

Java

Introduction

You are expected to complete the following task within 72 hours of receiving the assignment.

Plan Generator

In order to inform borrowers about the final repayment schedule, we need to have pre-calculated repayment plans throughout the lifetime of a loan.

To be able to calculate a repayment plan specific input parameters are necessary:



- duration (number of installments in months)
- nominal rate (annual interest rate)
- loan amount (principal amount)
- Date of Disbursement/Payout ("startDate")

These four parameters need to be input parameters.

The goal is to calculate a repayment plan for an annuity loan. Therefore the amount that the borrower has to pay back every month, consisting of principal and interest repayments, does not change (the last installment might be an exception).

The annuity amount has to be derived from three of the input parameters (duration, nominal interest rate, total loan amount) before starting the plan calculation.

(use http://financeformulas.net/Annuity_Payment_Formula.html as reference)

 **Note:** the nominal interest rate is an **annual** rate and must be converted to monthly before using in the annuity formula 

Example Loan Details after annuity calculation:

Loan Amount	5000 €
Interest Rate	5.00%
Duration	24 month
Start Date	01.01.2018
Annuity	219.36 €

Based on this information we are able to create a repayment plan for the borrower:

Date	Annuity (Borrower Payment Amount)	Principal	Interest	Initial Outstanding Principal	Remaining Outstanding Principal
01.01.2018	219.36 €	198.53 €	20.83 €	5000 €	4801.47 €
01.02.2018	219.36 €	199.35	20.01 €	4801.47 €	4602.12 €
...
01.12.2019	219.28 €	218.37 €	0.91 €	218.37 €	0 €

Calculation Basics:

- 1) For simplicity, we will have the following day convention: each month has **30 days**, a year has **360 days**.
- 2) Interest calculation

```
Interest = (Rate * Days in Month * Initial Outstanding Principal) / Days in Year
e.g. first installment = (0.05 * 30 * 5000.00) / 360 = 20.83 € (with rounding)
```

- 3) Principal Calculation

```
Principal = Annuity - Interest
e.g. first principal = 219.36 - 20.83 = 198.53 €
```

(if, calculated **principal** amount exceeds the initial outstanding principal amount, take initial outstanding principal amount instead >> can happen in the very last installment)

4) Borrower Payment Amount (Annuity)

Borrower Payment Amount(Annuity) = Principal + Interest e.g. first borrower payment = 198.53 + 20.83 = 219.36 €

Output:

Implement a web service that has one endpoint to generate a borrower plan via HTTP in JSON. Feel free to use any web service you are most comfortable with.

e.g. <http://localhost:8080/generate-plan> (POST)

Request:

Payload

```
{
  "loanAmount": "5000",
  "nominalRate": "5.0",
  "duration": 24,
  "startDate": "2018-01-01T00:00:01Z"
}
```

Response:

Payload

```
{
  "borrowerPayments":[
    {
      "borrowerPaymentAmount":"219.36",
      "date":"2018-01-01T00:00:00Z",
      "initialOutstandingPrincipal":"5000.00",
      "interest":"20.83",
      "principal":"198.53",
      "remainingOutstandingPrincipal":"4801.47"
    },
    {
      "borrowerPaymentAmount":"219.36",
      "date":"2018-02-01T00:00:00Z",
      "initialOutstandingPrincipal":"4801.47",
      "interest":"20.01",
      "principal":"199.35",
      "remainingOutstandingPrincipal":"4602.12"
    },
    ...
    {
      "borrowerPaymentAmount":"219.28",
      "date":"2019-12-01T00:00:00Z",
      "initialOutstandingPrincipal":"218.37",
      "interest":"0.91",
      "principal":"218.37",
      "remainingOutstandingPrincipal":"0"
    }
  ]
}
```

Closing Remarks:

Please design and implement the task at least in Java 8 or newer as a Maven project. The code should be production ready including proper test coverage. Together with the implementation please provide a short description for other developers on how to integrate and use this tool.

Your solution should be contained in one directory. Please send it via email as a zip file attachment to recruiting@lendico.de.