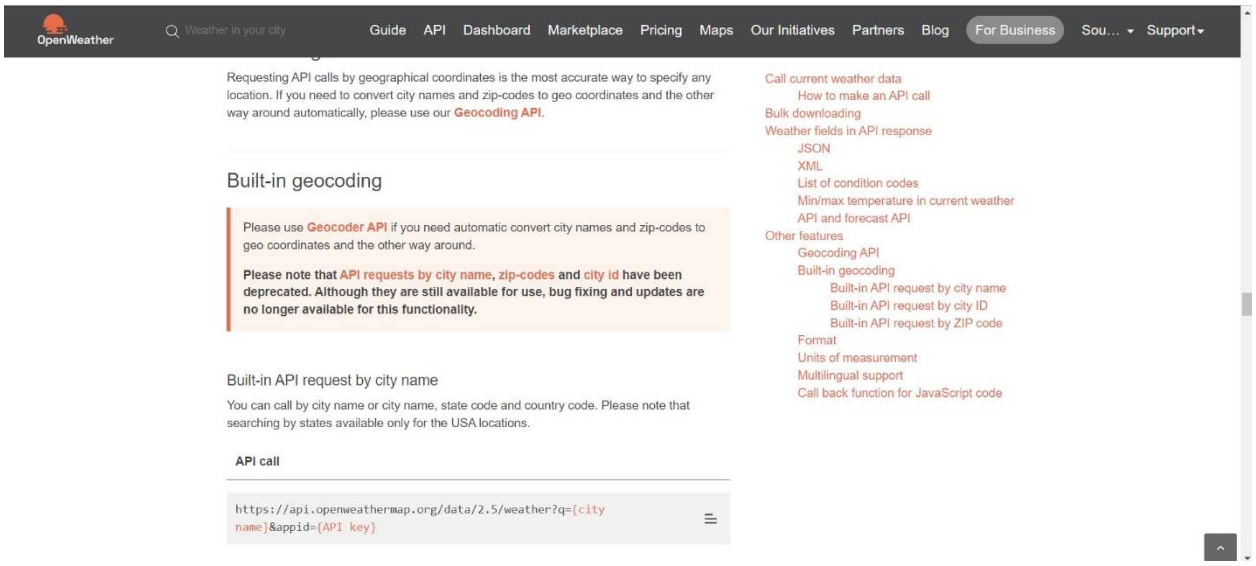
Project Development Phase

Sprint 2

Date	17 November 2022
Team ID	PNT2022TMID12941
Title	Signs with smart connectivity for better road safety

Goal: To extract data from Openweathermap website and simulate in IBM Watson IOT Platform.

Getting API from Openweathermap website:



Requesting API calls by geographical coordinates is the most accurate way to specify any location. If you need to convert city names and zip-codes to geo coordinates and the other way around automatically, please use our Geocoding API.

Built-in geocoding

Please use Geocoder API if you need automatic convert city names and zip-codes to geo coordinates and the other way around.

Please note that API requests by city name, zip-codes and city id have been deprecated. Although they are still available for use, bug fixing and updates are no longer available for this functionality.

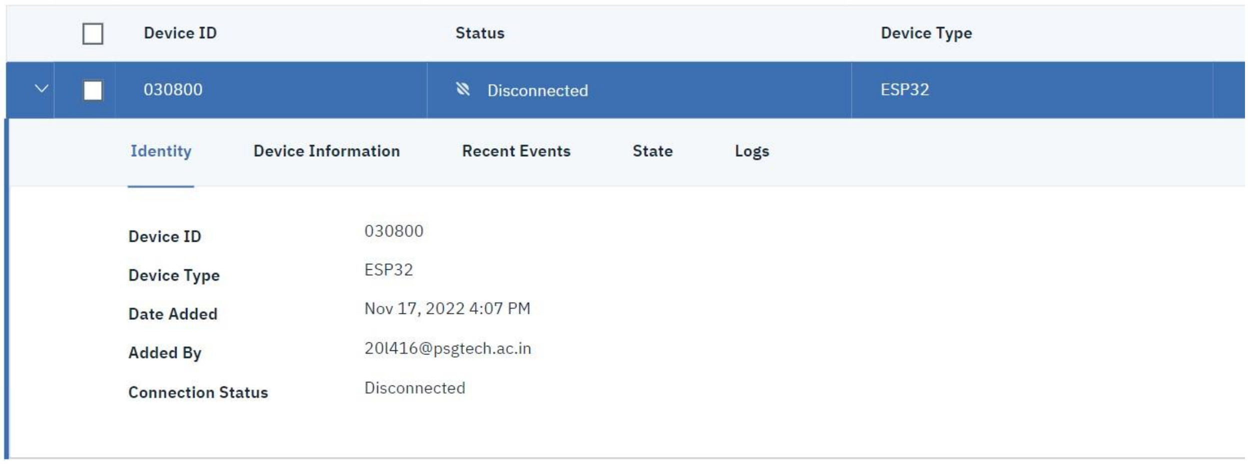
Built-in API request by city name

You can call by city name or city name, state code and country code. Please note that searching by states available only for the USA locations.

API call

```
https://api.openweathermap.org/data/2.5/weather?q={city
name}&appid={API key}
```

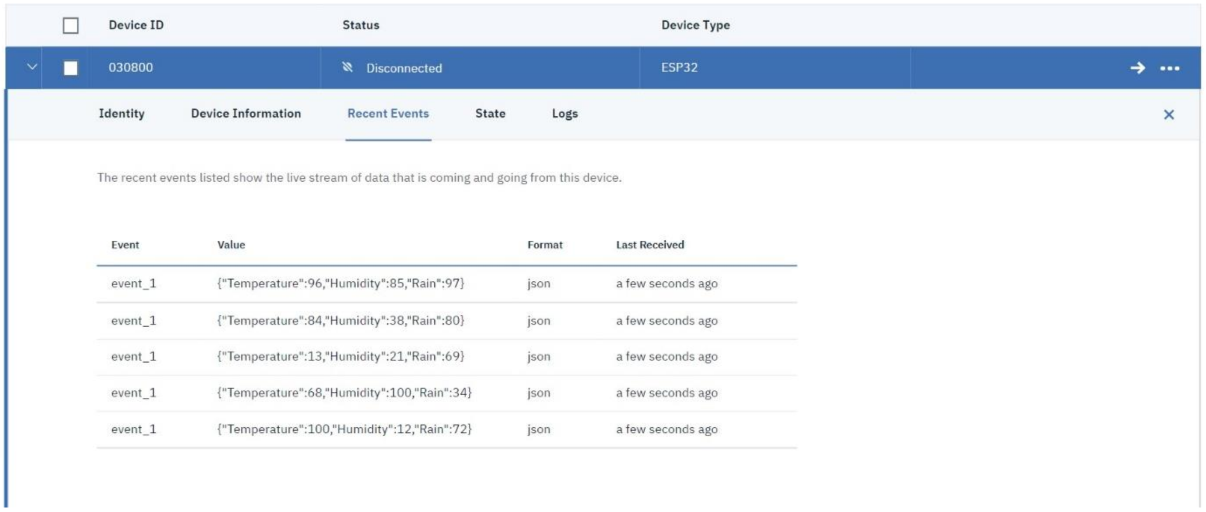
Configuring IBM Watson IOT Platform:



Device ID	Status	Device Type
030800	Disconnected	ESP32

Device ID	030800
Device Type	ESP32
Date Added	Nov 17, 2022 4:07 PM
Added By	201416@psgtech.ac.in
Connection Status	Disconnected

Output at simulation events:

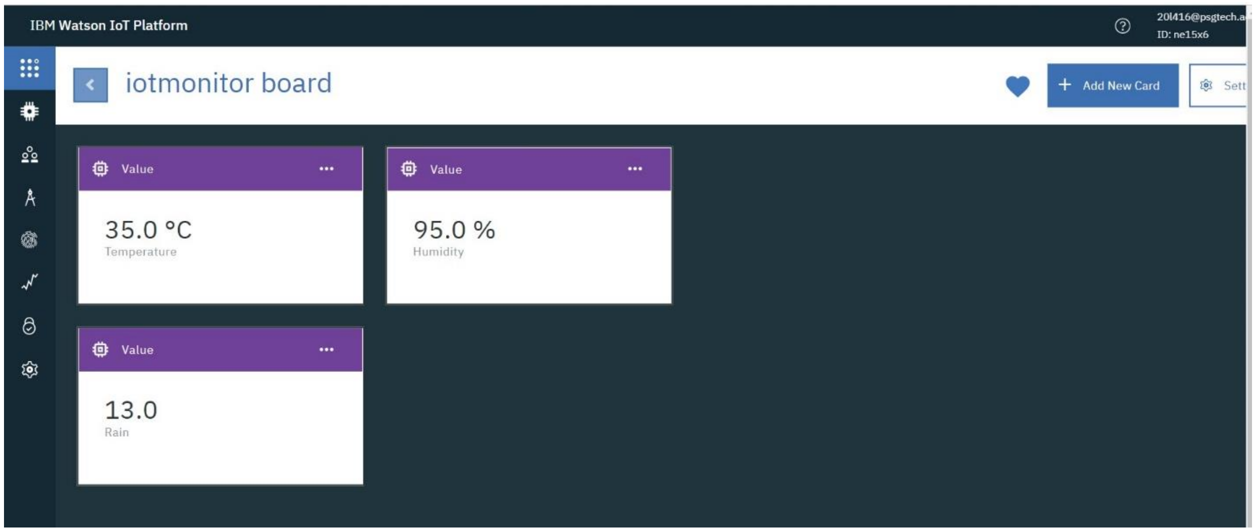


Device ID	Status	Device Type
030800	Disconnected	ESP32

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

Event	Value	Format	Last Received
event_1	{"Temperature":96,"Humidity":85,"Rain":97}	json	a few seconds ago
event_1	{"Temperature":84,"Humidity":38,"Rain":80}	json	a few seconds ago
event_1	{"Temperature":13,"Humidity":21,"Rain":69}	json	a few seconds ago
event_1	{"Temperature":68,"Humidity":100,"Rain":34}	json	a few seconds ago
event_1	{"Temperature":100,"Humidity":12,"Rain":72}	json	a few seconds ago

Output at IOT monitor board:



Python code:

i) weather.py

import requests as reqs

def get(myLocation,APIKEY):

apiURL ="https://api.openweathermap.org/data/2.5/weather?q={myLocation}&appid={APIKEY}"

responseJSON =(reqs.get(apiURL)).json()

returnObject = {

"temperature" : responseJSON['main']['temp'] - 273.15,

```
        "weather" : [responseJSON['weather'][_]['main'].lower() for _ in
range(len(responseJSON['weather']))],

        "visibility" : responseJSON['visibility']/100, # visibility in percentage where 10km is 100%
and 0km is 0%

        }

    if("rain" in responseJSON):

        returnObject["rain"] = [responseJSON["rain"][key] for key in responseJSON["rain"]]

        return(returnObject)
```

ii) brain.py

```
import weather

from datetime import datetime as dt

# IMPORT SECTION ENDS

#

# UTILITY LOGIC SECTION STARTS

def processConditions(myLocation,APIKEY,localityInfo):

    weatherData = weather.get(myLocation,APIKEY)

    finalSpeed = localityInfo["usualSpeedLimit"]if "rain" not in weatherData else
localityInfo["usualSpeedLimit"]/2

    finalSpeed = finalSpeed if weatherData["visibility"]>35 else finalSpeed/2

    if(localityInfo["hospitalsNearby"]):

        # hospital zone

        doNotHonk = True

    else:

        if(localityInfo["schools"]["schoolZone"]==False):

            # neither school nor hospital zone

            doNotHonk = False

        else:

                # school zone

                now = [dt.now().hour,dt.now().minute]
                activeTime = [list(map(int,_.split(":"))) for _ in localityInfo["schools"]["activeTime"]]
                doNotHonk = activeTime[0][0]<=now[0]<=activeTime [1][0] and
activeTime[0][1]<=now[1]<=activeTime[1][1]
                return({
                    "speed" : finalSpeed,
                    "doNotHonk" : doNotHonk
                })
```

iii) main.py

```
# IMPORT SECTION STARTS
import brain
# IMPORT SECTION ENDS
# -----------------------------------------------
# USER INPUT SECTION STARTS
myLocation = "Coimbatore,IN"
APIKEY = "bf4a8d480ee05c00952bf65b78ae826b"
# USER INPUT SECTION ENDS
# -----------------------------------------------
# MICRO-CONTROLLER CODE STARTS
print(brain.processConditions(myLocation,APIKEY,localityInfo))
'''
MICRO CONTROLLER CODE WILL BE ADDED IN SPRINT 2 AS PER OUR PLANNED
SPRINT
SCHEDULE
'''
```