

## แบบฝึกปฏิบัติ ครั้งที่ 3

เรื่อง สร้างโปรเจกต์ Spring ด้วย IntelliJ IDEA

วัตถุประสงค์ 1. เพื่อฝึกฝนการสร้าง Service ที่มีการสื่อสารแบบ Rest ด้วย Spring

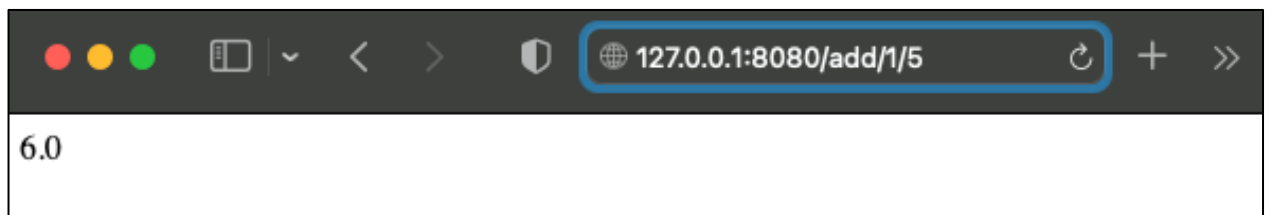
### ข้อที่ 1 สร้างคลาส MathService

1. ให้นักศึกษาพัฒนา MathService ด้วย Spring และเรียกใช้งานผ่าน Web Browser และโปรแกรม Postman

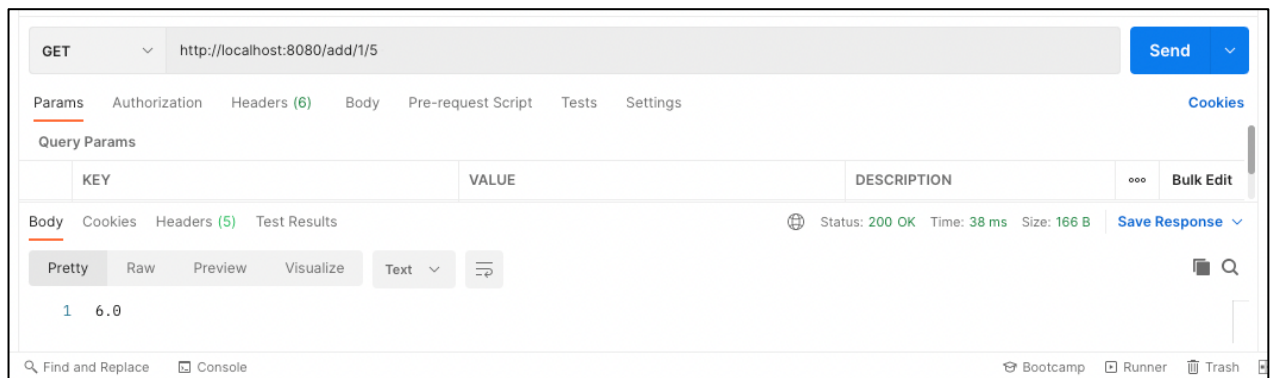
1.1. เมธอด `add()` จะรับค่าตัวเลขชนิด `double` มา 2 จำนวน และจะคืนค่าเป็นผลลัพธ์ของการบวก 2 จำนวนข้างต้นด้วยชนิดข้อมูลประเภท `String` โดยที่

- กำหนดให้ใช้ Annotation ที่ช่วยสกัดค่าพารามิเตอร์ที่ส่งมาผ่าน URL ด้วย `@PathVariable`
- กำหนดให้ URL ในการเรียกใช้งานเป็น `http://localhost:8080/add/1/5`

ตัวอย่างการเรียกใช้งานผ่าน Web Browser



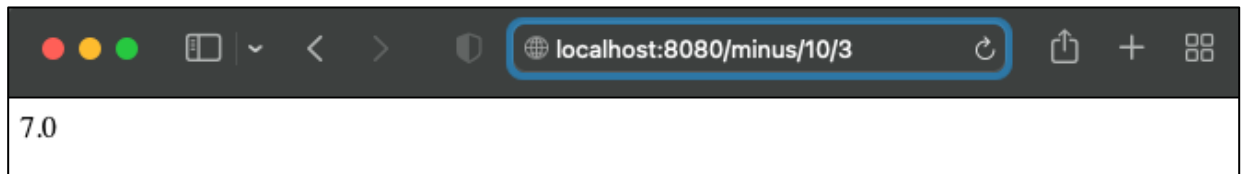
ตัวอย่างการเรียกใช้งานผ่านโปรแกรม POSTMAN



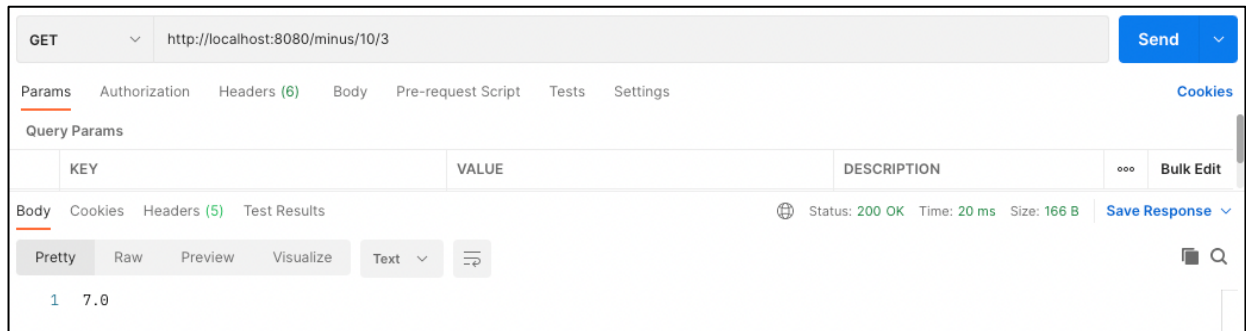
1.2. เมธอด `minus()` จะรับค่าตัวเลขชนิด `double` มา 2 จำนวน และจะคืนค่าเป็นผลลัพธ์ของการลบ 2 จำนวนข้างต้นด้วยชนิดข้อมูลประเภท `String` โดยที่

- กำหนดให้ใช้ Annotation ที่ช่วยสกัดค่าพารามิเตอร์ที่ส่งมาผ่าน URL ด้วย `@PathVariable`
- กำหนดให้ URL ในการเรียกใช้งานเป็น `http://localhost:8080/minus/10/3`

