

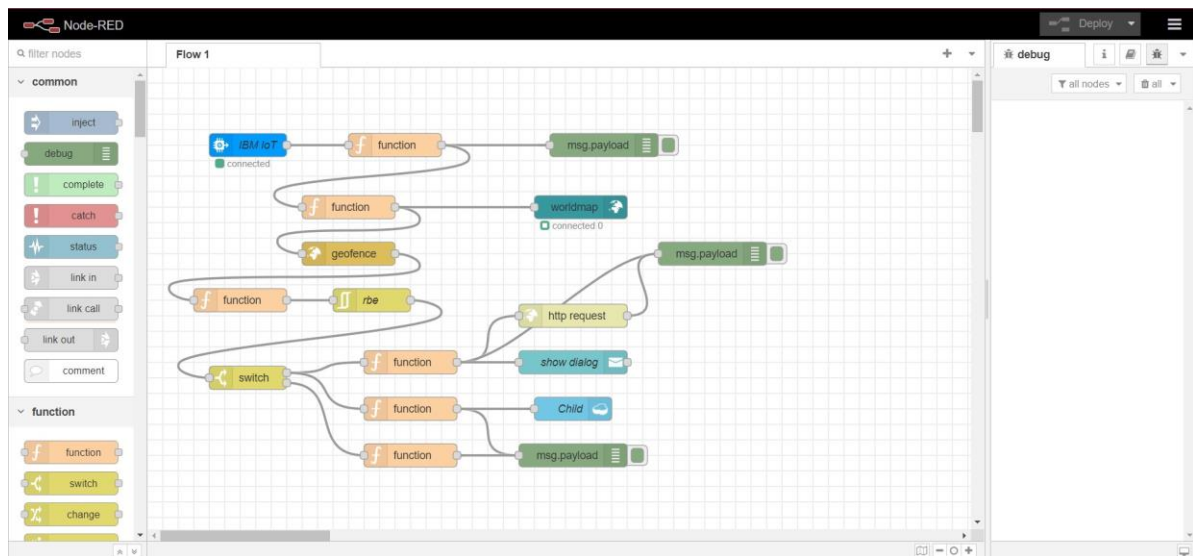
# Project Development – Delivery plan sprint-2

## IoT Based Safety Gadget for Child Safety Monitoring & Notification

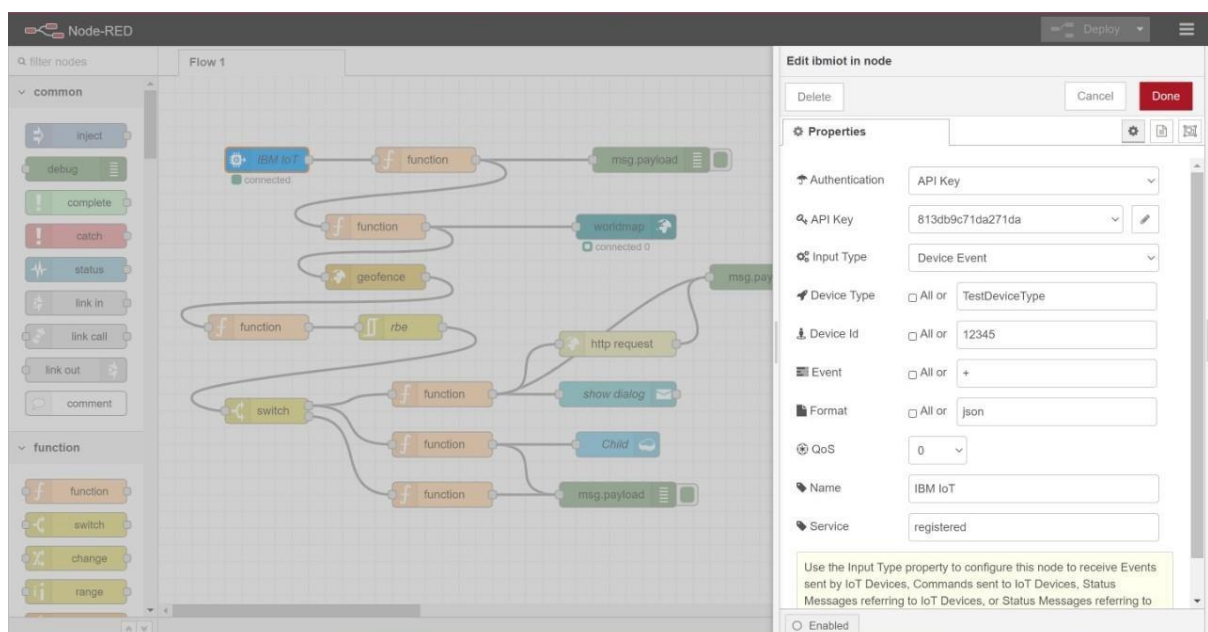
TEAM ID:PNT2022TMID27063

### Creating Node-Red service and connecting with IBM cloud

#### Creating Node-Red service:



#### Codes in each Node:



Node-RED interface showing the "Edit function node" dialog for a function node in the "Child Tracker" flow. The flow includes an "IBM IoT" node, a "function" node, a "geofence" node, and another "function" node. The "Edit function node" dialog is open, showing the "Properties" tab with the "Name" field set to "Name". The "On Message" tab is selected, displaying the following JavaScript 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;
```

The "Enabled" checkbox is checked. The dashboard on the right shows the "Child Tracker" tab with a "Map" node.

Node-RED interface showing the "Edit debug node" dialog for a debug node in the "Child Tracker" flow. The flow includes an "IBM IoT" node, a "function" node, a "worldmap" node, a "geofence" node, a "function" node, an "rbe" node, and a "switch" node. The "Edit debug node" dialog is open, showing the "Properties" tab with the "Output" field set to "msg.payload", the "To" field set to "debug window", and the "Name" field set to "Name". The "Enabled" checkbox is checked. The dashboard on the right shows the "Child Tracker" tab with a "Map" node.

Node-RED interface showing the "Edit function node" dialog for a function node in the "Child Tracker" flow. The flow includes an "IBM IoT" node, a "function" node, a "geofence" node, and another "function" node. The "Edit function node" dialog is open, showing the "Properties" tab with the "Name" field set to "Name". The "On Message" tab is selected, displaying the following JavaScript code:

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

The "Enabled" checkbox is checked. The dashboard on the right shows the "Child Tracker" tab with a "Map" node.

Node-RED interface showing a flow named "Flow 1" and the "Edit worldmap node" configuration panel.

**Flow 1:** The flow starts with an **IBM IoT** node (connected), followed by a **function** node. The output of the function node goes to a **worldmap** node (connected). The **worldmap** node's output goes to a **msg.payload** node. The **worldmap** node also has a **geofence** node connected to it. The **geofence** node's output goes to a **function** node, which then goes to a **switch** node. The **switch** node has four outputs: one to a **function** node, one to a **function** node, one to a **function** node, and one to a **function** node. The **function** nodes are connected to **msg.payload** nodes, **show dialog** nodes, and **Child** nodes.

**Edit worldmap node Properties:**

- Group: [Child Tracker] Map
- Size: auto
- Start: Latitude 17.4226372, Longitude 78.5456505, Zoom 16
- Map list: 7 selected
- Base map: ESRI Satellite
- Overlays: 5 selected
- Cluster when zoom level is less than 0 (0, off - 19)
- Max age: Remove markers after 600 seconds
- User menu: Show, Layer menu: Hide
- Lock map: False, Lock zoom: False
- Auto-pan: Disable, Right click: Disable
- Enabled: ☐

Node-RED interface showing a flow named "Flow 1" and the "Edit geofence node" configuration panel.

**Flow 1:** The flow starts with an **IBM IoT** node (connected), followed by a **function** node. The output of the function node goes to a **worldmap** node (connected). The **worldmap** node's output goes to a **msg.payload** node. The **worldmap** node also has a **geofence** node connected to it. The **geofence** node's output goes to a **function** node, which then goes to a **switch** node. The **switch** node has four outputs: one to a **function** node, one to a **function** node, one to a **function** node, and one to a **function** node. The **function** nodes are connected to **msg.payload** nodes, **show dialog** nodes, and **Child** nodes.

**Edit geofence node Properties:**

- Map: A map showing a geofence area in Chennai, India, centered around the area of Taramani and Perungudi.
- Floor: ground, Ceiling: Infinity
- Action: add "inarea" property
- Enable output of zones to WorldMap node: ☐
- Enabled: ☐

Node-RED interface showing a flow named "Child Tracker" and the "Edit function node" configuration panel.

**Flow 1:** The flow starts with an **IBM IoT** node (connected), followed by a **function** node. The output of the function node goes to a **worldmap** node (connected). The **worldmap** node's output goes to a **msg.payload** node. The **worldmap** node also has a **geofence** node connected to it. The **geofence** node's output goes to a **function** node, which then goes to a **switch** node. The **switch** node has four outputs: one to a **function** node, one to a **function** node, one to a **function** node, and one to a **function** node. The **function** nodes are connected to **msg.payload** nodes, **show dialog** nodes, and **Child** nodes.

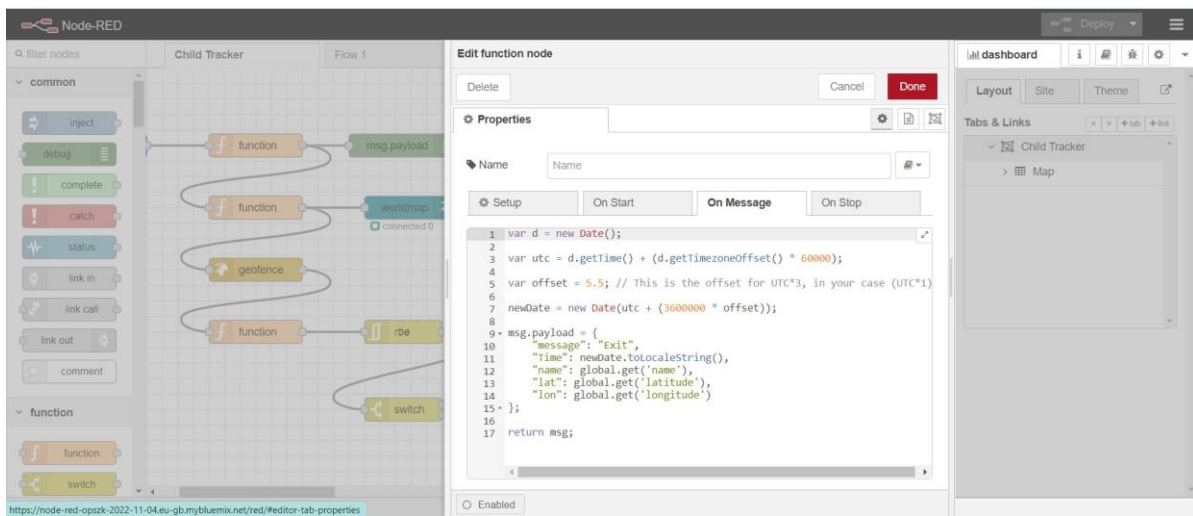
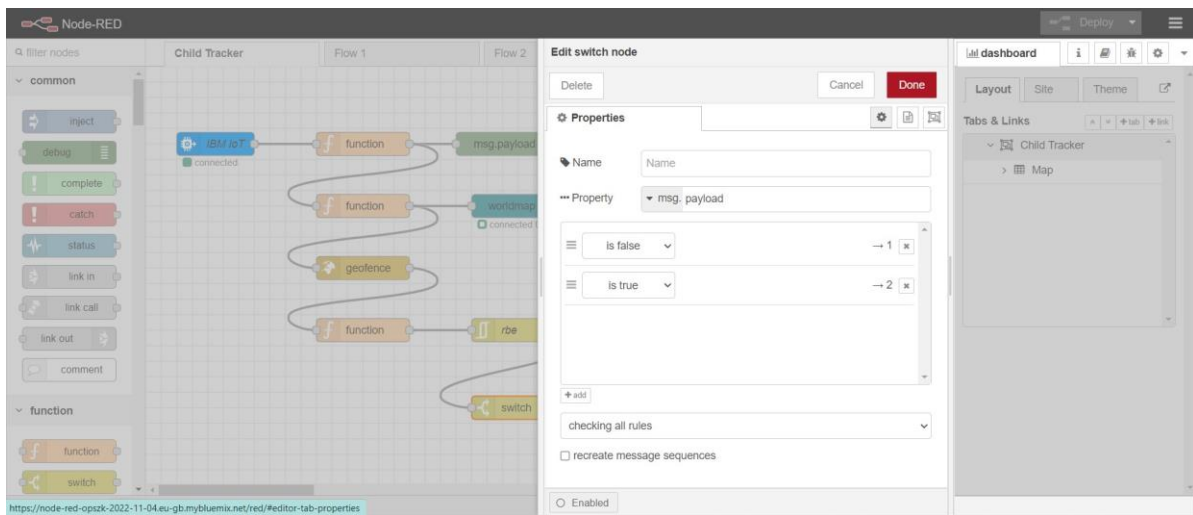
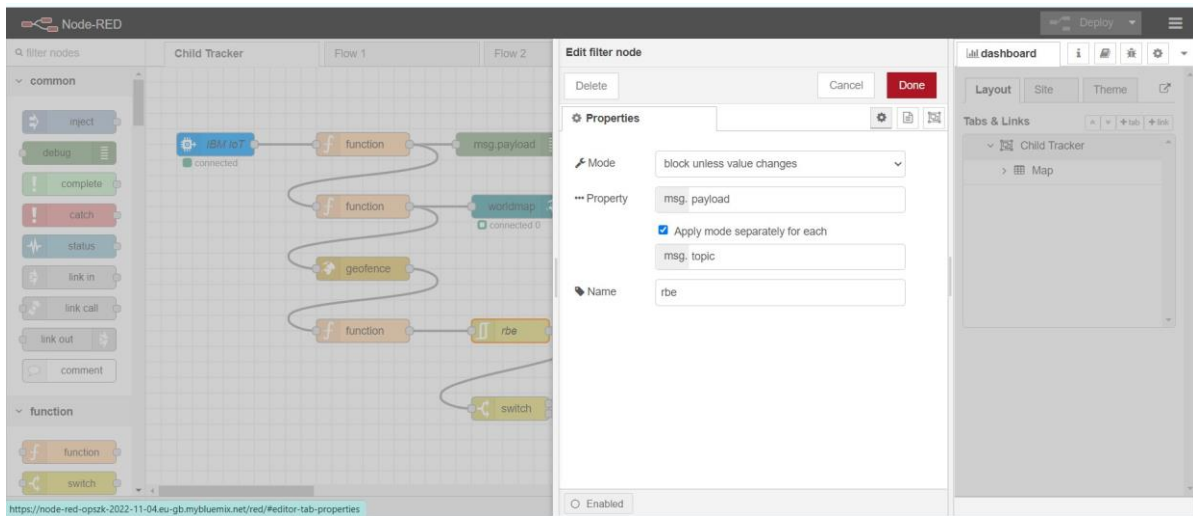
**Edit function node Properties:**

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

