DEVELOP A PYTHON SCRIPT TO PUBLISH AND SUBSCRIBE TO IBM IOT PLATFORM

Team id : PNT2022TMID53925

Project Name : Gas Leakage Monitoring and Alerting System

Develop python code :

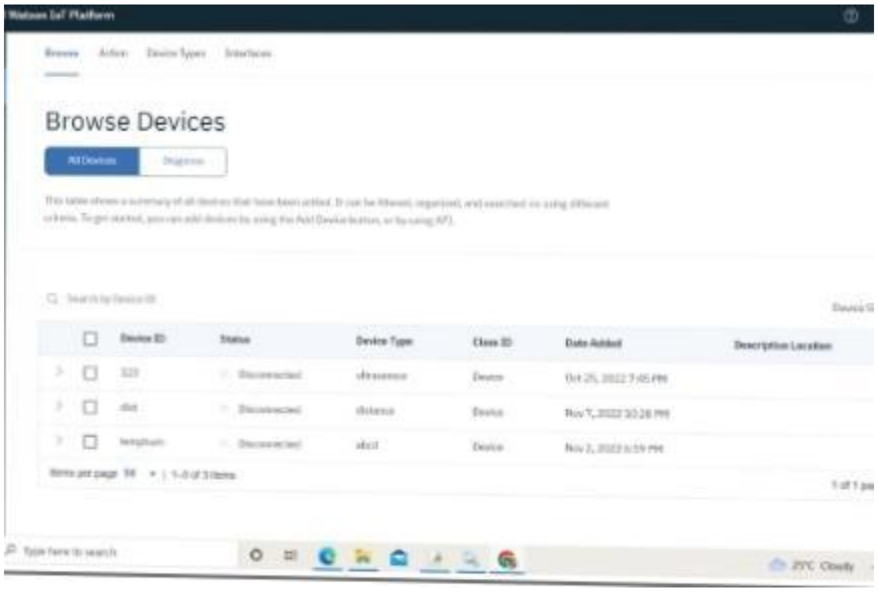
```
import time import sys

import ibmiotf.application

import ibmiotf.device

import random




#Provide your IBM Watson Device Credentials

organization = "u9pz01" deviceType = "abcd"

deviceId = "temphum" authMethod = "token"

authToken = "12345678"


# Initialize GPIO


def myCommandCallback(cmd):

   print("Command received: %s" % cmd.data['command'])

   status=cmd.data['command']

if status=="lighton":        print

("led is on")     elif status
```

```
=="lightoff":        print ("led is
off")    else:
        print("please send proper command")

    #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=random.randint(90,110)
        Humid=random.randint(60,100)

        data = { 'temp' : temp, 'Humid': Humid }
        #print data        def
myOnPublishCallback():
```

```
            print ("Published Temperature = %s C" % temp, "Humidity = %s %%" % Humid, "to
IBM Watson")


