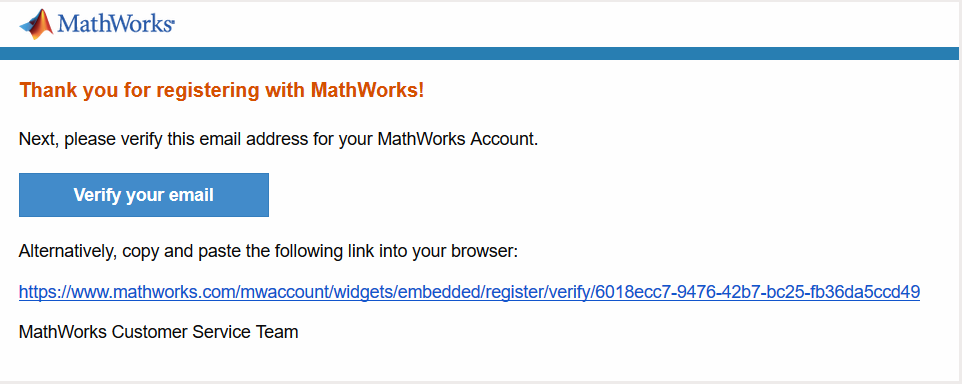


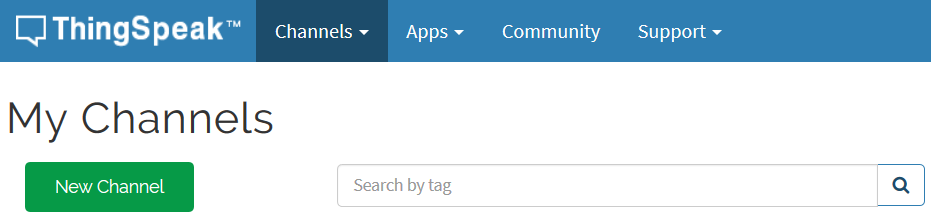
積木用途：發送HTTP請求將資料送到外部網站，使資料能共享並做進一步的應用。

說明：物聯網的應用千變萬化，在此僅介紹兩個應用：一是收集資料、分析、共享的ThingSpeak；另一個是串接網路服務的IFTTT。

1. **ThingSpeak**
2. 連上Thingspeak網站，網址是<https://thingspeak.com/>，並註冊一個帳號。註冊並沒有特殊程序，填完資料後，會發一封確認信到e-mail信箱，按下確認接鈕即可，ThingSpeak使用MathWork的帳號，所以收到的信件是由MathWorks寄發的。



1. 回到ThingSpeak，登入之後即可建立一個新的Channel。

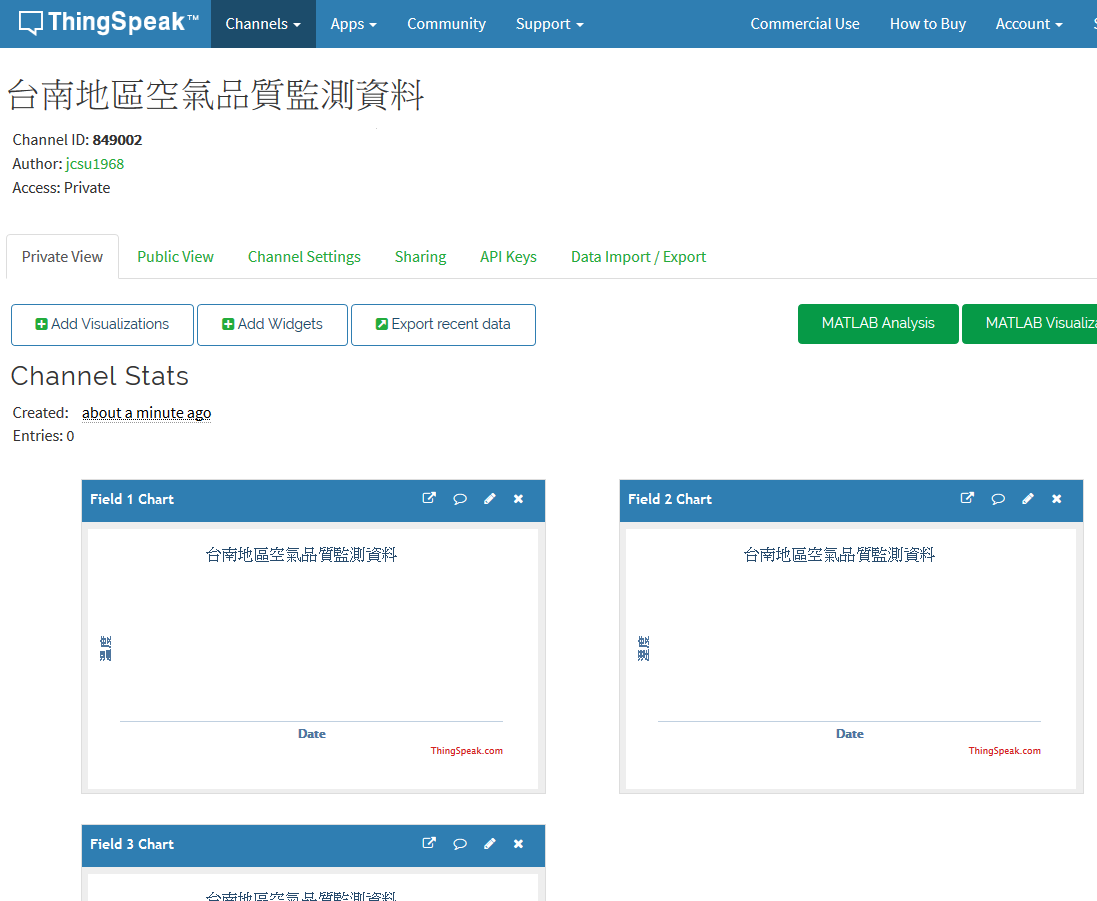


1. 接著填寫Channel的內容，簡單填寫Name(資料表名稱)、Field1(資料項標題)，要收集幾項資料就填幾欄，然後就可以按下表格最下方的Save Channel。

