**Traffic Analysis System Manual 2**

Thingsboard

<https://github.com/DRR-IGI/device_controller>

python requirement libs : paho-mqtt

**[light\_controller.py]**

**python light\_controller.py [-A device\_access\_token] [-H HOST\_NAME] [-t timeout]**

สร้าง RPC server สำหรับอุปกรณ์รองรับ 3 method

1. setLight ตั้งค่าสถานะของอุปกรณ์
2. getLight รับค่าสถานะของอุปกรณ์
3. limitLight สั่งเปิดไฟโดยให้ปิดเมื่อเวลาผ่านไป t วินาที

**example: python light\_controller.py -A LIGHT\_001 -H iot.sappasing.com -t 5**

สร้างเซิฟเวอร์สำครับอุปกรณ์ที่มี access token LIGHT\_001 บน thingsboard host iot.sappasing.com โดยเมื่อรับคำสั่ง limitLight จะตั้งค่าสถานะอุปกรณ์และปิดไฟในอีก 5 วินาที

**[run\_light0\_94.sh]**

**./run\_light0\_94.sh**

ตัวอย่าง shell script สั่งเปิด server ของอุปกรณ์ LIGHT\_000 ถึง LIGHT\_094

<https://github.com/DRR-IGI/rpc_call>

**[rpc\_call\_shell.py]**

**python rpc\_call\_shell.py [-D DEVICE\_ID] [-m rpc\_method] [-p rpc\_parameter]**

ส่ง rpc request ไปยัง server

Parameter สำหรับเปิดปิดไฟ

Yes, yes, True, true, t, y, 1 หมายถึงให้ “เปิดไฟ”

No, no, False, false, n, f, 0, -1 หมายถึงให้ “ปิดไฟ”

ตัวอย่าง **python rpc\_call\_shell.py -D 33b4a031-8660-11eb-b42e-550755035d0b -m setLight -p 0**

สั่งปิดไฟที่อุปกรณ์ device\_id 33b4a031-8660-11eb-b42e-550755035d0b

<https://github.com/DRR-IGI/light_distance_API>

python requirement libs: fastapi, uvicorn, psycopg2, pandas, numpy

เวอชั่นที่ใช้ตอนสร้าง (uvicorn==0.13.4, fastapi==0.63.0)

**[server.py]**

**python server.py**

สร้าง api server สำหรับคำนวนระยะห่างของอุปกรณ์ สามารถเข้าไปแก้ port ได้ใน server.py

เมื่อรันแล้วจะสามารถเข้าหน้า /docs ได้จะเป็น API manual ให้ลองใช้

**/api/device\_nearby** หาอุปกรณ์ที่ใกล้เคียงจากอุปกรณ์ที่กำหนด

Parameter:

src\_name ชื่ออุปกรณ์ตั้งต้น

src\_type ประเภทของอุปกรณ์ตั้งต้น

find\_type ประเภทของอุปกรณ์ที่จะค้นหา

dist ระยะห่างมากสุดที่สนใจ

Response:

{“device” : [ { "name ": " … ", " device\_id ": " … " ], […] }

Example :

Request url = <http://172.23.0.87:8888/api/device_nearby?src_name=cctv_7&src_type=cctv&find_type=light&dist=50>

Shell command : **curl -X 'GET' \**

**'http://172.23.0.87:8888/api/device\_nearby?src\_name=cctv\_7&src\_type=cctv&find\_type=light&dist=50' \**

**-H 'accept: application/json'**

**Response:**

{

"device": [

{

"name": "light\_42",

"device\_id": "33da50a0-8660-11eb-b42e-550755035d0b"

},

{

"name": "light\_43",

"device\_id": "33da77b0-8660-11eb-b42e-550755035d0b"

},

{

"name": "light\_44",

"device\_id": "33dcc1a0-8660-11eb-b42e-550755035d0b"

},

{

"name": "light\_45",

"device\_id": "33dd5de1-8660-11eb-b42e-550755035d0b"

}

]

}

**/api/device\_nearby\_gps** เช่นเดียวกับ **/api/device\_nearby** แต่จะหาจาก Latitude และ Longitude แทน

Parameter:

lat Latitude

lng Longitude

find\_type ประเภทของอุปกรณ์ที่จะค้นหา

dist ระยะห่างมากสุดที่สนใจ