

TEAM ID:	PNT2022TMID06928
PROJECT TITLE:	INDUSTRY SPECIFIC INTELLIGENT FIRE MANAGEMENT SYSTEM

SPRINT 3

```

import time

import sys

import ibmiotf.application

import ibmiotf.device

import random

# Provide your IBM Watson Device Credentials

organization = "gh6uoi"

deviceType = "fire"

deviceId = "fire123"

authMethod = "token"

authToken = "0123456789"

# 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 ()

def myCommandCallback(cmd) :
```

```

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

status = cmd.data['command']

def myCommandCallback(cmd) :

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

    status = cmd.data['command']

def myCommandCallback(cmd) :

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

    status = cmd.data['command']

while True :

    # Get Sensor Data from DHT11

    temp = random.randint ( 0, 100 )

    gas = random.randint ( 60, 200 )

    flame = random.randint ( 60, 200 )

    data = { 'temp' : temp, 'Gas' : gas, 'Flame': flame }

    # print data

    def myOnPublishCallback() :

        print ( "Published Temperature = %s C" % temp, "Gas = %s %" % gas, "Flame = %s %" % flame,
        "to IBM Watson")

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

        if not success :

            print ( "Not connected to IoT" )

        time.sleep ( 1 )

        deviceCli.commandCallback = myCommandCallback

# Initialize GPIO

if temp > 50 :

    print ( "buzzer is on" )

else :

    print ( "buzzer is off" )

```

```
if flame > 100 :
```

```
    print ( "sprinklers are on" )
```

```
else :
```

```
    print ( "sprinklers are off" )
```

```
if gas>100:
```

```
    print ( "exhaust fan is on" )
```

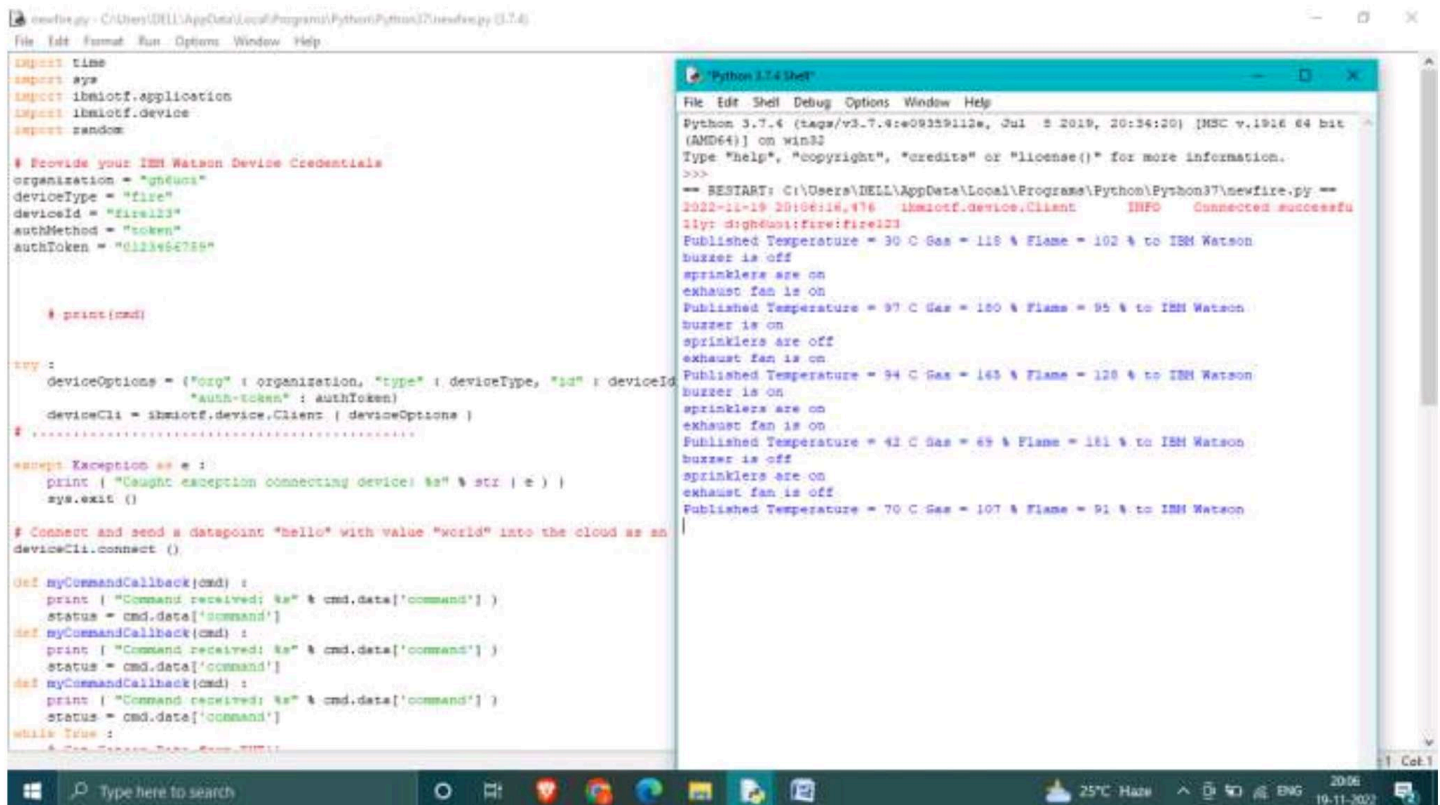
```
else :
```

```
    print ( "exhaust fan is off" )
```

```
# Disconnect the device and application from the cloud
```

```
deviceCli.disconnect ()
```

