScript 的腳本語言，能夠連接 Google 各項服務， Line 則是提供一個平台供我們達成聊天機器人的對話功能，以達到我們系統親切性目的，因此我們以 Line 為主， Google apps script 為輔，再加上以 Google 試算表(Google Spreadsheet) 來儲存資料等，來實作智慧化功能，建構一套以 Line 平台為主的對話互動式 Google 行事曆統計系統。

1-2 軟、硬體需求

1.Line

Line Messaging API 功能。開發者即可透過 Messaging API 將自己的服務內容串聯到 LINE 如:圖 1 案例圖說明。

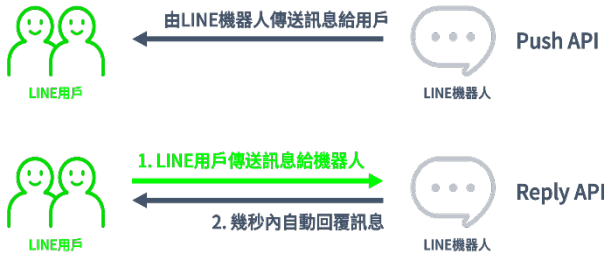


圖 1 Line@架構圖

對於服務提供商來說，可透過 LINE 聊天介面與該 LINE 帳號進行雙向互動，並操作各式線上服務，更可以於下班時間透過機器人即時回覆訊息。

2.Google Apps Script,Google 試算表和 Google 行事曆簡介

由 Google 推出基於 JavaScript 的腳本語言，它能夠輕易的將 Google 的各項服務(Gmail、Maps、Drive、Calendar、Forms)搭配應用， 透過這個功能， 便能在 Google 文件上自行開發撰寫各式各樣的功能，並透過 Google 試算表(Google Spreadsheet ）的實作儲存與讀取的功能，製作簡易資料庫功能。

第二章文獻探討

2-1 甚麼是API

API 是應用程式介面 (Application Programming Interface)，主要是扮演應用程式與應用程式之間的橋樑，它可以幫助開發者節省精力，並很快達到開發者希望的目的，因為網路平台的多樣性的發展，許多大型網站平台逐漸開放物件功能和資訊與其它網站進行串接、共享，不僅能接觸整合多元資訊，更可以提升跨平台使用者的友善性，達到平台多元發展的市占率。

2-2 Google Calendar

Google calendar 是 Google 針對個人時間管理所推出的線上服務，Google calendar 允許使用者管理自己並分享行事曆給好友，透過行事曆可以即時看到自己與親朋好友的時間安排，透過權限設定使用者可以自己決定行事曆要分享給誰、分享的程度與內容為何。

2-3 Google sheets

Google sheets 是一個和 Excel 相似的軟體，但是 sheets 是自動儲存並存放在 Google 雲端硬碟中。無論身在何處，都可以進行存取，而且也能在線上與其他 Google 帳戶進行同步、共用。在此專題中，我們建立了兩個表格，一個是先建立簡單的資料庫來儲存使用者的 Line Id、 Email、refresh_token 等等基本資料，或是一些使用者所想儲存的訊息，另一個則是負責針對句子進行解析，我們利用給予特定的單字一個編號，當句子裡有數個特定單字，就會將編號進行加總，得到的數字就會回傳相對的訊息。

2-4 Google 認證協定

Google API 使用開放授權(OAuth) 2.0 協定進行身分的驗證和授權，開放授權（OAuth）是一個開放標準，允許用戶讓第三方應用存取該用戶在某一網站上儲存的私密的資源（如相片，影片，聯絡人列表），而無需將用戶名稱和密碼提供給第三方應用。OAuth 允許用戶提供一個權杖，而不是用戶名稱和密碼來存取他們存放在特定服務提供者的資料。這樣，OAuth 讓用戶可以授權第三方網站存取他們儲存在另外服務提供者的某些特定資訊，而非所有內容，如:圖 2 案例圖說明。

圖 2Google 認證協定

2-5 Google AppsScript

App Script 是由 Google 基於 JavaScript 所開發的腳本語言，他能夠輕易地使用 Google 的各項服務，像是 Calendar、Gmail 或是 Sheet。使用 Apps Script 就可以做出像是自動化重複性任務或創建文檔等。另外我們可以公布 Web App，它會自動升成一個網址，我們就可以利用網址去連接 Line，圖 3 是目前可以

