

SPRINT-II

TOPIC: CREATE DEVICE IN THE IOT WATSON PLATFORM AND WORKFLOW FOR IOT SCENARIOS USING LOCAL NODE RED

Create The IBM Watson IoT Platform And A Device

The screenshot shows the IBM Watson IoT Platform dashboard. 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 with columns: Device ID, Status, Device Type, Class ID, and Date Added. The first row shows device ID 1814, status 'Connected', device type 'ESP32', class ID 'Device', and date added 'Nov 13, 2022 2:48 PM'. Below the table, a detailed view for device 1814 is shown, including fields for Device ID, Device Type, Date Added, Added By, and Connection Status. The connection status is 'Connected' with details: Connection Time: Nov 19, 2022 7:42 PM and Client Address: 42.111.148.36. A notification at the bottom right states '1 Simulation running'.

| Device ID | Status | Device Type | Class ID | Date Added |
|-----------|-----------|-------------|----------|----------------------|
| 1814 | Connected | ESP32 | Device | Nov 13, 2022 2:48 PM |

| Field | Value |
|-------------------|---|
| Device ID | 1814 |
| Device Type | ESP32 |
| Date Added | Nov 13, 2022 2:48 PM |
| Added By | yokeshjinendiran548@gmail.com |
| Connection Status | Connected Connection Time: Nov 19, 2022 7:42 PM Client Address: 42.111.148.36 |

The screenshot shows the IBM Watson IoT Platform dashboard, specifically the 'Recent Events' tab for device ID 1814. The top navigation bar and table are the same as in the previous screenshot. The 'Recent Events' section displays a message: 'The recent events listed show the live stream of data that is coming and going from this device.' Below this, a table lists recent events with columns: Event, Value, Format, and Last Received. The events are status updates with JSON payloads containing soil moisture, temperature, and humidity data. A notification at the bottom right states '1 Simulation running'.

| Event | Value | Format | Last Received |
|--------|---|--------|-------------------|
| status | {"soil_moisture":61,"temperature":43,"humidity":...} | json | a few seconds ago |
| status | {"soil_moisture":90,"temperature":70,"humidity":...} | json | a few seconds ago |
| status | {"soil_moisture":22,"temperature":-20,"humidity":...} | json | a few seconds ago |
| status | {"soil_moisture":99,"temperature":116,"humidity":...} | json | a few seconds ago |
| status | {"soil_moisture":85,"temperature":120,"humidity":...} | json | a few seconds ago |

Create Node-RED Service

The screenshot displays the Node-RED web interface in a browser. The address bar shows the URL: `169.51.203.65:30772/red/#flow/775a438e00663222`. The interface includes a left sidebar with node categories (common, function), a central workspace with a flow diagram, and a right sidebar with a debug console.

Flow Diagram:

- Input:** An `IBM IoT` node (connected) feeds into three function nodes: `Soil Moisture`, `Humidity`, and `Temperature`.
- Processing:** These function nodes connect to corresponding `msg.payload` nodes. A `switch` node is connected to the `msg.payload` nodes. The `switch` node connects to an `http request` node.
- Output:** The `http request` node connects to an `http` node. The `http` node connects to an `msg.payload` node.
- Control:** A `[get] /data` node connects to a `data` function node, which connects to an `http` node. The `http` node connects to an `msg.payload` node.
- Motor Control:** `Motor On` and `Motor Off` nodes connect to an `msg.payload` node. A `[get] /command` node connects to an `http` node, which connects to an `msg.payload` node.

Debug Console:

```
{ "command": "motoron" }
11/18/2022, 5:03:36 PM node: 9ec1acf858428dea
msg.payload: Object
{ "command": "motoroff" }
11/18/2022, 5:03:37 PM node: 9ec1acf858428dea
msg.payload: Object
{ "command": "motoron" }
11/18/2022, 5:03:38 PM node: 9ec1acf858428dea
msg.payload: Object
{ "command": "motoroff" }
11/18/2022, 5:03:40 PM node: 9ec1acf858428dea
msg.payload: Object
{ "command": "motoroff" }
11/18/2022, 5:03:40 PM node: 9ec1acf858428dea
msg.payload: Object
{ "command": "motoroff" }
11/18/2022, 5:03:41 PM node: 9ec1acf858428dea
msg.payload: Object
{ "command": "motoroff" }
```