

HAZARDOUS AREA MONITORING FOR INDUSTRIAL PLANT POWERED BY IOT

DEVELOPE THE PYTHON SCRIPT

Team ID-PNT2022TMID14654

We are getting temperature and heart rate of worker as input through the beacon scanner (python code)



```
import wiotp.sdk.device
import time
import random
myConfig = {
    "identity": {
        "orgId": "6yafic",
        "typeId": "Sprint1",
        "deviceId": "SprintID"
    },
    "auth": {
        "token": "sW(1QhEK't)4!jgrjD"
    }
}

def myCommandCallback(cmd):
    print("Message received from IBM IoT Platform: %s" % cmd.data['command'])
    m=cmd.data['command']

client = wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None)
client.connect()

while True:
    temp=random.randint(0,50)
    heart=random.randint(60,100)
    myData={'temperature':temp, 'heartrate':heart}
    client.publishEvent(eventId="sensus", msgFormat="json", data=myData, qos=0, onPublish=None)
    print("Published data Successfully: %s", myData)
    client.commandCallback = myCommandCallback
    time.sleep(5)
client.disconnect()
```

The image shows two windows. The left window is a Python script named 'Sprint1.py' located at 'C:\Users\user\Desktop\Arvin\Sprint1.py (3.7.0)'. The script imports 'wiotp.sdk.device', 'time', and 'random'. It defines a 'myConfig' dictionary with 'orgId', 'typeId', 'deviceId', and 'token'. It then creates a 'DeviceClient' and connects it. A 'while True' loop generates random temperature and heart rate data, publishes it as JSON, and prints success messages. A 'myCommandCallback' function prints received commands. The right window is a 'Python 3.7.0 Shell' showing the script's execution. It displays a restart message, connection logs, and three successful data publication events with sample values for temperature and heart rate.

```
import wiotp.sdk.device
import time
import random
myConfig = {
    "orgId": "6yafic",
    "typeId": "Sprint1",
    "deviceId": "SprintID"
},
"auth": {
    "token": "sW(iQhEK*t)4!jgrjD"
}

def myCommandCallback(cmd):
    print("Message received from IBM IoT Platform: %s" % cmd.data['command'])
    m=cmd.data['command']

client = wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None)
client.connect()

while True:
    temp=random.randint(0,50)
    heart=random.randint(60,100)
    myData={'temperature':temp, 'hearttrate':heart}
    client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0, onPublish=None)
    print("Published data Successfully: %s" % myData)
    client.commandCallback = myCommandCallback
    time.sleep(5)
client.disconnect()
```

```
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\user\Desktop\Arvin\Sprint1.py
2022-11-10 14:51:02.276 wiotp.sdk.device.client.DeviceClient INFO Connecte
d successfully: d:6yafic:Sprint1:SprintIDPublished data Successfully: %s
({'temperature': 14, 'hearttrate': 74})
Published data Successfully: %s ('temperature': 49, 'hearttrate': 89)
Published data Successfully: %s ('temperature': 2, 'hearttrate': 60)
```

Publish Data To The IBM Cloud

The screenshot shows the 'Browse Devices' page in the IBM Watson IoT Platform. The page has a sidebar with navigation icons and a top navigation bar with 'Browse', 'Action', 'Device Types', and 'Interfaces'. A 'Browse Devices' section includes an 'All Devices' button and a 'Diagnose' button. Below this, a table lists devices with columns for Device ID, Status, Device Type, Class ID, Date Added, and Descriptive Location. The table shows five devices: IBM_IoT_1 (Connected), IBM_IoT_2 (Connected), SprintID (Disconnected), f9lw6p (Disconnected), and sensoeid (Disconnected). A 'Device Simulator' toggle is set to 'On'. At the bottom, a pink banner indicates '3 Simulations running'.

Device ID	Status	Device Type	Class ID	Date Added	Descriptive Location
IBM_IoT_1	Connected	IBM_IoT	Device	14 Nov 2022 20:42	
IBM_IoT_2	Connected	IBM_IoT	Device	14 Nov 2022 20:42	
SprintID	Disconnected	Sprint1	Device	14 Nov 2022 20:36	
f9lw6p	Disconnected	IBM_IoT	Device	14	
sensoeid	Disconnected	Sensor	Device	14	

Sensors

Browse	Action	Device Types	Interfaces	Add Device +		
>		Sensorid	Disconnected	Sensor	Device	Nov 4, 2022 12:24 PM
▼		SprintID	Connected	Sprint1	Device	Oct 31, 2022 2:40 PM → ...

Identity	Device Information	Recent Events	State	Logs	X
The recent events listed show the live stream of data that is coming and going from this device.					
Event	Value	Format	Last Received		
status	{"temperature":13,"heartrate":62}	json	a few seconds ago		
status	{"temperature":6,"heartrate":96}	json	a few seconds ago		
status	{"temperature":25,"heartrate":77}	json	a few seconds ago		
status	{"temperature":19,"heartrate":82}	json	a few seconds ago		