

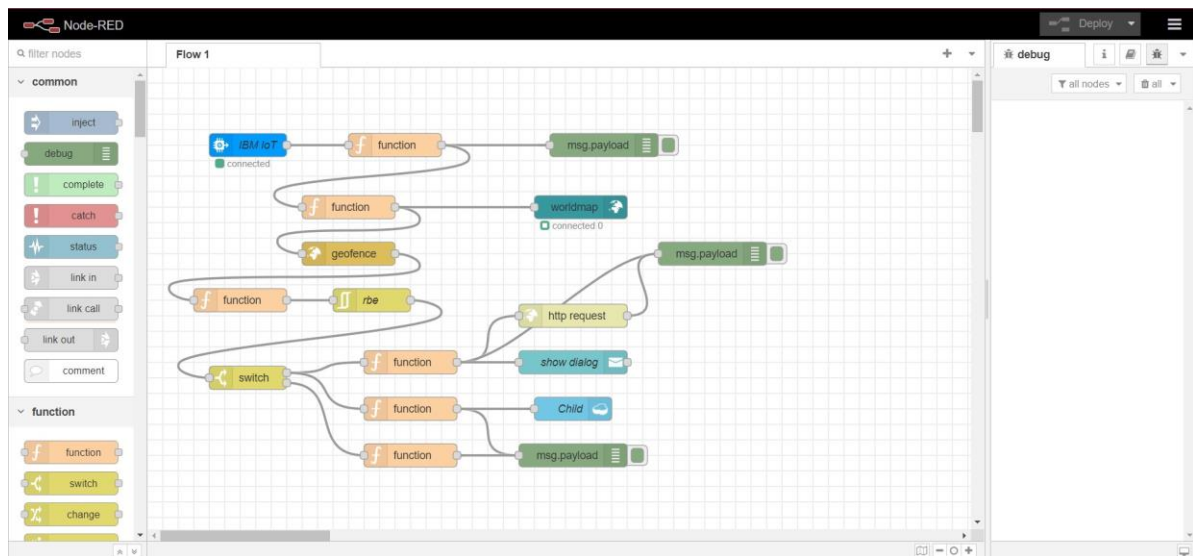
# Project Development –Sprint-3

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

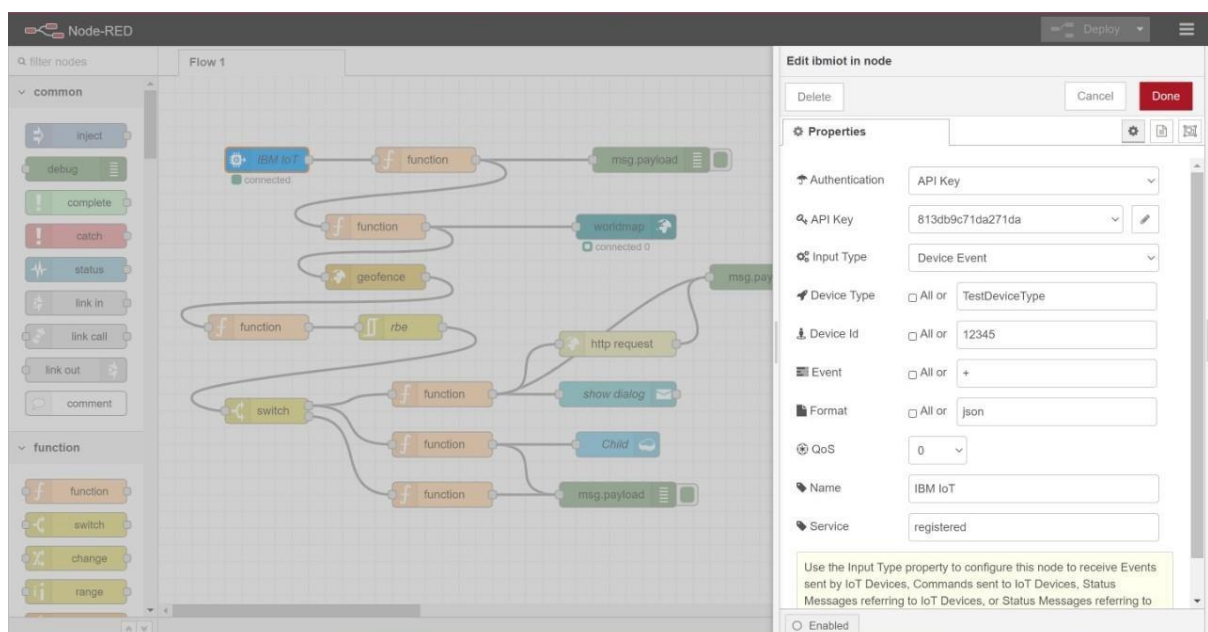
TEAM ID:PNT2022TMID36015

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

#### Creating Node-Red service:



#### Codes in each Node:



Node-RED interface showing a flow named "Child Tracker" in Flow 1. The flow starts with an "IBM IoT" node (connected), followed by a "function" node, then another "function" node, then a "geofence" node, and finally a third "function" node. The "Edit function node" panel is open, showing 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 "Properties" panel shows the node name as "Name". The "On Message" tab is selected. The "dashboard" panel on the right shows a "Child Tracker" tab with a "Map" sub-tab.

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

WhatsApp Image... .jpeg

Show all

Node-RED interface showing the same "Child Tracker" flow in Flow 1. The flow is extended to include a "msg.payload" node, a "worldmap" node (connected), and an "rbe" node. The "Edit debug node" panel is open, showing the following configuration:

- Output: msg.payload
- To: ☒ debug window
- ☐ system console
- ☐ node status (32 characters)
- Name: Name

The "Properties" panel shows the node name as "Name". The "On Message" tab is selected. The "dashboard" panel on the right shows a "Child Tracker" tab with a "Map" sub-tab.

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

Node-RED interface showing the same "Child Tracker" flow in Flow 1. The flow is extended to include a "function" node, a "geofence" node, and another "function" node. The "Edit function node" panel is open, showing 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 "Properties" panel shows the node name as "Name". The "On Message" tab is selected. The "dashboard" panel on the right shows a "Child Tracker" tab with a "Map" sub-tab.

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

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, then a **worldmap** node (connected). Below this, there is a **geofence** node, followed by a **function** node and an **rbe** node. A **switch** node branches the flow into three parallel **function** nodes. The top **function** node connects to an **http request** node, which then connects to a **show dialog** node. The middle **function** node connects to a **Child** node. The bottom **function** node connects to a **msg.payload** node.

