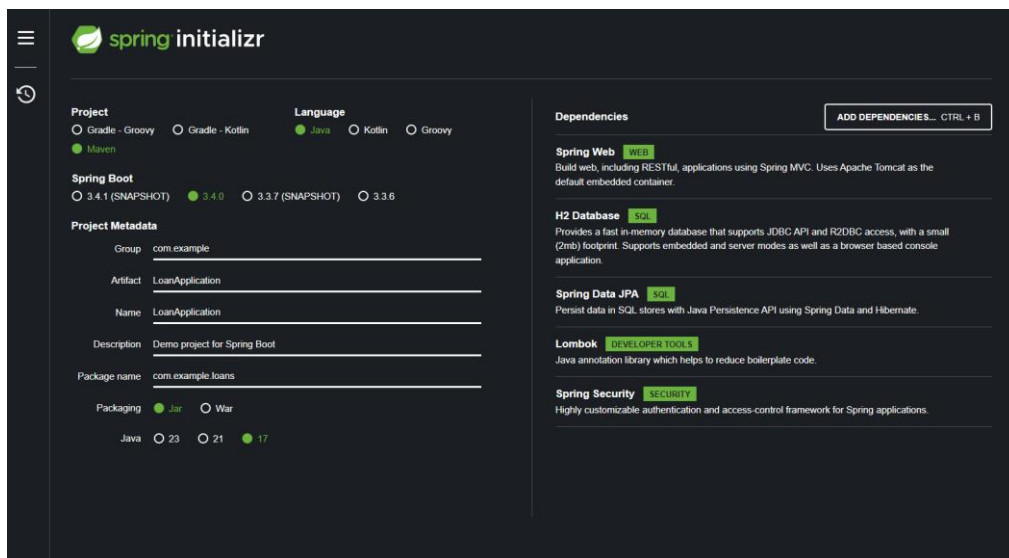


# Loan Application

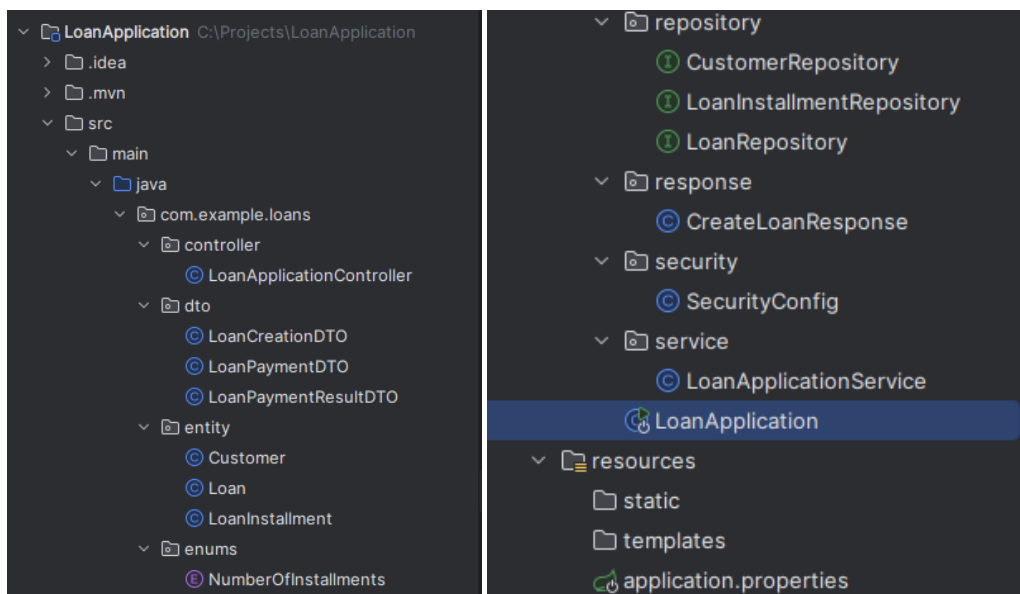
## General Information

Application is implemented for handling some loan operations.

With help of spring boot initializer we can add the dependencies, select the dependency management system and choose the Java version and then download the whole project and open it via our IDE, in my case IntelliJ IDEA.



It is basically a spring boot application had a hierarchy like below.



We have a rest controller named ***LoanApplicationController*** to handle the api calls.

It has 4 methods;

```
@PostMapping("/createLoan")
@GetMapping("/listLoansByCustomerId/{customerId}")
@GetMapping("/listInstallmentsByLoanId/{loanId}")
@PostMapping("/payLoan")
```

We have a Service named ***LoanApplicationService*** to handle the main logic for all operation details.

H2 database is used for data operations and we have 3 main entities named as ***Customer***, ***Loan*** and ***LoanInstallment***. Also we have repositories for these 3 entities named CustomerRepository, LoanRepository and LoanInstallmentRepository.

Basic authentication is used for authorizing api with a simple user and password via using **Spring Security** dependencies. For a configuration file I defined a class named ***SecurityConfig*** and defined some configurations in it.

We have a ***LoanApplicationServiceTest*** test class for handling some unit tests for our service.

Also we have ***application.properties*** file to make some pre configurations for our application.

## Running and Testing Project

First, we run the SpringBootApplication(in our case LoanApplication) and see the logs below. So at the end of the run operation, h2 database tables are created.

```
Maximum pool size: undefined/unknown
Hibernate: create global temporary table HTE_loans(is_paid boolean, loan_amount numeric(38,2), number_of_installment integer, rn_integer not null, create_date timestamp(6), customer_id bigint,
Hibernate: create global temporary table HTE_customers(credit_limit numeric(38,2), rn_integer not null, used_credit_limit numeric(38,2), id bigint, name varchar(255), sur_name varchar(255), pri
Hibernate: create global temporary table HTE_loan_installments(amount numeric(38,2), due_date date, is_paid boolean, paid_amount numeric(38,2), payment_date date, rn_integer not null, id bigint
2024-12-13T23:46:32.598+03:00 INFO 31408 --- [LoanApplication] [ restartedMain] o.s.e.t.j.p.i.JtaPlatformInitiator : HHH000489: No JTA platform available (set 'hibernate.transaction.jta.
Hibernate: create table customers (id bigint not null, credit_limit numeric(38,2), name varchar(255), sur_name varchar(255), used_credit_limit numeric(38,2), primary key (id))
Hibernate: create table loan_installments (id bigint not null, amount numeric(38,2), due_date date, is_paid boolean, loan_id bigint, paid_amount numeric(38,2), payment_date date, primary key (id
Hibernate: create table loans (id bigint not null, create_date timestamp(6), customer_id bigint, is_paid boolean, loan_amount numeric(38,2), number_of_installment integer, primary key (id))
Hibernate: create sequence customers_seq start with 1 increment by 50
Hibernate: create sequence loan_installments_seq start with 1 increment by 50
Hibernate: create sequence loans_seq start with 1 increment by 50
2024-12-13T23:46:32.636+03:00 INFO 31408 --- [LoanApplication] [ restartedMain] j.LocalContainerEntityManagerFactoryBean : Initialized JPA EntityManagerFactory for persistence unit 'default'
2024-12-13T23:46:32.878+03:00 INFO 31408 --- [LoanApplication] [ restartedMain] o.s.d.j.r.query.QueryEnhancerFactory : Hibernate is in classpath; If applicable, HQL parser will be used.
2024-12-13T23:46:33.261+03:00 WARN 31408 --- [LoanApplication] [ restartedMain] JpaBaseConfiguration$JpaWebConfiguration : spring.jpa.open-in-view is enabled by default. Therefore, database queries may be executed during view rendering. This is not recommended.
2024-12-13T23:46:33.308+03:00 INFO 31408 --- [LoanApplication] [ restartedMain] r$InitializeUserDetailsServiceConfigurer : Global AuthenticationManager configured with UserDetailsServiceImpl bean
2024-12-13T23:46:33.588+03:00 WARN 31408 --- [LoanApplication] [ restartedMain] o.s.c.a.web.builders.WebSecurity : You are asking Spring Security to ignore Ant [pattern='/h2-console/**
2024-12-13T23:46:33.821+03:00 INFO 31408 --- [LoanApplication] [ restartedMain] o.s.b.d.a.OptionalLiveReloadServer : LiveReload server is running on port 35729
2024-12-13T23:46:33.864+03:00 INFO 31408 --- [LoanApplication] [ restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port 8080 (http) with context path '/'
2024-12-13T23:46:33.874+03:00 INFO 31408 --- [LoanApplication] [ restartedMain] com.example.loans.LoanApplication : Started LoanApplication in 4.524 seconds (process running for 5.144)
```

Then we can login and see the tables in h2-console

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:mem:testdb

User Name: sa

Password:

Connect Test Connection

We can see all tables defined in h2 console and we will define a customer manually for applying our test cases for loan operations.

