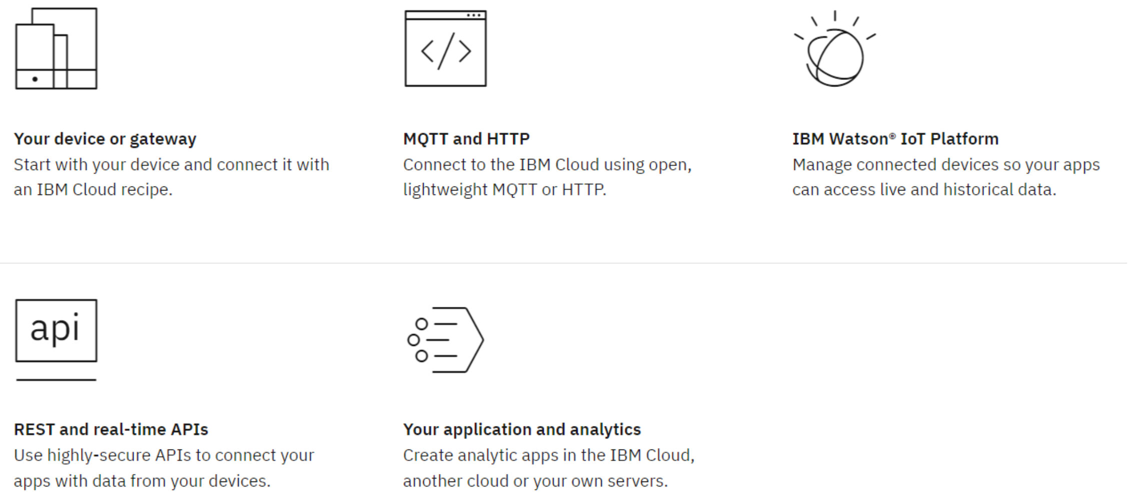
Sprint- 2

Team ID	**PNT2022TMID11481**
Project Title	**Smart Farmer -Iot Enabled Smart Farming Application**
Date	**14.11.2022**

IBM Watson and Python Integration:

By using Watson IoT Platform, you can collect connected device data and perform analytics on real-time data. The IBM Watson IoT Platform is a fully managed, Cloud-hosted service that provides device management capabilities as well as data collection and management in a time series format.



Your device or gateway
Start with your device and connect it with an IBM Cloud recipe.

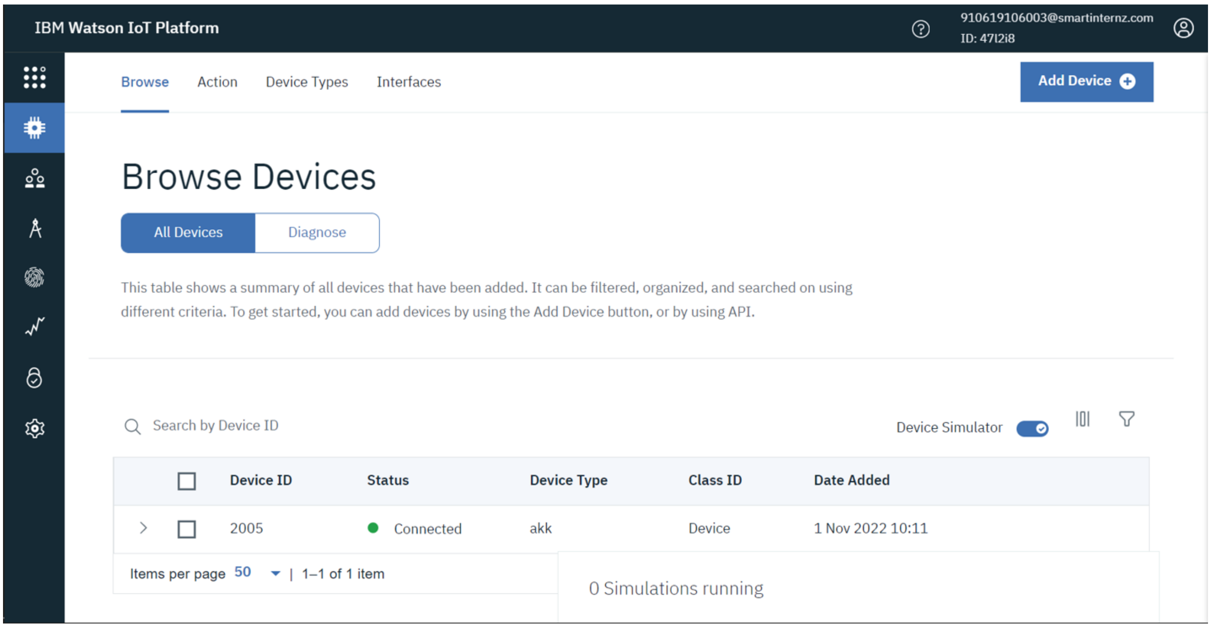
MQTT and HTTP
Connect to the IBM Cloud using open, lightweight MQTT or HTTP.

