# **Spring Boot CRUD Application**

**Description: This project created in the aim of to provide online books service provider with spring boot and restful API to represent all CRUD operation.**

# Pre-Requirements:

1. Java 1.8 or Higher Version
2. STS Tool (Spring Tool Suite)
3. MySQL Data base
4. MySQL Workbench
5. Post man (Test tool)

# Spring Boot Project Creation:

We can create Spring Boot application using Spring initializer <https://start.spring.io/>

Steps:

1. Select from below fields from the spring initializer.

* Project: Maven
* Language: Java
* Spring Boot: 3.4.3 or latest version
* Project Metadata: As per our project related naming conventions should provide details like group, artifact, name etc.
* Packaging: jar with java 17
* Dependencies: Add the below dependency starters
* spring-boot-starter-web,
* spring-boot-starter-security,
* spring-boot-starter-data-jpa
* mysql-connector-j

1. After clicking Generate then download the zip folder and extract and import as maven existing project into STS workspace.

# Create package inside project explore

**Controller package:** To implement all crud operations like POST, GET, PUT, DELETE etc.

**Service package:** To implement business logic

**Dao package:** To implement data base logic.

**Entity package:** To represent the whole request and response object

**Repository package:** To make connection between spring boot and MySQL data base

# Spring Security implantation:

Spring security is the default basic authentication for this we need to provide the spring security starter dependency as mentioned below.

Step 1:

A screenshot of a computer

AI-generated content may be incorrect.

Step 2: By default, authentication credentials as username: user and password: < It will automatically generate once we start the application need to copy from STS console>.

Step: If we want to our customized credentials, we need to change the application.properties file with below details.

spring.security.user.name=\*\*\*\*

spring.security.user.password=\*\*\*

# POST MAN Testing

Using POST man tool we can test all our crud operators like POST, GET, PUT and DELETE

1. **POST OPERATION** (@PostMapping("books"))

**End Point URL:** <http://localhost:8080/bookservice/books>

**Request Payload/Request Body:** Json format with post method

A screenshot of a computer

AI-generated content may be incorrect.

**Response Payload:** In Json format with 200 success code

A screenshot of a computer

AI-generated content may be incorrect.

1. **GET OPERATION (**@GetMapping("books"))

**Endpoint URL:** <http://localhost:8080/bookservice/books> (To fetch all books)

**Request Payload/Request Body:** NA

**A screenshot of a computer

AI-generated content may be incorrect.**

**Response Payload:** Json Format with list of books

**A screenshot of a computer

AI-generated content may be incorrect.**

* 1. **GET OPERATION By Id (**@GetMapping("books/{id}"))

**Endpoint URL:** <http://localhost:8080/bookservice/books/353> (To fetch single books with book Id)

**Request Payload/Request Body:** NA

**A screenshot of a computer

AI-generated content may be incorrect.**

**Response Payload:** Joson response payload with single book details with 200 status code

****

1. **PUT OPERATION (**@PutMapping("books/{id}"))

**Endpoint URL:** [**http://localhost:8080/bookservice/books/353**](http://localhost:8080/bookservice/books/353)It is used to update the existing resource or object with full or partial data.

**Request Payload/Request Body:** Json format with changed properties

**A screenshot of a computer

AI-generated content may be incorrect.**

**Response Payload:**

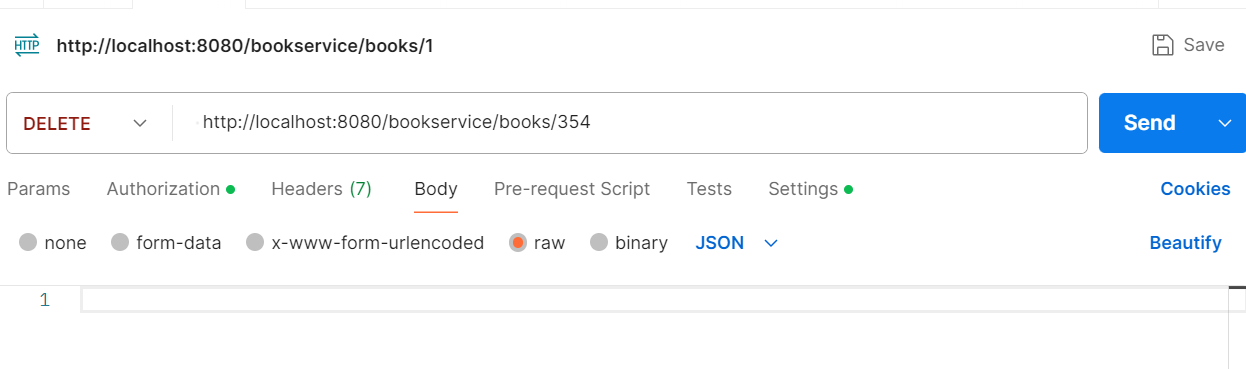
**A screenshot of a computer

AI-generated content may be incorrect.**

1. **DELETE OPERATION** (@DeleteMapping("books/{id}")**)**

**Endpoint URL:** http://localhost:8080/bookservice/books/354It is used to delete the existing resource or object with full or partial data.

**Request Payload/Request Body:** Json format with changed properties

****

**Response Payload:**

**A screenshot of a computer

AI-generated content may be incorrect.**