

Smart Waste Management System for Metropolitan Cities

Project Objectives - Develop The Python Script:

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

Code:

```
import wiotp.sdk
import time
import random
myConfig = {
    "identity": {
        "orgId": "nhpwjc",
        "typeId": "NodeMCU",
        "deviceId": "12345"
    },
    "auth": {
        "token": "123456789"
    }
}
lat="13.167589"
lon="80.248510"
name="point1"
icon="fa-trash-o"
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(0,100)
    if temp>60:
        icon="fa-trash"
    else:
        icon = "fa-trash-o"

myData={"Name":name,"Latitude":lat,"Longitude":lon,"Icon":icon,"FillPercent":temp}
    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()
```

Output:

PyCharm output:

The screenshot shows the PyCharm IDE with a Python script named `main.py` and its console output. The script defines a `main` function that connects to the IBM IoT Platform, publishes data, and handles commands. The console output shows the successful execution of the script, including the connection status and the published data.

```

def main():
    """Main function to connect to IBM IoT Platform and publish data"""
    # Define device configuration
    config = {
        "token": "123456789",
        "lat": "13.167589",
        "long": "80.248510"
    }

    # Define command callback function
    def myCommandCallback(cmd):
        print("Message received from IBM IoT Platform: %s" % cmd.data['command'])
        #cmd.data['command']

    # Create device client
    client = iotdp.sdk.device.DeviceClient(config=myConfig, loginHandlers=None)
    client.connect()

    # Publish data
    while True:
        temp = random.randint(0, 100)
        myData = {'Latitude': lat, 'Longitude': long, 'FillPercent': temp}
        client.publishEvent(event="status", msg=base64.b64encode(json.dumps(myData).encode('utf-8')).decode('utf-8'), qos=0, onPublish=None)
        print("Published data Successfully: %s" % myData)
        client.commandCallback = myCommandCallback
        time.sleep(5)
    client.disconnect()

if __name__ == '__main__':
    main()

```

Run console output:

```

"C:\Users\Arun K\PycharmProjects\pythonProject2\venv\Scripts\python.exe" "C:\Users\Arun K\PycharmProjects\pythonProject2\main.py"
Published data Successfully: %s {'Latitude': '13.167589', 'Longitude': '80.248510', 'FillPercent': 71}
2022-11-11 18:43:15.532 iotdp.sdk.device.client.DeviceClient INFO Connected successfully: dnhpwj:NodeMCU:12345
Published data Successfully: %s {'Latitude': '13.167589', 'Longitude': '80.248510', 'FillPercent': 100}
Published data Successfully: %s {'Latitude': '13.167589', 'Longitude': '80.248510', 'FillPercent': 31}

```

IBM IOT Output:

The screenshot shows the IBM Watson IoT Platform interface. The top navigation bar includes 'Browse', 'Action', 'Device Types', and 'Interfaces'. The main content area displays a table of devices, with the selected device (ID: 12345) showing its status as 'Connected' and type as 'NodeMCU'. Below the table, the 'Recent Events' tab is active, showing a list of events with columns for Event, Value, Format, and Last Received.

Event	Value	Format	Last Received
status	["Latitude": "13.167589", "Longitude": "80.248510", "FillPercent": 71]	json	a few seconds ago
status	["Latitude": "13.167589", "Longitude": "80.248510", "FillPercent": 100]	json	a few seconds ago
status	["Latitude": "13.167589", "Longitude": "80.248510", "FillPercent": 31]	json	a few seconds ago
status	["Latitude": "13.167589", "Longitude": "80.248510", "FillPercent": 100]	json	a few seconds ago
status	["Latitude": "13.167589", "Longitude": "80.248510", "FillPercent": 31]	json	a few seconds ago