**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 the same flow and the "Edit geofence node" configuration panel.

**Edit geofence node Properties:**

- Floor:** ground
- Ceiling:** Infinity
- Action:** add "inarea" property
- Enable output of zones to WorldMap node:** ☐
- Enabled:** ☐

Node-RED interface showing the flow, the "Edit function node" configuration panel, and the dashboard.

**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 layout with a site and theme. Under "Tabs & Links", there is a tab named "Child Tracker" which contains a link to "Map".

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

Node-RED interface showing the "Edit filter node" dialog for a node named "rbe". The dialog includes a "Properties" tab with the following settings:

- Mode: block unless value changes
- Property: msg.payload
- Apply mode separately for each: checked
- msg.topic
- Name: rbe

The background shows a flow diagram with nodes: inject, debug, complete, catch, status, link in, link call, link out, comment, function, switch, and msg.payload. The flow is titled "Child Tracker".

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

Node-RED interface showing the "Edit switch node" dialog for a node named "rbe". The dialog includes a "Properties" tab with the following settings:

- Name: Name
- Property: msg.payload
- Is false: 1
- Is true: 2
- checking all rules: checked
- recreate message sequences: unchecked

The background shows the same flow diagram as the previous image, titled "Child Tracker".

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

Node-RED interface showing the "Edit function node" dialog for a node named "rbe". The dialog includes a "Properties" tab with the following settings:

- Name: Name
- Setup: selected
- On Start: selected
- On Message: selected
- On Stop: selected

The function code is as follows:

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

The background shows the same flow diagram as the previous images, titled "Child Tracker".

<https://node-red-opszk-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

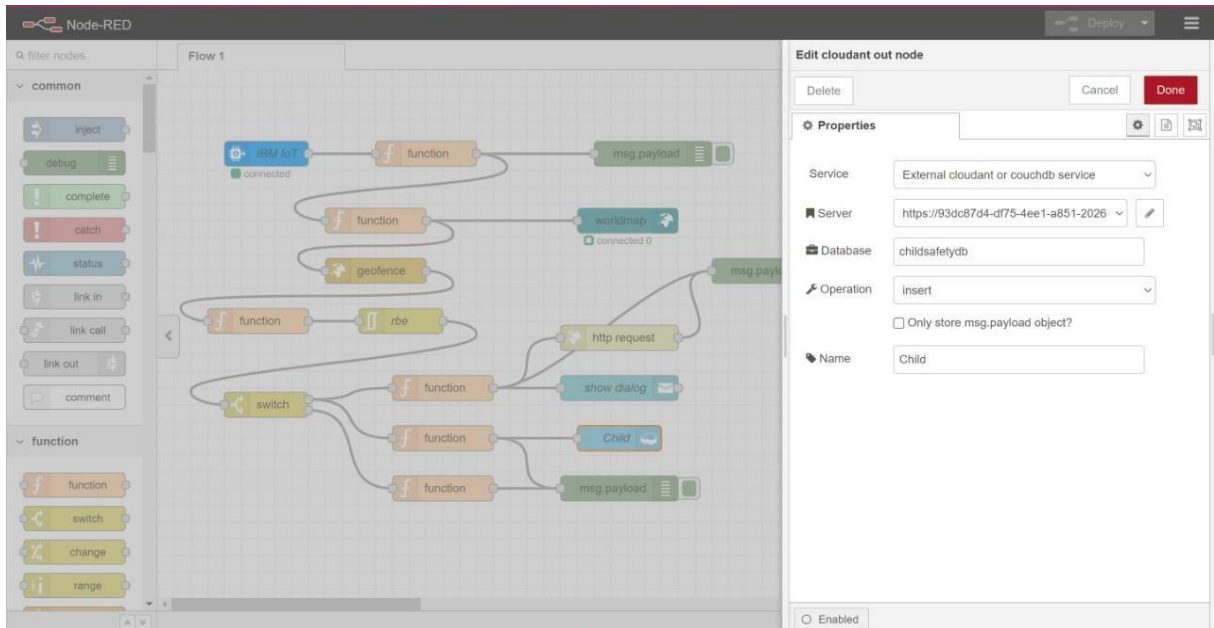
The right sidebar shows the 'dashboard' tab with 'Child Tracker' and 'Map' links.

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 'Browse API Keys' page in the IBM Watson IoT Platform. The page header includes the user's email (310819106007@smartinternz.com) and ID (4a1qxb). A 'Generate API Key' button is visible in the top right. The main content area is titled 'Browse API Keys' and includes a search bar. Below the search bar, a table lists the API keys:

| Key                | Description                      | Role                 | Expires |
|--------------------|----------------------------------|----------------------|---------|
| a-4a1qxb-d5wgvebrf | -                                | Standard Application | -       |
| a-4a1qxb-ecmygwzdc | API Key for the device simulator | Standard Application | -       |

At the bottom of the page, there is a status bar showing '1 Simulation running' and 'Apps using your microphone: Google Chrome'.

The image shows the 'Browse API Keys' page in the IBM Watson IoT Platform, displaying details for a specific API key. The page header includes the user's email (310819106007@smartinternz.com) and ID (4a1qxb). A 'Generate API Key' button is visible in the top right. The main content area is titled 'Browse API Keys' and includes a search bar. Below the search bar, a table lists the API keys:

| Key                | Description | Role                 | Expires |
|--------------------|-------------|----------------------|---------|
| a-4a1qxb-d5wgvebrf | -           | Standard Application | -       |

Below the table, there is a section titled 'API Key Information' with the following details:

- Key: a-4a1qxb-d5wgvebrf
- Description: -
- Date Added: Nov 10, 2022 2:20 PM
- Last Update: Nov 10, 2022 2:20 PM
- Last Edited By: 310819106007@smartinternz.com
- Expires: Never

At the bottom of the page, there is a status bar showing '1 Simulation running'.

## 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-zMRKvshayi"
    }
}

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:

