

User Manual

Book Management System

Author: Tim Mailman

Date: December 5th, 2021

Version: 1

Book Management System

Table of Contents

1.	Project Overview	1
1.1.	Scope	1
1.2.	Benefits and Values	1
1.3.	Platform Requirements	1
2.	Use Cases.....	2
2.1.	Run the Application.....	2
2.2.	Create New Book	4
2.3.	View Entered Books.....	5
2.4.	Use Case Update Book Details.....	6
2.5.	Deleting a Book	7
2.6.	Administration Delete by ID	8
2.7.	Administration Delete All Books	10
2.8.	Administration Post New Book.....	11
2.9.	Administration Update Book.....	12
2.10.	Administration Get All Books.....	13
2.11.	Administration Getting One Book By ID.....	14

1. Project Overview

The Book Management System is a web service application that allows a user to keep track of books in a database.

1.1. Scope

The user is able to create, read, update and delete books from a database. The user can view fields such as title, author, inventory, and price. The application is built using Java and Spring Boot.

1.2. Benefits and Values

The application allows a retail store or individual to keep inventory of their books. Some benefits of this application include:

- prevent the user from ordering too many books
- view inventory to know when to order more books
- relate fields to one another (see what author wrote what book)
- set and update the price of a book
- remove a book from the database

1.3. Platform Requirements

The application in its current condition requires a user to run it as a java application. Either using a Java Development Kit, or a Java Runtime Environment. The user must also be able to connect and browse the network that the application connects to.

The application also has some administrative features that require the use of a RESTful application such as Postman to make specific calls to the database.

Book Management System

2. Use Cases

What follows is a description of the use-case view of the software architecture. The images that follow are a visual representation of a user of the Book Management System performing its functions.

2.1. Run the Application

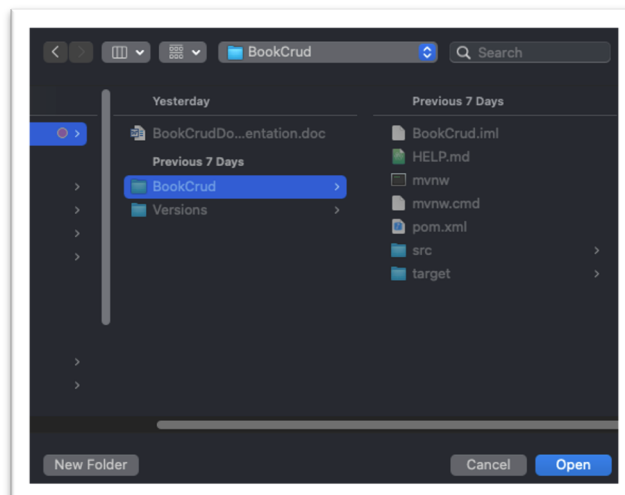
A user performing the necessary steps to run and view the application

Preconditions	Platform requirements met
Steps	<ol style="list-style-type: none">1. Open runtime/development application2. With runtime/development application, open 'BookCrud' application3. Run BookCrudApplication.java4. Open browser5. Enter URL 'localhost:8080/' in address bar
Postconditions	Application running

Open runtime/development application

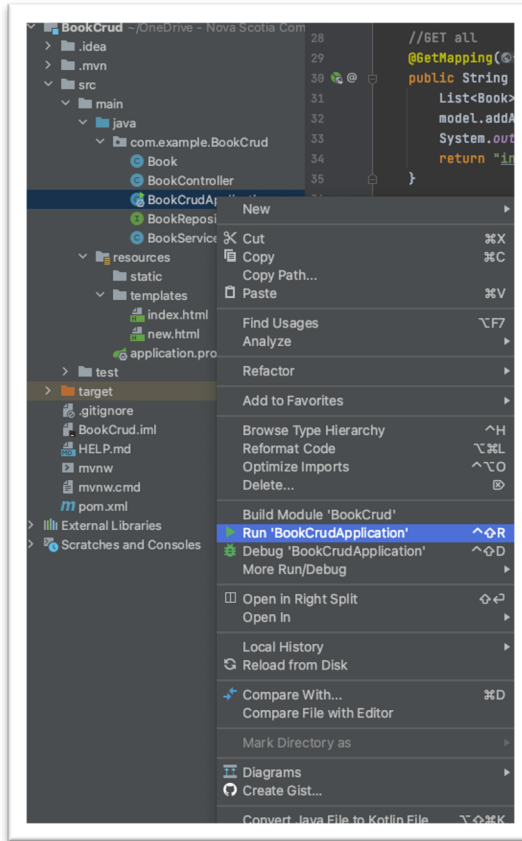


With runtime/development application, open 'BookCrud' application

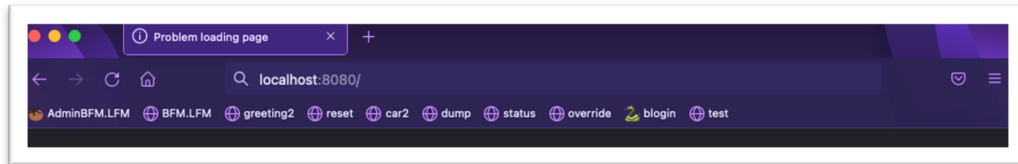


Book Management System

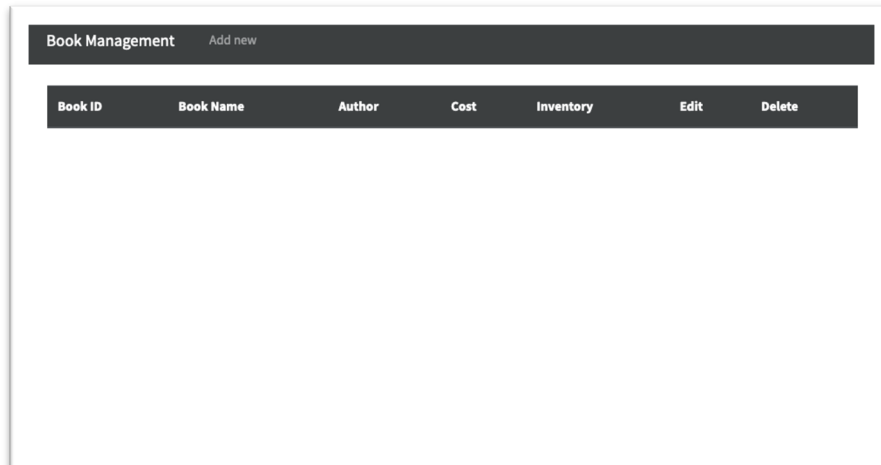
Run BookCrudApplication.java



Open browser



Enter URL 'localhost:8080/' in address bar



Book Management System

2.2. Create New Book

User adds a new book to the application

Preconditions	App Running
Steps	<ol style="list-style-type: none">1. User clicks “Add new”2. User enters respected data into the fields3. User clicks save
Postconditions	Book entered into app

User clicks “Add new”

The screenshot shows the 'Book Management' application interface. At the top, there is a dark header bar with 'Book Management' on the left and 'Add new' on the right. Below the header, there is a table with the following columns: Book ID, Book Name, Author, Cost, Inventory, Edit, and Delete. The table is currently empty.

User enters respected data into the fields

The screenshot shows the 'Create New Book' form in the 'Book Management' application. The header bar shows 'Book Management' and 'Home'. The form title is 'Create New Book'. The form contains the following fields:

- Book Name: Book Titles: How to Name Books
- Author: Tim Mailman
- Price: 89.99
- Inventory: 5

At the bottom of the form, there is a green 'Save' button.

Book Management System

User clicks save

Book Management Add new						
Book ID	Book Name	Author	Cost	Inventory	Edit	Delete
1	Book Titles: How to Name Books	Tim Mailman	89.99	5	Edit	Delete

2.3. View Entered Books

User viewing the entered books. This is dependent on the database containing at least 1 book entry. Otherwise, no books will show.

Preconditions	At least 1 book entered
Steps	<ol style="list-style-type: none">1. Run the application2. View fields
Postconditions	Data entered into program

View fields

Book Management Add new						
Book ID	Book Name	Author	Cost	Inventory	Edit	Delete
1	Book Titles: How to Name Books	Tim Mailman	89.99	5	Edit	Delete
2	Book Titles 2: How Not To Name Books	Tim Mailman	89.99	8	Edit	Delete
3	On Eating Catfood	Fernie B. Cat	1.0	1	Edit	Delete

Book Management System

2.4. Use Case Update Book Details

The user can select a book to update. The user may update the book name, author, cost, or inventory of the book.

Use Case Name	Update book
Preconditions	At least 1 book entered
Steps	<ol style="list-style-type: none">1. User clicks on 'Edit' for the book row they wish to edit2. Updates any/all fields they wish to update3. User clicks the 'Save' button
Postconditions	User updated a book with new information

User clicks on 'Edit' for the book row they wish to edit

Book Management

Add new

Book ID	Book Name	Author	Cost	Inventory	Edit	Delete
1	Book Titles: How to Name Books	Tim Mailman	89.99	5	Edit	Delete
2	Book Titles 2: How Not To Name Books	Tim Mailman	89.99	8	Edit	Delete
3	On Eating Catfood	Fernie B. Cat	1.0	1	Edit	Delete

update any/all fields they wish to update and clicks the green 'Save' button

Book Management

Home

Create New Book

Book Name

Book Titles 2: How Not To Name Books

Author

Tim Mailman

Price

59.99

Inventory

17

Save

Book Management System

Row 2 now updated with a new price and inventory count

Book Management Add new						
Book ID	Book Name	Author	Cost	Inventory	Edit	Delete
1	Book Titles: How to Name Books	Tim Mailman	89.99	5	Edit	Delete
2	Book Titles 2: How Not To Name Books	Tim Mailman	59.99	17	Edit	Delete
3	On Eating Catfood	Fernie B. Cat	1.0	1	Edit	Delete

2.5. Deleting a Book

The user chooses a book to remove from the list.

Preconditions	<ul style="list-style-type: none">At least 1 book entered
Steps	1. User clicks 'Delete' of the book they wish to remove
Postconditions	Book no longer exists and is not in the list

User clicks 'Delete' of the book they wish to remove

Book Management Add new						
Book ID	Book Name	Author	Cost	Inventory	Edit	Delete
1	Book Titles: How to Name Books	Tim Mailman	89.99	5	Edit	Delete
2	Book Titles 2: How Not To Name Books	Tim Mailman	59.99	17	Edit	Delete
3	On Eating Catfood	Fernie B. Cat	1.0	1	Edit	Delete

Book Management System

Book no longer exists and is not in the list

Book Management Add new						
Book ID	Book Name	Author	Cost	Inventory	Edit	Delete
1	Book Titles: How to Name Books	Tim Mailman	89.99	5	Edit	Delete
3	On Eating Catfood	Fernie B. Cat	1.0	1	Edit	Delete

2.6. Administration Delete by ID

Administrator deleting a book record with a DELETE REST request.

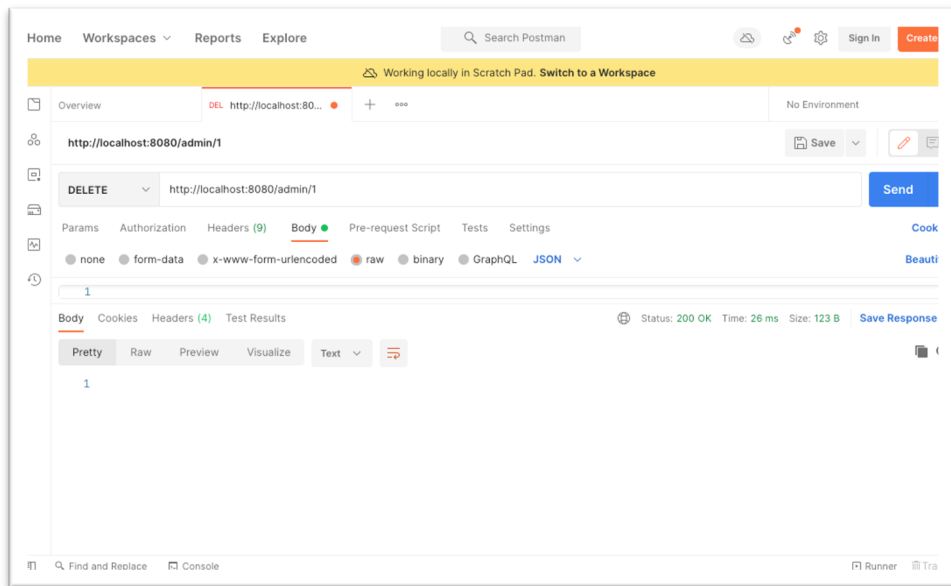
Preconditions	At least 1 book record
Steps	<ol style="list-style-type: none">Admin enters ID of book to be deleted in URL request (e.g. localhost:8080/admin/1)Sends DELETE request
Postconditions	Book deleted by ID

