

Develop A Python Script

Date	17 November 2022
Team ID	PNT2022TMID48061
Project Name	Project -IoT Based Safety Gadget for Child Safety Monitoring and Notification

Develop a python code for publishing the location (latitude and longitude) data to the IBM IoT Platform.

PYTHON SCRIPT:

```
child chip.py - C:/Users/ELCOT/Desktop/child chip.py (3.11.0)
File Edit Format Run Options Window Help

#IBM Watson IOT Platform
#pip install wiotp-sdk
import wiotp.sdk.device
import time
import random
import requests
import math
myConfig = {
    "identity": {
        "orgId": " ",
        "typeId": "chip",
        "deviceId": "0613"
    },
    "auth": {
        "token": " "
    }
}

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:
    name= "hema"
    #in area location

    #latitude= 17.4225176
    #longitude= 78.5458842

    #out area location

    latitude= 17.4225176
    longitude= 78.5458842
    myData={'name' : name, 'lat' :latitude, 'lon' :longitude}
    client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0, onPublish=None)
    print("Data published to IBM IoT platform: ",myData)
    client.commandCallback = myCommandCallback
    time.sleep(5)

client.disconnect()
```

INSTALL WIOTP PACKAGE:

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19044.2251]
(c) Microsoft Corporation. All rights reserved.

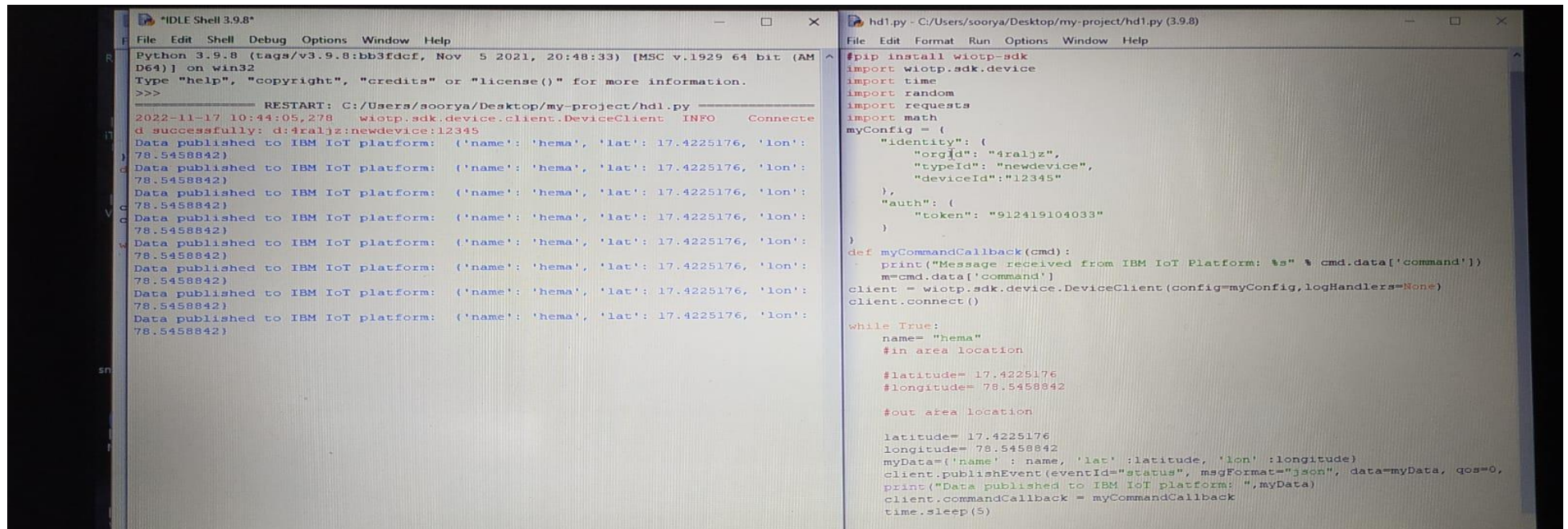
C:\WINDOWS\system32>pip install wiotp-sdk
Requirement already satisfied: wiotp-sdk in c:\python311\lib\site-packages (0.11.0)
Requirement already satisfied: iso8601>=0.1.12 in c:\python311\lib\site-packages (from wiotp-sdk) (1.1.0)
Requirement already satisfied: pytz>=2018.9 in c:\python311\lib\site-packages (from wiotp-sdk) (2022.6)
Requirement already satisfied: pyyaml>=3.13 in c:\python311\lib\site-packages (from wiotp-sdk) (6.0)
Requirement already satisfied: paho-mqtt>=1.5.0 in c:\python311\lib\site-packages (from wiotp-sdk) (1.6.1)
Requirement already satisfied: requests>=2.21.0 in c:\python311\lib\site-packages (from wiotp-sdk) (2.28.1)
Requirement already satisfied: requests-toolbelt>=0.8.0 in c:\python311\lib\site-packages (from wiotp-sdk) (0.10.1)
Requirement already satisfied: charset-normalizer<3,>=2 in c:\python311\lib\site-packages (from requests>=2.21.0->wiotp-sdk) (2.1.1)
Requirement already satisfied: idna<4,>=2.5 in c:\python311\lib\site-packages (from requests>=2.21.0->wiotp-sdk) (3.4)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\python311\lib\site-packages (from requests>=2.21.0->wiotp-sdk) (1.26.12)
Requirement already satisfied: certifi>=2017.4.17 in c:\python311\lib\site-packages (from requests>=2.21.0->wiotp-sdk) (2022.9.24)

