

Project development

Phase sprint-4

Team id	PNT2022TMID48884
Project name	Smart solution for railway

Code for (random temperature, road signs , speed limit , message):

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

myConfig = {
    #Configuration
    "identity": {
        "org Id": "prs76q",
        "type Id": "Ramya16",
        "device Id": "9629553721"
    },
    #API Key
    "auth": {
        "token": "Ramya06"
    }
}
```

#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 = "Dindigul, 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("")
    client.commandCallback = myCommandCallback
    time.sleep(5)
client.disconnect()
```

python simulation:

```
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": {
  "org Id": "prs76q",
  "type Id": "Ramyal6",
  "device Id": "9629553721"
}, #API Key
"auth": {
  "token": "Ramyag06"
}
}

#Receiving callbacks from IBM IOT platform def myCommandCallback(cmd):
print("Message received from IBM IoT Platform: %s" % cmd.data['command']) m=cmd.
client = wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None) client

#OpenWeatherMap Credentials
BASE_URL = "https://api.openweathermap.org/data/2.5/weather?" CITY = "Dindigul",
URL = BASE_URL + "q=" + CITY + "&units=metric"&"&appid=" + "f58e4720c739a54c439a

while True:
response = requests.get(URL)
if response.status_code == 200: data = response.json()
main = data['main'] temperature = main['temp'] humidity = main['humidity'] press
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':sp
client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0, onPu
print("Published data Successfully: ", myData) print("")
client.commandCallback = myCommandCallback time.sleep(5)
client.disconnect()

```

Import wiotp-sdk and ibm iot :

```

Microsoft Windows [Version 10.0.19044.2251]
(c) Microsoft Corporation. All rights reserved.

C:\Users\HP\AppData\Local\Programs\Python\Python37\Scripts>pip install wiotp-sdk
Collecting wiotp-sdk
  Downloading wiotp-sdk-0.11.0.tar.gz (96 kB)
    ----- 96.2/96.2 kB 239.4 kB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
Requirement already satisfied: iso8601>=0.1.12 in c:\users\hp\appdata\local\programs\python\python37\lib\site-packages (from wiotp-sdk) (1.1.0)
Requirement already satisfied: pytz>=2018.9 in c:\users\hp\appdata\local\programs\python\python37\lib\site-packages (from wiotp-sdk) (2022.6)
Collecting pyyaml>=3.13
  Downloading PyYAML-6.0-cp37-cp37m-win_amd64.whl (153 kB)
    ----- 153.2/153.2 kB 916.1 kB/s eta 0:00:00
Requirement already satisfied: paho-mqtt>=1.5.0 in c:\users\hp\appdata\local\programs\python\python37\lib\site-packages (from wiotp-sdk) (1.6.1)
Requirement already satisfied: requests>=2.21.0 in c:\users\hp\appdata\local\programs\python\python37\lib\site-packages (from wiotp-sdk) (2.28.1)
Requirement already satisfied: requests-toolbelt>=0.8.0 in c:\users\hp\appdata\local\programs\python\python37\lib\site-packages (from wiotp-sdk) (0.10.1)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\users\hp\appdata\local\programs\python\python37\lib\site-packages (from requests>=2.21.0->wiotp-sdk) (1.26.12)
Requirement already satisfied: idna<4,>=2.5 in c:\users\hp\appdata\local\programs\python\python37\lib\site-packages (from requests>=2.21.0->wiotp-sdk) (3.4)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\hp\appdata\local\programs\python\python37\lib\site-packages (from requests>=2.21.0->wiotp-sdk) (2022.9.24)
Requirement already satisfied: charset-normalizer<3,>=2 in c:\users\hp\appdata\local\programs\python\python37\lib\site-packages (from requests>=2.21.0->wiotp-sdk) (2.1.1)
Installing collected packages: pyyaml, wiotp-sdk
  DEPRECATION: wiotp-sdk is being installed using the legacy 'setup.py install' method, because it does not have a 'pyproject.toml' and the 'wheel' package is not installed. pip 23.1 will enforce this behaviour change. A possible replacement is to enable the '--use-pep517' option. Discussion can be found at https://github.com/pypa/pip/issues/8559
  Running setup.py install for wiotp-sdk ... done
Successfully installed pyyaml-6.0 wiotp-sdk-0.11.0

C:\Users\HP\AppData\Local\Programs\Python\Python37\Scripts>

