PUBLISH PYTHON

Date	16 Nov 2022
Team ID	PNT2022TMID14391
Project Name	Industry - specific intelligent management system

python.py - C:/Users/HP/Desktop/python.py (3.7.0)

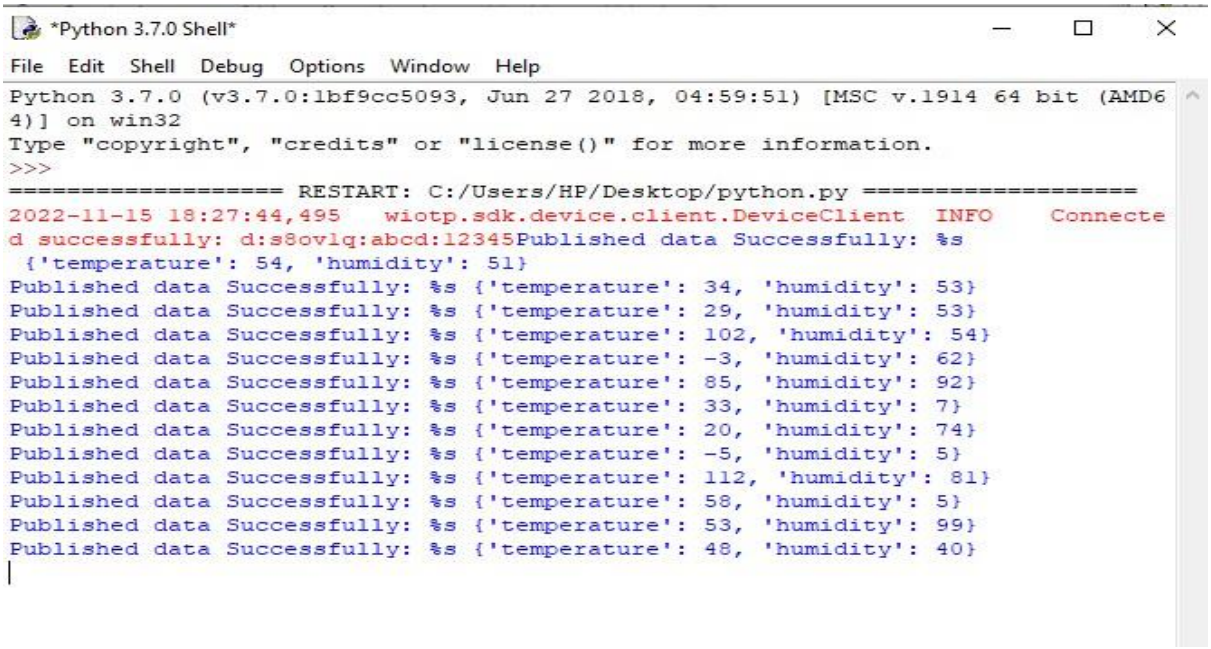
File Edit Format Run Options Window Help

```
#IBM Watson IOT Platform
#pip install wiotp-sdk
import wiotp.sdk.device
import time
import random
myConfig = {
    "identity": {
        "orgId": "s8ovlq",
        "typeId": "abcd",
        "deviceId":"12345"
    },
    "auth": {
        "token": "12345678"
    }
}

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:
    temp=random.randint(-20,125)
    hum=random.randint(0,100)
    myData={'temperature':temp, 'humidity':hum}
    client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0, onPublish=None)
    print("Published data Successfully: %s", myData)
    client.commandCallback = myCommandCallback
    time.sleep(2)
client.disconnect()
```



code

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


#Provide your IBM Watson Device Credentials
organization = "s8ov1q" deviceType = "abcd" deviceId =
"12345" 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")


try:
```

```
        deviceOptions = {"org": organization, "type": deviceType, "id": deviceId, "authmethod":
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()
```