Name:Nacheeket Balkrishna Pednekar

Link:

Aim of the assignment is to create a demo REST API to get information about books. The information retrieved from the REST API call should be displayed using a standalone HTML page

Following are high level tasks for the assignment

- Create a database for books with author information. Draw ER diagram for the same.

- Create one REST API to retrieve book information, you can use any programming

language to create demo/debug server for this REST API like Java or Python.

- Create static web page to get information from REST API and display it on the page. This

page should be an HTML code along with CSS and JavaScript. This page should be

standalone and directly opened as an html file. Should not be served through any

particular server.

- Project for REST API:

Creating a Rest API using spring boot

Technology use Java and for database mysql

1. **Create the Spring Boot Project.**
2. **Define Database configurations.**
3. **Create an Entity Class.**
4. **Create JPA Data Repository layer.**
5. **Create Rest Controllers and map API requests.**
6. **Create Unit Testing for API requests and run the unit testing.**
7. **Build and run the Project.**

Requirement

We need to create a simple REST API to store user data to MySQL database with that API so we can create, update, delete, and get users information.

Web — Full-stack web development with Tomcat and Spring MVC

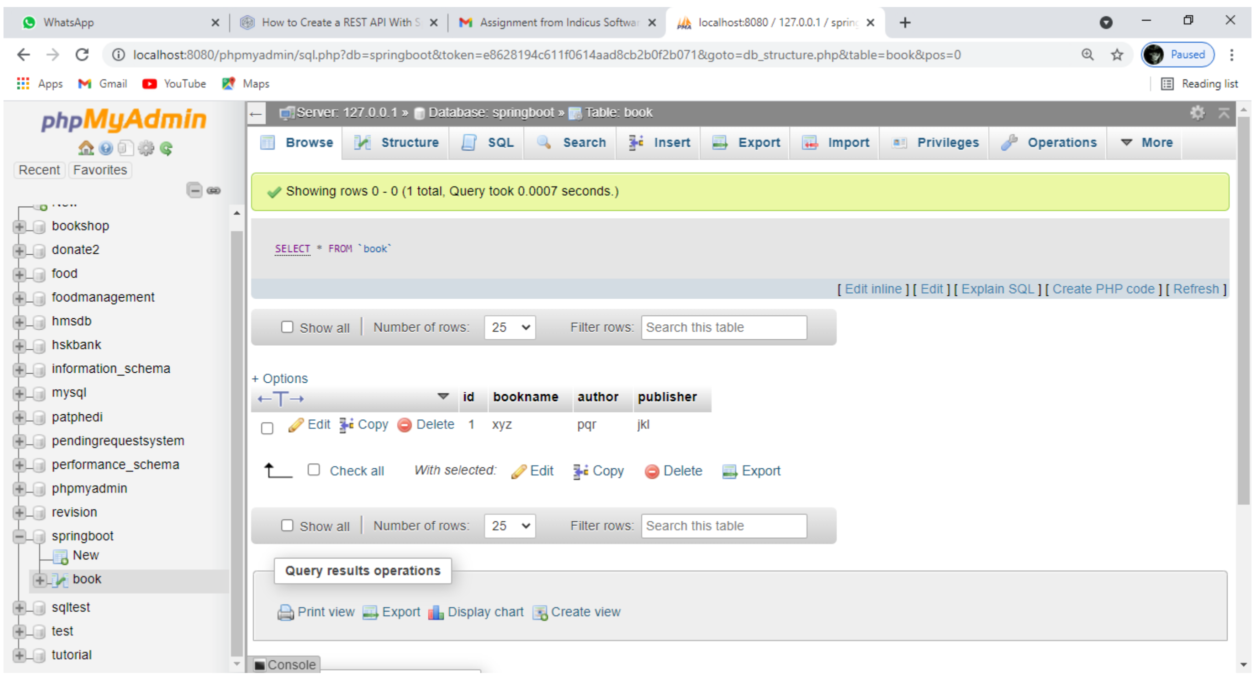
DevTools — Spring Boot Development Tools

JPA — Java Persistence API including spring-data-JPA, spring-orm, and Hibernate

MySQL — MySQL JDBC driver

Define Database Configurations

Next Create the database name called *springboot* in MySQL database server and define connection properties in spring-boot-rest-*api*-tutorial/src/main/resources/application.properties



