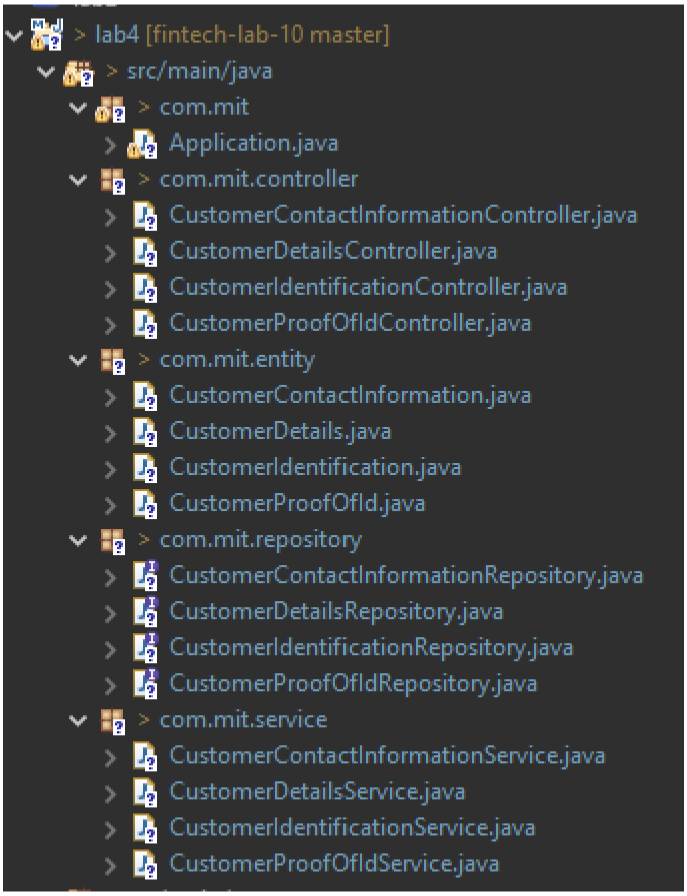
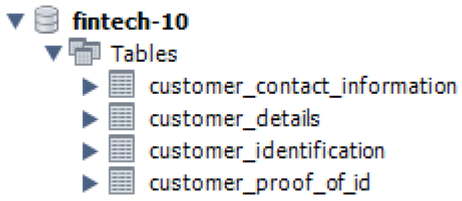
LAB 4

My file hierarchy after completing lab 4 for reference:



```
lab4 [fintech-lab-10 master]
  src/main/java
    com.mit
      Application.java
    com.mit.controller
      CustomerContactInformationController.java
      CustomerDetailsController.java
      CustomerIdentificationController.java
      CustomerProofOfIdController.java
    com.mit.entity
      CustomerContactInformation.java
      CustomerDetails.java
      CustomerIdentification.java
      CustomerProofOfId.java
    com.mit.repository
      CustomerContactInformationRepository.java
      CustomerDetailsRepository.java
      CustomerIdentificationRepository.java
      CustomerProofOfIdRepository.java
    com.mit.service
      CustomerContactInformationService.java
      CustomerDetailsService.java
      CustomerIdentificationService.java
      CustomerProofOfIdService.java
```

My SQL tables (auto-generated by Spring Boot):



```
fintech-10
  Tables
    customer_contact_information
    customer_details
    customer_identification
    customer_proof_of_id
```

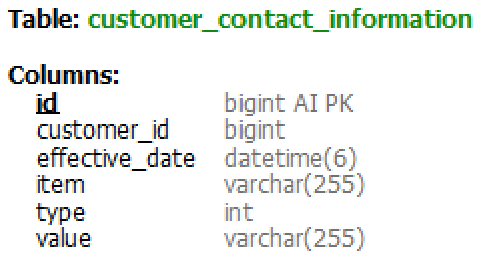


Table: customer_contact_information

Columns:

id	bigint AI PK
customer_id	bigint
effective_date	datetime(6)
item	varchar(255)
type	int
value	varchar(255)

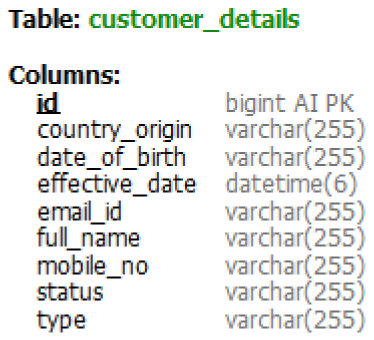


Table: customer_details

Columns:

id	bigint AI PK
country_origin	varchar(255)
date_of_birth	varchar(255)
effective_date	datetime(6)
email_id	varchar(255)
full_name	varchar(255)
mobile_no	varchar(255)
status	varchar(255)
type	varchar(255)

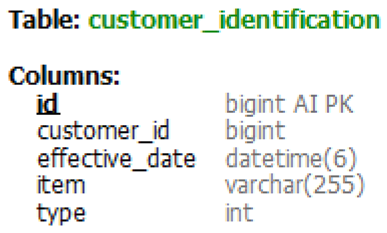


Table: customer_identification

Columns:

id	bigint AI PK
customer_id	bigint
effective_date	datetime(6)
item	varchar(255)
type	int

