

บทที่ 2

HTTP & RESTful API Basics

หัวข้อที่จะเรียนรู้

- พื้นฐาน HTTP Protocol
- HTTP Methods & Status Codes
- Request–Response Cycle
- หลักการของ RESTful API
- การทดสอบ API ด้วย Postman

HTTP Protocol

- Hypertext Transfer Protocol (HTTP) = โปรโตคอลสำหรับการสื่อสารบนเว็บ
- ใช้โมเดล Client–Server
- ทำงานบน TCP (Port 80) และ HTTPS (Port 443)
- Stateless → แต่ละ request แยกจากกัน

HTTP Request

ประกอบด้วย:

1. **Method** (GET, POST, PUT, DELETE)
2. **URL/Path** เช่น `/users/1`
3. **Headers** เช่น `Content-Type: application/json`
4. **Body** (กรณี POST/PUT)

ตัวอย่าง:

```
POST /users HTTP/1.1
Host: example.com
Content-Type: application/json
```

```
{ "name": "Alice", "email": "[alice@example.com](mailto:alice@example.com)" }
```

HTTP Response

ประกอบด้วย:

1. **Status Code** (200, 404, 500)
2. **Headers**
3. **Body (Payload)**

ตัวอย่าง:

```
HTTP/1.1 201 Created  
Content-Type: application/json  
  
{ "id": 1, "name": "Alice" }
```

HTTP Methods

- GET → ดึงข้อมูล (Read)
- POST → สร้างข้อมูลใหม่ (Create)
- PUT → แก้ไขข้อมูล (Update/Replace)
- PATCH → แก้ไขบางส่วน (Partial Update)
- DELETE → ลบข้อมูล

HTTP Status Codes

- **2xx Success** → 200 OK, 201 Created
- **4xx Client Error** → 400 Bad Request, 401 Unauthorized, 404 Not Found
- **5xx Server Error** → 500 Internal Server Error

RESTful API คืออะไร?

- REpresentational State Transfer
- รูปแบบการออกแบบ API บน HTTP
- หลักการสำคัญ:
 - ใช้ Resource (เช่น `/users` , `/books`)
 - ใช้ HTTP Methods แทน Action
 - Stateless
 - URI สื่อความหมาย

ตัวอย่าง RESTful API

Resource: users

Method	Path	Action
GET	/users	ดึงรายชื่อผู้ใช้
GET	/users/1	ดึงข้อมูลผู้ใช้
POST	/users	เพิ่มผู้ใช้ใหม่
PUT	/users/1	อัปเดตข้อมูลผู้ใช้
DELETE	/users/1	ลบผู้ใช้

Tools สำหรับทดสอบ API

- Postman
- Insomnia
- cURL
- HTTPie

Lab

1. เพิ่ม Endpoint ใหม่ใน Spring Boot

- `/hello/{name}` → ตอบ JSON `{ "message": "Hello, {name}" }`

2. ทดสอบด้วย Postman

3. ทดลองใช้ GET, POST, PUT, DELETE

Assignment

- สร้าง REST API `/echo` ที่รับ JSON และส่งกลับ JSON พร้อม timestamp
- ตัวอย่าง Input:

```
{ "data": "Backend" }
```

- ตัวอย่าง Output:

```
{ "data": "Backend", "timestamp": "2025-10-01T10:30:00" }
```