Assessment 6

Make the Book Store App functionality accessible via Web API.

<table>
<tr><td>getBook(name)</td><td>Returns the book object if found otherwise “Not found” exception.</td></tr>
<tr><td>getBooksByPageCount(pageCount)</td><td>Returns the books with pages >= the pageCount parameters. E.g. Calling the function with pageCount=200 should return all the books with pages &gt;= 200.</td></tr>
<tr><td>getBooksByAuthor(author)</td><td>Returns all the books authored by that specific author.
<b>Note</b>: some books have more than one author. You should consider those too and return them as well.</td></tr>
<tr><td>getBooksbyCatagory(category)</td><td>Returns the books for a particular category.
E.g. Calling the function with <i>category = Programming</i> should return all the programming books.</td></tr>
<tr><td>getAuthorsBookCount()</td><td>Returns a map that contains the author name and the number of books they have authored. E.g.
<table>
<tr><th>Author Name</th><th>Book Count</th></tr>
<tr><td>James</td><td>2</td></tr>
<tr><td>Ali</td><td>4</td></tr>
</table>
</td></tr>
</table>

Develop Web API to make the Book App functionality accessible via the web:

- Create **app.js** file, import express package, instantiate an express app, configure it then start it at port 8090.
- Create a **book-service.js** file and implement **BookService** class with methods to handle the requests listed in the table below. The handlers should use the provided methods from the **BookRepository** class.
- Create a **book-router.js** file and implement handlers for the Urls listed in the table below. The Url handlers should use the methods from the BookService class.

HTTP Verb	**Url**	**Functionality**
Get	/api/books?name=	Returns the book by name
Get	/api/books?pageCount=	Returns the books with pages >= the pageCount parameters. E.g. Calling the function with pageCount=200 should return all the books with pages >= 200.

<table>
<tr><td>Get</td><td>/api/books?author=</td><td>Returns all the books authored by that specific author.</td></tr>
<tr><td>Get</td><td>/api/books?category=</td><td>Returns the books for a particular category.
E.g. Calling the function with <i>category = Programming</i> should return all the programming books.</td></tr>
<tr><td>Get</td><td>/api/books/summary</td><td>Returns a map that contains the author name and the number of books they have authored. E.g.<table><tr><th>Author Name</th><th>Book Count</th></tr><tr><td>James</td><td>2</td></tr><tr><td>Ali</td><td>4</td></tr></table></td></tr>
<tr><td>Post</td><td>/api/books/</td><td>Adds a book</td></tr>
<tr><td>Put</td><td>/api/books/:isbn</td><td>Updates a book</td></tr>
<tr><td>Delete</td><td>/api/books/:isbn</td><td>Deletes a book</td></tr>
</table>

1. Test the Web API using Postman.
2. Create a **book-service.spec.js** file. Test the Web API using chai-http.

After you complete the lab, fill in the ***TestingDoc-Grading-Sheet.docx*** and save it inside **Assessment 6-Web API** folder. Sync your repository to push your work to Github.