

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

สร้าง API Server

- วาง Code และไฟล์ต่างๆ ของ project ใน Github
 - Project root ต้องมีไฟล์ README.md อธิบายเกี่ยวกับ project รวมถึงวิธีการ run ด้วย
- Deploy ให้สามารถ run และถูกเรียกใช้ได้นบน cloud host (สามารถเลือกใช้เจ้าไหนก็ได้)
- ใช้ environment variables และเก็บ Config ต่างๆ ในไฟล์ .env หรือ .env.local
 - ให้เก็บ API Token, URL, และ Path ที่ใช้ในการเรียก API Server อื่น ใน .env file (และให้เรียกใช้ใน code ได้จาก process.env) เช่น
 - LOG_URL=https://app-tracking.pockethost.io/api/collections/drone_logs/records
 - LOG_API_TOKEN=xxxxx
 - (หมายความว่าห้ามให้เห็นเนื้อของ API Token โดยตรงใน code)

ในการทำงาน API Server ของเราจะต้องเรียกใช้บริการจาก 2 servers นี้อีกต่อหนึ่ง:

- Server1: Drone Config Server
 - **API Endpoint :**
https://script.google.com/macros/s/AKfycbzwclqJRodyVjzYyY-NTQDb9cWG6Hoc5vGAABVtr5-jPA_ET_2lasrAJK4aeo5XoONiaA/exec
- Server 2: Drone Log Server
 - **API Endpoint:** https://app-tracking.pockethost.io/api/collections/drone_logs/records
 - **API Token (Bearer Token):** 20250901efx
 - ศึกษาวิธีการเรียกใช้งานได้จาก Document นี้: <https://pocketbase.io/docs/api-records/>
 - หัวข้อ List/Search records
 - หัวข้อ Create records

Path และ HTTP Methods ที่ API ของเราจะต้องมี

- GET /configs/{droneId}
 - การทำงาน
 - 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, weigh
 - ตัวอย่างหน้าตาของ response เช่น
 - {
 drone_id: 3001,
 drone_name: "Dot Dot",
 light: "on",
 country: "India",
 weight: 21
 }
- GET /status/{droneId}
 - 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
 - ตัวอย่าง response
 - {
 condition: "good"
 }

- GET /logs/{droneId}

- 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 และจำกัดจำนวนรายการที่ออกมาในผลลัพธ์ที่ 12 รายการ
 - หากสามารถทำ pagination ได้ มีคะแนนพิเศษ
 - ข้อมูลใน log ให้มีเฉพาะ drone_id, drone_name, created, country, celsius

- ตัวอย่าง 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 ตามที่ส่งมาใน request body นำไปสร้างข้อมูล log record ใน Server2 (Drone Log Server)
 - ข้อมูลที่ส่งไปให้ Server 2 (Drone Log Server) ทำการสร้าง log record ให้มีแค่เฉพาะ drone_id, drone_name, country, celsius

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

สร้าง Web Frontend

- ภาษาเลือกใช้ JavaScript หรือ TypeScript ก็ได้
- จะใช้ React.js หรือ Next.js ด้วยก็ได้ไม่บังคับ
- วาง Code และไฟล์ต่างๆ ของ project ใน Github
 - Project root ต้องมีไฟล์ README.md อธิบายเกี่ยวกับ project รวมถึงวิธีการ run ด้วย
- Deploy ให้สามารถ run และถูกเรียกใช้ได้นบน cloud host (สามารถเลือกใช้เจ้าไหนก็ได้)
- **มีคะแนนพิเศษหากทำ UI ได้สวยงาม เข้าใจและใช้งานง่าย** (จะใช้ UI Library ที่มีคนอื่นสร้างไว้แล้วก็ได้)
- ใช้ environment variables และเก็บ Config ต่างๆ ในไฟล์ .env หรือ .env.local
 - ให้เก็บเลข drone id ของเรา ใน .env file และเรียกใช้ใน code จาก process.env
 - เช่น DRONE_ID=3001
 - (หมายความว่าห้ามมีเลข drone id ในเนื้อ code)
- การทำงานเบื้องหลัง ให้เรียกใช้ API ที่เราทำไว้ใน Assignment #1
 - ไม่ให้เรียกบริการจริงๆจาก Drone Config Server และ Drone Log Server ของอาจารย์

Features

- Page #1: View Config
 - เป็นหน้าแสดงข้อมูล Config ของ drone id ที่กำหนดไว้ใน .env ไฟล์
 - ขอข้อมูล Config จาก API Server ที่สร้างไว้ใน Assignment #1
 - ข้อมูล Config ต้องเก็บไว้สำหรับใช้ตอน 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 12 รายการ
 - **หากสามารถทำ pagination ได้ มีคะแนนพิเศษ**
 - แต่ละรายการ Log record แสดงข้อมูลเป็น 1 row ใน table
 - ให้ข้อมูล 5 columns ดังนี้
 - Created
 - Country
 - Drone ID
 - Drone Name
 - Celsius