連接的服務。

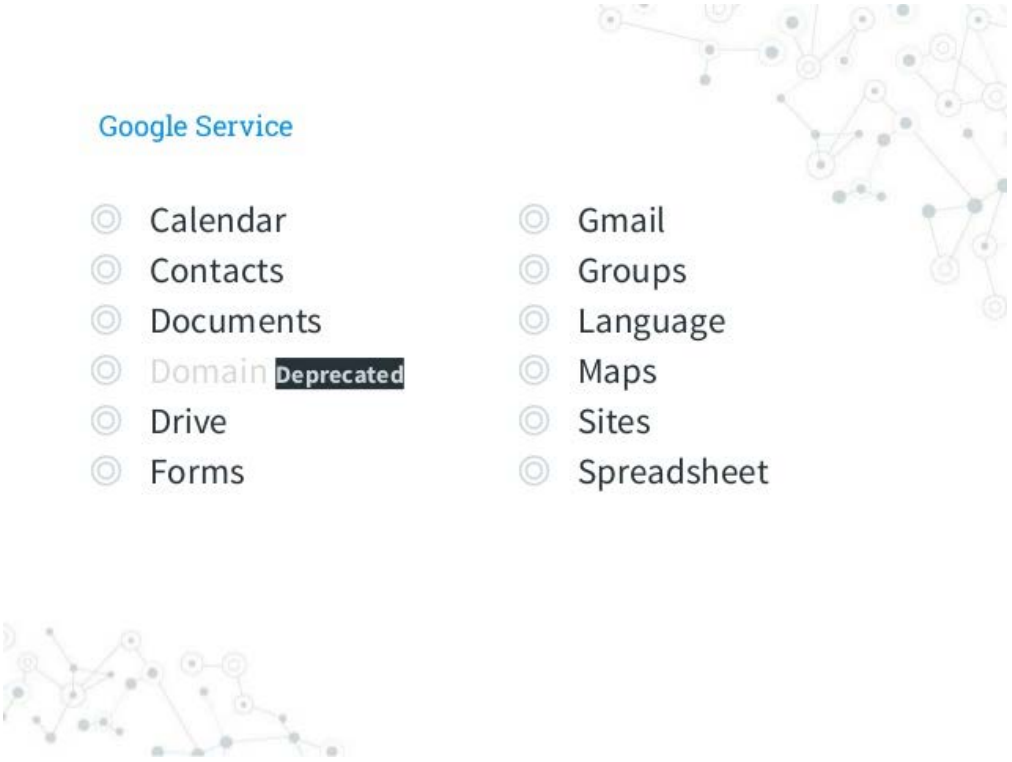


圖 3Google AppsScript

第三章系統功能

3-1 工作進度

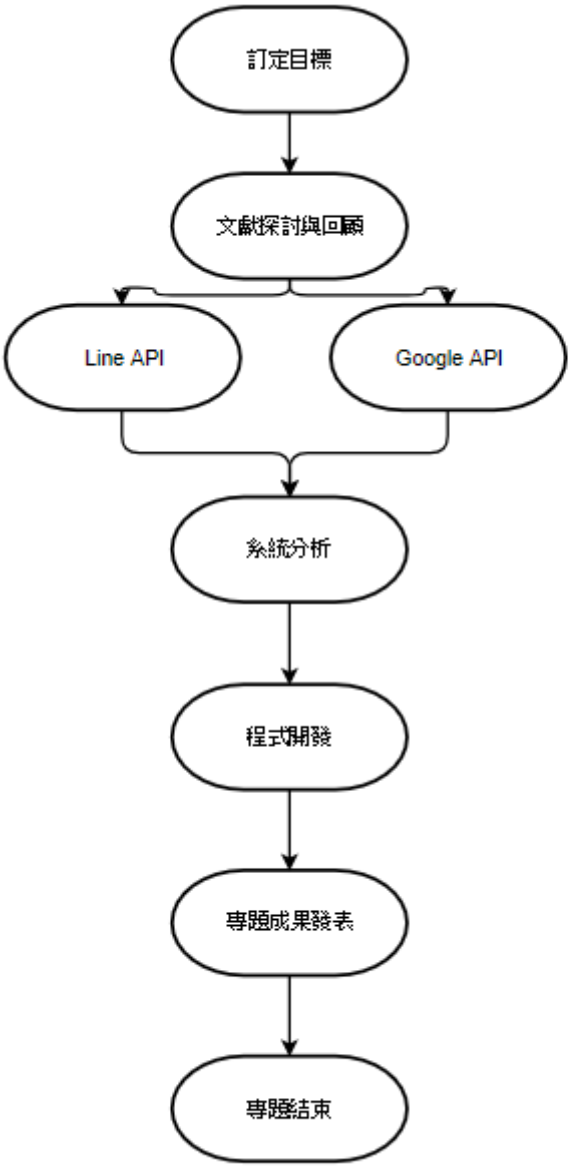


圖 4 工作進度

3-2 系統流程圖

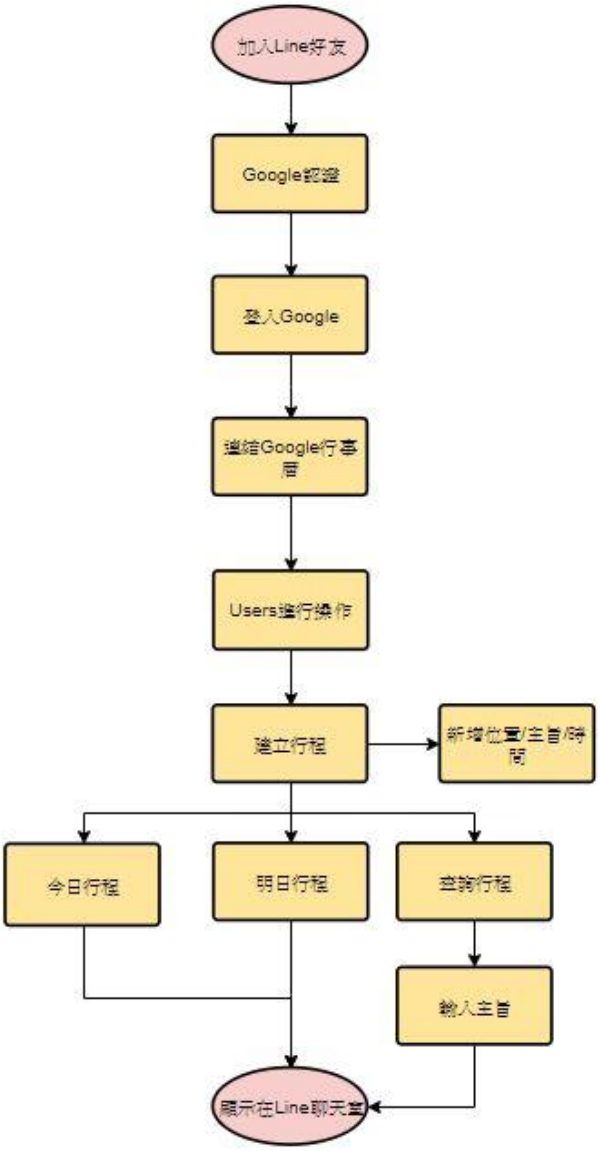


圖 5 系統流程圖

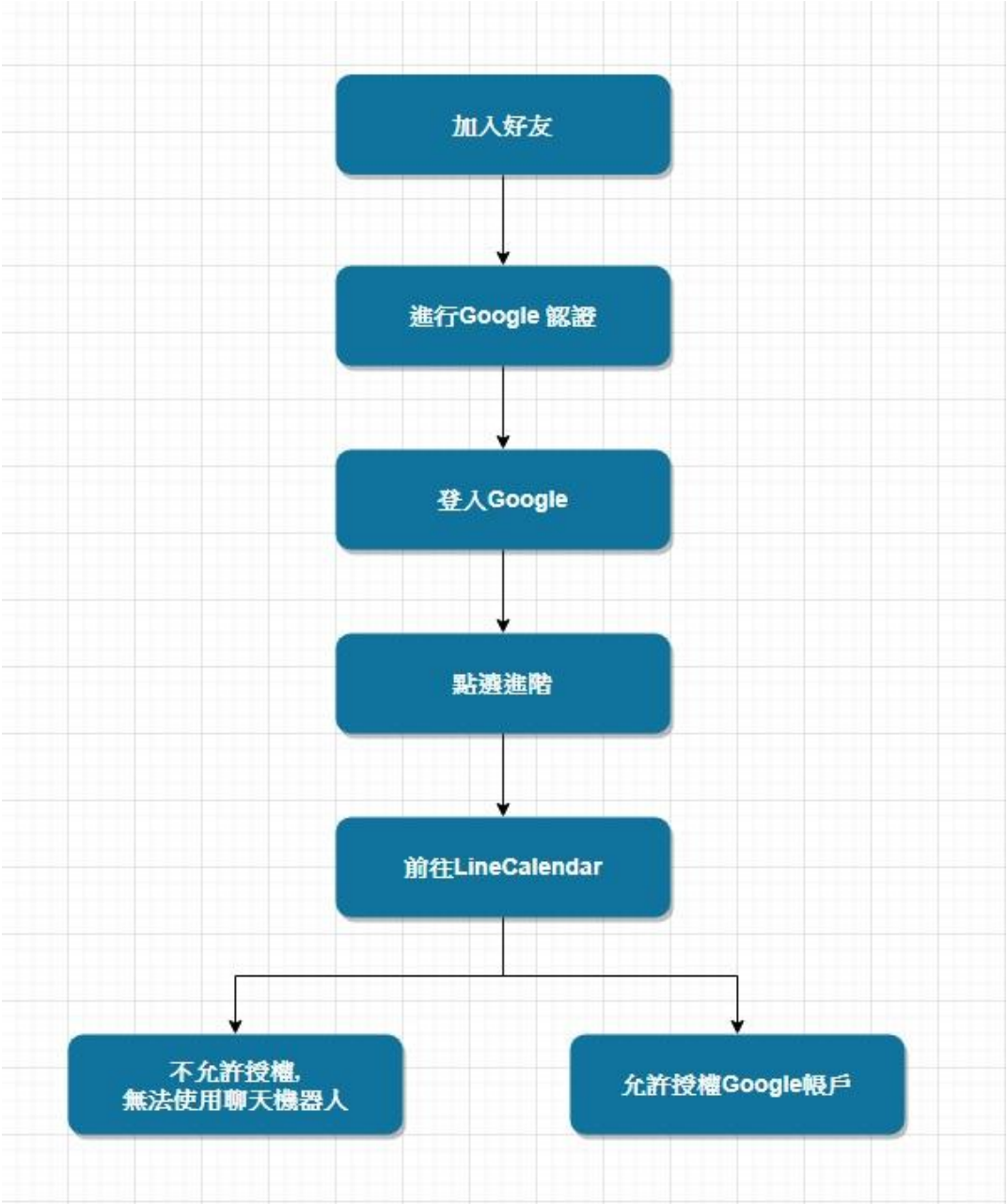

加入好友
進行Google 認證
登入Google
點選進階
前往LineCalendar
不允許授權，
無法使用聊天機器人
允許授權Google帳戶

圖 6 認證帳號流程

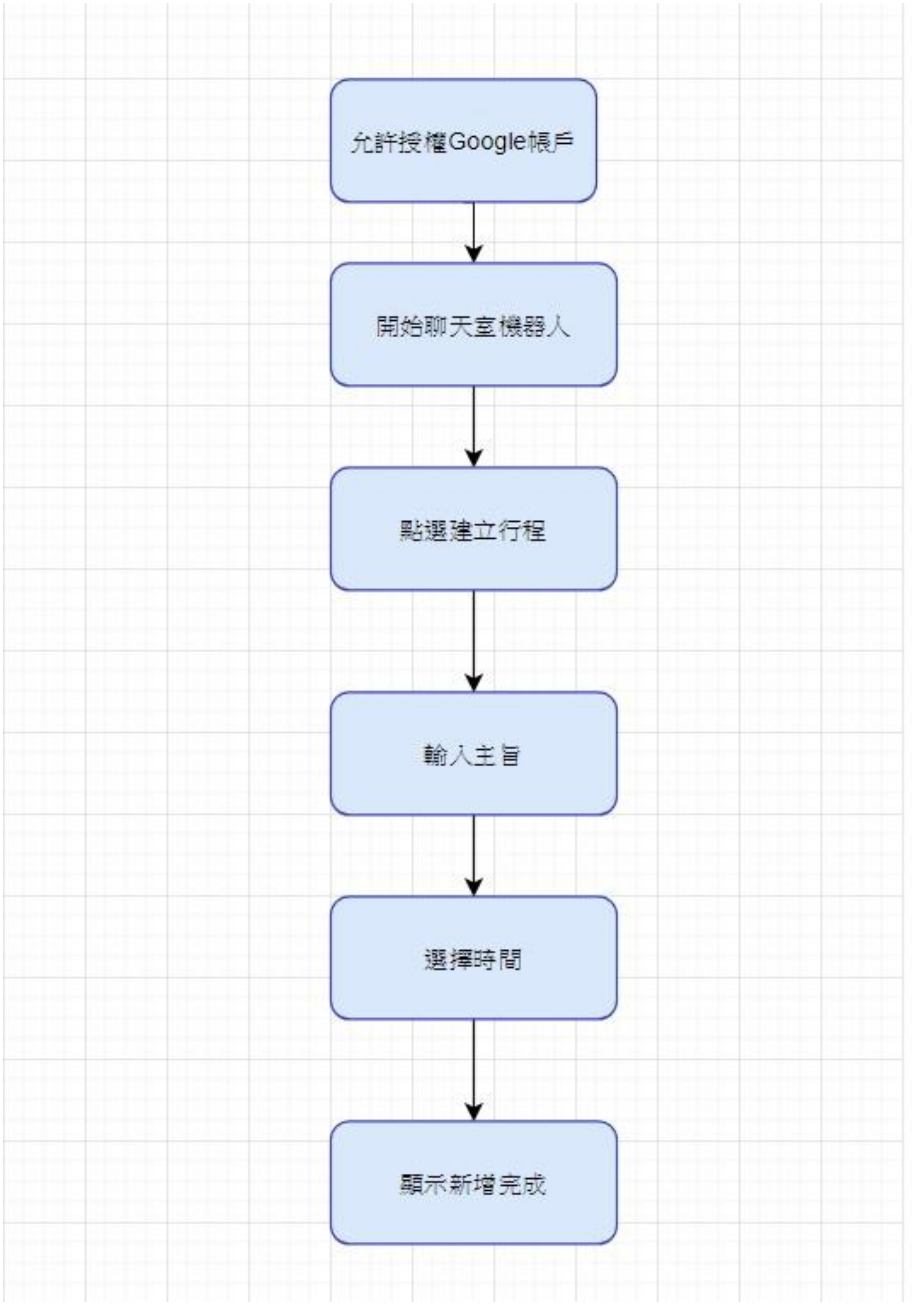


圖 7 建立行程之流程

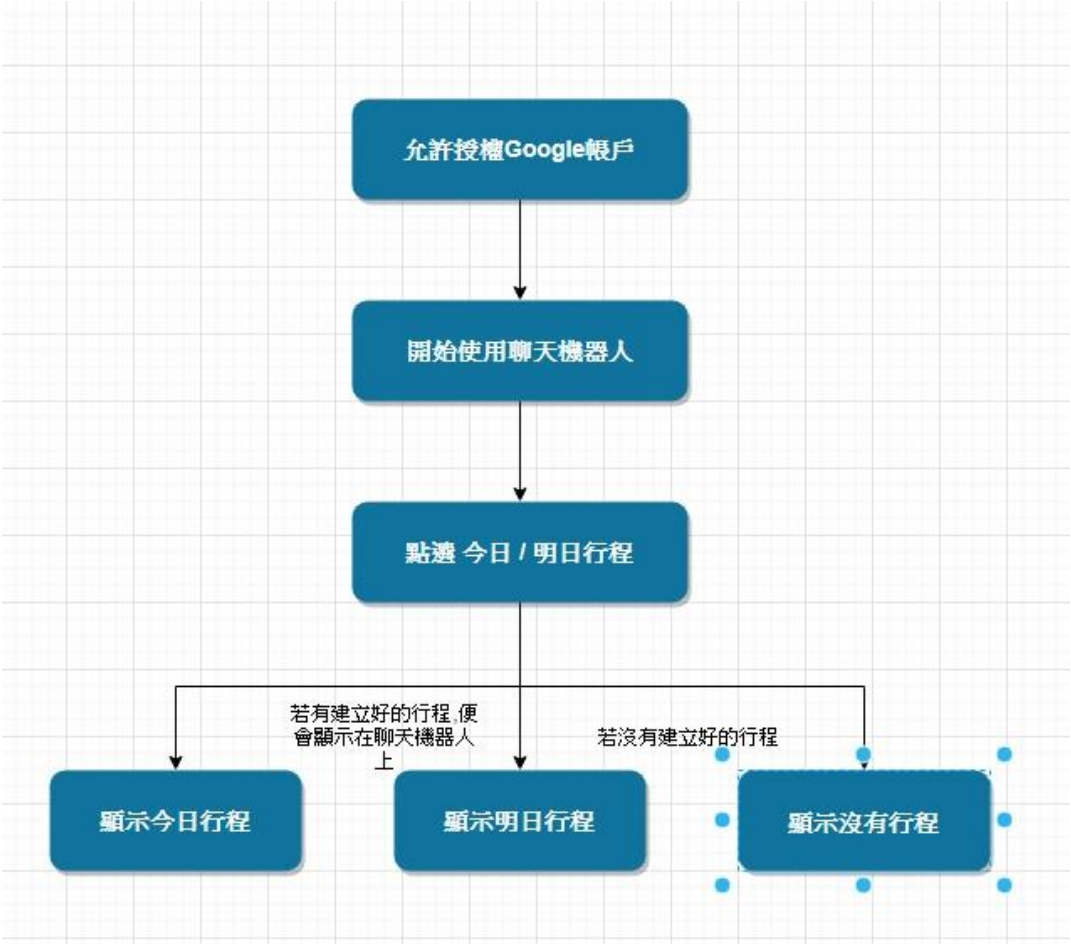


圖 8 顯示今日行程與明日行程之流程

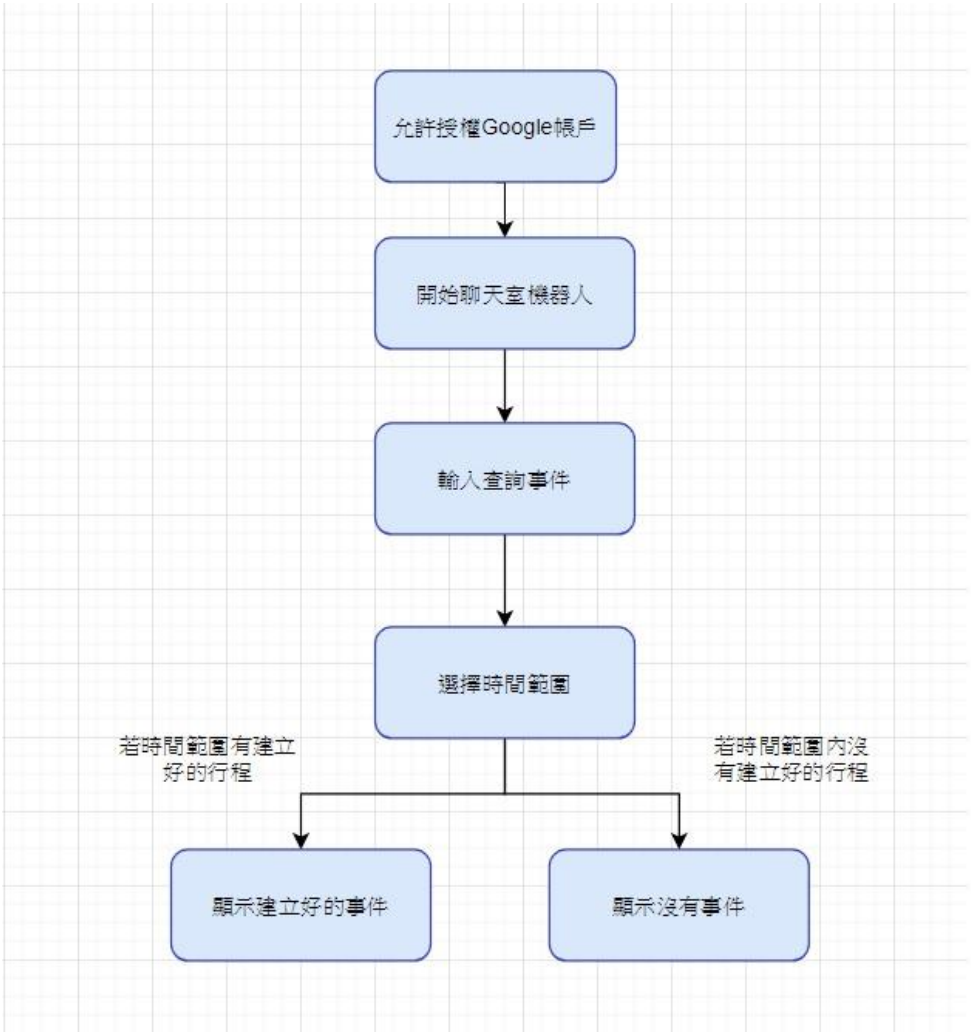

允許授權Google帳戶
開始聊天室機器人
輸入查詢事件
選擇時間範圍
若時間範圍有建立好的行程
若時間範圍內沒有建立好的行程
顯示建立好的事件
顯示沒有事件

圖 9 查詢行程之流程

第四章系統實作

一、建設 Line 官方帳號

前往網站(https://developers.line.biz/en/)並登入，如圖 10，登入或者是新增帳號。

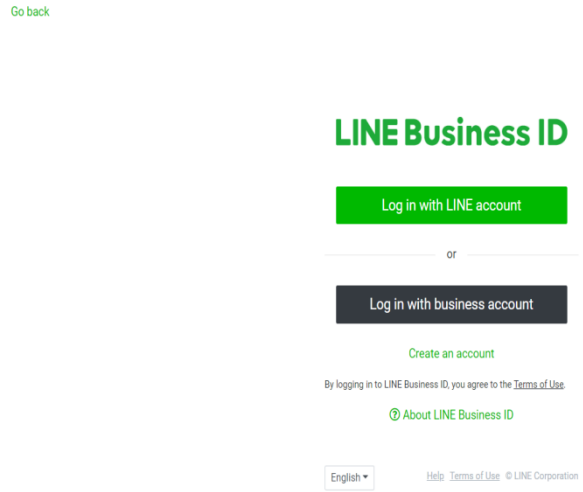


圖 10 建設 Line 官方帳號

先建立 Providers，再建立 Messaging API，如圖 11 案例圖說明。

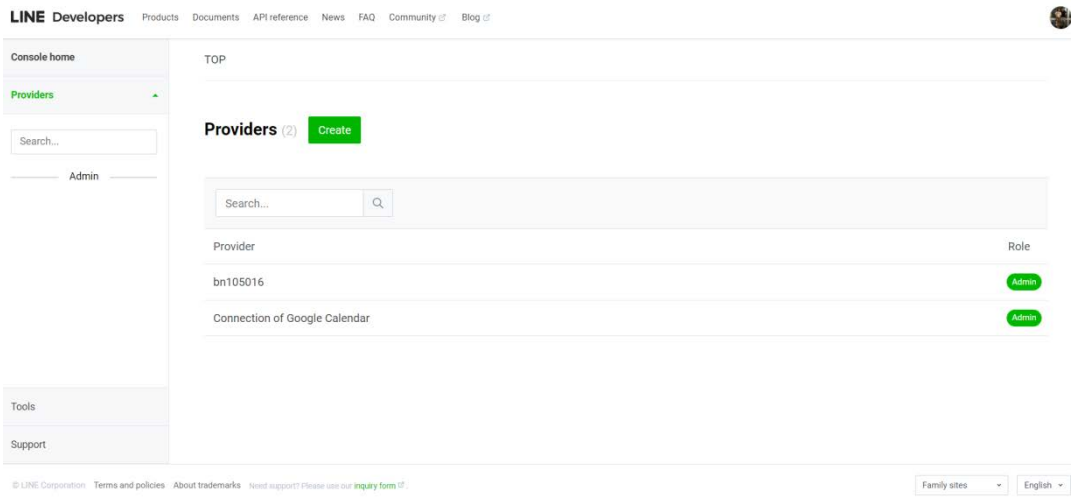


圖 11 建設 Line 官方帳號

點擊建立並輸入名稱，圖 12 案例圖說明

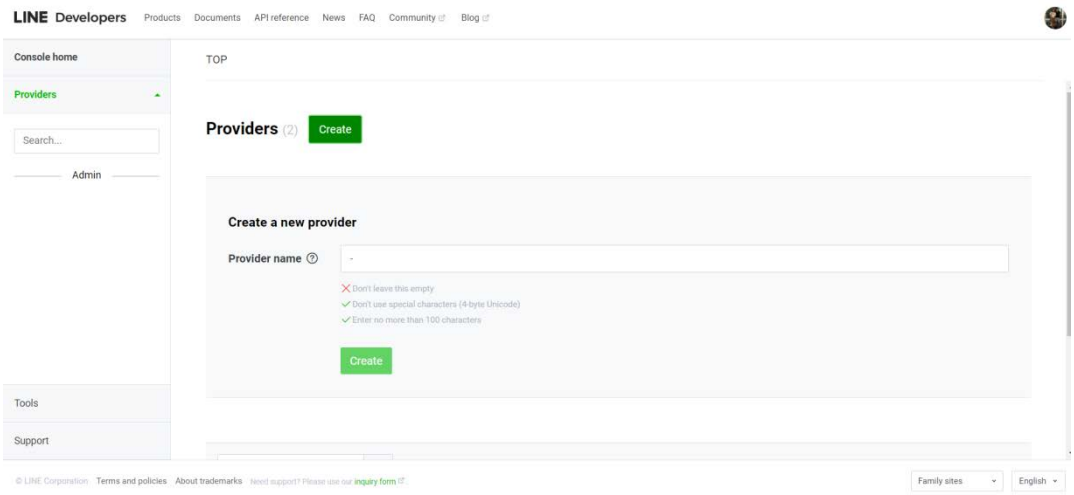


圖 11 建立 Providers

新增完後會出現如下圖，新增新的 channel，圖 12 案例圖說明

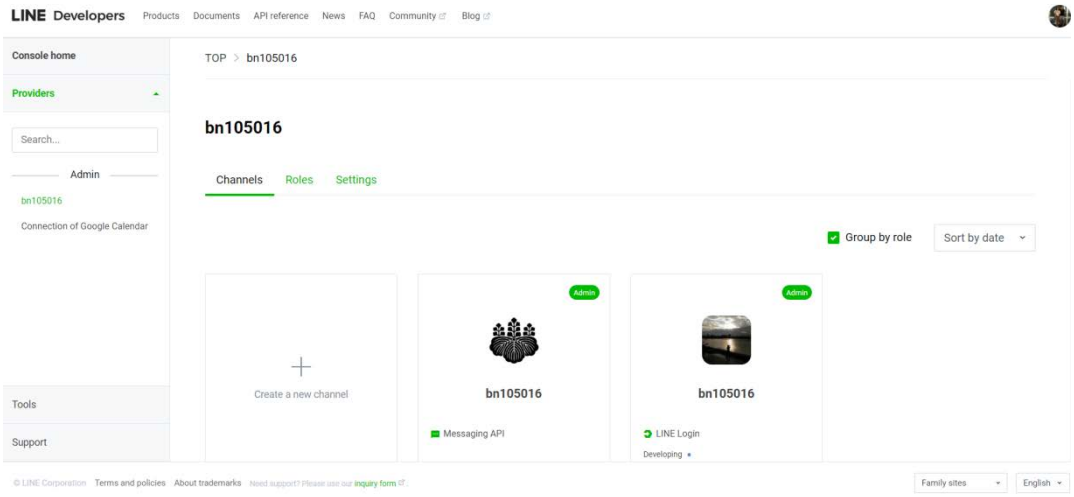


圖 12 新增新的 channel

點選 Messaging API，圖 13 案例圖說明

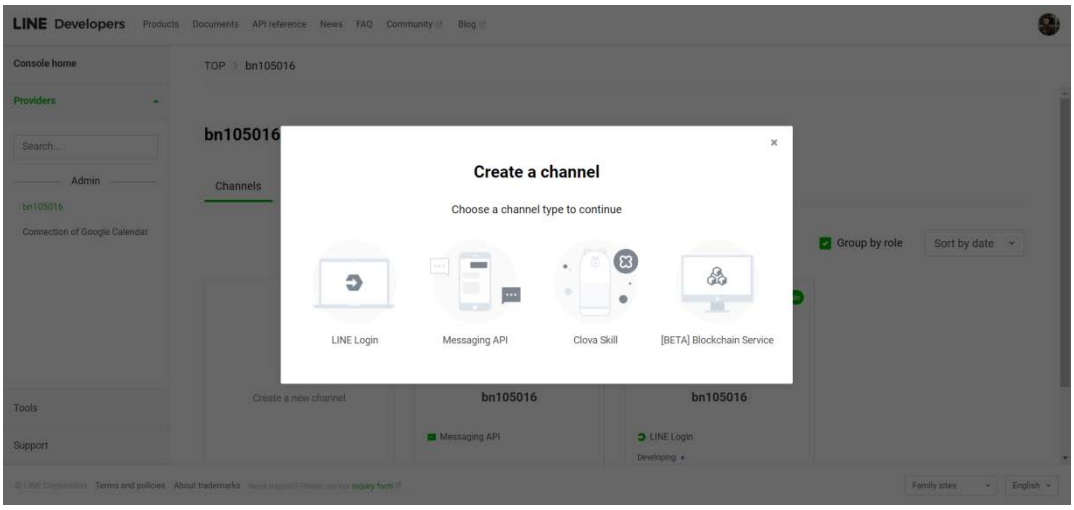


圖 13 點選 Messaging API

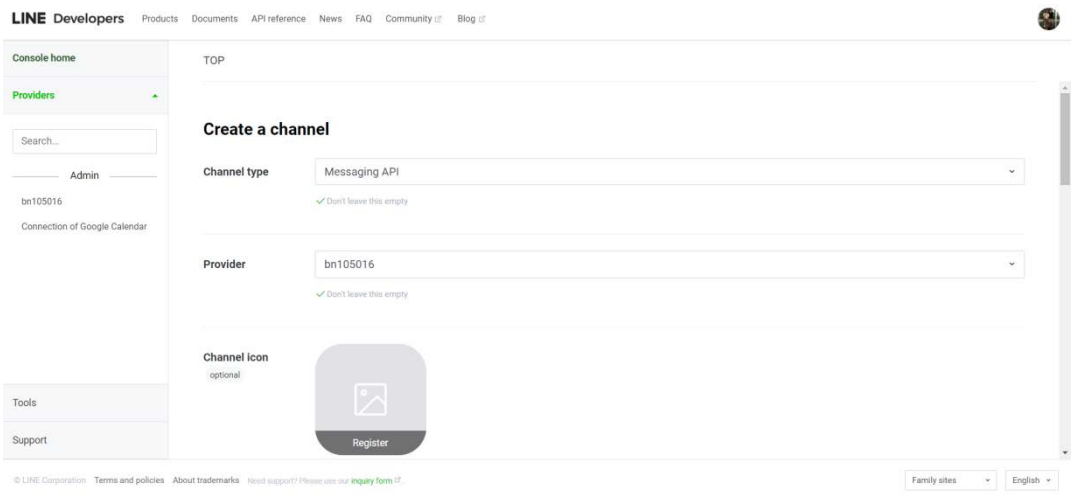


圖 14 建立 channel

新增名字，描述與種類，圖 15 案例說明

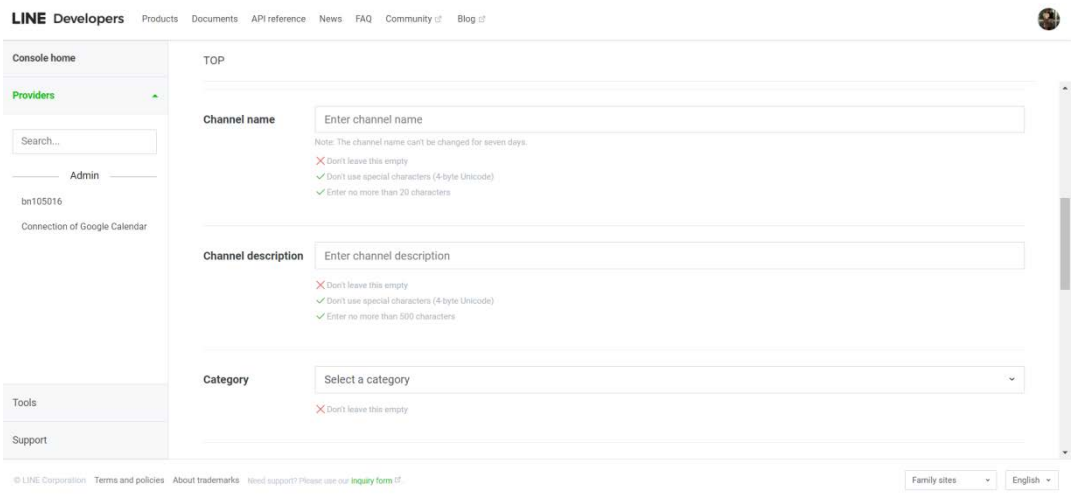


圖 15 新增 Channel 名字

新增子種類、Email，圖 16 案例說明

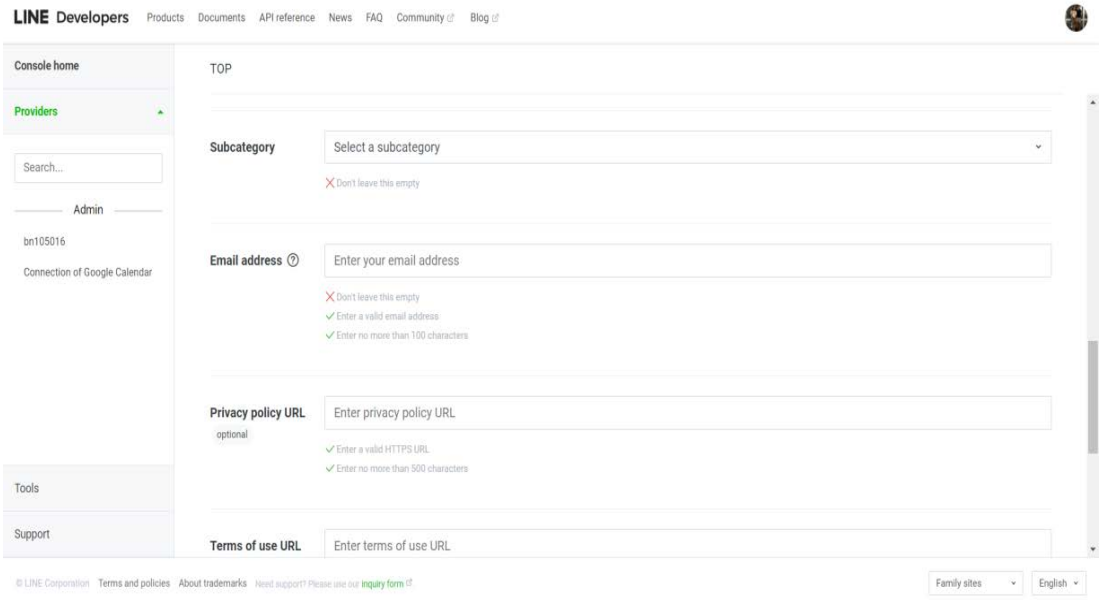


圖 16 新增其他項目

最後勾選同意並創立，圖 17 案例說明

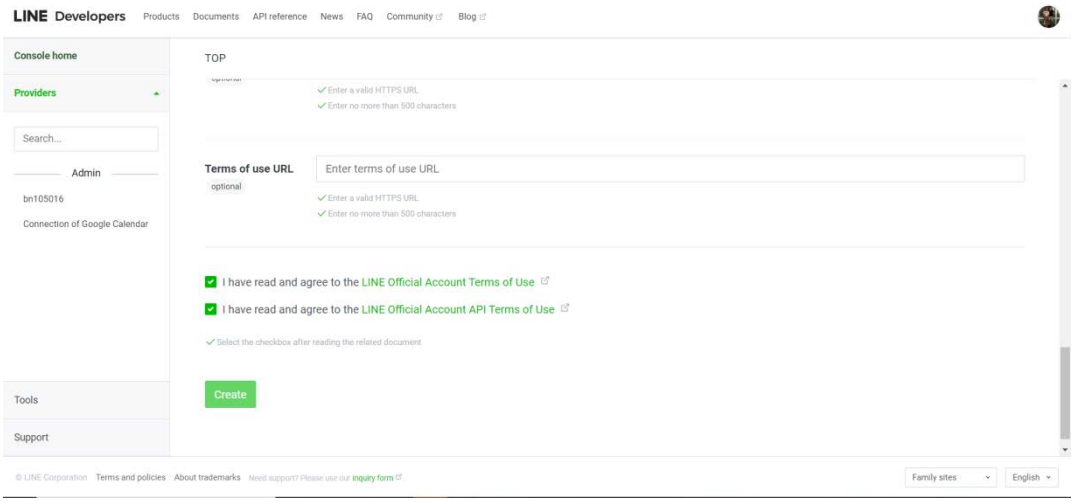


圖 17 同意並建立

建立完成，圖 18 案例說明

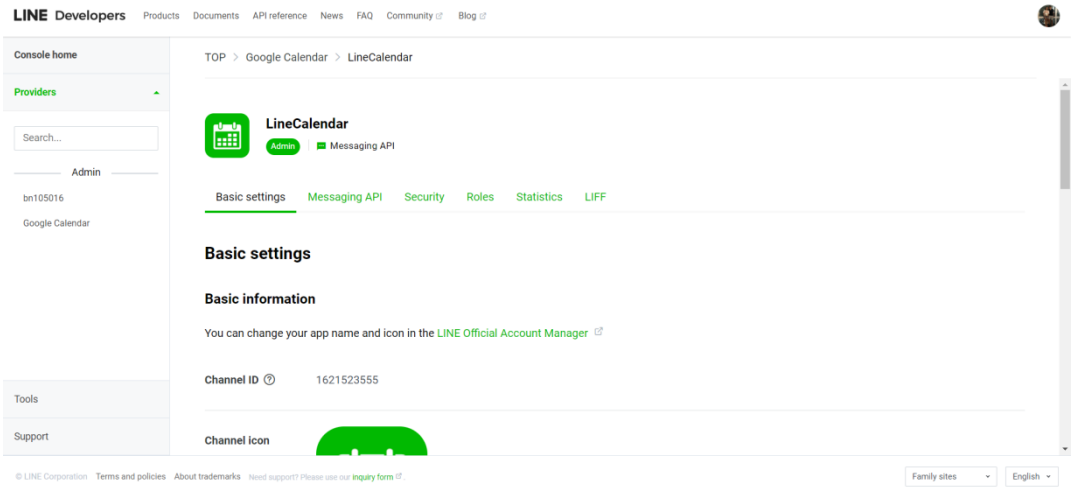


圖 18 建立完成後的畫面

二、使用 Google App Script 連接 Line

進入 Google Apps Script 點選新增 project，圖 19 案例說明

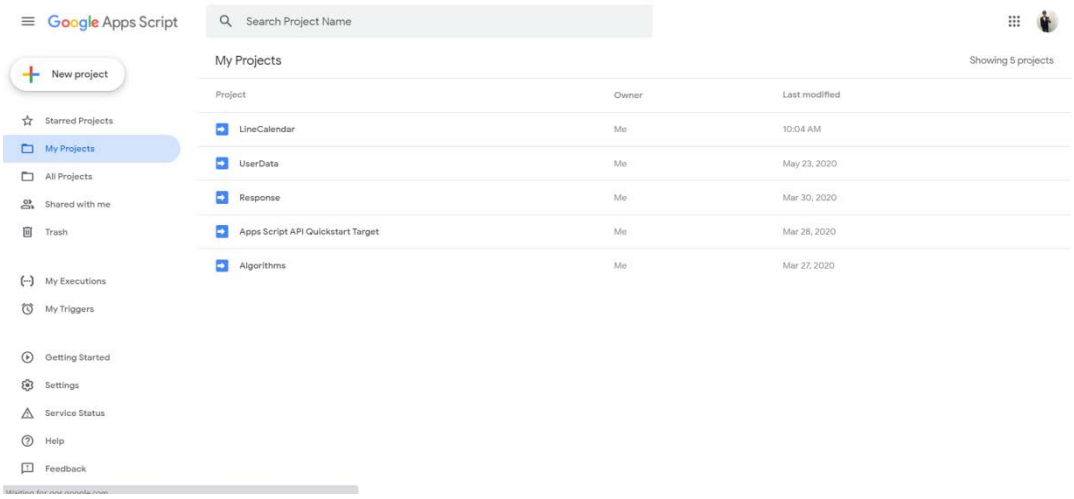


圖 19 進入 Google Apps Script

圖 20 建立完成後的畫面

在全部撰寫完後點 Publish 裡面的 Deploy as web app，圖 21 案說明。

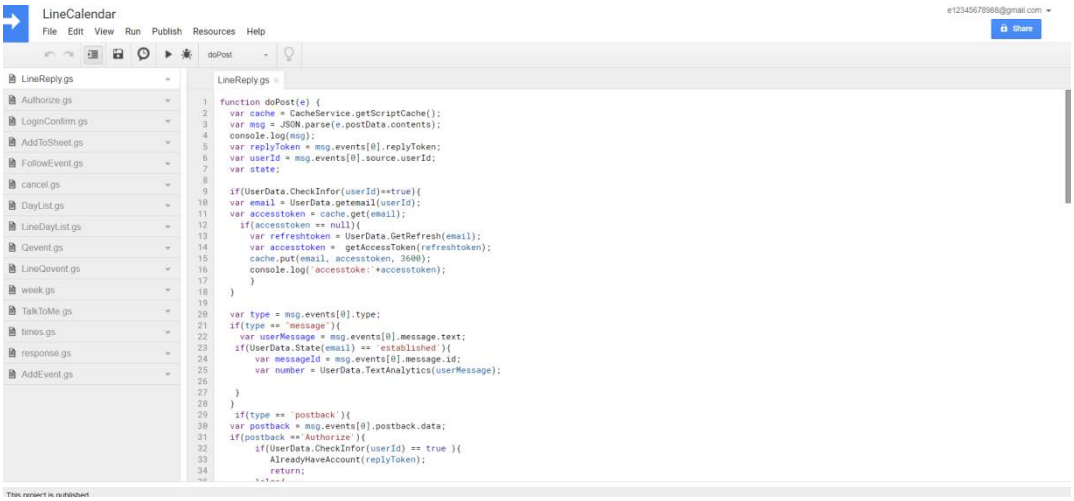


圖 21 完成程式撰寫

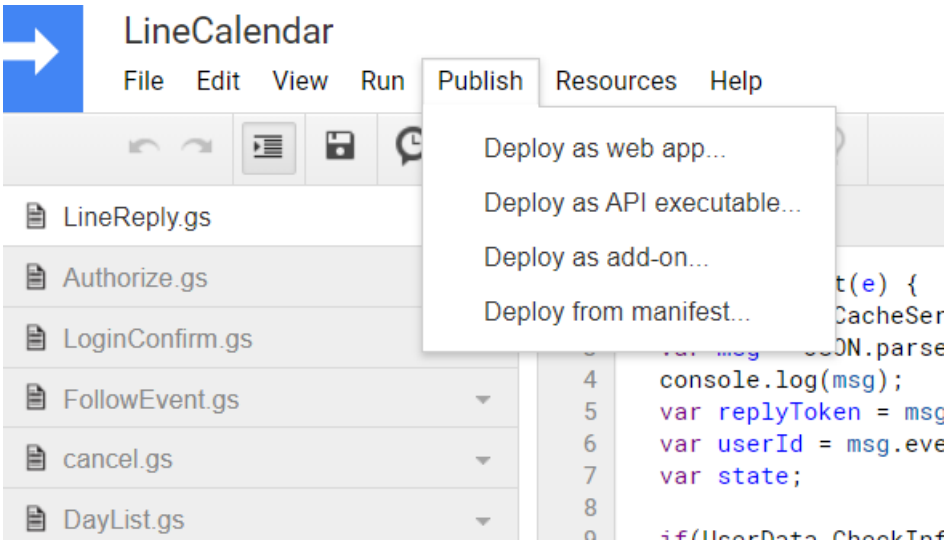


圖 22 發布畫面

建立新的版本，並且要任何人皆可訪問。

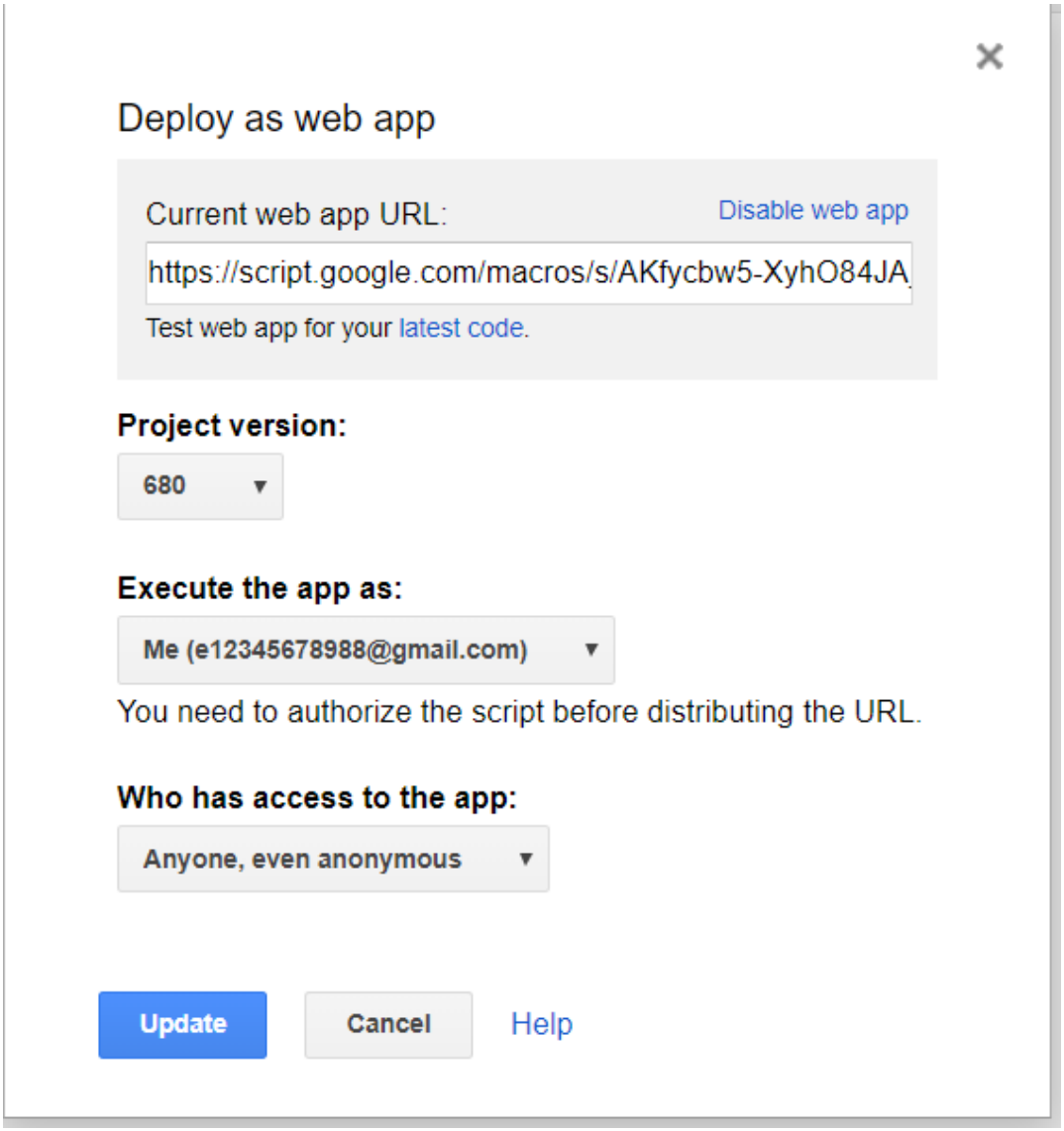


圖 23 發布成 web app

發佈完後回到 Line 的官方帳號管理往下滑將 Google APPScripts 發佈產生的 web app URL 放置到這裡並認證使用，如圖 24。

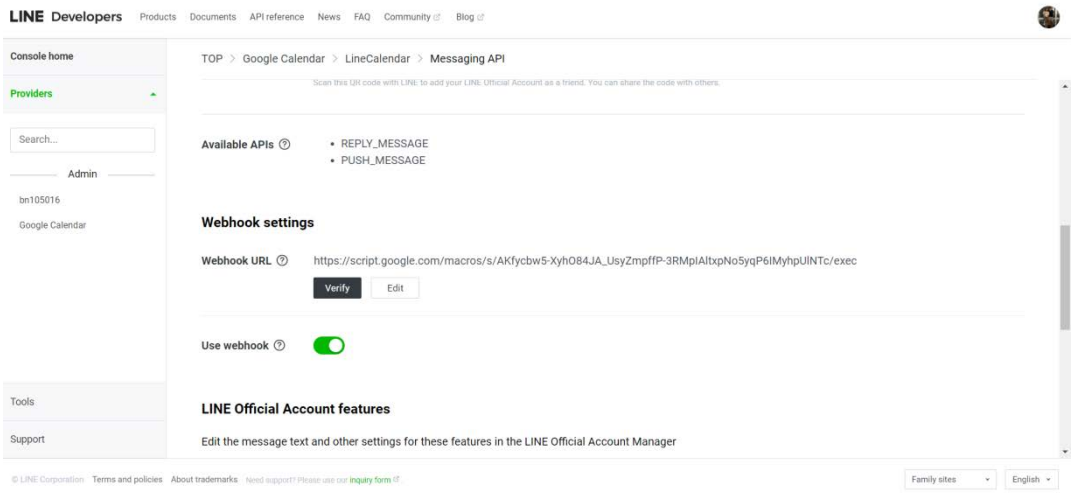


圖 24 認帳 webhook URL

這樣就完成 Google app script 與 Line 的連結。

三、在 App script 接受 Line 的訊息

LineReply.gs

```
1  function doPost(e) {
2    var cache = CacheService.getScriptCache();
3    var msg = JSON.parse(e.postData.contents);
4    console.log(msg);
5    var replyToken = msg.events[0].replyToken;
6    var userId = msg.events[0].source.userId;
```

