

PUBLISH DATA TO IBM CLOUD

```
import sys

import ibmiotf.application
import ibmiotf.device

import random


organisation = "4qbk92"

deviceType="rasperrypi"

deviceId ="12345"

authMethod="use-token-auth"

authToken ="123456789"


def myCommandCallback(cmd):

    print("Command recieved :%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":organisation ,"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))
```

```
deviceCli.connect()
```

```
while True:
```

```
    temp=random.randint(90,100)
```

```
    humid=random.randint(60,100)
```

```
    north=0
```

```
    south=0
```

```
    east=0
```

```
    west=10
```

```
    data ={'temp':temp , 'humidity':humid, 'North':north, 'South':south, 'East':east, 'West':west}
```

```
    def myOnPublishCallback():
```

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

```
success=deviceCli.publishEvent("IOTSENSOR","json",data,qos=0,on_publish=myOnPublishCa  
llback)
```

```
    if not success:
```

```
        print("Not connected to IoTF")
```

```
    time.sleep(10)
```

```
    deviceCli.commandCallback= myCommandCallback
```

```
deviceCli.disconnect()
```

main.py - D:\IBM NALAIYA THIRAN\PYTHON\main.py (3.10.8)

File Edit Format Run Options Window Help

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

organisation = "4gbk92"
deviceType="rasperrypi"
deviceId ="12345"
authMethod="use-token-auth"
authToken ="123456789"

def myCommandCallback(cmd):
    print("Command recieved :%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":organisation ,"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))

deviceCli.connect()

while True:
    temp=random.randint(90,100)
    humid=random.randint(60,100)
    north=0
    south=0
    east=0
    west=10

    data ={'temp':temp , 'humidity':humid, 'North':north, 'South':south, 'East':east, 'West':west}

    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

deviceCli.disconnect()
```

The screenshot shows the IBM Watson IoT Platform dashboard. The browser address bar displays the URL: 4gbk92.internetofthings.ibmcloud.com/dashboard/devices/browse. The dashboard header includes the IBM Watson IoT Platform logo and navigation tabs: Browse, Action, Device Types, and Interfaces. A sidebar on the left contains icons for various dashboard features. The main content area shows a modal window titled "Recent Events" with a close button (X). Inside the modal, a message states: "The recent events listed show the live stream of data that is coming and going from this device." Below this message is a table with the following columns: Event, Value, Format, and Last Received. The table contains five rows of data, all with the same values: Event: Data, Value: {"temp":24,"humidity":40,"North":0,"South":0,"E...}, Format: json, and Last Received: a few seconds ago. At the bottom of the modal, there is a summary bar showing a status icon, the ID 34567, the status "Disconnected", the device type "rasperrypi", and the label "Device".

Event	Value	Format	Last Received
Data	{"temp":24,"humidity":40,"North":0,"South":0,"E...	json	a few seconds ago
Data	{"temp":24,"humidity":40,"North":0,"South":0,"E...	json	a few seconds ago
Data	{"temp":24,"humidity":40,"North":0,"South":0,"E...	json	a few seconds ago
Data	{"temp":24,"humidity":40,"North":0,"South":0,"E...	json	a few seconds ago
Data	{"temp":24,"humidity":40,"North":0,"South":0,"E...	json	a few seconds ago