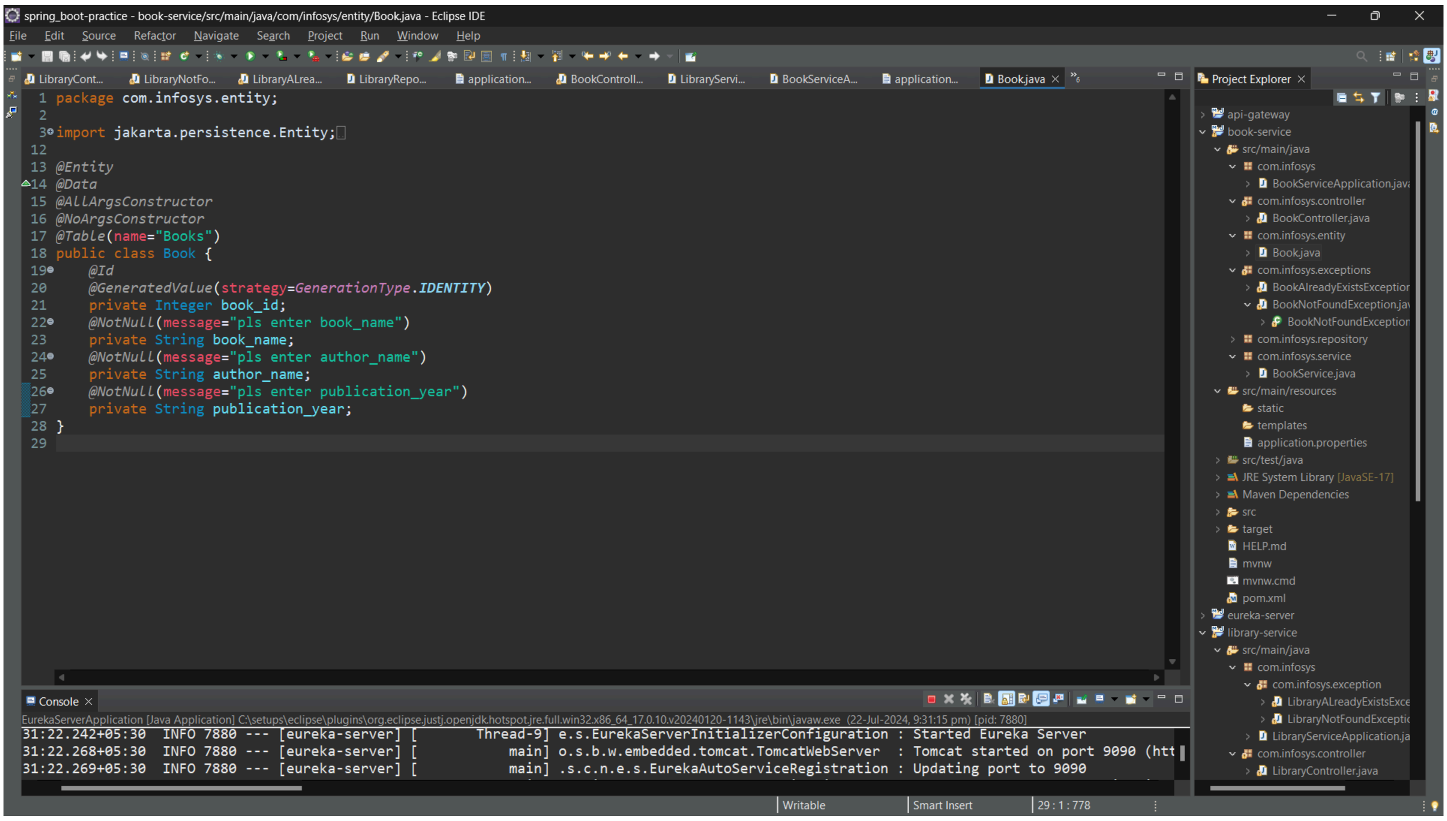
Final-Spring-Capstone-Project

Name-Prathamesh Sonje Id-72288231K

Book-Service

entity-

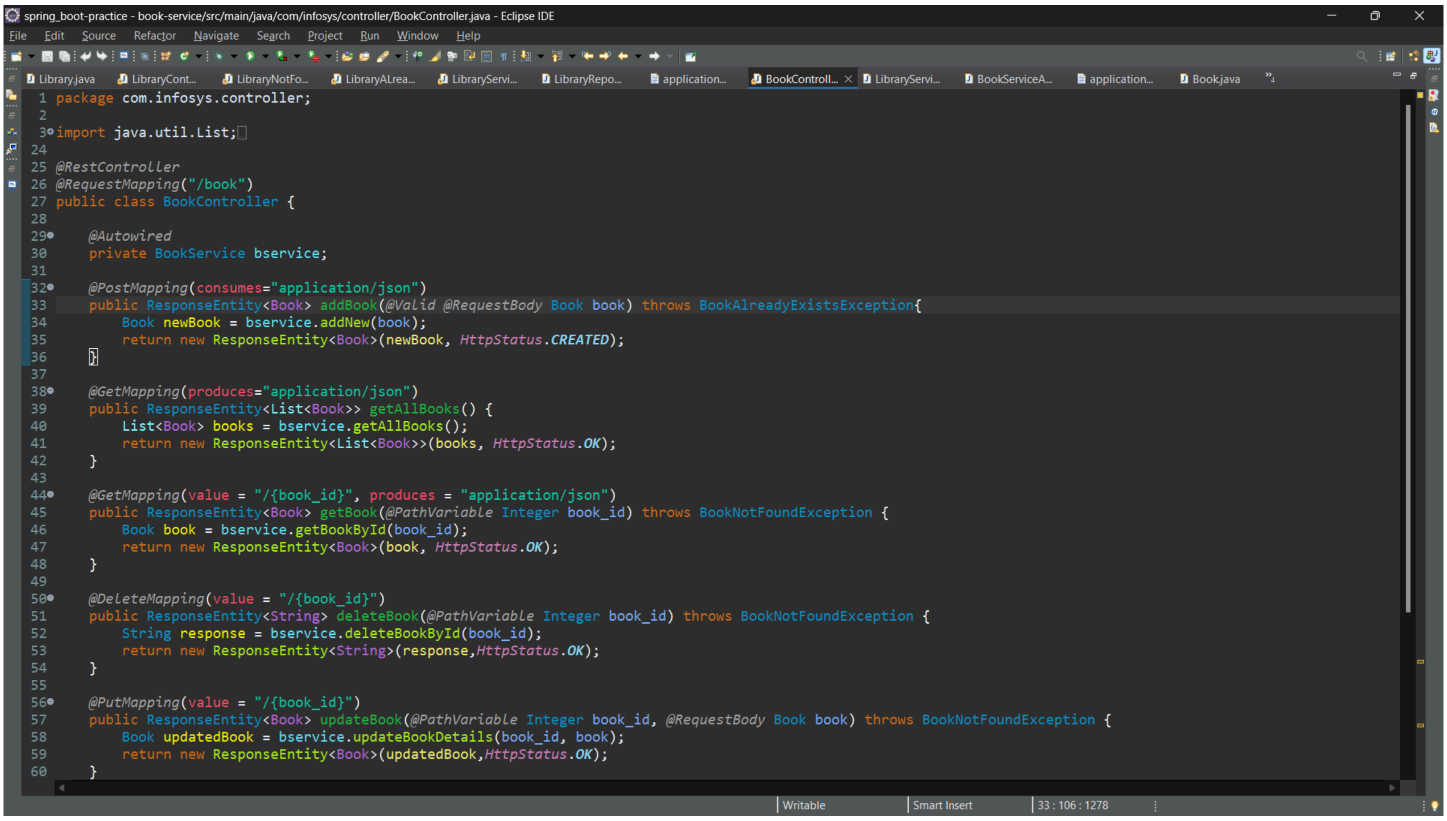


spring_boot-practice - book-service/src/main/java/com/infosys/entity/Book.java - Eclipse IDE

```java
package com.infosys.entity;

import jakarta.persistence.Entity;

@Entity
@Data
@AllArgsConstructor
@NoArgsConstructor
@Table(name="Books")
public class Book {
	@Id
	@GeneratedValue(strategy=GenerationType.IDENTITY)
	private Integer book_id;
	@NotNull(message="pls enter book_name")
	private String book_name;
	@NotNull(message="pls enter author_name")
	private String author_name;
	@NotNull(message="pls enter publication_year")
	private String publication_year;
}
```

```
EurekaServerApplication [Java Application] C:\setups\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.10.v20240120-1143\jre\bin\javaw.exe  (22-Jul-2024, 9:31:15 pm) [pid: 7880]
31:22.242+05:30  INFO 7880 --- [eureka-server] [       Thread-9] e.s.EurekaServerInitializerConfiguration : Started Eureka Server
31:22.268+05:30  INFO 7880 --- [eureka-server] [           main] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat started on port 9090 (htt
31:22.269+05:30  INFO 7880 --- [eureka-server] [           main] .s.c.n.e.s.EurekaAutoServiceRegistration : Updating port to 9090
```

controller-

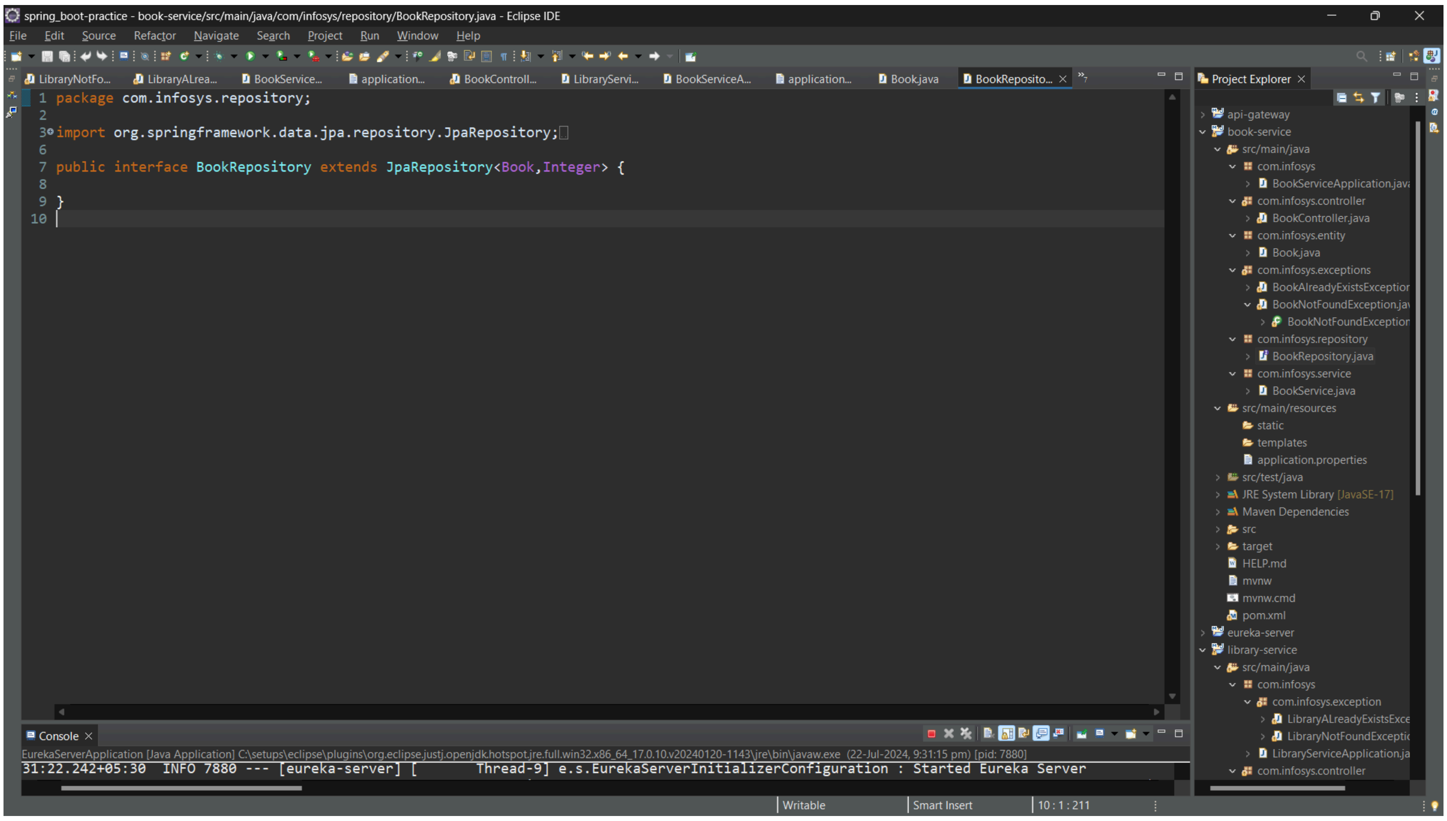


spring_boot_practice - book-service/src/main/java/com/infosys/controller/BookController.java - Eclipse IDE

```java
package com.infosys.controller;

import java.util.List;

@RestController
@RequestMapping("/book")
public class BookController {

    @Autowired
    private BookService bservice;

    @PostMapping(consumes="application/json")
    public ResponseEntity<Book> addBook(@Valid @RequestBody Book book) throws BookAlreadyExistsException{
        Book newBook = bservice.addNew(book);
        return new ResponseEntity<Book>(newBook, HttpStatus.CREATED);
    }

    @GetMapping(produces="application/json")
    public ResponseEntity<List<Book>> getAllBooks() {
        List<Book> books = bservice.getAllBooks();
        return new ResponseEntity<List<Book>>(books, HttpStatus.OK);
    }

    @GetMapping(value = "/{book_id}", produces = "application/json")
    public ResponseEntity<Book> getBook(@PathVariable Integer book_id) throws BookNotFoundException {
        Book book = bservice.getBookById(book_id);
        return new ResponseEntity<Book>(book, HttpStatus.OK);
    }

    @DeleteMapping(value = "/{book_id}")
    public ResponseEntity<String> deleteBook(@PathVariable Integer book_id) throws BookNotFoundException {
        String response = bservice.deleteBookById(book_id);
        return new ResponseEntity<String>(response,HttpStatus.OK);
    }

    @PutMapping(value = "/{book_id}")
    public ResponseEntity<Book> updateBook(@PathVariable Integer book_id, @RequestBody Book book) throws BookNotFoundException {
        Book updatedBook = bservice.updateBookDetails(book_id, book);
        return new ResponseEntity<Book>(updatedBook,HttpStatus.OK);
    }
```

Writable | Smart Insert | 33 : 106 : 1278

repository-



spring_boot_practice - book-service/src/main/java/com/infosys/repository/BookRepository.java - Eclipse IDE

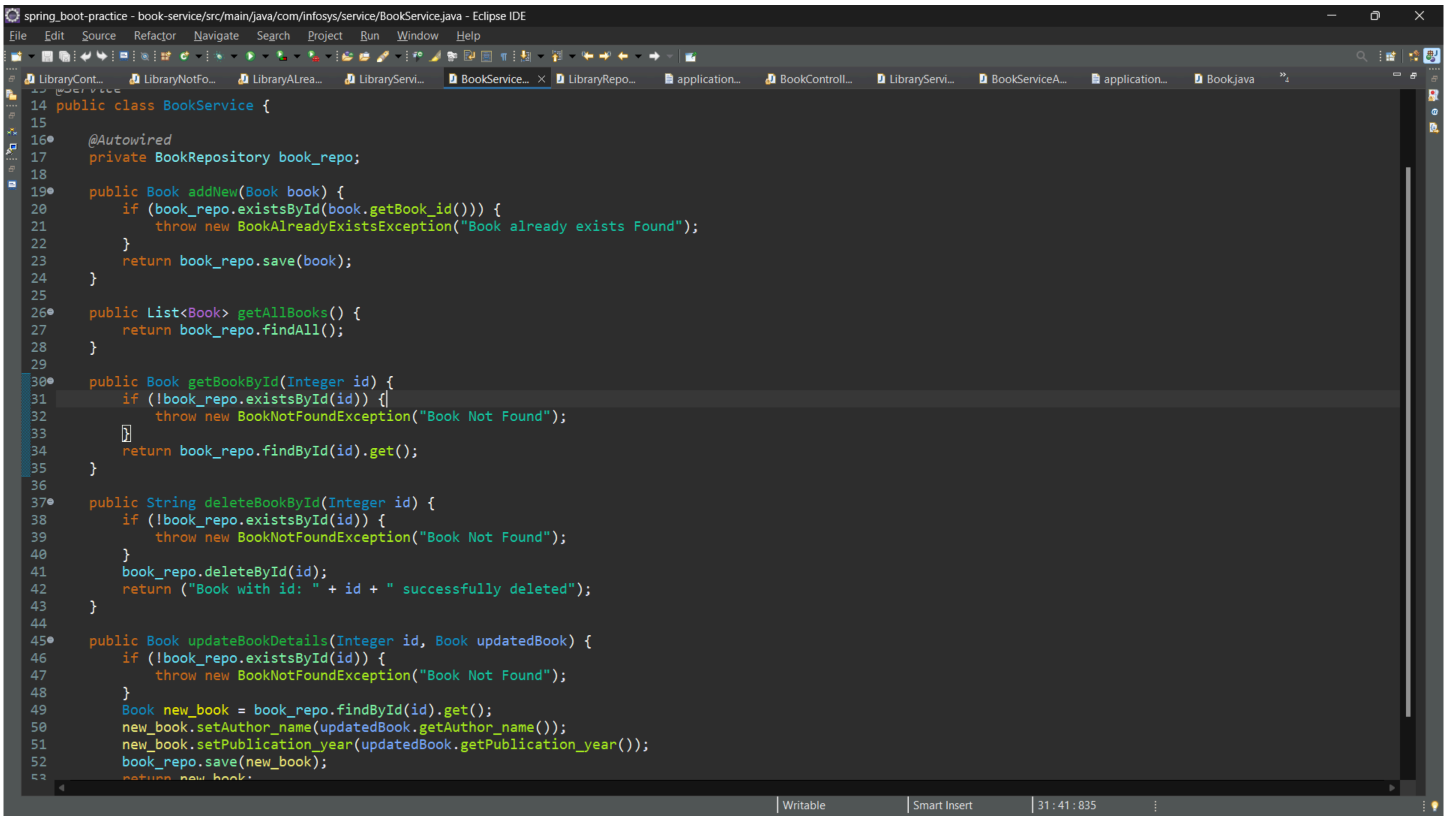
```java
package com.infosys.repository;

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

public interface BookRepository extends JpaRepository<Book,Integer> {

}
```

Console

```
EurekaServerApplication [Java Application] C:\setups\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.10.v20240120-1143\jre\bin\javaw.exe (22-Jul-2024, 9:31:15 pm) [pid: 7880]
31:22.242+05:30  INFO 7880 --- [eureka-server] [       Thread-9] e.s.EurekaServerInitializerConfiguration : Started Eureka Server
```

Writable | Smart Insert | 10 : 1 : 211

service-


spring_boot-practice - book-service/src/main/java/com/infosys/service/BookService.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
LibraryCont... LibraryNotFo... LibraryAlrea... LibraryServi... BookService... LibraryRepo... application... BookControll... LibraryServi... BookServiceA... application... Book.java
Writable
Smart Insert
31 : 41 : 835

```
@Service
public class BookService {

    @Autowired
    private BookRepository book_repo;

    public Book addNew(Book book) {
        if (book_repo.existsById(book.getBook_id())) {
            throw new BookAlreadyExistsException("Book already exists Found");
        }
        return book_repo.save(book);
    }

    public List<Book> getAllBooks() {
        return book_repo.findAll();
    }

    public Book getBookById(Integer id) {
        if (!book_repo.existsById(id)) {
            throw new BookNotFoundException("Book Not Found");
        }
        return book_repo.findById(id).get();
    }

    public String deleteBookById(Integer id) {
        if (!book_repo.existsById(id)) {
            throw new BookNotFoundException("Book Not Found");
        }
        book_repo.deleteById(id);
        return ("Book with id: " + id + " successfully deleted");
    }

    public Book updateBookDetails(Integer id, Book updatedBook) {
        if (!book_repo.existsById(id)) {
            throw new BookNotFoundException("Book Not Found");
        }
        Book new_book = book_repo.findById(id).get();
        new_book.setAuthor_name(updatedBook.getAuthor_name());
        new_book.setPublication_year(updatedBook.getPublication_year());
        book_repo.save(new_book);
        return new_book;
```

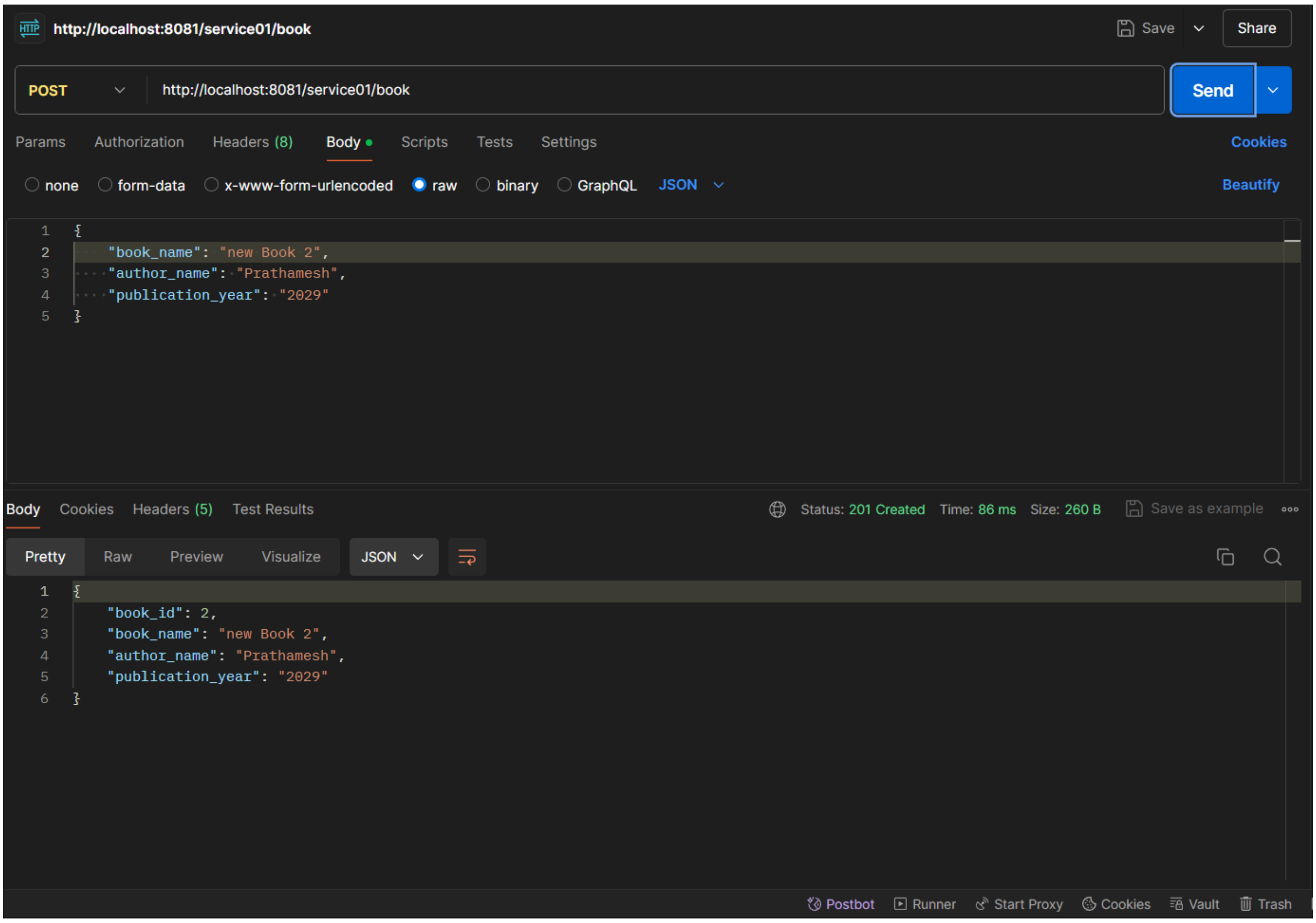
Q1. create endpoints:

2)Add New Book-

code -

```
  @PostMapping(consumes="application/json")
  public ResponseEntity<Book> addBook(@Valid @RequestBody Book book) throws BookAlreadyExistsException{
      Book newBook = bservice.addNew(book);
      return new ResponseEntity<Book>(newBook, HttpStatus.CREATED);
  }
```

output -



POST http://localhost:8081/service01/book

```
{
    "book_name": "new Book 2",
    "author_name": "Prathamesh",
    "publication_year": "2029"
}
```

Status: 201 Created Time: 86 ms Size: 260 B

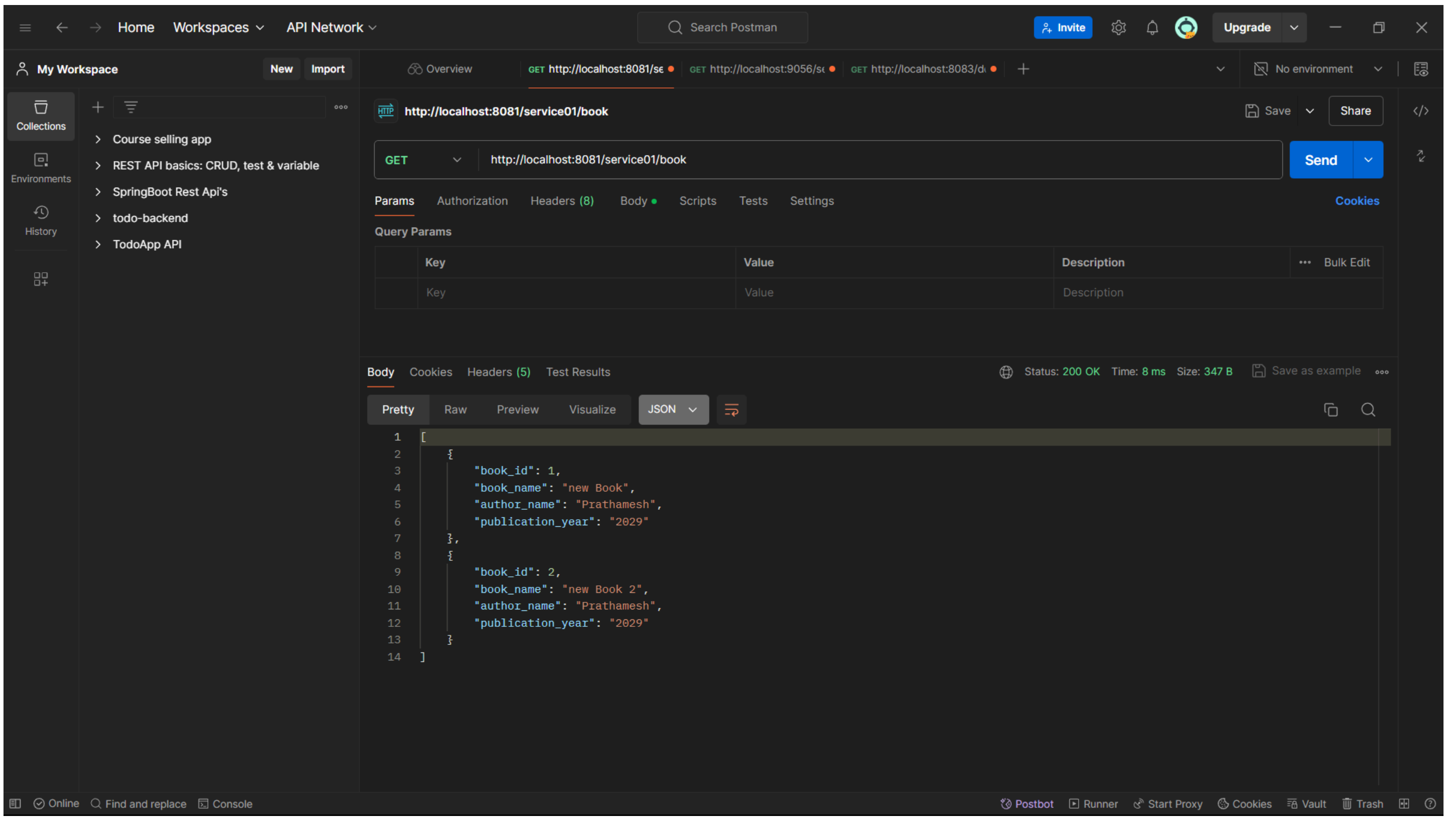
```
{
    "book_id": 2,
    "book_name": "new Book 2",
    "author_name": "Prathamesh",
    "publication_year": "2029"
}
```

show all book details in json

code -

```
  @GetMapping(produces="application/json")
  public ResponseEntity<List<Book>> getAllBooks() {
      List<Book> books = bservice.getAllBooks();
      return new ResponseEntity<List<Book>>(books, HttpStatus.OK);
  }
```

output-


GET http://localhost:8081/service01/book

Status: 200 OK Time: 8 ms Size: 347 B

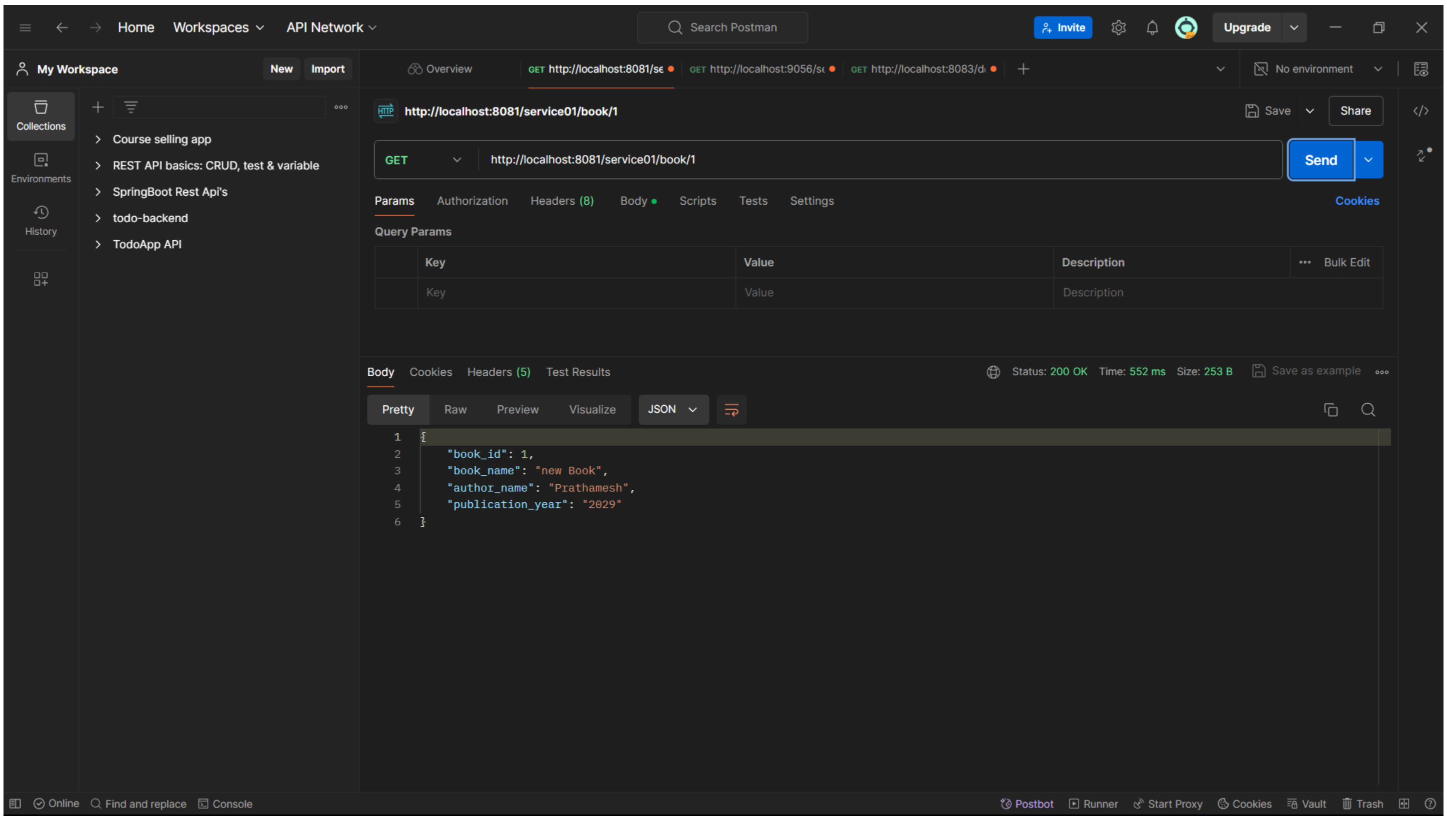
```
[
    {
        "book_id": 1,
        "book_name": "new Book",
        "author_name": "Prathamesh",
        "publication_year": "2029"
    },
    {
        "book_id": 2,
        "book_name": "new Book 2",
        "author_name": "Prathamesh",
        "publication_year": "2029"
    }
]
```

Search Book by Id -

code -

```
   @GetMapping(value = "/{book_id}", produces = "application/json")
   public ResponseEntity<Book> getBook(@PathVariable Integer book_id) throws
BookNotFoundException {
       Book book = bservice.getBookById(book_id);
       return new ResponseEntity<Book>(book, HttpStatus.OK);
   }
```

output -



Course selling app
REST API basics: CRUD, test & variable
SpringBoot Rest Api's
todo-backend
TodoApp API

GET http://localhost:8081/service01/book/1

Query Params

Key	Value	Description
Key	Value	Description

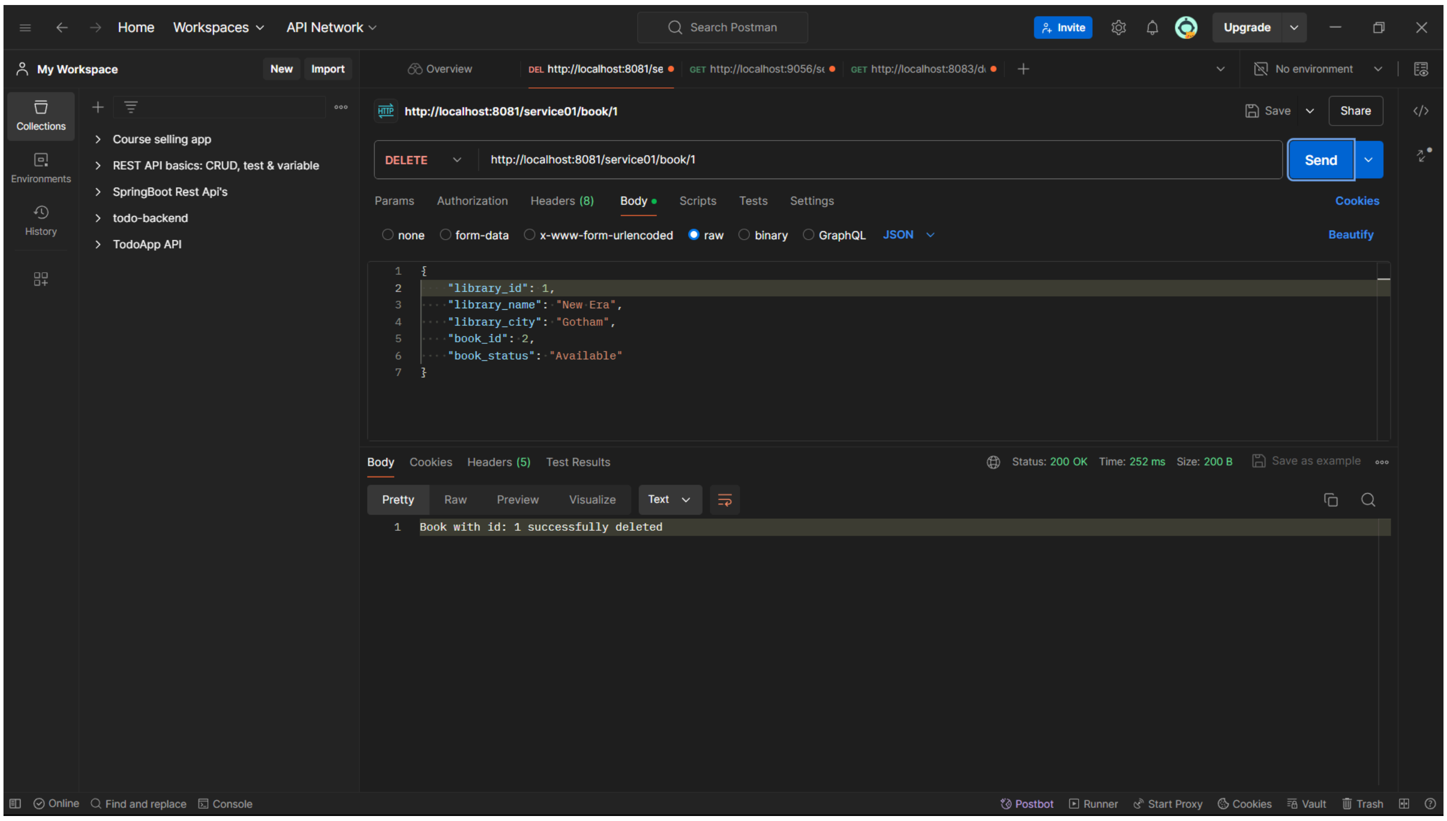
Body Cookies Headers (5) Test Results — Status: 200 OK Time: 552 ms Size: 253 B

```
{
    "book_id": 1,
    "book_name": "new Book",
    "author_name": "Prathamesh",
    "publication_year": "2029"
}
```

Delete book by given id-

code -

```
   @DeleteMapping(value = "/{book_id}")
   public ResponseEntity<String> deleteBook(@PathVariable Integer book_id) throws BookNotFoundException {
       String response = bservice.deleteBookById(book_id);
       return new ResponseEntity<String>(response,HttpStatus.OK);
   }
```



DELETE http://localhost:8081/service01/book/1

```
{
    "library_id": 1,
    "library_name": "New Era",
    "library_city": "Gotham",
    "book_id": 2,
    "book_status": "Available"
}
```

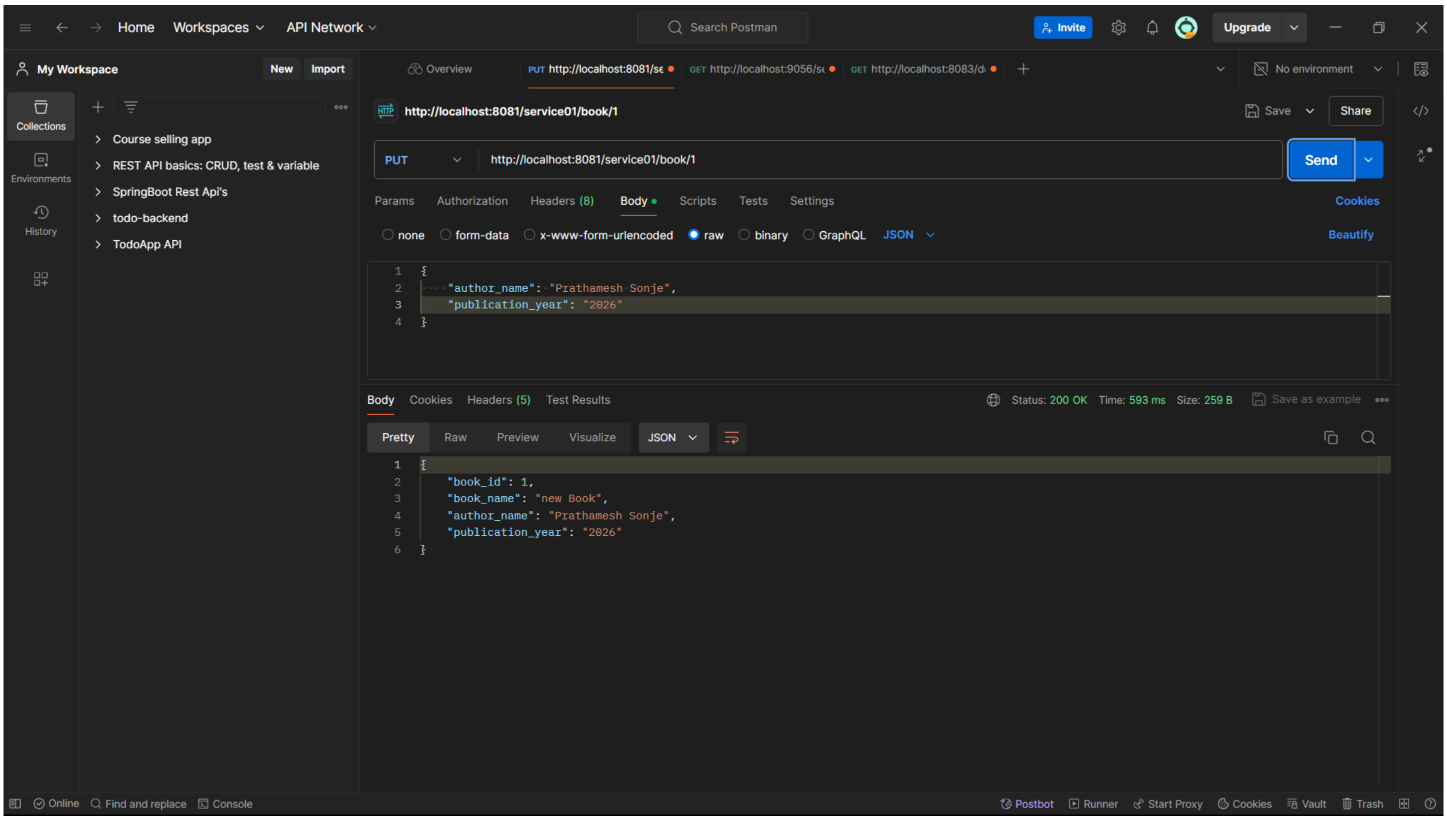
Status: 200 OK Time: 252 ms Size: 200 B

```
Book with id: 1 successfully deleted
```

update Book -

```
@PutMapping(value = "/{book_id}")
public ResponseEntity<Book> updateBook(@PathVariable Integer book_id, @RequestBody Book book) throws BookNotFoundException {
    Book updatedBook = bservice.updateBookDetails(book_id, book);
    return new ResponseEntity<Book>(updatedBook,HttpStatus.OK);
}
```

output -


PUT http://localhost:8081/service01/book/1

Request body (raw, JSON):

```
{
    "author_name": "Prathamesh Sonje",
    "publication_year": "2026"
}
```

Status: 200 OK Time: 593 ms Size: 259 B

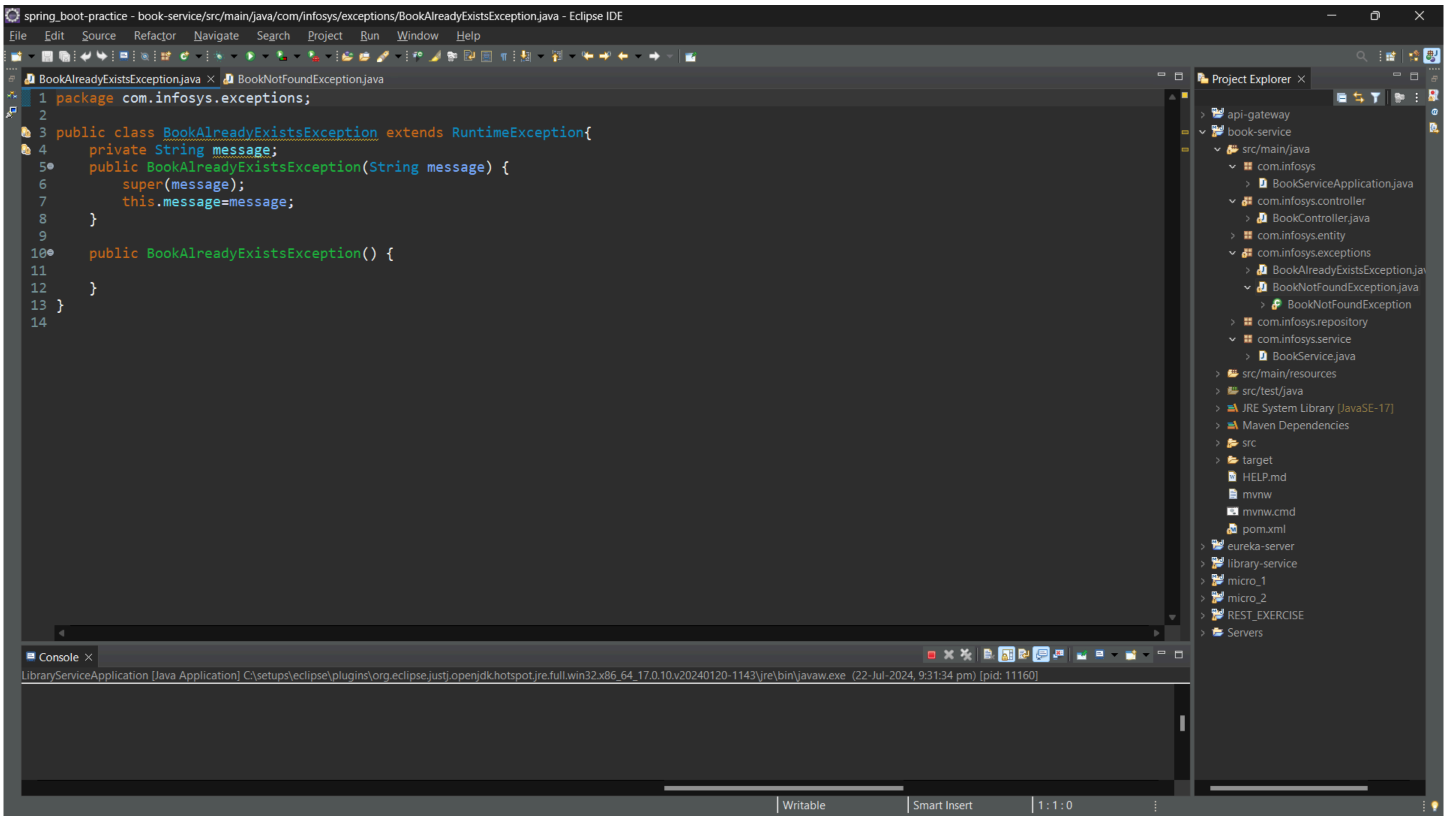
Response body:

```
{
    "book_id": 1,
    "book_name": "new Book",
    "author_name": "Prathamesh Sonje",
    "publication_year": "2026"
}
```

Q2: Exception Handling

```
package com.infosys.exceptions;


public class BookAlreadyExistsException extends RuntimeException{
    private String message;
    public BookAlreadyExistsException(String message) {
        super(message);
        this.message=message;
    }


    public BookAlreadyExistsException() {


    }
}
```

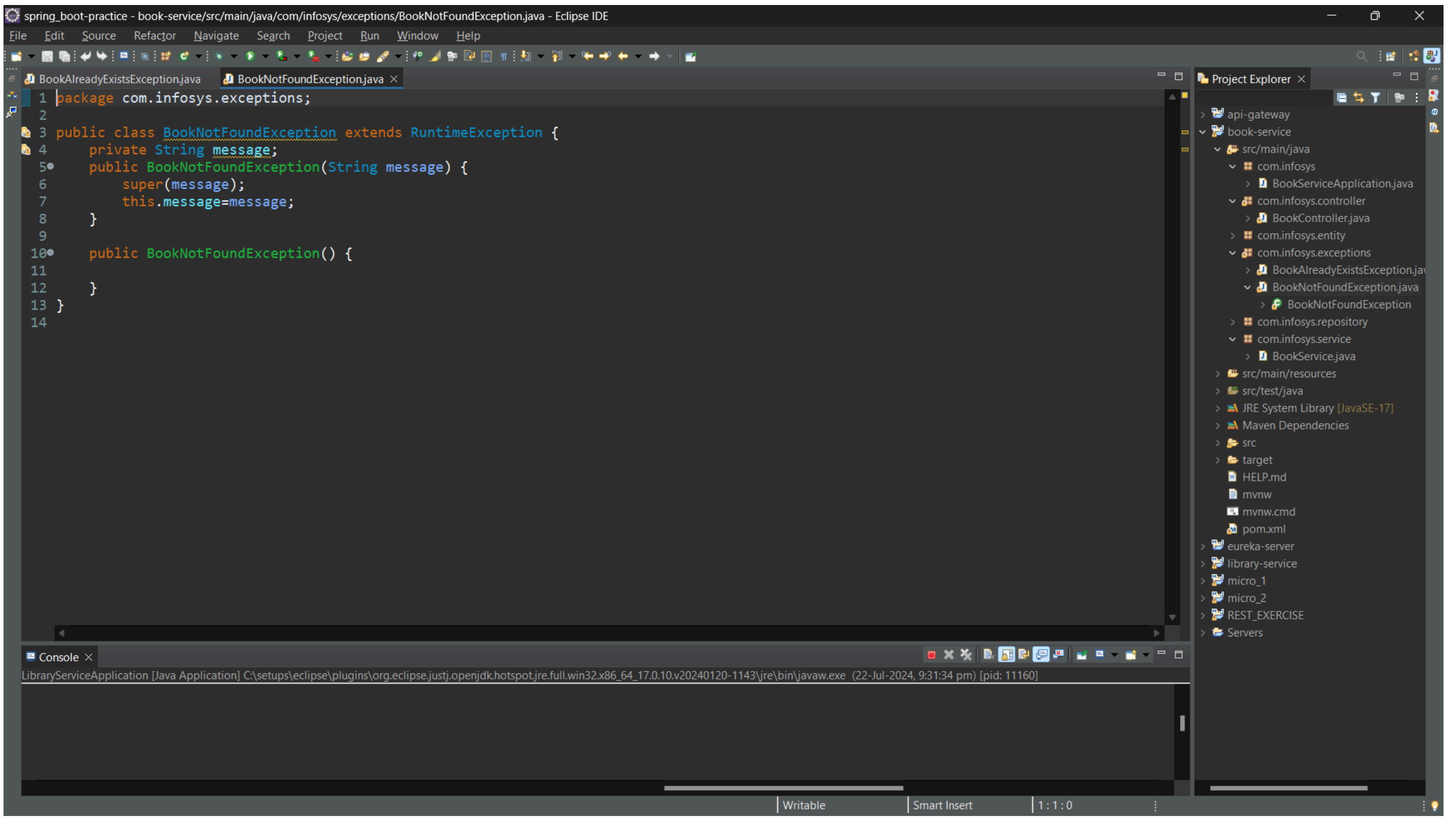
```
package com.infosys.exceptions;


public class BookNotFoundException extends RuntimeException {

    private String message;

    public BookNotFoundException(String message) {

        super(message);

        this.message=message;

    }



    public BookNotFoundException() {



    }

}
```


spring_boot-practice - book-service/src/main/java/com/infosys/exceptions/BookAlreadyExistsException.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
BookAlreadyExistsException.java
BookNotFoundException.java
Project Explorer
Console
Writable
Smart Insert
1 : 1 : 0

```
package com.infosys.exceptions;

public class BookAlreadyExistsException extends RuntimeException{
    private String message;
    public BookAlreadyExistsException(String message) {
        super(message);
        this.message=message;
    }

    public BookAlreadyExistsException() {

    }
}
```

```
api-gateway
book-service
  src/main/java
    com.infosys
      BookServiceApplication.java
    com.infosys.controller
      BookController.java
    com.infosys.entity
    com.infosys.exceptions
      BookAlreadyExistsException.ja
      BookNotFoundException.java
        BookNotFoundException
    com.infosys.repository
    com.infosys.service
      BookService.java
  src/main/resources
  src/test/java
  JRE System Library [JavaSE-17]
  Maven Dependencies
  src
  target
  HELP.md
  mvnw
  mvnw.cmd
  pom.xml
eureka-server
library-service
micro_1
micro_2
REST_EXERCISE
Servers
```

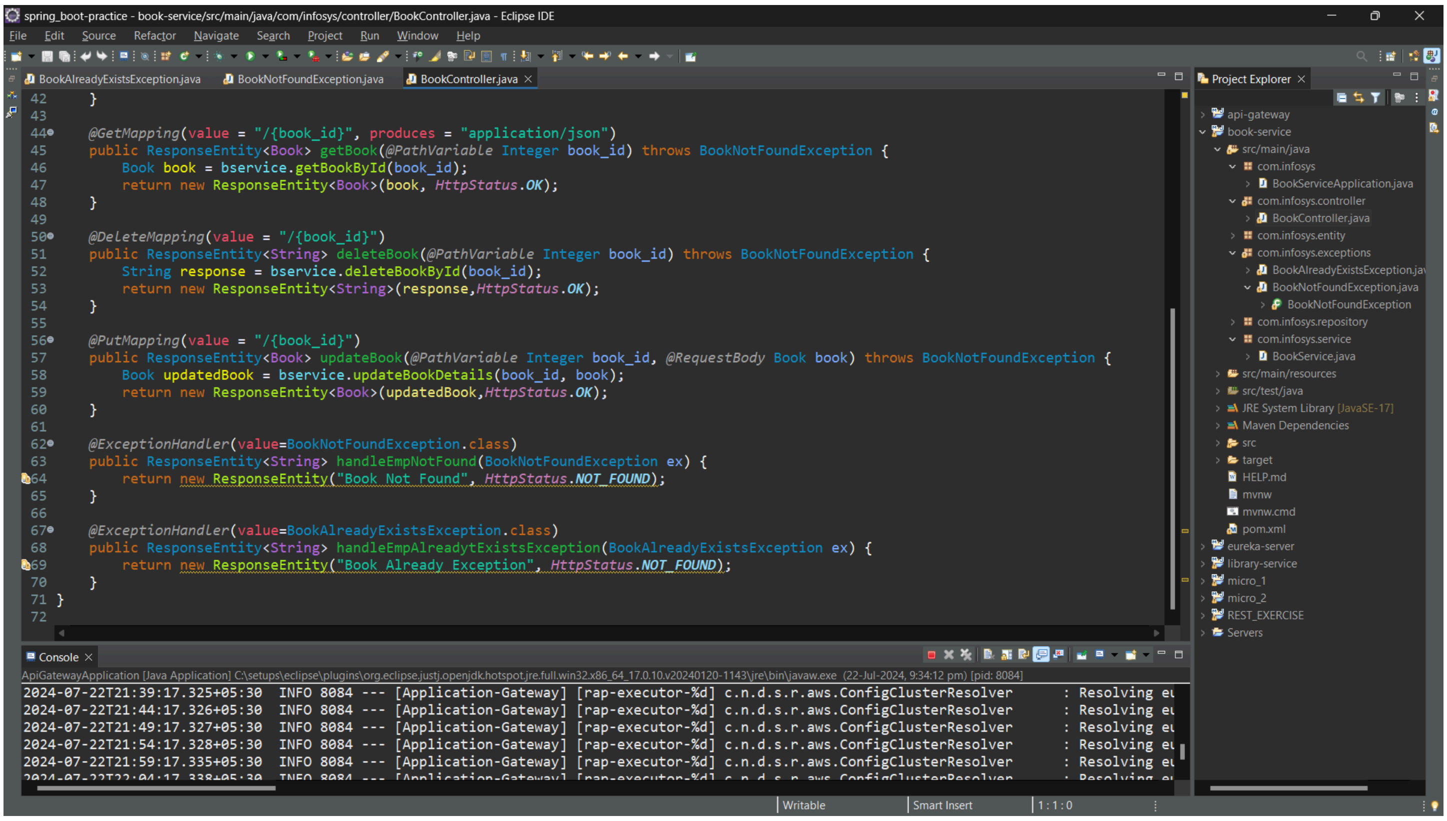
```
LibraryServiceApplication [Java Application] C:\setups\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.10.v20240120-1143\jre\bin\javaw.exe  (22-Jul-2024, 9:31:34 pm) [pid: 11160]
```



```java
package com.infosys.exceptions;

public class BookNotFoundException extends RuntimeException {
    private String message;
    public BookNotFoundException(String message) {
        super(message);
        this.message=message;
    }

    public BookNotFoundException() {

    }
}

```

```
LibraryServiceApplication [Java Application] C:\setups\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.10.v20240120-1143\jre\bin\javaw.exe  (22-Jul-2024, 9:31:34 pm) [pid: 11160]
```

```java
@ExceptionHandler(value=BookNotFoundException.class)

public ResponseEntity<String> handleEmpNotFound(BookNotFoundException ex) {

    return new ResponseEntity("Book Not Found", HttpStatus.NOT_FOUND);

}



@ExceptionHandler(value=BookAlreadyExistsException.class)

public ResponseEntity<String>
handleEmpAlreadytExistsException(BookAlreadyExistsException ex) {

    return new ResponseEntity("Book Already Exception", HttpStatus.NOT_FOUND);

}
```



spring_boot-practice - book-service/src/main/java/com/infosys/controller/BookController.java - Eclipse IDE

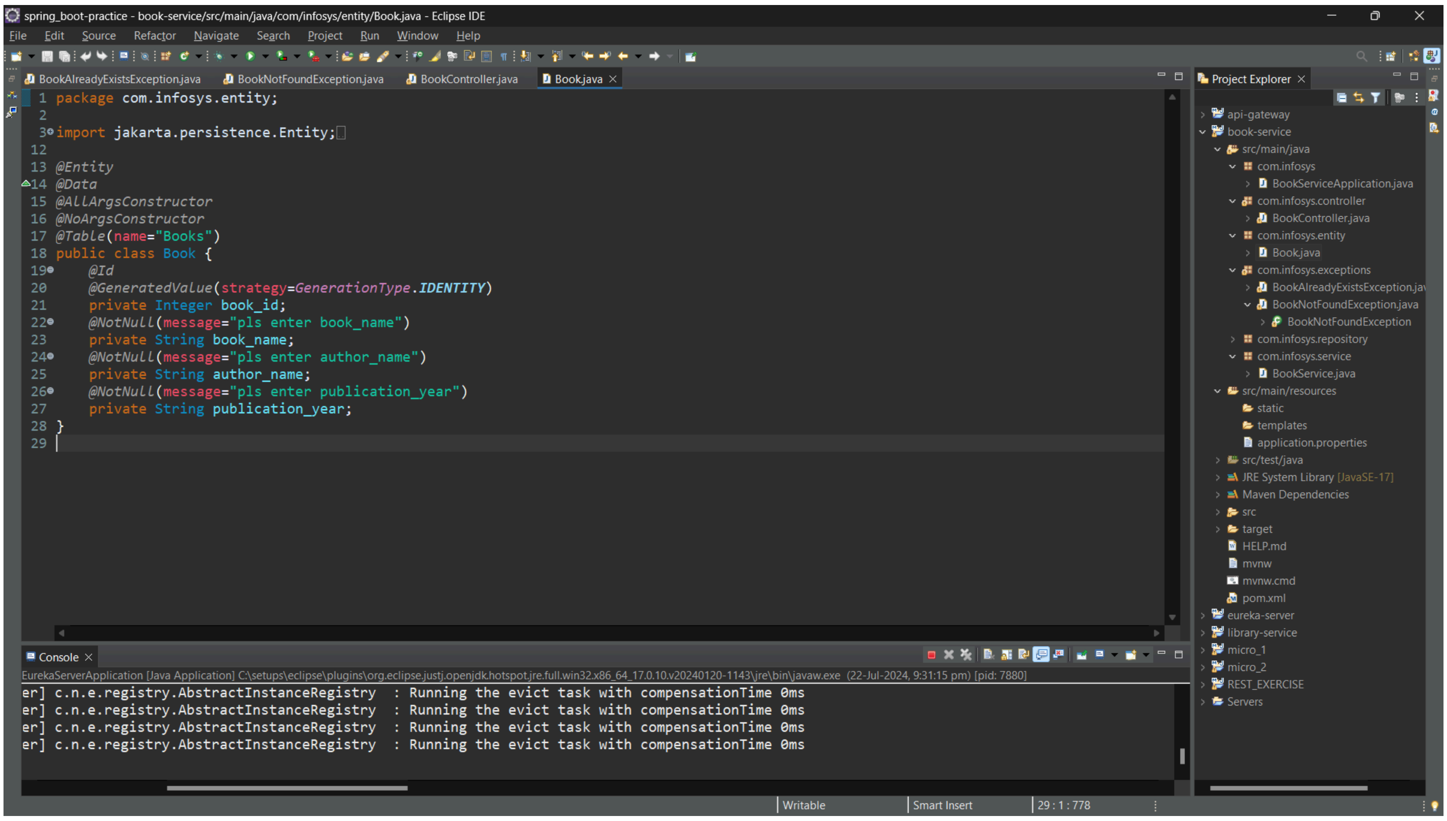
```java
    }

    @GetMapping(value = "/{book_id}", produces = "application/json")
    public ResponseEntity<Book> getBook(@PathVariable Integer book_id) throws BookNotFoundException {
        Book book = bservice.getBookById(book_id);
        return new ResponseEntity<Book>(book, HttpStatus.OK);
    }

    @DeleteMapping(value = "/{book_id}")
    public ResponseEntity<String> deleteBook(@PathVariable Integer book_id) throws BookNotFoundException {
        String response = bservice.deleteBookById(book_id);
        return new ResponseEntity<String>(response,HttpStatus.OK);
    }

    @PutMapping(value = "/{book_id}")
    public ResponseEntity<Book> updateBook(@PathVariable Integer book_id, @RequestBody Book book) throws BookNotFoundException {
        Book updatedBook = bservice.updateBookDetails(book_id, book);
        return new ResponseEntity<Book>(updatedBook,HttpStatus.OK);
    }

    @ExceptionHandler(value=BookNotFoundException.class)
    public ResponseEntity<String> handleEmpNotFound(BookNotFoundException ex) {
        return new ResponseEntity("Book Not Found", HttpStatus.NOT_FOUND);
    }

    @ExceptionHandler(value=BookAlreadyExistsException.class)
    public ResponseEntity<String> handleEmpAlreadytExistsException(BookAlreadyExistsException ex) {
        return new ResponseEntity("Book Already Exception", HttpStatus.NOT_FOUND);
    }
}
```

ApiGatewayApplication [Java Application] C:\setups\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.10.v20240120-1143\jre\bin\javaw.exe (22-Jul-2024, 9:34:12 pm) [pid: 8084]

```
2024-07-22T21:39:17.325+05:30  INFO 8084 --- [Application-Gateway] [rap-executor-%d] c.n.d.s.r.aws.ConfigClusterResolver      : Resolving eu
2024-07-22T21:44:17.326+05:30  INFO 8084 --- [Application-Gateway] [rap-executor-%d] c.n.d.s.r.aws.ConfigClusterResolver      : Resolving eu
2024-07-22T21:49:17.327+05:30  INFO 8084 --- [Application-Gateway] [rap-executor-%d] c.n.d.s.r.aws.ConfigClusterResolver      : Resolving eu
2024-07-22T21:54:17.328+05:30  INFO 8084 --- [Application-Gateway] [rap-executor-%d] c.n.d.s.r.aws.ConfigClusterResolver      : Resolving eu
2024-07-22T21:59:17.335+05:30  INFO 8084 --- [Application-Gateway] [rap-executor-%d] c.n.d.s.r.aws.ConfigClusterResolver      : Resolving eu
```

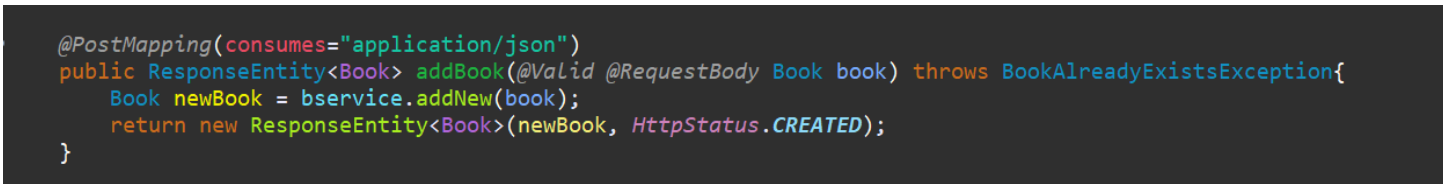
3) validate

```java
    @PostMapping(consumes="application/json")

    public ResponseEntity<Book> addBook(@Valid @RequestBody Book book) throws
BookAlreadyExistsException{

        Book newBook = bservice.addNew(book);

        return new ResponseEntity<Book>(newBook, HttpStatus.CREATED);

    }
```



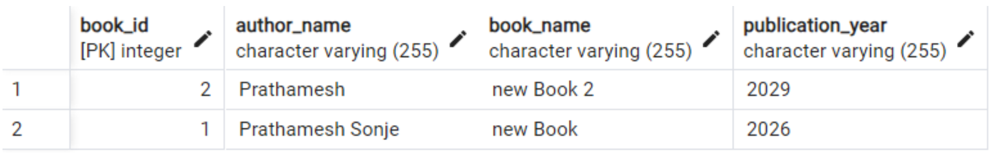
```java
package com.infosys.entity;

import jakarta.persistence.Entity;

@Entity
@Data
@AllArgsConstructor
@NoArgsConstructor
@Table(name="Books")
public class Book {
    @Id
    @GeneratedValue(strategy=GenerationType.IDENTITY)
    private Integer book_id;
    @NotNull(message="pls enter book_name")
    private String book_name;
    @NotNull(message="pls enter author_name")
    private String author_name;
    @NotNull(message="pls enter publication_year")
    private String publication_year;
}

```

```
EurekaServerApplication [Java Application] C:\setups\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.10.v20240120-1143\jre\bin\javaw.exe  (22-Jul-2024, 9:31:15 pm) [pid: 7880]
er] c.n.e.registry.AbstractInstanceRegistry  : Running the evict task with compensationTime 0ms
er] c.n.e.registry.AbstractInstanceRegistry  : Running the evict task with compensationTime 0ms
er] c.n.e.registry.AbstractInstanceRegistry  : Running the evict task with compensationTime 0ms
er] c.n.e.registry.AbstractInstanceRegistry  : Running the evict task with compensationTime 0ms
```



```java
    @PostMapping(consumes="application/json")
    public ResponseEntity<Book> addBook(@Valid @RequestBody Book book) throws BookAlreadyExistsException{
        Book newBook = bservice.addNew(book);
        return new ResponseEntity<Book>(newBook, HttpStatus.CREATED);
    }
```

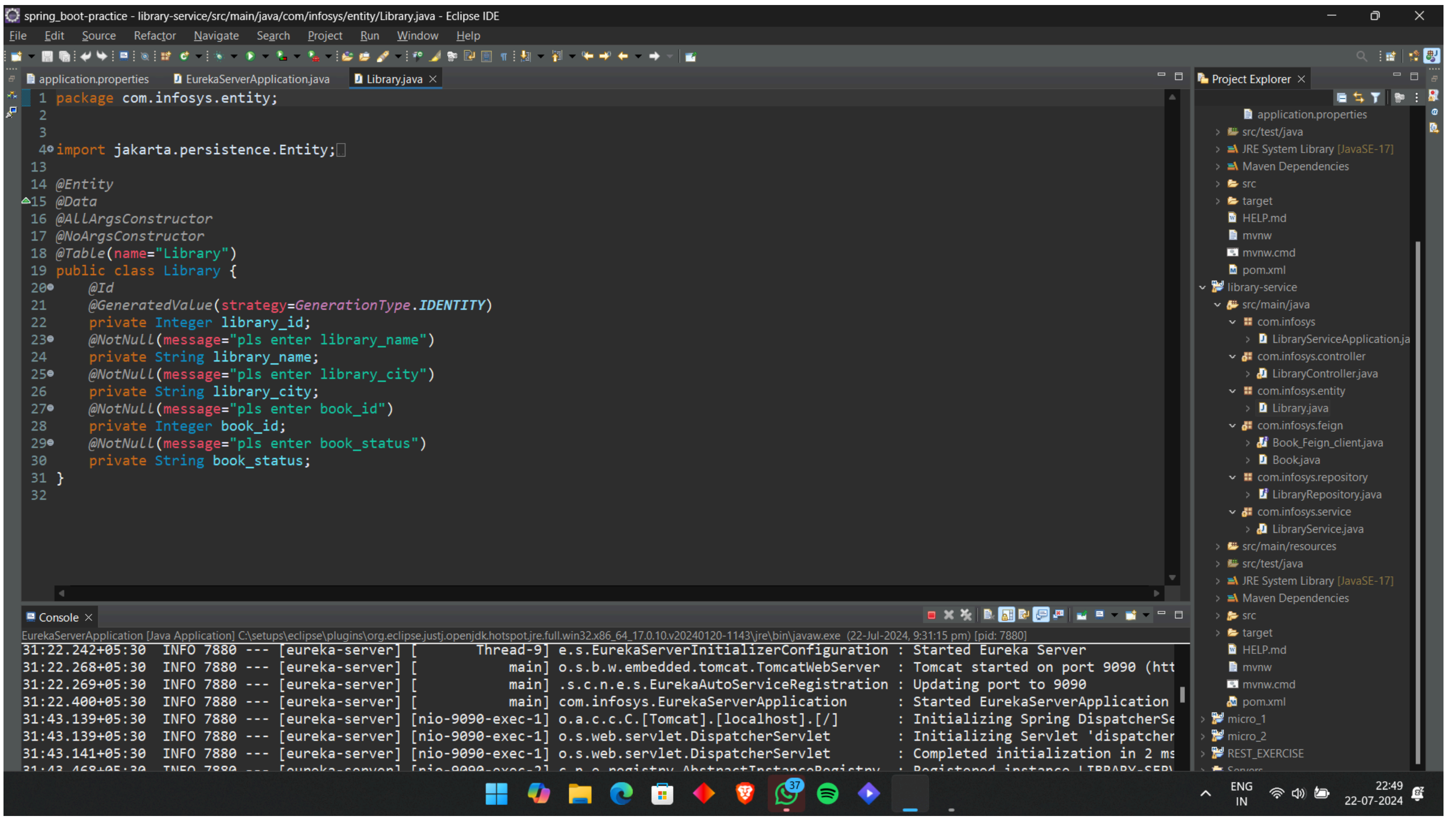
4) Database



	book_id [PK] integer	author_name character varying (255)	book_name character varying (255)	publication_year character varying (255)
1	2	Prathamesh	new Book 2	2029
2	1	Prathamesh Sonje	new Book	2026

Library-Service

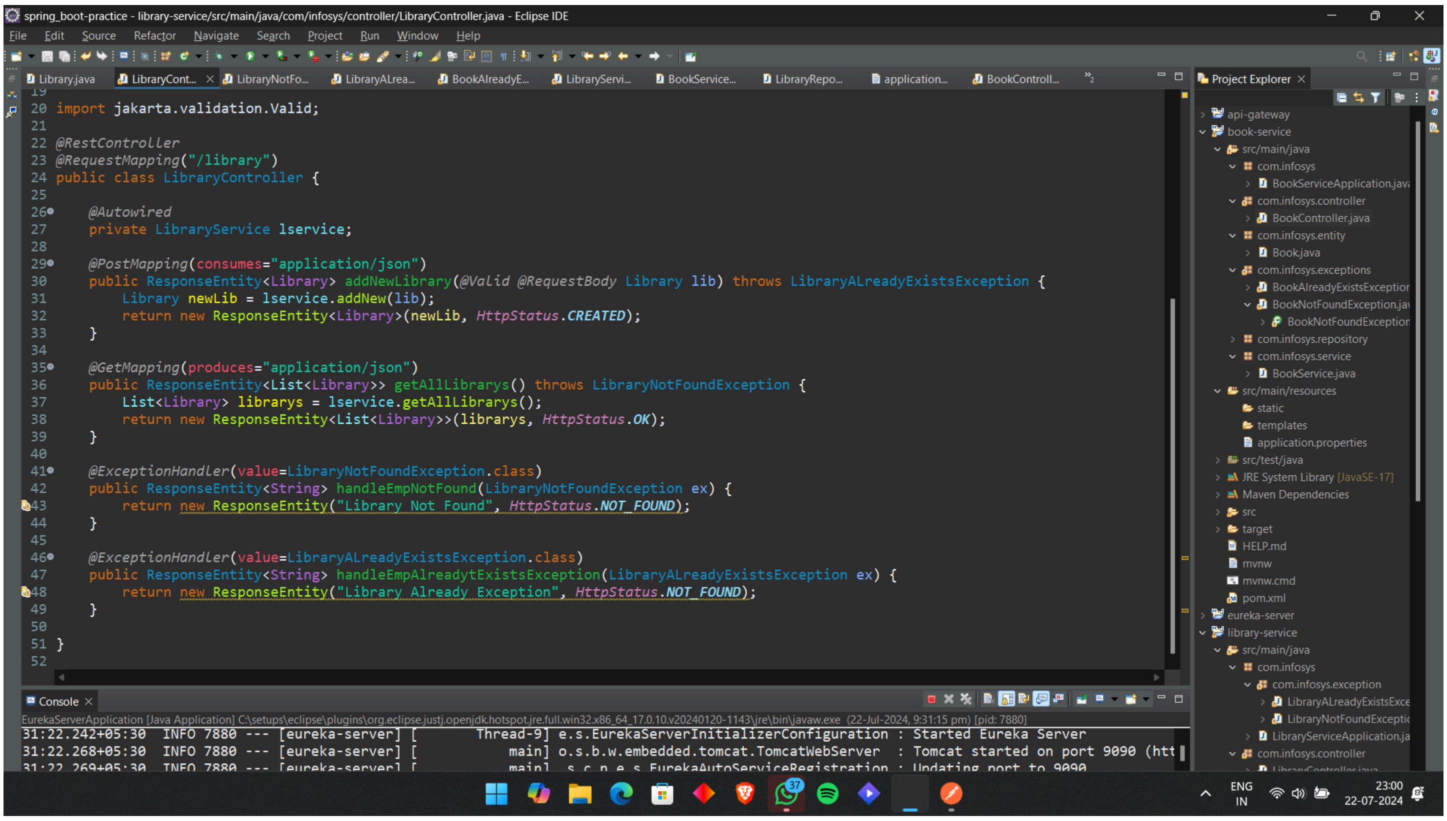
Entity-


spring_boot-practice - library-service/src/main/java/com/infosys/entity/Library.java - Eclipse IDE

```java
package com.infosys.entity;


import jakarta.persistence.Entity;

@Entity
@Data
@AllArgsConstructor
@NoArgsConstructor
@Table(name="Library")
public class Library {
    @Id
    @GeneratedValue(strategy=GenerationType.IDENTITY)
    private Integer library_id;
    @NotNull(message="pls enter library_name")
    private String library_name;
    @NotNull(message="pls enter library_city")
    private String library_city;
    @NotNull(message="pls enter book_id")
    private Integer book_id;
    @NotNull(message="pls enter book_status")
    private String book_status;
}
```

```
EurekaServerApplication [Java Application] C:\setups\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.10.v20240120-1143\jre\bin\javaw.exe  (22-Jul-2024, 9:31:15 pm) [pid: 7880]
31:22.242+05:30  INFO 7880 --- [eureka-server] [       Thread-9] e.s.EurekaServerInitializerConfiguration : Started Eureka Server
31:22.268+05:30  INFO 7880 --- [eureka-server] [           main] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat started on port 9090 (htt
31:22.269+05:30  INFO 7880 --- [eureka-server] [           main] .s.c.n.e.s.EurekaAutoServiceRegistration : Updating port to 9090
31:22.400+05:30  INFO 7880 --- [eureka-server] [           main] com.infosys.EurekaServerApplication      : Started EurekaServerApplication
31:43.139+05:30  INFO 7880 --- [eureka-server] [nio-9090-exec-1] o.a.c.c.C.[Tomcat].[localhost].[/]       : Initializing Spring DispatcherSe
31:43.139+05:30  INFO 7880 --- [eureka-server] [nio-9090-exec-1] o.s.web.servlet.DispatcherServlet        : Initializing Servlet 'dispatcher
31:43.141+05:30  INFO 7880 --- [eureka-server] [nio-9090-exec-1] o.s.web.servlet.DispatcherServlet        : Completed initialization in 2 ms
```

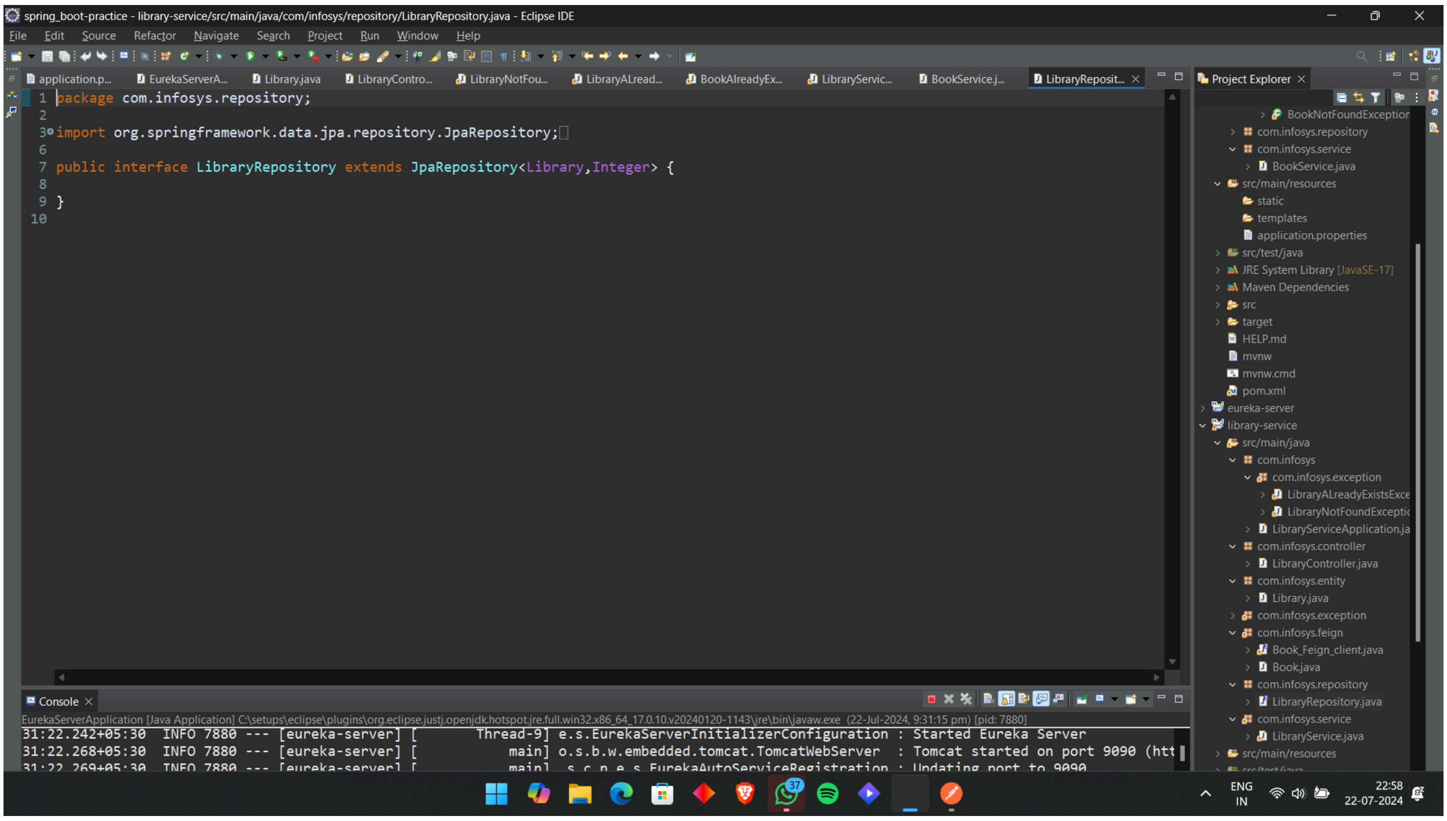
Controller-


spring_boot-practice - library-service/src/main/java/com/infosys/controller/LibraryController.java - Eclipse IDE

```java
import jakarta.validation.Valid;

@RestController
@RequestMapping("/library")
public class LibraryController {

    @Autowired
    private LibraryService lservice;

    @PostMapping(consumes="application/json")
    public ResponseEntity<Library> addNewLibrary(@Valid @RequestBody Library lib) throws LibraryALreadyExistsException {
        Library newLib = lservice.addNew(lib);
        return new ResponseEntity<Library>(newLib, HttpStatus.CREATED);
    }

    @GetMapping(produces="application/json")
    public ResponseEntity<List<Library>> getAllLibrarys() throws LibraryNotFoundException {
        List<Library> librarys = lservice.getAllLibrarys();
        return new ResponseEntity<List<Library>>(librarys, HttpStatus.OK);
    }

    @ExceptionHandler(value=LibraryNotFoundException.class)
    public ResponseEntity<String> handleEmpNotFound(LibraryNotFoundException ex) {
        return new ResponseEntity("Library Not Found", HttpStatus.NOT_FOUND);
    }

    @ExceptionHandler(value=LibraryALreadyExistsException.class)
    public ResponseEntity<String> handleEmpAlreadytExistsException(LibraryALreadyExistsException ex) {
        return new ResponseEntity("Library Already Exception", HttpStatus.NOT_FOUND);
    }

}
```

```
EurekaServerApplication [Java Application] C:\setups\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.10.v20240120-1143\jre\bin\javaw.exe  (22-Jul-2024, 9:31:15 pm) [pid: 7880]
31:22.242+05:30  INFO 7880 --- [eureka-server] [       Thread-9] e.s.EurekaServerInitializerConfiguration : Started Eureka Server
31:22.268+05:30  INFO 7880 --- [eureka-server] [           main] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat started on port 9090 (htt
```

repository-

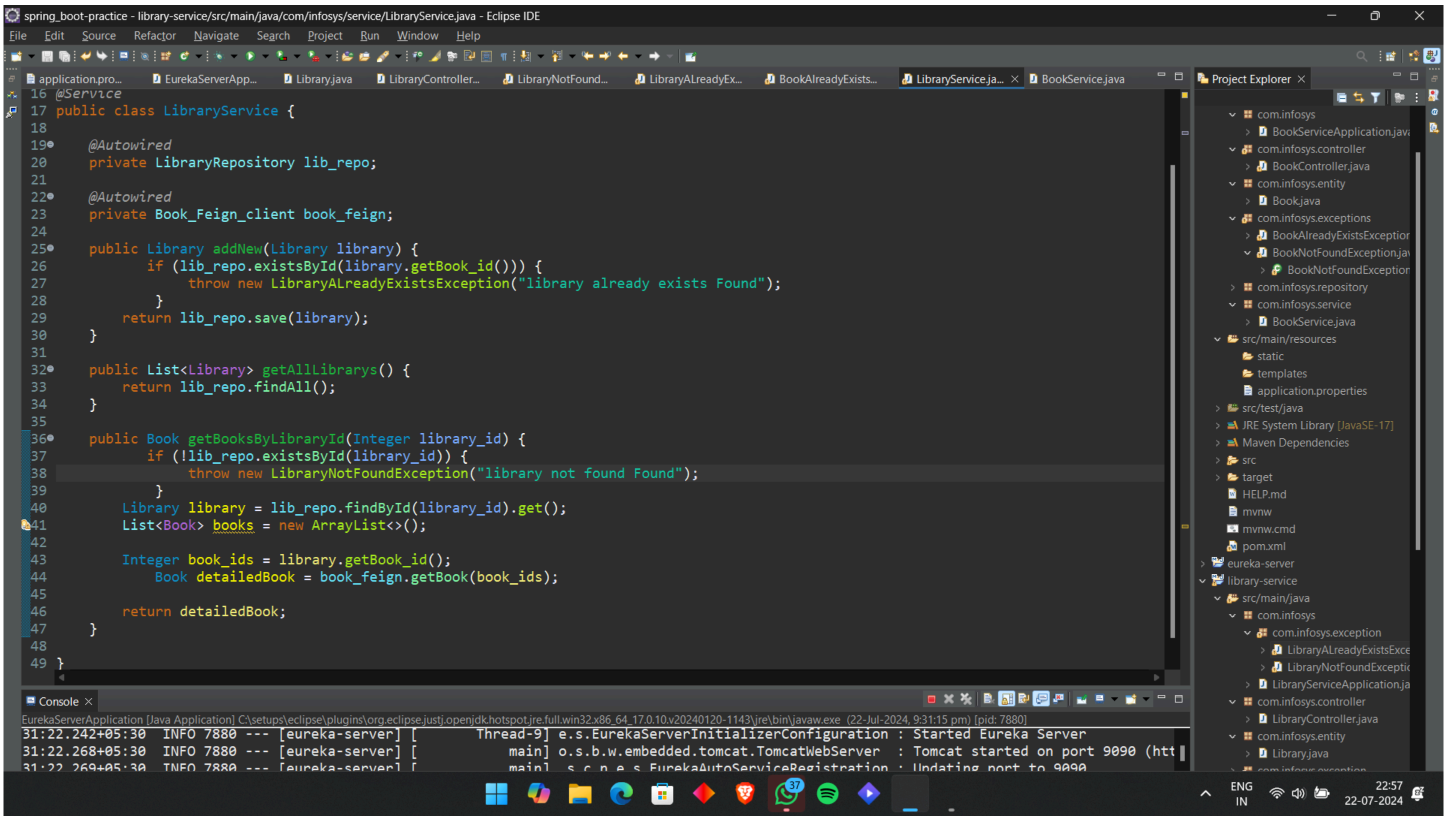


spring_boot-practice - library-service/src/main/java/com/infosys/repository/LibraryRepository.java - Eclipse IDE

```java
package com.infosys.repository;

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

public interface LibraryRepository extends JpaRepository<Library,Integer> {

}
```

```
EurekaServerApplication [Java Application] C:\setups\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.10.v20240120-1143\jre\bin\javaw.exe  (22-Jul-2024, 9:31:15 pm) [pid: 7880]
31:22.242+05:30  INFO 7880 --- [eureka-server] [           Thread-9] o.s.e.s.EurekaServerInitializerConfiguration : Started Eureka Server
31:22.268+05:30  INFO 7880 --- [eureka-server] [               main] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat started on port 9090 (htt
31:22.269+05:30  INFO 7880 --- [eureka-server] [               main] s.c.n.e.s.EurekaAutoServiceRegistration : Updating port to 9090
```

service -

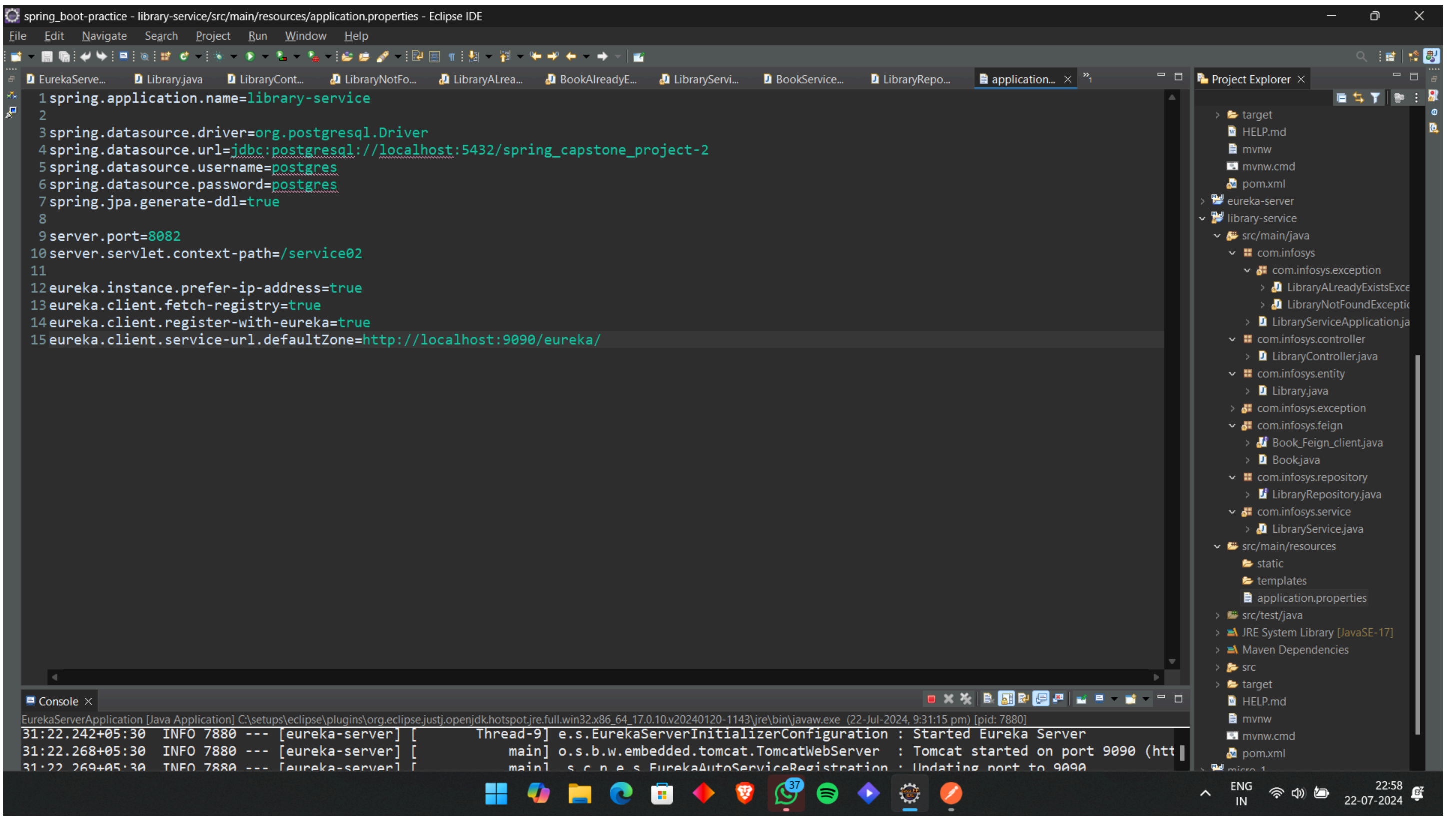


spring_boot-practice - library-service/src/main/java/com/infosys/service/LibraryService.java - Eclipse IDE

```java
@Service
public class LibraryService {

    @Autowired
    private LibraryRepository lib_repo;

    @Autowired
    private Book_Feign_client book_feign;

    public Library addNew(Library library) {
        if (lib_repo.existsById(library.getBook_id())) {
            throw new LibraryALreadyExistsException("library already exists Found");
        }
        return lib_repo.save(library);
    }

    public List<Library> getAllLibrarys() {
        return lib_repo.findAll();
    }

    public Book getBooksByLibraryId(Integer library_id) {
        if (!lib_repo.existsById(library_id)) {
            throw new LibraryNotFoundException("library not found Found");
        }
        Library library = lib_repo.findById(library_id).get();
        List<Book> books = new ArrayList<>();

        Integer book_ids = library.getBook_id();
            Book detailedBook = book_feign.getBook(book_ids);

        return detailedBook;
    }

}
```

```
EurekaServerApplication [Java Application] C:\setups\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.10.v20240120-1143\jre\bin\javaw.exe  (22-Jul-2024, 9:31:15 pm) [pid: 7880]
31:22.242+05:30  INFO 7880 --- [eureka-server] [           Thread-9] o.s.e.s.EurekaServerInitializerConfiguration : Started Eureka Server
31:22.268+05:30  INFO 7880 --- [eureka-server] [               main] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat started on port 9090 (htt
31:22.269+05:30  INFO 7880 --- [eureka-server] [               main] s.c.n.e.s.EurekaAutoServiceRegistration : Updating port to 9090
```

application - properties -



```
spring.application.name=library-service

spring.datasource.driver=org.postgresql.Driver
spring.datasource.url=jdbc:postgresql://localhost:5432/spring_capstone_project-2
spring.datasource.username=postgres
spring.datasource.password=postgres
spring.jpa.generate-ddl=true

server.port=8082
server.servlet.context-path=/service02

eureka.instance.prefer-ip-address=true
eureka.client.fetch-registry=true
eureka.client.register-with-eureka=true
eureka.client.service-url.defaultZone=http://localhost:9090/eureka/
```

```
EurekaServerApplication [Java Application] C:\setups\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.10.v20240120-1143\jre\bin\javaw.exe  (22-Jul-2024, 9:31:15 pm) [pid: 7880]
31:22.242+05:30  INFO 7880 --- [eureka-server] [       Thread-9] e.s.EurekaServerInitializerConfiguration : Started Eureka Server
31:22.268+05:30  INFO 7880 --- [eureka-server] [           main] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat started on port 9090 (htt
31:22.269+05:30  INFO 7880 --- [eureka-server] [           main] s.c.n.e.s.EurekaAutoServiceRegistration : Updating port to 9090
```

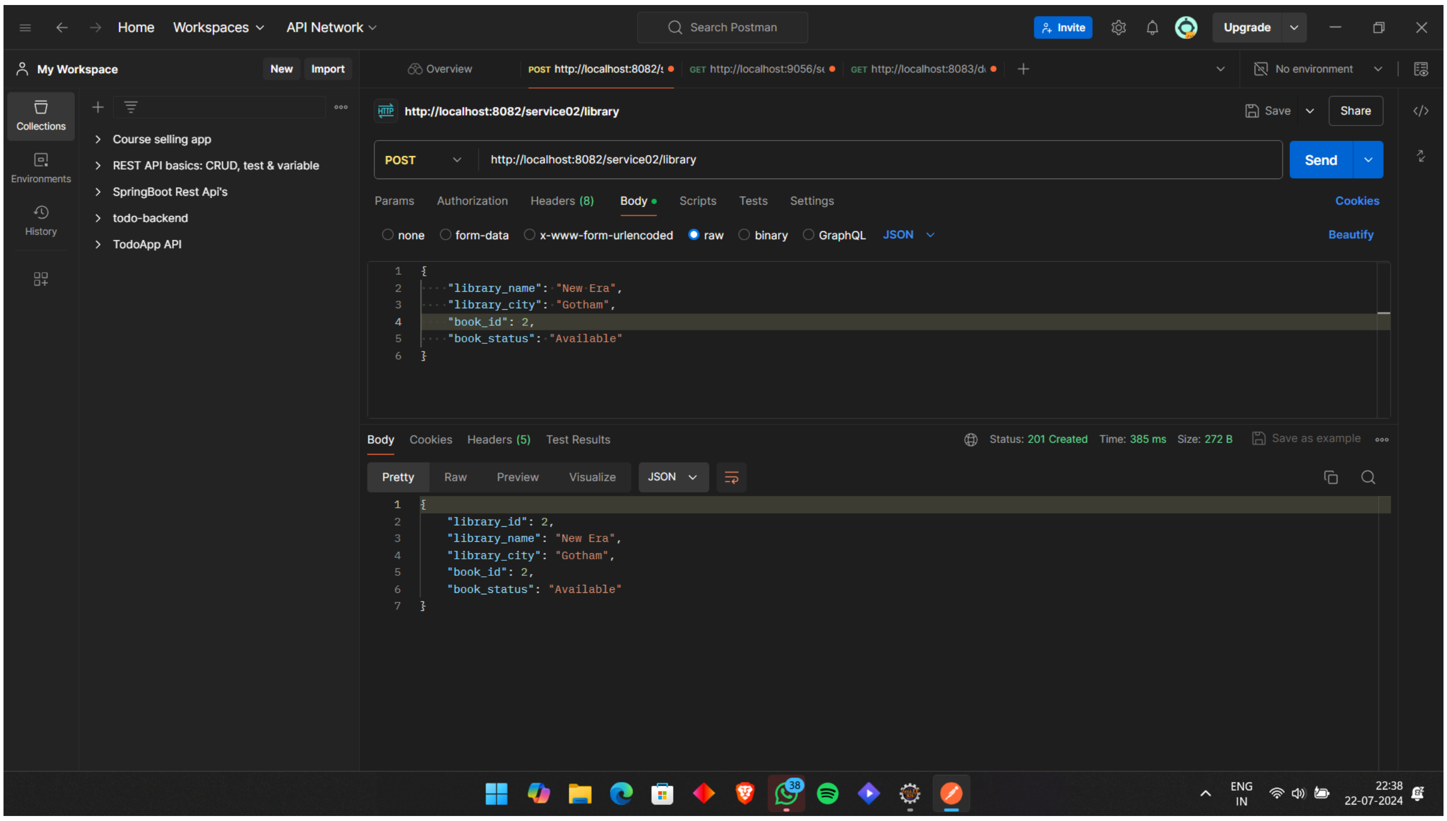
Q) create endpoints for library service

Add new Library -

code -

```
@PostMapping(consumes="application/json")
public ResponseEntity<Library> addNewLibrary(@Valid @RequestBody Library lib) {
    Library newLib = lservice.addNew(lib);
    return new ResponseEntity<Library>(newLib, HttpStatus.CREATED);
}
```

output -



http://localhost:8082/service02/library

POST http://localhost:8082/service02/library

Body: none / form-data / x-www-form-urlencoded / [✓ raw] / binary / GraphQL — JSON

```
{
    "library_name": "New Era",
    "library_city": "Gotham",
    "book_id": 2,
    "book_status": "Available"
}
```

Body Cookies Headers (5) Test Results   Status: 201 Created Time: 385 ms Size: 272 B

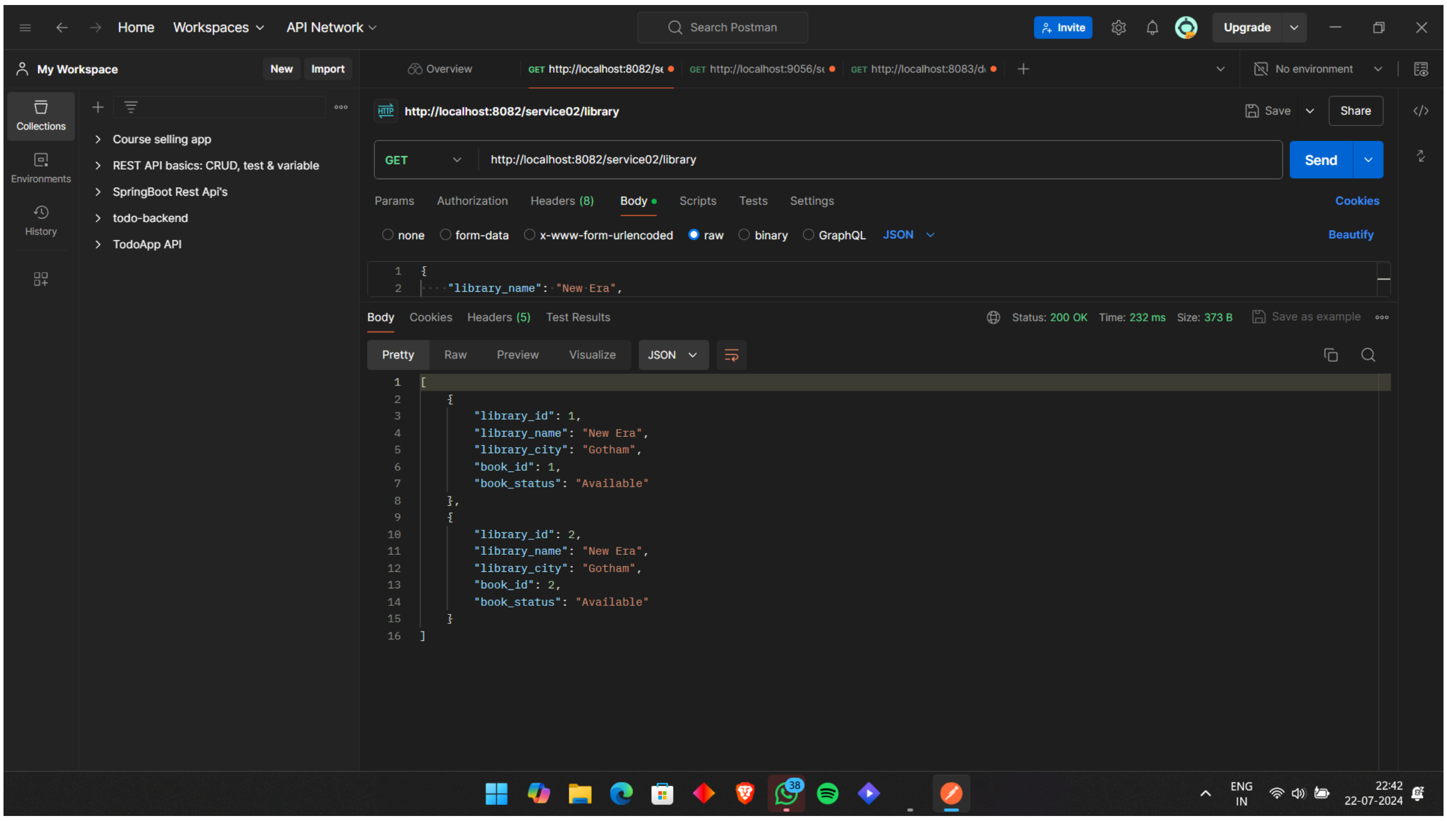
```
{
    "library_id": 2,
    "library_name": "New Era",
    "library_city": "Gotham",
    "book_id": 2,
    "book_status": "Available"
}
```

ENG IN 22:38 22-07-2024

Show all library details

output -

```
@GetMapping(produces="application/json")
public ResponseEntity<List<Library>> getAllLibrarys() {
    List<Library> librarys = lservice.getAllLibrarys();
    return new ResponseEntity<List<Library>>(librarys, HttpStatus.OK);
}
```


GET http://localhost:8082/service02/library

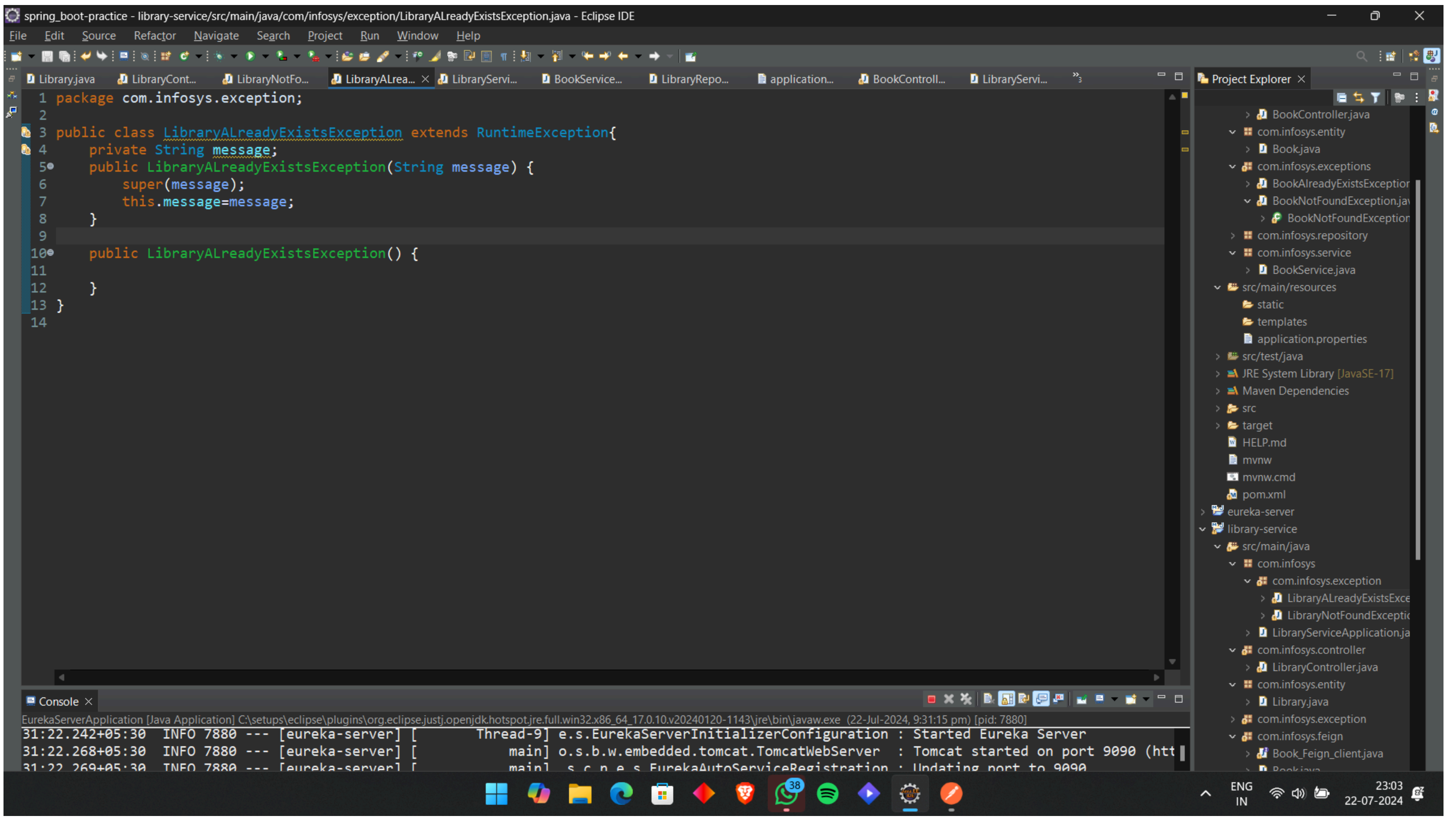
Body (raw, JSON):

```
{
    "library_name": "New Era",
```

Status: 200 OK Time: 232 ms Size: 373 B

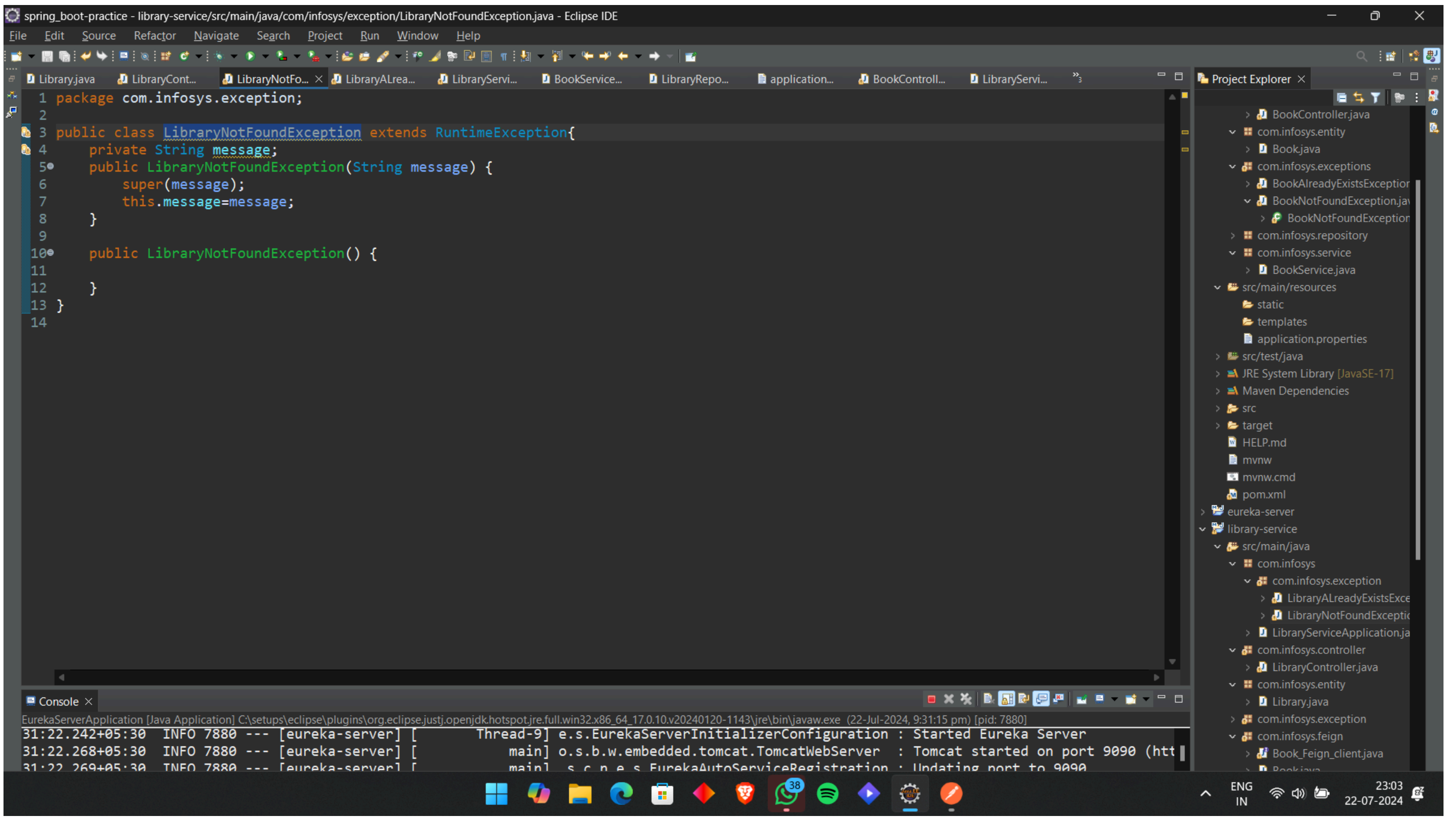
```
[
    {
        "library_id": 1,
        "library_name": "New Era",
        "library_city": "Gotham",
        "book_id": 1,
        "book_status": "Available"
    },
    {
        "library_id": 2,
        "library_name": "New Era",
        "library_city": "Gotham",
        "book_id": 2,
        "book_status": "Available"
    }
]
```

Exception Handling


spring_boot-practice - library-service/src/main/java/com/infosys/exception/LibraryALreadyExistsException.java - Eclipse IDE

```java
package com.infosys.exception;

public class LibraryALreadyExistsException extends RuntimeException{
    private String message;
    public LibraryALreadyExistsException(String message) {
        super(message);
        this.message=message;
    }

    public LibraryALreadyExistsException() {

    }
}
```

```
EurekaServerApplication [Java Application] C:\setups\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.10.v20240120-1143\jre\bin\javaw.exe  (22-Jul-2024, 9:31:15 pm) [pid: 7880]
31:22.242+05:30  INFO 7880 --- [eureka-server] [       Thread-9] e.s.EurekaServerInitializerConfiguration : Started Eureka Server
31:22.268+05:30  INFO 7880 --- [eureka-server] [           main] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat started on port 9090 (htt
31:22.269+05:30  INFO 7880 --- [eureka-server] [           main] s.c.n.e.s.EurekaAutoServiceRegistration : Updating port to 9090
```



spring_boot-practice - library-service/src/main/java/com/infosys/exception/LibraryNotFoundException.java - Eclipse IDE

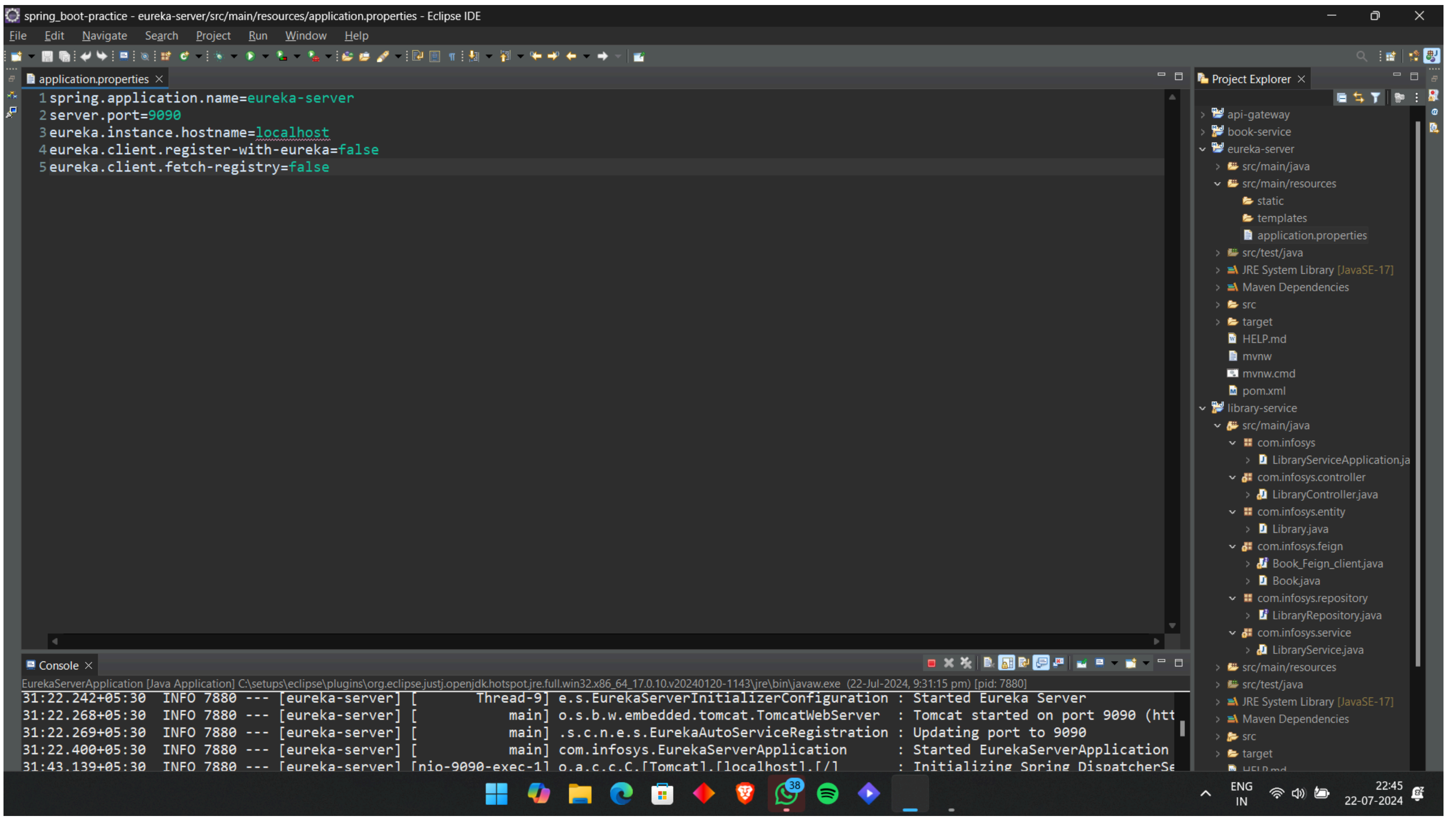
```java
package com.infosys.exception;

public class LibraryNotFoundException extends RuntimeException{
    private String message;
    public LibraryNotFoundException(String message) {
        super(message);
        this.message=message;
    }

    public LibraryNotFoundException() {

    }
}

```

```
EurekaServerApplication [Java Application] C:\setups\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.10.v20240120-1143\jre\bin\javaw.exe  (22-Jul-2024, 9:31:15 pm) [pid: 7880]
31:22.242+05:30  INFO 7880 --- [eureka-server] [       Thread-9] e.s.EurekaServerInitializerConfiguration : Started Eureka Server
31:22.268+05:30  INFO 7880 --- [eureka-server] [           main] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat started on port 9090 (htt
31:22.269+05:30  INFO 7880 --- [eureka-server] [           main] .s.c.n.e.s.EurekaAutoServiceRegistration : Updating port to 9090
```

Q4) Create Eureka Service Q5) eureka clients

Eureka-server

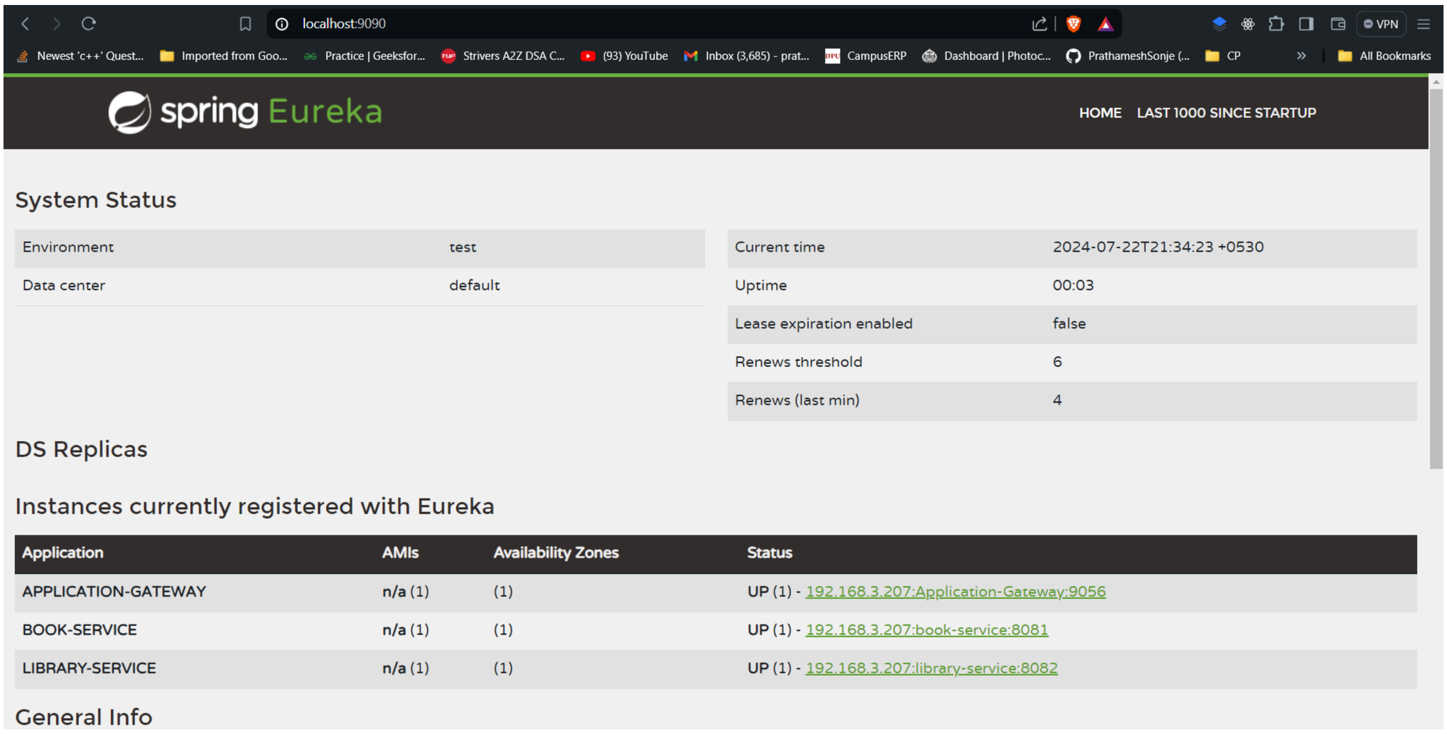
application properties



spring_boot-practice - eureka-server/src/main/resources/application.properties - Eclipse IDE

```
spring.application.name=eureka-server
server.port=9090
eureka.instance.hostname=localhost
eureka.client.register-with-eureka=false
eureka.client.fetch-registry=false
```

```
EurekaServerApplication [Java Application] C:\setups\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.10.v20240120-1143\jre\bin\javaw.exe  (22-Jul-2024, 9:31:15 pm) [pid: 7880]
31:22.242+05:30  INFO 7880 --- [eureka-server] [       Thread-9] e.s.EurekaServerInitializerConfiguration : Started Eureka Server
31:22.268+05:30  INFO 7880 --- [eureka-server] [           main] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat started on port 9090 (htt
31:22.269+05:30  INFO 7880 --- [eureka-server] [           main] .s.c.n.e.s.EurekaAutoServiceRegistration : Updating port to 9090
31:22.400+05:30  INFO 7880 --- [eureka-server] [           main] com.infosys.EurekaServerApplication      : Started EurekaServerApplication
31:43.139+05:30  INFO 7880 --- [eureka-server] [nio-9090-exec-1] o.a.c.c.C.[Tomcat].[localhost].[/]       : Initializing Spring DispatcherSe
```


localhost:9090
Newest 'c++' Quest... Imported from Goo... Practice | Geeksfor... Strivers A2Z DSA C... (93) YouTube Inbox (3,685) - prat... CampusERP Dashboard | Photoc... PrathameshSonje (... CP All Bookmarks
spring Eureka
HOME
LAST 1000 SINCE STARTUP

System Status

Environment	test
Data center	default

Current time	2024-07-22T21:34:23 +0530
Uptime	00:03
Lease expiration enabled	false
Renews threshold	6
Renews (last min)	4

DS Replicas

Instances currently registered with Eureka

Application	AMIs	Availability Zones	Status
APPLICATION-GATEWAY	n/a (1)	(1)	UP (1) - 192.168.3.207:Application-Gateway:9056
BOOK-SERVICE	n/a (1)	(1)	UP (1) - 192.168.3.207:book-service:8081
LIBRARY-SERVICE	n/a (1)	(1)	UP (1) - 192.168.3.207:library-service:8082

General Info

book- service

main-

```
package com.infosys;


import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.cloud.client.discovery.EnableDiscoveryClient;
import org.springframework.cloud.openfeign.EnableFeignClients;


@SpringBootApplication
@EnableDiscoveryClient
@EnableFeignClients
public class BookServiceApplication {


    public static void main(String[] args) {
        SpringApplication.run(BookServiceApplication.class, args);
    }


}
```

application-properties-

```
spring.application.name=book-service

spring.datasource.driver=org.postgresql.Driver
spring.datasource.url=jdbc:postgresql://localhost:5432/spring_capstone_project-2
spring.datasource.username=postgres
spring.datasource.password=postgres
spring.jpa.generate-ddl=true

server.port=8081
server.servlet.context-path=/service01

eureka.instance.prefer-ip-address=true
eureka.client.fetch-registry=true
eureka.client.register-with-eureka=true
eureka.client.service-url.defaultZone=http://localhost:9090/eureka/
```

library-service

main-

```
package com.infosys;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.cloud.client.discovery.EnableDiscoveryClient;

import org.springframework.cloud.openfeign.EnableFeignClients;


@SpringBootApplication

@EnableDiscoveryClient

@EnableFeignClients

public class LibraryServiceApplication {


    public static void main(String[] args) {

        SpringApplication.run(LibraryServiceApplication.class, args);

    }


}
```

application-properties

```
spring.application.name=library-service


spring.datasource.driver=org.postgresql.Driver

spring.datasource.url=jdbc:postgresql://localhost:5432/spring_capstone_project-2

spring.datasource.username=postgres

spring.datasource.password=postgres

spring.jpa.generate-ddl=true


server.port=8082

server.servlet.context-path=/service02
```

```
eureka.instance.prefer-ip-address=true

eureka.client.fetch-registry=true

eureka.client.register-with-eureka=true

eureka.client.service-url.defaultZone=http://localhost:9090/eureka/
```