```
1 msg.payload=msg.location.inarea
2 return msg;
```
- Enabled: ☐

**Dashboard:** The dashboard shows a "Child Tracker" tab with a "Map" view.

<https://node-red-opzk-2022-11-04-eu-gb.mybluemix.net/red/editor-tab-properties>



Node-RED interface showing the 'Edit function node' dialog. The flow is titled 'Child Tracker' and 'Flow 1'. The function node code is as follows:

```
1 var d = new Date();
2 var utc = d.getTime() + (d.getTimezoneOffset() * 60000);
3
4 var offset = 5.5; // This is the offset for UTC+3, in your case (UTC+1)
5
6 newDate = new Date(utc + (3600000* offset));
7
8
9 msg.payload={
10   "message": "Entry",
11   "time": newDate.toLocaleString(),
12   "name": global.get('name'),
13   "lat": global.get('latitude'),
14   "lon": global.get('longitude')
15 };
16
17 return msg;
```

The right sidebar shows the 'dashboard' tab with 'Child Tracker' and 'Map' links.

Node-RED interface showing the 'Edit http request node' dialog. The flow is titled 'Flow 1'. The http request node configuration is as follows:

- Method: GET
- URL: <https://www.fast2sms.com/dev/bulkV2?authorization=>
- 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

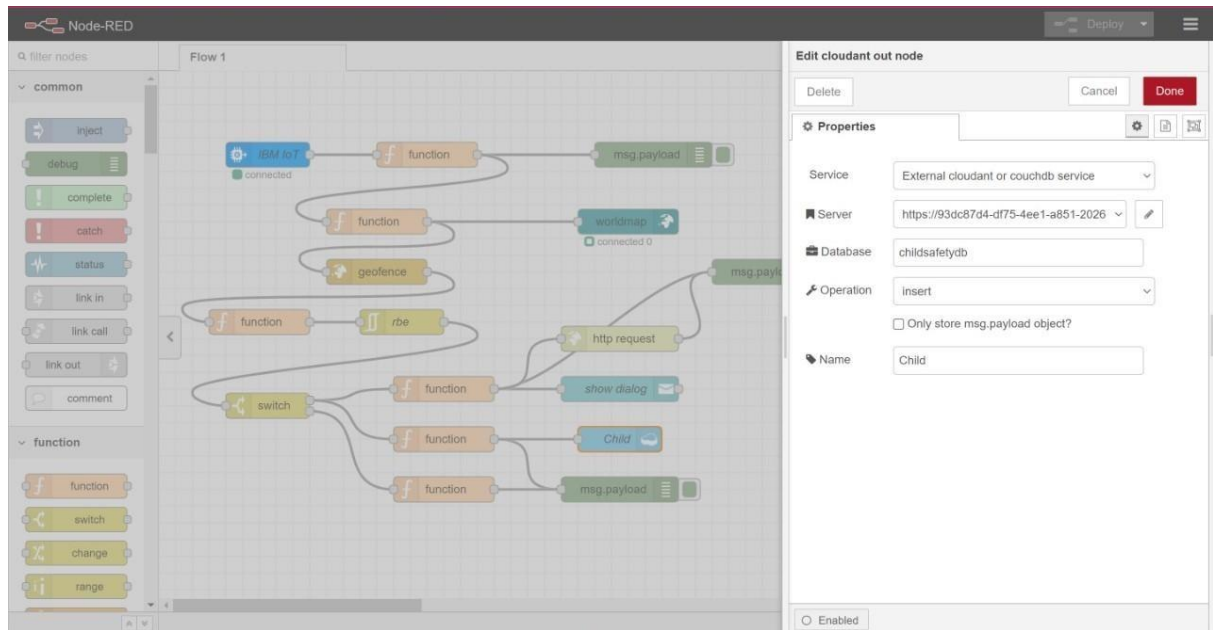
The right sidebar shows the 'dashboard' tab with 'Child Tracker' and 'Map' links.

Node-RED interface showing the 'Edit notification node' dialog. The flow is titled 'Child Tracker' and 'Flow 1'. The notification node configuration is as follows:

- Layout: OK / Cancel Dialog
- Send to all browser sessions: ☒
- Default action label: OK
- Secondary action label: (optional label for Cancel button)
- Accept raw HTML/JavaScript input in msg.payload to format popup: ☐
- Class: [msg.className]
- Topic: [msg.topic]
- Name: Show Dialoge

The right sidebar shows the 'dashboard' tab with 'Child Tracker' and 'Map' links.





## Connecting with IBM Cloud: Using IBM IOT node through the API key

The image shows the IBM Watson IoT Platform interface. The 'API Key Information' tab is selected, displaying details for a 'Standard Application' with the key 'a-jgry6x-vocds6jzm'. The 'API Key Information' table is as follows:

Key	Description	Role	Expires
a-jgry6x-vocds6jzm	-	-	-

The 'Access Control/Permissions' tab is also visible, showing the 'Last Edited By' as 'julianthomaspeniel16@gmail.com' and the 'Expires' as 'Never'.

## Transferring values from Python Code:

```
child.py - C:\Users\Anu\AppData\Local\Programs\Python\Python37\chld.py (37.0)
File Edit Format Run Options Window Help
import json
import wiotsdk.device

import time
myConfig = {

    "identity":{
        "orgId": "401qxb",
        "typeId": "TestDeviceType",
        "deviceId": "12345"
    },
    "auth": {
        "token": "pnhKvZn-zMRvshayi"
    }
}

client= wiotsdk.device.DeviceClient(config=myConfig, logHandlers=None)
client.connect()

while True:
    name = "Smartbridge"
    #in area location

    #latitude = 17.4225176
    #longitude = 78.5456842

    #out area location

    latitude= 17.4219272
    longitude= 78.5488793
    myData={"name": name, 'lat':latitude, 'lon': longitude}
    client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0, onPublish=None)
    print("Data published to IBM IoT platform: ",myData)
    time.sleep(5)

client.disconnect()
```

## Node-Red:

