

SPRINT 2

TOPIC : SIGNS WITH SMART CONNECTIVITY FOR BETTER ROAD SAFETY

TEAM ID: PNT2022 TMID06138

DATE : 5 NOV 2022

SOFTWARE USED:

- Python
- IBM Watson IOT.

US 1: Developing a python script to communicate between open-weather API and IOT platform.

PYTHON CODE:

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

myConfig = {
    "identity": {
        "orgId": "e4jrbo",
        "typeId": "SignsWithSmartConnectivity",
        "deviceId": "12345"
    },
```

```
"auth": {  
    "token": "1234567890"  
}  
}
```

```
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
```

```
URL =  
"http://api.openweathermap.org/data/2.5/weather?q=Kāraikkudi,IN&units=metric&appid=76e08ef85f6173baed5302d8d21a6d24"
```

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

```
humidity = main['humidity']
vis = (data['visibility']/1000)
place=data['name']
wea= data['weather'][0]['main']

vis_ms=""
if vis>=10:
    vis_ms+="Road is visible"
else:
    vis_ms+="Visiblity is Low, Drive safely"

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=random.randint(0,150)
if speed>=100:
```

```
    speedMsg="Speed Limit Exceeded"
elif speed>=60 and speed<100:
    speedMsg="Moderate Speed"
else:
    speedMsg="Slow and steady"

if temperature < 24:
    visibility="cold weather, Drive Slow"
elif temperature < 20:
    visibility="Bad Weather, Be Careful"
else:
    visibility="Clear Weather, Safe Journey"
```

```
sign=random.randint(0,6)
if sign==1:
    signMsg="Right Diversion"
elif sign==2:
    signMsg="Speed Breaker"
elif sign==3:
    signMsg="Left Diversion"
elif sign==4:
    signmsg="U Turn"
elif sign==5:
    signMsg="Under Repair"
else:
```

```
signMsg="""
```

```
myData={'Temperature':temperature, 'Visibility':vis, 'temp-msg':visibility,  
'Sign_msg':signMsg, 'Vis_msg':vis_ms, 'LM_msg':message,  
'Speed_msg':speedMsg, 'Humidity':humidity, 'Place':place, 'Weather':wea}
```

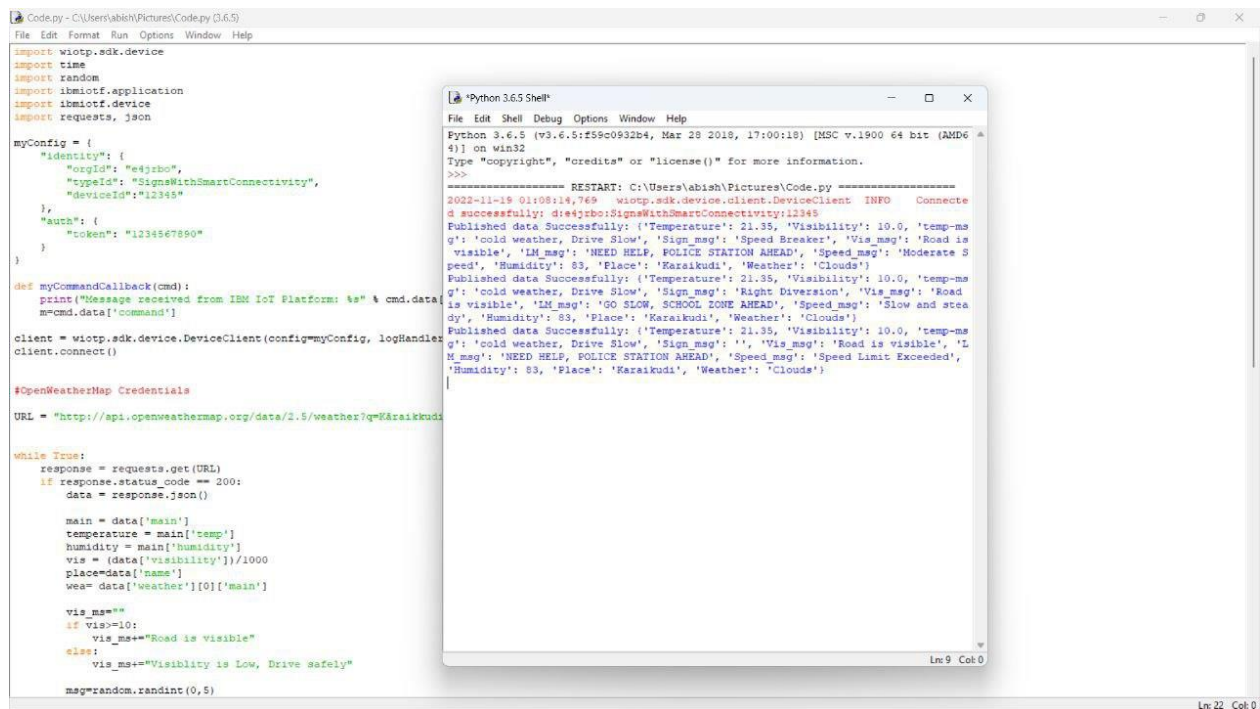
```
client.publishEvent(eventId="status", msgFormat="json", data=myData,  
qos=0, onPublish=None)
```

```
print("Published data Successfully:", myData)
```

```
client.commandCallback = myCommandCallback
```

```
time.sleep(2)
```

```
client.disconnect()
```

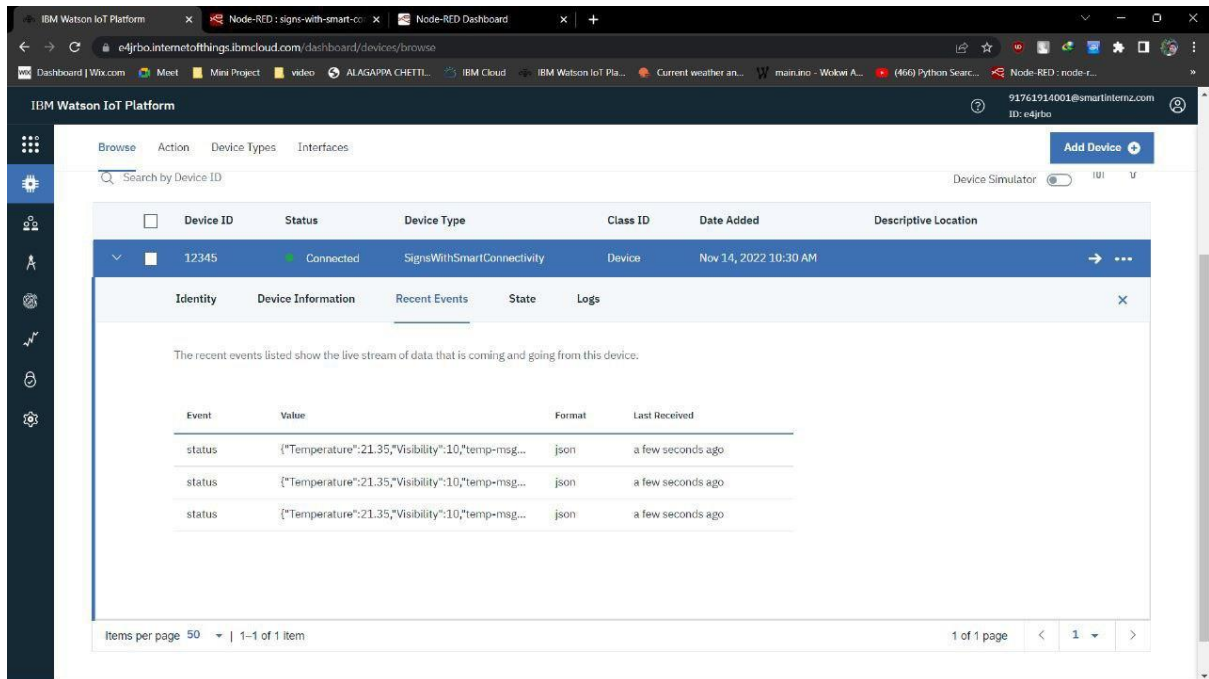


The screenshot shows a Python script in a Code Editor window titled 'Code.py - C:\Users\abish\Picture\Code.py (3.6.5)'. The script imports modules like `wiotp.sdk.device`, `time`, `random`, `ibmiotf.application`, `ibmiotf.device`, `requests`, and `json`. It defines a `myConfig` dictionary with identity and authentication details. A `myCommandCallback` function is defined to print received commands. The script then connects to a Wiotp device client and publishes data. It also fetches weather data from OpenWeatherMap and updates the `myData` dictionary. The script is running in a loop.

The output is shown in a 'Python 3.6.5 Shell' window. It displays the restart command, the connection status, and the published data. The data is a JSON object containing temperature, visibility, sign message, visibility message, LM message, speed message, humidity, place, and weather. The weather data is fetched from OpenWeatherMap and includes main, temperature, humidity, visibility, place, and weather details.

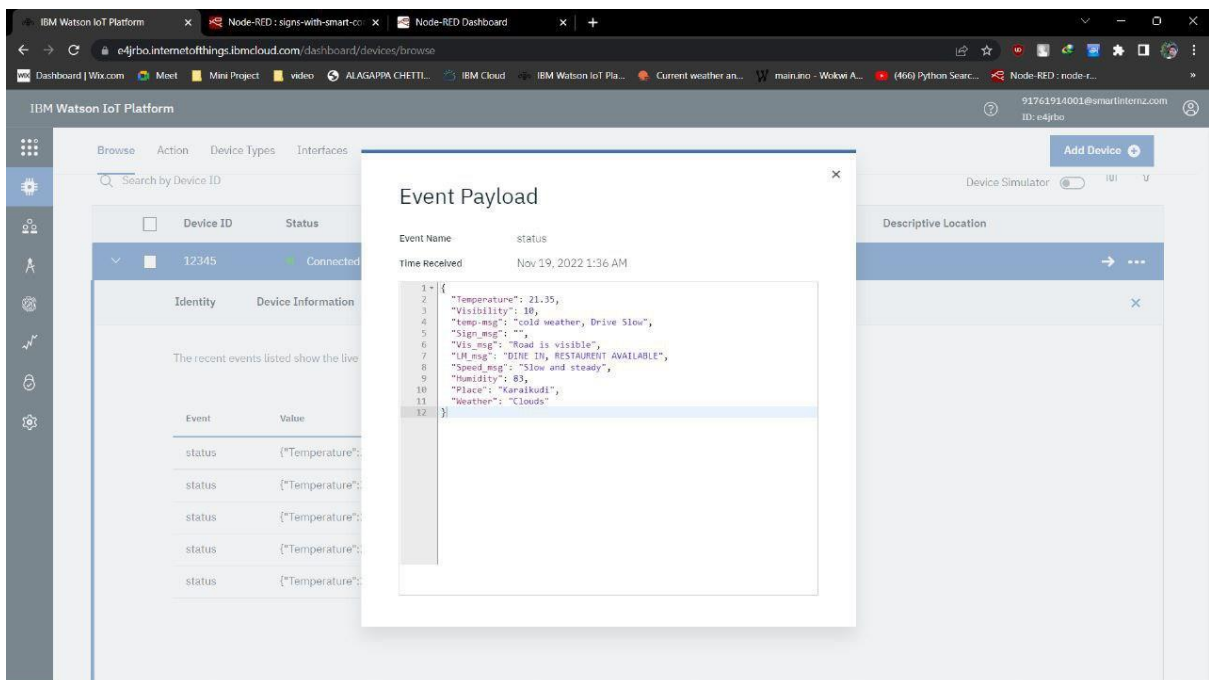
```
Python 3.6.5 Shell
Python 3.6.5 [v3.6.5:258c0932b4, Mar 28 2018, 17:00:18] [MSC v.1900 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
----- RESTART: C:\Users\abish\Picture\Code.py -----
2022-11-19 01:08:14.769 wiotp.sdk.device.client.DeviceClient INFO Connecte
d successfully: d1e4j3bo:SignsWithSmartConnectivity:12345
Published data Successfully: {'Temperature': 21.35, 'Visibility': 10.0, 'temp-ms
g': 'cold weather, Drive Slow', 'Sign_msg': 'Speed Breaker', 'Vis_msg': 'Road is
visible', 'LM_msg': 'NEED HELP, POLICE STATION AHEAD', 'Speed_msg': 'Moderate S
peed', 'Humidity': 83, 'Place': 'Karaikudi', 'Weather': 'Clouds'}
Published data Successfully: {'Temperature': 21.35, 'Visibility': 10.0, 'temp-ms
g': 'cold weather, Drive Slow', 'Sign_msg': 'Right Diversion', 'Vis_msg': 'Road
is visible', 'LM_msg': 'GO SLOW, SCHOOL ZONE AHEAD', 'Speed_msg': 'Slow and stea
dy', 'Humidity': 83, 'Place': 'Karaikudi', 'Weather': 'Clouds'}
Published data Successfully: {'Temperature': 21.35, 'Visibility': 10.0, 'temp-ms
g': 'cold weather, Drive Slow', 'Sign_msg': '', 'Vis_msg': 'Road is visible', 'L
M_msg': 'NEED HELP, POLICE STATION AHEAD', 'Speed_msg': 'Speed Limit Exceeded',
'Humidity': 83, 'Place': 'Karaikudi', 'Weather': 'Clouds'}
```

US 2: Getting data to IOT platform



The screenshot shows the IBM Watson IoT Platform dashboard. The top navigation bar includes 'Browse', 'Action', 'Device Types', and 'Interfaces'. A search bar is present with the text 'Search by Device ID'. The main content area displays a table of devices. The first device, ID 12345, is 'Connected' and has a 'SignsWithSmartConnectivity' device type. Below the device list, a modal window titled 'Recent Events' is open, showing a table of events. The events table has columns for 'Event', 'Value', 'Format', and 'Last Received'. The events listed are 'status' with a JSON payload, received 'a few seconds ago'.

Event	Value	Format	Last Received
status	{\"Temperature\":21.35,\"Visibility\":10,\"temp-msg...}	json	a few seconds ago
status	{\"Temperature\":21.35,\"Visibility\":10,\"temp-msg...}	json	a few seconds ago
status	{\"Temperature\":21.35,\"Visibility\":10,\"temp-msg...}	json	a few seconds ago



The screenshot shows the IBM Watson IoT Platform dashboard with an 'Event Payload' modal window open. The modal displays the event name 'status' and the time received 'Nov 19, 2022 1:36 AM'. The event payload is a JSON object with the following structure:

```
1 {
2   \"Temperature\": 21.35,
3   \"Visibility\": 10,
4   \"temp-msg\": \"cold weather, Drive Slow\",
5   \"Sign-msg\": \"\",
6   \"Vis-msg\": \"Road is visible\",
7   \"IH-msg\": \"DINE IN, RESTAURENT AVAILABLE\",
8   \"Speed-msg\": \"Slow and steady\",
9   \"Humidity\": 83,
10  \"Place\": \"Karaikudi\",
11  \"Weather\": \"Clouds\"
12 }
```