Adafruit IO

What is Adafruit IO?

[](https://learn.adafruit.com/assets/54701)

Adafruit.io is a *cloud service*- that just means we run it for you and you don't have to manage it. You can connect to it over the Internet. It's meant primarily for storing and then retrieving data but it can do a lot more than just that!

What can Adafruit IO do for me?

* Display your data in real-time, online
* Make your project internet-connected: Control motors, read sensor data, and more!
* Connect projects to web services like Twitter, RSS feeds, weather services, etc.
* Connect your project to other internet-enabled devices
* The best part? All of the above is do-able for **free**with Adafruit IO

Dashboards

Adafruit.io can handle and visualize multiple feeds of data. Want to display data from a [temperature-humidity sensor](https://www.adafruit.com/product/385) alongside data from an [air quality sensor](https://www.adafruit.com/product/3566) and add a button to turn on the air-conditioner in your room?

No problem! [Dashboards](https://learn.adafruit.com/adafruit-io-basics-dashboards) are a feature integrated into Adafruit IO which allow you to chart, graph, gauge, log, and display your data. You can view your dashboards from anywhere in the world.

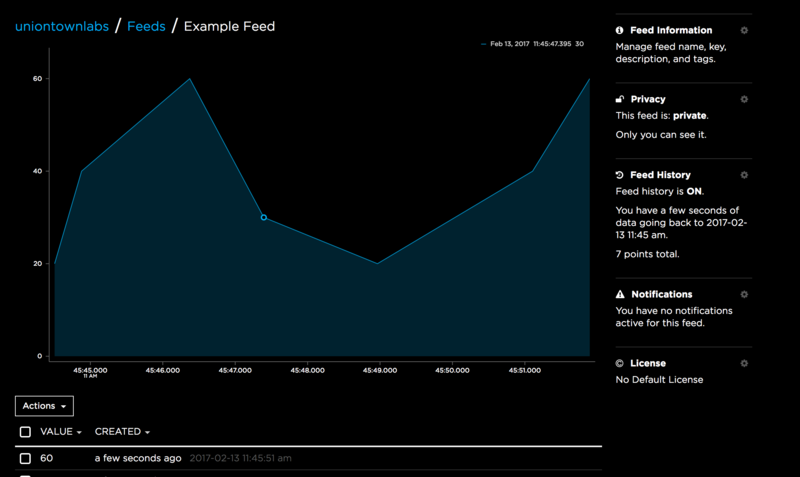
Triggers

Use triggers in Adafruit IO to control and react to your data. Configure triggers to email you when your system goes offline, react to a temperature sensor getting too hot, and publish a message to a new feed.

<https://learn.adafruit.com/welcome-to-adafruit-io/what-is-adafruit-io>

# FEEDS

## Overview

[](https://learn.adafruit.com/assets/39229)

Feeds are the core of the Adafruit IO system. The feed holds **metadata** about the data you push to Adafruit IO. This includes settings for whether the data is public or private, what license the stored sensor data falls under, and a general description of the data. The feed also contains the sensor **data**values that get pushed to Adafruit IO from your device.

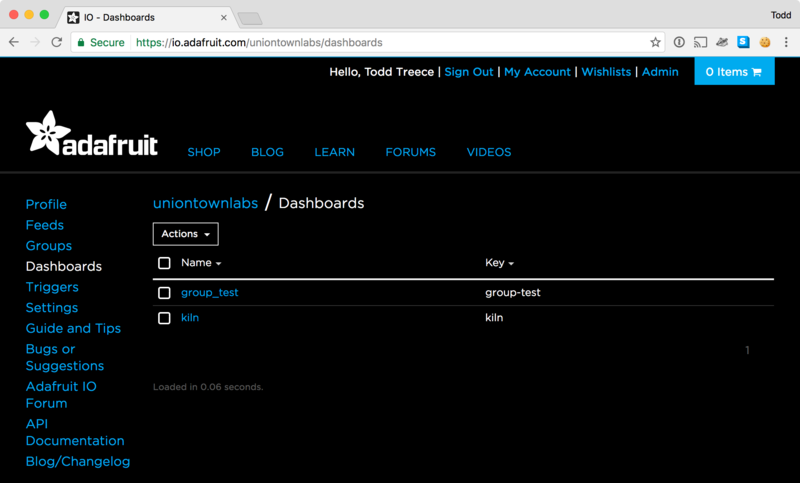
You will need to create one feed for each unique source of data you send to the system. For example, if you have a project with one temperature sensor and two humidity sensors, you would need to create three feeds. One feed for the temperature sensor, and one feed for each humidity sensor.

First, let's take a look at how to create a new feed using Adafruit IO.

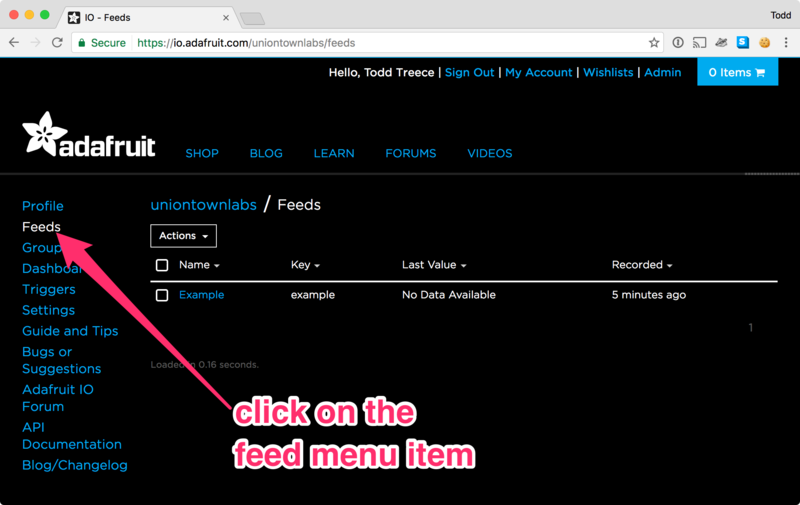
## Creating a Feed

[Like](https://learn.adafruit.com/guides/1001/favorites/55276.js)

