

DEVELOP PYTHON SCRIPT

```
ibm.py - C:\Users\manur\OneDrive\Desktop\ibm.py (3.7.0)
File Edit Format Run Options Window Help
import ibmiotf.application
import ibmiotf.device
import time
import random
import sys

organization = "12s377"
deviceType = "Raspberry"
deviceId = "123"
authMethod = "token"
authToken = "12345678"

moisture= random.randint(1, 14)
Humid = random.randint(1, 1000)
temp = random.randint(0, 100)

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()

deviceCli.connect()

while True:
    moisture= random.randint(1, 14)
    hum = random.randint(90, 110)
```

Ln: 1 Col: 0

```
ibm.py - C:\Users\manur\OneDrive\Desktop\ibm.py (3.7.0)
File Edit Format Run Options Window Help
    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()

deviceCli.connect()

while True:
    moisture= random.randint(1, 14)
    hum = random.randint(90, 110)
    temp = random.randint(0, 100)

    data = { 'hum': hum, 'temp': temp}

    def myOnPublishCallback():
        print("Published hum=%s" % hum, "temp:%s" % temp)

    success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0, on_publish=myOnPublishCallback)
    if not success:
        print("Not Connected to ibmiot")
        time.sleep(5)
    deviceCli.commandCallback = myCommandCallback

deviceCli.disconnect()
```

Ln: 1 Col: 0