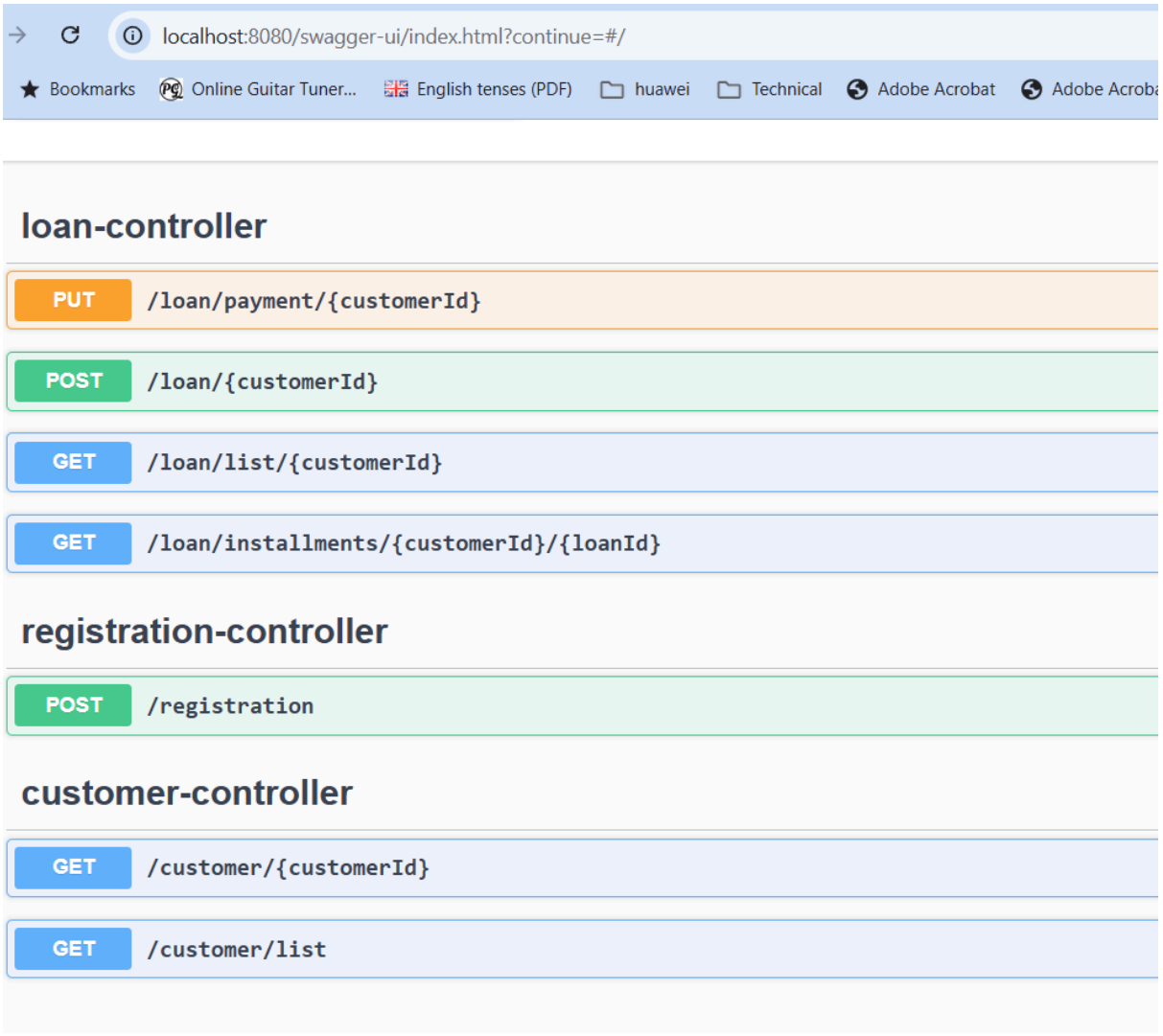
Customer Loan Service Application - User Manuel

This backend service application has been developed by using Spring Boot Framework, and H2 has been used as an in memory database.

Provided Endpoints

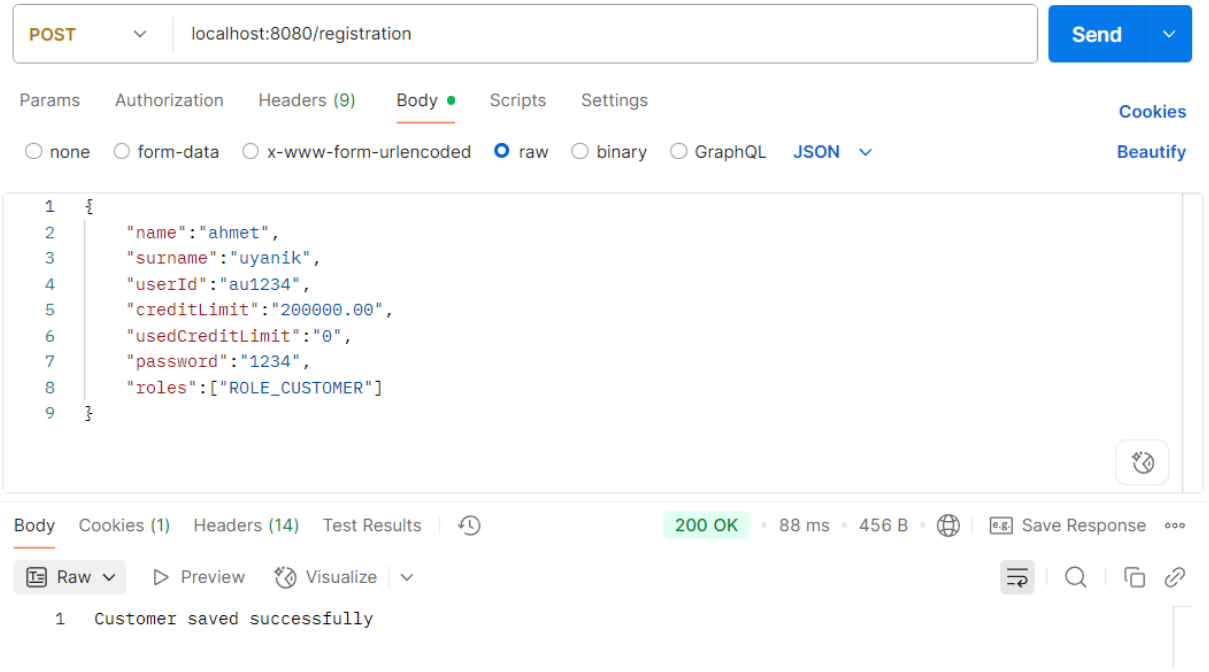
The below apis are provided in order to manage the customer loan process. As it was required, all endpoints (except /registration) require admin user or record owner [customer] authorization by specifying username and password with basic authentication.
There are two user roles; admin [ROLE_ADMIN], customer [ROLE_CUSTOMER]
The system admin user automatically initialized and created when the application started.

The admin user credentials;
username : admin
password : admin1



POST :/registration (Customer registration)

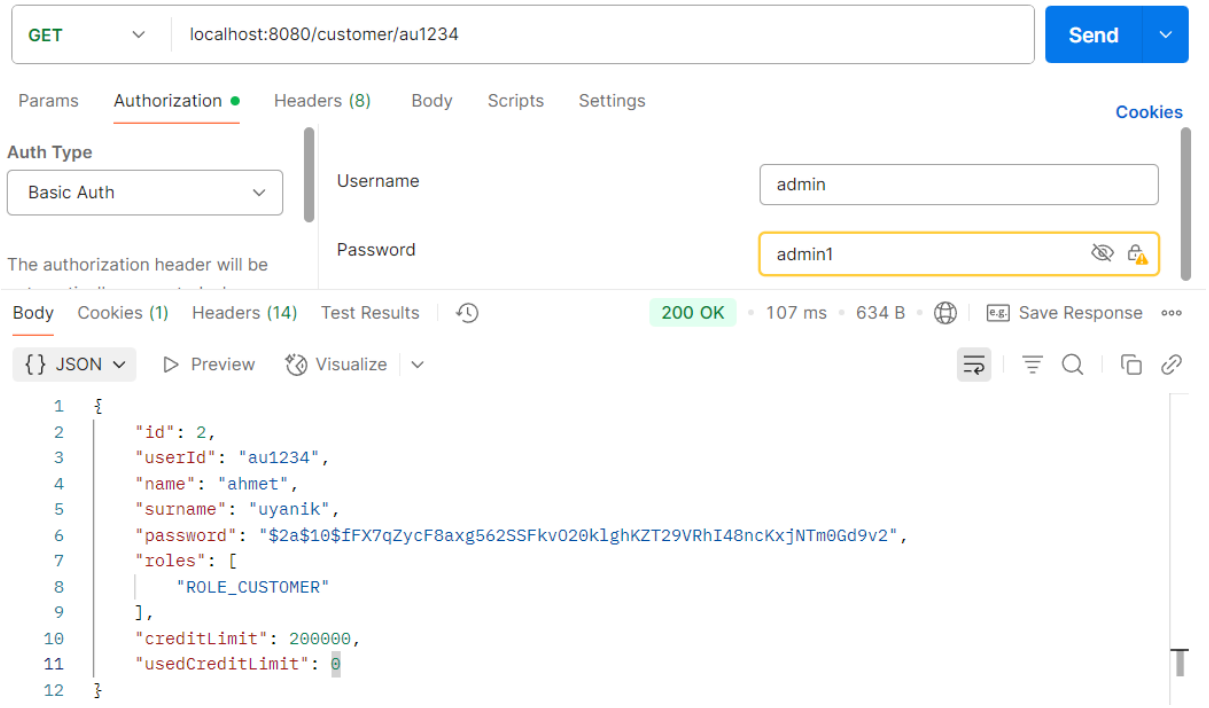
It is a public endpoint to create customer records on the system. On the request body, the customer information needs to be provided like; name, surname, userId, creditLimit…