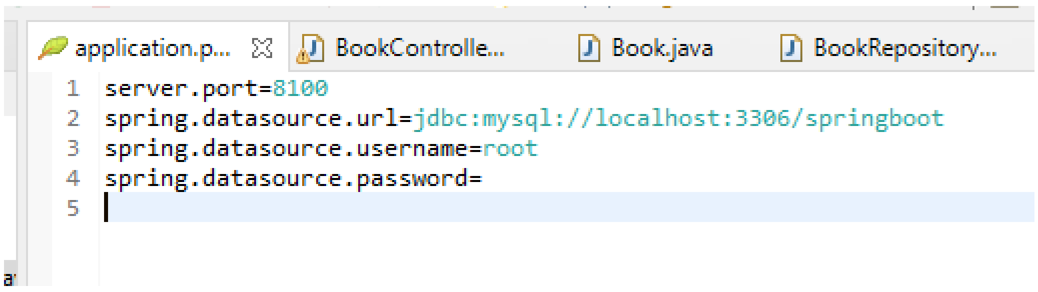
localhost:8080/phpmyadmin/sql.php?db=springboot&token=e8628194c611f0614aad8cb2b0f2b071&goto=db_structure.php&table=book&pos=0

Server: 127.0.0.1 » Database: springboot » Table: book

Showing rows 0 - 0 (1 total, Query took 0.0007 seconds.)

```
SELECT * FROM `book`
```

id	bookname	author	publisher
1	xyz	pqr	jkl



```
server.port=8100
spring.datasource.url=jdbc:mysql://localhost:3306/springboot
spring.datasource.username=root
spring.datasource.password=
```

Creating Entity Class

Book.java:

```java
package com.book.model;

import javax.persistence.*;

@Entity
@Table(name = "Book")
public class Book {
    private int id;
    private String bookname;
    private String author;
    private String publisher;

    public Book() {
    }

    public Book(int id, String bookname, String author, String publisher) {
        this.id = id;
        this.bookname = bookname;
        this.author = author;
        this.publisher = publisher;
    }

    @Id
    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    public String getBookname() {
        return bookname;
    }

    public void setBookname(String bookname) {
        this.bookname = bookname;
    }

    public String getAuthor() {
        return author;
    }

    public void setAuthor(String author) {
        this.author = author;
    }
    public String getPublisher() {
        return publisher;
    }

    public void setPublisher(String publisher) {
        this.publisher = publisher;
    }
}
```

Creating JPA Data Repository Layer:

package com.book.repository;

import com.book.model.Book;

import org.springframework.data.jpa.repository.JpaRepository;

public interface BookRepository extends JpaRepository<Book, Integer> {

}

Creating Rest Controllers and Map API Requests:

package com.book.controller;

import com.book.model.Book;

import com.book.service.BookService;

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

```
import org.springframework.http.HttpStatus;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.*;

import java.util.List;
import java.util.NoSuchElementException;

@CrossOrigin(maxAge = 3600)
@RestController
@RequestMapping("/books")
public class BookController {

    @Autowired
    BookService bookService;

    @GetMapping("")
    public List<Book> list() {
        return bookService.listAllBook();
    }

    @GetMapping("/{id}")
    public ResponseEntity<Book> get(@PathVariable Integer id) {
        try {
            Book book = bookService.getBook(id);
            return new ResponseEntity<Book>(book, HttpStatus.OK);
        }catch (NoSuchElementException e) {
            return new
ResponseEntity<Book>(HttpStatus.NOT_FOUND);
        }
    }

    @PostMapping("/addBook")
    public void add(@RequestBody Book book) {
        bookService.saveBook(book);
    }

    @PutMapping("/updateBook/{id}")
    public ResponseEntity<?> update(@RequestBody Book book,
@PathVariable Integer id) {
        try {
            Book existBook = bookService.getBook(id);
            book.setId(id);
            bookService.saveBook(book);

            return new ResponseEntity<>(HttpStatus.OK);
        } catch (NoSuchElementException e) {
            return new ResponseEntity<>(HttpStatus.NOT_FOUND);
        }
    }

    @DeleteMapping("deleteBook/{id}")
    public void delete(@PathVariable Integer id) {

        bookService.deleteBook(id);
    }
}
```

