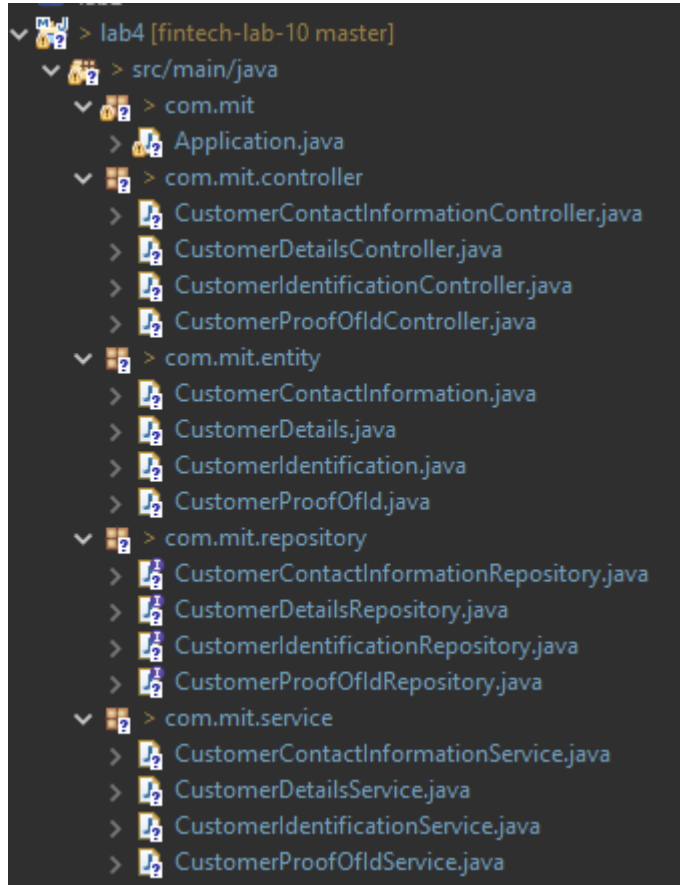


LAB 4

My file hierarchy after completing lab 4 for reference:



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

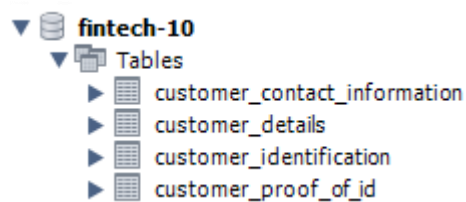


Table: customer_contact_information

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

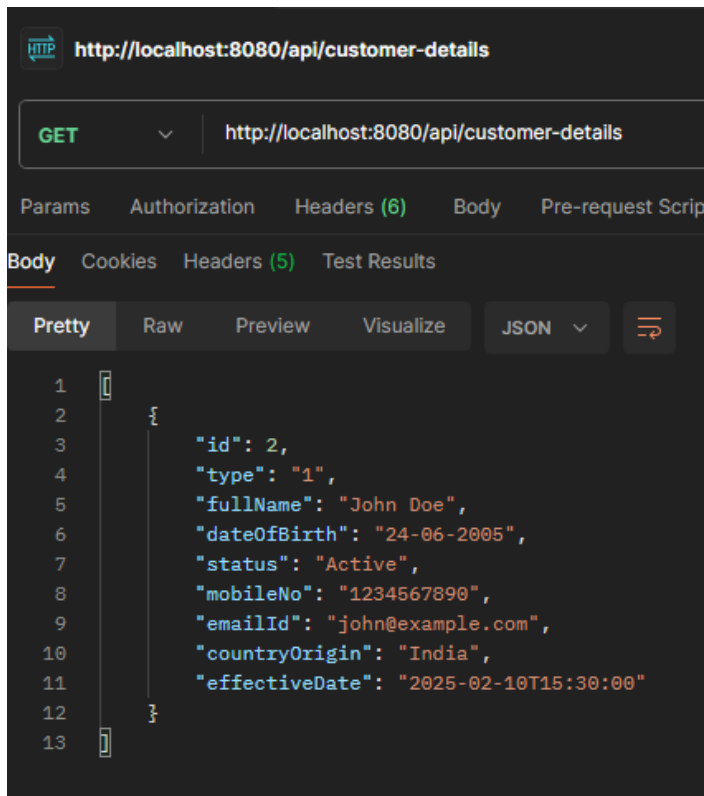
The screenshot displays a REST client interface with the following details:

- URL:** `http://localhost:8080/api/customer-details`
- Method:** **POST**
- Body Type:** **JSON** (selected from a dropdown menu that also includes none, form-data, x-www-form-urlencoded, raw, and binary).
- Body Content:**

```
1  {  
2    "type": "1",  
3    "fullName": "John Doe",  
4    "dateOfBirth": "24-06-2005",  
5    "status": "Active",  
6    "mobileNo": "1234567890",  
7    "emailId": "john@example.com",  
8    "countryOrigin": "India",  
9    "effectiveDate": "2025-02-10T15:30:00"  
10 }  
11
```
- Response Section:** Includes tabs for Body, Cookies, Headers (5), and Test Results. The **Body** tab is active, showing the response in **Pretty** JSON format:

```
1  {  
2    "id": 2,  
3    "type": "1",  
4    "fullName": "John Doe",  
5    "dateOfBirth": "24-06-2005",  
6    "status": "Active",  
7    "mobileNo": "1234567890",  
8    "emailId": "john@example.com",  
9    "countryOrigin": "India",  
10   "effectiveDate": "2025-02-10T15:30:00"  
11 }
```

Part 2: GET Request (all records)




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

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

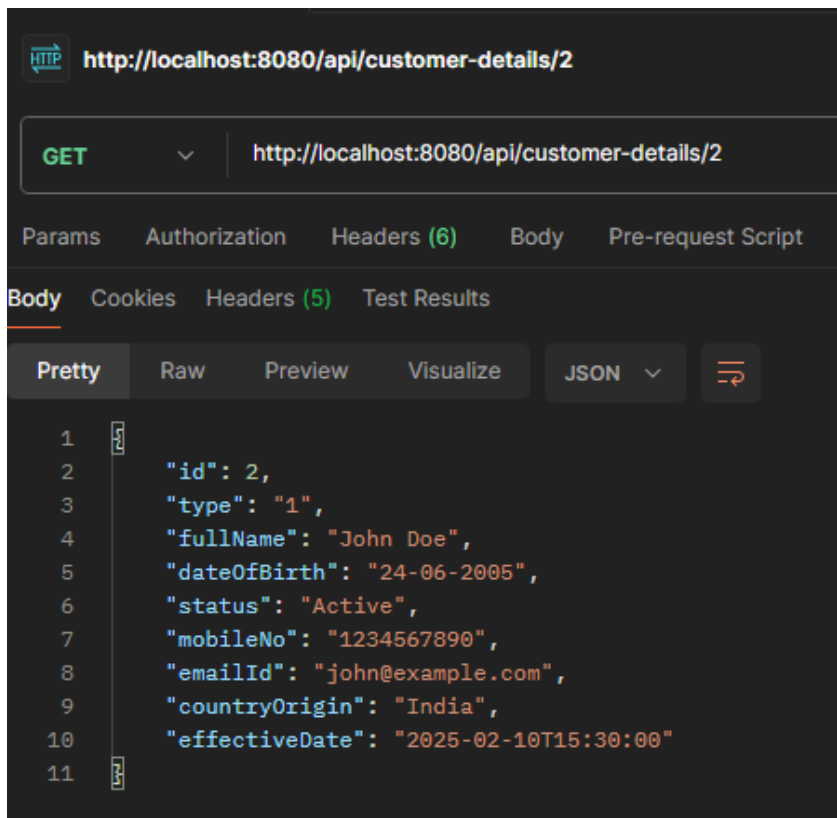
Params Authorization Headers (6) Body Pre-request Script

Body Cookies Headers (5) Test Results

Pretty Raw Preview Visualize JSON 

```
1 [
2   {
3     "id": 2,
4     "type": "1",
5     "fullName": "John Doe",
6     "dateOfBirth": "24-06-2005",
7     "status": "Active",
8     "mobileNo": "1234567890",
9     "emailId": "john@example.com",
10    "countryOrigin": "India",
11    "effectiveDate": "2025-02-10T15:30:00"
12  }
13 ]
```

Part 3: GET Request (by ID)




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

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

Params Authorization Headers (6) Body Pre-request Script

Body Cookies Headers (5) Test Results

Pretty Raw Preview Visualize JSON 

```
1 {
2   "id": 2,
3   "type": "1",
4   "fullName": "John Doe",
5   "dateOfBirth": "24-06-2005",
6   "status": "Active",
7   "mobileNo": "1234567890",
8   "emailId": "john@example.com",
9   "countryOrigin": "India",
10  "effectiveDate": "2025-02-10T15:30:00"
11 }
```

2. Establish Spring Boot MySQL Connection with CustomerIdentification table

Part 1: POST Request

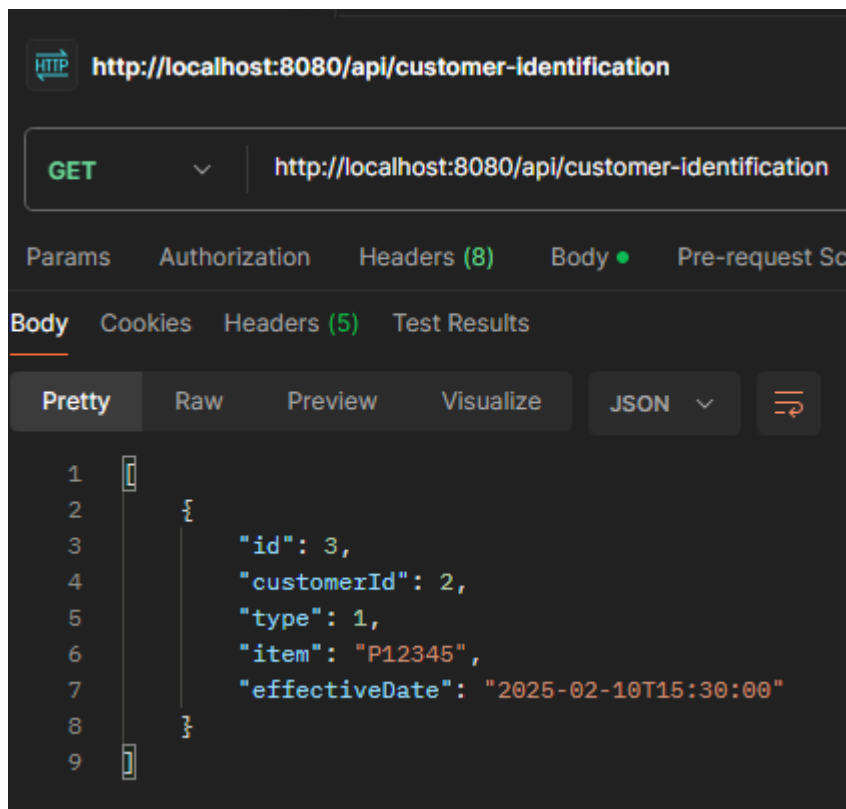
The screenshot displays a REST client interface for a POST request to `http://localhost:8080/api/customer-identification`. The request body is a JSON object with the following structure:

```
1 {  
2   "customerId": 2,  
3   "type": 1,  
4   "item": "P12345",  
5   "effectiveDate": "2025-02-10T15:30:00"  
6 }
```

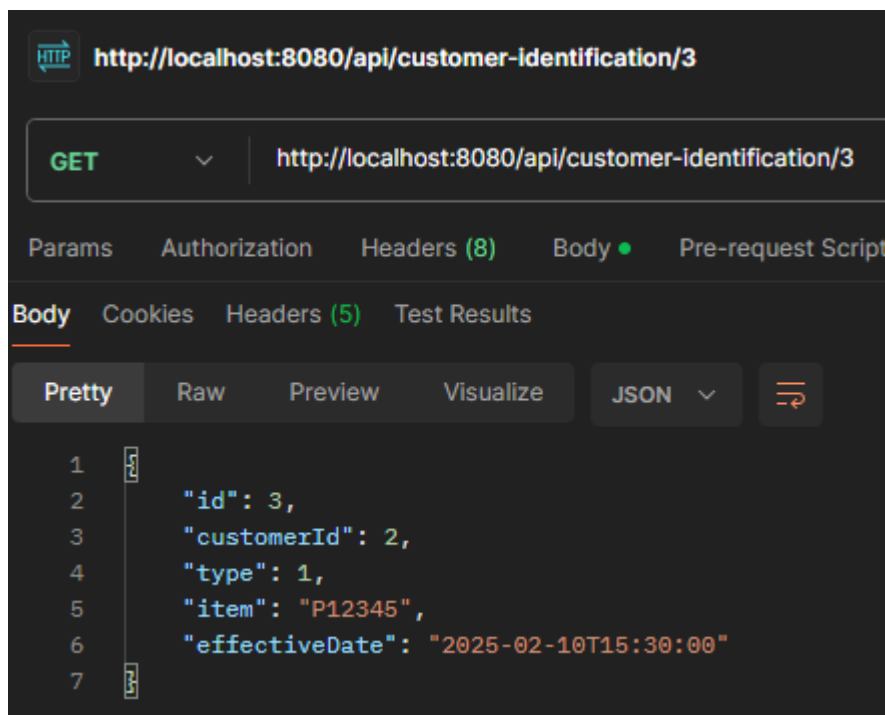
The response body is also shown in a pretty-printed JSON format:

```
1 {  
2   "id": 3,  
3   "customerId": 2,  
4   "type": 1,  
5   "item": "P12345",  
6   "effectiveDate": "2025-02-10T15:30:00"  
7 }
```

Part 2: GET Request (all records)



Part 3: GET Request (by ID)



3. Establish Spring Boot MySQL Connection with CustomerContactInformation table

Part 1: POST Request

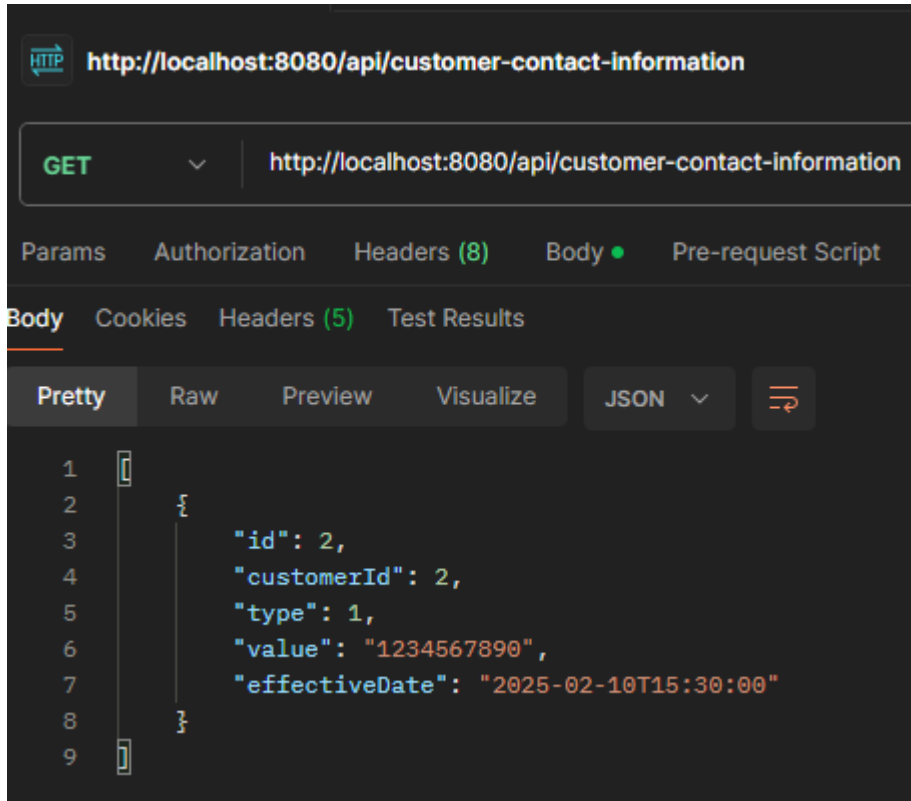
The screenshot displays a REST client interface for a POST request. The URL is `http://localhost:8080/api/customer-contact-information`. The request method is **POST**. The request body is set to **raw** and contains the following JSON:

```
1 {  
2   "customerId": 2,  
3   "type": 1,  
4   "value": "1234567890",  
5   "effectiveDate": "2025-02-10T15:30:00"  
6 }
```

The response section shows the **Body** tab selected, displaying the response in **Pretty** JSON format:

```
1 {  
2   "id": 2,  
3   "customerId": 2,  
4   "type": 1,  
5   "value": "1234567890",  
6   "effectiveDate": "2025-02-10T15:30:00"  
7 }
```

Part 2: GET Request (all records)




