

ASSIGNMENT 3:

Develop a code to upload the water tank level and light intensity values to the IBM IoT platform and visualize them in the web application.

CODE:

```
waterlevel.py - C:\Users\SUBHRAJIT HOOM\Downloads\waterlevel.py (3.9.6)
File Edit Format Run Options Window Help
import wiotp.sdk.device
import time
import random
myConfig = {
    "identity": {
        "orgId": "3zj2w8",
        "typeId": "vitdevice",
        "deviceId": "8335"
    },
    "auth": {
        "token": "8335920874"
    }
}

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()

while True:
    level=random.randint(0,1000)
    light=random.randint(0,100)
    myData={'Waterlevel':level, 'Lightintensity':light}
    client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0,
    print("Published data Successfully: %s", myData)
    client.commandCallback = myCommandCallback
    time.sleep(5)
client.disconnect()
```

NODE-RED OUTPUT:

The Node-RED interface displays a flow named 'Flow 1' with the following components:

- IBM IoT Node:** A blue node labeled 'connected' that receives data from the IBM IoT platform.
- Function Nodes:** Two orange nodes labeled 'waterlevel' and 'lightintensity' that process the incoming data.
- Output Nodes:** Two green nodes labeled 'Waterlevel' and 'Lightintensity' that output the processed data.
- msg payload Node:** A green node that outputs the raw message payload.

The debug console on the right shows the following logs:

```
7/19/2021, 10:57:30 AM node:1a4b377d-46a759
ok:2btypevitdeviceid8335devstatusmqttjson:
msg payload: Object
* { Waterlevel: 877, Lightintensity: 90 }
877
7/19/2021, 10:57:30 AM node:1a4b377d-46a759
ok:2btypevitdeviceid8335devstatusmqttjson:
msg payload: number
877
7/19/2021, 10:57:30 AM node:1a4b377d-46a759
ok:2btypevitdeviceid8335devstatusmqttjson:
msg payload: number
90
7/19/2021, 10:57:35 AM node:1a4b377d-46a759
ok:2btypevitdeviceid8335devstatusmqttjson:
msg payload: Object
* { Waterlevel: 716, Lightintensity: 74 }
716
7/19/2021, 10:57:35 AM node:1a4b377d-46a759
ok:2btypevitdeviceid8335devstatusmqttjson:
msg payload: number
716
```

IDLE Shell 3.9.6

File Edit Shell Debug Options Window Help

Python 3.9.6 (tags/v3.9.6:db3ff76, Jun 28 2021, 15:26:21) [MSC v.1929 64 bit D64] on win32

Type "help", "copyright", "credits" or "license()" for more information.

>>>

===== RESTART: C:\Users\SUBHRAJIT HOOM\Downloads\waterlevel.py =====

2021-07-19 10:57:25,349 wiotp.sdk.device.client.DeviceClient INFO Connected successfully: d:3zj2w8:vitdevice:8335

Published data Successfully: %s {'Waterlevel': 36, 'Lightintensity': 54}

Published data Successfully: %s {'Waterlevel': 877, 'Lightintensity': 90}

Published data Successfully: %s {'Waterlevel': 716, 'Lightintensity': 74}

Published data Successfully: %s {'Waterlevel': 859, 'Lightintensity': 85}

Published data Successfully: %s {'Waterlevel': 593, 'Lightintensity': 85}

Published data Successfully: %s {'Waterlevel': 784, 'Lightintensity': 81}

Published data Successfully: %s {'Waterlevel': 666, 'Lightintensity': 72}

Published data Successfully: %s {'Waterlevel': 722, 'Lightintensity': 67}

Published data Successfully: %s {'Waterlevel': 972, 'Lightintensity': 87}

Published data Successfully: %s {'Waterlevel': 239, 'Lightintensity': 41}

Published data Successfully: %s {'Waterlevel': 780, 'Lightintensity': 98}

Published data Successfully: %s {'Waterlevel': 262, 'Lightintensity': 15}

Published data Successfully: %s {'Waterlevel': 378, 'Lightintensity': 20}

Published data Successfully: %s {'Waterlevel': 974, 'Lightintensity': 67}

Published data Successfully: %s {'Waterlevel': 270, 'Lightintensity': 8}

Published data Successfully: %s {'Waterlevel': 24, 'Lightintensity': 86}

Published data Successfully: %s {'Waterlevel': 73, 'Lightintensity': 42}

Published data Successfully: %s {'Waterlevel': 792, 'Lightintensity': 45}

smart home application

Waterlevel



Lightintensity

