

INGHUB CASE STUDY

DOCUMENTATION

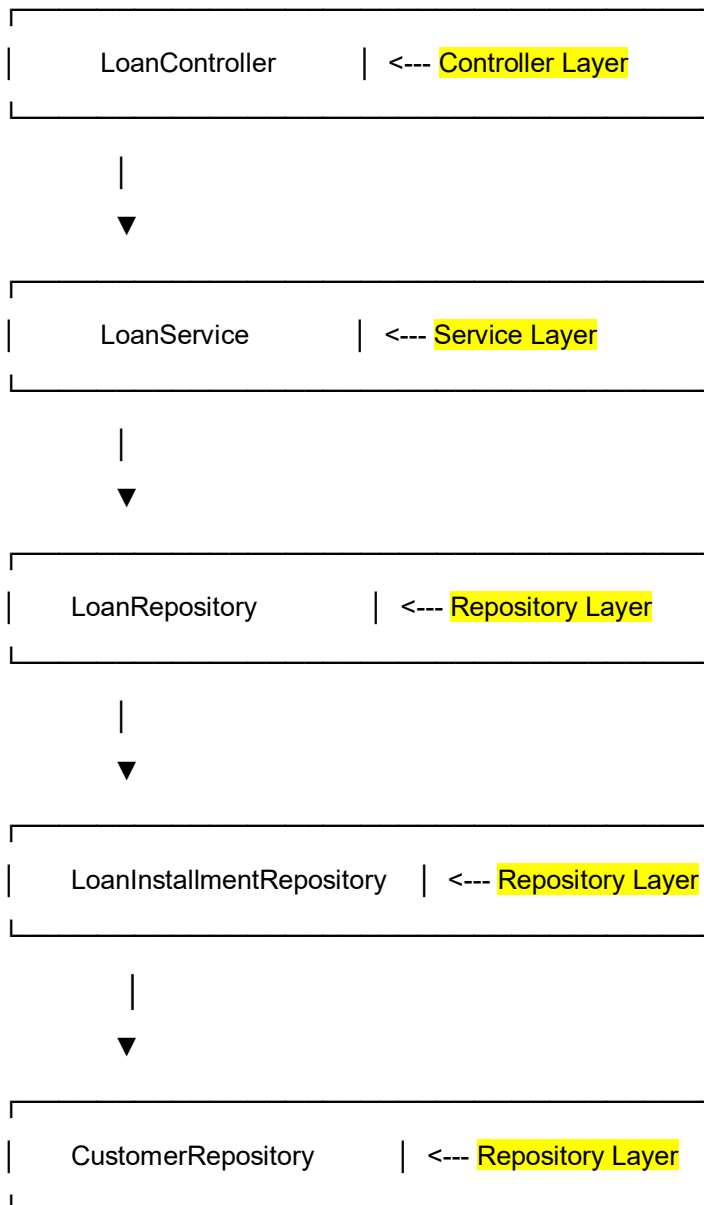
LAYERS:

In the project, we have three main layers:

- **Controller Layer:** Handles HTTP requests and delegates them to the service layer.
- **Service Layer:** Contains business logic and performs data processing.
- **Repository Layer:** Contains database operations

CLASS HIERARCHY:

Below is the class hierarchy and the relationships between the classes:



Project Overview

This application allows bank employees to create loans for customers, view loan installments, and make payments on loans.

Loan Creation: A loan is provided to a customer with a specific amount, interest rate, and installment plan.

Loan Payment: A customer can pay their loan installments. Installments are paid based on priority by due date and payment amount.

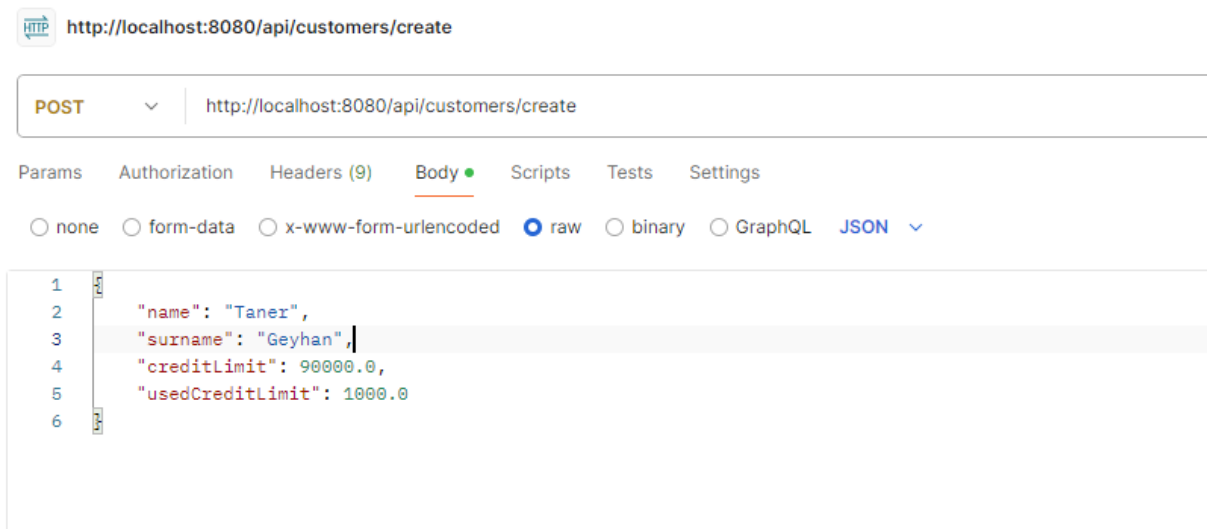
Technologies

- **Java 17**
- **Spring Boot Framework**
- **H2 Database:** In-memory database
- **Maven:** Dependency management and project build tool
- **Spring Data JPA:** Database operations
- **JUnit :** Unit testing
- **Postman:** API testing

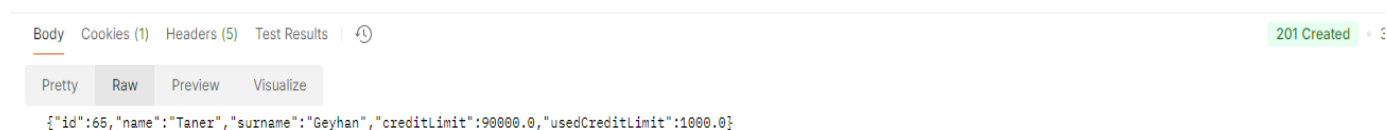
API Endpoints

1. Create Customer

- **URL:** <http://localhost:8080/api/customers/create>
- **Method:** **POST**



Response:



H2 Console:

localhost:8080/h2-console/login.jsp?jsessionid=b44a5878553415982602ff655cb63e41

English Preferences Tools Help

Login

Saved Settings: Generic H2 (Embedded)

Setting Name: Generic H2 (Embedded) Save Remove

Driver Class: org.h2.Driver

JDBC URL: jdbc:h2:file:./data/testdb

User Name: sa

Password:

Connect Test Connection

User Name: sa

Password: password

localhost:8080/h2-console/login.do?jsessionid=b44a5878553415982602ff655cb63e41

Auto commit Max rows: 1000 Auto complete Off Auto select On

jdbc:h2:file:./data/testdb
CUSTOMER
LOANS
LOAN_INSTALLMENTS
INFORMATION_SCHEMA
Users
H2 2.2.224 (2023-09-17)

Run Run Selected Auto complete Clear SQL statement:

SELECT * FROM CUSTOMER

SELECT * FROM CUSTOMER;

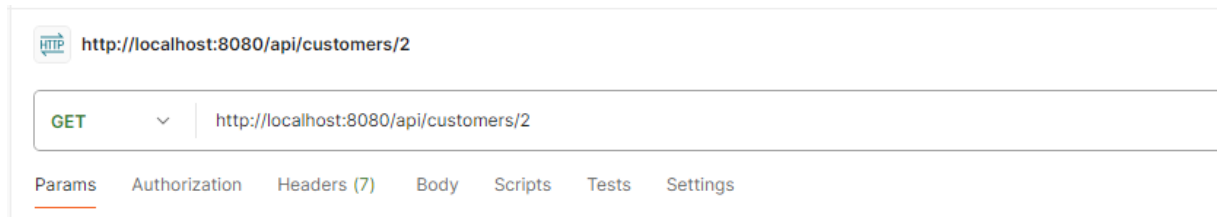
ID	CREDIT_LIMIT	NAME	SURNAME	USED_CREDIT_LIMIT	ROLE
1	75000.0	Ahmet	Mert	21000.0	null
2	90000.0	Taner	Geyhan	1000.0	null
33	90000.0	Taner	Geyhan	1000.0	null
65	90000.0	Taner	Geyhan	1000.0	null