BookService.java:

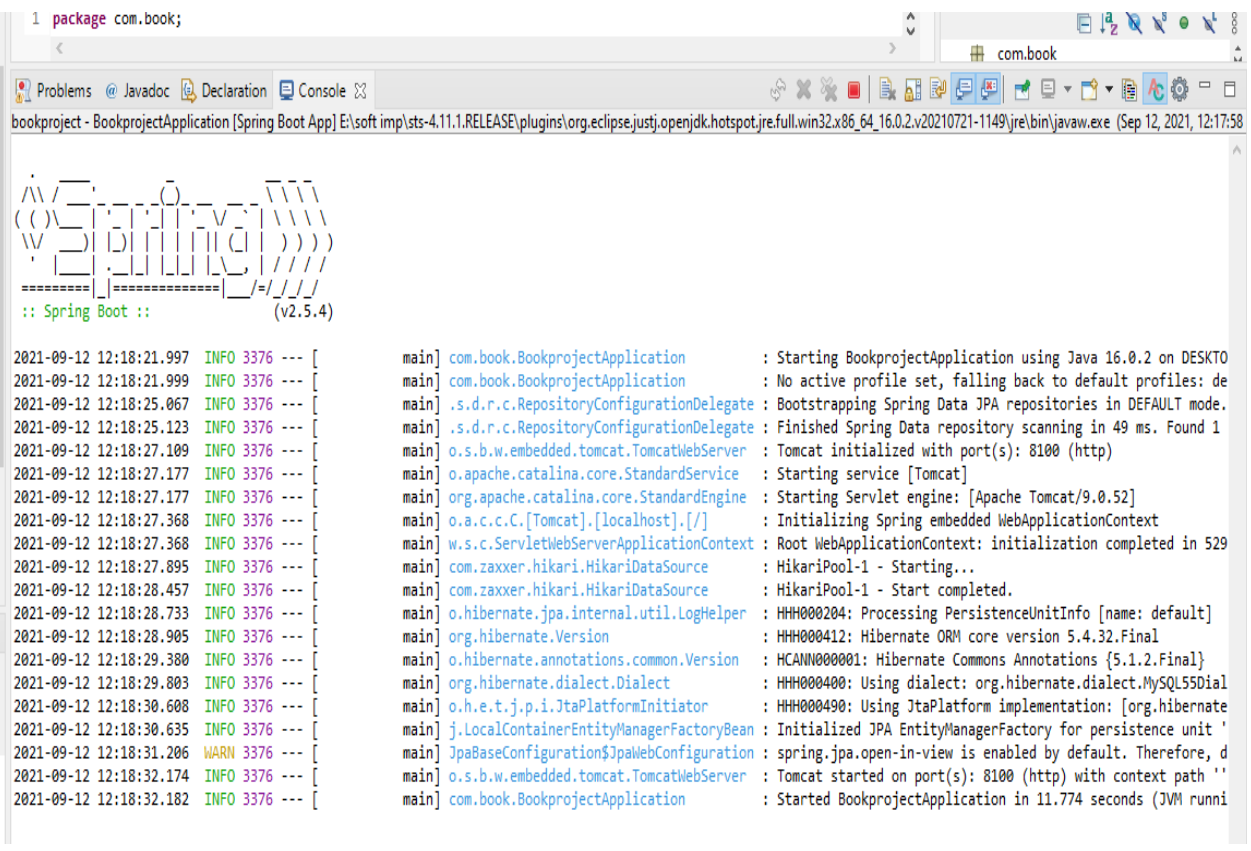
```
package com.book.service;

import com.book.model.Book;
import com.book.repository.BookRepository;
import
org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import javax.transaction.Transactional;
import java.util.List;

@Service
@Transactional
public class BookService {

    @Autowired
    private BookRepository bookRepository;

    public List<Book> listAllBook() {
        return bookRepository.findAll();
    }

    public void saveBook(Book book) {
        bookRepository.save(book);
    }
    public Book getBook(Integer id) {
        return bookRepository.findById(id).get();
    }

    public void deleteBook(Integer id) {
        bookRepository.deleteById(id);
    }
}
```

Creating html page to display the all records of books:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01
Transitional//EN">
<html>
  <head>
    <title></title>
    <meta http-equiv="Content-Type" content="text/html;
charset=UTF-8">
    <script language="javascript">
    var xmlhttp;
    function init() {
       xmlhttp = new XMLHttpRequest();

        var empno = document.getElementById("empno");
        var newurl = "http://localhost:8100/books";
        var url = "https://cdn-api.co-
vin.in/api/v2/appointment/sessions/public/findByDistrict?district
_id=395&date=11-09-2021";
        xmlhttp.open('GET',newurl,true);
        xmlhttp.send(null);
        xmlhttp.onreadystatechange = function() {
               if (xmlhttp.readyState == 4) {
                  if ( xmlhttp.status == 200) {

                    const obj = JSON.parse(xmlhttp.responseText);

                    var tempTable = "<table>"+
                        "<tr><th>Id </th><th>Book
Name</th><th>Author</th><th>Publisher</th></tr>";



                    for(var i = 0; i < obj.length; i++)
                    {

                        tempTable += "<tr><td>"+obj[i].id+
                          "</td><td>"+obj[i].bookname+
                            "</td><td>"+obj[i].author+
                            "</td><td>"+obj[i].publisher+
                            "</td></tr>";
                    }
                    document.getElementById("mydiv").innerHTML =
tempTable;
```

```
                }
                else
                      alert("Error ->" + xmlhttp.responseText);
             }
        };
    }

  </script>
  </head>
  <body  onload="init()">


   <div id="mydiv">
   </div>
  </body>
</html>
```

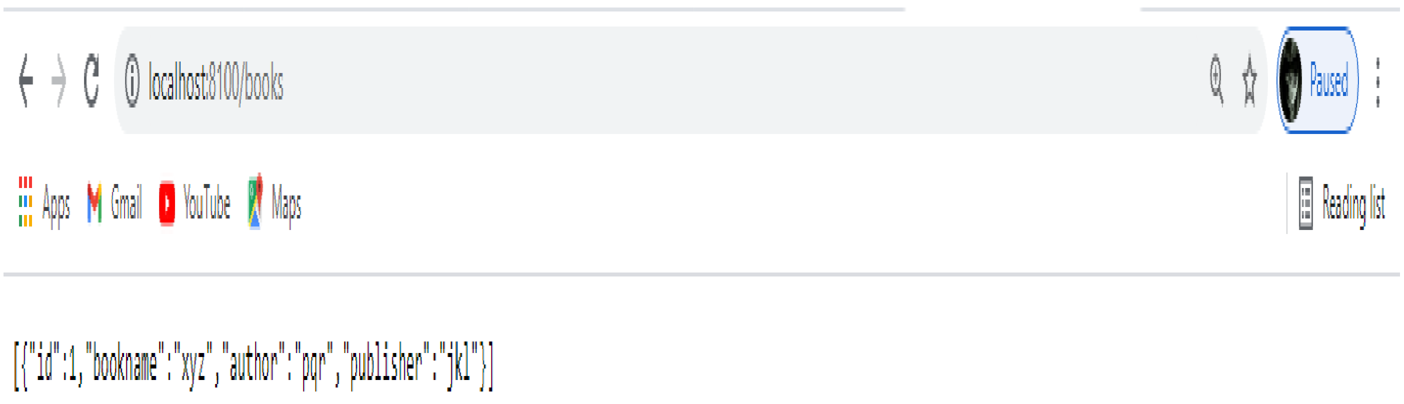
Outputs:



```
package com.book;
```

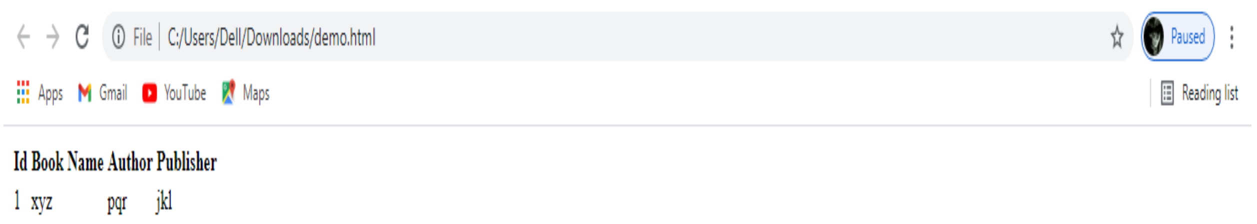
```
bookproject - BookprojectApplication [Spring Boot App] E:\soft imp\sts-4.11.1.RELEASE\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_16.0.2.v20210721-1149\jre\bin\javaw.exe (Sep 12, 2021, 12:17:58

  .   ____          _            __ _ _
 /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \ \
( ( )\___ | '_ | '_| | '_ \/ _` | \ \ \ \
 \\/  ___)| |_)| | | | | || (_| |  ) ) ) )
  '  |____| .__|_| |_|_| |_\__, | / / / /
 =========|_|==============|___/=/_/_/_/
 :: Spring Boot ::                (v2.5.4)

2021-09-12 12:18:21.997  INFO 3376 --- [           main] com.book.BookprojectApplication          : Starting BookprojectApplication using Java 16.0.2 on DESKTO
2021-09-12 12:18:21.999  INFO 3376 --- [           main] com.book.BookprojectApplication          : No active profile set, falling back to default profiles: de
2021-09-12 12:18:25.067  INFO 3376 --- [           main] .s.d.r.c.RepositoryConfigurationDelegate : Bootstrapping Spring Data JPA repositories in DEFAULT mode.
2021-09-12 12:18:25.123  INFO 3376 --- [           main] .s.d.r.c.RepositoryConfigurationDelegate : Finished Spring Data repository scanning in 49 ms. Found 1
2021-09-12 12:18:27.109  INFO 3376 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat initialized with port(s): 8100 (http)
2021-09-12 12:18:27.177  INFO 3376 --- [           main] o.apache.catalina.core.StandardService   : Starting service [Tomcat]
2021-09-12 12:18:27.177  INFO 3376 --- [           main] org.apache.catalina.core.StandardEngine  : Starting Servlet engine: [Apache Tomcat/9.0.52]
2021-09-12 12:18:27.368  INFO 3376 --- [           main] o.a.c.c.C.[Tomcat].[localhost].[/]       : Initializing Spring embedded WebApplicationContext
2021-09-12 12:18:27.368  INFO 3376 --- [           main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 529
2021-09-12 12:18:27.895  INFO 3376 --- [           main] com.zaxxer.hikari.HikariDataSource       : HikariPool-1 - Starting...
2021-09-12 12:18:28.457  INFO 3376 --- [           main] com.zaxxer.hikari.HikariDataSource       : HikariPool-1 - Start completed.
2021-09-12 12:18:28.733  INFO 3376 --- [           main] o.hibernate.jpa.internal.util.LogHelper  : HHH000204: Processing PersistenceUnitInfo [name: default]
2021-09-12 12:18:28.905  INFO 3376 --- [           main] org.hibernate.Version                    : HHH000412: Hibernate ORM core version 5.4.32.Final
2021-09-12 12:18:29.380  INFO 3376 --- [           main] o.hibernate.annotations.common.Version   : HCANN000001: Hibernate Commons Annotations {5.1.2.Final}
2021-09-12 12:18:29.803  INFO 3376 --- [           main] org.hibernate.dialect.Dialect            : HHH000400: Using dialect: org.hibernate.dialect.MySQL55Dial
2021-09-12 12:18:30.608  INFO 3376 --- [           main] o.h.e.t.j.p.i.JtaPlatformInitiator       : HHH000490: Using JtaPlatform implementation: [org.hibernate
2021-09-12 12:18:30.635  INFO 3376 --- [           main] j.LocalContainerEntityManagerFactoryBean : Initialized JPA EntityManagerFactory for persistence unit '
2021-09-12 12:18:31.206  WARN 3376 --- [           main] JpaBaseConfiguration$JpaWebConfiguration : spring.jpa.open-in-view is enabled by default. Therefore, d
2021-09-12 12:18:32.174  INFO 3376 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat started on port(s): 8100 (http) with context path ''
2021-09-12 12:18:32.182  INFO 3376 --- [           main] com.book.BookprojectApplication          : Started BookprojectApplication in 11.774 seconds (JVM runni
```

Running API:



```
[{"id":1,"bookname":"xyz","author":"pqr","publisher":"jkl"}]
```

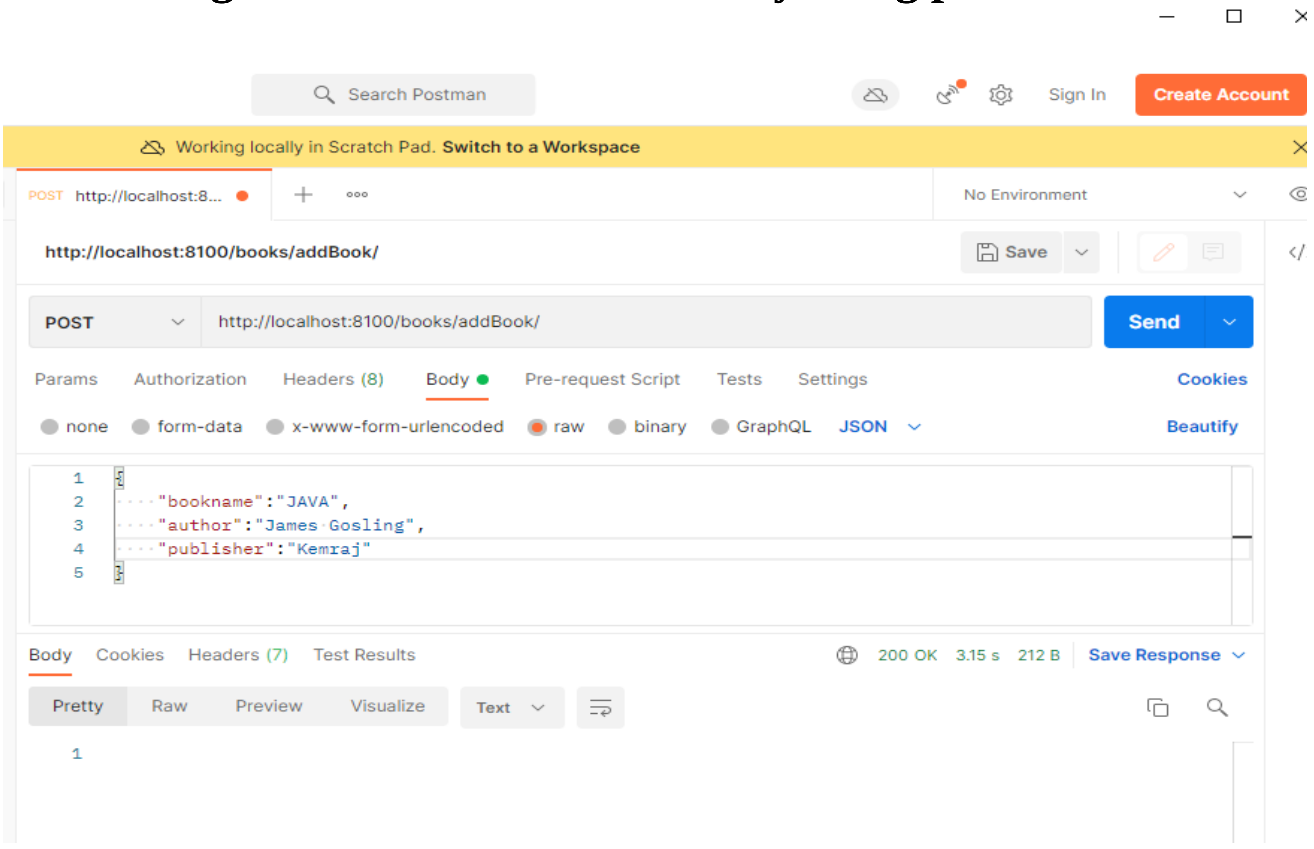
Showing the html page to user with book information:



Id	Book Name	Author	Publisher
1	xyz	pqr	jkl

Adding records of book using postman by sending the url

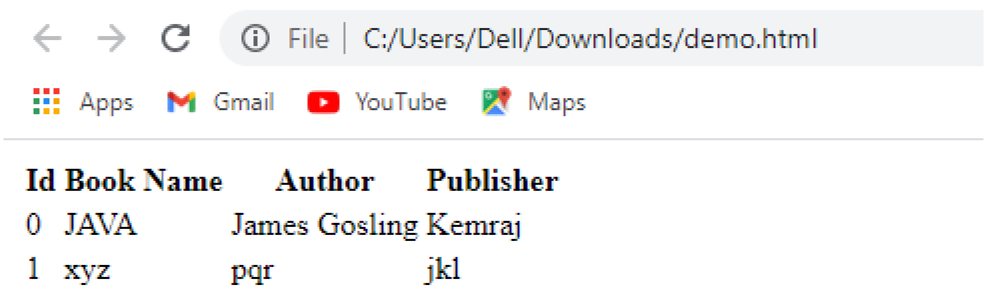
and adding records in Jason format by using post method:



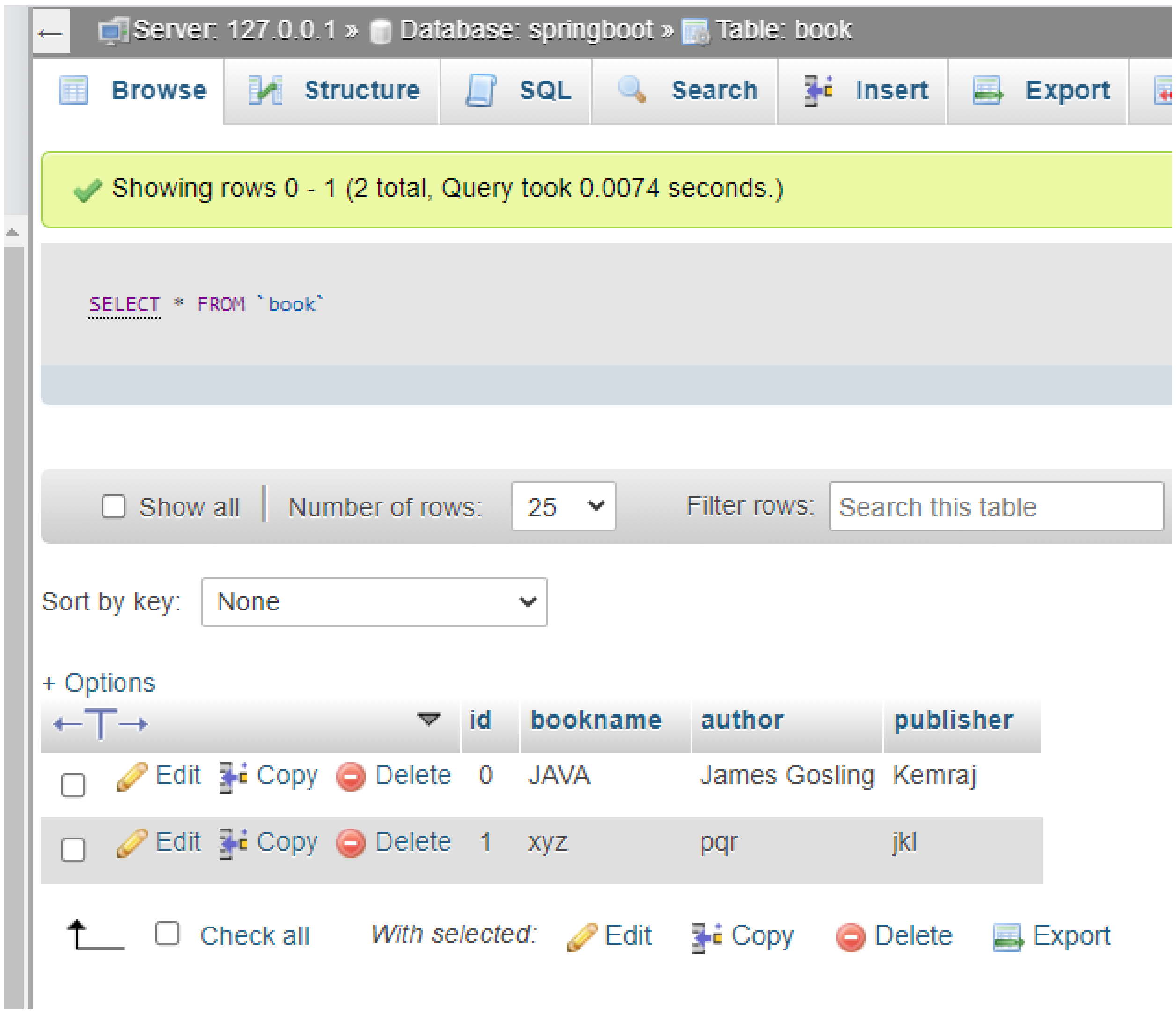
POST http://localhost:8100/books/addBook/

```
{
    "bookname":"JAVA",
    "author":"James Gosling",
    "publisher":"Kemraj"
}
```

200 OK 3.15 s 212 B



Id	Book Name	Author	Publisher
0	JAVA	James Gosling	Kemraj
1	xyz	pqr	jkl


Server: 127.0.0.1 » Database: springboot » Table: book
Browse
Structure
SQL
Search
Insert
Export

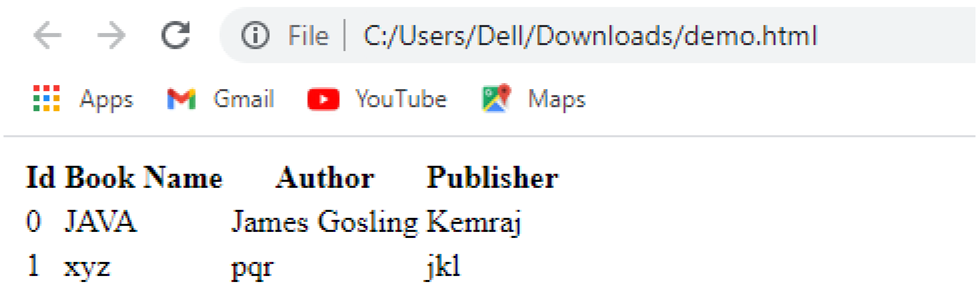
Showing rows 0 - 1 (2 total, Query took 0.0074 seconds.)