        success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0,
on_publish=myOnPublishCallback)

        if not success:

            print("Not connected to IoTF")

time.sleep(10)


        deviceCli.commandCallback = myCommandCallback


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

Publish data to IBM Cloud:

Step 1 : Open IBM WATSON IOT PLATFORM from I5BM catalog.



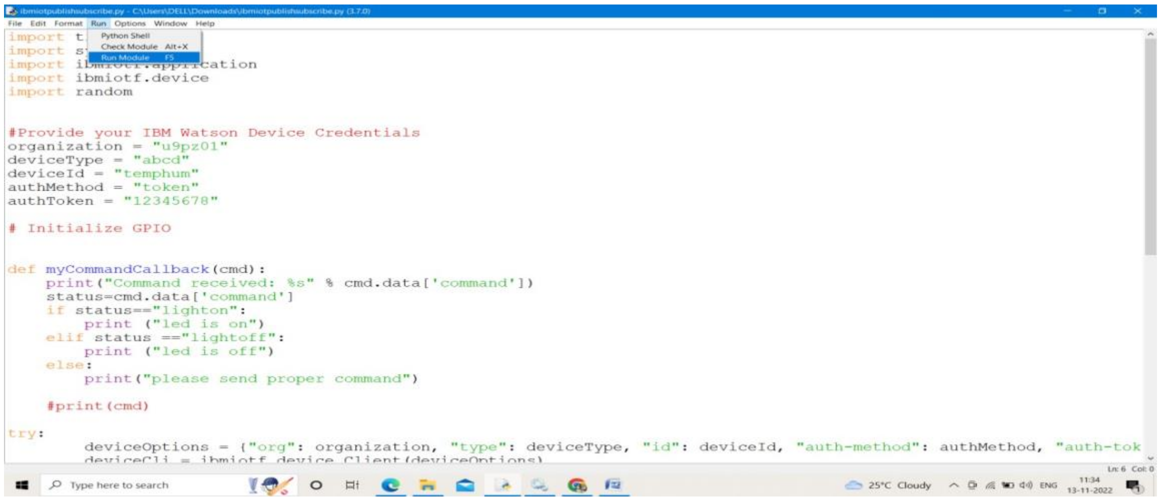
Watson IoT Platform

Browse Devices

	Device ID	Status	Device Type	Class ID	Date Added	Description Location
	123	Disconnected	ultrasonic	Device	Oct 25, 2022 7:05 PM	
	dist	Disconnected	distance	Device	Nov 7, 2022 10:28 PM	
	temphum	Disconnected	abcd	Device	Nov 2, 2022 6:59 PM	

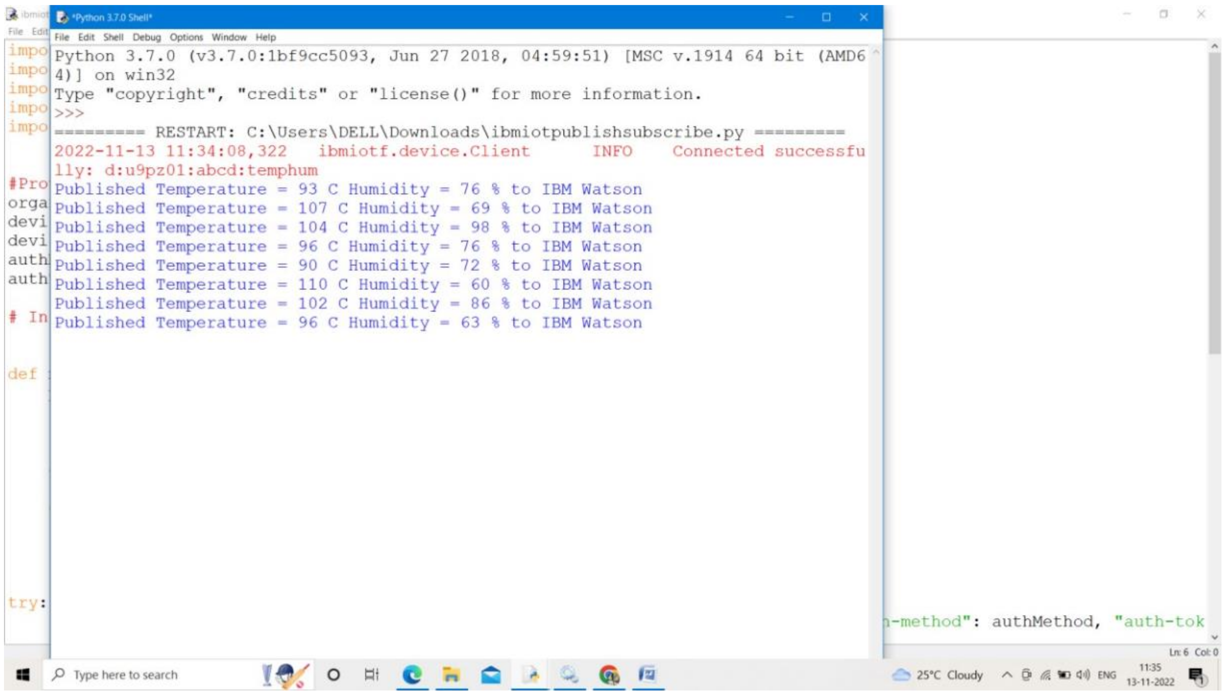
Items per page 10 | 1–3 of 3 items

Step 2 : Open IDLE Python 3.7.0 and Run the Python code.


ibmiotpublishsubscribe.py - C:\Users\DELL\Downloads\ibmiotpublishsubscribe.py (3.7.0)
File Edit Format Run Options Window Help
Python Shell
Check Module Alt+X
Run Module F5

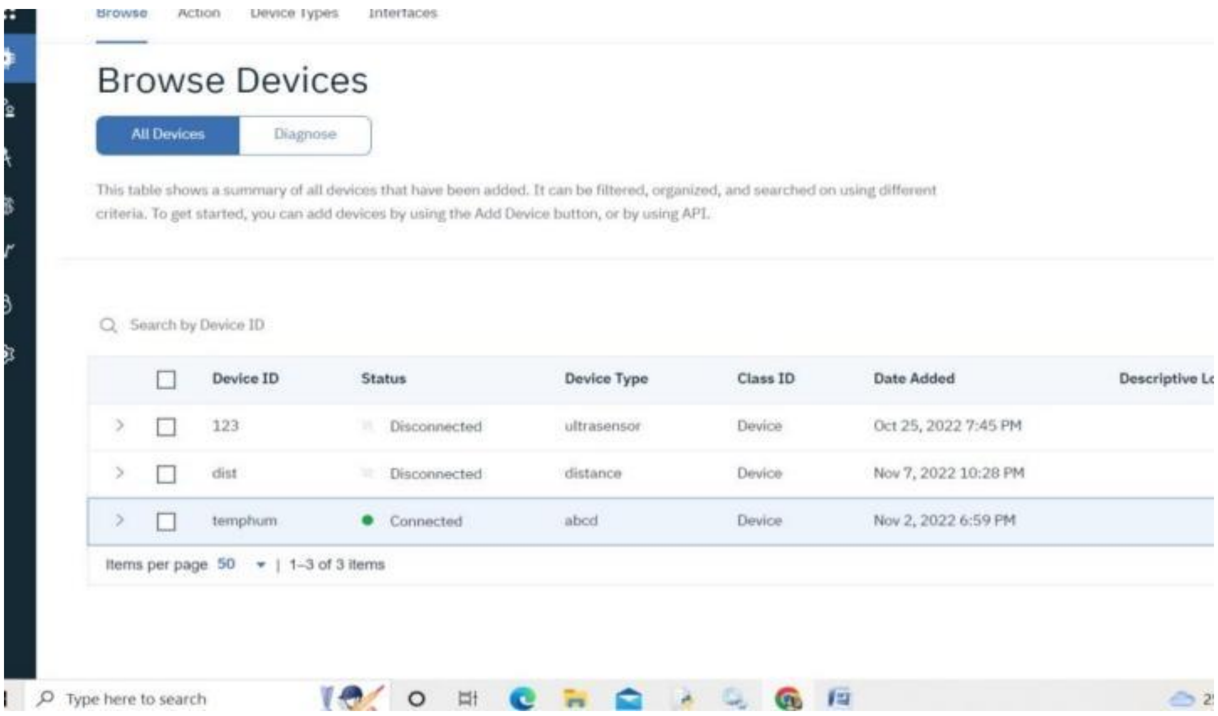
```
import t
import s
import ibmiotf.application
import ibmiotf.device
import random


#Provide your IBM Watson Device Credentials
organization = "u9pz01"
deviceType = "abcd"
deviceId = "temphum"
authMethod = "token"
authToken = "12345678"

# Initialize GPIO


def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    status=cmd.data['command']
    if status=="lighton":
        print ("led is on")
    elif status =="lightoff":
        print ("led is off")
    else:
        print("please send proper command")

    #print(cmd)

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

Step 3 : The random values for Temperature and Humidity are produced in the output. And the data is send to the IBM Watson IOT Platform.



```
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:59:51) [MSC v.1914 64 bit (AMD6
4)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
========= RESTART: C:\Users\DELL\Downloads\ibmiotpublishsubscribe.py =========
2022-11-13 11:34:08,322   ibmiotf.device.Client      INFO    Connected successfu
lly: d:u9pz01:abcd:temphum
Published Temperature = 93 C Humidity = 76 % to IBM Watson
Published Temperature = 107 C Humidity = 69 % to IBM Watson
Published Temperature = 104 C Humidity = 98 % to IBM Watson
Published Temperature = 96 C Humidity = 76 % to IBM Watson
Published Temperature = 90 C Humidity = 72 % to IBM Watson
Published Temperature = 110 C Humidity = 60 % to IBM Watson
Published Temperature = 102 C Humidity = 86 % to IBM Watson
Published Temperature = 96 C Humidity = 63 % to IBM Watson
```

Step 4 : In IBM Watson IOT Platform the status shows connected when the python code is made to run.



