

**Sprint – 2**  
**Team ID: PNT2022TMID40473**

CODE:

```
#IBM Watson IOT Platform
#pip install wiotp-sdk
from geopy.geocoders import Nominatim
import wiotp.sdk.device
import time
import random
myConfig = {
    "identity": {
        "orgId": "4ipem5",
        "typeId": "PNT2022MID40473",
        "deviceId": "1234"
    },
    "auth": {
        "token": "8072679097"
    }
}

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)
    g1=random.randint(0,100)
    g2=random.randint(0,100)
    f="EMERGENCY"
    loc = Nominatim(user_agent="GetLoc")
    getLoc = loc.geocode("kanchipuram")
    getLoc1= loc.geocode("chennai")
    lat=getLoc.latitude
    log=getLoc.longitude
    lat1=getLoc1.latitude
    log1=getLoc1.longitude

    myData={'temperature':temp, 'humidity':hum,'gas_1':g1}
    if g1 > 40:
        myData={'alert_g_1':f,'latitude':lat,'longitude':log}
    else:
        myData={'temperature':temp, 'humidity':hum,'gas_1':g1}
    if g2 > 40:
```

```

        myData1={'alert_g_2':f,'latitude':lat1,'longitude':log1}
    else:
        myData1={'temperature':temp, 'humidity':hum,'gas_2':g2}

    client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0,
onPublish=None)
    client.publishEvent(eventId="status", msgFormat="json", data=myData1, qos=0,
onPublish=None)
    print("Published data Successfully: %s", myData)
    print("Published data successfully: %s", myData1)
    client.commandCallback = myCommandCallback
    time.sleep(3)
    client.disconnect()

```

The screenshot displays a Windows desktop environment. On the left, a window titled "pycode2.py - C:\python\Python37\pycode2.py (3.7.4)" shows a Python script. The script imports necessary modules, defines a configuration object for the IBM Watson IoT Platform, and includes a command callback function. It then creates a DeviceClient, connects to the platform, and enters a loop where it generates random data for temperature, humidity, and gas, and fetches location data from a geocoding service. On the right, a "Python 3.7.4 Shell" window shows the execution output. The output consists of multiple lines of text, including "Published data Successfully: %s" and "Published data successfully: %s", followed by JSON objects containing the generated data and location information. The Windows taskbar at the bottom shows the system clock as 15:02 on 15-11-2022.

The screenshot displays the IBM Watson IoT Platform interface. At the top, the navigation bar includes 'Browse', 'Action', 'Device Types', and 'Interfaces'. A user profile dropdown shows the email '513419106003@smartinternz.com' and ID '4ipem5'. An 'Add Device' button is visible in the top right.

The main content area shows a device card for ID '1234', which is 'Connected'. The device name is 'PNT2022MID40473'. Below the card, a table titled 'Recent Events' displays a stream of data. The table has four columns: 'Event', 'Value', 'Format', and 'Last Received'. The events are status alerts with a value containing latitude information, all in 'json' format, and received 'a few seconds ago'.

At the bottom of the dashboard, a status bar shows another device '12365' as 'Disconnected' and a notification that '1 Simulation running'.

Event	Value	Format	Last Received
status	{"alert_g_2":"EMERGENCY","latitude":13.083693...	json	a few seconds ago
status	{"alert_g_1":"EMERGENCY","latitude":12.964716...	json	a few seconds ago
status	{"alert_g_2":"EMERGENCY","latitude":13.083693...	json	a few seconds ago
status	{"alert_g_1":"EMERGENCY","latitude":12.964716...	json	a few seconds ago
status	{"alert_g_2":"EMERGENCY","latitude":13.083693...	json	a few seconds ago