圖 25 接收 Line 的訊息的程式碼

因為我們的程式會向 LINE 發送 HTTP POST request，所以這裡要將名稱改為 doPost，後方接著一個 e 的參數。因為 e.parameters 需包含指定的 key 與 value，所以這裡使用 e.postData.contents 來獲取完成傳送的內容，但因為獲取的訊息內容為「字串」，所以再用 JSON.parse 來轉換資料格式。

CacheService 是用於短暫的儲存數據最後再使用 console.log 來印出收到的訊息剛剛有寫了 console.log 來顯示訊息，所以需要前往 Google Cloud Platform 的 Stackdriver 查看訊息記錄，如圖 26。

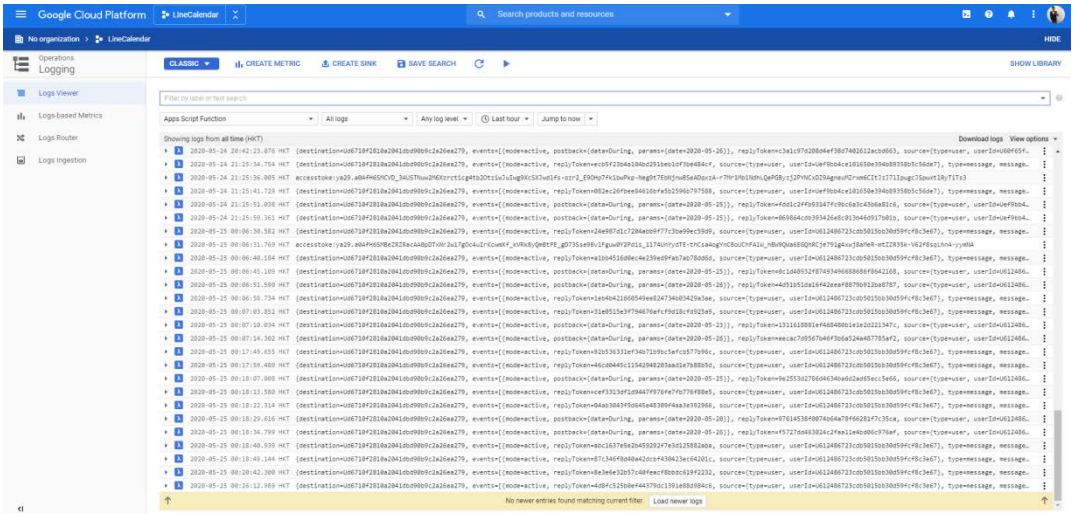


圖 26 查看接收到的訊息

msg.events[0].replyToken 便是這些訊息中的內容位置，msg 裡面的 event[0]裡面的 **replyToken**，**replyToken** 是用來回傳給當初傳送訊息的使用者，每次使用者傳送訊息後都會送出不同的 **replyToken**。**varuserId = msg.events[0].source.userId** 則是在 source 裡面的 userId，這是每個 Line 的帳號獨特的 Id，我們會需要儲存下來跟 email 一起保存，如圖 27。

圖 27 某段訊息詳細內容

四、建立登入 Google

使用 Google 的 OAuth2.0 身分驗證系統進行登入前，我們必須去 Google API 去設定，來獲取 OAuth2.0 的憑證，如圖 28。

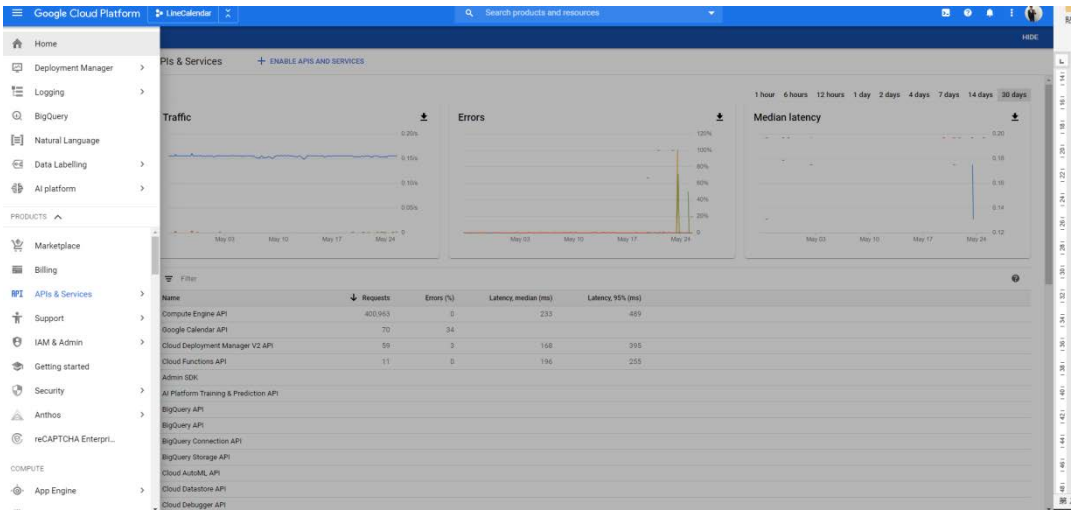


圖 28 Google Cloud Platform 畫面

進入到 Google Cloud Platform 點選左邊的 APIs & Service 裡面的 Credentials 之後點選 Create Credentials 選取 OAuth client ID，如圖 29。

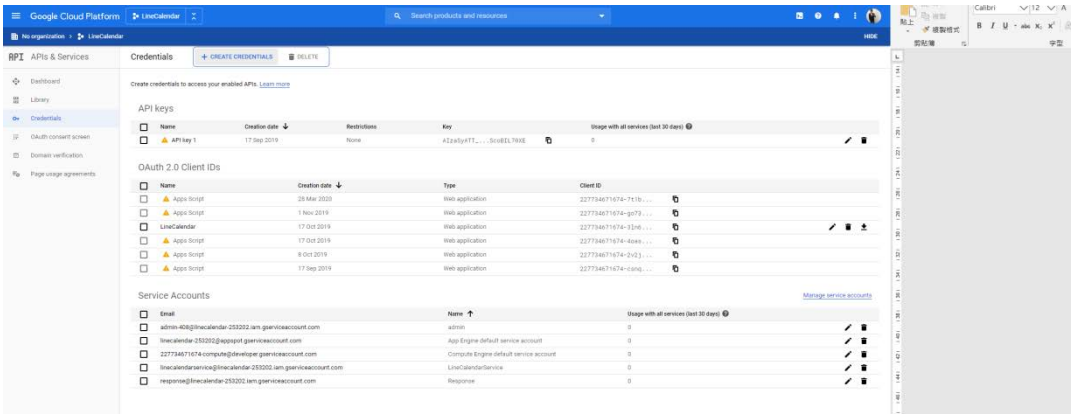


圖 29 API &Serive 畫面

在這邊 Application type 選取 Web application，如圖 30。

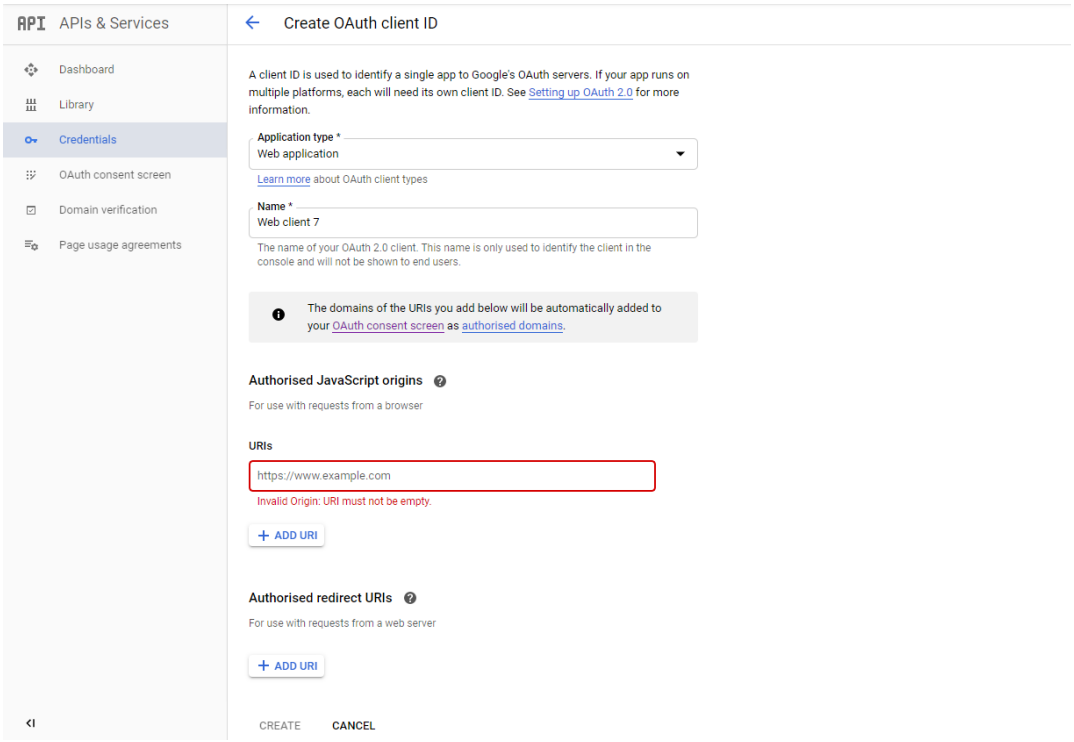


圖 30 建立 OAuth client ID

接下來 AuthorisedJavaScriptorigins 把 https://script.google.com 貼上。Authorised redirect URIs 則是把 web app url + /usercallback 和另一個網址如下 https://script.google.com/macros/d/ + Script ID 貼上。Script ID 可以在 File 裡面的 Project properties 找到，圖 31

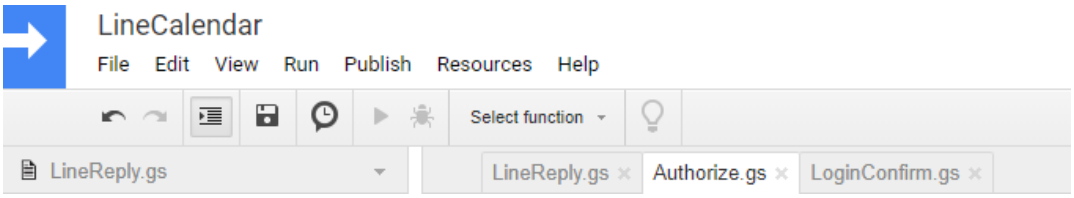


圖 31 App script 選單

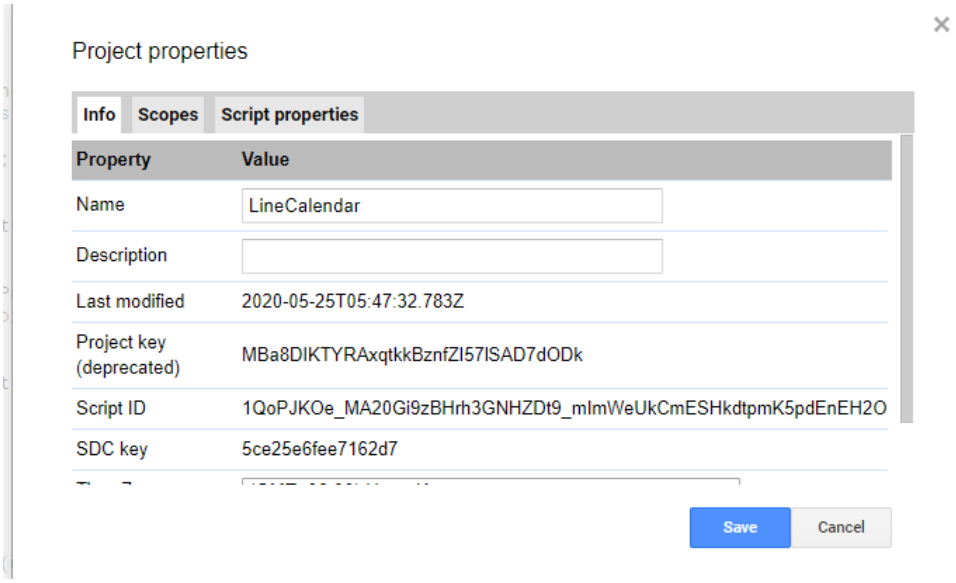


圖 32 File 裡的內容

新增完後會如圖 33。

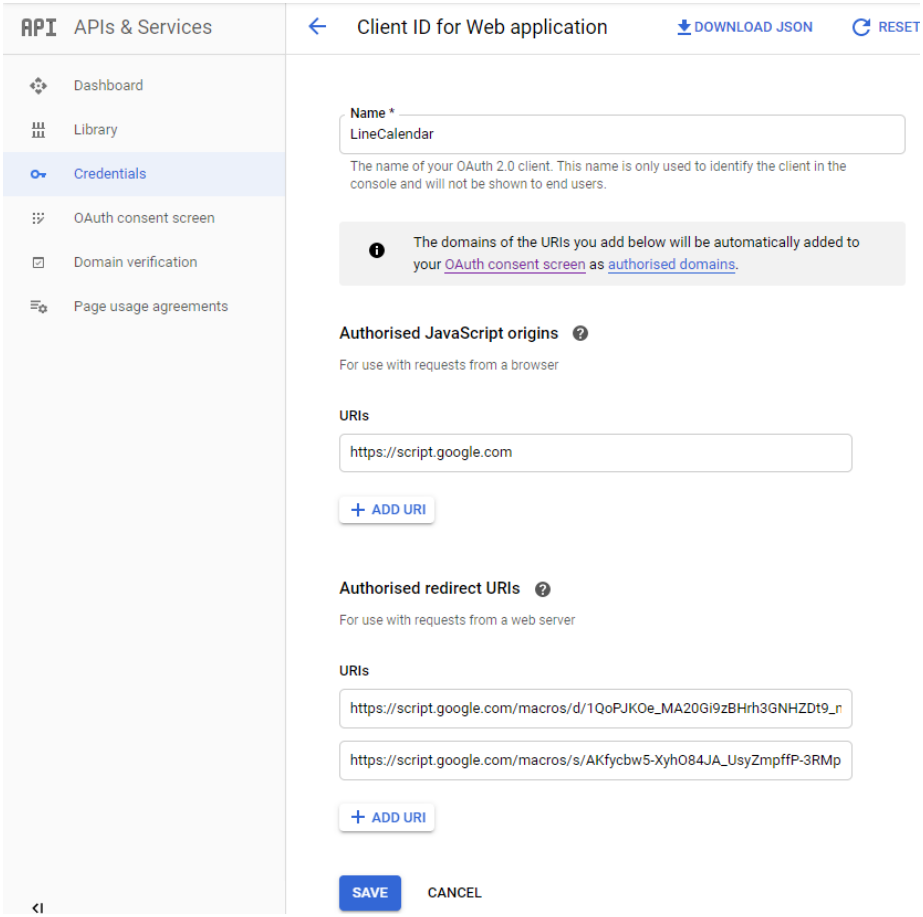


圖 33 Authorised redirect URL 設定

接下來點選旁邊的 OAuth consent screen，OAuth consent screen 是負責設定當你登入後下一步會出現是否同意我們使用某些訪問。

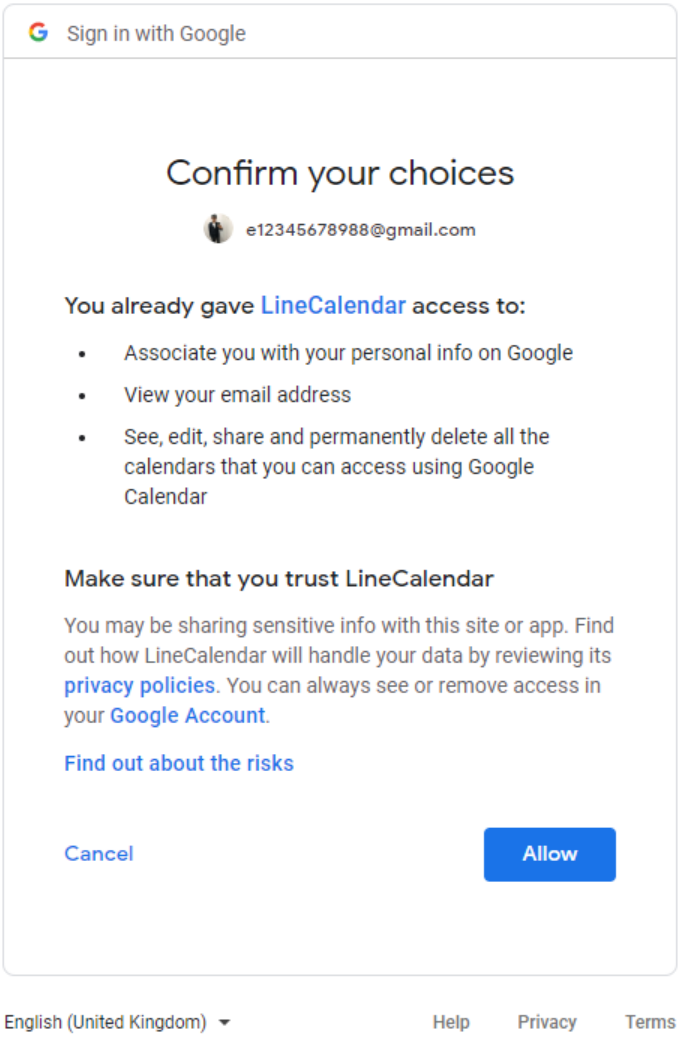

Sign in with Google
Confirm your choices
e12345678988@gmail.com
You already gave LineCalendar access to:
Associate you with your personal info on Google
View your email address
See, edit, share and permanently delete all the calendars that you can access using Google Calendar
Make sure that you trust LineCalendar
You may be sharing sensitive info with this site or app. Find out how LineCalendar will handle your data by reviewing its privacy policies. You can always see or remove access in your Google Account.
Find out about the risks
Cancel
Allow
English (United Kingdom)
Help
Privacy
Terms

圖 34 登入是否同意

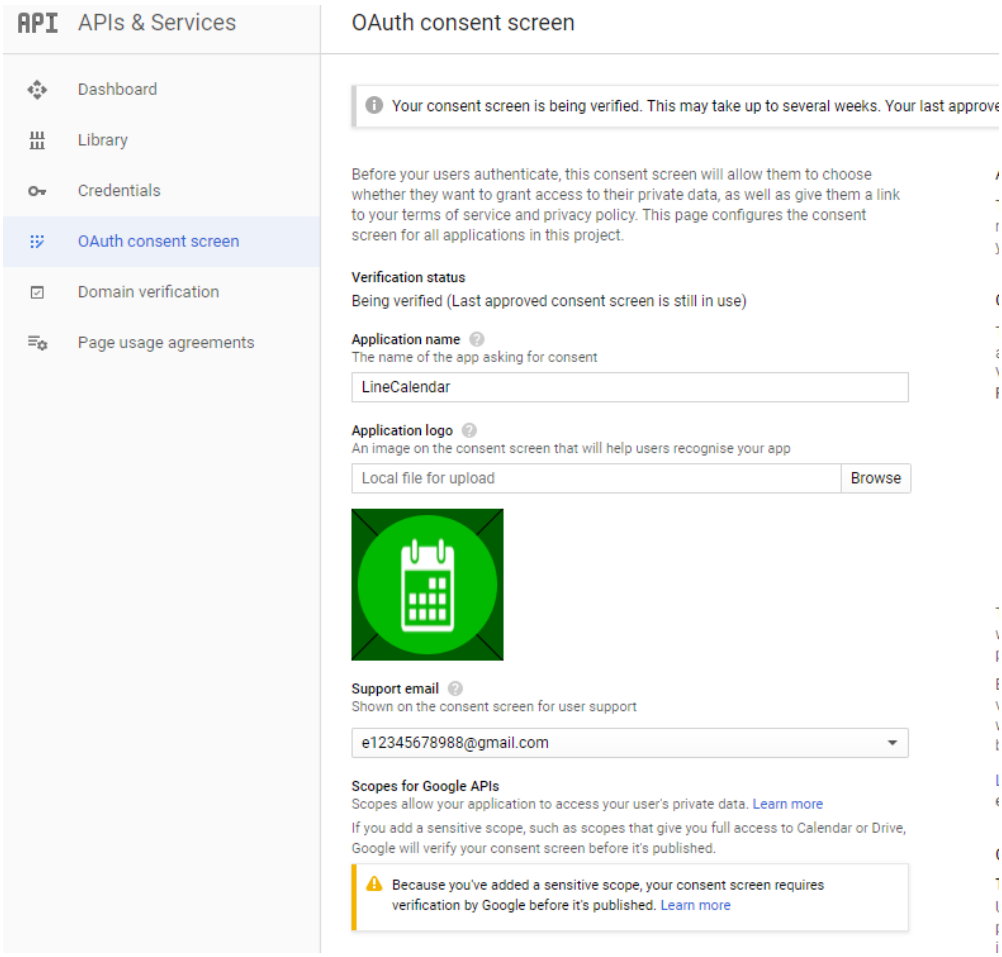


圖 35 OAuth 內容畫面設定

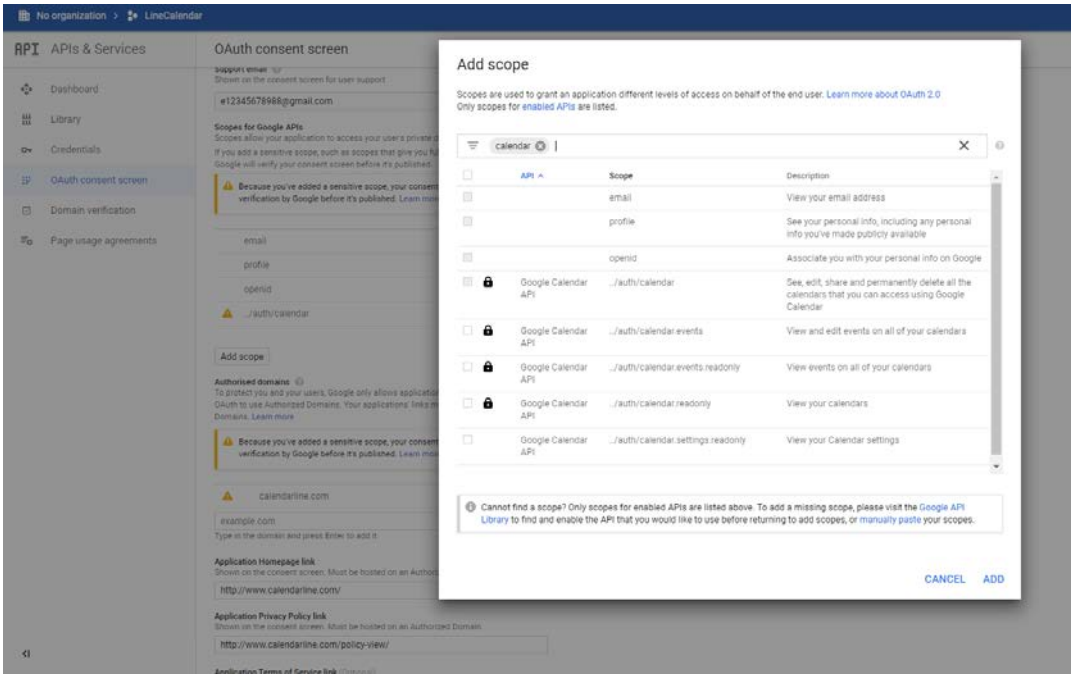


圖 36 範圍的設定畫面

接下來 Authorised dom 

ge link、Application Privacy Po与和隱私政策，這是為了保護用戶。

圖 37 設定畫面

我們需要建一個網址來讓使用者登入。

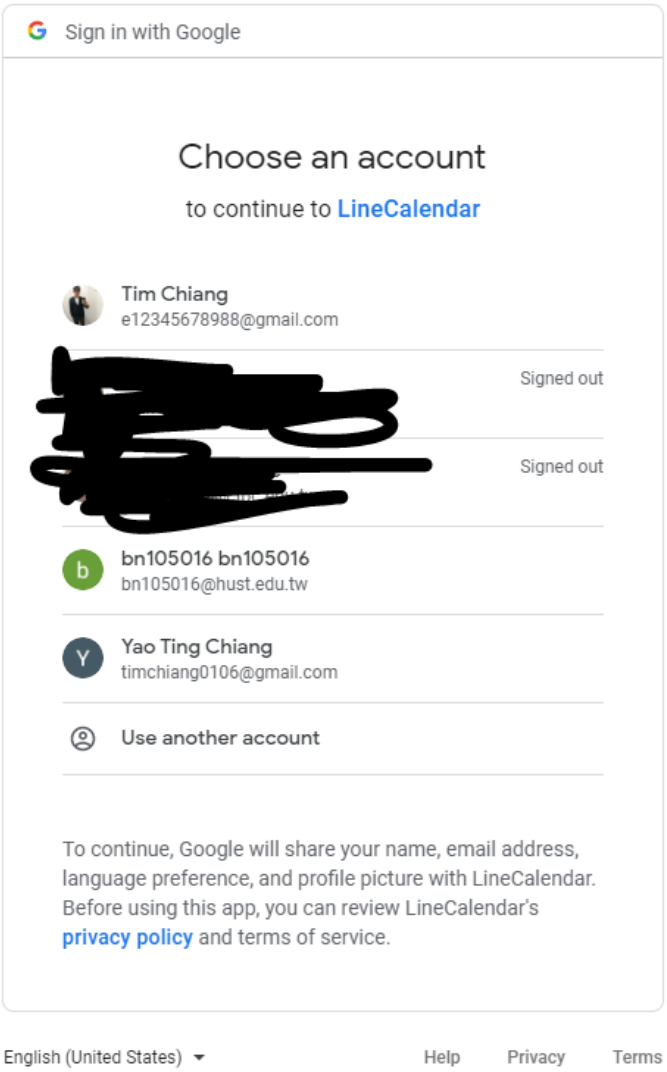


圖 38 登入畫面

這是我們讓使用者登入的網址

https://accounts.google.com/o/oauth2/v2/auth?response_type=code&scope=https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fcalendar%20https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fuserinfo.email&client_id=227734671674-3ln6msgcicpqabgm1smgpu11fasprntv.apps.googleusercontent.com&redirect_uri=https%3A%2F%2Fscript.google.com%2Fmacros%2Fs%2FAKfycbw5-XyhO84JA_UsyZmpffP-3RMpIA1txpNo5yqP6IMyhpU1NTc%2Fusercallback&prompt=consent&access_type=offline

可以把網址解析為範圍

scope=https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fcalendar%20https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fuserinfo.email

一個是 https://www.googleapis.com/auth/calendar 另一個是 http://www.googleapis.com/auth/userinfo.email

後者是因為我們要存取他的 Email 所以另外要求 response_type 正常的話是 codeclient_id 是當初創立 credentials 所得到的

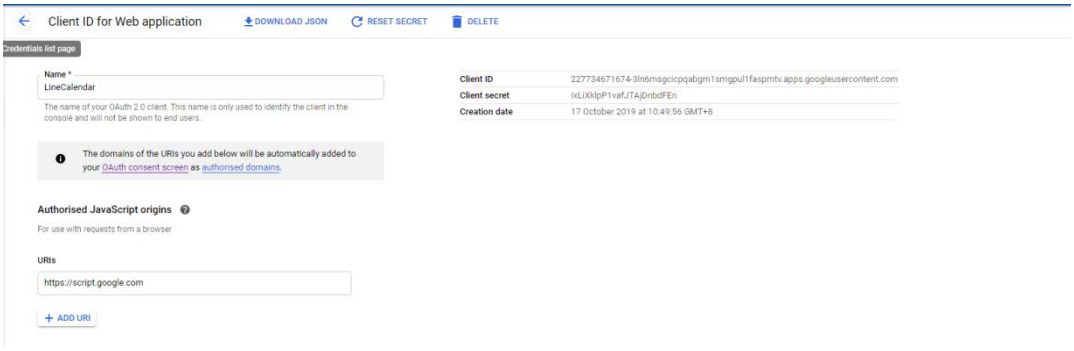


圖 39 Client ID 畫面

redirect_uri 是你的 web app URL 當設置完網址後可以儲存在

Project properties 裡面的 Script properites

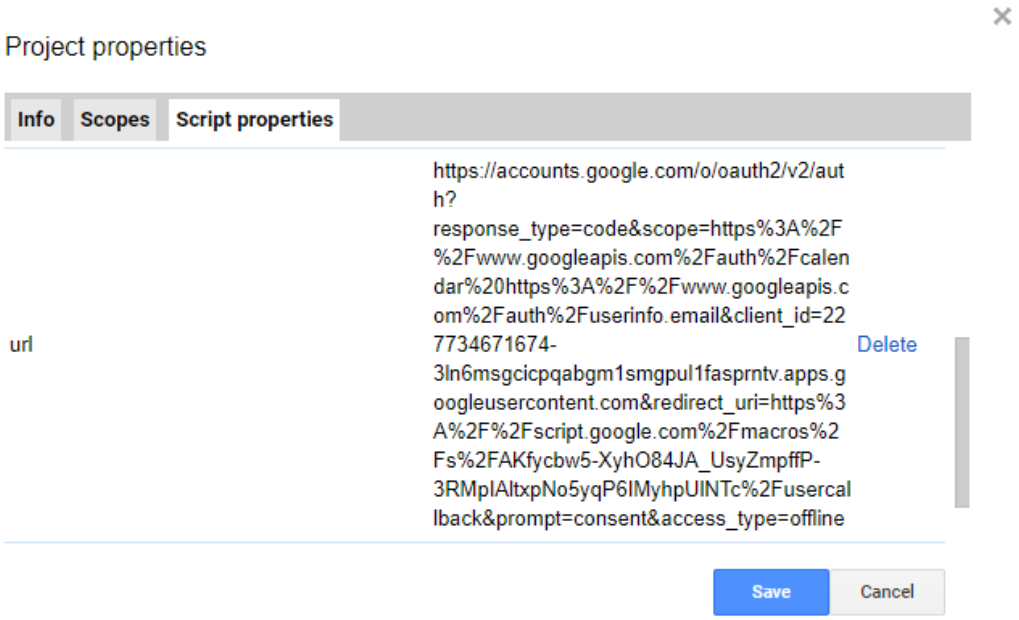


圖 40 Project properties 畫面

```
function doGet(e) {
  var props =  PropertiesService.getScriptProperties().getProperties();
  var url = 'https://oauth2.googleapis.com/token';
  var postPayload = {
      "code" : e.parameter.code ,
      "client_id" : props.client_id,
      "client_secret" : props.client_secret,
      "redirect_uri" : getGoogleCallbackURL(true),
      "grant_type" : "authorization_code",
    };
  var options = {
      "method" : "post",
      "payload" : postPayload,
    };
       var credentials = UrlFetchApp.fetch(url, options);
       var params = JSON.parse(credentials.getContentText());
       console.log(params);
       var refresh = params.refresh_token;
       var access = params.access_token;
       addtosheet(refresh,access);
       return HtmlService.createHtmlOutput('成功登入，接下來你可以關閉視窗，開始使用Google日曆的功能了');
}
```

圖 41 認證帳號部分程式碼

當使用者登入時，同意後會觸發 doGet(e)來執行程式碼，後方一樣接著一個 e 的參數，從 code 是一個只能使用一次的認證碼，可以得到 access token 和 ID tokenClient_id 和 Client_secret 都是從 credential 複製貼上過來 Redirect_url 則是自己當初設置 credential 裡的 Authorisedredirect URIs 的 web apps + /usercallback 因為最後會出現最後成功登入的網址

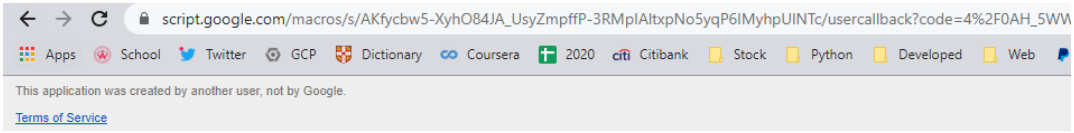


圖 42 登入完成後畫面

然後"method" 是"post",最後用 UrlFetchApp.fetch(url, options)來請求資訊

```
var credentials = UrlFetchApp.fetch(url, options);
var params = JSON.parse(credentials.getContentText());
console.log(params);
var refresh = params.refresh_token;
var access = params.access_token;
addtosheet(refresh,access);
return HtmlService.createHtmlOutput('成功登入，接下來你可以關閉視窗，開始使用Google日曆的功能了');
```

圖 43 認證帳號部分程式碼

一樣把得到的訊息改成 JSON 並且顯示在 Logging 裡面，並且把 refresh_token 和 access_token 儲存起來。

Refresh_token 是一個永久鑰匙，因為實際要求使用者行事曆訊息的是 access_token 但是最多只有 3600 秒的期限，在 access_token 過期後，我們需要用 refresh_token 再請求一次。

最後都將透過 addtosheet function 來儲存到雲端裡的 sheet 裡

HtmlService.createHtmlOutput('成功登入，接下來你可以關閉視窗，開始使用 Google 日曆的功能了');用來顯示成功登入的畫面。

```
function addtosheet(refresh,access) {
  var props = PropertiesService.getScriptProperties();
  var cache = CacheService.getScriptCache();
  var userid = props.getProperty('getPorfile');
  var email = getEmail(access);
  cache.put(email, access, 3600);
  UserData.AddUserData(userid,email, refresh, 'established');
}
```

圖 44 新增使用者資訊程式碼

PropertiesService.getScriptProperties();是準備來讀取儲存在 Scriptporperties 裡的數值 CacheService.getScriptCache();是準備短暫儲存 accesstoken。

props.getProperty('getPorfile'); 讀取儲存在 Script properites 裡的 getPorfile 的 UserId。

getEmail(access);用 getEmail function 來得到使用者的 Email

```
function getEmail(access){
  var url = 'https://www.googleapis.com/oauth2/v2/userinfo';
  var option = {
   'headers': {
      'Content-Type': 'application/json; charset=UTF-8',
      'Authorization': 'Bearer '+ access,
    },
    'method':'get'
  };
   var data = UrlFetchApp.fetch(url, option);
   var params = JSON.parse(data.getContentText());
   console.log(params);
   return params.email;
}
```

圖 45 得到使用者的 Email 程式碼

使用請求的網址和 headers，裡面須包含 access_token，然後 method 是 get，得到訊息後一樣轉成 JSON 然後回傳。

cache.put(email, access, 3600);

UserData.AddUserData(userid,email,refresh, 'established');

UserData 是另外一個 Project，用來儲存與讀取

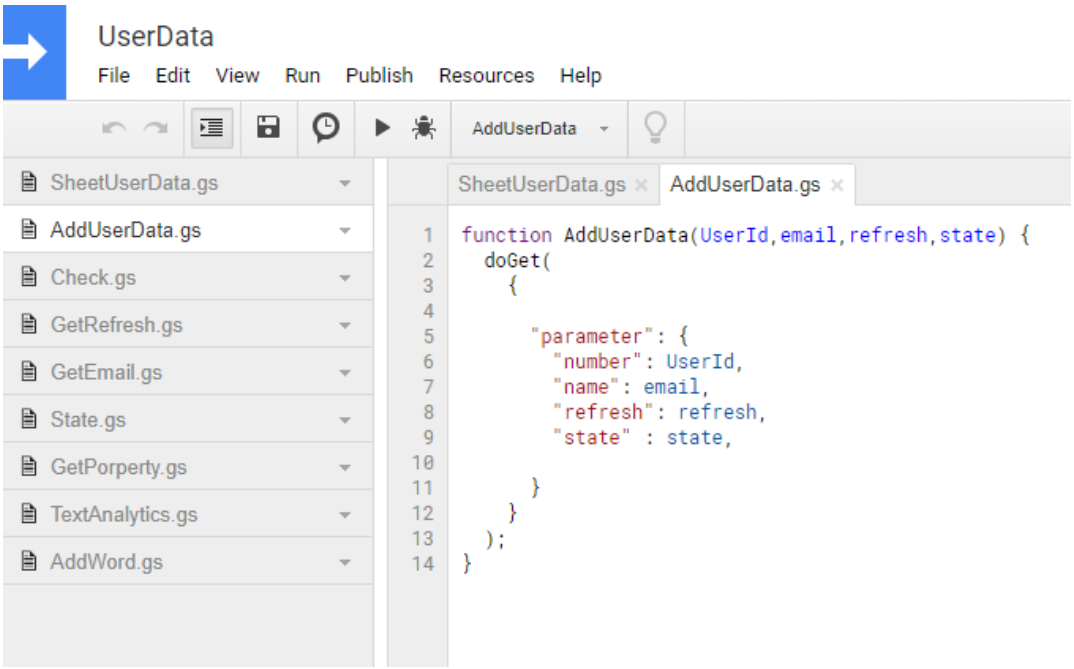


圖 46 新增到 Sheet 程式碼

要如何連接兩個 Project 點選你要主要的 project 上方選項的 resources 的 Libraries

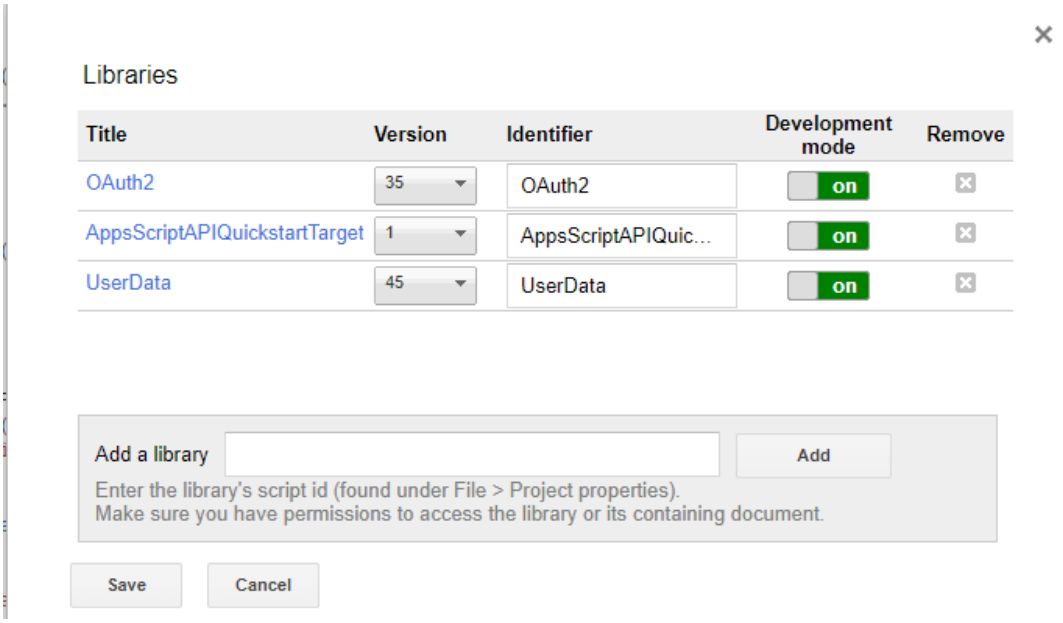


圖 47 連接兩個 project

在 Add a library 裡面放入第二個 project 的 Script ID 貼上後按下 Add，變會出現資訊，選擇你要的版本，並且將 Off 點下去便會啟動(On)

當使用者傳送訊息後，查看雲端的 Sheet 是否有他的 UserId

```
  if(UserData.CheckInfor(userId)==true){
  var email = UserData.getemail(userId);
  var accesstoken = cache.get(email);
    if(accesstoken == null){
      var refreshtoken = UserData.GetRefresh(email);
      var accesstoken =  getAccessToken(refreshtoken);
      cache.put(email, accesstoken, 3600);
      console.log('accesstoke:'+accesstoken);
      }
  }else{
    var props = PropertiesService.getScriptProperties()
    props.setProperty('getPorfile',userId)
    loginconfirm(replyToken);
    return;
  }
```

圖 48 查看雲端的程式碼

if(UserData.CheckInfor(userId)==true)是用來確認

```
function CheckInfor(userId) {
  var SpreadSheet = SpreadsheetApp.openById("1tJJ2ynTJC1sYLxG7Wx1iHvk0NRJ9LAoC_O7zQHdPXQI");
  var Sheet = SpreadSheet.getSheetByName("UserData");
  var values = Sheet.getDataRange().getValues();
  for(var i=1;i<values.length;i++){
    if(userId == values[i][0]){
      return true ;
    }
  }
}
```

圖 49 確認使用者的程式碼

UserData.getemail(userId);用 UserId 來得到儲存在雲端 Sheet 裡的 Email

```
function getemail(userId) {
  var SpreadSheet = SpreadsheetApp.openById("1tJJ2ynTJC1sYLxG7Wx1iHvk0NRJ9LAoC_O7zQHdPXQI");
  var Sheet = SpreadSheet.getSheetByName("UserData");
  var values = Sheet.getDataRange().getValues();
  for(var i=1;i<values.length;i++){
    if(userId == values[i][0]){
      return values[i][1];
    }
  }
}
```

圖 50 讀取使用者 Email 程式碼

varaccesstoken = cache.get(email)用 Email 來讀取使用者的 Accesstoken

if(accesstoken == null){ 如果 Accesstoken 回傳後是 null 無效的

varrefreshtoken = UserData.GetRefresh(email);

varaccesstoken= getAccessToken(refreshtoken);

先讀取使用者的 refreshtoken 來取得新的 accesstoken，我們使用 getAccessToken function 來得到新的 accesstoken

```
function getAccessToken(refresh_token) {
  var props = PropertiesService.getScriptProperties().getProperties();
  var url = 'https://accounts.google.com/o/oauth2/token';
  var postPayload = {
      'client_id': props.client_id,
      'client_secret': props.client_secret,
      'refresh_token': refresh_token,
      'grant_type': "refresh_token",
    };
  var options = {
      "method" : "post",
      "payload" : postPayload,
    };
      var credentials = UrlFetchApp.fetch(url, options);
      var params = JSON.parse(credentials.getContentText());

      return params.access_token;
}
```

圖 51 Get AccessToken 的程式碼

cache.put(email, accesstoken, 3600);

取得後重新放入到 cache 裡，第一個是 key 第二個是 value 最後

是過期時間。

如果使用者在雲端裡的 Sheet 沒有 UserId，則會先儲存他的 UserId 到 ScriptProperties，之後會透過 loginconfirm(replyToken);來傳送互動訊息給使用者

```
else{
var props = PropertiesService.getScriptProperties()
props.setProperty('getPorfile',userId)
loginconfirm(replyToken);
return;
  }
```

五、新增官方帳號的互動

圖 52 新增好友後畫面

```
function loginconfirm(replyToken) {
  var proper = PropertiesService.getScriptProperties();
  var CHANNEL_ACCESS_TOKEN = proper.getProperty('CHANNEL_ACCESS_TOKEN');
   var url = 'https://api.line.me/v2/bot/message/reply';
   var option = {
     'headers':{
           'Content-Type': 'application/json',
           'Authorization': 'Bearer ' + CHANNEL_ACCESS_TOKEN,
     },
      'method': 'post',
     'payload': JSON.stringify({
       'replyToken': replyToken,
       'messages': [{
         "type": "template",
         "altText": "this is a confirm template",
        "template": {
         "type": "confirm",
         "text": "登入Google",
         "actions": [
           {
             "type": "uri",
             "label": "登入",
             "uri": proper.getProperty('url'),
           },
           {
             "type": "message",
             "label": "不登入",
             "text": "不登入無法使用功能，請點選登入來享受在這裡使用Google日曆的服務"
           },
       ],
  },
        }],
}),
};
  UrlFetchApp.fetch(url, option);

   }
```

圖 53 Line 互動框框的程式碼

當我們需要傳送訊息給使用者時會需要用到 Channel access token，這時候要到 Line developers 你創立的 Messaging API 最底下可以得到

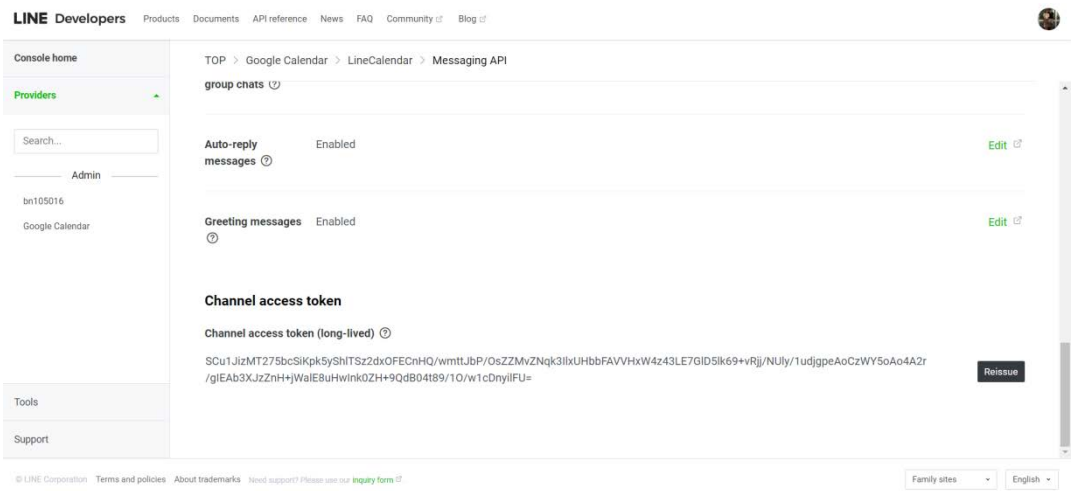


圖 54 Channel access token

得到後，可以事先儲存在 ScriptProperties，再透過

PropertiesService.getScriptProperties()來取得

https://api.line.me/v2/bot/message/reply 是用來請求回復的網址，

```
var option = {
  'headers':{
        'Content-Type': 'application/json',
        'Authorization': 'Bearer ' + CHANNEL_ACCESS_TOKEN,
  },
```

圖 55 headers 程式碼

需要在 headers 裡面的 Authorization 裡放入 Channel_Access_Token

```
    'method': 'post',
  'payload': JSON.stringify({
    'replyToken': replyToken,
    'messages': [{
      "type": "template",
      "altText": "this is a confirm template",
    "template": {
      "type": "confirm",
      "text": "登入Google",
      "actions": [
        {
          "type": "uri",
          "label": "登入",
          "uri": proper.getProperty('url'),
        },
        {
          "type": "message",
          "label": "不登入",
          "text": "不登入無法使用功能，請點選登入來享受在這裡使用Google日曆的服務"
        },
    ],
},
     }],
',

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

圖 56 程式碼

Line 回送訊息有特定的需求

HTTP request

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

Request headers

Content-Type	application/json
Authorization	Bearer `{channel access token}`

Request body

replyToken String Required	Reply token received via webhook
messages Array of message objects Required	Messages to send Max: 5

圖 57 Line 回送訊息有特定的需求

我們使用的種類是 template

Confirm template

Use the confirm template to send a message with two buttons.

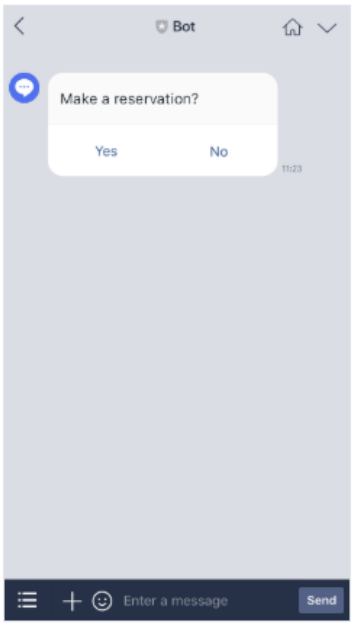


圖 58template

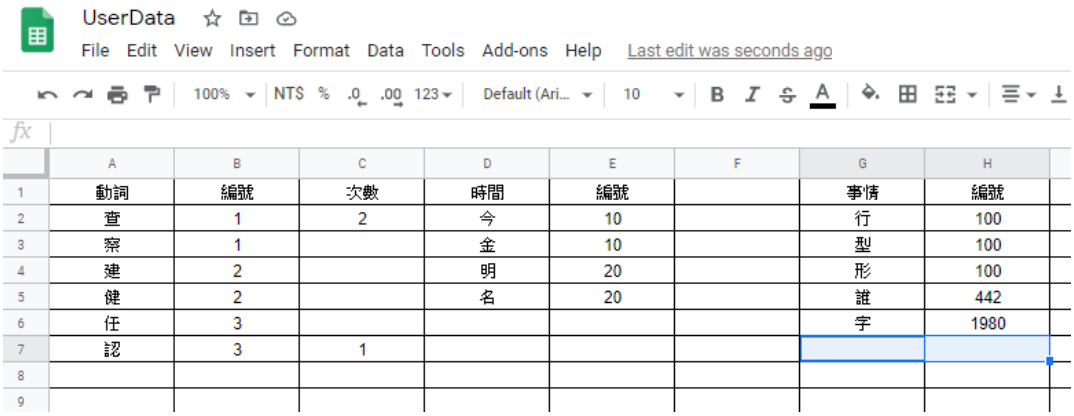
六、Line 的訊息種類

```
25  var type = msg.events[0].type;
26  if(type == "message"){
27    var userMessage = msg.events[0].message.text;
28   if(UserData.State(email) == 'established'){
29       var messageId = msg.events[0].message.id;
30       var number = UserData.TextAnalytics(userMessage);
31   }
32  }
33  if(type == 'postback'){
34  var postback = msg.events[0].postback.data;
35  if(postback =='Authorize'){
36       if(UserData.CheckInfor(userId) == true ){
37          AlreadyHaveAccount(replyToken);
38          return;
39       }else{
40         var props = PropertiesService.getScriptProperties()
41         props.setProperty('getPorfile',userId)
42         loginconfirm(replyToken);
43         return;
44       }
45    }
46    if(postback =='cancel'){cancel(replyToken);UserData.NewState(email, 'established');return;}
47
48  }
49
```

圖 59 Line 的訊息種類

Message 會是一般使用者傳送的訊息，另外第 28 行，是用來確認使用者的狀態是完成的而不是在進行其他動作，因為我們與使用者進行互動，所以是用來記錄使用者執行到哪個步驟。

30 行是用來對文字進行分析。



	A	B	C	D	E	F	G	H
1	動詞	編號	次數	時間	編號		事情	編號
2	查	1	2	今	10		行	100
3	察	1		金	10		型	100
4	建	2		明	20		形	100
5	健	2		名	20		誰	442
6	任	3					宇	1980
7	認	3	1					
8								
9								

圖 60 文字編號

如查詢、察巡，我們可以知道使用者是想要查詢，但是因中文有同音不同字的問題，所以我們給予第一個字相同編號。

建立、健力，也是一樣的問題。

postback 則是透過互動的方式傳送

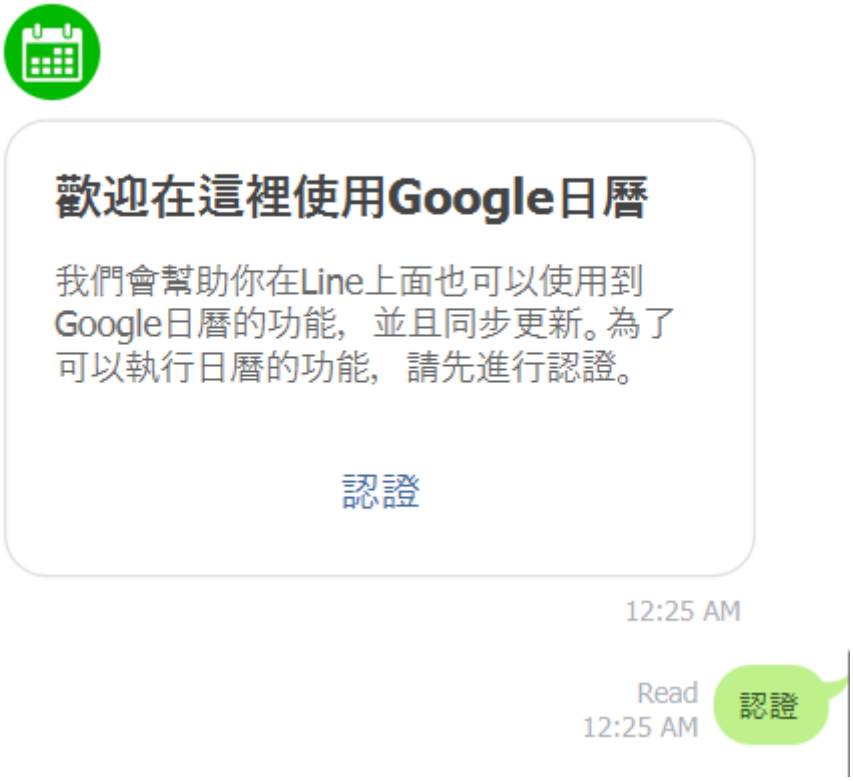


圖 61 Line 對話

當使用者點選認證後，會直接透過 postback 送出認證

如果 postback 是認證的話，確認是否有帳號。

如果是取消(Cancel)的話會傳送取消的回覆，並且將使用者的狀態改成完成。

```
function cancel(replyToken) {
  var proper = PropertiesService.getScriptProperties();
  var CHANNEL_ACCESS_TOKEN = proper.getProperty('CHANNEL_ACCESS_TOKEN');
   var url = 'https://api.line.me/v2/bot/message/reply';
  UrlFetchApp.fetch(url, {
      'headers': {
      'Content-Type': 'application/json; charset=UTF-8',
      'Authorization': 'Bearer ' + CHANNEL_ACCESS_TOKEN,
    },
    'method': 'post',
    'payload': JSON.stringify({
      'replyToken': replyToken,
      'messages': [{
        'type': 'text',
        'text': '取消完成',
      }],
    }),
  });
}
```

圖 62 取消的程式碼

這是最基本的回傳一般的文字

圖 63 LINE 對話

最後如果種類是 follow 則是當使用者加入好友時，會收到的種類

七、解析句子

```
87    switch(number){
88      case 1:
89      case 101:
90       LineQevent(replyToken);
91       UserData.NewState(email,  'quevent');
92       break;
```

圖 64 回復的程式碼

首先 number 是 `var number = UserData.TextAnalytics(userMessage);`

```
function TextAnalytics(userMessage) {
  var userMessage = '我是誰'
  var split = userMessage.split("");
  var SpreadSheet = SpreadsheetApp.openById("1tJJ2ynTJC1sYLxG7Wx1iHvk0NRJ9LAoC_O7zQHdPXQI");
  var Sheet = SpreadSheet.getSheetByName("TextAnalytics");
  var values = Sheet.getDataRange().getValues();
  var number=0;
  for(var x=0;x<split.length;x++){
    for(var i=0;i<7;i+=3){
     for(var j = 1;j<values.length;j++){
        if(split[x] == values[j][i]){
            number += values[j][i+1];

        }
     }
   }
  }
  Logger.log(number)
  return number;
}

```

圖 65 解析句子得程式碼

第 1 行是負責把收到的文字拆開來
第 2 行 SpreadsheetApp.openById(id)，id 則是可以在 sheet 的網址上找到

圖 66 id 網址

Values 就是各個格子的數值，[0][0]就是動詞[0][1]是編號 [1][0]是查

動詞	編號		時間	編號		事情	編號
查	1		今	10		行	100
察	1		金	10		型	100
建	2		明	20		形	100
健	2		名	20		誰	442
任	3					字	1980
認	3						

圖 67 文字編號

然後利用迴圈找出並計算數值，

如果是查詢事件次數，給予編號 101 假如使用者只說查詢 則是 1，

我們都給他做相同的查詢事件的動作，

並且把使用者的狀態改成搜尋中。

圖 68 LINE 對話

	A	B	C	D	E
1	userid	email	refresh	state	title
2	Uc9f9a91bd46d59f14ec62afee89f0771	katherine710@gmail.com	1//04hk8Tl1tolL6CgYIARAAGAQSNwF-L9Irm9hn9fo58qGoP3UhjKENS31AdBHxUJAA1M_ug6LiXyYcdVr3kGGMHmprROw9aqEXU_o	established	史上
3	U60f65ff88267b1f212a7a5dd1c02ce7e	bn105015@gmail.com	1//04yxawJzuXw-rCgYIARAAGAQSNwF-L9IrfR7_rgJnYOTPiUb7QWVmqsegmvWyerN86C2-SwmlTw8HZUttD53qE_JnbUUhzPM1e1U	established	測試
4	Uef9bb4ce101650e394b89358b5c56de7	sasasa60161@gmail.com	1//04Yw629okgn8nCgYIARAAGAQSNwF-L9Irar	established	上課
5	U612486723cdb5015b	e12345678988@gmail.com	1//04qJ09yKMYDvjCgYIARAAGAQSNwF-L9IrSddNUyaz0eM-2lWap6HUiVrklw3BxBaSVPkRyKqBfYkM1G5EYiiCSt93dRK1iuH7exg	quevent	星期一

另外，互動的訊息可以用 Line 官方提供的網站來修改。

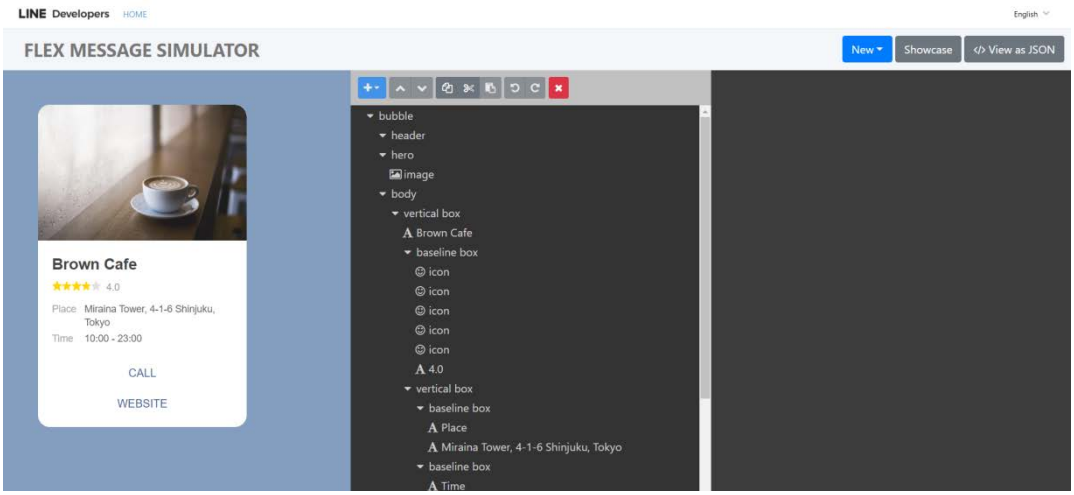


圖 70 修改圖

在複製到程式碼內

```
50    if(UserData.State(email)== 'quevent'){
51        UserData.setTitle(email ,userMessage);
52        stime(replyToken)
53        UserData.NewState(email, 'stime');
54        return;
55      }
56      if(UserData.State(email)== 'stime'){
57        UserData.stime(email, msg.events[0].postback.params.date);
58        etime(replyToken)
59        UserData.NewState(email, 'etime');
60        return;
61      }
62      if(UserData.State(email)== 'etime'){
63        UserData.etime(email,  msg.events[0].postback.params.date);
64        var all = UserData.getPorperty(email)
65
66        QeventList(replyToken,all,email)
67        UserData.NewState(email, 'established')
68        return;
69      }
```

圖 71 流程的程式碼

當使用者送出想查詢事件的主旨後，因為他得狀態是查詢中所以會送出。

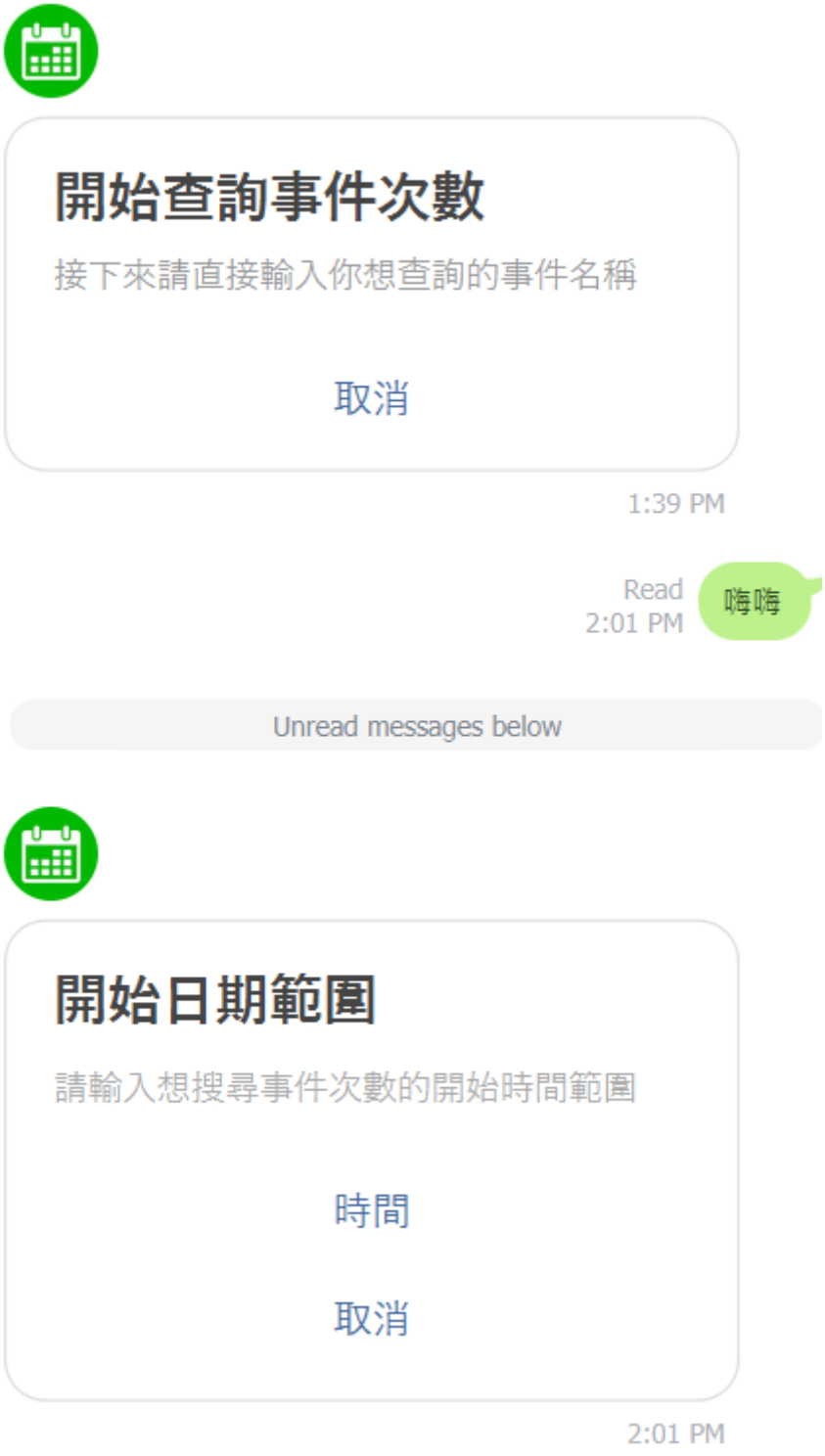


圖 72 LINE 對話

開始日期的對話，接下來是結束，會記錄全部的主旨和時間在 sheet 裡

```
252 function QeventList(replyToken,all,email) {
253   var proper = PropertiesService.getScriptProperties();
254   var itemlength = times(all,email);
255   if (itemlength == 0){noReasult(replyToken);return;};
256   var CHANNEL_ACCESS_TOKEN = proper.getProperty('CHANNEL_ACCESS_TOKEN');
257   var url = 'https://api.line.me/v2/bot/message/reply';
258   UrlFetchApp.fetch(url, {
259       'headers': {
260       'Content-Type': 'application/json; charset=UTF-8',
261       'Authorization': 'Bearer ' + CHANNEL_ACCESS_TOKEN,
262     },
263     'method': 'post',
264     'payload': JSON.stringify({
265       'replyToken': replyToken,
266       'messages': [{
267         'type': 'text',
268         'text': '在'+all[1]+'到'+all[2]+'總共有'+itemlength+'次'+all[0],
269       }],
270     }),
271   });
272 }
```

圖 73 Line 回復的程式碼

```
1  function times(all, email) {
2    var cache = CacheService.getScriptCache();
3
4    var accessToken = cache.get(email);
5
6    var q = all[0]
7    var stime = all[1]
8    var etime = all[2]
9
10   var url = 'https://www.googleapis.com/calendar/v3/calendars/'+email+'/events?access_token='+acce
11
12   var reasult  =  UrlFetchApp.fetch(url);
13   var item = JSON.parse(reasult.getContentText());
14   var itemlength = Object.keys(item.items).length;
15   return itemlength;
16 }
17
18
```

圖 74 搜尋事件次數的程式碼

第 254 行的 times 則是下圖的程式碼，用 accesstoken 和得到的主旨和開始結束時間來進行查詢，varurl = 'https://www.googleapis.com/calendar/v3/calendars/'+email+'/events?access_token='+accessToken+'&q='+q+'&timeMax='+etime+'T23%3A59%3A59%2B08%3A00&timeMin='+stime+'T00%3A00%3A00%2B08%3A00&alt=json&prettyPrint=true';

時間上要遵守 RFC3339 timestamp，所以再得到的時間後面加上 T23%3A59%3A59%2B08%3A00 和 T00%3A00%3A00%2B08%3A00。

得到的事件(itmes)多寡再回傳給 queventlist 並送出給使用者，假如沒有收到任何事件，則會回傳查無資料。

圖 75 Line 對話程式碼

```
93      case 2:
94      case 102:
95        addevent(replyToken)
96        UserData.NewState(email,  'add');
97        break;
98      case 3:
99        if(UserData.CheckInfor(userId) == true ){
100         AlreadyHaveAccount(replyToken);
101         return;
102       }else{
103         var props = PropertiesService.getScriptProperties()
104         props.setProperty('getPorfile',userId)
105         loginconfirm(replyToken);
106         return;
107       }
108       break;
```

圖 76 新增事件等程式碼

接下來是新增事件，和當使用者說認證時，我們會回傳是否已經登入了。

當使用者新增事件後，狀態改成新增中所以會執行下列程式碼。

```
70    if(UserData.State(email)== 'add'){
71       UserData.setTitle(email ,userMessage);
72       oneday(replyToken)
73       UserData.NewState(email, 'oneday');
74       return;
75     }
76     if(UserData.State(email)== 'oneday'){
77       var day = msg.events[0].postback.params.date
78       var title = UserData.getPorperty(email)
79       insert(replyToken, email, title,day)
80       UserData.NewState(email, 'established')
81       return;
82     }
83
```

圖 77 流程程式碼

最後再 insert 的 function 裡，將記錄下來的主旨與時間，送出給特定網址，一樣要加上 accesstoken。

```
function insert(replyToken, email, title,day){
  var proper = PropertiesService.getScriptProperties();
  var cache = CacheService.getScriptCache();
  var accessToken = cache.get(email);

  var summary = title[0]
  var postPayload = {

    "summary":summary,
    "start":{
    "date":day,
  },
    "end":{
    "date":day,
  },
  }
   var payload = JSON.stringify(postPayload);
   var options = {
      'contentType': 'application/json',
      "method" : "post",
      "payload" : payload,
    };
  var url = 'https://www.googleapis.com/calendar/v3/calendars/'+email+'/events?access_token='+accessToken+'&alt=json&prettyPrint=true';
  var test = UrlFetchApp.getRequest(url,options)
  var result = UrlFetchApp.fetch(url,options);
```

圖 78 新增事件的程式碼

最後是今日行程與名日行程，我的名字我是誰和不知道的句子

```
    case 10:
    case 110:
    case 111:
     GetTodayList(replyToken,email);
     break;

    case 120:
    case 121:
     GetTomorrowList(replyToken,email);
     break;

    case 442 :
    case 2000:
      yourName(userId,replyToken);
      break;
    default:
      unknow(replyToken);
```

圖 79 今明日行程的等程式碼

因為是查詢一整天的行程，所以需要從昨日的 23:59:59 查詢到明日的 00:00:00，

```
function TodayList(email) {
  var cache = CacheService.getScriptCache();
  var accessToken = cache.get(email);
  var tomorrow = new Date()
  tomorrow.setDate(tomorrow.getDate() + 1)
  var yesterday = new Date()
  yesterday.setDate(yesterday.getDate() - 1)
  var tomorrow =  Utilities.formatDate(tomorrow, "GMT+8", "yyyy-MM-dd");
  var yesterday =  Utilities.formatDate(yesterday, "GMT+8", "yyyy-MM-dd");
  var url = 'https://www.googleapis.com/calendar/v3/calendars/'+email+'/events?access_token='+accessToken+'&timeMax='+tomorrow+'T
  var reasult  =  UrlFetchApp.fetch(url);
  var item = JSON.parse(reasult.getContentText());

  var itemlength = Object.keys(item.items).length;
  if(itemlength==0){ return 0;}
   var title =  new Array();
   var time = new Array();
  for(i=0;i<itemlength;i++){
       title[i] = item.items[i].summary;
       var timelength = JSON.stringify(item.items[i].start).length;
    if(timelength <25){
      time[i] = 'All day';
    }else{
      time[i] = JSON.stringify(item.items[i].start.dateTime).substring(12,20);
    }
  }
  return [title, time];
}
```

圖 80 查詢今日的程式碼

另外可以使用 var url = 'https://api.line.me/v2/bot/profile/'+userId;來得到你 Line 上面的名字再加已回傳給使用者，算是小小互動

```
function yourName(userId,replyToken) {
  var proper = PropertiesService.getScriptProperties();
  var CHANNEL_ACCESS_TOKEN = proper.getProperty('CHANNEL_ACCESS_TOKEN');
  var url = 'https://api.line.me/v2/bot/profile/'+userId;
  var options = {
      'headers': {
      'Content-type': 'application/json; charset=UTF-8',
      'authorization': 'Bearer ' + CHANNEL_ACCESS_TOKEN,
    },
    'method':'get',
  }
  var profile = UrlFetchApp.fetch(url, options);
  var params = JSON.parse(profile.getContentText());
  var name = params.displayName;


  var url = 'https://api.line.me/v2/bot/message/reply';
  UrlFetchApp.fetch(url, {
      'headers': {
      'Content-Type': 'application/json; charset=UTF-8',
      'Authorization': 'Bearer ' + CHANNEL_ACCESS_TOKEN,
    },
    'method': 'post',
    'payload': JSON.stringify({
      'replyToken': replyToken,
      'messages': [{
        'type': 'text',
        'text': '你的Line上面名字是:' + name,
      }],
    }),
  });
}
```

圖 81 Line 回復的程式碼

最後當使用者說出不知道的句子時，會回傳下列 unknown 陣列裡的回覆