```
SELECT * FROM `book`
```

Show all
Number of rows: 25
Filter rows: Search this table
Sort by key: None
+ Options
Check all
With selected: Edit Copy Delete Export

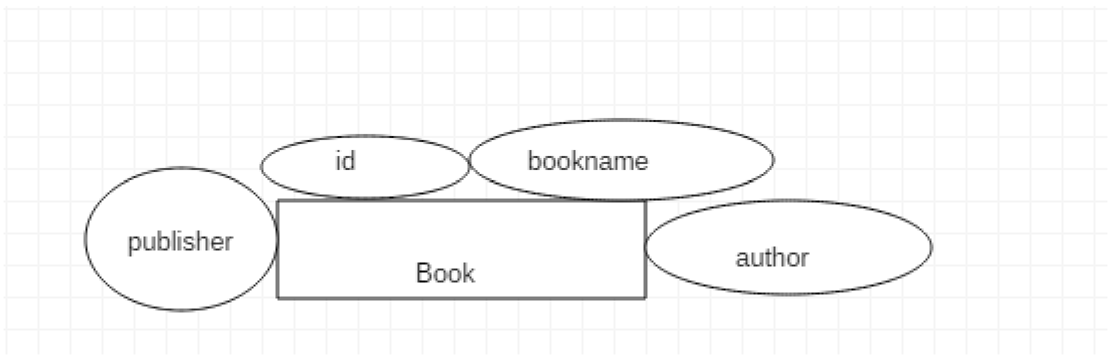
	id	bookname	author	publisher
Edit Copy Delete	0	JAVA	James Gosling	Kemraj
Edit Copy Delete	1	xyz	pqr	jkl

Displaying all Records To user:



Id	Book Name	Author	Publisher
0	JAVA	James Gosling	Kemraj
1	xyz	pqr	jkl

ER Diagram:


id
bookname
publisher
Book
author