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

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

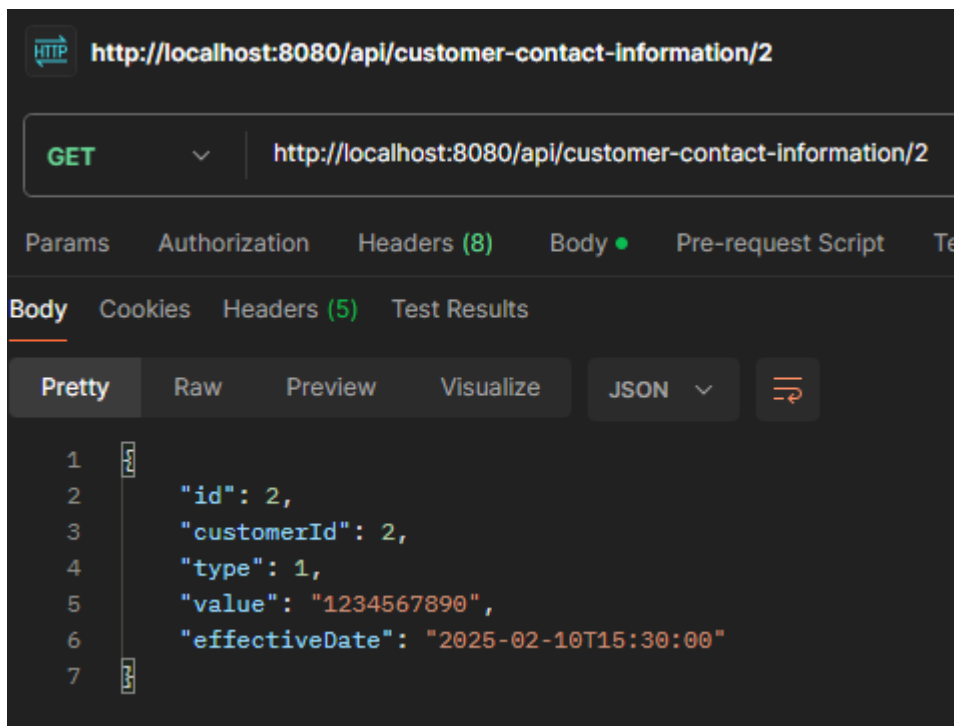
Params Authorization Headers (8) Body • Pre-request Script

Body Cookies Headers (5) Test Results

Pretty Raw Preview Visualize JSON 

```
1 []
2   {
3     "id": 2,
4     "customerId": 2,
5     "type": 1,
6     "value": "1234567890",
7     "effectiveDate": "2025-02-10T15:30:00"
8   }
9 []
```

Part 3: GET Request (by ID)




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

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

Params Authorization Headers (8) Body • Pre-request Script Test

Body Cookies Headers (5) Test Results

Pretty Raw Preview Visualize JSON 

```
1 {
2   "id": 2,
3   "customerId": 2,
4   "type": 1,
5   "value": "1234567890",
6   "effectiveDate": "2025-02-10T15:30:00"
7 }
```


4. Establish Spring Boot MySQL Connection with CustomerProofOfId table

Part 1: POST Request

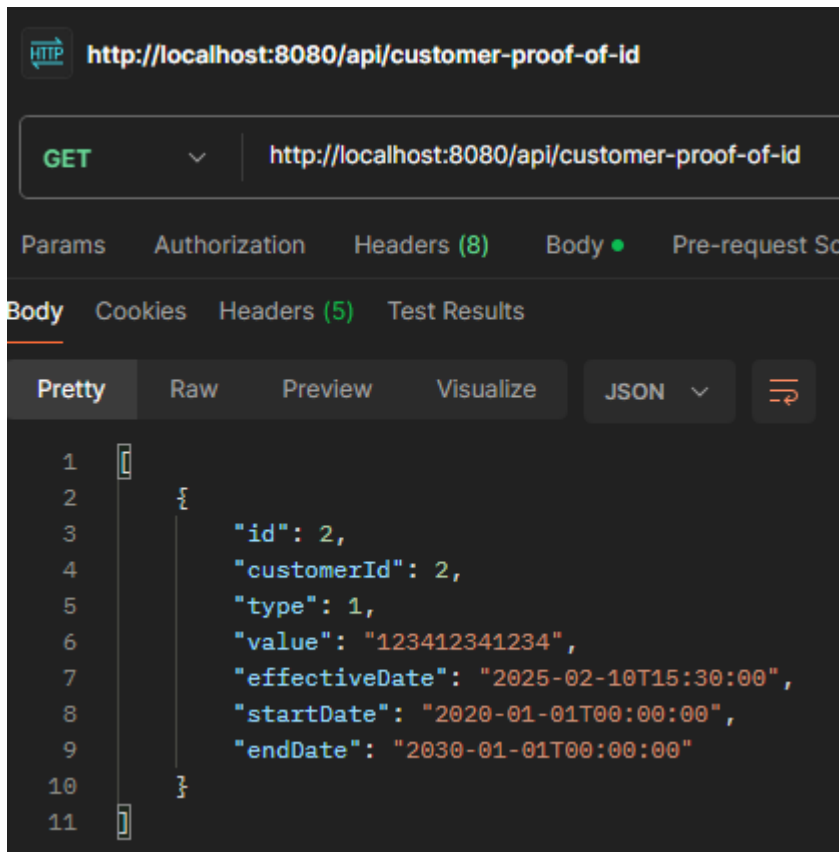
The screenshot displays a REST client interface for a POST request to `http://localhost:8080/api/customer-proof-of-id`. The request body is a JSON object with the following fields:

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

The response is shown in the 'Body' tab, formatted as JSON. It includes an additional `id` field with the value 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"
}
```

Part 2: GET Request (all records)



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

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

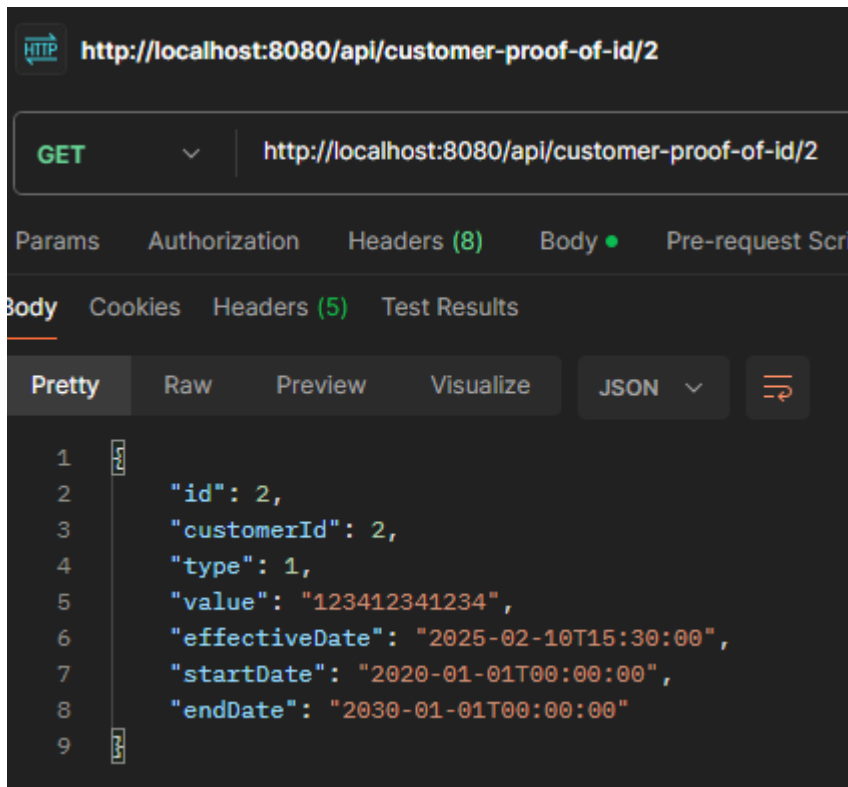
Params Authorization Headers (8) Body ● Pre-request Script

Body Cookies Headers (5) Test Results

Pretty Raw Preview Visualize JSON

```
1 [
2   {
3     "id": 2,
4     "customerId": 2,
5     "type": 1,
6     "value": "123412341234",
7     "effectiveDate": "2025-02-10T15:30:00",
8     "startDate": "2020-01-01T00:00:00",
9     "endDate": "2030-01-01T00:00:00"
10  }
11 ]
```

Part 3: GET Request (by ID)



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

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

Params Authorization Headers (8) Body ● Pre-request Script

Body Cookies Headers (5) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": 2,
3   "customerId": 2,
4   "type": 1,
5   "value": "123412341234",
6   "effectiveDate": "2025-02-10T15:30:00",
7   "startDate": "2020-01-01T00:00:00",
8   "endDate": "2030-01-01T00:00:00"
9 }
```