Spring Data - Exercise 1

Problem Statement

A library needs an application to manage its books with following functionalities:

1. Get book details
2. Add book details
3. Get book details for a given author name
4. Get book details whose price is greater than or equal to given price
5. Get book details whose price is less than given price
6. Get book details published between a given period
7. Get book details published after a given year
8. Get book details for a given author name and publisher
9. Update book details
10. Delete book details

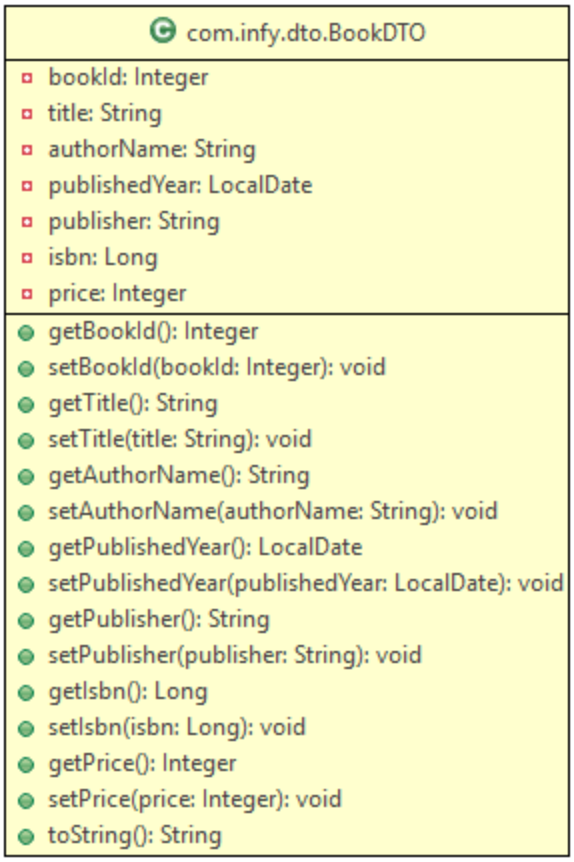
This application is partially implemented. You have to implement validator, service, entity classes and specify methods in repository interface according to the requirement. Follow the following instructions before starting coding:

1. Download the project for MySQL database.
2. Import the downloaded project in your workspace.
3. Modify the database credentials in **application.properties** file and execute the table scripts.

The description of different classes is given below:

1. DTO class (already implemented)

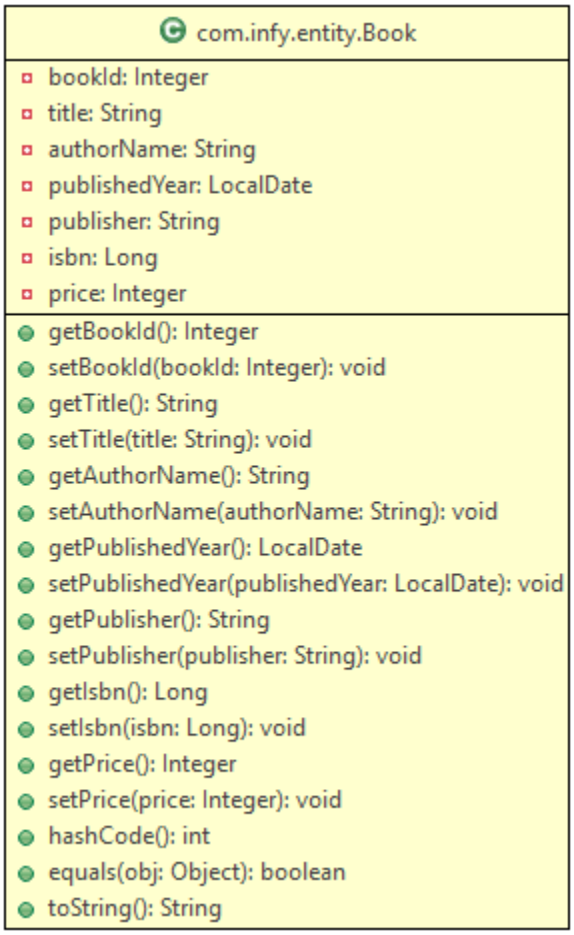
BookDTO



2. Entity class (to be implemented)

Book

- Implement this entity class according to the class diagram and description given below:



- It should be mapped to the Book table with bookId as identifier.

