

Project Development Phase Sprint IV

Date	14 November 2022
Team ID	PNT2022TMID47661
Project Name	Project - Signs with smart connectivity for better road safety

Code for print the random temperature, Road signs, Speed limit, Message :

(RandomValues.py)

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

myConfig = {
    #Configuration
    "identity": {
        "orgId": "0nyujc",
        "typeId": "abcde",
        "deviceId": "12345"
    },
    #API Key
    "auth": {
        "token": "12345678"
    }
}

#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 = "Salem, IN"
URL = BASE_URL + "q=" + CITY + "&units=metric"+"&appid=" + "f58e4720c739a54c439aba9b05176839"

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="GO SLOW, SCHOOL ZONE AHEAD"
        elif msg==2:
            message="NEED HELP, POLICE STATION AHEAD"
        elif msg==3:
            message="EMERGENCY, HOSPITAL NEARBY"
        elif msg==4:
            message="DINE IN, RESTAURENT AVAILABLE"
        elif msg==5:
            message="PETROL BUNK NEARBY"
        else:
            message=""

        #Speed Limit 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==2:
            signMsg="Speed Breaker"
        elif sign==3:
            signMsg="Left Diversion"
        elif sign==4:
            signmsg="U Turn"
        else:
            signMsg=""

        #Visibility
        if temperature < 24:
            visibility="Fog Ahead, Drive Slow"
        elif temperature < 20:
            visibility="Bad Weather"
        else:
            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: ", myData)
    print("-----")
    print("-----")
    client.commandCallback = myCommandCallback
    time.sleep(5)
client.disconnect()

```

Python Simulation :

```

ibmiotpublishsubscribe.py - C:\Users\ELCOT\Downloads\ibmiotpublishsubscribe.py (3.7.0)
File Edit Format Run Options Window Help

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

#Provide your IBM Watson Device Credentials
organization = "Onyujc"
deviceType = "abcde"
deviceId = "12345"
authMethod = "use-token-auth"
authToken = "12345678"

# Initialize GPIO
def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    status=cmd.data['command']
    if status=="lighton":
        print ("led is on")
    elif status == "lightoff":
        print ("led is off")
    else :
        print ("please send proper command")

try:
    deviceOptions = {"org": organization, "type": deviceType, "id": deviceId, "auth-method": authMethod, "auth-token": 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 "hello" with value "world" into the cloud as an event of type "greeting" 10 times
deviceCli.connect()

while True:
    #Get Sensor Data from DHT11

    temp=random.randint(90,110)
    Humid=random.randint(60,100)

```

Ln: 1 Col: 0

Import wiotp-sdk & ibmiotf :

```
Command Prompt
Microsoft Windows [Version 10.0.17763.1577]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\ELCOT>pip install wiotp-sdk
WARNING: Retrying (Retry(total=4, connect=None, read=None, redirect=None, status=None)) after connection broken by 'NewConnectionError('<pip._vendor.urllib3.connection.
HTTPSConnection object at 0x000001E9EF8E0390>: Failed to establish a new connection: [Errno 11001] getaddrinfo failed')': /simple/wiotp-sdk/
WARNING: Retrying (Retry(total=3, connect=None, read=None, redirect=None, status=None)) after connection broken by 'NewConnectionError('<pip._vendor.urllib3.connection.
HTTPSConnection object at 0x000001E9EF954510>: Failed to establish a new connection: [Errno 11001] getaddrinfo failed')': /simple/wiotp-sdk/
WARNING: Retrying (Retry(total=2, connect=None, read=None, redirect=None, status=None)) after connection broken by 'NewConnectionError('<pip._vendor.urllib3.connection.
HTTPSConnection object at 0x000001E9EF954850>: Failed to establish a new connection: [Errno 11001] getaddrinfo failed')': /simple/wiotp-sdk/
WARNING: Retrying (Retry(total=1, connect=None, read=None, redirect=None, status=None)) after connection broken by 'NewConnectionError('<pip._vendor.urllib3.connection.
HTTPSConnection object at 0x000001E9EF95D310>: Failed to establish a new connection: [Errno 11001] getaddrinfo failed')': /simple/wiotp-sdk/
WARNING: Retrying (Retry(total=0, connect=None, read=None, redirect=None, status=None)) after connection broken by 'NewConnectionError('<pip._vendor.urllib3.connection.
HTTPSConnection object at 0x000001E9EF95D810>: Failed to establish a new connection: [Errno 11001] getaddrinfo failed')': /simple/wiotp-sdk/
ERROR: Could not find a version that satisfies the requirement wiotp-sdk (from versions: none)
ERROR: No matching distribution found for wiotp-sdk

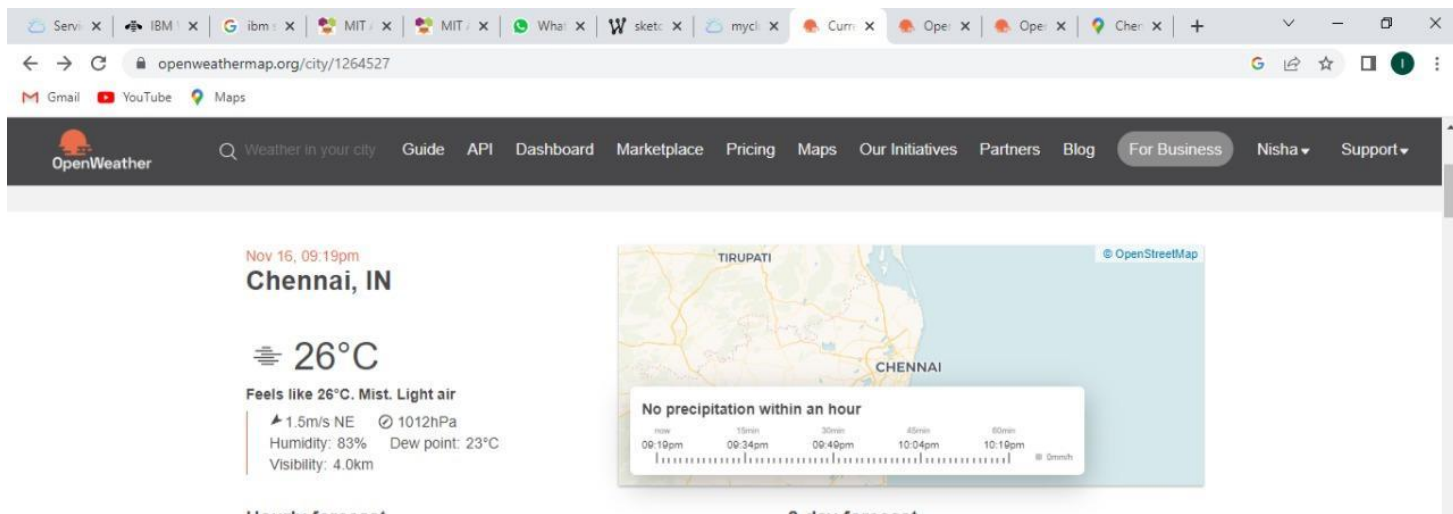
[notice] A new release of pip available: 22.3 -> 22.3.1
[notice] To update, run: python.exe -m pip install --upgrade pip

C:\Users\ELCOT>pip install ibmiotf
Requirement already satisfied: ibmiotf in c:\python311\lib\site-packages (0.4.0)
Requirement already satisfied: iso8601>=0.1.12 in c:\python311\lib\site-packages (from ibmiotf) (1.1.0)
Requirement already satisfied: pytz>=2017.3 in c:\python311\lib\site-packages (from ibmiotf) (2022.6)
Requirement already satisfied: paho-mqtt>=1.3.1 in c:\python311\lib\site-packages (from ibmiotf) (1.6.1)
Requirement already satisfied: requests>=2.18.4 in c:\python311\lib\site-packages (from ibmiotf) (2.28.1)
Requirement already satisfied: requests_toolbelt>=0.8.0 in c:\python311\lib\site-packages (from ibmiotf) (0.10.1)
Requirement already satisfied: charset-normalizer<3,>=2 in c:\python311\lib\site-packages (from requests>=2.18.4->ibmiotf) (2.1.1)
Requirement already satisfied: idna<4,>=2.5 in c:\python311\lib\site-packages (from requests>=2.18.4->ibmiotf) (3.4)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\python311\lib\site-packages (from requests>=2.18.4->ibmiotf) (1.26.12)
Requirement already satisfied: certifi>=2017.4.17 in c:\python311\lib\site-packages (from requests>=2.18.4->ibmiotf) (2022.9.24)

[notice] A new release of pip available: 22.3 -> 22.3.1
[notice] To update, run: python.exe -m pip install --upgrade pip

C:\Users\ELCOT>
```

OpenWeatherMap - (Ex.Chennai, IN) :



Python IDLE Output :

```
Python 3.7.0 Shell
File Edit Shell Debug Options Window Help
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:59:51) [MSC v.1914 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\ELCOT\Downloads\ibmiotpublishsubscribe.py =====
2022-11-07 20:01:24,074 ibmiotf.device.Client INFO Connected successfully: d:157uf3:abcd:7654321
Published Moisture = 90 deg C Temperature = 96 C Humidity = 76 % to IBM Watson
Published Moisture = 102 deg C Temperature = 110 C Humidity = 68 % to IBM Watson
Published Moisture = 45 deg C Temperature = 99 C Humidity = 100 % to IBM Watson
```