

SPRINT-II

Create Device in IoT Watson and Workflow using Node-Red

| | |
|--------------|--|
| Date | 29 th OCTOBER 2022 |
| Team ID | PNT2022TMID04339 |
| Project Name | Gas Leakage Monitoring and Alerting System |

Device in IoT Watson:

- Creating a device in IBM Watson IoT

IBM Watson IoT Platform

710019106014@smartinternz.com
ID: Otus0f

Browse Action Device Types Interfaces

Add Device

| Device ID | Status | Device Type | Class ID | Date Added |
|-----------|-----------|-------------|----------|---------------------|
| 01 | Connected | ESP32 | Device | Nov 6, 2022 9:53 AM |

Identity Device Information Recent Events State Logs

The recent events listed show the live stream of data that is coming and going from this device.

| Event | Value | Format | Last Received |
|-------|-------------------|--------|-------------------|
| event | {"gas_level":480} | json | a few seconds ago |
| event | {"gas_level":480} | json | a few seconds ago |
| event | {"gas_level":480} | json | a few seconds ago |
| event | {"gas_level":480} | json | a few seconds ago |

○ Creating a workflow in node red and giving connection for web ui.

Temperature

MESSAGE
GAS LEAKAGE ALERT

Sensor Data

gauge

695 ppm

0 2000

The screenshot displays the Node-RED web interface in a browser. The main workspace shows a flow named 'Flow 1' with the following components:

- IBM IoT** node (blue) with a 'connected' status, which branches into two paths.
- Status** function node (orange) connected to a **msg.payload** node (green).
- Gas** function node (orange) connected to a **gauge** node (blue).
- [get] /Data** node (yellow) connected to a **function** node (orange), which is then connected to an **http (200)** node (yellow).
- [get] /Sensordata** node (yellow) connected to a **function** node (orange), which is then connected to an **http** node (yellow).

The left sidebar shows a 'filter nodes' search bar and a 'network' category with nodes like 'mqtt in', 'mqtt out', 'http in', and 'http response'. The right sidebar shows a 'debug' console with a list of messages:

```
14/11/2022, 3:39:08 pm node: 40cc9a64c4fe1f96  
iot-2/type/ESP32/id/01/evt/event/fmt/json : msg.payload  
: Object  
  { gas_level: 241 }  
14/11/2022, 3:39:10 pm node: 40cc9a64c4fe1f96  
iot-2/type/ESP32/id/01/evt/event/fmt/json : msg.payload  
: Object  
  { gas_level: 241 }  
14/11/2022, 3:39:11 pm node: 40cc9a64c4fe1f96  
iot-2/type/ESP32/id/01/evt/event/fmt/json : msg.payload  
: Object  
  { gas_level: 241 }  
14/11/2022, 3:39:12 pm node: 40cc9a64c4fe1f96  
iot-2/type/ESP32/id/01/evt/event/fmt/json : msg.payload  
: Object  
  { gas_level: 241 }
```

Output:

