

Use Dashboard Nodes For Creating UI(Web App)

TEAM ID	PNT2022TMID41673
PROJECT TITLE	REAL TIME RIVER WATER QUALITY MONITORING AND CONTROL SYSTEM

Step 1: Sending data to the IBM Watson

The screenshot displays the IBM Watson IoT Platform interface. The top navigation bar includes 'Browse', 'Action', 'Device Types', and 'Interfaces'. A sidebar on the left contains various icons for navigation. The main content area shows a table of devices. The selected device, ID 123435, is in a 'Disconnected' state. Below the device information, the 'Recent Events' tab is active, showing a stream of data events. The events are listed in a table with columns: Event, Value, Format, and Last Received. The events are JSON objects containing 'Turbidity' and 'pH_value' data. A status bar at the bottom indicates '1 Simulation running'.

Event	Value	Format	Last Received
data	{"Turbidity":203,"pH_value":8}	json	a few seconds ago
data	{"Turbidity":597,"pH_value":12}	json	a few seconds ago
data	{"Turbidity":110,"pH_value":10}	json	a few seconds ago
data	{"Turbidity":608,"pH_value":9}	json	a few seconds ago
data	{"Turbidity":393,"pH_value":10}	json	a few seconds ago

Step 2: Configure the IBM IOT in the Node-red

The screenshot shows the Node-RED interface with a flow named 'Flow 1'. The flow includes an 'IBM IoT' node (connected), a 'Cloudant' node, and two function nodes labeled 'ph' and 'turbidity'. The 'IBM IoT' node is connected to the 'Cloudant' node, which is then connected to the 'ph' and 'turbidity' function nodes. The 'ph' function node is connected to the 'turbidity' function node, which is then connected to an 'http req' node. The 'http req' node is connected to a '[get] /sensor_data' node. The right sidebar shows the configuration for the 'IBM IoT' node, with fields for Name, API Key, API Token, Server-Name, Scalable, Application ID, Keep Alive, and Use Clean Session.

Flow 1

Edit ibmiot in node > Edit ibmiot node

Properties

- Name: lot api
- API Key: a-jj64y3-w7zkatzbl9
- API Token:
- Server-Name: jj64y3.messaging.internetofthings.ibmcloud.com
- Scalable: ☐
- Application ID:
- Keep Alive: 60 Seconds
- Use Clean Session: ☒

Step 3: Configure the Device in IOT in Node-red

The screenshot shows the Node-RED interface with a flow named 'Flow 1'. The flow includes an 'IBM IoT' node (green) connected to three function nodes: 'ph' (orange), 'Cloudant' (blue), and 'turbidity' (orange). Below these, there is a '[get] /sensor_data' node (green) connected to an 'http re' node (orange). The 'Edit ibmiot in node' configuration panel is open on the right, showing the following properties:

- Authentication: API Key
- API Key: lot api
- Input Type: Device Event
- Device Type: All or +
- Device Id: All or device id e.g. ab12cd231a21
- Event: All or +
- Format: All or json
- QoS: 0
- Name: IBM IoT
- Service: registered

A yellow note at the bottom of the panel states: "Use the Input Type property to configure this node to receive Events sent by IoT Devices, Commands sent to IoT Devices, Status Messages referring to IoT Devices, or Status Messages referring to IoT Applications."

Step 4: Add a new function node to separate the pH value

The screenshot shows the 'Edit function node' configuration panel. The 'Name' field is set to 'ph'. The 'On Message' tab is selected, and the following code is entered in the message editor:

```
1 msg.payload=msg.payload.ph
2 global.set('ph',msg.payload)
3 return msg;
```

Step 5: Add a new function node to separate the Turbidity value

Edit function node

Delete

Cancel

Done

⚙ Properties

⚙

📄

🖨

📌 Name

turbidity

📄

⚙ Setup

On Start

On Message

On Stop

1

msg.payload=msg.payload.turbidity

2

global.set('turbidity',msg.payload)

3

return msg;

Step 6: Add a Gauge from the Node-red Dashboard and Configure Gauge node and pass the pH value to the Gauge node

Edit gauge node

Delete

Cancel

Done

⚙ Properties

⚙

📄

🖨

📁 Group

[Home] River Water Quality Monitoring

✎

📺 Size

auto

☰ Type

Gauge

▼

🏷 Label

pH

🏷 Value format

{{value}}

🏷 Units

units

Range

min

0

max

14

Colour gradient

Sectors

0

...

optional

...

optional

...

14

🔍 debug

📄

⚙

🔍

⌵

▼ all nodes

🗑 all

▶ { ph: 14, turbidity: 0 }

11/13/2022, 11:59:35 AM node: 1c347adebc7ec9d8

iot-2/type/Nodered/id/12345/evt/status/fmt/json :

msg.payload : number

14

11/13/2022, 11:59:35 AM node: 1c347adebc7ec9d8

iot-2/type/Nodered/id/12345/evt/status/fmt/json :

msg.payload : number

0

11/13/2022, 11:59:35 AM node: 1c347adebc7ec9d8

iot-2/type/Nodered/id/12345/evt/status/fmt/json :

msg.payload : Object

▶ { ph: 12, turbidity: 7 }

11/13/2022, 11:59:35 AM node: 1c347adebc7ec9d8

iot-2/type/Nodered/id/12345/evt/status/fmt/json :

msg.payload : number

12

11/13/2022, 11:59:35 AM node: 1c347adebc7ec9d8

iot-2/type/Nodered/id/12345/evt/status/fmt/json :

msg.payload : number

7

Step 7: Add a Gauge from the Node-red Dashboard and Configure Gauge and pass the Turbidity value to the Gauge node

The screenshot shows the 'Edit gauge node' configuration window on the left and the 'debug' console on the right.

Edit gauge node configuration:

- Group:** [Home] River Water Quality Monitoring
- Size:** auto
- Type:** Donut
- Label:** turbidity
- Value format:** {{value}}
- Units:** units
- Range:** min 0, max 10
- Colour gradient:** Green, Yellow, Red
- Sectors:** 0, optional, optional, 10
- Class:** Optional CSS class name(s) for widget

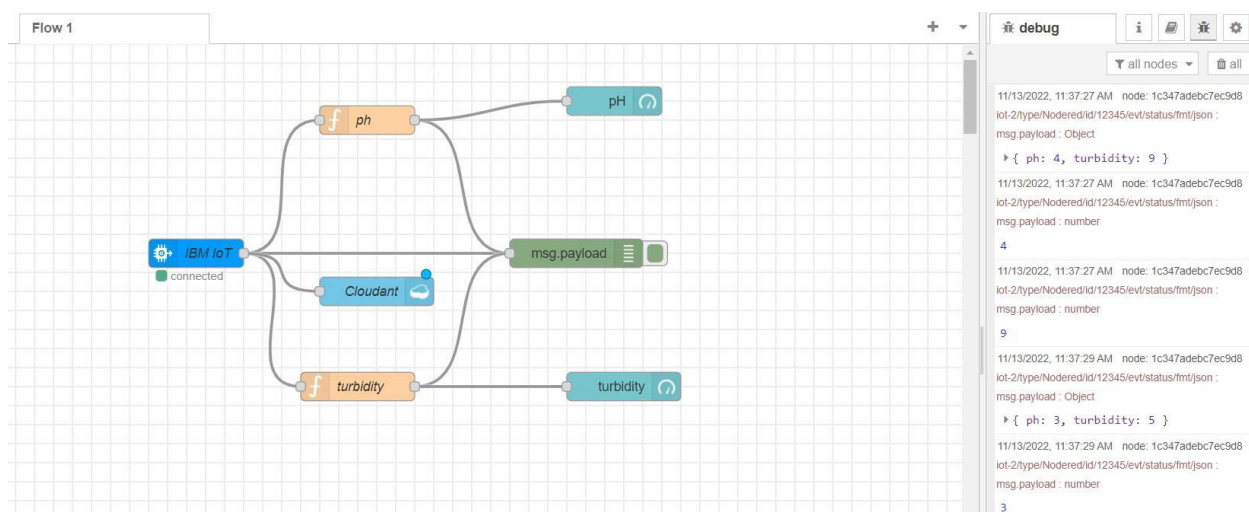
debug console:

```

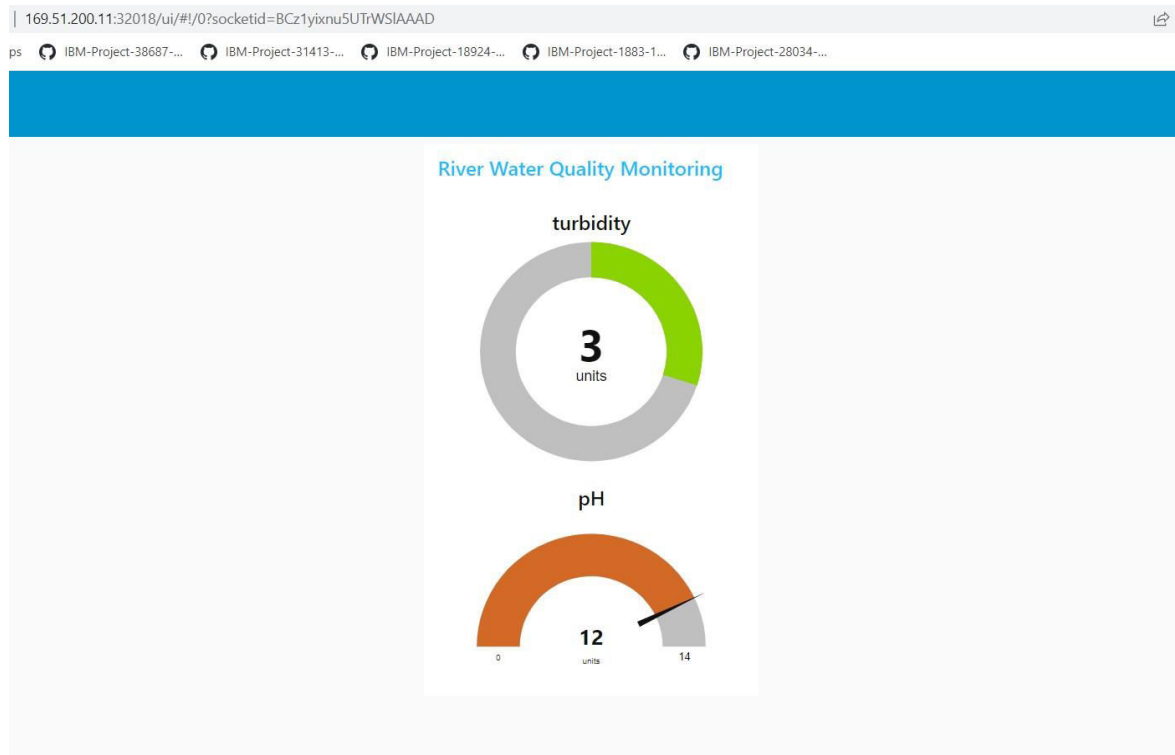
{ ph: 14, turbidity: 0 }
11/13/2022, 11:59:35 AM node: 1c347adabc7ec9d8
iot-2/type/Nodered/id/12345/evt/status/fmt/json :
msg.payload : number
14
11/13/2022, 11:59:35 AM node: 1c347adabc7ec9d8
iot-2/type/Nodered/id/12345/evt/status/fmt/json :
msg.payload : number
0
11/13/2022, 11:59:35 AM node: 1c347adabc7ec9d8
iot-2/type/Nodered/id/12345/evt/status/fmt/json :
msg.payload : Object
{ ph: 12, turbidity: 7 }
11/13/2022, 11:59:35 AM node: 1c347adabc7ec9d8
iot-2/type/Nodered/id/12345/evt/status/fmt/json :
msg.payload : number
12
11/13/2022, 11:59:35 AM node: 1c347adabc7ec9d8
iot-2/type/Nodered/id/12345/evt/status/fmt/json :
msg.payload : number
7
11/13/2022, 11:59:35 AM node: 1c347adabc7ec9d8
iot-2/type/Nodered/id/12345/evt/status/fmt/json :

```

Step 8: Configure all nodes and deploy the Node-red



Step 9: Go to Node-red UI we can now visualize the data which received from the IBM Watson



Step 10: We can change the Gauge type in the Gauge node and Visualize the data in different format

