

PYTHON CODE (GAS, TEMPERATURE, HUMIDITY, PRESSURE)

Date	16 NOVEMBER 2022
Team ID	PNT2022TMID06193
Project Name	GAS LEAKAGE MONITORING AND ALERTING SYSTEM FOR INDUSTRIES

PYTHON CODE

```
#IBM Watson IOT Platform
import wiotp.sdk.device
import time
import random
myConfig = {
"identity": {
    "orgId": "yf0dyy ",
    "typeId": "Faraaz ",
    "deviceId": "12345"
},
"auth": {
    "token": "VJTDPRX@f&4Vuox8ms "
}
}

def myCommandCallback(cmd):
print("Message received from IBM IoT Platform: %s" %
cmd.data['command'])
m=cmd.data['command']
client = wiotp.sdk.device.DeviceClient(config=myConfig,
logHandlers=None)
client.connect()
```

```

while True:

    gas=random.randint(0,100)

    temp=random.randint(0,100)

    hum=random.randint(0,100)

    pre=random.randint(0,100)

    myData={'Hazardous Gas':gas, 'Temperature':temp, 'Humidity':hum,
    'Pressure':pre }

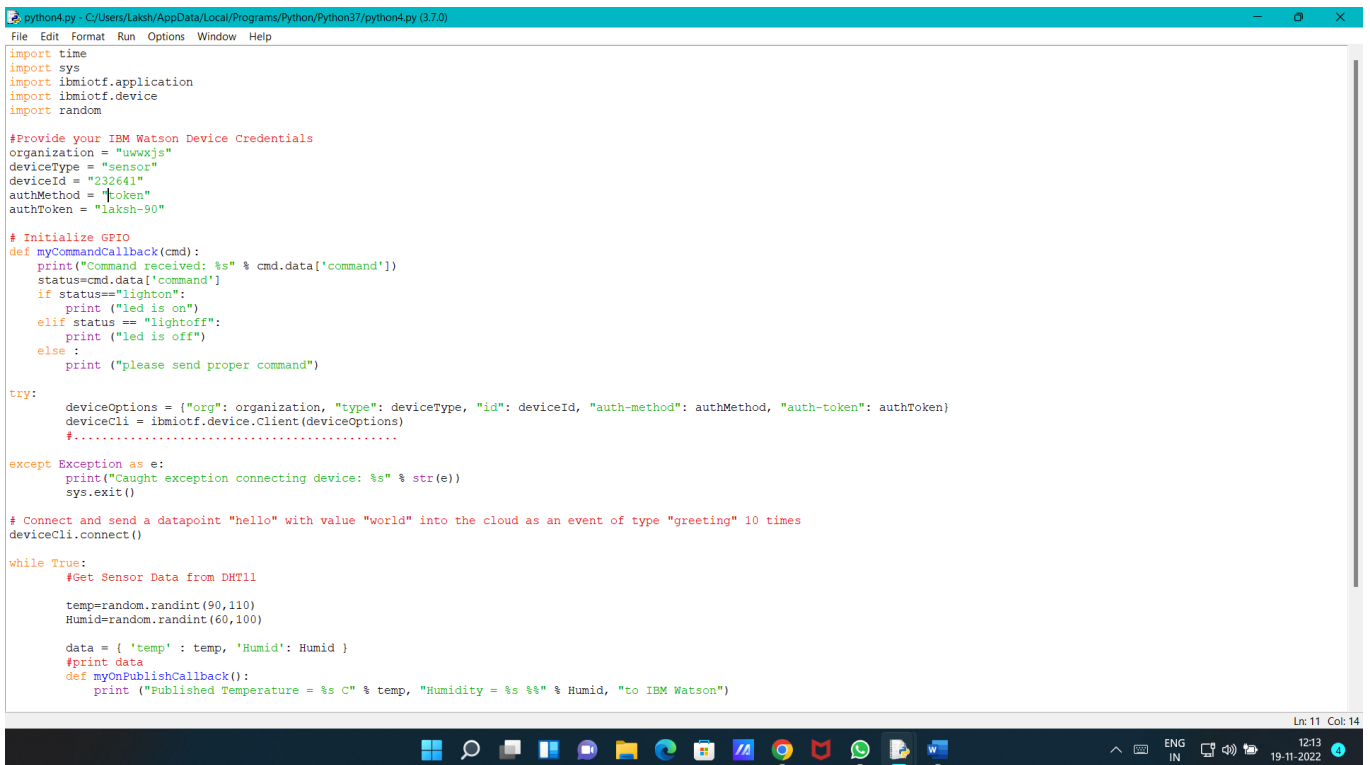
    client.publishEvent(eventId="status", msgFormat="json",
    data=myData,qos=0, onPublish=None)

    print("Published data Successfully:  %s",
    myData)      client.commandCallback      =
    myCommandCallbacktime.sleep(2)

client.disconnect()

```

OUTPUT:



```

python4.py - C:/Users/Laksh/AppData/Local/Programs/Python/Python37/python4.py (3.7.0)
File Edit Format Run Options Window Help

import time
import sys
import ibmiotf.application
import ibmiotf.device
import random

#Provide your IBM Watson Device Credentials
organization = "uwxkjs"
deviceType = "sensor"
deviceId = "232641"
authMethod = "token"
authToken = "laksh-90"

# 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")

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")

```

IBM Watson IoT Platform

91762014215@smartinternz.com
ID: uwwxjs

Search by Device ID

Device Simulator

Device ID	Status	Device Type	Class ID	Date Added	Descriptive Location
232641	Connected	sensor	Device	Nov 19, 2022 10:25 AM	

Identity Device Information Recent Events State Logs

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":95,"Humid":86}	json	34 minutes ago
IoTSensor	{"temp":94,"Humid":79}	json	35 minutes ago

Items per page 50 | 1-1 of 1 item

0 Simulations running

Node-RED

Flow 1

lakh:Hello Node-RED!

IBM IoT

msg.payload

debug

```
11/19/2022, 12:14:19 PM node: f2f2649a.0d0d98  
iot-2/type/sensor/id/232641/ev/IoTSensor/rmt/json :  
msg.payload : Object  
  { temp: 96, Humid: 100 }  
11/19/2022, 12:14:29 PM node: f2f2649a.0d0d98  
iot-2/type/sensor/id/232641/ev/IoTSensor/rmt/json :  
msg.payload : Object  
  { temp: 106, Humid: 89 }  
11/19/2022, 12:14:39 PM node: f2f2649a.0d0d98  
iot-2/type/sensor/id/232641/ev/IoTSensor/rmt/json :  
msg.payload : Object  
  { temp: 101, Humid: 78 }  
11/19/2022, 12:14:49 PM node: f2f2649a.0d0d98  
iot-2/type/sensor/id/232641/ev/IoTSensor/rmt/json :  
msg.payload : Object  
  { temp: 105, Humid: 74 }  
11/19/2022, 12:14:59 PM node: f2f2649a.0d0d98  
iot-2/type/sensor/id/232641/ev/IoTSensor/rmt/json :  
msg.payload : Object  
  { temp: 102, Humid: 91 }  
11/19/2022, 12:15:09 PM node: f2f2649a.0d0d98  
iot-2/type/sensor/id/232641/ev/IoTSensor/rmt/json :  
msg.payload : Object  
  { temp: 105, Humid: 96 }
```