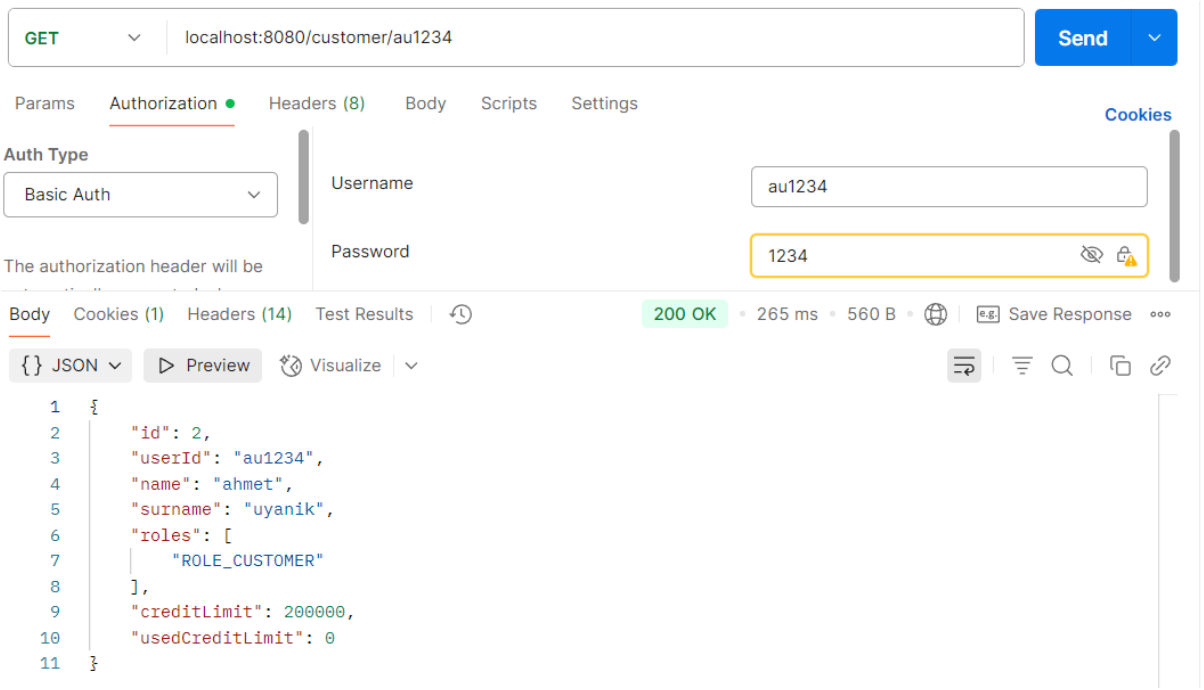
It should return a success message after the customer registered successfully.

GET :/customer/{userId} (customer details)

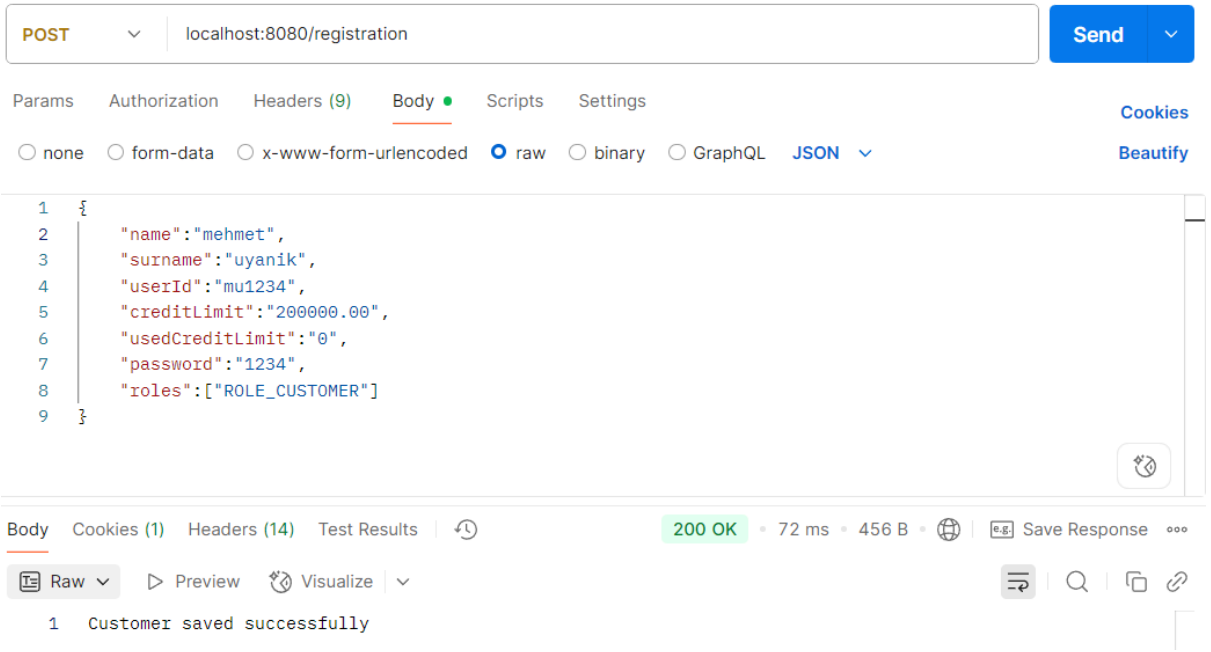
It is a secured endpoint to get customer details for a given userId. It requires basic auth (admin, customer [record owner])



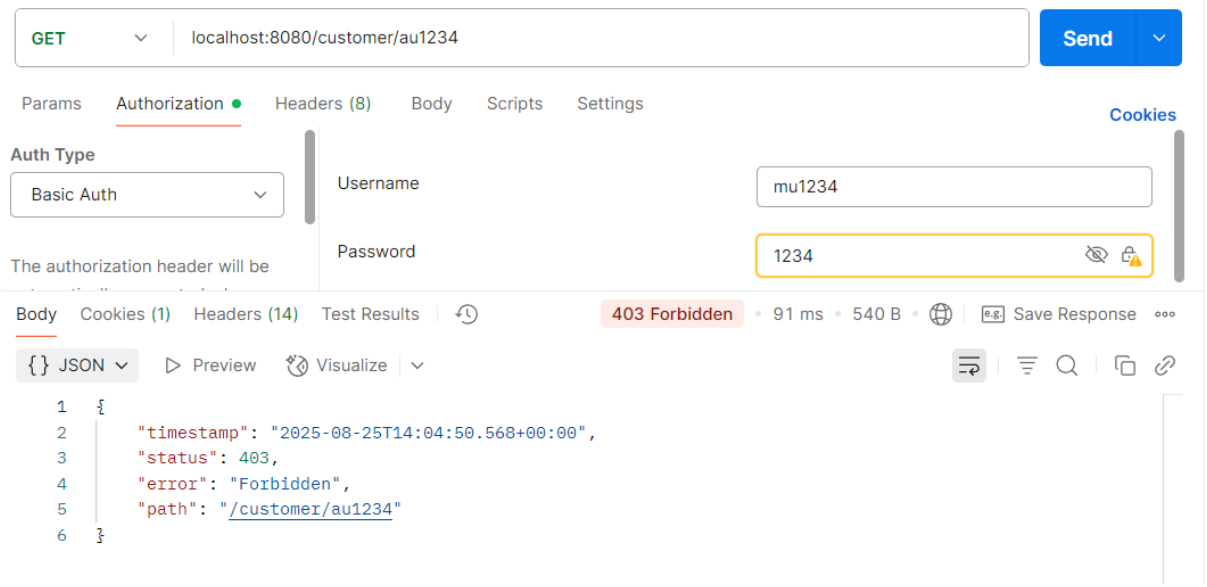
The api also accepts the record owner [customer] request as below



Lets create another customer, and check if the second customer can access the information of the first one.



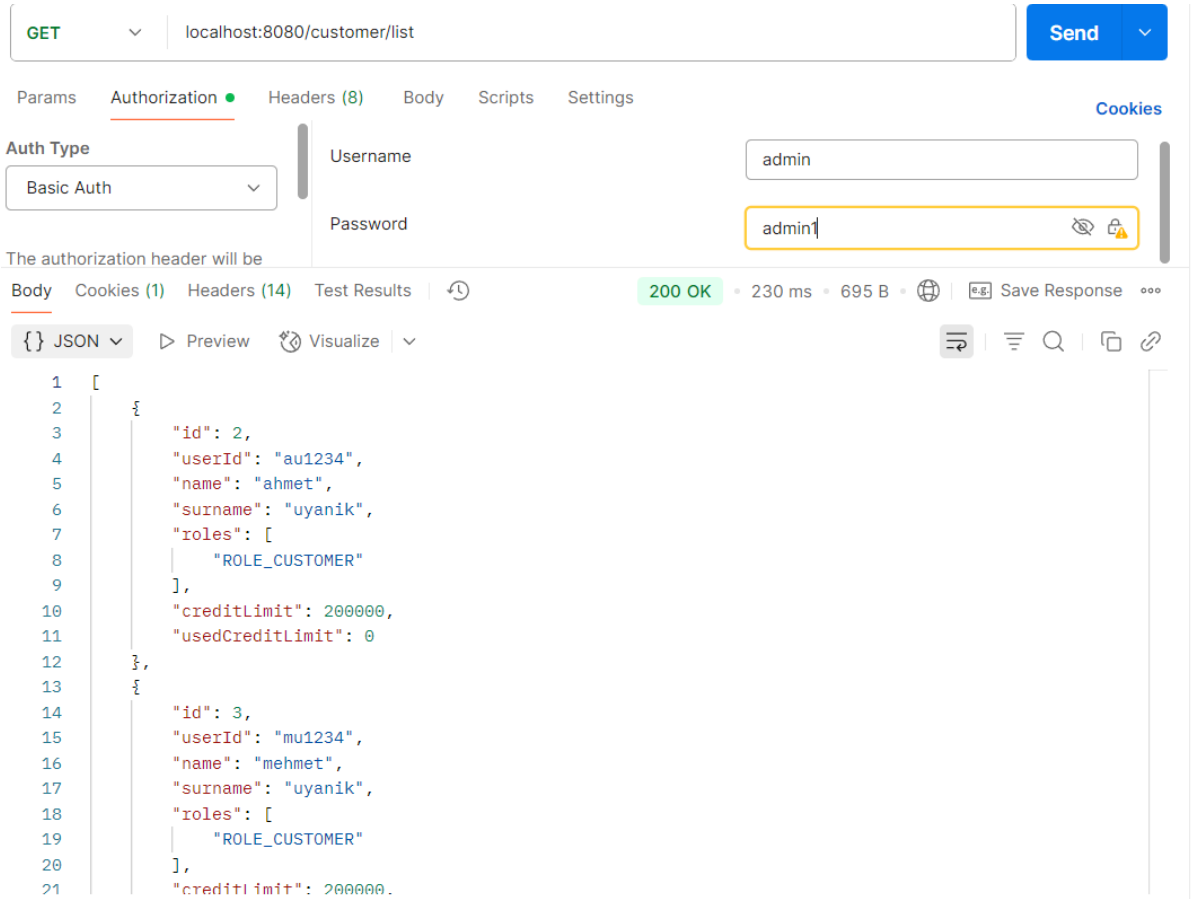
the second customer mu1234 was created, and lets try to access customer au1234 information with the customer mu1234



As we can see here, 403 Forbidden http message was returned by the application.

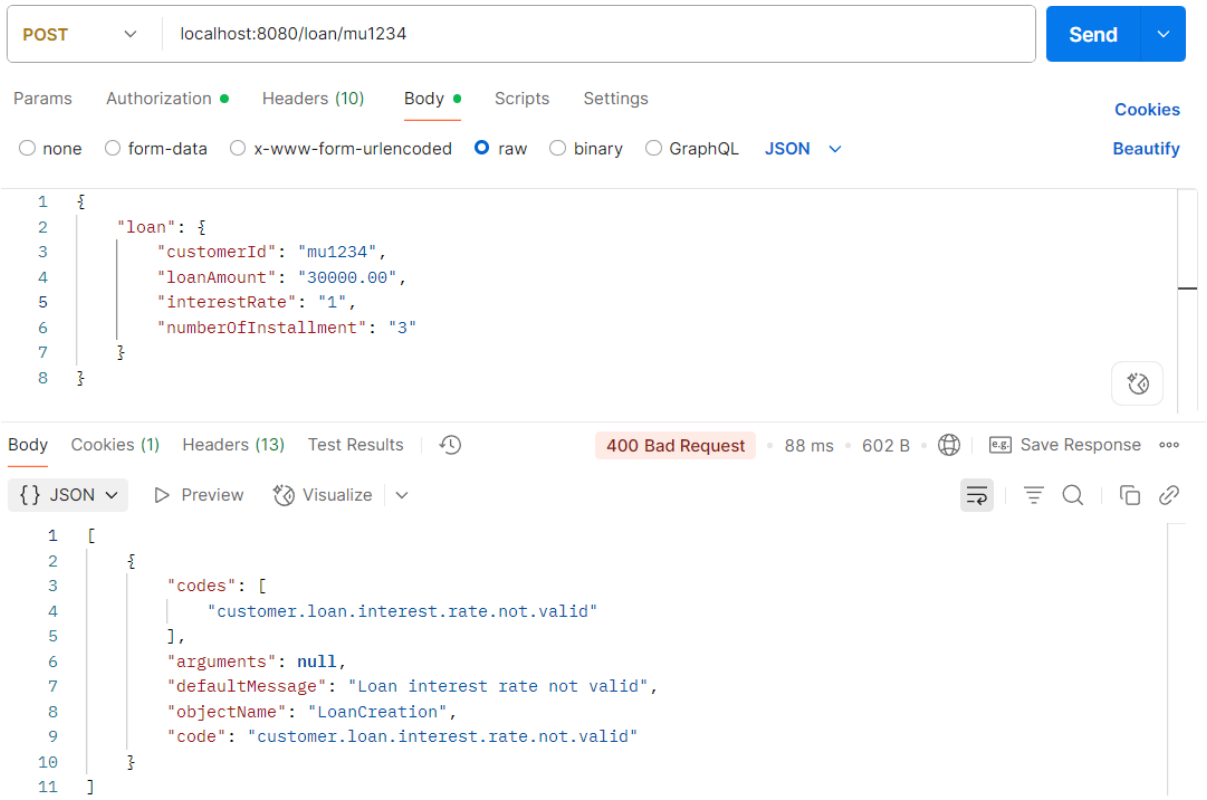
GET :/customer/list (all customer details)

It is a secured endpoint to list all registered customers in the system. Only admin user can invoke this



POST :/loan/{userId} (create loan)

It is a secured endpoint to create loan and loan installments with the request body as shown. There is a validator logic inside the app to check some conditions to determine whether it is eligible to create a loan for that customer. This endpoint can be triggered by the admin, or the owner customer.
First, let us check the validation. For example, the interest rate amount should be between 0.1 - 0.5 as it has been mentioned in the requirement document. Let's send interest rate as 1

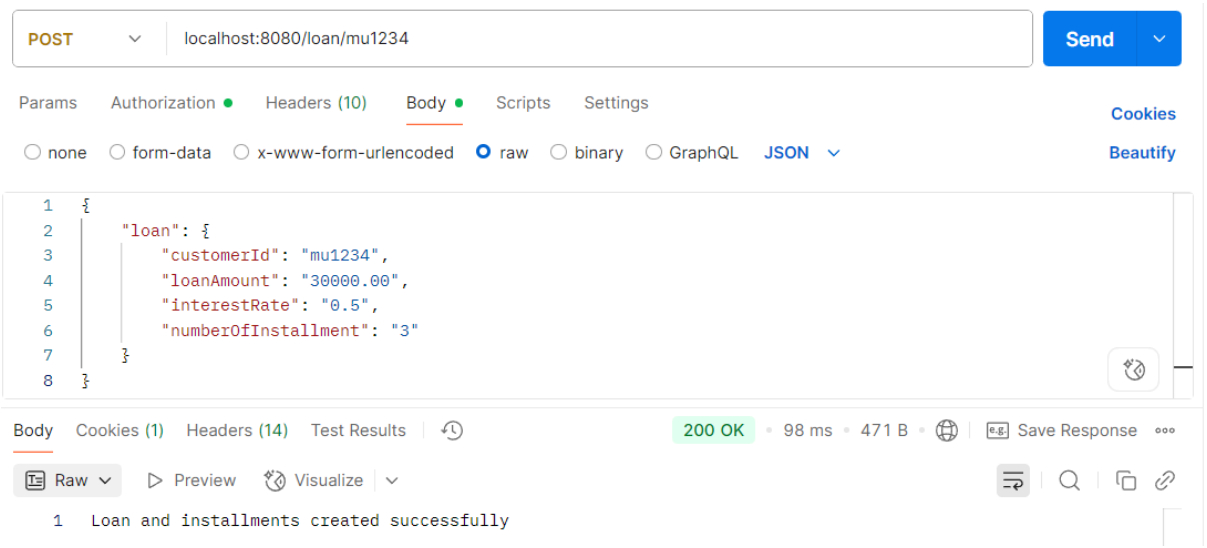


We expect that It gets failed by the validation since the interest rate is not fitting the required amount.

Let’s try a successful scenario..

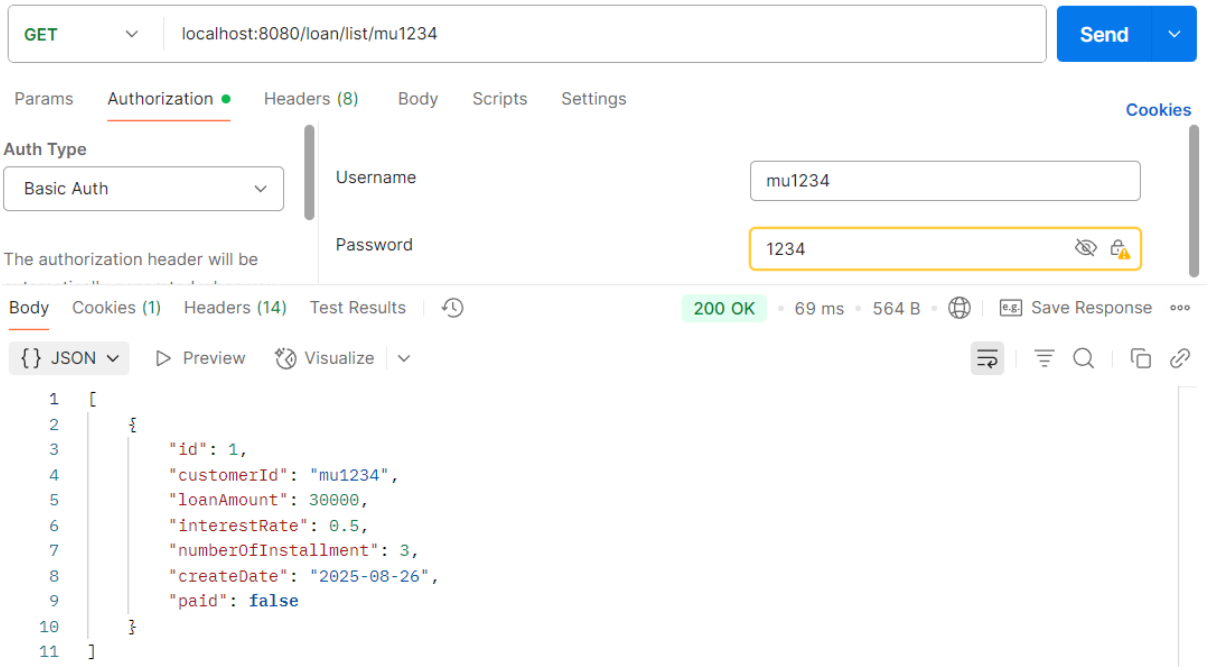
We set interest rate as 0.5, for the loan amount 30000.00, that makes 450000.00 for 3 month installment payment, and each installment amount 15000.00

As we can see below, since all conditions are met, it allows us to create a loan and installments accordingly. Some installment information like dueDate, isPaid is calculated and set programmatically. Therefore we do not include them in the request body.



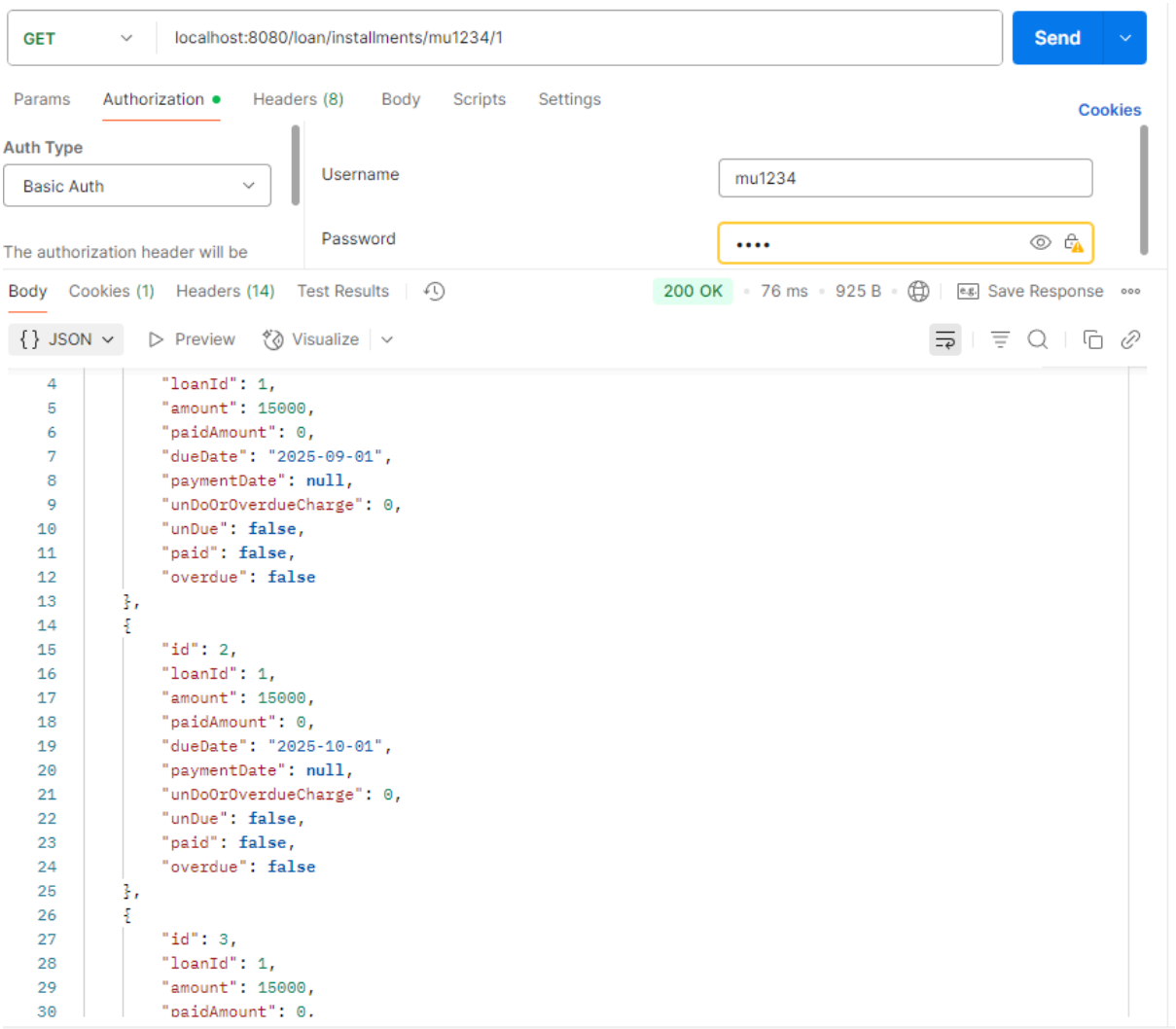
GET :/loan/list/{userId} (list loans for userId)

It is a secured endpoint to list loans for the given userId.



GET :/loan/installments/{userId}/{loanId} (list loan installments for userId)

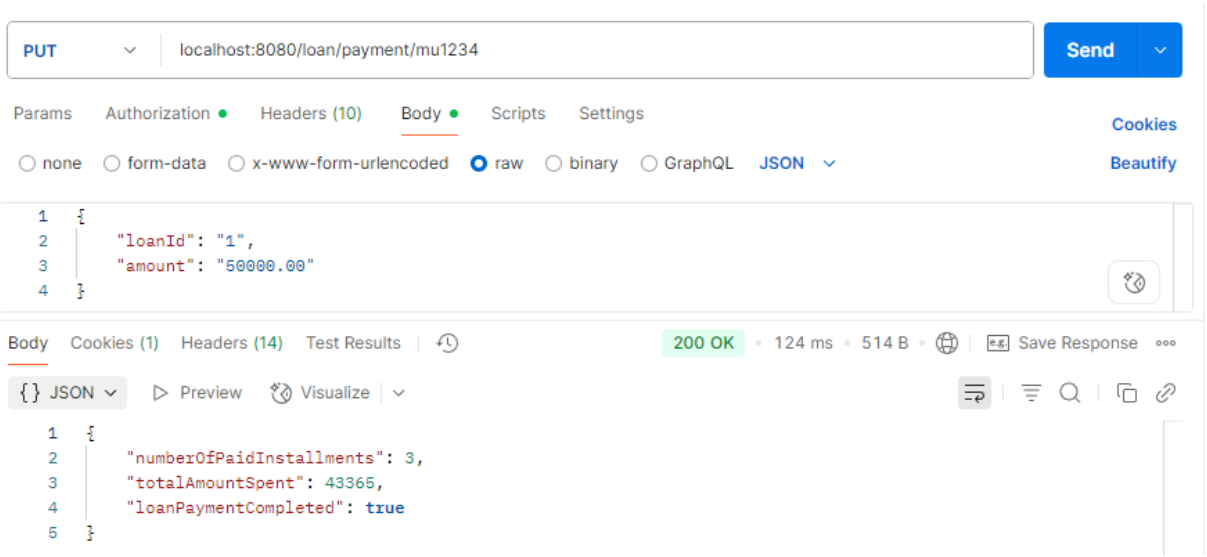
It is a secured endpoint to list the installments for a given userId and loanId