ตัวอย่างการเรียกใช้งานผ่าน Web Browser



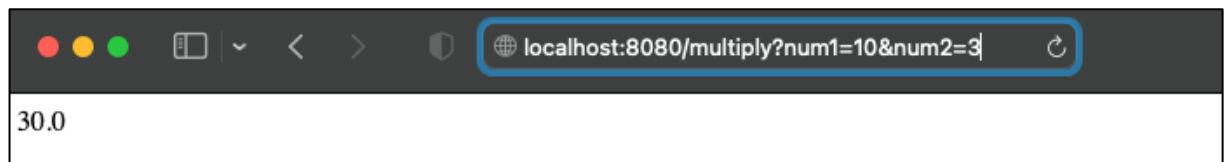
### ตัวอย่างการเรียกใช้งานผ่านโปรแกรม POSTMAN



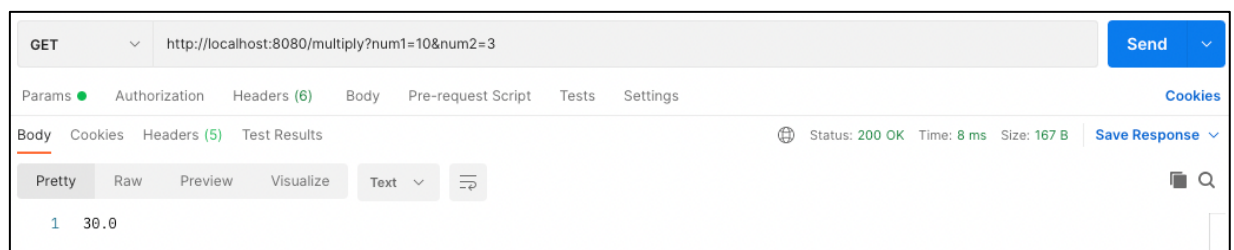
1.3. เมธอด `multiply()` จะรับค่าตัวเลขชนิด `double` มา 2 จำนวน และจะคืนค่าเป็นผลลัพธ์ของการคูณ 2 จำนวนข้างต้นด้วยชนิดข้อมูลประเภท `String` โดยที่

- กำหนดให้ใช้ Annotation ที่ช่วยสกัดค่าพารามิเตอร์ที่ส่งมาผ่าน URL ด้วย `@RequestParam`
- กำหนดให้ URL ในการเรียกใช้งานเป็น `http://localhost:8080/multiply?num1=10&num2=3`

### ตัวอย่างการเรียกใช้งานผ่าน Web Browser



### ตัวอย่างการเรียกใช้งานผ่านโปรแกรม POSTMAN

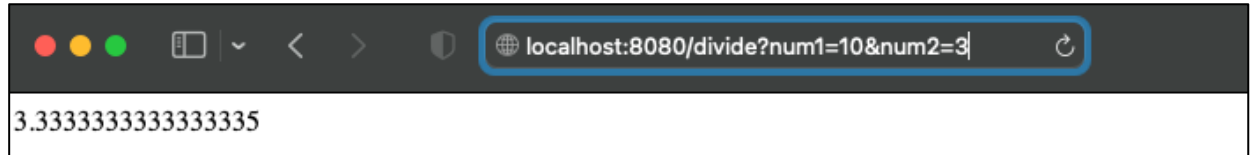


1.4. เมธอด `divide()` จะรับค่าตัวเลขชนิด `double` มา 2 จำนวน และจะคืนค่าเป็นผลลัพธ์ของการหาร 2 จำนวนข้างต้นด้วยชนิดข้อมูลประเภท `String` โดยที่

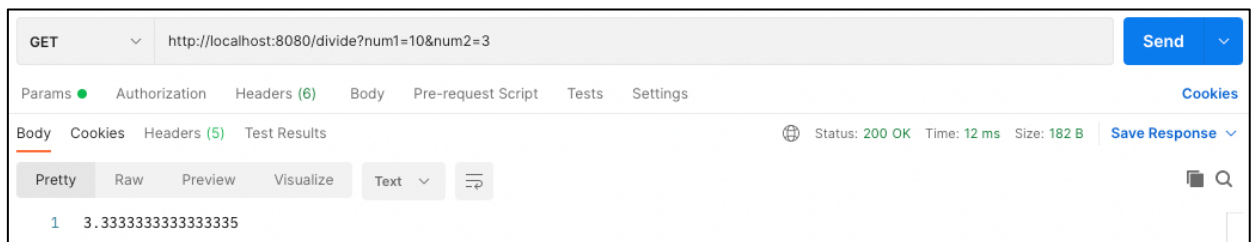
- กำหนดให้ใช้ Annotation ที่ช่วยสกัดค่าพารามิเตอร์ที่ส่งมาผ่าน URL ด้วย `@RequestParam`

- กำหนดให้ URL ในการเรียกใช้งานเป็น `http://localhost:8080/divide?num1=10&num2=3`

ตัวอย่างการเรียกใช้งานผ่าน Web Browser



ตัวอย่างการเรียกใช้งานผ่านโปรแกรม POSTMAN



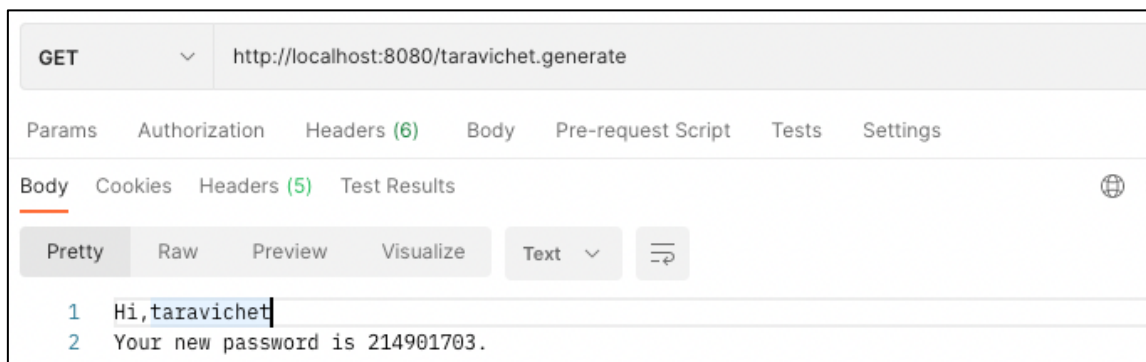
## ข้อที่ 2 สร้างคลาส GeneratePasswordService

