

# Assignment 3

Name: Kripa Karthik

Registration number: 19BEC0549

**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.**

## PYTHON CODE

```
import wiotp.sdk.device
```

```
import time
```

```
import random
```

```
myConfig = {
```

```
    "identity": {
```

```
        "orgId": "cp3p3y",
```

```
        "typeId": "firstdevice",
```

```
        "deviceId": "kris123"
```

```
    },
```

```
    "auth": {
```

```
        "token": "12345678"
```

```
    }
```

```
}
```

```
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:
```

```
    tanklevel=random.randint(0,125)
```

```
    lightint=random.randint(0,100)
```

```
    myData={'waterlevel':tanklevel, 'lightintensity':lightint}
```

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

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

```
    client.commandCallback = myCommandCallback
```

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

```
client.disconnect()
```

## SCREENSHOTS

The screenshot displays the Node-RED web interface. On the left, a palette of nodes is visible, including 'switch', 'slider', 'numeric', 'text input', 'date picker', 'colour picker', 'form', 'text', 'gauge', 'chart', 'audio out', 'notification', 'ui control', and 'template'. The main workspace shows a flow with the following components:

- An **IBM IoT** node (blue) with a 'connected' status indicator.
- Two function nodes (orange) labeled **tanklevel** and **lightintensity**.
- A **msg.payload** node (green) with a dropdown menu.
- A **water tank level** gauge node (blue) and a **light intensity** gauge node (blue).

The flow connects the IBM IoT node to the function nodes, which then connect to the respective gauge nodes. The msg.payload node is also connected to the function nodes.

On the right, the **debug** console shows a list of messages with the following details:

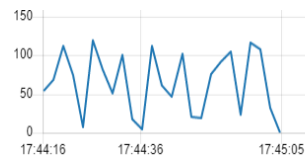
- Timestamp: 7/19/2021, 5:42:39 PM
- Node ID: node: 12f24e34.d20012
- Message: iot-2/type/firstdevice/id/kris123/ev/status/fmt/json : msg.payload : number
- Value: 32

Subsequent messages show values of 38, 13, 24, and 25, all with the same timestamp and node ID.

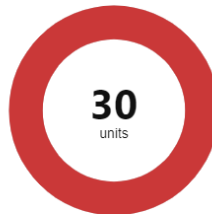
## smarthome application

### sensor data

#### water tank level



#### light intensity



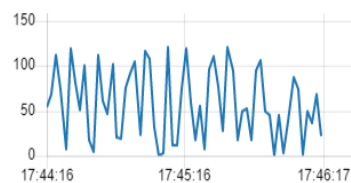
\*IDLE Shell 3.9.6\*

File Edit Shell Debug Options Window Help

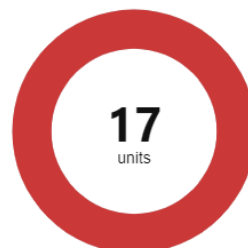
```
'lightintensity': 91}
Published data Successfully: %s {'waterlevel': 50,
'lightintensity': 2}
Published data Successfully: %s {'waterlevel': 53,
'lightintensity': 13}
Published data Successfully: %s {'waterlevel': 18,
'lightintensity': 61}
Published data Successfully: %s {'waterlevel': 94,
'lightintensity': 39}
Published data Successfully: %s {'waterlevel': 106,
'lightintensity': 2}
Published data Successfully: %s {'waterlevel': 50,
'lightintensity': 83}
Published data Successfully: %s {'waterlevel': 45,
'lightintensity': 84}
Published data Successfully: %s {'waterlevel': 0,
'lightintensity': 5}
Published data Successfully: %s {'waterlevel': 45,
'lightintensity': 63}
Published data Successfully: %s {'waterlevel': 3,
'lightintensity': 92}
Published data Successfully: %s {'waterlevel': 38,
'lightintensity': 77}
Published data Successfully: %s {'waterlevel': 87,
'lightintensity': 68}
Published data Successfully: %s {'waterlevel': 75,
'lightintensity': 59}
Published data Successfully: %s {'waterlevel': 1,
'lightintensity': 14}
Published data Successfully: %s {'waterlevel': 49,
'lightintensity': 94}
Published data Successfully: %s {'waterlevel': 37,
'lightintensity': 75}
Published data Successfully: %s {'waterlevel': 69,
'lightintensity': 48}
Published data Successfully: %s {'waterlevel': 22,
'lightintensity': 17}
```

### sensor data

#### water tank level



#### light intensity



```
import wiotp.sdk.device
import time
import random
myConfig = {
    "identity": {
        "orgId": "cp3p3y",
        "typeId": "firstdevice",
        "deviceId": "krisl23"
    },
    "auth": {
        "token": "12345678"
    }
}

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:
    tanklevel=random.randint(0,125)
    lightint=random.randint(0,100)
    myData={'waterlevel':tanklevel, 'lightintensity':lightint}
    client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0,
    print("Published data Successfully: %s", myData)
    client.commandCallback = myCommandCallback
    time.sleep(2)
client.disconnect()
|
```