

Exercises:

Introduction

We continue with the hibernateExercises project. In the previous workshop we used a `Many_To_One` relationship between `Book` and `Author`.

In this way, the `Book` class has information about the author. But the `Author` class does not have any information about the books.

We can get this information by having a one to many relationship in author class.

So we change the direction of the relationship from *many to one* from `Book` class to `Author` class to *one to many* relationship from `Author` to `Book`.

Overview

In this exercise:

- We will practice having two classes and make a one to many relationship between these two classes and see how it will affect our database.
- We use tables `Author` and `Book`. We map the `Author` class to the `Book` class and make a one to many relationship. It means that a book can be written by only one author but an author can write one or more books.
- In this exercise, we use `Set` as the collection.
- We test our code by creating some books and an author.

Tasks

- Do the necessary changes in the `Author` and `Book` classes (adding property, adding annotations in `Author`, removing annotation in `Book` and any code about the `Author`, changing the constructors, etc)
- Add a method in your `Author` class to add a book to the set of books written by the author.
- Remember that you need a get method to get the set of books.
- Drop both tables from `ij`.
- In the main method, create three books and one author and save (persist, insert) them in the database and run the code.
- Run the code and see the results in `ij`. How many tables do you have now?
- Now retrieve the author from the database by its id. (get method on session) and then retrieve the books for this specific author and print them.

- Change your setting from creating the third table to having foreign key using `@JoinColumn` (name="AUTHOR_FK")
- Drop tables in the ij.
- Test your code by creating new books. Run the code. How many tables do you have in ij? What did `@JoinColumn` do?