

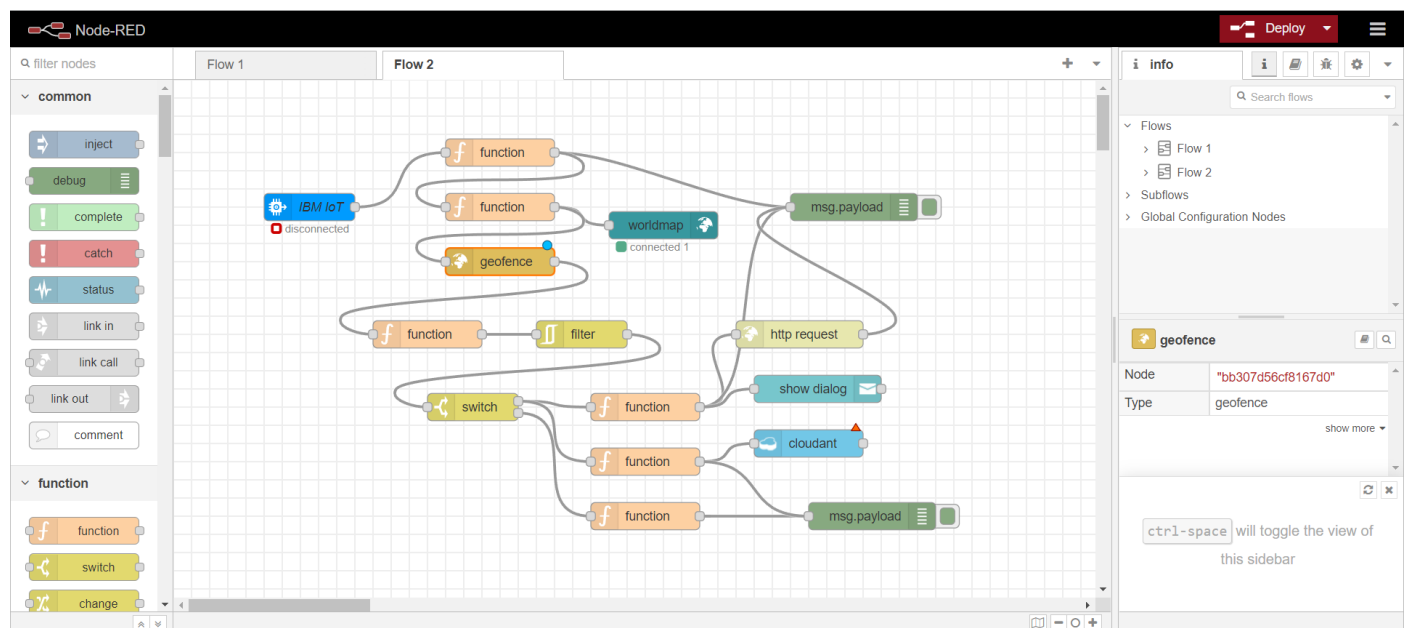
Project Development - Sprint 1

Iot Based Safety For Child Safety Monitoring & Notification

Team ID: PNT2022TMID20412

Creating Node - Red Service and Connecting with IBM cloud

Creating Node-Red Service :



Codes in Each Node:

Node-RED interface showing a flow diagram and the "Edit ibmiot in node" configuration panel.

Flow Diagram: The flow starts with an "IBM IoT" node (connected). It branches into two parallel paths. The top path consists of a "function" node, followed by a "worldmap" node (connected 1), and then a "msg.payload" output. The bottom path consists of a "function" node, followed by a "geofence" node, and then a "filter" node. Both paths converge into a "switch" node. From the "switch", the flow branches into three parallel paths: a "function" node leading to an "http request" node, a "function" node leading to a "show dialog" node, and a "function" node leading to a "Child" node. All three paths then converge into a final "msg.payload" output.

Edit ibmiot in node Properties:

- Authentication: API Key
- API Key: child_iot
- Input Type: Device Event
- Device Type: ☐ All or ☐ IOT_CHILD
- Device Id: ☐ All or 16022002
- Event: ☒ All or +
- Format: ☐ All or ☐ json
- QoS: 0
- Name: IBM IoT
- Service: registered
- Enabled: ☐ Enabled

Node-RED interface showing the same flow diagram and the "Edit function node" configuration panel.

Flow Diagram: The flow diagram is identical to the one in the first screenshot.

Edit function node Properties:

- Name: Name
- Setup: ☐ Setup ☐ On Start ☒ On Message ☐ On Stop
- Code:

```
1 var name = msg.payload.name
2 var lat = msg.payload.lat
3 var lon = msg.payload.lon
4 global.set('latitude',lat)
5 global.set('longitude',lon)
6 global.set('name',name)
7 return msg;
```

Enabled: ☐ Enabled

Node-RED interface showing the same flow diagram and the "Edit function node" configuration panel.

Flow Diagram: The flow diagram is identical to the one in the first screenshot.

Edit function node Properties:

- Name: Name
- Setup: ☐ Setup ☐ On Start ☒ On Message ☐ On Stop
- Code:

```
1 msg.payload={
2   'name':global.get('name'),
3   'lat':global.get('latitude'),
4   'lon':global.get('longitude')
5 }
6 return msg;
```

Enabled: ☐ Enabled

Node-RED interface showing a flow diagram and the "Edit geofence node" panel.

The flow diagram includes nodes for **storage** (Db2 in, cloudant in, dashDB in, Db2 out, cloudant out, dashDB out), **IBM Watson** (assistant, assistant workspace manager, assistant V2, discovery), **function**, **geofence**, **worldmap**, **filter**, **switch**, **http request**, and **show dialog**.

The "Edit geofence node" panel shows a map of Hyderabad, India, with a geofence area defined. The map includes labels for Sangareddy, Medchal-Malkajgiri, Secunderabad, and Hyderabad. The geofence is defined by a polygon on the map. The panel also includes a "Properties" section with a "Name" field and a "Floor" dropdown set to "ground".

Node-RED interface showing a flow diagram and the "Edit function node" panel.

The flow diagram is identical to the first screenshot, showing the same sequence of nodes and connections.

The "Edit function node" panel shows the "On Message" tab with the following code:

```
1 msg.payload = msg.location.inarea
2 return msg;
```

Node-RED interface showing a flow diagram and the "Edit switch node" panel.

The flow diagram is identical to the first screenshot, showing the same sequence of nodes and connections.

The "Edit switch node" panel shows the "Properties" section with the "Property" dropdown set to "msg: payload". The switch is configured with two rules:

- Rule 1: `is false` → 1
- Rule 2: `is true` → 2

The "checking all rules" checkbox is checked, and the "recreate message sequences" checkbox is unchecked. The "Enabled" checkbox is also checked.

Node-RED interface showing a flow diagram and the Edit function node configuration.

Flow Diagram: The flow starts with an **IBM IoT** node (connected). It branches into two parallel paths. The top path consists of a **function** node, followed by another **function** node, then a **worldmap** node (connected 1), and finally an **http request** node. The bottom path consists of a **function** node, followed by a **filter** node, then a **switch** node, and finally three **function** nodes. The **http request** node and the final **function** node in the bottom path are connected to a **show** node.

Edit function node configuration:

- Name:** Name
- On Message:**

```
1 var d = new Date();
2
3 var utc = d.getTime() + (d.getTimezoneOffset()*60000);
4
5 var offset = 5.5;
6
7 newDate = new Date(utc +(360000*offset));
8
9 msg.payload = {
10   "message": "Exit",
11   "Time": newDate.toLocaleString(),
12   "name": global.get('name'),
13   "lat": global.get('latitude'),
14   "lon": global.get('longitude')
15 };
16
17 return msg;
```
- Enabled:** ☐

Node-RED interface showing a flow diagram and the Edit function node configuration.

Flow Diagram: The flow is identical to the first screenshot, starting with an **IBM IoT** node and branching into two paths. The top path includes **function**, **function**, **worldmap** (connected 1), and **http request** nodes. The bottom path includes **function**, **filter**, **switch**, and three **function** nodes. The **http request** node and the final **function** node in the bottom path are connected to a **show** node.

Edit function node configuration:

- Name:** Name
- On Message:**

```
1
2 var d = new Date();
3
4 var utc = d.getTime() + (d.getTimezoneOffset()*60000);
5
6 var offset = 5.5;
7
8 newDate = new Date(utc +(360000*offset));
9
10 msg.payload = {
11   "message": "Entry",
12   "Time": newDate.toLocaleString(),
13   "name": global.get('name'),
14   "lat": global.get('latitude'),
15   "lon": global.get('longitude')
16 };
17
18 return msg;
```
- Enabled:** ☐

Node-RED interface showing a flow diagram and the Edit http request node configuration.

Flow Diagram: The flow is identical to the previous screenshots, starting with an **IBM IoT** node and branching into two paths. The top path includes **function**, **function**, **worldmap** (connected 1), and **http request** nodes. The bottom path includes **function**, **filter**, **switch**, and three **function** nodes. The **http request** node and the final **function** node in the bottom path are connected to a **show** node.

Edit http request node configuration:

- Method:** GET
- URL:** <https://www.fast2sms.com/dev/bulkV2?authorizati>
- Payload:** Ignore
- Enable secure (SSL/TLS) connection:** ☐
- Use authentication:** ☐
- Enable connection keep-alive:** ☐
- Use proxy:** ☐
- Only send non-2xx responses to Catch node:** ☐
- Return:** a UTF-8 string
- Name:** Name
- Enabled:** ☐