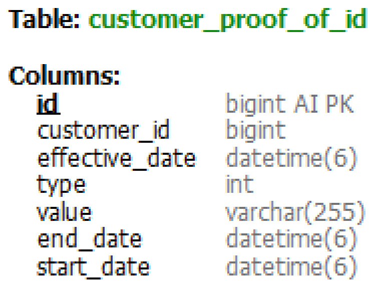


Table: customer_proof_of_id

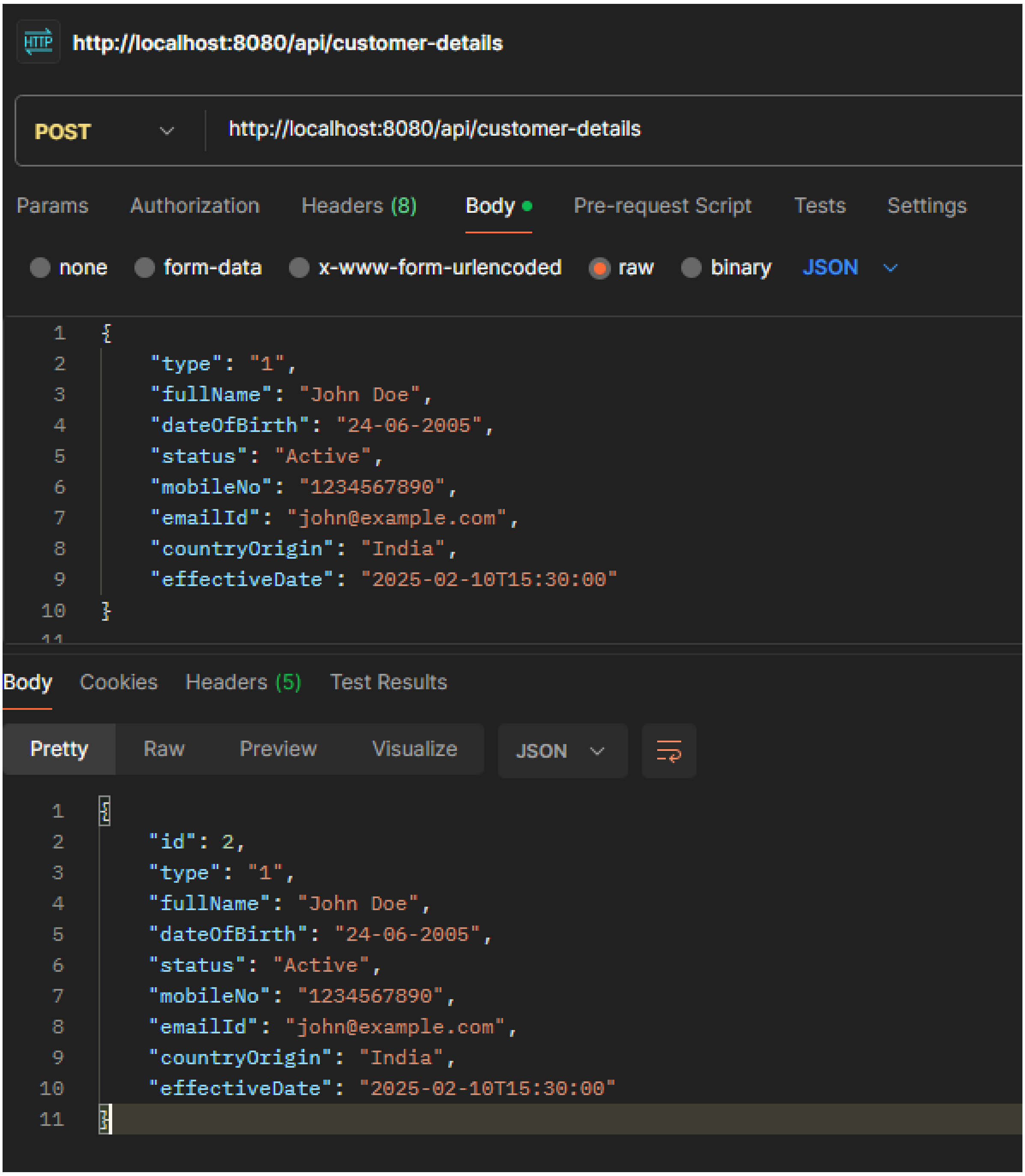
Columns:

id	bigint AI PK
customer_id	bigint
effective_date	datetime(6)
type	int
value	varchar(255)
end_date	datetime(6)
start_date	datetime(6)

QUESTIONS

1. **Establish Spring Boot MySQL Connection with CustomerDetail table**

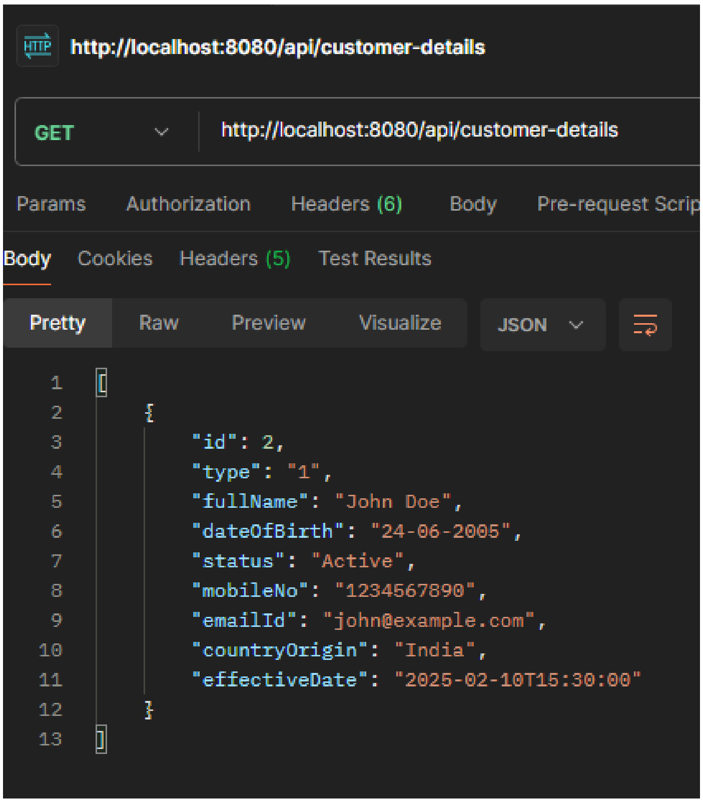
Part 1: POST Request



```
{
    "type": "1",
    "fullName": "John Doe",
    "dateOfBirth": "24-06-2005",
    "status": "Active",
    "mobileNo": "1234567890",
    "emailId": "john@example.com",
    "countryOrigin": "India",
    "effectiveDate": "2025-02-10T15:30:00"
}
```

```
{
    "id": 2,
    "type": "1",
    "fullName": "John Doe",
    "dateOfBirth": "24-06-2005",
    "status": "Active",
    "mobileNo": "1234567890",
    "emailId": "john@example.com",
    "countryOrigin": "India",
    "effectiveDate": "2025-02-10T15:30:00"
}
```

Part 2: GET Request (all records)

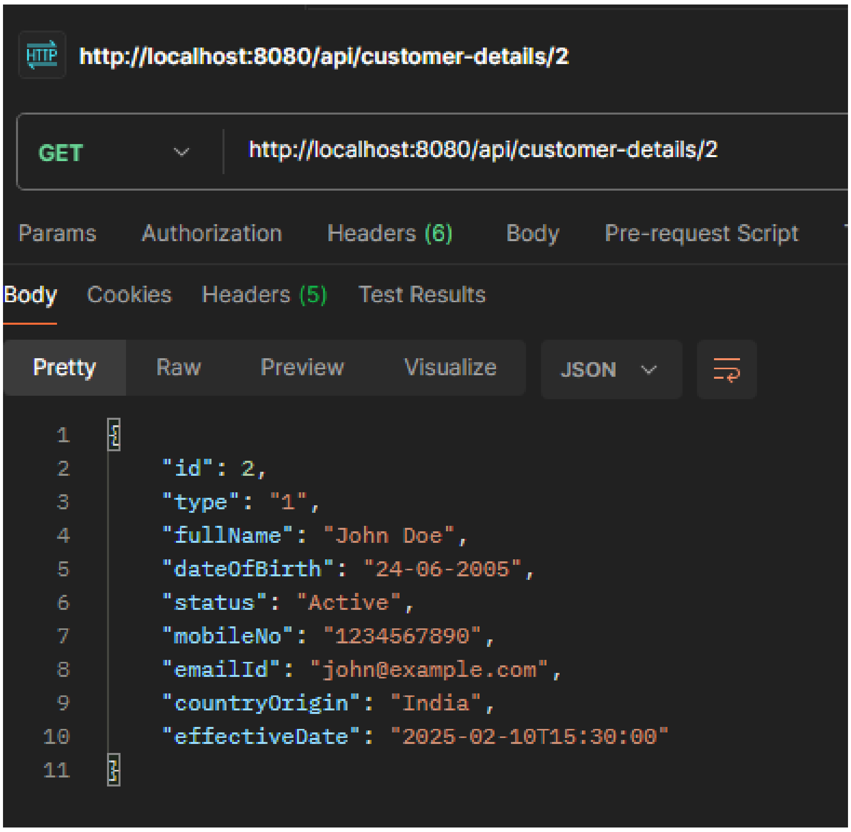


http://localhost:8080/api/customer-details

GET http://localhost:8080/api/customer-details

```
[
    {
        "id": 2,
        "type": "1",
        "fullName": "John Doe",
        "dateOfBirth": "24-06-2005",
        "status": "Active",
        "mobileNo": "1234567890",
        "emailId": "john@example.com",
        "countryOrigin": "India",
        "effectiveDate": "2025-02-10T15:30:00"
    }
]
```

Part 3: GET Request (by ID)



