

SPRINT 3

Date	10 November 2022
Team ID	PNT2022TMID43127
Project Name	Smart Waste Management System for Metropolitan Cities
Story Point	20

Python code (To Connect IBM Watson)

BIN 3:

```
import wiotp.sdk.device
import time
import random

myConfig = {
    "identity": {
        "orgId": "x3lifo",
        "typeId": "Bin_3",
        "deviceId": "1234"
    },
    "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:

```
    level=random.randint(0,10)
```

```
    weight=random.randint(0,10)
```

```
    myData={ 'name': 'Bin_3', 'lat': 15.092677, 'lon': 79.188314 , 'Level':level, 'Weight':weight }
```

```
    if weight == 10:
```

```
        print ('ALERT !! Weight is HIGH')
```

```
    if level == 10:
```

```
        print ('ALERT !! Level is HIGH')
```

```
    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()
```

Output in python IDLE:

```
bin_3.py - C:/desktop/bin_3.py (3.7.4)
File Edit Format Run Options Window Help

import wiotp.sdk.device
import time
import random
myConfig = {
    "identity": {
        "orgId": "x3lifo",
        "typeId": "Bin_3",
        "deviceId": "1234"
    },
    "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:
    level=random.randint(0,10)
    weight=random.randint(0,10)
    myData={ 'name': 'Bin_3', 'lat': 15.092677, 'lon': 79.188314 , 'Level':level
    if weight == 10:
        print ('ALERT !! Weight is HIGH')
    if level == 10:
        print ('ALERT !! Level is HIGH')
    client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0,o
    print ("Published data Successfully: %s", myData)
    client.commandCallback = myCommandCallback
    time.sleep(2)
client.disconnect()

Python 3.7.4 Shell
File Edit Shell Debug Options Window Help

Python 3.7.4 (tags/v3.7.4:09359112e, Jul 8 2019, 20:34:20) [MSC v.1916 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/desktop/bin_3.py =====
2022-11-17 19:46:52,905 wiotp.sdk.device.client.DeviceClient INFO Connecte
d successfully: diox3lifo:Bin_3:1234
Published data Successfully: %s ('name': 'Bin_3', 'lat': 15.092677, 'lon': 79.18
8314, 'Level': 2, 'Weight': 7)
Published data Successfully: %s ('name': 'Bin_3', 'lat': 15.092677, 'lon': 79.18
8314, 'Level': 3, 'Weight': 5)
Published data Successfully: %s ('name': 'Bin_3', 'lat': 15.092677, 'lon': 79.18
8314, 'Level': 9, 'Weight': 2)
ALERT !! Level is HIGH
Published data Successfully: %s ('name': 'Bin_3', 'lat': 15.092677, 'lon': 79.18
8314, 'Level': 10, 'Weight': 7)
Published data Successfully: %s ('name': 'Bin_3', 'lat': 15.092677, 'lon': 79.18
8314, 'Level': 8, 'Weight': 9)
ALERT !! Level is HIGH
Published data Successfully: %s ('name': 'Bin_3', 'lat': 15.092677, 'lon': 79.18
8314, 'Level': 10, 'Weight': 2)
Published data Successfully: %s ('name': 'Bin_3', 'lat': 15.092677, 'lon': 79.18
8314, 'Level': 0, 'Weight': 3)
```

IBM Watson IoT platform:

The screenshot shows the IBM Watson IoT Platform dashboard. The top navigation bar includes 'Browse', 'Action', 'Device Types', and 'Interfaces'. A search bar is present. The main content area displays a table of devices. The selected device (ID 1234) is shown in detail, including its status (Connected), device type (Bin_3), class ID (Device), and date added (Nov 17, 2022 9:52 AM). Below the device details, there is a section for 'Recent Events' showing a live stream of data. The events table has columns for Event, Value, Format, and Last Received. The events are JSON payloads containing location data (name, lat, lon) and status information. At the bottom, it indicates '0 Simulations running'.

Device ID	Status	Device Type	Class ID	Date Added	Descriptive Location
1234	Disconnected	Bin_1	Device	Nov 16, 2022 8:03 PM	
1234	Disconnected	Bin_2	Device	Nov 16, 2022 8:04 PM	
1234	Connected	Bin_3	Device	Nov 17, 2022 9:52 AM	

Event	Value	Format	Last Received
status	{"name":"Bin_3","lat":15.092677,"lon":79.188...	json	a few seconds ago
status	{"name":"Bin_3","lat":15.092677,"lon":79.188...	json	a few seconds ago
status	{"name":"Bin_3","lat":15.092677,"lon":79.188...	json	a few seconds ago
status	{"name":"Bin_3","lat":15.092677,"lon":79.188...	json	a few seconds ago

Node Red platform:

The screenshot shows the Node-RED dashboard. The top navigation bar includes 'Node-RED', 'Application Details - IBM Cloud', 'Node-RED : node-red-chbou', 'IBM Watson IoT Platform', and 'Node-RED Dashboard'. The main content area displays a flow diagram. The flow starts with an 'inject' node, followed by a 'debug' node. The data is then split into three parallel paths: one for 'Bin_Level' (level), one for 'Bin_Weight' (weight), and one for 'worldmap'. The 'level' and 'weight' paths are connected to 'level' and 'weight' nodes respectively. The 'worldmap' path is connected to a 'worldmap' node. The output of the flow is sent to a 'msg.payload' node. The right sidebar shows the 'debug' console with a list of messages, including JSON payloads containing location data (name, lat, lon) and status information.

```
graph LR
    inject[inject] --> debug[debug]
    debug --> Bin_Level[Bin_Level]
    debug --> Bin_Weight[Bin_Weight]
    debug --> worldmap[worldmap]
    Bin_Level --> level[level]
    Bin_Weight --> weight[weight]
    worldmap --> worldmap
    level --> msg_payload[msg.payload]
    weight --> msg_payload
    worldmap --> msg_payload
```

Output in Node Red:

