

Publish Data To the IBM Code

Date	16 November 2022
Team ID	PNT2022TMID02525
Project Name	Gas Leakage Monitoring and Alerting System
Maximum Marks	4 Marks

CODE

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

#Provide your IBM Watson Device Credentials
organization = "z0kljz"
deviceType = "OpenCV"
deviceId = "Fire_Detection"
authMethod = "token"
authToken = "Fire@123"

# 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")
```

```
        success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0,
on_publish=myOnPublishCallback)
```

```
        if not success:
```

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

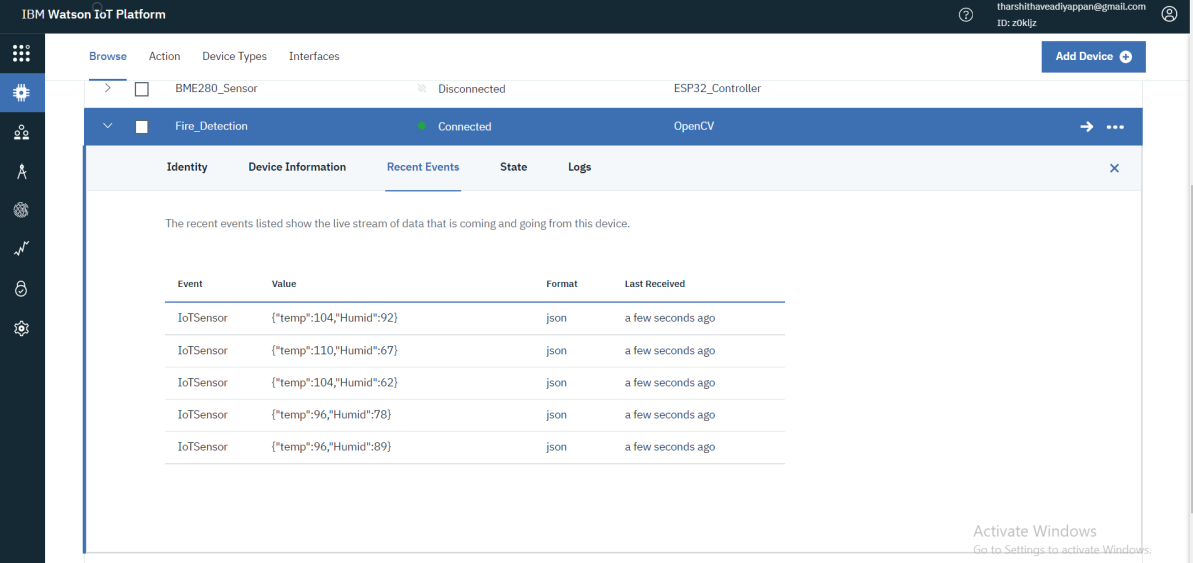
```
            time.sleep(10)
```

```
    deviceCli.commandCallback = myCommandCallback
```

```
# Disconnect the device and application from the cloud
```

```
deviceCli.disconnect()
```

OUTPUT



The screenshot displays the IBM Watson IoT Platform interface. The top navigation bar includes 'Browse', 'Action', 'Device Types', and 'Interfaces'. The main content area shows a list of devices. The 'Fire_Detection' device is selected, showing a 'Connected' status. Below the device list, the 'Recent Events' tab is active, displaying a table of events. The table has four columns: 'Event', 'Value', 'Format', and 'Last Received'. The events listed are all 'IoTSensor' events with JSON format, showing temperature and humidity data. The bottom of the interface shows a 'Temp_Humid' device with a 'Disconnected' status.

Event	Value	Format	Last Received
IoTSensor	{"temp":104,"Humid":92}	json	a few seconds ago
IoTSensor	{"temp":110,"Humid":67}	json	a few seconds ago
IoTSensor	{"temp":104,"Humid":62}	json	a few seconds ago
IoTSensor	{"temp":96,"Humid":78}	json	a few seconds ago
IoTSensor	{"temp":96,"Humid":89}	json	a few seconds ago