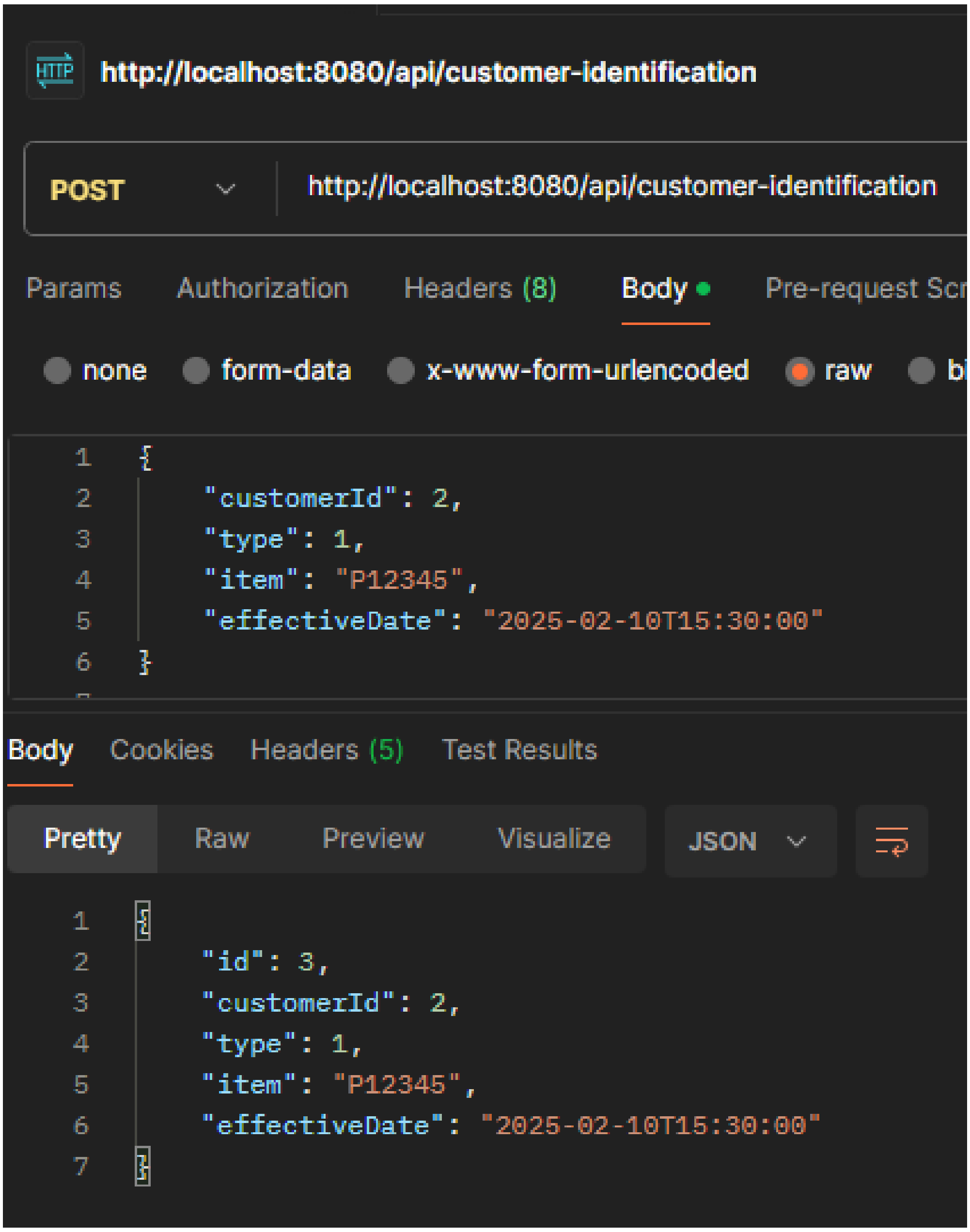
http://localhost:8080/api/customer-details/2

GET http://localhost:8080/api/customer-details/2

```
{
    "id": 2,
    "type": "1",
    "fullName": "John Doe",
    "dateOfBirth": "24-06-2005",
    "status": "Active",
    "mobileNo": "1234567890",
    "emailId": "john@example.com",
    "countryOrigin": "India",
    "effectiveDate": "2025-02-10T15:30:00"
}
```

2. Establish Spring Boot MySQL Connection with CustomerIdentification table

Part 1: POST Request

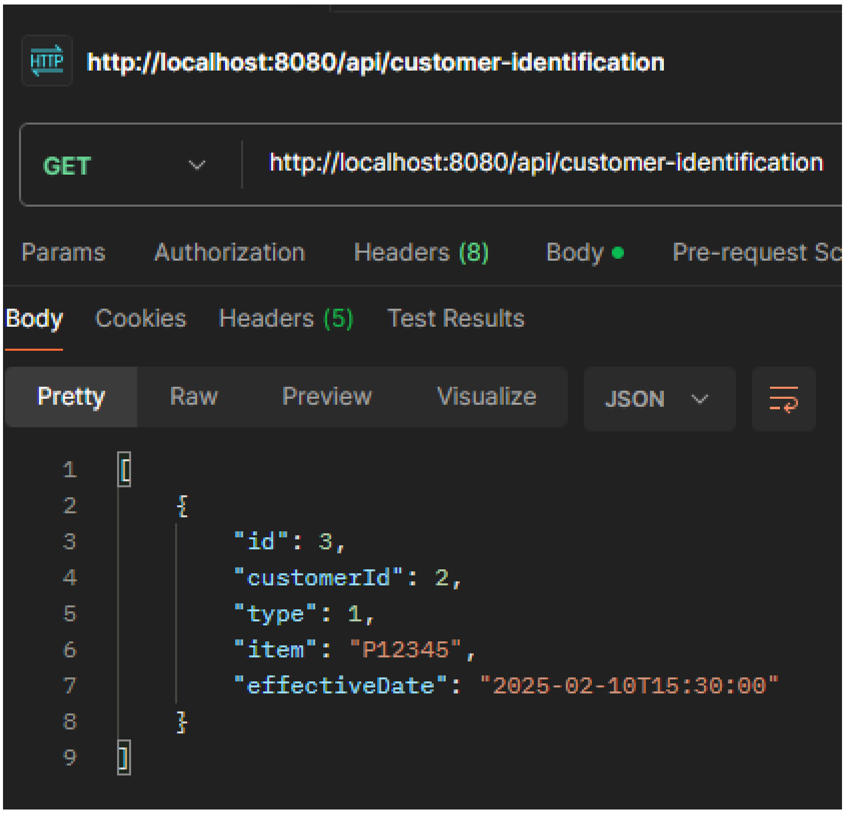


POST http://localhost:8080/api/customer-identification

```
{
    "customerId": 2,
    "type": 1,
    "item": "P12345",
    "effectiveDate": "2025-02-10T15:30:00"
}
```

```
{
    "id": 3,
    "customerId": 2,
    "type": 1,
    "item": "P12345",
    "effectiveDate": "2025-02-10T15:30:00"
}
```

Part 2: GET Request (all records)

