# HomePlantGuardian

Figure 1: Network Topology and Program Flowchart

**Project Link:** [**https://github.com/SPWwj/HomePlantGuardian**](https://github.com/SPWwj/HomePlantGuardian%09)

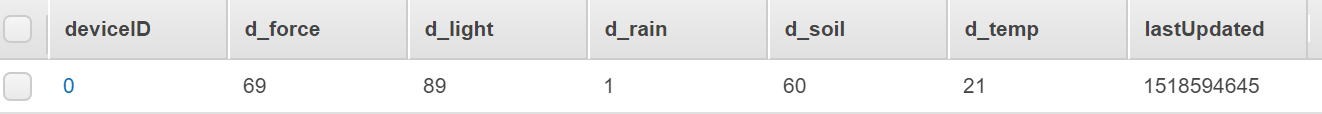
**Setting Up**

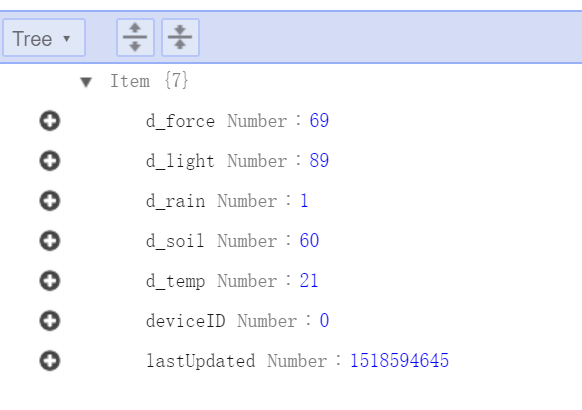
To build this project your will need an Amazon Web Services account and all the component shown on the Figure 1. (You can ignore GY85 sensor if you not intent to detect movement).

For setting up AWS IOT please refer to: <https://github.com/SPWwj/AWS-IOT-Landslide-Proj>. the IOT structure for both project is very similar. The only different is the DynamoDB and the lambda function. The difference will be cover on this Tutorial.

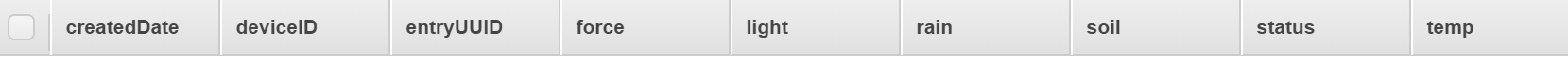
**AWS**

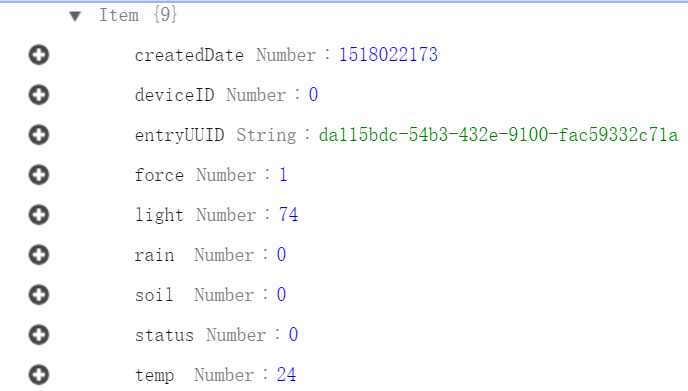
1.You have to create 2 table AWSThing and AWSThingDATA

AWSThing 



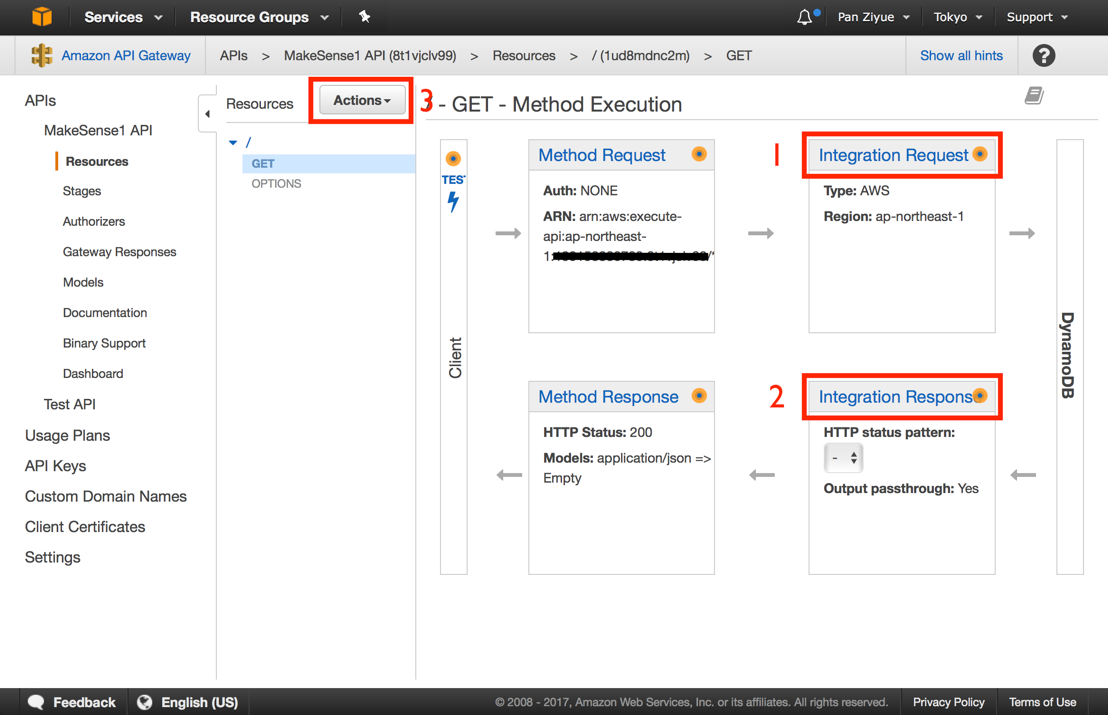
AWSThingDATA





Take note of data type

You must change the Integration Response code base on the format.



#set($inputRoot = $input.path('$'))

{

#foreach($elem in $inputRoot.Items)

"ID$elem.deviceID.N":

[$elem.lastUpdated.N, $elem.d\_force.N, $elem.d\_light.N, $elem.d\_rain.N, $elem.d\_soil.N, $elem.d\_temp.N]#if($foreach.hasNext),#end

#end

}

2.Do not forget to change the table name in integration Request to

{ "TableName": "AWSThing" }

3.Lastly The Lambda Source Code can be found in [**HomePlantGuardian**](https://github.com/SPWwj/HomePlantGuardian)/[NewProj](https://github.com/SPWwj/HomePlantGuardian/tree/master/NewProj)/**AWS Code**/

Serial Communication between ESP8266 and PIC18F4550

The source code for PIC18F4550 can be found in [**HomePlantGuardian**](https://github.com/SPWwj/HomePlantGuardian)/[NewProj](https://github.com/SPWwj/HomePlantGuardian/tree/master/NewProj)/**PIC\_ESP8266\_WiFi.X**/

This program waits for ‘.’ Command form ESP8266 and then send the data to ESP8266 through serial port.

The ESP8266 Source code can be found in [**HomePlantGuardian**](https://github.com/SPWwj/HomePlantGuardian)/[NewProj](https://github.com/SPWwj/HomePlantGuardian/tree/master/NewProj)/**Arduino**/

Wire Connection

PIC18F4550

Input:

RA0-Temperature Sensor

RA1-Soil Moisture Sensor

RA2-Force Sensor

RA3-Rain Sensor

RA4-LDR

Output:

RD0-Fan

RD1-Water Pump

RD2-Buzz

RD3-Water tank Cover

RD4-Light

Serial:  
RC7-RX

RC6-TX

ESP8266

TX-RC7

RX-RC6

VCC-3.3V

GND-GND

D3 Web interface

The D3 Web interface is using D3 Liquid Fill Gauge from Curtis Bratton’s Block 5e5ce9beee483220e2f6. Link: <http://bl.ocks.org/brattonc/5e5ce9beee483220e2f6>

The source code can be found in：

[**HomePlantGuardian**](https://github.com/SPWwj/HomePlantGuardian)/[NewProj](https://github.com/SPWwj/HomePlantGuardian/tree/master/NewProj)/[D3](https://github.com/SPWwj/HomePlantGuardian/tree/master/NewProj/D3)/**DestProject**/

The main file is [AWSDATA.html](https://github.com/SPWwj/HomePlantGuardian/blob/master/NewProj/D3/DestProject/AWSDATA.html)

Suggestion

18F4550 may face overloading. Currently delay is used to prevent this problem. However, Timeout is more effective in handling the overloading issue but not yet implement.

The source code folder includes an android application which allow user to subscribe and publish to the AWS IOT.

The AWSForm is a window application that allow user to control the device using window application.