(4 rows, 2 ms)

Edit

2. Get Customer By Id

- **URL:** <http://localhost:8080/api/customers/2>
- **Method:** **GET**




Response:



3. All Customer List

- **URL:** <http://localhost:8080/api/customers/>
- **Method:** **GET**

Response:

 http://localhost:8080/api/customers/

GET

▼

http://localhost:8080/api/customers/

ParamsAuthorizationHeaders (7)BodyScriptsTestsSettings

Query Params

	Key	Value
--	-----	-------

BodyCookies (1)Headers (5)Test Results🔄

PrettyRawPreviewVisualizeJSON ▼🔍

```
1  [  
2    {  
3      "id": 1,  
4      "name": "Ahmet",  
5      "surname": "Mert",  
6      "creditLimit": 75000.0,  
7      "usedCreditLimit": 21000.0  
8    },  
9    {  
10     "id": 2,  
11     "name": "Taner",  
12     "surname": "Geyhan",  
13     "creditLimit": 90000.0,  
14     "usedCreditLimit": 1000.0  
15   },  
16   {  
17     "id": 33,  
18     "name": "Taner",  
19     "surname": "Geyhan",  
20     "creditLimit": 90000.0,  
21     "usedCreditLimit": 1000.0  
22   },  
23   {  
24     "id": 65,  
25     "name": "Taner",  
26     "surname": "Geyhan",  
27     "creditLimit": 90000.0,  
28     "usedCreditLimit": 1000.0  
29   }  
30 ]
```

4. Create Loan

- **URL:**
<http://localhost:8080/api/loans/create?customerId=65&loanAmount=20000&interestRate=0.5&numberOfInstallments=6>
- **Method:** **POST**

Response:

Body Cookies (1) Headers (5) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": 33,
3   "customer": {
4     "id": 65,
5     "name": "Taner",
6     "surname": "Geyhan",
7     "creditLimit": 90000.0,
8     "usedCreditLimit": 21000.0
9   },
10  "loanAmount": 30000.0,
11  "numberOfInstallments": 6,
12  "isPaid": false,
13  "createDate": "2025-01-10"
14 }
```

200 OK

5. List Loans

- **URL:** <http://localhost:8080/api/loans/list?customerId=65>
- **Method:** **GET**

Response:

http://localhost:8080/api/loans/list?customerId=65

GET http://localhost:8080/api/loans/list?customerId=65

Params Authorization Headers (7) Body Scripts Tests Settings

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

This request does not have a body

Body Cookies (1) Headers (5) Test Results

Pretty Raw Preview Visualize JSON

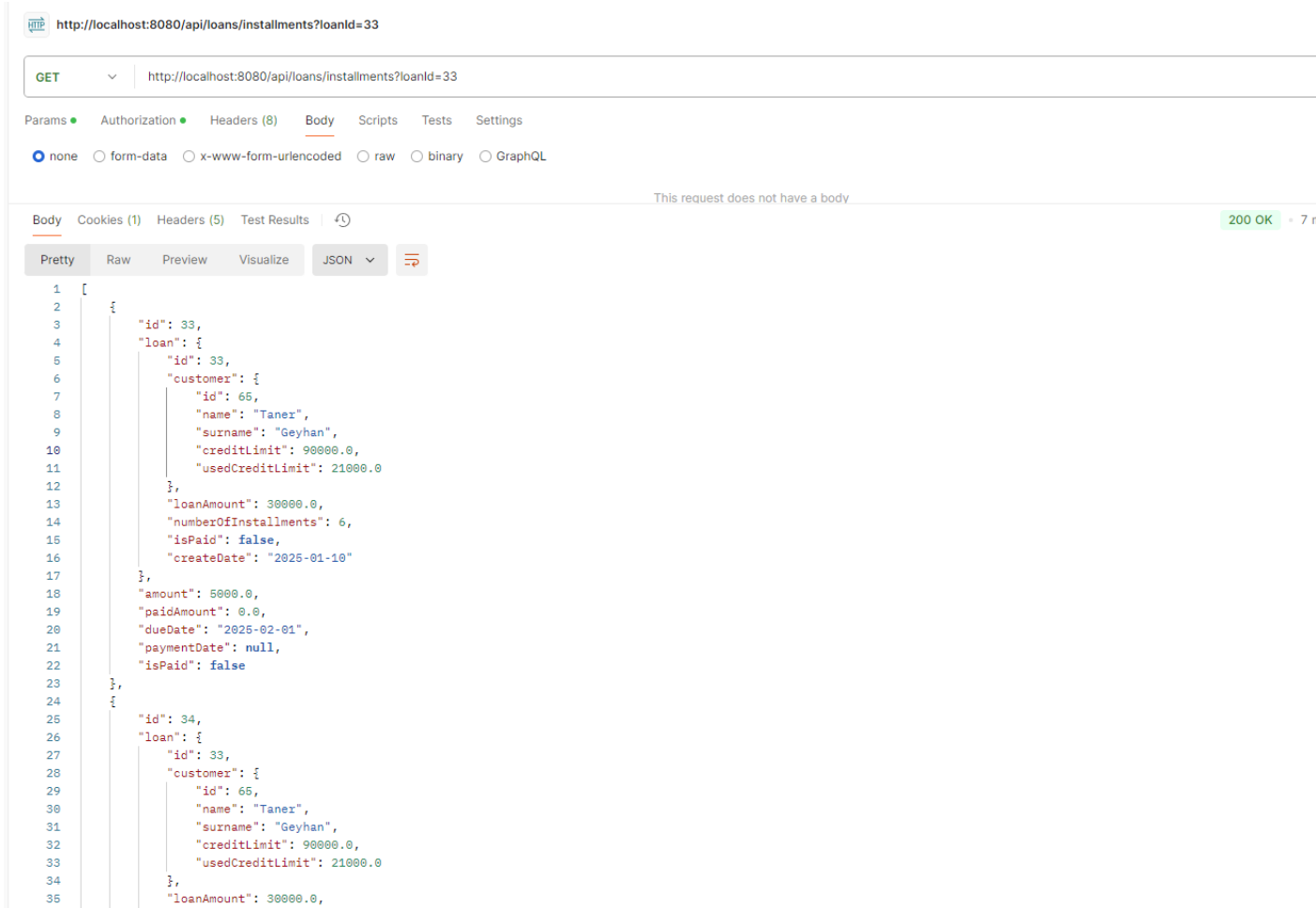
```
1 [
2   {
3     "id": 33,
4     "customer": {
5       "id": 65,
6       "name": "Taner",
7       "surname": "Geyhan",
8       "creditLimit": 90000.0,
9       "usedCreditLimit": 21000.0
10    },
11    "loanAmount": 30000.0,
12    "numberOfInstallments": 6,
13    "isPaid": false,
14    "createDate": "2025-01-10"
15  }
16 ]
```

200 OK

6. List Installments

- **URL:** `http://localhost:8080/api/loans/installments?loanId=33`
- **Method:** **GET**

Response:



http://localhost:8080/api/loans/installments?loanId=33

GET http://localhost:8080/api/loans/installments?loanId=33

Params Authorization Headers (8) Body Scripts Tests Settings

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

This request does not have a body

Body Cookies (1) Headers (5) Test Results 200 OK

Pretty Raw Preview Visualize JSON