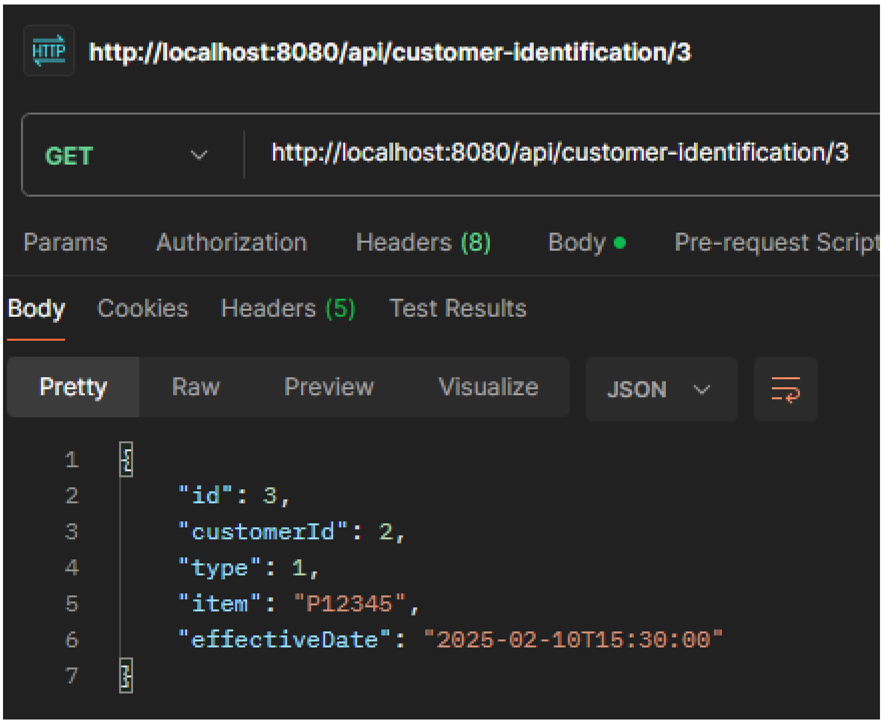


http://localhost:8080/api/customer-identification

GET http://localhost:8080/api/customer-identification

```
[
    {
        "id": 3,
        "customerId": 2,
        "type": 1,
        "item": "P12345",
        "effectiveDate": "2025-02-10T15:30:00"
    }
]
```

Part 3: GET Request (by ID)



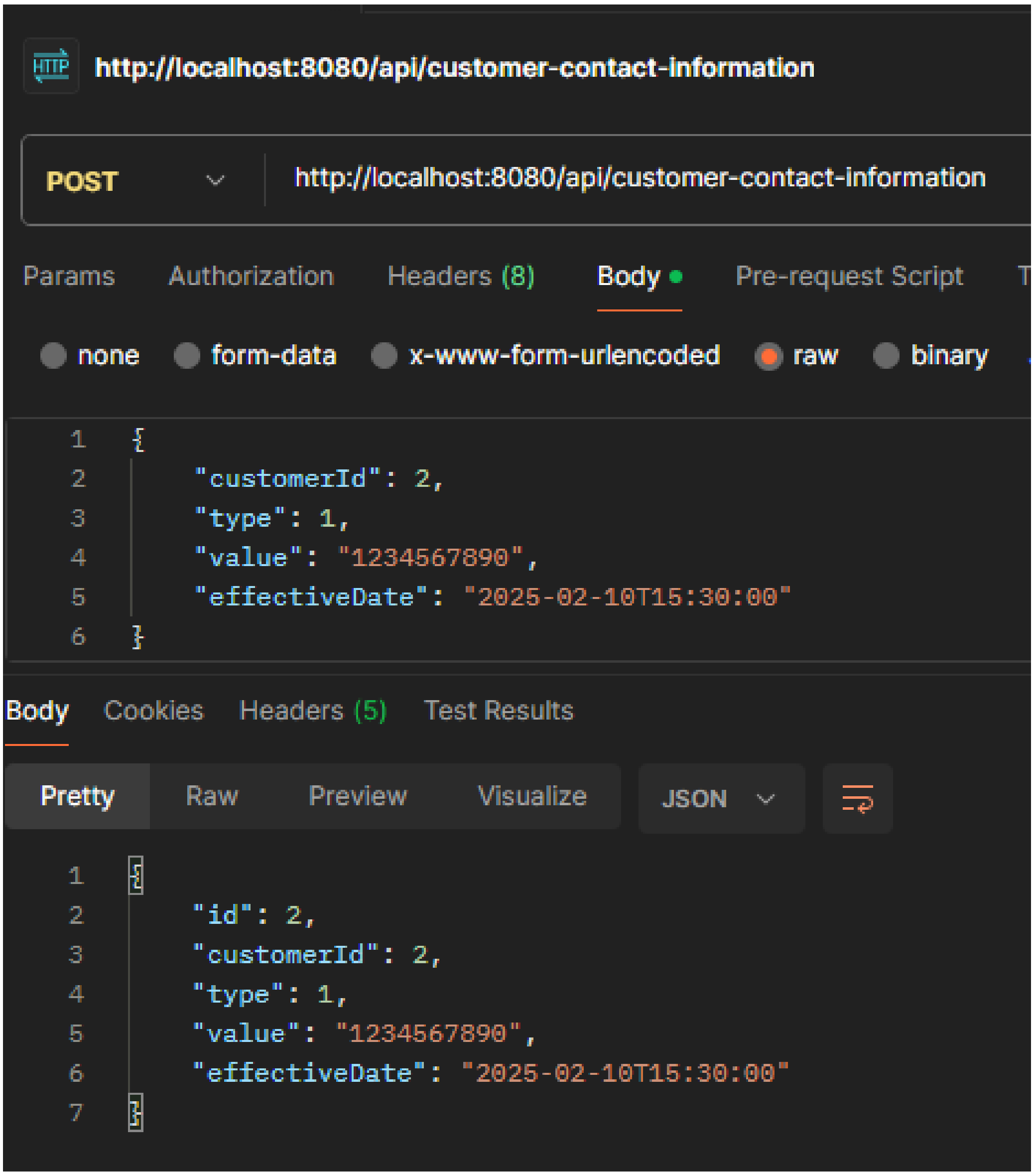
http://localhost:8080/api/customer-identification/3

