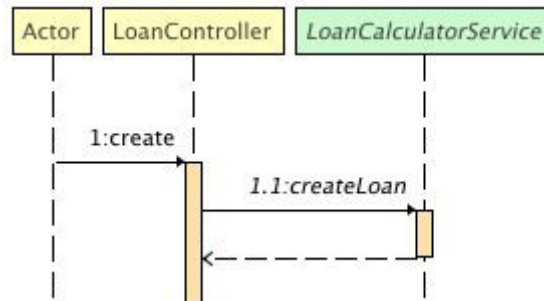


Loan Calculator Documentation

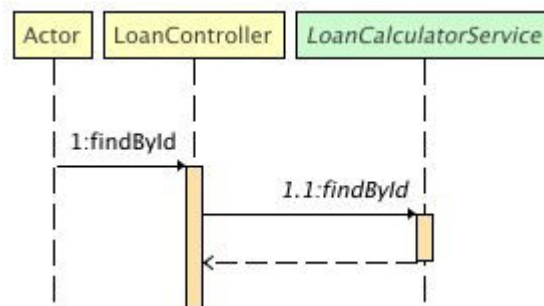
LoanCalculatorController.class

Method: *create(LoanRequestDTO)*



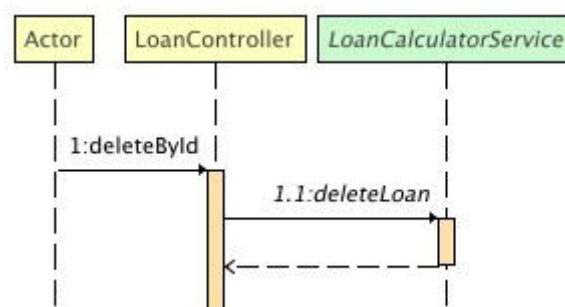
1. Client is calling *create()* method and passing the request to *api/loan-calculator/create*. Request is LoanRequestDTO object. Response is LoanResponseDTO object. HTTP method is POST.
2. Controller is calling *createLoan()* method in LoanCalculatorService.class.

Method: *findById(id)*



1. Client is calling *findById()* method and passing the id parameter to *api/loan-calculator/{id}* path. HTTP method is GET.
2. Controller is calling *findById()* method in LoanCalculatorService.class with provided id.

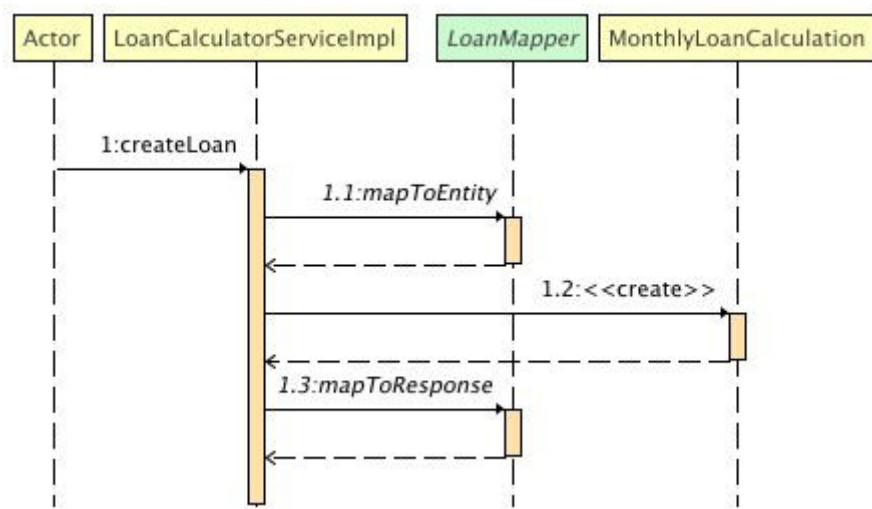
Method: *deleteById(id)*



1. Client is calling *deleteById()* method and passing the id parameter to *api/loan-calculator/{id}* path. HTTP method is DELETE.
2. Controller is calling *deleteLoan()* method in LoanCalculatorService.class with id from client.

