**Exercise 1: Configuring a Basic Spring Application**

**BookRepo.java**

package com.example.repository;

public class BookRepository {

    public void displayBooks() {

        System.out.println("Books displayed from the repository.");

    }

}

**Bookservice.java**

package com.example.service;

import com.example.repository.BookRepository;

public class BookService {

    private BookRepository bookRepository;

    public void setBookRepository(BookRepository bookRepository) {

        this.bookRepository = bookRepository;

    }

    public void listBooks() {

        System.out.println("BookService calling repository.");

        bookRepository.displayBooks();

    }

}

**App.java**

package com.example;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

import com.example.service.BookService

public class App {

    public static void main(String[] args) {

        ApplicationContext context = new ClassPathXmlApplicationContext("applicationContext.xml");

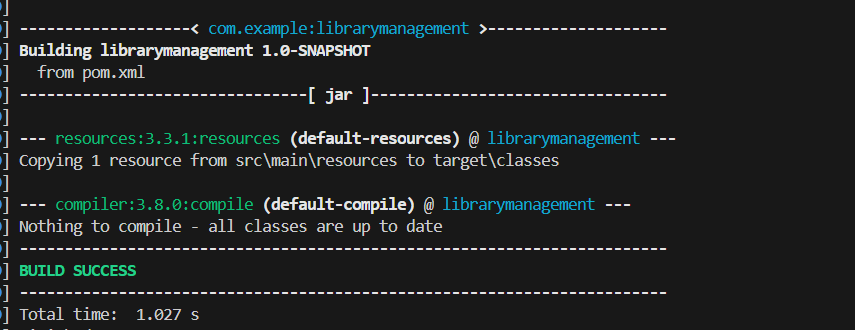
        BookService bs = (BookService) context.getBean("bookService");

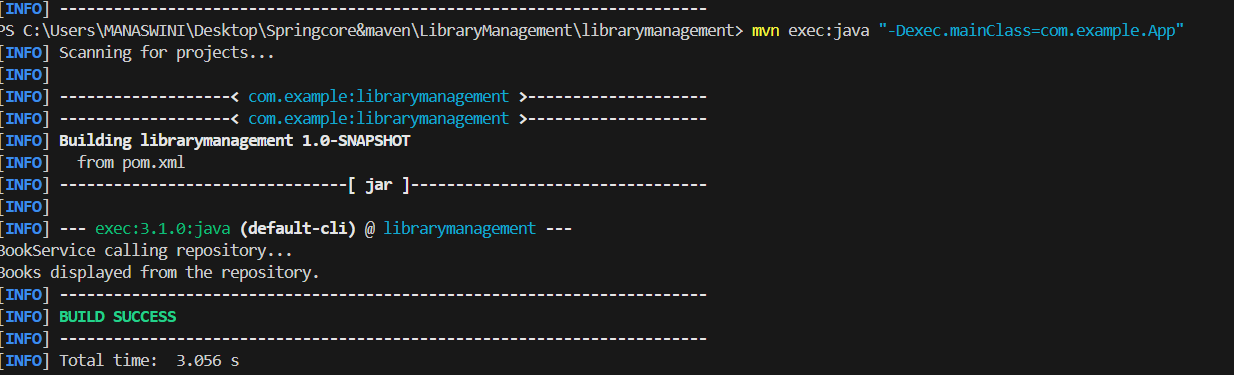
        bs.listBooks();

        ((ClassPathXmlApplicationContext) context).close();

    }

}





\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Exercise 2: Implementing Dependency Injection**

**BookService.java**

package com.example.service;

import com.example.repository.BookRepository;

public class BookService {

    private BookRepository bookRepository;

    // Setter for dependency injection

    public void setBookRepository(BookRepository bookRepository) {

        this.bookRepository = bookRepository;

    }

    public void listBooks() {

        System.out.println("Listing books via BookService...");

        bookRepository.printData();

    }

}

**App.java**

package com.example;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

import com.example.service.BookService;

public class App {

    public static void main(String[] args) {

        ApplicationContext context = new ClassPathXmlApplicationContext("applicationContext.xml");

        BookService bookService = (BookService) context.getBean("bookService");

        bookService.listBooks();

        ((ClassPathXmlApplicationContext) context).close();

    }

}

**BookRepo.java**

package com.example.repository;

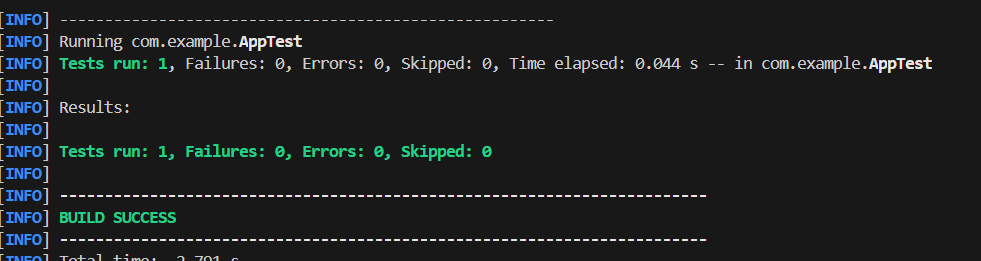
public class BookRepository {

    public void printData() {

        System.out.println("Accessing book data from repository...");

    }

}



**Exercise 3: Implementing Logging with Spring AOP**

**LoggingAspect.java**

package com.example.aspect;

import org.aspectj.lang.ProceedingJoinPoint;

import org.aspectj.lang.annotation.\*;

@Aspect

public class LoggingAspect {

    @Around("execution(\* com.example.service.\*.\*(..))")

    public Object logExecutionTime(ProceedingJoinPoint joinPoint) throws Throwable {

        long start = System.currentTimeMillis();

        Object result = joinPoint.proceed();

        long duration = System.currentTimeMillis() - start;

        System.out.println("[LOG] " + joinPoint.getSignature() + " executed in " + duration + "ms");

        return result;

    }

}

**BookRepo**

package com.example.repository;

import org.springframework.stereotype.Repository;

@Repository

public class BookRepository {

    public void save() {

        System.out.println("Saving book...");

    }

}

**Bookservice**

package com.example.service;

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

import org.springframework.stereotype.Service;

import com.example.repository.BookRepository;

@Service

public class BookService {

    private BookRepository bookRepository;

    @Autowired

    public void setBookRepository(BookRepository bookRepository) {

        this.bookRepository = bookRepository;

    }

    public void performOperation() {

        System.out.println("Inside BookService...");

        bookRepository.save();

    }

}

**App.java**

package com.example;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

import com.example.service.BookService;

public class App {

    public static void main(String[] args) {

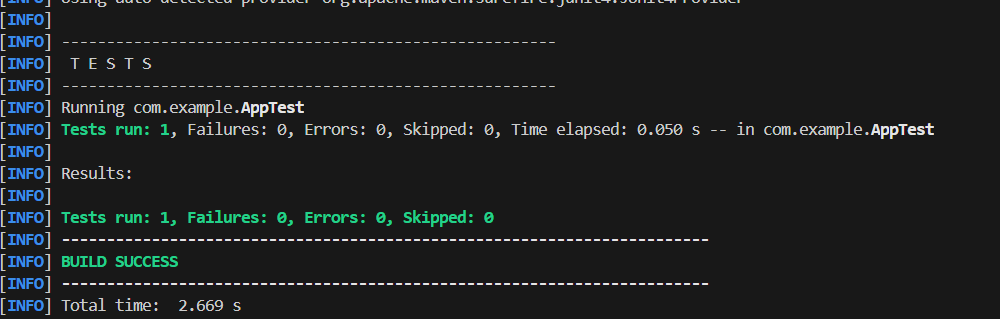
        ApplicationContext context = new ClassPathXmlApplicationContext("applicationContext.xml");

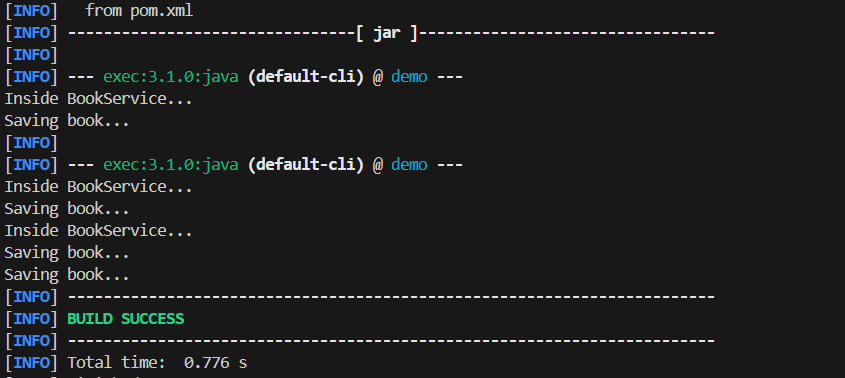
        BookService bookService = context.getBean("bookService", BookService.class);

        bookService.performOperation();

    }

}





**Exercise 4: Creating and Configuring a Maven Project**

package com.example.service;

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

import org.springframework.stereotype.Service;

import com.example.repository.BookRepository;

@Service

public class BookService {

    private BookRepository bookRepository;

    @Autowired

    public void setBookRepository(BookRepository bookRepository) {

        this.bookRepository = bookRepository;

    }

    public void performOperation() {

        System.out.println("Inside BookService...");

        bookRepository.save();

    }

}

**Bookrepo**

package com.example.repository;

import org.springframework.stereotype.Repository;

@Repository

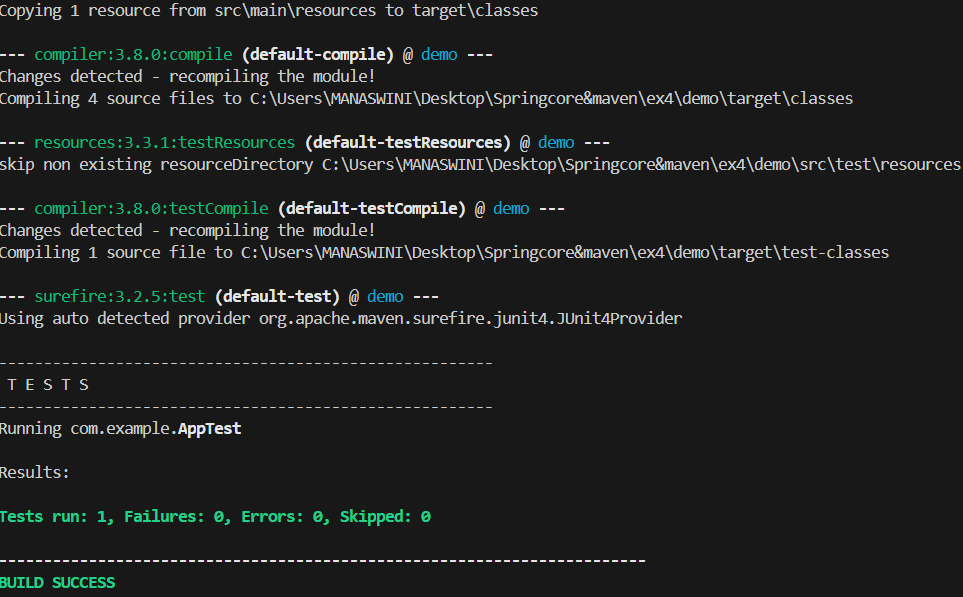
public class BookRepository {

    public void save() {

        System.out.println("Saving book...");

    }

}



**Exercise 5: Configuring the Spring IoC Container**

**Bookrepo.java**

package com.example.repository;

public class BookRepository {

    public void fetchBooks() {

        System.out.println("Fetching books from repository...");

    }

}

**App.java**

package com.example;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

import com.example.service.BookService;

public class App {

    public static void main(String[] args) {

        // Load Spring context from resources

        ApplicationContext context = new ClassPathXmlApplicationContext("applicationContext.xml");

        BookService bookService = (BookService) context.getBean("bookService");

        bookService.listBooks();

    }

}

**Service.java**

package com.example.service;

import com.example.repository.BookRepository;

public class BookService {

    private BookRepository bookRepository;

    public void setBookRepository(BookRepository bookRepository) {

        this.bookRepository = bookRepository;

    }

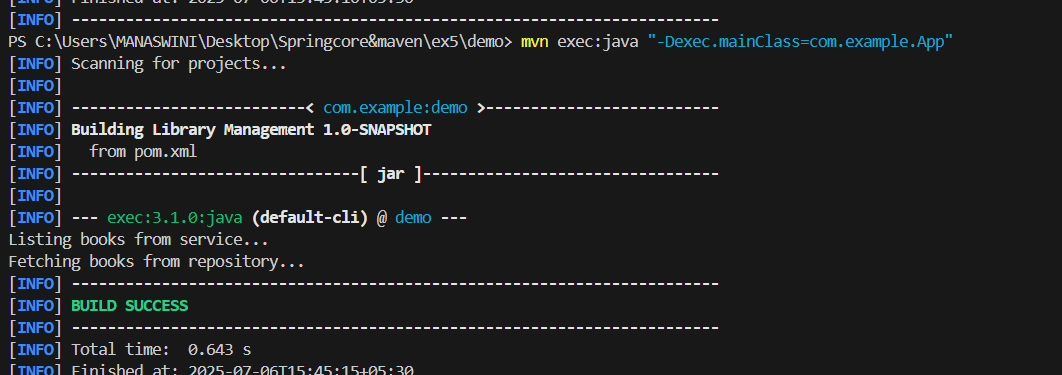
    public void listBooks() {

        System.out.println("Listing books from service.");

        bookRepository.fetchBooks();

    }

}



**Exercise 7: Implementing Constructor and Setter Injection**

package com.example.service;

import com.example.repository.BookRepository;

public class BookService {

    private String serviceName;

    private BookRepository bookRepository;

    public BookService(String serviceName) {

        this.serviceName = serviceName;

    }

    public void setBookRepository(BookRepository bookRepository) {

        this.bookRepository = bookRepository;

    }

    public void displayServiceInfo() {

        System.out.println(" Service Name: " + serviceName);

        bookRepository.fetchBooks();

    }

}

package com.example;

import com.example.service.BookService;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

public class App {

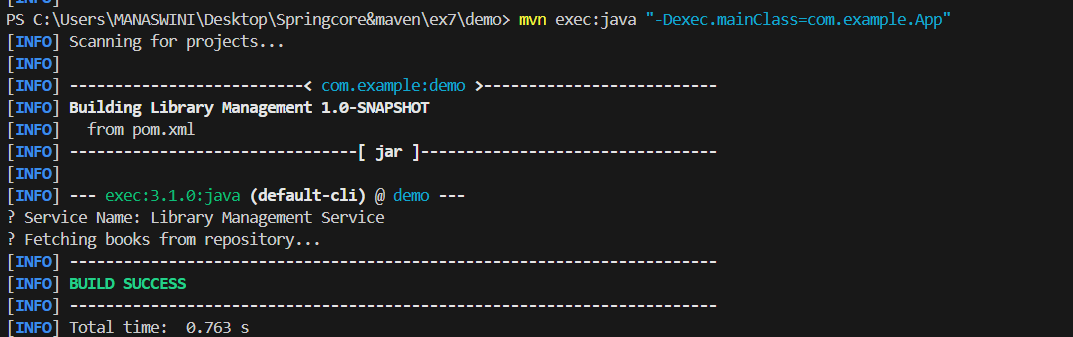
    public static void main(String[] args) {

        ApplicationContext context = new ClassPathXmlApplicationContext("applicationContext.xml");

        BookService bookService = (BookService) context.getBean("bookService");

        bookService.displayServiceInfo();

    }

} 

**Exercise 8: Implementing Basic AOP with Spring**

Logging.java

package com.example.aspect;

import org.aspectj.lang.annotation.After;

import org.aspectj.lang.annotation.Aspect;

import org.aspectj.lang.annotation.Before;

@Aspect

public class LoggingAspect {

    @Before("execution(\* com.example.service.BookService.performOperation(..))")

    public void beforeMethod() {

        System.out.println("[Before] BookService method is about to be executed.");

    }

    @After("execution(\* com.example.service.BookService.performOperation(..))")

    public void afterMethod() {

        System.out.println("[After] BookService method has executed.");

    }

}

Bookrepo

package com.example.repository;

public class BookRepository {

    public void listBooks() {

        System.out.println("Listing all books in repository.");

    }

}

package com.example.service;

import com.example.repository.BookRepository;

public class BookService {

    private BookRepository bookRepository;

    public BookService(BookRepository bookRepository) {

        this.bookRepository = bookRepository;

    }

    public void performOperation() {

        System.out.println("Performing book service operation...");

        bookRepository.listBooks();

    }

}