2. ให้นักศึกษาพัฒนา GeneratePasswordService ด้วย Spring และเรียกใช้งานผ่านโปรแกรม Postman
- 2.1. เมธอด `generate()` จะรับค่าชื่อของผู้ใช้งานชนิด String และจะคืนค่าเป็นข้อความในรูปแบบต่อไปนี้

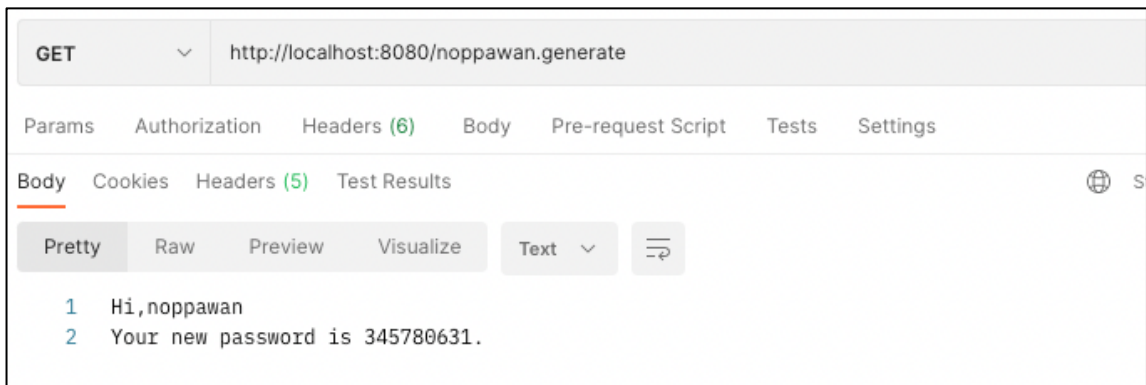
Hi, [ชื่อของผู้ใช้งาน]  
 Your new password is [รหัสผ่าน].

โดยที่ URL ในการเรียกใช้งานเป็น `http://localhost:8080/[ชื่อของผู้ใช้งาน].generate`  
 ตัวอย่างการเรียกใช้งานผ่านโปรแกรม POSTMAN

### ตัวอย่างที่ 1



### ตัวอย่างที่ 2



### ข้อที่ 3 สร้างคลาส Customer

3. ให้นักศึกษาพัฒนา Customer Service ด้วย Spring และเรียกใช้งานผ่าน Web Browser และโปรแกรม Postman

3.1. ให้นักศึกษาสร้างคลาส Customer ดังคลาสไดอะแกรมต่อไปนี้

Customer	
- ID	: String
- name	: String
- sex	: boolean
- age	: int
+ Customer()	
+ Customer(String ID, String n, boolean s, int a)	

3.2. ให้สร้างทุกคลาสเป็นไปตามหลักการ OOP อันประกอบด้วยหลักการห่อหุ้ม หลักการสืบทอด และการมีได้หลากหลายรูปแบบ

3.3. กำหนดให้ default constructor ของคลาส Customer ให้เรียกใช้ constructor อีกตัวโดยกำหนดให้

แอททริบิวต์	ID	name	sex	age
ค่า	“ ”	null	female	0

- แอททริบิวต์ sex จะเป็น true ก็ต่อเมื่อผู้ใช้กรอกมาเป็น “Male” หรือ “male” เท่านั้น ขณะแอททริบิวต์ sex จะเป็น false ก็ต่อเมื่อผู้ใช้กรอกมาเป็น “Female” หรือ “female” เท่านั้น
- แอททริบิวต์ age จะเป็นตัวเลขจำนวนเต็มมากกว่า 0 เท่านั้น ในกรณีที่ผู้ใช้กรอกค่าน้อยกว่า 0 จะกำหนดให้เป็น 0
- Constructor ที่มีการรับค่าของคลาส Customer นักศึกษาสามารถปรับเปลี่ยนชนิดข้อมูลเป็น String หรือ boolean ของตัวแปร s ก็ได้ ขึ้นกับความเหมาะสมของการเขียนโปรแกรมของนักศึกษา

3.4. กำหนดให้สร้าง setter และ getter ของทุกแอททริบิวต์ โดยมีเงื่อนไขตามข้อ 3.3

## ข้อที่ 4 สร้างคลาส CustomerController

4. ให้นักศึกษาสร้าง CustomerController เพื่อเป็น RestController ของคลาส Customer ดังคลาสไดอะแกรมต่อไปนี้

CustomerController	
- customers	: List<Customer>
+ getCustomers ()	: List<Customer>
+ getCustomerByID (String ID)	: Customer
+ getCustomerByName (String n)	: Customer
+ delCustomerByID (String ID)	: boolean
+ delCustomerByName (String n)	: boolean
+ addCustomer (String ID, String n, String s, int a)	: boolean

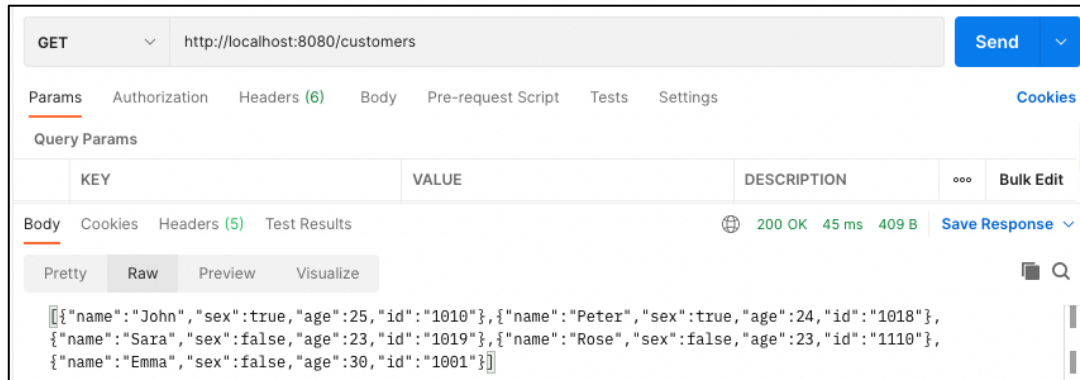
- 4.1. สร้างแอททริบิว customers **ประกาศ**เป็น List แต่จะ**สร้าง**เป็นอะไรก็ได้ จากนั้น กำหนดข้อมูลสมาชิกเก็บไว้ในแอททริบิว customers ดังนี้

id	name	sex	Age
1010	John	Male	25
1018	Peter	Male	24
1019	Sara	Female	23
1110	Rose	Female	23
1001	Emma	Female	30

