

PYTHON SCRIPT

| | |
|---------------|--|
| Date | 16 NOVEMBER 2022 |
| Team ID | PNT2022TMID15959 |
| Project Name | Project - Real-Time River Water Quality Monitoring and Control System |
| Maximum Marks | 4 Marks |

```

Python 3.7.0 Shell
File Edit Shell Debug Options Window Help
Python 3.7.0 (tags/v3.7.0:1bf9cc5093, Jun 27 2018, 04:59:51) [MSC v.1914 64 bit (AMD64)] on win32
Type "copyright", "credits()" or "license()" for more information.
>>>
===== RESTART: C:\Users\DELL\Downloads\ibmiotpublishsubscribe.py =====
2022-11-17 10:56:55,227 ibmiotf.device.Client INFO Connected successfully: diuo60re:AKASH:1234
Published temp = 69 'C Turbidity = 57 % phvalue = 9 % to IBM Watson
Published temp = 63 'C Turbidity = 18 % phvalue = 12 % to IBM Watson
Command received: lightoff
led is off
Published temp = 75 'C Turbidity = 93 % phvalue = 13 % to IBM Watson
Command received: lighton
led is on
Published temp = 89 'C Turbidity = 71 % phvalue = 11 % to IBM Watson
Published temp = 68 'C Turbidity = 44 % phvalue = 9 % to IBM Watson
Published temp = 74 'C Turbidity = 92 % phvalue = 5 % to IBM Watson
Published temp = 96 'C Turbidity = 60 % phvalue = 13 % to IBM Watson
Published temp = 89 'C Turbidity = 58 % phvalue = 10 % to IBM Watson
Published temp = 60 'C Turbidity = 66 % phvalue = 6 % to IBM Watson
Published temp = 90 'C Turbidity = 42 % phvalue = 10 % to IBM Watson
Published temp = 83 'C Turbidity = 70 % phvalue = 9 % to IBM Watson
Published temp = 80 'C Turbidity = 12 % phvalue = 9 % to IBM Watson
Published temp = 99 'C Turbidity = 72 % phvalue = 5 % to IBM Watson
Published temp = 90 'C Turbidity = 11 % phvalue = 12 % to IBM Watson
Published temp = 60 'C Turbidity = 14 % phvalue = 8 % to IBM Watson
Published temp = 61 'C Turbidity = 60 % phvalue = 8 % to IBM Watson
Published temp = 67 'C Turbidity = 1 % phvalue = 4 % to IBM Watson
Published temp = 93 'C Turbidity = 87 % phvalue = 3 % to IBM Watson
Published temp = 83 'C Turbidity = 36 % phvalue = 12 % to IBM Watson
Published temp = 64 'C Turbidity = 11 % phvalue = 2 % to IBM Watson
Published temp = 83 'C Turbidity = 48 % phvalue = 8 % to IBM Watson
Published temp = 92 'C Turbidity = 86 % phvalue = 4 % to IBM Watson
Published temp = 61 'C Turbidity = 6 % phvalue = 9 % to IBM Watson
Published temp = 94 'C Turbidity = 64 % phvalue = 5 % to IBM Watson
Command received: lightoff
led is off
Published temp = 69 'C Turbidity = 54 % phvalue = 9 % to IBM Watson
Published temp = 85 'C Turbidity = 74 % phvalue = 4 % to IBM Watson
Published temp = 87 'C Turbidity = 84 % phvalue = 4 % to IBM Watson

ibmiotpublishsubscribe.py - C:\Users\DELL\Downloads\ibmiotpublishsubscribe.py (3.7.0)
File Edit Format Run Options Window Help
def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    status=cmd.data['command']
    if status=="lighton":
        print ("led is on")
    else:
        print ("led is off")
    #print(cmd)

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 ew
deviceCli.connect()

while True:
    #Get Sensor Data from DHT11
    temp=random.randint(60,100)
    Turbidity=random.randint(0,100)
    phvalue=random.randint(2,14)

    data = { 'temp': temp, 'Turbidity': Turbidity, 'phvalue': phvalue}
    #print data
    def myOnPublishCallback():
        print ("Published temp = %s 'C" % temp, "Turbidity = %s %%" % Turbidity)

    success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0, on_publish=myOnPublishCallback)
    if not success:
        print("Not connected to IoT")
  
```

CODE:

import time

import sys

import ibmiotf.application

import ibmiotf.device

import random

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

```
organization = "uo60re"
```

```
deviceType = "AKASH"
```

```
deviceId = "1234"
```

```
authMethod = "token"
```

```
authToken = "12345678"
```

```
# Initialize GPIO
```

```
def myCommandCallback(cmd):
```

```
    print("Command received: %s" % cmd.data['command'])
```

```
    status=cmd.data['command']
```

```
    if status=="lighton":
```

```
        print ("led is on")
```

```
    else:
```

```
        print ("led is off")
```

```
    #print(cmd)
```

```
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(60,100)
```

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

```
    phvalue=random.randint(2,14)
```

```
    data = { 'temp' : temp, 'Turbidity': Turbidity,'phvalue': phvalue}
```

```
    #print data
```

```
    def myOnPublishCallback():
```

```
        print ("Published temp = %s 'C" % temp, "Turbidity = %s %" % Turbidity,"phvalue =
%s %" % phvalue, "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()
```