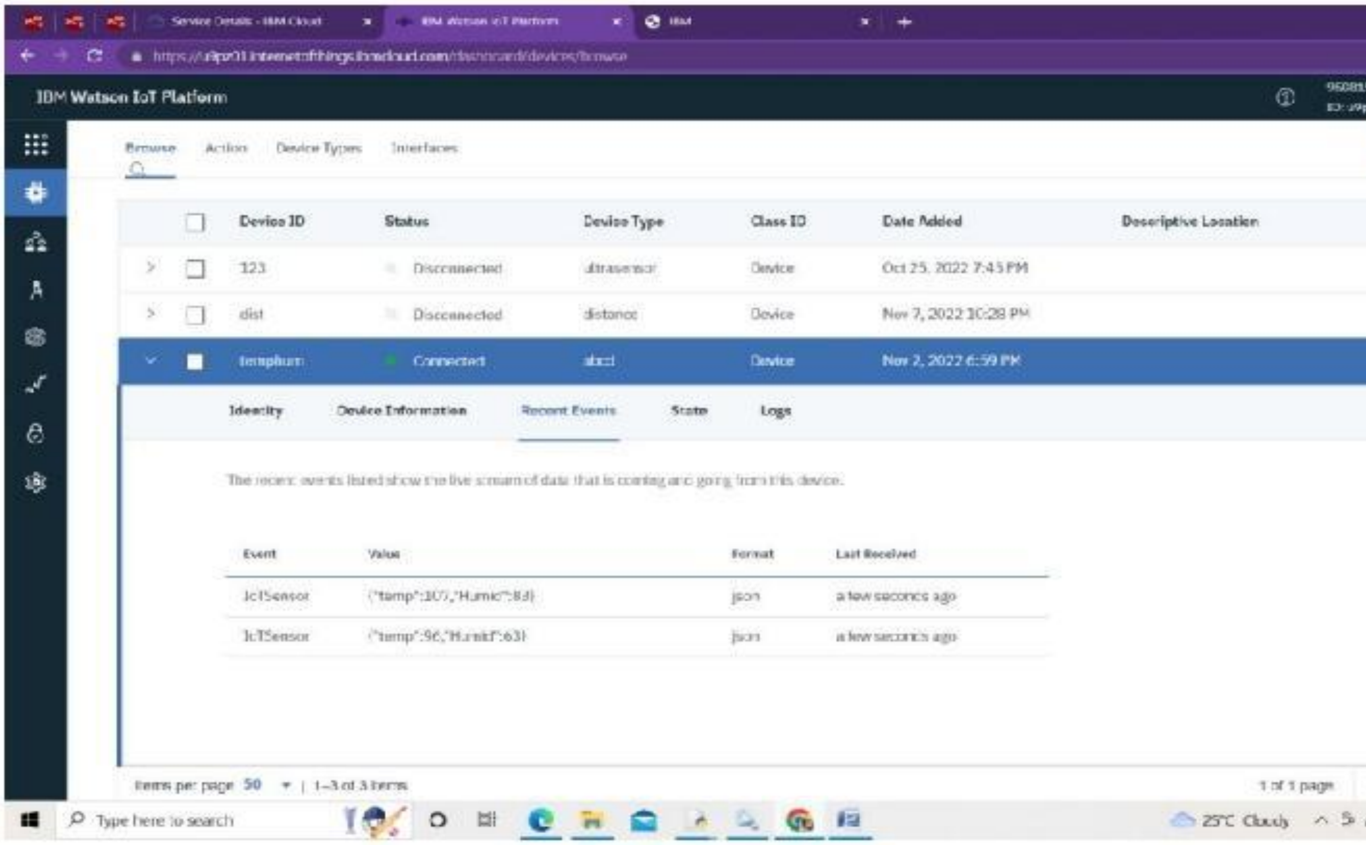
Browse Devices

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
123	Disconnected	ultrasensor	Device	Oct 25, 2022 7:45 PM
dist	Disconnected	distance	Device	Nov 7, 2022 10:28 PM
temphum	● Connected	abcd	Device	Nov 2, 2022 6:59 PM

Items per page 50 | 1–3 of 3 items

Step 5 : On clicking Recent Events we can see the Temperature and Humidity values from Python code is published to the IBM Watson IOT Platform.



Device ID	Status	Device Type	Class ID	Date Added	Descriptive Location
123	Disconnected	ultrasensor	Device	Oct 25, 2022 7:45 PM	
dist	Disconnected	distance	Device	Nov 7, 2022 10:28 PM	
temphum	Connected	abcd	Device	Nov 2, 2022 6:59 PM	

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":107,"Humid":83}	json	a few seconds ago
IoTSensor	{"temp":96,"Humid":63}	json	a few seconds ago

Items per page 50 | 1–3 of 3 items 1 of 1 page