

CODE :

Date	16 November 2022
Team ID	PNT2022TMID07803
Project Name	Project - Signs with smart connectivity for Better road safety

CODING & SOLUTIONING:

```
import wiotp.sdk.device
import time
import random
import ibmiotf.application
import ibmiotf.device
import requests, json
```

```
myConfig = {
#Configuration
"identity": {
"orgId": "3dpjnk",
"typeId": "Sign_Board",
"deviceId":"Board_1"},
#API Key
"auth": {
"token": "1234567890"
}
}
```

```
#Receiving callbacks from IBM IOT
```

```
platformdefmyCommandCallback(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()
```

#OpenWeatherMap Credentials

CITY = "Nagercoil"

URL = BASE_URL + "q=" + CITY + "&units=metric"+"&appid=" +
"01df65417ab3968e3fc2a38c4aee27bb"

while True:

response = requests.get(URL)

if response.status_code == 200:

data = response.json()

main = data['main']

temperature = main['temp']

humidity = main['humidity']

pressure = main['pressure']

report = data['visibility']

#messge part

msg=random.randint(0,5)

if msg==1:

message="SLOW DOWN, SCHOOL IS NEAR"

elif msg==2:

message="NEED HELP, POLICE STATION AHED"

elif msg==3:

message="EMERGENCY, HOSPITAL NEARBY"

elif msg==4:

message="DINE IN, RESTAURENT AVAILABLE"

else:

message=""

#Speed

#speedLimit part

speed=random.randint(0,150)

if speed>=100:

speedMsg=" Limit Exceeded"

elif speed>=60 and speed<100:

speedMsg="Moderate"

else:

speedMsg="Slow"

```
#Diversion part
sign=random.randint(0,5)
if sign==1:
    signMsg="Right Diversion"
elif sign==3:
    signMsg="Left Diversion"
elif sign==5:
    signmsg="U Turn"
else:
    signMsg=""
```

```
#Visibility
```

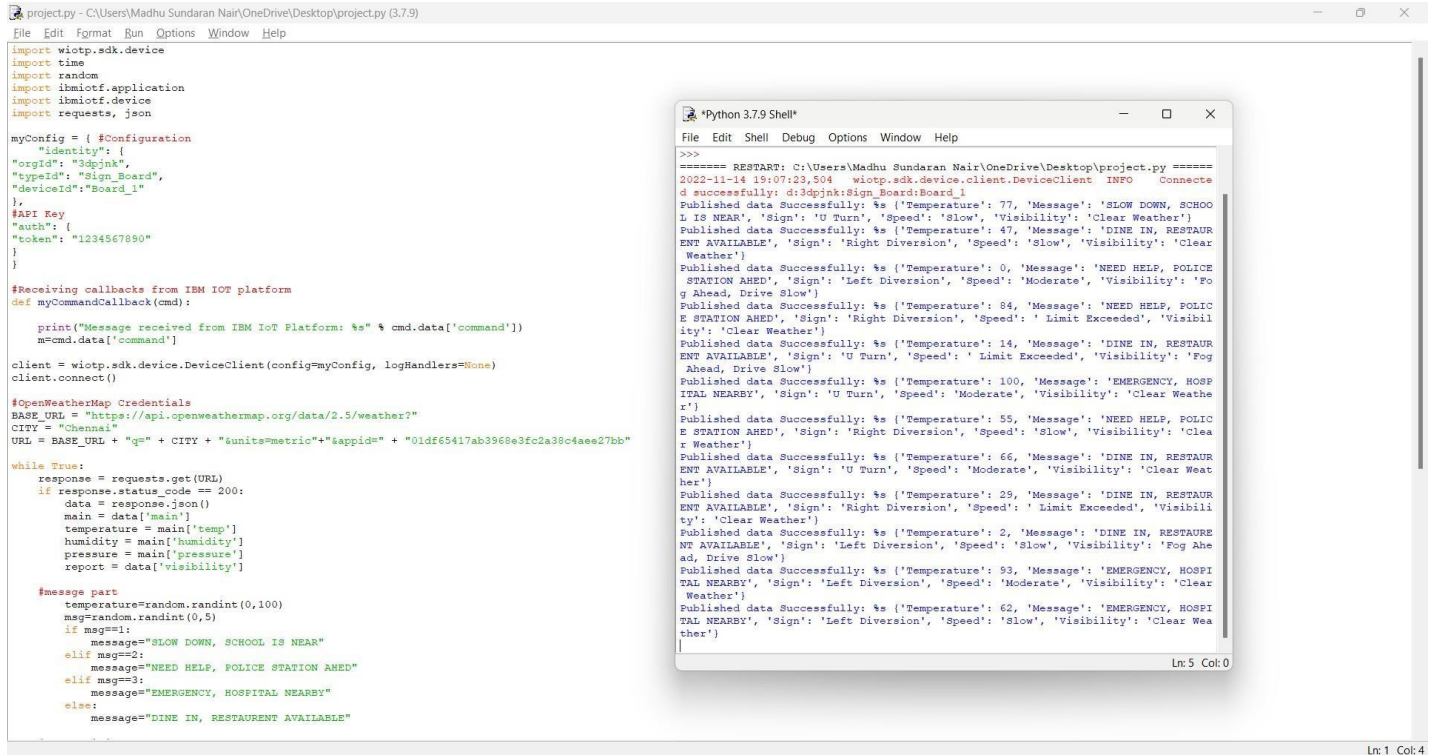
```
if temperature < 24:
    visibility="Fog Ahead, Drive Slow"
elif temperature < 20:
    visibility="Bad Weather"
elif temperature >24:
    visibility="Clear Weather"
else:
    print("Error in the HTTP request")
```

```
myData={'Temperature':temperature, 'Message':message, 'Sign':signMsg, 'Speed':speedMsg,
'Visibility':visibility}
client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0,
onPublish=None)
```

```
#PUBLISHING TO IOT WATSON
```

```
print("Published          data          Successfully:%s",myData)
client.commandCallback=myCommandCallbacktime.sleep(5)
client.disconnect()
```

