REST API:

com.ram.demo

com.ram.demo.controller

com.ram.demo.entities

com.ram.demo.services

===================================================

package com.ram.demo.entities;

public class Book {

private int id;

private String title;

private String author;

public Book(int id, String title, String author) {

super();

this.id = id;

this.title = title;

this.author = author;

}

public Book() {

super();

// TODO Auto-generated constructor stub

}

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public String getTitle() {

return title;

}

public void setTitle(String title) {

this.title = title;

}

public String getAuthor() {

return author;

}

public void setAuthor(String author) {

this.author = author;

}

}

========================================================

package com.ram.demo.services;

import java.util.ArrayList;

import java.util.List;

import org.springframework.stereotype.Component;

import com.ram.demo.entities.Book;

@Component

public class BookService {

private static List<Book> list=new ArrayList<Book>();

static {

list.add(new Book(101, "Java Book", "Ram"));

list.add(new Book(102, "python Book", "deepak"));

list.add(new Book(103, "DotNet Book", "shivam"));

}

public List<Book> getAllBook(){

return list;

}

public Book getBookById(int id) {

Book k=new Book();

for(Book b:list) {

if(b.getId()==id) {

k=b;

}

}

return k;

}

public void addBook(Book b) {

list.add(b);

}

public void deleteBookById(int id) {

for(int i=0;i<list.size();i++) {

if(id==list.get(i).getId()) {

list.remove(i);

System.out.println("Data Delete success");

}

}

}

public void updateBookById(Book book,int id) {

for(int i=0;i<list.size();i++) {

if(id==list.get(i).getId()) {

list.set(i, book);

System.out.println("Data update success");

}

}

}

}

==================================================

4. BookController

package com.ram.demo.controller;

import java.util.List;

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

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

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.PutMapping;

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

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

import com.ram.demo.entities.Book;

import com.ram.demo.services.BookService;

@RestController

public class BookController {

@Autowired

private BookService service;

public BookService getService() {

return service;

}

public void setService(BookService service) {

this.service = service;

}

//http://localhost:8080/books

@GetMapping("/books")

public List<Book> getBooks() {

return this.service.getAllBook();

}

//http://localhost:8080/book/101

@GetMapping("/book/{id}")

public Book getBooksById(@PathVariable("id")int id) {

return this.service.getBookById(id);

}

//http://localhost:8080/books

@PostMapping("/books")

/\*

{"id":105,

"title":"JavaScript",

"author":"kartik"

}

\*/

public void saveBook(@RequestBody Book book)

{ System.out.println("Data add Success");

this.service.addBook(book);

}

//http://localhost:8080/books/105

@PutMapping("/books/{id}")

public void updateBook(@RequestBody Book book,@PathVariable("id")int id)

{ System.out.println("Data update Success");

book.setId(id);

this.service.updateBookById(book, id);

}

//http://localhost:8080/book/103

@DeleteMapping("/book/{id}")

public void deleteBook(@PathVariable("id")int id) {

System.out.println("Data Deleted Success");

this.service.deleteBookById(id);

}

}

REST\_API\_without\_Database.txt

Open with

Displaying REST\_API\_without\_Database.txt.

# Class Notes REST API Without Database

More options

Ram Lovewanshi

•

Yesterday

[](https://drive.google.com/file/d/1CWOq60vzsrYBCo9SuuhQq0AlrxE32Pk3/view?usp=classroom_web&authuser=0" \o "REST_API_without_Database.txt" \t "https://classroom.google.com/c/NzE5OTc3NTAzNjE1/m/NzYzNDM1OTU5MjMy/_blank)

[REST\_API\_without\_Database.txt](https://drive.google.com/file/d/1CWOq60vzsrYBCo9SuuhQq0AlrxE32Pk3/view?usp=classroom_web&authuser=0" \o "REST_API_without_Database.txt" \t "https://classroom.google.com/c/NzE5OTc3NTAzNjE1/m/NzYzNDM1OTU5MjMy/_blank)

[Text](https://drive.google.com/file/d/1CWOq60vzsrYBCo9SuuhQq0AlrxE32Pk3/view?usp=classroom_web&authuser=0" \o "REST_API_without_Database.txt" \t "https://classroom.google.com/c/NzE5OTc3NTAzNjE1/m/NzYzNDM1OTU5MjMy/_blank)

Class comments

REST API:

com.ram.demo

com.ram.demo.controller

com.ram.demo.entities

com.ram.demo.services

===================================================

package com.ram.demo.entities;

public class Book {

private int id;

private String title;

private String author;

public Book(int id, String title, String author) {

super();

this.id = id;

this.title = title;

this.author = author;

}

public Book() {

super();

// TODO Auto-generated constructor stub

}

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public String getTitle() {

return title;

}

public void setTitle(String title) {

this.title = title;

}

public String getAuthor() {

return author;

}

public void setAuthor(String author) {

this.author = author;

}

}

========================================================

package com.ram.demo.services;

import java.util.ArrayList;

import java.util.List;

import org.springframework.stereotype.Component;

import com.ram.demo.entities.Book;

@Component

public class BookService {

private static List<Book> list=new ArrayList<Book>();

static {

list.add(new Book(101, "Java Book", "Ram"));

list.add(new Book(102, "python Book", "deepak"));

list.add(new Book(103, "DotNet Book", "shivam"));

}

public List<Book> getAllBook(){

return list;

}

public Book getBookById(int id) {

Book k=new Book();

for(Book b:list) {

if(b.getId()==id) {

k=b;

}

}

return k;

}

public void addBook(Book b) {

list.add(b);

}

public void deleteBookById(int id) {

for(int i=0;i<list.size();i++) {

if(id==list.get(i).getId()) {

list.remove(i);

System.out.println("Data Delete success");

}

}

}

public void updateBookById(Book book,int id) {

for(int i=0;i<list.size();i++) {

if(id==list.get(i).getId()) {

list.set(i, book);

System.out.println("Data update success");

}

}

}

}

==================================================

4. BookController

package com.ram.demo.controller;

import java.util.List;

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

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

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.PutMapping;

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

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

import com.ram.demo.entities.Book;

import com.ram.demo.services.BookService;

@RestController

public class BookController {

@Autowired

private BookService service;

public BookService getService() {

return service;

}

public void setService(BookService service) {

this.service = service;

}

//http://localhost:8080/books

@GetMapping("/books")

public List<Book> getBooks() {

return this.service.getAllBook();

}

//http://localhost:8080/book/101

@GetMapping("/book/{id}")

public Book getBooksById(@PathVariable("id")int id) {

return this.service.getBookById(id);

}

//http://localhost:8080/books

@PostMapping("/books")

/\*

{"id":105,

"title":"JavaScript",

"author":"kartik"

}

\*/

public void saveBook(@RequestBody Book book)

{ System.out.println("Data add Success");

this.service.addBook(book);

}

//http://localhost:8080/books/105

@PutMapping("/books/{id}")

public void updateBook(@RequestBody Book book,@PathVariable("id")int id)

{ System.out.println("Data update Success");

book.setId(id);

this.service.updateBookById(book, id);

}

//http://localhost:8080/book/103

@DeleteMapping("/book/{id}")

public void deleteBook(@PathVariable("id")int id) {

System.out.println("Data Deleted Success");

this.service.deleteBookById(id);

}

}

REST\_API\_without\_Database.txt

Open with

Displaying REST\_API\_without\_Database.txt.

# Class Notes REST API Without Database

More options

Ram Lovewanshi

•

Yesterday

[](https://drive.google.com/file/d/1CWOq60vzsrYBCo9SuuhQq0AlrxE32Pk3/view?usp=classroom_web&authuser=0" \o "REST_API_without_Database.txt" \t "https://classroom.google.com/c/NzE5OTc3NTAzNjE1/m/NzYzNDM1OTU5MjMy/_blank)

[REST\_API\_without\_Database.txt](https://drive.google.com/file/d/1CWOq60vzsrYBCo9SuuhQq0AlrxE32Pk3/view?usp=classroom_web&authuser=0" \o "REST_API_without_Database.txt" \t "https://classroom.google.com/c/NzE5OTc3NTAzNjE1/m/NzYzNDM1OTU5MjMy/_blank)

[Text](https://drive.google.com/file/d/1CWOq60vzsrYBCo9SuuhQq0AlrxE32Pk3/view?usp=classroom_web&authuser=0" \o "REST_API_without_Database.txt" \t "https://classroom.google.com/c/NzE5OTc3NTAzNjE1/m/NzYzNDM1OTU5MjMy/_blank)

Class comments