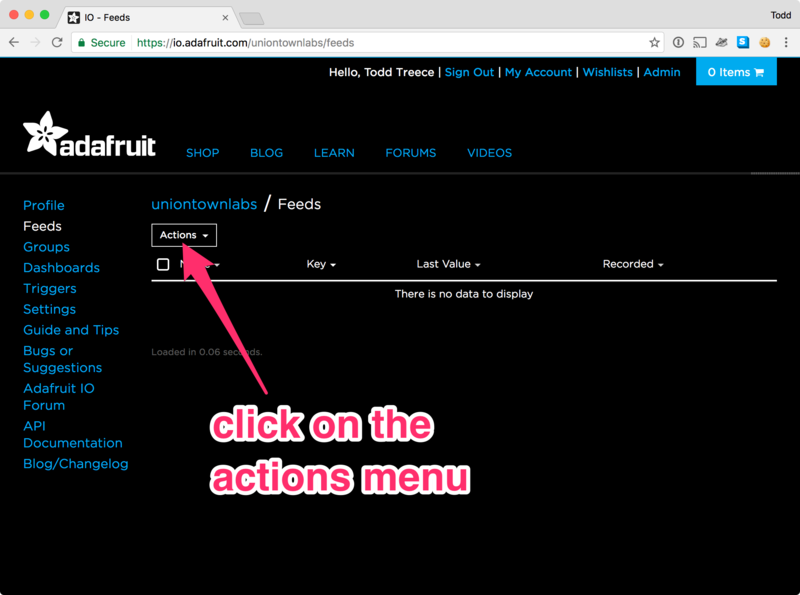
Creating a Feed on Adafruit IO is a very simple process. When you login or register to your [io.adafruit.com](https://io.adafruit.com/) account, you will see a list of your current dashboards like the list shown below.

[](https://learn.adafruit.com/assets/39199)

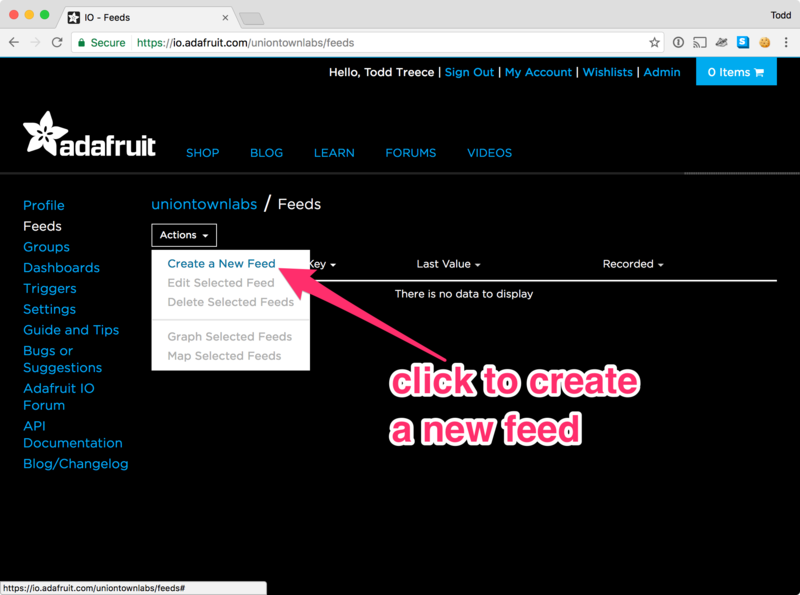
Click the**Feeds** link on the left hand side of the screen to navigate to the Feeds list.

[](https://learn.adafruit.com/assets/39200)

Next, click on the **Actions** menu on the left hand side of the screen.

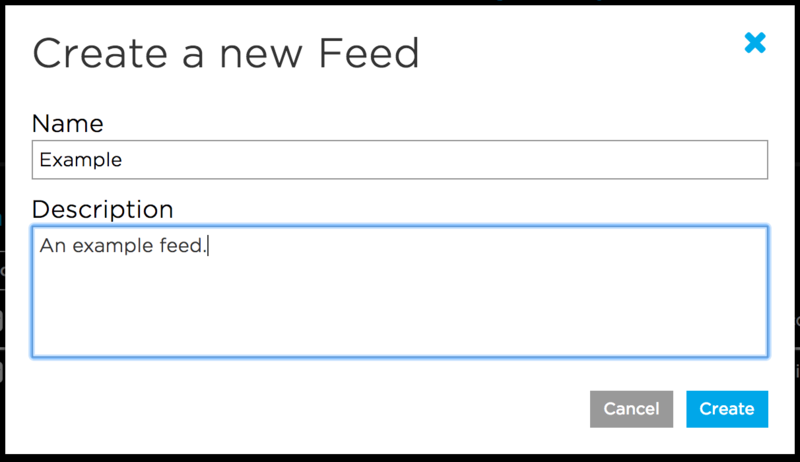
[](https://learn.adafruit.com/assets/39194)

Next, click on the **Create a New Feed** menu item.

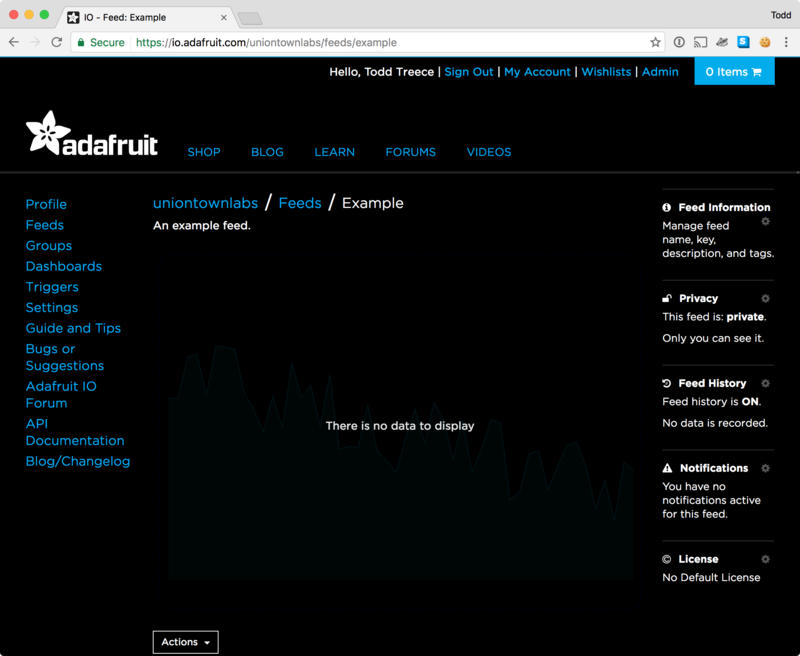
[](https://learn.adafruit.com/assets/39195)