IBM Watson® IoT Platform
Manage connected devices so your apps can access live and historical data.

REST and real-time APIs
Use highly-secure APIs to connect your apps with data from your devices.

Your application and analytics
Create analytic apps in the IBM Cloud, another cloud or your own servers.

Using the Device Created in IBM Watson:



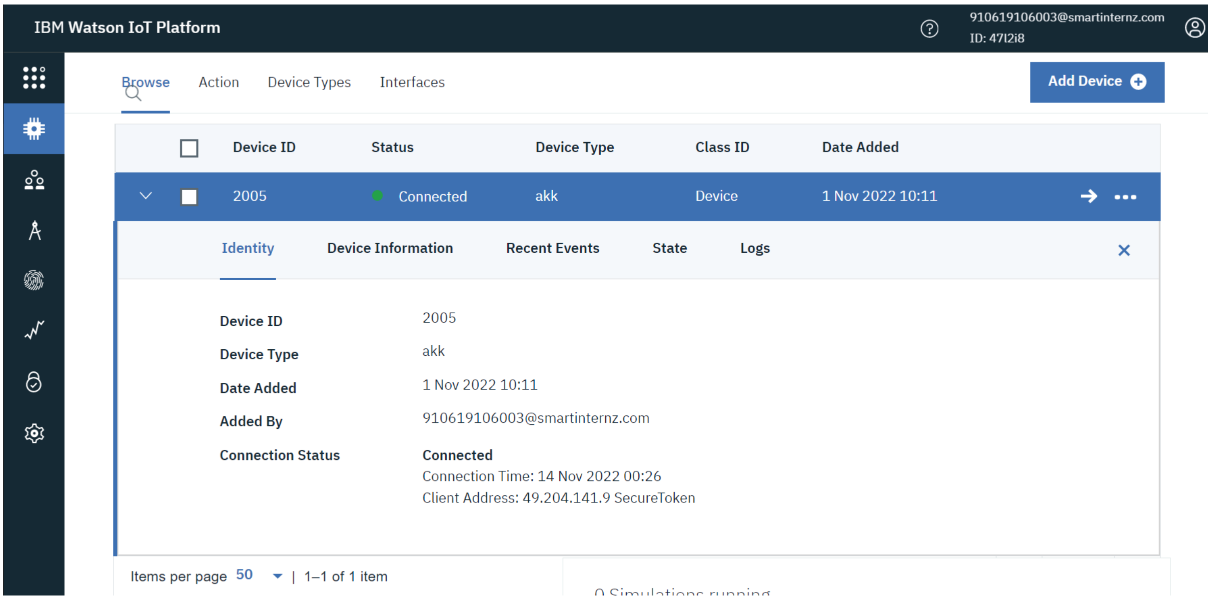
Browse Devices

This table shows a summary of all devices that have been added. It can be filtered, organized, and searched on using different criteria. To get started, you can add devices by using the Add Device button, or by using API.

	Device ID	Status	Device Type	Class ID	Date Added
□	2005	● Connected	akk	Device	1 Nov 2022 10:11

Items per page 50 | 1–1 of 1 item

0 Simulations running



	Device ID	Status	Device Type	Class ID	Date Added
□	2005	● Connected	akk	Device	1 Nov 2022 10:11

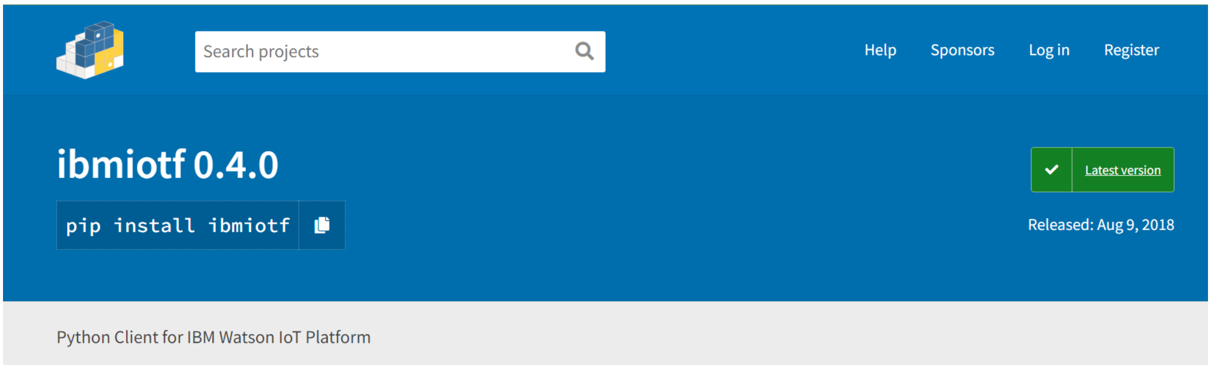
Device ID	2005
Device Type	akk
Date Added	1 Nov 2022 10:11
Added By	910619106003@smartinternz.com
Connection Status	Connected Connection Time: 14 Nov 2022 00:26 Client Address: 49.204.141.9 SecureToken

Items per page 50 | 1–1 of 1 item

0 Simulations running

Connected sign shows that it is connected and live

Python code execution:



ibmiotf 0.4.0

`pip install ibmiotf`

Released: Aug 9, 2018

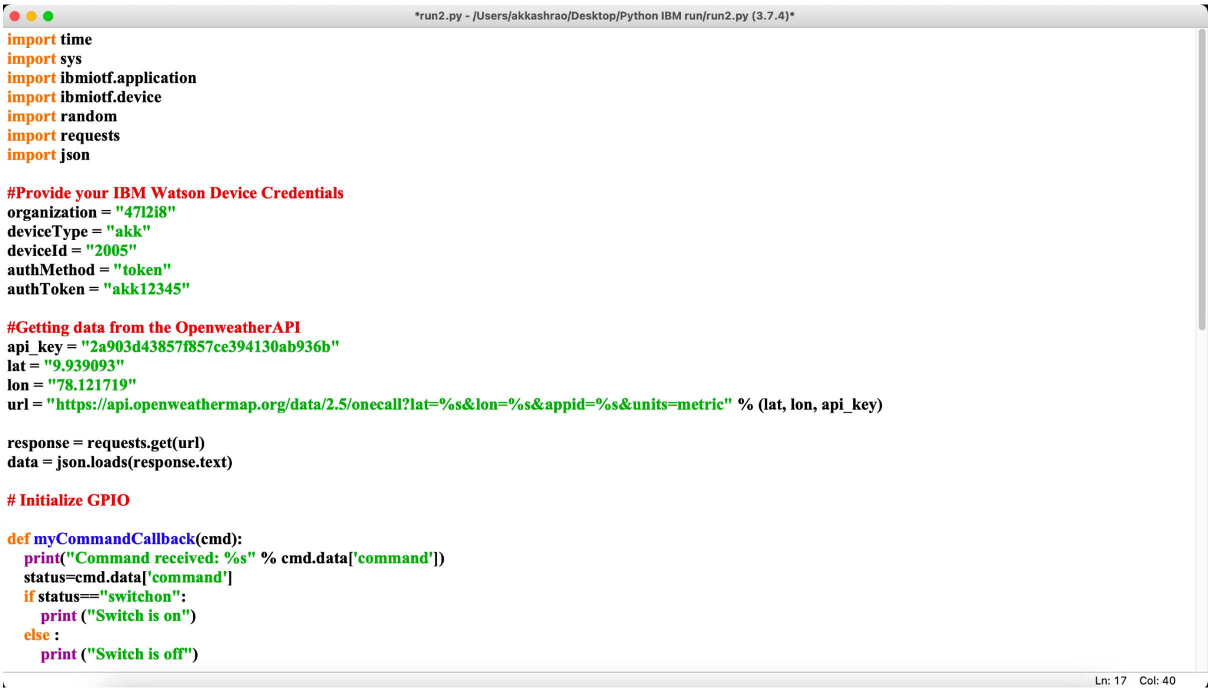
Python Client for IBM Watson IoT Platform

Install this package : Python Client for IBM Watson IoT Platform

Python code:

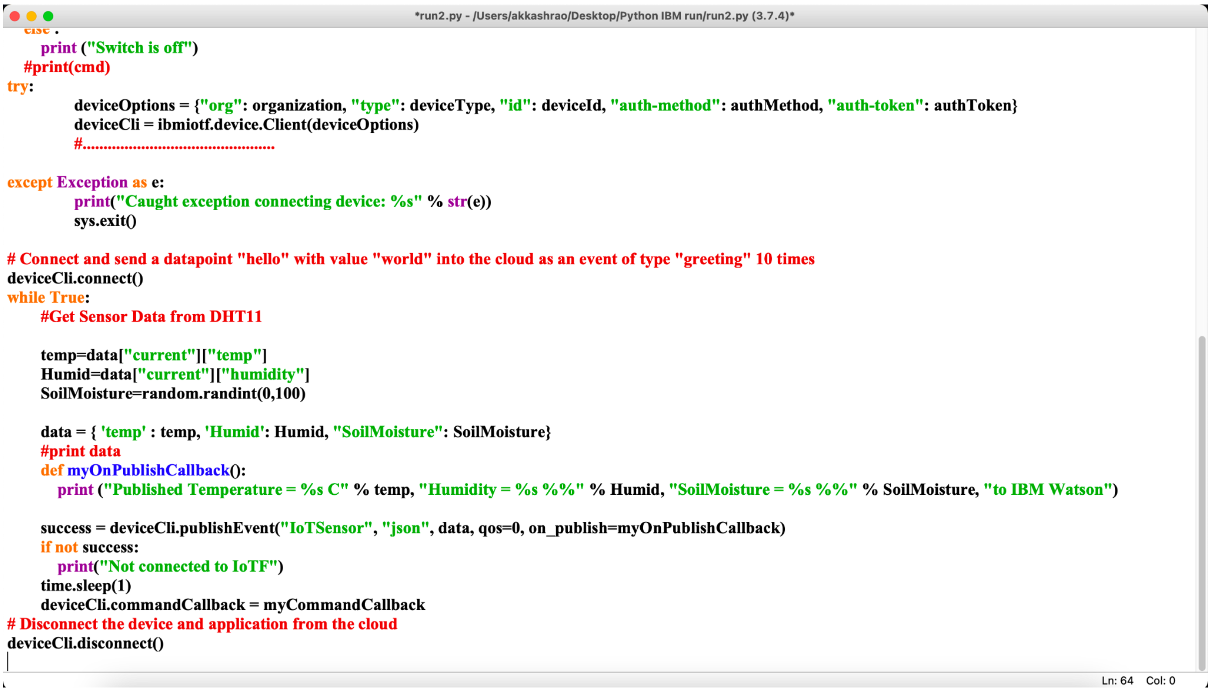
```
import time
import sys
import ibmiotf.application
import ibmiotf.device
import random
import requests
import json

#Provide your IBM Watson Device Credentials
organization = "47l2i8"
deviceType = "akk"
deviceId = "2005"
authMethod = "token"
authToken = "confidential"

#Getting data from the OpenweatherAPI
api_key = "confidential"
lat = "9.939093"
lon = "78.121719"
url =
"https://api.openweathermap.org/data/2.5/onecall?lat=%s&lon=%s&appid=%s&units
=metric" % (lat, lon, api_key)

response = requests.get(url)
data = json.loads(response.text)

# Initialize GPIO

def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    status=cmd.data['command']
    if status=="switchon":
        print ("Switch is on")
    else :
        print ("Switch is off")

    #print(cmd)

try:
          deviceOptions = {"org": organization, "type": deviceType, "id": deviceId,
"auth-method": authMethod, "auth-token": authToken}
          deviceCli = ibmiotf.device.Client(deviceOptions)
          #............................................

except Exception as e:
          print("Caught exception connecting device: %s" % str(e))
          sys.exit()

# Connect and send a datapoint "hello" with value "world" into the cloud as an event
of type "greeting" 10 times
deviceCli.connect()

while True:
        #Get Sensor Data from DHT11

        temp=data["current"]["temp"]
        Humid=data["current"]["humidity"]
        SoilMoisture=random.randint(0,100)

        data = { 'temp' : temp, 'Humid': Humid, "SoilMoisture": SoilMoisture}

        #print data
        def myOnPublishCallback():
            print ("Published Temperature = %s C" % temp, "Humidity = %s %%" %
Humid, "SoilMoisture = %s %%" % SoilMoisture, "to IBM Watson")

        success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0,
on_publish=myOnPublishCallback)
        if not success:
            print("Not connected to IoTF")
        time.sleep(1)

        deviceCli.commandCallback = myCommandCallback
```

```
# Disconnect the device and application from the cloud
deviceCli.disconnect()
```



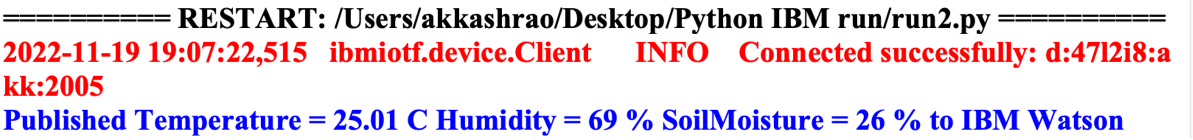
run2.py - /Users/akkashrao/Desktop/Python IBM run/run2.py (3.7.4)
Ln: 17   Col: 40

```
import time
import sys
import ibmiotf.application
import ibmiotf.device
import random
import requests
import json

#Provide your IBM Watson Device Credentials
organization = "47l2i8"
deviceType = "akk"
deviceId = "2005"
authMethod = "token"
authToken = "akk12345"

#Getting data from the OpenweatherAPI
api_key = "2a903d43857f857ce394130ab936b"
lat = "9.939093"
lon = "78.121719"
url = "https://api.openweathermap.org/data/2.5/onecall?lat=%s&lon=%s&appid=%s&units=metric" % (lat, lon, api_key)

response = requests.get(url)
data = json.loads(response.text)

# Initialize GPIO

def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    status=cmd.data['command']
    if status=="switchon":
        print ("Switch is on")
    else :
        print ("Switch is off")
```



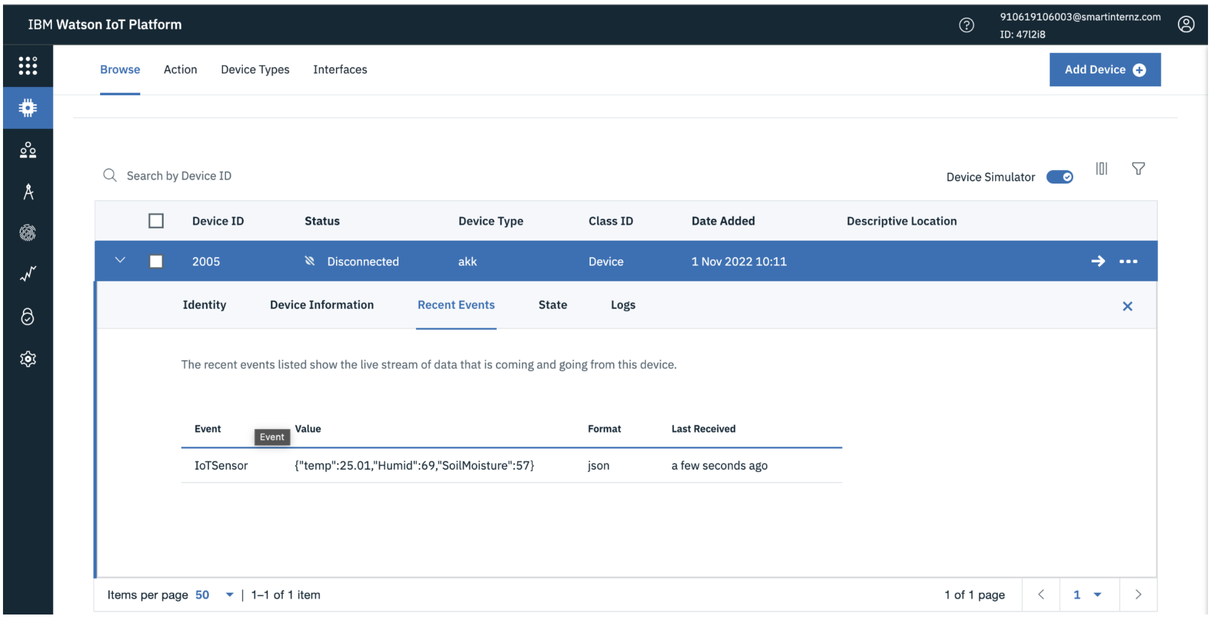
run2.py - /Users/akkashrao/Desktop/Python IBM run/run2.py (3.7.4)
Ln: 64   Col: 0

```
    else :
        print ("Switch is off")
    #print(cmd)
try:
                deviceOptions = {"org": organization, "type": deviceType, "id": deviceId, "auth-method": authMethod, "auth-token": authToken}
                deviceCli = ibmiotf.device.Client(deviceOptions)
                #..............................................

except Exception as e:
                print("Caught exception connecting device: %s" % str(e))
                sys.exit()

# Connect and send a datapoint "hello" with value "world" into the cloud as an event of type "greeting" 10 times
deviceCli.connect()
while True:
        #Get Sensor Data from DHT11

        temp=data["current"]["temp"]
        Humid=data["current"]["humidity"]
        SoilMoisture=random.randint(0,100)

        data = { 'temp' : temp, 'Humid': Humid, "SoilMoisture": SoilMoisture}
        #print data
        def myOnPublishCallback():
            print ("Published Temperature = %s C" % temp, "Humidity = %s %%" % Humid, "SoilMoisture = %s %%" % SoilMoisture, "to IBM Watson")

        success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0, on_publish=myOnPublishCallback)
        if not success:
            print("Not connected to IoTF")
        time.sleep(1)
        deviceCli.commandCallback = myCommandCallback
# Disconnect the device and application from the cloud
deviceCli.disconnect()
```



```
========== RESTART: /Users/akkashrao/Desktop/Python IBM run/run2.py ==========
2022-11-19 19:07:22,515   ibmiotf.device.Client     INFO    Connected successfully: d:47l2i8:a
kk:2005
Published Temperature = 25.01 C Humidity = 69 % SoilMoisture = 26 % to IBM Watson
```

