

```

1 package com.infy.entity;
2
3 import java.time.LocalDate;
4
5 @Entity
6 public class Book {
7     @Id
8     private Integer bookId;
9
10    @Column(name="book_name")
11    private String title;
12    private String authorName;
13
14    @Column(name="year")
15    private LocalDate publishedYear;
16    private String publisher;
17    private Long isbn;
18    private Integer price;
19    public Integer getBookId() {
20        return bookId;
21    }
22    public String getTitle() {
23        return title;
24    }
25    public String getAuthorName() {
26        return authorName;
27    }
28    public LocalDate getPublishedYear() {
29        return publishedYear;
30    }
31    public String getPublisher() {
32        return publisher;
33    }
34    public Long getIsbn() {
35        return isbn;
36    }
37    public Integer getPrice() {
38        return price;
39    }
40    public void setBookId(Integer bookId) {
41        this.bookId = bookId;
42    }
43    public void setTitle(String title) {
44        this.title = title;
45    }
46
47
48
49

```

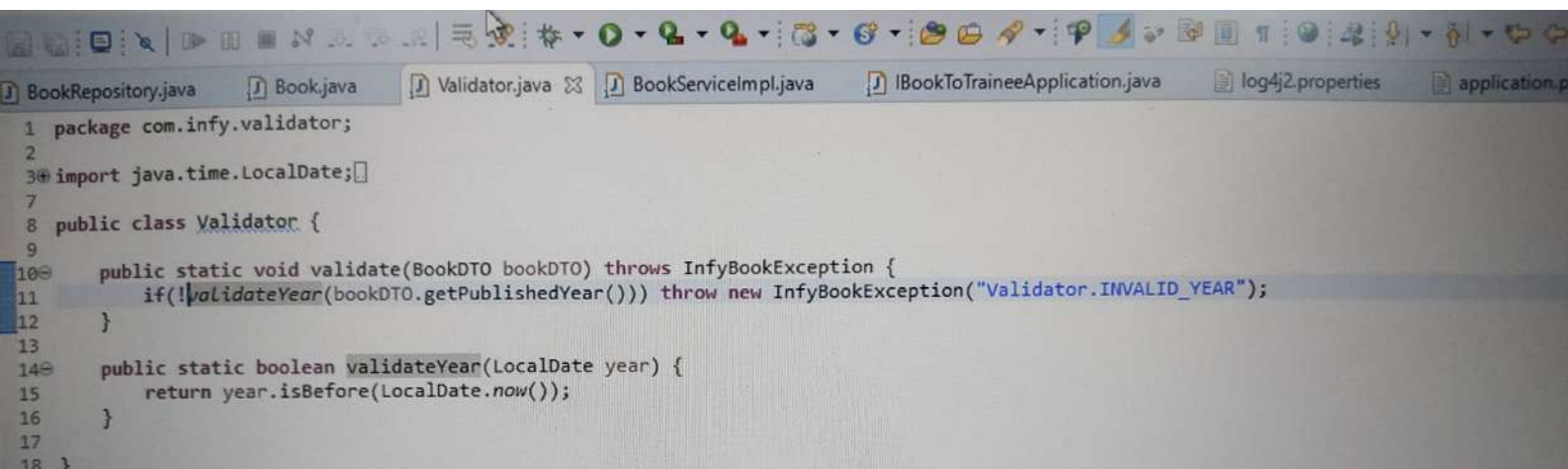
```
49     }
50     public void setAuthorName(String authorName) {
51         this.authorName = authorName;
52     }
53     public void setPublishedYear(LocalDate publishedYear) {
54         this.publishedYear = publishedYear;
55     }
56     public void setPublisher(String publisher) {
57         this.publisher = publisher;
58     }
59     public void setIsbn(Long isbn) {
60         this.isbn = isbn;
61     }
62     public void setPrice(Integer price) {
63         this.price = price;
64     }
65     @Override
66     public int hashCode() {
67         final int prime = 31;
68         int result = 1;
69         result = prime * result + ((bookId == null) ? 0 : bookId.hashCode());
70         return result;
71     }
72     @Override
73     public boolean equals(Object obj) {
74         if (this == obj)
75             return true;
76         if (obj == null)
77             return false;
78         if (getClass() != obj.getClass())
79             return false;
80         Book other = (Book) obj;
81         if (bookId == null) {
82             if (other.bookId != null)
83                 return false;
84         } else if (!bookId.equals(other.bookId))
85             return false;
86         return true;
87     }
88     @Override
89     public String toString() {
90         return "Book [bookId=" + bookId + ", title=" + title + ", authorName=" + authorName + ", publishedYear="
91             + publishedYear + ", publisher=" + publisher + ", isbn=" + isbn + ", price=" + price + "];"
92     }
93 }
```

eclipse-workspace - IBookToTrainee/src/main/java/com/infy/repository/BookRepository.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

