

Develop a python script Publish Data to the IBM Cloud

Date	09 November 2022
Team ID	PNT2022TMID28837
Project Name	SMART WASTE MANAGEMENT SYSTEM FOR METROPOLITIAN CITIES
Maximum Marks	4 Marks

The screenshot displays a Jupyter Notebook environment with a single code cell. The code defines a configuration object, a callback function, and a loop that publishes random temperature and humidity data to the IBM Cloud IoT Platform. The output shows successful connections and data publications.

```
myconfig = {
    "identity": {
        "orgId": "ktymlx",
        "typeId": "new",
        "deviceId": "09876"
    },
    "auth": {
        "token": "Kamesh@2002"
    }
}

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(-20,125)
    hum=random.randint(0,100)
    myData={'temperature':temp, 'humidity':hum}
    client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0, onPublish=None)
    print("Published data Successfully: %s", myData)
    client.commandCallback = myCommandCallback
    time.sleep(2)
    client.disconnect()
```

2022-11-09 06:00:17,784 wiotp.sdk.device.client.DeviceClient INFO Connected successfully: d:ktymlx:new:09876
INFO:wiotp.sdk.device.client.DeviceClient:Connected successfully: d:ktymlx:new:09876
Published data Successfully: %s {'temperature': 99, 'humidity': 14}
Published data Successfully: %s {'temperature': 104, 'humidity': 9}
Published data Successfully: %s {'temperature': -14, 'humidity': 86}
Published data Successfully: %s {'temperature': 12, 'humidity': 26}
Published data Successfully: %s {'temperature': 28, 'humidity': 36}
Published data Successfully: %s {'temperature': -16, 'humidity': 83}
Published data Successfully: %s {'temperature': 67, 'humidity': 1}
Published data Successfully: %s {'temperature': -20, 'humidity': 100}

The screenshot displays the IBM Watson IoT Platform interface. At the top, there's a navigation bar with 'Browse', 'Action', 'Device Types', and 'Interfaces'. A search bar is present with the text 'Search by Device ID'. Below this, a table lists devices. The selected device '09876' is shown in a detailed view with tabs for 'Identity', 'Device Information', 'Recent Events', 'State', and 'Logs'. The 'Recent Events' tab is active, showing a table of events with columns: Event, Value, Format, and Last Received. The events are as follows:

Event	Value	Format	Last Received
status	{"temperature":-5,"humidity":23}	json	a few seconds ago
status	{"temperature":2,"humidity":18}	json	a few seconds ago
status	{"temperature":25,"humidity":66}	json	a few seconds ago
status	{"temperature":43,"humidity":83}	json	a few seconds ago

At the bottom of the dashboard, there's a footer with 'Items per page 50' and '1 of 1 page'.

Program :

```
#IBM Watson IOT Platform
#pip install wiotp-sdk
import wiotp.sdk.device
import time
import random
myConfig = {
    "identity": {
        "orgId": "ktymlx",
        "typeId": "new",
        "deviceId": "09876"
    },
    "auth": {
        "token": "Kamesh@2002"
    }
}

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(-20,125)
hum=random.randint(0,100)
myData={'temperature':temp, 'humidity':hum}
client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0,
onPublish=None)
print("Published data Successfully: %s", myData)
client.commandCallback = myCommandCallback
time.sleep(2)
client.disconnect()
```