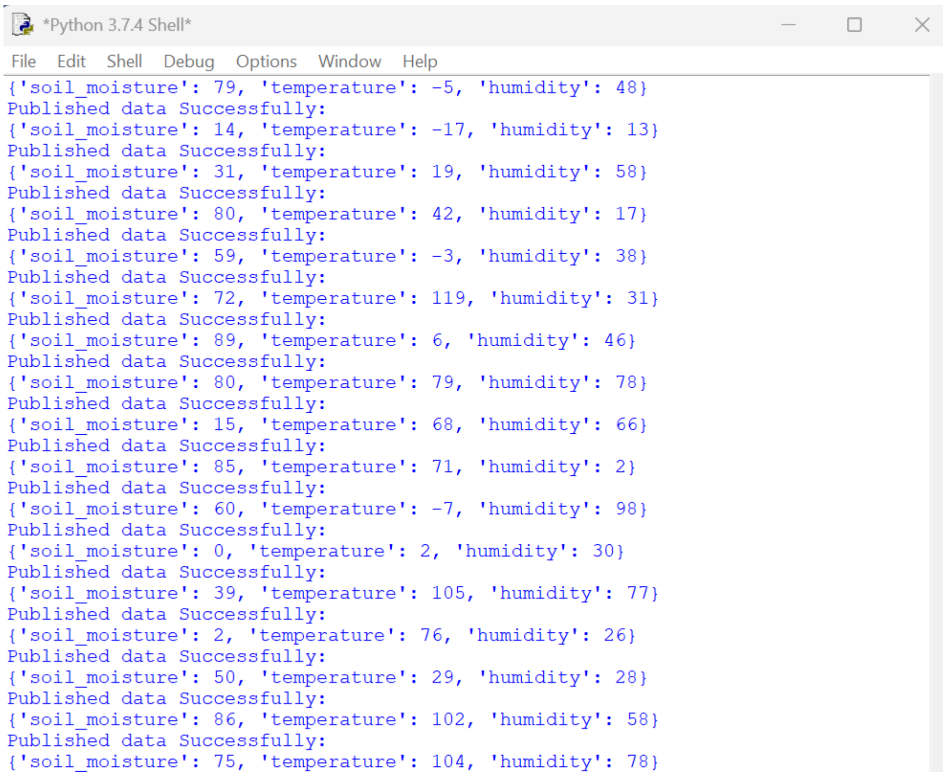
Smart Farmer - IoT Enabled Smart Farming Application

SPRINT DELIVERY-2

Team ID: PNT2022TMID43384

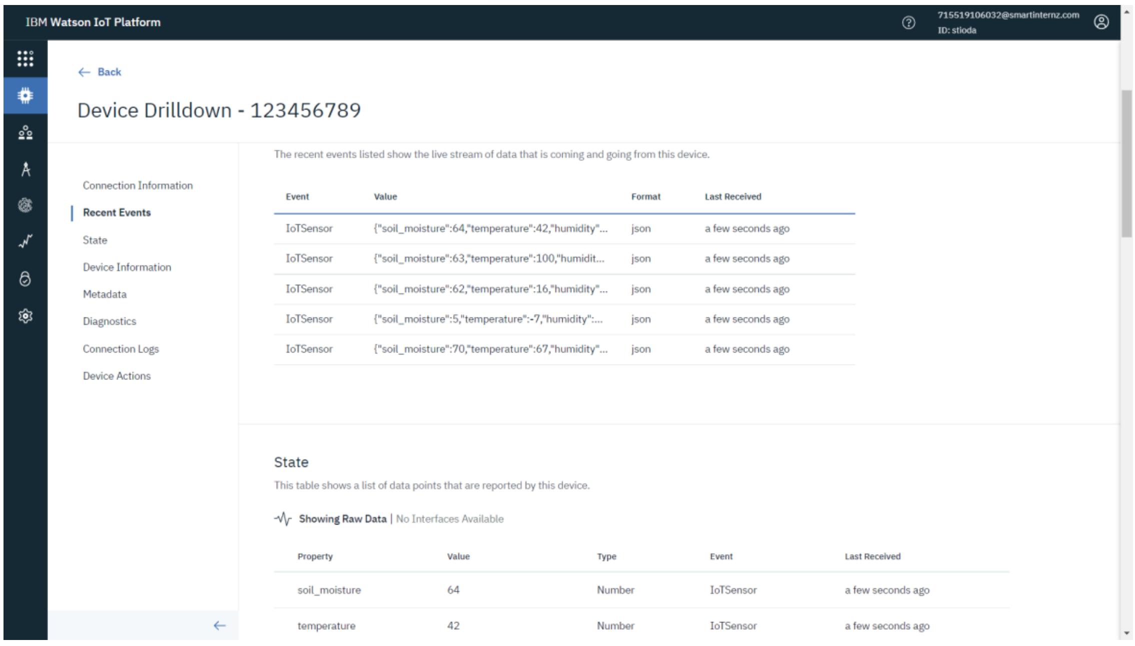
6. BUILDING PROJECT:

6.1. Connect Python script to IBM Watson IoT platform



```
{'soil_moisture': 79, 'temperature': -5, 'humidity': 48}
Published data Successfully:
{'soil_moisture': 14, 'temperature': -17, 'humidity': 13}
Published data Successfully:
{'soil_moisture': 31, 'temperature': 19, 'humidity': 58}
Published data Successfully:
{'soil_moisture': 80, 'temperature': 42, 'humidity': 17}
Published data Successfully:
{'soil_moisture': 59, 'temperature': -3, 'humidity': 38}
Published data Successfully:
{'soil_moisture': 72, 'temperature': 119, 'humidity': 31}
Published data Successfully:
{'soil_moisture': 89, 'temperature': 6, 'humidity': 46}
Published data Successfully:
{'soil_moisture': 80, 'temperature': 79, 'humidity': 78}
Published data Successfully:
{'soil_moisture': 15, 'temperature': 68, 'humidity': 66}
Published data Successfully:
{'soil_moisture': 85, 'temperature': 71, 'humidity': 2}
Published data Successfully:
{'soil_moisture': 60, 'temperature': -7, 'humidity': 98}
Published data Successfully:
{'soil_moisture': 0, 'temperature': 2, 'humidity': 30}
Published data Successfully:
{'soil_moisture': 39, 'temperature': 105, 'humidity': 77}
Published data Successfully:
{'soil_moisture': 2, 'temperature': 76, 'humidity': 26}
Published data Successfully:
{'soil_moisture': 50, 'temperature': 29, 'humidity': 28}
Published data Successfully:
{'soil_moisture': 86, 'temperature': 102, 'humidity': 58}
Published data Successfully:
{'soil_moisture': 75, 'temperature': 104, 'humidity': 78}
```

Connecting python script to Watson and sending data to IBM Watson IoT Platform.



Device Drilldown - 123456789

The recent events listed show the live stream of data that is coming and going from this device.

Event	Value	Format	Last Received
IoTSensor	{"soil_moisture":64,"temperature":42,"humidity"...	json	a few seconds ago
IoTSensor	{"soil_moisture":63,"temperature":100,"humidit...	json	a few seconds ago
IoTSensor	{"soil_moisture":62,"temperature":16,"humidity"...	json	a few seconds ago
IoTSensor	{"soil_moisture":5,"temperature":-7,"humidity":...	json	a few seconds ago
IoTSensor	{"soil_moisture":70,"temperature":67,"humidity"...	json	a few seconds ago

State

This table shows a list of data points that are reported by this device.

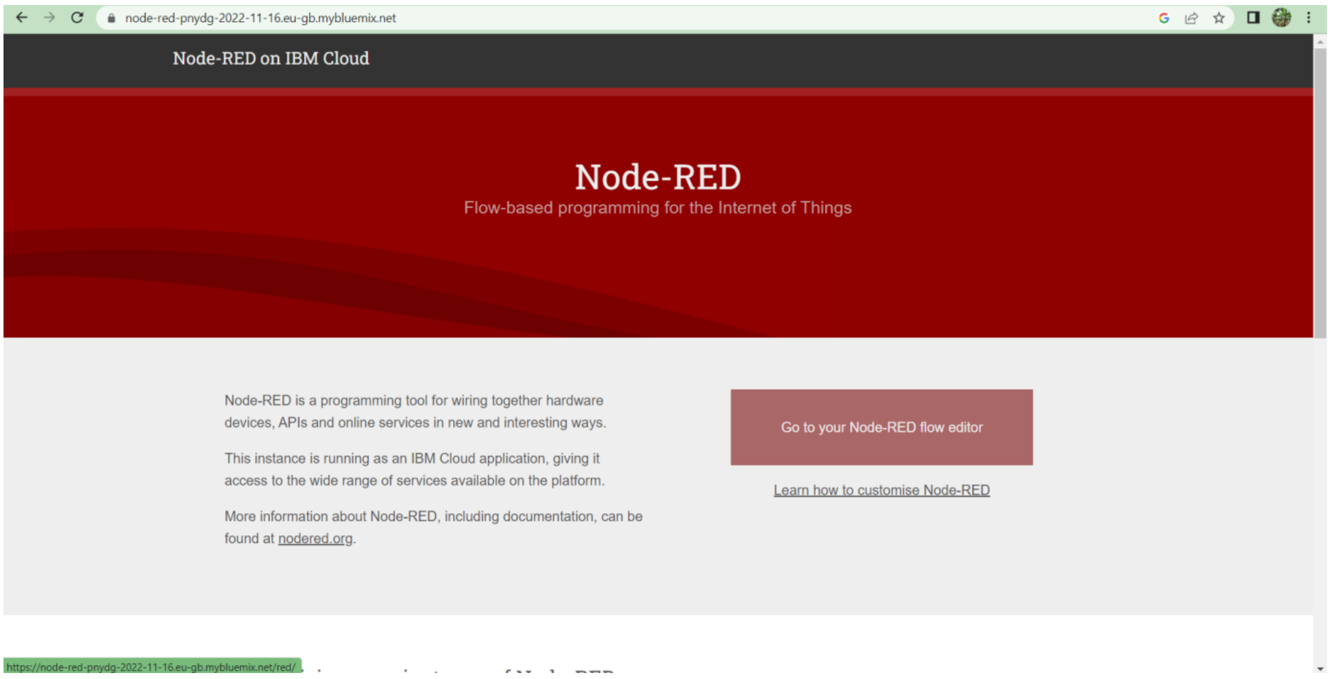
Showing Raw Data | No Interfaces Available

Property	Value	Type	Event	Last Received
soil_moisture	64	Number	IoTSensor	a few seconds ago
temperature	42	Number	IoTSensor	a few seconds ago

6.2. CONFIGURATION OF NODE-RED TO COLLECT IBM CLOUD DATA

Steps to configuring Node-RED

1.Click on Node-RED flow editor where we will be redirected to the Node-RED flow editor



node-red-pnydg-2022-11-16.eu-gb.mybluemix.net

Node-RED on IBM Cloud

Node-RED

Flow-based programming for the Internet of Things

Node-RED is a programming tool for wiring together hardware devices, APIs and online services in new and interesting ways.

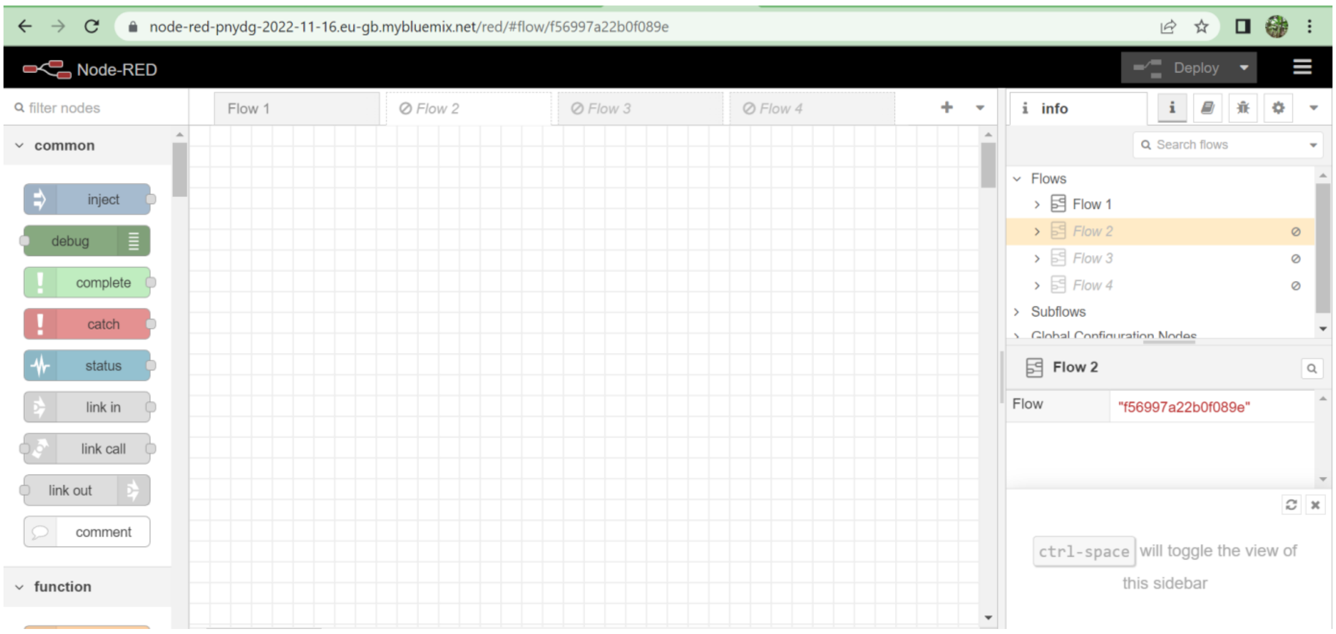
This instance is running as an IBM Cloud application, giving it access to the wide range of services available on the platform.

More information about Node-RED, including documentation, can be found at nodered.org.

Go to your Node-RED flow editor

Learn how to customise Node-RED

https://node-red-pnydg-2022-11-16.eu-gb.mybluemix.net/red/

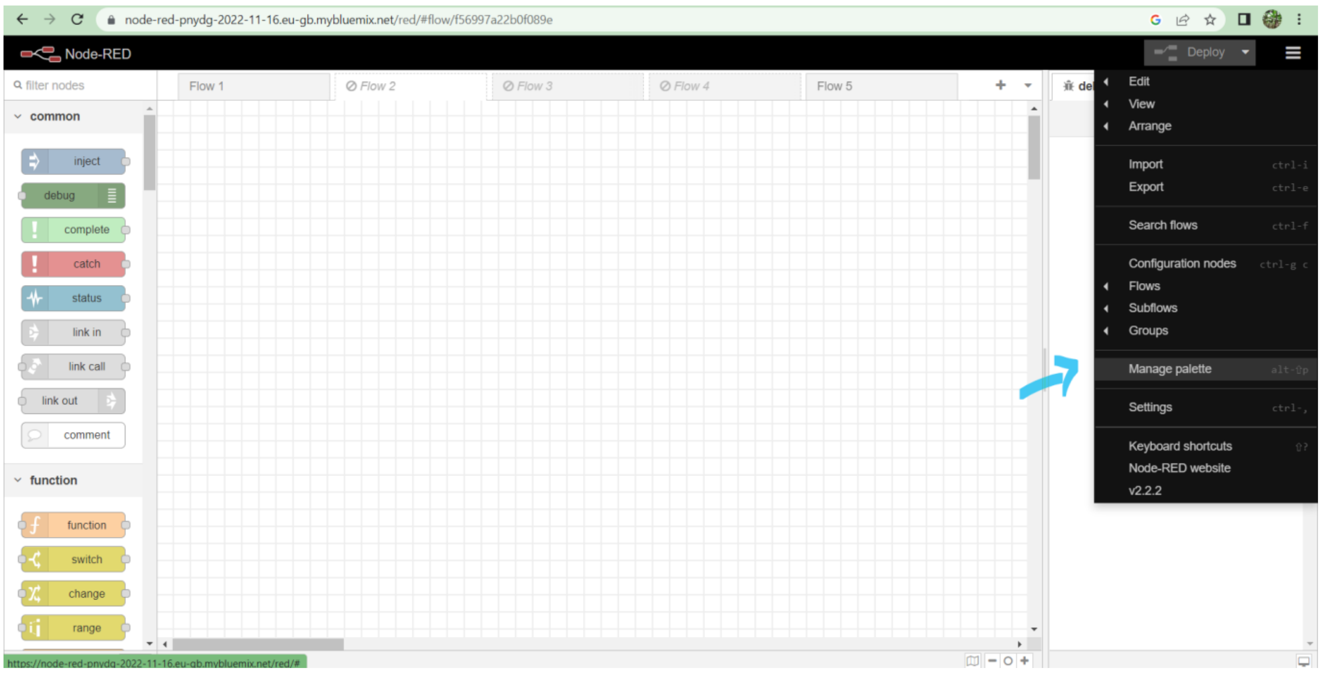


node-red-pnydg-2022-11-16.eu-gb.mybluemix.net/red/#flow/f56997a22b0f089e

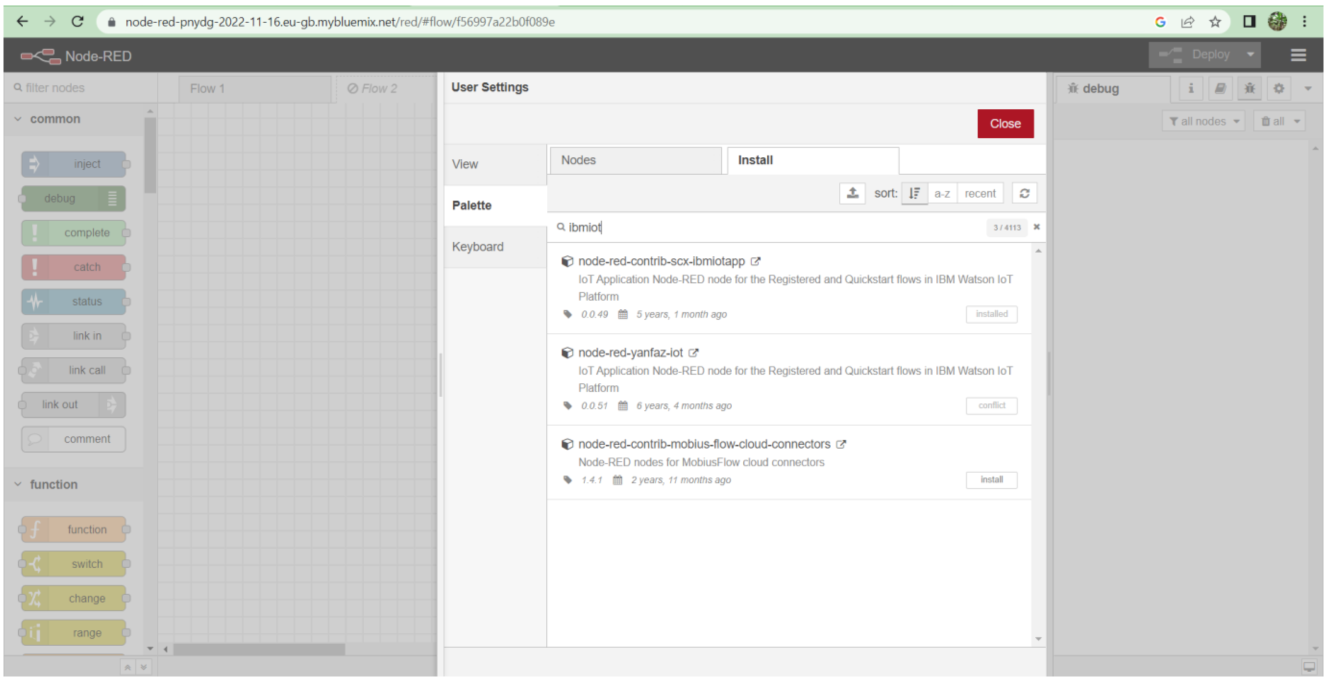
Flow "f56997a22b0f089e"

ctrl-space will toggle the view of this sidebar

2.To install IBM nodes in Node-RED flow editor click on manage palette in the menu option which is on the top-right of the screen.



3.In install section search for ibmiot and install the ibm nodes to flow editor

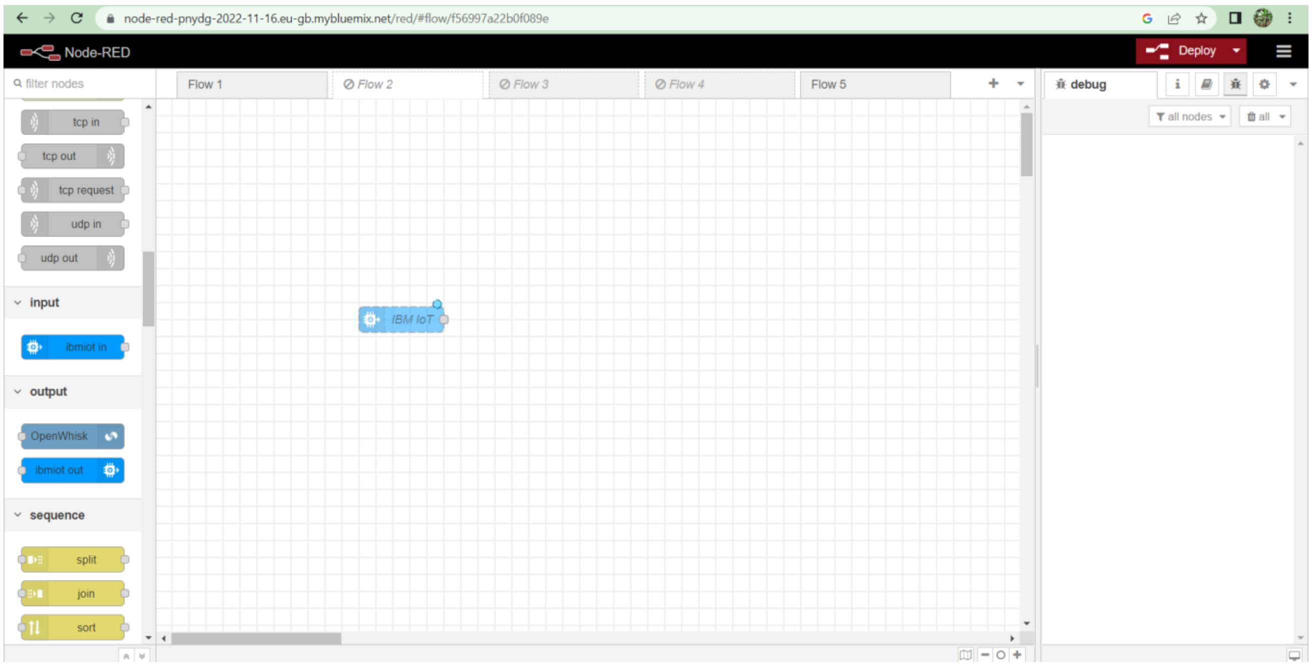


node-red-contrib-scx-ibmiotapp
IoT Application Node-RED node for the Registered and Quickstart flows in IBM Watson IoT Platform
0.0.49 5 years, 1 month ago installed

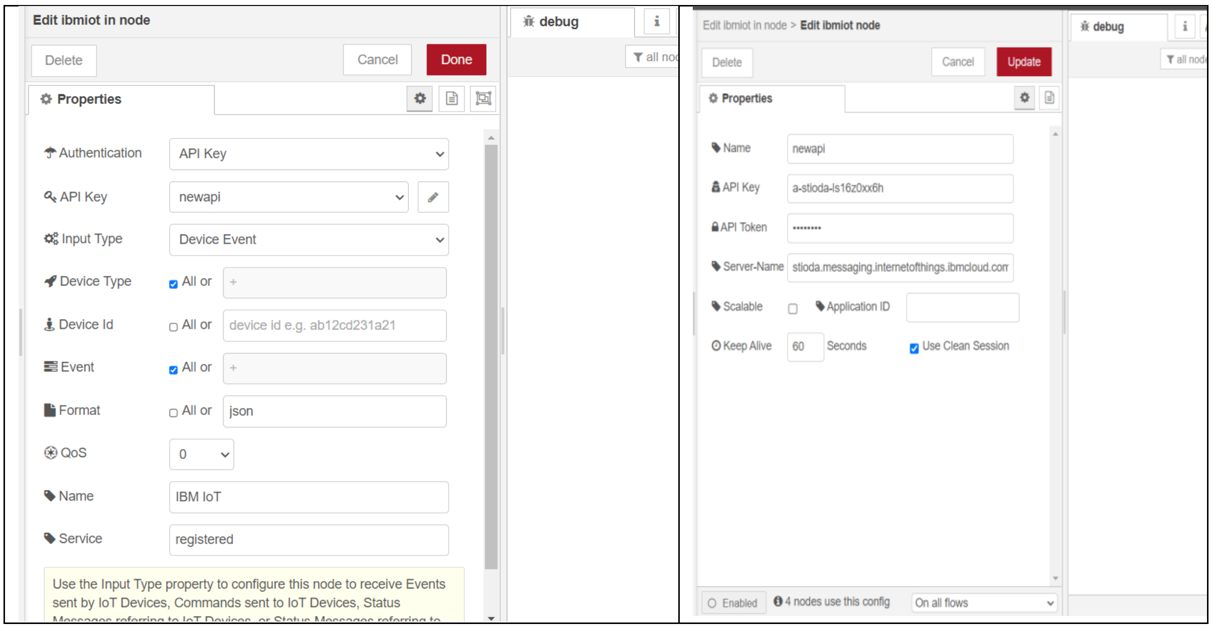
node-red-yanfaz-iot
IoT Application Node-RED node for the Registered and Quickstart flows in IBM Watson IoT Platform
0.0.51 6 years, 4 months ago conflict

node-red-contrib-mobius-flow-cloud-connectors
Node-RED nodes for MobiusFlow cloud connectors
1.4.1 2 years, 11 months ago install

4.To retrieve the data from the IBM IoT platform by using Node-RED IBM IoT Input node and double click on the IBM IoT input node



5.Select API Key from Authentication in properties and in API key paste API key, API Token and server name and update it.

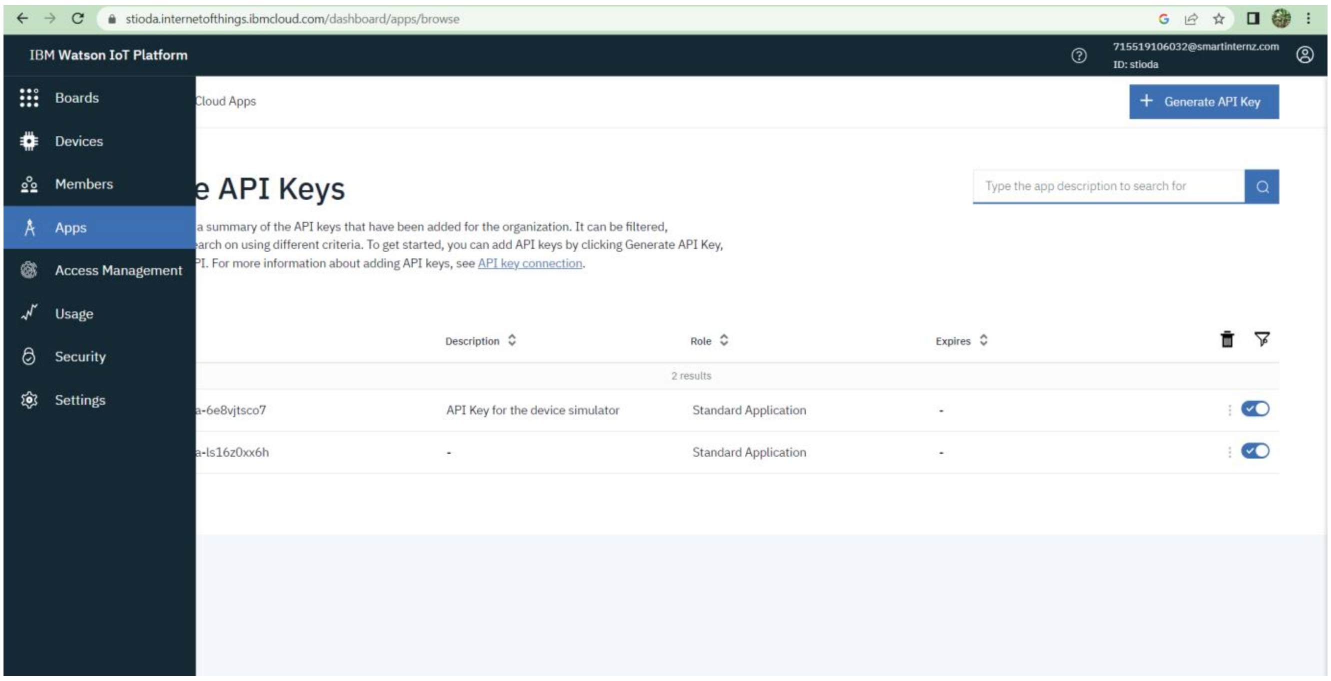


Authentication: API Key
API Key: newapi
Input Type: Device Event
Device Type: ☑ All or +
Device Id: ☐ All or device id e.g. ab12cd231a21
Event: ☑ All or +
Format: ☐ All or json
QoS: 0
Name: IBM IoT
Service: registered

Use the Input Type property to configure this node to receive Events sent by IoT Devices, Commands sent to IoT Devices, Status Messages referring to IoT Devices, or Status Messages referring to

Name: newapi
API Key: a-stioda-ls16z0xx6h
API Token: ••••••••
Server-Name: stioda.messaging.internetofthings.ibmcloud.com
Scalable: ☐ Application ID
Keep Alive: 60 Seconds ☑ Use Clean Session
Enabled
4 nodes use this config
On all flows

6.To generate API Key go to IBM IoT platform. In Apps section, click on Generate API key and click next for information. In permissions section select Standard application as role and click on generate API key and copy the API key and Authentication token .

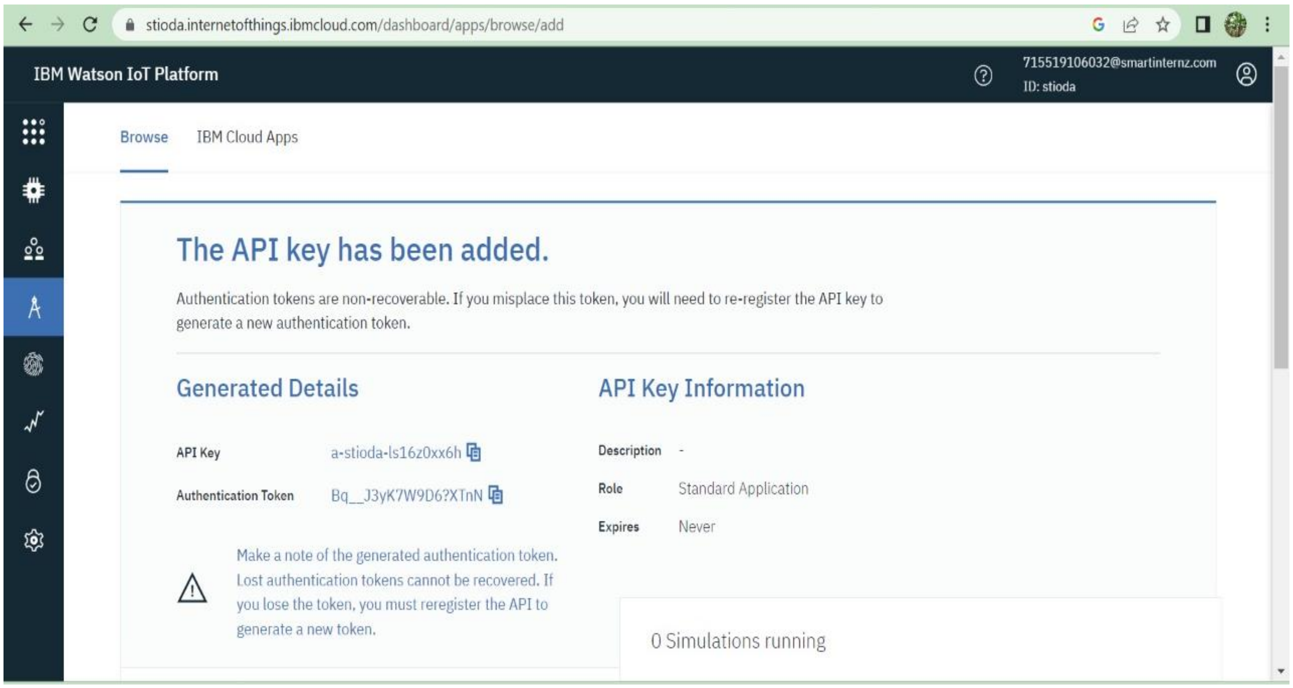

stioda.internetofthings.ibmcloud.com/dashboard/apps/browse
IBM Watson IoT Platform
715519106032@smartinternz.com
ID: stioda
Boards
Devices
Members
Apps
Access Management
Usage
Security
Settings
Cloud Apps
Generate API Key
Type the app description to search for

e API Keys

a summary of the API keys that have been added for the organization. It can be filtered, arch on using different criteria. To get started, you can add API keys by clicking Generate API Key, PI. For more information about adding API keys, see API key connection.

	Description	Role	Expires
a-6e8vjtsco7	API Key for the device simulator	Standard Application	-
a-ls16z0xx6h	-	Standard Application	-

2 results


stioda.internetofthings.ibmcloud.com/dashboard/apps/browse/add
IBM Watson IoT Platform
715519106032@smartinternz.com
ID: stioda
Browse
IBM Cloud Apps

The API key has been added.

Authentication tokens are non-recoverable. If you misplace this token, you will need to re-register the API key to generate a new authentication token.

Generated Details

API Key a-stioda-ls16z0xx6h

Authentication Token Bq__J3yK7W9D6?XTnN

Make a note of the generated authentication token. Lost authentication tokens cannot be recovered. If you lose the token, you must reregister the API to generate a new token.

API Key Information

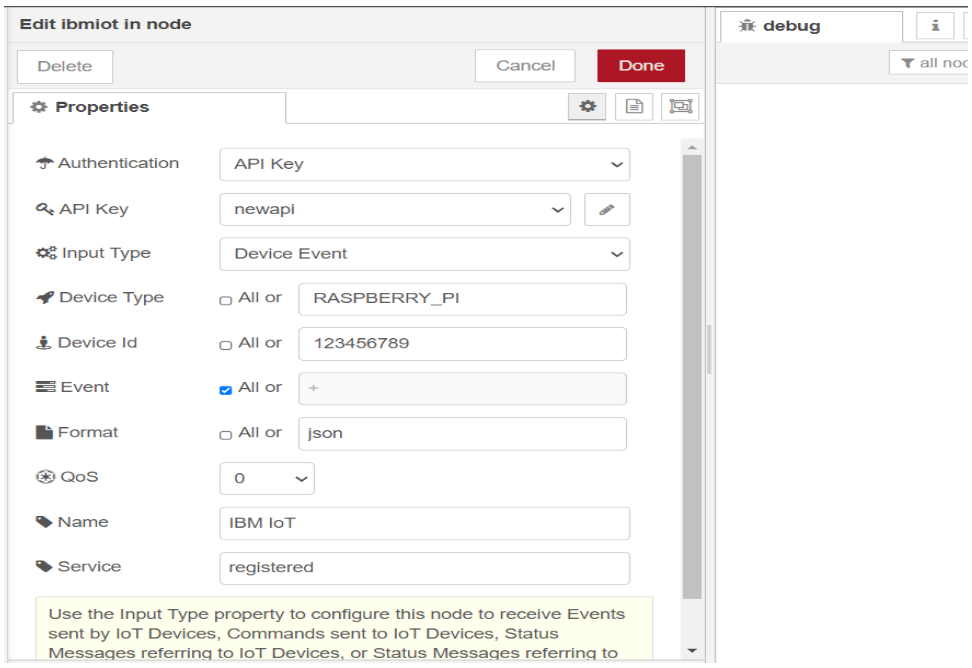
Description -

Role Standard Application

Expires Never

0 Simulations running

7. Update the input type as event, device type, device id format in the properties section and click on done.

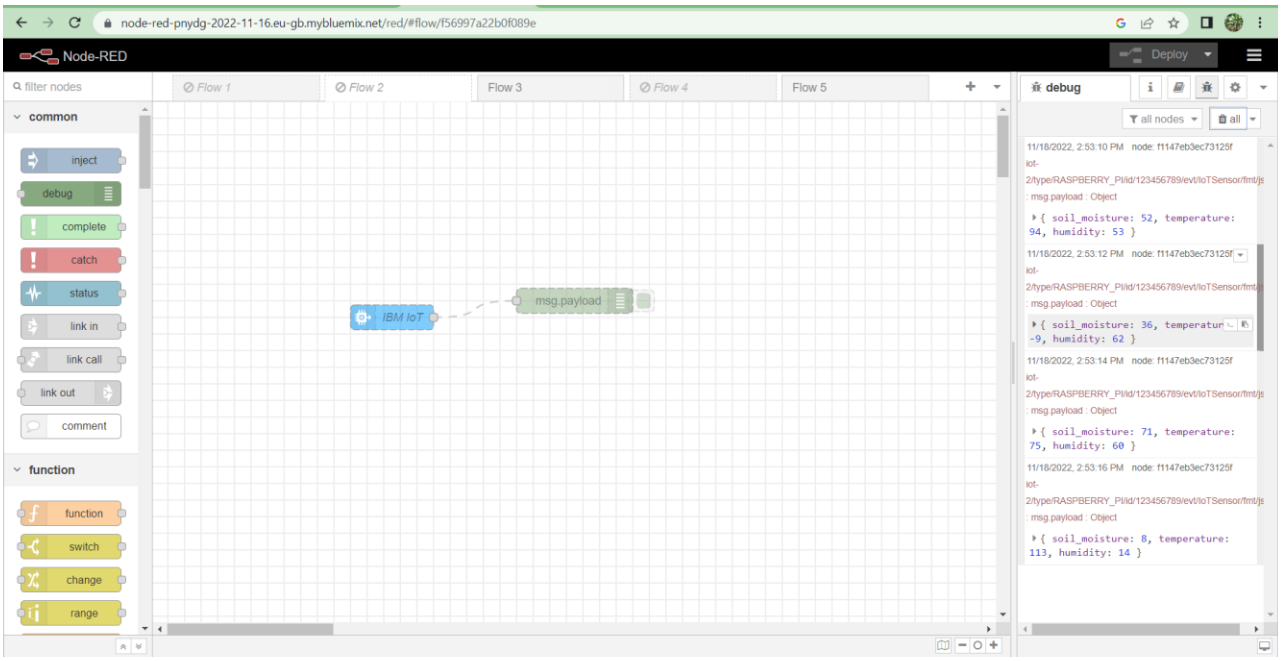


Edit ibmiot in node

Authentication: API Key
API Key: newapi
Input Type: Device Event
Device Type: ☐ All or RASPBERRY_PI
Device Id: ☐ All or 123456789
Event: ☑ All or +
Format: ☐ All or json
QoS: 0
Name: IBM IoT
Service: registered

Use the Input Type property to configure this node to receive Events sent by IoT Devices, Commands sent to IoT Devices, Status Messages referring to IoT Devices, or Status Messages referring to

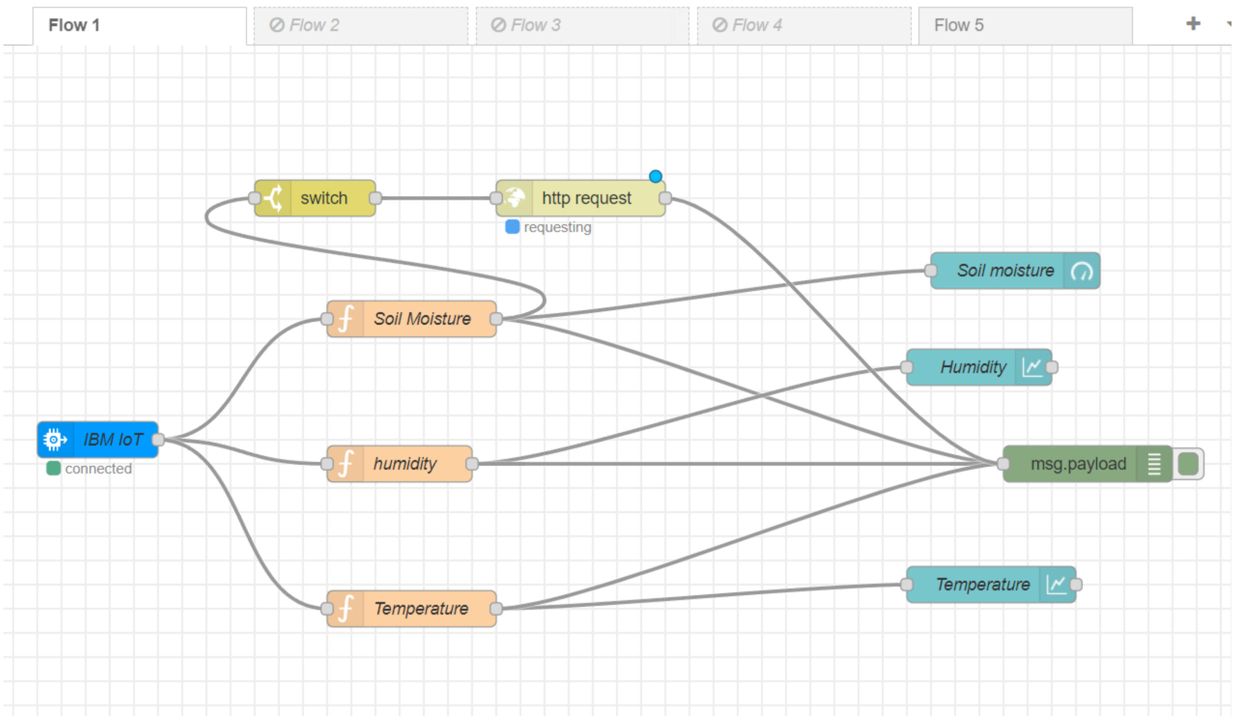
8.Place debug node in the flow editor and click on deploy to see the temperature, humidity and soil moisture value in the debug mode.



node-red-pnydg-2022-11-16.eu-gb.mybluemix.net/red/#flow/f56997a22b0f089e

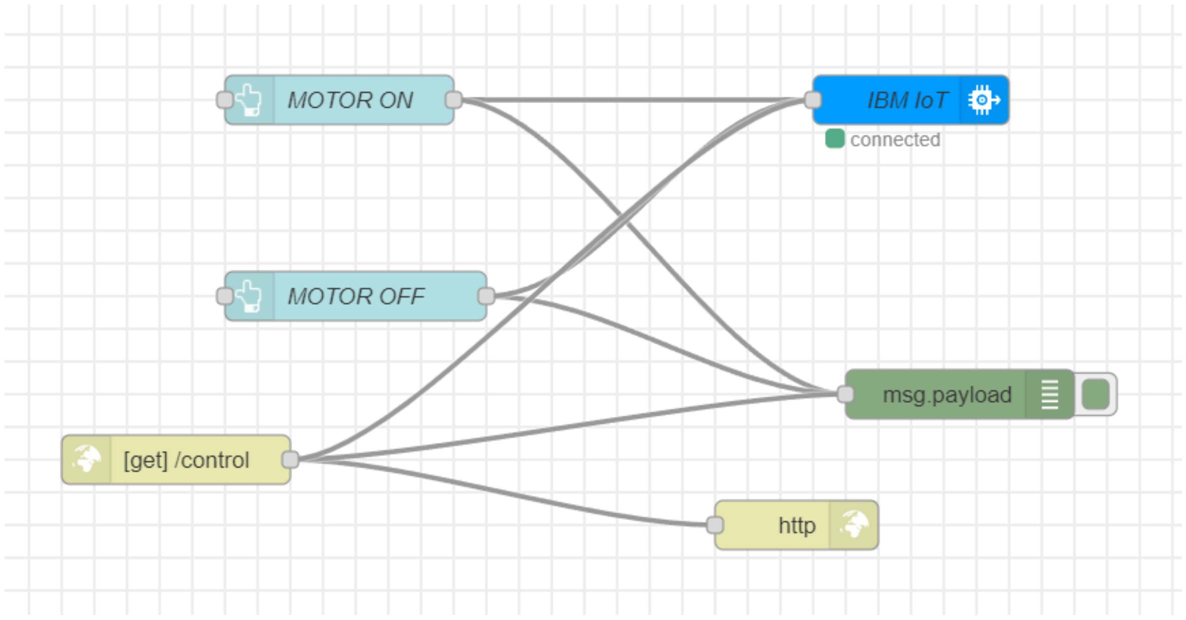
```
11/18/2022, 2:53:10 PM   node: f1147eb3ec73125f
iot-2/type/RASPBERRY_PI/id/123456789/evt/IoTSensor/fmt/js
: msg.payload : Object
▸{ soil_moisture: 52, temperature: 94, humidity: 53 }
11/18/2022, 2:53:12 PM   node: f1147eb3ec73125f
iot-2/type/RASPBERRY_PI/id/123456789/evt/IoTSensor/fmt/js
: msg.payload : Object
▸{ soil_moisture: 36, temperatur
-9, humidity: 62 }
11/18/2022, 2:53:14 PM   node: f1147eb3ec73125f
iot-2/type/RASPBERRY_PI/id/123456789/evt/IoTSensor/fmt/js
: msg.payload : Object
▸{ soil_moisture: 71, temperature: 75, humidity: 60 }
11/18/2022, 2:53:16 PM   node: f1147eb3ec73125f
iot-2/type/RASPBERRY_PI/id/123456789/evt/IoTSensor/fmt/js
: msg.payload : Object
▸{ soil_moisture: 8, temperature: 113, humidity: 14 }
```

9.connect the required nodes to retrieve data from IBM Watson IoT Platform

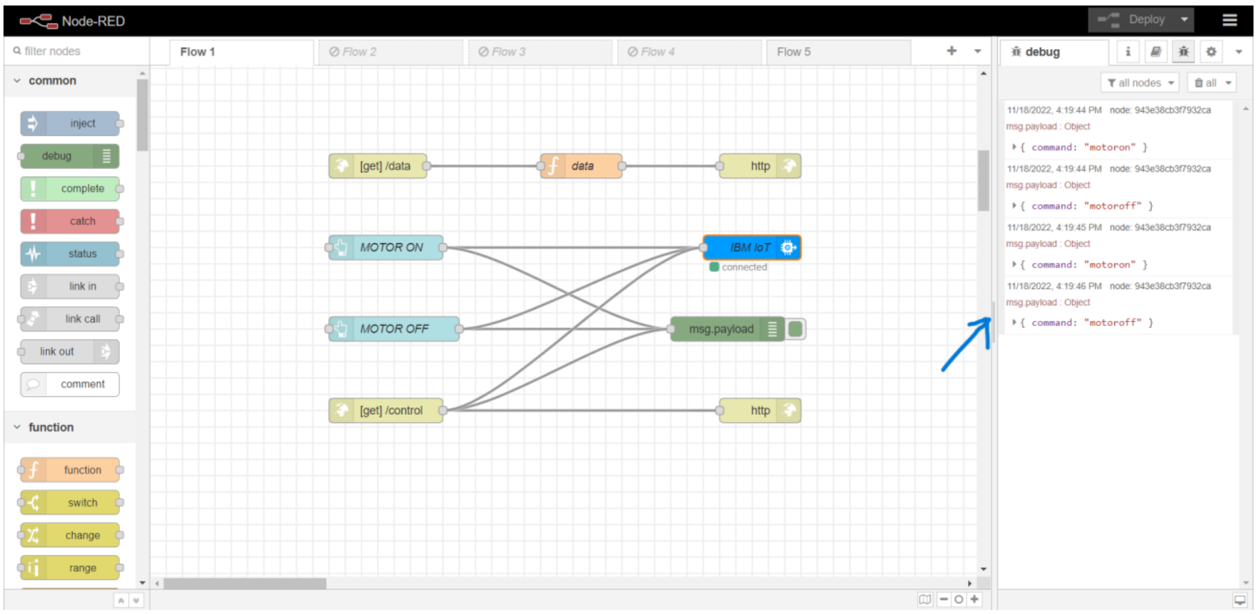


6.3.. CONFIGURATION OF NODE-RED TO SEND COMMANDS TO IBM CLOUD

1.connect the motor on and motor off nodes to send command to IBM Watson IoT platform through IBM node

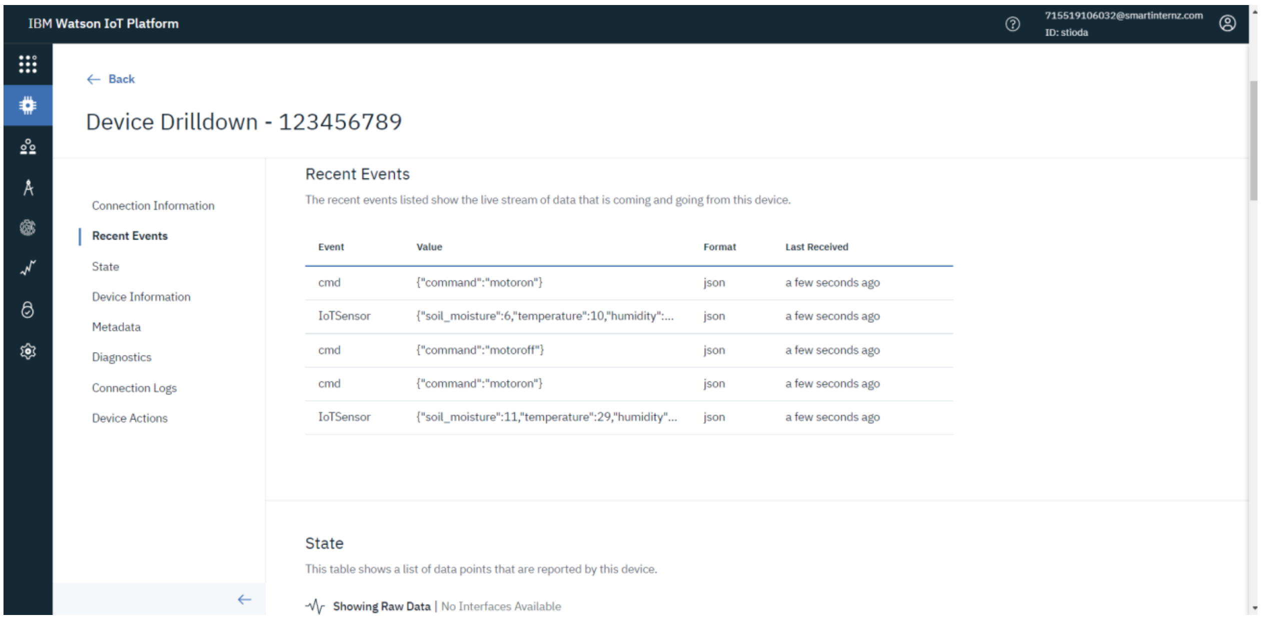


2. sending a motor on and off command through node red



```
11/18/2022, 4:19:44 PM   node: 943e38cb3f7932ca
msg.payload : Object
▸{ command: "motoron" }
11/18/2022, 4:19:44 PM   node: 943e38cb3f7932ca
msg.payload : Object
▸{ command: "motoroff" }
11/18/2022, 4:19:45 PM   node: 943e38cb3f7932ca
msg.payload : Object
▸{ command: "motoron" }
11/18/2022, 4:19:46 PM   node: 943e38cb3f7932ca
msg.payload : Object
▸{ command: "motoroff" }
```

3. We can see the data in IBM Watson device



Device Drilldown - 123456789

Recent Events

The recent events listed show the live stream of data that is coming and going from this device.

Event	Value	Format	Last Received
cmd	{"command":"motoron"}	json	a few seconds ago
IoTSensor	{"soil_moisture":6,"temperature":10,"humidity":...	json	a few seconds ago
cmd	{"command":"motoroff"}	json	a few seconds ago
cmd	{"command":"motoron"}	json	a few seconds ago
IoTSensor	{"soil_moisture":11,"temperature":29,"humidity"...	json	a few seconds ago

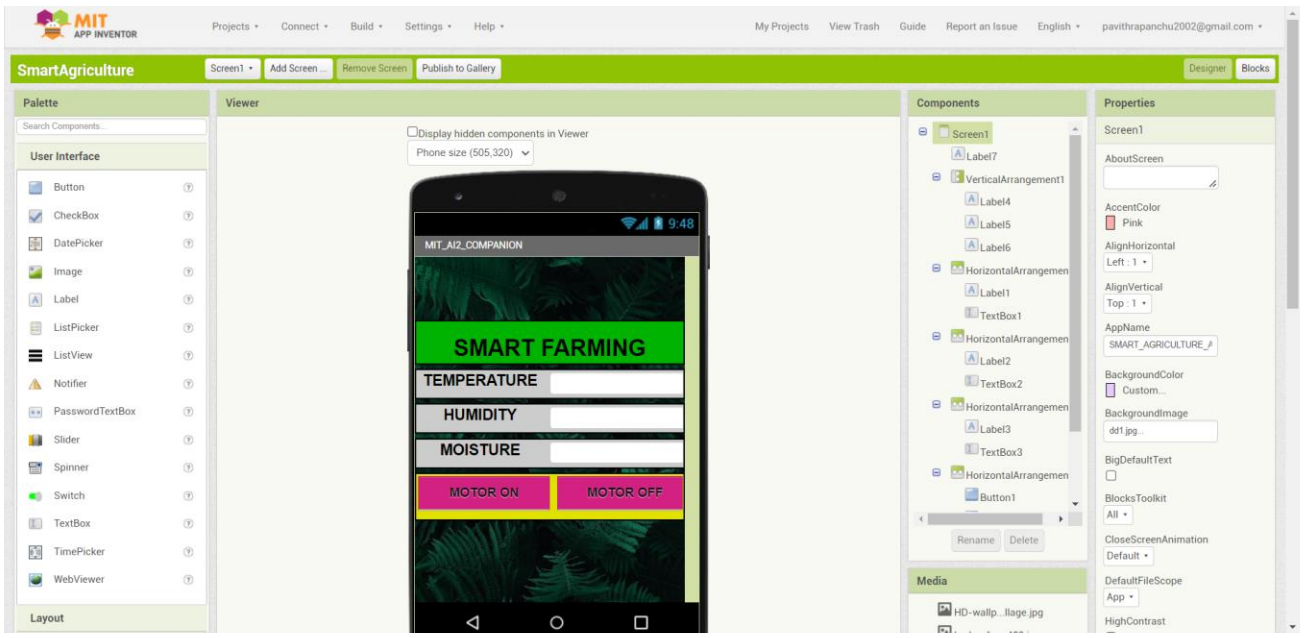
State

This table shows a list of data points that are reported by this device.

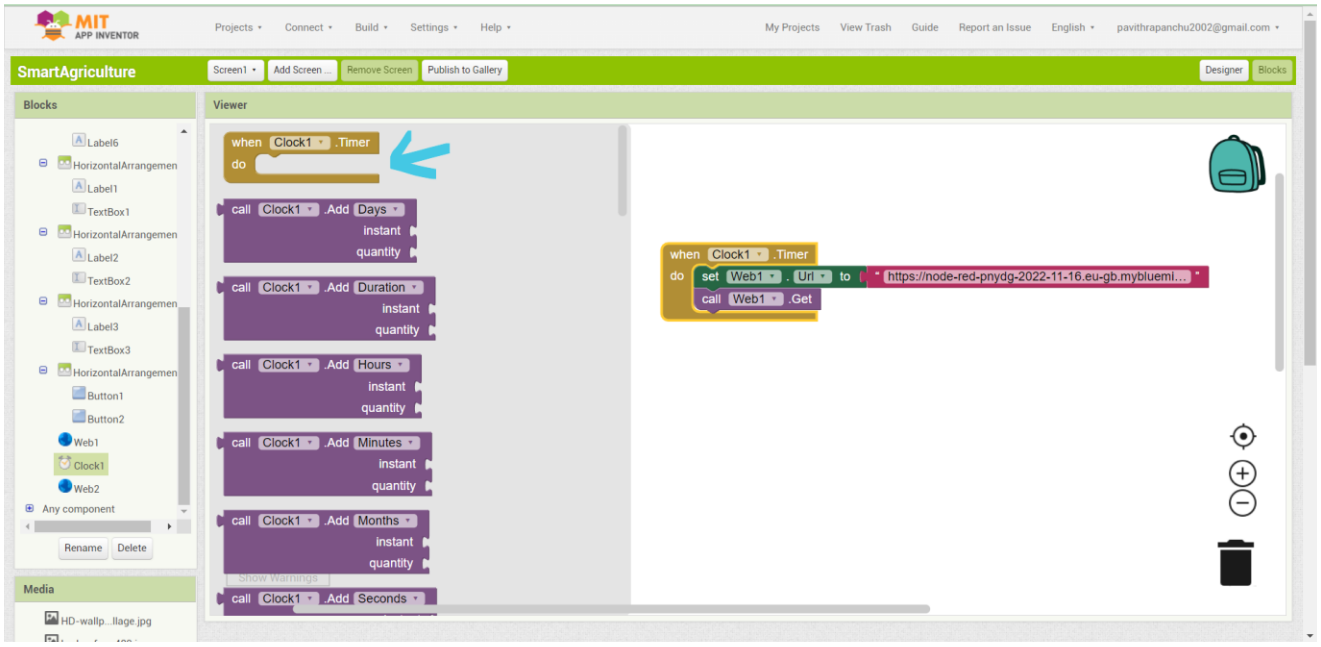
Showing Raw Data | No Interfaces Available

6.4 . CONFIGURATION OF MIT APP INVENTOR AND CONNECTING MIT APP INVENTOR TO NODE RED

1.Add screen and add required text field, labels and buttons in screen

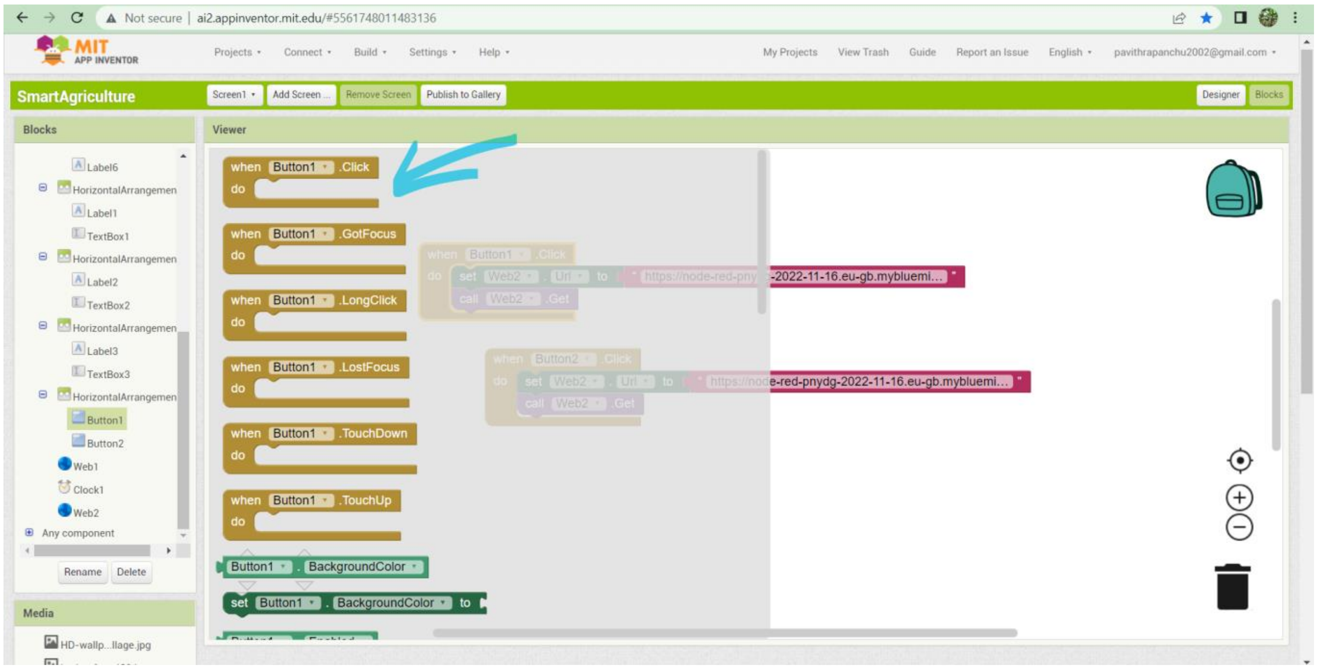


2. Go to block section and add clock1 timer block and set web1 URL to [https://node-red-pnydg-2022-11-16.eu-gb.mybluemix.net/data and call the web1](https://node-red-pnydg-2022-11-16.eu-gb.mybluemix.net/data) to get data from it.



```
when Clock1 .Timer
do

call Clock1 .Add Days
    instant
    quantity
call Clock1 .Add Duration
    instant
    quantity
call Clock1 .Add Hours
    instant
    quantity
call Clock1 .Add Minutes
    instant
    quantity
call Clock1 .Add Months
    instant
    quantity
call Clock1 .Add Seconds

when Clock1 .Timer
do  set Web1 . Url to " https://node-red-pnydg-2022-11-16.eu-gb.mybluemi... "
    call Web1 .Get
```

3. Add button for motor on and off it will send the command to IBM Watson IoT Platform through Node-RED .



when Button1 .Click
do

when Button1 .GotFocus
do

when Button1 .LongClick
do

when Button1 .LostFocus
do

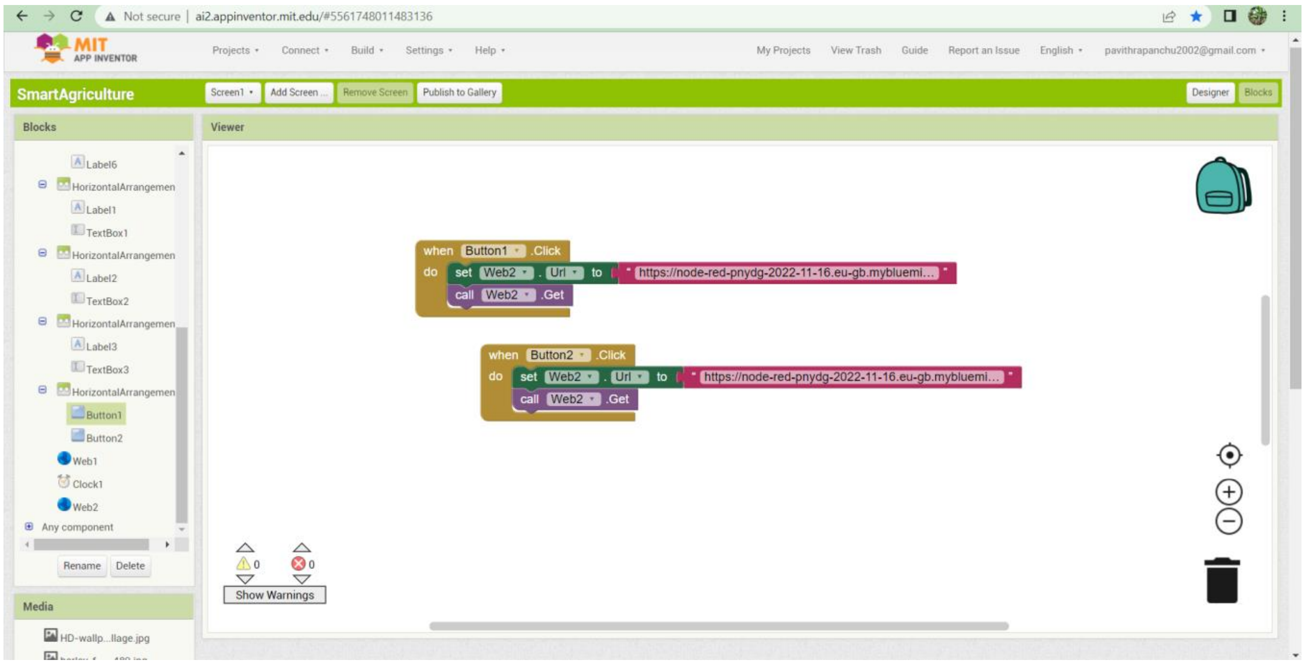
when Button1 .TouchDown
do

when Button1 .TouchUp
do

Button1 . BackgroundColor

set Button1 . BackgroundColor to

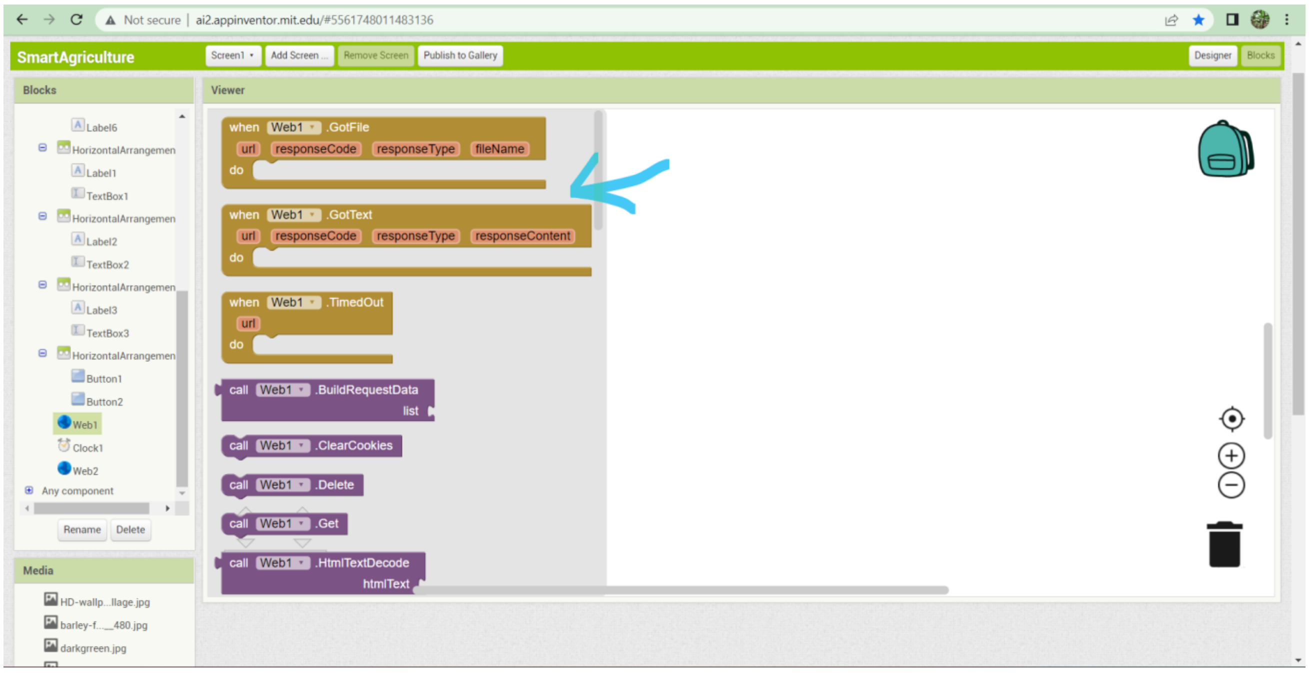
4.In button add URL links for motor on and motor off https://node-red-pnydg-2022-11-16.eugb.mybluemix.net/control?command=motoron and https://node-red-pnydg-2022-11-16.eu-gb.mybluemix.net/control?command=motoroff



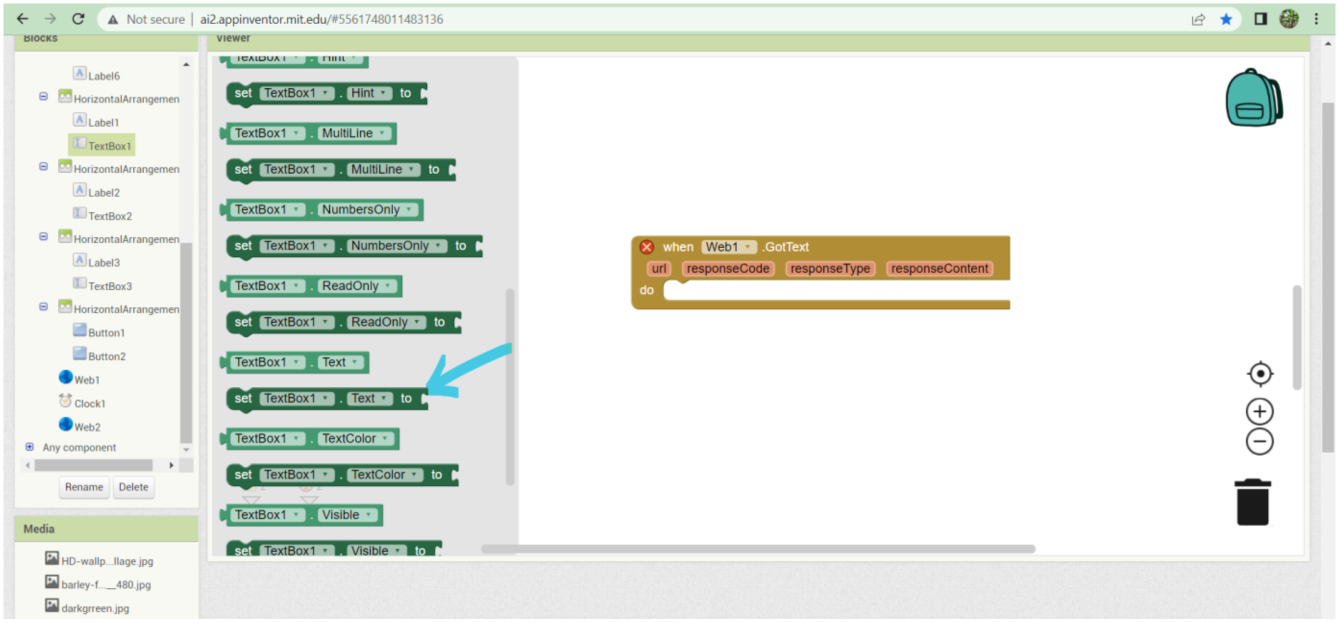
when Button1 .Click
do set Web2 . Url to " https://node-red-pnydg-2022-11-16.eu-gb.mybluemi... "
call Web2 .Get

when Button2 .Click
do set Web2 . Url to " https://node-red-pnydg-2022-11-16.eu-gb.mybluemi... "
call Web2 .Get

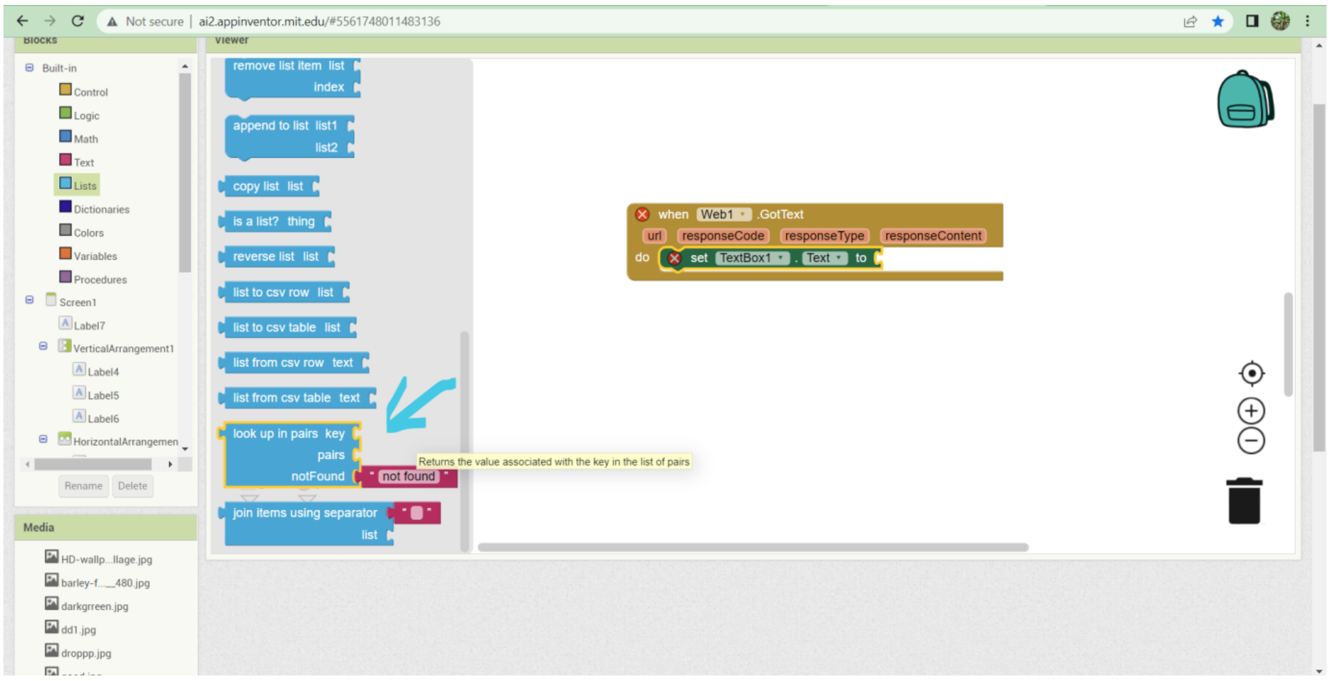
5. when web1 component got data and we will show in the text field so drag the required blocks and join them.



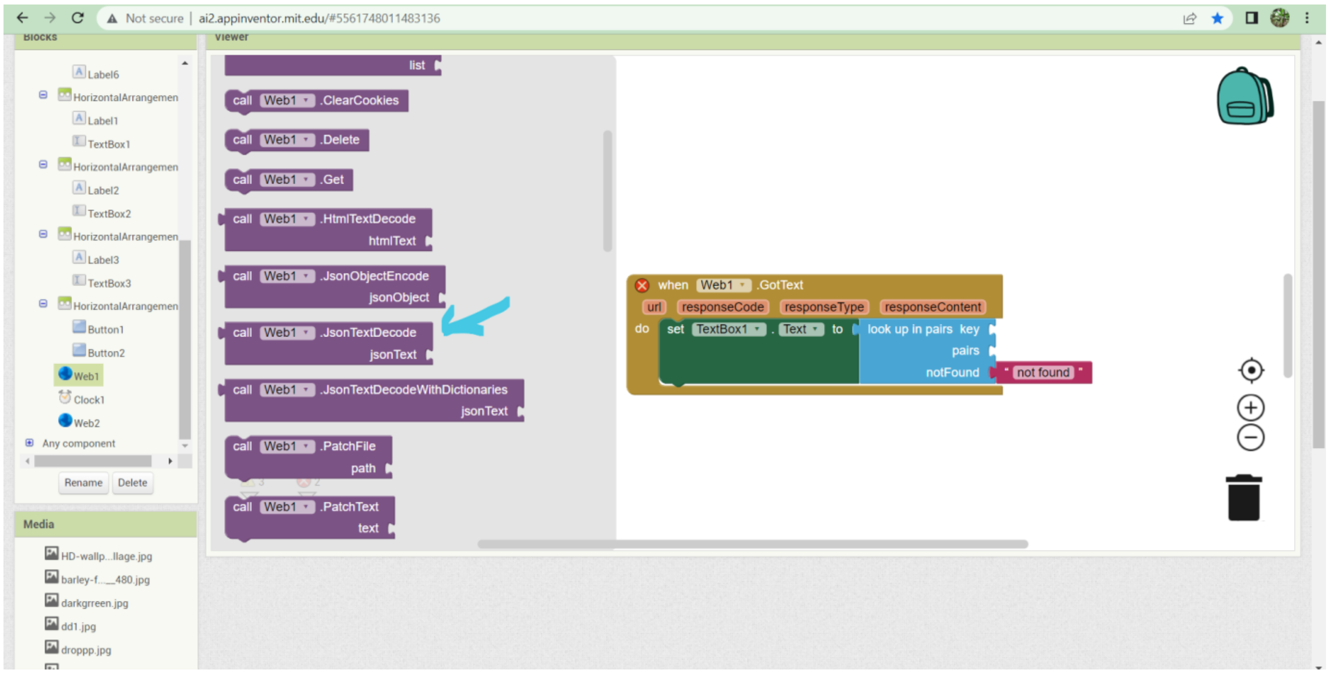
```
when Web1 .GotFile
  url  responseCode  responseType  fileName
do

when Web1 .GotText
  url  responseCode  responseType  responseContent
do

when Web1 .TimedOut
  url
do

call Web1 .BuildRequestData
  list

call Web1 .ClearCookies

call Web1 .Delete

call Web1 .Get

call Web1 .HtmlTextDecode
  htmlText
```



```
set TextBox1 . Hint to
TextBox1 . MultiLine
set TextBox1 . MultiLine to
TextBox1 . NumbersOnly
set TextBox1 . NumbersOnly to
TextBox1 . ReadOnly
set TextBox1 . ReadOnly to
TextBox1 . Text
set TextBox1 . Text to
TextBox1 . TextColor
set TextBox1 . TextColor to
TextBox1 . Visible
set TextBox1 . Visible to

when Web1 .GotText
  url  responseCode  responseType  responseContent
do
```

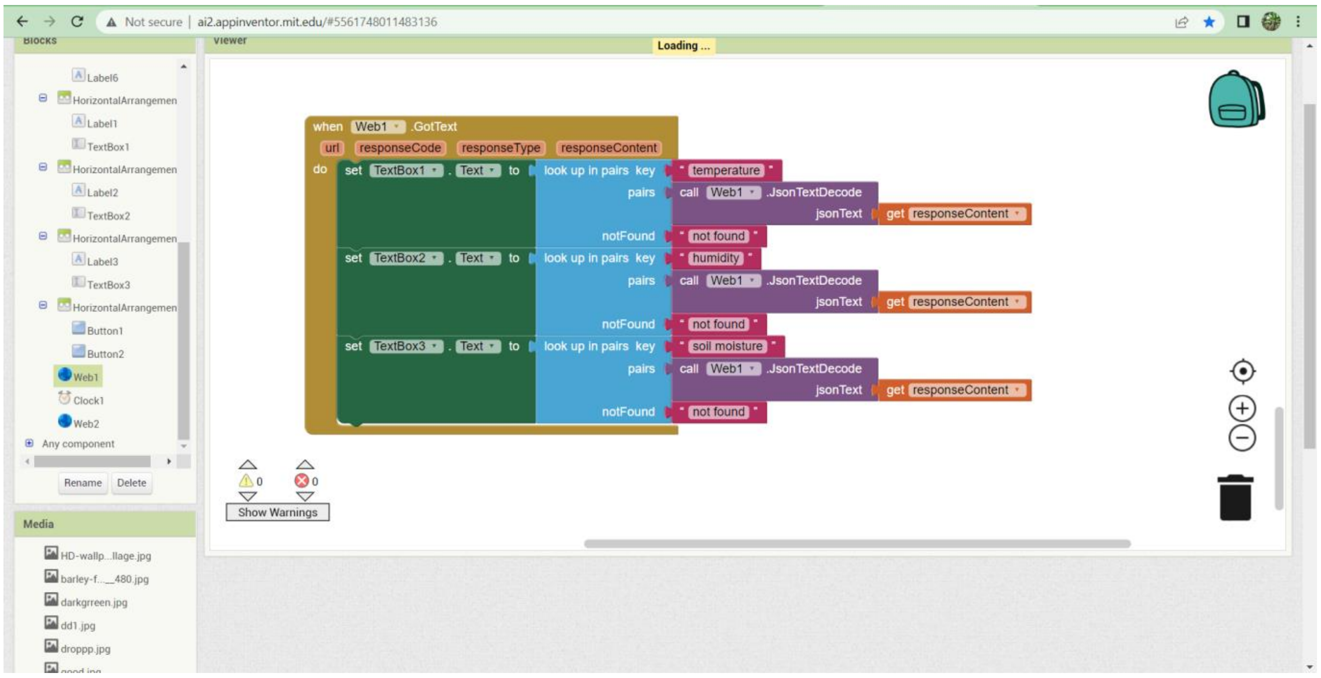


```
when Web1 .GotText
  url responseCode responseType responseContent
do set TextBox1 . Text to
```

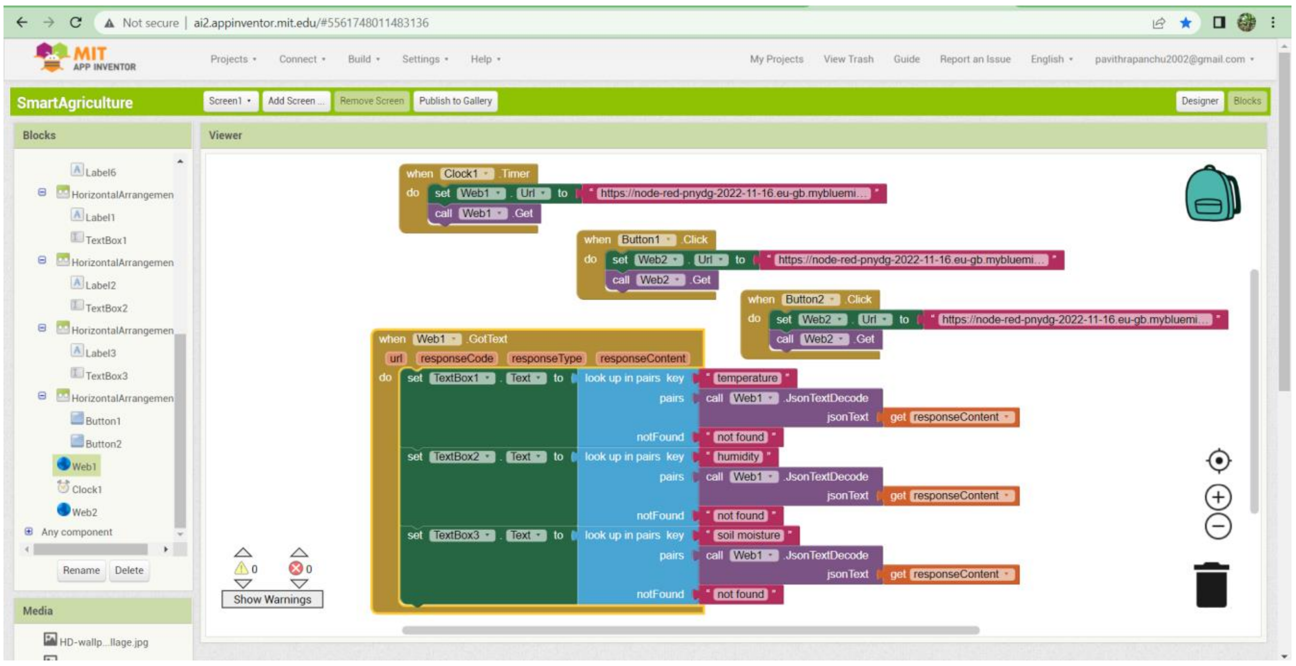


```
when Web1 .GotText
  url responseCode responseType responseContent
do set TextBox1 . Text to look up in pairs key
                                            pairs
                                            notFound "not found"
```

6. And also add blocks for the humidity and soil moisture also



```
when Web1 .GotText
  url  responseCode  responseType  responseContent
do  set TextBox1 . Text to  look up in pairs  key  " temperature "
                                              pairs  call Web1 .JsonTextDecode
                                                            jsonText  get responseContent
                                           notFound  " not found "
    set TextBox2 . Text to  look up in pairs  key  " humidity "
                                              pairs  call Web1 .JsonTextDecode
                                                            jsonText  get responseContent
                                           notFound  " not found "
    set TextBox3 . Text to  look up in pairs  key  " soil moisture "
                                              pairs  call Web1 .JsonTextDecode
                                                            jsonText  get responseContent
                                           notFound  " not found "
```



```
when Clock1 .Timer
do  set Web1 . Url to  " https://node-red-pnydg-2022-11-16.eu-gb.mybluemi... "
    call Web1 .Get

when Button1 .Click
do  set Web2 . Url to  " https://node-red-pnydg-2022-11-16.eu-gb.mybluemi... "
    call Web2 .Get

when Button2 .Click
do  set Web2 . Url to  " https://node-red-pnydg-2022-11-16.eu-gb.mybluemi... "
    call Web2 .Get

when Web1 .GotText
  url  responseCode  responseType  responseContent
do  set TextBox1 . Text to  look up in pairs  key  " temperature "
                                              pairs  call Web1 .JsonTextDecode
                                                            jsonText  get responseContent
                                           notFound  " not found "
    set TextBox2 . Text to  look up in pairs  key  " humidity "
                                              pairs  call Web1 .JsonTextDecode
                                                            jsonText  get responseContent
                                           notFound  " not found "
    set TextBox3 . Text to  look up in pairs  key  " soil moisture "
                                              pairs  call Web1 .JsonTextDecode
                                                            jsonText  get responseContent
                                           notFound  " not found "
```