

Sprint – 3

Team ID: PNT2022TMID10955

Python Code:

Sprint - 3

Team ID: PNT2022TMID10221

```
import time import
```

```
sys
```

```
import ibmiotf.application
```

```
import      ibmiotf.device
```

```
import random
```

```
#Provide your IBM Watson Device Credentials
```

```
organization = "lcft5g" deviceType = "Final"
```

```
deviceId = "Hello" authMethod = "token"
```

```
authToken = "8300113450"
```

```
try:
```

```
    deviceOptions = {"org": organization, "type": deviceType, "id": deviceId,  
"auth-method": authMethod, "auth-token": authToken} deviceCli
```

```
    = ibmiotf.device.Client(deviceOptions)
```

```
    #.....
```

```

except Exception as e:
    print("Caught
exception connecting
device: %s" % str(e))
    sys.exit()
# Connect and send a datapoint "hello" with value "world" into the cloud as an
event of type "greeting" 10 times deviceCli.connect()

```

```

while True:

```

```

    #Get Sensor Data from DHT11 temp=random.randint(0,100)

```

```

    Humid=random.randint(0,100)

```

```

    Gas=random.randint(0,100)

```

```

    data = { 'temp' : temp, 'Humid': Humid,'Gas':gas }

```

```

    #print      data      def

```

```

    myOnPublishCallback():

```

```

        print ("Published Temperature = %s C" % temp, "Humidity = %s %" %
Humid, "Gas Concentration = %s"%Gas"to IBM Watson")

```

```

        success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0,
on_publish=myOnPublishCallback) if

```

```

        not success:

```

```

            print("Not connected to IoT")

```

```

        time.sleep(10)

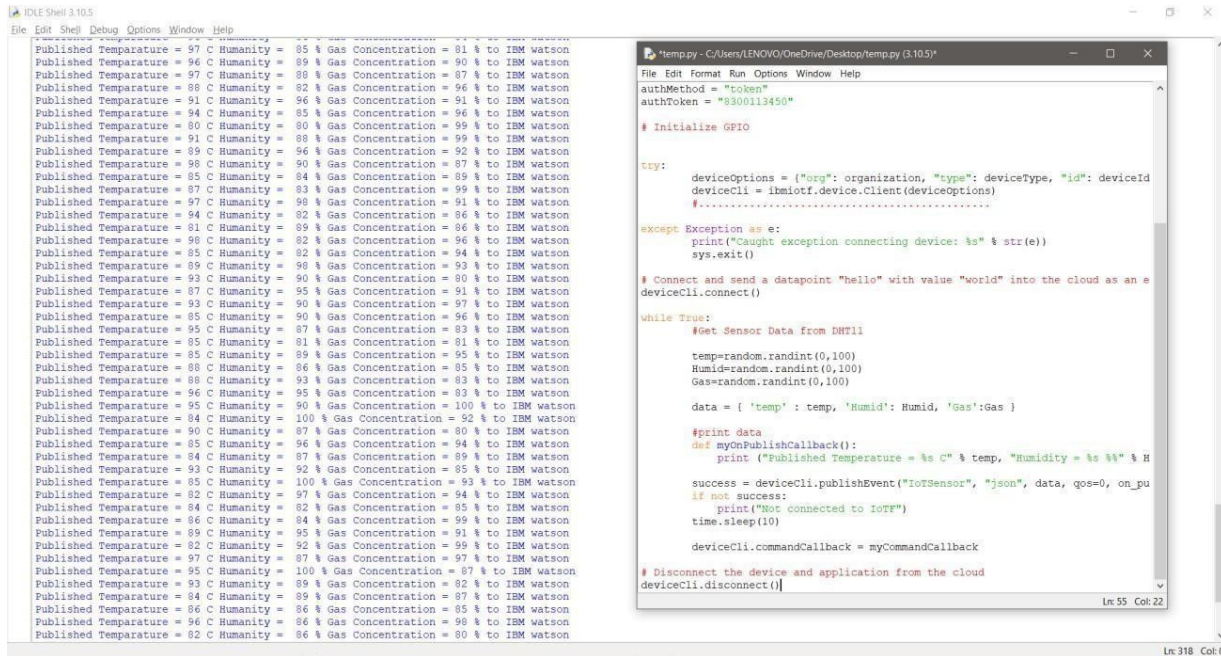
```

```
deviceCli.commandCallback = myCommandCallback
```

Disconnect the device and application from the cloud

```
deviceCli.disconnect()
```

Output:



The screenshot shows a Python IDE with two windows. The left window displays a list of 40 log entries, each showing published sensor data (Temperature, Humidity, Gas Concentration) for an IBM Watson IoT device. The right window shows the Python script that generates this output. The script includes comments for initialization, data generation, publishing, and disconnection.

```
Published Temperature = 97 C Humanity = 85 % Gas Concentration = 81 % to IBM watson
Published Temperature = 96 C Humanity = 89 % Gas Concentration = 90 % to IBM watson
Published Temperature = 97 C Humanity = 88 % Gas Concentration = 87 % to IBM watson
Published Temperature = 88 C Humanity = 82 % Gas Concentration = 96 % to IBM watson
Published Temperature = 91 C Humanity = 96 % Gas Concentration = 91 % to IBM watson
Published Temperature = 94 C Humanity = 85 % Gas Concentration = 96 % to IBM watson
Published Temperature = 80 C Humanity = 90 % Gas Concentration = 99 % to IBM watson
Published Temperature = 91 C Humanity = 88 % Gas Concentration = 99 % to IBM watson
Published Temperature = 89 C Humanity = 96 % Gas Concentration = 92 % to IBM watson
Published Temperature = 96 C Humanity = 90 % Gas Concentration = 87 % to IBM watson
Published Temperature = 85 C Humanity = 84 % Gas Concentration = 89 % to IBM watson
Published Temperature = 87 C Humanity = 83 % Gas Concentration = 99 % to IBM watson
Published Temperature = 97 C Humanity = 98 % Gas Concentration = 91 % to IBM watson
Published Temperature = 94 C Humanity = 82 % Gas Concentration = 96 % to IBM watson
Published Temperature = 81 C Humanity = 89 % Gas Concentration = 96 % to IBM watson
Published Temperature = 98 C Humanity = 82 % Gas Concentration = 94 % to IBM watson
Published Temperature = 89 C Humanity = 98 % Gas Concentration = 93 % to IBM watson
Published Temperature = 93 C Humanity = 90 % Gas Concentration = 90 % to IBM watson
Published Temperature = 87 C Humanity = 95 % Gas Concentration = 91 % to IBM watson
Published Temperature = 93 C Humanity = 90 % Gas Concentration = 97 % to IBM watson
Published Temperature = 85 C Humanity = 90 % Gas Concentration = 96 % to IBM watson
Published Temperature = 95 C Humanity = 87 % Gas Concentration = 83 % to IBM watson
Published Temperature = 85 C Humanity = 81 % Gas Concentration = 81 % to IBM watson
Published Temperature = 85 C Humanity = 89 % Gas Concentration = 95 % to IBM watson
Published Temperature = 86 C Humanity = 86 % Gas Concentration = 85 % to IBM watson
Published Temperature = 88 C Humanity = 93 % Gas Concentration = 83 % to IBM watson
Published Temperature = 96 C Humanity = 95 % Gas Concentration = 83 % to IBM watson
Published Temperature = 90 C Humanity = 90 % Gas Concentration = 100 % to IBM watson
Published Temperature = 84 C Humanity = 100 % Gas Concentration = 92 % to IBM watson
Published Temperature = 90 C Humanity = 87 % Gas Concentration = 80 % to IBM watson
Published Temperature = 85 C Humanity = 96 % Gas Concentration = 94 % to IBM watson
Published Temperature = 84 C Humanity = 87 % Gas Concentration = 89 % to IBM watson
Published Temperature = 93 C Humanity = 92 % Gas Concentration = 95 % to IBM watson
Published Temperature = 85 C Humanity = 100 % Gas Concentration = 93 % to IBM watson
Published Temperature = 82 C Humanity = 97 % Gas Concentration = 94 % to IBM watson
Published Temperature = 84 C Humanity = 82 % Gas Concentration = 85 % to IBM watson
Published Temperature = 86 C Humanity = 84 % Gas Concentration = 99 % to IBM watson
Published Temperature = 89 C Humanity = 95 % Gas Concentration = 91 % to IBM watson
Published Temperature = 82 C Humanity = 92 % Gas Concentration = 99 % to IBM watson
Published Temperature = 87 C Humanity = 97 % Gas Concentration = 97 % to IBM watson
Published Temperature = 95 C Humanity = 100 % Gas Concentration = 87 % to IBM watson
Published Temperature = 89 % Gas Concentration = 82 % to IBM watson
Published Temperature = 84 C Humanity = 89 % Gas Concentration = 97 % to IBM watson
Published Temperature = 86 C Humanity = 86 % Gas Concentration = 85 % to IBM watson
Published Temperature = 86 % Gas Concentration = 98 % to IBM watson
Published Temperature = 82 C Humanity = 86 % Gas Concentration = 80 % to IBM watson
```

```
Temp.py - C:/Users/LENOVO/OneDrive/Desktop/temp.py (3.10.5)
File Edit Format Run Options Window Help

authMethod = "token"
authToken = "9300113450"

# Initialize GPIO

try:
    deviceOptions = {"org": organization, "type": deviceType, "id": deviceId}
    deviceCli = ibmiotf.device.Client(deviceOptions)
    # .....
except Exception as e:
    print("caught exception connecting device: %s" % str(e))
    sys.exit()

# Connect and send a datapoint "hello" with value "world" into the cloud as an e
deviceCli.connect()

while True:
    #Get Sensor Data from DHT11
    temp=random.randint(0,100)
    Humid=random.randint(0,100)
    Gas=random.randint(0,100)

    data = { 'temp': temp, 'Humid': Humid, 'Gas':Gas }

    #print data
    def myOnPublishCallback():
        print ("Published Temperature = %s C" % temp, "Humidity = %s %%" % H

    success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0, on_pu
    if not success:
        print("Not connected to IoTFF")
        time.sleep(10)

    deviceCli.commandCallback = myCommandCallback

# Disconnect the device and application from the cloud
deviceCli.disconnect()
```