C:\WINDOWS\system32>
```

```
wiotp.py - C:/Users/ELCOT/Desktop/wiotp.py (3.11.0)
File Edit Format Run Options Window Help
import wiotp

IDLE Shell 3.11.0
File Edit Shell Debug Options Window Help
Python 3.11.0 (main, Oct 24 2022, 18:26:48) [MSC v.1933 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
>>> ===== RESTART: C:/Users/ELCOT/Desktop/wiotp.py =====
>>> import wiotp
>>>
```

OUTPUT :



The image shows two side-by-side windows from an IDE. The left window, titled 'IDLE Shell 3.9.8', displays the output of a Python script. It shows a restart message and a series of log entries indicating successful connections and data publishing to the IBM IoT platform. The right window, titled 'hd1.py - C:/Users/soorya/Desktop/my-project/hd1.py (3.9.8)', shows the source code of the script. The code imports necessary modules, defines a configuration object, and sets up a command callback to simulate data publishing.

```
Python 3.9.8 (tags/v3.9.8:bb3fcdcf, Nov 5 2021, 20:48:33) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
RESTART: C:/Users/soorya/Desktop/my-project/hd1.py
2022-11-17 10:44:05,278 wiotp.sdk.device.client.DeviceClient INFO Connecte
d successfully: d:4raljz:newdevice:12345
Data published to IBM IoT platform: ('name': 'hema', 'lat': 17.4225176, 'lon':
78.5458842)
Data published to IBM IoT platform: ('name': 'hema', 'lat': 17.4225176, 'lon':
78.5458842)
Data published to IBM IoT platform: ('name': 'hema', 'lat': 17.4225176, 'lon':
78.5458842)
Data published to IBM IoT platform: ('name': 'hema', 'lat': 17.4225176, 'lon':
78.5458842)
Data published to IBM IoT platform: ('name': 'hema', 'lat': 17.4225176, 'lon':
78.5458842)
Data published to IBM IoT platform: ('name': 'hema', 'lat': 17.4225176, 'lon':
78.5458842)
Data published to IBM IoT platform: ('name': 'hema', 'lat': 17.4225176, 'lon':
78.5458842)
Data published to IBM IoT platform: ('name': 'hema', 'lat': 17.4225176, 'lon':
78.5458842)
Data published to IBM IoT platform: ('name': 'hema', 'lat': 17.4225176, 'lon':
78.5458842)

#pip install wiotp-sdk
import wiotp.sdk.device
import time
import random
import requests
import math

myConfig = {
    "identity": {
        "orgId": "4raljz",
        "typeId": "newdevice",
        "deviceId": "12345"
    },
    "auth": {
        "token": "912419104033"
    }
}

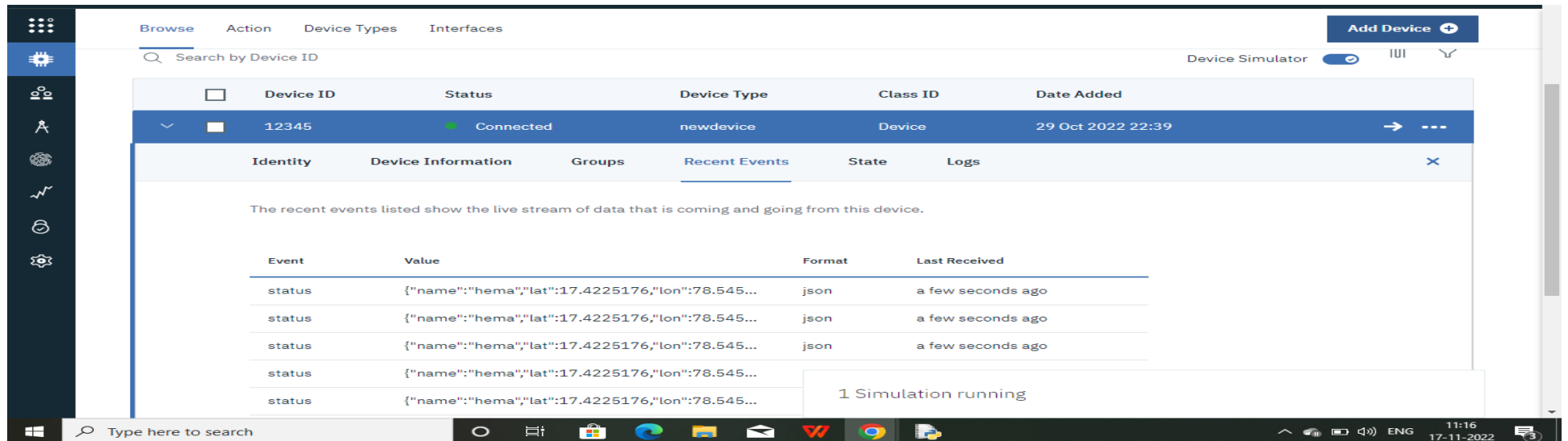
def myCommandCallback(cmd):
    print("Message received from IBM IoT Platform: %s" % cmd.data['command'])
    m=cmd.data['command']
    client = wiotp.sdk.device.DeviceClient(config=myConfig,logHandlers=None)
    client.connect()

while True:
    name= "hema"
    #in area location

    #latitude= 17.4225176
    #longitude= 78.5458842

    #out area location

    latitude= 17.4225176
    longitude= 78.5458842
    myData={'name' : name, 'lat' :latitude, 'lon' :longitude}
    client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0,
    print("Data published to IBM IoT platform: ",myData)
    client.commandCallback = myCommandCallback
    time.sleep(5)
```



The image shows the IBM IoT Platform Device Simulator interface. The top navigation bar includes 'Browse', 'Action', 'Device Types', and 'Interfaces'. A search bar is present with the text 'Search by Device ID'. The main content area displays a table of devices. The first device, with ID '12345', is shown as 'Connected' and of type 'newdevice'. Below the table, there is a section for 'Recent Events' which shows a stream of data. A notification at the bottom indicates '1 Simulation running'.

Device ID	Status	Device Type	Class ID	Date Added
12345	Connected	newdevice	Device	29 Oct 2022 22:39

Event	Value	Format	Last Received
status	{"name":"hema","lat":17.4225176,"lon":78.545...	json	a few seconds ago
status	{"name":"hema","lat":17.4225176,"lon":78.545...	json	a few seconds ago
status	{"name":"hema","lat":17.4225176,"lon":78.545...	json	a few seconds ago
status	{"name":"hema","lat":17.4225176,"lon":78.545...		
status	{"name":"hema","lat":17.4225176,"lon":78.545...		

1 Simulation running