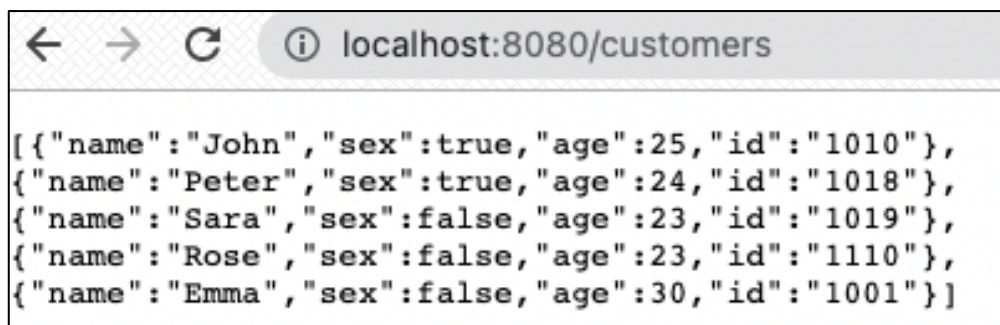
หมายเหตุ ให้นักศึกษาประกาศและกำหนดค่าของแอททริบิว customers เลย

4.2. ให้นักศึกษาร่างเมธอด `getCustomers()` `getCustomerByID(String ID)` และ `getCustomerByName(String n)` เพื่อให้บริการข้อมูลลูกค้าจากเงื่อนไขต่อไปนี้

- เมธอด `getCustomers()` จะคืนค่าข้อมูลลูกค้าทั้งหมดให้กับผู้เรียกใช้งาน และกำหนดให้ URL ในการเรียกใช้งานเป็น `http://localhost/customers` ด้วยตัวดำเนินการ GET ผลลัพธ์ดังแสดงในภาพ

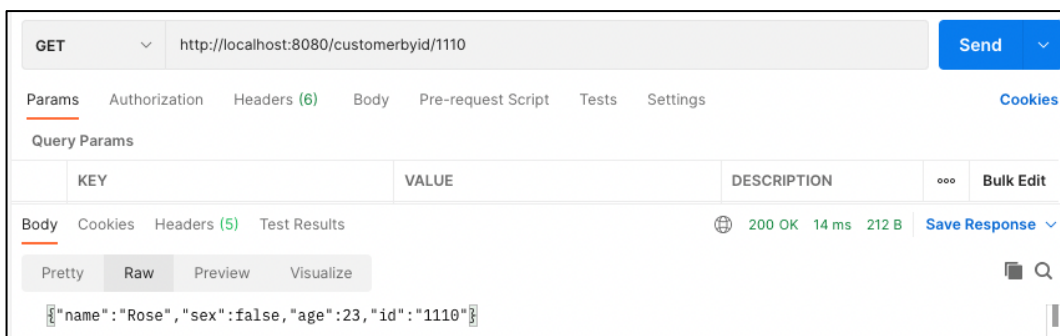


ตัวอย่างผลลัพธ์การเรียกใช้งานผ่านโปรแกรม POSTMAN

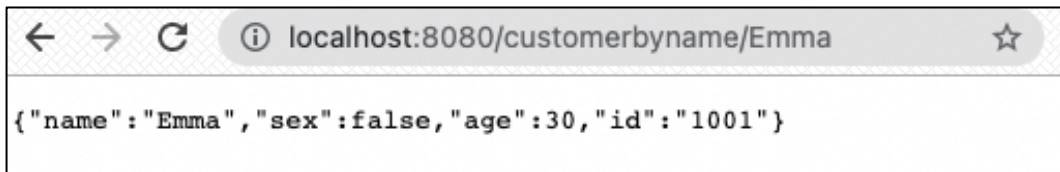


ตัวอย่างผลลัพธ์การเรียกใช้งานผ่าน Web Browser

- เมธอด `getCustomerByID(String ID)` จะคืนค่าข้อมูลลูกค้าที่มี ID ตามที่กำหนดให้กับผู้เรียกใช้งาน และกำหนดให้ URL ในการเรียกใช้งานเป็น `http://localhost/customerbyid/{id}` ด้วยตัวดำเนินการ GET ผลลัพธ์ดังแสดงในภาพ

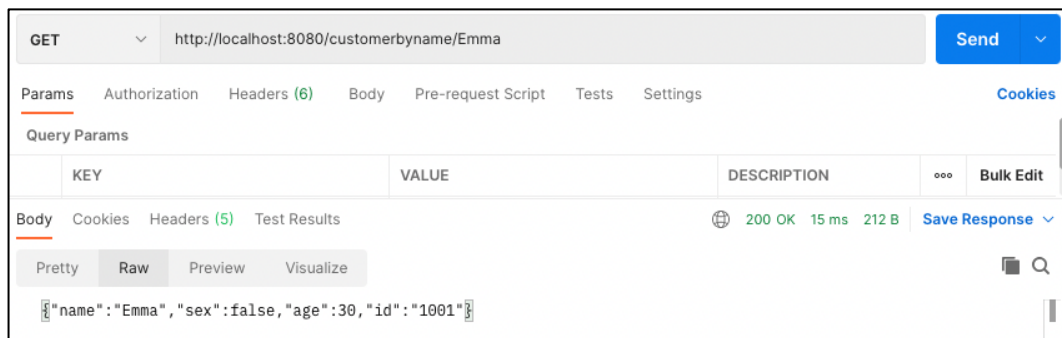


ตัวอย่างผลลัพธ์การเรียกใช้งานผ่านโปรแกรม POSTMAN

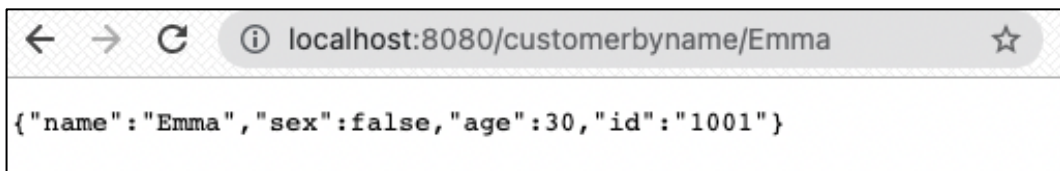


ตัวอย่างผลลัพธ์การเรียกใช้งานผ่าน Web Browser

- เมธอด `getCustomerByName(String n)` จะคืนค่าข้อมูลลูกค้าที่มี name ตามที่กำหนด และกำหนดให้ URL ในการเรียกใช้งานเป็น `http://localhost/customerbyname/{n}` ด้วยตัวดำเนินการ GET ผลลัพธ์ดังแสดงในภาพ



