



KMUTT
สถาบันเทคโนโลยี
พระจอมเกล้า
เจ้าคุณทหารลาดกระบัง

KMITL
FICHT

iCTE
KMUTT
IoT System &
Information
Engineering



01236255 Introduction to Internet of Things

Lab 1 : Cloud IoT Platform with NETPIE

พศ.ดร.รนวิชญ์ อนุวงศ์พิบูจ

หลักสูตรวิศวกรรมระบบไอโอทีและสารสนเทศ ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง



Final Section

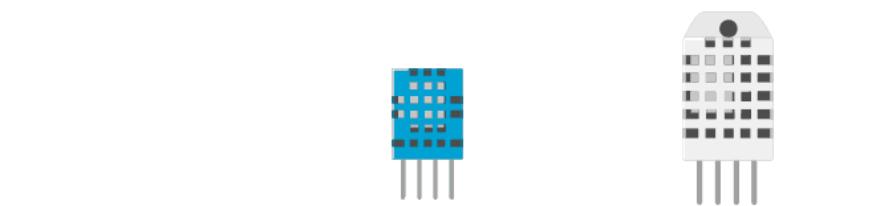
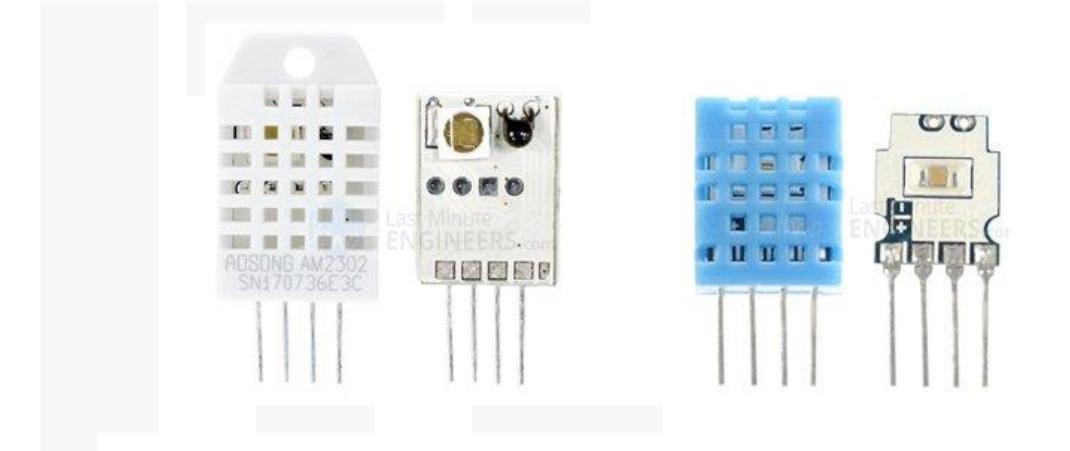
- Lecture
 - IoT Architecture, Hardware & Device
 - Basic IoT Networks & Communications
 - IoT Data Processing
- Lab
 - DHT11 Temp&Humi Sensor
 - NETPIE IoT Cloud Platform
 - Basic IoT Network Programming (TCP/UDP/HTTP)
 - Node-Red Programming
 - HTTP & API
 - MQTT
 - Dashboard, Notify, Data Storage
- Project

DHT11

Temperature & Humidity Sensor

DHT11

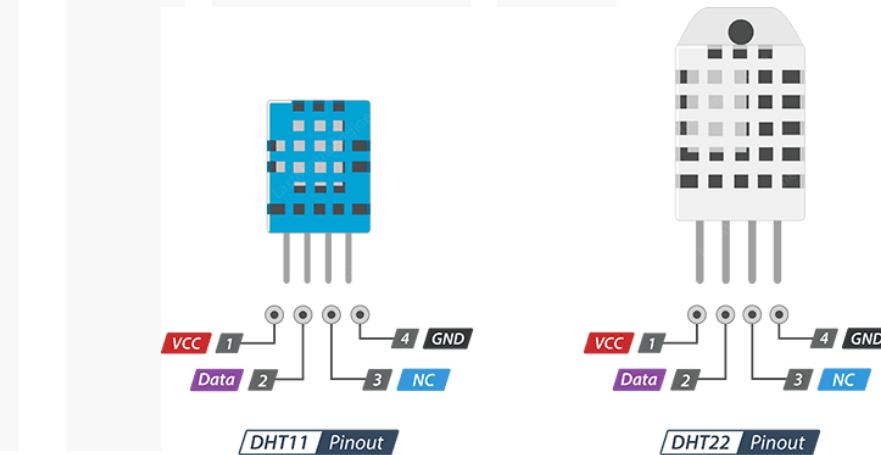
- DHT11 / 22 เป็นเซ็นเซอร์ที่ใช้วัดอุณหภูมิ และวัดความชื้นได้อีกด้วย มี labore รีพร้อมใช้งานกับ Arduino มีราคาถูก และสามารถใช้วัดค่าได้เที่ยงตรงกว่ามาก เพราะให้ Output ออกมาในรูปแบบของดิจิตอล ใช้วัดอุณหภูมิอากาศโดยรอบ ปัจจุบันเป็นเซ็นเซอร์ซึ่งเป็นที่นิยมใช้เนื่องจากสามารถใช้งานได้ง่าย และวัดค่าออกมาได้อย่างมีประสิทธิภาพ ซึ่งเซ็นเซอร์นี้จะมีสองประเภท DHT11 และ DHT22 ภายใต้เป็น Resistive Humidity Sensor (สำหรับวัดความชื้นในอากาศ) และ Thermistor (สำหรับวัดอุณหภูมิ)



	DHT11	DHT22
Operating Voltage	3 to 5V	3 to 5V
Max Operating Current	2.5mA max	2.5mA max
Humidity Range	20-80% / 5%	0-100% / 2-5%
Temperature Range	0-50°C / ± 2°C	-40 to 80°C / ± 0.5°C
Sampling Rate	1 Hz (reading every second)	0.5 Hz (reading every 2 seconds)
Body size	15.5mm x 12mm x 5.5mm	15.1mm x 25mm x 7.7mm
Advantage	Ultra low cost	More Accurate

DHT11

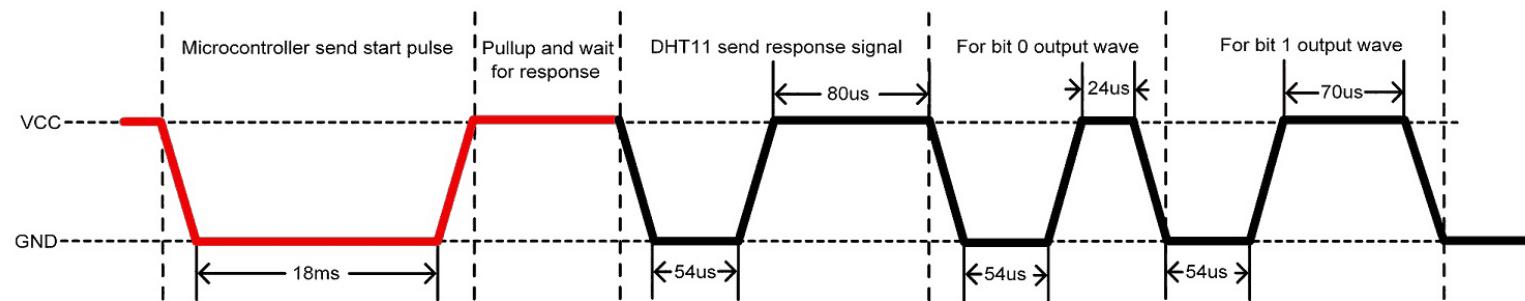
- ในการเชื่อมต่อ กับบอร์ดทดลองนั้นจะใช้เพียง 1 สาย (one wire) ในการสื่อสารเชื่อมต่อ ค่าต่าง ๆ หรือ ระดับแรงดันจะเปลี่ยนแปลงไปตามค่าลอจิก 0 หรือ 1 ที่ส่งมากับสาย โดยในการสื่อสารนั้นจะมีทั้งหมดสามขั้นตอนคือ จะส่ง request ไปยัง DHT11 จากนั้น เซ็นเซอร์จะส่ง response pulse กลับมา และเริ่มส่งข้อมูล 40 บิต ไปให้กับไมโครคอนโทรลเลอร์



Last Minute
ENGINEERS.com

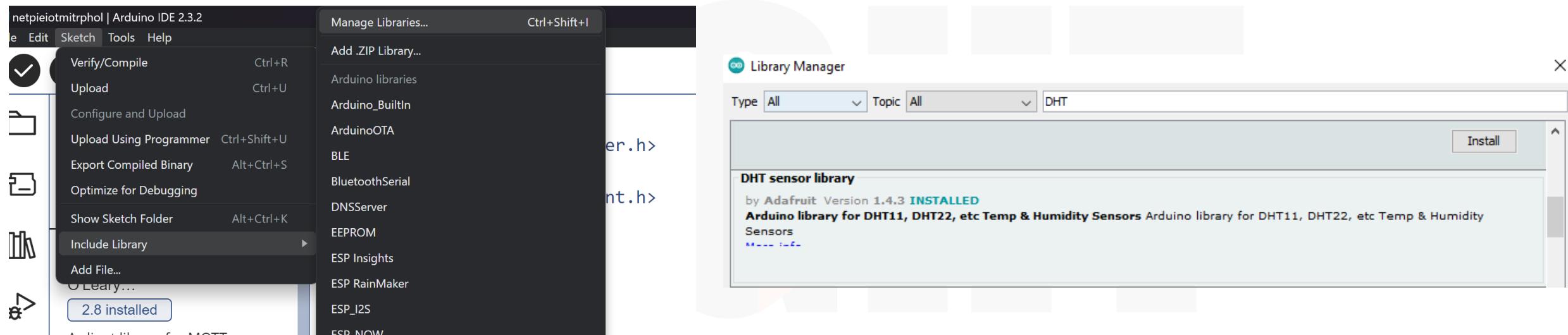
RH = Decimal of 00100101 = 37%

Temperature = Decimal of 00011001 = 25°C



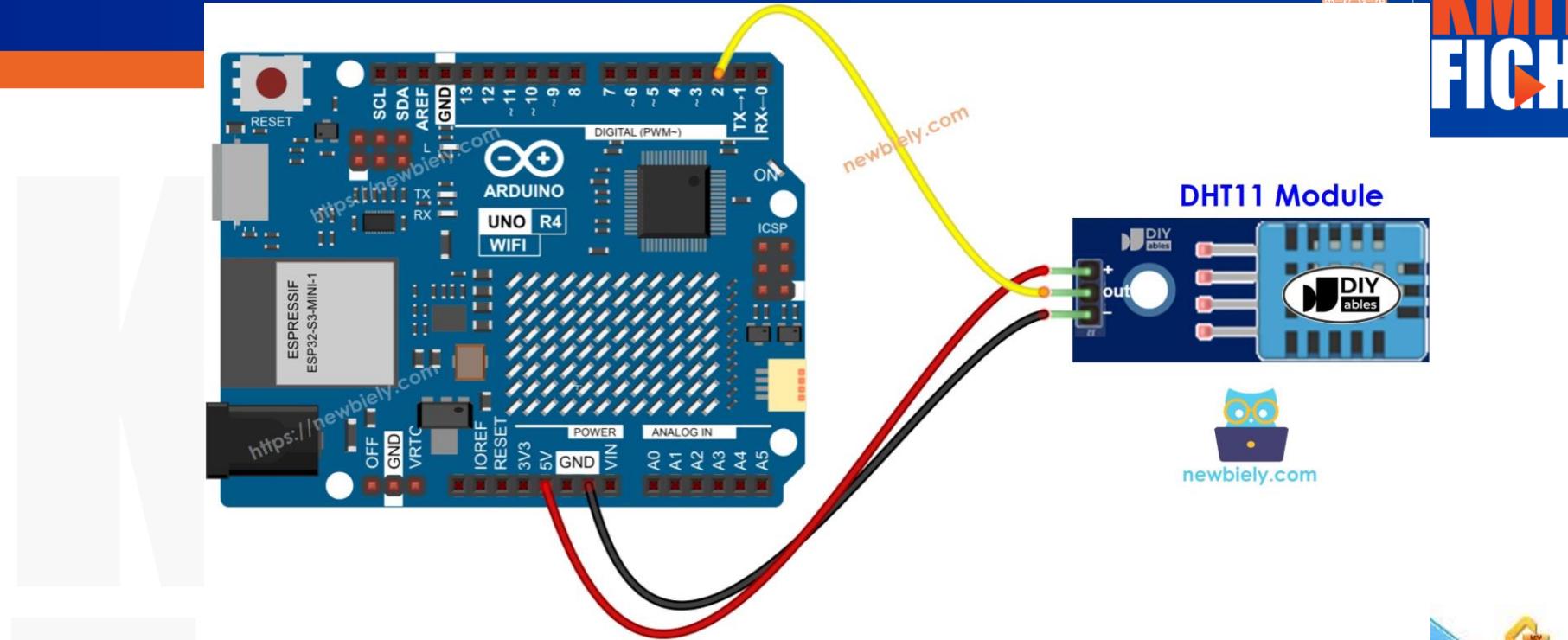
DHT11

- เพื่อให้สะดวกในการใช้งาน ทำการดาวน์โหลด Search จาก Arduino IDE โดยที่เมนูบาร์ด้านบนไปที่ Sketch -> Include Libraries -> Manage Libraries
- ในหน้าต่างที่ปรากฏให้ค้นหา DHT sensor library จากนั้นกดปุ่ม Install โดยจะมีการถามให้ติดตั้ง Lib เพิ่มเติมให้เลือก “Install All” รอนติดตั้งเสร็จ (จนขึ้นตัวอักษรสีเขียวว่า Installed)





DHT11



การเชื่อมต่อระหว่าง Arduino UNO R4 และ Sensor เชื่อมต่อตามตารางด้านล่างนี้

DHT11/DHT22	Arduino UNO R4
VCC หรือ +	5V
DATA หรือ OUT	2
GND หรือ -	GND

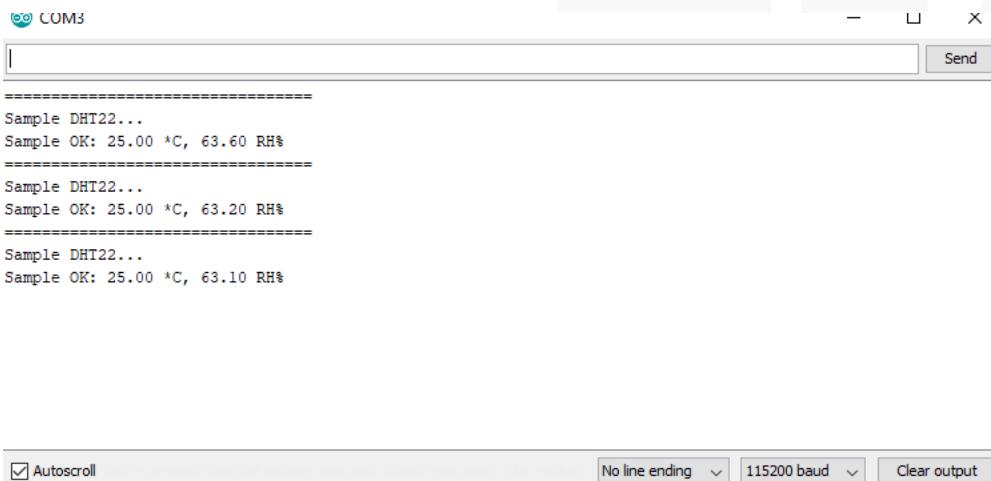
** ถ้าตัวที่มี 4 ขา ขาที่สาม (นับจากหันหน้าอุปกรณ์เข้าเรา) จะไม่ได้ใช้

*** ทั้งนี้ บอร์ด DHT11 อาจมีการเชื่อมต่อขาหลายรูปแบบ ให้ดูสัญลักษณ์ที่ขาเป็นหลัก อย่าดูตามว่าจะเป็นหลัก



DHT11

- จนนี้อัพโหลดโค้ดลงบอร์ด
- เปิด Serial Monitor ว่ามีแสดงผลค่าขึ้นมาหรือไม่



```

#include "DHT.h"
#define DHTPIN 2
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);

void setup() {
  Serial.begin(9600);
  Serial.println(F("DHTxx test!"));
  dht.begin();
}

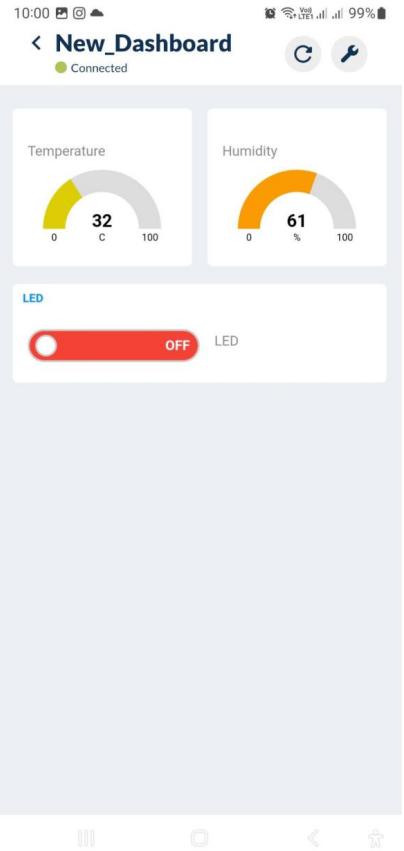
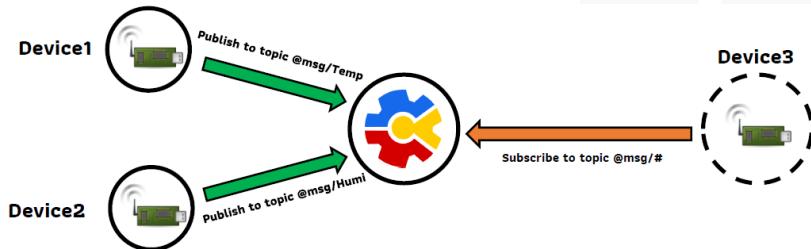
void loop() {
  delay(2000);
  float h = dht.readHumidity(); // Read temperature as Celsius (the default)
  float t = dht.readTemperature(); // Read temperature as Fahrenheit (isFahrenheit = true)
  float f = dht.readTemperature(true);

  // Check if any reads failed and exit early (to try again).
  if (isnan(h) || isnan(t) || isnan(f)) {
    Serial.println(F("Failed to read from DHT sensor!"));
    return;
  }

  Serial.print(F("Humidity: "));
  Serial.print(h);
  Serial.print(F("% Temperature: "));
  Serial.print(t);
  Serial.print(F("°C "));
  Serial.print(f);
}
  
```

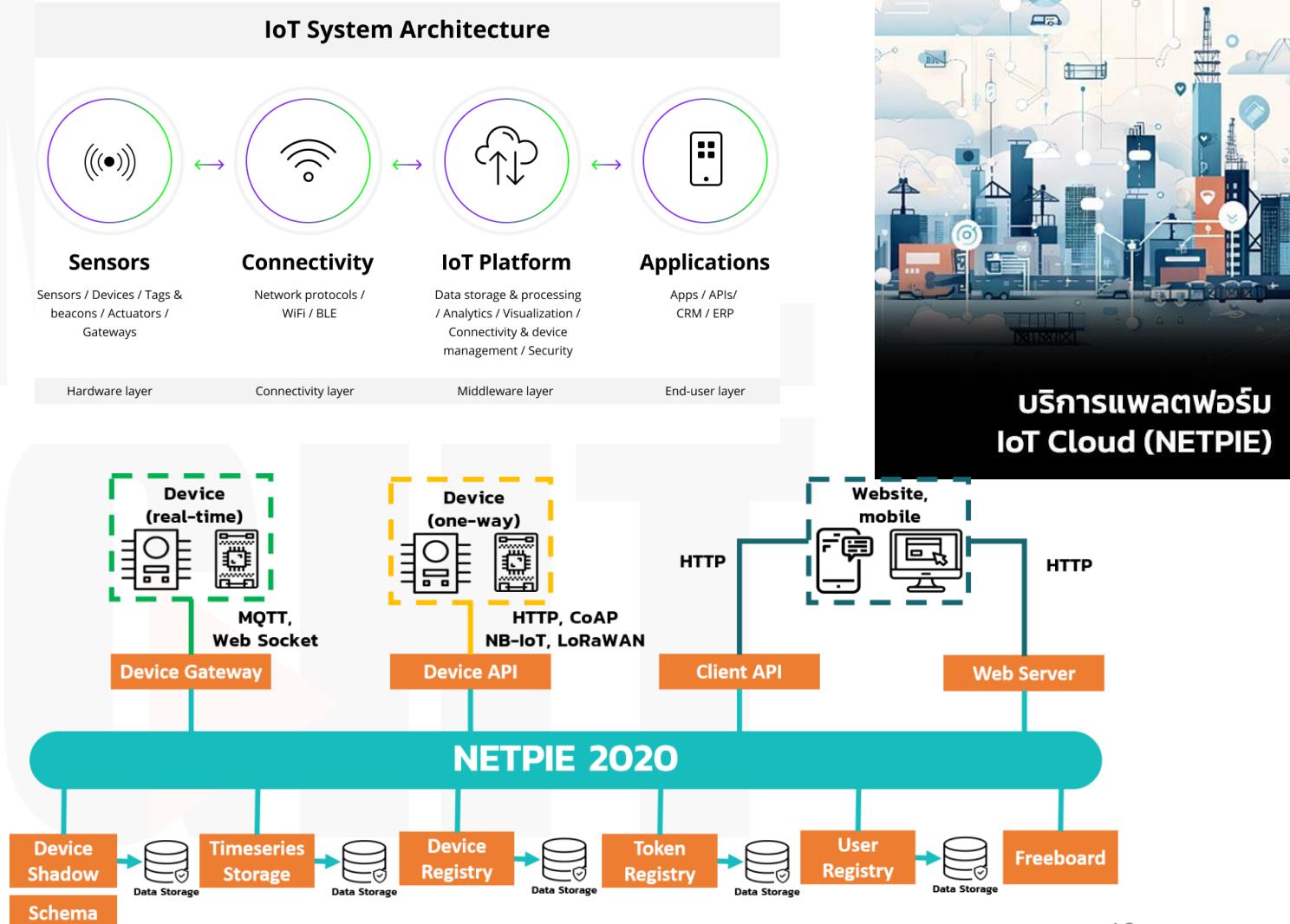


NETPIE 2020



NETPIE

- NETPIE คือ IoT Cloud Platform ที่เปิดให้บุคคลทั่วไปใช้งานได้ โดยแพลตฟอร์มจะช่วยให้อุปกรณ์ต่างๆ สามารถสื่อสารกันได้ เกิดการรับ-ส่งข้อมูลระหว่างอุปกรณ์แบบ real-time ทำให้ผู้ใช้งานทราบถึงข้อมูลของอุปกรณ์ ณ เวลานั้นๆ ไม่ว่าผู้ใช้งานจะอยู่ที่ไหนเวลาใดก็ตาม



NETPIE

- NETPIE คือ IoT Cloud Platform ที่เปิดให้บุคคลทั่วไปใช้งานได้ โดยแพลตฟอร์มจะช่วยให้อุปกรณ์ต่างๆ สามารถสื่อสารกันได้ เกิดการรับ-ส่งข้อมูลระหว่างอุปกรณ์แบบ real-time ทำให้ผู้ใช้งานทราบถึงข้อมูลของอุปกรณ์ ณ เวลานั้นๆ ไม่ว่าผู้ใช้งานจะอยู่ที่ไหนเวลาใดก็ตาม

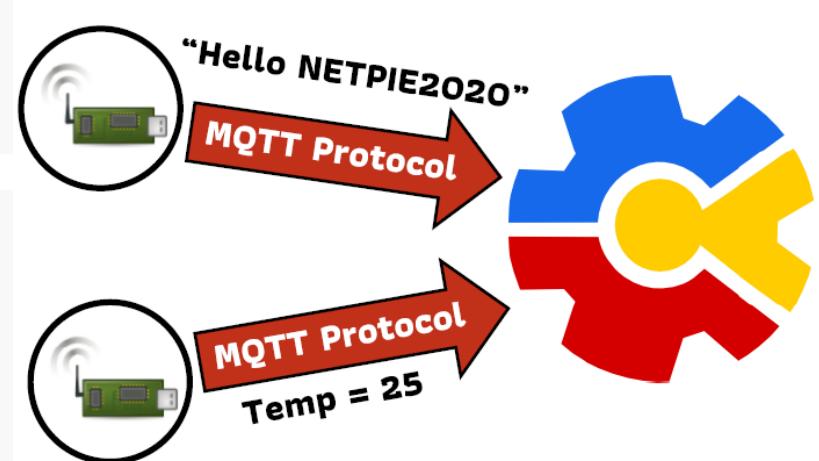


- 1. Monitoring คือ การแสดงค่าข้อมูลของอุปกรณ์หรือเซนเซอร์แบบ Real-Time
- 2. Controlling คือ การควบคุมการทำงานอุปกรณ์ต่างๆ ผ่าน Cloud Platform
- 3. Data Storage คือ การเก็บข้อมูลที่ได้จากเซนเซอร์หรืออุปกรณ์
- 4. Notification คือ การแจ้งเตือนความผิดปกติของเซนเซอร์หรืออุปกรณ์จากที่ได้กำหนดไว้

NETPIE

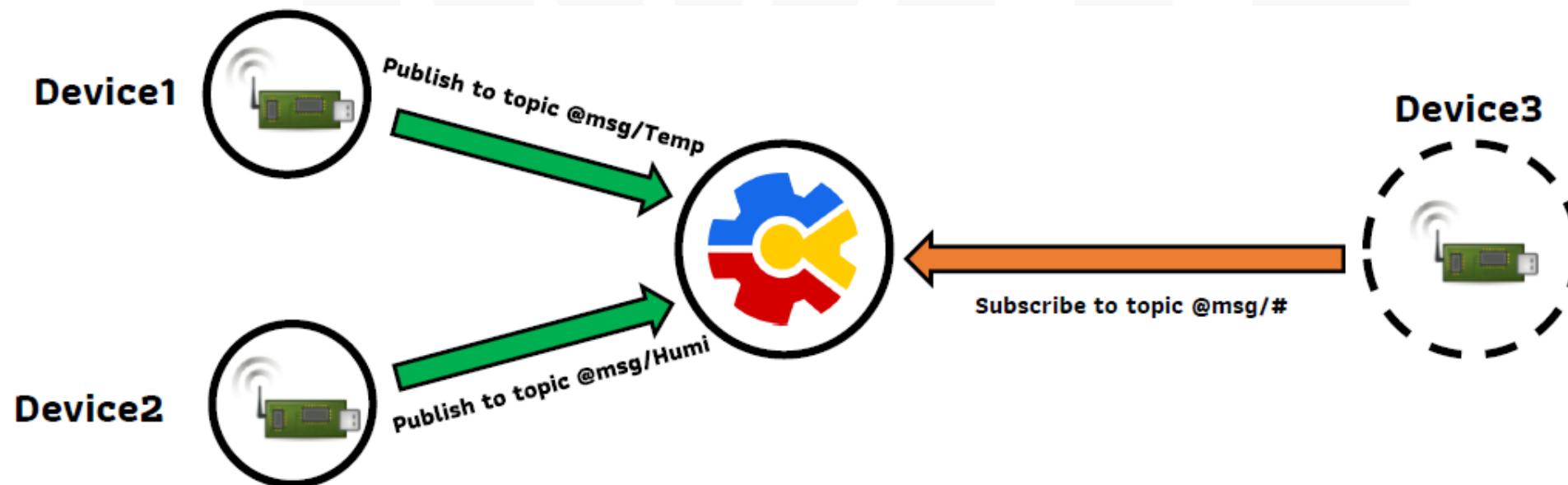
- ข้อความที่ส่งไปยัง NETPIE2020 ผ่าน MQTT แบ่งเป็น 2 ชนิด
 - แบบ Message เป็นการส่งข้อความผ่าน MQTT โดย อ้างอิง Topic ซึ่งรูปแบบ topic คือ @msg/topic
 - แบบ Realtime Data (SHADOW) เป็นการส่งข้อความ ผ่าน MQTT ไปยัง Device Shadow เพื่อบันทึกข้อความ ล่าสุด มีรูปแบบ topic คือ @shadow/data/update ซึ่ง รูปแบบข้อความเป็น JSON ตัวอย่างเช่น

```
{'data': {'temperature': 24, 'humidity': 56}}
```



NETPIE

- Topic แบบ Wildcard : สามารถใช้สัญลักษณ์ # เป็นส่วนหนึ่งใน topic ซึ่งจะตรงกับค่าทุกๆ ทาง / ของค่า
- ถ้า Subscribe topic /# จะเทียบเท่ากับการ Subscribe All



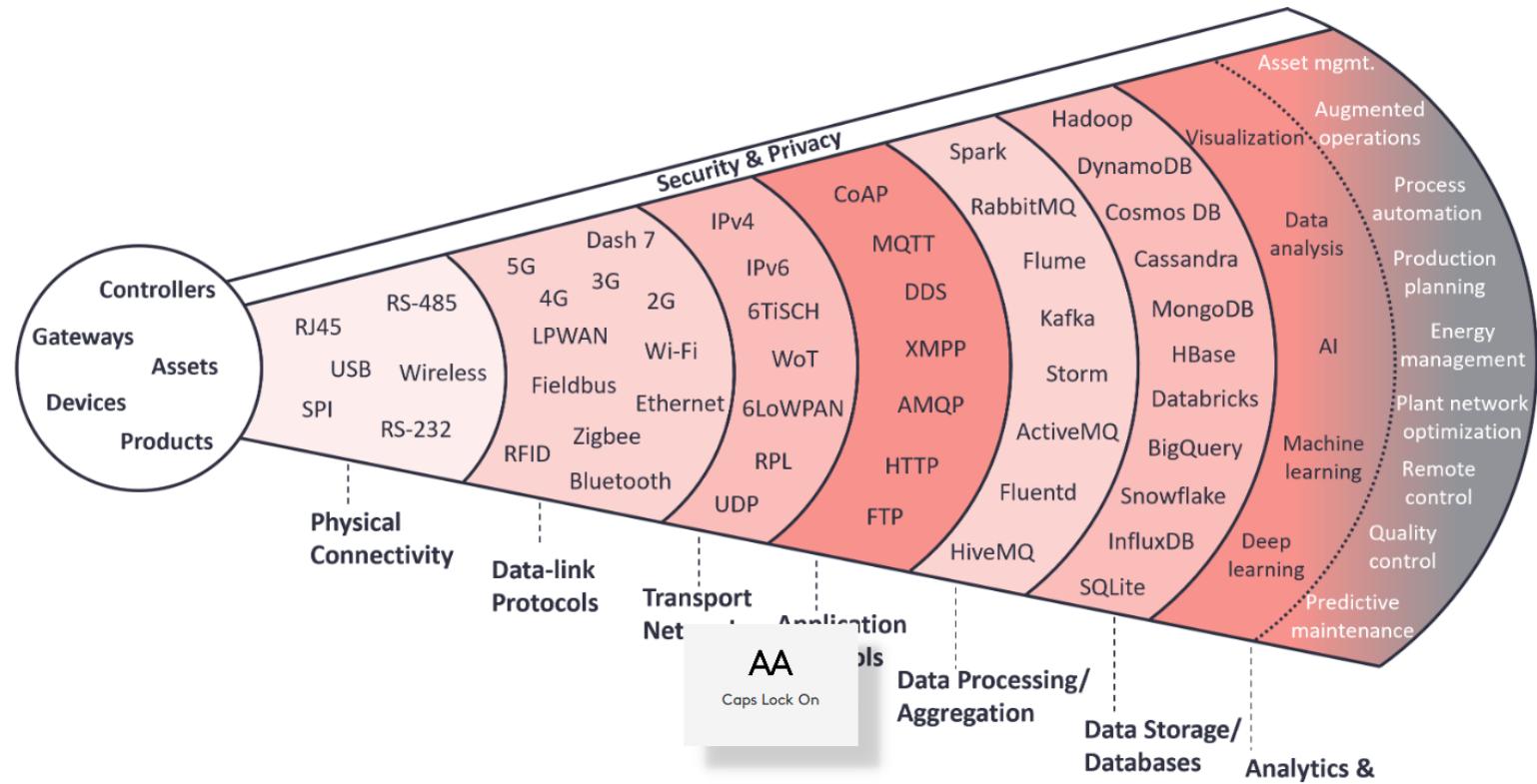
IoT Protocols

IoT ANALYTICS

Sept 2022 – List non-exhaustive

Your Global IoT Market Research P

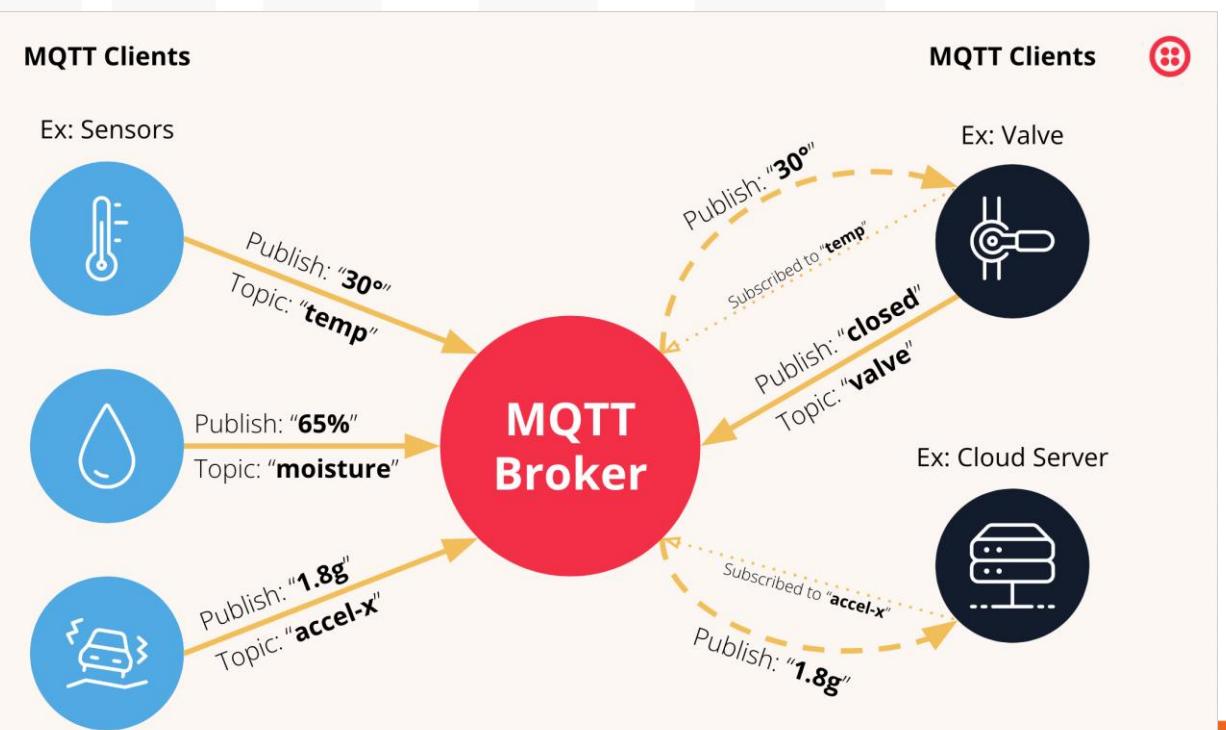
The IoT protocol landscape: From device to applications



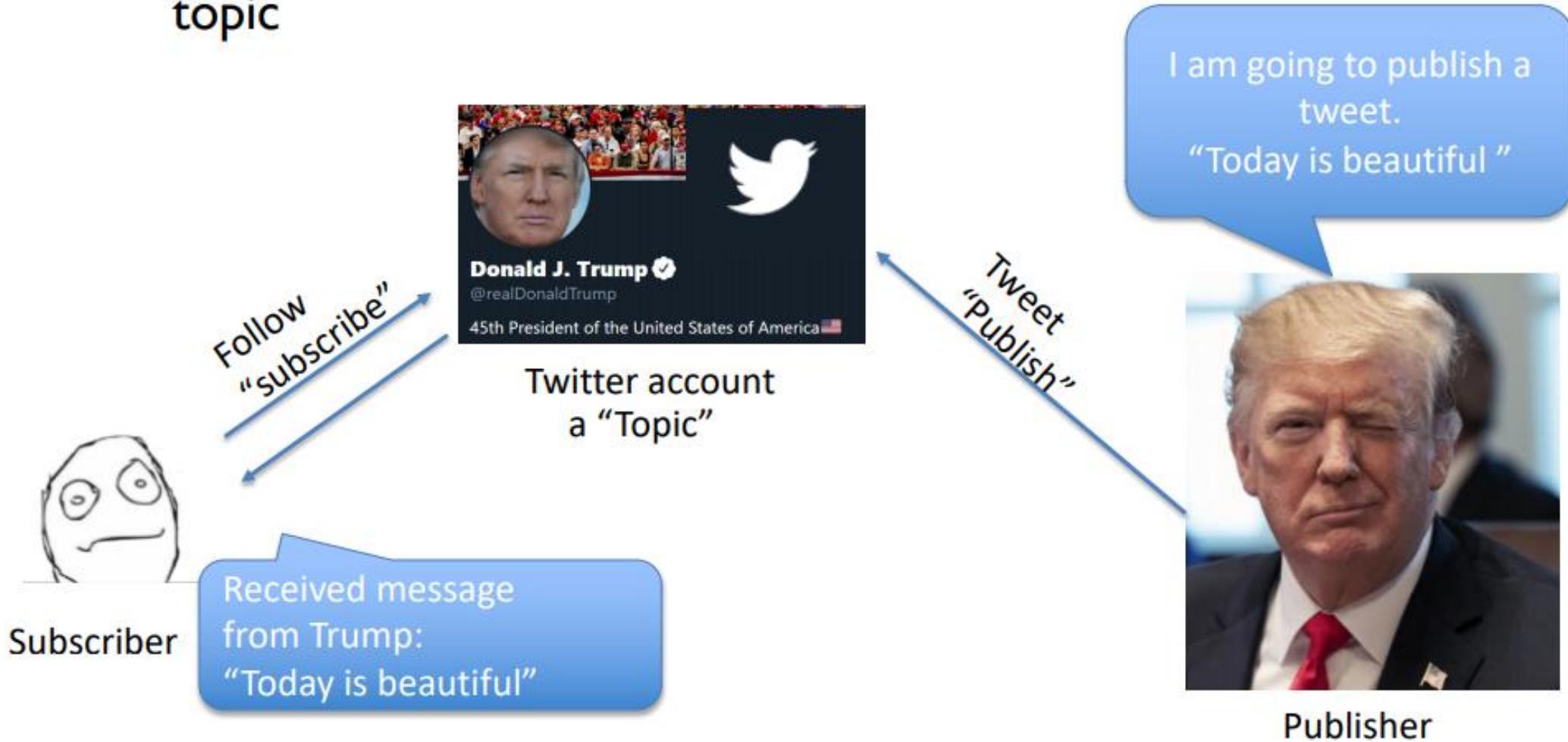
MQTT Protocols

- MQTT Protocol (Message Queue Telemetry Transport) เป็น Protocol ที่มีขนาดเล็กในชั้น Application Layer ที่ออกแบบมาเพื่อสนับสนุนการทำงานของระบบ IoT โดยการสนับสนุนการเชื่อมต่อระหว่างอุปกรณ์กับอุปกรณ์ (Machine-to-Machine : M2M) มีการทำงานรับส่งข้อมูลแบบ Publish/Subscribe โดยมีตัวกลางในการสื่อสารคือ MQTT Broker ทำหน้าที่จัดลำดับการรับ-ส่งข้อมูลระหว่างอุปกรณ์และทิ้งที่เป็น Publish/Subscribe

- Publisher (ผู้เผยแพร่ข้อมูล)
- Subscriber (ผู้ติดตาม)
- Topic (หัวข้อที่จะสื่อสาร)
- Broker (เซิฟเวอร์กลาง)



- You can define your own Topic (Twitter Account)
- Subscriber can receive the message you published to your topic



MQTT Protocols

Publish/Subscribe

- Publish ก็คือการส่งข้อมูลแต่จะต้องบอกด้วยว่าข้อมูลที่เราส่งไปนั้น จะส่งไปใน Topic ไหน ส่วน Subscribe คือการรับข้อมูลแต่จะรับข้อมูลเฉพาะที่มาจากการ Topic ที่เรา Subscribe เท่านั้น

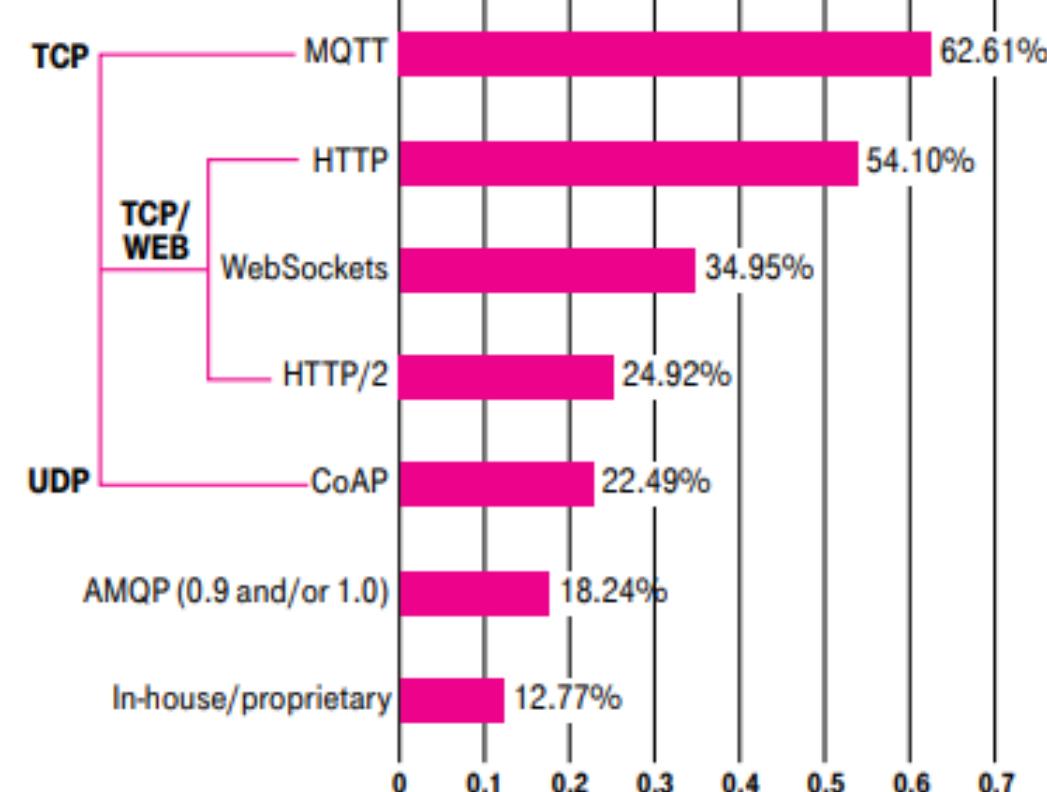
Topic

- คือหัวข้อที่เราสนใจ ซึ่งเอาไว้บ่งบอกว่า สนใจที่จะส่งข้อมูลไปยัง topic นี้หรือรับข้อมูลสำหรับ topics นี้อยู่ตลอด ตัวอย่าง เช่น home/office/temperature หรือ home/thermostat/temperature เป็นต้น

Broker

- คือตัวกลางที่จะรับข้อมูลมาทั้งหมดจาก clients (publisher) ไม่ว่าจะเป็น topics อะไรก็ตาม แล้วทำการจัดการส่งข้อมูลไปยัง clients (subscriber) ที่ได้ทำการ subscribe สำหรับ topic ที่ได้รับข้อมูลมา นั้นสามารถหา global broker หรือ cloud MQTT broker ได้ในหลาย ๆ เว็บไซต์ เช่น hivemq public broker

IoT Protocols



<https://www.daviteq.com/blog/en/what-are-the-most-popular-iot-protocols/>

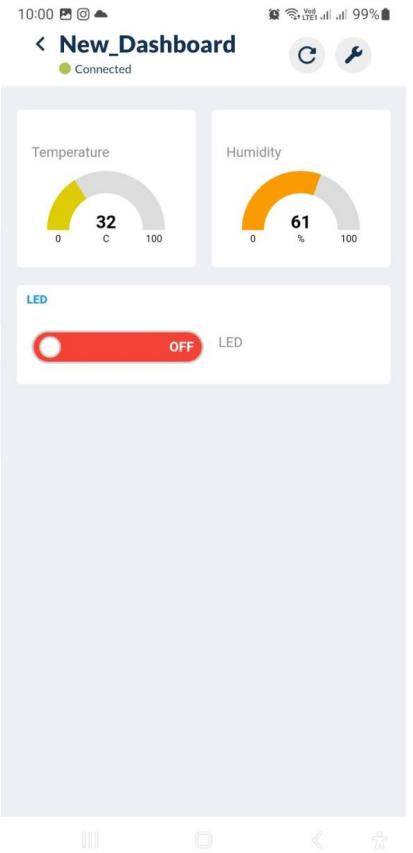
Limitations

NETPIE2020 Quota

Connecting limit	10 Devices
Project	3 Projects
Real time message	9,000,000 Messages/Month
Data Storage	1,000,000 Points-Month
Shadow read/write	500,000 Operations/Month
API Call	800,000 Operations/Month
Trigger and Action	5,000 Operations/Month
Dashboard	3 Freeboards/Project
Freeboard Connection	3 Concurrent Views



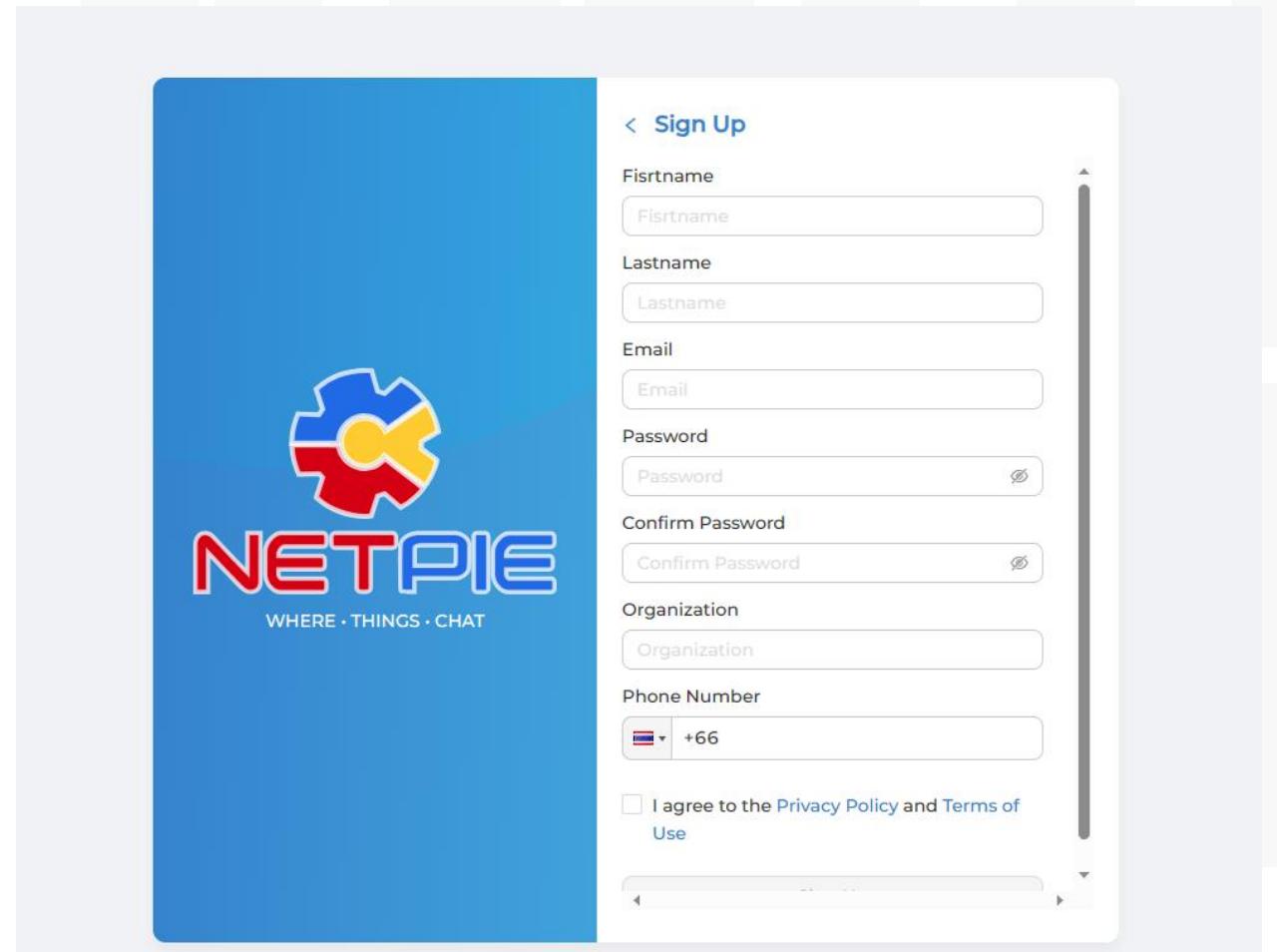
เริ่มต้นการทดลอง NETPIE 2020



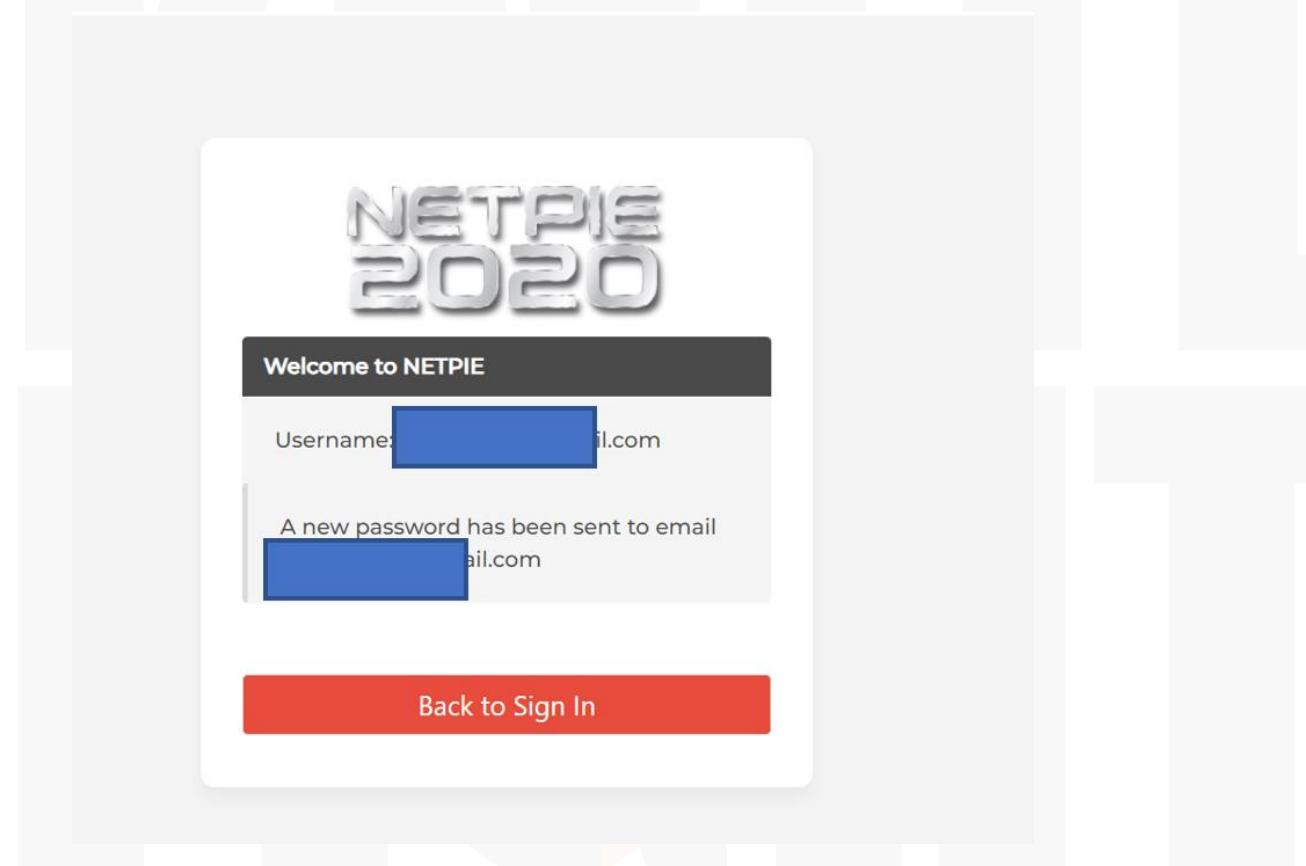
1. เข้าไปที่เว็บไซต์ <http://netpie.io> จากนั้นกดปุ่ม Sign Up เพื่อลงทะเบียนใช้งาน

The screenshot shows the official website for NETPIE 2020. At the top, there is a navigation bar with links for Home, Ecosystem, Developer, and a search icon. On the right side of the header, there are 'LOGIN' and 'Sign up' buttons. The main banner features the text 'NETPIE 2020' in large, bold letters, followed by 'From Makers Nation Toward Smart Nation'. Below the banner, there is a large illustration of a cloud-based IoT ecosystem. The cloud contains a central gear icon and various icons representing different IoT devices and services like solar panels, a car, a robot arm, and a conveyor belt. In front of the cloud, there are icons for a smartphone, a laptop displaying a lock icon, and a person holding a tablet. Two buttons at the bottom left are labeled 'GET STARTED' and 'WATCH VIDEO'.

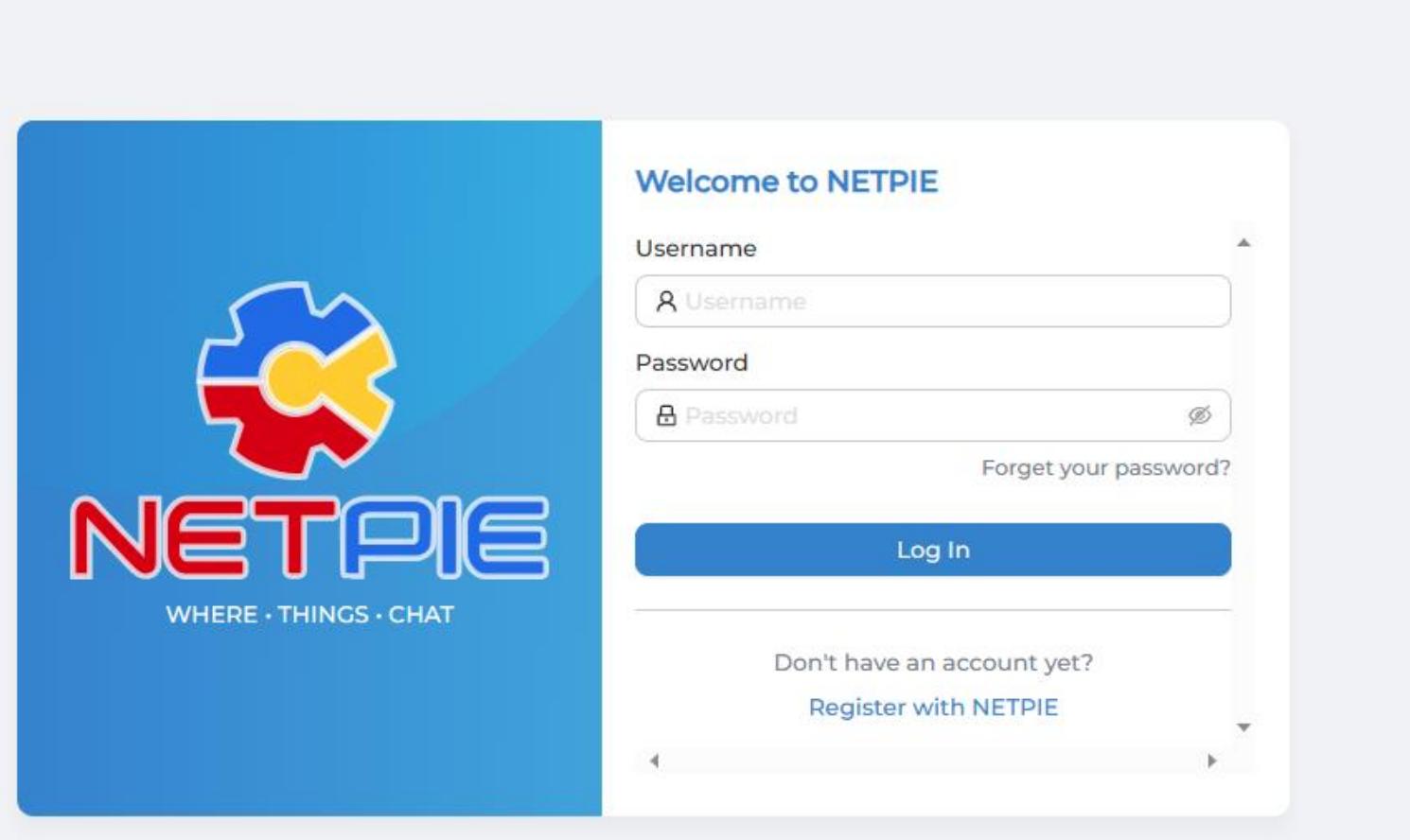
2. ทำการลงทะเบียนโดยใส่ อีเมล์ ชื่อ นามสกุล องค์กร และเบอร์โทรศัพท์ จากนั้นกดปุ่ม SIGN UP



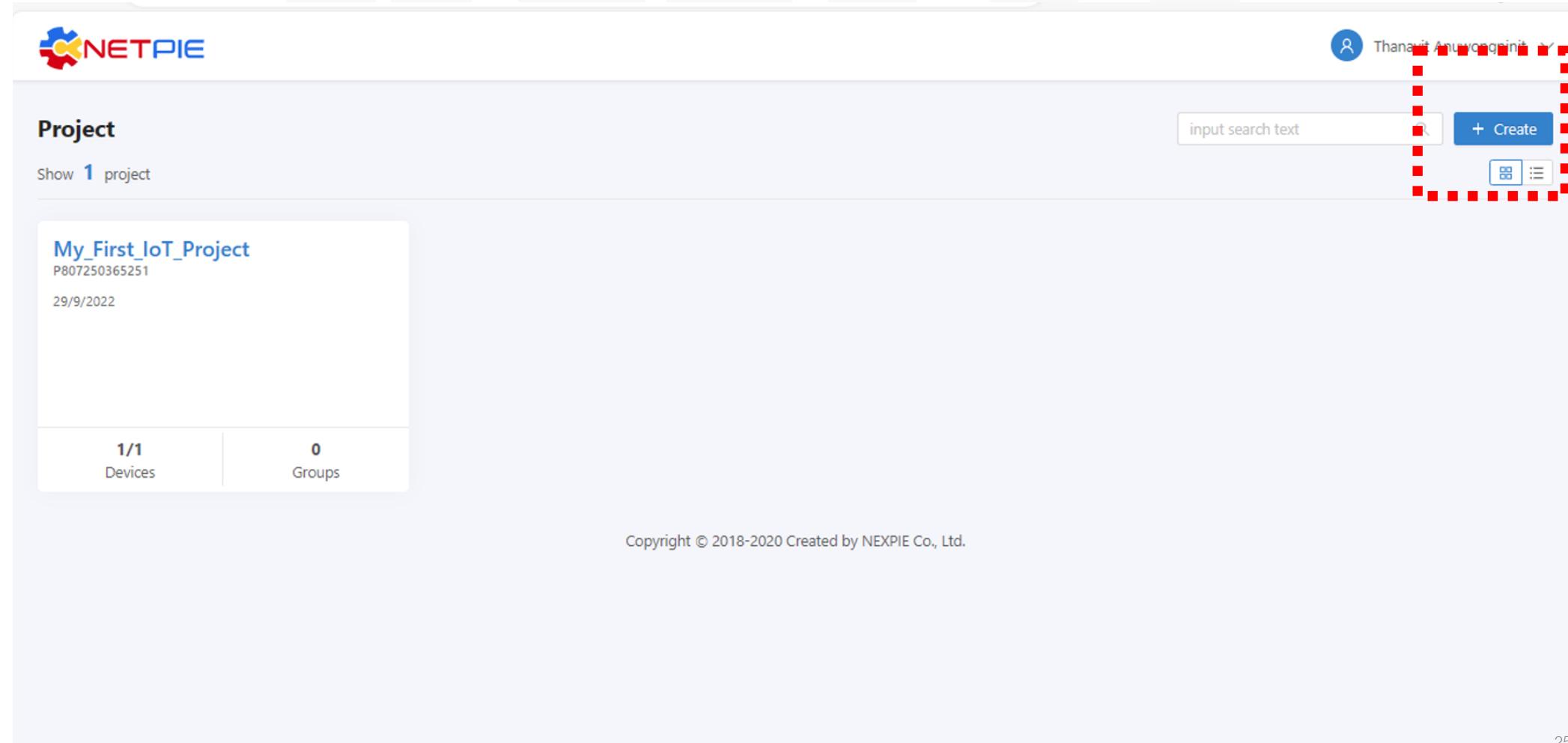
3. ระบบจะส่งส่งอีเมลยืนยันไปยัง Email ที่ทำการลงทะเบียนไว้



4. กลับมายังหน้าเว็บไซต์ netpie จากนั้นทำการ Login ด้วย Email และ Password ที่ทำการลงทะเบียนไว้

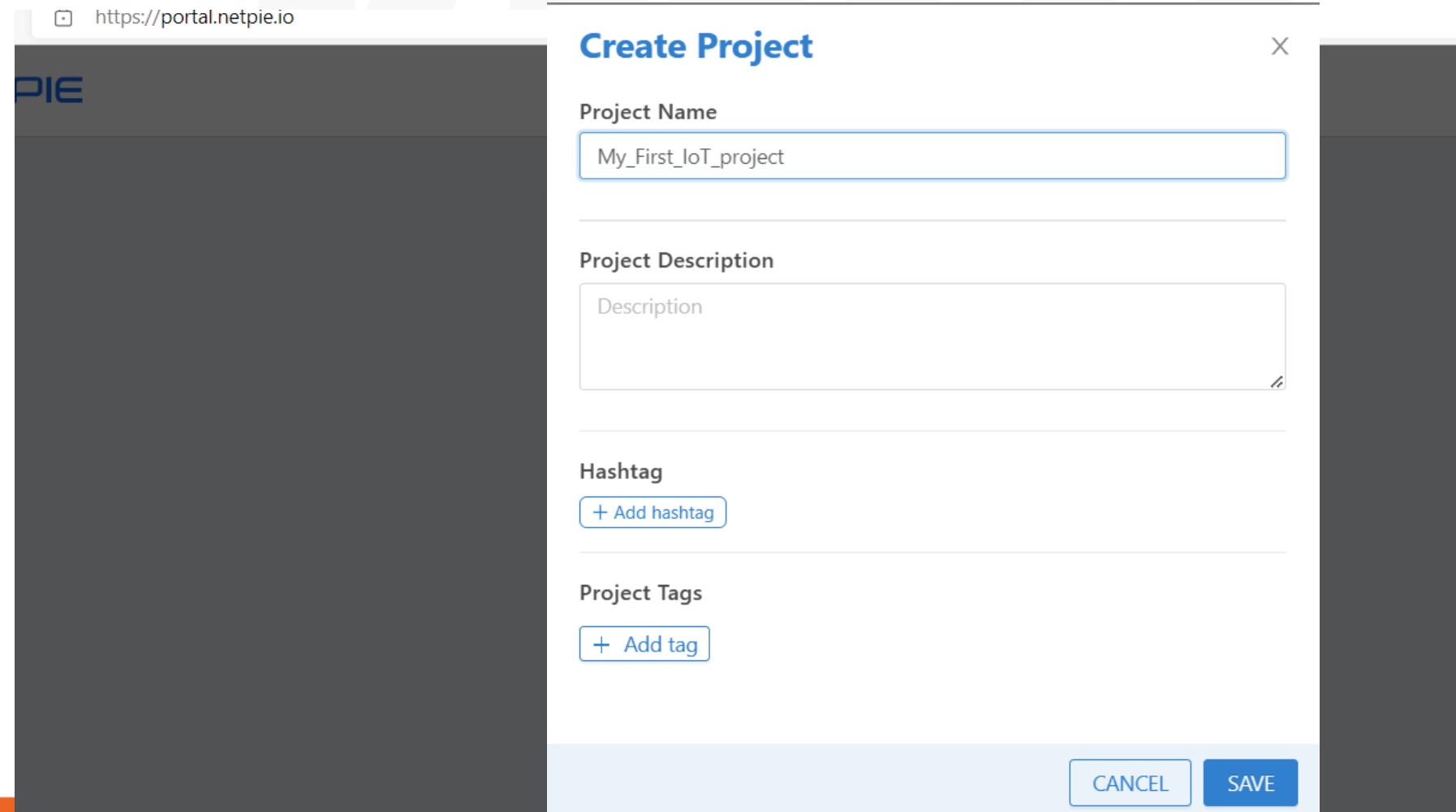


5. ระบบจะเข้าสู่หน้าแรก จากนั้นให้กดปุ่ม + Create บนขวาหน้าจอ เพื่อสร้างโปรเจ็คใหม่



The screenshot shows the NETPIE Project Management interface. At the top, there is a header with the NETPIE logo and a user profile for 'Thanawit Anuwongponkit'. Below the header, there is a search bar labeled 'input search text' and a blue button labeled '+ Create' with a plus sign icon. On the left, there is a sidebar titled 'Project' with the text 'Show 1 project'. A single project card is displayed, titled 'My_First_IoT_Project' with the ID 'P807250365251'. The project card includes the creation date '29/9/2022'. At the bottom of the project card, there are two status indicators: '1/1 Devices' and '0 Groups'. At the very bottom of the page, there is a copyright notice: 'Copyright © 2018-2020 Created by NEXPIE Co., Ltd.'

6. ตั้งชื่อโปรเจ็คใหม่ จากนั้นกดปุ่ม Save



7. Project ใหม่ที่สร้างจะปรากฏเข้าไปยังโปรเจ็คที่ทำการสร้างขึ้นมา

The screenshot shows a web browser window for the NETPIE portal at the URL <https://portal.netpie.io/p/P560016244454/overview>. The left sidebar has sections for PROJECT (+ Add Project), WORKSPACE (Overview selected), and SETTINGS (Setting). The main area displays the 'IoTE_Project / overview' page, which includes a 'Detail' section with a Project ID of P560016244454 and a 'Copy' button, and a 'Devices' section showing 0 devices (0 online, 0 offline). A success message 'Create Project Success' is visible in the top right corner.

8. หน้า Dashboard ของโปรเจ็ค ให้ทำการเลือก “Device”

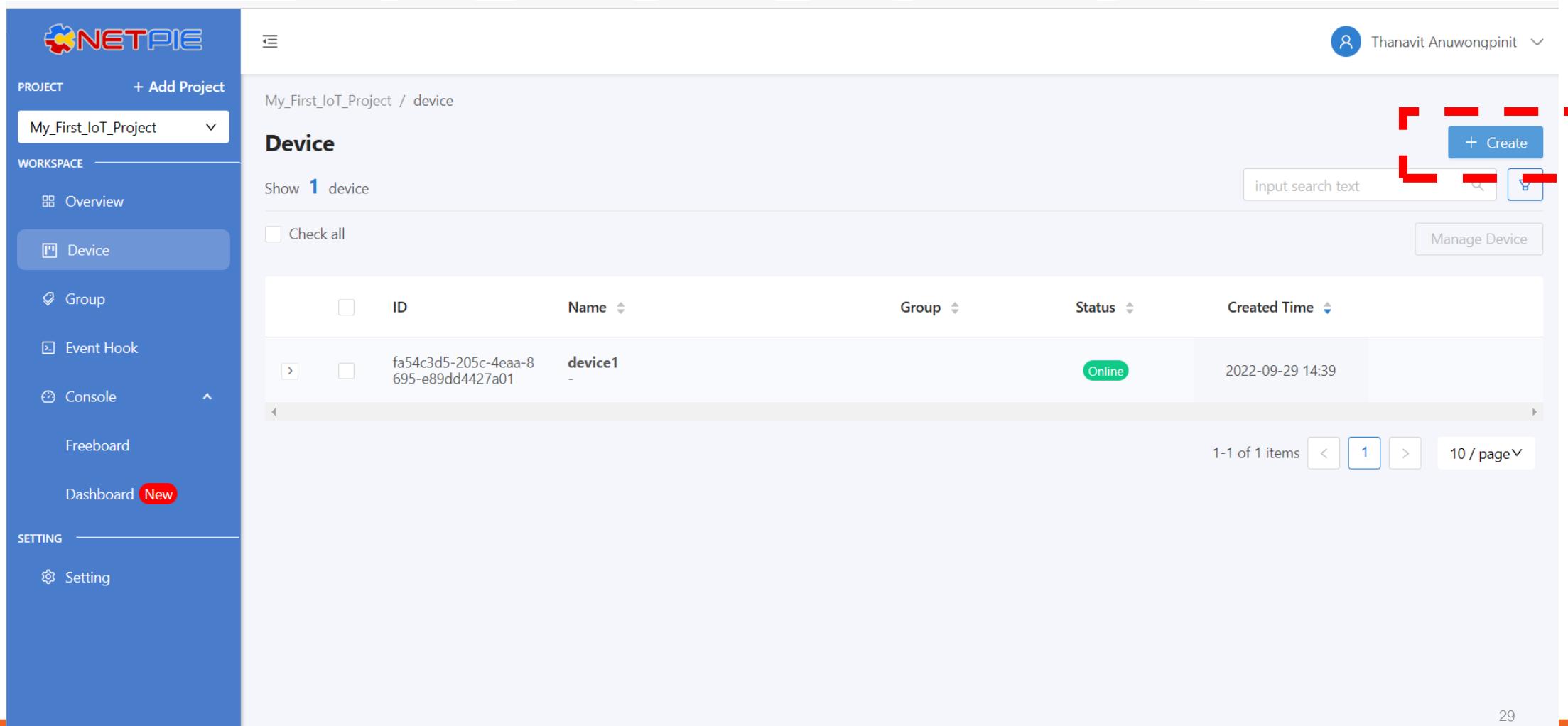
The screenshot shows the NETPIE Platform interface. On the left, there is a sidebar with the following menu items:

- PROJECT: My_First_IoT_Project
- WORKSPACE:
 - Overview
 - Device** (highlighted with a red border)
 - Group
- Event Hook
- Console
- Freeboard
- Dashboard (New)
- SETTING: Setting

The main content area is titled "Device" and shows the following details:

- Show 1 device
- Check all checkbox
- + Create button
- input search text search icon
- Manage Device button
- Table header: ID, Name, Group, Status, Created Time
- Table data:
 - ID: fa54c3d5-205c-4eaa-8695-e89dd4427a01, Name: device1, Group: -, Status: Online, Created Time: 2022-09-29 14:39
- Pagination: 1-1 of 1 items, page number 1, 10 / page

9. จากนั้นกดปุ่ม Create เพื่อสร้าง Device ใหม่ ขึ้นมา

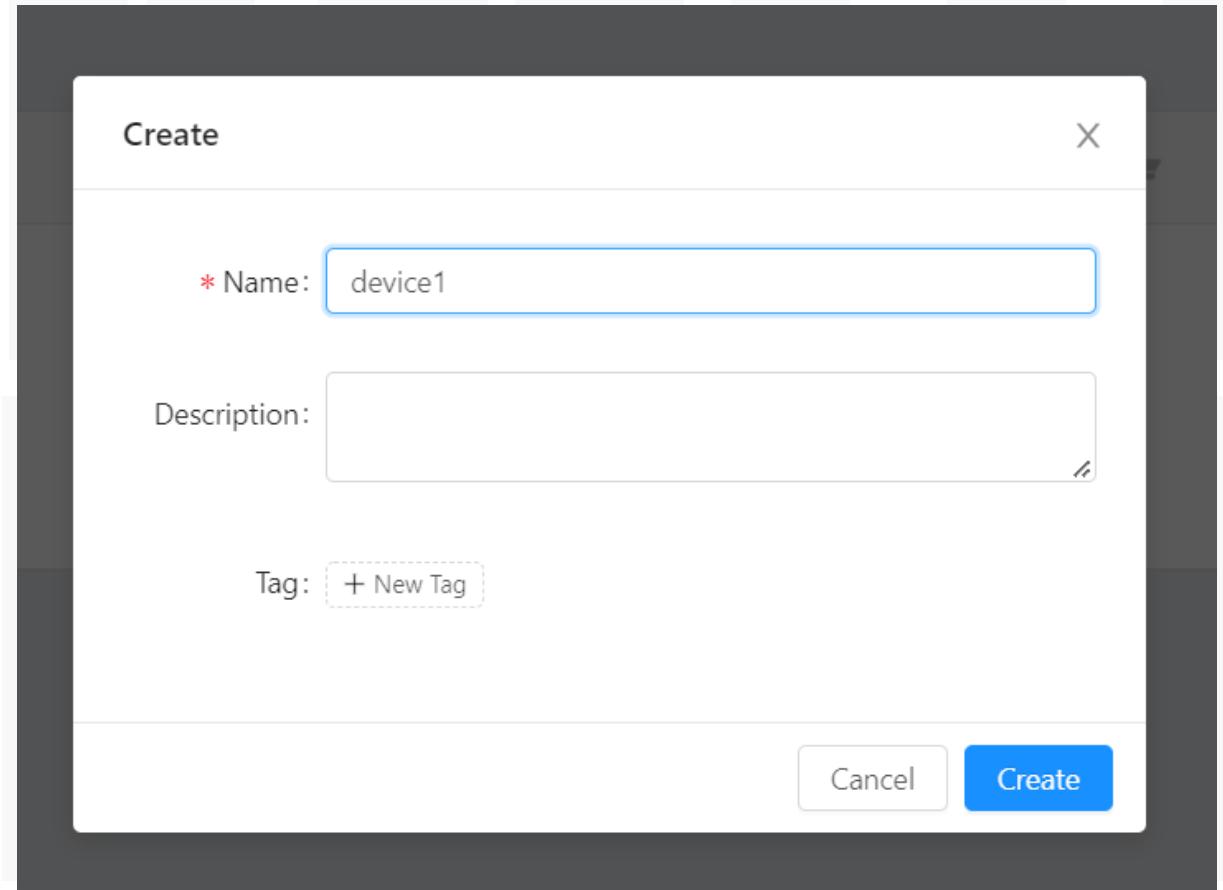


The screenshot shows the NETPIE web interface for managing devices. On the left, a sidebar menu includes options like PROJECT, WORKSPACE (Overview, Device, Group, Event Hook, Console, Freeboard, Dashboard), and SETTING (Setting). The main area is titled 'Device' and shows a table with one item:

ID	Name	Group	Status	Created Time
fa54c3d5-205c-4eaa-8695-e89dd4427a01	device1	-	Online	2022-09-29 14:39

On the right side of the interface, there is a red box highlighting the '+ Create' button, which is used to add a new device.

10. ตั้งชื่ออุปกรณ์ที่สร้างขึ้นใหม่ ในที่นี่ เช่น device1 จากนั้นกดปุ่ม Create



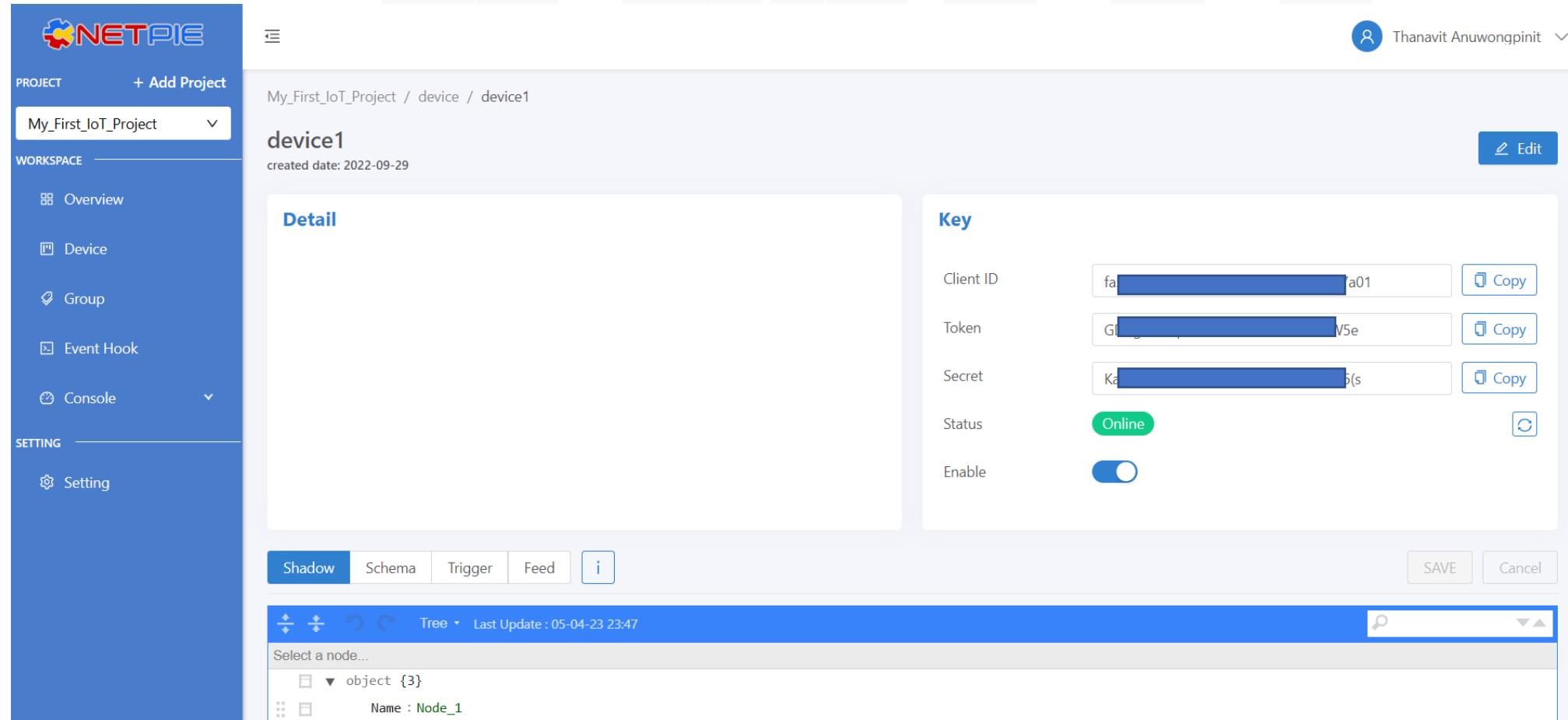
11. จะเห็นว่า device1 ได้ถูกสร้างขึ้นมาเรียบร้อยแล้ว จากนั้นคลิกเข้าไปที่ device1 ที่เพิ่งสร้างขึ้นมาสำเร็จ

The screenshot shows the NETPIE Device Management interface. At the top right, a modal window displays a green checkmark icon and the text "Create Device Success". Below the modal, there is a search bar with placeholder text "input search text" and a "Manage Device" button. On the left, a sidebar shows the path "IoTE_Project / device" and a "Device" section with a "Create" button. The main area displays a table with one device entry:

ID	Name	Group	Status	Created Time
72f50d4f-ac69-40bf-95b0-c53e5846a60d	Device1	-	Offline	2023-09-16 22:14

A red arrow points from the text "Device1 ที่เพิ่งสร้างขึ้นมาสำเร็จ" in the previous slide to the "Device1" entry in the table.

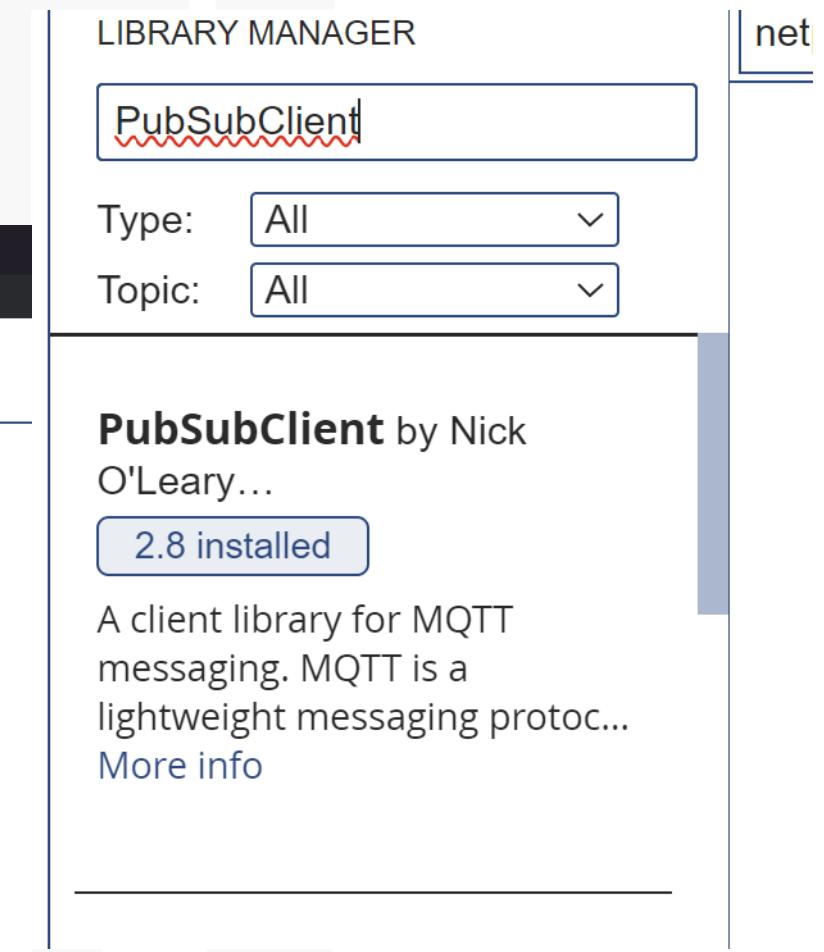
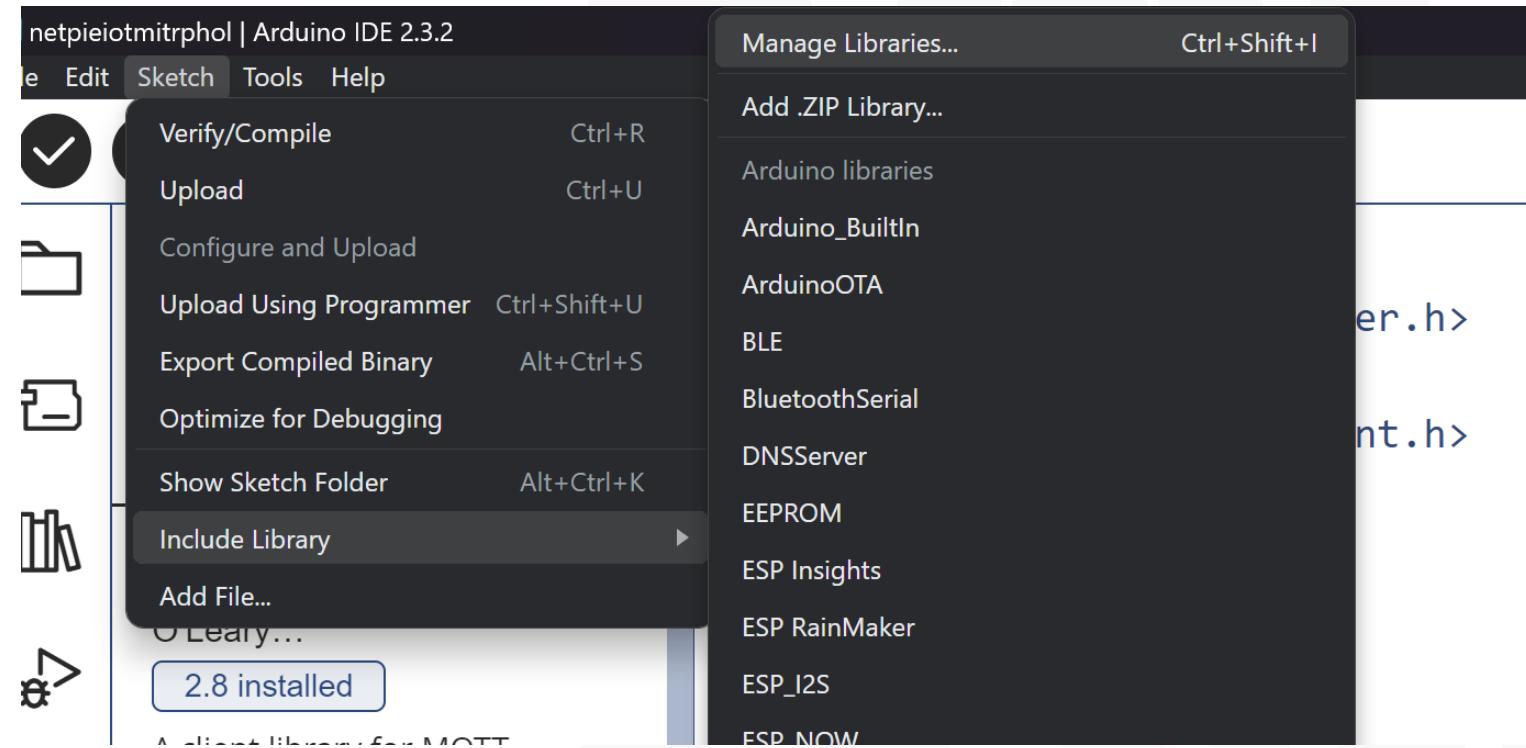
12. ใน device1 จะปรากฏหน้าจอแสดงสิ่งต่าง ๆ ได้แก่ Key ต่าง ๆ , สถานะรวมถึง Format ข้อมูลต่าง ๆ เช่น Shadow/Schema



The screenshot shows the NETPIE web interface for managing IoT devices. On the left, there's a sidebar with navigation links: PROJECT (+ Add Project), WORKSPACE (My_First_IoT_Project selected), and SETTINGS (Setting). The main content area is titled 'device1' (created date: 2022-09-29) and includes tabs for 'Detail', 'Key', 'Shadow', 'Schema', 'Trigger', 'Feed', and an info icon ('i'). The 'Key' tab displays fields for Client ID, Token, Secret, Status (set to 'Online'), and an 'Enable' toggle switch. Below the tabs is a tree view labeled 'Tree' with a message 'Last Update : 05-04-23 23:47'. The tree structure shows 'object {3}' and a node named 'Name : Node_1'. At the bottom right are 'SAVE' and 'Cancel' buttons.

13. ติดตั้งไลบรารี PubSubClient

Sketch > Include Library > Manage Library



14. นำโค้ด netpie_lab มาทำการแก้ไข บรรทัดที่ 20-21 / 24-26

```
9 const char* ssid = "YOUR_SSID"; //อย่าลืมแก้ไข SSID ของ WIFI ที่จะให้ NodeMCU ไปเขื่อนต่อ
10 const char* password = "YOUR_PASSWORD"; //อย่าลืมแก้ไข PASSWORD ของ I
11 const char* mqtt_server = "broker.netpie.io"; //อย่าลืมแก้ไข BROKER
12 const int mqtt_port = 1883;
13 const char* mqtt_Client = "fa54c3d5-205c-4eaa-8695-xxxxxxxx"; //อย่า
14 const char* mqtt_username = "GDAgJLLwTp4is4fPhwTLxxxxxxxxx"; //อย่า
15 const char* mqtt_password = "KaQBToBx0~7FNMcK*dD$xxxxxxxxxx"; //อย่า
16
```



อย่าลืมแก้ไข
MQTT_Client_ID
MQTT_Token
MQTT_Secret

อธิบาย code

```

40
41 void callback(char* topic, byte* payload, unsigned int length) {
42     Serial.print("Message arrived [");
43     Serial.print(topic);
44     Serial.print("] ");
45     String message;
46     for (int i = 0; i < length; i++) {
47         message = message + char(payload[i]);
48     }
49 }
50 Serial.println(message);
51 if(String(topic) == "@msg/operator"){
52     if (message == "ON"){
53         digitalWrite(LED_PIN,HIGH);
54         //client.publish("@shadow/data/update", "{\"data\" : {\"led\" : \"on\"}}");
55         Serial.println("LED ON"); }
56     else if (message == "OFF") {
57         digitalWrite(LED_PIN,LOW);
58         //client.publish("@shadow/data/update", "{\"data\" : {\"led\" : \"off\"}}");
59         Serial.println("LED OFF"); }
60     }
61 }
62

```

ถ้ามีข้อความเข้ามา ให้เก็บในตัวแปร message และตรวจสอบว่าข้อความเข้ามา เป็นคำว่าอะไร

ถ้ามีข้อมูลที่ subscribe มาจะให้ไปทำงานที่นี่

อธิบาย code

```

63 void setup() {
64     Serial.begin(115200);
65     pinMode(LED_PIN,OUTPUT);
66     dht.begin(); ——————  

67     Serial.println();
68     Serial.print("Connecting to ");
69     Serial.println(ssid);
70     WiFi.begin(ssid, password); //เชื่อมต่อ กับ WIFI
71     while (WiFi.status() != WL_CONNECTED) {  

72         delay(500);
73         Serial.print(".");
74     }
75     Serial.println("");
76     Serial.println("WiFi connected");
77     Serial.println("IP address: ");
78     Serial.println(WiFi.localIP()); //เชื่อมต่อ กับ WIFI สำเร็จ แสดง IP
79     client.setServer(mqtt_server, mqtt_port); //กำหนด MQTT_BROKER PORT
80     client.setCallback(callback); //ตั้งค่าฟังก์ชันที่จะทำงานเมื่อมีข้อมูลเข้ามาผ่านการ subscribe
81     client.subscribe("@msg/operator");
82 }
```

ตั้งค่าเริ่มต้น pin/Serial

ตรวจสอบสถานะการณ์เชื่อมต่อ WiFi

แสดงผลหมายเลข IP ที่ ESP32 ได้รับ

การเชื่อมต่อ กับ MQTT Server และ
PORT1883

Topic ที่บอร์ดไป
subscribe

ตั้งค่าฟังก์ชันที่จะทำงานเมื่อมีการ subscribe ใน
topic บรรทัดที่ 81 เกิดขึ้น ที่นี่คือ callback

เริ่มต้นใช้งาน NETPIE 2020

อธิบาย code

```
--  
84 void loop() {  
85     if (!client.connected()) {  
86         reconnect();  
87     }  
88     client.loop();  
89     long now = millis();  
90     if (now - lastMsg > 5000) { //จับเวลาส่งข้อมูลทุกๆ 5 วินาที  
91         lastMsg = now;  
92         ++value;  
93         float h = dht.readHumidity();  
94         float t = dht.readTemperature();  
95         DataString = "{\"data\":{\"temperature\":"+ (String)t+", \"humidity\":"+ (String)h+"}}";  
96         // Example of data : {"data":{"temperature":25 , "humidity": 60}}  
97         DataString.toCharArray(msg, 100);  
98         Serial.println("Hello NETPIE2020");  
99         Serial.print("Publish message: ");  
100        Serial.println(msg);  
101        client.publish("@shadow/data/update", msg); // อย่าลืมแก้ไข TOPIC ที่จะทำการ PUBLISH ไปยัง MQTT BROKER  
102    }  
103    delay(1);  
104 }
```

จับเวลาส่งข้อมูลทุกๆ 5000 มิลลิวินาที

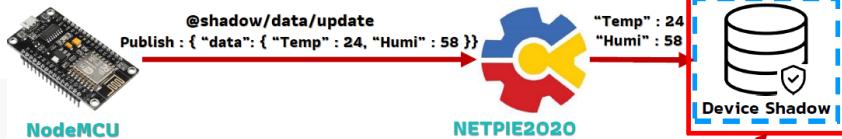
อ่านค่าอุณหภูมิ (t) และความชื้น (h)

สร้างเทมเพลตข้อมูลเป็นแบบ shadow ของ netpie

แปลงเป็น Char Array

ทำการ Publish ข้อมูลไปยัง Topic

@shadow/data/update

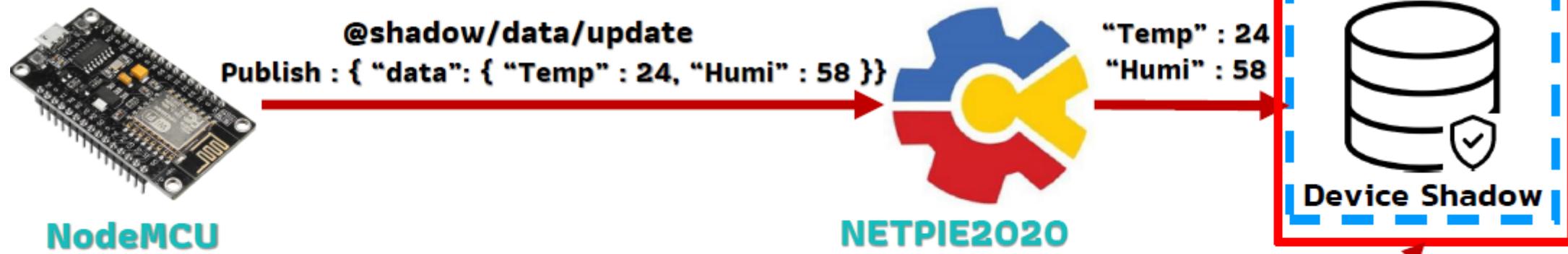


แม่การจะจัดเก็บข้อมูลลง Device Shadow ได้นั้นจำเป็นต้องสร้าง Device Schema ก่อน

Device Data Management

Device Shadow

ตัวอย่างการส่งข้อมูลในรูปแบบ JSON

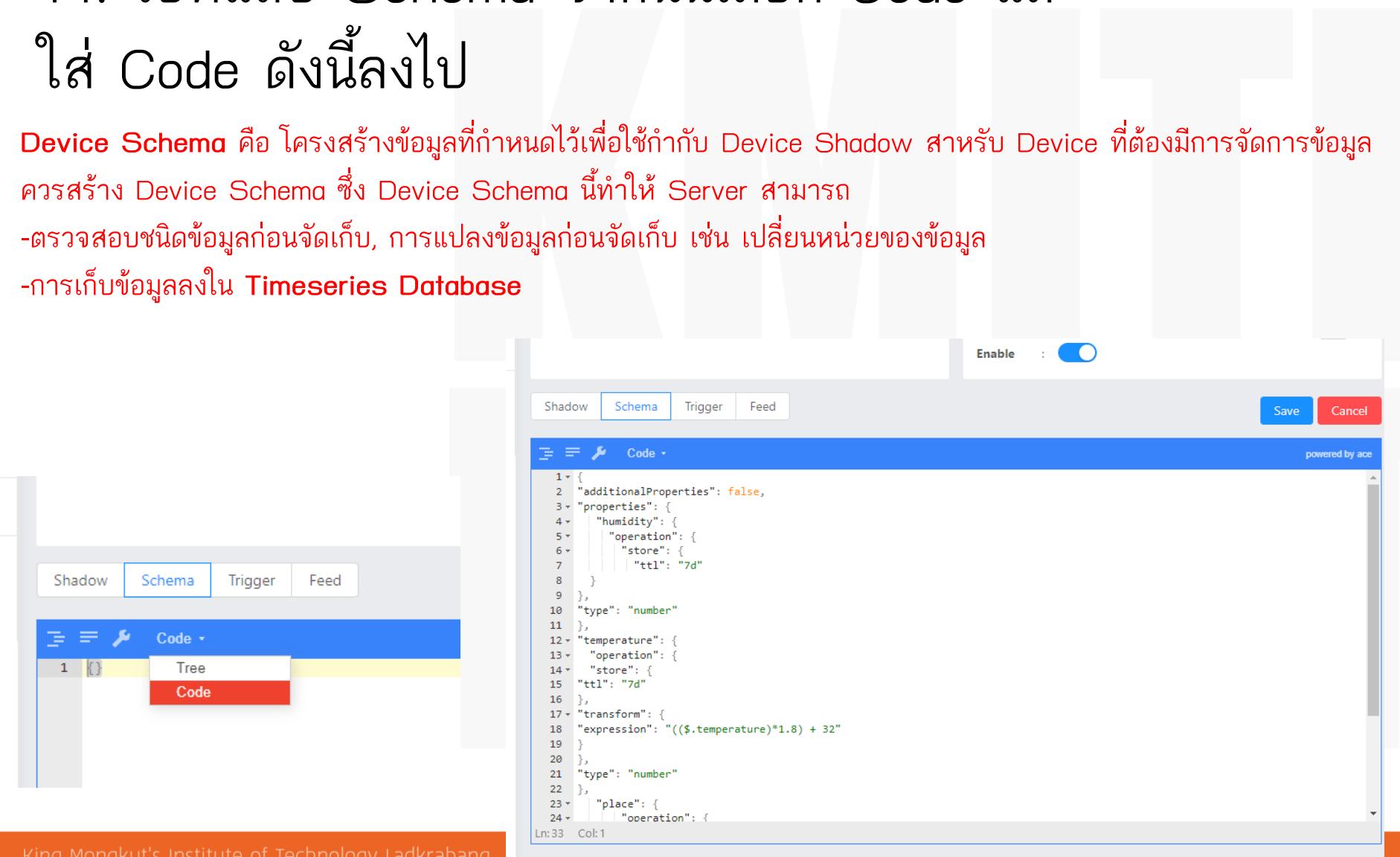


แต่การจะจัดเก็บข้อมูลลง Device Shadow ได้นั้นจำเป็นต้องสร้าง Device Schema ก่อน

14. ไปที่ແນ Schema จากนั้นเลือก Code และ¹ ใส่ Code ดังนี้ลงไป

Device Schema คือ โค้ดสร้างข้อมูลที่กำหนดไว้เพื่อใช้กำกับ Device Shadow สำหรับ Device ที่ต้องมีการจัดการข้อมูล
ควรสร้าง Device Schema ซึ่ง Device Schema นี้ทำให้ Server สามารถ

- ตรวจสอบชนิดข้อมูลก่อนจัดเก็บ การแปลงข้อมูลก่อนจัดเก็บ เช่น เปลี่ยนหน่วยของข้อมูล
- การเก็บข้อมูลลงใน Timeseries Database



Workshop 4 : ส่งค่า Temperature และ Humidity จาก NodeMCU ไปยัง NETPIE2020 และบันทึกใน Timeseries DB

Device Schema ในรูปแบบ JSON

ตั้งค่า **additionalProperties = false**
เพื่อไม่ให้บันทึกค่าอื่นที่นอกเหนือจาก properties ลง Shadow

จัดเก็บตัวแปรประเภทคือ **humidity**
โดยมีคุณสมบัติคือ

- จัดเก็บจำนวน 7 วัน
- จัดเก็บตัวแปรประเภท number

```

{
  "additionalProperties": false,
  "properties": {
    "humidity": {
      "operation": {
        "store": {
          "ttl": "7d"
        }
      },
      "type": "number"
    }
  },
  "temperature": {
    "operation": {
      "store": {
        "ttl": "7d"
      },
      "transform": {
        "expression": "[$.temperature]*1.8 + 32"
      }
    },
    "type": "number"
  },
  "place": {
    "operation": {
      "store": {
        "ttl": "7d"
      }
    },
    "type": "string"
  }
}
  
```

Device Data Management

Workshop 4 : ส่งค่า Temperature และ Humidity จาก NodeMCU ไปยัง NETPIE2020 และบันทึกใน Timeseries DB

Device Schema ในรูปแบบ JSON

```
{
  "additionalProperties": false,
  "properties": {
    "humidity": {
      "operation": {
        "store": {
          "ttl": "7d"
        }
      },
      "type": "number"
    },
    "temperature": {
      "operation": {
        "store": {
          "ttl": "7d"
        },
        "transform": {
          "expression": "[($.temperature)*1.8] + 32"
        }
      },
      "type": "number"
    },
    "place": {
      "operation": {
        "store": {
          "ttl": "7d"
        }
      },
      "type": "string"
    }
  }
}
```

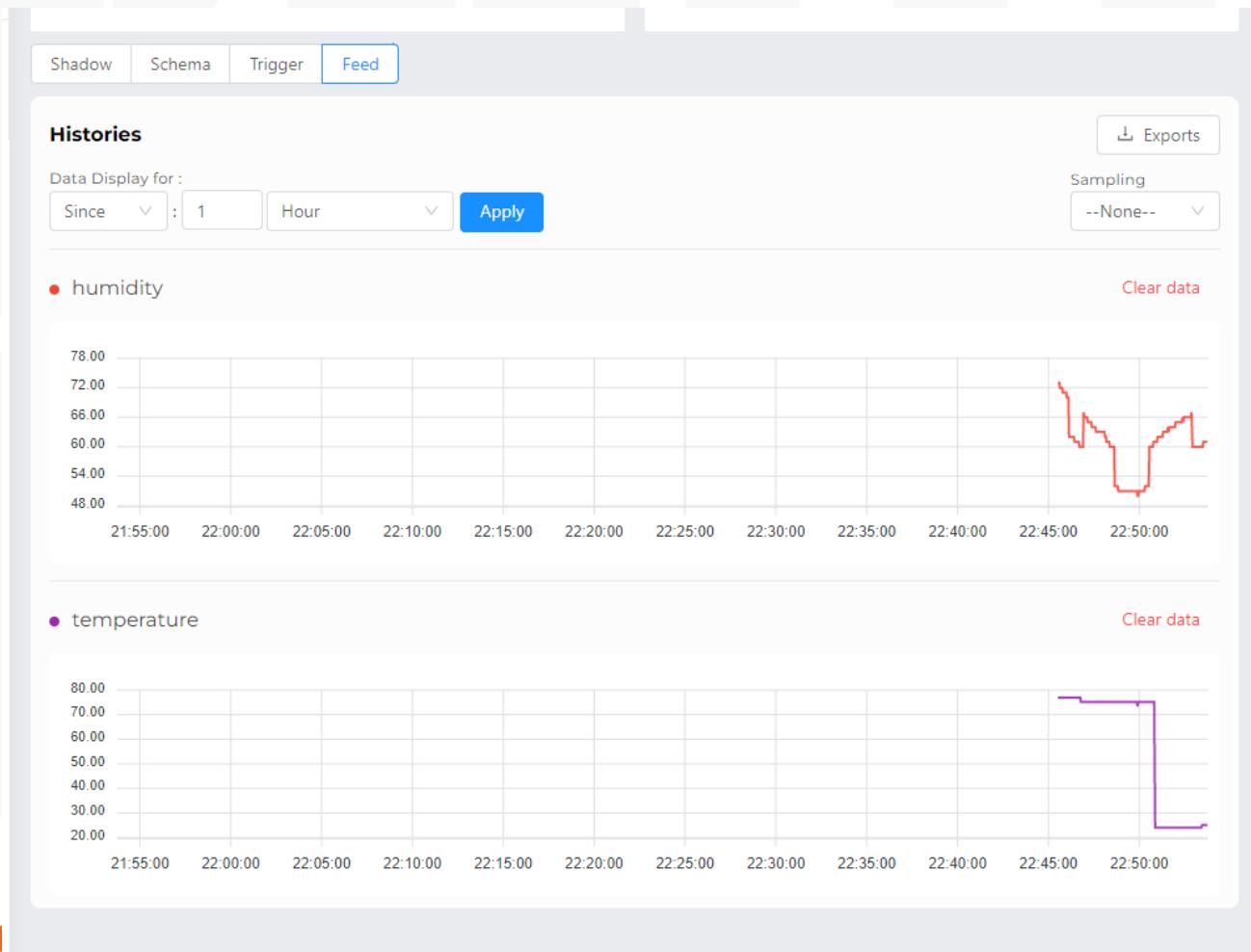
จัดเก็บตัวแปรคือ **temperature** โดยมีคุณสมบัติคือ

- จัดเก็บจำนวน 7 วัน
- แปลงข้อมูลอุณหภูมิ
- จัดเก็บตัวแปรประเภท number

จัดเก็บตัวแปรคือ **place** โดยมีคุณสมบัติคือ

- จัดเก็บจำนวน 7 วัน
- จัดเก็บตัวแปรประเภท string

15. ที่ແບ Feed จะแสดงข้อมูลแบบเรียลไทม์ ที่สามารถเก็บไว้ได้ 7 วัน



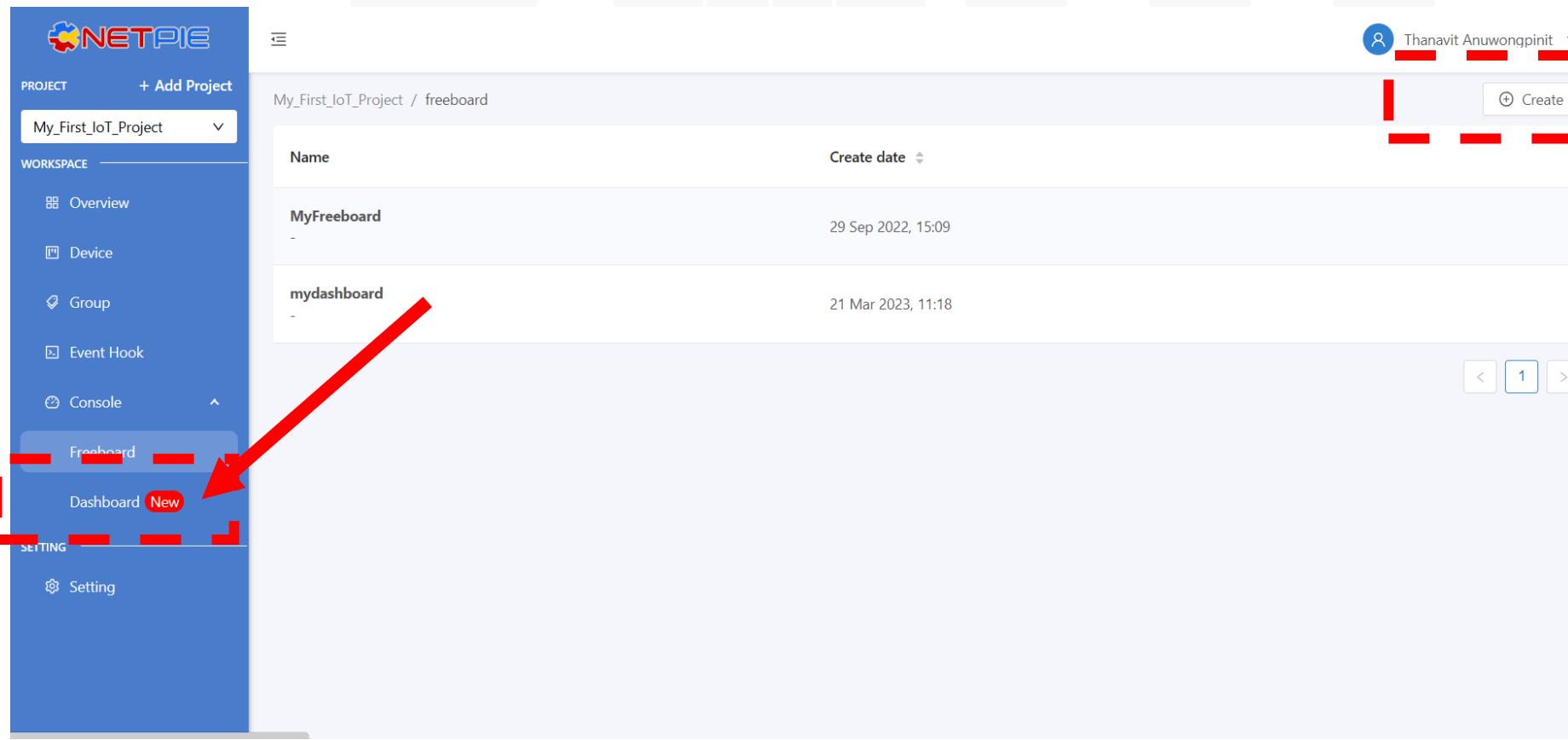
การสร้าง Dashboard บน Netpie 2020



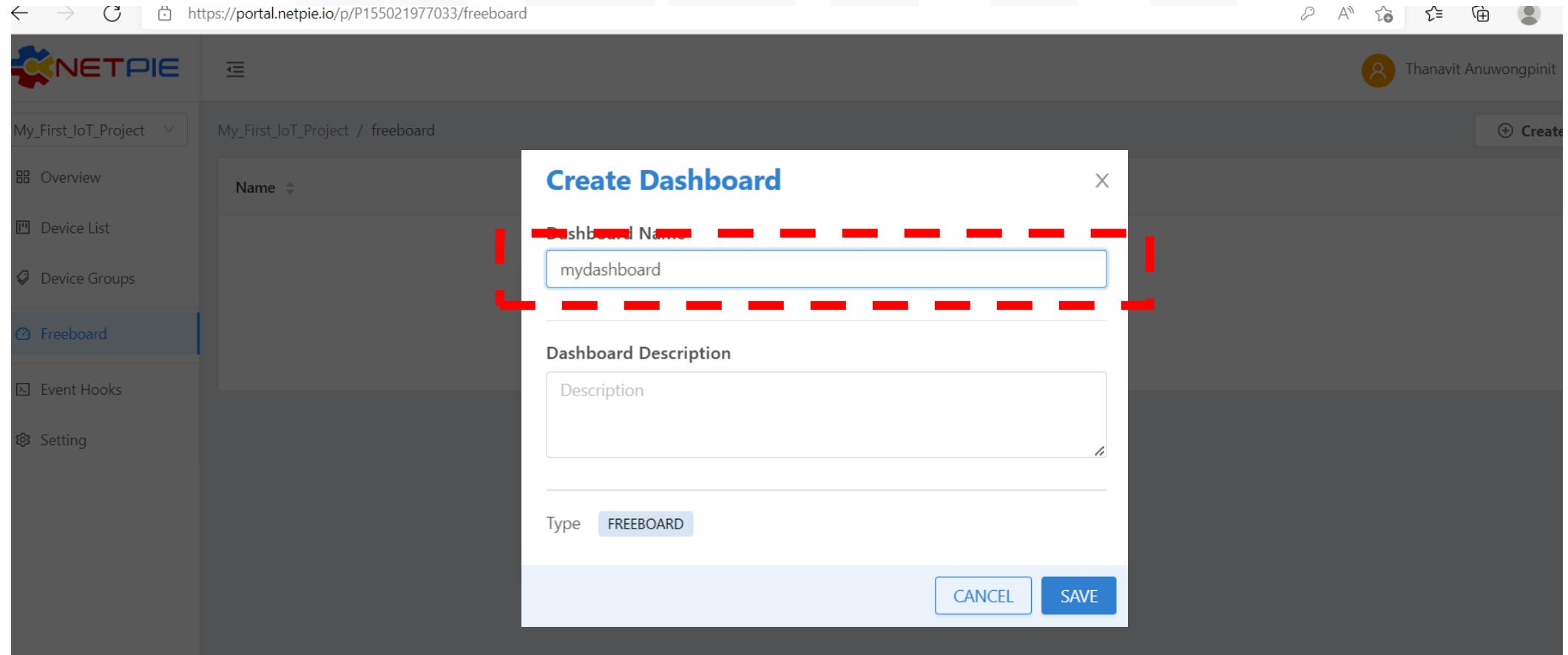
NETPIE2020 สามารถสร้าง Dashboard สำหรับการควบคุมและการแสดงผล ข้อมูลที่ได้มา Device Shadow ซึ่งทางผู้พัฒนาได้พัฒนา Widget Plugins ขึ้นมาให้ ผู้ใช้งานสามารถทำตามความต้องการได้หลากหลาย เช่น สามารถสร้างปุ่มควบคุมและใส่ค่าสั่ง Javascript สำหรับ Action ต่างๆ รวมถึงใช้ใน Mobile App ได้

Dashboard ด้วย NETPIE Freeboard

1. ที่หน้า Project ให้ทำการเลือกเมนู Console จากนั้นเลือก Freeboard จากนั้นให้กดปุ่ม Create เพื่อสร้าง Freeboard ขึ้นใหม่



2. ตั้งชื่อใหม่ในที่นี่ชื่อ MyDashboard



จากนั้นจะเห็น New Dashboard ถูกสร้างขึ้นมา จากนั้นคลิกเข้าไปที่ New_Dashboard ที่สร้างขึ้นมา

Dashboard			+ Create
Name	Device	Create Date	
New_Dashboard	-	2023-09-16 21:48	

จากนั้นไม่ที่ແບມ່ນ “Setting” จากนั้น กดປຸນ + Add device |

New_Dashboard Connected

Dashboard Setting

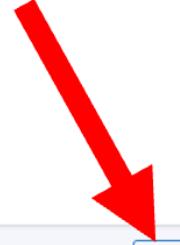
Name: New_Dashboard

Description: Description

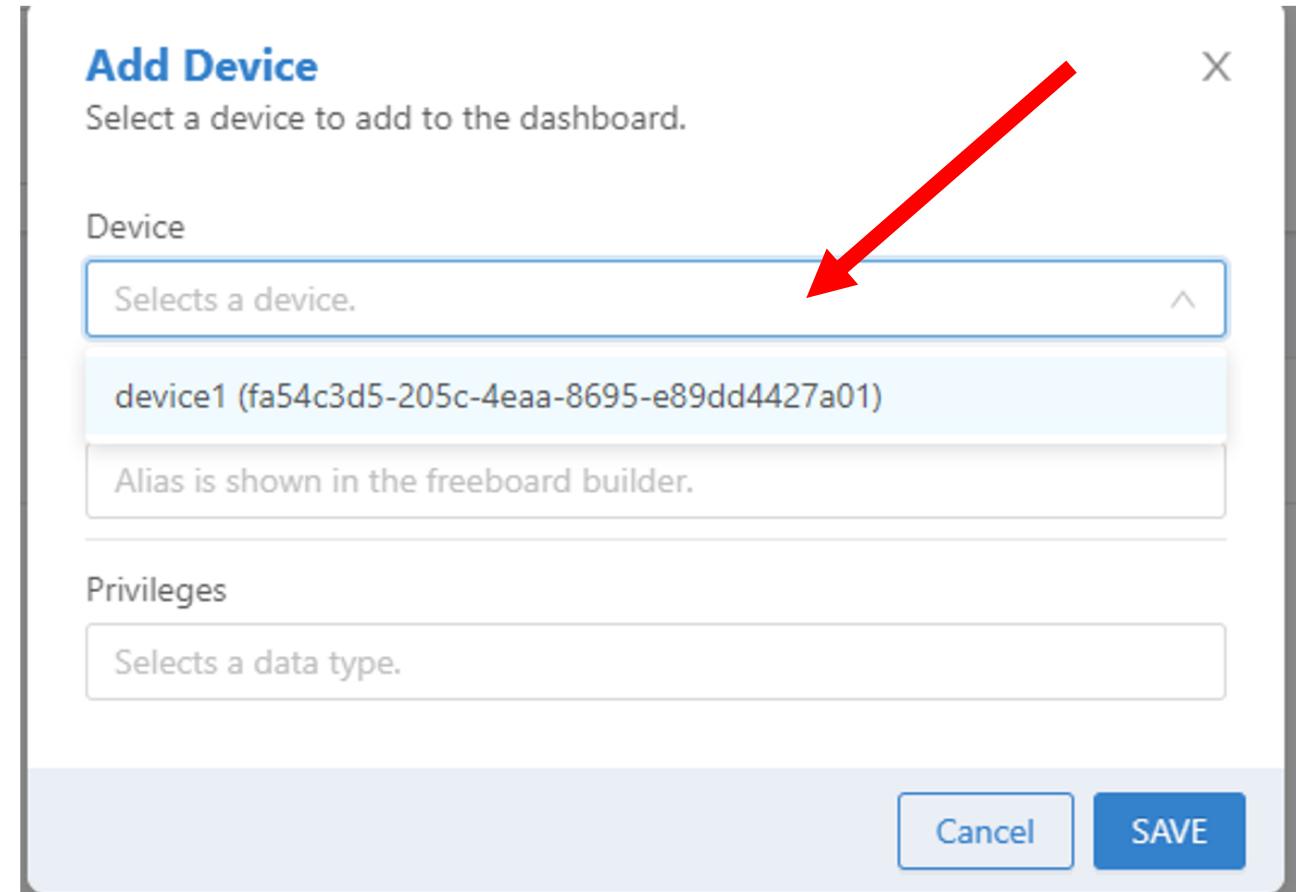
Show 0 Dashboard datasource

+ Add device

ID	Alias	Privileges	Create Date
No device yet You haven't connected any devices to freeboard yet. To add a device, click the "+ Add Device" button in the upper right corner.			



จากนั้นให้ทำการเลือก device ที่ทำการสร้างไว้ จากการทดลองเดิม (ในที่นี้คือ device1)



จากนั้นที่ Privileges ให้เลือกว่า จะสามารถทำอะไรได้บ้าง

“Subscribe Message” คือ การรับข้อมูลใน Topic ที่ต้องการ

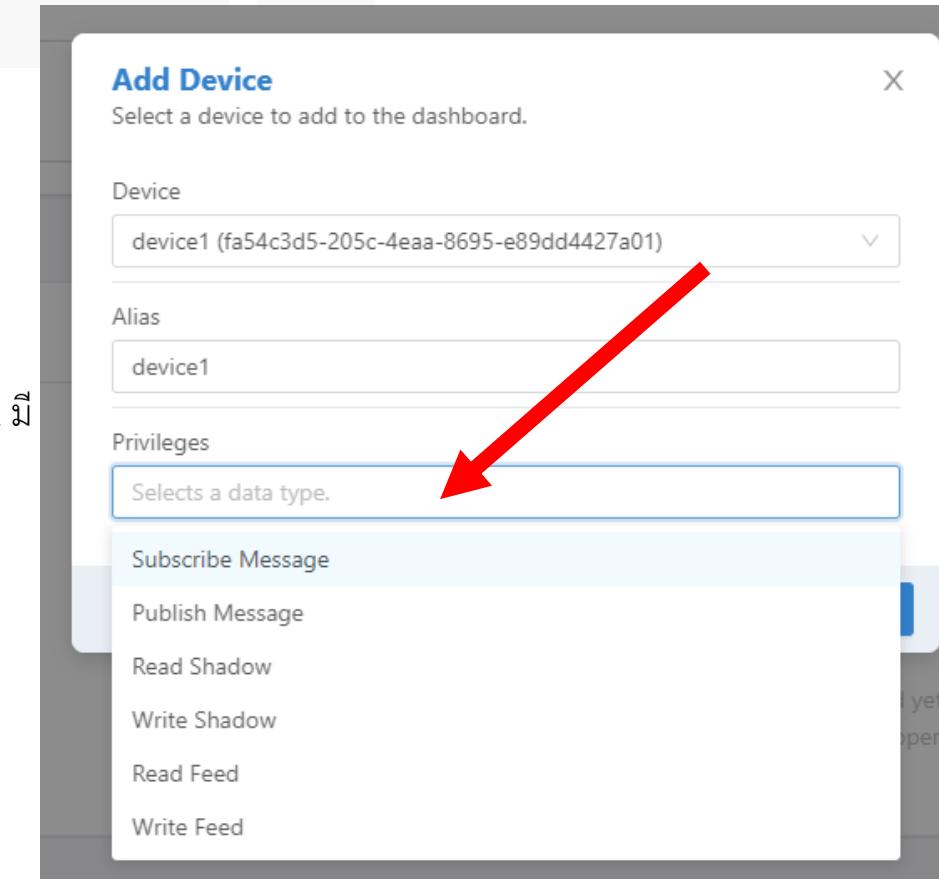
“Publish Message” คือ การส่งข้อมูลไปยัง Topic ที่ต้องการ

“Shadow Read” คือ จำนวนการอ่าน Shadow โดยดูจากขนาดของ Shadow ที่ถูกอ่านออกมานั้น แต่ละ Block มีขนาดไม่เกิน 1 kilobyte

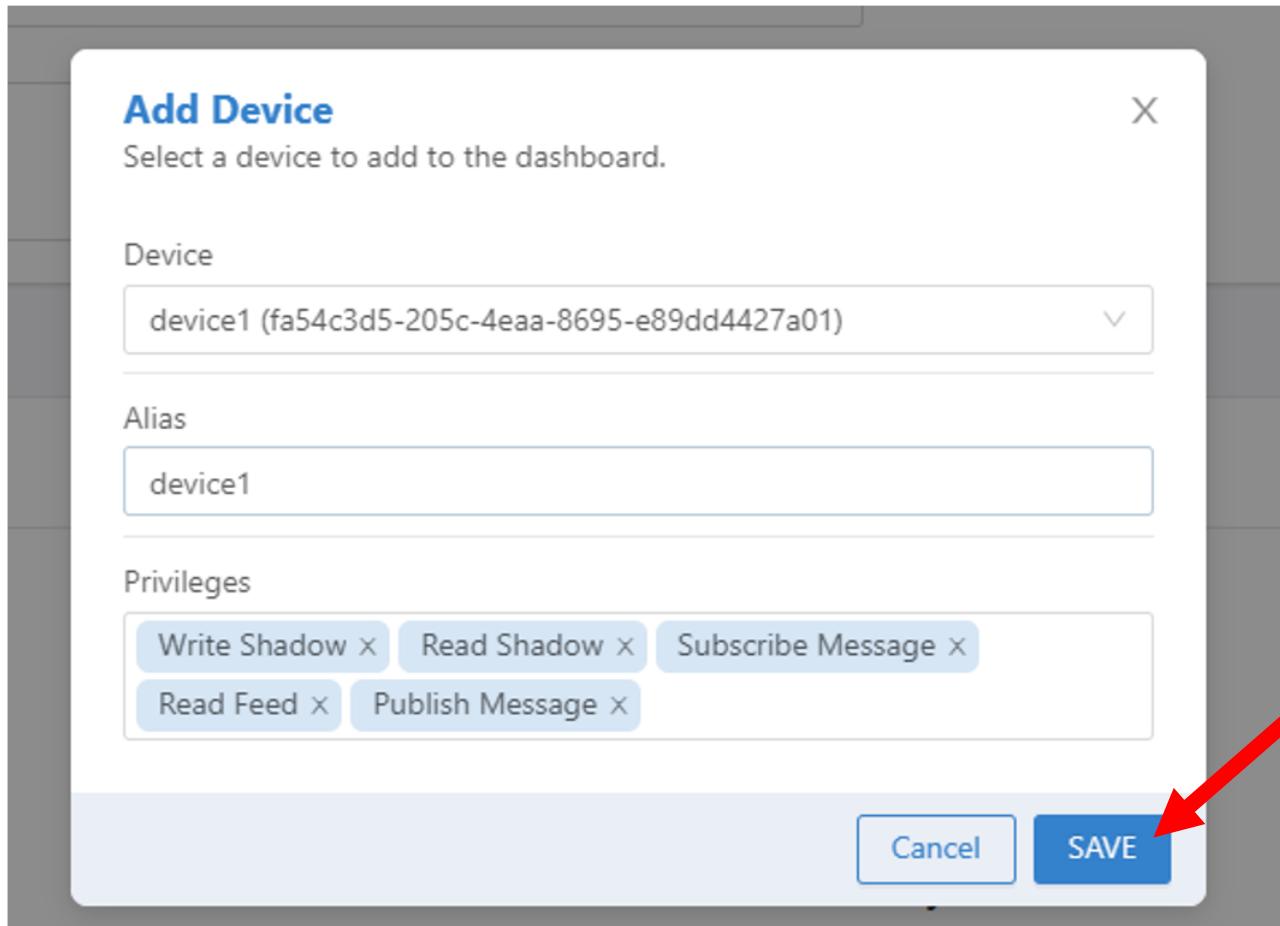
“Shadow Write” คือ จำนวนการเขียน Shadow โดยดูจากขนาดของ Shadow ที่ส่งไปเขียน แต่ละ Block มีขนาดไม่เกิน 1 kilobyte

“Read Feed” คือ การอ่านข้อมูลจาก feed หรือชุดข้อมูลที่ส่งเข้าไป

“Write Feed” คือ การสร้าง feed หรือชุดข้อมูลที่สามารถส่งเข้าไป



จากนั้นกดปุ่ม Save



จากนั้นจะเห็นว่า device1 ได้ถูกสร้างขึ้นมาเชื่อมต่อกับแดชบอร์ดนี้

New_Dashboard Connected

Dashboard Setting

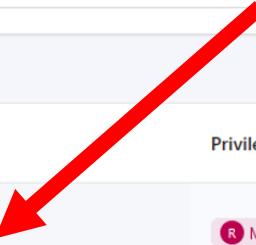
Name: New_Dashboard

Description: Description

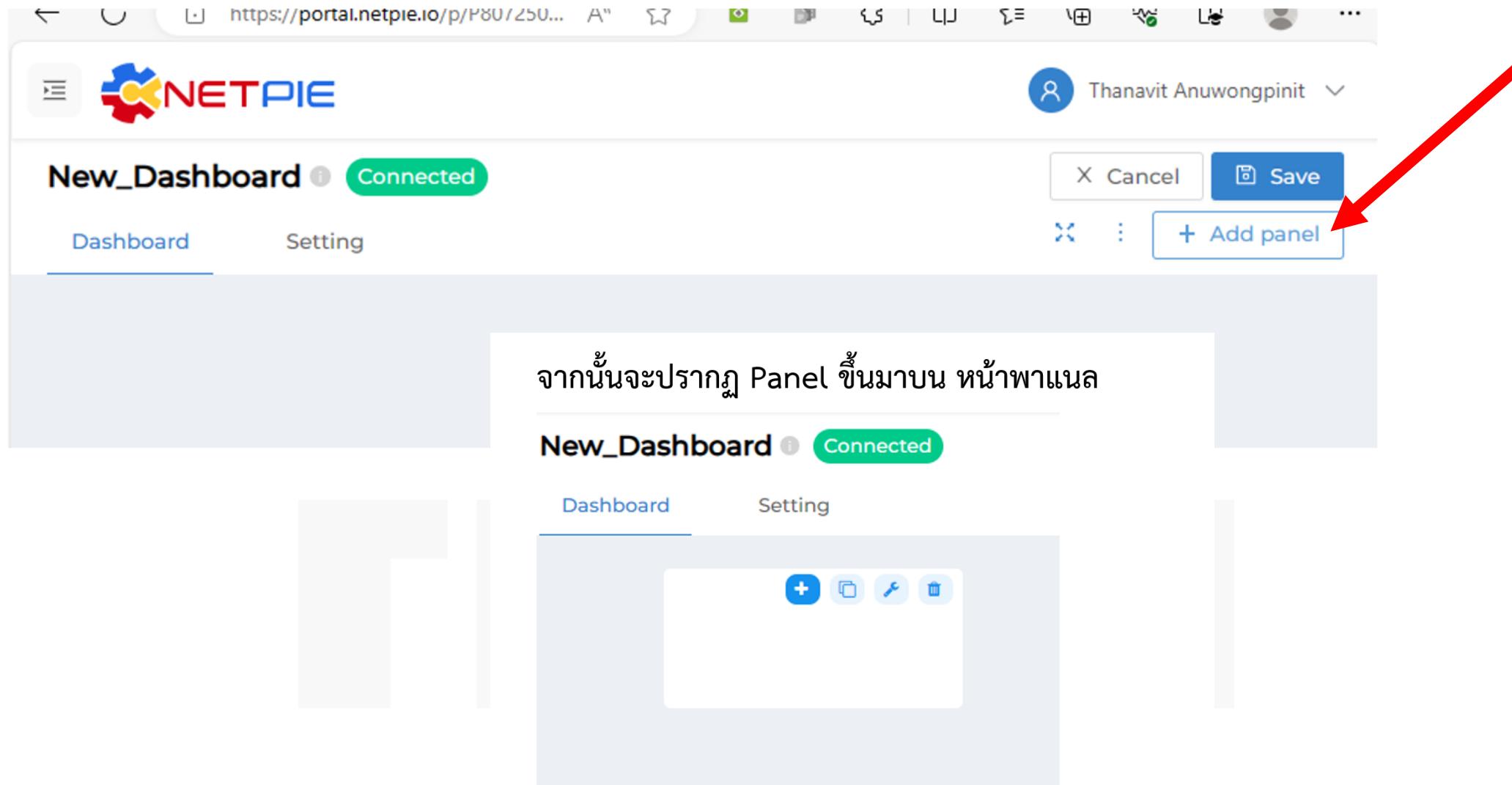
Show 1 Dashboard datasource

+ Add device

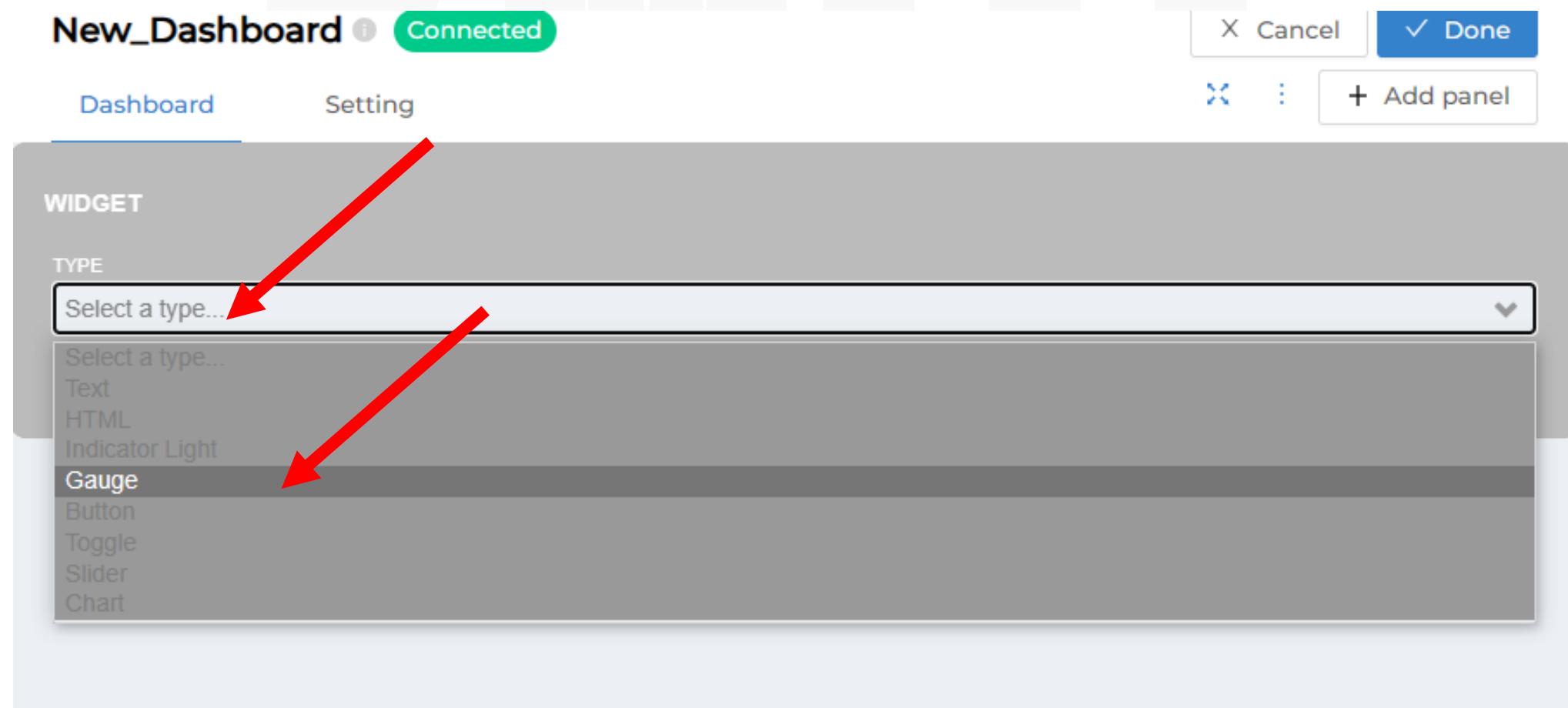
ID	Alias	Privileges	Create Date
fa54c3d5-205c-4eaa-8695-e89dd4427a01	device1	R Message W Message R Shadow W Shadow R Feed	2023-09-16 21:53



ที่ແບແນ່ງກລັບມາທີ່ ແບແນ່ງ “Dashbaord” ຈາກນິ້ນກດປຸ່ມ + Add Panel



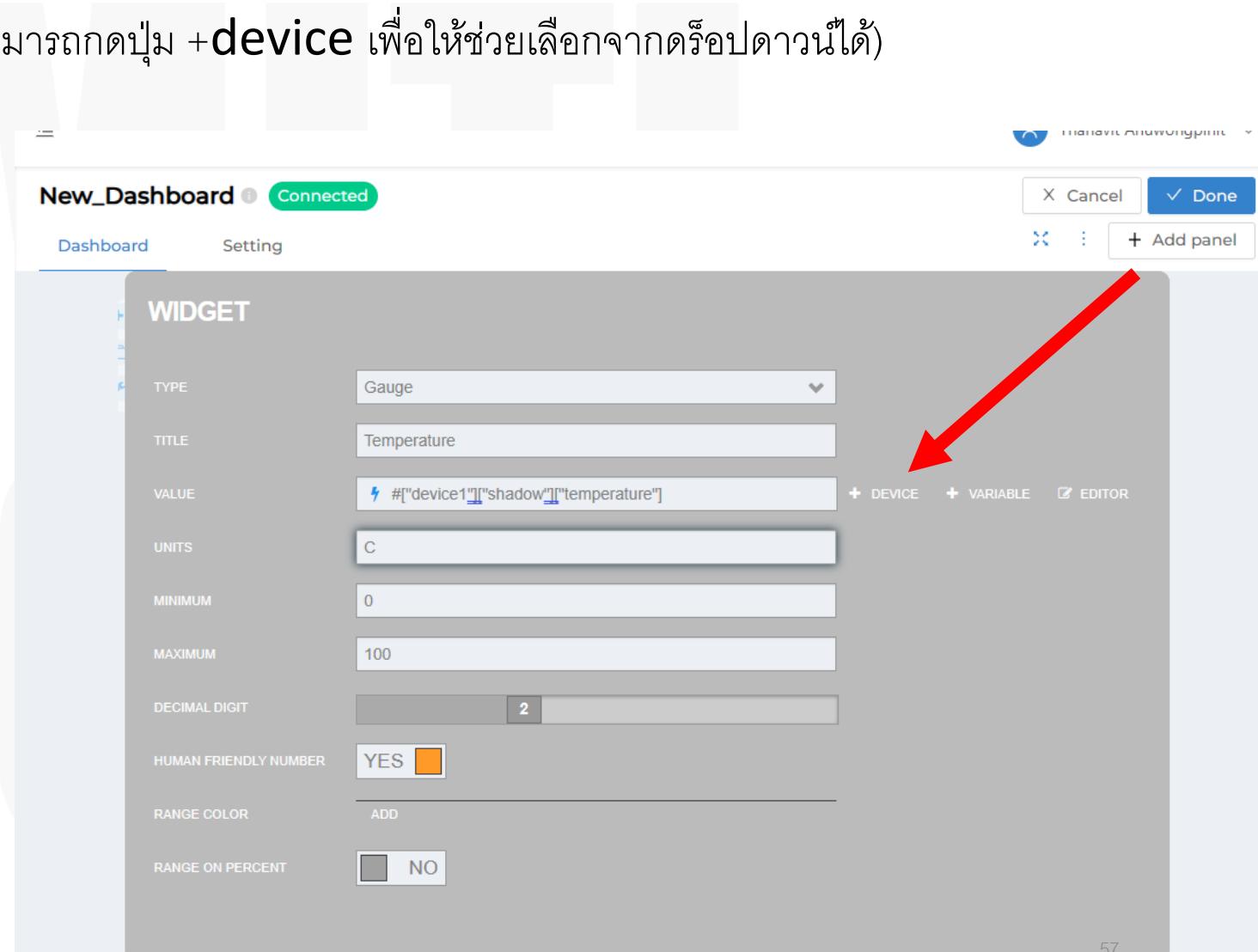
- จากนั้นกดเครื่องหมาย + บนพานแนลที่สร้างขึ้นมา บนกระดาน จะปรากฏ Widget ให้เลือก
- โดยในที่นี่ จะสร้างเกจวัดอุณหภูมิ ให้เลือก “Gauge” (คล้ายกับ Freeboard ที่ผ่านมา)



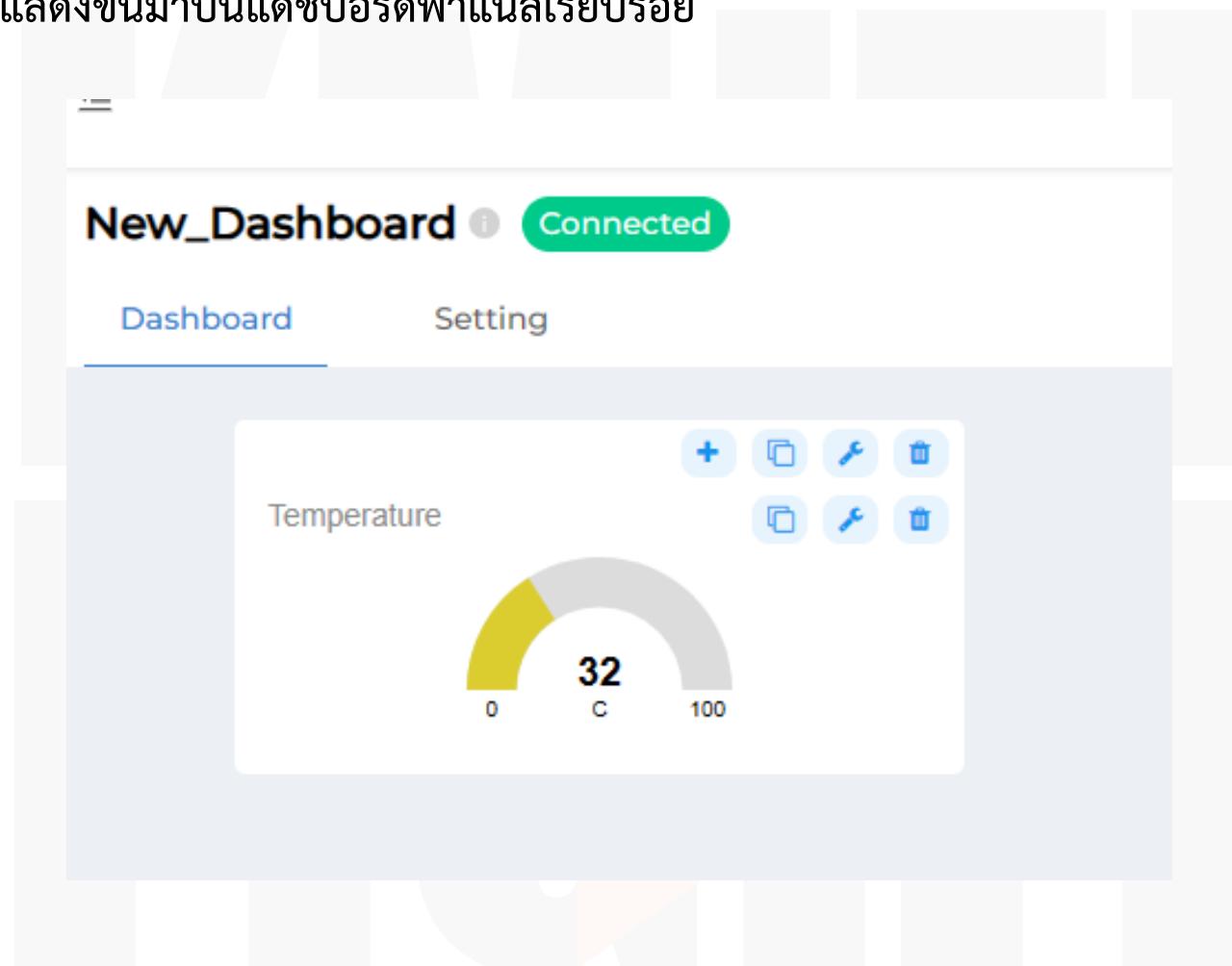
- จากนั้นใส่รายละเอียดต่าง ๆ ลงไป
- โดยที่ **VALUE** จะคล้ายกันกับใน Freeboard (สามารถกดปุ่ม **+device** เพื่อให้ช่วยเลือกจากเครือข่ายที่ได้)

`#[“device1”][“shadow”][“Temperature”]`

จากนั้นกดปุ่ม Done ที่มุมบนขวา



- จะเห็นได้ว่าเกจวัดอุณหภูมิแสดงขึ้นมาบนแดชบอร์ดพาแนลเรียบร้อย



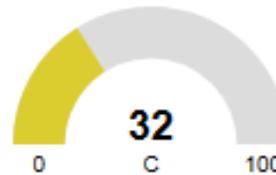
- จะเห็นได้ว่าเกจวัดอุณหภูมิแสดงขึ้นมาบนแดชบอร์ดพาแนลเรียบร้อย

New_Dashboard Connected

Dashboard

Setting

Temperature



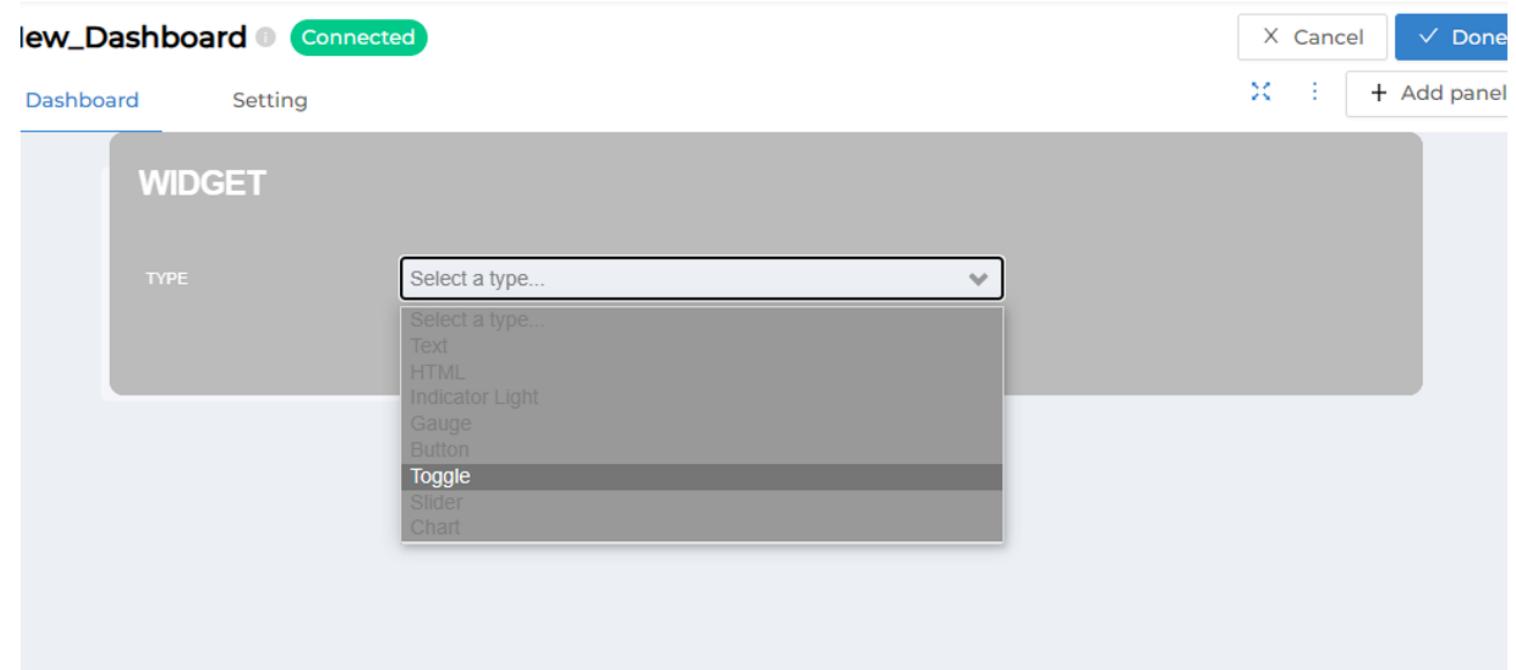
อย่าลืมกดปุ่ม Save ที่ด้านบนขวา

X Cancel

Save

+ Add panel

ทำการสร้างปุ่มกดแบบ Toggle



- เลือก ONTOGGLEON ACTION
(เมื่อสวิตซ์เป็น ON ให้ส่งแมส
เสด ON ไปที่ topic
@msg/operator)
- และ ONTOGGLEOFF ACTION
(เมื่อสวิตซ์เป็น OFF ให้ส่งแมส
เสด OFF ไปที่ topic
@msg/operator)

WIDGET

A simple toggle widget that can perform Javascript action.

TYPE: Toggle

LABEL: LED

TOGGLE ON COLOR: Green

OR

TOGGLE OFF COLOR: Red

OR

TOGGLE ON CAPTION: ON

TOGGLE OFF CAPTION: OFF

TOGGLE STATE:

Add a condition to switch a toggle state here. Otherwise it just toggle by click.

ONTOGGLEON ACTION: `#"device1".publishMsg("operator","ON")`

JS code to run when a toggle is switched to ON

ONTOGGLEOFF ACTION: `#"device1".publishMsg("operator","OFF")`

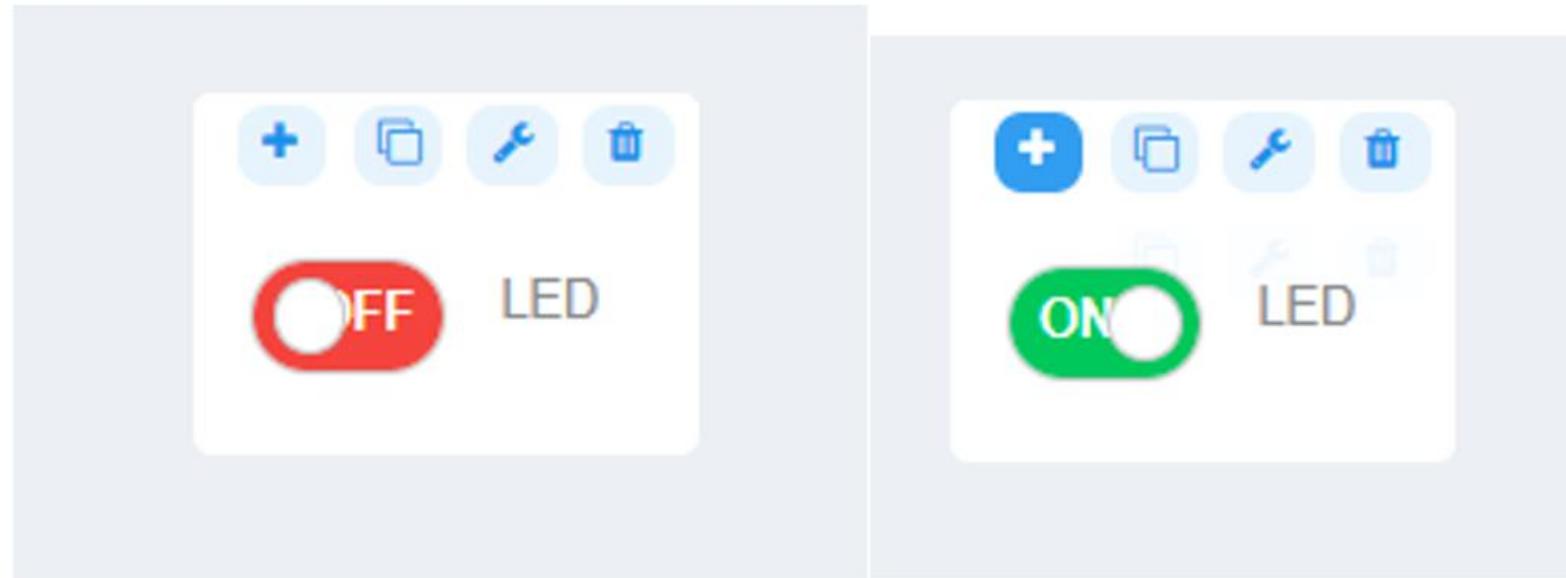
JS code to run when a toggle is switched to OFF

ONINITIAL ACTION:

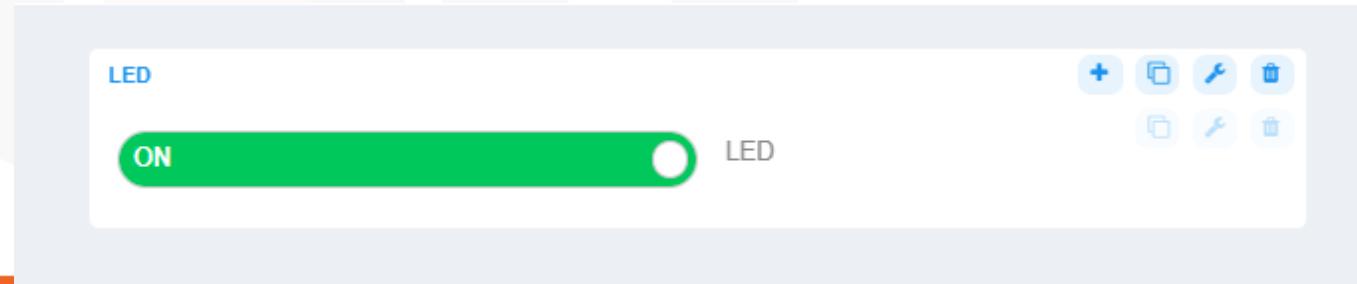
JS code to run after a toggle is created

+ DEVICE + VARIABLE EDITOR

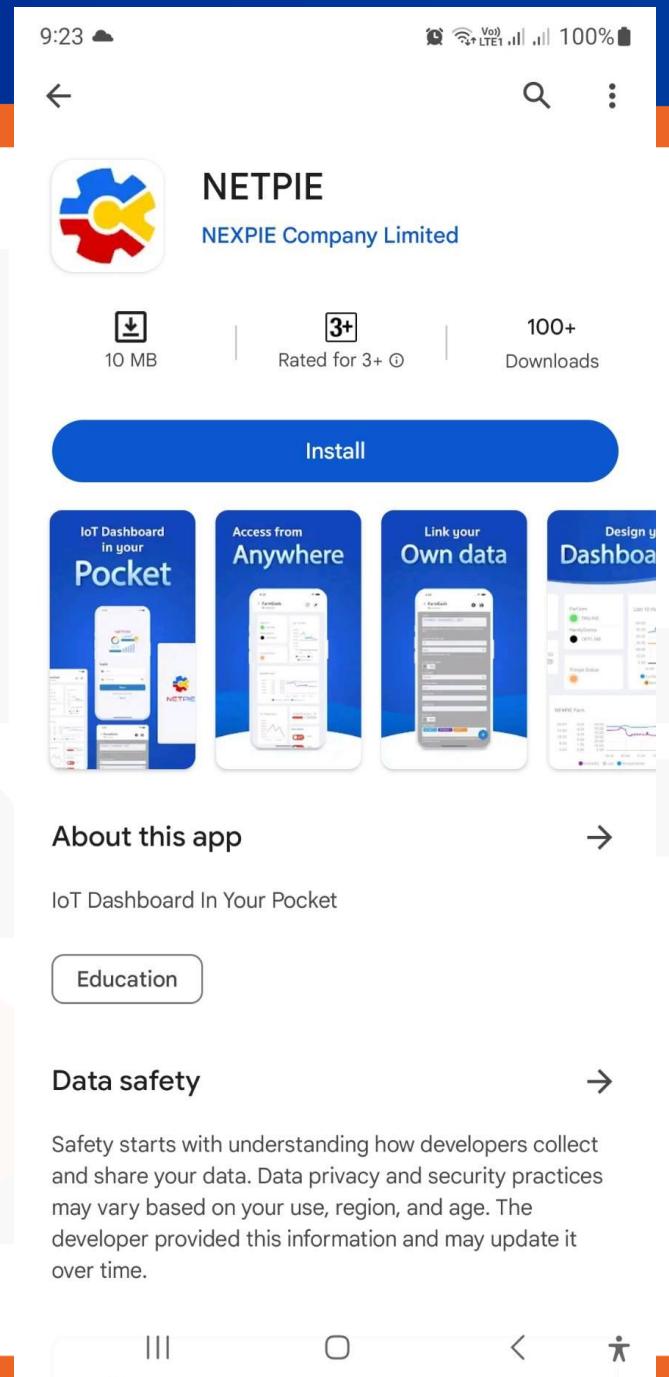
จากนั้นกด Save จะปรากฏปุ่ม Toggle ขึ้นมา



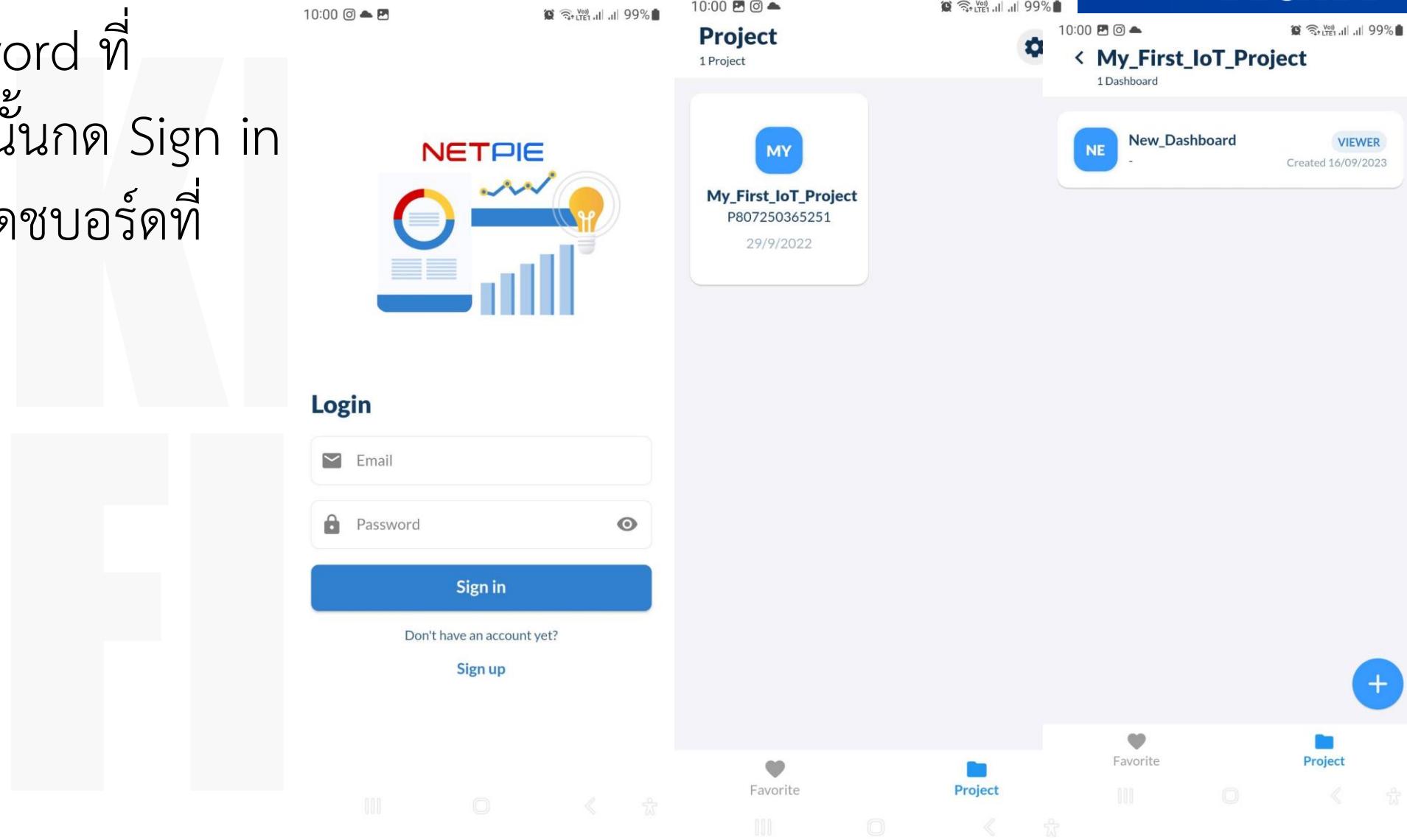
- สามารถกดปุ่มประจำเพื่อปรับขนาด และซื้อพาแนลได้



สามารถดาวน์โหลดแอพพลิเคชัน NETPIE ใน iOS / Android ได้



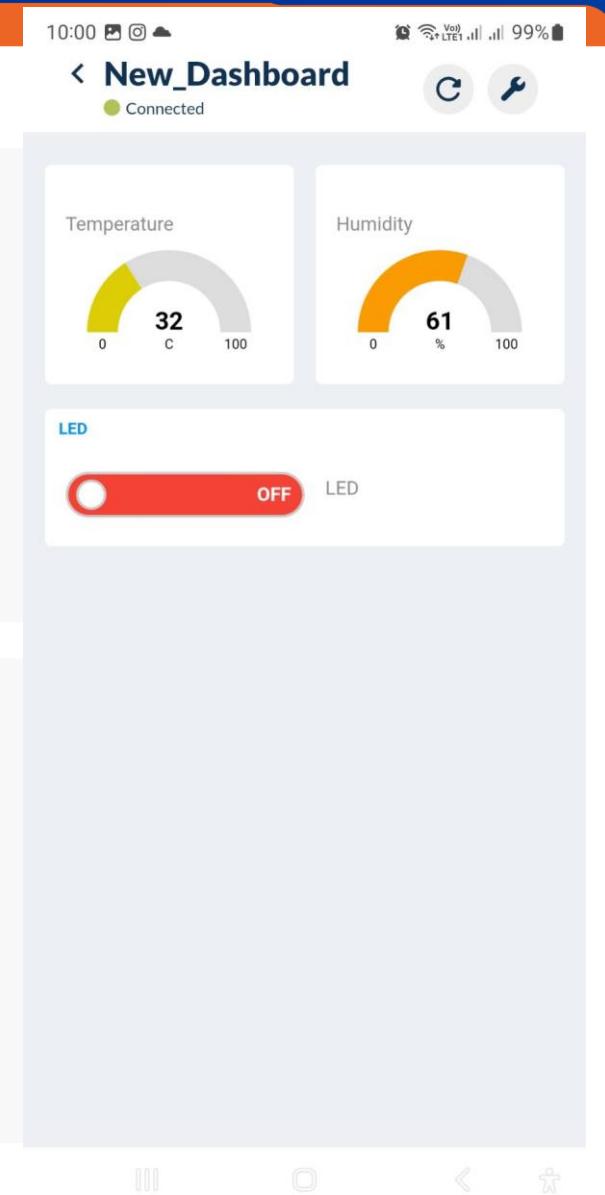
ใส่ Email / Password ที่
ลงทะเบียนไว้ จากนั้นกด Sign in
เลือกโปรเจ็คและเดชบอร์ดที่
สร้างไว้



The screenshots illustrate the NetPie platform's user interface:

- Login Screen:** Shows fields for Email and Password, a Sign in button, and links for "Don't have an account yet?" and "Sign up".
- Project Dashboard:** Displays a summary of "1 Project" with a thumbnail for "My_First_IoT_Project" (P807250365251) created on 29/9/2022.
- Project Detail View:** Shows the "My_First_IoT_Project" dashboard, which includes a "New_Dashboard" card and a "VIEWER" section indicating it was created on 16/09/2023.

เข้าหน้าโปรเจ็ค และเดชบอร์ด
ไปจะเจอ พาเนลต่าง ๆ ที่ได้
สร้างเอาไว้



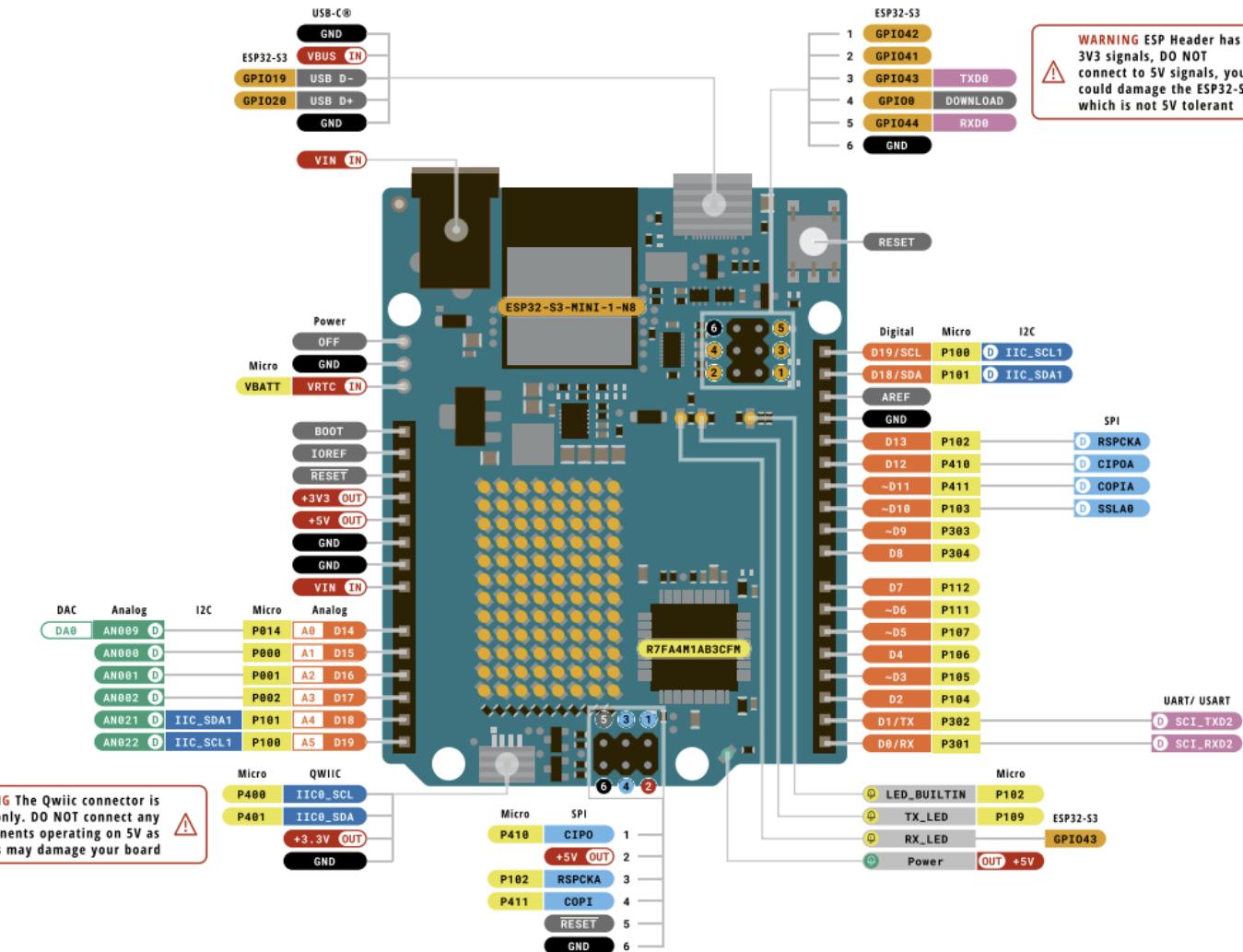
การส่งงาน

- แสดงผลการอ่านค่าเซ็นเซอร์ DHT11 และสั่งเปิด/ปิดไฟ ผ่าน NETPIE แพลตฟอร์ม (เว็บหรือบนสมาร์ทโฟนก็ได้) จากนั้นเรียกอาการร้าย / พี TA ตรวจ
- ทำเซิร์จภายในเวลาเรียนได้คะแนนเต็ม ทำเซิร์จไม่ทันเวลาหักคะแนน

การบ้าน

- ให้เพิ่ม Sensor อีก 1 ชนิด (อะไรมีก็ได้) แล้วแก้ Code แล้วแสดงการอ่านค่าใน NETPIE
- เพิ่มปุ่ม TOGGLE หรือ BUTTON เมื่อกดแล้วให้ LED Matrix แสดงผล
- ถ่ายเป็นคลิปวิดีโอสั้นๆ Upload ลง YOUTUBE ส่งเป็น Private Link ใน Microsoft Teams
- ส่งก่อน 23.59 น. ของวันจันทร์หน้า (23/9/67)

A



References

- Introducing NETPIE2020 & Workshop, Piyawat Jomsathan, Cyber-Physical Systems (CPS)National Electronics and Computer Technology Center (NECTEC), Thailand
- <https://netpie.io/tutorials>



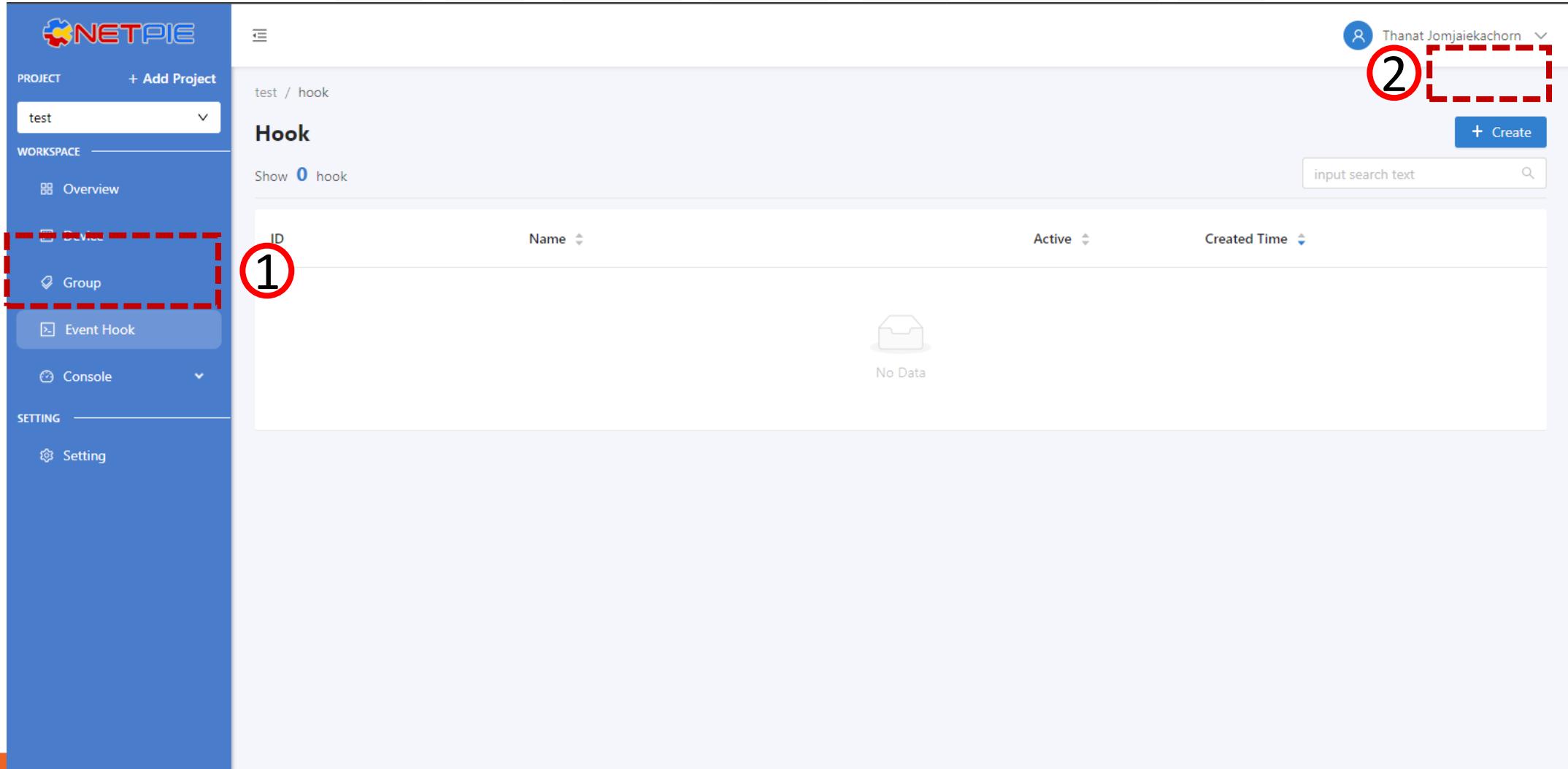
KMITL
สถาบันเทคโนโลยี
พระจอมเกล้า
เจ้าคุณทหารลาดกระบัง

KMITL
FIGHT

การทำระบบแจ้งเตือนผ่าน LINE บน Netpie 2020

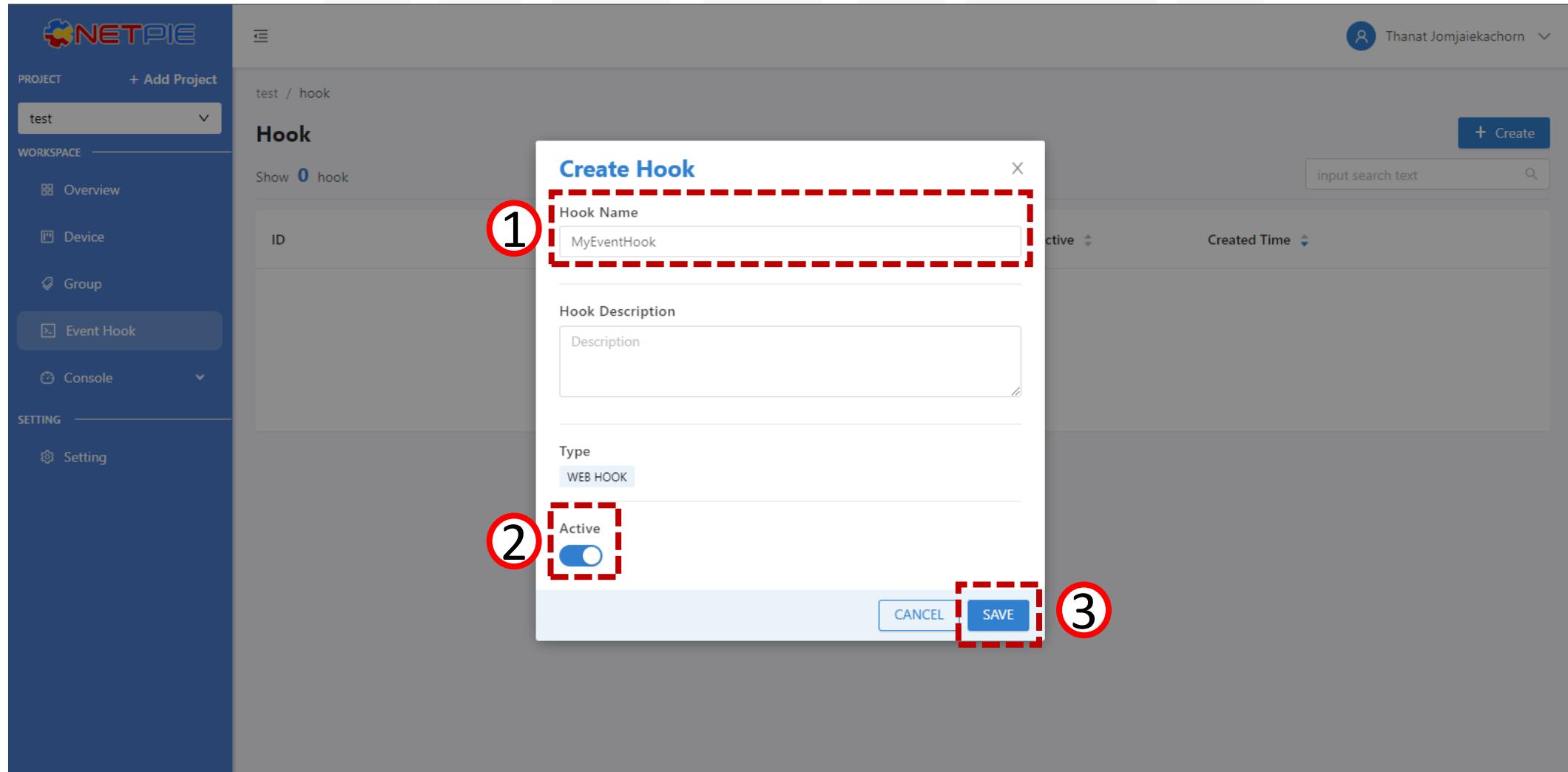


1. ที่หน้า Project ให้ทำการเลือกเมนู Event Hook จากนั้นให้กดปุ่ม Create เพื่อสร้าง Hook ขึ้นมาใหม่

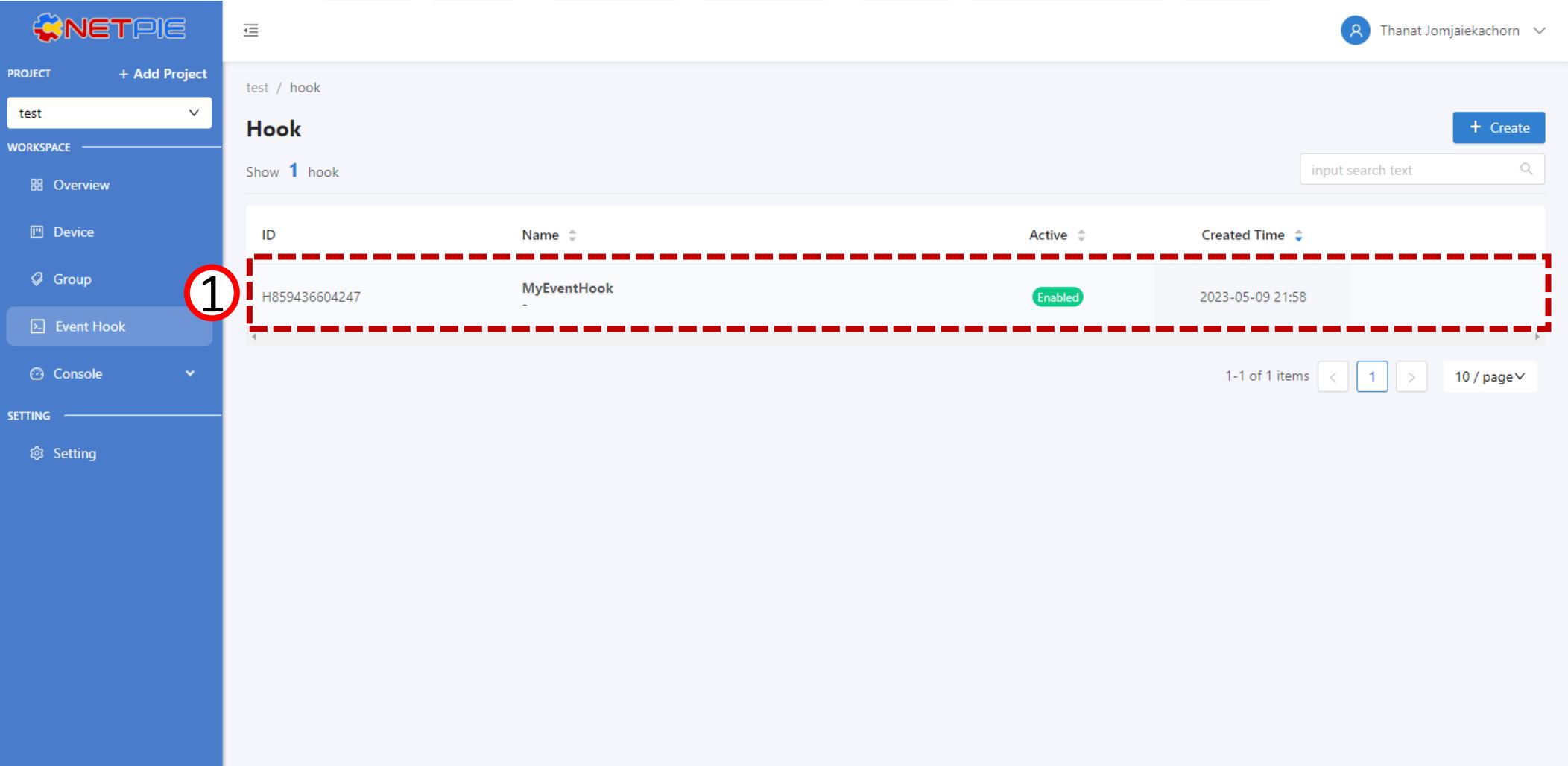


The screenshot shows the NETPIE platform interface. On the left sidebar, under the WORKSPACE section, the 'Event Hook' item is highlighted with a red dashed box and circled with a red number '1'. In the main content area, the 'Hook' page is displayed with a search bar and a 'Create' button. A user profile 'Thanat Jomjaiekachorn' is shown at the top right. The 'Event Hook' table has columns for ID, Name, Active, and Created Time. It currently displays 'No Data'. A red dashed box and circled with a red number '2' covers the entire right side of the screen, from the user profile down to the bottom right corner.

2. ตั้งชื่อใหม่ใน Hook Name โดยใช้ชื่อว่า MyEventHook และกดปุ่ม Active จากนั้นกดปุ่ม Save



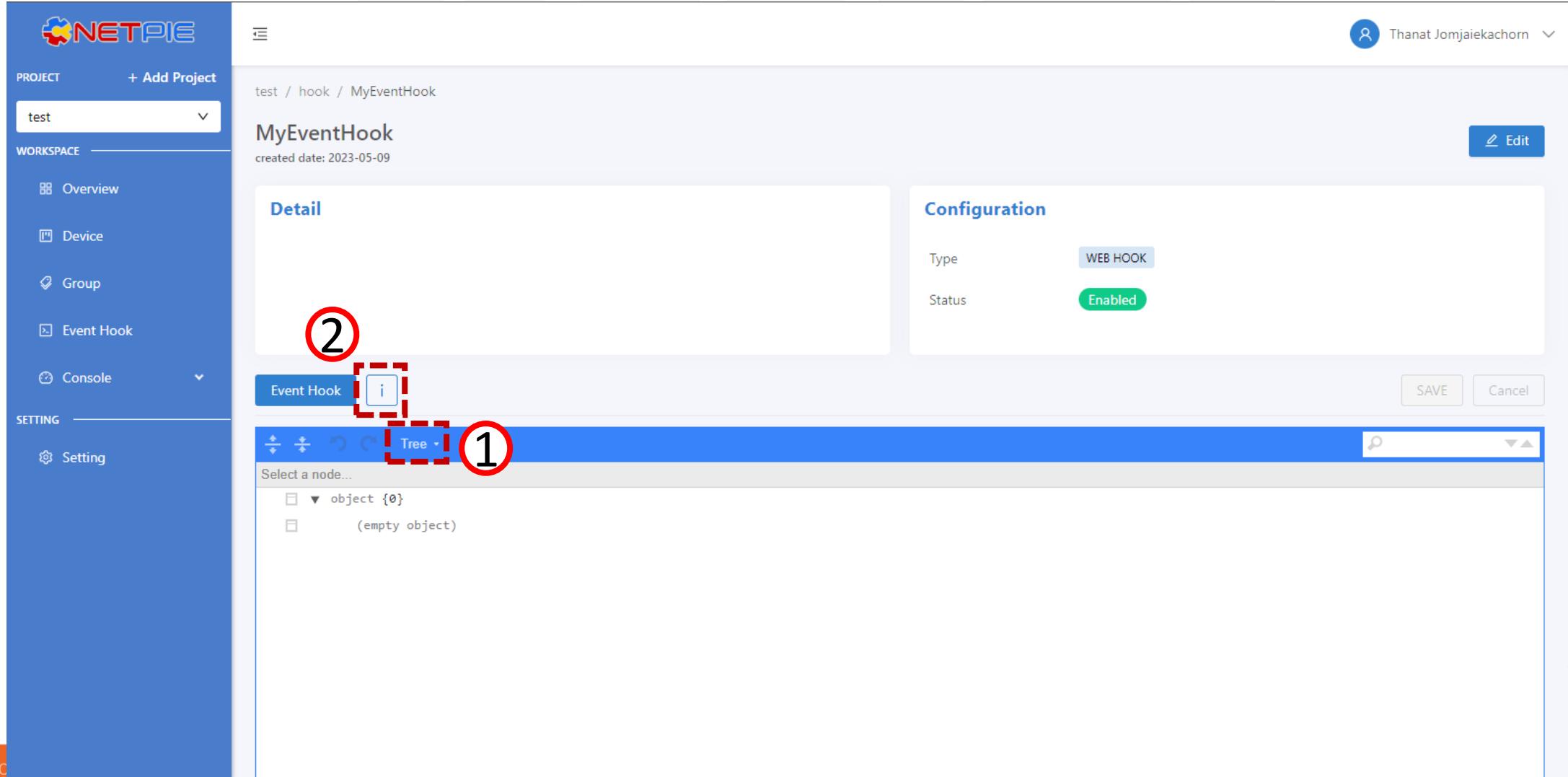
3. จะปรากฏ MyEventHook ที่ได้ทำการสร้างขึ้นมาสำเร็จ จากนั้นคลิกเข้าไปที่ MyEventHook ที่สร้างขึ้นมา



The screenshot shows the NETPIE platform interface. On the left, a sidebar menu includes options like PROJECT, WORKSPACE (Overview, Device, Group, Event Hook), and CONSOLE. The Event Hook option is highlighted with a red circle containing the number '1'. The main area displays a table titled 'Hook' with one item listed. The table columns are ID, Name, Active, and Created Time. The single row shows an ID of H859436604247, a Name of 'MyEventHook', an Active status of 'Enabled', and a Created Time of 2023-05-09 21:58.

ID	Name	Active	Created Time
H859436604247	MyEventHook	Enabled	2023-05-09 21:58

4. จะเข้าสู่หน้า MyEventHook ที่สร้างขึ้นมา จากนั้นที่กดปุ่มเปลี่ยนจาก Tree เป็น code จากนั้นให้กดปุ่มตัว i ที่ใกล้กับปุ่ม Event Hook



The screenshot shows the NETPIE platform interface. On the left, there's a sidebar with 'PROJECT' (test selected), 'WORKSPACE' (Overview, Device, Group, Event Hook selected), and 'SETTING' (Console, Setting). The main area shows a project named 'test / hook / MyEventHook' created on 2023-05-09. The 'Detail' tab is active. On the right, there's a 'Configuration' section with 'Type: WEB HOOK' and 'Status: Enabled'. Below it is an 'Event Hook' button. At the bottom, there's a toolbar with icons for up, down, left, right, and a search bar. A red circle labeled '1' is over the 'Event Hook' button, and another red circle labeled '2' is over the 'i' icon next to it. The tree view below shows 'object {0}' and '(empty object)'.

5. ให้นำข้อความที่ Copy เมื่อสไลเดอร์ก่อนไปวางลงในช่อง Code จากนั้นกดปุ่ม Save

Event Hook i

SAVE Cancel

powered by ace

```
1  {  
2   "body": "message={{context.msg}} {{context.Humidity}}",  
3   "header": {  
4     "Authorization": "Bearer {{context.linetoken}}",  
5     "Content-Type": "application/x-www-form-urlencoded"  
6   },  
7   "method": "POST",  
8   "uri": "https://notify-api.line.me/api/notify"  
9 }  
10
```

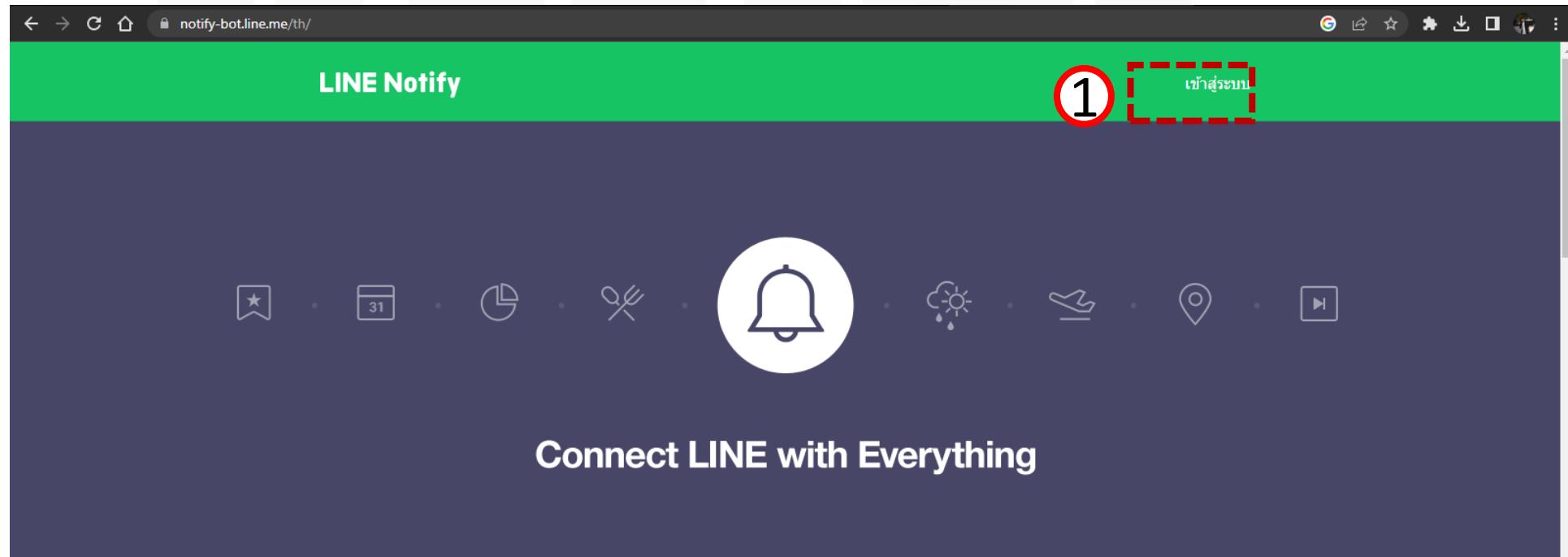
Ln: 10 Col: 1

ที่ແກบด้านล่าง เปลี่ยนเป็น mode code
แล้วทำการแปะ setting

```
{  
  "body": "message={{context.msg}} {{context.Humidity}}",  
  "header": {  
    "Authorization": "Bearer {{context.linetoken}}",  
    "Content-Type": "application/x-www-form-urlencoded"  
  },  
  "method": "POST",  
  "uri": "https://notify-api.line.me/api/notify"  
}
```

แล้วทำการกด save

6. เข้าไปที่เว็บไซต์ <https://notify-bot.line.me/th/> จากนั้นกดปุ่มเข้าสู่ระบบเพื่อเข้าใช้งาน Line Notify



รับการแจ้งเตือนจากเว็บไซต์ LINE

หลังเสร็จสิ้นการเชื่อมต่อกับเว็บไซต์ LINE คุณจะได้รับการแจ้งเตือนจากบัญชีทางการ "LINE Notify" ซึ่งให้บริการโดย LINE
คุณสามารถเชื่อมต่อกับบริการที่หลากหลาย และรับการแจ้งเตือนทางกลุ่มได้ด้วย

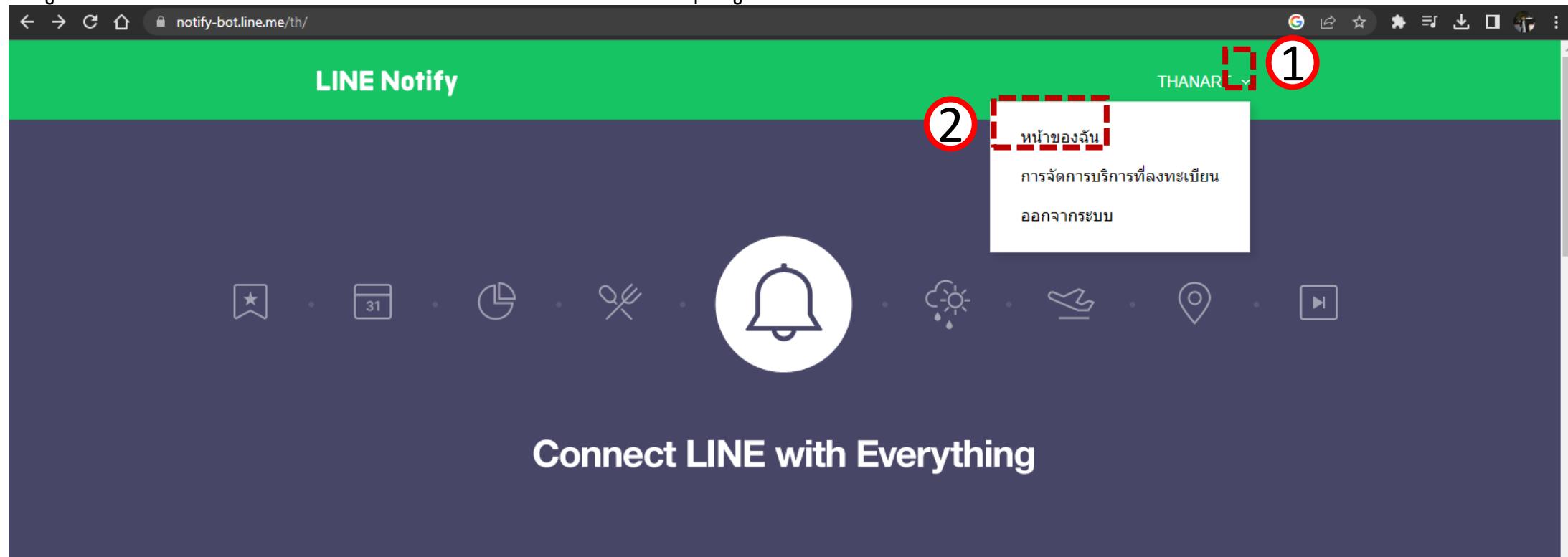
7. ทำการเข้าสู่ระบบโดยใช้ อีเมล และรหัสผ่าน จากนั้นกดปุ่มเข้าสู่ระบบ



8. หลังจากนั้นใส่รหัสยืนยันตัวตนในโทรศัพท์มือถือ



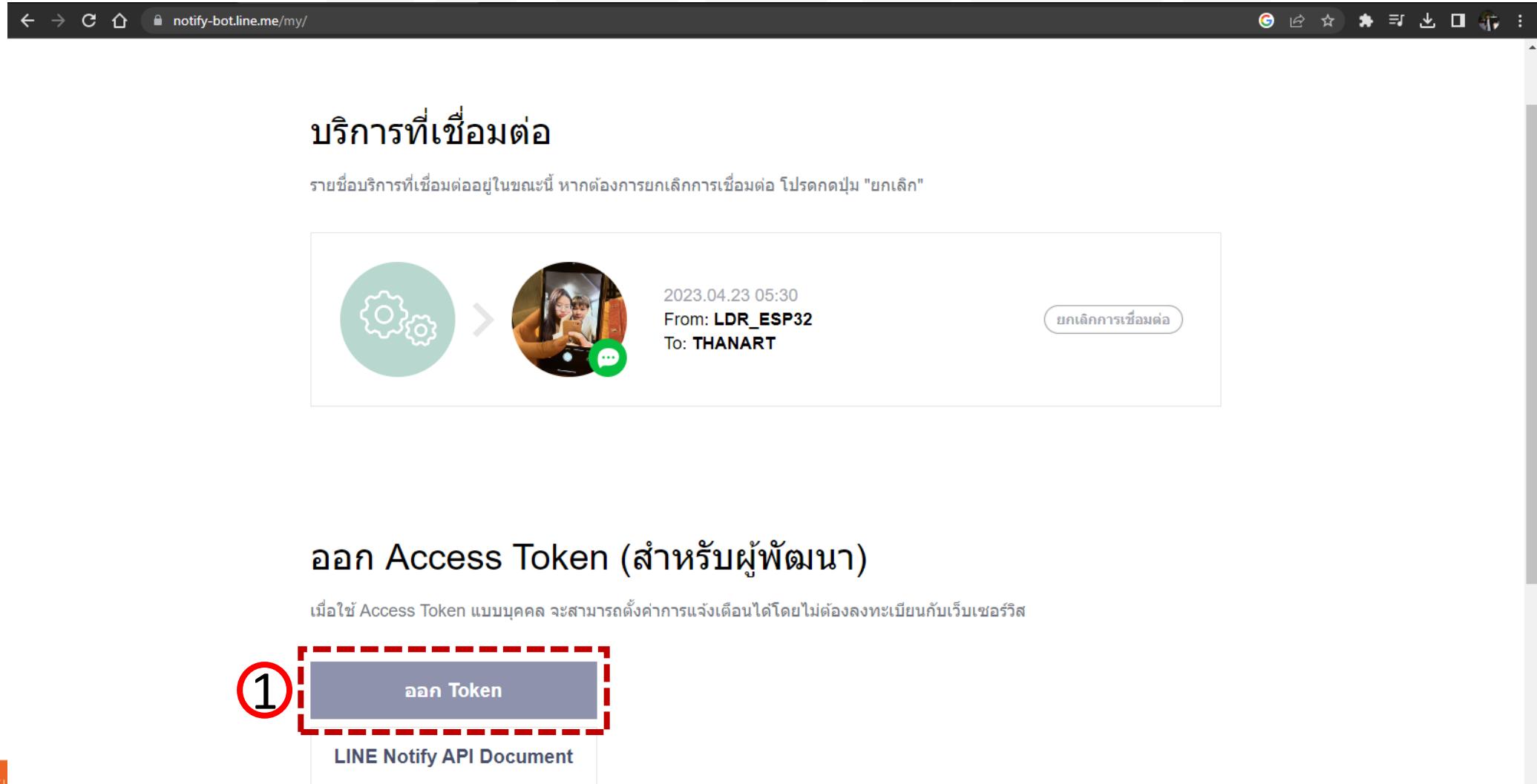
9. จะเข้าสู่หน้าแรกของ Line Notify จากนั้นกดไปที่ปุ่มลูกศรซึ่งลง และเลือกหน้าของฉัน



รับการแจ้งเตือนจากเว็บไซต์ LINE

หลังเสร็จสิ้นการเชื่อมต่อ กับเว็บไซต์ LINE คุณจะได้รับการแจ้งเตือนจากบัญชีทางการ "LINE Notify" ซึ่งให้บริการโดย LINE
คุณสามารถเชื่อมต่อ กับบริการที่หลากหลาย และรับการแจ้งเตือนทางกลุ่มได้ด้วย

10. จะเข้าสู่หน้าแรกของบริการที่เชื่อมต่อ จากนั้นกดปุ่มออก Token



บริการที่เชื่อมต่อ

รายชื่อบริการที่เชื่อมต่ออยู่ในขณะนี้ หากต้องการยกเลิกการเชื่อมต่อ โปรดกดปุ่ม "ยกเลิก"

2023.04.23 05:30
From: LDR_ESP32
To: THANART

ยกเลิกการเชื่อมต่อ

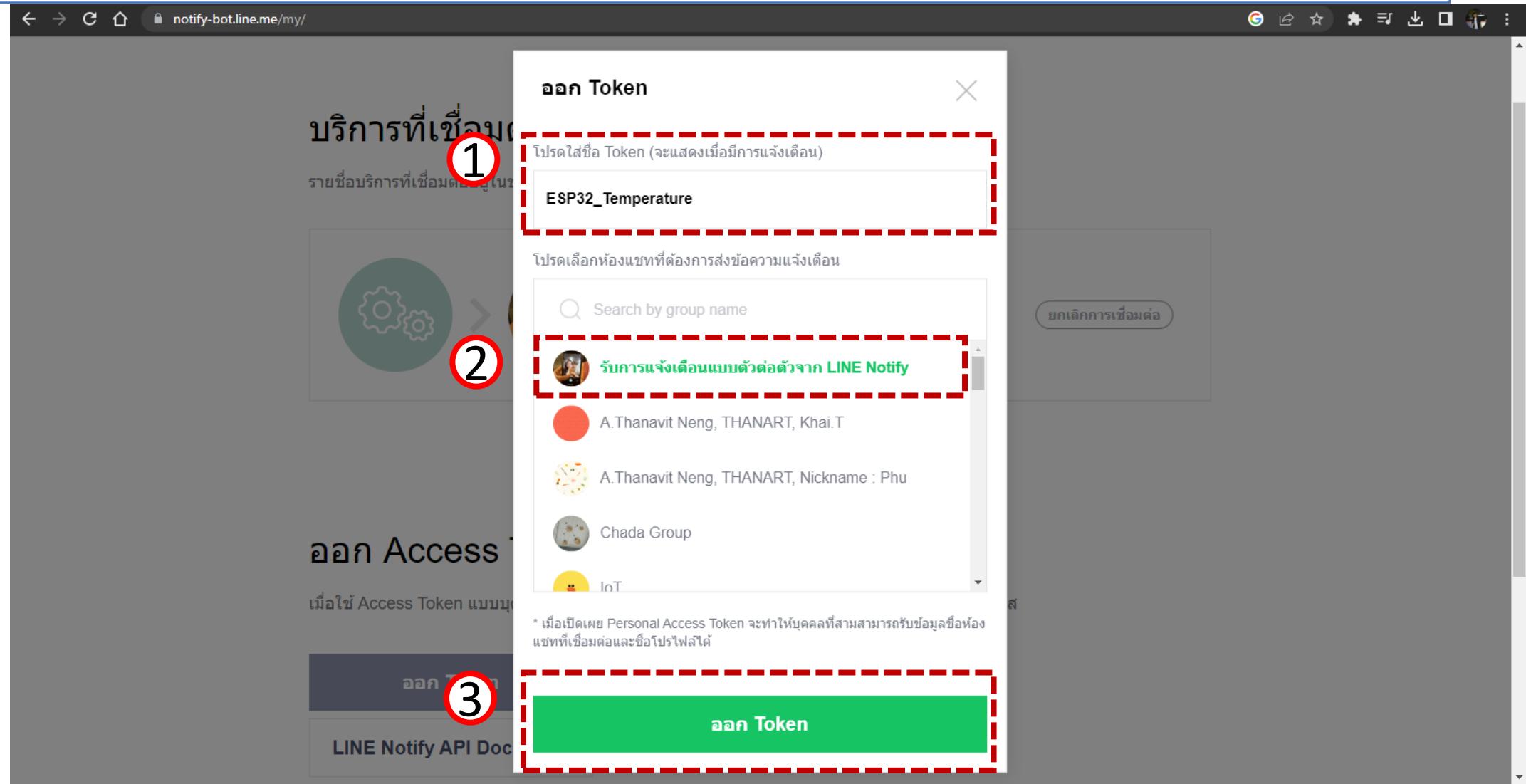
ออก Access Token (สำหรับผู้พัฒนา)

เมื่อใช้ Access Token แบบบุคคล จะสามารถตั้งค่าการแจ้งเตือนได้โดยไม่ต้องลงทะเบียนกับเว็บไซต์

1 **ออก Token**

LINE Notify API Document

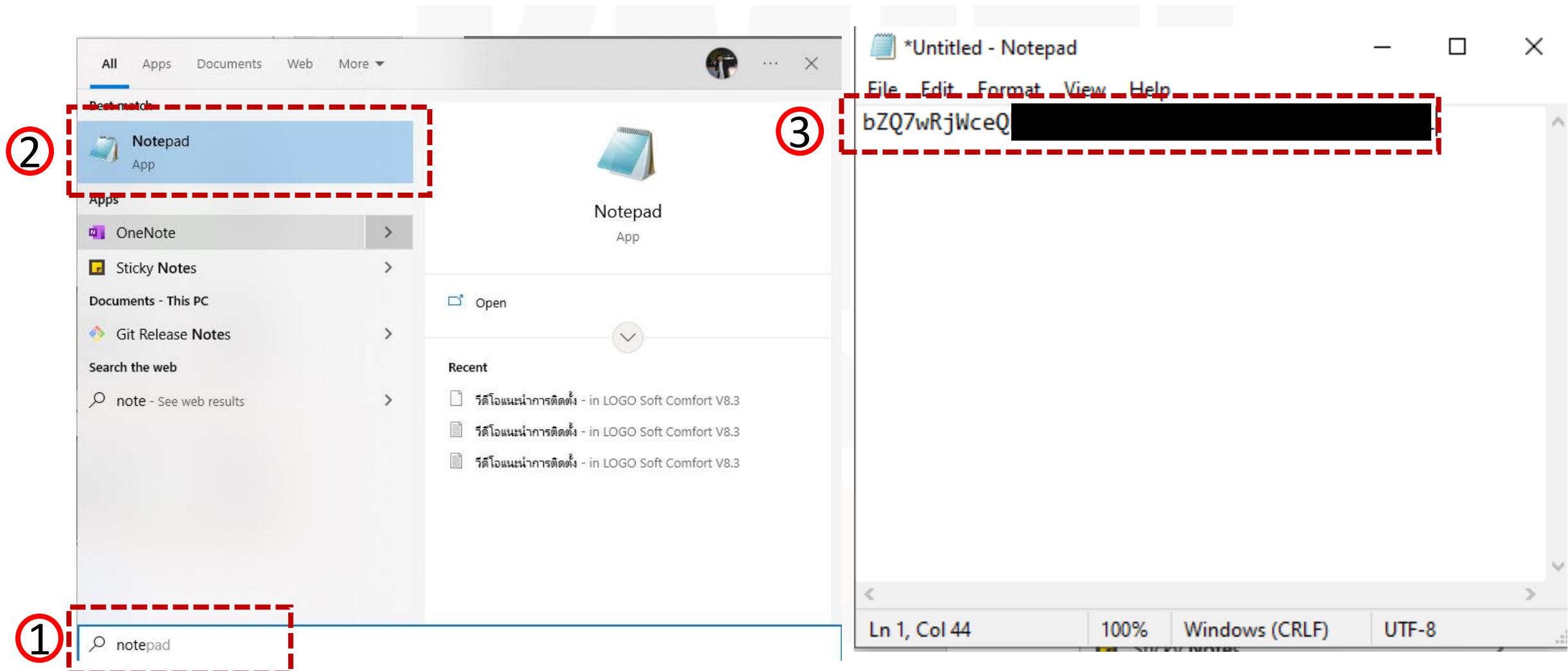
11. จะปรากฏหน้าต่างให้ใส่ชื่อ Token = ESP32_Temperature และห้องแชทให้กดเลือกเป็นรับการแจ้งเตือนแบบตัวต่อตัวจาก LINE Notify จากนั้นกดปุ่มອอก Token



12. จะปรากฏหน้าต่าง Token ที่ออกให้ จากนั้นกดปุ่มคัดลอก หลังจากนั้นกดปุ่มปิดหน้าต่าง หมายเหตุ ถ้าไม่ได้คัดลอก Token และกดปิดหน้าต่างไปแล้ว โดยต้องเริ่มทำใหม่ตั้งแต่ข้อที่ 12 เป็นต้นไป

The screenshot shows a web browser window for 'notify-bot.line.me/my/'. The main page has a dark background with text in Thai. A modal window is open in the center, titled 'Token ที่ออก' (Generated Token). Inside the modal, there is a text input field containing the token 'bZQ7wRjWceQI [REDACTED]'. Below the field, there is explanatory text in Thai: 'ถ้าออกจากหน้านี้ ระบบจะไม่แสดง Token ที่ออกใหม่อีกต่อไป โปรดคัดลอก Token ก่อนออกจากหน้านี้'. At the bottom of the modal, there are two buttons: a green 'คัดลอก' (Copy) button with a red border and a grey 'ปิด' (Close) button. Red circles with numbers '1' and '2' are overlaid on the image: circle 1 is over the 'Copy' button, and circle 2 is over the 'Close' button.

13. หลังจากนั้นเปิด Notepad โดยค้นหาจากปุ่ม Windows ด้านล่างซ้าย หลังจากนั้นนำ LINE_TOKEN ที่คัดลอกมาวาง และ Save ใน Notepad



14. กลับมาที่หน้า Project ใน Netpie 2020 ให้ทำการเลือกเมนู Device จากนั้นคลิกเข้าไปที่ Device ที่ต้องการให้แจ้งเตือน

The screenshot shows the Netpie 2020 application interface. On the left, there is a sidebar with the following sections and items:

- PROJECT**: + Add Project, test
- WORKSPACE**: Overview (highlighted with a red dashed box), Device (highlighted with a red dashed box), Group, Event Hook, Console
- SETTING**: Setting

The main content area is titled "Device" and shows the following details:

- test / device
- Show 1 device
- + Create
- input search text
- Manage Device

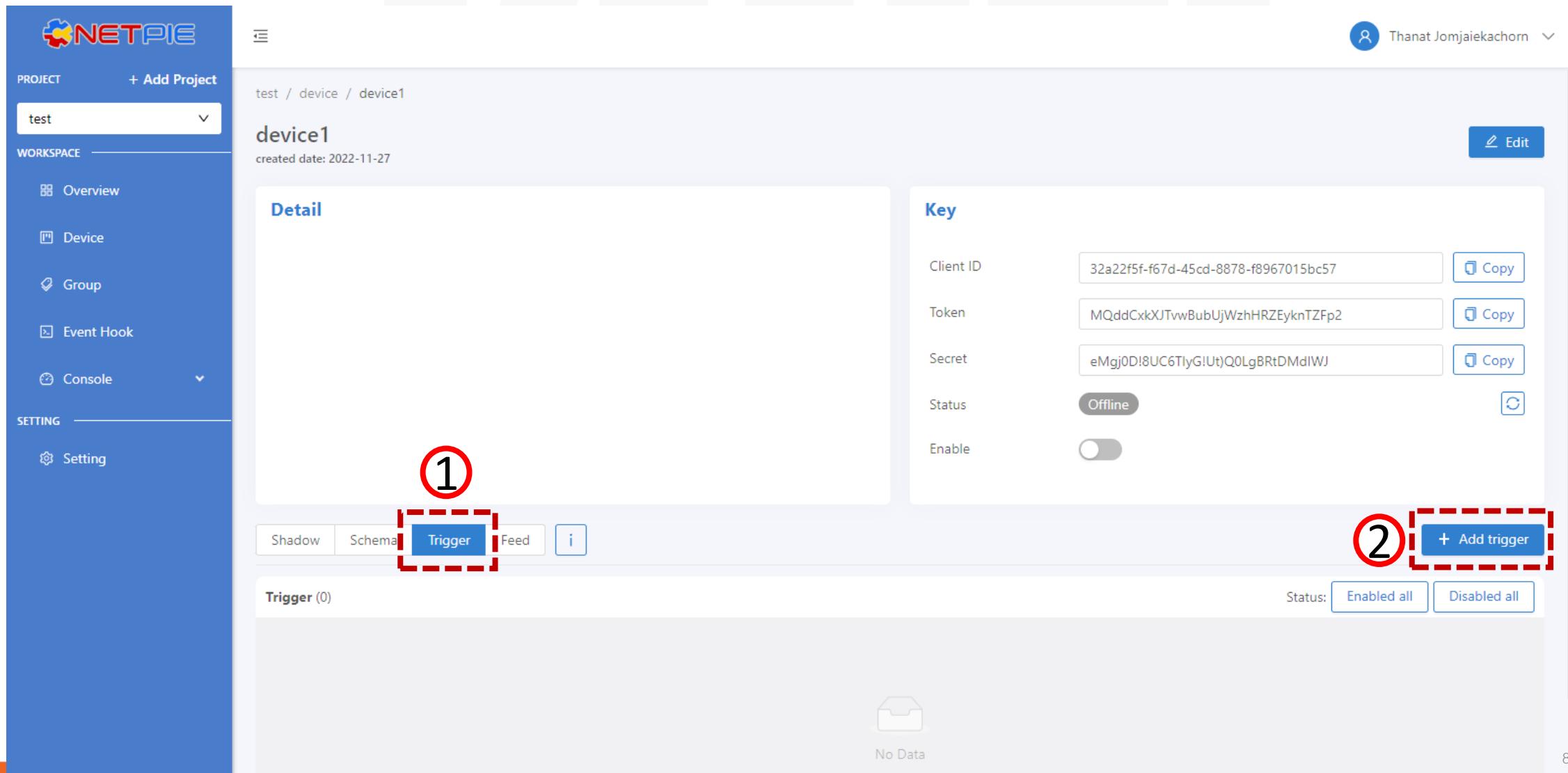
A table displays one device entry:

ID	Name	Group	Status	Created Time
32a22f5f-f67d-45cd-8878-f8967015bc57	device1	-	Offline	2022-11-27 09:02

Two red circles with numbers indicate specific points of interest:

- ① Points to the "Device" item in the sidebar.
- ② Points to the device entry in the table.

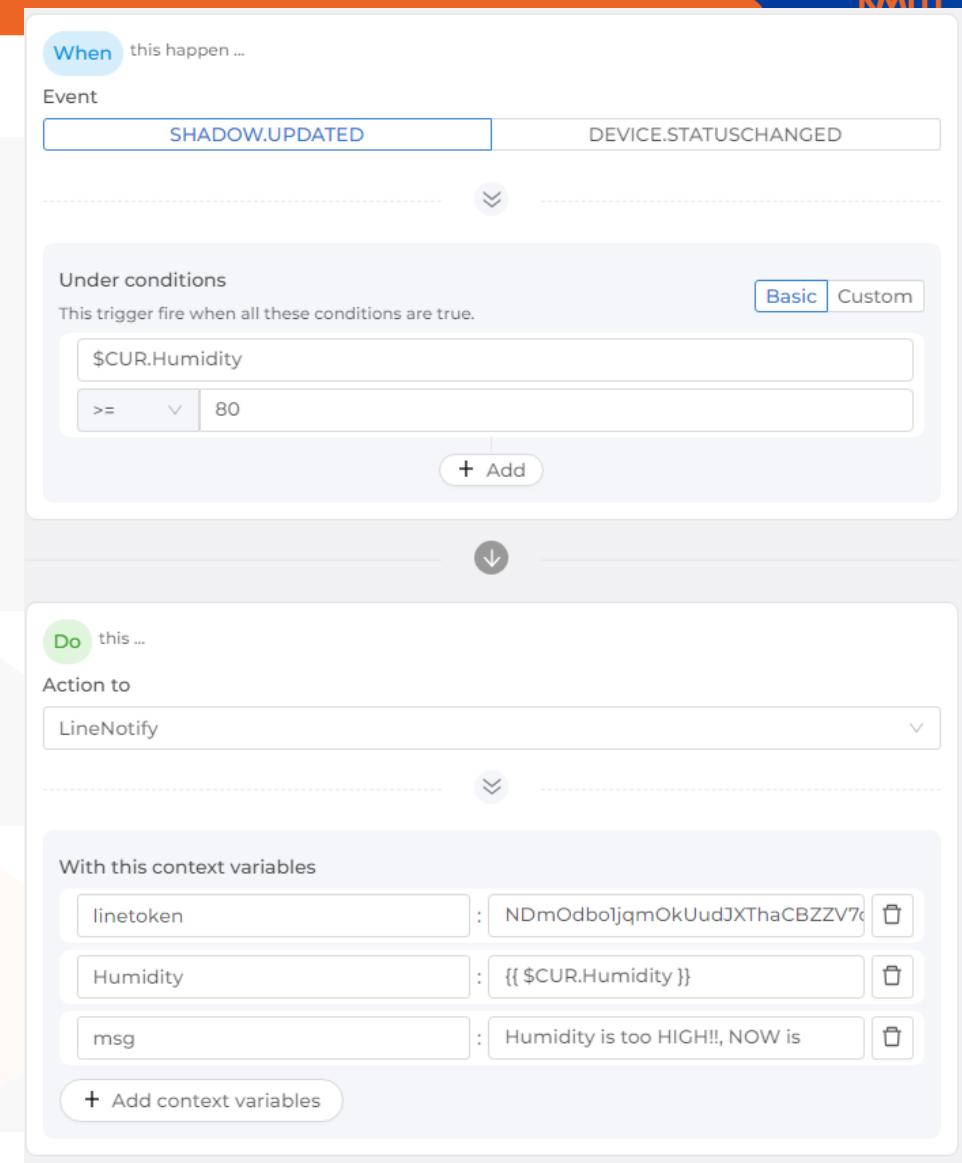
15. ให้เลือกกดปุ่มไปที่ Trigger จากนั้นกดปุ่ม Add Trigger เพื่อสร้าง Trigger ขึ้นมาใหม่



The screenshot shows the NETPIE Platform interface for managing projects and devices. On the left sidebar, under the 'PROJECT' section, 'test' is selected. Under 'WORKSPACE', 'device1' is listed with a creation date of 2022-11-27 and an 'Edit' button. The main area displays the 'device1' detail page. A red circle labeled '1' highlights the 'Trigger' tab in the navigation bar, which is currently active. Another red circle labeled '2' highlights the '+ Add trigger' button located at the bottom right of the 'Trigger' section. The 'Trigger' section shows 'Trigger (0)' and status buttons for 'Enabled all' and 'Disabled all'. To the right, there is a 'Key' section with fields for Client ID, Token, Secret, Status (set to Offline), and an 'Enable' toggle switch.

16. จะปรับกฎหน้าให้ใส่ข้อมูล โดย กดปุ่ม Status

Trigger Title ให้ใส่ข้อความว่า temperature ให้เลือก Event เป็น Shadow.updated จากนั้นใน Under conditions บรรทัดแรกให้ใส่ข้อความว่า \$CUR.Humidity บรรทัดต่อมาให้เลือกเงื่อนไข เช่น ≥ 80



Cancel

Edit

86

17. ต่อมา Action to ให้เลือกเป็น MyEventHook

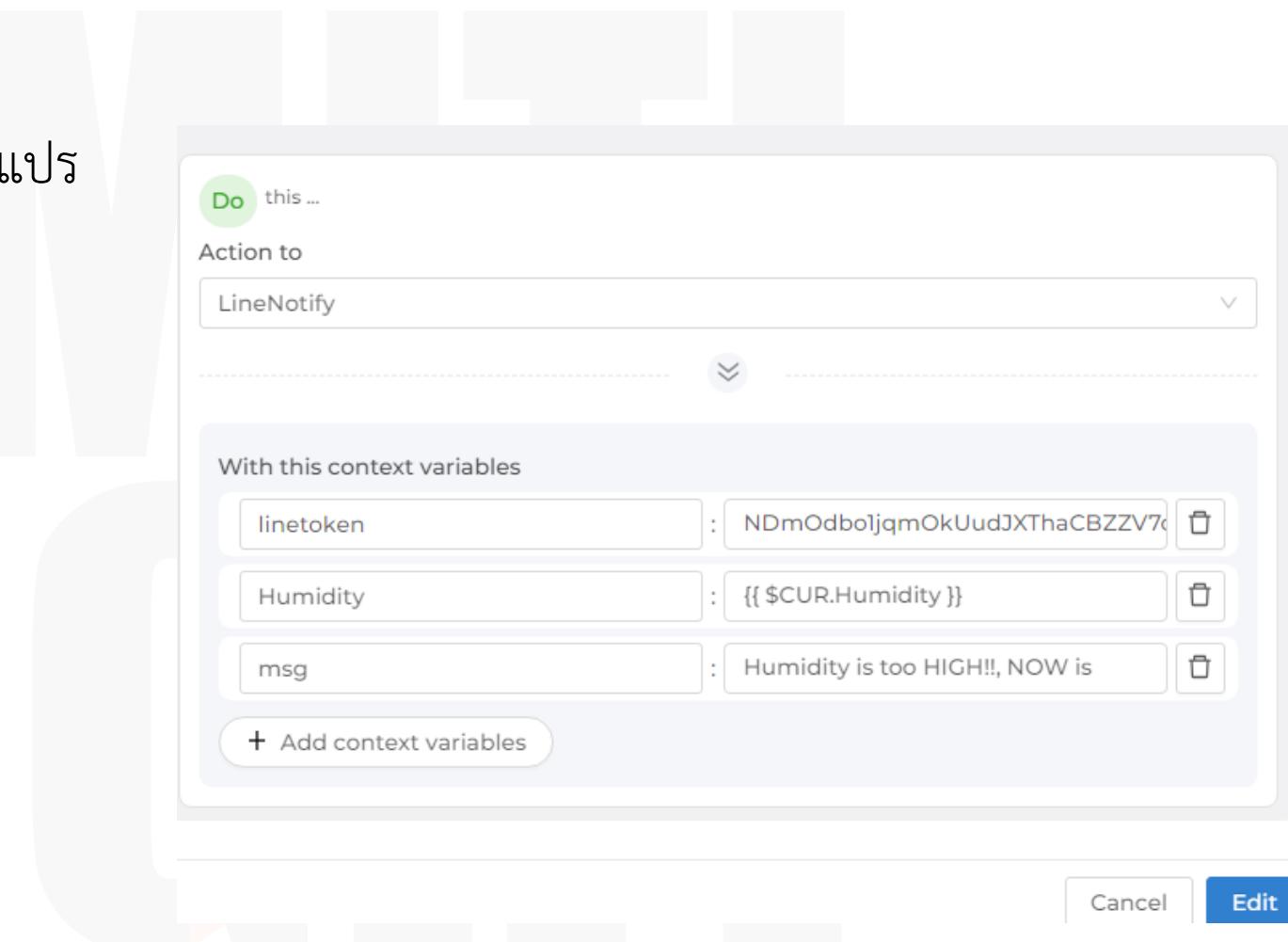
ใน With this context variables มี 3 ตัวแปร โดยการเพิ่มตัวแปรทำได้จากการกดปุ่ม Add context variables ดังนี้

linetoken = LINE_TOKEN

humidity = {{ \$CUR.Humidity }}

msg= SHOW MESSAGE

จากนั้นกดปุ่ม Add



จากรูปด้านบนข้อมูลที่ต้องระบุมีดังนี้

- Trigger ID (string) : รหัสของ Trigger ซึ่งระบบจะสร้างให้อัตโนมัติหรือผู้ใช้ต้องการกำหนดเองก็ได้
- Status : สถานะเปิด/ปิดการใช้งาน Trigger
- Trigger Title (string): ชื่อหรือคำอธินายสั้น ๆ เกี่ยวกับ Trigger
- Event : ประเภทการเปลี่ยนแปลงข้อมูลของ Device (Device Shadow) มี 2 ประเภทให้เลือก ดังนี้
 - `SHADOW.UPDATED` จะเกิด Trigger เมื่อ Device Shadow Data มีการเปลี่ยนแปลงตรงตามเงื่อนไข (`Under conditions`) ที่กำหนดไว้
 - `DEVICE.STATUSCHANGED` จะเกิด Trigger เมื่อ Device เป็นสถานะการเชื่อมต่อ Platform (Online/Offline) และตรงตามเงื่อนไข (`Under conditions`) ที่กำหนดไว้ ซึ่งการกำหนดเงื่อนไขสำหรับ Trigger Event นี้มีได้ 3 รูปแบบ ดังนี้
 - ต้องการให้ Trigger ทุกครั้งที่สถานะการเชื่อมต่อ Platform เป็นไปในรูปแบบ Online เป็น Offline หรือ Offline เป็น Online ให้เช็คเงื่อนไข (`Under conditions`) ให้เป็นจริงเสมอ เช่น `true == true` หรือ `1 == 1` เป็นต้น
 - ต้องการให้ Trigger ในกรณีที่เปลี่ยนสถานะเป็น Online เท่านั้น ให้เช็ค ให้เช็คเงื่อนไข (`Under conditions`) เป็น `$NEW.STATUS == 1`
 - ต้องการให้ Trigger ในกรณีที่เปลี่ยนสถานะเป็น Offline เท่านั้น ให้เช็ค ให้เช็คเงื่อนไข (`Under conditions`) เป็น `$NEW.STATUS == 0`
- Under conditions : เมื่อเลือก `Event` และส่วนนี้ถึงปรากฏ ช่องสามารถเลือก UI ที่จะใช้กำหนดได้ 2 แบบ คือ
 - แบบ Basic จะเป็นฟอร์มให้ระบุเงื่อนไข โดยระบุค่าทั้ง 2 ฝั่ง (ค่าส่วนนี้สามารถใส่ Operators เช่น `>`, `<`, `=`, `>=`, `<=`, `!=`) ที่จะทำการเปรียบเทียบด้วย Comparisons ที่ต้องการ ถ้ามีหลายเงื่อนไขให้คลิกที่ปุ่ม Add และแต่งเงื่อนไขต่อไปด้วยเครื่องหมาย `&&` (and) หรือ `||` (or)
 - แบบ Custom จะเป็นการกำหนดเงื่อนไขแบบพิมพ์ได้เองอิสระ หรือใช้ในกรณีที่เงื่อนไขมีความซับซ้อนมากจน UI แบบ Basic ไม่รองรับในการกรอก การกำหนดเงื่อนไขแบบนี้ ถ้าเงื่อนไขมีความซับซ้อนหรือกรอกไม่ถูกต้อง ระบบจะไม่สามารถแปลงกลับไปเป็นแบบ Basic เพื่อแสดงผลให้ได้ ดังนั้นควรส่งผลให้ผู้ใช้ไม่สามารถคลิกเลือกไปที่แบบ Basic ได้จนกว่าจะปรับแก้ให้ระบบสามารถแปลงกลับ หรือลบข้อมูลที่กรอกไว้ในแบบ Custom ออกหมดก่อนที่จะคลิกกลับไปที่แบบ Basic

การอ้างอิงค่า Shadow ใน Trigger

สำหรับการอ้างอิงค่าตัวแปร Shadow สามารถเรียกใช้ใน Condition หรือ Context Variable ของ Trigger มีรูปแบบดังนี้

- `$CUR.พาธ.ของ.ตัว.แปร` ค่าปัจจุบันล่าสุดที่ถูกอัพเดท (\$NEW merge \$PREV) โดยขึ้นต้นด้วย \$CUR ตามด้วย Path ตามโครงสร้างใน Shadow
- `$NEW.พาธ.ของ.ตัว.แปร` ค่าใหม่ที่ส่งมาอัพเดทลง Shadow โดยขึ้นต้นด้วย \$NEW ตามด้วย Path ตามโครงสร้างใน Shadow
- `$PREV.พาธ.ของ.ตัว.แปร` ค่าก่อนหน้าที่จะถูกอัพเดทลง Shadow โดยขึ้นต้นด้วย \$PREV ตามด้วย Path ตามโครงสร้างใน Shadow

การอ้างอิงตัวแปรอื่น ๆ ภายในระบบ

สำหรับการอ้างอิงค่าตัวแปรที่ระบบมีให้เรียกใช้ใน Condition หรือ Context Variable ของ Trigger มีรูปแบบดังนี้

- `$DEVICEID` รหัสของ Device ที่เป็นเจ้าของ Shadow
- `$ALIAS` ชื่อของ Device ที่เป็นเจ้าของ Shadow
- `$PROJECTID` รหัสของ Project ที่ Shadow สั้งกัด
- `$PROJECTNAME` ชื่อของ Project ที่ Shadow สั้งกัด
- `$GROUPID` รหัสของ Group ที่ Shadow สั้งกัด
- `$GROUPNAME` ชื่อของ Group ที่ Shadow สั้งกัด
- `$NEW.STATUS` รหัสสถานะปัจจุบันของ Device (1 คือ online, 0 คือ offline)
- `$NEW.STATUSTEXT` ข้อความสถานะปัจจุบันของ Device (online คือ เชื่อมต่อ Platform อยู่, offline คือ ไม่ได้เชื่อมต่อ Platform)
- `$OLD.STATUS` รหัสสถานะก่อนหน้าของ Device (1 คือ online, 0 คือ offline)
- `$OLD.STATUSTEXT` ข้อความสถานะก่อนหน้าของ Device (online คือ เชื่อมต่อ Platform อยู่, offline คือ ไม่ได้เชื่อมต่อ Platform)

18. จะ pragqu หน้า Trigger ที่สร้างขึ้นมาใหม่

Trigger (1)

Status: [Enabled all](#) [Disabled all](#)

ID : TD215394478167 | Status :

When

Event : SHADOW UPDATED

Under conditions : \$CUR.Humidity >= 80

Do

Action to : notify

with this context variables :

linetoken : NDmOdbo1jqmOkUudJXThaCBZZV7cb1fit96MXqGchZ1 Humidity : {{ \$CUR.Humidity }}

msg : **Humidity is too HIGH!! , NOW is**

19. ดูใน LINE NOTIFY ว่ามีเด้งแจ้งเตือนขึ้นมาหรือไม่