GET http://localhost:8080/api/customer-identification/3

```
{
    "id": 3,
    "customerId": 2,
    "type": 1,
    "item": "P12345",
    "effectiveDate": "2025-02-10T15:30:00"
}
```

3. Establish Spring Boot MySQL Connection with CustomerContactInformation table

Part 1: POST Request

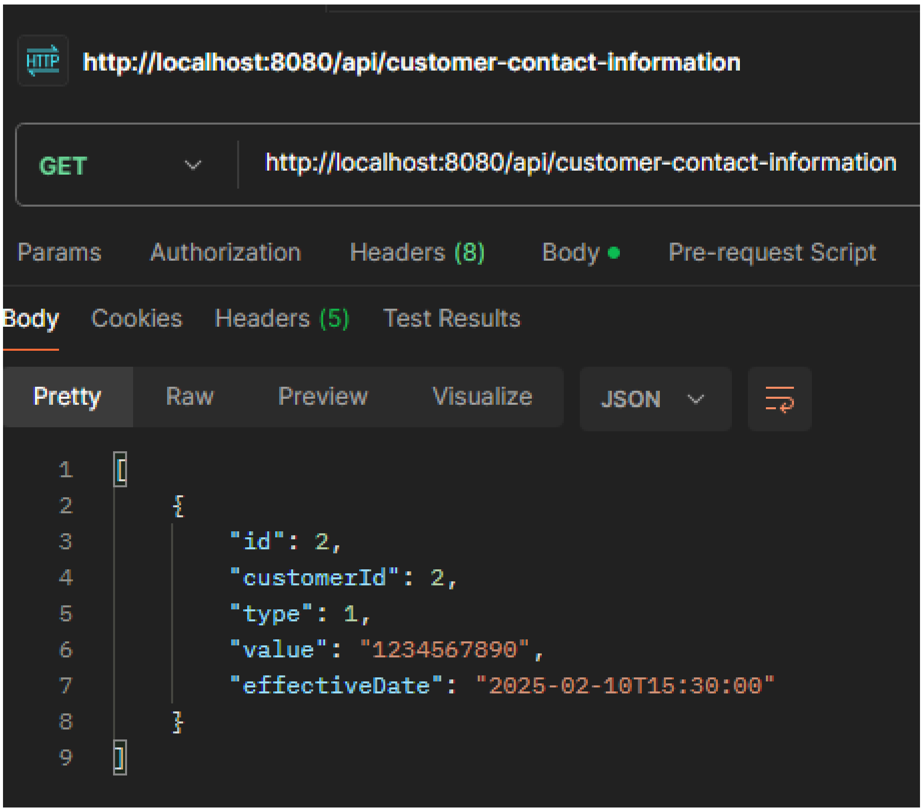


POST http://localhost:8080/api/customer-contact-information

```
{
    "customerId": 2,
    "type": 1,
    "value": "1234567890",
    "effectiveDate": "2025-02-10T15:30:00"
}
```

```
{
    "id": 2,
    "customerId": 2,
    "type": 1,
    "value": "1234567890",
    "effectiveDate": "2025-02-10T15:30:00"
}
```

Part 2: GET Request (all records)

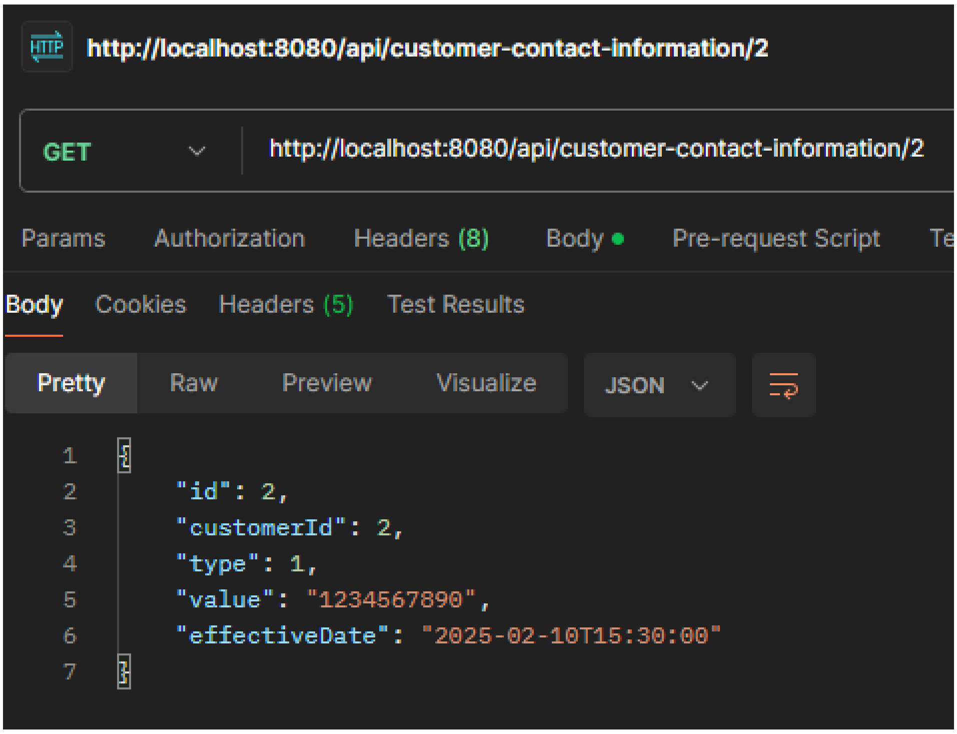


http://localhost:8080/api/customer-contact-information

GET http://localhost:8080/api/customer-contact-information

```
[
    {
        "id": 2,
        "customerId": 2,
        "type": 1,
        "value": "1234567890",
        "effectiveDate": "2025-02-10T15:30:00"
    }
]
```

