

**ECO KIT**

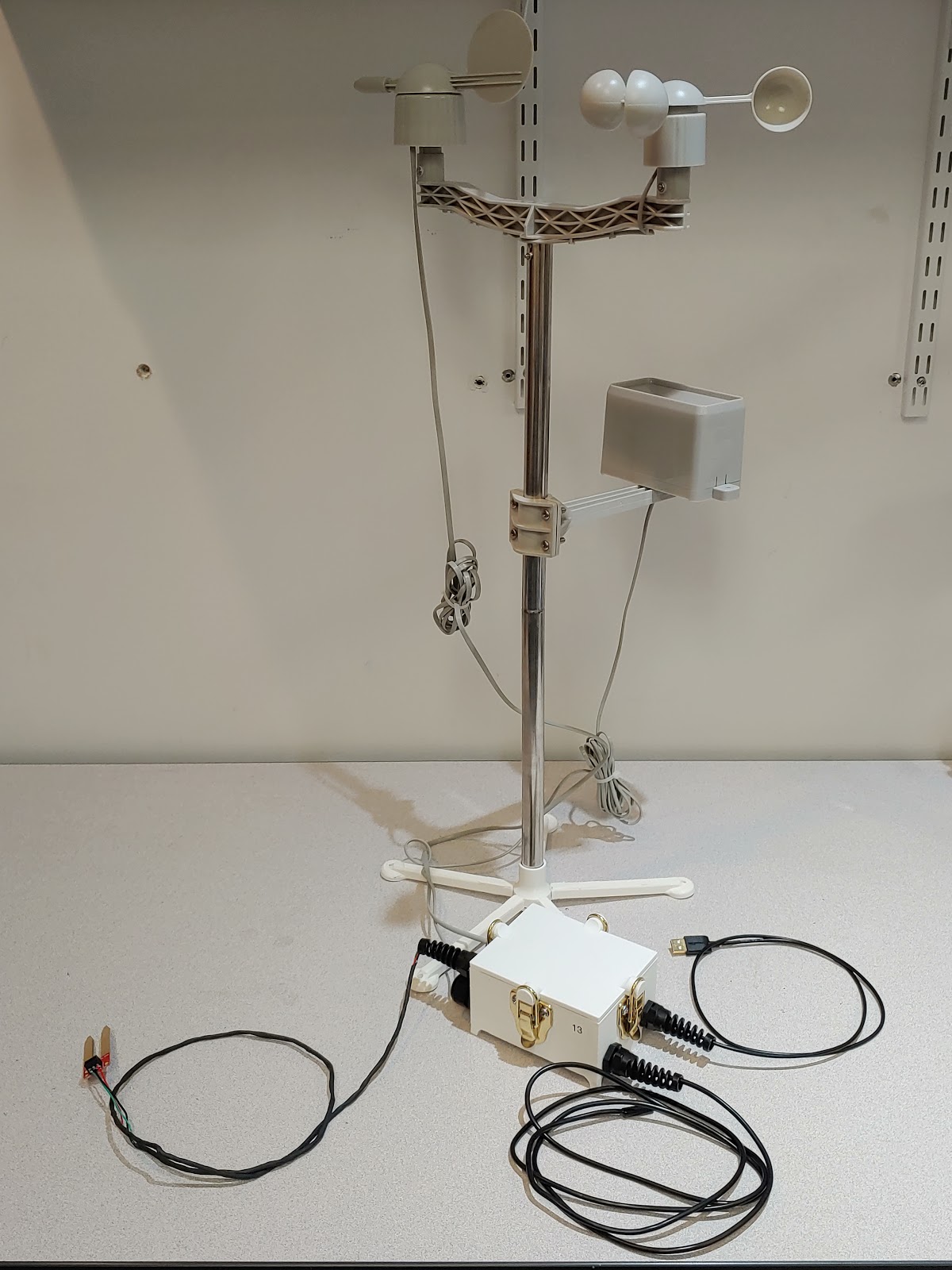


Table of content

WIFI setup – 3

Eco Kit setup – 3

Photon color codes - 3

Downloading the weather app- 4

Running App - 5

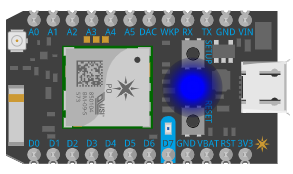
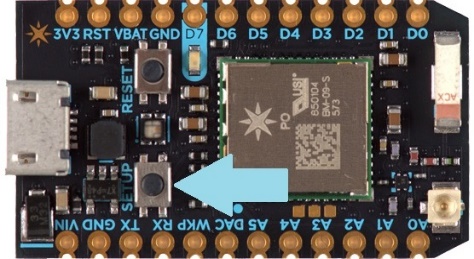
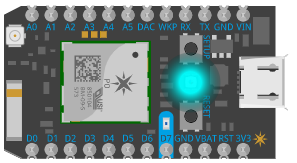
Using App – 6

Getting the Data from the Photon - 7

Setting Up Google Sheets - 8

Adding new weather station to weather app - 16

Connecting Wi-Fi to photon:

1. Power up photon
2. Navigate to [https://setup.particle.io](https://setup.particle.io/)
3. Select Photon
4. Press continue with local file and download the setup file
5. Click the download and a new window should open up
6. Hold down SETUP on your Photon for 3 seconds until the LED begins blinking dark blue
7. From your Wi-Fi network list, connect to the one named 'Photon-...'
8. If it displays “Error: HTTP timed out” reload the page
9. If your photon is breathing cyan you successfully connected
10. Select complete setup without naming your device
11. You can exit the page now

Eco Kit setup

1. Find a flat location to setup the Eco Kit
2. Place the soil moisture sensor and soil temperature sensor into dirt that you want to monitor
3. Plug the usb cable into a computer or USB wall charger to power the system up
4. The weather station should start reporting data
5. If data isn't reporting check the inside of the box and the photon board status will be displayed by the color it is emitting

