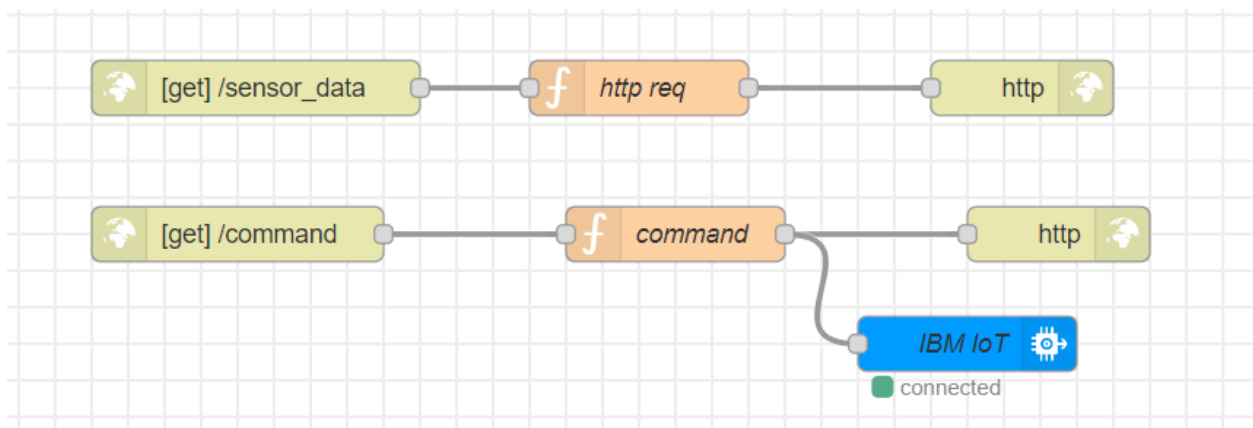


Building Mobile App

TEAM ID	PNT2022TMID48692
PROJECT TITLE	REAL TIME RIVER WATER QUALITY MONITORING AND CONTROL SYSTEM

Configure The Mobile App For Controlling Motor Using Buttons

Step 1: In Node-red use Http input and Http response for creating API for the Command from the Application and connect it to the IBM IOT Out



Step 2: Edit the Command function to pass the command value to the IBM IOT

Delete

Cancel

Done

⚙️ Properties

⚙️ 📄 🖨️

🔖 Name

command

📄 ▼

⚙️ Setup

On Start

On Message

On Stop

1 msg.payload=msg.payload.command

2 msg.payload={'command':msg.payload}

3 return msg;

↗️

Step 3: Configure the IBM IOT Out node and change the Output Type as Device Command

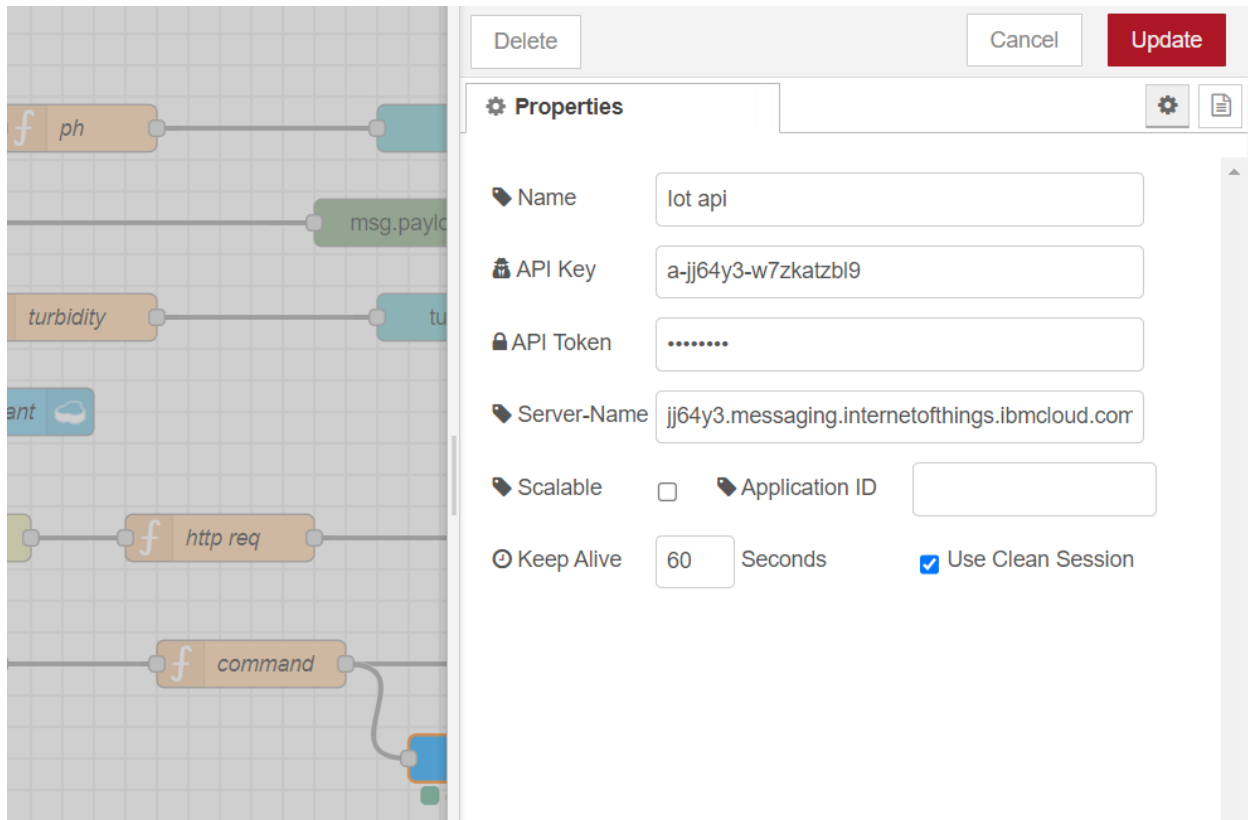
Edit ibmiot out node

Delete Cancel Done

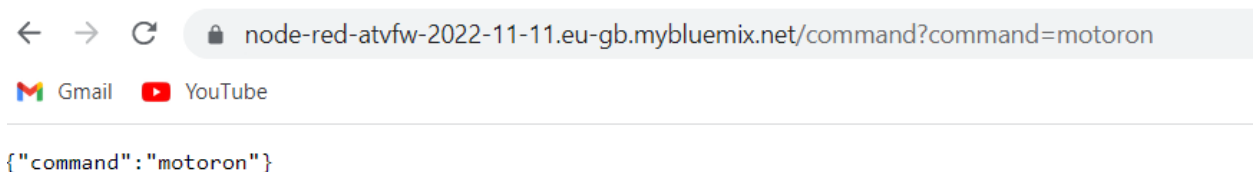
Properties

- Authentication: API Key
- API Key: lot api
- Output Type: Device Command
- Device Type: Nodered
- Device Id: 12345
- Command Type: command
- Format: json
- Data: command
- QoS: 0
- Name: IBM IoT
- Service: registered

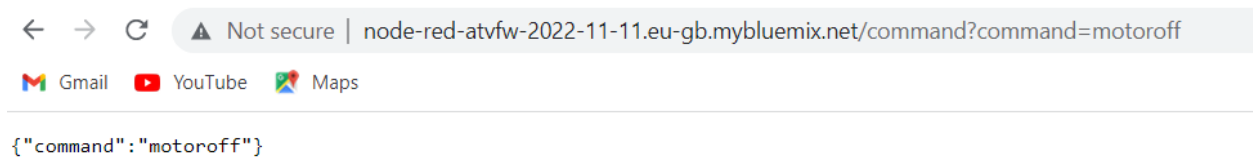
Step 4: Configure the API Key in the IBM IOT Out Node



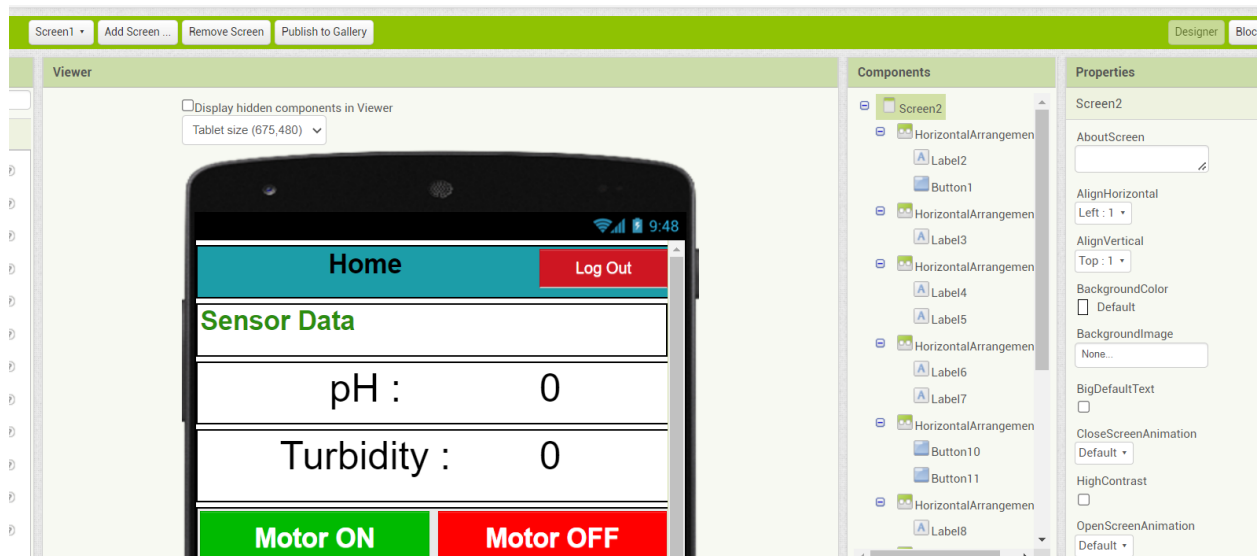
Step 5: In the Command API when we pass the command as motor on it show result as given below



Step 6: In the Command API when we pass the command as motor off it show result as given below



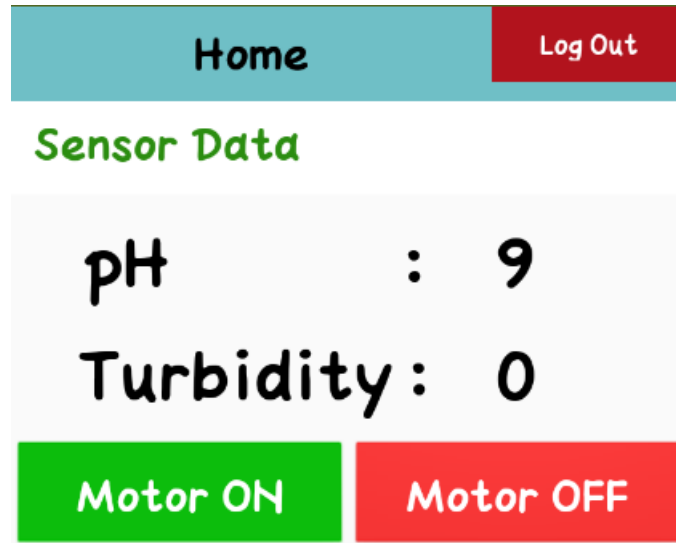
Step 7: In the MIT Inventor app now add the two buttons Motor ON and Motor OFF for controlling the motors



Step 8: Use the Button Blocks for Setting the Web URL and call the Web using Get Method



Step 9: In the App we can have two buttons and data will be shown in the App



Step 10: When we click the Motor ON in the App we receive the command in the Python

```
Published data Successfully: {'ph': 5, 'turbidity': 4}
Published data Successfully: {'ph': 6, 'turbidity': 8}
Published data Successfully: {'ph': 9, 'turbidity': 9}
Published data Successfully: {'ph': 9, 'turbidity': 2}
Published data Successfully: {'ph': 12, 'turbidity': 8}
Published data Successfully: {'ph': 5, 'turbidity': 4}
Published data Successfully: {'ph': 0, 'turbidity': 9}
Message received from IBM IoT Platform:motoron
Motor is turned ON
Published data Successfully: {'ph': 13, 'turbidity': 2}
Published data Successfully: {'ph': 5, 'turbidity': 8}
Published data Successfully: {'ph': 7, 'turbidity': 7}
```

Step 10: When we click the Motor OFF in the App we receive the command in the Python

```
Published data Successfully: {'ph': 10, 'turbidity': 7}
Published data Successfully: {'ph': 4, 'turbidity': 7}
Published data Successfully: {'ph': 3, 'turbidity': 7}
Published data Successfully: {'ph': 10, 'turbidity': 2}
Published data Successfully: {'ph': 14, 'turbidity': 10}
Published data Successfully: {'ph': 0, 'turbidity': 1}
Message received from IBM IoT Platform:motoroff
Motor is turned OFF
Published data Successfully: {'ph': 7, 'turbidity': 6}
Published data Successfully: {'ph': 0, 'turbidity': 0}
Published data Successfully: {'ph': 12, 'turbidity': 6}
Published data Successfully: {'ph': 0, 'turbidity': 11}
```