



บทที่ 12

Web API

ดร.ธีระยุทธ ทองเครือ

ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์

มหาวิทยาลัยขอนแก่น



Web API



- ❖ Web API อาจเรียกอีกชื่อหนึ่งว่า Web Service คือ ส่วนหนึ่งของระบบ ที่เปิดให้บริการข้อมูลสำหรับ Application เช่น Mobile Application, Web Application, IoT Application
- ❖ รับ HTTP Request และส่ง HTTP Response บนเครือข่าย internet
- ❖ การทำงานมีความคล้ายคลึงกับฟังก์ชัน หรือเมธอด คือ สามารถรับ input และ return ผลลัพธ์กลับได้
- ❖ HTTP Response มีรูปแบบที่สามารถนำไปประมวลผลต่อได้ เช่น XML, JSON



ชนิดของ Web API



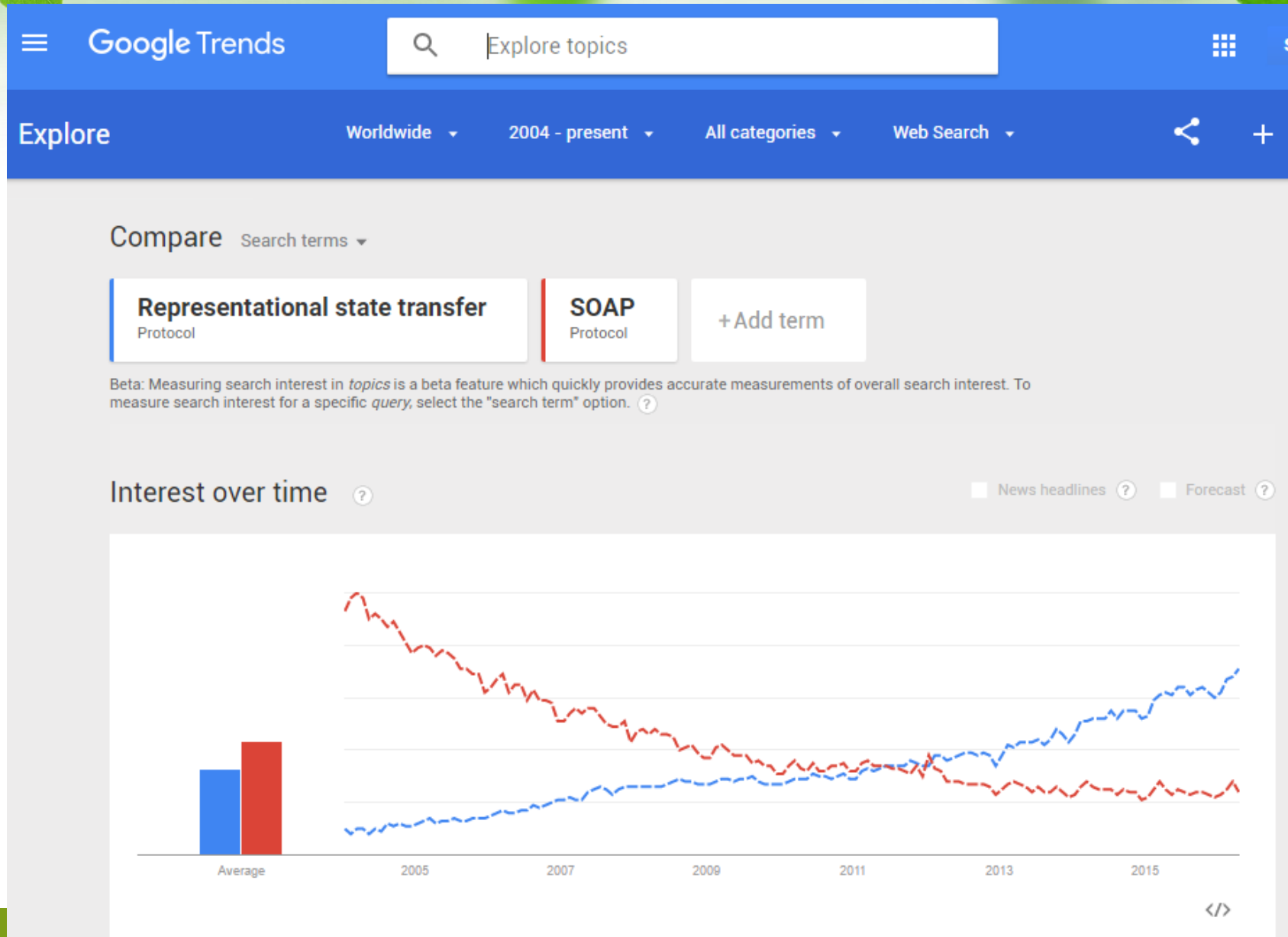
❖ RESTful API

- แลกเปลี่ยนข้อมูลในรูปแบบ JSON หรือ XML
- เรียกใช้จาก URL ได้โดยตรง

❖ SOAP-Based Web Services (WS-* standards)

- แลกเปลี่ยนข้อมูลในรูปแบบ XML เท่านั้น
- เรียกใช้โดยส่งคำร้องในรูปแบบ SOAP (Simple Object Access Protocol)

RESTful API และ SOAP





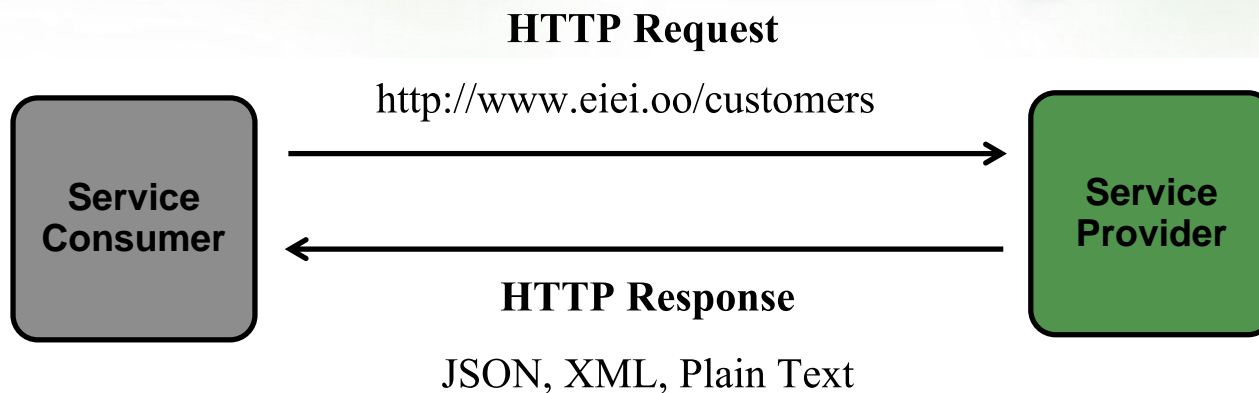
RESTful API



- ❖ REST (**RE**presentational **S**tate **T**ransfer) คือ สถาปัตยกรรมของระบบที่ออกแบบสำหรับการให้บริการ โดยมองข้อมูลทุกอย่างเป็น Resource ที่อ้างอิงได้ด้วย URI
- ❖ เว็บเซอร์วิสที่สร้างตามเงื่อนไขของ REST จะเรียกว่า "**RESTful Web Services**"
- ❖ มักถูกเรียกว่า Web API เช่น Facebook มีเว็บเซอร์วิสแบบ REST เรียกว่า Graph API
- ❖ REST ถูกนำเสนอโดยวิทยานิพนธ์ของนักศึกษาปริญญาเอก ชื่อ Roy Thomas Fielding ในปี ค.ศ. 2000



การทำงานของ RESTful API



Consumer Processing

1. เตรียมข้อมูลที่ Service ต้องการ
2. ส่งข้อมูลไปยัง Service ด้วย URL

Service Processing

1. รับข้อมูลจากผู้ใช้ส่งมา
2. ประมวลผล
3. สร้างข้อความตอบกลับ



การส่งคำร้องไปยัง RESTful API



- ❖ RESTful API ถูกเรียกใช้ด้วย URL ดังนั้นผู้ให้บริการจะต้องอธิบายรูปแบบ URL การเรียกใช้แก่นักพัฒนาที่ต้องการมาใช้บริการ
- ❖ Path ที่ใช้เข้าถึง RESTful API มีหลากหลายรูปแบบ แต่มักจะเป็นคำนาม และมีความหมายเป็นข้อมูลที่ต้องการ เช่น
 - /customers ใช้ในการขอข้อมูลลูกค้าทั้งหมด
 - /customers/{id} เช่น /customers/5 ใช้ในการขอข้อมูลลูกค้าที่มีรหัสลูกค้าเป็น 5
 - /customers?gender=male&province=khonkaen ใช้ในการระบุเงื่อนไขข้อมูลลูกค้าที่ต้องการ
 - /orders/{id}/lineitems เช่น /orders/5/lineitems ขอข้อมูลรายการย่อยของใบสั่งซื้อตามรหัสที่ระบุ



การเรียก RESTful API



- ❖ RESTful API อาจใช้ URL เดียวกัน แต่กำหนด HTTP Request Method ที่แตกต่างกัน เพื่อบ่งชี้การทำงาน

Method	Database Operation	การทำงาน
GET	SELECT	ขอข้อมูลแบบรายการ หรือขอข้อมูล 1 รายการ
POST	INSERT	เพิ่มข้อมูล (ไม่รู้ ID)
PUT	UPDATE	แก้ไขข้อมูล (ต้องรู้ ID)
DELETE	DELETE	ลบข้อมูล (ต้องรู้ ID)



การเรียก RESTful API



❖ <http://www.eiei.oo/customers/5>

GET – ดึงข้อมูลลูกค้าหมายเลข 5

PUT - แก้ไขข้อมูลลูกค้าหมายเลข 5

DELETE - ลบข้อมูลลูกค้าหมายเลข 5

❖ <http://www.eiei.oo/customers>

GET – ดึงข้อมูลลูกค้าทั้งหมด

POST – เพิ่มข้อมูลลูกค้า



การส่งข้อมูลไปกับ URL



- ❖ ในบางครั้ง RESTful API อาจต้องการข้อมูลนอกเหนือจากที่ส่งไปกับ URL เช่น การเพิ่ม หรือแก้ไขข้อมูลในฐานข้อมูล อาจต้องส่งข้อมูลในฟิลด์ต่างๆ
- ❖ Web API สามารถรับข้อมูลในส่วน HTTP Request Body ซึ่งจะอยู่ในรูปแบบใด ขึ้นกับผู้ให้บริการ เช่น JSON, XML, Plain Text
- ❖ การส่งข้อมูลทำได้โดย
 - นักพัฒนาเขียนโค้ดโปรแกรมเพื่อแนบข้อมูลไปกับ HTTP Request
 - ใช้โปรแกรมในการทดสอบ RESTful API เช่น Postman, SoapUI



Postman: <http://www.getpostman.com>



❖ Postman คือ Chrome Application สำหรับทดสอบการรับและส่งข้อมูลกับ RESTful API

The screenshot shows the Postman interface with the following callouts:

1. ระบุ HTTP Request Method (Specify HTTP Request Method)
2. ระบุ URL (Specify URL)
3. เลือกใส่ข้อมูลส่วน HTTP Request Body (Select to enter HTTP Request Body information)
4. เลือกรูปแบบข้อมูลเป็น JSON (Select data format as JSON)
5. ใส่ข้อมูลในรูปแบบ JSON (Enter data in JSON format)
6. เริ่มส่งคำร้องไปยัง Web API (Start sending request to Web API)

The interface shows a POST request to `http://localhost:8080/employee/` with a JSON body:

```
{
  "fullName": "Theerayut Thongkrau",
  "position": "Web Developer",
  "department": {"depId": 1}
}
```

The response status is 200 OK and the time taken is 75 ms. The response body is shown in the bottom panel:

```
{
  "empId": 6,
  "fullName": "Theerayut Thongkrau",
  "position": "Web Developer",
  "department": {
    "depId": 1,
    "deptName": null
  }
}
```



RESTful API ให้บริการจัดการข้อมูลสินค้า



Service ให้บริการ	HTTP Method	URL ในการเรียกเว็บเซอร์วิส รูปแบบตามที่กำหนดใน @Path	ข้อมูลเข้า	ข้อมูลส่งกลับ @Produces
ให้รายการข้อมูล สินค้า	GET	http://localhost:8080/products	-	รายการสินค้าทั้งหมด
ให้ข้อมูลสินค้าตาม รหัส	GET	http://localhost:8080/products/{id}	แวนบรหัสสินค้า มากับ URL	ข้อมูลสินค้า 1 ชิ้น
เพิ่มข้อมูลสินค้า	POST	http://localhost:8080/products	ข้อมูลสินค้า รูปแบบ JSON	ข้อมูลสินค้าที่เพิ่ม ใหม่ พร้อมรหัส
แก้ไขข้อมูลสินค้า	PUT	http://localhost:8080/products/{id}	ข้อมูลสินค้า รูปแบบ JSON	ข้อมูลสินค้าที่แก้ไข
ลบข้อมูลสินค้า	DELETE	http://localhost:8080/products/{id}	แวนบรหัสสินค้า มากับ URL	-



การส่ง Error response ของ RESTful API



การส่ง response จาก API ที่เกิดข้อผิดพลาดจะต้องเป็นไปตามกฎของ HTTP

❖ มี Status code สำหรับ Error ที่ถูกต้อง และส่ง response ในรูปแบบที่ผู้ใช้ request มา เช่น ผู้ใช้ยอมรับข้อมูลในแบบ XML หรือ JSON

■ 4xx เป็น error ผั่ง client หรือผู้ใช้งาน

- 400 สำหรับการ validate ข้อมูล
- 401 และ 403 จะเกี่ยวกับ permission การใช้งาน
- 404 ไม่พบข้อมูล

■ 5xx เป็น error ผ่น server หรือผู้ให้บริการ เช่น 500 และ 503 เกี่ยวกับ server error

❖ สนใจ Accept header เสมอ

❖ ส่งข้อมูลกลับไปให้ถูกต้องตาม Content-Type



โครงสร้าง RESTful API บน Spring

ระบุ annotation เพื่อให้ Spring Boot Container รู้ว่าคลาสสามารถรับ request ได้แบบ REST

```
@RestController  
public class MyAPI {
```

```
@GetMapping("/ชื่อ path เข้าถึงเมธอด")  
ResponseEntity<ชื่อคลาสผลลัพธ์> ชื่อเมธอด() {
```

ระบุ path และ method ในการเข้าถึง

```
// สร้าง Object หรือขอจาก Repository Interface
```

ระบุชื่อคลาสที่ต้องการส่งกลับ อาจเป็น
JavaBean, Entity Class หรือ List ได้

object ที่ระบุ

```
return new ResponseEntity<ชื่อคลาสผลลัพธ์>(ชื่อobjectผลลัพธ์, HttpStatus.OK);  
}
```

```
}
```

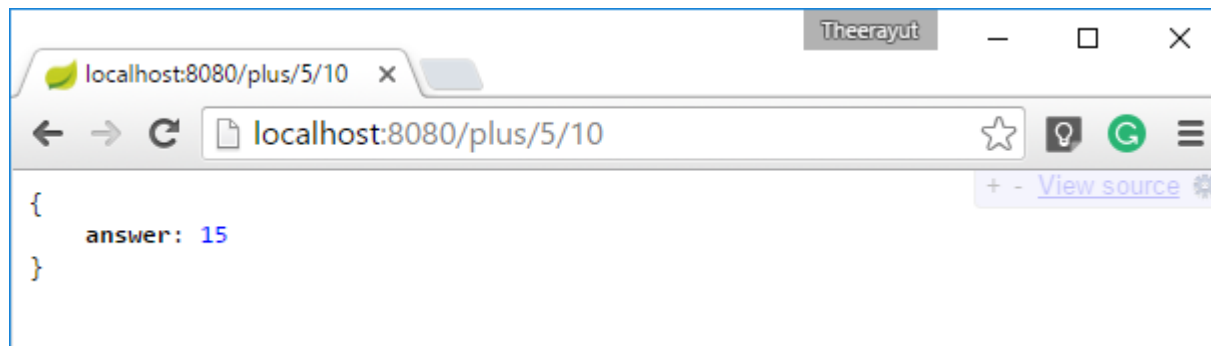
กิจกรรม

❖ สร้าง RESTful API ให้บริการบวกเลข โดยส่งตัวเลขที่ต้องการหาผลบวกทาง URL ซึ่งมีรูปแบบดังนี้

- GET /plus/{ค่าที่1}/{ค่าที่2}

เช่น `http://localhost:8080/plus/5/10`

❖ ตัวอย่างผลลัพธ์ที่ได้





กิจกรรม

❖ สร้าง RESTful API ให้บริการบวกเลข โดยส่งตัวเลขที่ต้องการหาผลบวกในส่วน HTTP Request Body ซึ่งมีรูปแบบดังนี้

- POST /plus

เช่น `http://localhost:8080/plus`

- ตัวอย่างข้อมูลแบบ JSON

```
{  
  "a": 5,  
  "b": 10  
}
```

❖ ใช้โปรแกรม Postman ในการทดสอบ

RunnerImportBuilderTeam Library

http://localhost:8080/plus+No environment

POST

http://localhost:8080/plus

Params

Send

Save

Authorization

Headers (1)

Body

Pre-request Script

Tests

Generate Code

form-data

x-www-form-urlencoded

raw

binary

JSON (application/json)

1

{

2

"a": 5,

3

"b": 10

4

}

Body

Cookies

Headers (4)

Tests

Status: 200 OK

Time: 28 ms

Pretty

Raw

Preview

JSON

1

{

2

"answer": 15

3

}



กิจกรรม

- ❖ ingsร้าง RESTful API ให้บริการคำนวณค่า BMI (Body Mass Index) ซึ่งมีสูตรดังนี้

$$BMI = \frac{Weight (kg)}{Height^2 (m^2)}$$

- ❖ ให้บริการ 2 ั้งแบบ GET และ POST

- ❖ แบบ GET

- รูปแบบ Request URL: `http://localhost:8080/bmi/{weight}/{height}`
- ตัวอย่าง: `http://localhost:8080/bmi/56/173`
- ตัวอย่าง Response: `{ "answer": "18.710948833968633" }`

- ❖ แบบ POST

- รูปแบบ Request URL: `http://localhost:8080/bmi`
- ตัวอย่างข้อมูล: `{ "weight": "56", "height": "173" }`
- ตัวอย่าง Response: `{ "answer": "18.710949246550168" }`