```

```

Microsoft Windows [Version 10.0.19044.2251]
(c) Microsoft Corporation. All rights reserved.

.:Users\HP\AppData\Local\Programs\Python\Python37\Scripts>pip install ibmiotf
Requirement already satisfied: ibmiotf in c:\users\hp\appdata\local\programs\python\python37\lib\site-packages (0.4.0)
Requirement already satisfied: iso8601>=0.1.12 in c:\users\hp\appdata\local\programs\python\python37\lib\site-packages (from ibmiotf) (1.1.0)
Requirement already satisfied: pytz>=2017.3 in c:\users\hp\appdata\local\programs\python\python37\lib\site-packages (from ibmiotf) (2022.6)
Requirement already satisfied: paho-mqtt>=1.3.1 in c:\users\hp\appdata\local\programs\python\python37\lib\site-packages (from ibmiotf) (1.6.1)
Requirement already satisfied: requests>=2.18.4 in c:\users\hp\appdata\local\programs\python\python37\lib\site-packages (from ibmiotf) (2.28.1)
Requirement already satisfied: requests-toolbelt>=0.8.0 in c:\users\hp\appdata\local\programs\python\python37\lib\site-packages (from ibmiotf) (0.10.1)
Requirement already satisfied: idna<4,>=2.5 in c:\users\hp\appdata\local\programs\python\python37\lib\site-packages (from requests>=2.18.4->ibmiotf) (3.4)
Requirement already satisfied: charset-normalizer<3,>=2 in c:\users\hp\appdata\local\programs\python\python37\lib\site-packages (from requests>=2.18.4->ibmiotf) (2.1.1)

Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\users\hp\appdata\local\programs\python\python37\lib\site-packages (from requests>=2.18.4->ibmiotf) (1.26.12)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\hp\appdata\local\programs\python\python37\lib\site-packages (from requests>=2.18.4->ibmiotf) (2022.9.24)

.:Users\HP\AppData\Local\Programs\Python\Python37\Scripts>

```

OpenWeatherMap – (Ex : Dindigul,IN):

The screenshot shows the OpenWeatherMap website interface. The browser address bar displays the URL `openweathermap.org/find?q=dindigul`. The website header includes the OpenWeather logo, a search bar with the text "Weather in your city", and navigation links for Guide, API, Dashboard, Marketplace, Pricing, Maps, Our Initiatives, Partners, Blog, For Business, and Support. The main content area features a large orange search bar with the text "dindigul" and a "Search" button. Below the search bar, the weather information for Dindigul, IN is displayed, including a weather icon (broken clouds), the temperature (25.8°C), and a description (broken clouds). The temperature is shown in a large, bold font. Below the temperature, the text "Dindigul, IN broken clouds" is displayed. The temperature is also shown in a smaller font as "25.8°C". Below the temperature, the text "temperature from 25.5 to 25.5 °C, wind 2.29 m/s, clouds 77 %, 1015 hpa" is displayed. Below the temperature, the text "Geo coords [10.35, 77.95]" is displayed. Below the weather information, the text "Search engine is very flexible. How it works:" is displayed. Below the text, a list of instructions is provided: "To make it more precise put the city's name, comma, 2-letter country code (ISO3166). You will get all proper cities in chosen country. The order is important - the first is city name then comma then country. Example - London, GB or New York, US."

Python IDLE Output:



```
Python 3.8.5 Shell
File Edit Shell Debug Options Window Help

Published data Successfully: {'Temperature': 26.03, 'Message': 'EMERGENCY, HOSPITAL NEARBY', 'Sign': 'Speed Breaker', 'Speed': 'Limit Exceeded', 'Visibility': 'Clear Weather'}

Published data Successfully: {'Temperature': 26.03, 'Message': 'GO SLOW, SCHOOL / COLLEGE ZONE AHEAD', 'Sign': 'Right Diversion', 'Speed': 'Moderate', 'Visibility': 'Clear Weather'}

Published data Successfully: {'Temperature': 26.03, 'Message': 'PETROL BUNK NEARBY', 'Sign': 'Speed Breaker', 'Speed': 'Limit Exceeded', 'Visibility': 'Clear Weather'}

Published data Successfully: {'Temperature': 26.03, 'Message': 'EMERGENCY, HOSPITAL NEARBY', 'Sign': 'Speed Breaker', 'Speed': 'Slow', 'Visibility': 'Clear Weather'}

Published data Successfully: {'Temperature': 26.03, 'Message': '', 'Sign': '', 'Speed': 'Limit Exceeded', 'Visibility': 'Clear Weather'}

Published data Successfully: {'Temperature': 26.03, 'Message': 'EMERGENCY, HOSPITAL NEARBY', 'Sign': '', 'Speed': 'Moderate', 'Visibility': 'Clear Weather'}

Published data Successfully: {'Temperature': 26.03, 'Message': 'EMERGENCY, HOSPITAL NEARBY', 'Sign': '', 'Speed': 'Slow', 'Visibility': 'Clear Weather'}

Published data Successfully: {'Temperature': 26.03, 'Message': 'NEED HELP, POLICE STATION AHEAD', 'Sign': 'Left Diversion', 'Speed': 'Moderate', 'Visibility': 'Clear Weather'}
```

Line 24 Col 0