OUTPUT:



The screenshot displays a Windows desktop environment. On the left, a command prompt window titled 'cmd' shows the execution of a Python script named 'newfire.py'. The script imports modules like 'time', 'sys', 'ibmiotf.application', 'ibmiotf.device', and 'random'. It defines device credentials and a callback function 'myCommandCallback'. The script connects to the cloud and sends a data point 'hello' with the value 'world'. On the right, a 'Python 3.7.4 Shell' window shows the output of the script. The output indicates a successful connection to IBM Watson IoT and displays sensor data: Published Temperature, Gas, and Flame levels, along with the status of the sprinklers and exhaust fan. The output is formatted with color-coded text (green for success, red for error, blue for info).

```
cmd: C:\Users\DELL\AppData\Local\Programs\Python\Python37\newfire.py (3.7.4)
File Edit Format Run Options Window Help

import time
import sys
import ibmiotf.application
import ibmiotf.device
import random

# Provide your IBM Watson Device Credentials
organization = "ghduna"
deviceType = "fire"
deviceId = "fire123"
authMethod = "token"
authToken = "0123456789"

# print(cmd)

try:
    deviceOptions = ("org" : organization, "type" : deviceType, "id" : deviceId,
                    "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
deviceCli.connect ()

def myCommandCallback(cmd):
    print ( "Command received: %s" % cmd.data['command'] )
    status = cmd.data['command']
def myCommandCallback(cmd):
    print ( "Command received: %s" % cmd.data['command'] )
    status = cmd.data['command']
def myCommandCallback(cmd):
    print ( "Command received: %s" % cmd.data['command'] )
    status = cmd.data['command']
while True:
    A Command Received From Device
```

```
Python 3.7.4 Shell
File Edit Shell Debug Options Window Help

Python 3.7.4 (tags/v3.7.4:08359112e, Jul 8 2019, 20:34:20) [MSC v.1916 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
== BESTART: C:\Users\DELL\AppData\Local\Programs\Python\Python37\newfire.py ==
2022-11-19 20:06:16.476 ibmiotf.device.Client INFO Connected successfully
11yr dighdoot:fire:fire123
Published Temperature = 30 C Gas = 118 % Flame = 102 % to IBM Watson
buzzer is off
sprinklers are on
exhaust fan is on
Published Temperature = 37 C Gas = 150 % Flame = 95 % to IBM Watson
buzzer is on
sprinklers are off
exhaust fan is on
Published Temperature = 34 C Gas = 145 % Flame = 128 % to IBM Watson
buzzer is on
sprinklers are on
exhaust fan is on
Published Temperature = 42 C Gas = 89 % Flame = 181 % to IBM Watson
buzzer is off
sprinklers are on
exhaust fan is off
Published Temperature = 70 C Gas = 107 % Flame = 91 % to IBM Watson
|
```

```
Python 3.7.4 Shell
File Edit Shell Debug Options Window Help
buzzer is off
sprinklers are off
exhaust fan is on
Published Temperature = 93 C Gas = 161 % Flame = 154 % to IBM Watson
buzzer is on
sprinklers are on
exhaust fan is on
Published Temperature = 7 C Gas = 187 % Flame = 170 % to IBM Watson
buzzer is off
sprinklers are on
exhaust fan is on
Published Temperature = 87 C Gas = 104 % Flame = 176 % to IBM Watson
buzzer is on
sprinklers are on
exhaust fan is on
Published Temperature = 74 C Gas = 133 % Flame = 91 % to IBM Watson
buzzer is on
sprinklers are off
exhaust fan is on
Published Temperature = 25 C Gas = 191 % Flame = 107 % to IBM Watson
buzzer is off
sprinklers are on
exhaust fan is on
Published Temperature = 73 C Gas = 174 % Flame = 112 % to IBM Watson
buzzer is on
sprinklers are on
exhaust fan is on
Published Temperature = 10 C Gas = 91 % Flame = 162 % to IBM Watson
buzzer is off
sprinklers are on
exhaust fan is off
Published Temperature = 56 C Gas = 65 % Flame = 133 % to IBM Watson
buzzer is on
sprinklers are on
exhaust fan is off
Published Temperature = 30 C Gas = 91 % Flame = 167 % to IBM Watson
buzzer is off
sprinklers are on
exhaust fan is off
Published Temperature = 42 C Gas = 140 % Flame = 172 % to IBM Watson
Ln 27 Col 0
```