

DEVELOP A WEB APPLICATION USING NODE-RED SERVICE

DEVELOP THE WEB APPLICATION USING NODE-RED SERVICE

Date	14 November 2022
Team ID	PNT2022TMID48307
Project Name	Industry-specific intelligent fire management system
Maximum Marks	8 Marks

Service Details - IBM Watson IoT Platform | Node-RED: node-red | Node-RED Dashboard | IBM | IoT-B8-2A4E (Afternoon) | IBM

https://node-red-1kluv-2022-11-13.au-syd.mybluemix.net/red/#flow/3e5b7d2a5cf2e52b

Node-RED

Deploy

filter nodes

Flow 1

button
dropdown
slider
numeric
text input
switch
date picker
colour picker
text
form
gauge
chart
audio out
notification
ui control
template

IBM IoT

Temperature

Humidity

msg.payload

temp

hum

dashboard

Layout Site Theme

Tabs & Links

- Temperature
 - SENSOR DATA
- SENSOR DATA
 - Temperature
 - Humidity

26°C Cloudy

19:32 13-11-2022

node-red-lkluv-2022-11-13.au-syd.mybluemix.net/red/#flow/3e5b7d2a5cf2e52b

Node RED

Flow 1

common

- inject
- debug
- complete
- catch
- status
- link in
- link call
- link out
- comment

function

- function
- switch
- change
- range
- template
- delay
- trigger
- filter

Flow 1 Diagram:

```
graph LR
    IBMIoT[IBM IoT] --> Temperature
    IBMIoT --> Humidity
    Temperature --> msgPayload1[msg.payload]
    Temperature --> TemperatureOut[Temperature]
    Humidity --> msgPayload2[msg.payload]
    Humidity --> HumidityOut[Humidity]
    FANON[FAN ON] --> IBMIoT
    FANON --> msgPayload3[msg.payload]
    FANOFF[FAN OFF] --> IBMIoT
    FANOFF --> msgPayload3
    GETCMD[GET /command] --> http1[http]
    GETDATA[GET /data] --> Webpage[Webpage]
    Webpage --> http2[http]
```

debug

all nodes

11/16/2022, 8:12:31 PM node: 6b7afb0d048e27d3
iot-2?type/NodeId/1234ev/status/rmt/json :
msg.payload : number
71

11/16/2022, 8:12:50 PM node: 6b7afb0d048e27d3
iot-2?type/NodeId/1234ev/status/rmt/json :
msg.payload : number
4

11/16/2022, 8:12:51 PM node: 6b7afb0d048e27d3
iot-2?type/NodeId/1234ev/status/rmt/json :
msg.payload : number
32

11/16/2022, 8:13:10 PM node: 6b7afb0d048e27d3
iot-2?type/NodeId/1234ev/status/rmt/json :
msg.payload : number
25

11/16/2022, 8:13:11 PM node: 6b7afb0d048e27d3
iot-2?type/NodeId/1234ev/status/rmt/json :
msg.payload : number
82

11/16/2022, 8:13:31 PM node: 6b7afb0d048e27d3
iot-2?type/NodeId/1234ev/status/rmt/json :
msg.payload : number
62

11/16/2022, 8:13:31 PM node: 6b7afb0d048e27d3
iot-2?type/NodeId/1234ev/status/rmt/json :
msg.payload : number
43

11/16/2022, 8:13:50 PM node: 6b7afb0d048e27d3
iot-2?type/NodeId/1234ev/status/rmt/json :
msg.payload : number
96

11/16/2022, 8:13:51 PM node: 6b7afb0d048e27d3

SENSORapp.apk

30°C Cloudy

20:27 16-11-2022

Node-RED interface showing a flow diagram and a code editor window.

Flow Diagram:

- Input: `mqtt` (blue node) receives data from `mqtt` (green node).
- Processing: `mqtt` data is split into `temperature` and `humidity` (orange nodes).
- Output: `temperature` and `humidity` data is sent to `mqtt` (green node).
- Control: `mqtt` data is also sent to `mqtt` (green node).
- Monitoring: `mqtt` data is sent to `mqtt` (green node).

Code Editor Window:

```
var mqtt = require('mqtt');
var client = mqtt.connect('mqtt://192.168.1.100:1883');

client.on('connect', function() {
  console.log('Connected to MQTT broker');
  client.subscribe('temperature');
  client.subscribe('humidity');
});

client.on('message', function(topic, message) {
  console.log('Received message: ' + topic + ' - ' + message);
  // Process the message here
});
```