Part 3: GET Request (by ID)



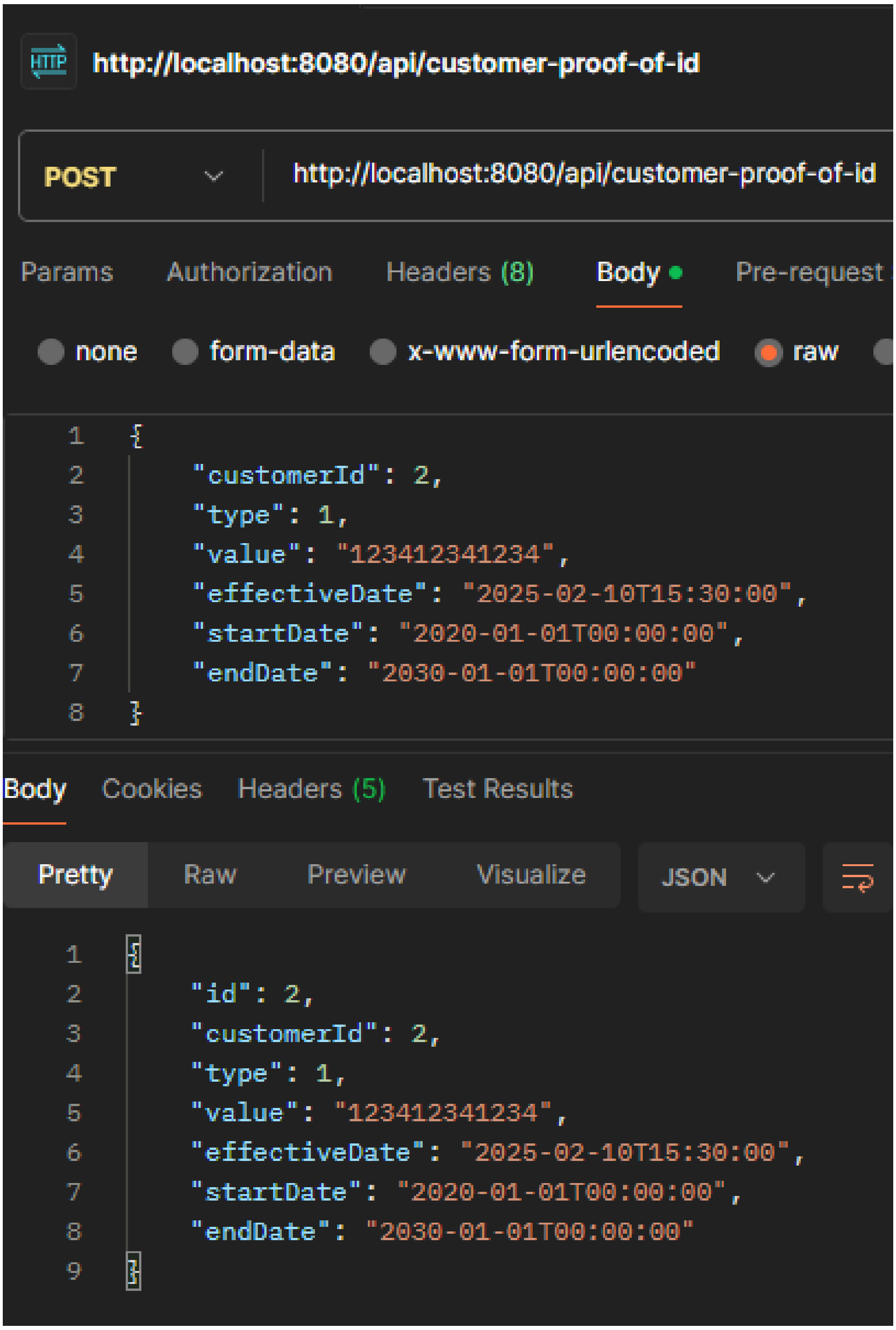
http://localhost:8080/api/customer-contact-information/2

GET http://localhost:8080/api/customer-contact-information/2

```
{
    "id": 2,
    "customerId": 2,
    "type": 1,
    "value": "1234567890",
    "effectiveDate": "2025-02-10T15:30:00"
}
```

4. Establish Spring Boot MySQL Connection with CustomerProofOfId table

Part 1: POST Request



http://localhost:8080/api/customer-proof-of-id

POST http://localhost:8080/api/customer-proof-of-id

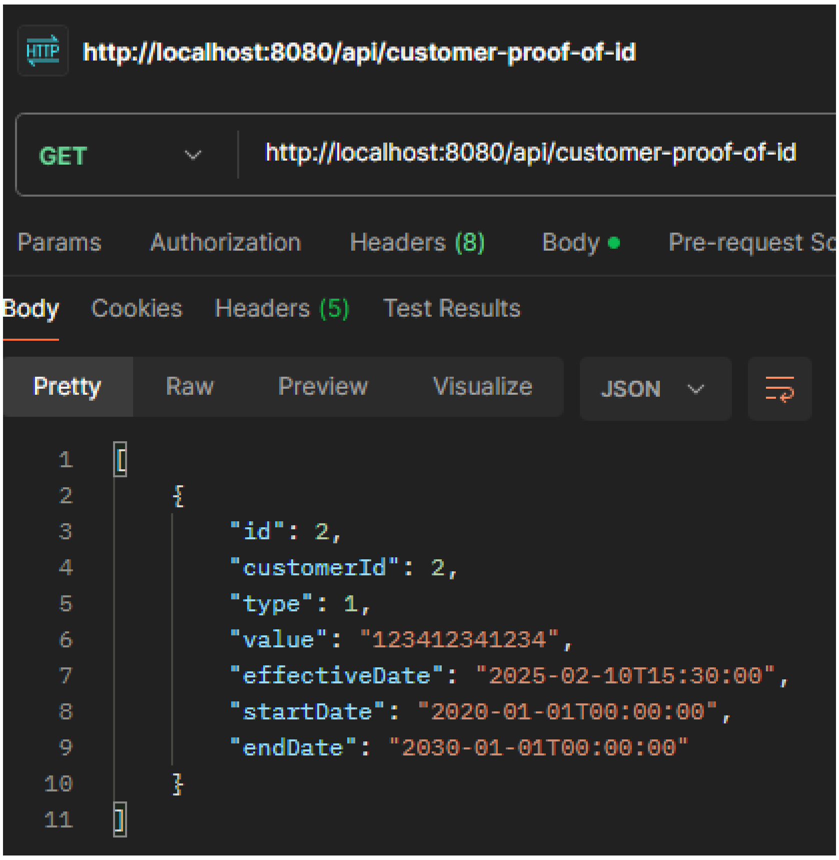
○ none ○ form-data ○ x-www-form-urlencoded ● raw

```
{
    "customerId": 2,
    "type": 1,
    "value": "123412341234",
    "effectiveDate": "2025-02-10T15:30:00",
    "startDate": "2020-01-01T00:00:00",
    "endDate": "2030-01-01T00:00:00"
}
```

Body

```
{
    "id": 2,
    "customerId": 2,
    "type": 1,
    "value": "123412341234",
    "effectiveDate": "2025-02-10T15:30:00",
    "startDate": "2020-01-01T00:00:00",
    "endDate": "2030-01-01T00:00:00"
}
```

Part 2: GET Request (all records)

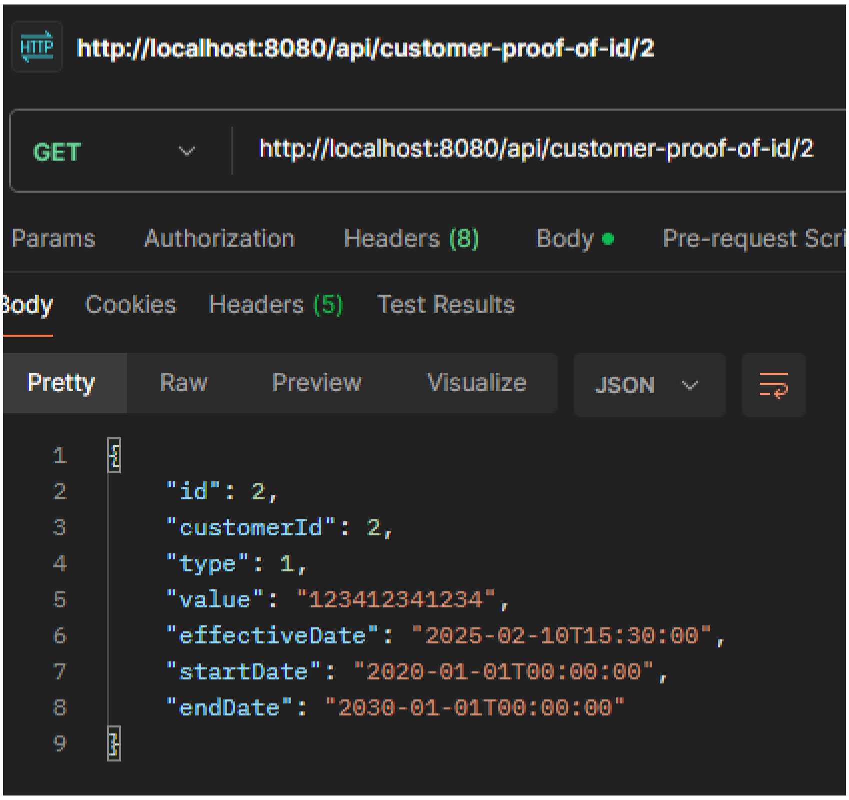


http://localhost:8080/api/customer-proof-of-id

GET http://localhost:8080/api/customer-proof-of-id

```
[
    {
        "id": 2,
        "customerId": 2,
        "type": 1,
        "value": "123412341234",
        "effectiveDate": "2025-02-10T15:30:00",
        "startDate": "2020-01-01T00:00:00",
        "endDate": "2030-01-01T00:00:00"
    }
]
```

Part 3: GET Request (by ID)



http://localhost:8080/api/customer-proof-of-id/2

GET http://localhost:8080/api/customer-proof-of-id/2

```
{
    "id": 2,
    "customerId": 2,
    "type": 1,
    "value": "123412341234",
    "effectiveDate": "2025-02-10T15:30:00",
    "startDate": "2020-01-01T00:00:00",
    "endDate": "2030-01-01T00:00:00"
}
```