ตัวอย่างผลลัพธ์การเรียกใช้งานผ่านโปรแกรม POSTMAN

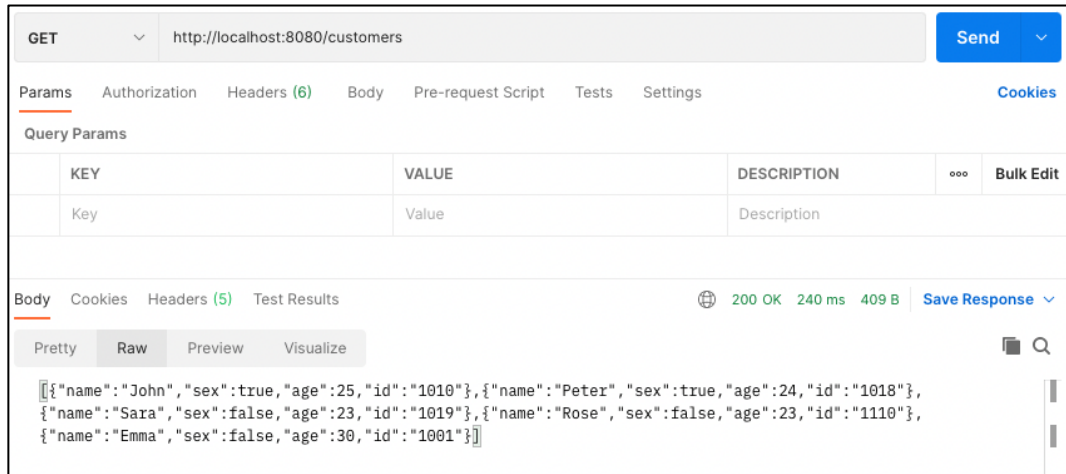


ตัวอย่างผลลัพธ์การเรียกใช้งานผ่าน Web Browser



5. ให้นักศึกษาร่างเมธอด `delCustomerByID(String ID)` และ `delCustomerByName(String n)` เพื่อลบข้อมูลลูกค้าจากเงื่อนไขต่อไปนี้

5.1. เมธอด `delCustomerByID(String ID)` จะลบข้อมูลลูกค้าที่มี ID ตามที่กำหนดจากข้อมูลของแอททริบิวต์ `customer` และกำหนดให้ URL ในการเรียกใช้งานเป็น `http://localhost/customerDelById/{id}` ด้วยตัวดำเนินการ DELETE ผลลัพธ์ดังแสดงในภาพ



GET `http://localhost:8080/customers` Send

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Query Params

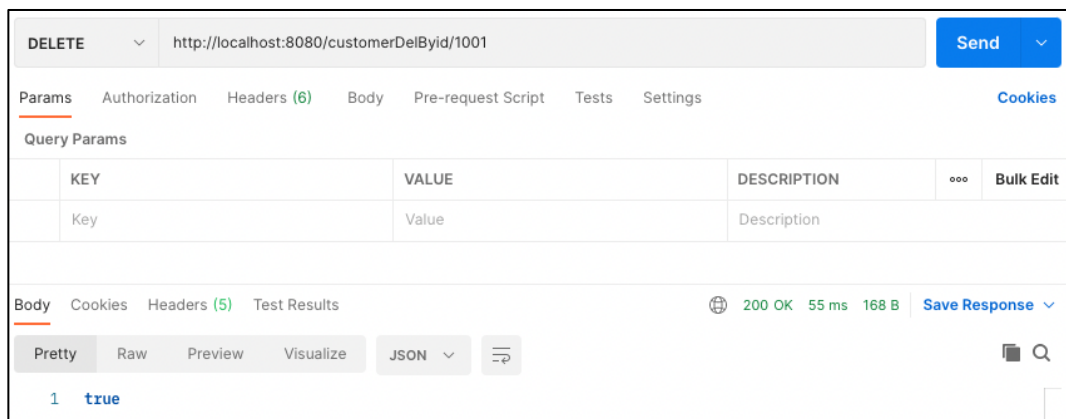
KEY	VALUE	DESCRIPTION	...	Bulk Edit
Key	Value	Description		

Body Cookies Headers (5) Test Results 200 OK 240 ms 409 B Save Response

Pretty Raw Preview Visualize

```
[{"name": "John", "sex": true, "age": 25, "id": "1010"}, {"name": "Peter", "sex": true, "age": 24, "id": "1018"}, {"name": "Sara", "sex": false, "age": 23, "id": "1019"}, {"name": "Rose", "sex": false, "age": 23, "id": "1110"}, {"name": "Emma", "sex": false, "age": 30, "id": "1001"}]
```

ข้อมูลก่อนการเรียกใช้งาน `http://localhost/customerDelById/{id}`



DELETE `http://localhost:8080/customerDelById/1001` Send

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Query Params

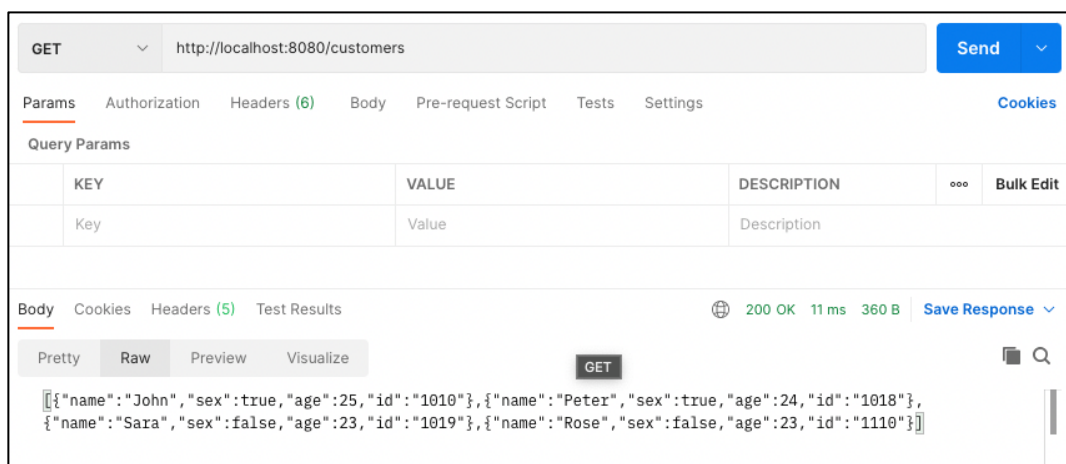
KEY	VALUE	DESCRIPTION	...	Bulk Edit
Key	Value	Description		

Body Cookies Headers (5) Test Results 200 OK 55 ms 168 B Save Response

Pretty Raw Preview Visualize JSON

```
1 true
```

เรียกใช้งาน `http://localhost/customerDelById/1001`



GET `http://localhost:8080/customers` Send

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION	...	Bulk Edit
Key	Value	Description		

Body Cookies Headers (5) Test Results 200 OK 11 ms 360 B Save Response

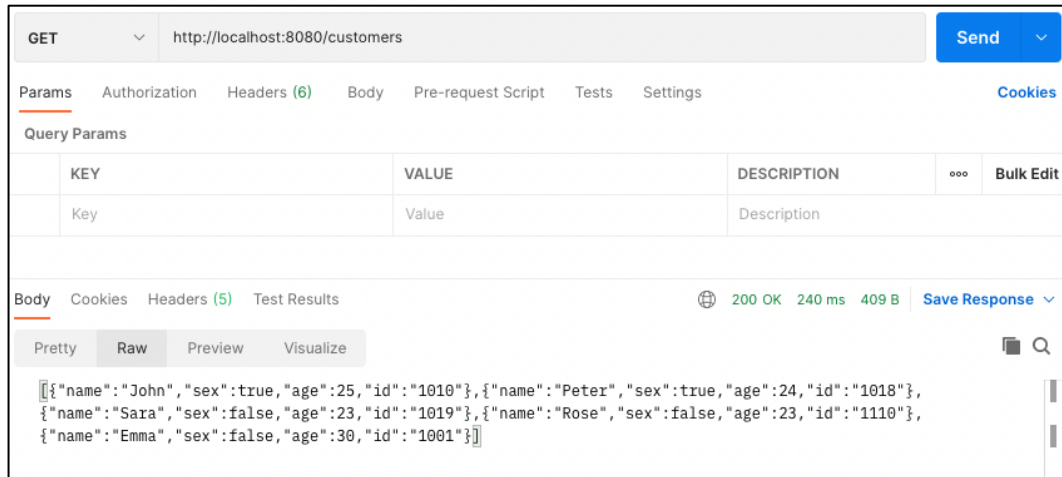
Pretty Raw Preview Visualize GET

```
[{"name": "John", "sex": true, "age": 25, "id": "1010"}, {"name": "Peter", "sex": true, "age": 24, "id": "1018"}, {"name": "Sara", "sex": false, "age": 23, "id": "1019"}, {"name": "Rose", "sex": false, "age": 23, "id": "1110"}]
```

ข้อมูลหลังการเรียกใช้งาน `http://localhost/customerDelById/{id}`

5.2. เมธอด `delCustomerByName(String n)` จะลบข้อมูลลูกค้าที่มี name ตามที่กำหนดจากข้อมูลของแอพพลิเคชัน customer และกำหนดให้ URL ในการเรียกใช้งานเป็น

`http://localhost/customerDelByname/{n}` ด้วยตัวดำเนินการ DELETE ผลลัพธ์ดังแสดงในภาพ



GET `http://localhost:8080/customers` Send

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Query Params

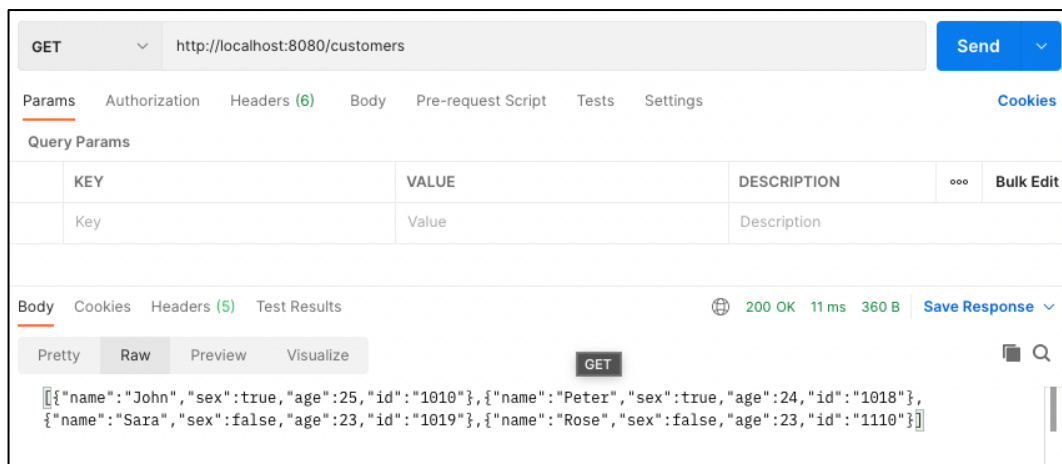
KEY	VALUE	DESCRIPTION	...	Bulk Edit
Key	Value	Description		

Body Cookies Headers (5) Test Results 200 OK 240 ms 409 B Save Response

Pretty Raw Preview Visualize

```
[{"name": "John", "sex": true, "age": 25, "id": "1010"}, {"name": "Peter", "sex": true, "age": 24, "id": "1018"}, {"name": "Sara", "sex": false, "age": 23, "id": "1019"}, {"name": "Rose", "sex": false, "age": 23, "id": "1110"}, {"name": "Emma", "sex": false, "age": 30, "id": "1001"}]
```

ข้อมูลก่อนการเรียกใช้งาน `http://localhost/customerDelByname/{n}`



GET `http://localhost:8080/customers` Send

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION	...	Bulk Edit
Key	Value	Description		

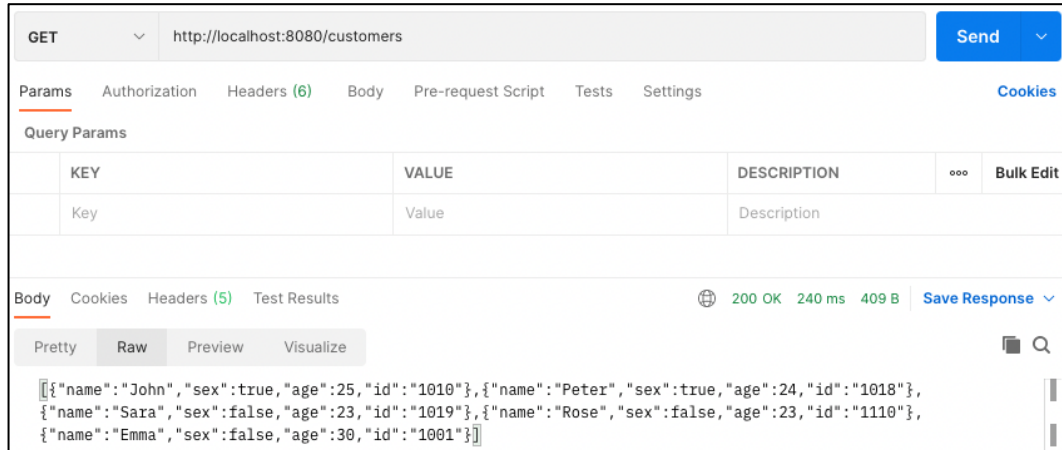
Body Cookies Headers (5) Test Results 200 OK 11 ms 360 B Save Response

Pretty Raw Preview Visualize GET

```
[{"name": "John", "sex": true, "age": 25, "id": "1010"}, {"name": "Peter", "sex": true, "age": 24, "id": "1018"}, {"name": "Rose", "sex": false, "age": 23, "id": "1110"}]
```

ข้อมูลหลังการเรียกใช้งาน `http://localhost/customerDelByname/{n}`

6. ให้นักศึกษาร่างเมธอด `addCustomer(String ID, String n, String s, int a)` เพื่อเพิ่มข้อมูลลูกค้าใหม่ลงในข้อมูลของแอพทริวิ customer และกำหนดให้ URL ในการเรียกใช้งานเป็น `http://localhost/addCustomer` ด้วยตัวดำเนินการ GET ผลลัพธ์ดังแสดงในภาพ



GET `http://localhost:8080/customers` Send

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Query Params

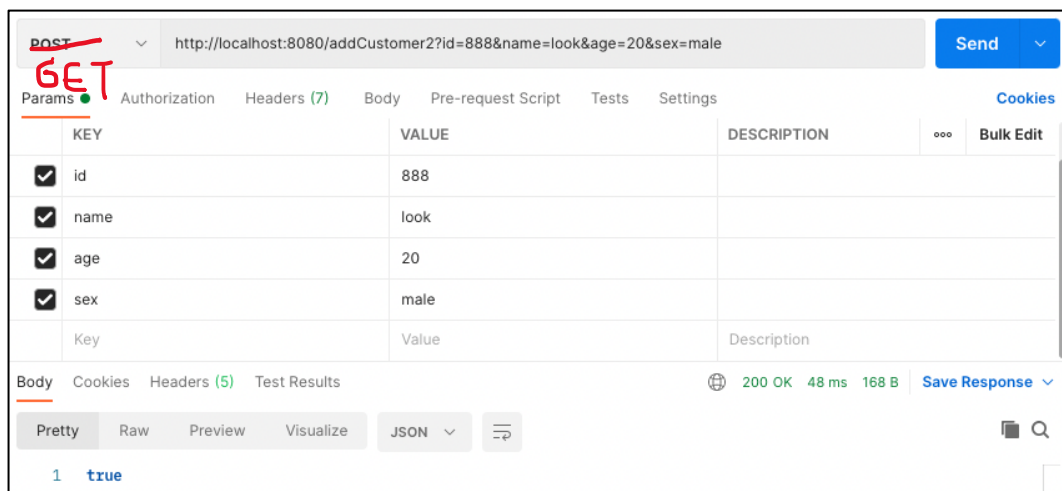
KEY	VALUE	DESCRIPTION	...	Bulk Edit
Key	Value	Description		

Body Cookies Headers (5) Test Results 200 OK 240 ms 409 B Save Response

Pretty Raw Preview Visualize

```
[{"name": "John", "sex": true, "age": 25, "id": "1010"}, {"name": "Peter", "sex": true, "age": 24, "id": "1018"}, {"name": "Sara", "sex": false, "age": 23, "id": "1019"}, {"name": "Rose", "sex": false, "age": 23, "id": "1110"}, {"name": "Emma", "sex": false, "age": 30, "id": "1001"}]
```

ข้อมูลก่อนการเรียกใช้งาน `http://localhost/addCustomer`



POST `http://localhost:8080/addCustomer?id=888&name=look&age=20&sex=male` Send

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

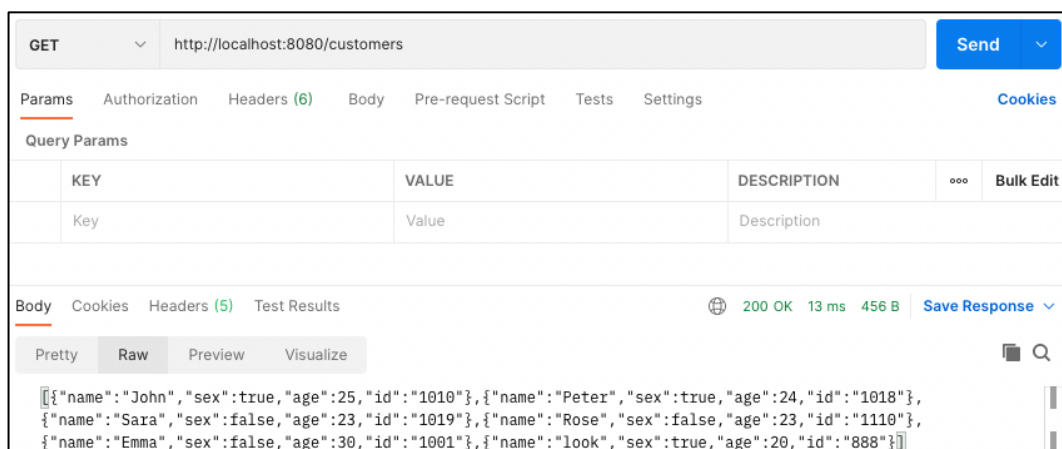
KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/> id	888			
<input checked="" type="checkbox"/> name	look			
<input checked="" type="checkbox"/> age	20			
<input checked="" type="checkbox"/> sex	male			
Key	Value	Description		

Body Cookies Headers (5) Test Results 200 OK 48 ms 168 B Save Response

Pretty Raw Preview Visualize JSON

```
1 true
```

เรียกใช้งาน `http://localhost/addCustomer` โดยกำหนดให้ `id = 888`, `name = look`, `age = 20` และ `sex = male`