A form modal will launch, and you will be presented with two text inputs:

* **Name** - A short descriptive title of your data. Letters, numbers, and spaces are valid characters, and this field is *required*. The feed name must be *unique* for your account.
* **Description** - A long form description of your data. This field is not required, but it's useful to provide a detailed description if your feed will be public.

[](https://learn.adafruit.com/assets/39201)

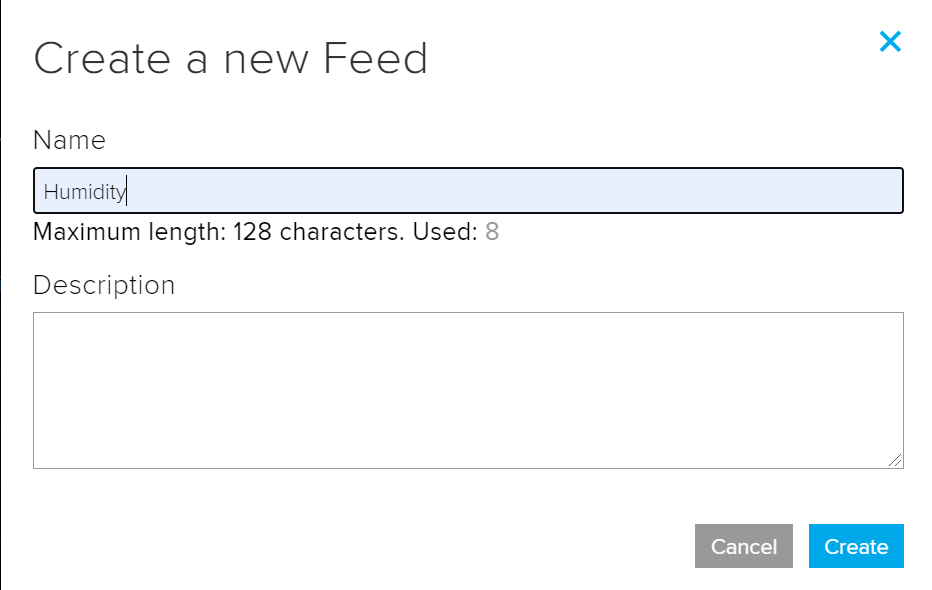
Click the Create Feed button once you have finished entering your feed's name and description. You will then be redirected to your new feed.

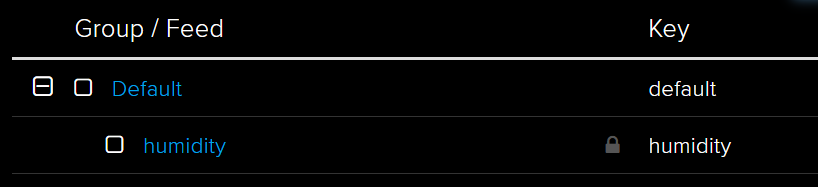
[](https://learn.adafruit.com/assets/39196)

<https://learn.adafruit.com/adafruit-io-basics-feeds/creating-a-feed>

# Humidity Feed

Let us create a humidity Feed and upload data to the feed.





The Feed Key will be used later to upload data to the feed

Let us write a Python code to get humidity values from the Arduino upload data to the feed.

First install the Adafruit IO library for python:

pip3 install adafruit-io

After loging in to Adafruit IO click on “Adafruit IO Key”. The “Username” and “Active Key” will be used to upload data to the humidity feed.

Code

from Adafruit\_IO import Client, RequestError, Feed #adafruit libraries

import threading #Threading is explained later

import serial

import time

ADAFRUIT\_IO\_KEY = 'Insert\_Key\_here'

ADAFRUIT\_IO\_USERNAME = 'Insert\_Username\_here'

#create an instance of the Adafruit IO Client

aio = Client(ADAFRUIT\_IO\_USERNAME, ADAFRUIT\_IO\_KEY)

#Enter the feed key

humidity\_feed = aio.feeds('humidity')

ser = serial.Serial('COM10',baudrate = 9600, timeout=3)

#ser = serial.Serial('/dev/rfcomm0',baudrate = 9600, timeout=3) #For Raspberry Pi

time.sleep(3)

def startLogger():

ser.write(b'h')

humidityval = ser.readline().decode('utf-8')

if len(humidityval) > 0:

if humidityval[0] == 'h':

humidityval = int(humidityval[1:])

print("Humidity sensor val: ",humidityval)

#Send data to the humidity feed in AdafruitIO

aio.send\_data(humidity\_feed.key, humidityval)

