

## PUBLISH DATA TO THE IBM CLOUD

Date	31 October 2022
Team ID	Based PNT2022TMID12751
Project Name	IoT based Smart Waste Management System For Metropolitan Cities

### Steps

Install json, wiotp, time modules in python and develop a code with corresponding credentials to publish the temperature and humidity

```
#IBM watson IOT PLATFORM
#pip install wiotp-sdk
import wiotp.sdk.device
import time
import random
myConfig = { "identity": {
    {
    "orgId": "ip9sem",
    "typeId": "smartwaste123",
    "deviceId": "76013" },
    "auth": { "token": "12345678" }
    }
}
def myCommandCallback(cmd):
    print("Message received from IBM IoT Platform: %s" % cmd.data['command'])
    m=cmd.data['command']

client = wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None)
client.connect()
while True:
    temp=random.randint(-20,125)
    hum=random.randint(0,100)
    myData={'temperature':temp, 'humidity':hum}
    client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0, onPublish=None)
    print("Published data Successfully: %s", myData)
    client.commandCallback = myCommandCallback
    time.sleep(2)

client.disconnect()
|
```

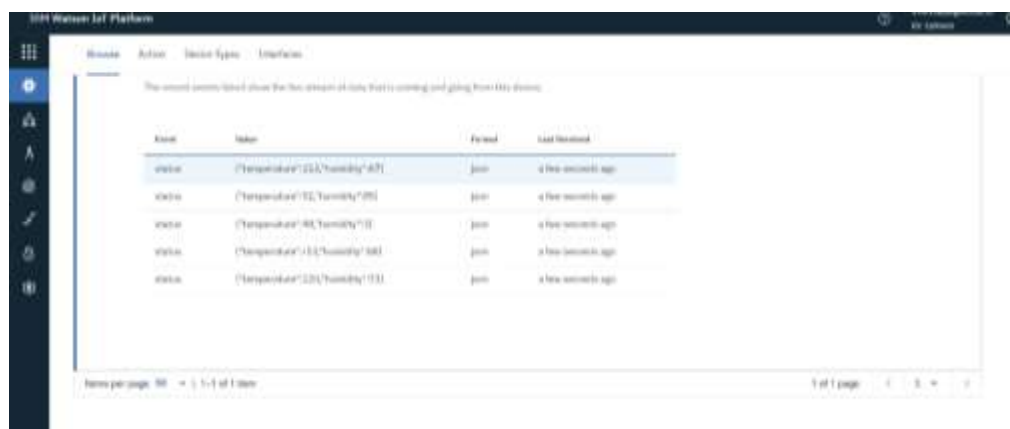
Save and Run the code therefore temperature and humidity are published in Node RED

```

Published data Successfully: %s ('temperature': 66, 'humidity': 41)
Published data Successfully: %s ('temperature': -9, 'humidity': 37)
Published data Successfully: %s ('temperature': 19, 'humidity': 83)
Published data Successfully: %s ('temperature': 52, 'humidity': 58)
Published data Successfully: %s ('temperature': 27, 'humidity': 83)
Published data Successfully: %s ('temperature': 20, 'humidity': 28)
Published data Successfully: %s ('temperature': -12, 'humidity': 52)
Published data Successfully: %s ('temperature': 64, 'humidity': 97)
Published data Successfully: %s ('temperature': -4, 'humidity': 32)
Published data Successfully: %s ('temperature': 88, 'humidity': 44)
Published data Successfully: %s ('temperature': 59, 'humidity': 8)
Published data Successfully: %s ('temperature': 57, 'humidity': 62)
Published data Successfully: %s ('temperature': 25, 'humidity': 97)
Published data Successfully: %s ('temperature': 96, 'humidity': 17)
Published data Successfully: %s ('temperature': 102, 'humidity': 7)
Published data Successfully: %s ('temperature': 75, 'humidity': 13)
Published data Successfully: %s ('temperature': 41, 'humidity': 81)
Published data Successfully: %s ('temperature': 58, 'humidity': 85)
Published data Successfully: %s ('temperature': 120, 'humidity': 73)
Published data Successfully: %s ('temperature': -13, 'humidity': 88)
Published data Successfully: %s ('temperature': 48, 'humidity': 3)
Published data Successfully: %s ('temperature': 51, 'humidity': 85)
Published data Successfully: %s ('temperature': 113, 'humidity': 67)
Published data Successfully: %s ('temperature': -7, 'humidity': 15)
Published data Successfully: %s ('temperature': 55, 'humidity': 1)
Published data Successfully: %s ('temperature': 124, 'humidity': 32)
Published data Successfully: %s ('temperature': -6, 'humidity': 88)
Published data Successfully: %s ('temperature': 92, 'humidity': 84)
Published data Successfully: %s ('temperature': 105, 'humidity': 63)
Published data Successfully: %s ('temperature': 56, 'humidity': 67)
Published data Successfully: %s ('temperature': 63, 'humidity': 29)
Published data Successfully: %s ('temperature': 122, 'humidity': 53)
Published data Successfully: %s ('temperature': 27, 'humidity': 22)
Published data Successfully: %s ('temperature': 72, 'humidity': 27)
Published data Successfully: %s ('temperature': 44, 'humidity': 5)
Published data Successfully: %s ('temperature': 42, 'humidity': 89)
Published data Successfully: %s ('temperature': 95, 'humidity': 44)
Published data Successfully: %s ('temperature': 57, 'humidity': 92)
Published data Successfully: %s ('temperature': 75, 'humidity': 63)

```

The published data (i.e.) temperature and humidity are seen in Node RED



The screenshot shows the Node-RED interface with a table of published data. The table has columns: Name, Value, Format, and Last Received. It lists four data points for 'station' with temperature and humidity values.

Name	Value	Format	Last Received
station	['temperature': 113, 'humidity': 67]	json	a few seconds ago
station	['temperature': 11, 'humidity': 85]	json	a few seconds ago
station	['temperature': 48, 'humidity': 3]	json	a few seconds ago
station	['temperature': 113, 'humidity': 67]	json	a few seconds ago
station	['temperature': 122, 'humidity': 53]	json	a few seconds ago