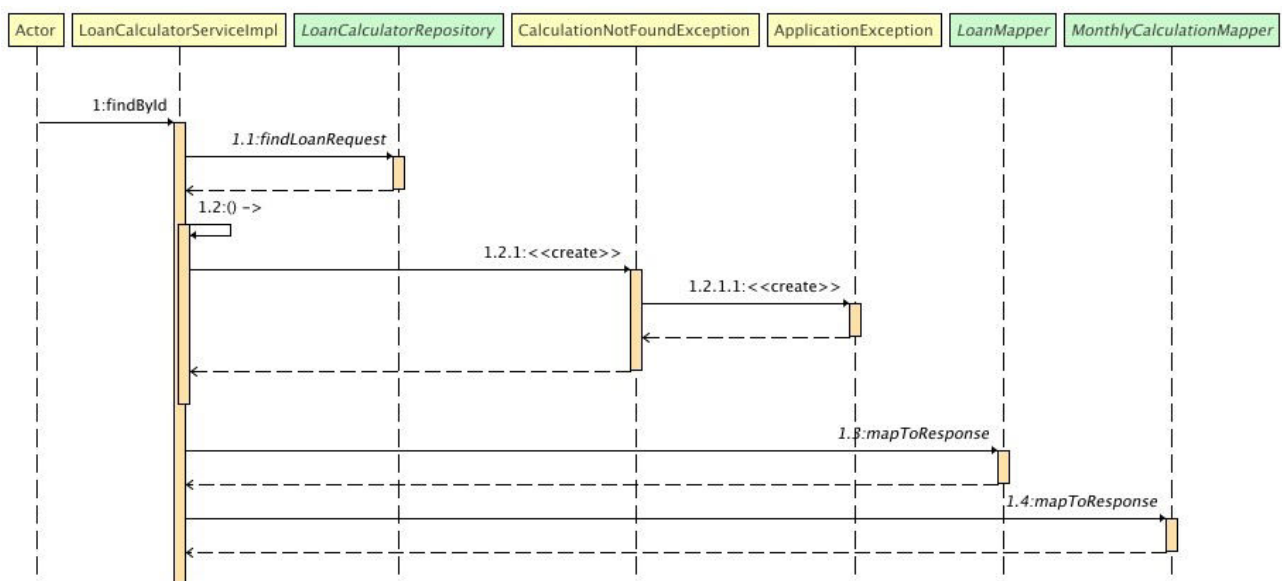
LoanCalculatorService.class

Method: *createLoan(LoanRequestDTO)*



1. Controller is passing *LoanRequestDTO* to ***createLoan()*** method to *LoanCalculatorService*.
2. *LoanMapper* is getting the passed request and mapping it into *LoanRequest* entity which will be saved into database after calculating results for each month.
3. Request will be saved into database.
4. *LoanMapper* will call ***mapToResponse()*** method to map calculated result to *LoanResponseDTO* object which will be sent as response.

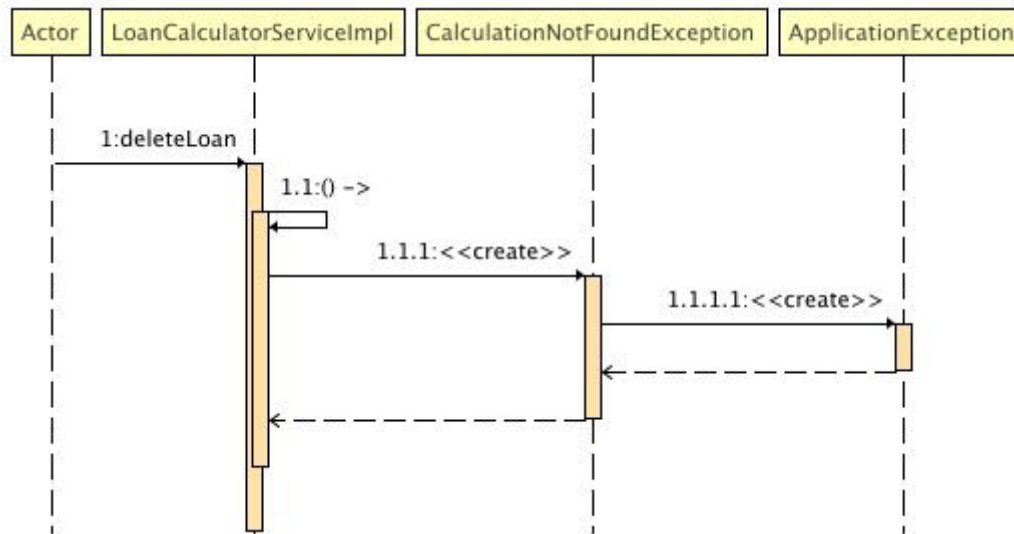
Method: *findById(id)*



1. Controller is passing *id* from client to ***findById()*** method to *LoanCalculatorService*.

2. In service class, we will use LoanCalculatorRepository to call ***findLoanRequest()*** method with provided id and expect to get LoanRequest object from database as a result. Otherwise we will get CalculationNotFoundException.
3. From retrieved request we will get all monthly calculations (MonthlyLoanCalculation.class objects) and with MonthlyCalculatorMapper we will map them to MonthlyLoanCalculationDTO objects.
4. In the last step we will have all DTO results where we are able to set them into LoanResponseDTO object as a result.

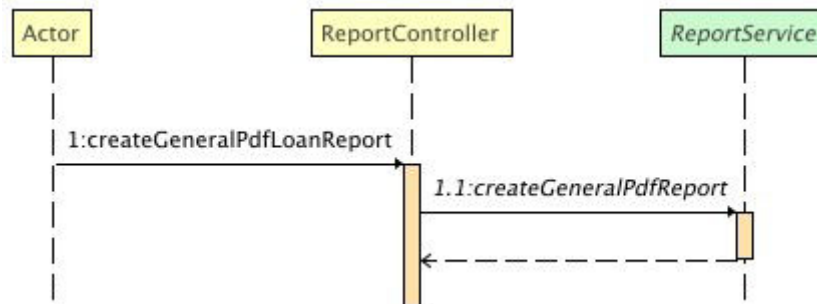
Method: *deleteLoan(id)*



1. Controller is passing id from client to ***deleteLoan()*** method into LoanCalculatorService.
2. In service class, we will use LoanCalculatorRepository to call ***findById()*** method with provided id to check does the object with provided id exist in database and to retrieve it. Otherwise we will get CalculationNotFoundException.
3. If the exception is not occurred and we have the entity with provided ID in database, repository will call ***deleteById()*** method and remove desired entity.

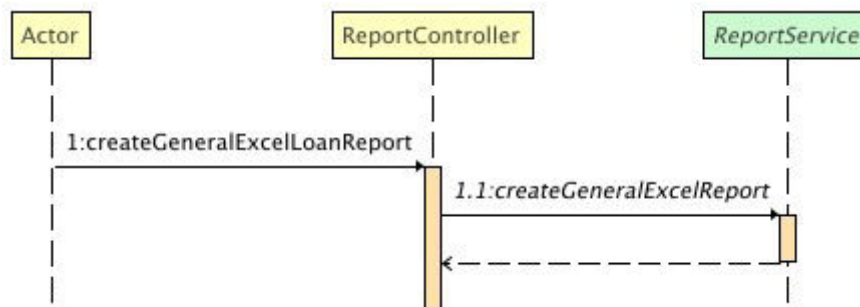
ReportController.class

Method: *createGeneralPdfLoanReport(loanId)*



1. Client is calling ***createGeneralPdfLoanReport()*** method and passing the ID parameter to *api/loan-calculator/report/pdf/{loanId}* path. Expected response is *ResponseEntity<String>* and HTTP method is GET.
2. In controller, ReportService will call ***createGeneralPdfReport()*** with provided ID as param.

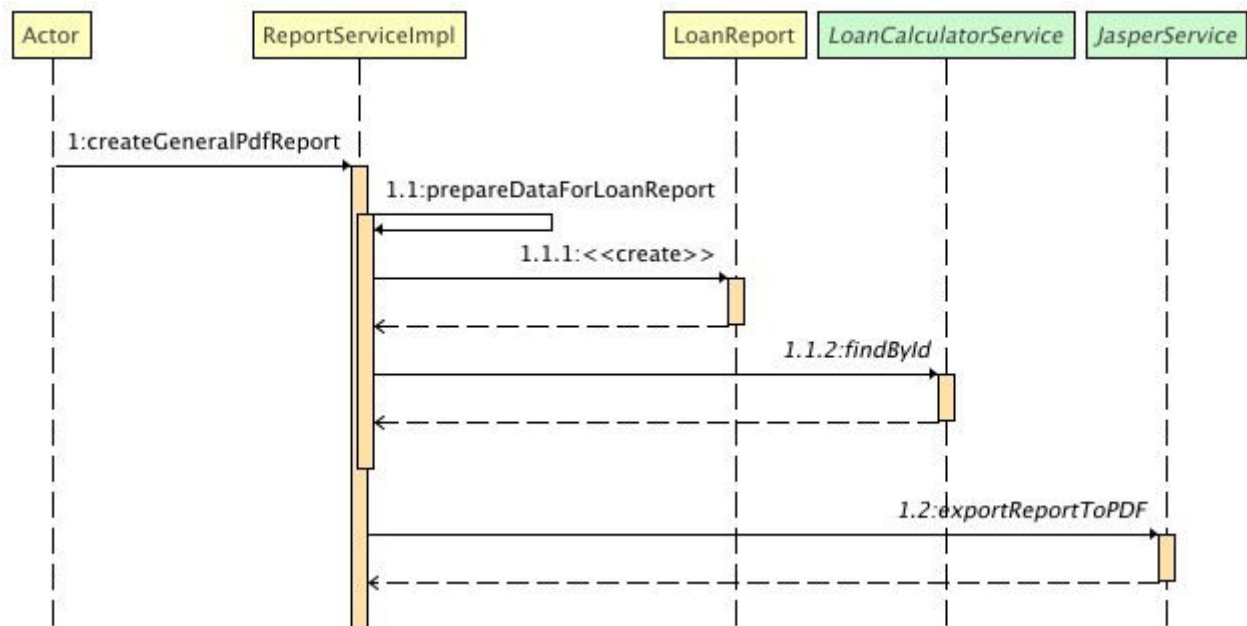
Method: *createGeneralExcelLoanReport(loanId)*



1. Client is calling ***createGeneralPdfLoanReport()*** method and passing the ID parameter to *api/loan-calculator/report/excel/{loanId}* path. Expected response is *ResponseEntity<String>* and HTTP method is GET.
2. In controller, ReportService will call ***createGeneralExcelReport()*** with provided ID as param.

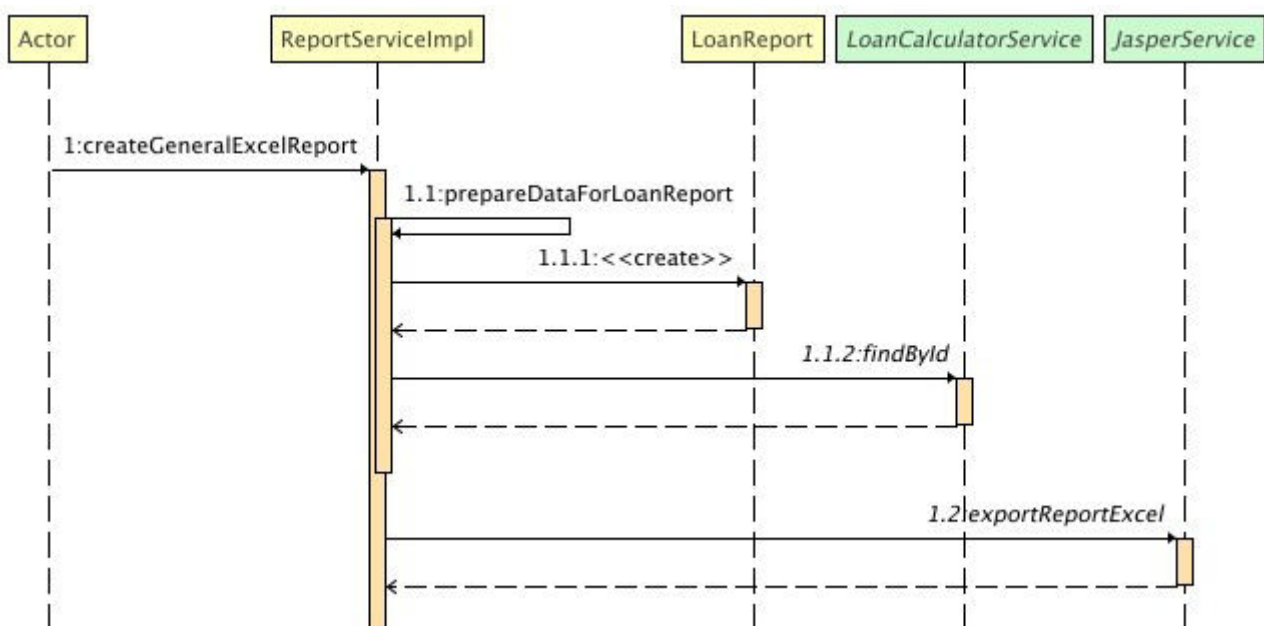
ReportService.class

Method: *createGeneralPdfReport(id)*



1. Controller is passing *id* to *createGeneralPdfReport()* method to ReportService.
2. In private method we will set and prepare desired data for PDF report and expect LoanReport object as a result.
3. In HashMap we will put populated result as a parameter.
4. We will use JasperService to call *exportReportToPdf()* method with the path where we stored our .jrxml file (template for report), map with parameters and collection if we have one (in this case we do, list of MonthlyLoanCalculationDTO objects).

Method: *createGeneralExcelReport(id)*



1. Controller is passing id to ***createGeneralExcelReport()*** method to ReportService.
2. Like we do for generating PDF reports, in private method we will set and prepare desired data for Excel report and expect LoanReport object as a result.
3. In HashMap we will put populated result as a parameter.
4. We will use JasperService to call ***exportReportExcel()*** method with the path where we stored our .jrxml file (template for report), map with parameters and collection if we have one (list of MonthlyLoanCalculationDTO objects).