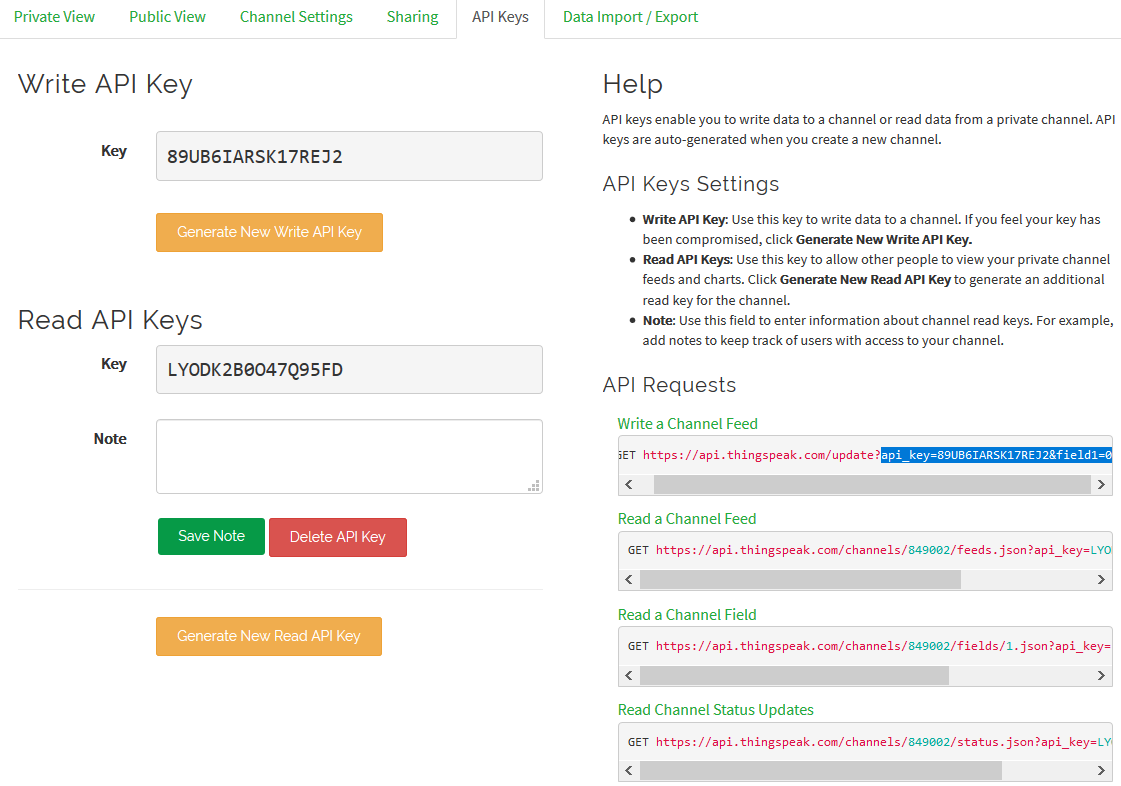




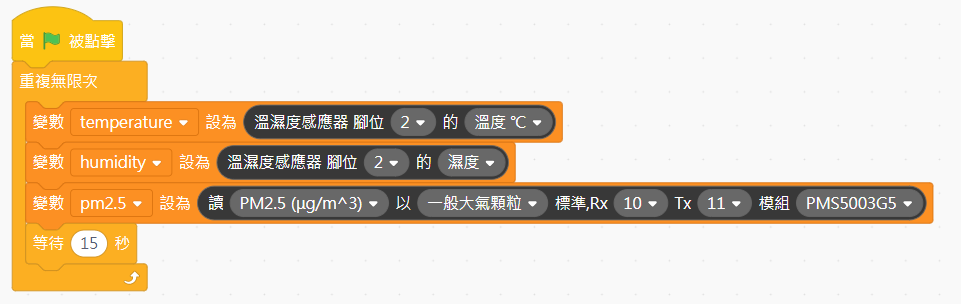
1. Save Channel之後可以看到三個資料項的統計圖框已經建立好，接著請按下API Keys，看看如何讀寫資料。



1. 這裡我們比較關注的是右下角的API Requests，尤其是Write a Channel Feed，是我們要寫進程式裡的資訊，請把Write a Channel欄位裡從”api\_key”以後的文字全部選取並複製起來。



1. 回到NKNU BLOCK，這次會用到CHT11溫溼度感測器與PMS5003空氣品質感測器。先寫好讀取感測器資料的部分，並依照說明完成HTTP Request參數字串的組合。



ThingSpeak的免費授權寫入資料時，每筆資料至少須間隔15秒

把剛剛複製的Write a Channel Feed字串貼在這裡並刪掉最後一個字”0”