#call the thread every 300 seconds

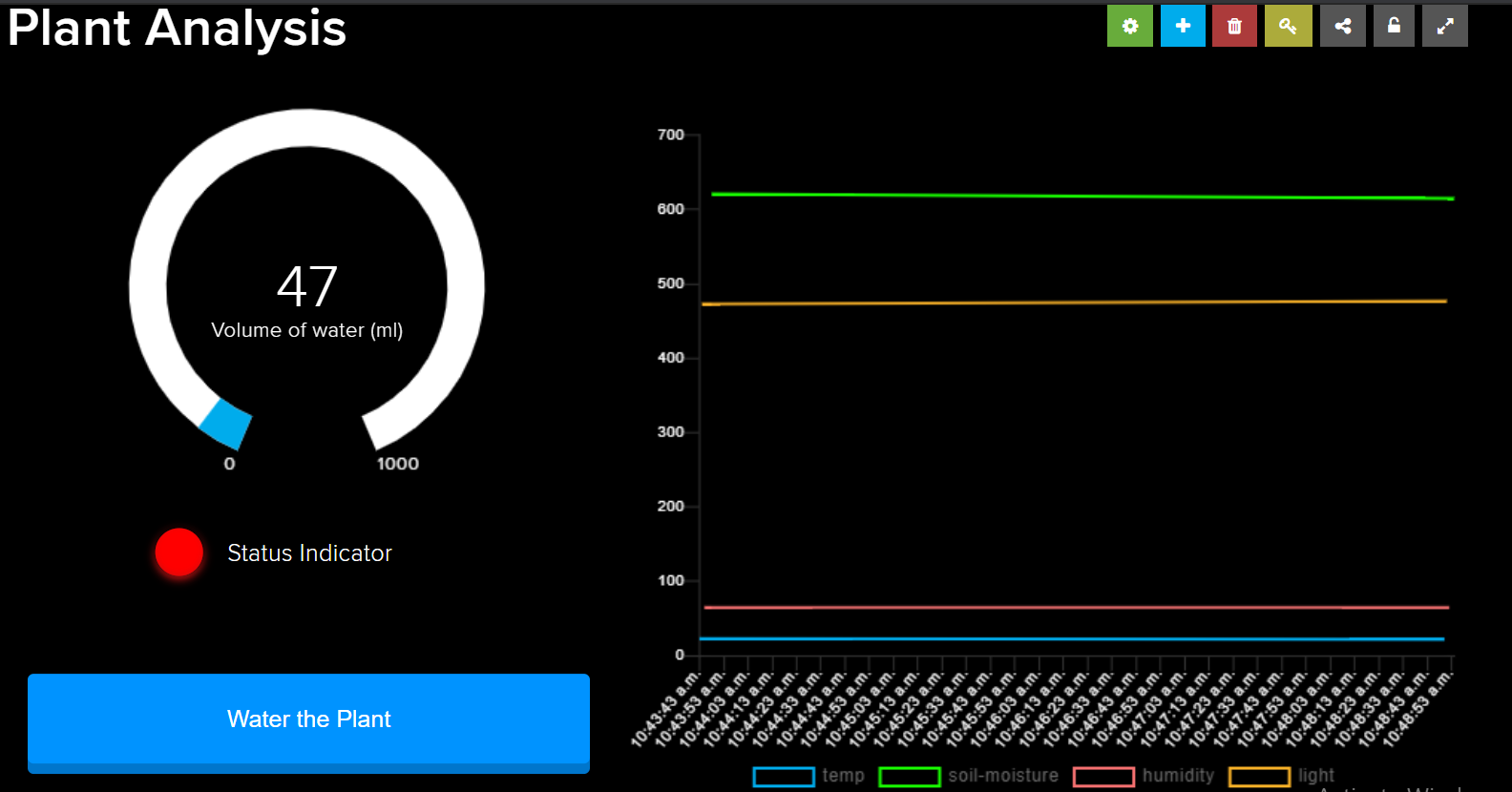
threading.Timer(300,startLogger).start()

startLogger() #start the thread

In the above program we have used threading, A thread is a separate flow of execution. This means that your program will have two things happening at once. We can have a thread run repeatedly after a specific duration. We have used a thread called “startlogger” to receive sensor data from the Arduino and post it Adafruit IO. In the above code the thread runs every 300 seconds(5 minutes).

# Dashboard

# Overview



Dashboards allow you to visualize data and control Adafruit IO connected projects from any modern web browser. Widgets such as charts, sliders, and buttons are available to help you quickly get your IoT project up and running without the need for any custom code.

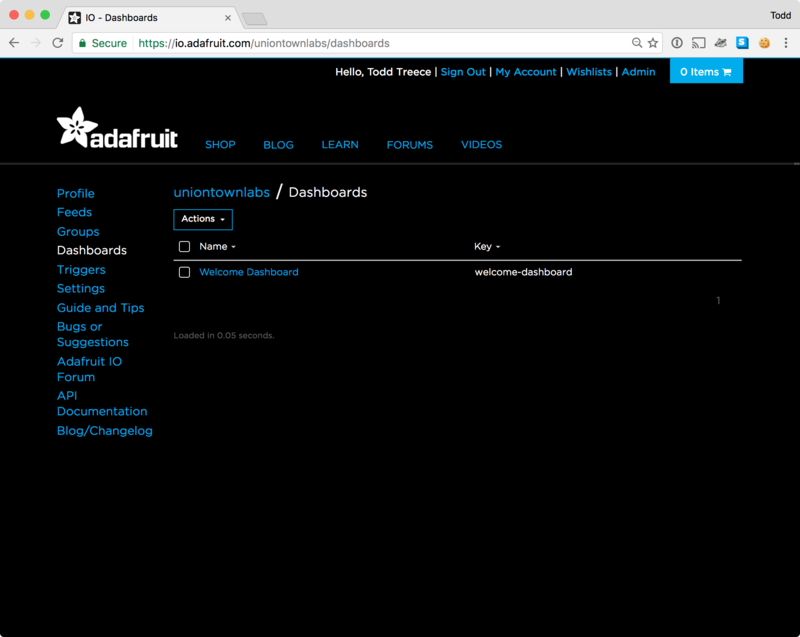
If you are new to Adafruit IO, you may want to start with the [Adafruit IO Feeds guide](https://learn.adafruit.com/adafruit-io-basics-feeds) before you continue with this guide. If you are comfortable with feeds, then you are ready to create your first dashboard.