Admin decides which book to delete

Book Management Add new						
Book ID	Book Name	Author	Cost	Inventory	Edit	Delete
1	Book Titles: How to Name Books	Tim Mailman	89.99	5	Edit	Delete
2	Book Titles 2: How Not To Name Books	Tim Mailman	59.99	17	Edit	Delete
3	On Eating Catfood	Fernie B. Cat	1.0	1	Edit	Delete

Book Management System

Admin enters ID of book to be deleted in URL request and sends 'DELETE' request



Record now deleted

Book Management Add new						
Book ID	Book Name	Author	Cost	Inventory	Edit	Delete
2	Book Titles 2: How Not To Name Books	Tim Mailman	59.0	17	Edit	Delete
3	On Eating Catfood	Fernie B. Cat	1.0	1	Edit	Delete

Book Management System

2.7. Administration Delete All Books

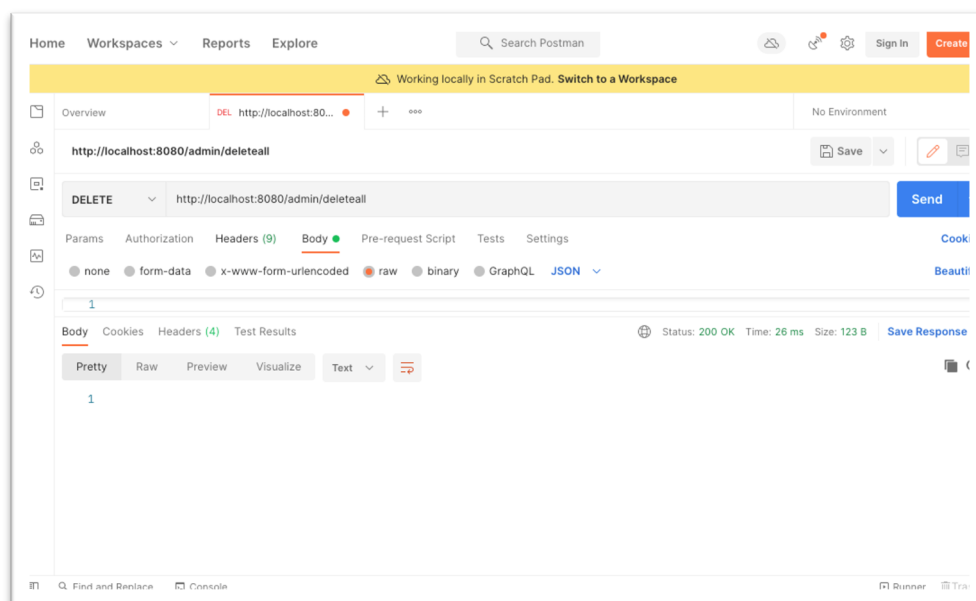
Admin deleting all book records with a DELETE REST request

Preconditions	At least 1 book record
Steps	1. Admin makes a DELETE URL request on the URL map 'deleteall' (e.g. localhost:8080/admin/deleteall)
Postconditions	All book records are deleted

Book Database contains some books

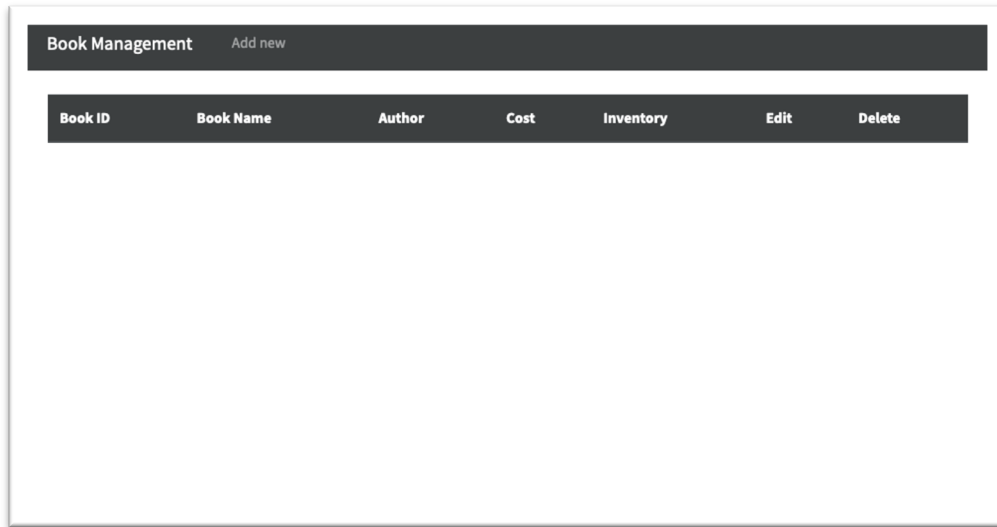
Book Management Add new						
Book ID	Book Name	Author	Cost	Inventory	Edit	Delete
2	Book Titles 2: How Not To Name Books	Tim Mailman	59.0	17	Edit	Delete
3	On Eating Catfood	Fernie B. Cat	1.0	1	Edit	Delete

DELETE request to '/admin/deleteall' map



Book Management System

All books removed

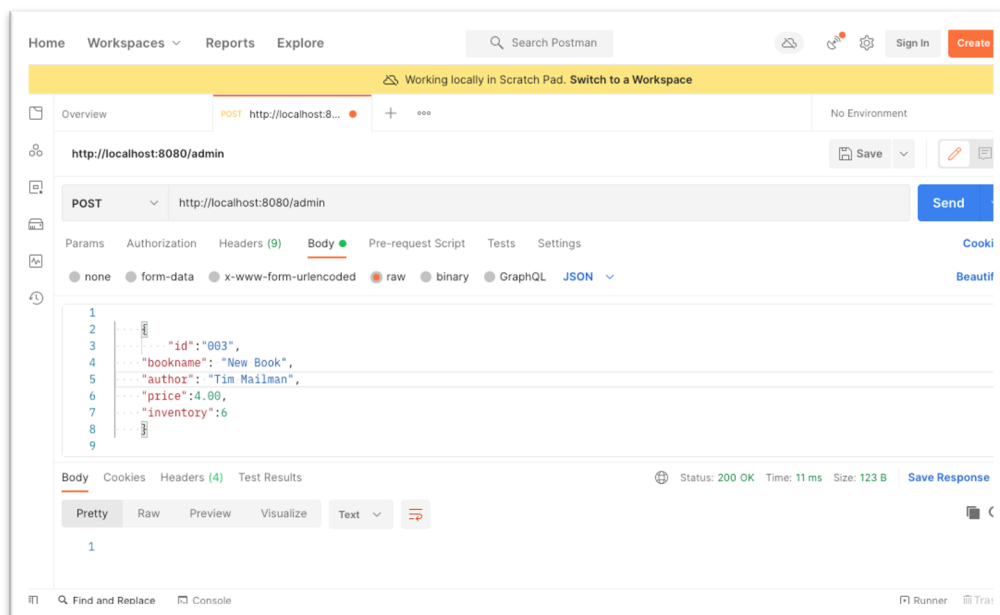


2.8. Administration Post New Book

Admin adding a new book with a REST request

Preconditions	Application running
Steps	<ol style="list-style-type: none">1. Admin writes in REST controller body of book details2. Sends POST request
Postconditions	New book entered

Admin writes in REST controller body of book details



Book Management System

New Book entered

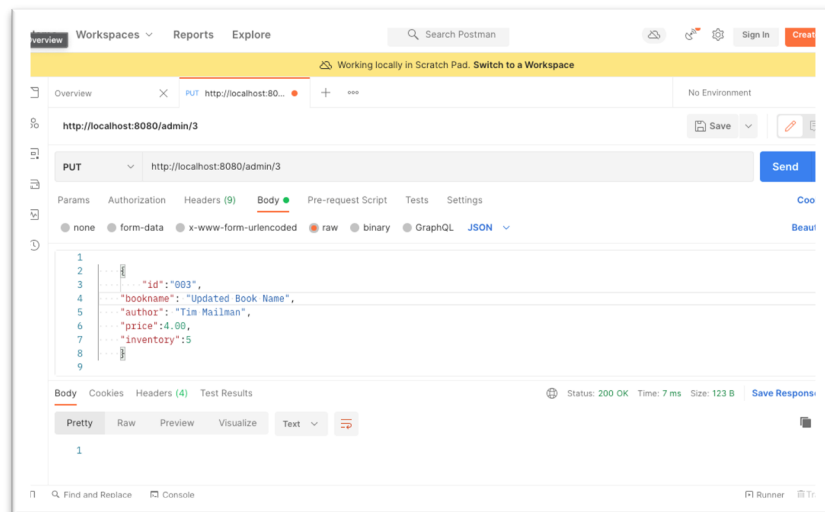
Book Management Add new						
Book ID	Book Name	Author	Cost	Inventory	Edit	Delete
3	New Book	Tim Mailman	4.0	6	Edit	Delete

2.9. Administration Update Book

Admin adding a new book with a REST request

Preconditions	At least 1 book entered
Steps	<ol style="list-style-type: none">Admin writes in REST controller body of book detailsSends PUT request
Postconditions	Book updated

Admin writes in REST controller body of book details



Book Management System

Updated result

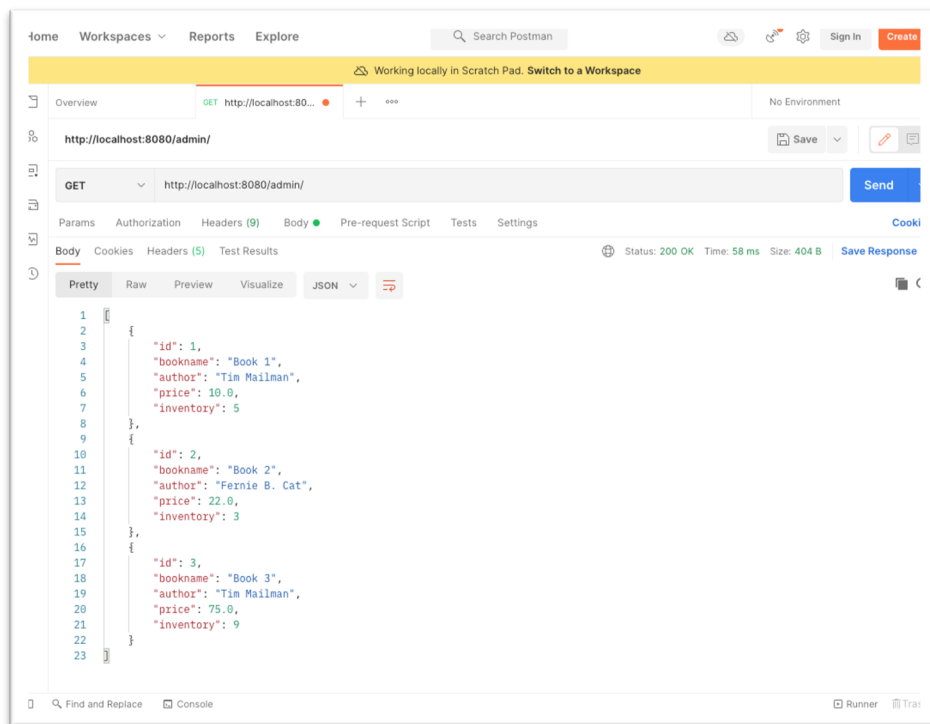
Book Management Add new						
Book ID	Book Name	Author	Cost	Inventory	Edit	Delete
3	Updated Book Name	Tim Mailman	4.0	5	Edit	Delete

2.10. Administration Get All Books

Admin adding a new book with a REST request

Preconditions	At least 1 book entered
Steps	1. Admin sends GET request to URL(e.g. localhost:8080/admin)
Postconditions	All books in database displayed

Administrator getting all content in database



Book Management System

2.11. Administration Getting One Book By ID

Admin adding a new book with a REST request

Preconditions	At least 1 book entered
Steps	1. Admin sends GET request to URL(e.g. localhost:8080/admin/2)
Postconditions	Book with specified ID is displayed

Administrator getting a Book with the ID of '2'

