# 实验四：图书信息的维护

## 实验目的

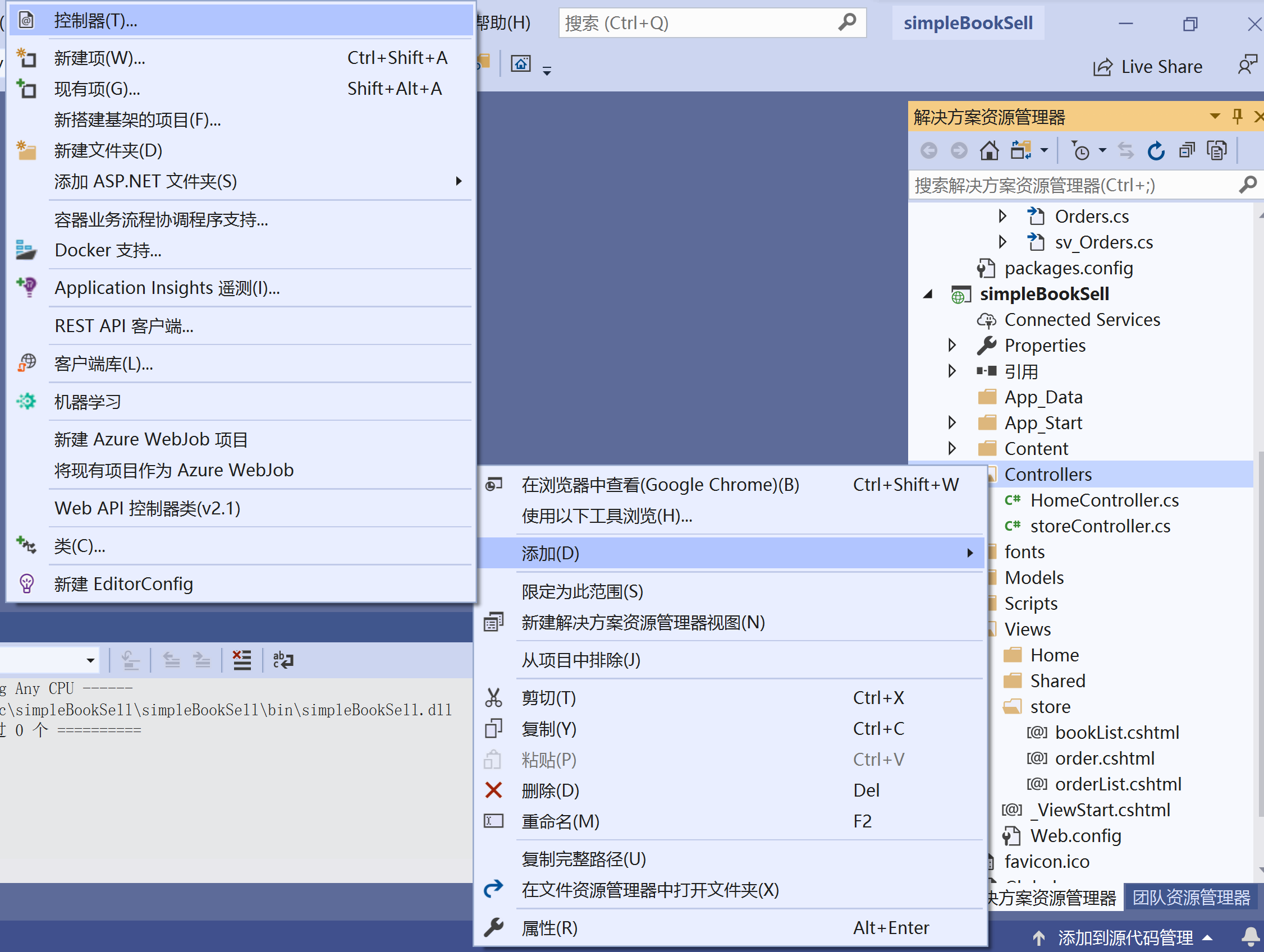
掌握EF模型在MVC框架中的灵活运用，掌握Entity Framework常用的数据访问方法。