```
function unknow(replyToken) {
  var unknown = ['你能換種說法在說一次嗎','請你換種說法試試','我會試者努力學習，你可以用別的方式說一次嗎?'];

  var random = Math.floor(Math.random()*4);

  var proper = PropertiesService.getScriptProperties();
  var CHANNEL_ACCESS_TOKEN = proper.getProperty('CHANNEL_ACCESS_TOKEN');
  var url = 'https://api.line.me/v2/bot/message/reply';
  UrlFetchApp.fetch(url, {
      'headers': {
      'Content-Type': 'application/json; charset=UTF-8',
      'Authorization': 'Bearer ' + CHANNEL_ACCESS_TOKEN,
    },
    'method': 'post',
    'payload': JSON.stringify({
      'replyToken': replyToken,
      'messages': [{
        'type': 'text',
        'text': unknown[random],
      }],
    }),
  });
}
```

圖 82 回傳不確定的程式碼

八、Line 操作說明

在開始操作前，需要先加好友，這是 ID@198txmnz 和 QRcode, 加完好友後，並且點進去聊天室，會出現如圖 4-1 畫面

圖 4-1 加入好友之後的畫面。

聊天室機器人會在第一段會先傳送「您好，使用者姓名」，這是在 Line 官方帳號管理網頁裡就可以設定的功能，第二段式歡迎詞則會簡單介紹有甚麼樣的功能，往下請點選認證的按鈕。

在點選認證後，會出現如圖 4-2 的畫面，聊天室機器人會傳送是否確認的框框，這時請點選登入。

圖 4-2 登入 Google 選項

點選登入後，會出現這個應用程式未經驗證如圖 4-3 的畫面。

圖 4-3 應用程式未經驗證

此時點選進階，點選進階後會出現此畫面如圖 4-4，再點選前往 LineCalendar。

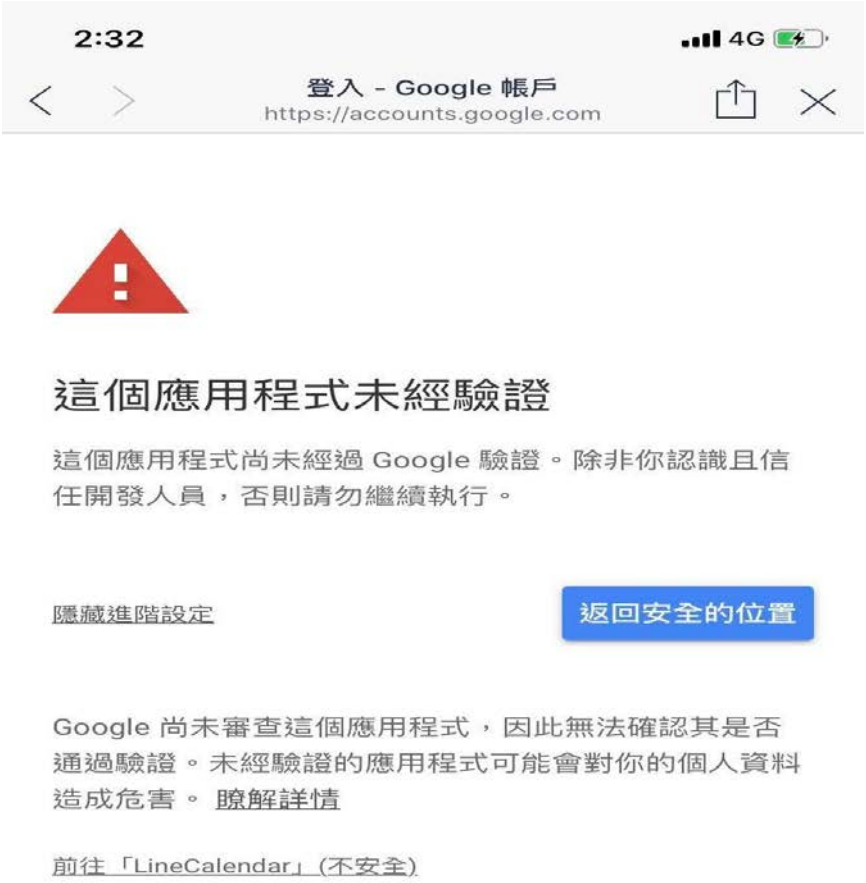


圖 4-4 前往 LineCalendar

前往之後會出現此畫面如圖 4-5，點選允許。

圖 4-5 允許授權于 LineCalendar

畫面顯示如圖 4-6 代表登入成功就能關掉畫面。

圖 4-6 登入成功

驗證成功後，便能開始使用此聊天機器人，功能如下圖 4-7。

圖 4-7 聊天機器人功能

建立行程，開始建立事件，如下圖 4-8。

圖 4-8 使用聊天機器人的畫面

建立完成事件後，便會要您新增時間如圖 4-9。

圖 4-9 使用聊天機器人的畫面

查詢行程，打出所建立過的行程，就會出現行程的次數，如圖 4-10。

圖 4-10 查詢行程次數

顯示今日有無行程，如圖 4-11。

圖 4-11 顯示行程

第五章問題討論

5-1 Google 的認證問題

首先，Oauth2.0 會有以下人物:

- 第三方應用程式(Third-party application)
- 用戶(Resource Owner)
- 認證服務器(Authorization serve)

當第三方應用程式向認證服務器要 access_token，認證伺服器會在生成之前，先詢問 Google 用戶能不能授權，用戶同意後，認證伺服器會生成並發行 access_token 給第三方應用程式，而第三方應用程式變可以使用 access_token 來獲取他所要求範圍內的資源。

而在要 access_token 之前的的步驟是建立一個授權請求，在這個請求中，需要放置一些參數來完整這個請求。

必須要有的參數是 client_id、redirect_uri、scope，另外有兩個建議的是 access_type 和 state, 其中 state 參數是指目前的狀態，可以為任何值，認證服務器會原封不動的返回這個值，主要是防範 CSRF 攻擊。

那要如何把 state 加進去，在 Apps Script 裡面有一個類別是 StateTokenBuilder，他會生成一個 state 的任意字串，於是在當初建構的時候變有使用到了這個方法，但是每一次點選完同意第三方應用程式使用權限時，會出現如圖 5-1 的畫面，試著更改過其他字串，但是只要有包含 state，就會得到這個畫面。最後我們把 state 拿掉，每一次的登入都有成功，但是 CSRF(跨站請求偽造)還是不能忽視。

在未來，我們會找尋其他安全的登入方法或是找到可以讓

state 正確的方法來讓使用者登入。

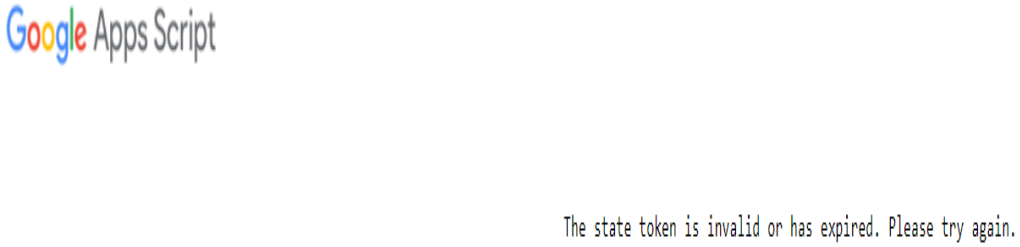


圖 5-1 state token is invalid or has expired

5-2 Google 驗證問題

當進行登入時，會出現如圖 5-2 的畫面，因為使用了 https://www.googleapis.com/auth/calendar 這個敏感的範圍，只要 Google 使用者同意後，我就可進行編輯、新增、刪除和修改等動作，所以 Google 要求進行驗證在公開之前。所以我需要認證我的網域，為此我去架設了一個網站並且簡單放置了隱私政策的網頁，並且傳送申請。

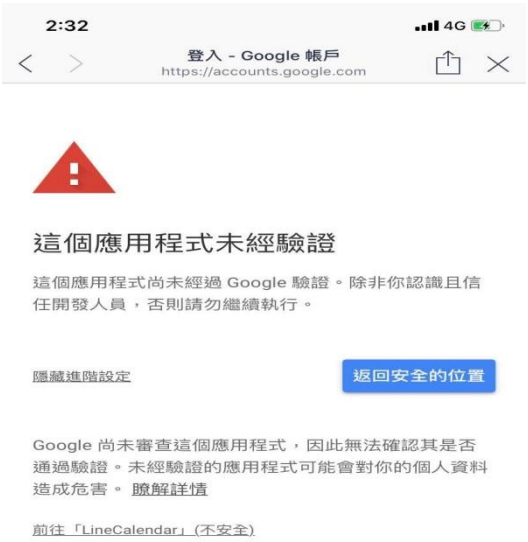


圖 5-2 應用程式未經驗證

而最終得到了圖 5-3 的畫面，為什麼他依然還是未經驗證，因為如果你在應用程序中有要求限制或敏感的範圍，開發者就需要解釋你的用途，來確保用戶的安全，包含:

1. 要是經過驗證的網域
2. 用戶可以準確、簡單的進行訪問
3. 透明的說明請求 Google 用戶數據的目的
4. 加強描述應用程式如何增加功能
5. 有連結可以到隱私政策網頁裡，網頁裡要說明如何使用，存儲 Google 用戶的數據

必須要包含這些內容，並重新更新，Google 才會繼續進行驗證，未來會在網頁上進行更詳細的補充，並且完整隱私政策裡的內容，讓用戶可以使用安心。

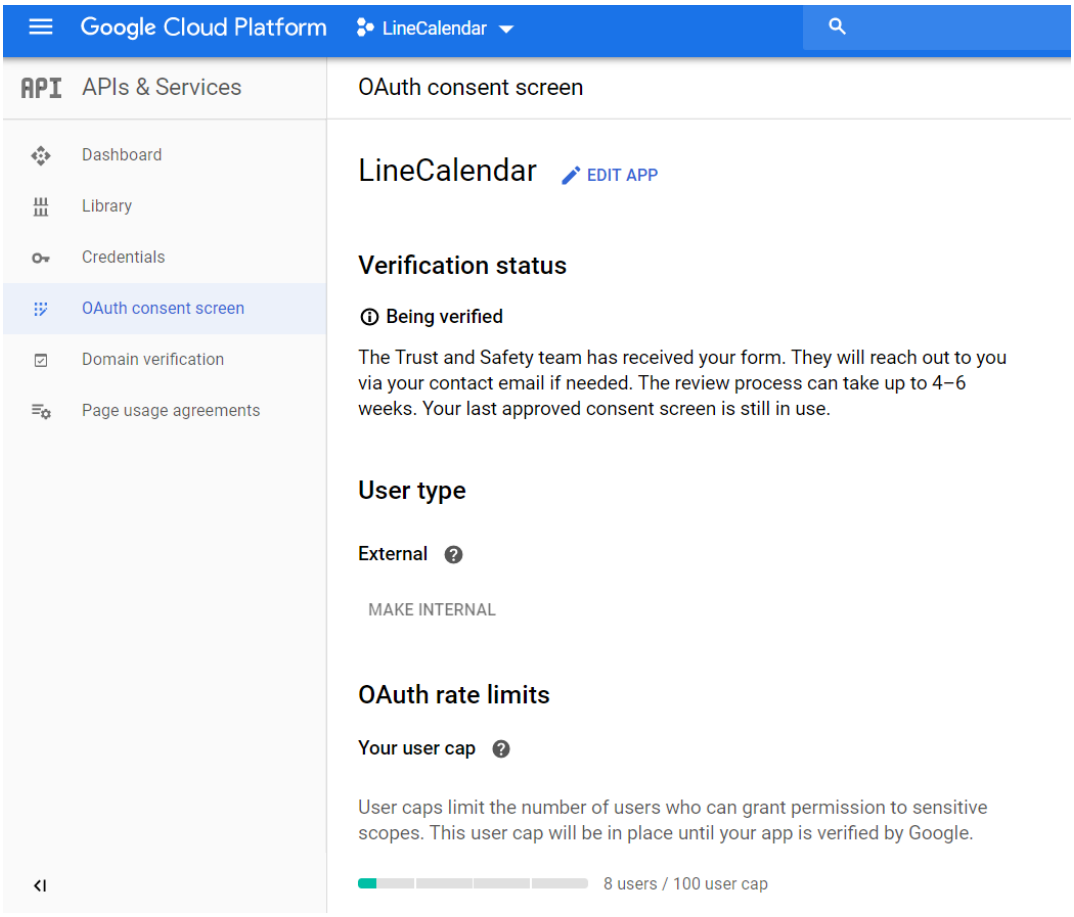

Google Cloud Platform
LineCalendar
APIs & Services
Dashboard
Library
Credentials
OAuth consent screen
Domain verification
Page usage agreements
OAuth consent screen
LineCalendar
EDIT APP
Verification status
Being verified
The Trust and Safety team has received your form. They will reach out to you via your contact email if needed. The review process can take up to 4–6 weeks. Your last approved consent screen is still in use.
User type
External
MAKE INTERNAL
OAuth rate limits
Your user cap
User caps limit the number of users who can grant permission to sensitive scopes. This user cap will be in place until your app is verified by Google.
8 users / 100 user cap

圖 5-3 正在驗證中

5-3 待改善之處

日曆的多樣化:

目前的在 Line 上僅能執行簡單的查詢行程，建立行程，並未活化日曆的功能，例如說顯示台灣的國慶假日或是喜歡看籃球的，可以查詢到球隊的比賽日等等日常生活。

更完整的詞庫:

目前為了讓使用者能輕鬆使用，我們假設了一些使用者可能會使用的句子和 Google 語音翻譯可能會翻錯的字來建立了一個非常小的詞庫並簡單的分類，未來希望能加強這個詞庫，來達到更方便的使用。

完成 Google 的驗證:

目前使用者登入時依然會看到還沒經過驗證，未來會完整隱私條款和服務條款等的內容，讓使用者安心。

結論

隨著科技的日新月異，「人手一機」已漸漸成為人們生活的一部分，甚至使用者橫跨全部的年齡層，使用率極為普遍，手機的分類也從「家電類」轉成一種「消耗類」的型態，高淘汰率造成產業無限的商機。

由於雲端技術持續成長，行動 App 開發者將會設計出更多以雲端驅動的 App，優勢在於可以在不佔用手機容量的情況下，更快速、容易地獲取資料。手機中安裝一堆佔容量、使用頻率低的 App 是許多人共同的困擾，即時 App 指的是在傳統 App 中增加 Instant 支援。

對話式體驗、聊天機器人持續成為趨勢，企業在 LINE、Whatsapp、Messenger、Skype、WeChat 中導入聊天機器人，能提供 24 小時、不輸真人的個人化客服。以 Line 為例，平台上約有 2000 萬個企業希望透過聊天機器人觸及到更多潛在客戶。此外在現有 App 中導入 Watson、、Line Bot、Google 行事曆、Google Map 等，具備自然語言理解的對話工具，也能創造不同以往的用戶體驗。回顧 2017 年，全球 App 的熱門趨勢主要聚焦在 Google、社群媒體、遊戲、生活風格等類別，App 代表的不僅只是服務的一環，同時也代表品牌形象、行銷、網站導流、提升消費者體驗等意義，隨著時序進入 2018，這些趨勢也將重新定義新 App 經濟。

參考文獻

[1]Mike, 2018,10 個新手必知的 JavaScript 實用技巧

https://medium.com/i-am-mike/10%E5%80%8B%E6%96%B0%E6%89%8B%E5%BF%85%E7%9F%A5%E7%9A%84-javascrip-%E5%AF%A6%E7%94%A8%E6%8A%80%E5%B7%A7-75b55d7c3e47

[2]Arong.chiu ,2018, Google Apps Script 實作 LINE Bot

https://medium.com/@f1236920001/google-apps-script%E5%AF%A6%E4%BD%9Cline-bot-b8e613b81fd1

[3]Javescript Reference

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference

[4]javascript 教程

http://www.runoob.com/js/js-tutorial.html

[5] Google app script

https://developers.google.com/apps-script

[6] Google Calendar API

https://developers.google.com/apps-script

[7]仙草奶綠,2018 , 取得機器人好友帳號 ID + 發送資料的實作 (Line API Push Message)

https://dotblogs.com.tw/milkgreenteaprograme_c_sharp/2018/05/13/214926

[8]陳鈞,2019,以 Google 試算表作為簡易資料庫(上)--資料庫的建立及寫入

https://blog.maki0419.com/2015/06/google-database.html

[9]Yu-Cheng Chuang,2013,OAuth 2.0 筆記 (1) 世界觀

https://blog.yorkxin.org/2013/09/30/oauth2-1-introduction.html

[10]Bryan Helmig,2018,How to Use the Google Calendar API

https://zapier.com/engineering/how-to-use-the-google-calendar-api/

[11]chihokyo,2018,【簡易圖解】『 OAuth2.0』猴子都能懂的图解 https://learnku.com/articles/20031

[12]2018, OAuth2 for Apps Script

https://github.com/gsuitedevs/apps-script-oauth2

[14]Ryan Boyd, 2012, How do I get an access token using UrlFetchApp with GAS

https://stackoverflow.com/questions/11063749/how-do-i-get-an-access-token-using-urlfetchapp-with-gas

[15]AmitAgarwal, 2017,How to Use Google Cloud APIs with Apps Script

https://ctrlq.org/code/20534-google-cloud-apps-script

[17]陳慶培,蘇敬倫, 楊智程,2011,專題管理系統--助教子系統

網

址:http://140.134.131.145/upload/paper_uni/1001pdf/%E5%B0

%88%E9%A1%8C%E7%AE%A1%E7%90%86%E7%B3%BB%E7%B5%B1-%E5%8A%A9%E6%95%99%E5%AD%90%E7%B3%BB%E7%B5%B1.pdf

82