

BOOKSTORE APPLICATION

Description:

- **Coding Language** : Java (version 11)
- **Software** : IntelliJ IDEA
- **Technology** : Spring Boot(version 2.7.1)
- **Dependencies:** Spring web, Spring Devtools, Validation, Lombok, My SQL driver, JPA, JWT, JSM, spring starter test.
- **Rest API used** : POST, GET, PUT, DELETE
- **Server** : Tomcat (version 9.0)
- **Third party server used** : POSTMAN

Summary:

For Bookstore Application we have to create User Registration APIs, Book APIs, Cart APIs and Order APIs. First in this project user need to register user details (ex. First name, last name, Address, Email Id, password) after registering, user will get the mail to his /her registered email Id. In Book APIs we need to add Books details (ex. Book Name, Author Name, Book description, Book Img, Price and Quantity). In Cart APIs we need to add Cart details (ex. Book Id, user ID and Quantity). In order APIs we need to add order details (ex. Quantity, Address, User ID, Book Id and Price).To create APIs we need to create seven different Layers that are **Controller, Dto, Model, Exception, Repository/Dao, Service and Util** for User Registration, Book, Cart and order.

USER REGISTRATION API

SECTION A – AUTHENTICATION STEPS

GENERAL INFORMATION

Use HTTP Basic Authentication (<http://localhost:5000/user>) for all the API calls.

Link for accessing the API:

[https://
url](https://url) (or)
[https://
ip:5000](https://ip:5000)

The url/ip is same as your dashboard's url/ip

EXAMPLE

(Get User API) <https://url>

CREDENTIALS

The API credentials can be found on </settings>

Username is "API Token"

Password is "API Key"

NOTE – In all the API calls if you want to get the Sub zones details using the top zones API Key and token you can post sub zones Use id.

SECTION B – ADD USER API

GENERAL INFORMATION POST Method (registration):

To add a user send a post request to </user/register> with the following data:

Key	Mandatory	Value
first Name	Yes	String
last Name	Yes	String
email	Yes	String
address	yes	String
password	yes	String

Sample Output

{

"message": "User Registered Successfully", "data":

"eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpZGF9rZXkiOiJlyfQ.vwWGxIYh8A
_RhQpMBGJ_HM9QLqS1Tsw62t6EcwB_1TQ"

}

Note: Mail will sent to registered Email Id with token.

SECTION C – Post user login (login)

GENERAL INFORMATION POST Method (login):

</user/login>

Input:

key	Mandatory	Value
email	yes	String(abc123@gmail.com)
password	yes	String(Set what you want as password it should not already taken)

Sample Output

```
{  
  "message": "login successful ", "data":  
  {  
    "userId": 22,  
    "firstName": "Akshay",  
    "lastName": "Kumar",  
    "email": "akshaysportive@gmail.com",  
    "address": "Bangalore",  
    "password": "9154601535"  
  }  
}
```

SECTION D – GET ALL USERS OF AN ACCOUNT API

GENERAL INFORMATION GET Method:

[/user/getAll](#)

Sample Output

```
{
  "message": "User retrieved successfully (:)",
  "data": [
    {
      "userId": 1,
      "firstName": "Akshay",
      "lastName": "kumar",
      "email": "akshaysportive@gmail.com",
      "address": "bangalore",
      "password": "560056"
    },
    {
      "userId": 2,
      "firstName": "Pavan",
      "lastName": "Kumar",
      "email": "pavan123@gmail.com",
      "address": "bangalore",
      "password": "560056"
    }
  ]
}
```

SECTION E – GET USER DETAILS by Token API

GENERAL INFORMATION

GET Method:

[/user/getAll/{token}](#)

Argument	Mandatory	Value
Token	yes	String

Sample Output

```

{"message": "User retrieved successfully (:)",

"data": {

    "userId": 1,
    "firstName": "Akshay",
    "lastName": "kumar",
    "email": "akshaysportive@gmail.com",
    "address": "bangalore",
    "password": "560056"
  }
}

```

SECTION E – GET USER DETAILS by Token API

GENERAL INFORMATION

GET Method:

[/user/update/{id}](#)

Argument	Mandatory	Value
Id	yes	String

Sample input [/user/update/2](#)

```

{
  "firstName": "Akshay",
  "lastName": "kumar",
  "email": "akshay@gmail.com",
  "address": "bangalore",
  "password": "aghhskk"
}

```

Data required to update:

Key(body)	Mandatory	Value
First Name	Yes	String
Last Name	Yes	String
Email	Yes	String
Address	Yes	String
password	Yes	String

Sample Output

```
{  
  "message": "User Record updated successfully (:",  
  "data": {  
    "userId": 1,  
    "firstName": "Akshay",  
    "lastName": "kumar",  
    "email": "akshay@gmail.com",  
    "address": "bangalore",  
    "password": " aghhskk "  
  }  
}
```

Book Store API

SECTION A

GENERAL INFORMATION

Use HTTP Basic Authentication (<http://localhost:5000/book>) for all the API calls.

Link for accessing the API:

[https://](https://url)
url (or
)
[https://](https://ip:5000)
[/ip:5000](https://ip:5000)
[0](https://ip:5000)

The url/ip is same as your dashboard`s url/ip

EXAMPLE

(Get Book API) <https://url>

NOTE – In all the API calls if you want to get the Sub zones details using the top zones API Key and token you can post sub zones Use id.

SECTION B – ADD Book API

GENERAL INFORMATION POST Method (Add Books):

To add a user send a post request to </book/insert> with the following data:

Key	Mandatory	Value
Book Name	Yes	String
Author Name	Yes	String
Book Description	Yes	String
Book image	Yes	String
Price	Yes	Integer
Quantity	Yes	Integer

Sample Output

```
{
  "message": "New Book Added in Book Store",
  "data": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpZGF9rZXkiOjM4fQ.sdjl1yh81OvfTaR1EocYTBctGW0WHvGInzrj2ENTDKsE"
}
```

Note: Mail will sent to registered Email Id with token.

SECTION C – GET All Books

GENERAL INFORMATION GET Method:

[/book/getAll](#)

Sample Output

```
{
  "message": "Books retrieved successfully (:)",
  "data": [
    {
      "bookId": 1,
      "bookName": "Rockettt",
      "authorName": "Kritin",
      "bookDescription": "read this",
      "bookImg": "Blue img",
      "price": 100,
      "quantity": 56
    },
    {
      "bookId": 2,
      "bookName": "Lost in space",
      "authorName": "Akshay",
      "bookDescription": "read this",
      "bookImg": "Blue",
      "price": 100,
      "quantity": 56
    }
  ]
}
```

SECTION D – GET Book by Id using Token

GENERAL INFORMATION GET Method:

[/book/getById/{token}](#)

Input:

Parameter	Mandatory	Value
------------------	------------------	--------------

Token	Yes	String
-------	-----	--------

Sample Output

```
{
  "message": "Data retrieved successfully by id (:)",
  "data": {
    "bookId": 38,
    "bookName": "Note",
    "authorName": "Pavan",
    "bookDescription": "Please read",
    "bookImg": "Blue color",
    "price": 100,
    "quantity": 15
  }
}
```

SECTION E – DELETE Book by Token API

GENERAL INFORMATION

DELETE Method:

[/book/delete/{token}](#)

Parameter	Mandatory	Value
Token	yes	String

Sample Output

```
{
  "message": "Book Record deleted successfully",
  "data": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpZm9rZXkiOiM4fQ.sdjlYh81OvfTaR1EocYTBctGW0WHvGInzrj2ENTDKsE"
}
```

SECTION F – UPDATE BOOK by Token API

GENERAL INFORMATION

PUT Method:

[/book/update/{token}](#)

Parameter	Mandatory	Value
token	yes	String

Sample input [/book/update/2](#)

```
{
  "bookName": "Hello",
  "authorName": "Akshay",
  "bookDescription": "Please read",
  "bookImg": "Blue color",
  "price": 100,
  "quantity": 15
}
```

Data required to update:

Key(body)	Mandatory	Value
Book Name	Yes	String
Author Name	Yes	String
Book Description	Yes	String
Book Img	Yes	String
Price	Yes	Integer
Quantity	Yes	Integer

Sample Output

```
{
  "message": " Book Record updated successfully by Id",
  "data": {
    "bookId": 40,
    "bookName": "Hello",
    "authorName": "Akshay",
    "bookDescription": "Please read",
    "bookImg": "Blue color",
    "price": 100,
    "quantity": 15
  }
}
```

Cart API

SECTION A

GENERAL INFORMATION

Use HTTP Basic Authentication (<http://localhost:5000/cart>) for all the API calls.

Link for accessing the API:

<https://>
[url](#) (or
)
<https://>

[/ip:5000](#)

The url/ip is same as your dashboard's url/ip

EXAMPLE

(Get cart API) <https://url>

NOTE – In all the API calls if you want to get the Sub zones details using the top zones API Key and token you can post sub zones cart id.

SECTION B – ADD cart API

GENERAL INFORMATION POST Method (Add Books):

To add a user send a post request to [/cart/create](#) with the following data:

Key	Mandatory	Value
User Id	Yes	String
Book id	Yes	String
Quantity	Yes	String

Sample Output

```
{
  "message": "cart Added successfully !",
  "data": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpZGF9rZXkiOiJQxfQ.T-yB2H8LsARc0uHDLsoN264gl37eBD5bgN4XfrER4-g"
}
```

Note: Mail will sent to registered Email Id with token.

SECTION C – GET ALL CART details by using Token

GENERAL INFORMATION GET Method:

[/cart/getAll/{token}](#)

Input:

<i>Parameter</i>	<i>Mandatory</i>	<i>Value</i>
Token	Yes	String

Sample Output

```
{
  "message": "Get call Success",
  "data": [
    {
      "cartId": 15,
      "user": {
        "userId": 1,
        "firstName": "Akshay",
        "lastName": "Kumar",
        "email": "akshaysportive@gmail.com",
        "address": "girinagar",
        "password": "kqtvikegkryzjixa"
      },
      "book": {
        "bookId": 3,
        "bookName": "Lost in space",
        "authorName": "Akshay",
        "bookDescription": "read this",
        "bookImg": "Blue",
        "price": 100,
        "quantity": 56
      },
      "quantity": 2
    },
    {
      "cartId": 16,
      "user": {
        "userId": 1,
        "firstName": "Akshay",
        "lastName": "Kumar",
        "email": "akshaysportive@gmail.com",
        "address": "girinagar",
        "password": "kqtvikegkryzjixa"
      },
      "book": {
        "bookId": 5,
        "bookName": "Fighting Robots",
```

```

        "authorName": "Kv Ram",
        "bookDescription": "read this",
        "bookImg": "Robots",
        "price": 100,
        "quantity": 16
    },
    "quantity": 2
},
{
    "cartId": 17,
    "user": {
        "userId": 1,
        "firstName": "Akshay",
        "lastName": "Kumar",
        "email": "akshaysportive@gmail.com",
        "address": "girinagar",
        "password": "kqtvikegkryzjixa"
    },
    "book": {
        "bookId": 5,
        "bookName": "Fighting Robots",
        "authorName": "Kv Ram",
        "bookDescription": "read this",
        "bookImg": "Robots",
        "price": 100,
        "quantity": 16
    },
    "quantity": 2
}
}
}

```

SECTION D – GET Cart by Id using Token

GENERAL INFORMATION GET Method:

[/cart/getById/{token}](#)

Input:

<i>Parameter</i>	<i>Mandatory</i>	<i>Value</i>
Token	Yes	String

Sample Output

```
{
  "message": "Cart details retrieved successfully",
  "data": {
    "cartId": 41,
    "user": {
      "userId": 1,
      "firstName": "Akshay",
      "lastName": "Kumar",
      "email": "akshaysportive@gmail.com",
      "address": "girinagar",
      "password": "kqtvikegkryzjixa"
    },
    "book": {
      "bookId": 2,
      "bookName": "Rockettt",
      "authorName": "Kritin",
      "bookDescription": "read this",
      "bookImg": "Blue img",
      "price": 100,
      "quantity": 56
    },
    "quantity": 8
  }
}
```

SECTION E – DELETE Cart by Token API

GENERAL INFORMATION

DELETE Method:

[/cart/delete/{token}](#)

<i>Parameter</i>	<i>Mandatory</i>	<i>Value</i>
Token	yes	String

Sample Output

```
{
  "message": "Cart delete successfully",
  "data": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpZm9rZXkiOiJQxfQ.T-yB2H8LsARc0uHDLsoN264gl37eBD5bgN4XfrER4-g"
}
```

SECTION F – UPDATE Cart by Token API

GENERAL INFORMATION

PUT Method:

[/cart/updateById/{token}](#)

Parameter	Mandatory	Value
token	yes	String

Sample input [/cart/updateById/{token}](#)

```
{
  "userId": "1",
  "bookId": "3",
  "quantity": "8"
}
```

Data required to update:

Key(body)	Mandatory	Value
User Id	Yes	String
Book Id	Yes	String
Quantity	Yes	Integer

Sample Output


```
{
  "message": " Cart Record updated successfully by Id",
  "data": {
    "cartId": 41,
    "user": {
      "userId": 1,
      "firstName": "Akshay",
      "lastName": "Kumar",
      "email": "akshaysportive@gmail.com",
      "address": "girinagar",
      "password": "kqtvikegkryzjixa"
    },
    "book": {
      "bookId": 3,
      "bookName": "Lost in space",
      "authorName": "Akshay",
      "bookDescription": "read this",
      "bookImg": "Blue",
      "price": 100,
      "quantity": 56
    },
    "quantity": 8
  }
}
```

Order API

SECTION A

GENERAL INFORMATION

Use HTTP Basic Authentication (<http://localhost:5000/order>) for all the API calls.

Link for accessing the API:

[https://](https://url)
url (or
)
[https://](https://ip:5000)
[/ip:500](https://ip:5000)
[0](https://ip:5000)

The url/ip is same as your dashboard's url/ip

EXAMPLE

(Get cart API) <https://url>

NOTE – In all the API calls if you want to get the Sub zones details using the top zones API Key and token you can post sub zones cart id.

SECTION B – ADD Order API

GENERAL INFORMATION POST Method (Insert Order):

To add a user send a post request to [/order/insert](#) with the following data:

Key	Mandatory	Value
Quantity	Yes	Integer
Address	Yes	String
User Id	Yes	Integer
Book Id	Yes	Integer
Price	Yes	Integer

Sample Output

```
{"message": "Order placed successfully !",  
  
  "data":  
  "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpZm9rZXkiOiJzfQ.qQSe10r  
  DRPptQdAsPZJOe85Bn2V1LfjLQ2c1toMGUQg" }
```

Note: Mail will sent to registered Email Id with token.

SECTION C – GET All Order details by using Token

GENERAL INFORMATION GET Method:

[/order/getAllOrders/{token}](#)

Input:

Parameter	Mandatory	Value
Token	Yes	String

Sample Output

```
{
  "message": "All records retrieved successfully !",
  "data": [
    {
      "orderId": 23,
      "date": "2022-07-21",
      "price": 500,
      "quantity": 1,
      "address": "Bangaloor",
      "user": {
        "userId": 4,
        "firstName": "Vishaka",
        "lastName": "Kadem",
        "email": "vishaka@gmail.com",
        "address": "Hyadrabad",
        "password": "9999999999"
      },
      "book": {
```

```
"bookId": 19,
"bookName": "First",
"authorName": "Sameer",
"bookDescription": "1",
"bookImg": "2",
"price": 500,
"quantity": 0
},
"cancel": true
},
{
  "orderId": 24,
  "date": "2022-07-21",
  "price": 500,
  "quantity": 1,
  "address": "Hydrabad",
  "user": {
    "userId": 5,
    "firstName": "Sam",
    "lastName": "syed",
    "email": null,
    "address": "HNo:22-313,SSR Nagar,Nandikotkur",
    "password": "830967"
  },
  "book": {
    "bookId": 19,
    "bookName": "First",
    "authorName": "Sameer",
```

```
    "bookDescription": "1",
    "bookImg": "2",
    "price": 500,
    "quantity": 0
  },
  "cancel": true
},
{
  "orderId": 27,
  "date": "2022-07-21",
  "price": 600,
  "quantity": 1,
  "address": "Hydrabad",
  "user": {
    "userId": 6,
    "firstName": "Sam",
    "lastName": "syed",
    "email": "sam@gmail.com",
    "address": "HNo:22-313,SSR Nagar,Nandikotkur",
    "password": "830967"
  },
  "book": {
    "bookId": 25,
    "bookName": "Second",
    "authorName": "Akshay",
    "bookDescription": "6",
    "bookImg": "5",
    "price": 600,
```

```

        "quantity": 2
      },
      "cancel": true
    }
  ]
}

```

SECTION D – GET Order by Id using Token

GENERAL INFORMATION GET Method:

[/order/getById/{token}](#)

Input:

<i>Parameter</i>	<i>Mandatory</i>	<i>Value</i>
Token	Yes	String

Sample Output

```

{
  "message": "Get record by Id retrieved successfully !",
  "data": {
    "orderId": 23,
    "date": "2022-07-21",
    "price": 500,
    "quantity": 1,
    "address": "Bangaloor",
    "user": {
      "userId": 4,
      "firstName": "Vishaka",
      "lastName": "Kadem",
      "email": "vishaka@gmail.com",
      "address": "Hyadrabad",
      "password": "9999999999"
    }
  },

```

```

"book": {
  "bookId": 19,
  "bookName": "First",
  "authorName": "Sameer",
  "bookDescription": "1",
  "bookImg": "2",
  "price": 500,
  "quantity": 0
},
"cancel": true
}

```

SECTION E – Cancel Order by Token API

GENERAL INFORMATION

PUT Method:

[/order/cancelOrder/{token}/{userId}](#)

<i>Parameter</i>	<i>Mandatory</i>	<i>Value</i>
token	yes	String
use Id	Yes	Integer

Sample Output {

"message": "Cancel order successfully !

"data": {

 "orderId": 23,

 "date": "2022-07-21",

 "price": 500,

 "quantity": 1,

 "address": "Bangalore",

 "user": {

```
"userId": 4,  
"firstName": "Vishaka",  
"lastName": "Kadem",  
"email": "vishaka@gmail.com",  
"address": "Hyadrabad",  
"password": "9999999999"  
},  
"book": {  
  "bookId": 19,  
  "bookName": "First",  
  "authorName": "Sameer",  
  "bookDescription": "1",  
  "bookImg": "2",  
  "price": 500,  
  "quantity": 0  
},  
"cancel": true  
}
```