

Java Backend Developer Case (Credit Module Challenge)

You are expected to write a backend Loan API for a bank so that their employees can create, list and pay loans for their customers.

Your backend service should have those endpoints:

- Create Loan: Create a new loan for a given customer, amount, interest rate and number of installments
 - You should check if customer has enough limit to get this new loan
 - Number of installments can only be 6, 9, 12, 24
 - Interest rate can be between 0.1 – 0.5
 - All installments should have same amount. Total amount for loan should be $\text{amount} * (1 + \text{interest rate})$
 - Due Date of Installments should be first day of months. So the first installment's due date should be the first day of next month.
- List Loans: List loans for a given customer
 - If you want you can add more filters like number of installments, is paid etc.
- List Installment: List installments for a given loan
- Pay Loan: Pay installment for a given loan and amount
 - Endpoint can pay multiple installments with respect to amount sent with some restriction described below.
 - Installments should be paid wholly or not at all. So if installments amount is 10 and you send 20, 2 installments can be paid. If you send 15, only 1 can be paid. If you send 5, no installments can be paid.
 - Earliest installment should be paid first and if there are more money then you should continue to next installment.
 - Installments have due date that still more than 3 calendar months cannot be paid. So if we were in January, you could pay only for January, February and March installments.
 - A result should be returned to inform how many installments paid, total amount spent and if loan is paid completely.
 - Necessary updates should be done in customer, loan and installment tables.

Requirements:

- Documentation is important. Prepare a document, readme, about how to use the application.
- All endpoints should be authorized with an admin user and password
- All info should be stored in database as below:

Customer: id, name, surname, creditLimit, usedCreditLimit

Loan: id, customerId, loanAmount, numberOfInstallment, createDate, isPaid

LoanInstallment: id, loanId, amount, paidAmount, dueDate, paymentDate, isPaid

Implementation:

We would like you to build a java application using Spring Boot Framework. You can use h2 db as database. You are expected to write some unit tests. Try to build and design your code as it will be deployed to prod (or at least test env 😊) Make sure to add information how to build and run the project.

Bonus 1: Instead of securing endpoints with an admin username and password. Create an authorization so that while ADMIN users can operate for all customers, CUSTOMER role users can operate for themselves.

Bonus 2: For reward and penalty add this logic to paying loan flow:

- If an installment is paid before due date, make a discount equal to $\text{installmentAmount} * 0.001 * (\text{number of days before due date})$ So in this case paidAmount of installment will be lower than amount.
- If an installment is paid after due date, add a penalty equal to $\text{installmentAmount} * 0.001 * (\text{number of days after due date})$ So in this case paidAmount of installment will be higher than amount.