

บทที่ 5

RESTful API Design Best Practices

หัวข้อที่จะเรียนรู้

- หลักการออกแบบ RESTful API
- การตั้งชื่อ Resource และ Endpoint
- การจัดการ Parameters, Filtering, Pagination
- การจัดการ Error Response
- การทำเอกสาร API ด้วย Swagger/OpenAPI

หลักการของ RESTful API

- ใช้ Resource-oriented URIs
- ใช้ HTTP Methods แทน Action
- Stateless → ไม่มี session state ฝั่ง server
- ใช้ HTTP Status Codes อย่างถูกต้อง
- Response ควรเป็น JSON ที่อ่านง่าย

การตั้งชื่อ Resource

✓ ควรใช้ คำนามพหูพจน์ (Plural Nouns)

- `/users` , `/books` , `/orders`

✓ ใช้ hierarchical structure

- `/users/1/orders`

✗ หลีกเลี่ยง Verb ใน URL

- `/getUser` , `/createBook`

การใช้ HTTP Methods

Method	Action	Example
GET	Read	/users/1
POST	Create	/users
PUT	Update/Replace	/users/1
PATCH	Partial Update	/users/1
DELETE	Delete	/users/1

Filtering & Pagination

Query Parameters:

- `/books?author=Alice`
- `/users?role=admin&active=true`

Pagination:

- `/users?page=2&size=20`
- Response ควรรีคืน `total` , `page` , `size`

Error Handling

ควรคืน Response ที่เป็น JSON มาตรฐาน

ตัวอย่าง:

```
{  
  "status": 404,  
  "error": "Not Found",  
  "message": "User with id=99 not found",  
  "timestamp": "2025-09-30T10:20:00"  
}
```

API Versioning

- ใช้ใน Path: `/api/v1/users`
- หรือใช้ใน Header: `Accept: application/vnd.myapp.v1+json`
- สำคัญเมื่อมีการเปลี่ยน API ที่ไม่ backward-compatible

Swagger / OpenAPI

- **Swagger/OpenAPI** = มาตรฐานเอกสาร API
- สามารถ:
 - อธิบาย Endpoints, Parameters, Responses
 - สร้าง UI สำหรับทดลอง API
- Spring Boot ใช้ **springdoc-openapi** หรือ **Swagger UI**

Spring Boot + Swagger Example

Dependency (Maven):

```
<dependency>  
  <groupId>org.springdoc</groupId>  
  <artifactId>springdoc-openapi-starter-webmvc-ui</artifactId>  
  <version>2.2.0</version>  
</dependency>
```

Swagger UI:

- รันแล้วเปิด → <http://localhost:8080/swagger-ui.html>

ตัวอย่าง Swagger Spec

```
paths:
  /users:
    get:
      summary: Get all users
      responses:
        '200':
          description: OK
    post:
      summary: Create new user
      requestBody:
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/User'
```

Lab

1. ออกแบบ API สำหรับ **Library Management System**

- `/books` , `/members` , `/borrows`
- รองรับ CRUD + Filtering

2. เพิ่ม Swagger UI ให้ Spring Boot Project

3. ตรวจสอบการทำงานผ่าน Swagger UI

Assignment

- พัฒนา API Library Management:
 - `/books` (GET/POST/PUT/DELETE)
 - `/members` (CRUD)
 - `/borrows` (ยืม-คืนหนังสือ)
- ส่ง:
 - Source code
 - Swagger/OpenAPI Documentation
 - ตัวอย่าง Response JSON