-
-
-
-
-
-

1.
2.
3.
4.
5.
1.

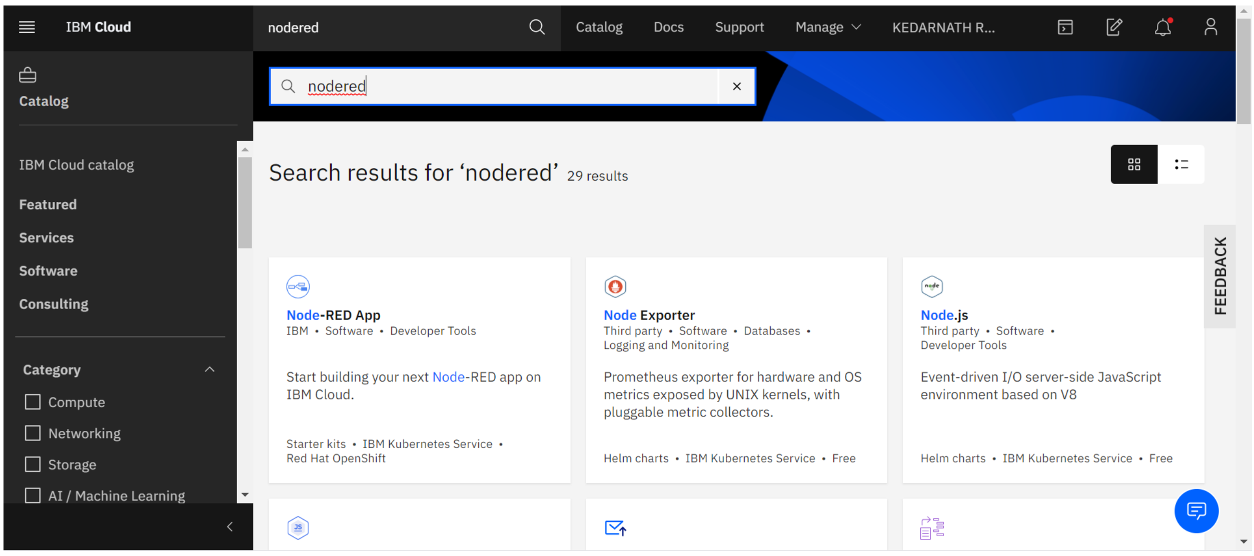
Now create that service

After creatin that service press on the launch button on the dash board

After launching the watson iot platform add a device in it

After this come to the cloud main page

Search nodered in the catalog, click on the nodered app and create nodered application



IBM Cloud

nodered

Catalog Docs Support Manage KEDARNATH R...

Catalog

IBM Cloud catalog

Featured

Services

Software

Consulting

Category

□ Compute

□ Networking

□ Storage

□ AI / Machine Learning

nodered

Search results for 'nodered' 29 results

Node-RED App
IBM • Software • Developer Tools
Start building your next Node-RED app on IBM Cloud.
Starter kits • IBM Kubernetes Service • Red Hat OpenShift

Node Exporter
Third party • Software • Databases • Logging and Monitoring
Prometheus exporter for hardware and OS metrics exposed by UNIX kernels, with pluggable metric collectors.
Helm charts • IBM Kubernetes Service • Free

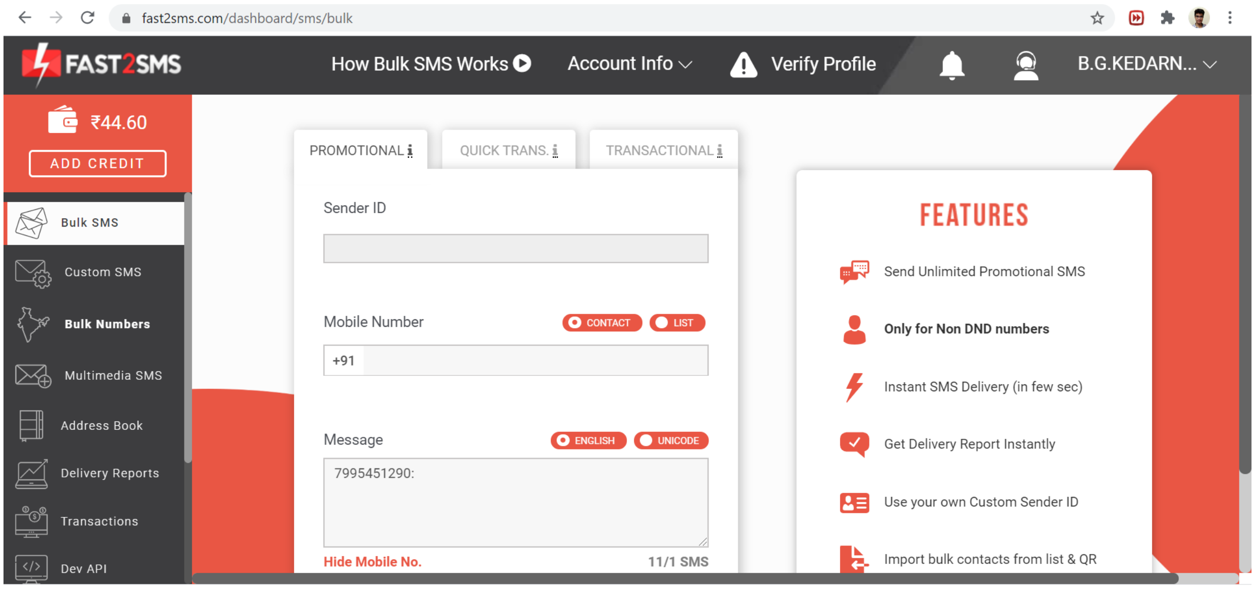
Node.js
Third party • Software • Developer Tools
Event-driven I/O server-side JavaScript environment based on V8
Helm charts • IBM Kubernetes Service • Free

FEEDBACK

2.Create a fast2sms account (for sending alert messages):-

Now create fast2sms account to send alert messages to the user

Search fast2sms in browser open that website and create an account there



fast2sms.com/dashboard/sms/bulk

FAST2SMS How Bulk SMS Works Account Info Verify Profile B.G.KEDARN...

₹44.60

ADD CREDIT

Bulk SMS

Custom SMS

Bulk Numbers

Multimedia SMS

Address Book

Delivery Reports

Transactions

Dev API

PROMOTIONAL QUICK TRANS. TRANSACTIONAL

Sender ID

Mobile Number CONTACT LIST

+91

Message ENGLISH UNICODE

7995451290:

Hide Mobile No. 11/1 SMS

FEATURES

Send Unlimited Promotional SMS

Only for Non DND numbers

Instant SMS Delivery (in few sec)

Get Delivery Report Instantly

Use your own Custom Sender ID

Import bulk contacts from list & QR

3.Code snippet for sendind sensor data to the watsoniot platform and for sending alert messages to the user

Note:- we dont have any sensors to send the data to the cloud so we send sensor data with python code

The following code is the code used for this task

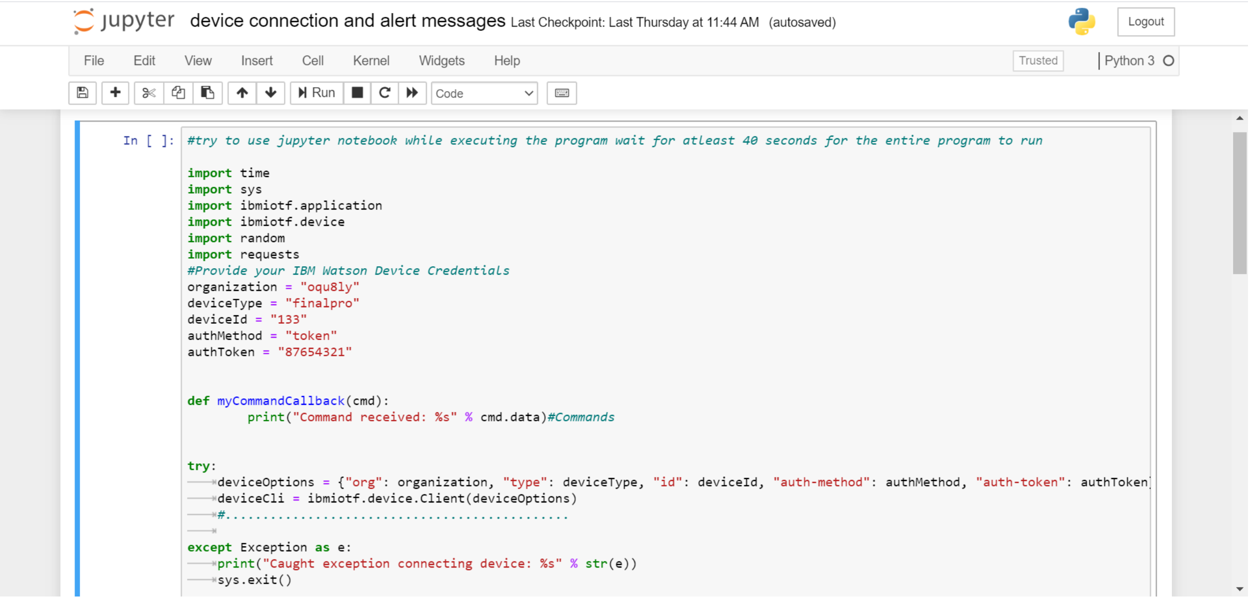
In the below code enter the credentials of the device that you created in the watson iot platform

PYTHON CODE

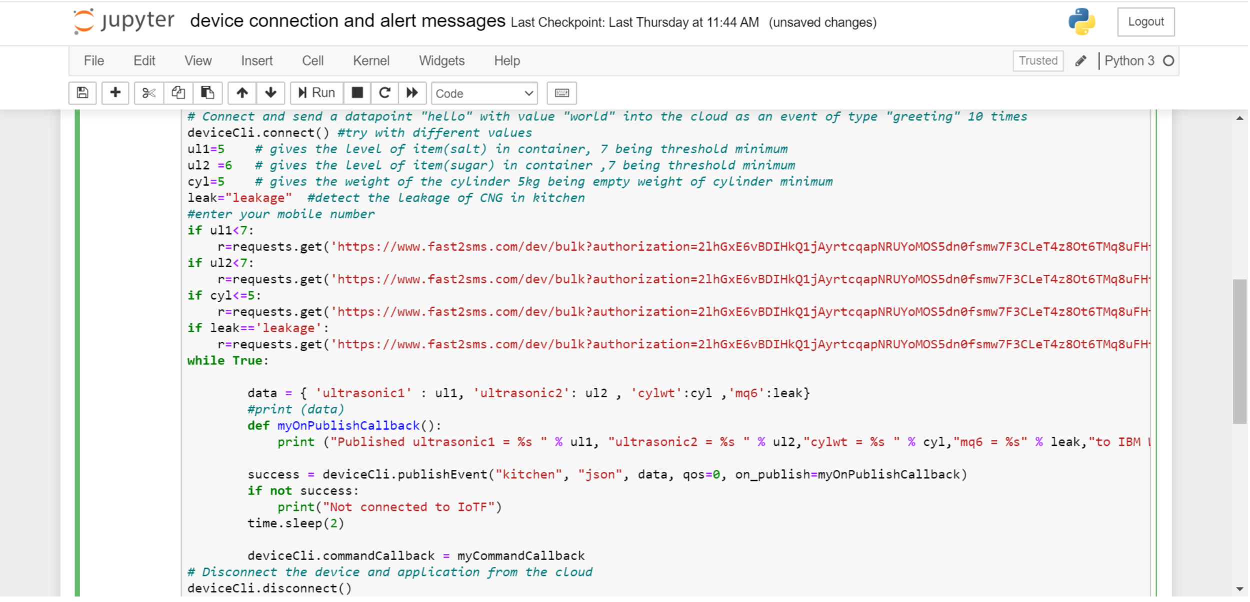
```
import time

import sys

import ibmiotf.application

import ibmiotf.device

import random

import requests

#Provide your IBM Watson Device Credentials

organization = "oqu8ly"

deviceType = "finalpro"

deviceId = "133"

authMethod = "token"

authToken = "87654321"



def myCommandCallback(cmd):

        print("Command received: %s" %
cmd.data)#Commands



try:
        deviceOptions = {"org": organization, "type": deviceType, "id": deviceId, "auth-method": authMethod, "auth-token": authToken}

        deviceCli = ibmiotf.device.Client(deviceOptions)

        #..............................................


except Exception as e:

        print("Caught exception connecting device: %s" % str(e))

        sys.exit()


# Connect and send a datapoint "hello" with value "world" into the cloud as an event of type "greeting" 10 times

deviceCli.connect() #try with different values

ul1=5    # gives the level of item(salt) in container, 7 being threshold minimum

ul2 =6   # gives the level of item(sugar) in container ,7 being threshold minimum

cyl=5    # gives the weight of the cylinder 5kg being empty weight of cylinder minimum

leak="leakage"  #detect the leakage of CNG in kitchen

#enter your mobile number

if ul1<7:


r=requests.get('https://www.fast2sms.com/dev/bulk?authorization=2lhGxE6vBDIHkQ1jAyrtcqapNRUYoMOS5dn0fsmw7F3CLeT4z8Ot6TMq8uFHfZEVycsB3GCz9wx2piNa&sender_id=FSTSMS&message=YOUR SALT IS ABOUT TO
COMPLETE&language=english&route=p&numbers=9014459578')

if ul2<7:

r=requests.get('https://www.fast2sms.com/dev/bulk?authorization=2lhGxE6vBDIHkQ1jAyrtcqapNRUYoMOS5dn0fsmw7F3CLeT4z8Ot6TMq8uFHfZEVycsB3GCz9wx2piNa&sender_id=FSTSMS&message=YOUR SUGAR IS ABOUT TO COMPLETE&language=english&route=p&numbers=9014459578')

if cyl<=5:

r=requests.get('https://www.fast2sms.com/dev/bulk?authorization=2lhGxE6vBDIHkQ1jAyrtcqapNRUYoMOS5dn0fsmw7F3CLeT4z8Ot6TMq8uFHfZEVycsB3GCz9wx2piNa&sender_id=FSTSMS&message=CYLINDER IS OVER&language=english&route=p&numbers=9014459578')

if leak=='leakage':

r=requests.get('https://www.fast2sms.com/dev/bulk?authorization=2lhGxE6vBDIHkQ1jAyrtcqapNRUYoMOS5dn0fsmw7F3CLeT4z8Ot6TMq8uFHfZEVycsB3GCz9wx2piNa&sender_id=FSTSMS&message=GAS IS BEING LEAKED FROM CYLINDER AND EXHAUST FAN HAS BEEN SWITCHED ON&language=english&route=p&numbers=9014459578')

while True:

        data = { 'ultrasonic1' : ul1, 'ultrasonic2': ul2 , 'cylwt':cyl ,'mq6':leak}

        #print (data)
```

```
    def myOnPublishCallback():
        print ("Published ultrasonic1 = %s " % ul1, "ultrasonic2 = %s " % ul2,"cylwt = %s " % cyl,"mq6 = %s" % leak,"to IBM Watson")


    success = deviceCli.publishEvent("kitchen", "json", data, qos=0, on_publish=myOnPublishCallback)
    if not success:
        print("Not connected to IoTF")
    time.sleep(2)


    deviceCli.commandCallback = myCommandCallback



# Disconnect the device and application from the cloud
deviceCli.disconnect()
```

