IoTManagement System

IoTManagement System

Table of Contents

About

This repo contains

- Backend
- RPiClient Software
- Client auto-Installer script
- Detailed instructions

for IoTManagement System.

Getting Started

These instructions will get you a copy of the project up and running on you raspberry pi.

Prerequisites

Turn on your Raspberry Pi and execute the following commands

```
- sudo apt update
- sudo apt upgrade
```

RPiClient Installation

Pre-configured Image

1. Download Raspberry Pi iamge with RPiClient-rs pre-configured and flash it to your Raspberry Pi.
2. ssh into the Raspberry Pi and execute the following command to get the MAC Address:

```
sudo systemctl status RPiClient-rs.service
```

 Logs

Server Details

Monitoring

- pm2 list
- pm2 monit

List of Packages installed on server

- Mosquitto Broker
- NodeJS, NPM, Node, NVM
- PM2
- ufw
- mongod
- mongo-express

Version Details

- Node v12.16.1
- NPM v6.13.4

Server Links

- MQTT Broker Link: 44.195.192.158:1883
- Backend Link: 44.195.192.158:3000

Backend

- Backend is based on NodeJS and it is being managed by PM2. It starts automatically on server start.

MQTT Topic Details

Topics List

Logs

1. iotm-sys/device/logs `(all log messages are published to this topic) READ-ONLY`

Fimrware

2. iotm-sys/device/update/* (global firmware update files are sent to this topic) WRITE-ONLY
3. iotm-sys/device/update/[macaddress] (the fimrware file for specific device is sent to this topic {replace [macaddress] with the Mac address of the device without : in the address}) WRITE-ONLY
4. iotm-sys/device/firmware/all (global firmware update files are received at this topic) READ-ONLY
5. iotm-sys/device/firmware/[macaddress] (the fimrware file for specific device are received at this topic {replace [macaddress] with the Mac address of the device without : in the address}) READ-ONLY
6. iotm-sys/device/heartbeat/[macaddress] (MAC Address of the online device is sent to this topic {replace [macaddress] with the Mac address of the device without : in the address}) READ-ONLY

Device Management

6. iotm-sys/device/add (for adding a new device message format 'deviceName;macAddress;updatedAt') WRTIE-ONLY

Device OS

7. iotm-sys/device/upgrade/* (global device OS upgrade) WRITE-ONLY
8. iotm-sys/device/upgrade/[macaddress] (specific device OS upgrade, replace [macaddress] with device mac address without : chars) WRITE-ONLY
9. iotm-sys/device/osug/all (global OS upgrade instructions are received at this topic) READ-ONLY
10. iotm-sys/device/osug/[macaddress] (OS upgrade instructions for specific device are received at this topic {replace [macaddress] with the Mac address of the device without : in the address}) READ-ONLY
11. iotm-sys/device/info/[macaddress] (device and os info of specific device can be requested from this topic) WRITE-ONLY

API Details

Add Device

```
POST http://44.195.192.158:3000/v1/addDevice
```

Parameter	Type	Description
operation	string	**Required**. *value of operation should be 'add'*
name	string	**Required**. *value of param could be a name*
macAddress	string	**Required**. *value of param should be a MAC Address of your RPi Device being displayed by RPiClient Installer*

Parameter	Type	Description
updatedAt	string	**Required**. *value of param should be the current timestamp*

Upgrade OS

```
POST http://44.195.192.158:3000/v1/upgrade
```

Parameter	Type	Description
operation	string	**Required**. *value of operation should be 'upgrade'*
devices	string	**Required**. *value of devices param could be 'all' or 'device MAC Address'*

Update Firmware

```
POST http://44.195.192.158:3000/v1/update
```

Parameter	Type	Description
operation	string	**Required**. *value of operation should be 'update'*
devices	string	**Required**. *value of devices param could be 'all' or 'device MAC Address'*
programFile	multipart/form-data	**Required**. *a Firmware file to be sent to repective device(s)*

List Devices

```
GET http://44.195.192.158:3000/v1/listAll
```

Parameter	Type	Description
nothing		

Responses

Many API endpoints return the JSON representation of the resources created or edited. However, if an invalid request is submitted, or some other error occurs, Gophish returns a JSON response in the following format:

```
{
  "status"  : int,
  "message" : string
}
```

The `message` attribute contains a message commonly used to indicate errors or to return the logged status/

The `status` attribute describes if the transaction was successful or not.

Status Codes

IoTManagementSystem Backend returns the following status codes in its API:

Status Code	Description
200	`OK`
201	`CREATED`
400	`BAD REQUEST`
404	`NOT FOUND`
500	`INTERNAL SERVER ERROR`

Usage

1. Download Raspberry Pi iamge with RPiClient-rs pre-configured and flash it to your Raspberry Pi.
2. ssh into the Raspberry Pi and execute the following command to get the MAC Address:

```
sudo systemctl status RPiClient-rs.service
```

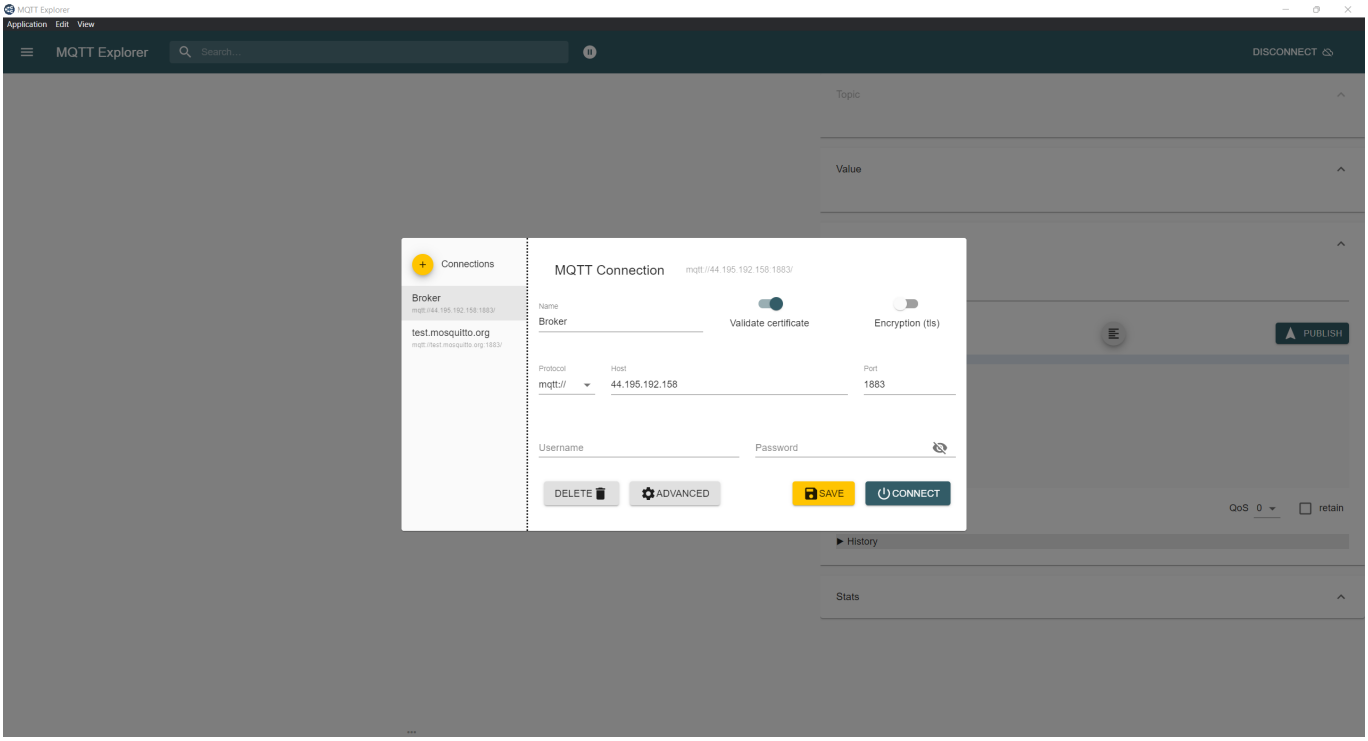
3. Add the device with the MAC Address collected in the previous step to the database using addDevice API endpoint mentioned above
4. Interact with the device with using MAC Address, or interact with all the devices in the system by using `all` in devices parameter of the API.

Test

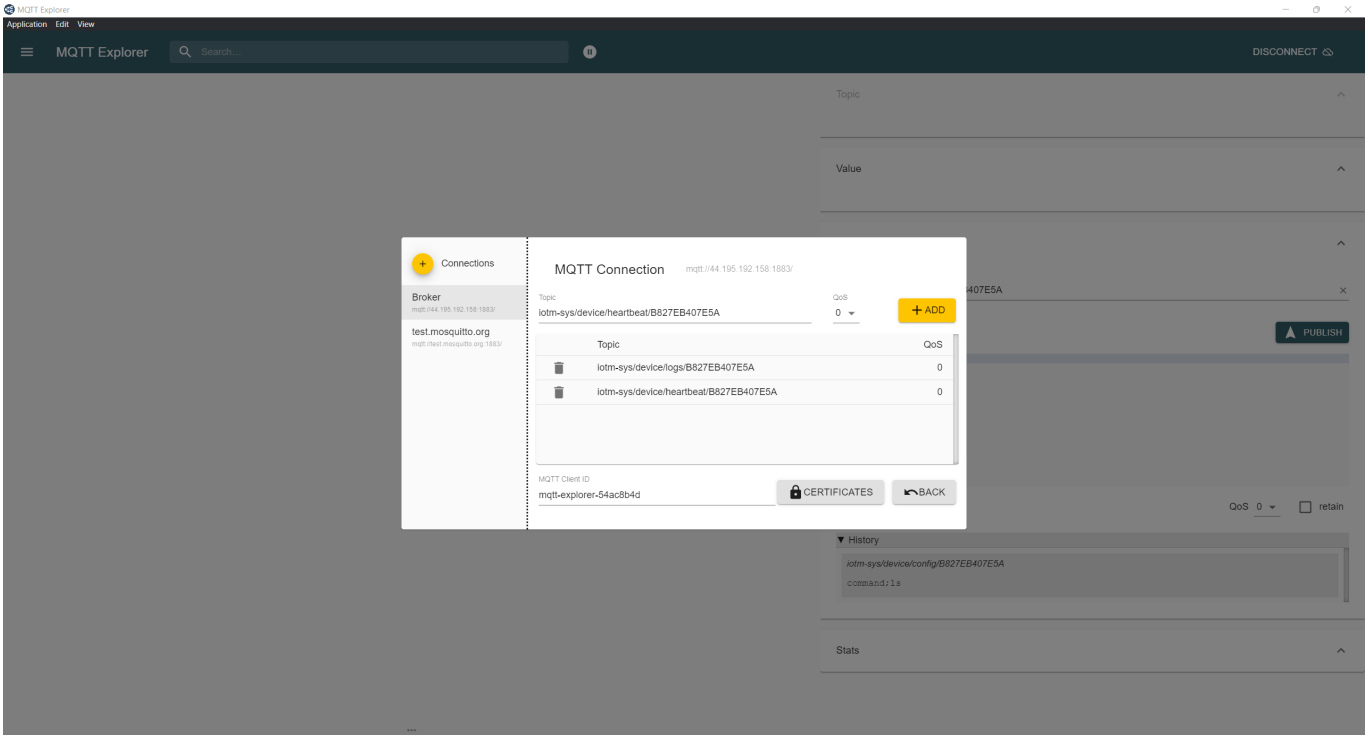
Use MQTT Explorer to test the remote communication over the internet. You can run MQTT Explorer or any computer placed anywhere on the internet.

Install and Open MQTT Explorer

1. Add a new connection with following details



2. Then add follwoing advaced options

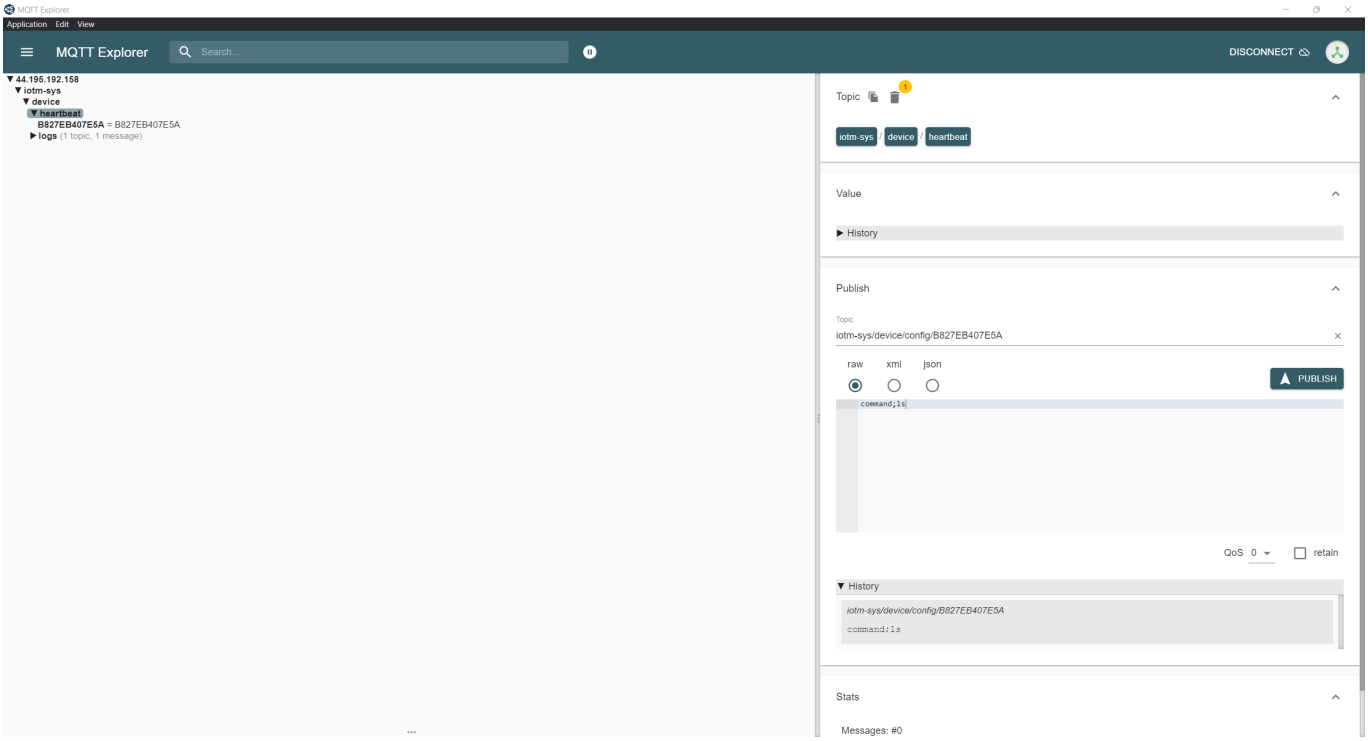


3. ssh into the Raspberry Pi and execute the following command to get the MAC Address:

```
sudo systemctl status RPiClient-rs.service
```

 Logs

4. Then you can publish to various topics and see the response on the same screen. You can get the topics list from the MQTT Topic Details section above.



Built Using

- NodeJS - JS Framework for Backend Programming
- Eclipse Paho MQTT - MQTT Client for Backend and RPiClient Software
- MongoDB - Database for Managing devices
- Rust - Systems Programming Language. For programming RPi Client

Demo Videos

- Complete Demo Part 1: https://youtu.be/d15zIwMxJ3w
- - This is a part 1 of complete demo of IoT Management System, showing how to install the Client on Raspberry Pi and run it.
- Complete Demo Part 2: https://youtu.be/kUgdPix0l-g
- - Part 2 of complete demo showing how to interact with all the devices or specific devices in the system using API.
- Demo of Rust-based RPiClient: https://www.youtube.com/watch?v=OvejznGeAbU

Authors

- @Nauman3S - Development and Deployment