

SPRINT – 3

DATE	12 NOVEMBER 2022
TEAM ID	B2-2M4E
PROJECT NAME	Smart Waste Management System for Metropolitan Cities

PYTHON CODE : [To connect IBM WATSON]

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

#Provide your IBM Watson Device Credentials
organization = "zncs13"
deviceType = "SENSOR"
deviceId = "SENSOR-23"
authMethod = "use-token
auth"
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")
    else :
        print ("led is off")
#print(cmd)

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 thecloud as  
an event of type "greeting" 10 times
```

```
deviceCli.connect()
```

```
while True:
```

```
    #Get Sensor Data from DHT11
```

```
    level=random.randint(0,100)
```

```
    weight=random.randint(0,100)
```

```
    data = { 'level' : level, 'weight': weight }
```

```
    #print data
```

```
    def myOnPublishCallback():
```

```
        print ("Published level = %s C" % level, "weight = %s %% "
```

```
% weight, "to IBM Watson")
```

```
    success = deviceCli.publishEvent("IoTSensor", "json", data,
```

```
    qos=0, on_publish=myOnPublishCallback)
```

```
if not success:
```

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

```
    time.sleep(1)
```

```
    deviceCli.commandCallback =
```

```
myCommandCallbackif (level>=75):
```

```
    print("Full LED ON")
```

```
# Disconnect the device and application from the cloud  
deviceCli.disconnect()
```

OUTPUT :

The screenshot displays the IBM Watson IoT Platform interface. The main view shows a table of recent events for a device named 'abcd'. A modal window is open for configuring a new event type.

Event	Status	Format	Age
eventabcd11	[{"variableNumber": 5, "level": 47, "weight": 947}]	json	4 m
eventabcd11	[{"variableNumber": 27, "level": 6, "weight": 275}]	json	4 m
eventabcd11	[{"variableNumber": 80, "level": 13, "weight": 605}]	json	2 m
eventabcd11	[{"variableNumber": 34, "level": 34, "weight": 224}]	json	3 m
eventabcd11	[{"variableNumber": 45, "level": 54, "weight": 405}]	json	8 m

Items per page: 10 → 1-1 of 1 item

Device Type: abcd

Events New event type

Event type name: eventabcd11 Send

Schedule: Every Minute

Repeat: Specify the event payload in the editor window or by uploading a CSV file.

```

0 {
1   "variableNumber": "variable(0, 100)"
2   "level": "variable(0, 100)"
3   "weight": "variable(0, 1000)"
4 }
5
  
```

Upload a CSV file Cancel Save

```

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

#Provide your IBM Watson Device Credentials
organization = "cbsej"
deviceType = "abcd"
deviceId = "1234"
authMethod = "token"
authToken = "12345678"

# Initialize GPIO

def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data["command"])
    status=cmd.data["command"]
    if status=="lights":
        print ("led is on")
    else:
        print ("led is off")

    #print(cmd)

try:
    deviceOptions = {"org": organization, "type": de
  
```

```

Python 3.7.0 Shell
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:59:51) [MSC v.1914 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
- RESTART: C:\Users\navar\Dropbox\PC\Downloads\ibmiotpublishsubscribe (1).py -
2022-11-13 11:52:44,654 ibmiotf.device.Client INFO Connected successfully: d:cbsej;abc
d:1234
Published level = 82 C weight = 64 % to IBM Watson
Full LED ON
Published level = 5 C weight = 2 % to IBM Watson
Published level = 22 C weight = 57 % to IBM Watson
Published level = 83 C weight = 69 % to IBM Watson
Full LED ON
Published level = 16 C weight = 12 % to IBM Watson
Published level = 19 C weight = 91 % to IBM Watson
Published level = 35 C weight = 77 % to IBM Watson
Published level = 22 C weight = 46 % to IBM Watson
Published level = 85 C weight = 68 % to IBM Watson
Full LED ON
Published level = 36 C weight = 88 % to IBM Watson
Published level = 69 C weight = 72 % to IBM Watson
Published level = 14 C weight = 3 % to IBM Watson
Published level = 99 C weight = 0 % to IBM Watson
  
```


