

Use Dashboard Nodes For Creating UI(Web App)

TEAM ID	PNT2022TMID48692
PROJECT TITLE	REAL TIME RIVER WATER QUALITY MONITORING AND CONTROL SYSTEM

Step 1: Sending data to the IBM Watson

The screenshot shows the IBM Watson IoT Platform interface. A device with ID 12345 is listed as 'Connected'. The 'Recent Events' tab is active, showing a table of data points:

Event	Value	Format	Last Received
status	{"ph":13,"turbidity":4}	json	a few seconds ago
status	{"ph":7,"turbidity":1}	json	a few seconds ago
status	{"ph":9,"turbidity":2}	json	a few seconds ago
status	{"ph":1,"turbidity":3}	json	a few seconds ago
status	{"ph":13,"turbidity":0}	json	a few seconds ago

Step2: Configure the IBM IOT in the Node-red

The screenshot shows the Node-RED interface with the 'IBM IoT' node configured. The 'Properties' panel on the right is open, showing the following settings:

- Name: lot api
- API Key: a-ij64y3-w7zkatzbl9
- API Token:
- Server-Name: ij64y3.messaging.internetofthings.ibmcloud.com
- Scalable: ☐
- Application ID:
- Keep Alive: 60 Seconds
- Use Clean Session: ☒

Step 3: Configure the Device in IOT in

Flow 1

Edit ibmiot in node

Delete Cancel Done

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

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

Edit function node

Delete Cancel Done

Properties

Name: ph

Setup On Start **On Message** On Stop

```
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 panel 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

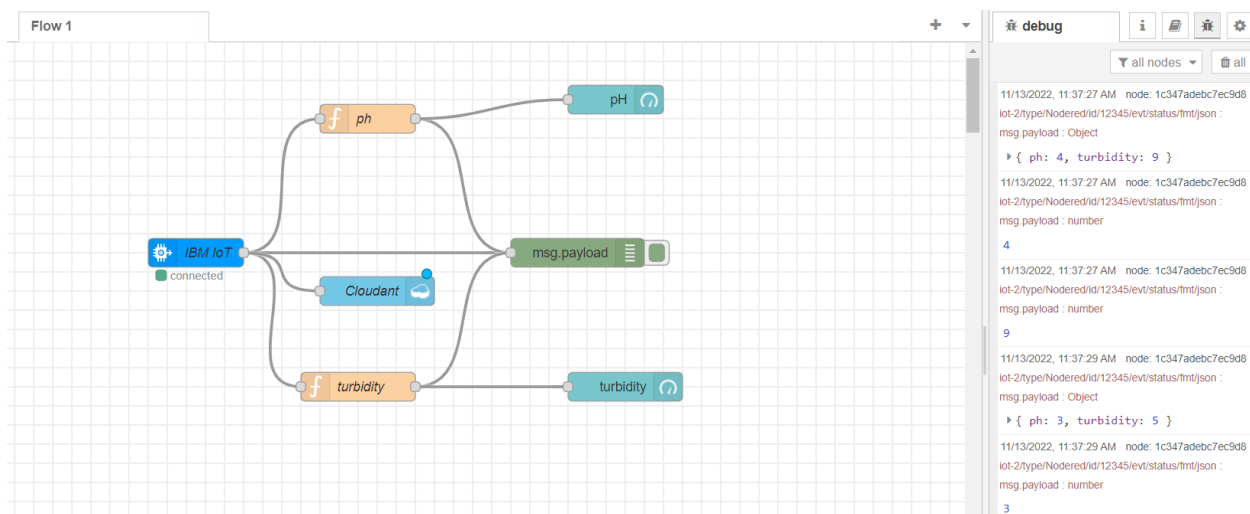
Debug console:

```

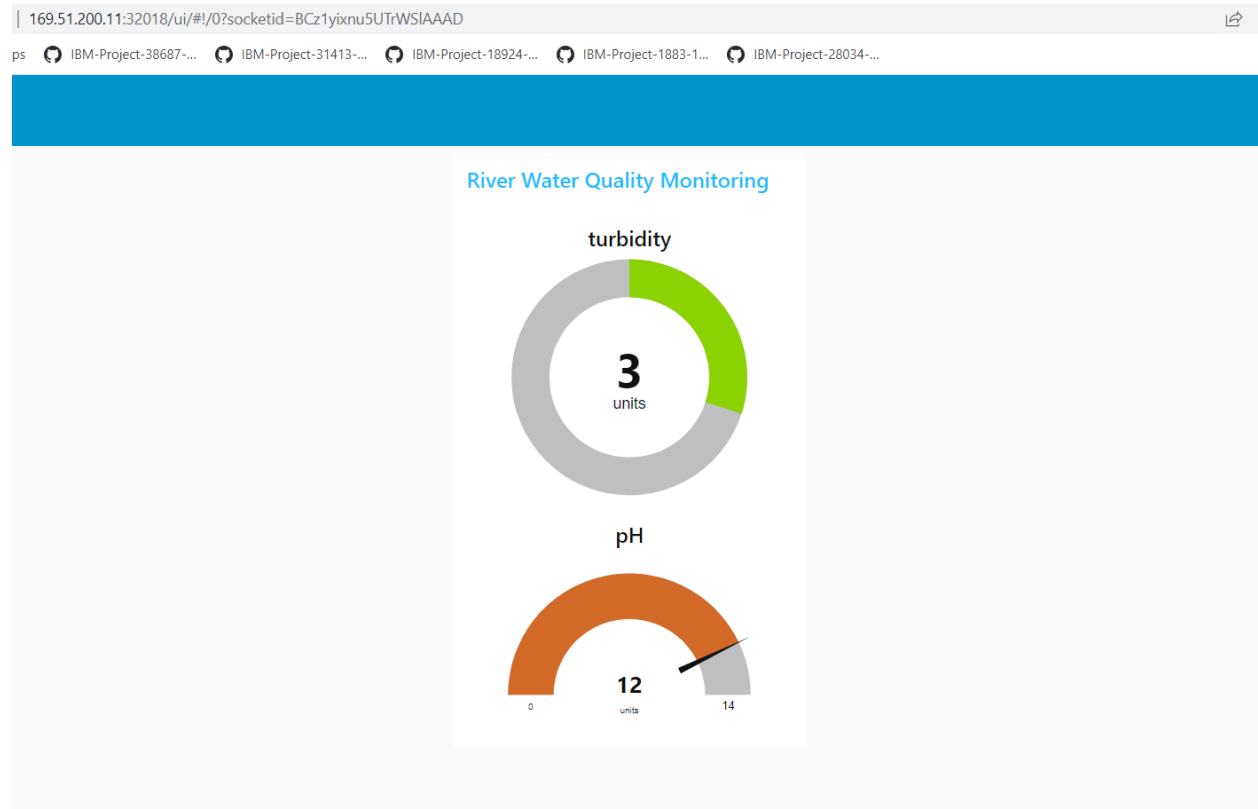
{ ph: 14, turbidity: 0 }
11/13/2022, 11:59:35 AM node: 1c347adabc7ec9d8
iot-2/type/Nodered/id/12345/evt/status/fmt/json :
msg.payload : number
14
11/13/2022, 11:59:35 AM node: 1c347adabc7ec9d8
iot-2/type/Nodered/id/12345/evt/status/fmt/json :
msg.payload : number
0
11/13/2022, 11:59:35 AM node: 1c347adabc7ec9d8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

