Book Store

This is the basic application used to store books where we can make it simple application which includes

- initially 5 books are added when application initialize and each book have unique ISBN(International Standard Book Number) which is the Book id we choosen
- this simple application having below functionality
  ** Shows list of Available Books
  ** Add the new Book
  ** Update the Existing Book
  ** search the book by any name
  ** get the book by Passing ISBN book id

Getting Started

Just pull the repository, its a spring boot maven application so there is nothing required to install anything you need only IDE (Eclipse,Intellij,Netbeans)
Also Executable Jar is provided in repository you can directly run the jar by below command
java -jar Bookstore-0.0.1-SNAPSHOT.jar

Prerequisites

Only IDE required
Chrome Browser or Postman to trigger the rest api

Installing

- If running the jar nothing required only run the jar as given command above and which will trigger the application and deploy on tomcat on port 8080
- Running from IDE :
  Just pull the project in IDE and right click on project and Run the spring boot applilcation, it will automatically deploy the application

Run Steps

Once Jar run or Run from IDE open the chrome browser or Postman tool

URl : http://localhost:8080/books/

Show all books (5 books added while application initialization)
GET

http://localhost:8080/books/allbooks – (show all books)

add the new Books (pass below json body format)
POST

http://localhost:8080/books/add

request Body in Json format :

{

“bookCategory” : “Technology”,

“bookISBN_ID” : “123”,

“bookName” : “Java my Favourite”,

“description” : “Java Fundamental”

}

Update the Book (Id as parameter and json body to update the fields)

PUT

http://localhost:8080/books/update/{ISBN_ID}

e.g : http://localhost:8080/books/update/0134685997

```
request body to update :
 {
    "bookCategory": "History",
    "bookName": "The 1st  World War",
    "description": "History of 1st world war"
}
```

search for books (pass book name as input)

GET

it will search all books where name matching and show the book data

http://localhost:8080/books/search/{BookName}

e.g http://localhost:8080/books/search/java – (searching for ‘java’ here)

Junit Test

Junit test added for acceptance test few test added

- Test for : testaddBook – pass httpstatus.created

- Test for : testGetBook – pass – httpstatus.ok
- test for : testaddExistingBook – pass – httpstatus.conflicts

Coading Style

Book Class (Model)

Attributes

```
* ISBN_ID : Unique Book Id   -- String
* BookName : Name of Book   -- String
* Category : category of book like Technology,Fiction,Drama.. -- String
* Description : Information about the book   -- String
```

ISBN : ISBN(International Standard Book Number) taken this as Unique id to identify book because its easily available on Net for users to search for any books
this taken as a String because sometime ISBN is conatins Dash(-) between the numbers

BookStore Interface (Model)

```
   having below methods
   ** public Book AddBook(Book book);
   ** public Map<String, Book> getListOfAvalableBooks();
   ** public Book updateBook(String id, Book book);
```

BookStoreService (service class)

BookStoreController (Controller)

Unit Test

TestControllerMethods

```
** testaddBook()
** testaddExistingBook()
** testGetBook()
```

User Exception Handler

```
** BookStoreExceptionHandler class
** BookStoreHandlerAdvice class
```