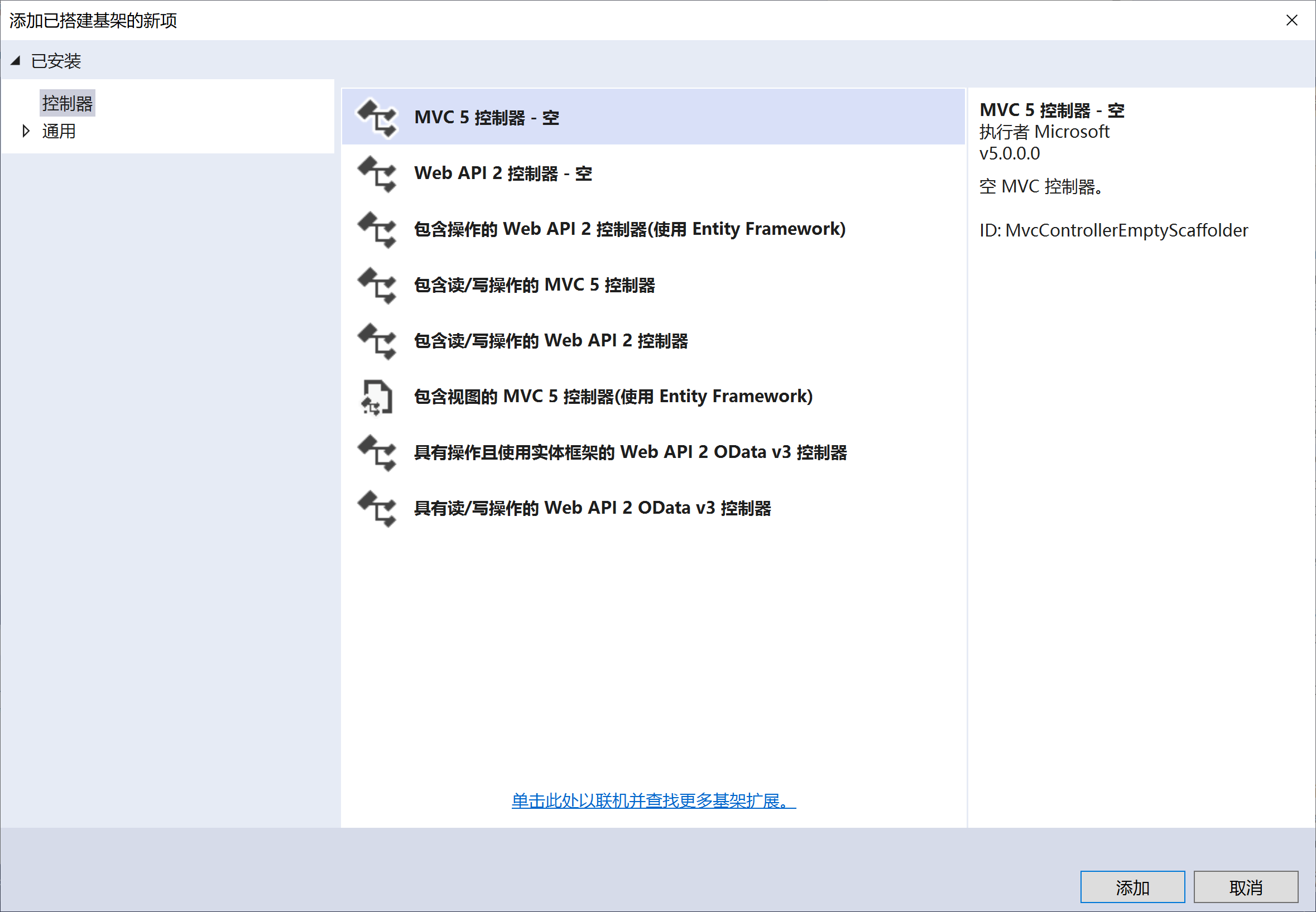
## 实验内容

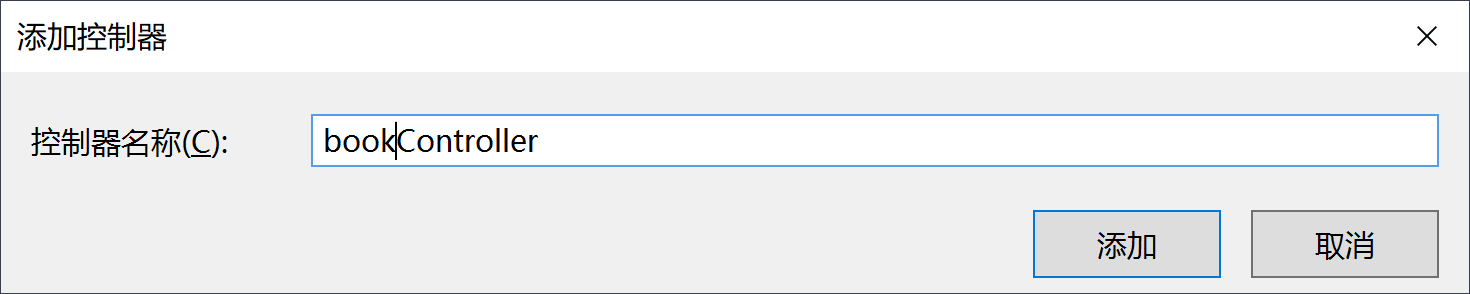
1. 掌握数据维护程序的简单写法。
2. 掌握EF新增代码的写法（2种）
3. 掌握EF修改代码的写法（3种）
4. 掌握EF删除代码的写法
5. 掌握EF单个实体对象的查询方法
6. 掌握EF集合对象的查询方法

## 实验步骤

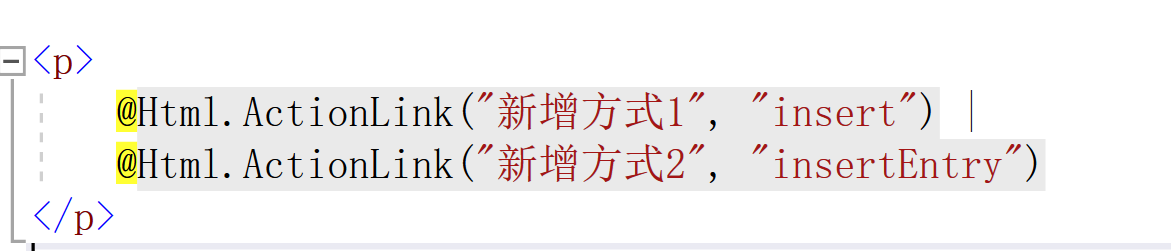
1. **创建控制器“bookController”用于维护图书信息的相关操作**
2. 打开“实验三”的项目 “simpleBookSell”，在网站种添加新的控制器“bookController”用于维护图书信息。







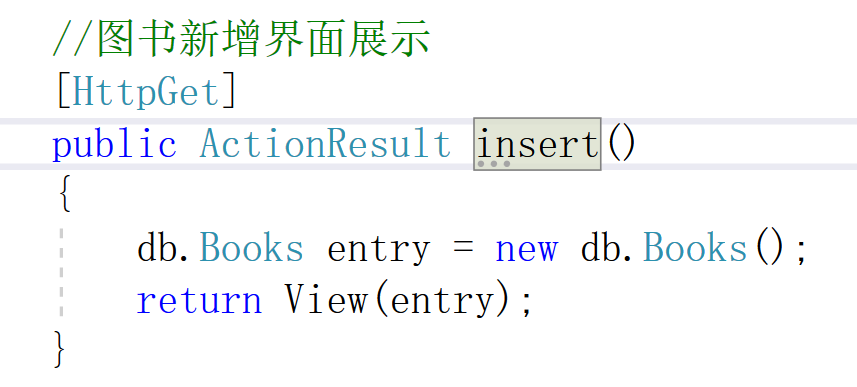
1. 先将之前storeController中的bookList这个Action移植到本控制器中，作为经营者维护图书信息的主界面，注意视图文件也需要移植过来，可以新建也可以直接把文件复制到/View/book/目录下。
2. 在图书展示页面中修改“新增”Action，分别实现两种方式的图书新增操作。

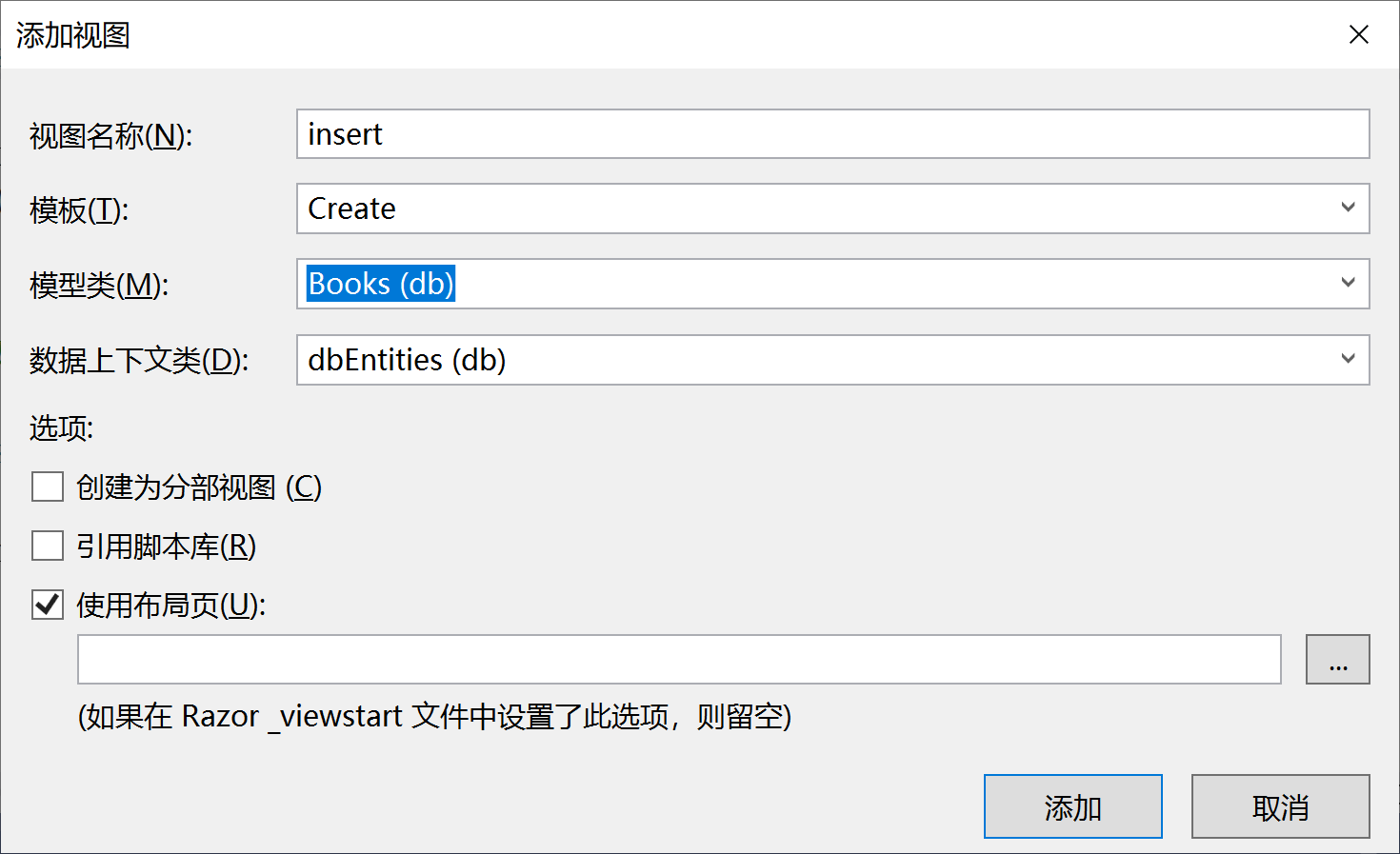


1. **新增图书**
2. 完成新增方式1的操作。在控制器中增加两个“insert”Action，其中一个属性设为[HttpGet]用于展示新增界面，另一个属性为[HttpPost]用于保存新增数据。

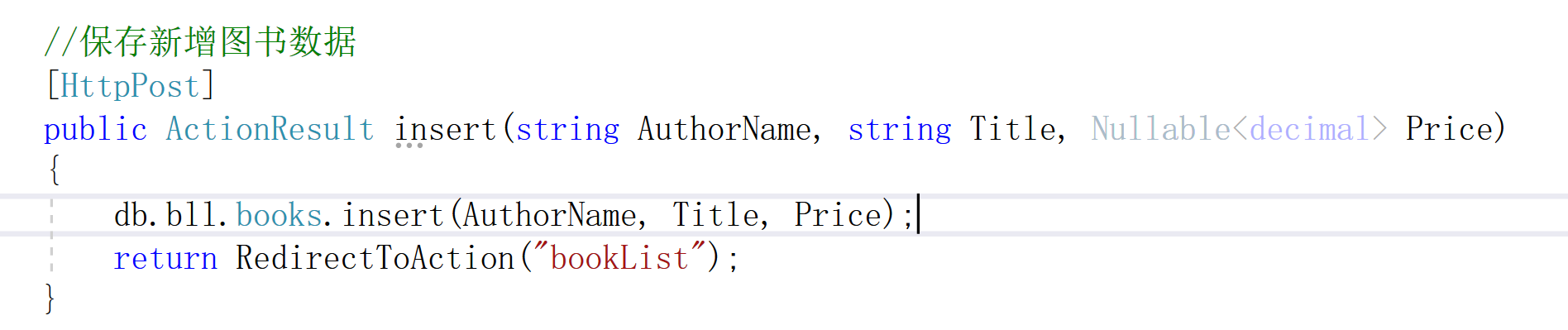


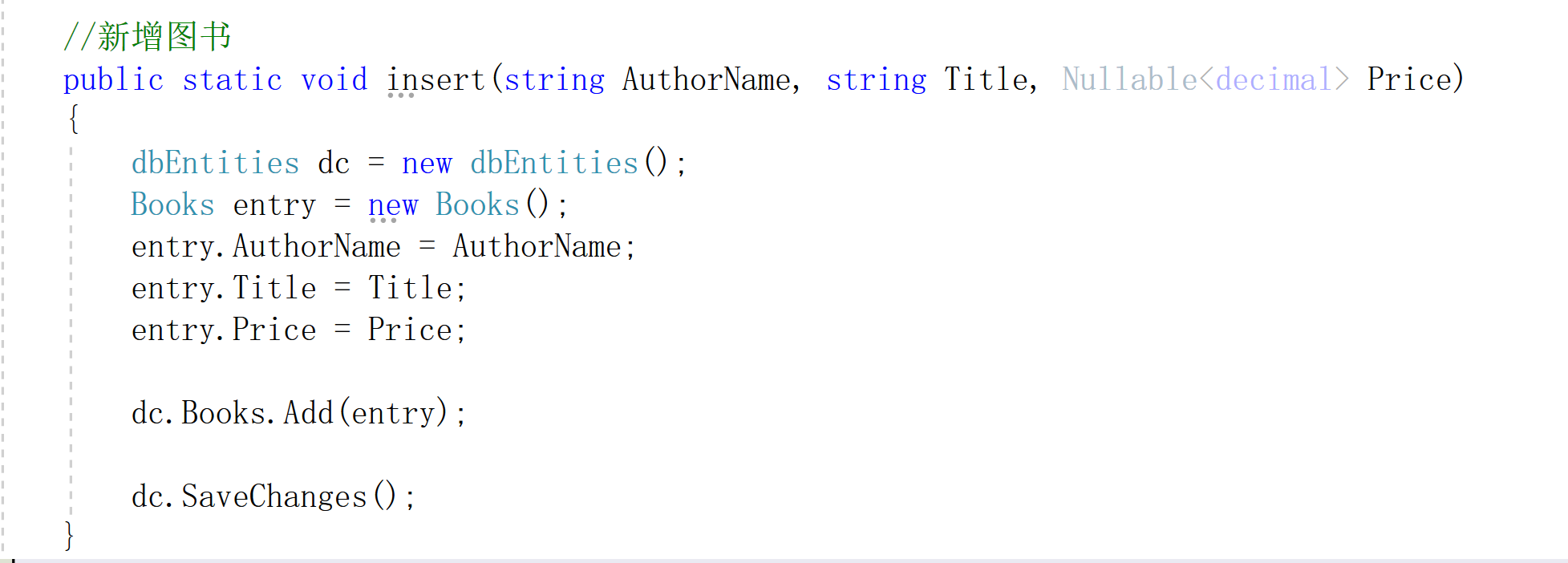
1. 实现新增方式一的视图，首先定义一个空的图书模型实体作为入参传入视图，之后建立强类型视图，支架类型选择Create，注意必须传入参数才可建立强类型视图，否则页面加载会报错。





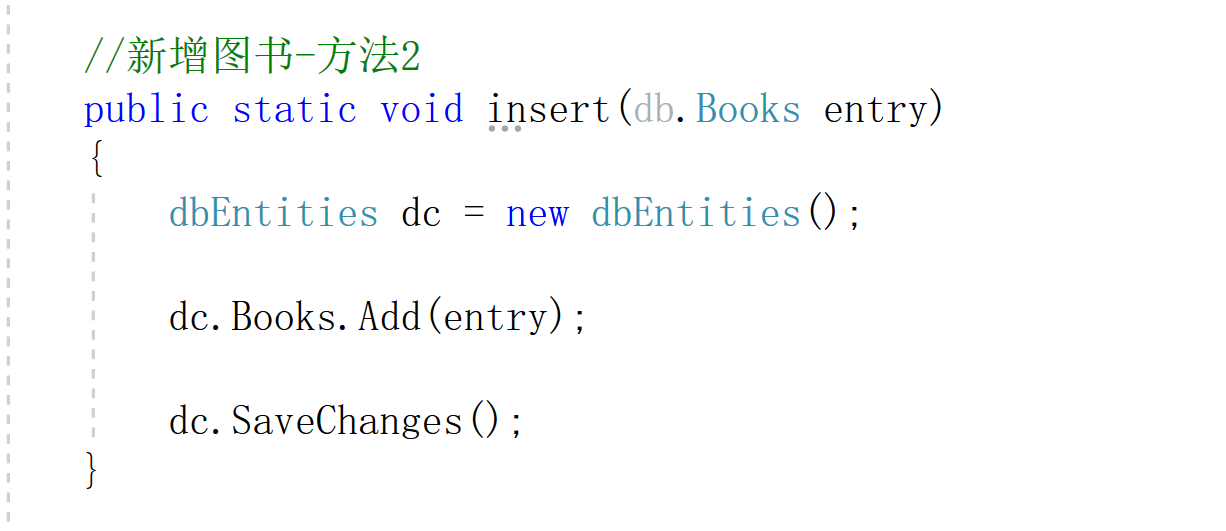
1. 修改视图文件，暂时删除图书封面的上传。
2. 实现新增方式一的数据保存Action，在dbLibrary库的books.cs中同步增加insert函数，参数和insert的action保持一致。



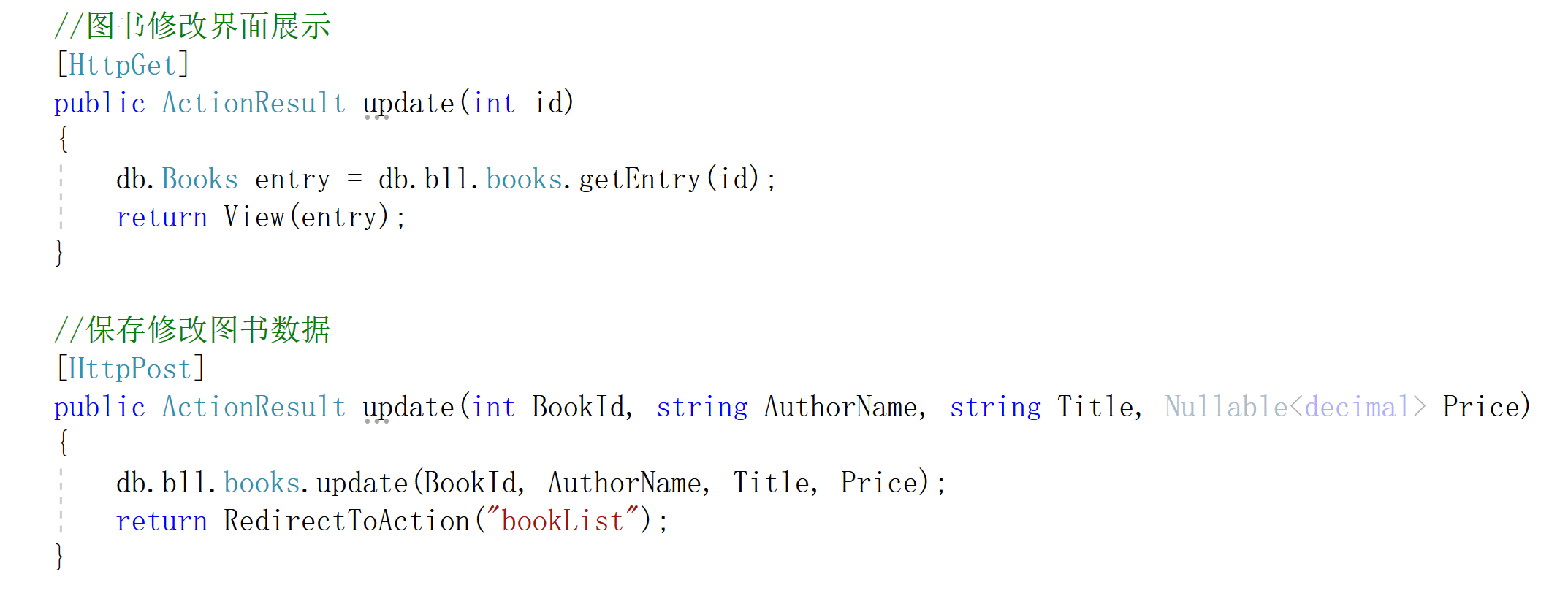


1. 实现新增方式二的View和Action，需要为insertEntry重新增加视图，不可共用方式1的视图，否则点击数据提交会跳到方式1的保存Action。

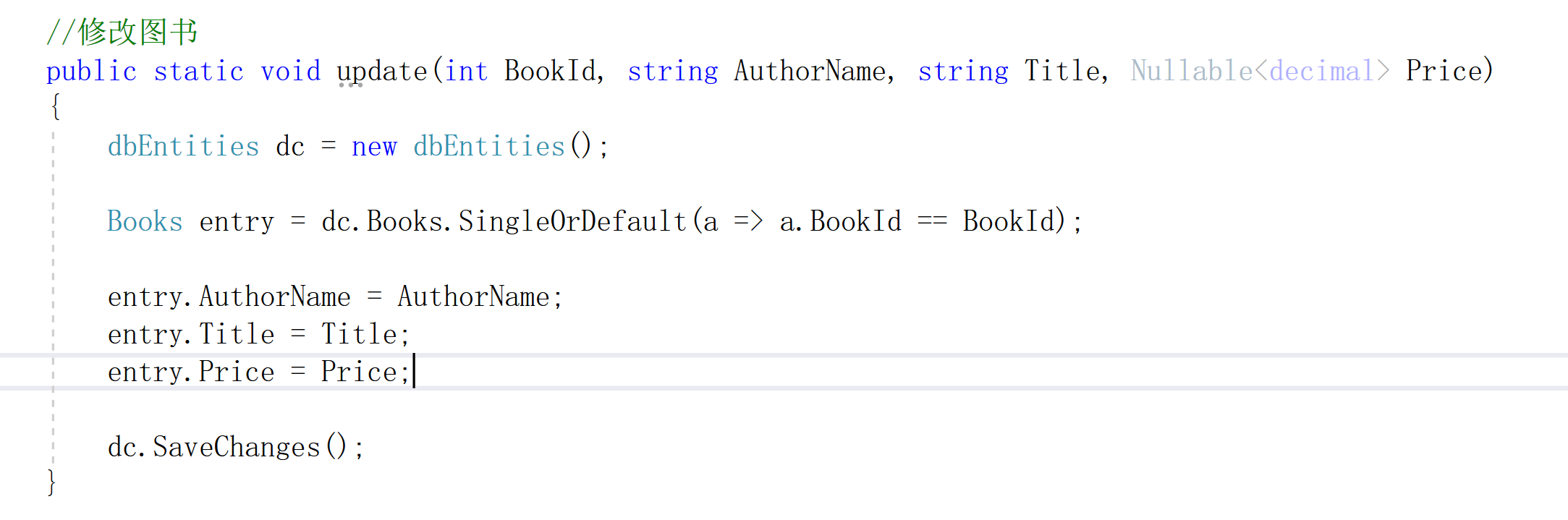




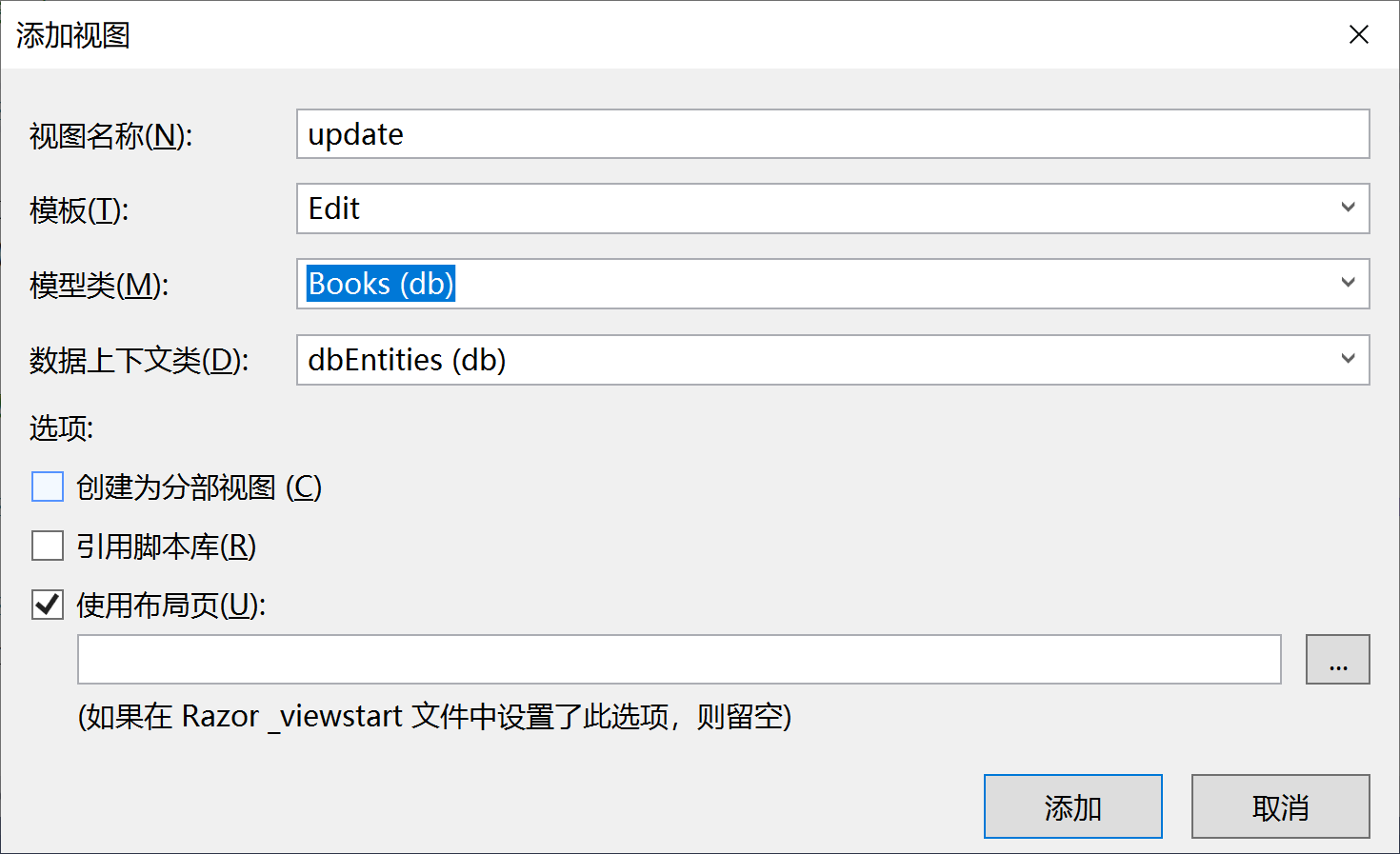
1. **图书修改功能实现**
2. 完成图书修改方式1的实现。新增两个“update”的Action，一个用于展示修改界面一个用于保存修改的图书信息。写法可以参考新增图书，注意此时需要传入需要修改图书的ID才能展示正确的图书信息。



1. 在业务类中实现图书信息的保存。



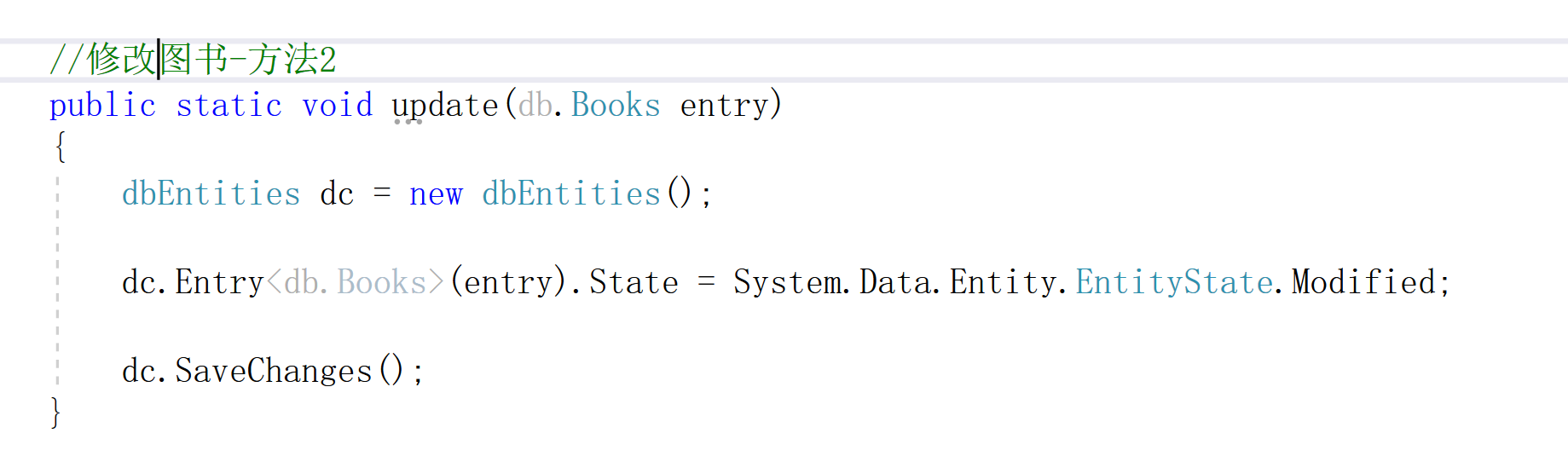
1. 创建修改页面的视图，注意支架选择“Edit”类型。注意生成的视图如果包含BookId字段，要把它隐藏起来，不允许用户修改主键信息，里面的封面图也暂时删除。运行查看效果。



1. 完成图书修改方式2的实现。新增两个“updateEntry”的Action，仿照新增方式二实现。



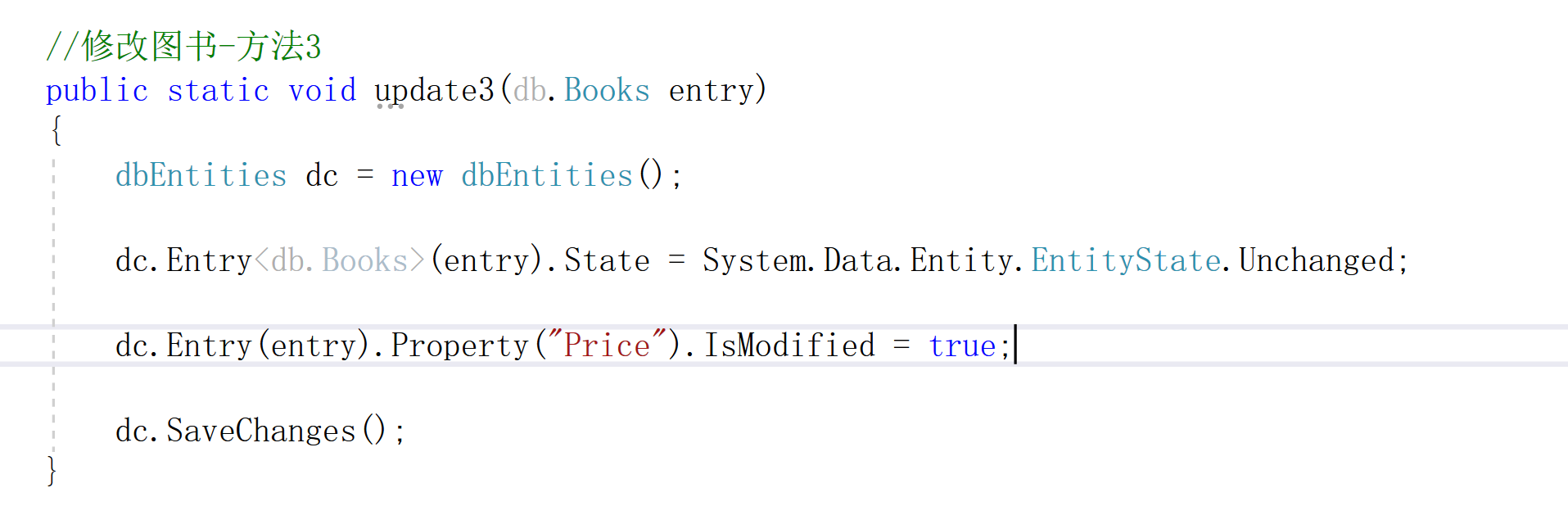
1. 新增业务类代码。



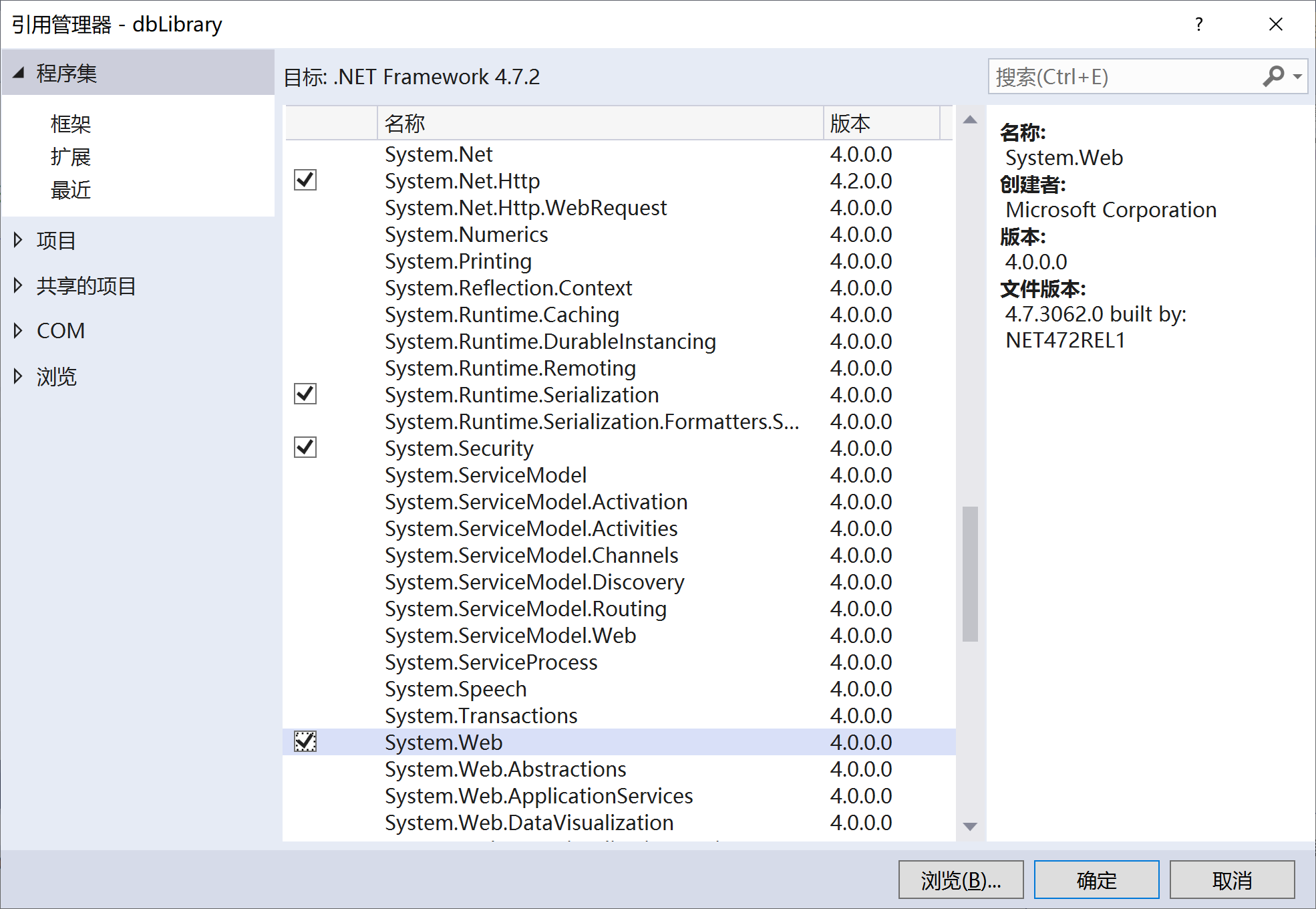
1. 创建修改方式二的视图文件，内容调整方法和修改方式1相同。
2. 完成图书修改方式3的实现。新增两个“update3”的Action，和修改方式2类似。



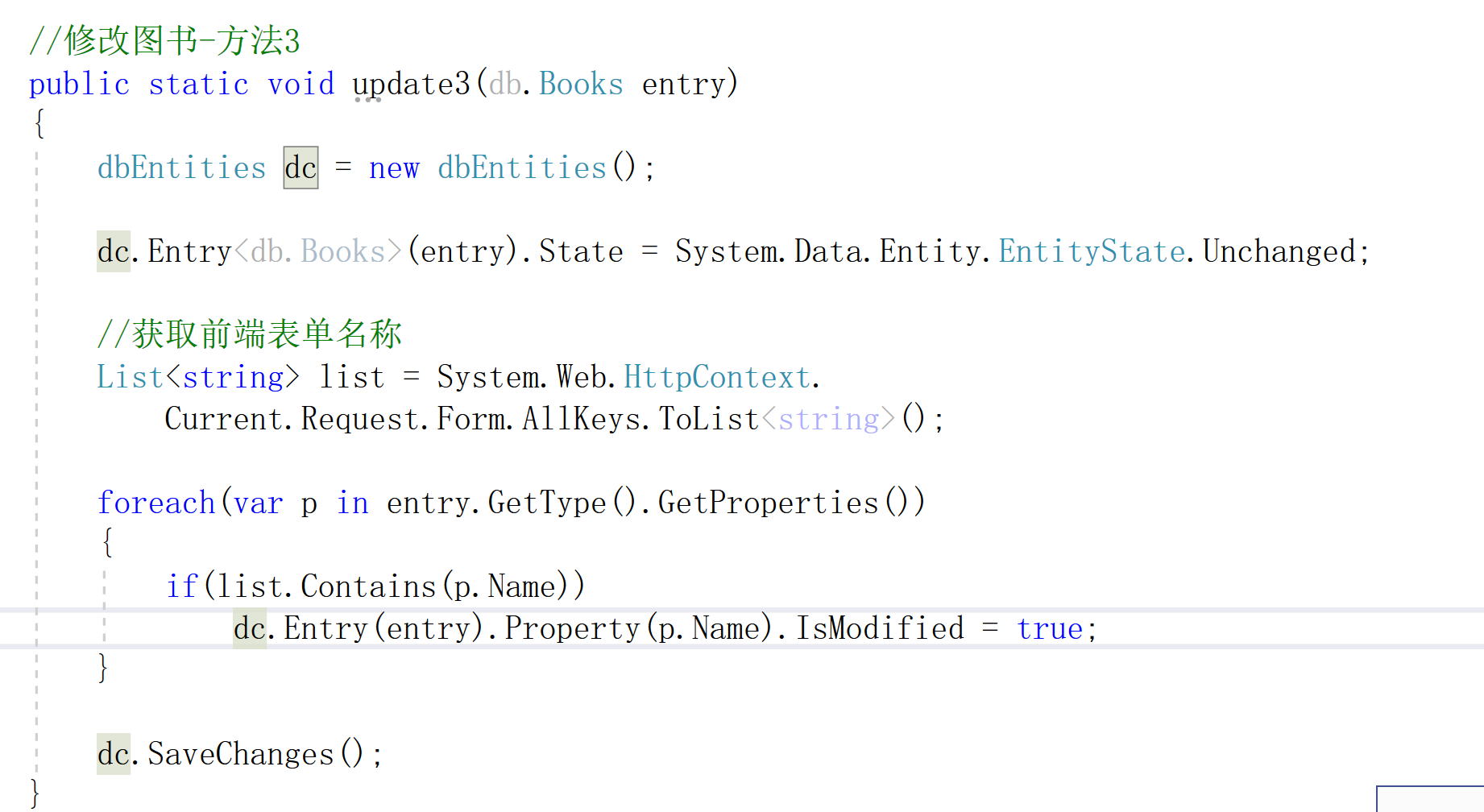
1. 新增业务类代码，新建update3视图，运行查看效果。



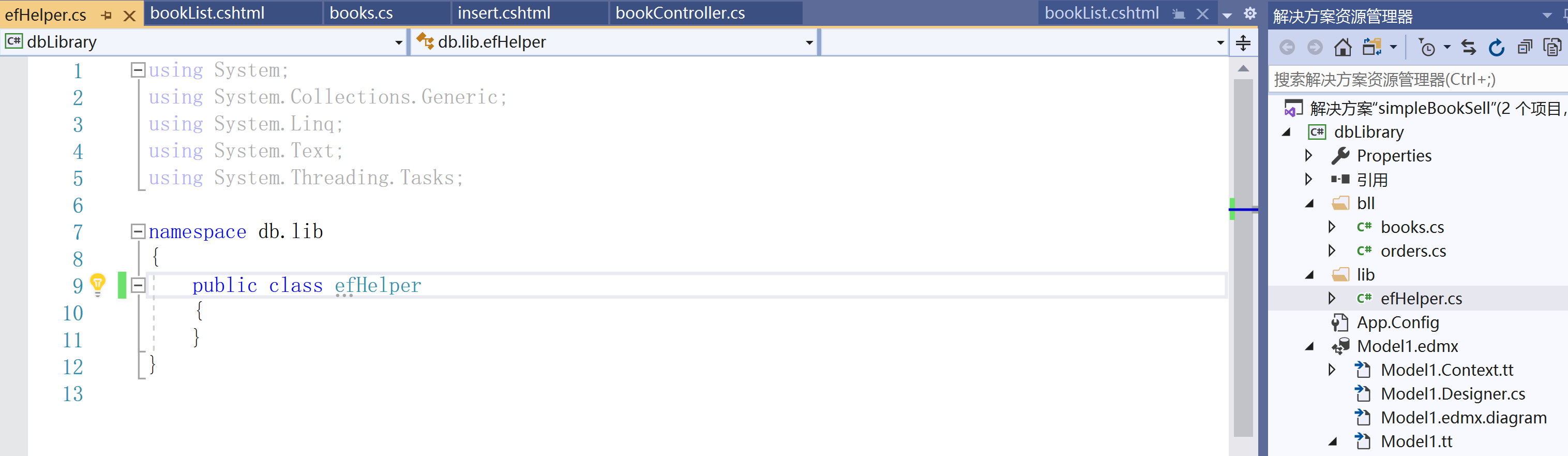
1. 优化业务类代码。获取当前页面表单，在dbLibrary中添加引用，System.web



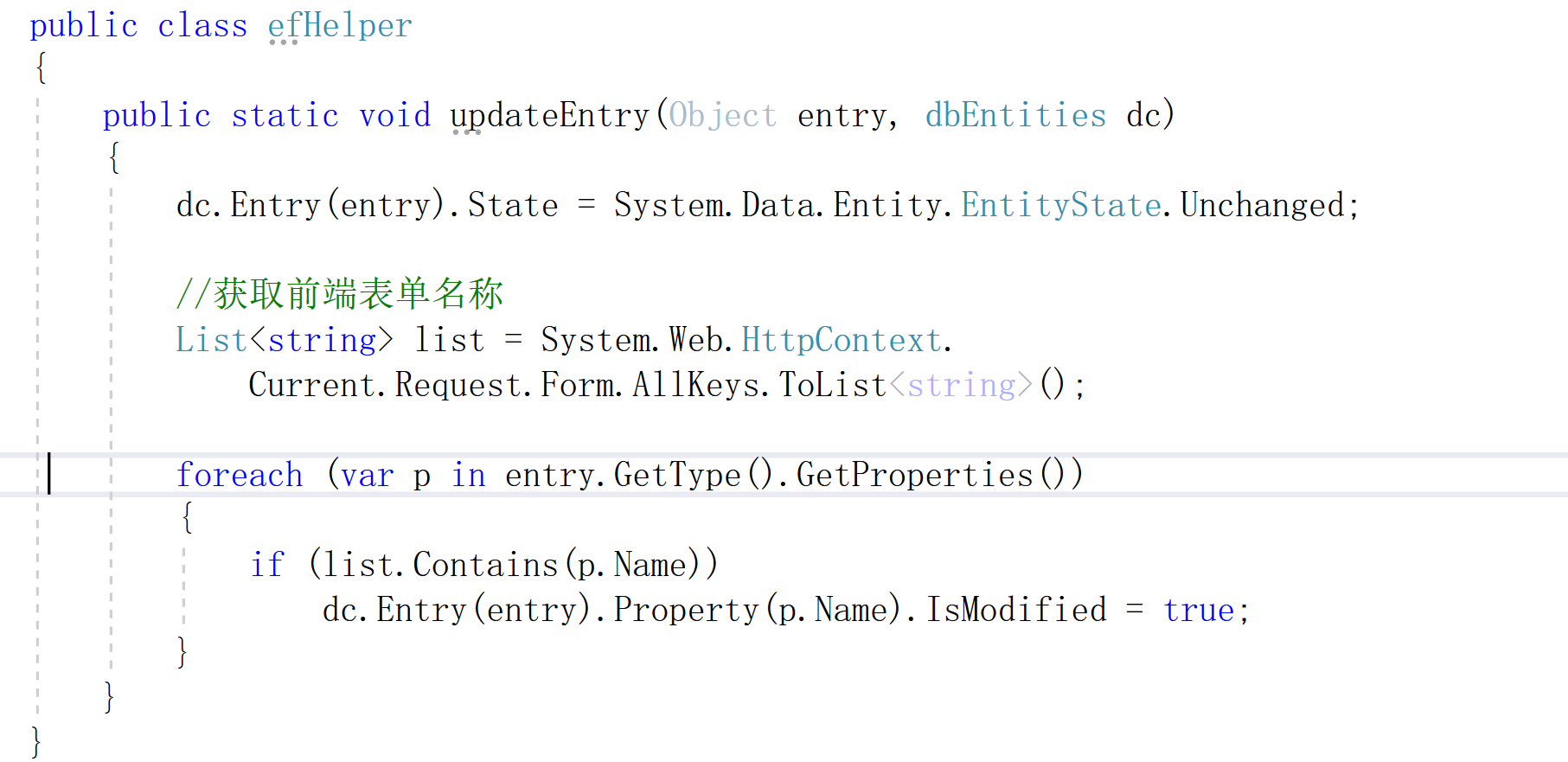
1. 获取当前页面表单，修改业务类代码，试着写出以下函数伪代码。



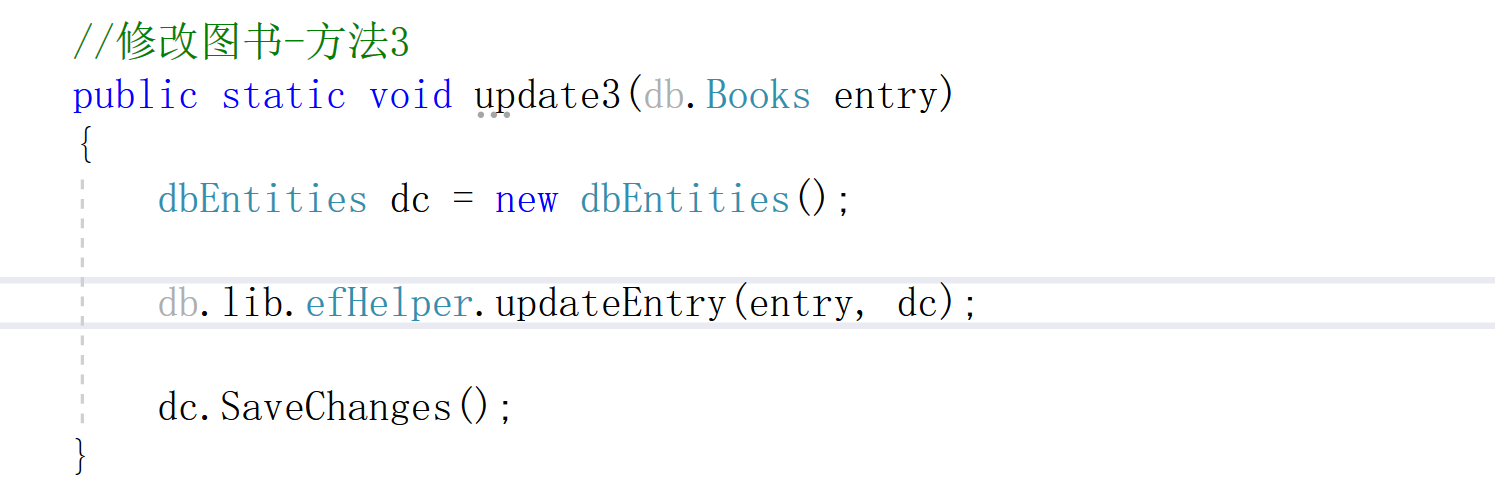
1. 优化修改方式3的代码封装方法，在dbLibrary中新建lib文件夹用于提供通用的业务函数。



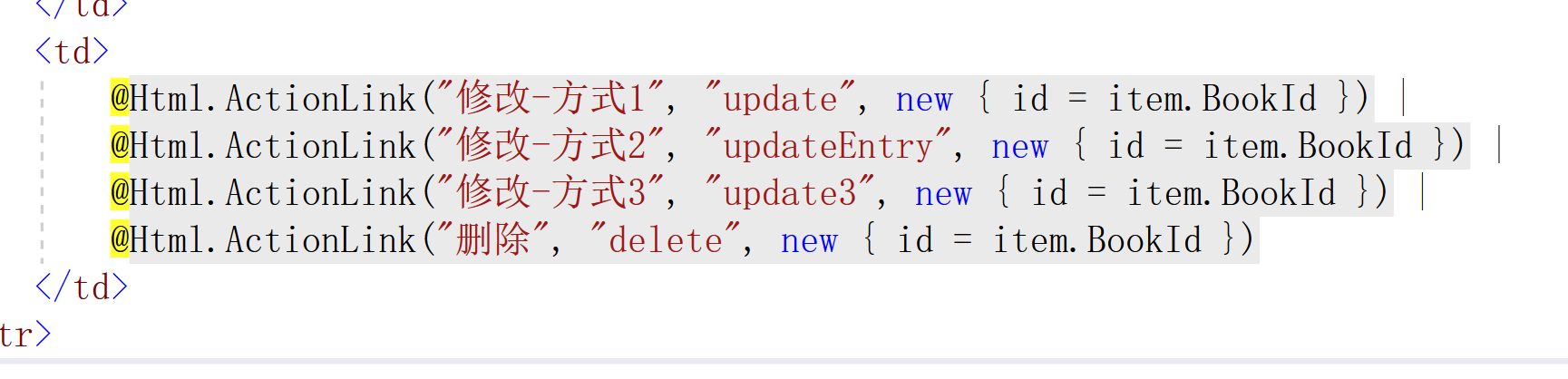
1. 增加一个新的更新数据实体的通用函数



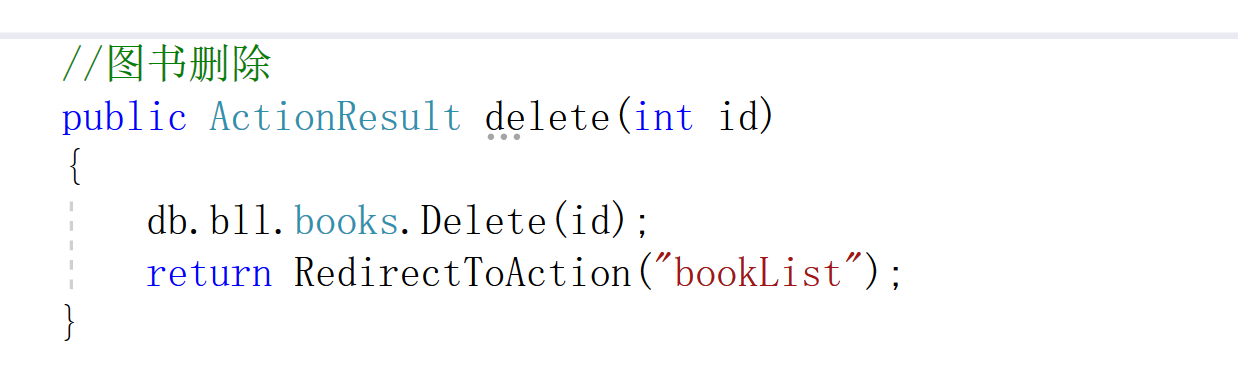
1. 修改业务类的函数，运行代码查看效果。



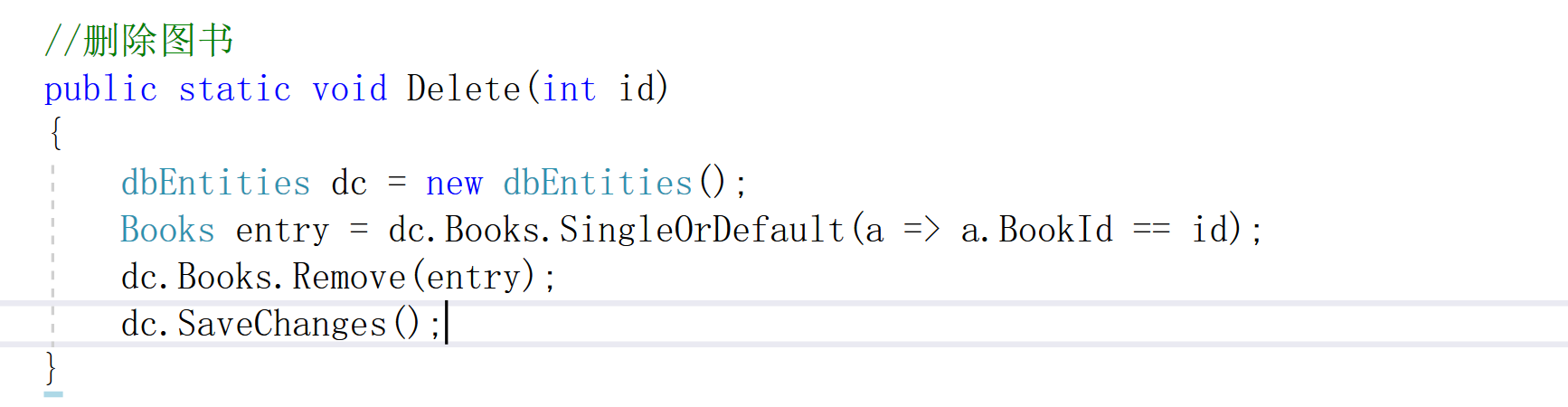
1. **删除图书**
2. 在图书类表中，更新后面增加图书的ActionLink。



1. 创建删除的action并实现。



1. 在业务类中完成delete操作，运行代码查看效果。



作业：

完成图书Detail展示界面的相关操作！