

PROJECT DEVELOPMENT – DELIVERY OF SPRINT – 4

Date	14 November 2022
Team ID	PNT2022TMID46406
Project Title	Signs With Smart Connectivity for Better RoadSafety

SPRINT-4 (USN - 6)

- **Information about Traffic around the area is gathered as a data. And the data is further encoded.**
- **Data collected from sprint 2 & sprint 3 is deployed in NodeRed service to link API.**

STEP 1: Developing a python script from Open Weather API.

```
spr4.py - C:\Users\ADMIN\Desktop\spr4.py (3.7.0)
File Edit Format Run Options Window Help
import wiotp.sdk.device #importing library files for connecting with CLOUD,sdk=softwa
import requests #for API request
import json #converting it to json(key:values)
import sys
myConfig = {
    "identity": {
        "orgId": "uaortj",
        "typeId": "Monitor_devicetype", #configuration wit CLOUD,finding identity
        "deviceId": "Monitor_deviceid"
    },
    "auth": {
        "token": "sngsl23monitor" #authenticating with cloud device
    }
}
#TRAFFIC AND FATAL SITUATION ALERT MESSAGE DISPLAYING IN WEB UI WHNN THE
client = wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None) #initial
client.connect()
ALERT=""
NOTIFY=""
def myCommandCallback(cmd):
    print("Message received from IBM IoT Platform: %s" % cmd.data['command'])
    m=cmd.data['command']
    #THIS IF COMDITION BLOCK IS FOR TRAFFIC AND FATAL SITUATION ALERT MESSAGE DISP
    ALERT=""
    NOTIFY=""
    if(m=="TRAFFIC"):
        ALERT="TRAFFIC - PLEASE WAIT OR PREFER ANOTHER ROUTE"
        print("*****//PLEASE WAIT OR PREFER ANOTHER ROUTE//*****")
    elif(m=="ACCIDENT"):
        ALERT="ACCIDENT - TAKE DIVERSION"
        print("*****//TAKE DIVERSION//*****")
    elif(m=="MESSAGE"):
        ALERT="HAVE A NICE DAY!"
        print("HAVE A NICE DAY!")
        #THE BELOW CONDITION BLOCK IS TO DISPLAY HOSPITAL ,SCHOOL, AND RESTAURANT REGI
    if(m=="SCHOOL"):
        NOTIFY="SCHOOL REGION MAINTAIN SPEED LIMIT BELOW 40KM/HR"
        print("SCHOOL REGION MAINTAIN SPEED LIMIT BELOW 40KM/HR")
    elif(m=="HOSPITAL"):
        NOTIFY="HOSPITAL REGION DONT USE HORN"
        print("HOSPITAL REGION DONT USE HORN")

Python 3.7.0 Shell
File Edit Shell Debug Options Window Help
n': '', 'DescriptionOfWeather': 'haze', 'visibility': 5000, 'RecommendationForV
isibility': 'visibility range is ideal for vehciles', 'WindSpeed': 6.17, 'LOCAT
ION': 'Chennai,%20IN')
=====
{'temperature': 303.14, 'TemperatureRecommendation': 'Temperature is higher tha
n ideal value', 'humidity': 62, 'WeatherCondition': 'Haze', 'SpeedRecommendatio
n': '', 'DescriptionOfWeather': 'haze', 'visibility': 5000, 'RecommendationForV
isibility': 'visibility range is ideal for vehciles', 'WindSpeed': 6.17, 'LOCAT
ION': 'Chennai,%20IN')
=====
{'temperature': 303.14, 'TemperatureRecommendation': 'Temperature is higher tha
n ideal value', 'humidity': 62, 'WeatherCondition': 'Haze', 'SpeedRecommendatio
n': '', 'DescriptionOfWeather': 'haze', 'visibility': 5000, 'RecommendationForV
isibility': 'visibility range is ideal for vehciles', 'WindSpeed': 6.17, 'LOCAT
ION': 'Chennai,%20IN')
=====
{'temperature': 303.14, 'TemperatureRecommendation': 'Temperature is higher tha
n ideal value', 'humidity': 62, 'WeatherCondition': 'Haze', 'SpeedRecommendatio
n': '', 'DescriptionOfWeather': 'haze', 'visibility': 5000, 'RecommendationForV
isibility': 'visibility range is ideal for vehciles', 'WindSpeed': 6.17, 'LOCAT
ION': 'Chennai,%20IN')
=====
{'temperature': 303.14, 'TemperatureRecommendation': 'Temperature is higher tha
n ideal value', 'humidity': 62, 'WeatherCondition': 'Haze', 'SpeedRecommendatio
n': '', 'DescriptionOfWeather': 'haze', 'visibility': 5000, 'RecommendationForV
isibility': 'visibility range is ideal for vehciles', 'WindSpeed': 6.17, 'LOCAT
ION': 'Chennai,%20IN')
=====
{'temperature': 303.14, 'TemperatureRecommendation': 'Temperature is higher tha
n ideal value', 'humidity': 62, 'WeatherCondition': 'Haze', 'SpeedRecommendatio
n': '', 'DescriptionOfWeather': 'haze', 'visibility': 5000, 'RecommendationForV
isibility': 'visibility range is ideal for vehciles', 'WindSpeed': 6.17, 'LOCAT
ION': 'Chennai,%20IN')
=====
{'temperature': 303.14, 'TemperatureRecommendation': 'Temperature is higher tha
n ideal value', 'humidity': 62, 'WeatherCondition': 'Haze', 'SpeedRecommendatio
n': '', 'DescriptionOfWeather': 'haze', 'visibility': 5000, 'RecommendationForV
isibility': 'visibility range is ideal for vehciles', 'WindSpeed': 6.17, 'LOCAT
ION': 'Chennai,%20IN')
=====
Ln: 1 Col: 1
Ln: 135 Col: 95
Type here to search 28°C 12:12 19-11-2022
```

STEP 2: By running the above Python Script, we can see the conditions of the current location using Open Weather API and IBM Cloud.

The screenshot displays the IBM Watson IoT Platform dashboard. The top navigation bar includes 'Browse', 'Action', 'Device Types', and 'Interfaces'. A sidebar on the left contains various icons for navigation. The main content area shows a device named 'RoadSafety_deviceid' with a status of 'Connected'. Below the device name, there are tabs for 'Identity', 'Device Information', 'Recent Events', 'State', and 'Logs'. The 'Recent Events' tab is active, showing a table of events. The table has columns for 'Event', 'Value', 'Format', and 'Last Received'. The events listed are all from the device '123456' and represent temperature recommendations in JSON format, received a few seconds ago.

Event	Value	Format	Last Received
123456	{"temperature":296.14,"TemperatureRecommen...	json	a few seconds ago
123456	{"temperature":296.14,"TemperatureRecommen...	json	a few seconds ago
123456	{"temperature":296.14,"TemperatureRecommen...	json	a few seconds ago
123456	{"temperature":296.14,"TemperatureRecommen...	json	a few seconds ago
123456	{"temperature":296.14,"TemperatureRecommen...	json	a few seconds ago

STEP 3: Create a Node – RED flow using the Node – RED Flow Editor.

The screenshot displays the Node-RED web interface in a browser. The browser's address bar shows the URL: `node-red-ncvoj-2022-11-14.eu-gb.mybluemix.net/red/#flow/2f5ce064b3a86f59`. The interface includes a top navigation bar with tabs for 'Node-RED', 'IBM Watson IoT Platform', 'Node-RED Dashboard', 'IBM-Project-1265-16583821', and 'New Tab'. Below this, a tabbed interface shows 'Flow 1' through 'Flow 5', with 'Flow 4' currently selected.

The main workspace of Flow 4 contains a complex flow diagram. On the left, there are several input nodes: 'School zone', 'Hospital zone', 'Restaurant', 'Traffic', 'Accident', and 'Info'. These are connected to two function nodes labeled 'Notify' and 'Alert'. Both function nodes are connected to two 'IBM IoT' nodes, which are marked as 'connected'. These IoT nodes then connect to a central hub that branches out to multiple output nodes: 'msg payload', 'Temperature', 'Humidity', 'Visibility', 'WindSpeed', 'Temperature abc', 'Speed abc', 'Visibility abc', 'Status abc', and 'Description abc'. At the bottom of the flow, there is a sequence of nodes: 'timestamp', 'abc', 'text input', and 'audio out'.

On the right side of the interface, a 'debug' console is open, showing a log of messages. The messages are JSON objects containing sensor data and recommendations. For example, one message is:

```
{ temperature: 296.14, TemperatureRecommendation: "Temperature is higher than ide...", humidity: 88, WeatherCondition: "Mist", SpeedRecommendation: "30KM/HR and switch on the head..." }
```

. The console also shows the timestamp '11/19/2022, 7:37:37 AM' and the node ID 'node: 906ecfe12afdc02a'.

The bottom of the image shows a Windows taskbar with a search bar, several application icons, and a system tray displaying the date '19-11-2022' and time '07:44'.

STEP 4: We can see the road safety instructions displayed in the Node – RED URL and it can be linked with the digital board