Photon color codes

Breathing cyan - Board is connected to wifi

Blinking cyan - Actively connecting to wifi

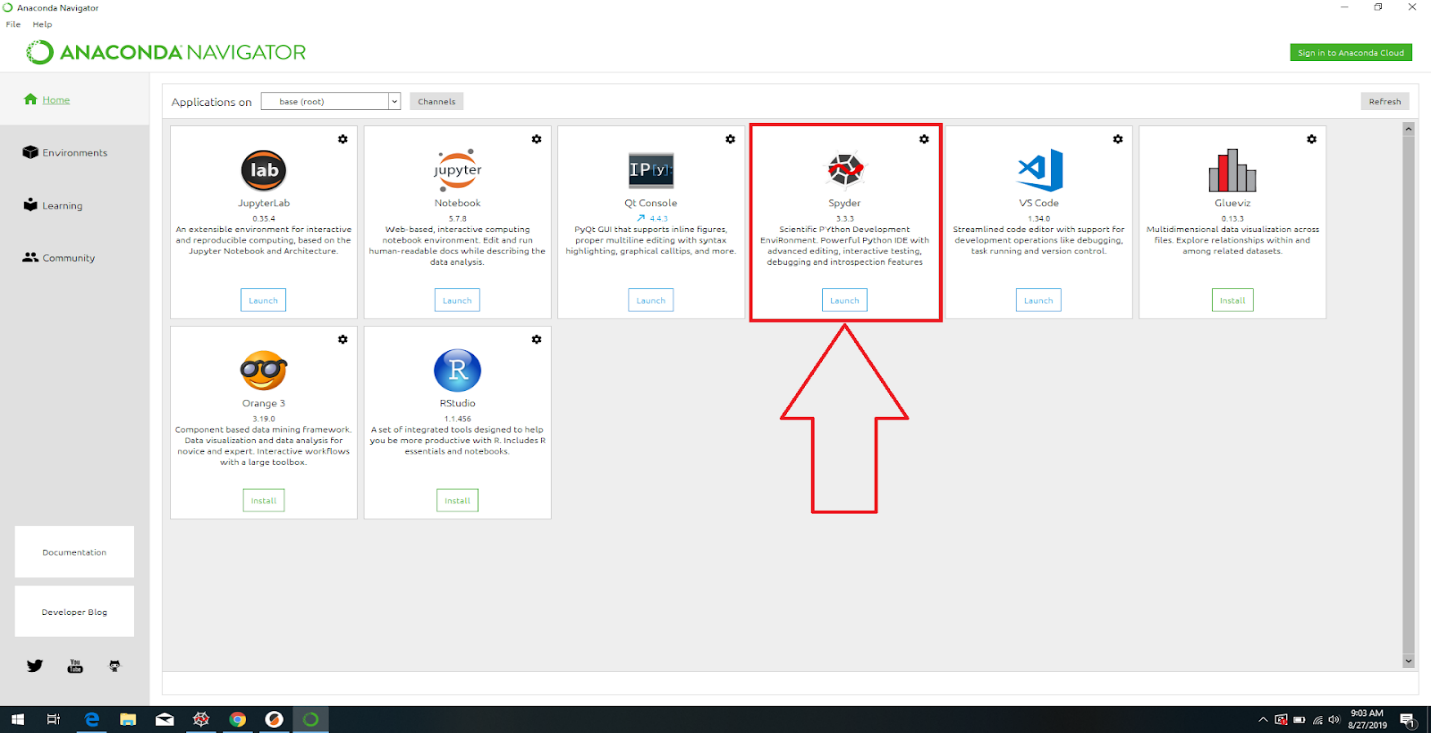
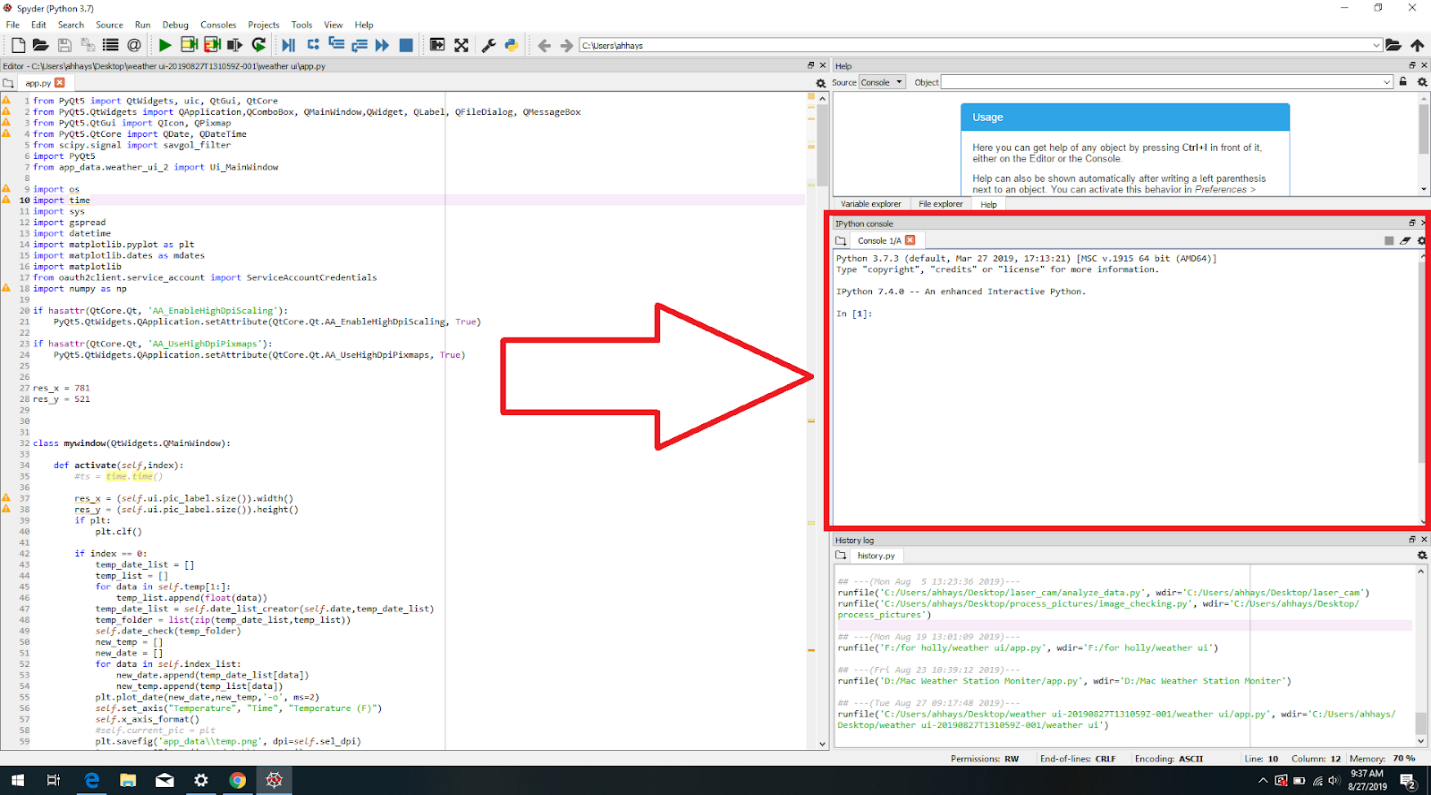
Blinking green - Looking for wifi