GET `http://localhost:8080/customers` Send

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION	...	Bulk Edit
Key	Value	Description		

Body Cookies Headers (5) Test Results 200 OK 13 ms 456 B Save Response

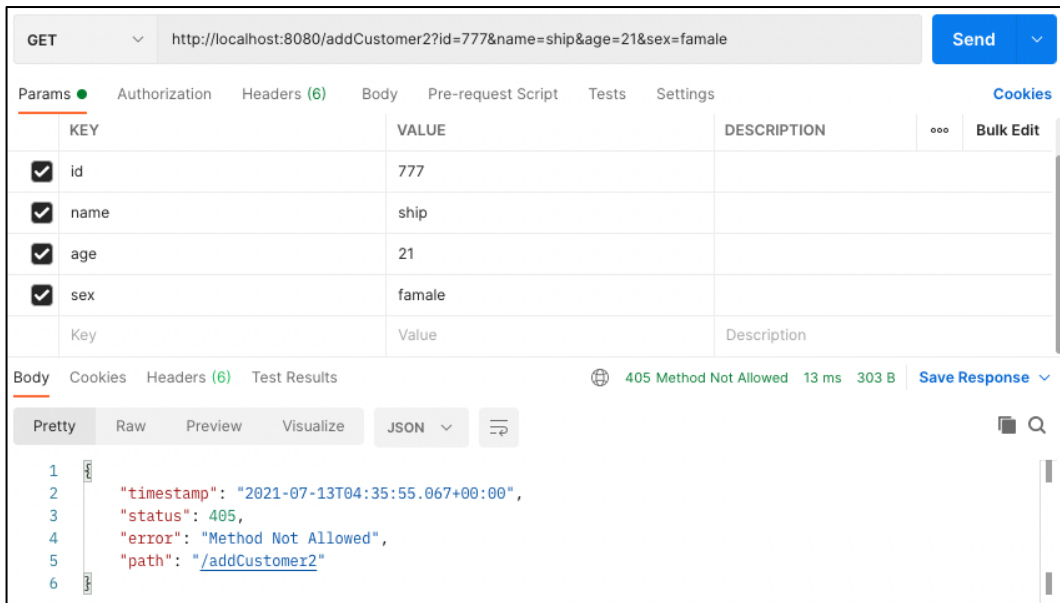
Pretty Raw Preview Visualize

```
[{"name": "John", "sex": true, "age": 25, "id": "1010"}, {"name": "Peter", "sex": true, "age": 24, "id": "1018"}, {"name": "Sara", "sex": false, "age": 23, "id": "1019"}, {"name": "Rose", "sex": false, "age": 23, "id": "1110"}, {"name": "Emma", "sex": false, "age": 30, "id": "1001"}, {"name": "look", "sex": true, "age": 20, "id": "888"}]
```

ข้อมูลหลังการเรียกใช้งาน `http://localhost/addCustomer`

7. ให้นักศึกษาลองปรับเมธอด `addCustomer(String ID, String n, String s, int a)` ให้เรียกใช้งานผ่าน `http://localhost/addCustomer2` ด้วยตัวดำเนินการ POST และเปรียบเทียบว่าผลลัพธ์เหมือนหรือแตกต่างจากข้อ 6

8. จากภาพต่อไปนี้ทำไมถึง เกิด Error



GET `http://localhost:8080/addCustomer2?id=777&name=ship&age=21&sex=female` Send

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/> id	777			
<input checked="" type="checkbox"/> name	ship			
<input checked="" type="checkbox"/> age	21			
<input checked="" type="checkbox"/> sex	female			
Key	Value	Description		

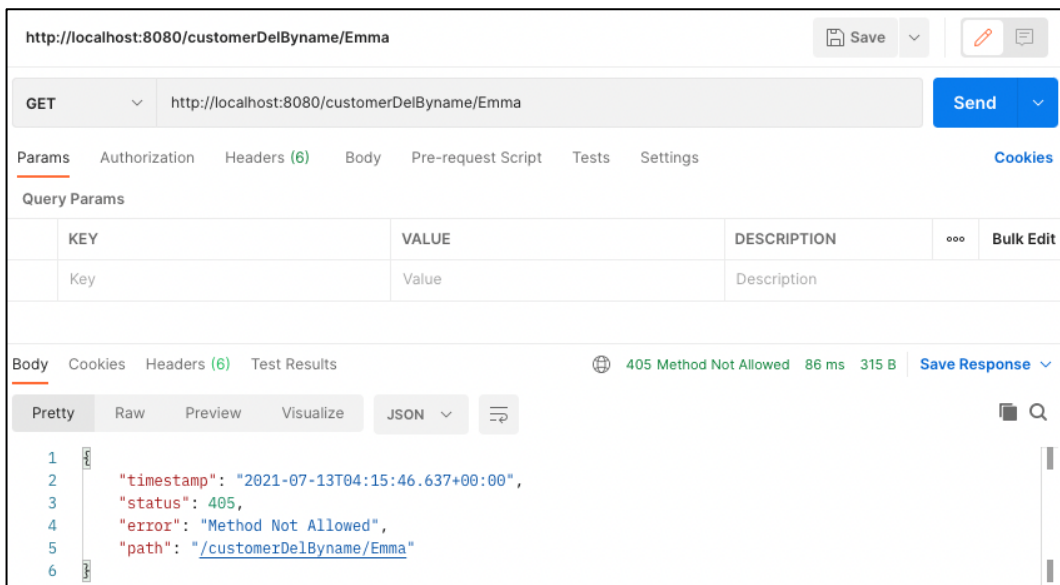
Body Cookies Headers (6) Test Results 405 Method Not Allowed 13 ms 303 B Save Response

Pretty Raw Preview Visualize JSON

```

1  {
2    "timestamp": "2021-07-13T04:35:55.067+00:00",
3    "status": 405,
4    "error": "Method Not Allowed",
5    "path": "/addCustomer2"
6  }

```



`http://localhost:8080/customerDelByname/Emma` Save

GET `http://localhost:8080/customerDelByname/Emma` Send

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION	...	Bulk Edit
Key	Value	Description		

Body Cookies Headers (6) Test Results 405 Method Not Allowed 86 ms 315 B Save Response

Pretty Raw Preview Visualize JSON

```

1  {
2    "timestamp": "2021-07-13T04:15:46.637+00:00",
3    "status": 405,
4    "error": "Method Not Allowed",
5    "path": "/customerDelByname/Emma"
6  }

```