Project

Book

1.0

Version

G. Jongen, J. Pieck, E. Steegmans, B. Van Impe

Authors

CONTENT

1 Introduction

OPO	Back-End Development	
Place in the OPO	Overall project	**Team (2 students)**
Documentation	Learning material in Toledo Labs of the User application	
Prerequisites	• Part 1: none • Part 2: both Book app (part 1) and User app (labs) are finished	
Learning Goals	• You can implement an application with all technologies presented in this OPO • You can analyze and implement user stories • You can work in team (2 students) • You can manage feedback • You are mastering the concepts of OO programming learned in the course • You can write clean code according to design principles introduced in the course • You can use GitHub as version control system • You can debug your code on an efficient way using the debugging tool of the editor	
Product Goals	Part 1: You create the back-end for a book registration service. Books can be added, deleted, updated, searched…. Part 2: You create the back-end for a library service. Users can borrow books, bring them back, …	

The project consists of 2 modules

1. Book: An application to register, update, search, … books of a library
2. Library: An application to manage library facilities: a registered user can borrow registered books, bring them back etc.

Module 1 (Book) runs parallel to the user-labs. In the labs (user) you exercise specific competences in addition to the theory lessons and demo code. The book project is more open.

In module 2, the User application of the labs meets the Book application of module 1. That will be impossible if you haven't completed both.

During the exam, it is allowed to use your self-written code as documentation (open laptop exam). Therefore, it is important to complete the project and to regularly ask feedback.

The project is divided in several parts. Ideally you complete each part before a new topic is introduced.

2 Before you start

2.1 Create a team

A team consists of 2 students of the same class group.

Register your team at Toledo.

In case you have a very good reason to get an exception to pairing, contact your lector.

We expect that team members work together and preferably simultaneously (pair programming). If the cooperation becomes impossible or too difficult, contact your lector. We will try to find a solution.

All students are expected to complete all tasks, even when they work individually.

2.2 Create repo in GitHub Classroom Assignment

You are expected to share your code with your teammate and lector via repo created as assignment in our GitHub classroom.

Link to the assignment: https://classroom.github.com/a/qpVFIsF9

See manual “Create and share a project”, part 1.

Name of your group is the same as your Toledo-team.

3 Asking the library for data (get request)

OPO	Back-End Development	
Place in the OPO	Java Basics	**Team**
Documentation	• Demo • Course Material	
Prerequisites	• You finished user labo 1 – 3	
Learning Goals	• domain – service – rest API with GET request • You can create a Spring Boot Project from scratch • You can analyze a user story and design a corresponding class diagram • You can write all necessary code to implement the back end as a Spring Boot project of a given user story • You can link your project to a given front end • You can share a project on GitHub and collaborate on it	
Product Goals	• A Spring Boot Project with name as described in the manual • All given tests run *(only minor adjustments are allowed, e.g. port number of local host)* • You have implemented story 1-3 (including front-end) • Your project is pushed to our GitHub classroom	

3.1 Create a Spring Boot Project and push it to the GitHub classroom

We created an assignment in a GitHub classroom. Before you start coding, follow step 2-4 in the tutorial "Create and share your project" to create a (shared) Spring Boot project connected to the repo you created via the GitHub classroom.

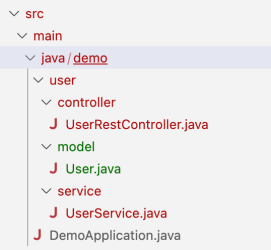
3.2 Create a class diagram and implement it

Read story 1-3. Create a class diagram necessary for their implementation.

You already know that you have to put the Java classes in the folder “src/main/java/*ownPath”* because we are working with the Spring Boot Framework. In Java, there are conventions to further structure files. We choose the following structure:

- Parent package “book” for the book project
- Subpackage “model” for the domain class “Book”
- “service” for “BookService”
- “controller” for the restcontroller

Create those packages in VS Code (as "New Folder"). Remark that the class “DemoApplication” with main method is in the root package “demo” and not in a subpackage.



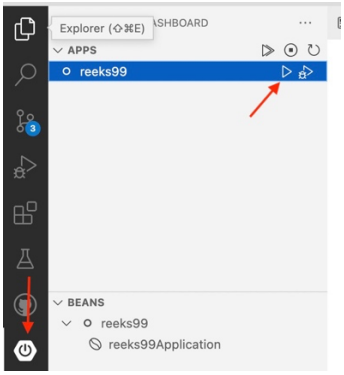
Implement all classes and methods of your class diagram. Run the given test classes until they pass. You can find the test classes on Toledo.

3.3 Make a Rest Controller

Write the necessary code so that the url’s of the stories work.

Test them in Thunderbird. Check the necessary parameters and the obtained responses.

Before you can test in Thunderbird, you must start your application:



3.4 Front End

Display your library in the browser according to the stories. Use the front end with basic functionalities you can find on Toledo. Make them work. It is not allowed to change url’s in JS.