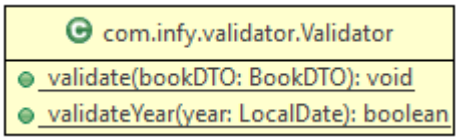
3. Repository (to be implemented)

BookRepository

- Create this interface by extending appropriate repository interface to perform database operations using Book entity class.
- Add following methods to this interface :
    - It should accept the name of the author and return all the books authored by him as List<Book>.
    - It should accept the price of book and return all the books whose price is greater than or equal to given price as List<Book>.
    - It should accept the price of book and return all the books whose price is less than the given price as List<Book>.
    - It should accept range of dates and return all the books which are published between a given dates as List<Book>.
    - It should accept date and return all the books which are published after the given date as List<Book>.
    - It should accept the name of author and publisher and return all the books having same author and publisher as List<Book>

4. Validator class (to be implemented)

This class validates the book details. Implement this class according to the class diagram and description given below:



Method Description

validate(Book book)

- This method validates the book's publishing year by calling validateYear() and throws exception with message "Validator.INVALID_YEAR" if year is invalid.

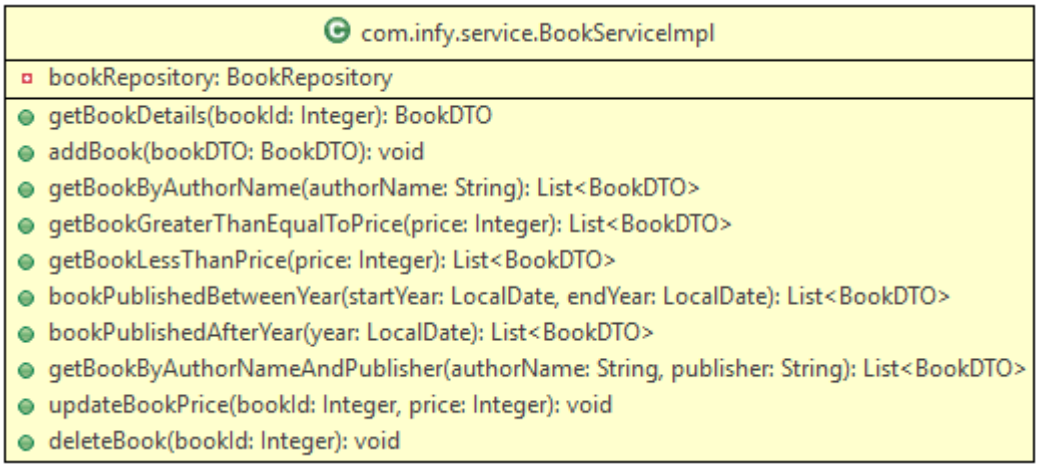
validateYear(LocalDate year)

- This method validates the publishing year of the book according to the following rules:
  - It should not be after the present date.
  - It should be before the present date.
- If the year is valid, it returns **true** else it returns **false**.

4. Service class (to be implemented)

BookServiceImpl class

This is the service class of the application. Implement this class according to the class diagram and description given below:



- Inject bookRepository using autowiring.
- Annotate it with appropriate stereotype annotation with value as "bookService":
- Annotate it with appropriate annotation for managing transactions.

Method Description

getBookDetails(Integer bookId)

- This method fetches book details based on bookId.
- Invoke the appropriate method of BookRepository to fetch book details based on bookId.
- If books are found, then return the book details.
- Else, throw an object of InfyBookException with the message Service.BOOK_DETAILS_NOT_FOUND.

addBook(BookDTO bookDTO)

- This method adds book details to the database and returns the title of the book.
- Invoke the validate() of Validator class to validate the book details.
- If the validation is successful, invoke the appropriate method of BookRepository to check if the given book is already present in database or not.
- If the book is already present, throw an object of InfyBookException with the message Service.BOOK_ALREADY_PRESENT.
- Else add the book details to the database by invoking appropriate method of BookRepository and return the title of the book.

getBookByAuthorName(String authorName)

- This method fetches the books authored by given author.
- Invoke the appropriate method of BookRepository to fetch the the books authored by given author.
- If books are found, then return the books.
- Else, throw an object of InfyBookException with the message Service.BOOK_NOT_FOUND.

getBookGreaterThanEqualToPrice (Integer price)

- This method fetches the books whose price greater than or equal to a given price
- Invoke the appropriate method of BookRepository, which fetches the books whose price greater than or equal to a given price
- If books are found, then return the books.
- Else, throw an object of InfyBookException with the message Service.BOOKS_NOT_FOUND.

getBookLessThanPrice (Integer price)

- This method fetches the books whose price less than given price.
- Invoke the appropriate method of BookRepository, whichfetches the books whose price less than given price.
- If books are found, then return the books.
- Else, throw an object of InfyBookException with the message Service.BOOKS_NOT_FOUND.

bookPublishedBetweenYear(LocalDate startYear, LocalDate endYear)

- This method fetches the books which are published between startYear and endYear.
- Invoke the appropriate method of BookRepository, which fetches books which are published between startYear and endYear.
- If books are found, then return the books.
- Else, throw an object of InfyBookException with the message Service.BOOKS_NOT_FOUND.

bookPublishedAfterYear(LocalDate year)

- This method fetches the books which are published after the given year.
- Invoke the appropriate method of BookRepository, which fetches books which are published after the given year.
- If books are found, then return the books.
- Else, throw an object of InfyBookException with the message Service.BOOKS_NOT_FOUND.

getBookByAuthorNameAndPublisher (String authorName, String publisher)

- This method fetches the books published by a particular author and publisher.
- Invoke the appropriate method of BookRepository, which fetches the books published by a particular author and publisher.
- If books are found, then return the books.
- Else, throw an object of InfyBookException with the message Service.BOOKS_NOT_FOUND.

updateBookPrice(Integer bookId, Integer newPrice)

- This method updates the price of the book based on bookId.
- Invoke the appropriate method of BookRepository which fetches the book details based on bookId.
- If book is not found, throw an object of InfyBookException with the message "Service.BOOK_NOT_FOUND".
- Else, update the book price in database by invoking appropriate method of BookRepository.

deleteBook(Integer bookId)

- This method deletes the book details based on bookId.
- Invoke the appropriate method of BookRepository which fetches the book details based on bookId.
- If book is not found throw an object of InfyBookException with the message "Service.BOOK_NOT_FOUND".
- Else, delete the book from the database by invoking appropriate method of BookRepository interface.

5. IBookToTraineeApplication (already implemented)

This class is used to run the application. The methods of this class are already implemented. You can modify the inputs to check output for various scenarios. Some valid input values and expected output is given below:

getBookDetails()

- **Input**
  - bookId = 1001
- **Output**

```
BookDTO [bookId=1001, title=The Immortals of Meluha, authorName=Amish Tripathi, publishedYear=2010-02-03, publisher=Westland Press, isbn=7893451237893, price=540]
```

addBook()

- **Input**
  - bookId = 1010
  - title=”The Da Vinci Code”
  - authorName=”Dan Brown”
  - year=18-Apr-2003
  - publisher=”Doubleday”
  - isbn=1456987609875l
  - price=980
- Output

```
Book details successfully added.
```

getBookByAuthorName()

- Input
  - authorName=”Nicholas Sparks”,
- Output

```
BookDTO [bookId=1003, title=The Notebook, authorName=Nicholas Sparks, publishedYear=1996-03-15, publisher=Warner Books, isbn=7893400211822, price=845]
BookDTO [bookId=1006, title=Message in a Bottle, authorName=Nicholas Sparks, publishedYear=1998-03-15, publisher=Warner Books, isbn=2293411211822, price=820]
BookDTO [bookId=1007, title=The Choice, authorName=Nicholas Sparks, publishedYear=2007-09-24, publisher=Grand Central, isbn=8993499211894, price=711]
```

getBookGreaterThanEqualToPrice()

- Input
  - price= 700

- Output

```
BookDTO [bookId=1002, title=The Alchemist, authorName=Paulo Coelho, publishedYear=1993-01-15, publisher=Harper Collins, isbn=7893400211893, price=700]
BookDTO [bookId=1003, title=The Notebook, authorName=Nicholas Sparks, publishedYear=1996-03-15, publisher=Warner Books, isbn=7893400211822, price=845]
BookDTO [bookId=1006, title=Message in a Bottle, authorName=Nicholas Sparks, publishedYear=1998-03-15, publisher=Warner Books, isbn=2293411211822, price=820]
BookDTO [bookId=1007, title=The Choice, authorName=Nicholas Sparks, publishedYear=2007-09-24, publisher=Grand Central, isbn=8993499211894, price=711]
BookDTO [bookId=1010, title=The Da Vinci Code, authorName=Dan Brown, publishedYear=2003-04-18, publisher=Doubleday, isbn=1456987609875, price=980]
```

getBookLessThanPrice(

- Input
  - price= 600
- Output

```
BookDTO [bookId=1001, title=The Immortals of Meluha, authorName=Amish Tripathi, publishedYear=2010-02-03, publisher=Westland Press, isbn=7893451237893, price=540]
BookDTO [bookId=1009, title=The Oath of Vayuputra, authorName=Amish Tripathi, publishedYear=2013-07-23, publisher=Westland Press, isbn=4567099211894, price=540]
```

bookPublishedBetweenYear()

- Input
  - startYear=22-12-1990
  - endYear= 22-12-2000
- Output

```
BookDTO [bookId=1002, title=The Alchemist, authorName=Paulo Coelho, publishedYear=1993-01-15, publisher=Harper Collins, isbn=7893400211893, price=700]
BookDTO [bookId=1003, title=The Notebook, authorName=Nicholas Sparks, publishedYear=1996-03-15, publisher=Warner Books, isbn=7893400211822, price=845]
BookDTO [bookId=1005, title=A Fine Balance, authorName=Rohinton Mistry, publishedYear=1995-03-22, publisher=McClelland, isbn=1289560027889, price=679]
BookDTO [bookId=1006, title=Message in a Bottle, authorName=Nicholas Sparks, publishedYear=1998-03-15, publisher=Warner Books, isbn=2293411211822, price=820]
```

bookPublishedAfterYear()

- Input
  - year=22-12-2000
- Output

```
BookDTO [bookId=1001, title=The Immortals of Meluha, authorName=Amish Tripathi, publishedYear=2010-02-03, publisher=Westland Press, isbn=7893451237893, price=540]
BookDTO [bookId=1007, title=The Choice, authorName=Nicholas Sparks, publishedYear=2007-09-24, publisher=Grand Central, isbn=8993499211894, price=711]
BookDTO [bookId=1008, title=The Secret of Nagas, authorName=Amish Tripathi, publishedYear=2011-07-21, publisher=Westland Press, isbn=1290099211894, price=650]
BookDTO [bookId=1009, title=The Oath of Vayuputra, authorName=Amish Tripathi, publishedYear=2013-07-23, publisher=Westland Press, isbn=4567099211894, price=540]
BookDTO [bookId=1010, title=The Da Vinci Code, authorName=Dan Brown, publishedYear=2003-04-18, publisher=Doubleday, isbn=1456987609875, price=980]
```

getBookByAuthorNameAndPublisher()

- Input
  - authorName=”Amish Tripathi”
  - publisher=”Westland Press”
- Output

```
BookDTO [bookId=1001, title=The Immortals of Meluha, authorName=Amish Tripathi, publishedYear=2010-02-03, publisher=Westland Press, isbn=7893451237893, price=540]
BookDTO [bookId=1008, title=The Secret of Nagas, authorName=Amish Tripathi, publishedYear=2011-07-21, publisher=Westland Press, isbn=1290099211894, price=650]
BookDTO [bookId=1009, title=The Oath of Vayuputra, authorName=Amish Tripathi, publishedYear=2013-07-23, publisher=Westland Press, isbn=4567099211894, price=540]
```

updateBookPrice()

- Input
  - bookId = 1005
  - price = 850
- Output

```
Book price successfully updated.
```

deleteBook()

- Input
  - bookId = 1005
- Output

```
Book details successfully deleted.
```