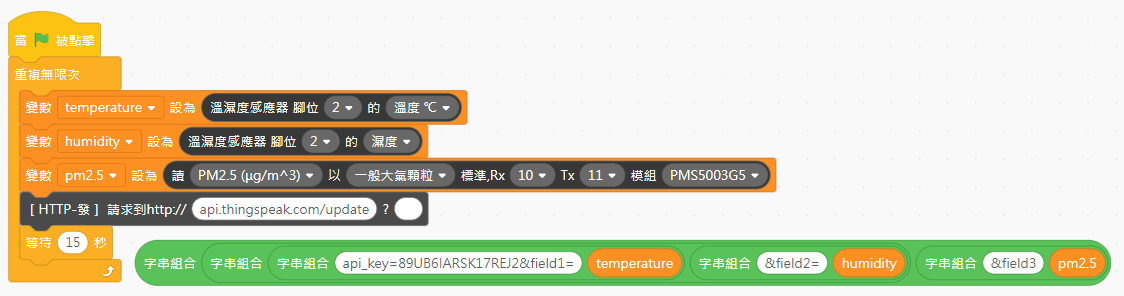


1. 進一步組合HTTP Request參數字串

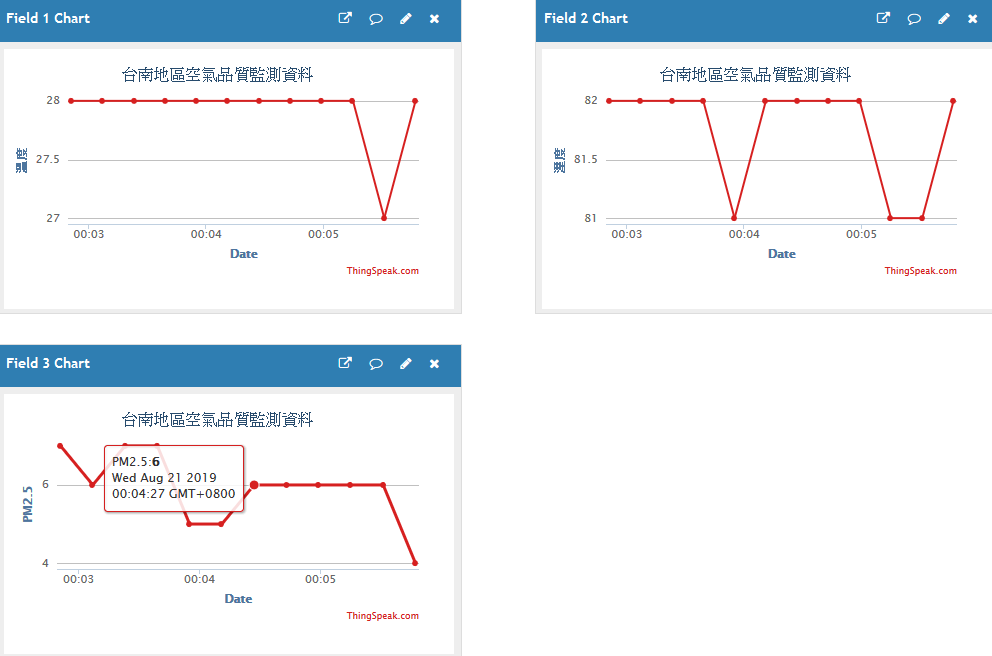


1. 接著就可以把發送http請求積木拉進程式中，記得在”http://”後面填上網址” api.thingspeak.com/update”，並將組合好的參數字串拉進積木的第2欄。

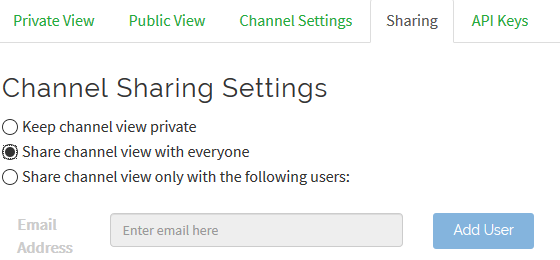
api.thingspeak.com/update



1. 執行程式後，就可以在ThingSpeak網站看到即時繪製的統計圖了。您也可將程式燒錄到Arduino Nano，只要給予電源，不需連接電腦也可以運作。



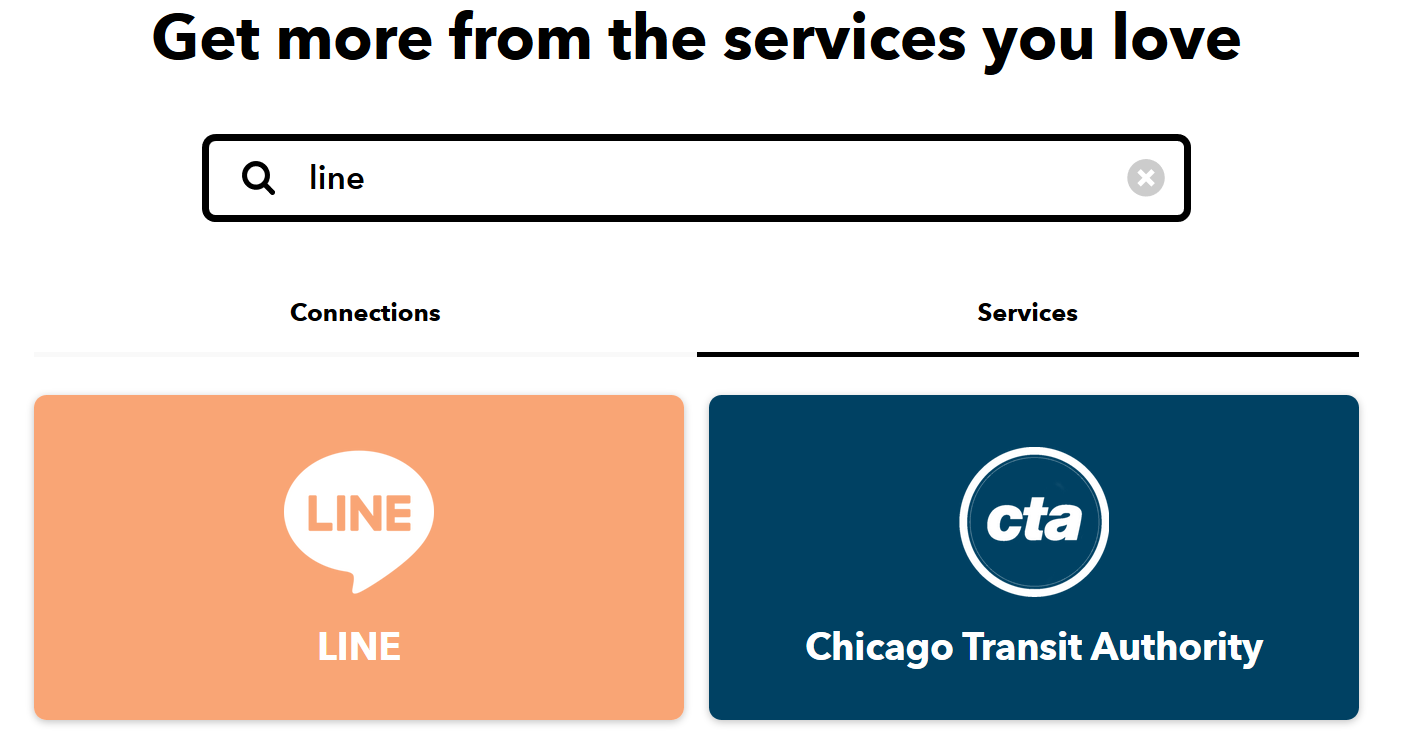
1. 不過這些資料只有上傳者才看得到，如果要分享給別人，請到Sharing項目中改變分享方式。



1. 除了運用感測器即時監控，一筆一筆加進資料外，也可在”Data Import/Export”整批上傳或下載資料。
2. **IFTTT**

IFTTT(If this then that)是一個網路自動聯結服務，讓我們不用寫程式也可以將常用的網路服務串在一起，例如Facebook、LINE、Gmail、Google Drive……等，當某個網路服務發生某個條件後，觸發另一個網路服務進行某個反應行動。底下就以”當溫度、溼度異常時，就在LINE上通知我”為例來介紹IFTTT的操作過程。

1. 首先到IFTTT網站註冊一個帳號，可以使用Google或Facebook帳號登入，或使用email註冊一個新帳號。
2. 登入後會顯示IFTTT可以串接的服務，因為我們要使用LINE，所以直接搜尋LINE，接著點右方的Service，就可以找到LINE，請點選LINE圖示。

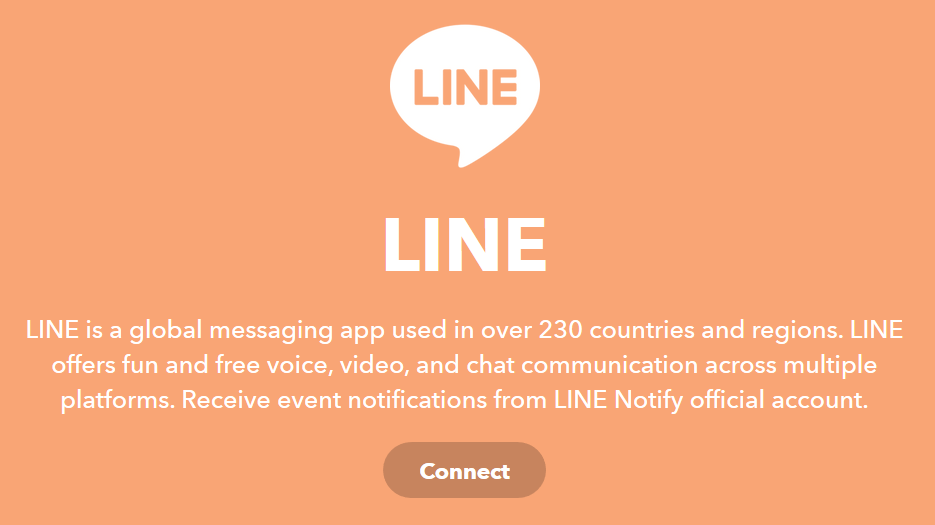


**3**

**2**

**1**

1. 點選Connect按鈕，以便連結LINE官方所提供的Notify通知服務。



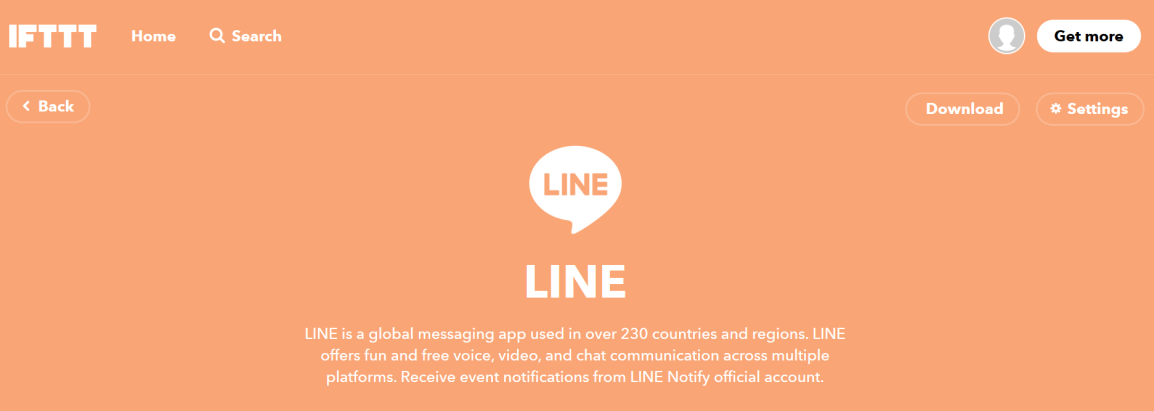
1. 接下來會進入LINE的登入畫面，請用自己的帳號密號登入。

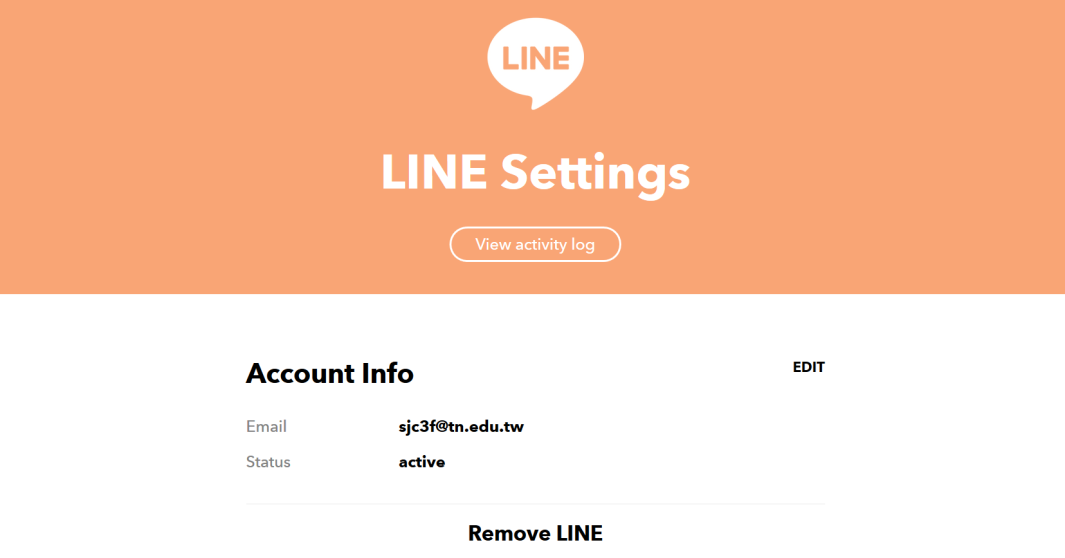


1. 接著會徵求您同意連動，同意後會將「LINE Notify」官方帳號加入好友。

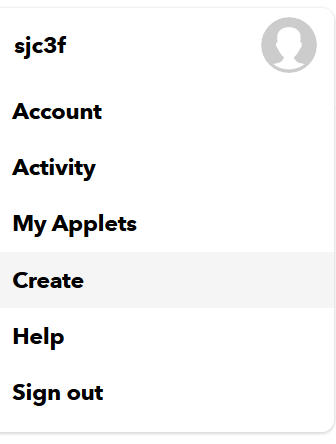


1. 連動完會回到IFTTT畫面，點擊右上方的Settings, 就可以看到Line服務已啟動，如果想取消，點擊畫面下方的Remove LINE即可。

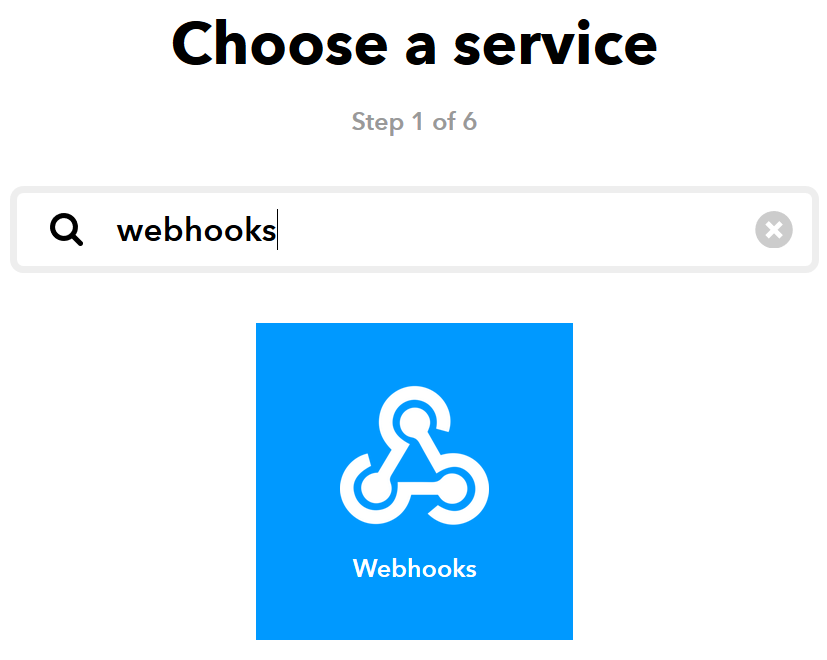




1. 建立好LINE Notify的連結後，回到IFTTT首頁，要開始處理發送訊息的服務。請點擊右上方帳號人頭圖示，選擇Create，接著點選if + This Then That的”This”。



