11\_RestFulAPI\_Assignment3

Code Snippet:-

Book.java:

**package** com.training;

**import** java.time.LocalDate;

**public** **class** Book {

String bookId;

String title;

**double** price;

Integer volume;

LocalDate publishDate;

@Override

**public** String toString() {

**return** "Book [bookId=" + bookId + ", title=" + title + ", price="

+ price + ", volume=" + volume + ", publishDate=" + publishDate

+ "]";

}

**public** Book(String bookId, String title, **double** price, Integer volume,

LocalDate publishDate) {

**super**();

**this**.bookId = bookId;

**this**.title = title;

**this**.price = price;

**this**.volume = volume;

**this**.publishDate = publishDate;

}

**public** Book(){

}

**public** String getBookId() {

**return** bookId;

}

**public** **void** setBookId(String bookId) {

**this**.bookId = bookId;

}

**public** String getTitle() {

**return** title;

}

**public** **void** setTitle(String title) {

**this**.title = title;

}

**public** **double** getPrice() {

**return** price;

}

**public** **void** setPrice(**double** price) {

**this**.price = price;

}

**public** Integer getVolume() {

**return** volume;

}

**public** **void** setVolume(Integer volume) {

**this**.volume = volume;

}

**public** LocalDate getPublishDate() {

**return** publishDate;

}

**public** **void** setPublishDate(LocalDate publishDate) {

**this**.publishDate = publishDate;

}

}

**BookDao.java:**

**package** com.training;

**import** java.time.LocalDate;

**public** **class** Book {

String bookId;

String title;

**double** price;

Integer volume;

LocalDate publishDate;

@Override

**public** String toString() {

**return** "Book [bookId=" + bookId + ", title=" + title + ", price="

+ price + ", volume=" + volume + ", publishDate=" + publishDate

+ "]";

}

**public** Book(String bookId, String title, **double** price, Integer volume,

LocalDate publishDate) {

**super**();

**this**.bookId = bookId;

**this**.title = title;

**this**.price = price;

**this**.volume = volume;

**this**.publishDate = publishDate;

}

**public** Book(){

}

**public** String getBookId() {

**return** bookId;

}

**public** **void** setBookId(String bookId) {

**this**.bookId = bookId;

}

**public** String getTitle() {

**return** title;

}

**public** **void** setTitle(String title) {

**this**.title = title;

}

**public** **double** getPrice() {

**return** price;

}

**public** **void** setPrice(**double** price) {

**this**.price = price;

}

**public** Integer getVolume() {

**return** volume;

}

**public** **void** setVolume(Integer volume) {

**this**.volume = volume;

}

**public** LocalDate getPublishDate() {

**return** publishDate;

}

**public** **void** setPublishDate(LocalDate publishDate) {

**this**.publishDate = publishDate;

}

}

**Controller class:**

**PostController.java**

package com.training;

import java.util.List;

import javax.validation.Valid;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.data.domain.Pageable;

import org.springframework.http.MediaType;

import org.springframework.web.bind.annotation.GetMapping;

import org.springframework.web.bind.annotation.PathVariable;

import org.springframework.web.bind.annotation.PostMapping;

import org.springframework.web.bind.annotation.RequestBody;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RequestMethod;

import org.springframework.web.bind.annotation.ResponseBody;

import org.springframework.web.bind.annotation.RestController;

@RestController

public class PostController {

@Autowired

private BookDao bookDAO;

@GetMapping("/welcome")

public String welcomeMe(Pageable pageable) {

return "Welcome1";

}

@RequestMapping(value = "/books", //

method = RequestMethod.GET, //

produces = { MediaType.APPLICATION\_JSON\_VALUE, //

MediaType.APPLICATION\_XML\_VALUE })

@ResponseBody

public List<Book> getBooks() {

System.out.println("................................................getBooks.........");

List<Book> list = bookDAO.getBooks();

return list;

}

@RequestMapping(value = "/book/{bookNo}", //

method = RequestMethod.GET, //

produces = { MediaType.APPLICATION\_JSON\_VALUE, //

MediaType.APPLICATION\_XML\_VALUE })

@ResponseBody

public Book getBook(@PathVariable("bookNo") String bookNo) {

System.out.println("bookNo........."+bookNo);

return bookDAO.getBook(bookNo);

}

@RequestMapping(value = "/addbook", //

method = RequestMethod.POST, //

produces = { MediaType.APPLICATION\_JSON\_VALUE, //

MediaType.APPLICATION\_XML\_VALUE })

@ResponseBody

public Book addBook(@RequestBody Book book) {

System.out.println("(Service Side) Creating Book: " + book.getBookId());

return bookDAO.addBook(book);

}

@RequestMapping(value = "/deletebook/{bookNo}", //

method = RequestMethod.DELETE, //

produces = { MediaType.APPLICATION\_JSON\_VALUE, MediaType.APPLICATION\_XML\_VALUE })

@ResponseBody

public void deleteBook(@PathVariable("bookNo") String bookNo) {

System.out.println("(Service Side) Deleting book: " + bookNo);

bookDAO.deleteBook(bookNo);

