

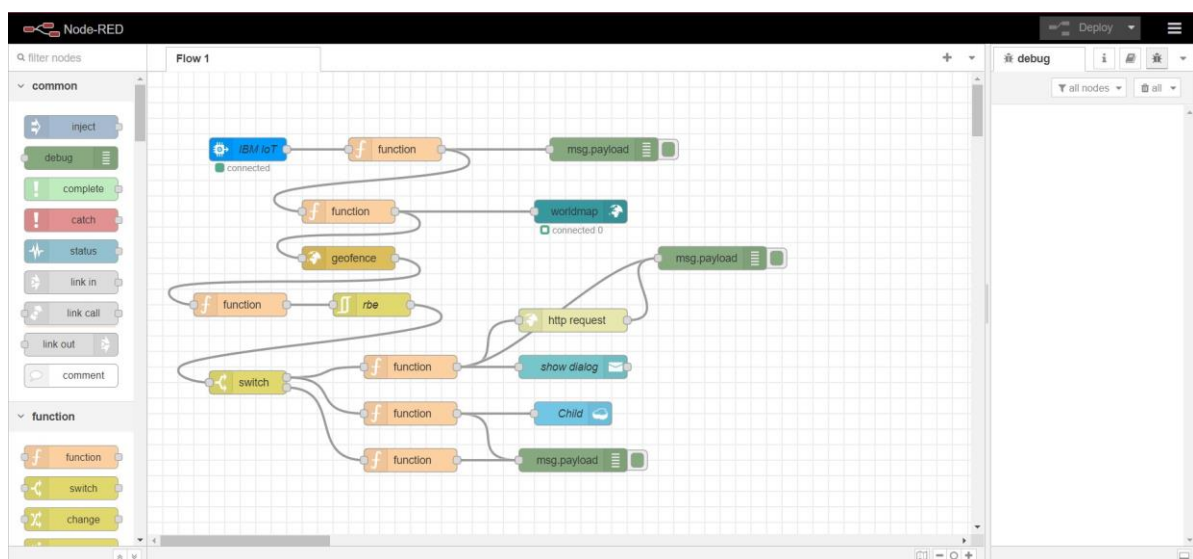
Project Development – Delivery plan sprint-2

IoT Based Safety Gadget for Child Safety Monitoring & Notification

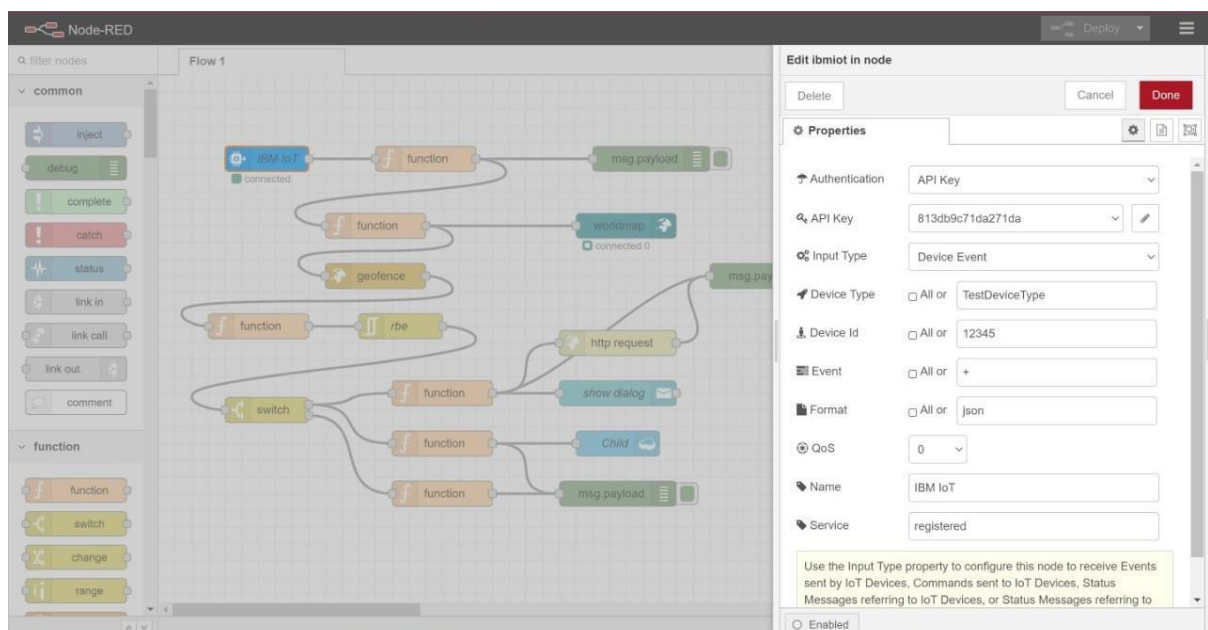
TEAM ID:PNT2022TMID16367

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. The flow is titled "Child Tracker" and contains an "IBM IoT" node connected to a series of function nodes, including a "geofence" node. The "Edit function node" dialog is open, showing the "On Message" tab with 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" section shows the node name as "Name". The "Enabled" checkbox is checked. The "dashboard" sidebar is visible on the right, showing the "Child Tracker" tab and a "Map" node.

Node-RED interface showing the "Edit debug node" dialog. The flow is titled "Child Tracker" and contains an "IBM IoT" node connected to a series of function nodes, including a "geofence" node. The "Edit debug node" dialog is open, showing the "Output" section with the following settings:

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

The "Properties" section shows the node name as "Name". The "Enabled" checkbox is checked. The "dashboard" sidebar is visible on the right, showing the "Child Tracker" tab and a "Map" node.

Node-RED interface showing the "Edit function node" dialog. The flow is titled "Child Tracker" and contains an "IBM IoT" node connected to a series of function nodes, including a "geofence" node. The "Edit function node" dialog is open, showing the "On Message" tab with 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" section shows the node name as "Name". The "Enabled" checkbox is checked. The "dashboard" sidebar is visible on the right, showing the "Child Tracker" tab and 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 connects to a **worldmap** node (connected). The **worldmap** node connects to a **msg.payload** node. The **worldmap** node also connects to a **geofence** node. The **geofence** node connects to a **function** node. The **function** node connects to a **switch** node. The **switch** node has two outputs: one to a **function** node and another to a **function** node. The **function** node connects to a **http request** node. The **http request** node connects to a **show dialog** node. The **show dialog** node connects to a **Child** node. The **Child** node connects to a **msg.payload** node.

Edit worldmap node:

- 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 connects to a **worldmap** node (connected). The **worldmap** node connects to a **msg.payload** node. The **worldmap** node also connects to a **geofence** node. The **geofence** node connects to a **function** node. The **function** node connects to a **switch** node. The **switch** node has two outputs: one to a **function** node and another to a **function** node. The **function** node connects to a **http request** node. The **http request** node connects to a **show dialog** node. The **show dialog** node connects to a **Child** node. The **Child** node connects to a **msg.payload** node.

Edit geofence node:

- Properties:**
- Map:** A map showing a geofence area around a location in Chennai, India. The geofence is a purple circle.
- Floor:** ground
- Ceiling:** Infinity
- Action:** add "inarea" property
- Enable output of zones to WorldMap node:** ☐
- Enabled:** ☐

Node-RED interface showing a flow named "Flow 1" 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 connects to a **worldmap** node (connected). The **worldmap** node connects to a **msg.payload** node. The **worldmap** node also connects to a **geofence** node. The **geofence** node connects to a **function** node. The **function** node connects to a **switch** node. The **switch** node has two outputs: one to a **function** node and another to a **function** node. The **function** node connects to a **http request** node. The **http request** node connects to a **show dialog** node. The **show dialog** node connects to a **Child** node. The **Child** node connects to a **msg.payload** 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:

- Layout:** Site, Theme
- Tabs & Links:**
- Child Tracker:**
- 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, and switch. The "Edit filter node" dialog 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" dialog is open, showing properties: Name (Name), Property (msg.payload), and rules: Is false (1), 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" dialog is open, showing the "On Message" tab with the following 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 the 'Edit function node' dialog. The function 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 interface also shows a sidebar with nodes (inject, debug, complete, catch, status, link in, link call, link out, comment) and a dashboard view with tabs for 'Child Tracker' and 'Map'.

Node-RED interface showing the 'Edit http request node' dialog. The properties are:

- 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

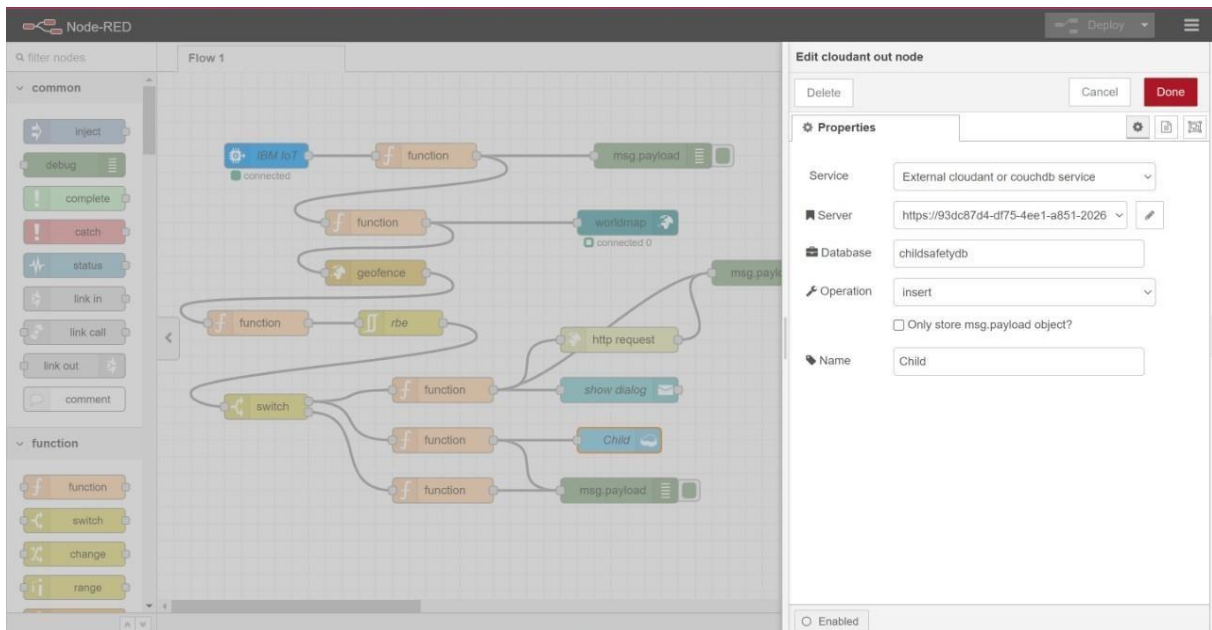
The interface also shows a sidebar with nodes (inject, debug, complete, catch, status, link in, link call, link out, comment) and a dashboard view with tabs for 'Child Tracker' and 'Map'.

Node-RED interface showing the 'Edit notification node' dialog. The properties are:

- 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

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

The interface also shows a sidebar with nodes (inject, debug, complete, catch, status, link in, link call, link out, comment) and a dashboard view with tabs for 'Child Tracker' and 'Map'.



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 '1916158@saec.ac.in' and the ID '4nppac'. A 'Generate API Key' button is visible. The main content area displays a table of API keys with columns for 'Key', 'Description', 'Role', and 'Expires'. There are 2 results listed:

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 page header includes the user '1916158@saec.ac.in' and the ID '4nppac'. A 'Generate API Key' button is visible. The main content area displays a table of API keys with columns for 'Key', 'Description', 'Role', and 'Expires'. There are 2 results listed:

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

Below the table, there is a section for 'API Key Information' and 'Access Control/Permissions'. The 'API Key Information' section shows the following details:

Field	Value
Key	a-4o1qxb-d5wguvebrf
Description	-
Date Added	Nov 10, 2022 2:20 PM
Last Update	Nov 10, 2022 2:20 PM

The 'Access Control/Permissions' section shows the following details:

Field	Value
Last Edited By	1916158@saec.ac.in
Expires	Never

At the bottom, it indicates '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:

