

软件测试 作业2 单元测试

黄培正 521021910454

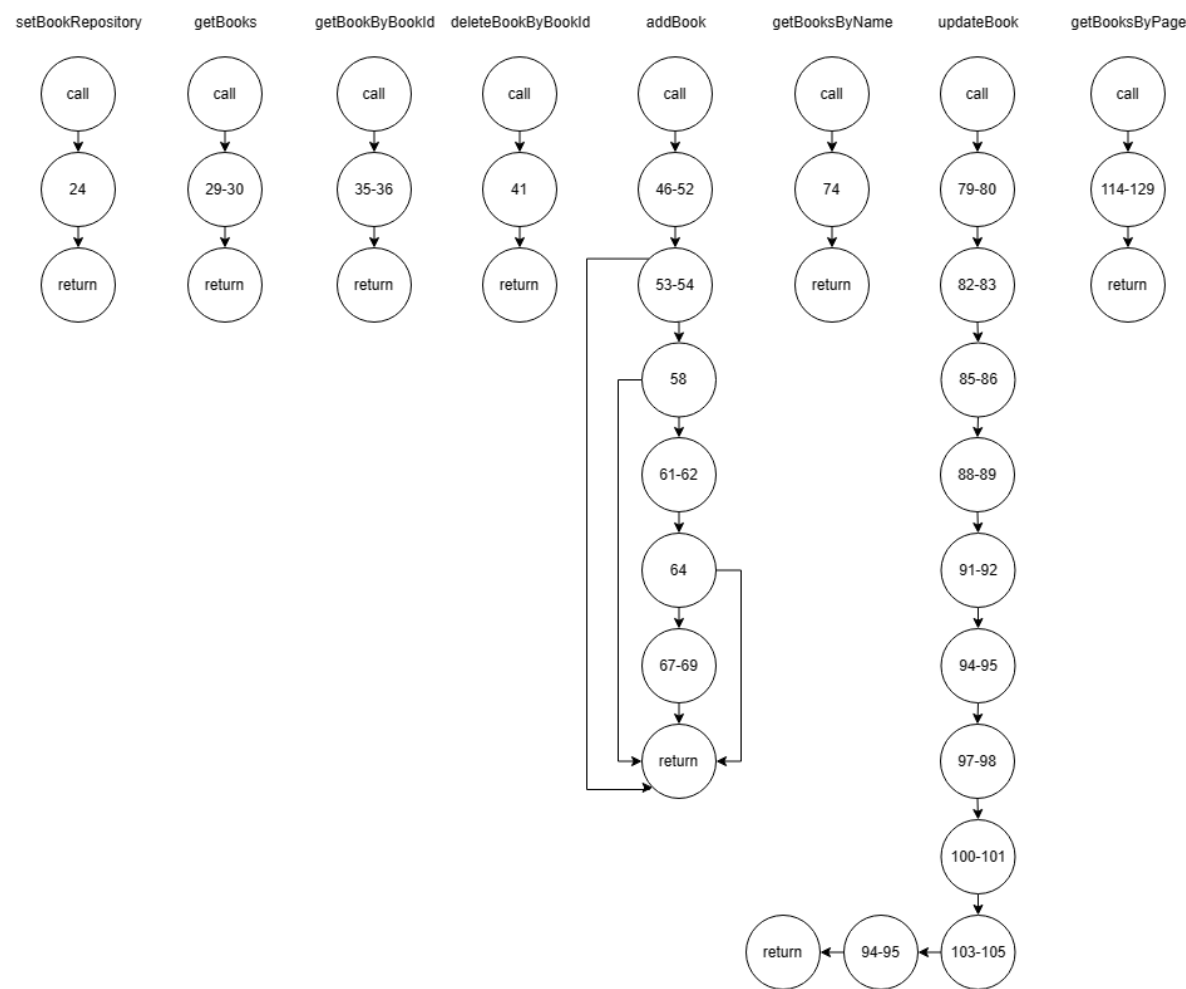
被测代码

我负责测试整个后端项目中的Book、Cart等的相关代码，包含它们的dao、entity、service等部分。

dao和daoimpl部分

dao中只有方法的声明，因此不需要写测试代码，只要对impl部分测试。BookDaoImpl和CartDaoImpl代码类似，这里以BookDaoImpl.java为例分析测试。

对每个待测试文件中的每个待测方法，绘制对应的DD路径，节点中的数字为待测试类所在文件的代码行号。



测试代码部分，@Mock 其中的 BookRepository bookRepository 再 @InjectMocks BookDaoImpl bookDao 注入mock，使用模拟的bookDao进行测试验证。使用 Mockito.verify、Mockito.times、assertSame 等方法测试代码的正确性。（具体见src/test文件夹下的具体测试代码）

考虑到多数被测代码仅是调用repository提供的方法，数据流我们仅分析较复杂的 addBook、updateBook 方法。

待测函数	变量	路径开始节点	路径结束节点	是否定义清除
addBook	params	46	52	是

待测函数	变量	路径开始节点	路径结束节点	是否定义清除
\	name	46	69	是
\	author	47	69	是
\	isbn	48	69	是
\	description	49	69	是
\	image	50	69	是
\	type	51	69	是
\	brief	52	69	是
\	price	61	69	是
\	inventory	67	69	是

待测函数	变量	路径开始节点	路径结束节点	是否定义清除
updateBook	params	79	107	是
\	bookId	79	108	是
\	name	82	83	是
\	author	85	86	是
\	isbn	88	89	是
\	description	91	92	是
\	image	94	95	是
\	type	97	98	是
\	brief	100	101	是
\	price	103	105	是
\	inventory	107	108	是

entity部分

这部分代码是定义book和cartItem实体类，用于映射数据库中的对应信息表，它会自动生成类的getters、setters、equals、hashCode和toString方法。我们只需要测试这些方法是否正确即可，具体见测试代码。这里不涉及DD路径和数据流。

repository层

Repository层是在应用程序中负责处理数据持久化和数据库操作的一层。它主要包含与数据存储相关的逻辑，提供了一种将应用程序中的数据对象持久化到数据库中的方式。我们使用了JpaRepository（是Spring Data JPA提供的一个接口，它扩展了JPA的标准接口）提供的一组基本的数据库操作方法，进行数据访问和操作。

在使用junit5进行测试时，这部分默认是100%覆盖的，因此我们仅对 `BookRepository` 进行了测试。使用Spring Boot 提供的 `@DataJpaTest` 注解和JUnit进行测试，方法参考[使用JUnit和@DataJpaTest测试Spring Data Repository - spring 中文网 \(springdoc.cn\)](#)，具体代码见src/test。

service和serviceimpl部分

service仅是方法的声明，具体实现在serviceimpl部分，因此只要对serviceimpl部分进行测试。

具体到serviceimpl的实现，只调用了dao提供的方法接口，所以测试只需要验证是否成功调用dao提供的方法。这里不涉及DD路径和数据流。