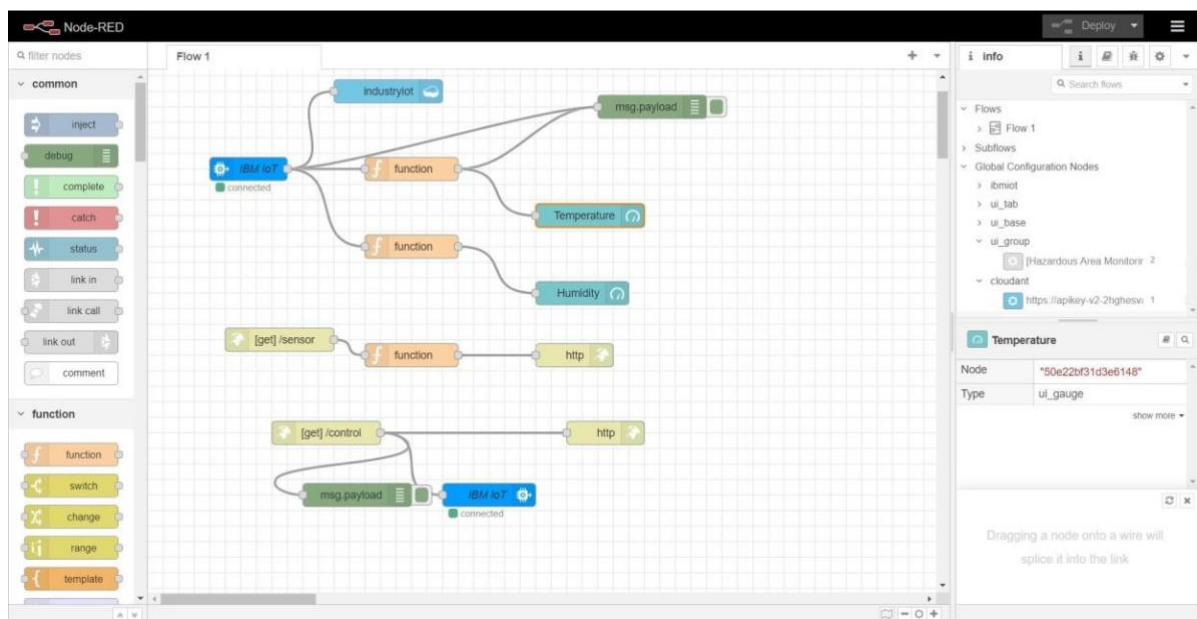


Configure The Application To Receive The Data From Cloud

Node red flow created to get values



Configuring function to fetch the desired value

The screenshot shows the Node-RED interface with a flow named 'Flow 1'. The flow includes an 'IBM IoT' node (connected), an 'industryiot' node, and two 'function' nodes. Below these, there are '[get] /sensor' and '[get] /control' nodes, both connected to a 'msg.payload' node. The 'Edit function node' dialog is open, showing the 'On Message' tab. The code in the dialog is:

```
1 msg.payload = msg.payload.temp;  
2 global.set('t',msg.payload)  
3 return msg;
```

The dialog also has a 'Name' field, a 'Setup' tab, and an 'Enabled' checkbox.

The screenshot shows the Node-RED interface with a flow named 'Flow 1'. The flow includes an 'IBM IoT' node (connected), an 'industryiot' node, and two 'function' nodes. Below these, there are '[get] /sensor' and '[get] /control' nodes, both connected to a 'msg.payload' node. The 'Edit function node' dialog is open, showing the 'On Message' tab. The code in the dialog is:

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

The dialog also has a 'Name' field, a 'Setup' tab, and an 'Enabled' checkbox.

App Blocks to render the values and display it in app

The screenshot displays the MIT App Inventor web interface for a project named "IoT_Interface". The interface is divided into several sections:

- Top Bar:** Includes navigation links like "Projects", "Connect", "Build", "Settings", and "Help". It also shows the user's profile "jagangoguria@gmail.com".
- Left Panel (Blocks):** A sidebar containing various built-in blocks categorized under "Built-in", "Screen1", and "Media".
- Right Panel (Viewer):** The main workspace where the app's logic is built using a block-based programming language. The logic includes:
 - A "when Clock1.Timer" block triggering a "do" block that sets "Web1.Uri" to "http://159.122.177.234:30241/sensor" and calls "Web1.Get".
 - A "when Web1.GetTest" block with a "do" block that processes the response. It uses "lock up in pairs" to extract "temperature" and "humidity" from the response content, then calls "Web1.JsonTextDecode" and "Web1.JsonText" to decode and display the data in "Text1" and "Text2" respectively.
 - Two "when Button1.Click" and "when Button2.Click" blocks that trigger "do" blocks to set "Web2.Uri" to "http://159.122.177.234:30241/control?command=on" and "http://159.122.177.234:30241/control?command=off" respectively, and call "Web2.Get".
- Bottom Bar:** Contains a "Show Warnings" button and a "Project Policy and Terms of Use" link.