

Python code to publish and subscribe to IBM IOT platform:

```
import wiotp.sdk.device
```

```
import time
```

```
import os
```

```
import datetime
```

```
import random
```

```
myConfig = {
```

```
    "identity": {
```

```
        "orgId" : "xcjqgo",
```

```
        "typeId" : "ESP32",
```

```
        "deviceId" : "IBM17"
```

```
    },
```

```
    "auth": {
```

```
        "token" : "12345678"
```

```
    }
```

```
}
```

```
client=wiotp.sdk.device.DeviceClient(config=myConfig,logH  
ndlers=None)
```

```
client.connect()
```

```
def myCommandCallback(cmd):
```

```
    print("Message received from IBM IoT platform: %s" %  
cmd.data ['command'])
```

```
m=cmd.data['command']
if(m=="motoron"):
    print("motor is switched on")
elif(m=="motoroff"):
    print("motor is switched off")
print(" ")
while True:
    soil=random.randint(0,100)
    temp=random.randint(-20,125)
    hum=random.randint(0,100)

    myData={'soil_moisture':soil,'temperature':temp,'humidity':hum}

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

    print("published data successfully: %s",myData)

    time.sleep(2)

    client.commandCallback=myCommandCallback
client.disconnect()
```

```
pythonProject - IBM.py
1 import wiotp.sdk.device
2 import time
3 import os
4 import datetime
5 import random
6 myConfig = {
7     "identity": {
8         "angle": "xcl99g",
9         "typeId": "ESP32",
10        "deviceId": "IBM17",
11    },
12    "auth": {
13        "token": "12345678"
14    }
15 }
16 client=wiotp.sdk.device.DeviceClient(config=myConfig,loghandlers=None)
17 client.connect()
18 def myCommandCallback(cmd):
19     print("Message received from IBM IoT platform: %s" % cmd.data["command"])
20     m=cmd.data["command"]
21     if(m=="motoron"):
22         print("motor is switched on")
23     elif(m=="motoroff"):
24         print("motor is switched off")
25     print(' ')
26 while True:
27     soil=random.randint(0,100)
28     temp=random.randint(-20,125)
29     hum=random.randint(0,100)
30     myData={"soil_moisture":soil,"temperature":temp,"humidity":hum}
31     client.publishEvent(eventId="status",contentType="json",data=myData,qos=0,onPublish=None)
32     print("published data successfully: %s"%myData)
33     time.sleep(2)
34     client.commandCallback=myCommandCallback
35     client.disconnect()
```

IBM Watson IoT Platform

Browse

Action

Device Types

Interfaces

All Devices

Diagnose

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	Descriptive Location	Added By	Device Class
12345678	Disconnected	ESP32	Device	Nov 17, 2022 10:57 AM		anbu8428@gmail.com	
IBM17	Disconnected	ESP32	Device	Nov 9, 2022 10:42 AM		anbu8428@gmail.com	

Identity

Device Information

Recent Events

State

Logs

Device ID

Device Type

Date Added

Added By

Connection Status

IBM17

ESP32

Nov 9, 2022 10:42 AM

anbu8428@gmail.com

Disconnected

Items per page 50 | 1-2 of 2 items

1 of 1 page

0 Simulations running