

# SPRINT 1

## TEAM ID: PNT2022TMID11075

### REAL TIME RIVER-WATER QUALITY MONITORING AND CONTROL SYSTEM

#### PYTHON CODE:

```
import time
import sys
import ibmiotf.application
import ibmiotf.device
import random
#Provide your IBM Watson Device Credentials
organization = "84708c"
deviceType = "abcd"
deviceId = "12345"
authMethod = "token"
authToken = "12345678"
def myCommandCallback (cmd):
    print ("Command received: %s" % cmd.data['command'])
    status=cmd.data['command']
    if status== "motoron":
        print ("motor is on")
    elif status == "motoroff":
        print ("motor is off")
    else:
        print ("please send proper command")
try:
    deviceOptions = {"org": organization, "type": deviceType, "id": deviceId,
"auth-method":authMethod, "auth-token":authToken}
    deviceCli= ibmiotf.device.Client (deviceOptions)

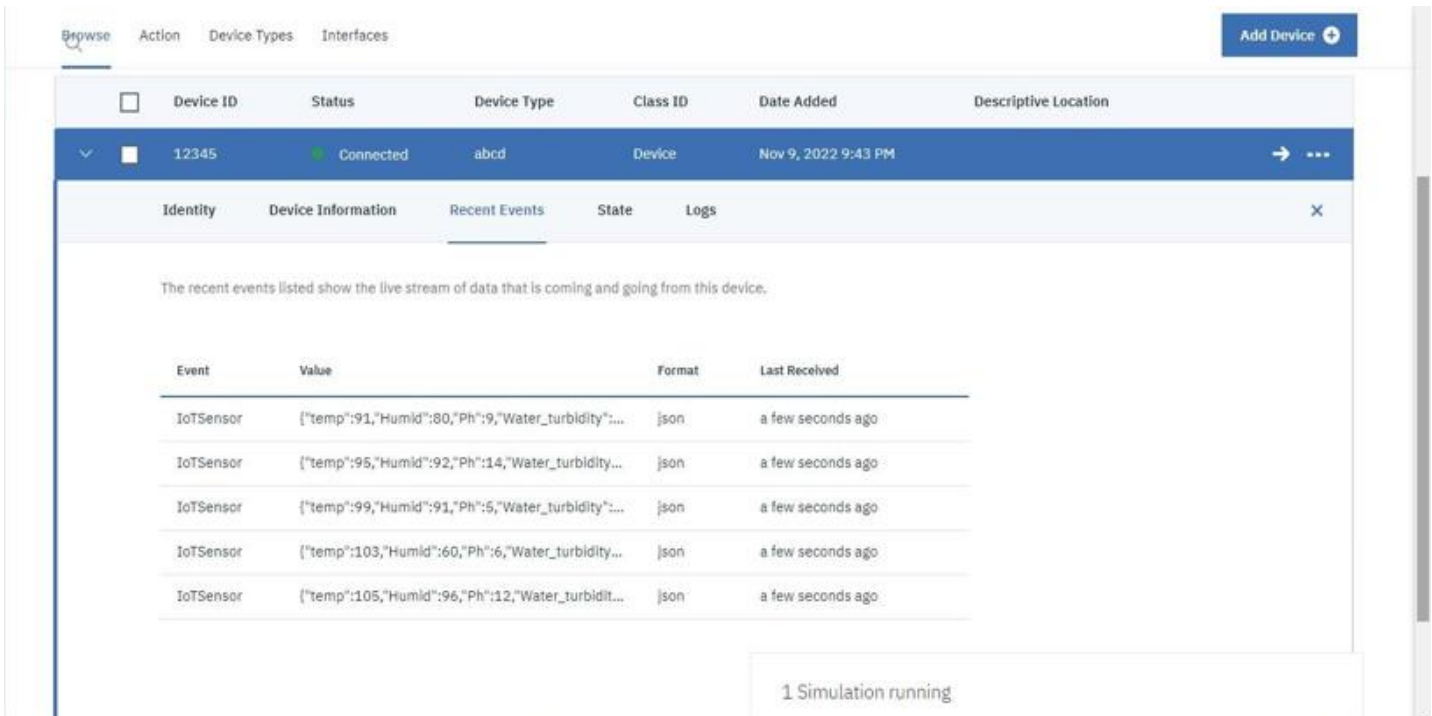
#..
except Exception as e:
    print ("Caught evention connecting device: %s" % str(e))
    sys.exit()
```

```

deviceCli.connect()
while True:
    temp=random.randint (90,110)
    Humid=random.randint (60,100)
    Ph=random.randint (0,14)
    Water_turbidity=random.randint (15,60)
    data = {'temp' : temp, 'Humid': Humid, 'Ph' : Ph, 'Water_turbidity' :
Water_turbidity}
    def myonPublishCallback():
        print ("Published Temperature = %s C" % temp, "Humidity = %s %% " %
Humid,"Ph = %s" % Ph,"Water Turbidity = %s NTU" % Water_turbidity, "to
IBM Watson")
    success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0,
on_publish = myonPublishCallback)
    if not success:
        print("Not connected to IOTF")
        time.sleep (10)
    deviceCli.commandCallback = myCommandCallback
deviceCli.disconnect()

```

## OUTPUT:



The screenshot shows the IBM Watson IoT Platform interface. At the top, there are tabs for 'Browse', 'Action', 'Device Types', and 'Interfaces'. A blue 'Add Device' button is in the top right corner. Below the tabs, a table lists devices. The first device, with ID '12345', is shown as 'Connected' with a green status icon. Below the device list, a 'Recent Events' tab is selected, showing a stream of data events from the device. The events are listed in a table with columns: Event, Value, Format, and Last Received. The events are all from the 'IoTSensor' and are in 'json' format. The values are JSON objects containing temperature, humidity, pH, and water turbidity data. The 'Last Received' column shows 'a few seconds ago' for each event. At the bottom right, a status box indicates '1 Simulation running'.

Event	Value	Format	Last Received
IoTSensor	{"temp":91,"Humid":80,"Ph":9,"Water_turbidity":...	json	a few seconds ago
IoTSensor	{"temp":95,"Humid":92,"Ph":14,"Water_turbidity":...	json	a few seconds ago
IoTSensor	{"temp":99,"Humid":91,"Ph":5,"Water_turbidity":...	json	a few seconds ago
IoTSensor	{"temp":103,"Humid":60,"Ph":6,"Water_turbidity":...	json	a few seconds ago
IoTSensor	{"temp":105,"Humid":96,"Ph":12,"Water_turbidit...	json	a few seconds ago

1 Simulation running