Output:



The image shows a Python script in a text editor and its execution output in a terminal window. The script, named `project.py`, is located at `C:\Users\Madhu Sundaran Nair\OneDrive\Desktop\project.py` and is version 3.7.9. It imports `wiotp.sdk.device`, `time`, `random`, `ibmiotf.application`, `ibmiotf.device`, `requests`, and `json`. It defines a `myConfig` dictionary with fields for `identity` (including `orgId`, `typeId`, and `deviceId`), `API Key` (including `auth` and `token`), and a `myCommandCallback` function. The script then initializes a `DeviceClient` and connects to the IBM IoT platform. It also defines `OpenWeatherMap` credentials and a `while True` loop that fetches weather data from the OpenWeatherMap API and publishes it to the IBM IoT platform. The output in the terminal window shows the script's execution, including the restart command, the connection to the IBM IoT platform, and the successful publication of weather data to the `Sign_Board:Board_1` device. The output also shows the weather data being published, including temperature, humidity, pressure, and visibility, along with a message indicating the weather status (e.g., 'SLOW DOWN, SCHOOL IS NEAR', 'NEED HELP, POLICE STATION AHEAD', 'EMERGENCY, HOSPITAL NEARBY', 'DINE IN, RESTAURANT AVAILABLE').

```
project.py - C:\Users\Madhu Sundaran Nair\OneDrive\Desktop\project.py (3.7.9)
File Edit Format Run Options Window Help

import wiotp.sdk.device
import time
import random
import ibmiotf.application
import ibmiotf.device
import requests, json

myConfig = { #Configuration
    "identity": {
        "orgId": "3dpjnk",
        "typeId": "Sign_Board",
        "deviceId": "Board_1"
    },
    #API Key
    "auth": {
        "token": "1234567890"
    }
}

#Receiving callbacks from IBM IoT platform
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()

#OpenWeatherMap Credentials
BASE_URL = "https://api.openweathermap.org/data/2.5/weather?"
CITY = "Chennai"
URL = BASE_URL + "q=" + CITY + "&units=metric"&appid=" + "01df65417ab3968e3fc2a38c4aee27bb"

while True:
    response = requests.get(URL)
    if response.status_code == 200:
        data = response.json()
        main = data['main']
        temperature = main['temp']
        humidity = main['humidity']
        pressure = main['pressure']
        report = data['visibility']

        #message part
        temperature=random.randint(0,100)
        msg=random.randint(0,5)
        if msg==1:
            message="SLOW DOWN, SCHOOL IS NEAR"
        elif msg==2:
            message="NEED HELP, POLICE STATION AHEAD"
        elif msg==3:
            message="EMERGENCY, HOSPITAL NEARBY"
        else:
            message="DINE IN, RESTAURANT AVAILABLE"

Python 3.7.9 Shell
File Edit Shell Debug Options Window Help

>>>
===== RESTART: C:\Users\Madhu Sundaran Nair\OneDrive\Desktop\project.py =====
2022-11-14 19:07:23.504 wiotp.sdk.device.client.DeviceClient INFO Connecte
d successfully: d:3dpjnk:Sign_Board:Board_1
Published data Successfully: %s ('Temperature': 77, 'Message': 'SLOW DOWN, SCHOO
L IS NEAR', 'Sign': 'U Turn', 'Speed': 'Slow', 'Visibility': 'Clear Weather')
Published data Successfully: %s ('Temperature': 47, 'Message': 'DINE IN, RESTAUR
ENT AVAILABLE', 'Sign': 'Right Diversion', 'Speed': 'Slow', 'Visibility': 'Clear
Weather')
Published data Successfully: %s ('Temperature': 0, 'Message': 'NEED HELP, POLIC
E STATION AHEAD', 'Sign': 'Left Diversion', 'Speed': 'Moderate', 'Visibility': 'Fo
g Ahead, Drive Slow')
Published data Successfully: %s ('Temperature': 84, 'Message': 'NEED HELP, POLIC
E STATION AHEAD', 'Sign': 'Right Diversion', 'Speed': 'Limit Exceeded', 'Visibil
ity': 'Clear Weather')
Published data Successfully: %s ('Temperature': 14, 'Message': 'DINE IN, RESTAUR
ENT AVAILABLE', 'Sign': 'U Turn', 'Speed': 'Limit Exceeded', 'Visibility': 'Fog A
head, Drive Slow')
Published data Successfully: %s ('Temperature': 100, 'Message': 'EMERGENCY, HOSPI
TAL NEARBY', 'Sign': 'U Turn', 'Speed': 'Moderate', 'Visibility': 'Clear Weathe
r')
Published data Successfully: %s ('Temperature': 55, 'Message': 'NEED HELP, POLIC
E STATION AHEAD', 'Sign': 'Right Diversion', 'Speed': 'Slow', 'Visibility': 'Clea
r Weather')
Published data Successfully: %s ('Temperature': 66, 'Message': 'DINE IN, RESTAUR
ENT AVAILABLE', 'Sign': 'U Turn', 'Speed': 'Moderate', 'Visibility': 'Clear Weat
her')
Published data Successfully: %s ('Temperature': 25, 'Message': 'DINE IN, RESTAUR
ENT AVAILABLE', 'Sign': 'Right Diversion', 'Speed': 'Limit Exceeded', 'Visibil
ity': 'Clear Weather')
Published data Successfully: %s ('Temperature': 2, 'Message': 'DINE IN, RESTAUR
ENT AVAILABLE', 'Sign': 'Left Diversion', 'Speed': 'Slow', 'Visibility': 'Fog Ahe
ad, Drive Slow')
Published data Successfully: %s ('Temperature': 93, 'Message': 'EMERGENCY, HOSPI
TAL NEARBY', 'Sign': 'Left Diversion', 'Speed': 'Moderate', 'Visibility': 'Clear
Weather')
Published data Successfully: %s ('Temperature': 62, 'Message': 'EMERGENCY, HOSPI
TAL NEARBY', 'Sign': 'Left Diversion', 'Speed': 'Slow', 'Visibility': 'Clear Wea
ther')
Ln:5 Col:0
```

GitHub Link :

<https://github.com/IBM-EPBL/IBM-Project-33273-1660217657>

Project demo link:

<https://youtu.be/TkGzAzMyVwE>