NOTE:- DONT PLACE MESSAGE CODE INSIDE THE LOOP IF YOU DO THAT THE FLOW OF MESSAGES WONT STOP


jupyter
device connection and alert messages Last Checkpoint: Last Thursday at 11:44 AM (autosaved)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3
Run
Code
In [ ]:

```
#try to use jupyter notebook while executing the program wait for atleast 40 seconds for the entire program to run

import time
import sys
import ibmiotf.application
import ibmiotf.device
import random
import requests
#Provide your IBM Watson Device Credentials
organization = "oqu8ly"
deviceType = "finalpro"
deviceId = "133"
authMethod = "token"
authToken = "87654321"


def myCommandCallback(cmd):
        print("Command received: %s" % cmd.data)#Commands


try:
    deviceOptions = {"org": organization, "type": deviceType, "id": deviceId, "auth-method": authMethod, "auth-token": authToken
    deviceCli = ibmiotf.device.Client(deviceOptions)
    #..............................................
    
except Exception as e:
    print("Caught exception connecting device: %s" % str(e))
    sys.exit()
```


jupyter
device connection and alert messages Last Checkpoint: Last Thursday at 11:44 AM (unsaved changes)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3
Run
Code

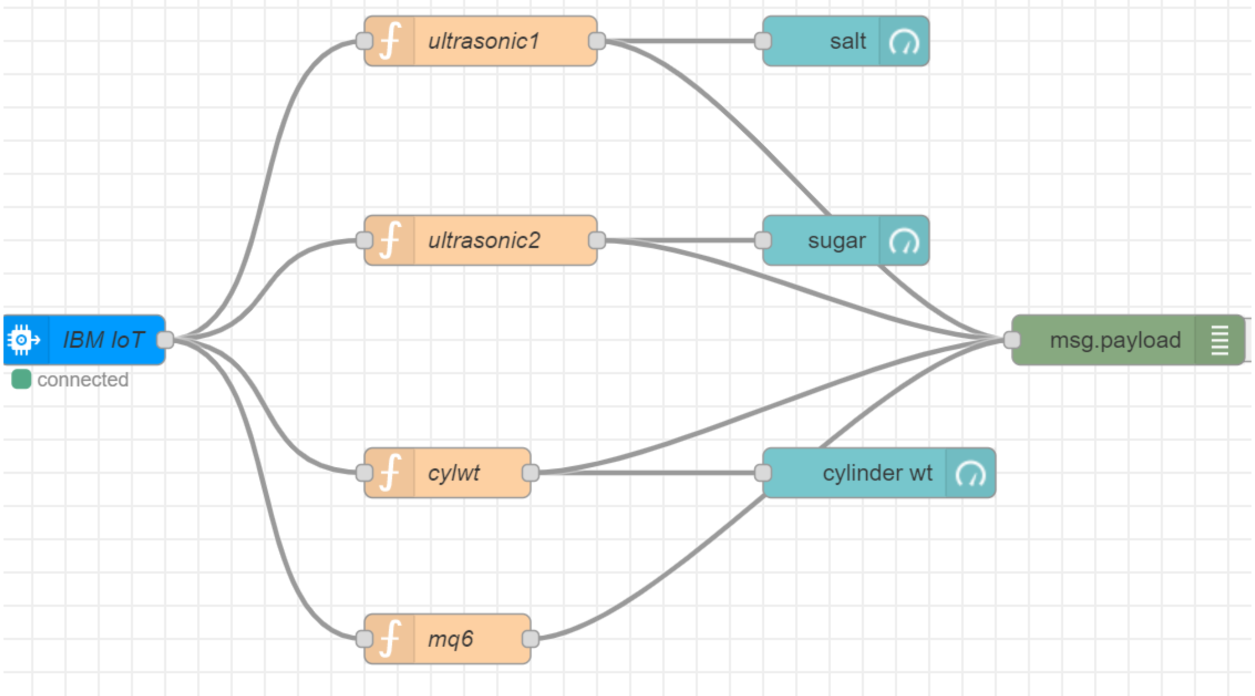
```
# Connect and send a datapoint "hello" with value "world" into the cloud as an event of type "greeting" 10 times
deviceCli.connect() #try with different values
ul1=5    # gives the level of item(salt) in container, 7 being threshold minimum
ul2 =6   # gives the level of item(sugar) in container ,7 being threshold minimum
cyl=5    # gives the weight of the cylinder 5kg being empty weight of cylinder minimum
leak="leakage"  #detect the leakage of CNG in kitchen
#enter your mobile number
if ul1<7:
    r=requests.get('https://www.fast2sms.com/dev/bulk?authorization=2lhGxE6vBDIHkQ1jAyrtcqapNRUYoMOS5dn0fsmw7F3CLeT4z8Ot6TMq8uFH
if ul2<7:
    r=requests.get('https://www.fast2sms.com/dev/bulk?authorization=2lhGxE6vBDIHkQ1jAyrtcqapNRUYoMOS5dn0fsmw7F3CLeT4z8Ot6TMq8uFH
if cyl<=5:
    r=requests.get('https://www.fast2sms.com/dev/bulk?authorization=2lhGxE6vBDIHkQ1jAyrtcqapNRUYoMOS5dn0fsmw7F3CLeT4z8Ot6TMq8uFH
if leak=='leakage':
    r=requests.get('https://www.fast2sms.com/dev/bulk?authorization=2lhGxE6vBDIHkQ1jAyrtcqapNRUYoMOS5dn0fsmw7F3CLeT4z8Ot6TMq8uFH
while True:

        data = { 'ultrasonic1' : ul1, 'ultrasonic2': ul2 , 'cylwt':cyl ,'mq6':leak}
        #print (data)
        def myOnPublishCallback():
            print ("Published ultrasonic1 = %s " % ul1, "ultrasonic2 = %s " % ul2,"cylwt = %s " % cyl,"mq6 = %s" % leak,"to IBM

        success = deviceCli.publishEvent("kitchen", "json", data, qos=0, on_publish=myOnPublishCallback)
        if not success:
            print("Not connected to IoTF")
        time.sleep(2)

        deviceCli.commandCallback = myCommandCallback
# Disconnect the device and application from the cloud
deviceCli.disconnect()
```

4. Create the nodered flow to get data from the device and http request to commuicate with the mobile app

We need to create two flows to do this task

Flow1:-

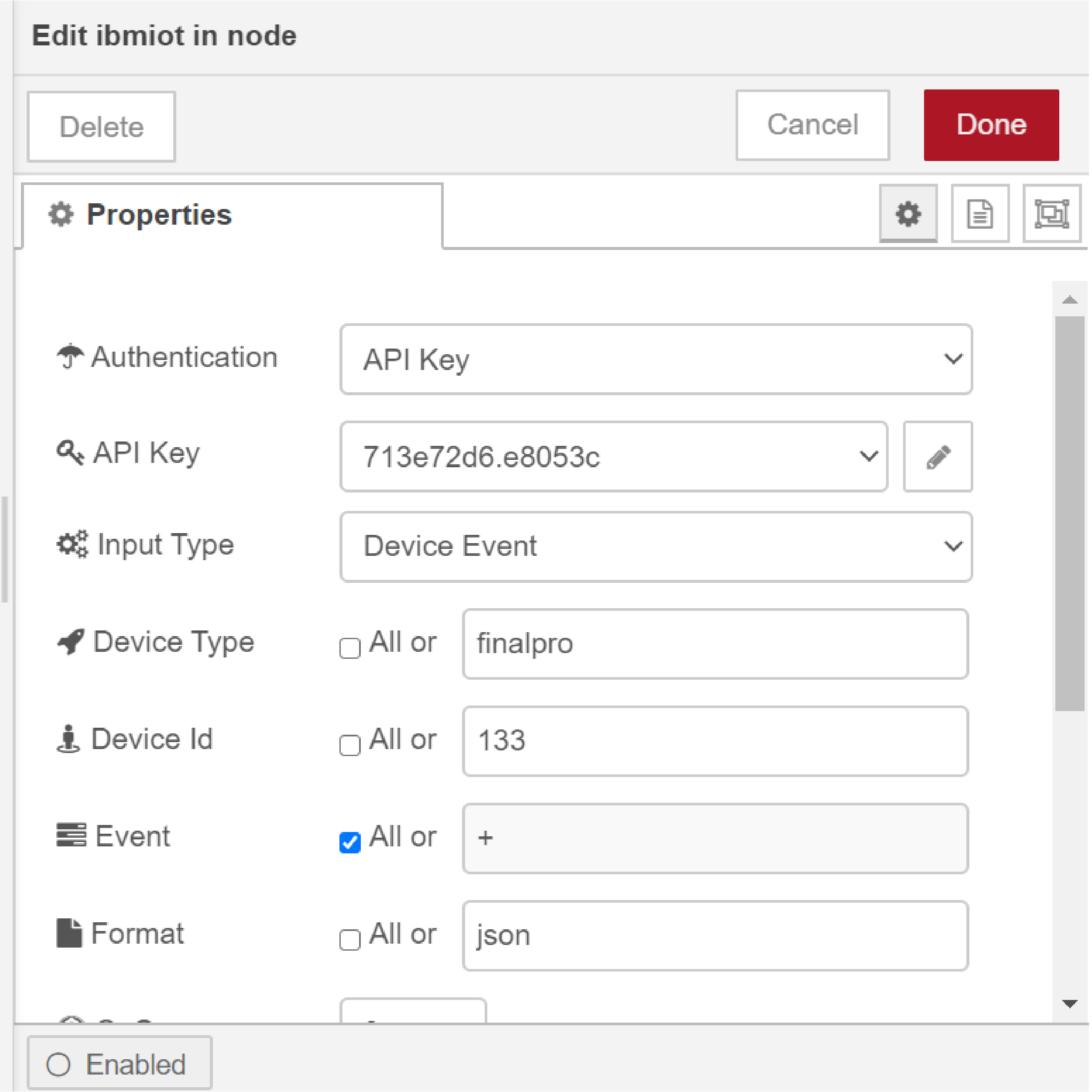
To get data from the device



To connect ibmiot device node to the device double click on the node and enter the device credentials of the device that you have

in your watson iot platform

like this



Authentication: API Key

API Key: 713e72d6.e8053c

Input Type: Device Event

Device Type: □ All or finalpro

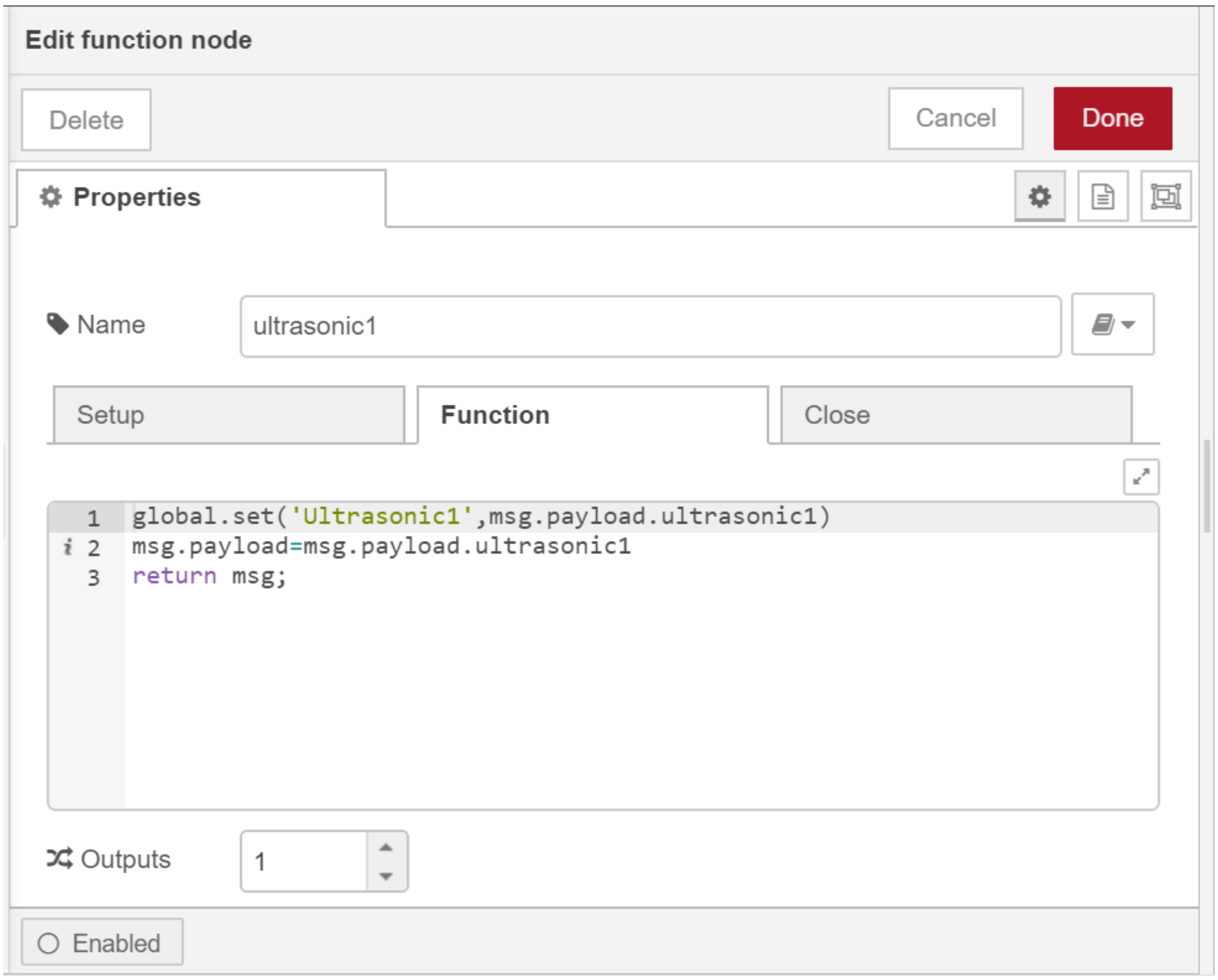
Device Id: □ All or 133

Event: ☑ All or +

Format: □ All or json

now the data that comes from the device is combined you need to parse the data and display data individually

code the function node like this



Name: ultrasonic1

```
global.set('Ultrasonic1',msg.payload.ultrasonic1)
msg.payload=msg.payload.ultrasonic1
return msg;
```

Outputs: 1

codes of all 4 function nodes

1.

```
global.set('Ultrasonic1',msg.payload.ultrasonic1)

msg.payload=msg.payload.ultrasonic1

return msg;
```

2.

```
global.set('Ultrasonic2',msg.payload.ultrasonic2)

msg.payload=msg.payload.ultrasonic2

return msg;
```

3.

```
global.set('Cylwt',msg.payload.cylwt)

msg.payload=msg.payload.cylwt

return msg;
```

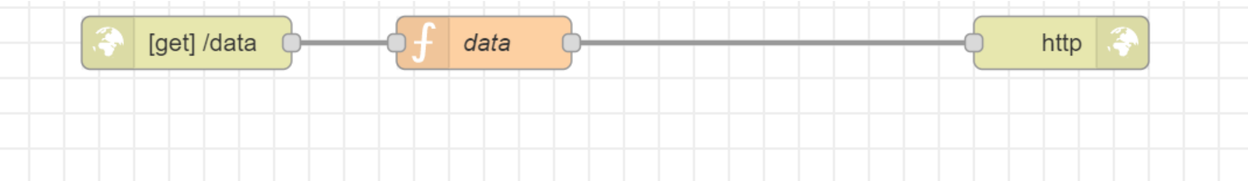
4.

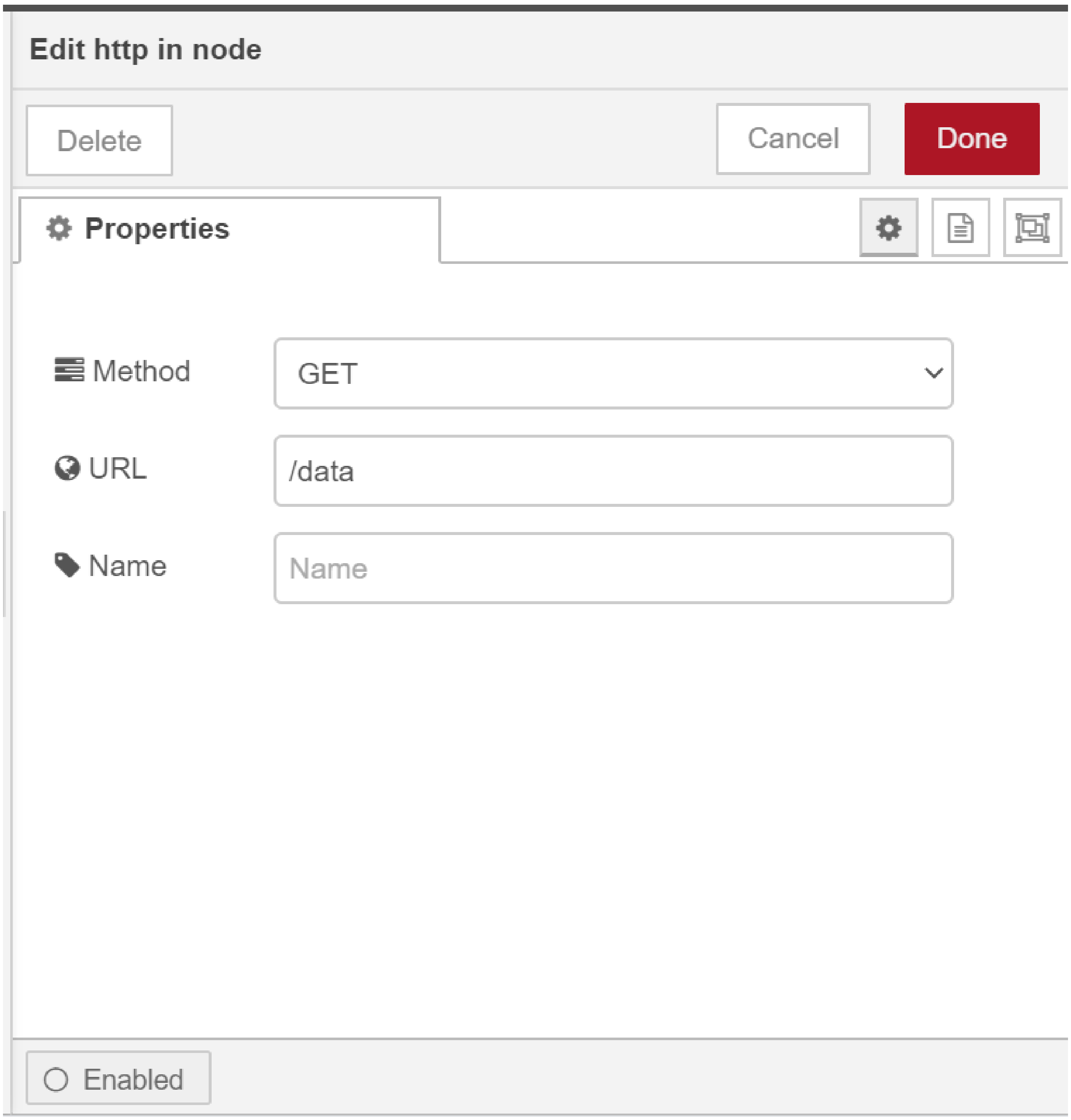
```
global.set('Mq6',msg.payload.mq6)

msg.payload=msg.payload.mq6

return msg;
```

connect those function nodes to gauges to display information on the dashboard

Flow2:-

To create http request to communicate with mobile app



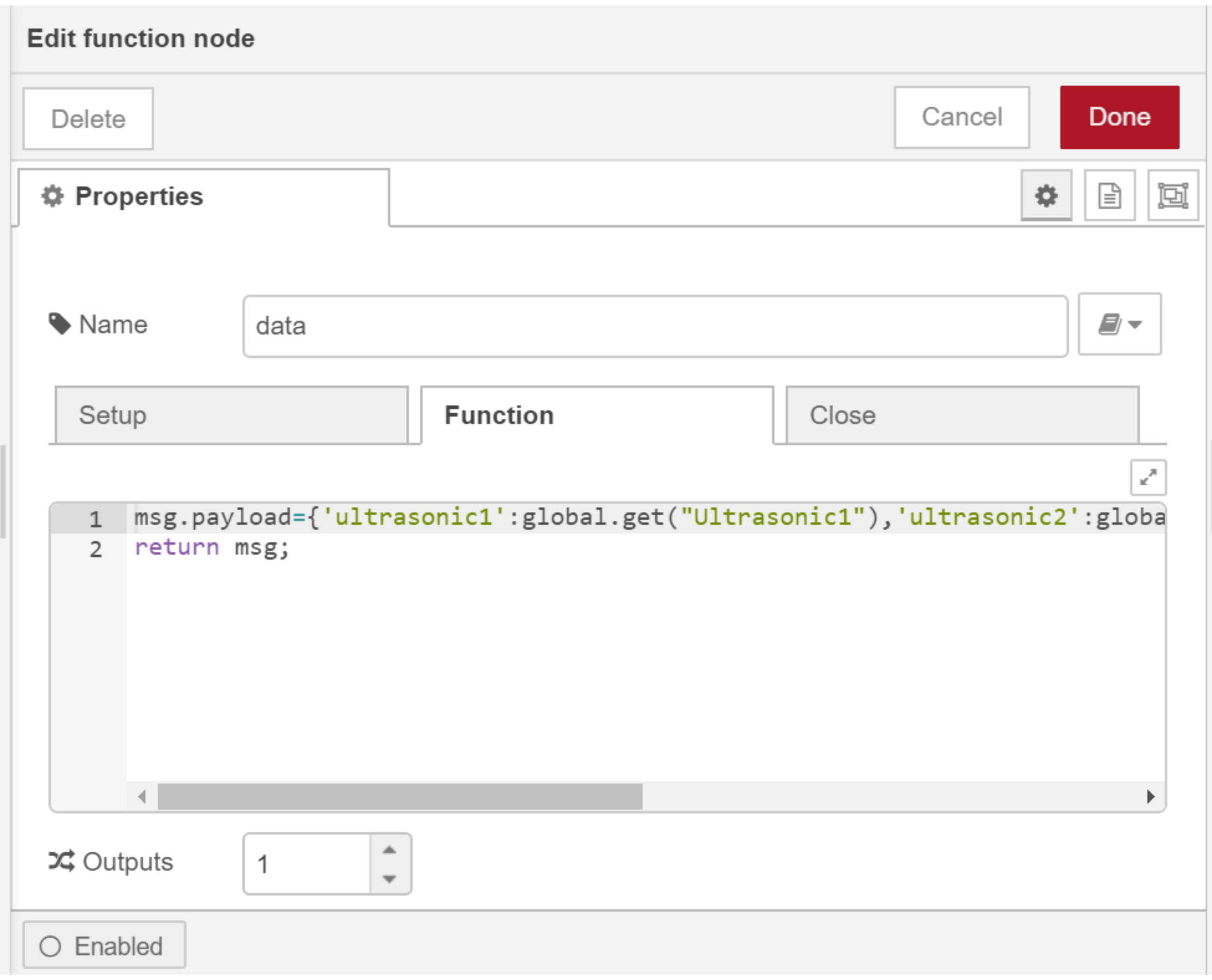


Method: GET

URL: /data

Name: Name

configure httpin node like this



Name: data

```
msg.payload={'ultrasonic1':global.get("Ultrasonic1"),'ultrasonic2':globa
return msg;
```

Outputs: 1

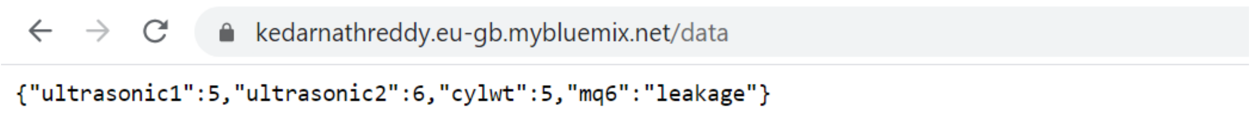
function node like this

code for the function node

```
msg.payload={'ultrasonic1':global.get("Ultrasonic1"),'ultrasonic2':global.get("Ultrasonic2"),'cylwt':global.get("Cylwt"),'mq6':global.get("Mq6")}

return msg;
```

By this flow we are sending data to the server

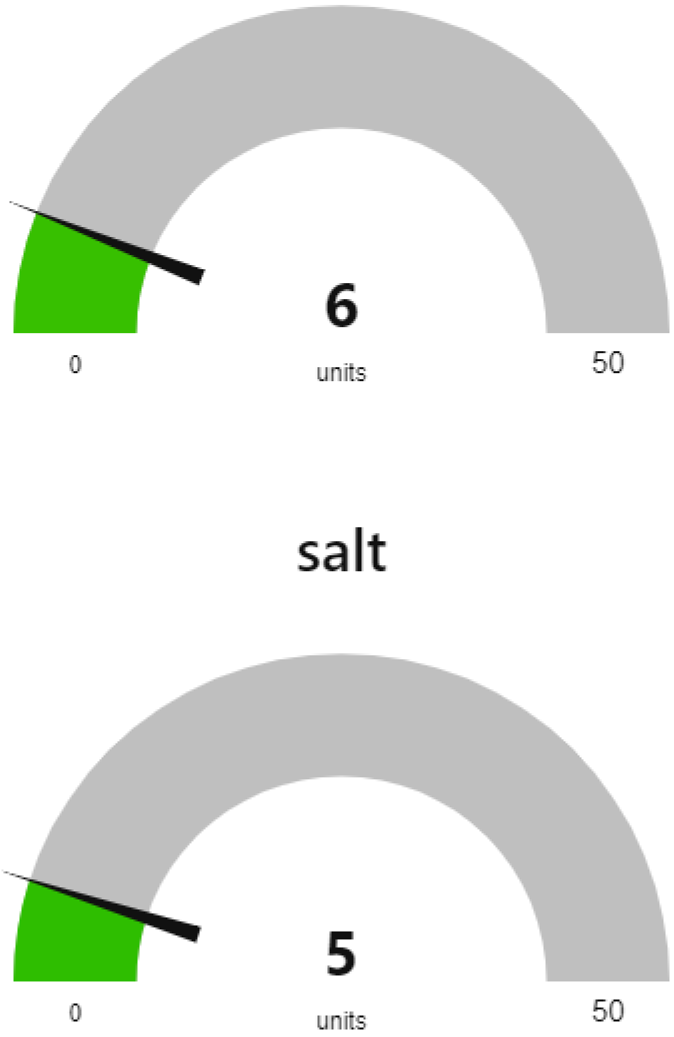
The data that has been sent to the server will be like this

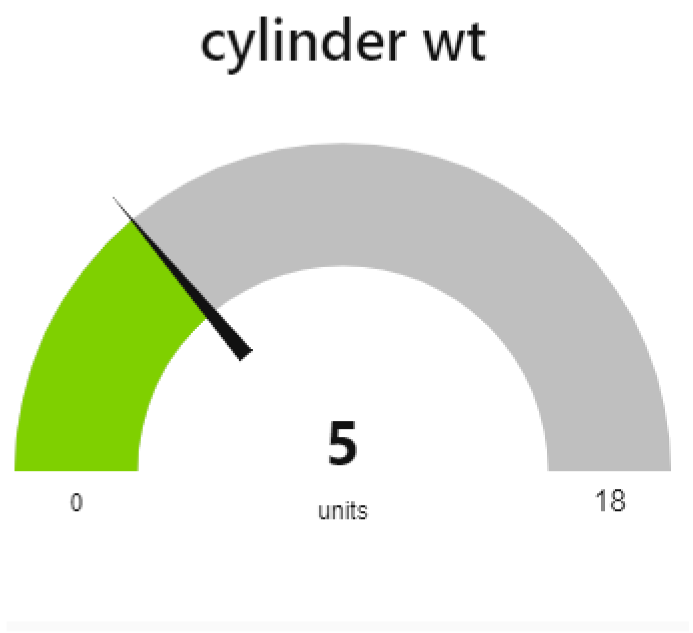


kedarnathreddy.eu-gb.mybluemix.net/data

```
{"ultrasonic1":5,"ultrasonic2":6,"cylwt":5,"mq6":"leakage"}
```

the web app ui will be like this





note:- cylwt is the weight of cylinder with 5 being empty weight

ultrasonic1 gives level of sugar and ultrasonic2 gives level of salt

in their respective jars

ultrasonic1 and ultrasonic2 are names sensors kept in jars

5. Create a mobile app using MIT APP INVENTOR and configure it to get data from the cloud

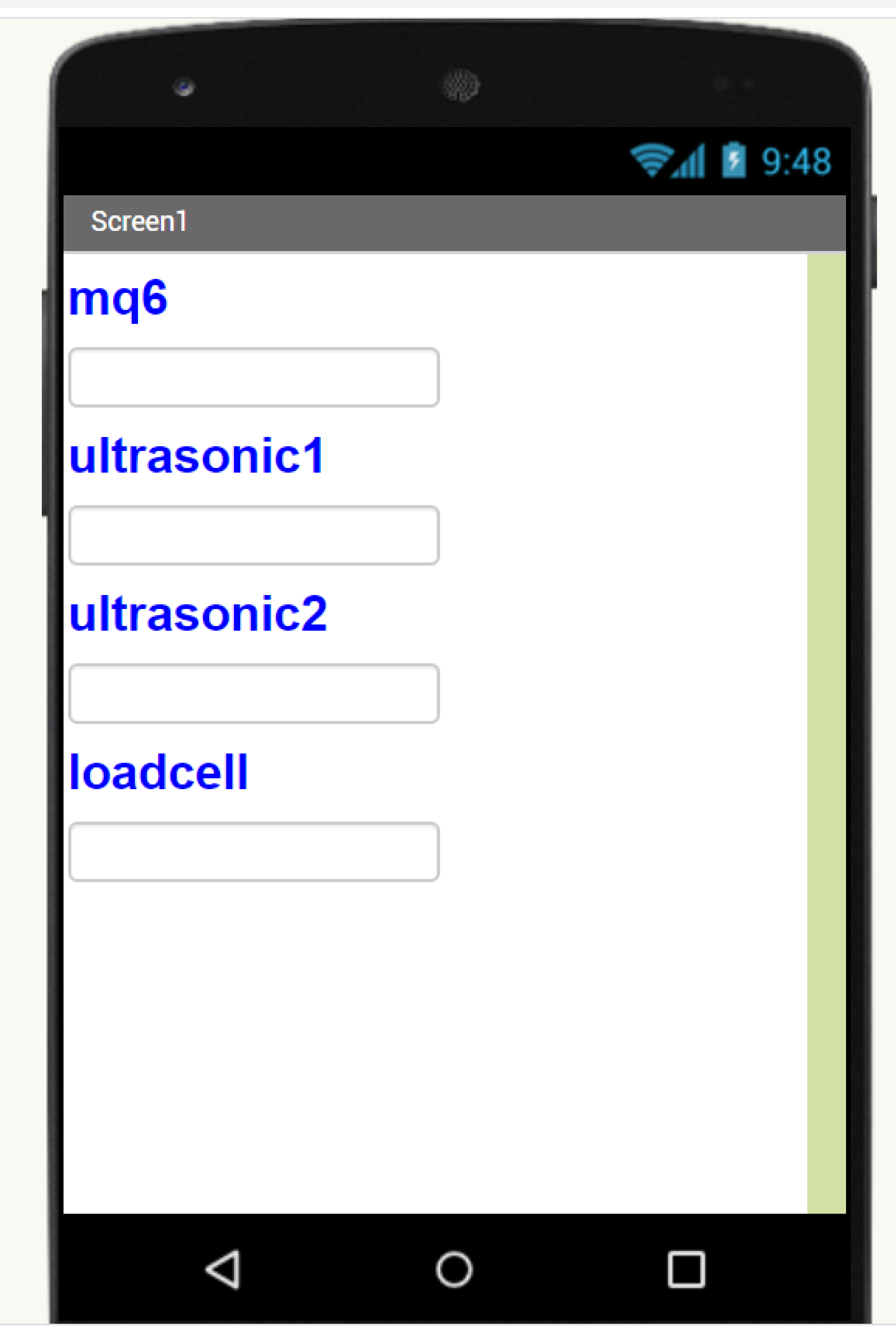
Open your browser and search 'mit app inventor' and open the website

Click on 'create apps' on the dashboard and login with your google account

Give the name of your project you should not give spaces in your project name

configure the ui of your app like this it should have 4 lables and

their respective text boxes



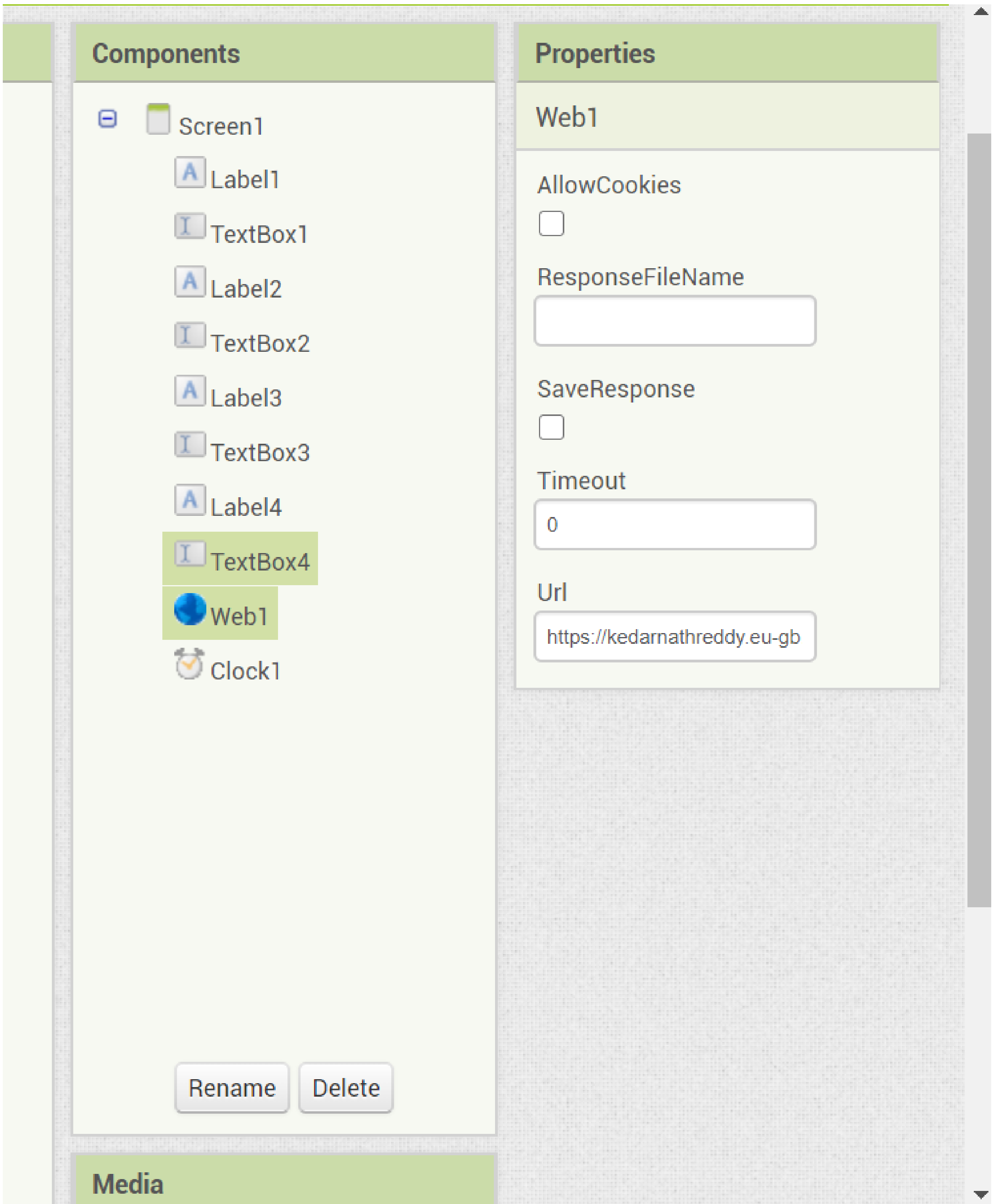
mq6 tells if there is any cng leakage in the kitchen

ultrasonic1 and ultrasonic2 gives the level of salt and sugar in the

respective jars

loadcell gives the weight of the cylinder

Now drop the web from connectivity on to the board and enter the url in the web



Web1

AllowCookies
□

ResponseFileName

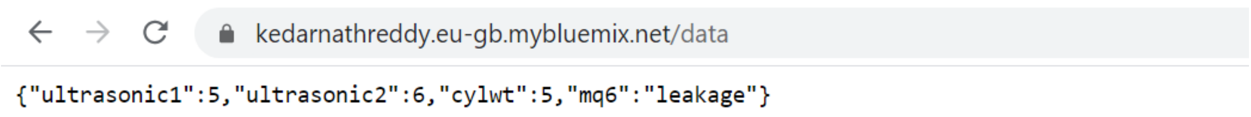
SaveResponse
□

Timeout
0

Url
https://kedarnathreddy.eu-gb

Note:-the app receives data from the url that you enter in web so you should enter the url that receives data from ibm device

Enter this url

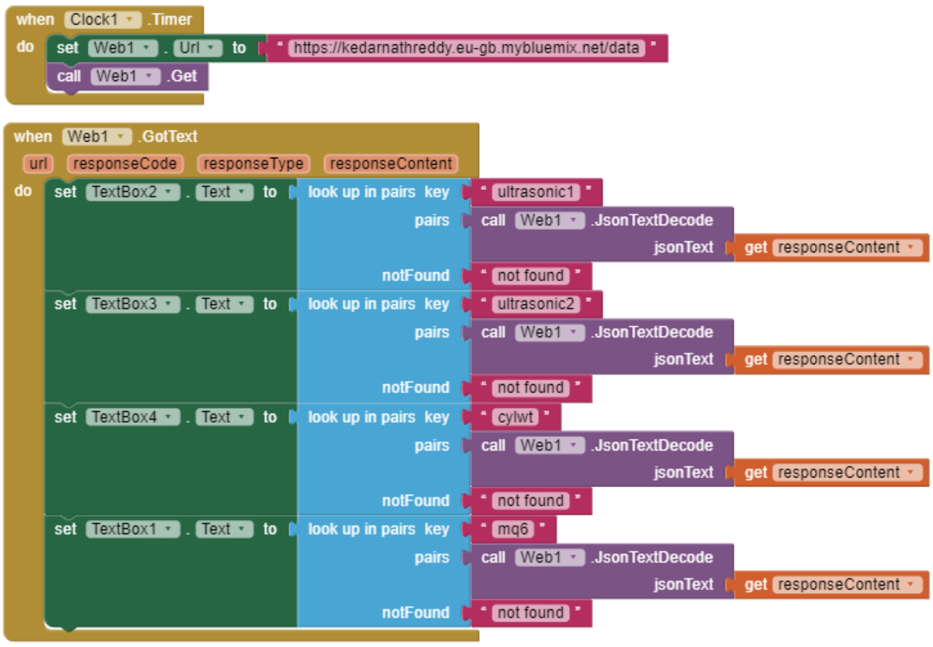


kedarnathreddy.eu-gb.mybluemix.net/data

{"ultrasonic1":5,"ultrasonic2":6,"cylwt":5,"mq6":"leakage"}

Now click on the blocks on top right corner of screen and start arrenging the blocks to create backend of the app

Set the blocks in this manner for the text boxes



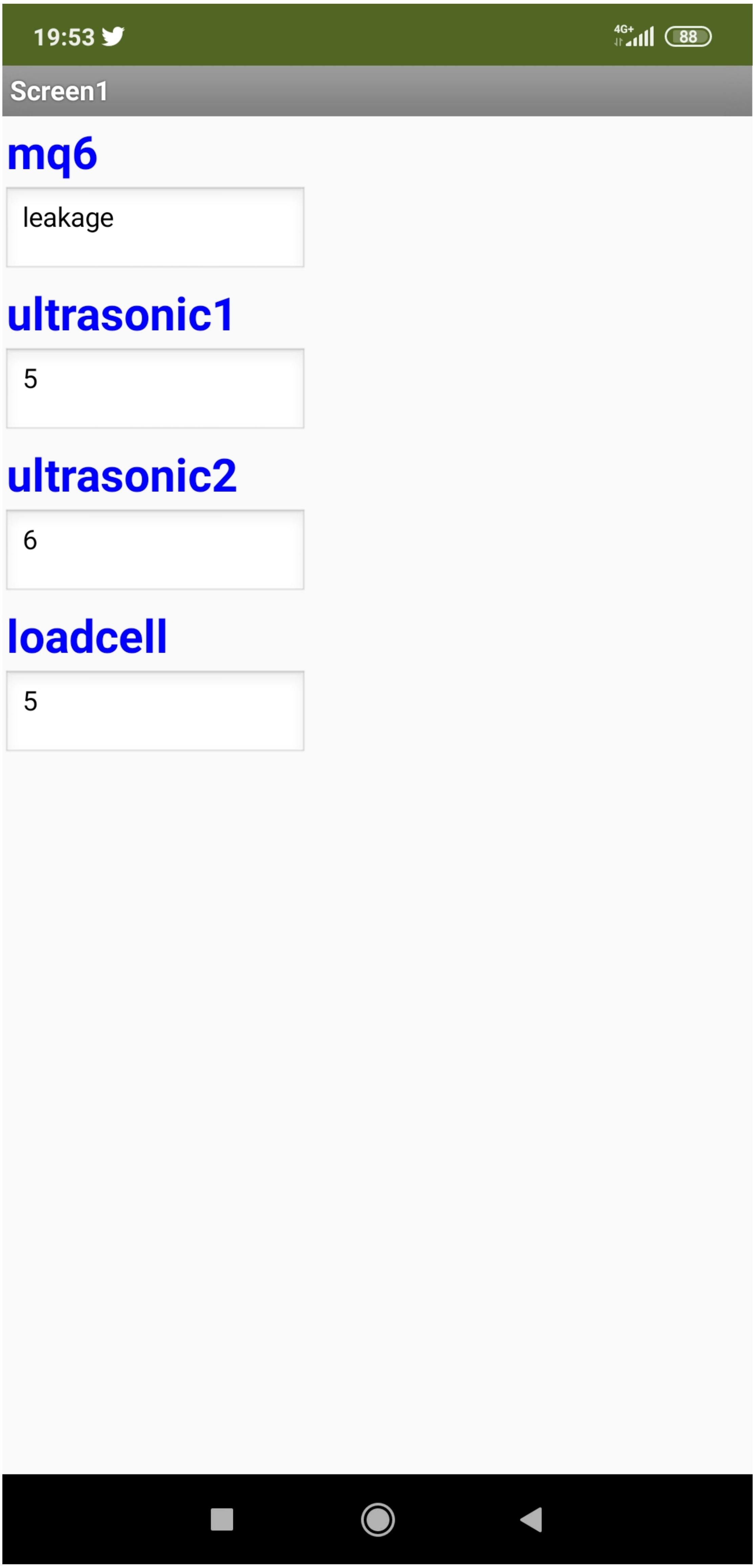
```
when Clock1 .Timer
do  set Web1 . Url to " https://kedarnathreddy.eu-gb.mybluemix.net/data "
    call Web1 .Get

when Web1 .GotText
 url  responseCode  responseType  responseContent
do  set TextBox2 . Text to  look up in pairs  key  " ultrasonic1 "
                                              pairs  call Web1 .JsonTextDecode
                                                                    jsonText  get responseContent
                                           notFound  " not found "
    set TextBox3 . Text to  look up in pairs  key  " ultrasonic2 "
                                              pairs  call Web1 .JsonTextDecode
                                                                    jsonText  get responseContent
                                           notFound  " not found "
    set TextBox4 . Text to  look up in pairs  key  " cylwt "
                                              pairs  call Web1 .JsonTextDecode
                                                                    jsonText  get responseContent
                                           notFound  " not found "
    set TextBox1 . Text to  look up in pairs  key  " mq6 "
                                              pairs  call Web1 .JsonTextDecode
                                                                    jsonText  get responseContent
                                           notFound  " not found "
```

These blocks are there to decode data that is in the form of json

and display then in their respective text boxes

Now everything is done click on build option on the top of dashboard and download the apk file, install it in your mobile

the app onened on mobile will be like this



mq6

leakage

ultrasonic1

5

ultrasonic2

6

loadcell

5

This is end of the report

THE END