

## Sprint 2

Date	5 November 2022
Team ID	PNT2022TMID20530
Project Name	Smart waste management system for metropolitan cities
Story Points	15

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

# watson device details

organization = "evd8ss"

devicType = "abcd"

deviceId = "1234"

authMethod= "token"

authToken= "12345678"

#generate random values for random variables (temperature&humidity)

def myCommandCallback(cmd):

    global a

    print("command recieved:%s"%cmd.data['command'])

    control=cmd.data['command']

    print(control)
```

try:

```
deviceOptions={"org": organization, "type": devicType,"id": deviceId,"auth-  
method":authMethod,"authtoken":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 "temp" with value integer value into the cloud as a type  
of event for every 10 seconds

```
deviceCli.connect()
```

while True:

```
distance= random.randint(10,70)
```

```
loadcell= random.randint(5,15)
```

```
data= {'dist':distance,'load':loadcell}
```

```
if loadcell < 13 and loadcell > 15:
```

```
    load = "90 %"
```

```
elif loadcell < 8 and loadcell > 12:
```

```
    load = "60 %"
```

```
elif loadcell < 4 and loadcell > 7:
```

```
    load = "40 %"
```

```
else: load = "0 %"
```

```
if distance < 15:
```

```
    dist = 'Risk warning:' 'Dumpster poundage getting high, Time to collect :) 90 %'
```

```
elif distance < 40 and distance >16:
```

```
    dist = 'Risk warning:' 'dumpster is above 60%'
```

```
elif distance < 60 and distance > 41:
```

```
    dist = 'Risk warning:' '40 %'
```

```
else:
```

```
    dist = 'Risk warning:' '17 %'
```

```
if load == "90 %" or distance == "90 %":
```

```
    warn = 'alert : ' ' Dumpster poundage getting high, Time to collect :)'
```

```
elif load == "60 %" or distance == "60 %":
```

```
    warn = 'alert : ' 'dumpster is above 60%'
```

```
else :
```

```
    warn = 'alert : ' 'No need to collect right now '
```

```
def myOnPublishCallback(lat=10.678991,long=78.177731):
```

[illegible]

## IBM Watson IoT platform :

The screenshot displays the IBM Watson IoT Platform dashboard. The main interface shows a list of devices with columns for Device ID, Status, Device Type, Class ID, and Date. A device with ID 1234 and type raspberrypi is selected. A modal window titled 'Simulations' is open, showing a list of simulated events. The events are generated by the device 'raspberrypi\_1' and consist of random number values.

Device ID	Status	Device Type	Class ID	Date
1234	Disconnected	abcd	Device	Nov 18, 2022 9:28 PM
1234	Disconnected	raspberrypi	Device	Nov 18, 2022 9:28 PM

Device Information for ID 1234:

- Device ID: 1234
- Device Type: raspberrypi
- Date Added: Nov 18, 2022 9:28 PM
- Added By: 1911110@nec.edu.in
- Connection Status: Disconnected

Simulation Events:

- event\_1: raspberrypi \* raspberrypi\_1 x 6
- ({"randomNumber": 4})
- ({"randomNumber": 73})
- ({"randomNumber": 52})
- ({"randomNumber": 87})
- ({"randomNumber": 26})
- ({"randomNumber": 6})

## Node Red Platform :

The screenshot shows the Node-RED web interface. The main workspace displays a flow diagram with the following components:

- Input:** A 'tcp request' node.
- Processing:** Two function nodes labeled 'weight' and 'location'.
- Output:** Two output nodes labeled 'Weight' and 'Location'.
- Message Payload:** A 'msg.payload' node.

The flow is configured to process incoming TCP requests and output weight and location data. The interface also includes a sidebar with various node categories (input, output, sequence) and a debug console on the right.