

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 list, the 'Recent Events' tab is active, showing a stream of data events. The events table has columns for Event, Value, Format, and Last Received. The data shows pH and Turbidity readings in JSON format. 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'. Below the flow, there is a '[get] /sensor_data' node connected to an 'http req' node. On the right, the 'Edit ibmiot in node > Edit ibmiot node' dialog is open, showing the configuration for the IBM IoT node. The configuration includes fields for Name, API Key, API Token, Server-Name, Scalable, Application ID, Keep Alive, and Use Clean Session.

Edit ibmiot in node > Edit ibmiot node

Buttons: Delete, Cancel, Update

Properties

- 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: ☒

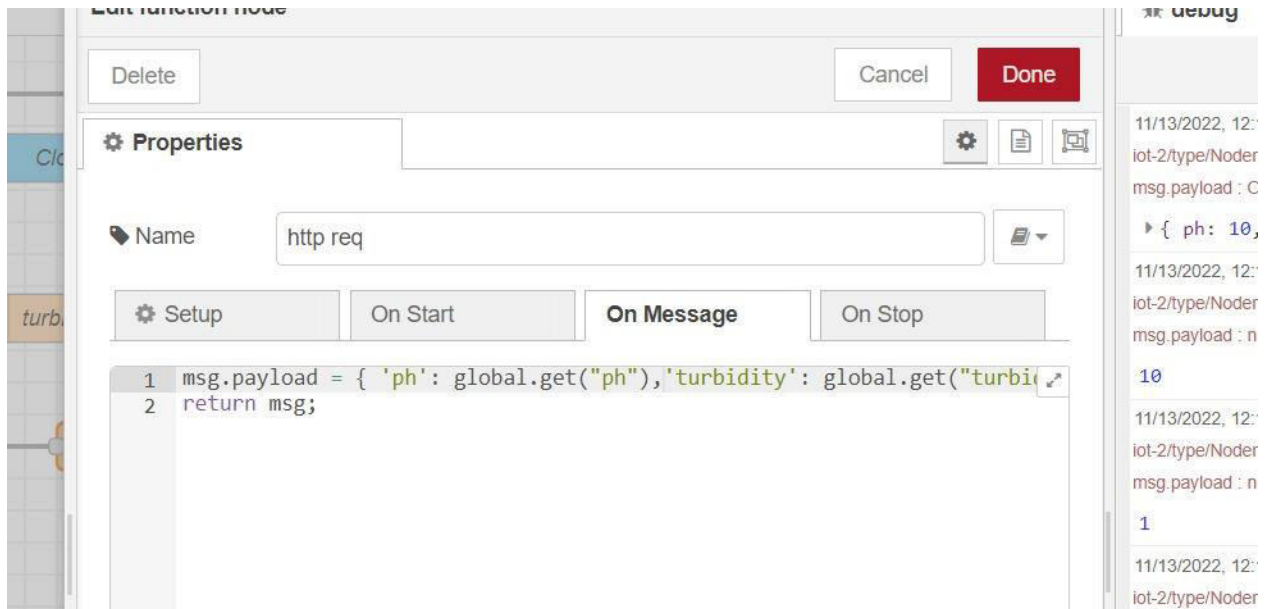
Step 3: Configure the IBM IoT in Node-red

The screenshot shows the Node-RED web interface. On the left, a flow named 'Flow 1' is visible on a grid. It contains an 'IBM IoT' node (blue with a gear icon) which is connected to three function nodes: 'ph', 'Cloudant', and 'turbidity'. Below these, there is a '[get] /sensor_data' node (green with a plus icon) connected to an 'http req' node (orange with a plus icon). On the right, the 'Edit ibmiot in node' dialog is open. It has a 'Delete' button, 'Cancel' and 'Done' buttons, and a 'Properties' section. The properties are: 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), and Service (registered). A yellow tooltip at the bottom right of the dialog reads: '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.'

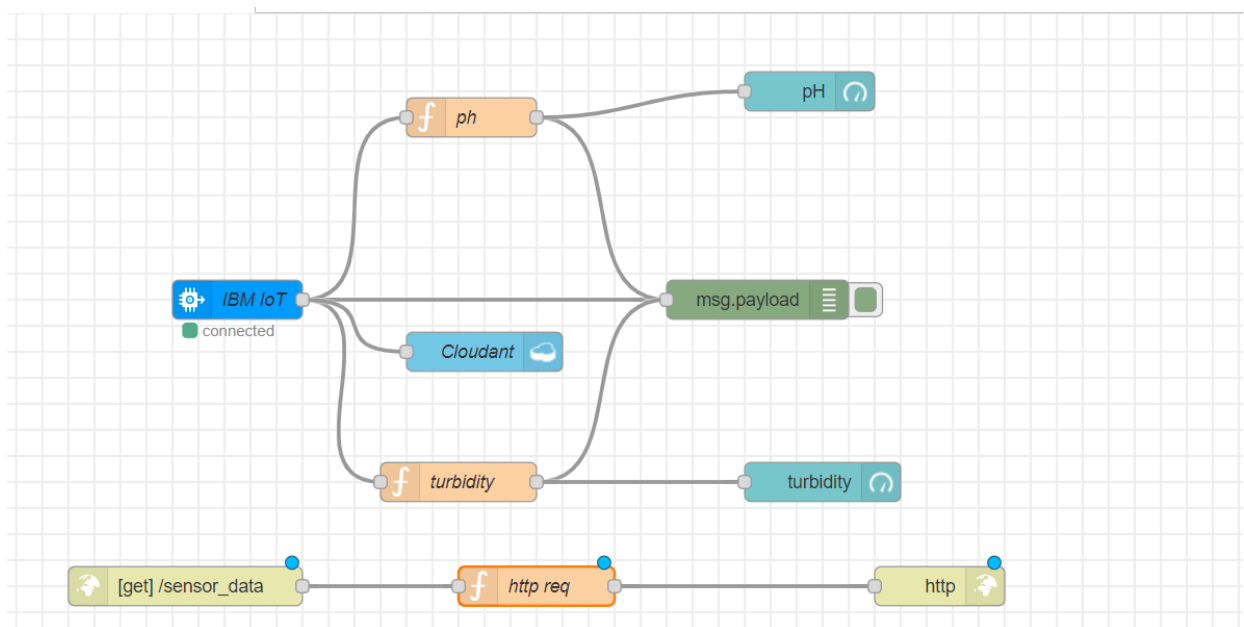
Step 4: Add a Http in and Http response node, Configure the Http in node and set the Method as GET and create an HTTP API for communicating with the mobile application

The screenshot shows the Node-RED web interface after adding a new node. The flow 'Flow 1' now includes an 'IBM IoT' node connected to 'ph', 'Cloudant', and 'turbidity' function nodes. Below these, a new '[get] /sensor_data' node (green with a plus icon) has been added, connected to an existing 'http req' node. On the right, the 'Edit http in node' dialog is open. It has 'Delete', 'Cancel', and 'Done' buttons, and a 'Properties' section. The properties are: Method (GET), URL (/sensor_data), and Name (Name).

Step 5: Add a function node to combine the pH and Turbidity value



Step 6: Configure all nodes and Deploy the Node-red



Step 7: Go to the API URL in node red and now you can able to view the data from the IBM Watson