<https://learn.adafruit.com/adafruit-io-basics-dashboards>

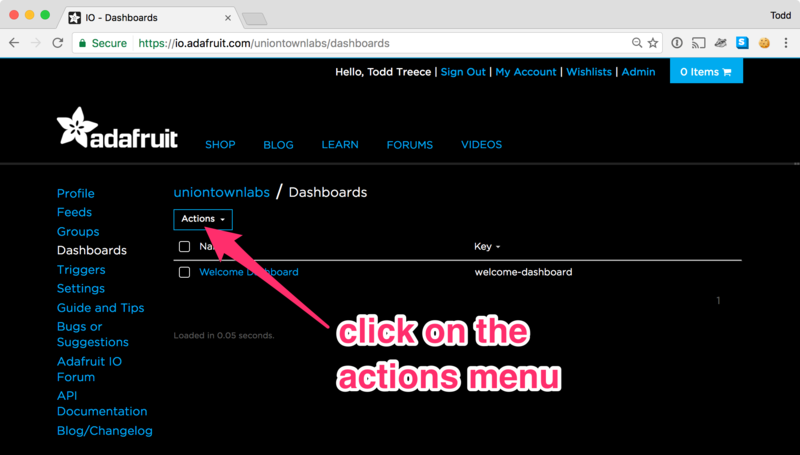
# Creating a Dashboard

[Like](https://learn.adafruit.com/guides/1002/favorites.js)

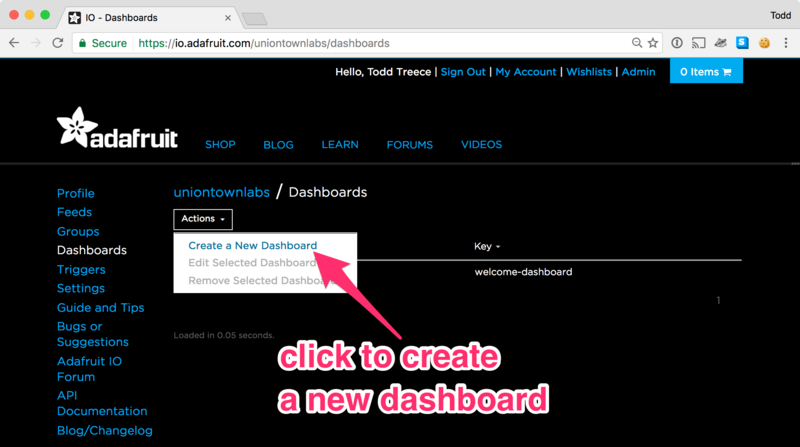
When you login to your [io.adafruit.com](https://io.adafruit.com/) account, you will be redirected to your list of dashboards. It will look like the page seen below.

[](https://learn.adafruit.com/assets/39234)

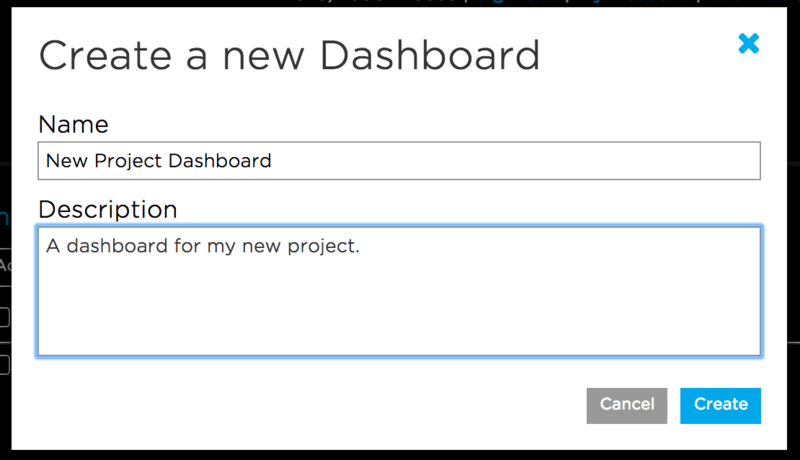
Your list of dashboards will only have the Welcome Dashboard when it is first loaded. You can start the dashboard creation process by clicking the **Actions** menu on the upper left hand side of the screen.

[](https://learn.adafruit.com/assets/39235)

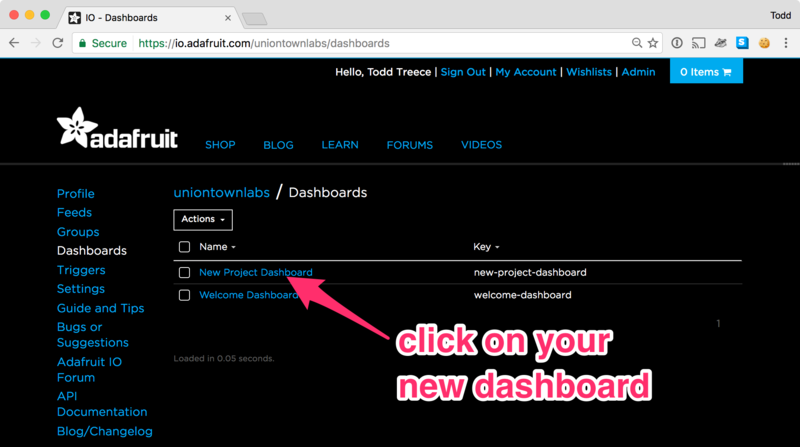
Next, select **Create a New Dashboard** from the dropdown menu.

[](https://learn.adafruit.com/assets/39236)

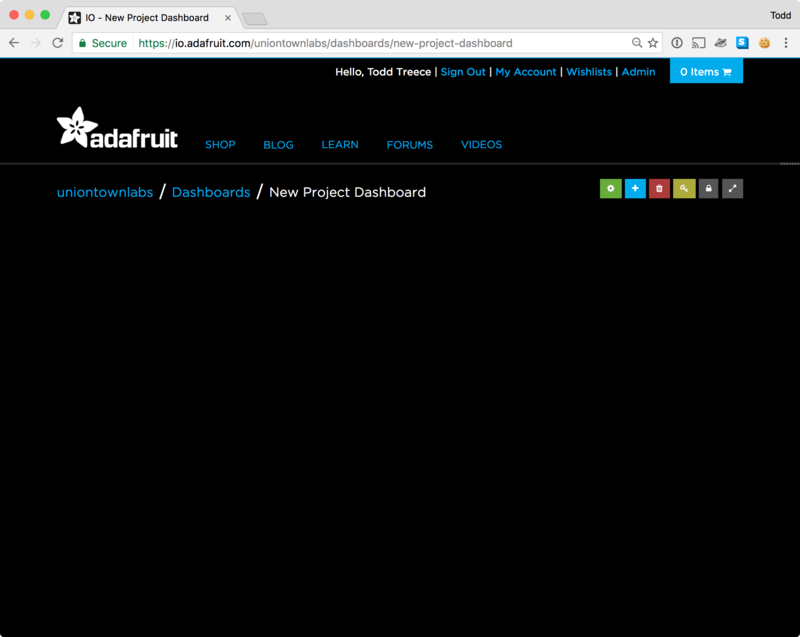
You can then enter the name and description of your new dashboard, and click the **Create** button once you are finished.

[](https://learn.adafruit.com/assets/39237)

Once your dashboard has been created, click on the name of your new dashboard to load it.

[](https://learn.adafruit.com/assets/39238)

You should now see your new blank dashboard.

[](https://learn.adafruit.com/assets/39239)

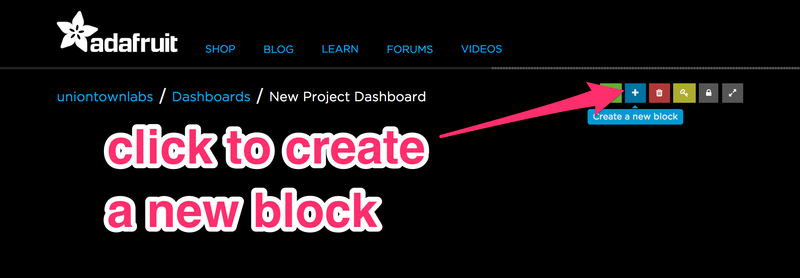
https://learn.adafruit.com/adafruit-io-basics-dashboards/creating-a-dashboard

Let us create a dashboard for our poject.

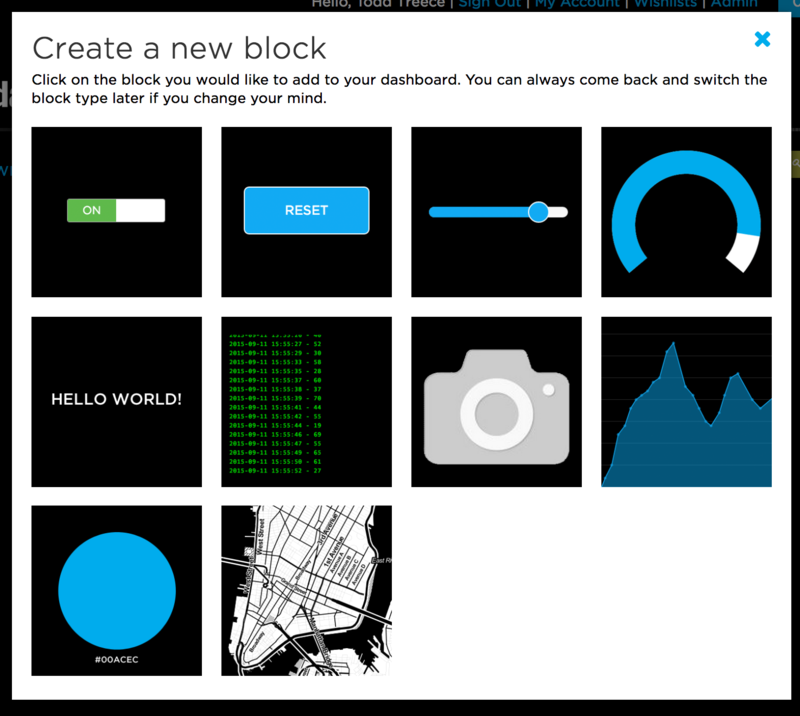
# 

# Adding Blocks

Blocks are widgets that you can add to your dashboard. There are some blocks that can be used as outputs, and some that can be used as inputs. To add a new block, you can click the **+**(plus) button on the upper right hand side of the dashboard.

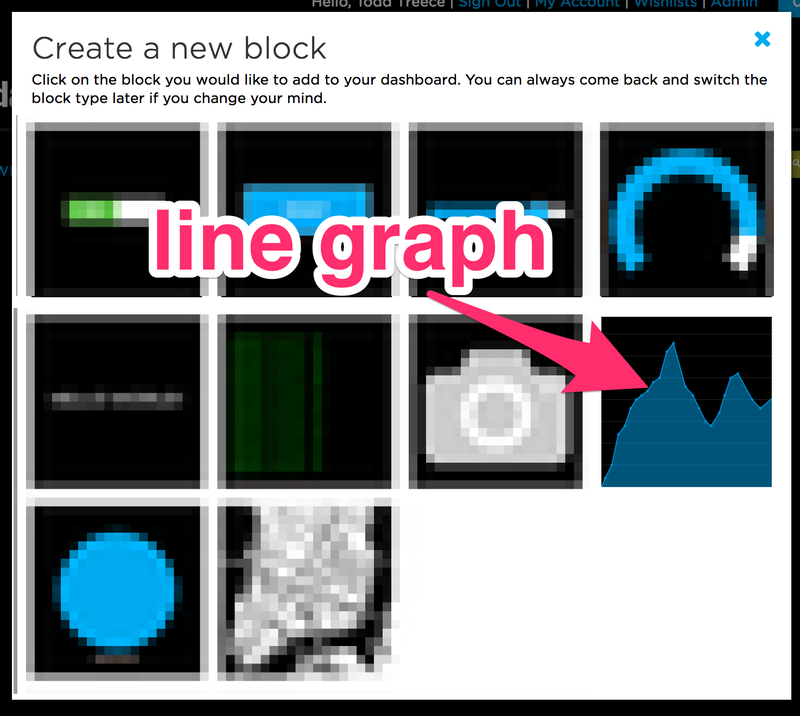
[](https://learn.adafruit.com/assets/39240)

You will then be presented with a list of block types to choose from, like the one seen below.

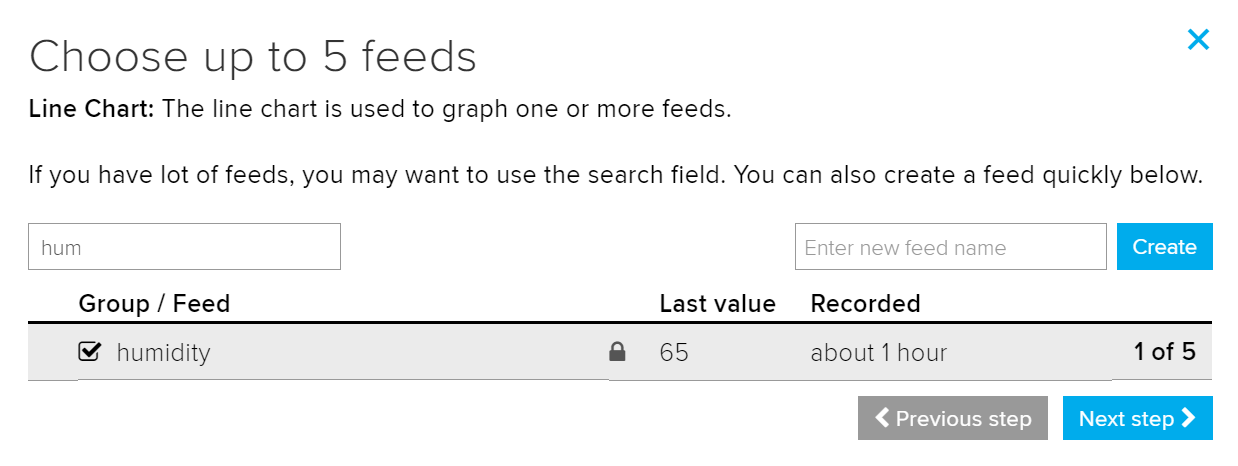
[](https://learn.adafruit.com/assets/39246)

For a detailed explanation regarding each block go to: <https://learn.adafruit.com/adafruit-io-basics-dashboards/adding-blocks>

Let us add a line chart to display the humidity value.



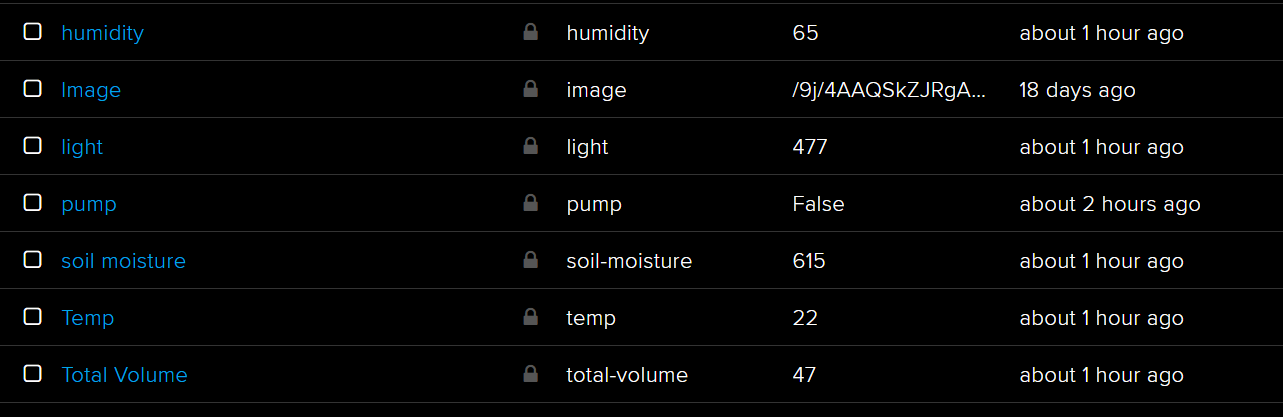
Upto 5 feeds can be displayed on the line chart, lets select “humidity”

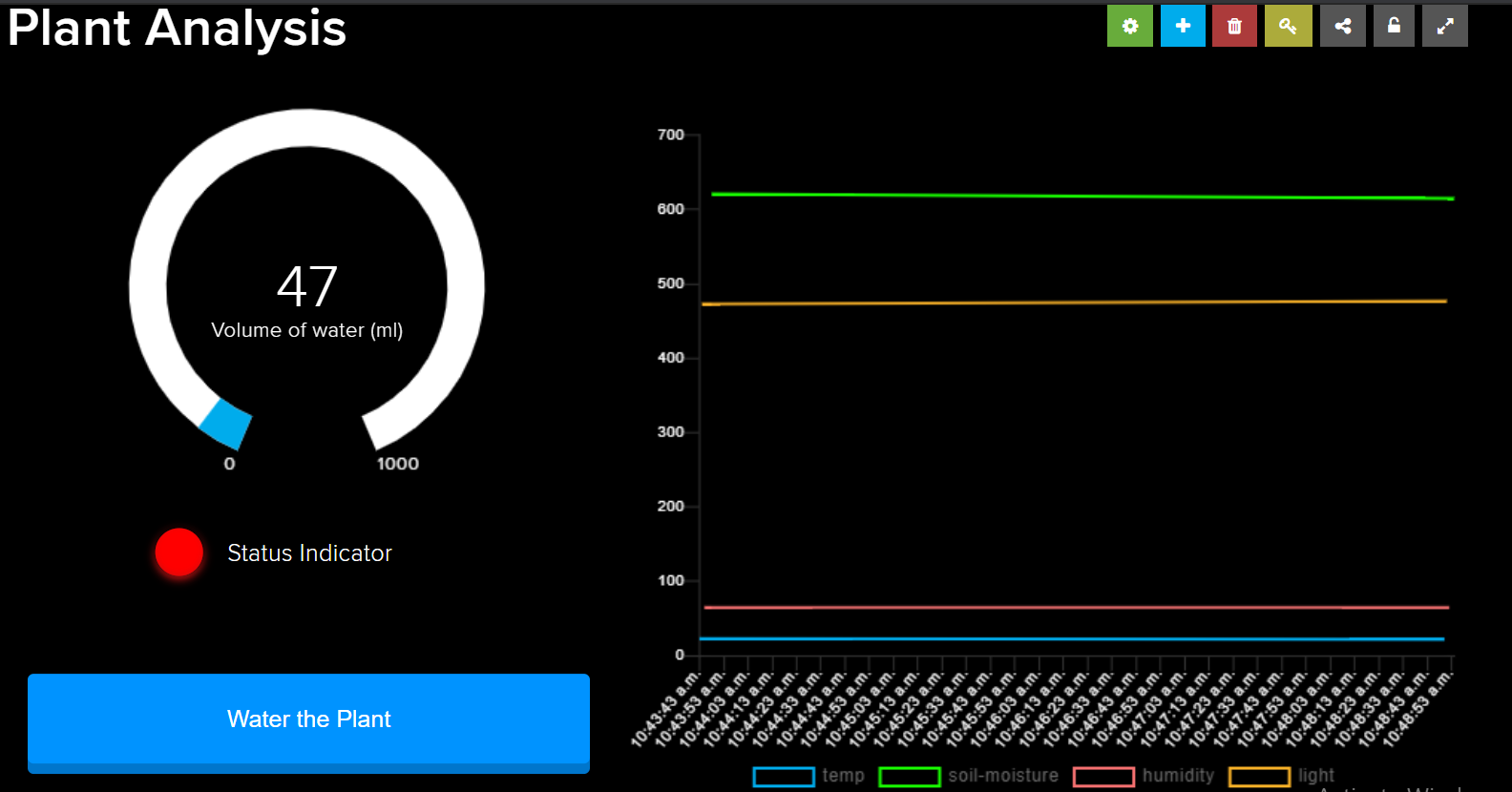


The block settings allows you to customize the line chart, create the block.

If we run the python code we can see humidity values plotted on the line chart.

Similarly we can create multiple feeds and multiple blocks to display all the sensor data.





# Triggers

## What are Adafruit IO Triggers?

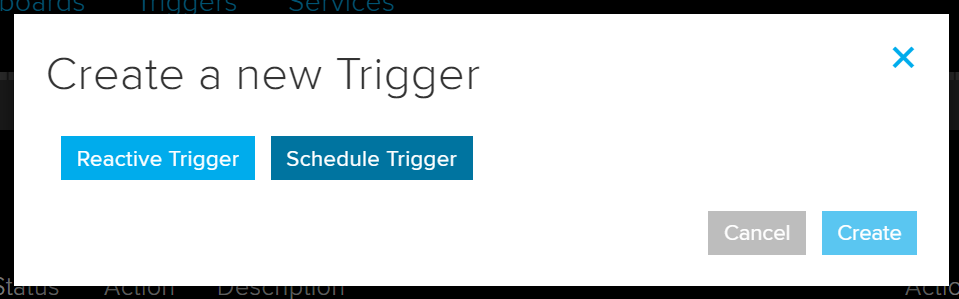
Adafruit IO **Triggers add some lightweight logic to your IoT project without writing extra code**. Triggers are a way to do something when a certain situation occurs. This guide focuses on the simplest trigger type - **scheduled triggers**.

<https://learn.adafruit.com/adafruit-io-basics-scheduled-triggers>

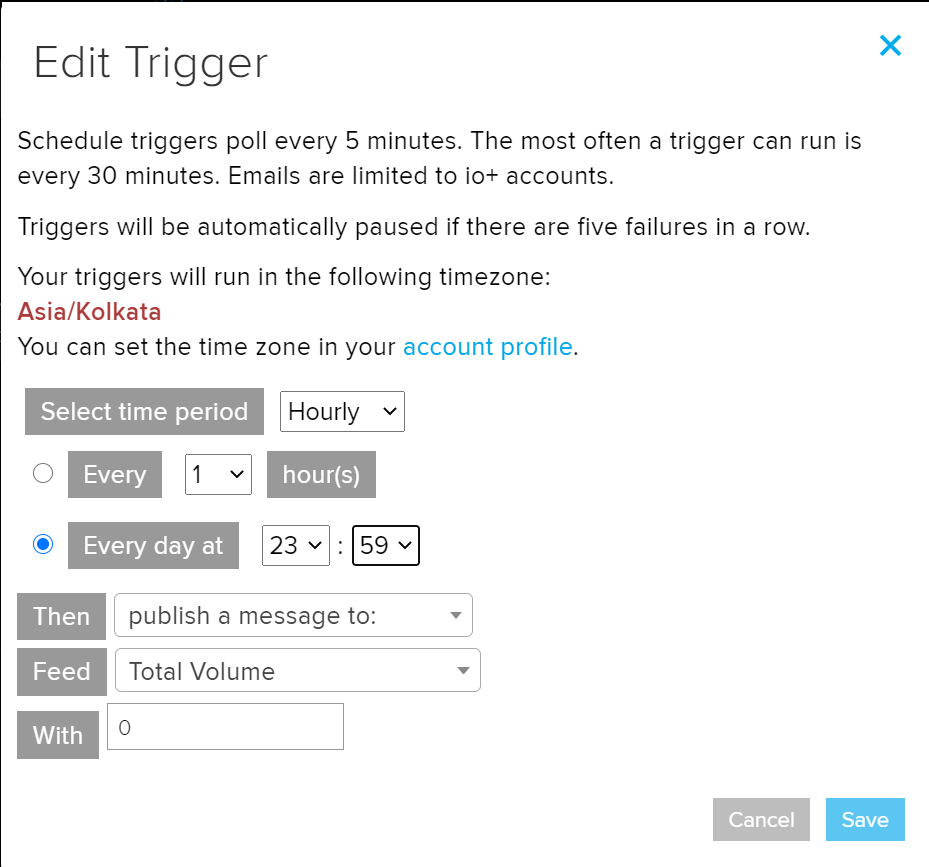
For example we automatically water the plant if the soil moisture sensors value goes above a threshold. This can be created as reactive trigger.

Let us create a Trigger to automatically reset the Volume of water poured every day, so we can see the amount of water the plant received each day.

Go to the Trigger section and create a schedule Trigger



We can set the total volume feed to 0 at 23:59



Further Tutorials:

<https://learn.adafruit.com/series/adafruit-io-basics>

<https://learn.adafruit.com/all-the-internet-of-things-episode-four-adafruit-io/introduction>