**CODE:**

package com.example.demo;

import com.example.demo.model.Book;

import com.fasterxml.jackson.databind.ObjectMapper;

import org.junit.jupiter.api.\*;

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

import org.springframework.boot.test.autoconfigure.web.servlet.AutoConfigureMockMvc;

import org.springframework.boot.test.context.SpringBootTest;

import org.springframework.test.context.ActiveProfiles;

import org.springframework.http.MediaType;

import org.springframework.test.web.servlet.MockMvc;

import static org.assertj.core.api.Assertions.assertThat;

import static org.springframework.test.web.servlet.request.MockMvcRequestBuilders.\*;

import static org.springframework.test.web.servlet.result.MockMvcResultMatchers.\*;

@SpringBootTest

@AutoConfigureMockMvc

@ActiveProfiles("test")

@TestMethodOrder(MethodOrderer.OrderAnnotation.class)

class BookControllerIntegrationTest {

@Autowired private MockMvc mvc;

@Autowired private ObjectMapper mapper;

private static Long createdId;

@Test @Order(1)

void createBook() throws Exception {

Book b = new Book("Clean Code", "Robert C. Martin");

String json = mapper.writeValueAsString(b);

String location =

mvc.perform(post("/books")

.contentType(MediaType.APPLICATION\_JSON)

.content(json))

.andExpect(status().isCreated())

.andReturn().getResponse().getHeader("Location");

assertThat(location).isNotNull();

createdId = Long.valueOf(location.substring(location.lastIndexOf('/') + 1));

}

@Test @Order(2)

void listContainsCreatedBook() throws Exception {

mvc.perform(get("/books"))

.andExpect(status().isOk())

.andExpect(jsonPath("$[0].title").value("Clean Code"));

}

@Test @Order(3)

void fetchById() throws Exception {

mvc.perform(get("/books/{id}", createdId))

.andExpect(status().isOk())

.andExpect(jsonPath("$.author").value("Robert C. Martin"));

}

}

package com.example.demo.controller;

import com.example.demo.model.Book;

import com.example.demo.service.BookService;

import org.springframework.http.ResponseEntity;

import org.springframework.web.bind.annotation.\*;

import java.net.URI;

import java.util.List;

@RestController

@RequestMapping("/books")

public class BookController {

private final BookService svc;

public BookController(BookService svc) {

this.svc = svc;

}

@PostMapping

public ResponseEntity<Book> create(@RequestBody Book book) {

Book saved = svc.save(book);

return ResponseEntity.created(URI.create("/books/" + saved.getId())).body(saved);

}

@GetMapping

public List<Book> all() {

return svc.findAll();

}

@GetMapping("/{id}")

public ResponseEntity<Book> one(@PathVariable Long id) {

Book found = svc.findById(id);

return found != null ? ResponseEntity.ok(found) : ResponseEntity.notFound().build();

}

}

package com.example.demo.model;

import jakarta.persistence.\*;

@Entity

public class Book {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private Long id;

private String title;

private String author;

protected Book() {}

public Book(String title, String author) {

this.title = title;

this.author = author;

}

public Long getId() { return id; }

public String getTitle() { return title; }

public String getAuthor() { return author; }

public void setTitle(String title) { this.title = title; }

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

}

package com.example.demo.repository;

import com.example.demo.model.Book;

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

public interface BookRepository extends JpaRepository<Book, Long> {}

package com.example.demo;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class DemoApplication {

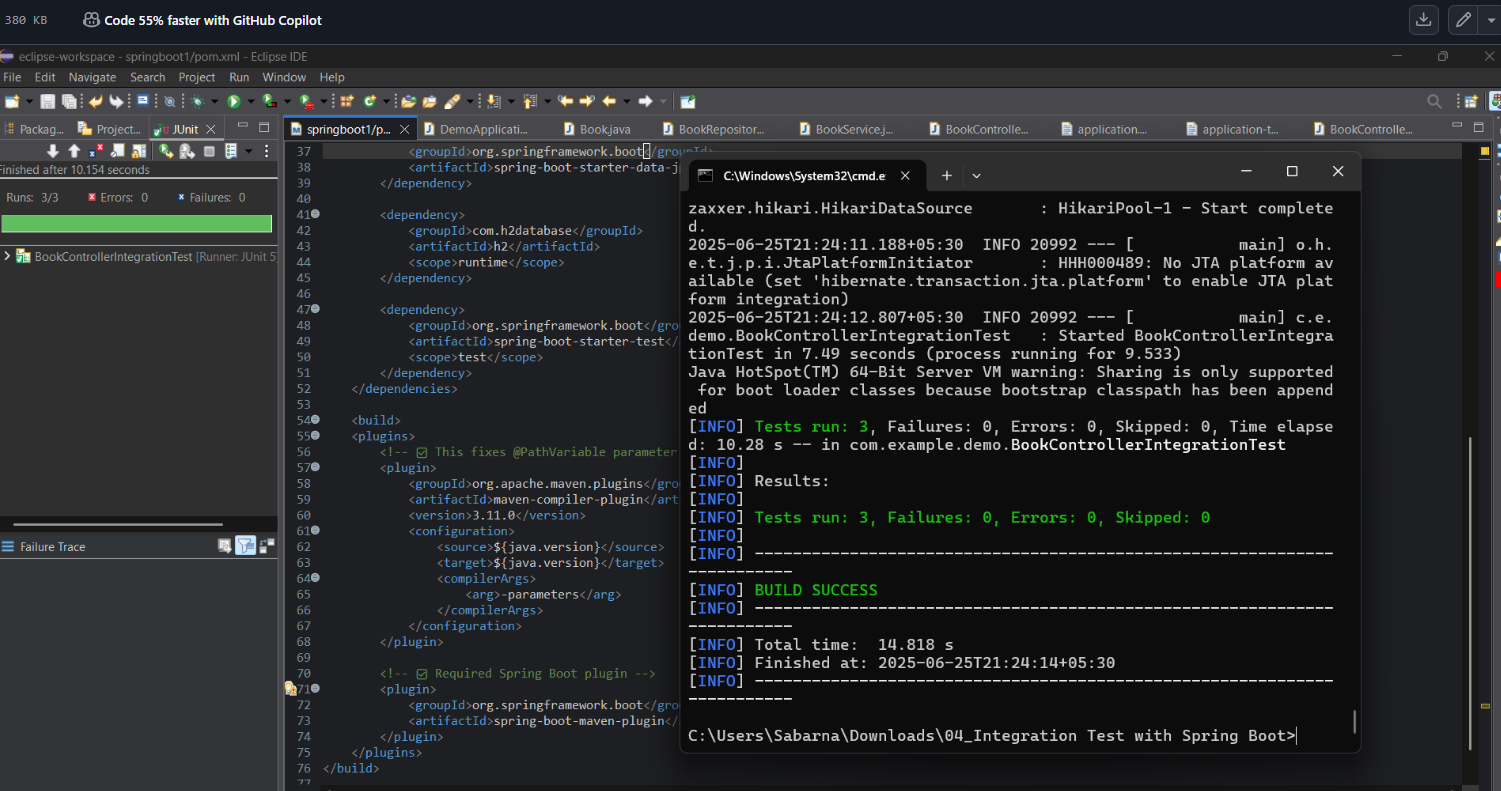
public static void main(String[] args) {

SpringApplication.run(DemoApplication.class, args);

}

}

**OUTPUT:**

****