

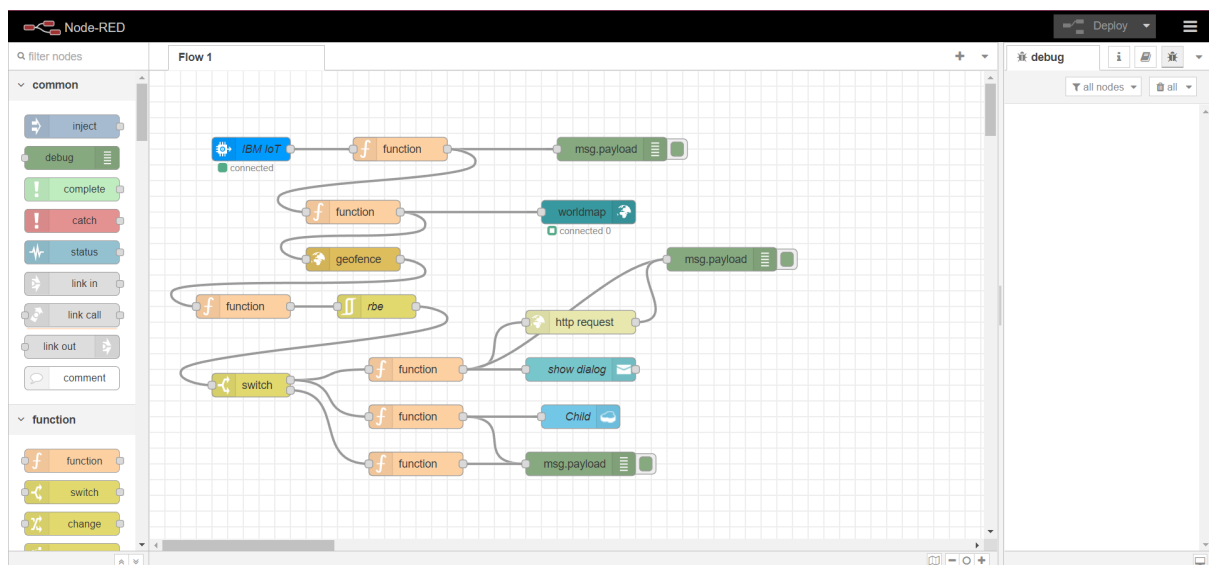
# Project Development – Delivery plan sprint-2

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

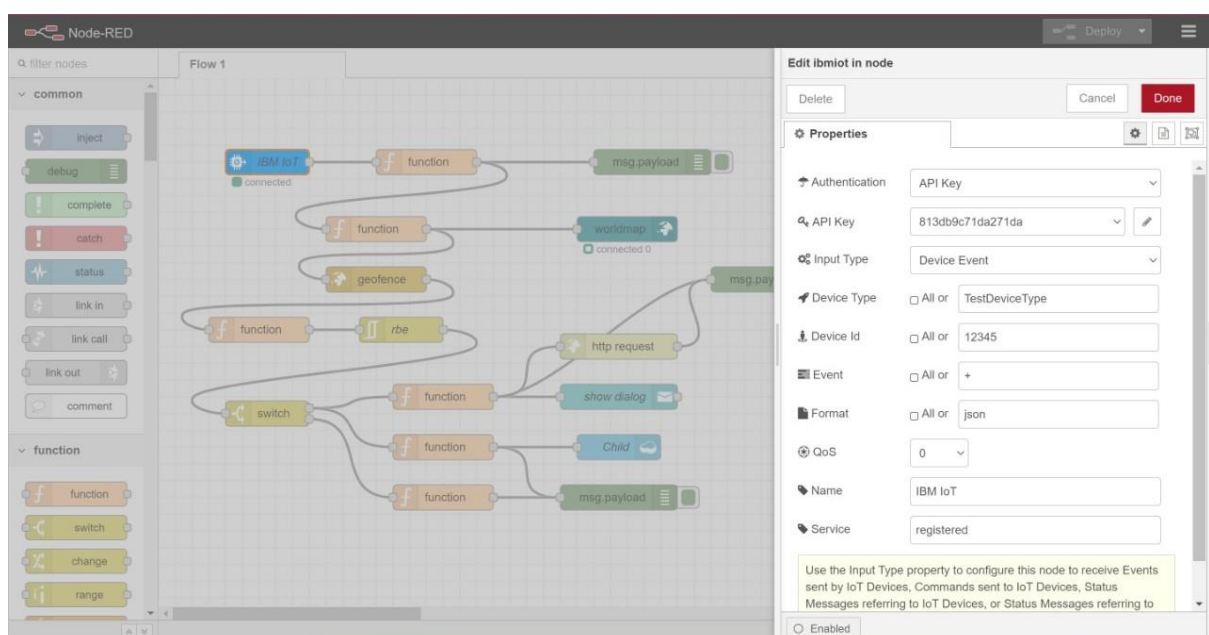
TEAM ID:PNT2022TMID27117

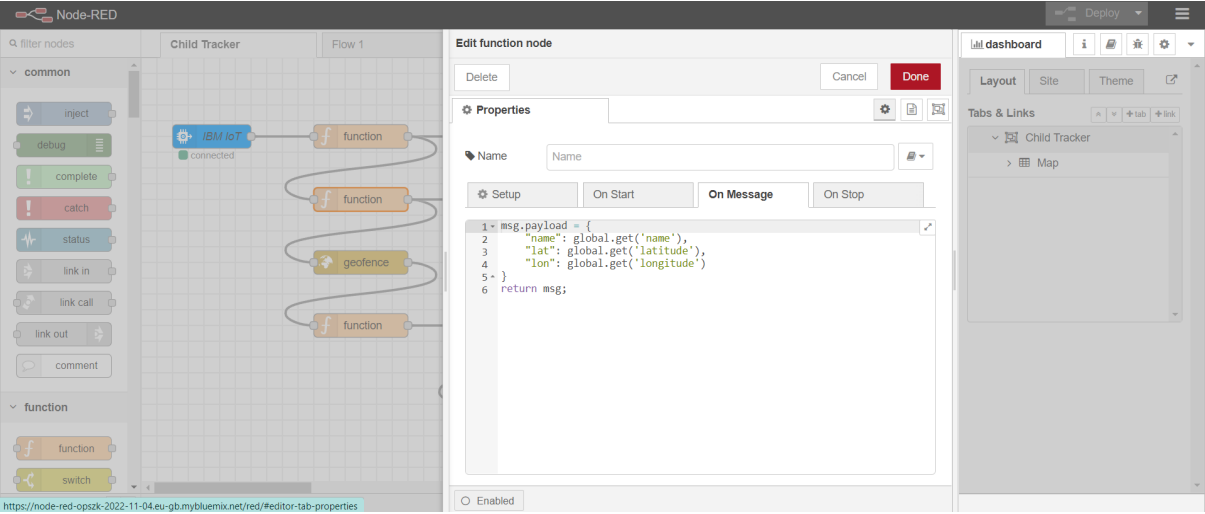
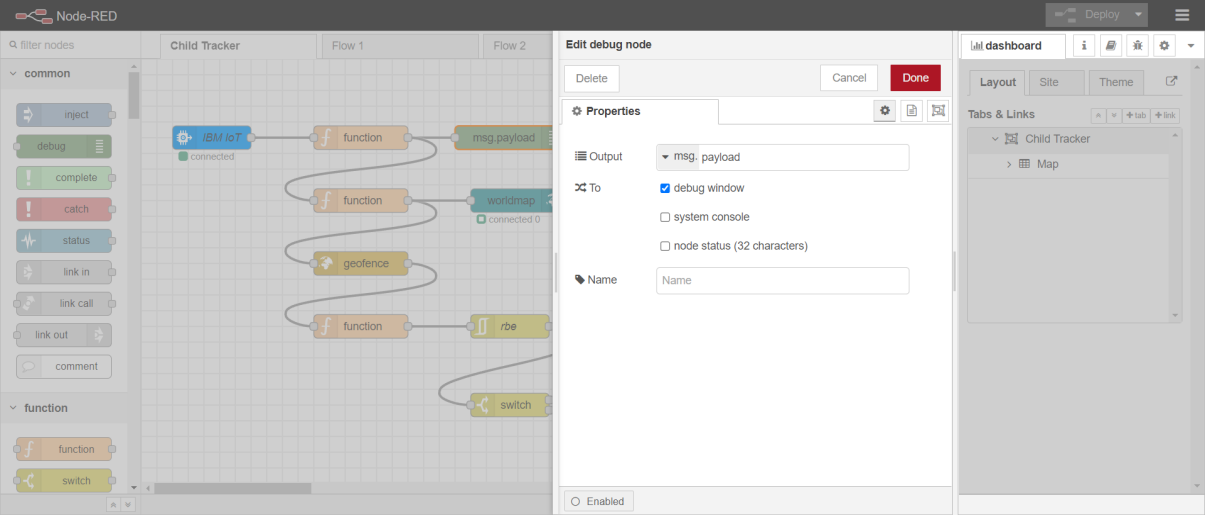
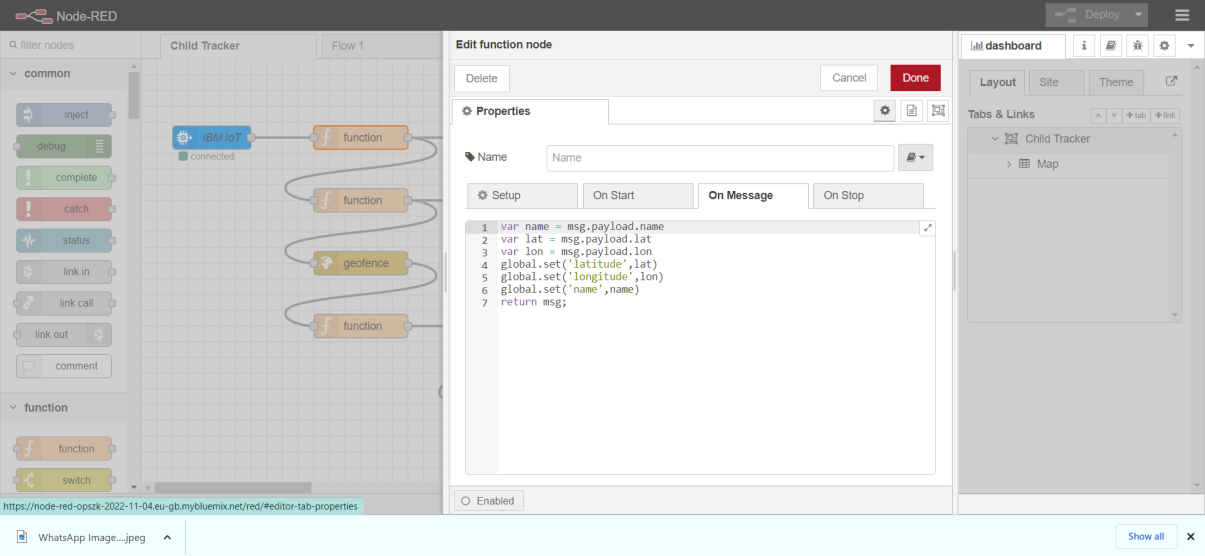
### Creating Node-Red service and connecting with IBM cloud

### Creating Node-Red service:



### Codes in each 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 to a `function` node. This is followed by another `function` node, then a `worldmap` node (connected 0). The flow then branches into two paths: one through a `geofence` node and another through an `rbe` node. Both paths converge at a `switch` node. From the `switch`, the flow branches into three paths: one through a `function` node to an `http request` node, another through a `function` node to a `show dialog` node, and a third through a `function` node to a `Child` node. All three paths converge at a final `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: ☐

The geofence node is visualized on a map showing a purple circular area over a city street map.

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

**Child Tracker Flow:** The flow starts with an `IBM IoT` node connected to a `function` node. This is followed by another `function` node, then a `geofence` node, and finally a `function` node.

**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 named "Child Tracker" containing a map.

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

Node-RED interface showing a flow named "Child Tracker" with nodes: inject, debug, complete, catch, status, link in, link call, link out, comment, function, switch, msg.payload, worldmap, rbe, and switch. The "Edit filter node" panel is open, showing properties: Mode (block unless value changes), Property (msg. payload), Apply mode separately for each (checked), msg. topic, and Name (rbe).

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

Node-RED interface showing the same flow. The "Edit switch node" panel is open, showing properties: Name (Name), Property (msg. payload), and rules: is false (→ 1) and is true (→ 2). The "checking all rules" checkbox is checked.

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

Node-RED interface showing the same flow. The "Edit function node" panel is open, showing the function code:

```
1 var d = new Date();
2
3 var utc = d.getTime() + (d.getTimezoneOffset() * 60000);
4
5 var offset = 5.5; // This is the offset for UTC*3, in your case (UTC*1)
6
7 newDate = new Date(utc + (3600000 * 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;
```

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

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

**Flow 1:** The flow starts with a "function" node, followed by a "msg.payload" node, then another "function" node, a "worldmap" node, a "geofence" node, another "function" node, an "rbe" node, and finally a "switch" node.

**Edit function node:**

- Name: Name
- Setup: On Message
- Code:

```
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":"Entry",
8   "Time": newDate.toLocaleString(),
9   "name": global.get('name'),
10  "lat": global.get('latitude'),
11  "lon": global.get('longitude')
12 };
13
14 return msg;
```

Enabled

Node-RED interface showing a flow named "Flow 1" and the "Edit http request node" dialog.

**Flow 1:** The flow starts with a "function" node, followed by a "msg.payload" node, then another "function" node, a "worldmap" node, a "geofence" node, another "function" node, an "rbe" node, and finally a "switch" node. The "switch" node has four outputs: "function", "Child", "show dialog", and "msg.payload".

**Edit http request node:**

- 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

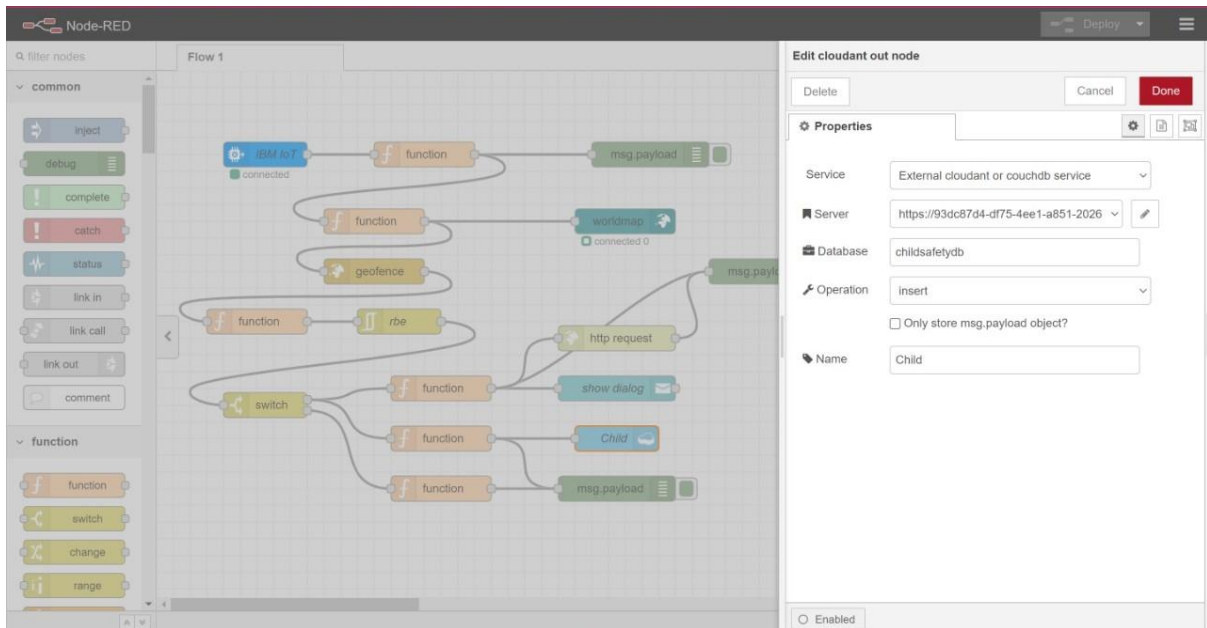
Node-RED interface showing a flow named "Child Tracker" and the "Edit notification node" dialog.

**Flow 1:** The flow starts with a "function" node, followed by a "msg.payload" node, then another "function" node, a "worldmap" node, a "geofence" node, another "function" node, an "rbe" node, and finally a "switch" node. The "switch" node has four outputs: "function", "Child", "show dialog", and "msg.payload".

**Edit notification node:**

- 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 Dialog
- Note: checking Accept raw HTML/JavaScript can allow injection of

Enabled



## 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 and ID. A search bar is present. Below the search bar, a table lists the API keys. The table has columns for Key, Description, Role, and Expires. There are 2 results shown.

Key	Description	Role	Expires
a-4o1qxb-d5wguvebrf	-	Standard Application	-
a-4o1qxb-ecmygwzdc	API Key for the device simulator	Standard Application	-

At the bottom, it indicates '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 detailed information for a specific API key. The table lists the API keys, and the selected key is highlighted. Below the table, there is a section for 'API Key Information' and 'Access Control/Permissions'.

Key	Description	Role	Expires
a-4o1qxb-d5wguvebrf	-	Standard Application	-

API Key Information

Key	Description	Last Edited By	Expires
a-4o1qxb-d5wguvebrf	-	310819106007@smartinternz.com	Never

Access Control/Permissions

Date Added	Last Update
Nov 10, 2022 2:20 PM	Nov 10, 2022 2:20 PM

At the bottom, it indicates '1 Simulation running'.



## Transferring values from Python Code:

```

child.py - C:\Users\Anu\AppData\Local\Programs\Python\Python37\chil... (py37) (3.7.0)
File Edit Format Run Options Window Help

import json
import wiotp.sdk.device

myConfig = {

    "identity":{f
        "orgid":"40lqxb",
        "typeid":"TestDeviceType",
        "deviceid": "12345"
    },
    "auth":{
        "token":"gnhXvZn-sMMKvsixyl"
    }
}

client= wiotp.sdk.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= 78.5488783
    myData={"name": name, 'lat':latitude, 'lon': longitude}
    client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0, onPublish=None)
    print("Data published to IHM IoT platform: ",myData)
    time.sleep(5)

client.disconnect()
```

## Node-Red:

