

### SPRINT-3

PYTHON CODE:

```
import wiotp.sdk.device
import time
import os
import datetime
import random

myConfig = {
    "identity":{
        "orgId":"jpj8ce",
        "typeId":"NodeMCU",
        "deviceId":"0001"
    },
    "auth":{
        "token":"12345678"
    }
}

client = wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None)
client.connect()

def myCommandCallback(cmd):
    print("Message received from IBM IoT platform: %s" % cmd.data['command'])
    m=cmd.data['command']
    if(m=="show"):
        print("Output is displayed")
    elif(m=="hide"):
        print("Output is not displayed")
    print(" ")
```

while True:

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

```
    temperature=random.randint(0,60)
```

```
    ph=random.randint(1,14)
```

```
    myData={'toxic':toxic,'temperature':temperature,'ph':ph}
```

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

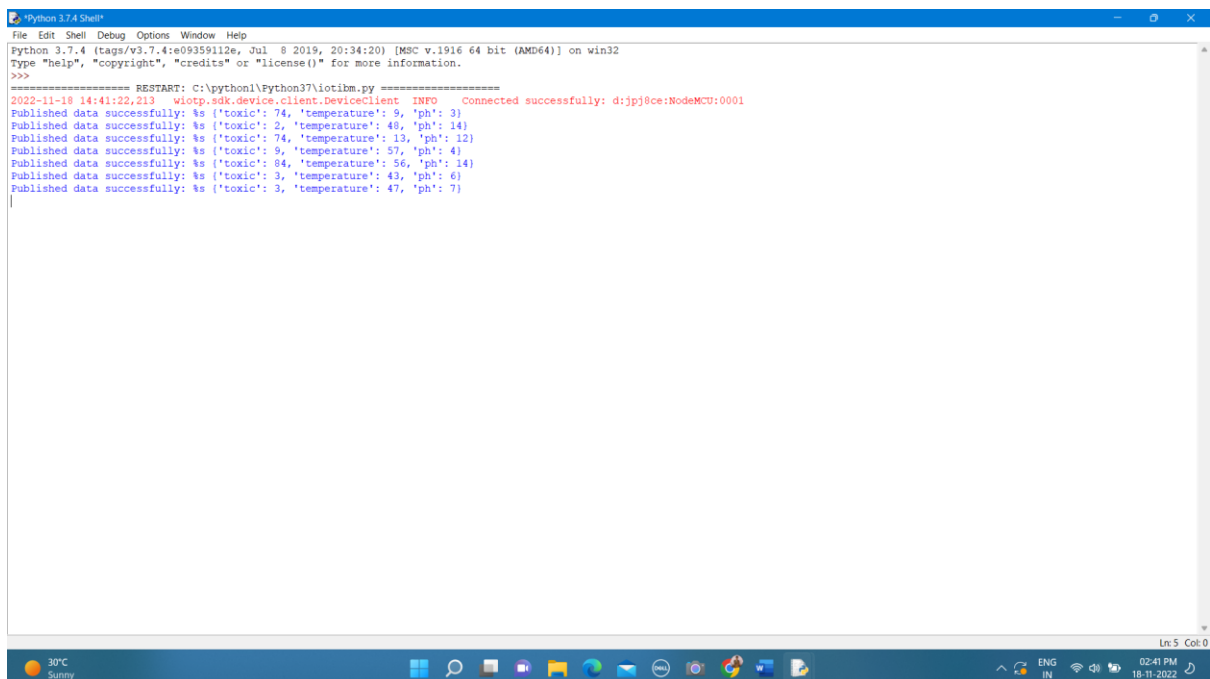
```
    print("Published data successfully: %s", myData)
```

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

```
    client.commandCallback = myCommandCallback
```

```
    client.disconnect()
```

PYTHON OUTPUT:



```
Python 3.7.4 Shell
File Edit Shell Debug Options Window Help
Python 3.7.4 [tags/v3.7.4:09359112e, Jul 8 2019, 20:34:20] [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\python1\Python37\iotlm.py =====
2022-11-18 14:41:22,213 wiotp.sdk.device.client.DeviceClient INFO Connected successfully: d:jpj8ce:NodeMCU:0001
Published data successfully: %s {'toxic': 74, 'temperature': 9, 'ph': 3}
Published data successfully: %s {'toxic': 2, 'temperature': 48, 'ph': 14}
Published data successfully: %s {'toxic': 74, 'temperature': 13, 'ph': 12}
Published data successfully: %s {'toxic': 9, 'temperature': 57, 'ph': 4}
Published data successfully: %s {'toxic': 94, 'temperature': 56, 'ph': 14}
Published data successfully: %s {'toxic': 3, 'temperature': 43, 'ph': 6}
Published data successfully: %s {'toxic': 3, 'temperature': 47, 'ph': 7}
|
```

## COMPARISON B/W PYTHON AND NODE-RED:

The image displays a side-by-side comparison of two programming environments: Python 3.7.4 Shell and Node-RED.

**Python 3.7.4 Shell (Left):** The terminal window shows a continuous stream of JSON data being published successfully. The data is structured as follows:

```
Published data successfully: { 'toxic': 42, 'temperature': 0, 'ph': 3 }
Published data successfully: { 'toxic': 32, 'temperature': 26, 'ph': 8 }
Published data successfully: { 'toxic': 13, 'temperature': 48, 'ph': 13 }
Published data successfully: { 'toxic': 37, 'temperature': 15, 'ph': 11 }
Published data successfully: { 'toxic': 60, 'temperature': 29, 'ph': 11 }
Published data successfully: { 'toxic': 40, 'temperature': 26, 'ph': 9 }
Published data successfully: { 'toxic': 32, 'temperature': 51, 'ph': 5 }
Published data successfully: { 'toxic': 7, 'temperature': 28, 'ph': 1 }
Published data successfully: { 'toxic': 37, 'temperature': 8, 'ph': 9 }
Published data successfully: { 'toxic': 5, 'temperature': 40, 'ph': 3 }
Published data successfully: { 'toxic': 0, 'temperature': 41, 'ph': 11 }
Published data successfully: { 'toxic': 14, 'temperature': 19, 'ph': 14 }
Published data successfully: { 'toxic': 85, 'temperature': 16, 'ph': 12 }
Published data successfully: { 'toxic': 27, 'temperature': 44, 'ph': 8 }
Published data successfully: { 'toxic': 22, 'temperature': 45, 'ph': 14 }
Published data successfully: { 'toxic': 31, 'temperature': 26, 'ph': 4 }
Published data successfully: { 'toxic': 88, 'temperature': 29, 'ph': 11 }
Published data successfully: { 'toxic': 78, 'temperature': 20, 'ph': 13 }
Published data successfully: { 'toxic': 1, 'temperature': 16, 'ph': 2 }
Published data successfully: { 'toxic': 28, 'temperature': 55, 'ph': 2 }
Published data successfully: { 'toxic': 90, 'temperature': 44, 'ph': 3 }
Published data successfully: { 'toxic': 100, 'temperature': 17, 'ph': 10 }
Published data successfully: { 'toxic': 97, 'temperature': 22, 'ph': 3 }
Published data successfully: { 'toxic': 4, 'temperature': 13, 'ph': 11 }
Published data successfully: { 'toxic': 44, 'temperature': 40, 'ph': 2 }
Published data successfully: { 'toxic': 47, 'temperature': 14, 'ph': 1 }
Published data successfully: { 'toxic': 16, 'temperature': 53, 'ph': 5 }
Published data successfully: { 'toxic': 74, 'temperature': 56, 'ph': 3 }
Published data successfully: { 'toxic': 26, 'temperature': 24, 'ph': 12 }
Published data successfully: { 'toxic': 77, 'temperature': 46, 'ph': 5 }
Published data successfully: { 'toxic': 61, 'temperature': 37, 'ph': 6 }
Published data successfully: { 'toxic': 80, 'temperature': 32, 'ph': 14 }
Published data successfully: { 'toxic': 10, 'temperature': 45, 'ph': 8 }
Published data successfully: { 'toxic': 83, 'temperature': 10, 'ph': 12 }
Published data successfully: { 'toxic': 84, 'temperature': 0, 'ph': 12 }
Published data successfully: { 'toxic': 72, 'temperature': 58, 'ph': 5 }
Published data successfully: { 'toxic': 86, 'temperature': 15, 'ph': 2 }
Published data successfully: { 'toxic': 85, 'temperature': 7, 'ph': 3 }
Published data successfully: { 'toxic': 94, 'temperature': 10, 'ph': 3 }
Published data successfully: { 'toxic': 67, 'temperature': 14, 'ph': 8 }
Published data successfully: { 'toxic': 25, 'temperature': 55, 'ph': 10 }
Published data successfully: { 'toxic': 34, 'temperature': 2, 'ph': 10 }
Published data successfully: { 'toxic': 30, 'temperature': 15, 'ph': 9 }
Published data successfully: { 'toxic': 4, 'temperature': 5, 'ph': 5 }
Published data successfully: { 'toxic': 85, 'temperature': 18, 'ph': 8 }
```

**Node-RED (Right):** The Node-RED web interface shows a flow diagram with nodes like inject, switch, http request, and http. The debug console on the right displays the same JSON data as the Python shell, confirming the data flow.