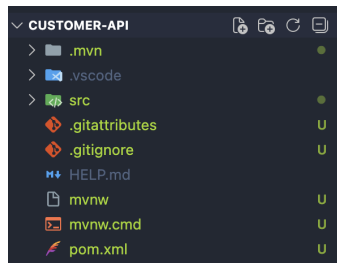


Lab 8

Github site & link

<https://github.com/alexis0704/web-dev-practice-lab/tree/main/Lab8/customer-api>

Task 1.1: Create Spring Boot Project



Configuration:

Spring Boot: 4.0.0

Group: com.example

Artifact: customer-api

Java: 17

Task 1.2, 1.3, 2.1, 2.2, 2.3, 3.1, 3.2, 3.3, 4.1, 4.2: Database Setup, Create Customer Entity, Create Request DTO, Create Response DTO, Create Error Response DTO, Create Repository, Create Service Interface, Implement Service, Create Basic REST Controller, Add Exception Handling

I simply pasted in the given code in the guide.

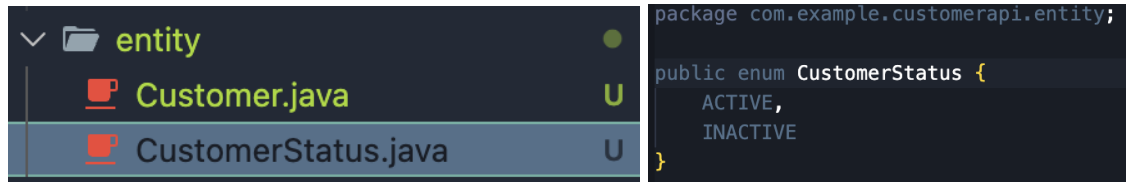
However, there's a bug as follows

```
private CustomerResponseDTO convertToResponseDTO(Customer customer) {
    CustomerResponseDTO dto = new CustomerResponseDTO();
    dto.setId(customer.getId());
    dto.setCustomerCode(customer.getCustomerCode());
    dto.setFullName(customer.getFullName());
    dto.setEmail(customer.getEmail());
    dto.setPhone(customer.getPhone());
    dto.setAddress(customer.getAddress());
    dto.setStatus(customer.getStatus().toString()); Enum.toString() is defined in an inaccessible class or interface
    dto.setCreatedAt(customer.getCreatedAt());
    return dto;
}
```

```
// Enum for status
enum CustomerStatus {
    ACTIVE,
    INACTIVE
}
```

The service couldn't use the method of the CustomerStatus enum, defined in the Customer.java entity file. Here, I think we can either modify the Customer's getStatus() method to implement toString(), or we can create a new CustomerStatus enum class.

I go with the second approach, because it improves maintainability and readability. Also, it avoids modifying the entity's basic method unnecessarily.

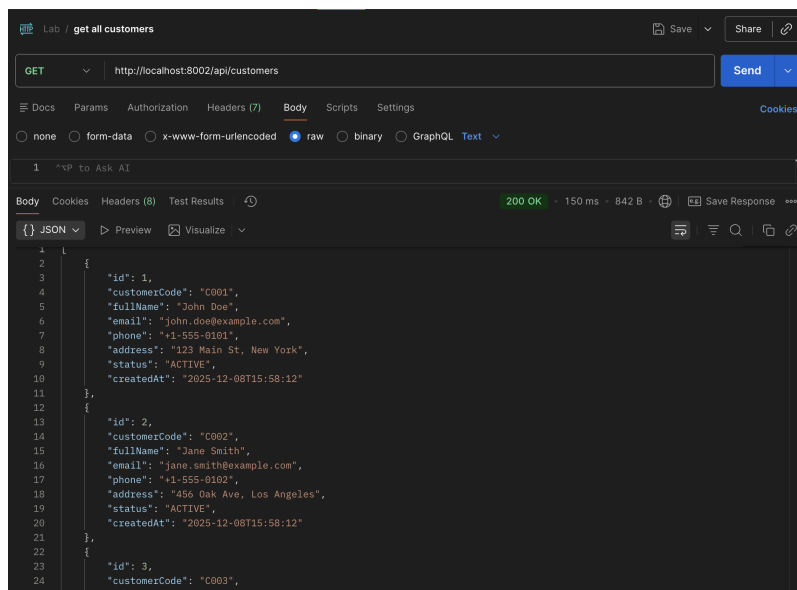


After moving CustomerStatus to its own public enum class, the method shows no bug anymore:



Testing all endpoints with Postman:

Get all customers



```
{
  "id": 3,
  "customerCode": "C003",
  "fullName": "Bob Johnson",
  "email": "bob.johnson@example.com",
  "phone": "+1-555-0103",
  "address": "789 Pine Rd, Chicago",
  "status": "ACTIVE",
  "createdAt": "2025-12-08T15:58:12"
}
```

Get customer by ID

Lab / get customer by ID

GET <http://localhost:8002/api/customers/1> Send

Docs Params Authorization Headers (7) Body Scripts Settings Cookies

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL Text

1 ^XP to Ask AI

Body Cookies Headers (8) Test Results 200 OK - 56 ms - 444 B Save Response

{ } JSON Preview Visualize

```
1 {
2   "id": 1,
3   "customerCode": "C001",
4   "fullName": "John Doe",
5   "email": "john.doe@example.com",
6   "phone": "+1-555-0101",
7   "address": "123 Main St, New York",
8   "status": "ACTIVE",
9   "createdAt": "2025-12-08T15:58:12"
10 }
```

Create new customer

Lab / create new customer

POST <http://localhost:8002/api/customers> Send

Docs Params Authorization Headers (9) Body Scripts Settings Cookies

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL JSON Schema Beautify

```
1 {
2   "customerCode": "C004",
3   "fullName": "Dan",
4   "email": "dan@gmail.com",
5   "phone": "0988888888",
6   "address": "123 Street B, City A",
7   "status": "ACTIVE"
8 }
```

Body Cookies Headers (8) Test Results 201 Created - 48 ms - 442 B Save Response

{ } JSON Preview Visualize

```
1 {
2   "id": 4,
3   "customerCode": "C004",
4   "fullName": "Dan",
5   "email": "dan@gmail.com",
6   "phone": "0988888888",
7   "address": "123 Street B, City A",
8   "status": "ACTIVE",
9   "createdAt": "2025-12-08T09:42:11.531179"
10 }
```

Update Customer

The screenshot shows a Postman client with a PUT request to `http://localhost:8002/api/customers/1`. The request body is a JSON object with the following fields: `customerCode`, `fullName`, `email`, `phone`, `address`, and `status`. The response is a 200 OK status with a JSON body containing the updated customer details, including an `id` and a `createdAt` timestamp.

```
PUT http://localhost:8002/api/customers/1
```

```
{
  "customerCode": "C111",
  "fullName": "JohnJohnJohn",
  "email": "john@gmail.com",
  "phone": "8999999999",
  "address": "123 Street A, City B",
  "status": "ACTIVE"
}
```

200 OK • 109 ms • 440 B

```
{
  "id": 1,
  "customerCode": "C001",
  "fullName": "JohnJohnJohn",
  "email": "john@gmail.com",
  "phone": "8999999999",
  "address": "123 Street A, City B",
  "status": "ACTIVE",
  "createdAt": "2025-12-08T15:58:12"
}
```

Delete customer

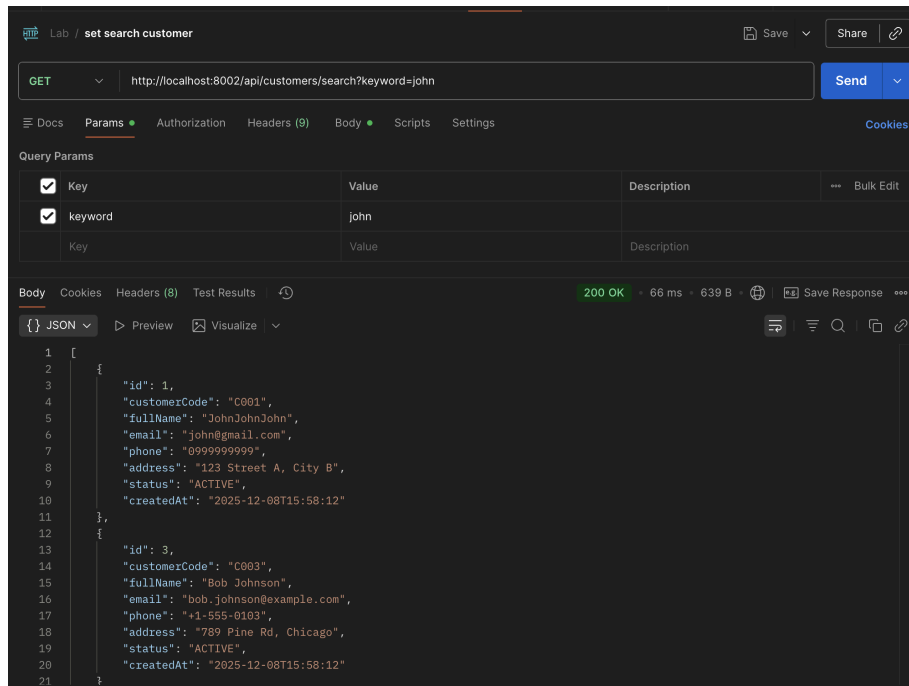
The screenshot shows a Postman client with a DELETE request to `http://localhost:8002/api/customers/4`. The response is a 200 OK status with a JSON body containing a success message.

```
DELETE http://localhost:8002/api/customers/4
```

