

PROJECT DEVELOPMENT PHASE

SPRINT-3 TEST CASE

Date	08 November 2022
Team ID	PNT2022TMID49497
Project Name	Smart farmer-IOT Enabled Smart Farming Application
Maximum Marks	8 Marks

Recent Event:

The screenshot displays the IBM Watson IoT Platform interface. The top navigation bar includes tabs for 'Browse', 'Action', 'Device Types', and 'Interfaces'. A search bar labeled 'Search by Device ID' is present. The main content area shows a table of devices, with 'demo123' selected. Below the table, the 'Recent Events' tab is active, displaying a list of events with columns for Event, Value, Format, and Last Received. The events are JSON objects containing temperature, humidity, and moisture data.

Device ID	Status	Device Type	Class ID	Date Added
demo123	Connected	raspberrypi	Device	Nov 1, 2022 6:46 PM

Event	Value	Format	Last Received
IoTSensor	{"temp":56,"hum":95,"moist":54}	json	a few seconds ago
IoTSensor	{"temp":59,"hum":35,"moist":77}	json	a few seconds ago
IoTSensor	{"temp":45,"hum":97,"moist":5}	json	a few seconds ago

Node-Red Output:

Smart Agriculture:

Node-RED interface showing a 'smart agriculture' flow. The flow starts with an IBM IoT node, branching into Soil Moisture, Humidity, and Temperature nodes. Each of these branches into a corresponding sensor node (Soil Moisture, Humidity, Temperature) and a msg.payload node. The msg.payload nodes are then connected to a function node labeled 'data', which is connected to an http node. Below this, there are Motor ON and Motor OFF nodes connected to a msg.payload node, which is also connected to an http node. The debug console on the right shows the output of the flow, including the msg.payload values for the sensors and the motor status.

```
graph LR
    IoT[IBM IoT] --> SM[Soil Moisture]
    IoT --> H[Humidity]
    IoT --> T[Temperature]
    SM --> SM_Sensor[Soil Moisture]
    H --> H_Sensor[Humidity]
    T --> T_Sensor[Temperature]
    SM_Sensor --> MP1[msg payload]
    H_Sensor --> MP2[msg payload]
    T_Sensor --> MP3[msg payload]
    MP1 --> F[data]
    MP2 --> F
    MP3 --> F
    F --> HTTP1[http]
    MotorON[Motor ON] --> MP4[msg payload]
    MotorOFF[Motor OFF] --> MP4
    MP4 --> HTTP2[http]
```

Debug console output:

```
11/8/2022, 11:45:41 AM node: f9e68ea00925e813 iot-
2/type/raspberrypi/d/demo123/ev/IoTSensor/fmt/json :
msg.payload : number
72
11/8/2022, 11:45:46 AM node: f9e68ea00925e813 iot-
2/type/raspberrypi/d/demo123/ev/IoTSensor/fmt/json :
msg.payload : number
40
11/8/2022, 11:45:46 AM node: f9e68ea00925e813 iot-
2/type/raspberrypi/d/demo123/ev/IoTSensor/fmt/json :
msg.payload : number
92
11/8/2022, 11:45:46 AM node: f9e68ea00925e813 iot-
2/type/raspberrypi/d/demo123/ev/IoTSensor/fmt/json :
msg.payload : number
60
```

OpenWeather:

Node-RED interface showing an 'Open Weather' flow. The flow starts with a timestamp node, followed by a Madurai node, then a function node. The function node branches into six parallel paths, each consisting of a function node connected to a msg.payload node. The msg.payload nodes are then connected to a function node labeled 'details', which is connected to an http node. The debug console on the right shows the output of the flow, including the msg.payload values for the weather data and the details.

```
graph LR
    Timestamp[timestamp] --> Madurai[Madurai]
    Madurai --> F1[function]
    F1 --> MP1[msg payload]
    F1 --> MP2[msg payload]
    F1 --> MP3[msg payload]
    F1 --> MP4[msg payload]
    F1 --> MP5[msg payload]
    F1 --> MP6[msg payload]
    MP1 --> F2[function]
    MP2 --> F2
    MP3 --> F2
    MP4 --> F2
    MP5 --> F2
    MP6 --> F2
    F2 --> HTTP[http]
```

Debug console output:

```
11/8/2022, 11:46:14 AM node: 8bec6846f15be2e
msg.payload : string[7]
"Madurai"
11/8/2022, 11:46:14 AM node: de8dc0b392a0bf7d
msg.payload : string[4]
"Haze"
11/8/2022, 11:46:14 AM node: 746d3e8946df3717
msg.payload : number
30
11/8/2022, 11:46:14 AM node: cb72fe3d0558ed7f
msg.payload : number
66
11/8/2022, 11:46:14 AM node: eee3ce1a01bb184a
msg.payload : number
3.6
```

Output:

The screenshot shows a Windows desktop with a taskbar at the bottom. The taskbar includes the Start button, a search icon, and several pinned application icons: File Explorer, Google Chrome, and a terminal window. The terminal window is titled "Python 3.7.0 Shell" and displays the output of a Python script. The script is a Raspberry Pi client for the IBM Watson IoT platform, which sends sensor data (Temperature, Humidity, and Soil Moisture) to the cloud. The terminal output shows the script running successfully, with the sensor data being sent to the IBM Watson cloud. The script is located at F:\cse4\Downloads\ibmiotpublishsubscribe (2).py (3.7.0).

```

import time
import sys
import ibmiotf.application
import ibmiotf.device
import random

#Provide your IBM Watson organization = "puubdh"
organization = "puubdh"
deviceType = "raspberrypi"
deviceId = "demol23"
authMethod = "token"
authToken = "1234567890"

# Initialize GPIO

def myCommandCallback(cmd):
    print("Command received: {}".format(cmd))
    status = cmd.data.get('status')
    if status == "motoron":
        print("Motor is on")
        Published Temperature = 20 C Humidity = 5 % Soil Moisture = 34 to IBM Watson
    elif status == "motoroff":
        print("Motor is off")
        Published Temperature = 7 C Humidity = 29 % Soil Moisture = 18 to IBM Watson
    else:
        print("Invalid command")
    #print(cmd)

try:
    deviceOptions = {
        "device-type": deviceType,
        "device-id": deviceId,
        "auth-method": authMethod,
        "auth-token": authToken
    }
    deviceCli = ibmiotf.device.Client(deviceOptions)

except Exception as e:
    print("Caught exception: {}".format(e))
    sys.exit()

# Connect and send data
deviceCli.connect()

```

Python 3.7.0 Shell

File Edit Shell Debug Options Window Help

Python 3.7.0 (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: F:\cse4\Downloads\ibmiotpublishsubscribe (2).py =====

2022-11-08 11:43:53,821 ibmiotf.device.Client INFO Connected successfully: d:\puubdh\raspberrypi\demol23

Published Temperature = 85 C Humidity = 80 % Soil Moisture = 57 to IBM Watson

Published Temperature = 41 C Humidity = 80 % Soil Moisture = 80 to IBM Watson

Command received: motoron

Motor is on

Published Temperature = 20 C Humidity = 5 % Soil Moisture = 34 to IBM Watson

Command received: motoroff

Motor is off

Published Temperature = 7 C Humidity = 29 % Soil Moisture = 18 to IBM Watson

10 times

Ln: 24 Col: 21