BookRepository.java Book.java Validator.java BookServiceImpl.java IBookToTraineeApplication.java

```
1 package com.infy.repository;
2
3 import java.time.LocalDate;
4 import java.util.List;
5
6
7 |
8 import org.springframework.data.repository.CrudRepository;
9 import org.springframework.stereotype.Repository;
10
11 import com.infy.entity.Book;
12
13 @Repository(value="bookRepository")
14 public interface BookRepository extends CrudRepository<Book, Integer> {
15
16     List<Book> findByAuthorName(String authorName);
17
18     List<Book> findByPriceGreaterThanOrEqual(Integer price);
19
20     List<Book> findByPriceLessThan(Integer price);
21
22     List<Book> findByPublishedYearBetween(LocalDate fromDate, LocalDate toDate);
23
24     List<Book> findByPublishedYearAfter(LocalDate fromDate);
25
26     // @Query("SELECT B FROM Book B WHERE B.authorName = ?1 AND B.publisher ?2")
27     List<Book> findByAuthorNameAndPublisher(String authorName, String publisher);
28
29 }
```



```
1 package com.infy.validator;
2
3 import java.time.LocalDate;
4
5 public class Validator {
6
7     public static void validate(BookDTO bookDTO) throws InfyBookException {
8         if(!validateYear(bookDTO.getPublishedYear())) throw new InfyBookException("Validator.INVALID_YEAR");
9     }
10
11     public static boolean validateYear(LocalDate year) {
12         return year.isBefore(LocalDate.now());
13     }
14 }
15
16
17
18 }
```


BookRepository.java Book.java Validator.java BookServiceImpl.java IBookToTraineeApplication.java log4j2.properties

```
1 package com.infy.service;
2
3 import java.time.LocalDate;
4 import java.util.ArrayList;
5 import java.util.List;
6 import java.util.Optional;
7
8 import javax.transaction.Transactional;
9
10 import org.springframework.beans.factory.annotation.Autowired;
11 import org.springframework.stereotype.Service;
12
13 import com.infy.dto.BookDTO;
14 import com.infy.entity.Book;
15 import com.infy.exception.InfyBookException;
16 import com.infy.repository.BookRepository;
17 import com.infy.validator.Validator;
18
19 @Service(value="bookService")
20 @Transactional
21 public class BookServiceImpl implements BookService {
22
23     @Autowired
24     private BookRepository bookRepository;
25
26     @Override
27     public BookDTO getBookDetails(Integer bookId) throws InfyBookException {
28         Optional<Book> optional = bookRepository.findById(bookId);
29
30         Book book = optional.orElseThrow(() -> new InfyBookException("Service.BOOK_DETAILS_NOT_FOUND"));
31
32         BookDTO bookDTO = new BookDTO();
33
34         bookDTO.setBookId(book.getBookId());
35         bookDTO.setAuthorName(book.getAuthorName());
36         bookDTO.setIsbn(book.getIsbn());
37         bookDTO.setPrice(book.getPrice());
38         bookDTO.setPublishedYear(book.getPublishedYear());
39         bookDTO.setPublisher(book.getPublisher());
40         bookDTO.setTitle(book.getTitle());
41
42         return bookDTO;
43     }
44
45     @Override
```

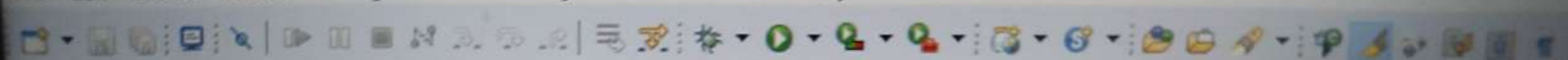
```

BookRepository.java  Book.java  Validator.java  BookServiceImpl.java  IBookToTraineeApplication.java  log4j2.properties  application
45 @Override
46 public void addBook(BookDTO bookDTO) throws InfyBookException {
47     Validator.validate(bookDTO);
48
49     Optional<Book> optional = bookRepository.findById(bookDTO.getBookId());
50
51     if(optional.isPresent())
52         throw new InfyBookException("Service.BOOK_ALREADY_PRESENT");
53
54     Book book = new Book();
55
56     book.setBookId(bookDTO.getBookId());
57     book.setAuthorName(bookDTO.getAuthorName());
58     book.setIsbn(bookDTO.getIsbn());
59     book.setPrice(bookDTO.getPrice());
60     book.setPublishedYear(bookDTO.getPublishedYear());
61     book.setPublisher(bookDTO.getPublisher());
62     book.setTitle(bookDTO.getTitle());
63
64     bookRepository.save(book);
65 }
66
67 @Override
68 public List<BookDTO> getBookByAuthorName(String authorName) throws InfyBookException {
69     List<Book> books = bookRepository.findByAuthorName(authorName);
70     if(books.isEmpty()) throw new InfyBookException("Service.BOOK_NOT_FOUND");
71
72     List<BookDTO> bookDTOs = new ArrayList<BookDTO>();
73
74     for(Book book:books) {
75         BookDTO bookDTO = new BookDTO();
76
77         bookDTO.setBookId(book.getBookId());
78         bookDTO.setAuthorName(book.getAuthorName());
79         bookDTO.setIsbn(book.getIsbn());
80         bookDTO.setPrice(book.getPrice());
81         bookDTO.setPublishedYear(book.getPublishedYear());
82         bookDTO.setPublisher(book.getPublisher());
83         bookDTO.setTitle(book.getTitle());
84
85         bookDTOs.add(bookDTO);
86     }
87
88     return bookDTOs;
89 }

```

eclipse-workspace - IBook_ToTrainee/src/main/java/com/infy/service/BookServiceImpl.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help



BookRepository.java Book.java Validator.java BookServiceImpl.java IBookToTraineeApplication.java

```
91 @Override
92 public List<BookDTO> getBookGreaterThanOrEqualToPrice(Integer price) throws InfyBookException {
93     List<Book> books = bookRepository.findByPriceGreaterThanOrEqualTo(price);
94     if(books.isEmpty()) throw new InfyBookException("Service.BOOK_NOT_FOUND");
95
96     List<BookDTO> bookDTOS = new ArrayList<BookDTO>();
97
98     for(Book book:books) {
99         BookDTO bookDTO = new BookDTO();
100
101         bookDTO.setBookId(book.getBookId());
102         bookDTO.setAuthorName(book.getAuthorName());
103         bookDTO.setIsbn(book.getIsbn());
104         bookDTO.setPrice(book.getPrice());
105         bookDTO.setPublishedYear(book.getPublishedYear());
106         bookDTO.setPublisher(book.getPublisher());
107         bookDTO.setTitle(book.getTitle());
108
109         bookDTOS.add(bookDTO);
110     }
111
112     return bookDTOS;
113 }
114
```

```
BookRepository.java Book.java Validator.java BookServiceImpl.java IBookToTraineeApplication.java log4j2.properties
112     return bookDTOs;
113 }
114
115 @Override
116 public List<BookDTO> getBookLessThanPrice(Integer price) throws InfyBookException {
117     List<Book> books = bookRepository.findByPriceLessThan(price);
118     System.out.println(books.isEmpty());
119     if(books.isEmpty()) throw new InfyBookException("Service.BOOK_NOT_FOUND");
120
121     List<BookDTO> bookDTOs = new ArrayList<BookDTO>();
122
123     for(Book book:books) {
124         BookDTO bookDTO = new BookDTO();
125
126         bookDTO.setBookId(book.getBookId());
127         bookDTO.setAuthorName(book.getAuthorName());
128         bookDTO.setIsbn(book.getIsbn());
129         bookDTO.setPrice(book.getPrice());
130         bookDTO.setPublishedYear(book.getPublishedYear());
131         bookDTO.setPublisher(book.getPublisher());
132         bookDTO.setTitle(book.getTitle());
133
134         bookDTOs.add(bookDTO);
135     }
136
137     return bookDTOs;
138 }
139
```


eclipse-workspace - IBook_ToTrainee/src/main/java/com/infy/service/BookServiceImpl.java - Eclipse IDE

Edit Source Refactor Navigate Search Project Run Window Help



BookRepository.java Book.java Validator.java BookServiceImpl.java IBookToTraineeApplication.java log4j2.properties app

```
139
140 @Override
141 public List<BookDTO> bookPublishedBetweenYear(LocalDate startYear, LocalDate endYear) throws InfyBookException {
142     List<Book> books = bookRepository.findByPublishedYearBetween(startYear, endYear);
143     System.out.println(books.isEmpty());
144     if(books.isEmpty()) throw new InfyBookException("Service.BOOK_NOT_FOUND");
145
146     List<BookDTO> bookDTOS = new ArrayList<BookDTO>();
147
148     for(Book book:books) {
149         BookDTO bookDTO = new BookDTO();
150
151         bookDTO.setBookId(book.getBookId());
152         bookDTO.setAuthorName(book.getAuthorName());
153         bookDTO.setIsbn(book.getIsbn());
154         bookDTO.setPrice(book.getPrice());
155         bookDTO.setPublishedYear(book.getPublishedYear());
156         bookDTO.setPublisher(book.getPublisher());
157         bookDTO.setTitle(book.getTitle());
158
159         bookDTOS.add(bookDTO);
160     }
161
162     return bookDTOS;
163 }
164
```

eclipse-workspace - IBook_ToTrainee/src/main/java/com/infy/service/BookServiceImpl.java - Eclipse IDE

Edit Source Refactor Navigate Search Project Run Window Help



BookRepository.java Book.java Validator.java BookServiceImpl.java IBookToTraineeApplication.java log4j2.properties

```
164
165 @Override
166 public List<BookDTO> bookPublishedAfterYear(LocalDate year) throws InfyBookException {
167     List<Book> books = bookRepository.findByPublishedYearAfter(year);
168     System.out.println(books.isEmpty());
169     if(books.isEmpty()) throw new InfyBookException("Service.BOOK_NOT_FOUND");
170
171     List<BookDTO> bookDTOS = new ArrayList<BookDTO>();
172
173     for(Book book:books) {
174         BookDTO bookDTO = new BookDTO();
175
176         bookDTO.setBookId(book.getBookId());
177         bookDTO.setAuthorName(book.getAuthorName());
178         bookDTO.setIsbn(book.getIsbn());
179         bookDTO.setPrice(book.getPrice());
180         bookDTO.setPublishedYear(book.getPublishedYear());
181         bookDTO.setPublisher(book.getPublisher());
182         bookDTO.setTitle(book.getTitle());
183
184         bookDTOS.add(bookDTO);
185     }
186
187     return bookDTOS;
188 }
189
```

eclipse-workspace - IBook_ToTrainee/src/main/java/com/infy/service/BookServiceImpl.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help



BookRepository.java Book.java Validator.java BookServiceImpl.java IBookToTraineeApplication.java log4j2.properties application.properties

```
190 @Override
191 public List<BookDTO> getBookByAuthorNameAndPublisher(String authorName, String publisher) throws InfyBookException {
192     List<Book> books = bookRepository.findByAuthorNameAndPublisher(authorName, publisher);
193     System.out.println(books.isEmpty());
194     if(books.isEmpty()) throw new InfyBookException("Service.BOOK_NOT_FOUND");
195
196     List<BookDTO> bookDTOS = new ArrayList<BookDTO>();
197
198     for(Book book:books) {
199         BookDTO bookDTO = new BookDTO();
200
201         bookDTO.setBookId(book.getBookId());
202         bookDTO.setAuthorName(book.getAuthorName());
203         bookDTO.setIsbn(book.getIsbn());
204         bookDTO.setPrice(book.getPrice());
205         bookDTO.setPublishedYear(book.getPublishedYear());
206         bookDTO.setPublisher(book.getPublisher());
207         bookDTO.setTitle(book.getTitle());
208
209         bookDTOS.add(bookDTO);
210     }
211
212     return bookDTOS;
213 }
214
215
216 @Override
217 public void updateBookPrice(Integer bookId, Integer price) throws InfyBookException {
218
219     Optional<Book> optional = bookRepository.findById(bookId);
220
221     Book book = optional.orElseThrow(()-> new InfyBookException("Service.BOOK_NOT_FOUND"));
222     book.setPrice(price);
223 }
224
225 @Override
226 public void deleteBook(Integer bookId) throws InfyBookException {
227     Optional<Book> optional = bookRepository.findById(bookId);
228
229     optional.orElseThrow(()-> new InfyBookException("Service.BOOK_NOT_FOUND"));
230     bookRepository.deleteById(bookId);
231 }
232
233 }
234
```