Developing a python script

Date	09 November 2022
Team Id	PNT2022TMID43363
Title	Hazardous Area Monitoring for Industrial Plant using IoT

Python Code

Python Script IOT.py - C:\Users\91934\Downloads\Python Script IOT.py (3.7.0)

File Edit Format Run Options Window Help

```
import ibmiotf.application
import ibmiotf.device
import random

#Provide your IBM Watson Device Credentials
organization = "4wj0mx"
deviceType = "NodeMCU"
deviceId = "IoT001"
authMethod = "token"
authToken = "1234567890"

# Initialize GPIO
def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    status=cmd.data['command']
    if status =="lighton":
        print("led in on")
    else :
        print ("led is off")
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)
        data = {'temp' : temp, 'humid' : humid}
        #print data
        def myOnPublishCallback():
            print ("Published Temperature = %s C" % temp, "Humidity:%s" %humid)

        success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0, on_publish=myOnPublishCallback)
        if not success:
            print("Not connected to IoTF")
        time.sleep(1)

        deviceCli.commandCallback = myCommandCallback

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

Generated Values

Publishing the value to IBM cloud