

# Assignment #1 Create an API Server with Node.js

สร้าง API Server (ถ้า deploy ให้รันบน cloud ได้มีคะแนนพิเศษ)

โดยเราจะต้องใช้บริการจาก 2 servers นี้

- Server1: Drone Config Server
  - [https://script.google.com/macros/s/AKfycbzwclqJRodyVjzYyY-NTQDb9cWG6Hoc5vGAABVtr5-jPA\\_ET\\_2lasrAJK4aeo5XoONiaA/exec](https://script.google.com/macros/s/AKfycbzwclqJRodyVjzYyY-NTQDb9cWG6Hoc5vGAABVtr5-jPA_ET_2lasrAJK4aeo5XoONiaA/exec)
- Server 2: Drone Log Server (PocketBase)
  - [https://app-tracking.pockethost.io/api/collections/drone\\_logs/records](https://app-tracking.pockethost.io/api/collections/drone_logs/records)
  - API Document ของ PocketBase อยู่ที่ <https://pocketbase.io/docs/api-records/>
    - หัวข้อ List/Search records
    - หัวข้อ Create records

## HTTP Methods ที่จะต้องรองรับ

- GET /configs/yourDroneId
  - อธิบายการทำงาน
    - รับ input parameter จาก request เป็นเลข id ของ drone มา
    - GET เรียกขอข้อมูลจาก Server1 (Drone Config Server)
    - ให้ response กลับไปเป็น JSON ของข้อมูล config เฉพาะของ drone id นั้น
    - ข้อมูลใน config ให้มีเฉพาะ drone\_id, drone\_name, light, country, weight
  - ตัวอย่างข้อมูล response
    - ```
{
  drone_id: 3001,
  drone_name: "Dot Dot",
  light: "on",
  country: "India",
  weight: 21
}
```
- GET /status/yourDroneId
  - อธิบายการทำงาน
    - รับ input parameter จาก request เป็นเลข id ของ drone มา
    - GET เรียกขอข้อมูลจาก Server1 (Drone Config Server)
    - ให้ response กลับไปเป็น JSON ของข้อมูล config เฉพาะของ drone id นั้น
    - ข้อมูลใน config ให้มีเฉพาะ condition
  - ตัวอย่างข้อมูล response
    - ```
{
  condition: "good"
}
```
- GET /logs/yourDroneId
  - อธิบายการทำงาน
    - รับ input parameter จาก request เป็นเลข id ของ drone มา
    - GET เรียกขอข้อมูลจาก Server2 (Drone Log Server)
    - ให้ response กลับไปเป็น JSON Array ของข้อมูล logs เฉพาะของ drone id นั้น เรียงจาก created ล่าสุดขึ้นก่อน และจำกัด 25 รายการ
    - ข้อมูลใน log ให้มีเฉพาะ drone\_id, drone\_name, created, country, celsius
  - ตัวอย่างข้อมูล response
    - ```
[
  {
    drone_id: 3001,
    drone_name: "Dot Dot",
    created: "2024-09-22 07:37:32.111Z",
```

```

        country: "India",
        celsius: 45
    },
    {
        drone_id: 3001,
        drone_name: "Dot Dot",
        created: "2024-09-22 07:37:57.411Z",
        country: "India",
        celsius: 46
    }
]

```

- POST /logs
  - อธิบายการทำงาน
    - รับข้อมูล JSON ที่ส่งมาใน request body นำไปสร้างข้อมูล log ด้วยการเรียกใช้บริการของ Server2 (Drone Log Server)
    - ข้อมูลที่ส่งไปให้ Server 2 (Drone Log Server) สร้าง log record ให้ส่งเฉพาะ drone\_id, drone\_name, country, celsius

## Assignment #2 Create a Web Client with HTML, CSS, and JavaScript

สร้าง web client (ถ้า host บน cloud ได้มีคณณพิเศษ, ถ้า UI สวยงามมีคณณพิเศษ)

จะใช้ UI Framework / Library รวมด้วยก็ได้ไม่บังคับ

- Features
  - Config: .env file
    - เก็บเลข drone id ไว้ใน .env ไฟล์
    - ตัวอย่างเช่น
      - DRONE\_ID=3001
  - Page #1: View Config
    - เป็นหน้าแสดงข้อมูล Config ของ drone id ที่กำหนดไว้ใน .env ไฟล์
    - ขอข้อมูล Config จาก API Server ที่สร้างไว้ใน Assignment #1
    - ข้อมูล Config เก็บไว้สำหรับใช้ตอน Page #2 และ Page #3 ด้วย
    - แสดงข้อมูลในหน้า web page ดังนี้
      - Drone ID
      - Drone Name
      - Light
      - Country
  - Page #2: Temperature Log Form
    - แสดง Form เปล่าสำหรับให้ user ป้อนข้อมูล
      - Input: Temperature in celsius
      - Button: Submit data
    - เมื่อ user กดปุ่ม Submit data ให้ใช้บริการ API Server ของ Assignment #1
      - POST /logs
      - ข้อมูลที่ส่งไปให้สร้าง log record ให้ส่งเฉพาะข้อมูลดังนี้
        - drone\_id, drone\_name, country (ที่ได้มาจากค่า Config)
        - celsius (ได้มาจากค่าตัวเลขที่ user กรอกมาใน Form)
  - Page #3: View Logs
    - ใช้ <table>
    - แสดงรายการ Log เฉพาะของ drone id โดยเรียงตามเวลา created ล่าสุดขึ้นก่อน, จำกัด 25 รายการ
    - แต่ละรายการ Log แสดงข้อมูลเป็น 1 row ใน table
    - แสดงข้อมูล 5 columns เช่น
      - Created: "2024-09-22 07:37:32.111Z"
      - Country: "Pakistan"
      - Drone ID: 3001
      - Drone Name: "Dot Dot"
      - Celsius: 45