Run Run Selected Auto complete Clear SQL statement:

SELECT \* FROM CUSTOMERS

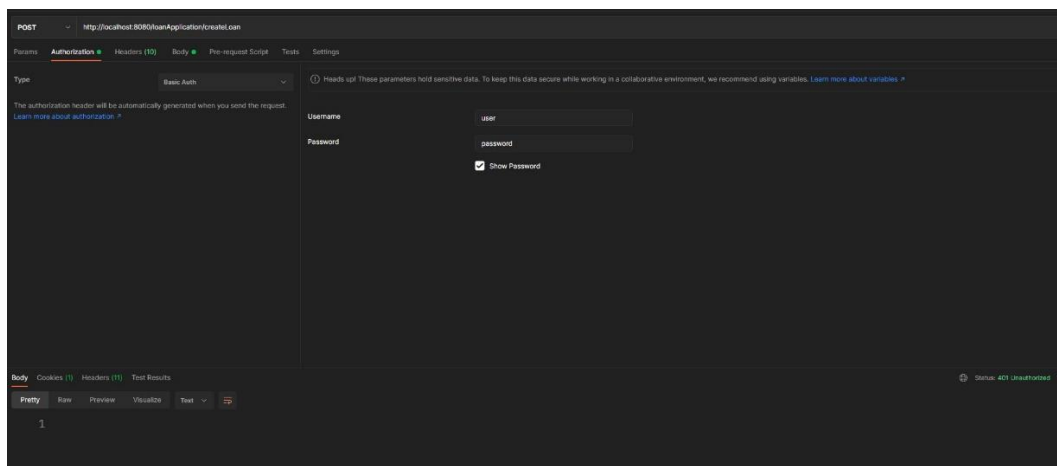
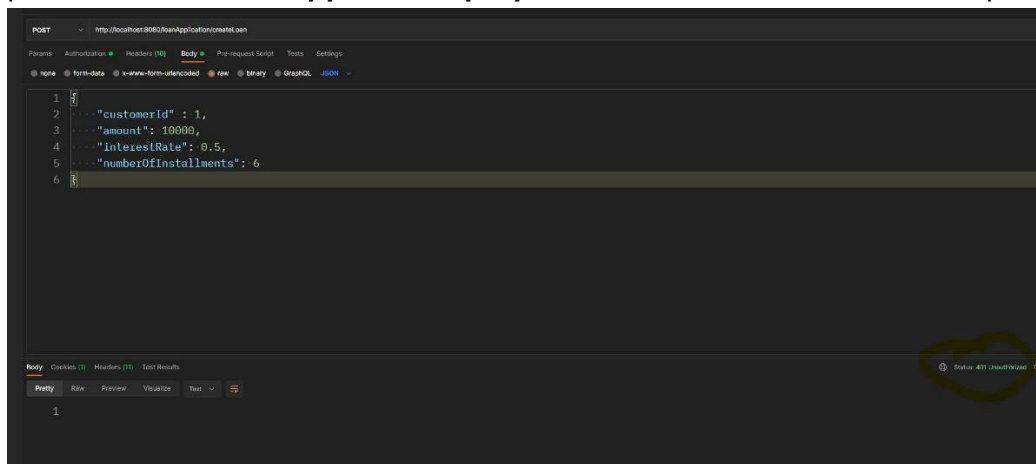
@edit SELECT \* FROM CUSTOMERS;

Action	ID	CREDIT_LIMIT	NAME	SUR_NAME	USED_CREDIT_LIMIT
✓ ✕	1	100000	Orhan	Senturk	0

(no rows, 0 ms)

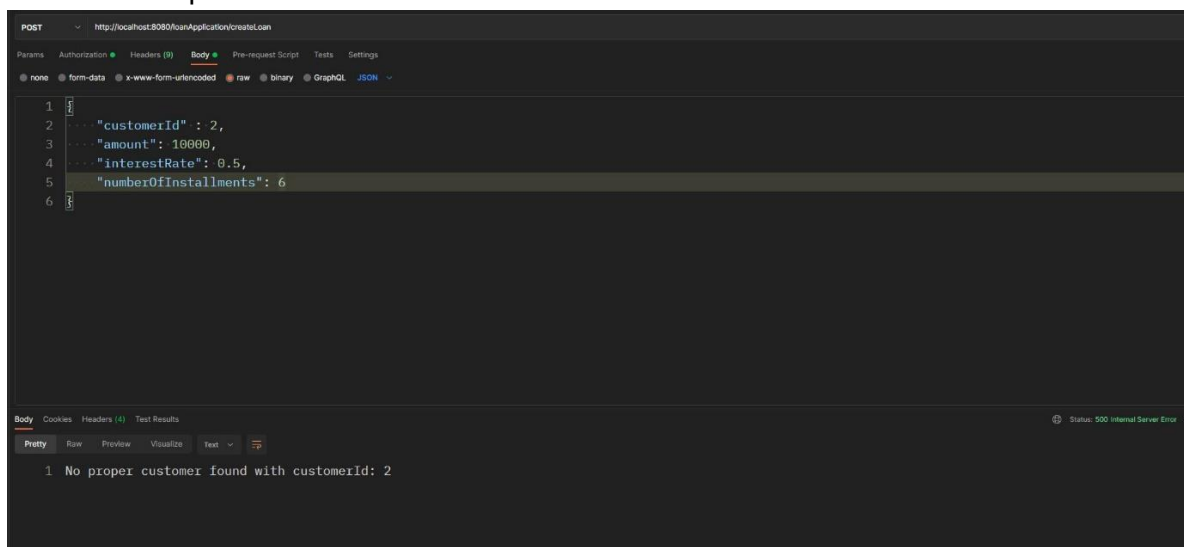
Now we can try to send requests with using **Postman Tool**.

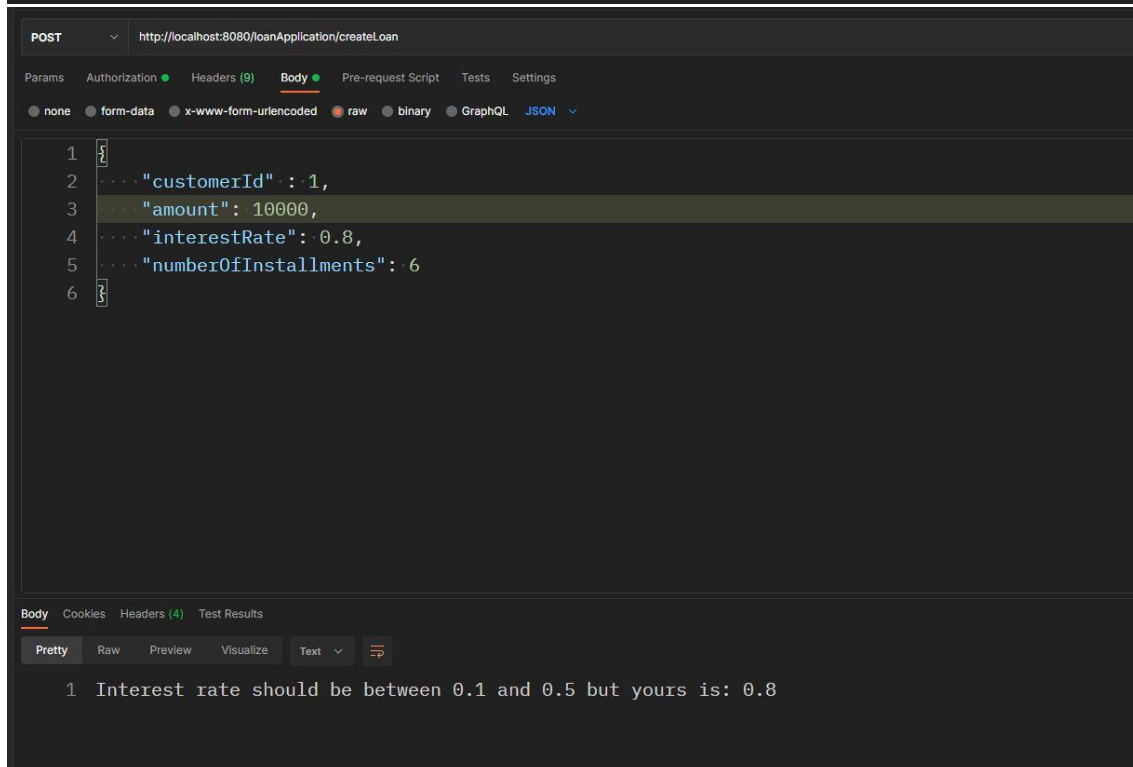
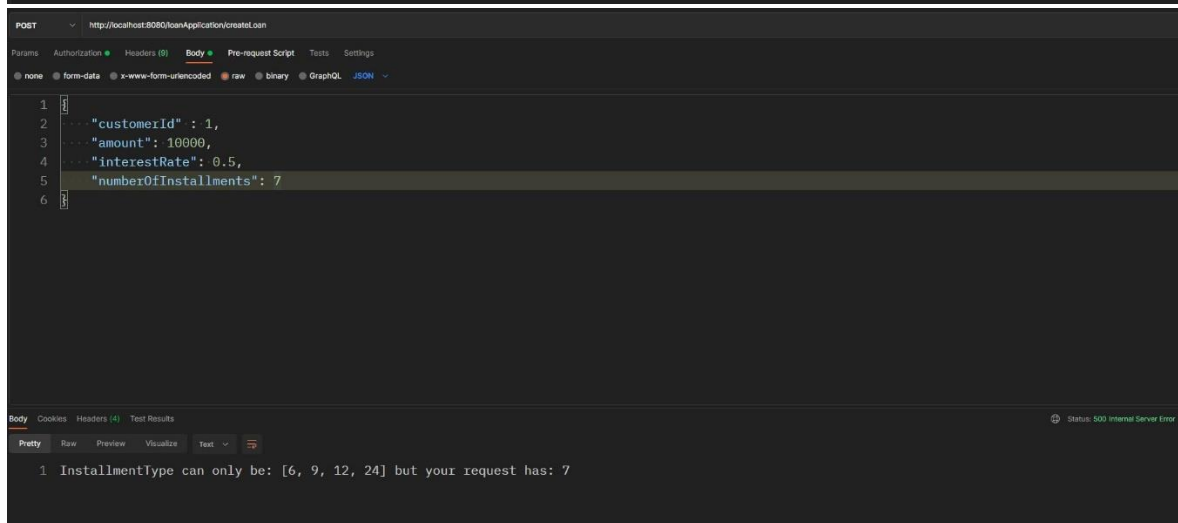
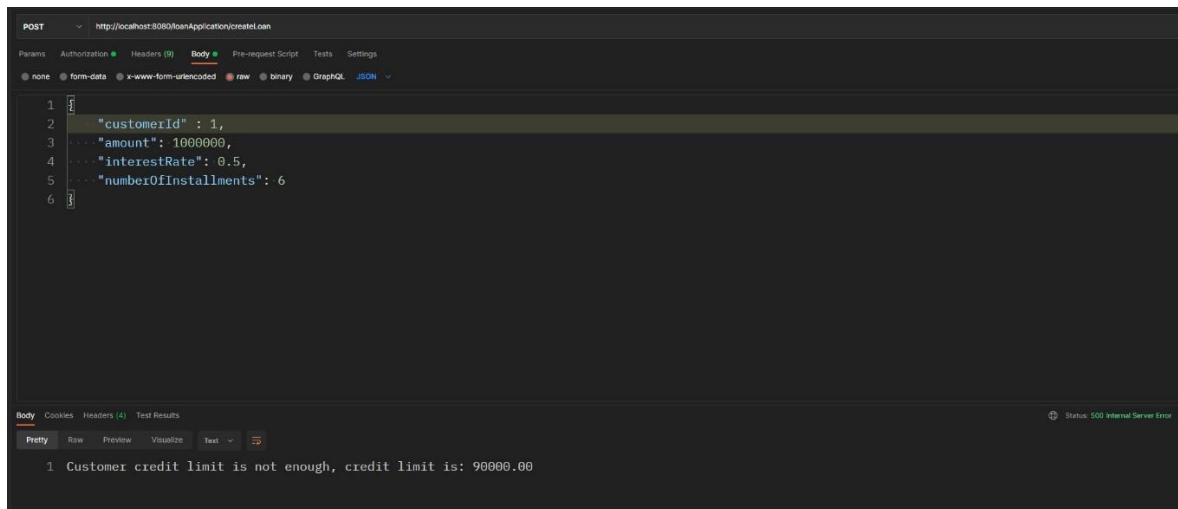
We have a basic authentication to authorize our apis and without entering this user and password defined in ***application.properties*** file we can not reach our apis.



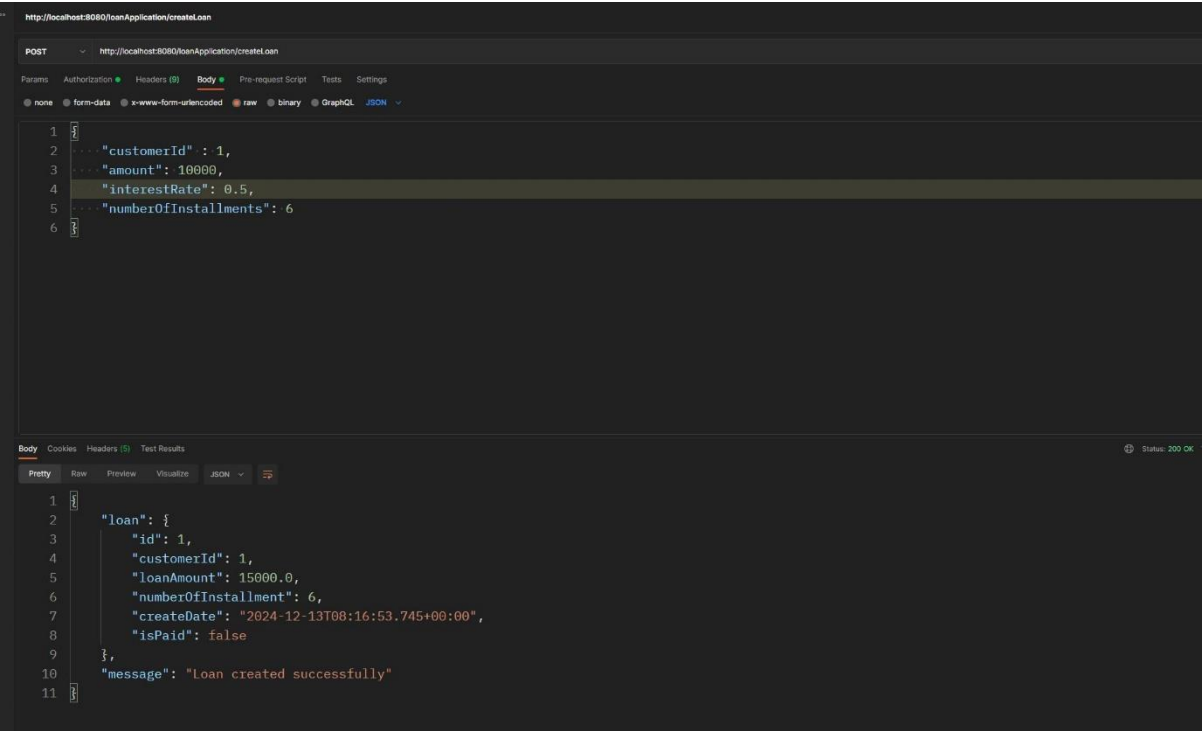
## Api Calls and Check The Results

createLoan operation error cases





createLoan operation:



after the loan created, database tables look like;

Run
Run Selected
Auto complete
Clear
SQL statement:

SELECT \* FROM CUSTOMERS;

SELECT \* FROM LOANS;

SELECT \* FROM LOAN\_INSTALLMENTS;

SELECT \* FROM CUSTOMERS;

ID	CREDIT_LIMIT	NAME	SUR_NAME	USED_CREDIT_LIMIT
1	90000.00	Orhan	Senturk	10000.00

(1 row, 0 ms)

SELECT \* FROM LOANS;

ID	CREATE_DATE	CUSTOMER_ID	IS_PAID	LOAN_AMOUNT	NUMBER_OF_INSTALLMENT
1	2024-12-13 10:58:49.588	1	FALSE	15000.00	6

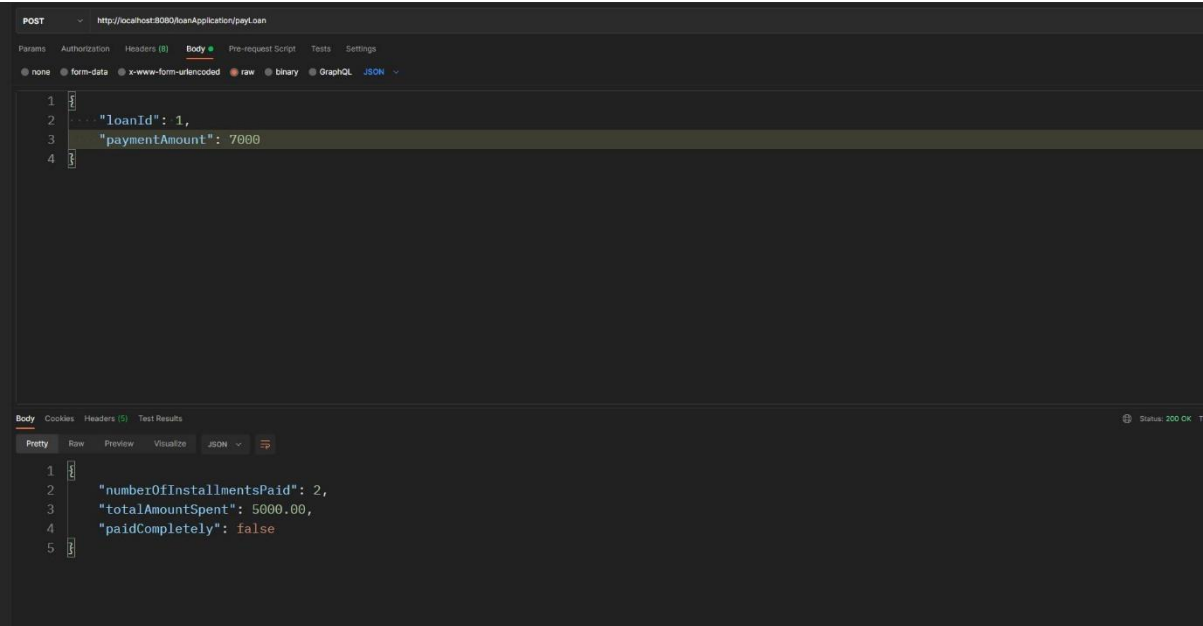
(1 row, 1 ms)

SELECT \* FROM LOAN\_INSTALLMENTS;

ID	AMOUNT	DUE_DATE	IS_PAID	LOAN_ID	PAID_AMOUNT	PAYMENT_DATE
1	2500.00	2025-01-01	FALSE	1	0.00	null
2	2500.00	2025-02-01	FALSE	1	0.00	null
3	2500.00	2025-03-01	FALSE	1	0.00	null
4	2500.00	2025-04-01	FALSE	1	0.00	null
5	2500.00	2025-05-01	FALSE	1	0.00	null
6	2500.00	2025-06-01	FALSE	1	0.00	null

(6 rows, 0 ms)

payLoan operation:



after the payLoan operation, database look like;

SELECT \* FROM CUSTOMERS;

SELECT \* FROM LOANS;

SELECT \* FROM LOAN\_INSTALLMENTS;

SELECT \* FROM CUSTOMERS;

ID	CREDIT_LIMIT	NAME	SUR_NAME	USED_CREDIT_LIMIT
1	95000.00	Orhan	Senturk	10000.00

(1 row, 0 ms)

SELECT \* FROM LOANS;

ID	CREATE_DATE	CUSTOMER_ID	IS_PAID	LOAN_AMOUNT	NUMBER_OF_INSTALLMENT
1	2024-12-13 10:58:49.588	1	FALSE	15000.00	6

(1 row, 0 ms)

SELECT \* FROM LOAN\_INSTALLMENTS;

ID	AMOUNT	DUE_DATE	IS_PAID	LOAN_ID	PAID_AMOUNT	PAYMENT_DATE
1	2500.00	2025-01-01	TRUE	1	2500.00	2024-12-13
2	2500.00	2025-02-01	TRUE	1	2500.00	2024-12-13
3	2500.00	2025-03-01	FALSE	1	0.00	null
4	2500.00	2025-04-01	FALSE	1	0.00	null
5	2500.00	2025-05-01	FALSE	1	0.00	null
6	2500.00	2025-06-01	FALSE	1	0.00	null

(6 rows, 1 ms)

listLoansByCustomerId api:

GET

http://localhost:8080/loanApplication/listLoansByCustomerId/2

Params

Authorization

Headers (6)

Body

Pre-request Script

Tests

Settings

Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body

Cookies

Headers (5)

Test Results

Pretty

Raw

Preview

Visualize

Text

Status: 204 No Content

GET

http://localhost:8080/loanApplication/listLoansByCustomerId/1

Params

Authorization

Headers (6)

Body

Pre-request Script

Tests

Settings

Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body

Cookies

Headers (5)

Test Results

Pretty

Raw

Preview

Visualize

JSON

Status: 200 OK

1

{

2

"id": 1,

3

"customerId": 1,

4

"loanAmount": 15000.00,

5

"numberOfInstallment": 6,

6

"createDate": "2024-12-13T07:58:49.588+00:00",

7

"isPaid": false

8

}

9

10



listLoanInstallmentsByLoanId api:

GET

localhost:8080/loanApplication/listInstallmentsByLoanId/2

Params

Authorization

Headers (6)

Body

Pre-request Script

Tests

Settings

Query Params

KEY	VALUE	DESCRIPTION
key	Value	Description

Body

Cookies

Headers (3)

Test Results

1

Raw

Preview

Visualize

Text

Status: 204 No Content

GET

localhost:8080/loanApplication/listInstallmentsByLoanId/1

Params

Authorization

Headers (6)

Body

Pre-request Script

Tests

Settings

Query Params

KEY	VALUE	DESCRIPTION
key	Value	Description

Body

Cookies

Headers (3)

Test Results

1

Raw

Preview

Visualize

JSON

Status: 200 OK

```
1 {
2   {
3     "id": 1,
4     "loanId": 1,
5     "amount": 2500.00,
6     "paidAmount": 0.00,
7     "dueDate": "2025-01-01",
8     "paymentDate": null,
9     "isPaid": false
10  },
11  {
12    "id": 2,
13    "loanId": 1,
14    "amount": 2500.00,
15    "paidAmount": 0.00,
16    "dueDate": "2025-02-01",
17    "paymentDate": null,
18    "isPaid": false
19  },
20  {
21    "id": 3,
22    "loanId": 1,
23    "amount": 2500.00,
24    "paidAmount": 0.00,
25    "dueDate": "2025-03-01",
26    "paymentDate": null,
27    "isPaid": false
28  },
29  {
30    "id": 4,
31    "loanId": 1,
32    "amount": 2500.00,
33    "paidAmount": 0.00,
34    "dueDate": "2025-04-01",
35    "paymentDate": null,
36    "isPaid": false
37  },
38  {
39    "id": 5,
40    "loanId": 1,
41    "amount": 2500.00,
42    "paidAmount": 0.00,
43    "dueDate": "2025-05-01",
44    "paymentDate": null,
45    "isPaid": false
46  },
47  {
48    "id": 6,
49    "loanId": 1,
50    "amount": 2500.00,
51    "paidAmount": 0.00,
52    "dueDate": "2025-06-01",
53    "paymentDate": null,
54    "isPaid": false
55  },
56  }
```

# Unit Tests

We simply have these test cases and run them successfully.

