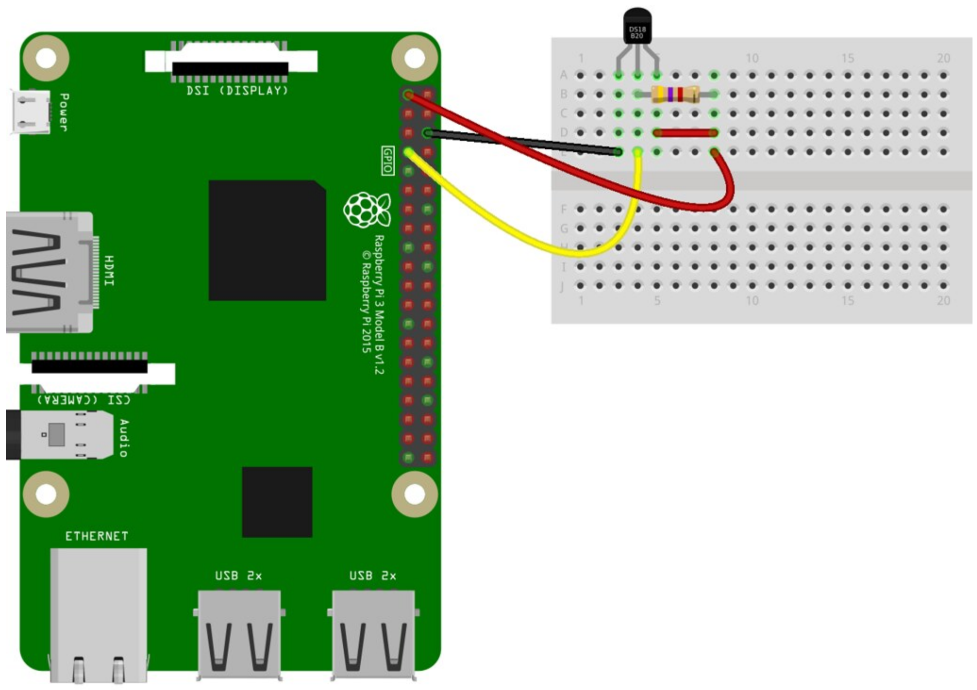
SPRINT 1

Sprint 1 focuses on allowing users to get local data from beacons on their devices.

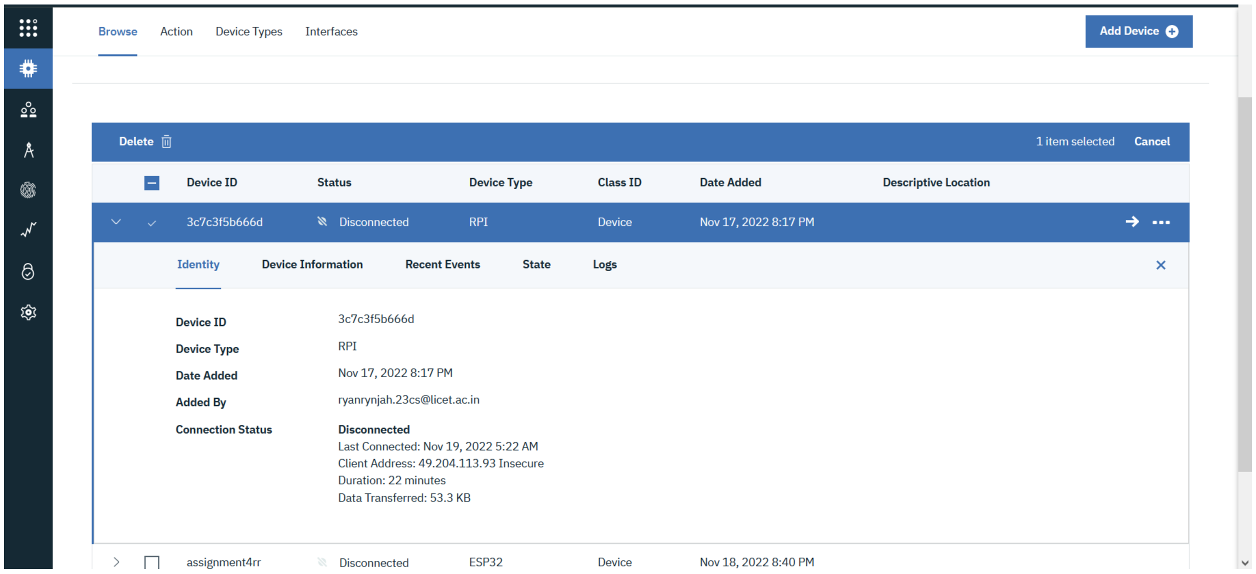
IoT device - Raspberry Pi 3B+



```python
import time
import random
import paho.mqtt.client as mqtt
import json

#from w1thermsensor import W1ThermSensor
#sensor = W1ThermSensor()
#def Temp():
    #return temperature = sensor.get_temperature()

#The above lines of code would be used when getting temperature data from a DS18B20 sensor.
#Due to hardware limitations we are simulating values using random function.
```

```python
def Temp():
    return random.randint(0,99);

ORG = "csgusn"
DEVICE_TYPE = "RPI"
TOKEN = "1123581321"
DEVICE_ID = "3c7c3f5b666d"  #Credentials of device as per created on IBM
IoT platform.

server = ORG + ".messaging.internetofthings.ibmcloud.com";
pubTopic1 = "iot-2/evt/status1/fmt/json";   #event named status 1 in JSON
format

authMethod = "use-token-auth";
token = TOKEN;
clientId = "d:" + ORG + ":" + DEVICE_TYPE + ":" + DEVICE_ID;

mqttc = mqtt.Client(client_id=clientId)
mqttc.username_pw_set(authMethod, token)
mqttc.connect(server, 1883, 60)         #Connecting using MQ Telemetry
Transport Protocol

while True:

    tempDict = { "d": {"temperature": Temp()} };    #Temporary storage in
a dictionary

    tempJson = json.dumps(tempDict);    #Conversion from dictionary to
JSON

    mqttc.publish(pubTopic1, tempJson)      #Publish payload
    print("Reading Taken");

    time.sleep(5);
```

IBM IoT Platform



Browse | Action | Device Types | Interfaces | Add Device

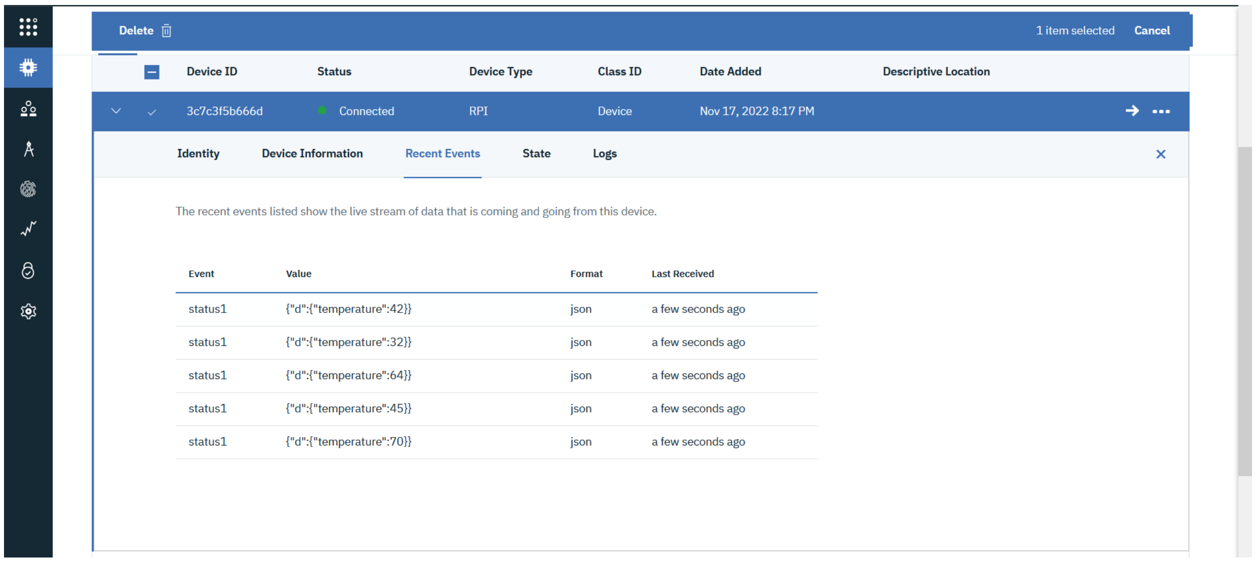
Delete | 1 item selected | Cancel

Device ID	Status	Device Type	Class ID	Date Added	Descriptive Location
3c7c3f5b666d	Disconnected	RPI	Device	Nov 17, 2022 8:17 PM	
assignment4rr	Disconnected	ESP32	Device	Nov 18, 2022 8:40 PM	

Identity | Device Information | Recent Events | State | Logs

Device ID	3c7c3f5b666d
Device Type	RPI
Date Added	Nov 17, 2022 8:17 PM
Added By	ryanrynjah.23cs@licet.ac.in
Connection Status	Disconnected Last Connected: Nov 19, 2022 5:22 AM Client Address: 49.204.113.93 Insecure Duration: 22 minutes Data Transferred: 53.3 KB

The device as created in IBM watson platform. On running the program, Status changes to Connected and published temperature data is shown in recent events.



Delete | 1 item selected | Cancel

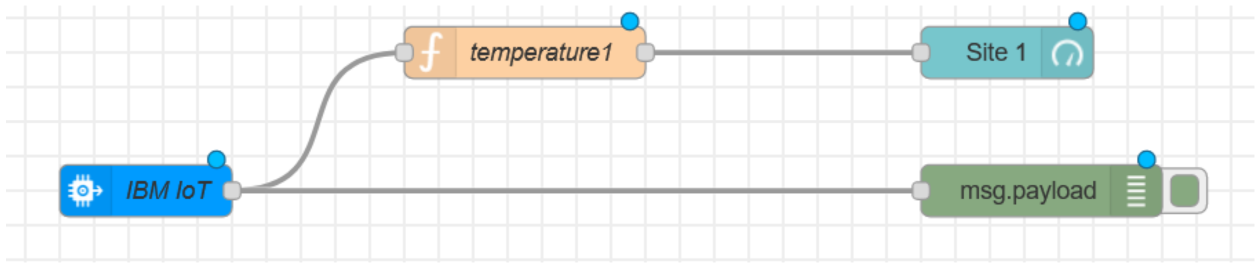
Device ID	Status	Device Type	Class ID	Date Added	Descriptive Location
3c7c3f5b666d	Connected	RPI	Device	Nov 17, 2022 8:17 PM	

Identity | Device Information | Recent Events | State | Logs

The recent events listed show the live stream of data that is coming and going from this device.

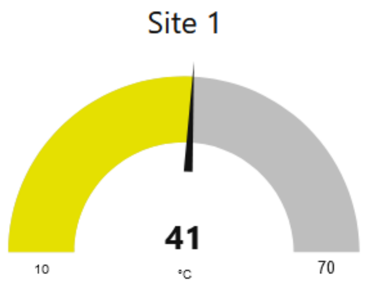
Event	Value	Format	Last Received
status1	{"d":{"temperature":42}}	json	a few seconds ago
status1	{"d":{"temperature":32}}	json	a few seconds ago
status1	{"d":{"temperature":64}}	json	a few seconds ago
status1	{"d":{"temperature":45}}	json	a few seconds ago
status1	{"d":{"temperature":70}}	json	a few seconds ago

Node-RED

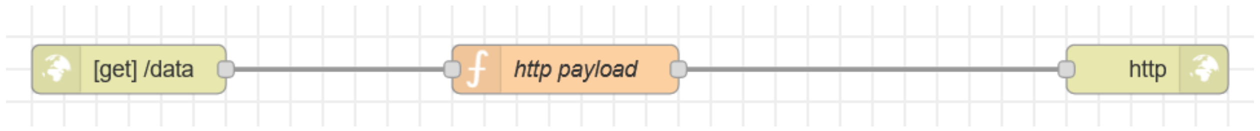


1. IBM IoT IN node allows us to get data from IBM IoT device.
2. Temperature is a function that extracts the payload.
   global.set("temperature1",msg.payload.d.temperature1)
   msg.payload=msg.payload.d.temperature1
   return msg;

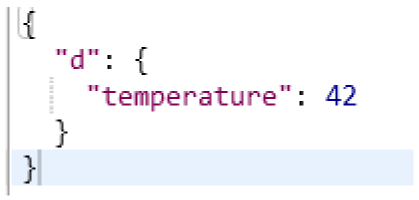
3. Site is a gauge to view values on the Node-RED dashboard.



4. msg.payload allows for debugging.



5. [get]/data is a GET HTTP method which sends the json data through the http node.

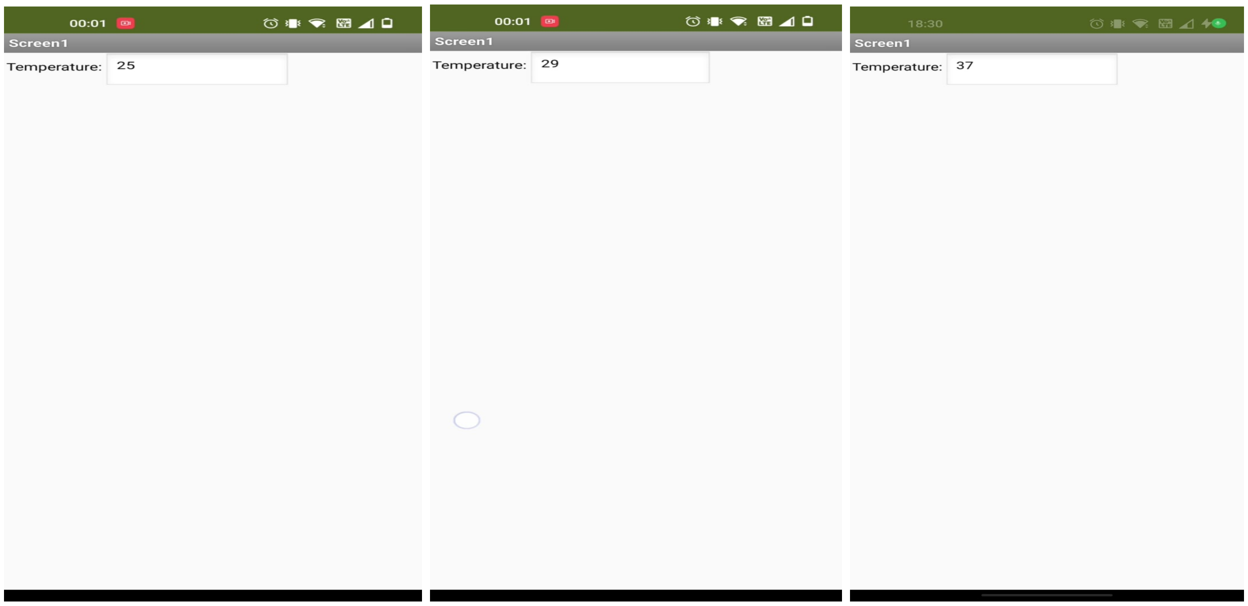


```
{
  "d": {
    "temperature": 42
  }
}
```

6. This allows us to export the data through HTTP for use by other applications.

User Device

Employee user interface is an Android Application.

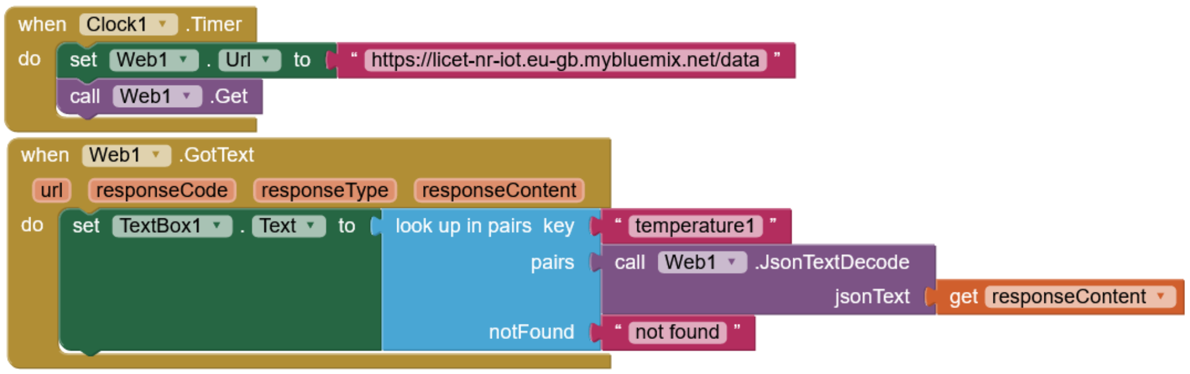


Screen1
Temperature: 25

Screen1
Temperature: 29

Screen1
Temperature: 37

MIT App Inventor



```
when Clock1 .Timer
do  set Web1 . Url to "https://licet-nr-iot.eu-gb.mybluemix.net/data"
    call Web1 .Get

when Web1 .GotText
 url  responseCode  responseType  responseContent
do  set TextBox1 . Text to  look up in pairs  key  "temperature1"
                                              pairs  call Web1 .JsonTextDecode
                                                                     jsonText  get responseContent
                                           notFound  "not found"
```

The app fetches the JSON data from the HTTP out from Node-RED, extracts the relevant data and pastes it in a text box.