Cucumber output  
  
21:38:57.475 -> ..WiFi connected

21:38:59.731 -> MAC Address: 7C:DF:A1:00:AD:6E

21:38:59.731 -> POST to: http://10.133.243.210:8080/mac-text/lookup

21:38:59.816 -> Backend response: {"description":"sarayut 1","token":"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJtYWMiOiI3QzpERjpBMTowMDpBRDo2RSJ9.cBmUn3Xo75Q\_vnjsYOHkgDFLtJuMzmIJfrwRAGM7eM0"}

21:38:59.816 -> Got token: eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJtYWMiOiI3QzpERjpBMTowMDpBRDo2RSJ9.cBmUn3Xo75Q\_vnjsYOHkgDFLtJuMzmIJfrwRAGM7eM0

21:38:59.816 -> Description: sarayut 1

21:38:59.816 -> Connecting MQTT...failed, rc=-2

21:39:07.834 -> Connecting MQTT...failed, rc=-2

**เพิ่มใหม่ทั้งหมด (7 จุดหลัก)**

| **จุด** | **สิ่งที่เพิ่ม** | **ทำอะไร** |
| --- | --- | --- |
| 1 | #include <HTTPClient.h> | ใช้ส่ง HTTP POST ไป backend |
| 2 | const char\* backend\_url = ".../mac-text/lookup" | URL สำหรับขอ token |
| 3 | String description, token; | เก็บข้อมูลจาก backend |
| 4 | bool getMacToken() | ฟังก์ชันเรียก backend ด้วย MAC |
| 5 | เรียก getMacToken() ใน setup() | หลัง WiFi ต่อ → ขอ token ก่อน MQTT |
| 6 | เพิ่ม description, token ใน JSON payload | ส่งไป MQTT ด้วย |
| 7 | ตรวจสอบ token.isEmpty() ใน reconnect() | ถ้าไม่มี token → รีสตาร์ท |

// 1. เพิ่ม library

#include <HTTPClient.h>

// 2. URL backend

const char\* backend\_url = "http://10.133.243.210:8080/mac-text/lookup";

// 3. ตัวแปรเก็บผล

String description, token;

// 4. ฟังก์ชันขอ token

bool getMacToken() {

HTTPClient http;

http.begin(backend\_url);

http.addHeader("Content-Type", "application/json");

StaticJsonDocument<100> doc;

doc["mac\_address"] = sensorId;

String payload;

serializeJson(doc, payload);

int httpCode = http.POST(payload);

if (httpCode == 200) {

String response = http.getString();

Serial.println("Backend response: " + response);

StaticJsonDocument<300> resDoc;

DeserializationError error = deserializeJson(resDoc, response);

if (!error) {

description = resDoc["description"].as<String>();

token = resDoc["token"].as<String>();

Serial.println("Got token: " + token);

return true;

}

}

Serial.println("Failed to get token, HTTP: " + String(httpCode));

http.end();

return false;

}

// 5. เรียกใน setup() หลัง WiFi

if (!getMacToken()) {

Serial.println("Failed to get token, restarting...");

delay(5000);

ESP.restart();

}

// 6. เพิ่มใน payload

doc["description"] = description;

doc["token"] = token;

// 7. ป้องกัน reconnect ถ้าไม่มี token

if (sensorId == "00:00:00:00:00:00" || token.isEmpty()) {

Serial.println("Invalid sensorId/token, restarting...");

ESP.restart();

}

WiFi ต่อ → ดึง MAC →

เรียก backend → ได้ token + description →

สร้าง MQTT topic → ส่งข้อมูลทุก 10 วินาที

**สิ่งที่ ไม่เปลี่ยน**

* DHT, Light sensor, NTP, MQTT reconnect logic
* USE\_LOCAL, WiFi credentials

A screenshot of a computer

AI-generated content may be incorrect.  
Schema เพิ่ม token

A computer screen shot of text

AI-generated content may be incorrect.  
Model เพิ่ม token generator

A screenshot of a computer program

AI-generated content may be incorrect.  
ตั้งค่ากุญแจที่นี่ สำคัญ กุญแจห้ามเปลี่ยนมั่วๆ ต้องสร้างแบบมีหลักการ ให้ใช้ตัวนี้ gen กุญแจ

A screenshot of a computer program

AI-generated content may be incorrect.  
gen กุญแจที่ file python นี้ ใช้ code runner ในการ run เหมือนเดิม

A screen shot of a computer code

AI-generated content may be incorrect.  
controller ไปเรียก token generator จาก model มา run

สรุปหลักการของ web server  
model <- 🡪 controller

ทั้ง model controller จะมี schema คอยประกบ check datatype