

# Book management API

## Source Code

### 1. BookInformationAPI (Main Class)

```
package com.example.demo;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.context.annotation.ComponentScan;

@SpringBootApplication
@ComponentScan(value="com.example.demo.*")
public class BookInformationAPI {

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

### 2 . BookEntity (Class)

Spring Boot Book Details Project - Simple Version

```
package com.example.demo.entity;

import jakarta.persistence.*;
import lombok.Data;

@Entity
@Table(name = "BookDetails")
@Data
public class BookEntity {

    @Id
```

```
@Column(name = "BOOKID")
private int bookId;

@Column(name = "BOOKNAME")
private String bookName;

@Column(name = "BOOKAUTHOR")
private String bookAuthor;

@Column(name = "BOOKPRICE")
private int bookPrice;

@Column(name = "PUBLISHEDBY")
private String publishedBy;
}
```

### **3. BookRepository.java (Interface)**

```
package com.example.demo.repo;

import org.springframework.data.jpa.repository.JpaRepository;
import com.example.demo.entity.BookEntity;

public interface BookRepository extends JpaRepository<BookEntity, Integer> {
}
```

### **4. Error Message (Class)**

```
package com.example.demo.errormessage;

public class ErrorMessage {
    private String errorMessage;

    public String getErrorMessage() {
        return errorMessage;
    }

    public void setErrorMessage(String errorMessage) {
        this.errorMessage = errorMessage;
    }
}
```

## **5. BookDetailsService (Interface)**

```
package com.example.demo.service;

import com.example.demo.entity.BookEntity;
import com.example.demo.errormessage.ErrorMessage;

public interface BookDetailsService {

    BookEntity getBookDetails(int bookId);

    ErrorMessage getErrorMessage();

}
```

## **6. BookServiceImp (Class)**

```
package com.example.demo.service;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;
import com.example.demo.entity.BookEntity;
import com.example.demo.errormessage.ErrorMessage;
import com.example.demo.repo.BookRepository;

@Service

public class BookServiceImp implements BookDetailsService {

    @Autowired

    private BookRepository bookRepository;

    public BookEntity getBookDetails(int bookId) {

        return bookRepository.findById(bookId).get();

    }

    public ErrorMessage getErrorMessage() {

        ErrorMessage error = new ErrorMessage();

        error.setErrorMessage("Invalid Book ID");

        return error;

    }

}
```

## **7. BookDetailsController (Class)**

```
package com.example.demo.controller;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.HttpStatus;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.*;
import com.example.demo.entity.BookEntity;
import com.example.demo.errormessage.ErrorMessage;
import com.example.demo.service.BookDetailsService;

@RestController

public class BookDetailsController {

    @Autowired

    private BookDetailsService service;

    @GetMapping("/getBookDetails/{bookId}")

    public ResponseEntity<?> getBookDetails(@PathVariable int bookId) {

        if (bookId >= 101 && bookId <= 105) {

            BookEntity details = service.getBookDetails(bookId);

            return new ResponseEntity<>(details, HttpStatus.OK);

        } else {

            ErrorMessage error = service.getErrorMessage();

            return new ResponseEntity<>(error, HttpStatus.BAD_REQUEST);

        }

    }

}
```

## **8. application.properties**

```
spring.application.name=GetMappingBookDetails

spring.datasource.url=jdbc:oracle:thin:@//localhost:1521/xe
```

```
spring.datasource.username=hr
spring.datasource.password=9360mah
spring.datasource.driver-class-name=oracle.jdbc.driver.OracleDriver
server.port=8093
```

## **9. pom.xml**

```
<dependencies>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web</artifactId>
  </dependency>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-data-jpa</artifactId>
  </dependency>
  <dependency>
    <groupId>com.oracle.database.jdbc</groupId>
    <artifactId>ojdbc8</artifactId>
    <version>19.8.0.0</version>
  </dependency>
  <dependency>
    <groupId>org.projectlombok</groupId>
    <artifactId>lombok</artifactId>
    <scope>provided</scope>
  </dependency>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-test</artifactId>
    <scope>test</scope>
  </dependency>
```

</dependencies>