Blinking dark blue - waiting for you to setup wifi

Breathing dark blue - wifi on but cannot connect to network

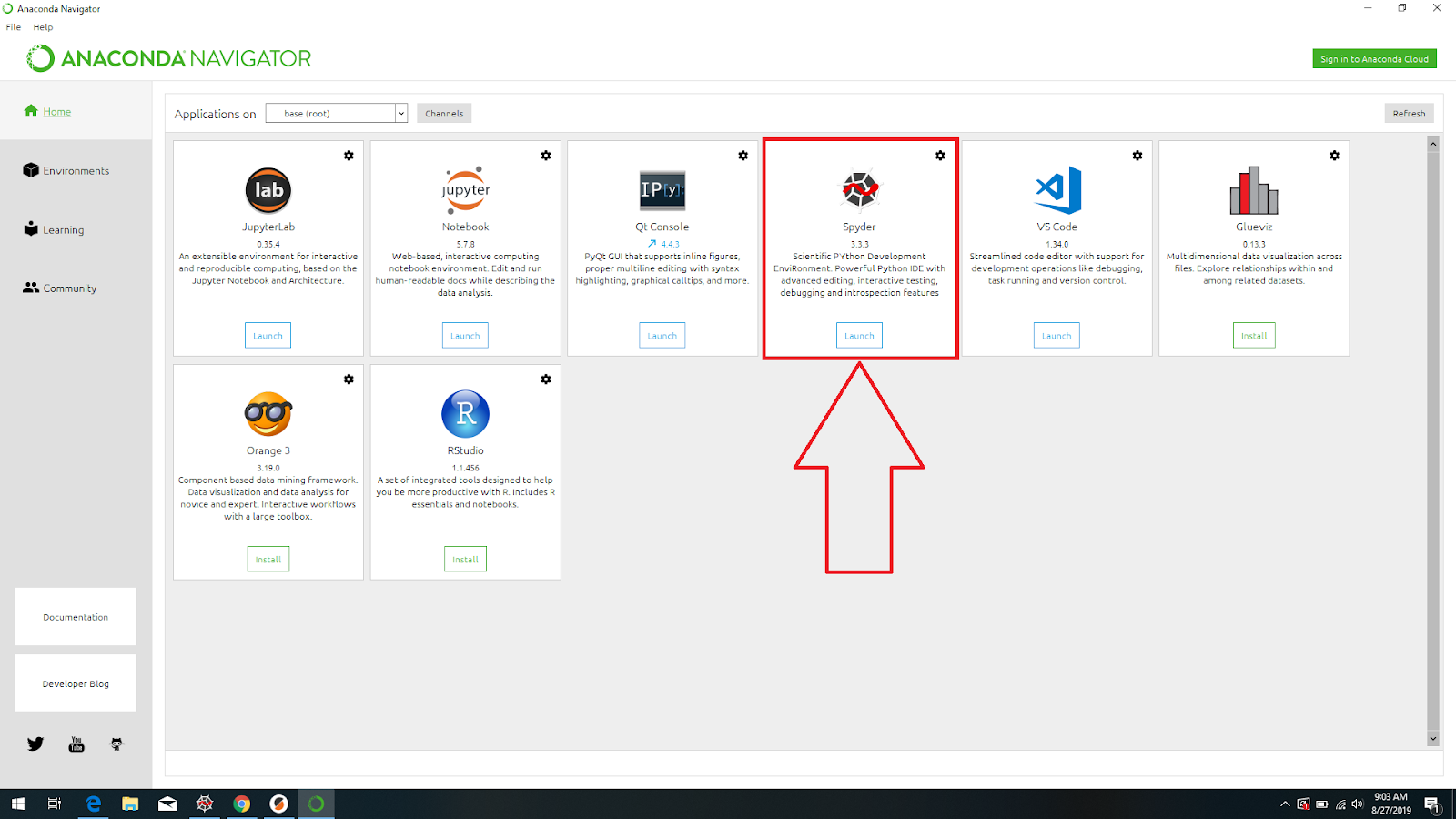
Breathing green - not connected to cloud

Downloading the weather app

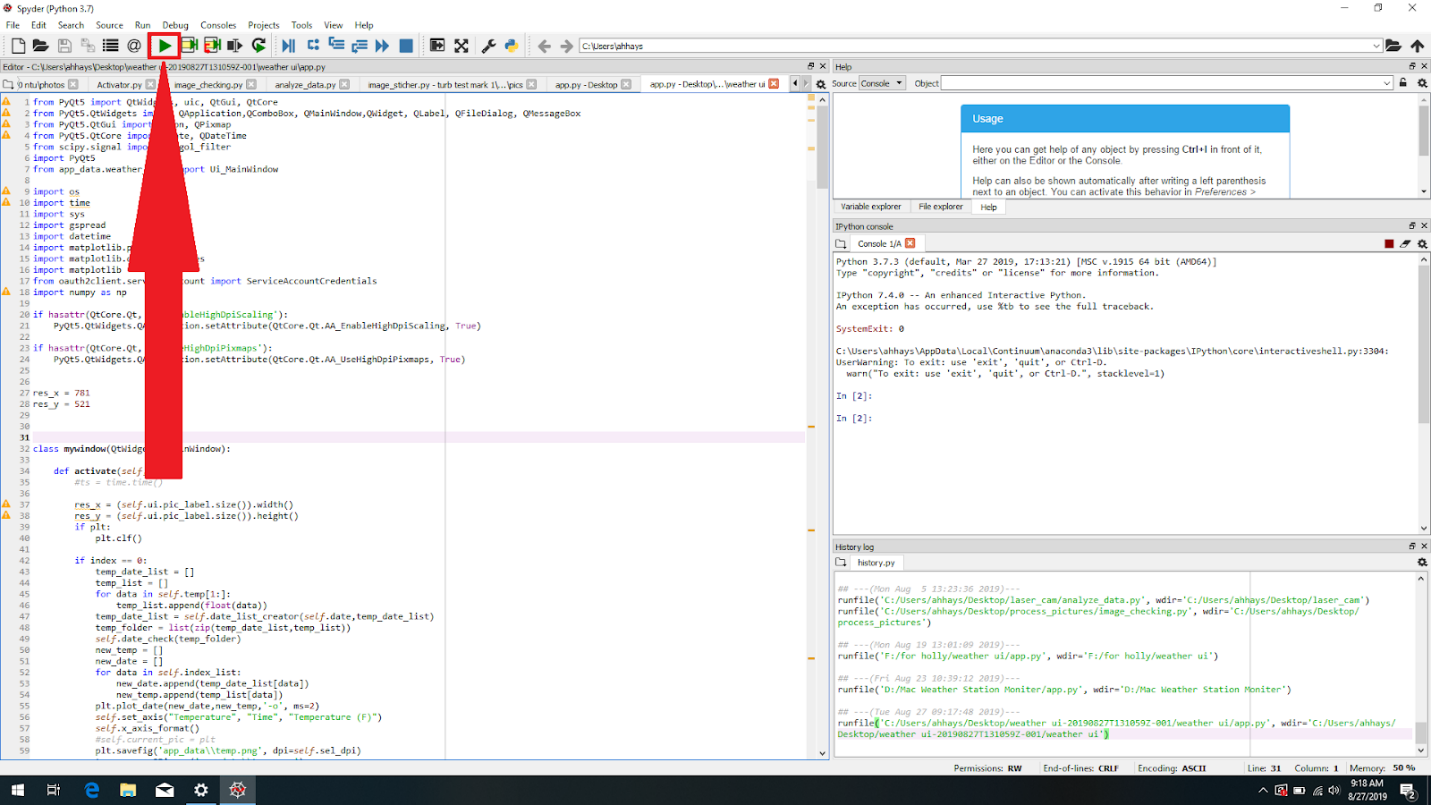
1. Download a python ide for this app we will use anaconda.
2. Go to <https://www.anaconda.com/distribution/#download-section> and download the latest anaconda version for your operating system
3. Once anaconda is downloaded type in “Anaconda Navigator” to your search bar and left click “Anaconda Navigator”
4. Click launch on the Spyder app
5. Type “pip install gspread” and “pip install oauth2client” into the console located in the right side of the screen

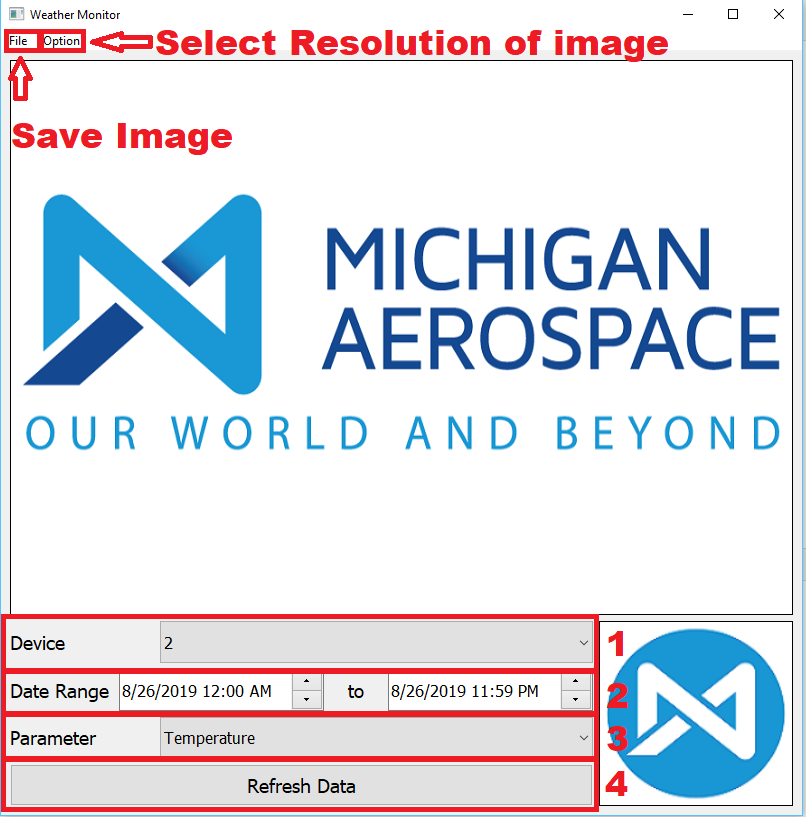
Running app

1. Launch Anaconda Navigator
2. Select launch on Spyder



1. Select file from top right of spyder window and select “open”
2. Navigate to app.py in the weather ui folder and open it
3. Run app by selecting green arrow in the top of the spider window

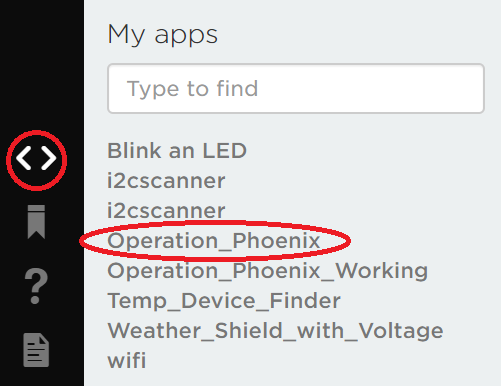


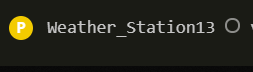


Once you have the Weather Monitor app running to get data you will need to:

1. Select a Device
2. Choose the data range of data that you want to see
3. Select which weather phenomenon you want to see
4. Then hit refresh data to obtain a graph of the data
5. This app downloads the data from google sheets and uses it to create these graphs. When you refresh data, it will take the newest points in the data range from google sheets if it doesn’t already have them.
6. To save the image, select file on the top left of the window and then save image and select the directory you want to save the image to.
7. To improve the resolution of this image, select options at the top left of the window and change the image resolution to a higher dpi

Getting the Data from the Photon

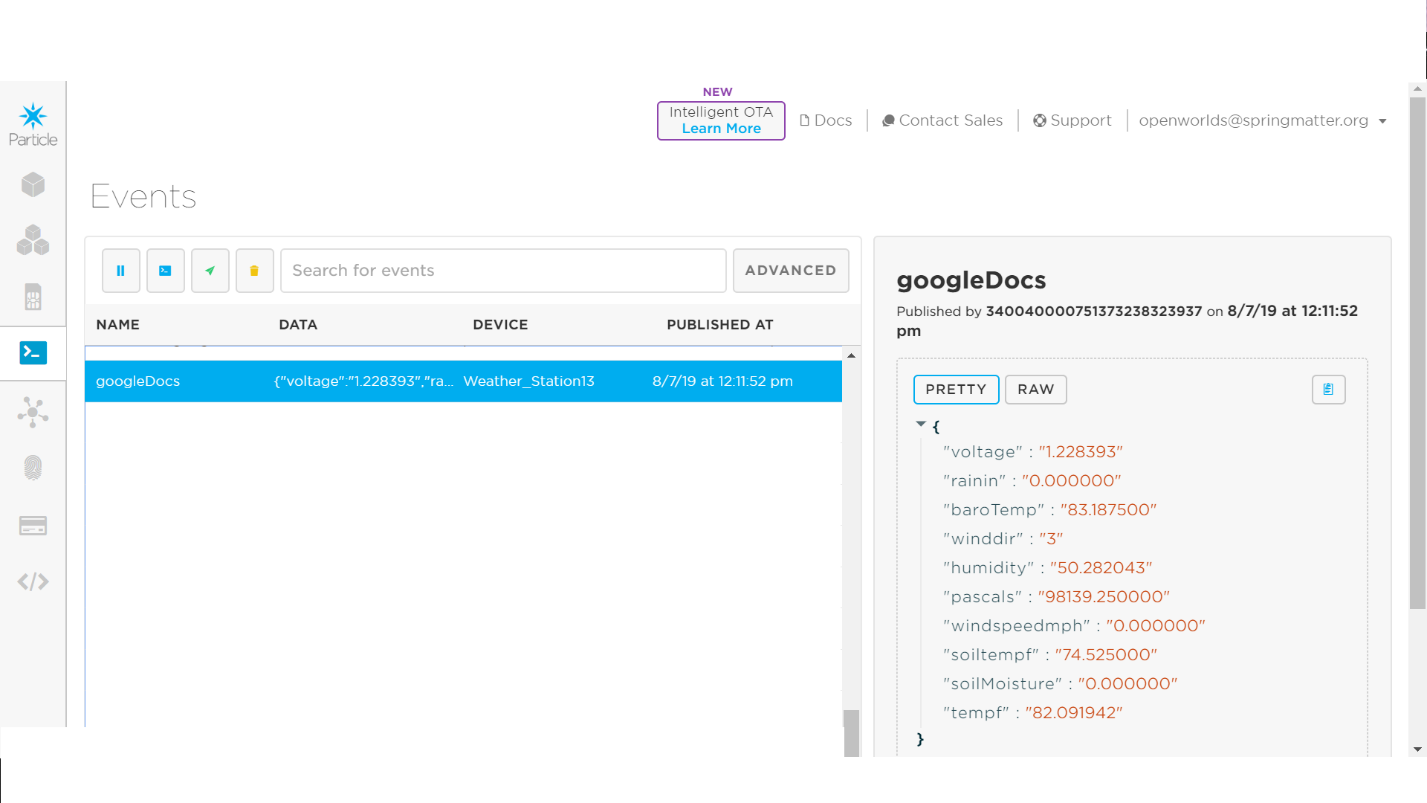
1. Go to particle.io
2. Login with [openworlds@springmatter.org](mailto:openworlds@springmatter.org) and the password Storm1777
3. Now, go to build.particle.io
4. First you will need to flash the Operation\_Phoenix code to your device. To do so, you should click the two arrows on the side of the screen then click on Operation\_Phoenix. 
5. Once you have selected the correct device in the bottom right corner, click on the lightning bolt in the top left corner.

1. Once you have flashed the code to your device, the photon will begin to send data to Particle.io because of the Particle.publish line in the code.

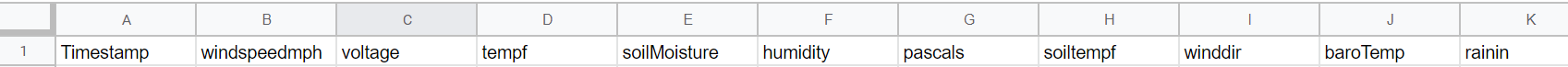


Line 312 is the line you can change to modify the time between readings. If you change it you will need to reflash the code.

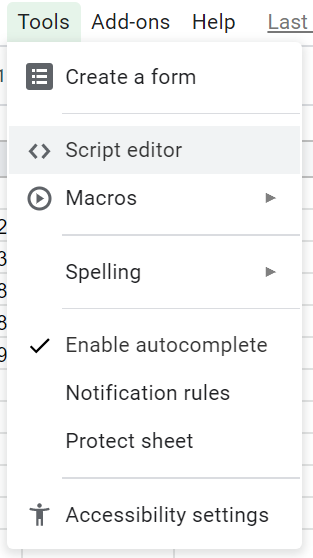
1. You can see data being published in the Particle.io Console

Setting Up Google Sheets

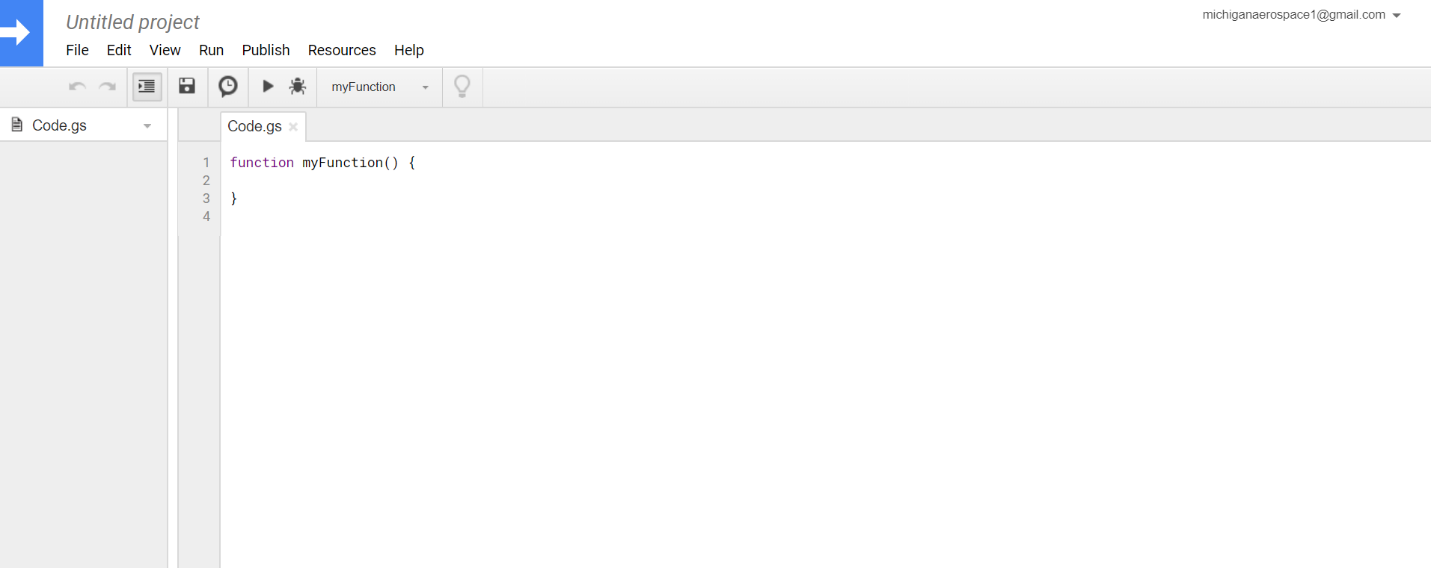
1. Once there is data streaming into Particle.io you can set up a Google sheet for a webhook to send this data to. You must use google chrome to get this to work
2. Login to the email: [michiganaerospace1@gmail.com](mailto:michiganaerospace1@gmail.com) password: michaero
3. Create a new Google Sheet in Google Drive and Name it (Weather Station #)
4. Once you open your google sheet, you need to name the columns as shown below



1. Next you need to open the script editor in Google Sheets. To do this click on the Tools tab at the top, the click on Script editor



1. A Screen like this will pop up. Give the project a name (Weather Station # Project)



1. Delete what is currently under Code.gs and paste this code:

// original from: http://mashe.hawksey.info/2014/07/google-sheets-as-a-database-insert-with-apps-script-using-postget-methods-with-ajax-example/

function doGet(e){

return handleResponse(e);

}

// Usage

// 1. Enter sheet name where data is to be written below

var SHEET\_NAME = "Sheet1";

// 2. Run > setup

//

// 3. Publish > Deploy as web app

// - enter Project Version name and click 'Save New Version'

// - set security level and enable service (most likely execute as 'me' and access 'anyone, even anonymously)

//

// 4. Copy the 'Current web app URL' and post this in your form/script action

//

// 5. Insert column names on your destination sheet matching the parameter names of the data you are passing in (exactly matching case)

var SCRIPT\_PROP = PropertiesService.getScriptProperties(); // new property service

// If you don't want to expose either GET or POST methods you can comment out the appropriate function

function doPost(e){

return handleResponse(e);

}

function handleResponse(e) {

// shortly after my original solution Google announced the LockService[1]

// this prevents concurrent access overwritting data

// [1] http://googleappsdeveloper.blogspot.co.uk/2011/10/concurrency-and-google-apps-script.html

// we want a public lock, one that locks for all invocations

var lock = LockService.getPublicLock();

lock.waitLock(30000); // wait 30 seconds before conceding defeat.

try {

// next set where we write the data - you could write to multiple/alternate destinations

var doc = SpreadsheetApp.openById(SCRIPT\_PROP.getProperty("key"));

var sheet = doc.getSheetByName(SHEET\_NAME);

// we'll assume header is in row 1 but you can override with header\_row in GET/POST data

var headRow = e.parameter.header\_row || 1;

var headers = sheet.getRange(1, 1, 1, sheet.getLastColumn()).getValues()[0];

var nextRow = sheet.getLastRow()+1; // get next row

var row = [];

// loop through the header columns

for (i in headers){

if (headers[i] == "Timestamp"){ // special case if you include a 'Timestamp' column

row.push(new Date());

} else { // else use header name to get data

row.push(e.parameter[headers[i]]);

}

}

// more efficient to set values as [][] array than individually

sheet.getRange(nextRow, 1, 1, row.length).setValues([row]);

// return json success results

return ContentService

.createTextOutput(JSON.stringify({"result":"success", "row": nextRow}))

.setMimeType(ContentService.MimeType.JSON);

} catch(e){

// if error return this

return ContentService

.createTextOutput(JSON.stringify({"result":"error", "error": e}))

.setMimeType(ContentService.MimeType.JSON);

} finally { //release lock

lock.releaseLock();

}

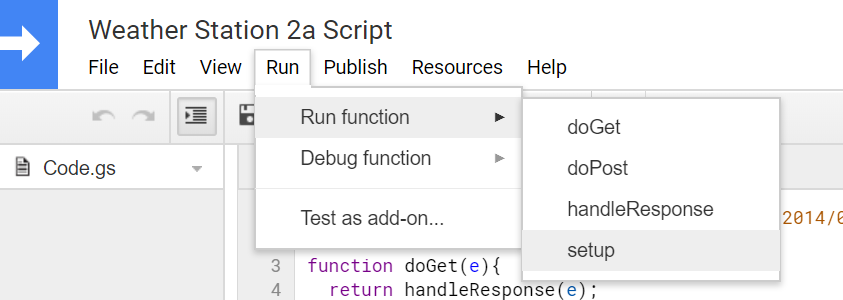
}

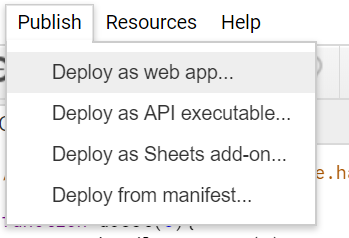
function setup() {

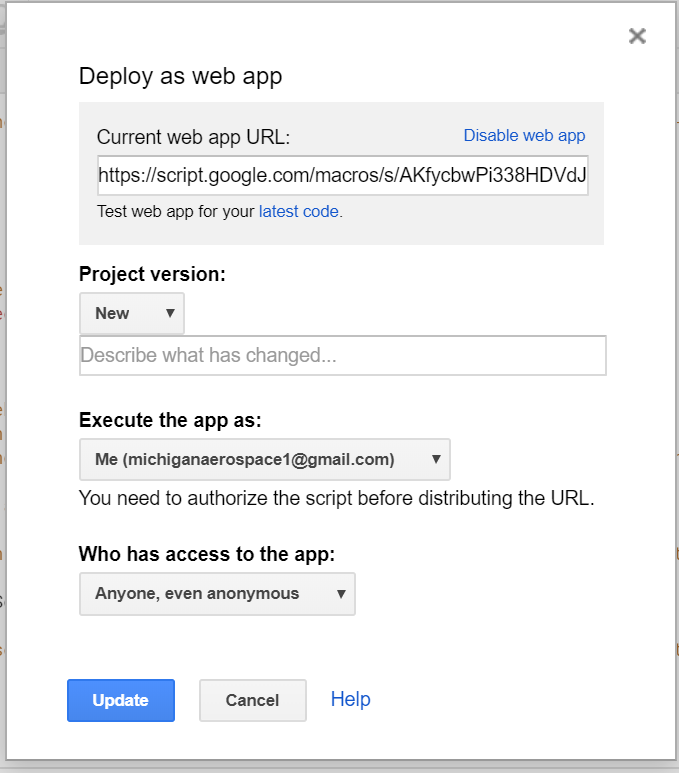
var doc = SpreadsheetApp.getActiveSpreadsheet();

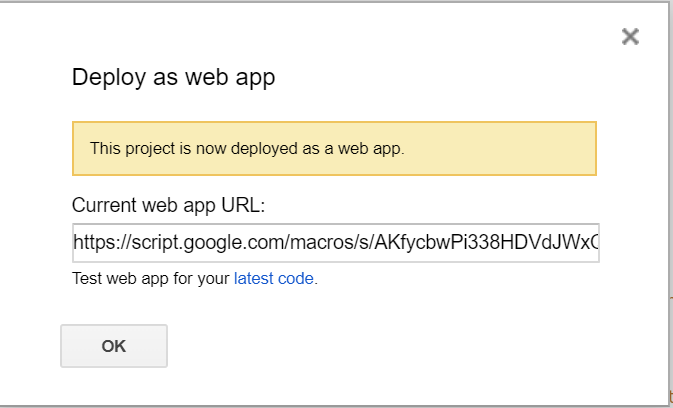
SCRIPT\_PROP.setProperty("key", doc.getId());

}

1. Save this code then Run setup. 
2. A screen will come up about permissions with the document. You will need to hit advanced settings then allow.
3. Next, click on the Publish tab and the Deploy as a Web App



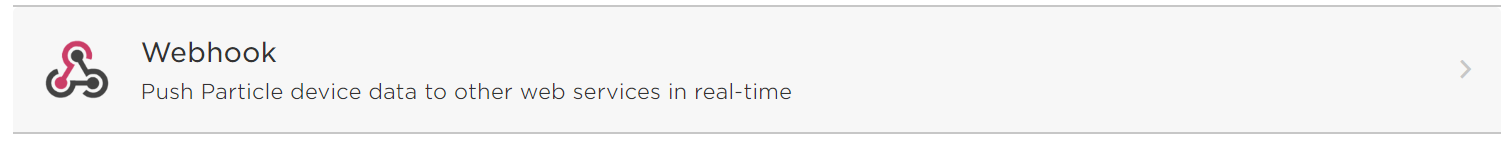
1. That will bring up this screen: 
2. Set the Project version to new, the execute app as to Me and the access to anyone even anonymous. Click Deploy.
3. That will bring up this screen:

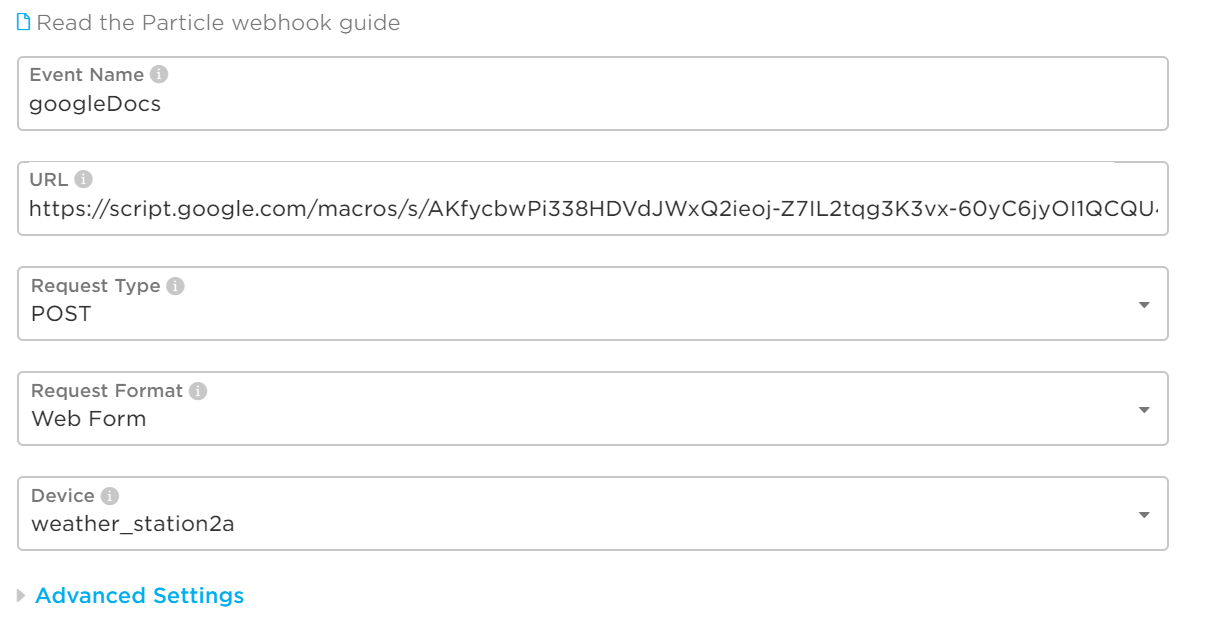
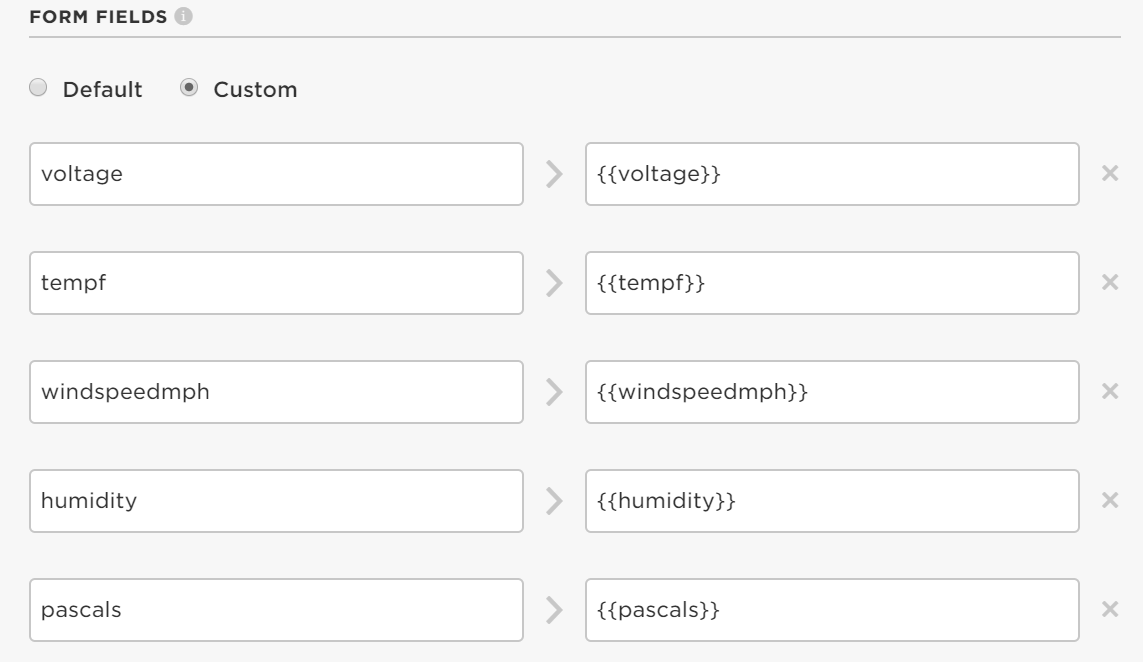
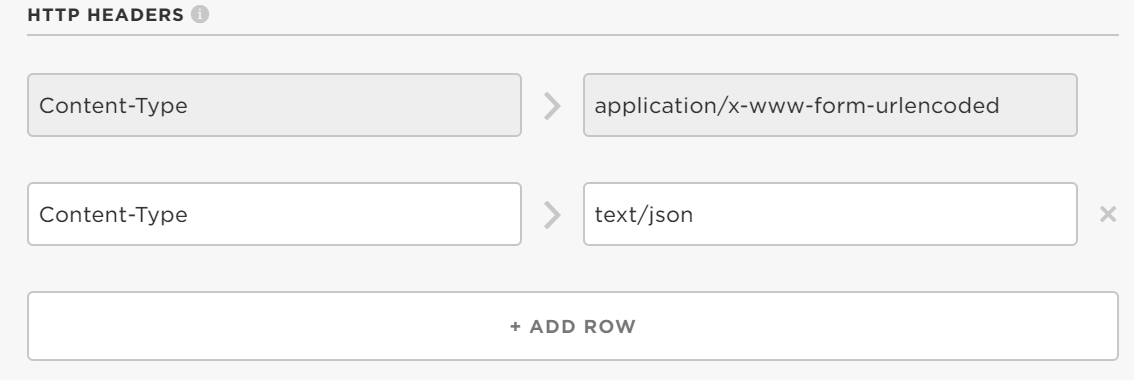
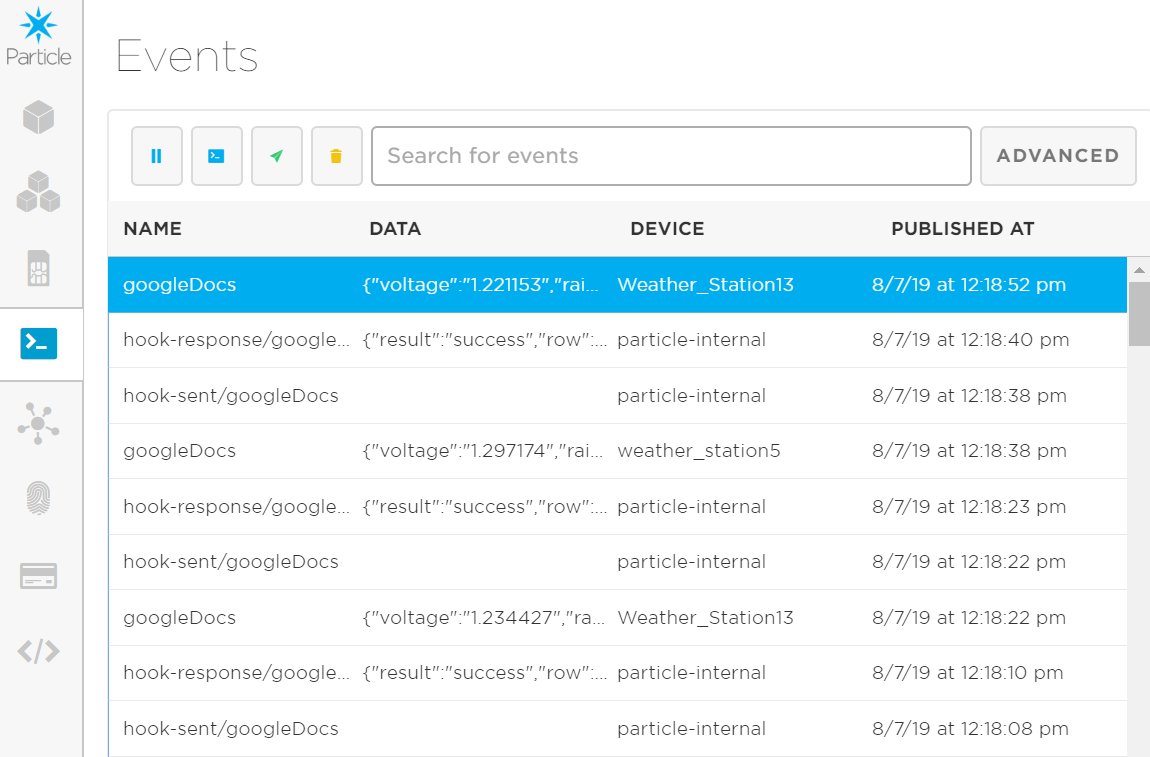
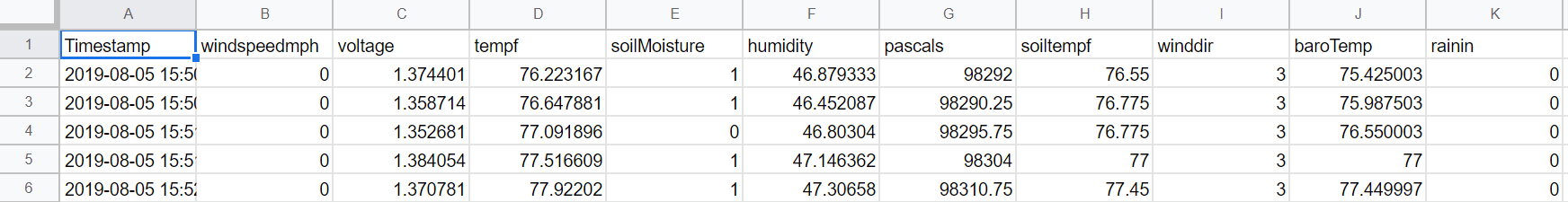


1. Copy the Current web app URL to be used in a Particle Webhook and click OK

Setting Up a Particle Webhook

1. Open the Particle.io console and click on integrations 
2. Click on new integration 
3. Click on Webhook



1. Enter the event name googleDocs. Paste the URL you copied from the google script. Select your device.
2. Click on Advanced Settings then custom for form fields. Enter each field as seen below
3. Under Http Headers enter as seen below 
4. At the bottom of the page click create webhook. If you are editing your webhook, you will need to hit save. 
5. You should now be able to go to the events tab on the Particle.io console and see hooks being sent to Google Sheets
6. Data should now be streaming in your Google Sheet

Adding new weather station to weather app

1. Navigate to the client secret file located in the app\_data or main folder
2. Right click and select edit
3. Select and copy the client\_email which is “weather@weather-247217.iam.gserviceaccount.com”
4. Return to the google sheet and share this with our client\_email so it can access the sheet.
5. Run the weather app if not familiar check the section on running the app
6. In the top left section of weather app select option and then Add Device
7. A pop-up window should ask you the number of your weather station
8. Enter the weather station number from your google sheets example (Weather Station 5) just enter “5”