200 OK • 64 ms • 296 B

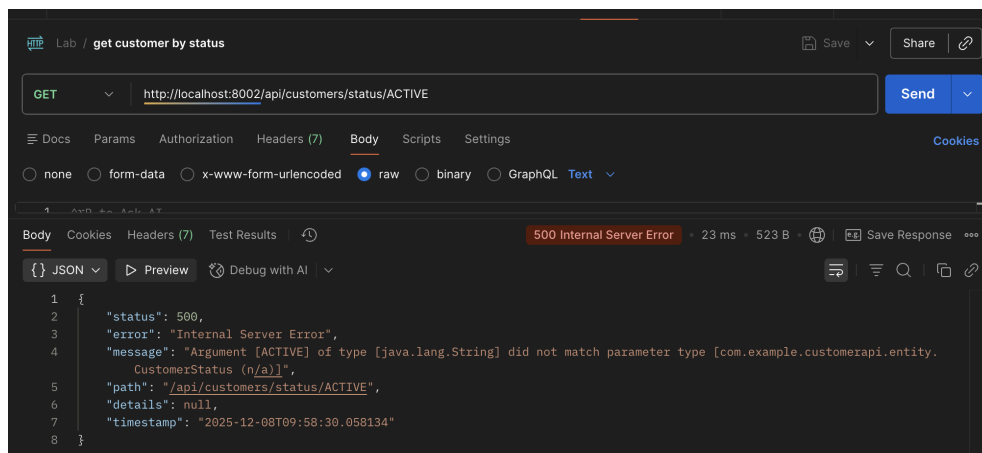
```
{
  "message": "Customer deleted successfully"
}
```

Search customer



Get customer by status

There's a type mismatch bug here for enum.



This is because the entity's attribute is CustomerStatus, but the whole flow from the Controller, to Service and to Repository throw in String status.

To fix the bug, I make the Service class convert the string to the enum type and make the Repository class save the enum:

- Old repository method

```
List<Customer> findByStatus(String status);
```

- New repository method

```
List<Customer> findByStatus(CustomerStatus status);
```

- Old service method

```
@Override
public List<CustomerResponseDTO> getCustomersByStatus(String status) {
    return customerRepository.findByStatus(status).stream().map(this::convertToResponseDTO)
        .collect(Collectors.toList());
}
```

- New service method

```
@Override
public List<CustomerResponseDTO> getCustomersByStatus(String status) {
    CustomerStatus enumStatus = CustomerStatus.valueOf(status.toUpperCase());
    return customerRepository.findByStatus(enumStatus)
        .stream()
        .map(this::convertToResponseDTO)
        .collect(Collectors.toList());
}
```

Result of the bug resolve:

The screenshot shows a REST client interface with the following details:

- Method:** GET
- URL:** http://localhost:8002/api/customers/status/ACTIVE
- Status:** 200 OK
- Time:** 91 ms
- Size:** 838 B
- Body:** A JSON array of three customer objects, all with status "ACTIVE".

```
[
  {
    "id": 1,
    "customerCode": "C001",
    "fullName": "JohnJohnJohn",
    "email": "john@gmail.com",
    "phone": "0999999999",
    "address": "123 Street A, City B",
    "status": "ACTIVE",
    "createdAt": "2025-12-08T15:58:12"
  },
  {
    "id": 2,
    "customerCode": "C002",
    "fullName": "Jane Smith",
    "email": "jane.smith@example.com",
    "phone": "+1-555-0102",
    "address": "456 Oak Ave, Los Angeles",
    "status": "ACTIVE",
    "createdAt": "2025-12-08T15:58:12"
  },
  {
    "id": 3,
    "customerCode": "C003",
    "fullName": "Bob Johnson",
    "email": "bob.johnson@example.com"
  }
]
```