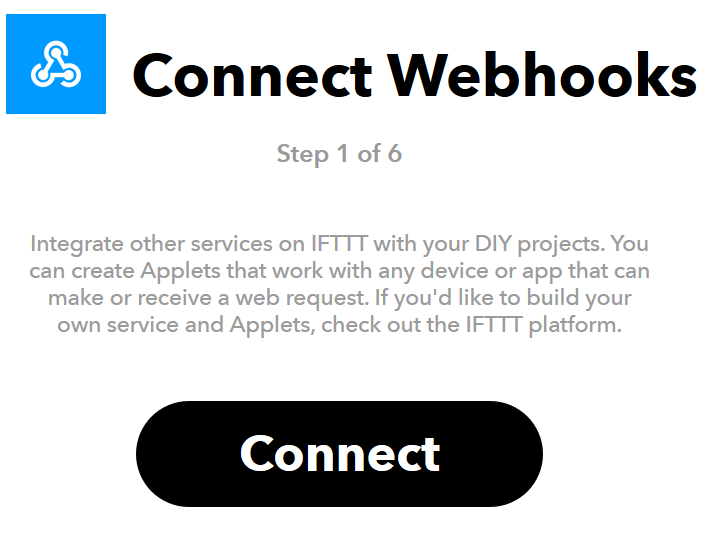
1. 搜尋Webhooks服務，找到後點擊Webhooks圖示。

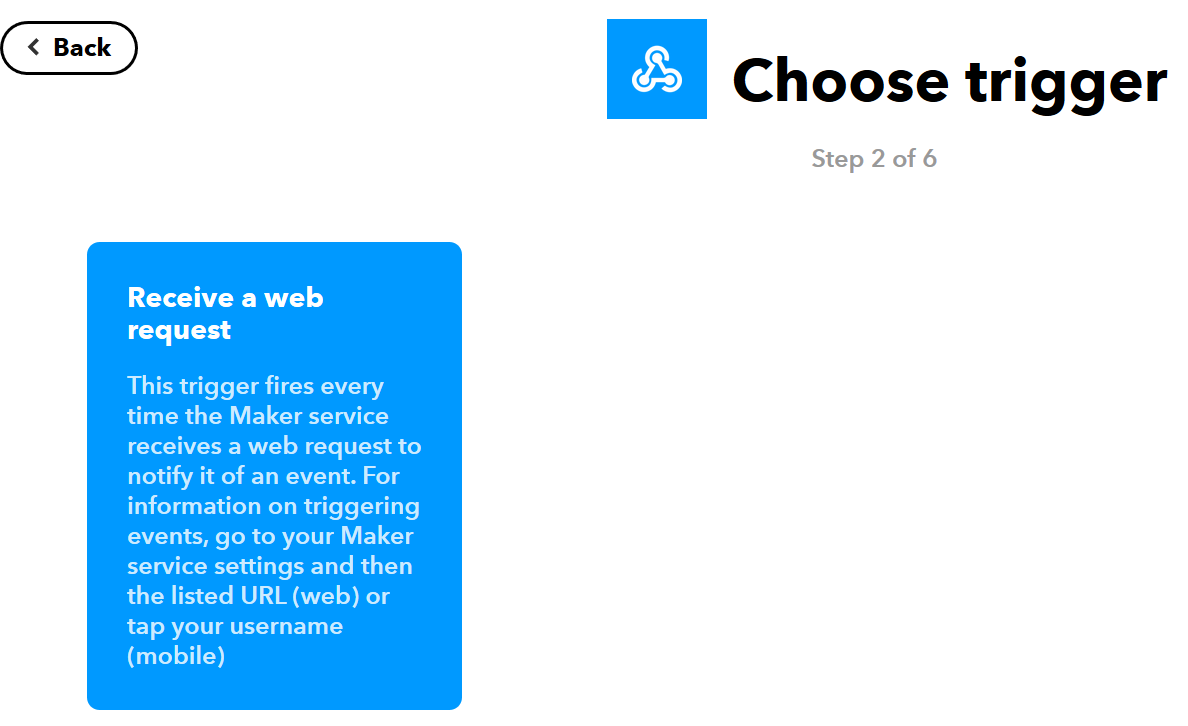


**1**

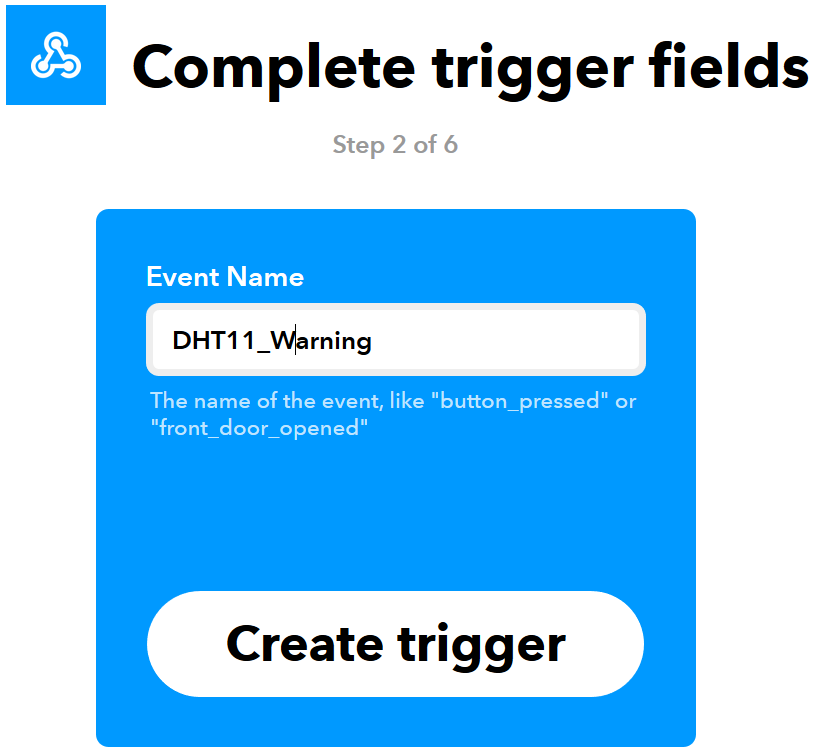
**2**

1. 連接Webhooks服務之後，點選Receive a web request 觸發服務。





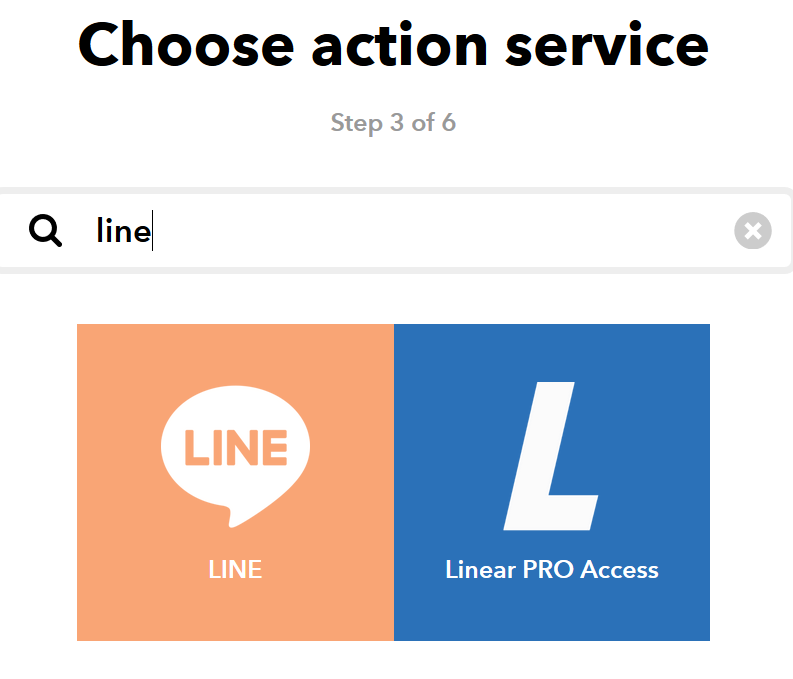
輸入觸發事件名稱後，按下Create trigger 按鈕就完成Webhooks的設定。



1. 回到前一個畫面後，可以看到If This Then That 的This已經變成Webhooks的圖示，接著請點擊That選擇要被觸發的服務。

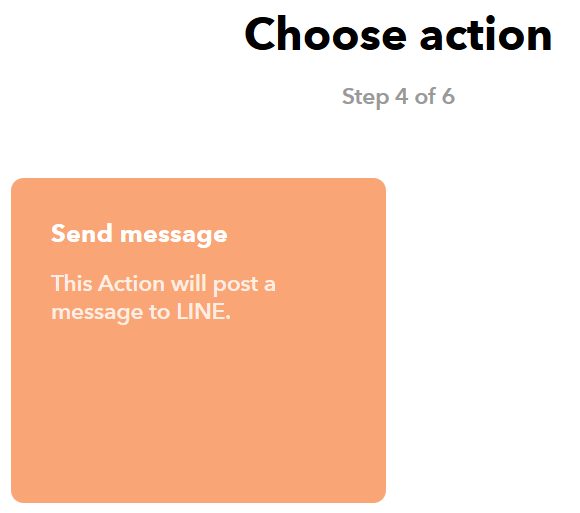


1. 因為是要用Line發送訊息，所以搜尋line，找到後按下LINE圖示，然後點擊”Send message”



**2**

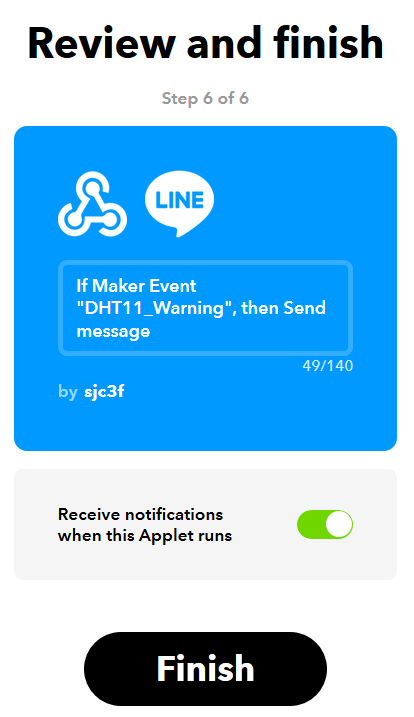
**1**



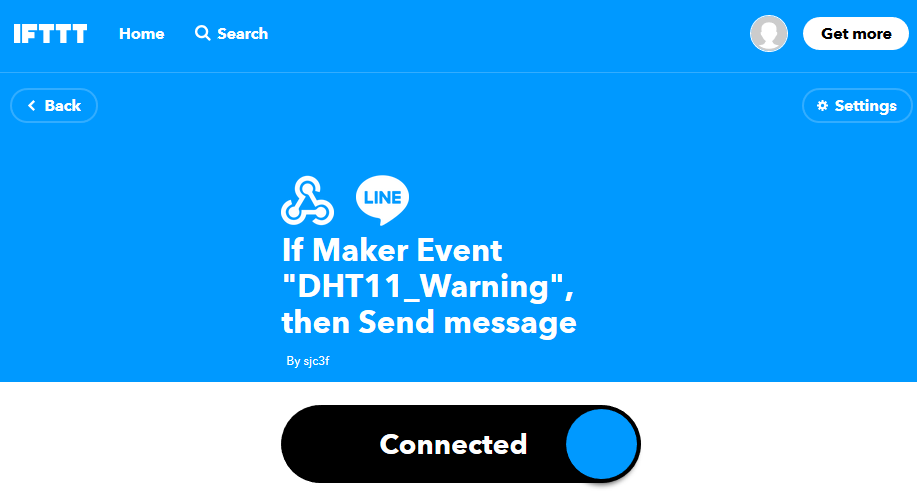
1. 設定傳送訊息的格式，訊息標籤可以自行修改，由於DHT11只能偵測溫度與溼度，所以只要保留{{Value1}}與{{Value2}}，注意外部兩組大括號{{}}不可刪除，<br>是換行的意思，編輯好確認之後按下Create action即可建立被觸發的行為了。

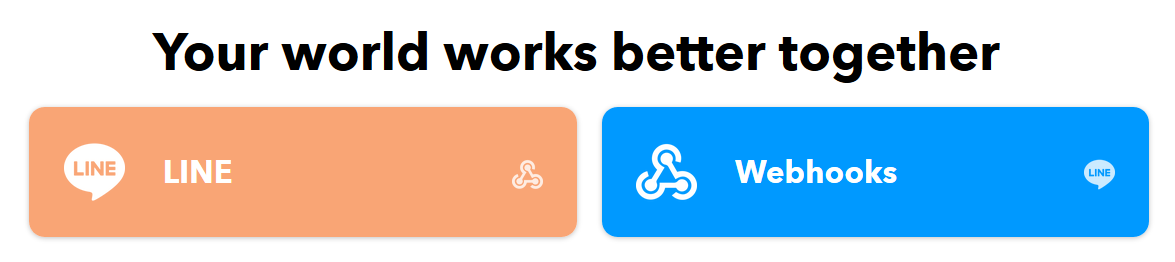
1. 按下Finish按鈕完成 if Maker Event “DHT11\_Warning”, then Send message 服務連結。

