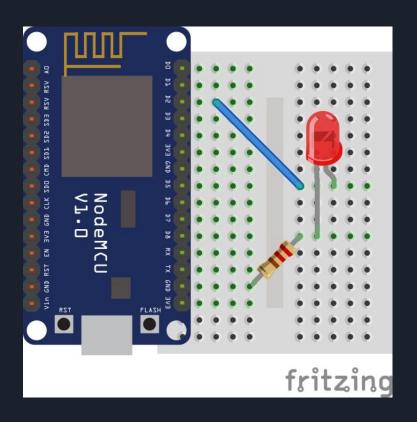
IOT Lab Assignment 5

Flask restful api

Wire in another led a different color (you will need 2 leds)



Flask (restful api)



A RESTful API is an application program interface (API) that uses HTTP requests to GET,
 PUT, POST and DELETE data.

sudo pip install flask flask-restful

Flask example

```
from flask import Flask
from flask restful import Resource, Api
app = Flask(name)
api = Api(app)
class HelloWorld(Resource):
    def get(self):
       return {'hello': 'world'}
api.add resource(HelloWorld, '/')
if name == ' main ':
    app.run(debug=True)
```

Layout of my api (includes mqtt, influxdb)

```
import paho.mqtt.client as mqtt
    import datetime
   from influxdb import InfluxDBClient
    from flask restful import Resource, Api
   broker address="192.168.1.4" #broker address (your pis ip address)
10 > def on connect(client, userdata, flags, rc):-
   dbclient = InfluxDBClient('0.0.0.0', 8086, 'root', 'root', 'sensordata')
   client.on message = on message
   client.on connect = on connect
   app = Flask( name )
   led state={'red':0, 'green':0}
   class Led(Resource):#define api resource
       def post(self):#post method change led and update led state ■
   api.add resource(Led, '/led')#add api resource
   client.loop start()#start client
```

Ngrok (get your pi a public url for your api)

- You will need to make a free account.
- https://ngrok.com/download
- https://bin.equinox.io/c/4VmDzA7iaHb/ngrok-stable-linux-arm.zip (use the arm image not arm64)

Download & setup ngrok

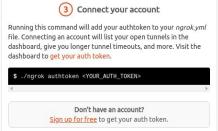
Get started with ngrok in just a few seconds.



 Mac OS X
 Windows
 Mac (32-bit)
 Windows (32-bit)
 Linux (ARM)

 Linux (ARM64)
 Linux (32-bit)
 FreeBSD (64-Bit)
 FreeBSD (32-bit)







Run Ngrok

• ./ngrok http 5000 (flask runs on port 5000)

```
Session Status
         Account
         2.2.8000000000000000000000000
Version
         Region
Web Interface
         http://127.0.0.1:4040@@@@@@@@@@@@@@@@@@@@@@@
         Forwarding
         Forwarding
                       Connections
         ttl
              rt1
                rt5
              0.00
                0.00
                  0.02
                     0.11
                       POSTORORORORORORORORORORORORORO
                 GET0000000000000000000000
              POSTORORORORORORORORORORORORO
                 GET00000000000000000000000
              GET000000000000000000000000
GET00000000000000000000000
              POSTORORORORORORORORORORORORORORO
```

Curl (how to call your api)

```
[dpivonka@dhcp-41-5 arduino-1.8.6]$ curl -X GET http://d356eb65.ngrok.io/led
   "green": 0,
   "red": 0
[dpivonka@dhcp-41-5 arduino-1.8.6]$ curl -d '{"led":"green", "state":"on"}' -X POST http://d356eb65.ngrok.io/led
   "green": 1,
   "red": 0
[dpivonka@dhcp-41-5 arduino-1.8.6]$ curl -d '{"led":"red", "state":"on"}' -X POST http://d356eb65.ngrok.io/led
   "green": 1,
   "red": 1
[dpivonka@dhcp-41-5 arduino-1.8.6]$ curl -d '{"led":"green", "state":"off"}' -X POST http://d356eb65.ngrok.io/led
   "green": 0,
   "red": 1
[dpivonka@dhcp-41-5 arduino-1.8.6]$ curl -d '{"led":"red", "state":"off"}' -X POST http://d356eb65.ngrok.io/led
   "green": 0,
   "red": 0
[dpivonka@dhcp-41-5 arduino-1.8.6]$ curl -X GET <u>http://d356eb65.ngrok.io/led</u>
   "green": 0,
   "red": 0
```

Assignment

- Add second led to breadboard
- Setup arduino to receive messages to control both leds separately
- Install flask and ngrok
- Create restful api to control both leds
- One post endpoint that receives a json containing the led you want to change and the state you want to change it to
- One get endpoint that returns the led states

Due by next class email me a video of yours dpivonka@redhat.com

Assignment video