Recent Events in IBM Watson IoT Platform:



□	Device ID	Status	Device Type	Class ID	Date Added	Descriptive Location
□	2005	Disconnected	akk	Device	1 Nov 2022 10:11	

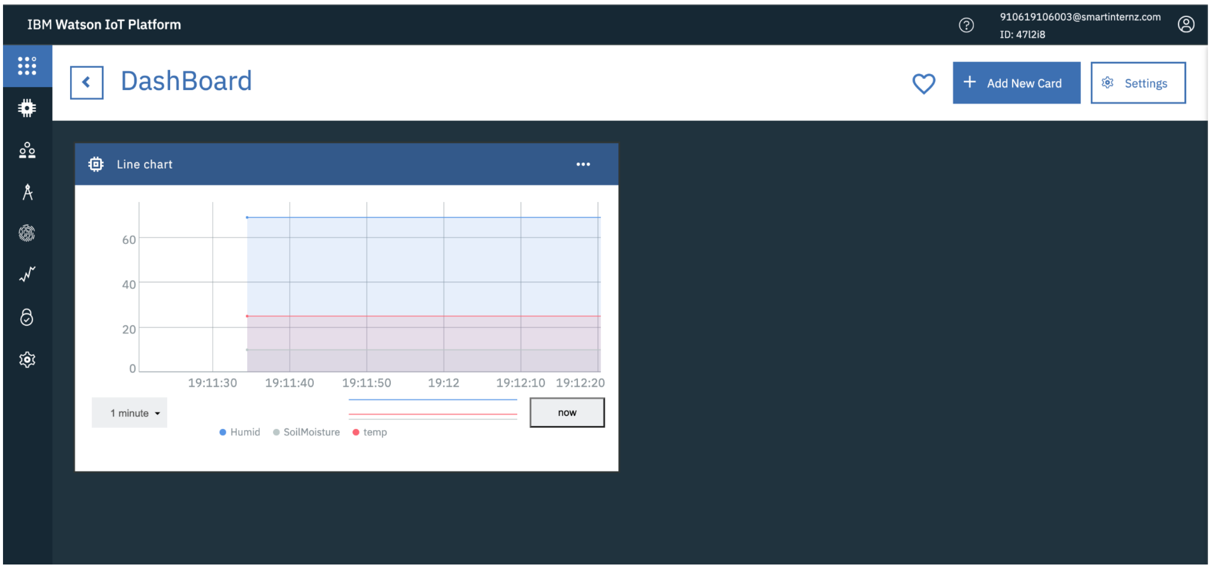
The recent events listed show the live stream of data that is coming and going from this device.

Event	Value	Format	Last Received
IoTSensor	{"temp":25.01,"Humid":69,"SoilMoisture":57}	json	a few seconds ago

Items per page 50 | 1–1 of 1 item

1 of 1 page 1

Boards in IBM Platform:


IBM Watson IoT Platform
910619106003@smartinternz.com
ID: 4712i8
DashBoard
Add New Card
Settings
Line chart
60
40
20
0
19:11:30
19:11:40
19:11:50
19:12
19:12:10
19:12:20
1 minute
now
Humid
SoilMoisture
temp