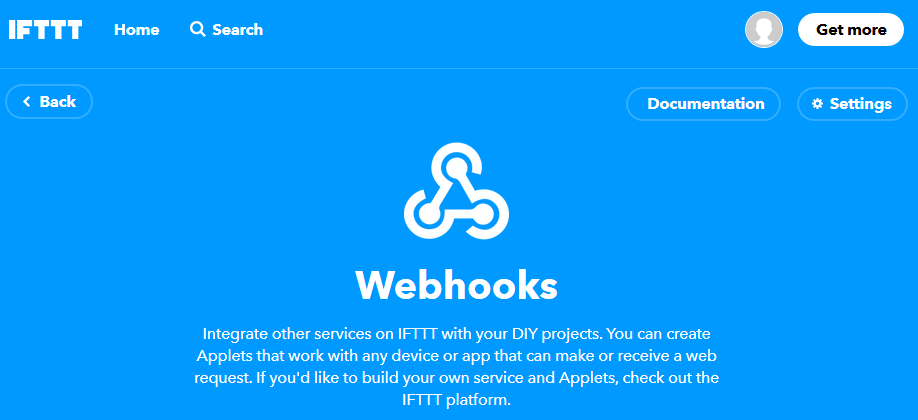


連結完成，若需要修改連結設定，請點擊右上方Settings。

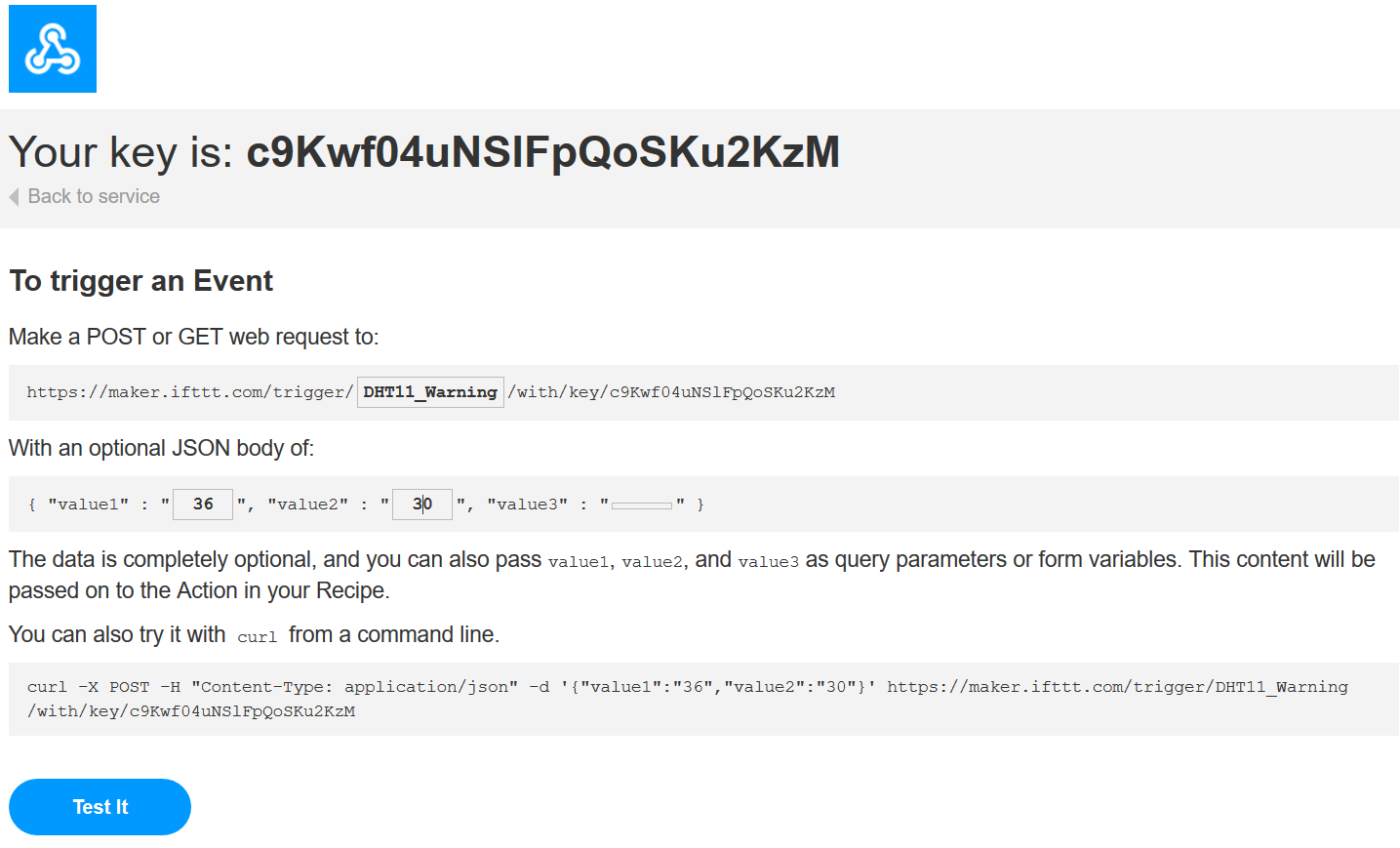


1. 連結設定完成後，怎麼使用呢？先回到IFTTT首頁，點擊右邊的Webhooks，然後再點擊右上方的Documentation。

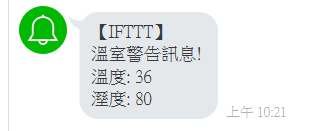




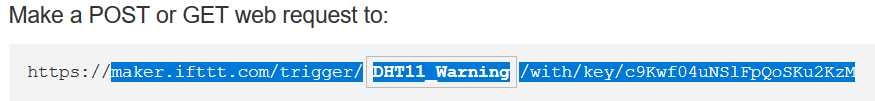
1. 輸入觸發事件名稱與模擬訊息，按下Test It 按鈕測試是否能將訊息傳送給LINE。



成功的話，LINE就會收到底下的訊息。



1. 測試成功的話，請框選Make a POST or GET web request to: 底下欄位中的訊息(從maker開始選到最末端)並複製起來。



1. NKNU BLOCK的程式中，先處理好DHT11溫度、溼度的偵測，以及需要傳送通知的數值範圍，然後將複製好的字中貼入http請求積木的第一個欄位，串接要傳送的參數，放進http請求積木的第二個欄位。若測試無誤，可以將程式燒錄至Arduino Nano中，就不需要連接電腦了。

