

Date	17 November 2022
Team ID	PNT2022TMID34892
Project Name	Smart waste management system for metropolitan cities
Story Points	15

Sprint 2

Develop the python code to find the GPS location using Latitude and Longitude (random values) and send it to Node red using IBM Watson platform and view location of bins on map

PYTHON CODE :

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

myConfig = {
    "identity": {
        "orgId": "fzv53v",
        "typeId": "Bin",
        "deviceId": "Bin_1"
    },
    "auth": {
        "token": "1234567890"
    }
}

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()

def pub (data):

    client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0,
onPublish=None)

    print ("Published data Successfully: %s", myData)

while True:

    myData={'name': 'Bin1', 'lat': 13.092677, 'lon': 80.188314}

    pub (myData)

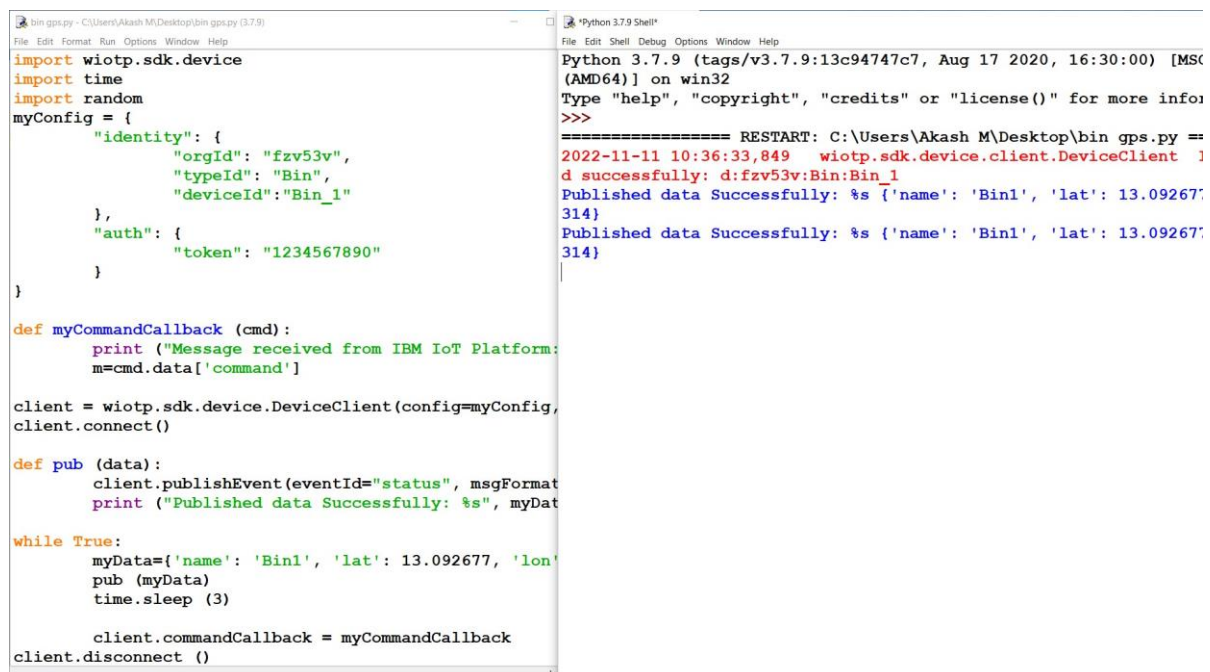
    time.sleep (3)

    client.commandCallback = myCommandCallback

client.disconnect ()

```

Output in python IDLE :



```

bin gps.py - C:\Users\Akash M\Desktop\bin gps.py (3.7.9)
File Edit Format Run Options Window Help
import wiotp.sdk.device
import time
import random
myConfig = {
    "identity": {
        "orgId": "fzv53v",
        "typeId": "Bin",
        "deviceId": "Bin_1"
    },
    "auth": {
        "token": "1234567890"
    }
}

def myCommandCallback (cmd):
    print ("Message received from IBM IoT Platform: ", cmd)
    m=cmd.data['command']

client = wiotp.sdk.device.DeviceClient(config=myConfig,
client.connect()

def pub (data):
    client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0, onPublish=None)
    print ("Published data Successfully: %s", myData)

while True:
    myData={'name': 'Bin1', 'lat': 13.092677, 'lon': 80.188314}
    pub (myData)
    time.sleep (3)

    client.commandCallback = myCommandCallback
client.disconnect ()

```

```

Python 3.7.9 (tags/v3.7.9:13c94747c7, Aug 17 2020, 16:30:00) [MSO
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more info
>>>
===== RESTART: C:\Users\Akash M\Desktop\bin gps.py ==
2022-11-11 10:36:33,849 wiotp.sdk.device.client.DeviceClient ]
d successfully: d:fzv53v:Bin:Bin_1
Published data Successfully: %s {'name': 'Bin1', 'lat': 13.092677, 'lon': 80.188314}
Published data Successfully: %s {'name': 'Bin1', 'lat': 13.092677, 'lon': 80.188314}

```

IBM Watson IOT platform :

The screenshot displays the IBM Watson IoT Platform dashboard. The top navigation bar includes tabs for 'Service Details - IBM Cloud', 'IBM Watson IoT Platform', 'Sprint_1.docx - Google Docs', 'Node-RED: node-red-usb', 'Node-RED Dashboard', and 'IBM'. The main content area shows a table of devices with columns: Device ID, Status, Device Type, Class ID, Date Added, and Descriptive Location. A device with ID 12345 is shown as 'Disconnected' with a 'Rasp' device type. Below the table, a detailed view for device 12345 is shown, including its identity, device information, recent events, state, and logs. The device information section lists: Device ID (12345), Device Type (Rasp), Date Added (Nov 12, 2022 10:39 AM), Added By (mithradhanalakshmi915@gmail.com), and Connection Status (Disconnected). The bottom status bar indicates '1 Simulation running'.

Device ID	Status	Device Type	Class ID	Date Added	Descriptive Location
12345	Disconnected	Rasp	Device	Nov 12, 2022 10:39 AM	

Device Information:

- Device ID: 12345
- Device Type: Rasp
- Date Added: Nov 12, 2022 10:39 AM
- Added By: mithradhanalakshmi915@gmail.com
- Connection Status: Disconnected

Node Red Platform :

The screenshot shows the Node-RED web interface. The top navigation bar includes tabs for 'Service Details - IBM Cloud', 'IBM Watson IoT Platform', 'Node-RED: node-red-usb', 'Signs with Smart Connectivity', and 'IBM'. The main content area displays a flow diagram for 'Flow 1'. The flow starts with an 'IBM IoT' node (connected) which branches into two parallel paths. The top path goes through a 'weight' function node to a 'Weight' output node. The bottom path goes through a 'location' function node to a 'Location' output node. Both output nodes are connected to a 'msg.payload' node. The right sidebar shows a 'dashboard' tab with 'Layout', 'Site', and 'Theme' options. Below this, there are 'Tabs & Links' for 'Smart Waste Management System' and 'Smart Garbage System'.

```
graph LR; IoT[IBM IoT] --> weight[weight]; IoT --> location[location]; weight --> Weight[Weight]; location --> Location[Location]; Weight --> payload[msg.payload]; Location --> payload;
```