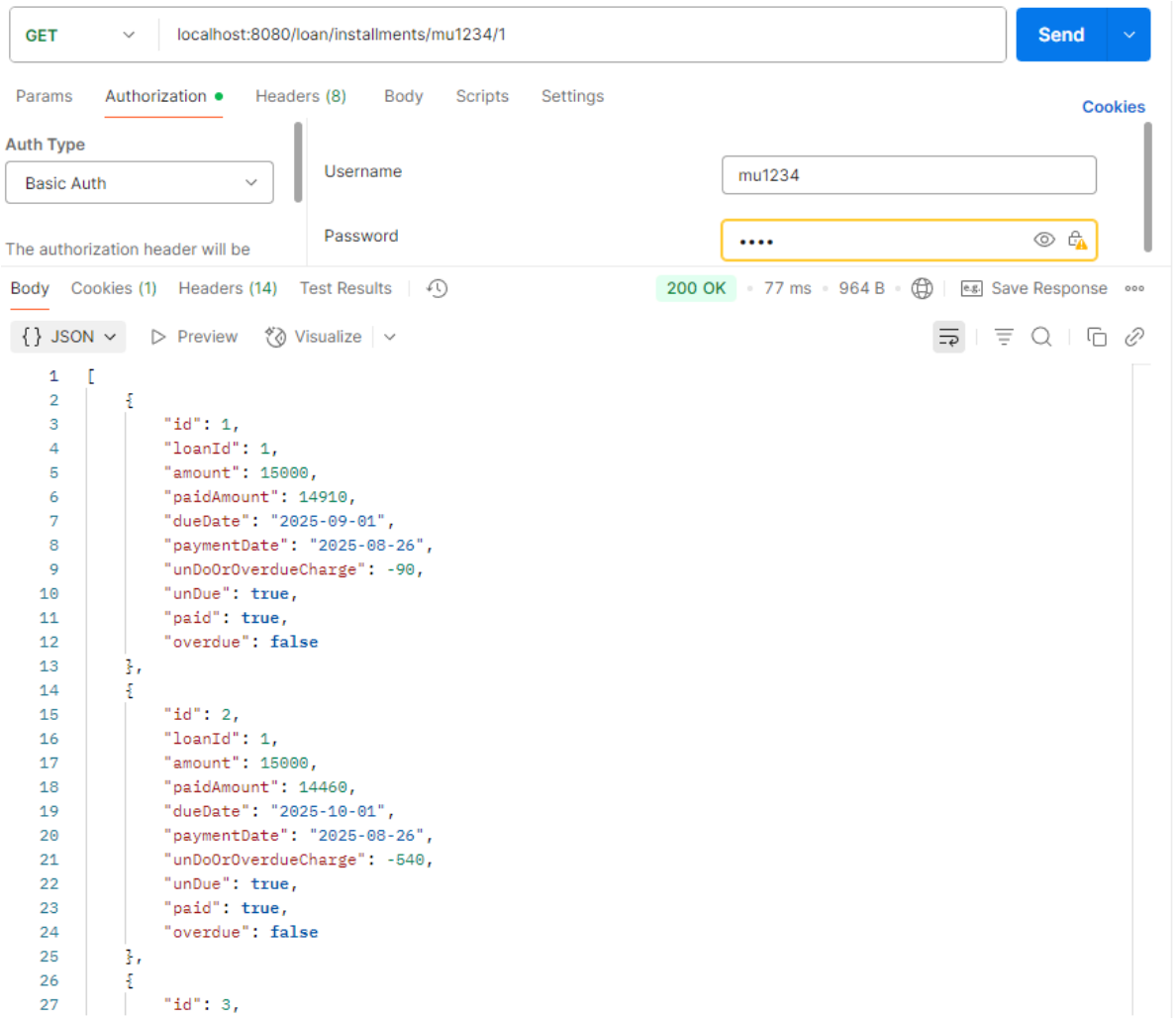
unDo and overdue fields are set to false initially, they are used to mark whether the installment is paid before due date, or after due date. I will add the reward or penalty amount into the unDoOrOverdueCharge field accordingly.

PUT :/loan/payment/{userId} (pay the loan for userId)

It is a secured endpoint to pay the loan based on the amount given in the request. There is also a validation behind this endpoint to check some conditions given in the requirement document.
On the request body, we specify the amount, and the loan id to pay.



On the response we can see; number of installments paid, total amount spent to pay, and if the loan was paid completely.
After that you can also see the installments isPaid status was changed to true, and payment Date also was set programatically after payment.



H2-Console (DB)

The h2 database credentials defined as,
username:sa,
password:sa.
It can be checked the data using h2-console as shown below.