```
1 [
2   {
3     "id": 33,
4     "loan": {
5       "id": 33,
6       "customer": {
7         "id": 65,
8         "name": "Taner",
9         "surname": "Geyhan",
10        "creditLimit": 90000.0,
11        "usedCreditLimit": 21000.0
12      },
13      "loanAmount": 30000.0,
14      "numberOfInstallments": 6,
15      "isPaid": false,
16      "createDate": "2025-01-10"
17    },
18    "amount": 5000.0,
19    "paidAmount": 0.0,
20    "dueDate": "2025-02-01",
21    "paymentDate": null,
22    "isPaid": false
23  },
24  {
25    "id": 34,
26    "loan": {
27      "id": 33,
28      "customer": {
29        "id": 65,
30        "name": "Taner",
31        "surname": "Geyhan",
32        "creditLimit": 90000.0,
33        "usedCreditLimit": 21000.0
34      },
35      "loanAmount": 30000.0,
```

7. Pay Loan Installments

- **URL:** <http://localhost:8080/api/loans/pay?loanId=33&paymentAmount=5000.0>
- **Method:** **POST**

Response:

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

- URL:** `http://localhost:8080/api/loans/pay?loanId=33&paymentAmount=5000.0`
- Method:** `POST`
- Body:** `{ "paidInstallments": 1, "totalPaid": 5000.0, "isLoanPaid": false, "customer": { "id": 65, "name": "Taner", "surname": "Geyhan", "creditLimit": 90000.0, "usedCreditLimit": 21000.0 } }`
- Status:** `200 OK`
- Response Time:** `17 ms`
- Response Size:** `327 B`

H2 DATABASE:

LOANS TABLE:

The screenshot shows the H2 Database console with the following details:

- Database:** `jdbc:h2:file:./data/testdb`
- Schema:** `INFORMATION_SCHEMA`
- Table:** `LOANS`
- SQL Statement:** `SELECT * FROM LOANS;`
- Results:**

ID	CREATE_DATE	IS_PAID	LOAN_AMOUNT	NUMBER_OF_INSTALLMENTS	CUSTOMER_ID
1	2025-01-10	TRUE	30000.0	6	1
33	2025-01-10	FALSE	30000.0	6	65

(2 rows, 0 ms)

LOAN_INSTALLMENTS TABLE:

localhost:8080/h2-console/login.do?jsessionid=b44a5878553415982602ff655cb63e41

Auto commit ☒ Max rows: 1000 Auto complete Off Auto select On

jdbc:h2:file:./data/testdb

- CUSTOMER
- LOANS
- LOAN_INSTALLMENTS
- INFORMATION_SCHEMA
- Users
- H2 2.2.224 (2023-09-17)

Run Run Selected Auto complete Clear SQL statement:

SELECT * FROM LOAN_INSTALLMENTS

SELECT * FROM LOAN_INSTALLMENTS;

ID	AMOUNT	DUE_DATE	IS_PAID	PAID_AMOUNT	PAYMENT_DATE	LOAN_ID
1	5000.0	2025-02-01	TRUE	5000.0	2025-01-10	1
2	5000.0	2025-03-01	TRUE	5000.0	2025-01-10	1
3	5000.0	2025-04-01	TRUE	5000.0	2025-01-10	1
4	5000.0	2025-05-01	TRUE	5000.0	2025-01-10	1
5	5000.0	2025-06-01	TRUE	5000.0	2025-01-10	1
6	5000.0	2025-07-01	TRUE	5000.0	2025-01-10	1
33	5000.0	2025-02-01	TRUE	5000.0	2025-01-10	33
34	5000.0	2025-03-01	FALSE	0.0	null	33
35	5000.0	2025-04-01	FALSE	0.0	null	33
36	5000.0	2025-05-01	FALSE	0.0	null	33
37	5000.0	2025-06-01	FALSE	0.0	null	33
38	5000.0	2025-07-01	FALSE	0.0	null	33

(12 rows, 0 ms)

Edit

CUSTOMER TABLE:

Auto commit ☒ Max rows: 1000 Auto complete Off Auto select On

jdbc:h2:file:./data/testdb

- CUSTOMER
- LOANS
- LOAN_INSTALLMENTS
- INFORMATION_SCHEMA
- Users
- H2 2.2.224 (2023-09-17)

Run Run Selected Auto complete Clear SQL statement:

SELECT * FROM CUSTOMER

SELECT * FROM CUSTOMER;

ID	CREDIT_LIMIT	NAME	SURNAME	USED_CREDIT_LIMIT	ROLE
1	75000.0	Ahmet	Mert	21000.0	null
2	90000.0	Taner	Geyhan	1000.0	null
33	90000.0	Taner	Geyhan	1000.0	null
65	90000.0	Taner	Geyhan	21000.0	null

(4 rows, 0 ms)

Edit