}

}

Test class:

package com.training;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class SpringCrudApplication {

public static void main(String[] args) {

SpringApplication.run(SpringCrudApplication.class, args);

}

}

Results:

Pom.xml:

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<project xmlns=*"http://maven.apache.org/POM/4.0.0"* xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*

xsi:schemaLocation=*"http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd"*>

<modelVersion>4.0.0</modelVersion>

<groupId>com.example</groupId>

<artifactId>hibernate-one-to-many-demo</artifactId>

<version>0.0.1-SNAPSHOT</version>

<packaging>jar</packaging>

<name>hibernate-one-to-many-demo</name>

<description>Demo project for Spring Boot</description>

<parent>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-parent</artifactId>

<version>2.0.0.RELEASE</version>

<relativePath/> <!-- lookup parent from repository -->

</parent>

<properties>

<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

<project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>

<java.version>1.8</java.version>

</properties>

<dependencies>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-data-jpa</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-web</artifactId>

</dependency>

<dependency>

<groupId>mysql</groupId>

<artifactId>mysql-connector-java</artifactId>

<scope>runtime</scope>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-test</artifactId>

<scope>test</scope>

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-maven-plugin</artifactId>

</plugin>

</plugins>

</build>

</project>

**Angular Client:**

<!DOCTYPE html>

<html>

<head>

</head>

<body>

<div ng-app=*"myapps"*>

<div ng-controller=*"PostController"*>

<table>

<tr>

<th>BookId</th>

<th>Price</th>

<th>volume</th>

<th>subtitle</th>

<th>publishdate</th>

</tr>

<tr ng-repeat=*"pr in listBooks"*>

<td>{{pr.bookId}}</td>

<td>{{pr.price}}</td>

<td>{{pr.volume}}</td>

<td>{{pr.subtitle}}</td>

<td>{{pr.publishDate}}</td>

</tr>

</table>

</div>

</div>

<script type=*"text/javascript"*

src=*"https://ajax.googleapis.com/ajax/libs/angularjs/1.4.7/angular.min.js"*></script>

<script type=*"text/javascript"*>

**var** myapp = angular.module('myapps', []);

myapp.controller('postController', **function**($scope, $http) {

$http.get('http://localhost:8080/books')

.success(**function**(response) {

$scope.listBooks = response.book;

});

});

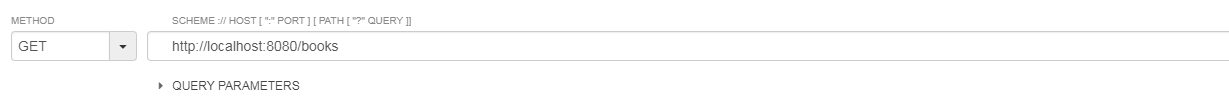
</script>

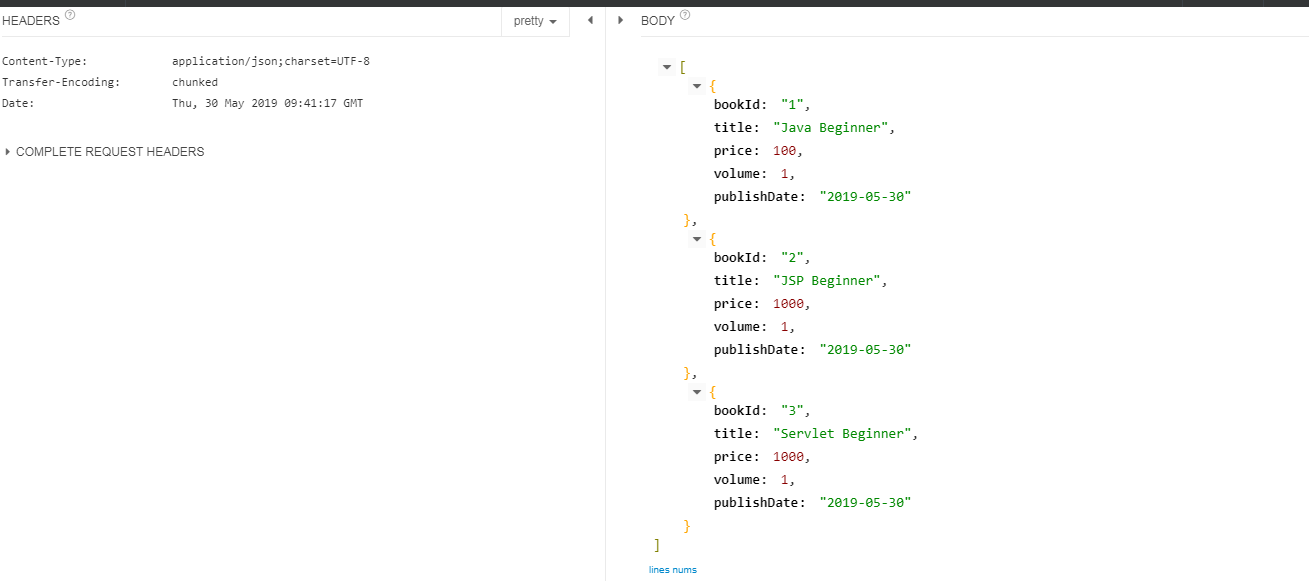
</body>

</html>

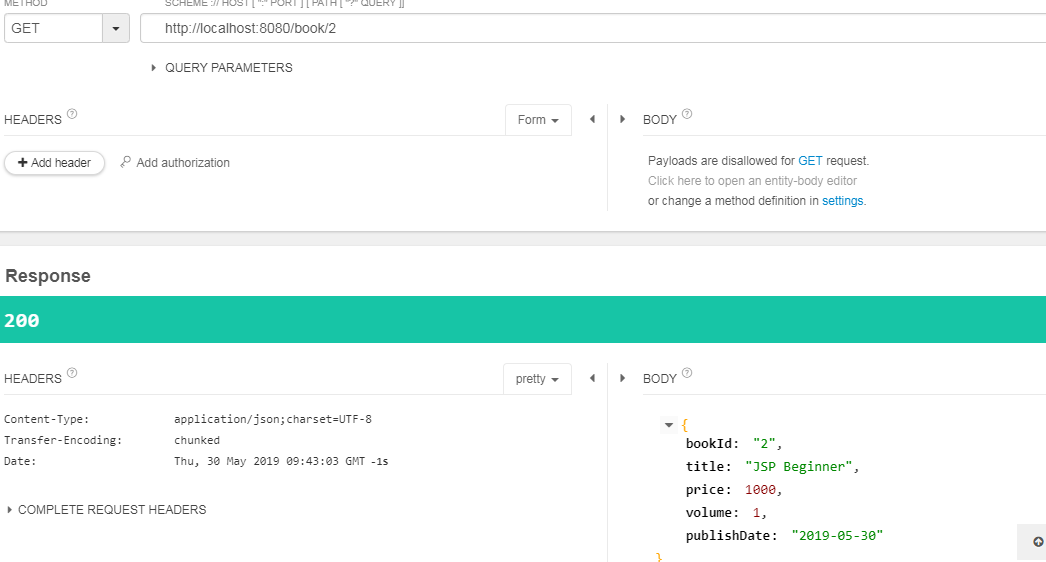
**Results:**

Get all books:

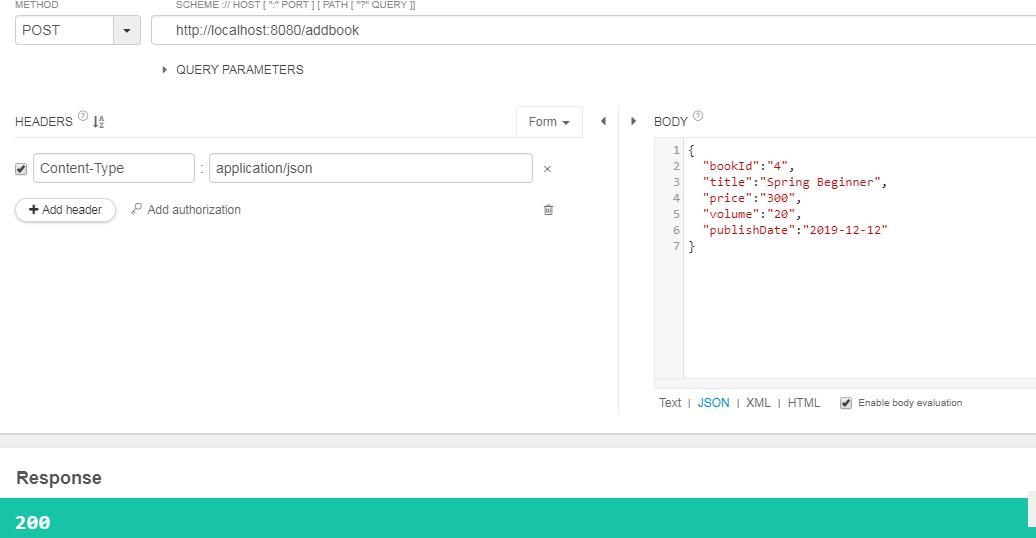




Get all books by id:



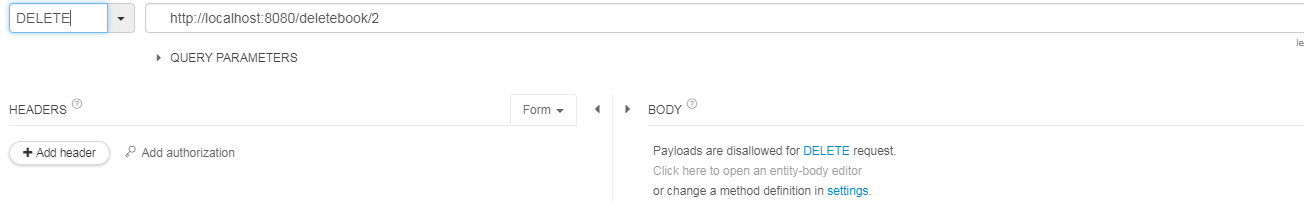
Post a book:



Get book after post:



Delete a book:



After delete:

