6 Merge project “Book” with labo “User”

Until now, the "book" project and the "user" lab were two separate applications. As promised at the beginning of this document, now comes the time to bring them together.

Preparations

Make the following preparations

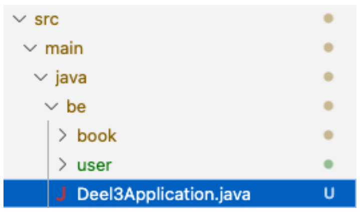
- Carefully test the code of both projects. Make sure all tests color green. Run your request-collection in Thunder Client one more time and assure yourself that everything works as requested.
- Push the code of both projects to their respective GitHub repo. Label those commits (release) e.g. with v2. That way, if it all goes wrong, you can be sure you can go back to a point where your code was still fine.
- If you are working in pairs:
    - Select one user-project that you will continue with.
    - Decide on whose laptop you will perform the merge. Don't do it partly on one, partly on the other laptop because that's guaranteed to cause major merge conflicts.

Workflow

In what follows, we start from project "Book" and add the code of labo “User” to this project. After that, we will work only with the (extended) project Book and leave the labo “User” for what it is.

Start now and open project Book.

Root Directory

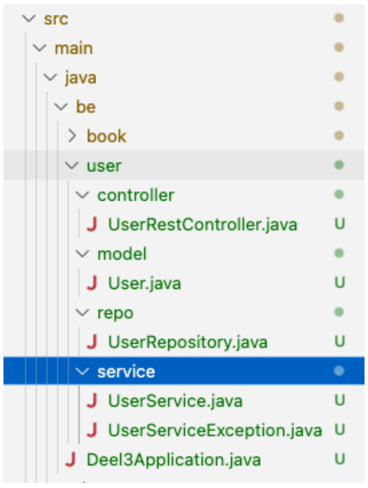


```
src
  main
    java
      be
        book
        user
        Deel3Application.java   U
```

Your project's basic document structure may be different from ours.

In what follows, we will call the directory where the java-file with annotation @SpringBootApplication (and the main method) resides the "root directory". In the screenshot above the java-file is called “Deel3Application.java” and “main/java/be” is the root directory. If your project looks different, e.g. “BookApplication.java” in “main/java/be/series1-2-3”, you may leave it as it is. If the main method is now in the package "book", you must drag it one level higher, otherwise your application will not recognize the classes in the package user.

Create packages for user classes and import them into the “Book” project



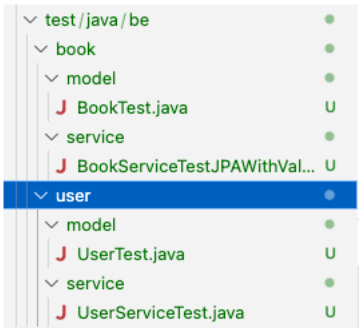
```
src
  main
    java
      be
        book
        user
          controller
            UserRestController.java      U
          model
            User.java                    U
          repo
            UserRepository.java          U
          service
            UserService.java             U
            UserServiceException.java    U
        Deel3Application.java            U
```

Create in your root directory the package “user” and subpackages “controller”, “model”, “repo” and “service”.

Import (copy) the corresponding user classes to the appropriate package. Refactor the classes if necessary (e.g. initial package declaration, imports).

Rename the two ServiceException classes to respectively UserServiceException and BookServiceException.

Import Test Classes for User



```
test/java/be
  book
    model
      BookTest.java                 U
    service
      BookServiceTestJPAWithVal...  U
  user
    model
      UserTest.java                 U
    service
      UserServiceTest.java          U
```

Do the same for the user test classes in the root test directory.

Clean your Java Workspace

You made a lot of fundamental adjustments to your project. Probably VS Code is a bit confused. Therefore, clear all of VS Code's cache: choose "F1 > Clean Java Language Server Workspace". Also delete the "data" folder where the h2 database is kept. Finally, also delete the folder "target" containing the .class files.

Run your project

Fingers crossed and run your project. All tests should pass. Run the request-collections “User” and “Book” in Thunder Client. No errors 500 or 405 should appear.
Test also both front-ends.

Commit to GitHub

Commit the code to GitHub. Your teammate can pull the code to their own computer and test the functionalities.

7 CRUD

In computer programming, create, read, update, and delete (often referred to via the acronym CRUD) are the four basic operations of persistent storage. CRUD is also sometimes used to describe user interface conventions that facilitate viewing, searching, and changing information using computer-based forms and reports.

https://en.wikipedia.org/wiki/Create,_read,_update_and_delete

As for the project “Book”, we have already coded “create”, “read” and “delete”. We are left with the update functionality. Implement story 7 so that the book app includes all CRUD operations.

Don’t forget to complete the class diagram.

You can find some tests in the startercode on GitHub (deel-3/test). Copy these test methods to the book test class you already have in your project.

There is no front-end to test this functionality. Create an extra request in your Thunder Client collection.