ผลการเรียกใช้งาน

1.GET ทั้งหมด

```
PS C:\งานของกันตวิชญ์ปี3\Principle\lab5> curl.exe http://localhost:8080/coffees
[{"id":1,"name":"Espresso","price":45.0},{"id":2,"name":"Latte","price":55.0}]
PS C:\งานของกันตวิชญ์ปี3\Principle\lab5>
```

2.GET ตาม id

```
PS C:\งานของกันตวิชญ์ปี3\Principle\lab5> curl.exe http://localhost:8080/coffees/1
{"id":1,"name":"Espresso","price":45.0}
PS C:\งานของกันตวิชญ์ปี3\Principle\lab5>
```

3.POST เพิ่มเมนูใหม่

```
PS C:\งานของกันตวิชญ์ปี3\Principle\lab5> curl.exe -X POST http://localhost:8080/coffees
`
>>   -H "Content-Type: application/json" `
>>   -d '{\"name\":\"Cappuccino\",\"price\":60.0}'
{"id":3,"name":"Cappuccino","price":60.0}
PS C:\งานของกันตวิชญ์ปี3\Principle\lab5>
```

```
PS C:\งานของกันตวิชญ์ปี3\Principle\lab5> curl.exe http://localhost:8080/coffees
[{"id":1,"name":"Espresso","price":45.0},{"id":2,"name":"Latte","price":55.0},{"id":
3,"name":"Cappuccino","price":60.0}]
PS C:\งานของกันตวิชญ์ปี3\Principle\lab5>
```

4.PUT แก้ไข

```
PS C:\งานของกันตวิชญ์ปี3\Principle\lab5> curl.exe -X PUT http://localhost:8080/coffees/
2 `
>>   -H "Content-Type: application/json" `
>>   -d '{\"name\":\"Latte\",\"price\":50.0}'
{"id":2,"name":"Latte","price":50.0}
PS C:\งานของกันตวิชญ์ปี3\Principle\lab5>
```

5.DELETE ลบ

```
PS C:\งานของกันตวิชญ์ปี3\Principle\lab5> curl.exe -X DELETE http://localhost:8080/coffe
es/3
PS C:\งานของกันตวิชญ์ปี3\Principle\lab5> curl.exe http://localhost:8080/coffees
[{"id":1,"name":"Espresso","price":45.0},{"id":2,"name":"Latte","price":50.0}]
PS C:\งานของกันตวิชญ์ปี3\Principle\lab5>
```

6.กรณี 404

```
PS C:\งานของกันตวิชญ์ปี3\Principle\lab5> curl.exe -i http://localhost:8080/coffees/9
HTTP/1.1 404
Content-Length: 0
Date: Sun, 26 Jul 2026 09:04:22 GMT
```

Discussion

1. HTTP method แต่ละตัวต่างกันอย่างไร

- GET ใช้สำหรับ "ขอดูข้อมูล" โดยไม่เปลี่ยนแปลงอะไรในระบบ เช่น GET /coffees ดูเมนูทั้งหมด
- POST ใช้ "สร้างข้อมูลใหม่" เช่น POST /coffees เพิ่มกาแฟรายการใหม่
- PUT ใช้ "แก้ไขข้อมูลเดิม" เช่น PUT /coffees/2 เปลี่ยนราคา Latte
- DELETE ใช้ "ลบข้อมูล" เช่น DELETE /coffees/3 ลบเมนูออกจากลิสต์

2. ทำไมต้องแยก Controller กับ Service

Controller	Service
Controller มีหน้าที่รับ HTTP request และส่ง response	Service มีหน้าที่เก็บ logic การทำงานจริง (เพิ่ม/ลบ/แก้ไขข้อมูล)

การแยกกันทำให้แต่ละคลาสมีความรับผิดชอบเดียว หากโปรแกรมโตขึ้น เช่น ต้องเปลี่ยนจากเก็บข้อมูลใน List เป็นฐานข้อมูลจริง ก็แก้แค่ใน Service โดยไม่กระทบ Controller เลย นอกจากนี้ยังทดสอบ logic ใน Service ได้ง่ายกว่าโดยไม่ต้องยิง HTTP request จริง

3. ข้อมูลใน List หายไปตอนไหน และถ้าไม่อยากให้หายควรทำอย่างไร

ข้อมูลที่เก็บใน List แบบ in-memory จะหายไปทันทีที่แอปพลิเคชันหยุดทำงาน เพราะข้อมูลอยู่ใน RAM ไม่ได้ถูกบันทึกลงดิสก์ ถ้าอยากให้ข้อมูลไม่หาย ควรเปลี่ยนไปเก็บในฐานข้อมูล เช่น MySQL, PostgreSQL เป็นต้น ผ่านการใช้ Spring Data JPA แทนการใช้ List ธรรมดา เพื่อให้ข้อมูลถูกบันทึกแบบถาวร

4. Annotation แต่ละตัวทำหน้าที่อะไร

- @RestController — บอก Spring ว่าคลาสนี้เป็น controller ที่รับ HTTP request และคืนค่าเป็น JSON โดยอัตโนมัติ
- @GetMapping — กำหนดว่า method นี้จะทำงานเมื่อมี HTTP GET request เข้ามาที่ path ที่ระบุ
- @PostMapping — กำหนดว่า method นี้จะทำงานเมื่อมี HTTP POST request เข้ามา ใช้สำหรับสร้างข้อมูลใหม่
- @PathVariable — ดึงค่าจากส่วนหนึ่งของ URL มาใช้เป็นตัวแปร เช่น /coffees/{id} ดึงค่า id มาใช้ในโค้ด
- @RequestBody — แปลงข้อมูล JSON ที่ client ส่งมาใน body ของ request ให้กลายเป็น Java object โดยอัตโนมัติ