**实验八 Spring项目搭建**

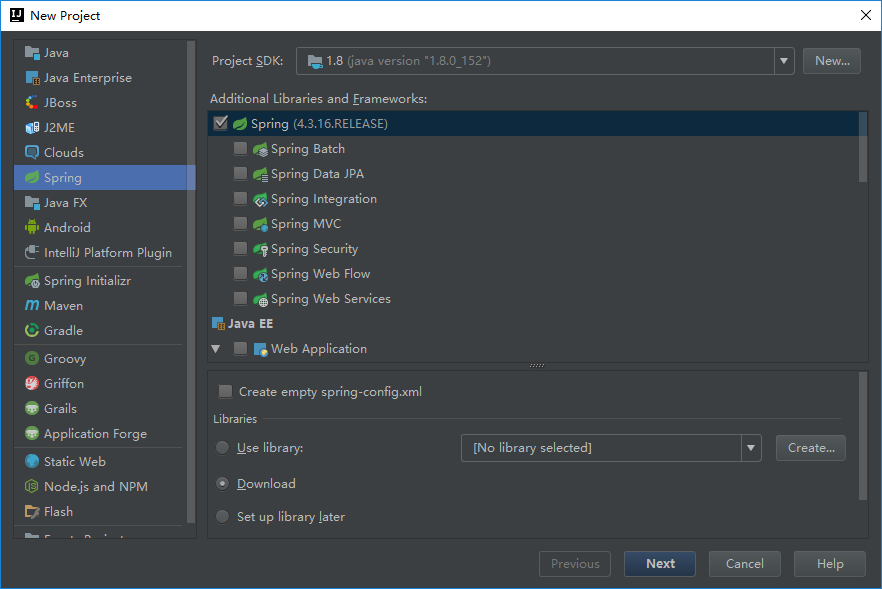
### 一、实验目的

使用Spring搭建项目框架，并熟悉SpringBean的属性。

### 二、实验项目内容

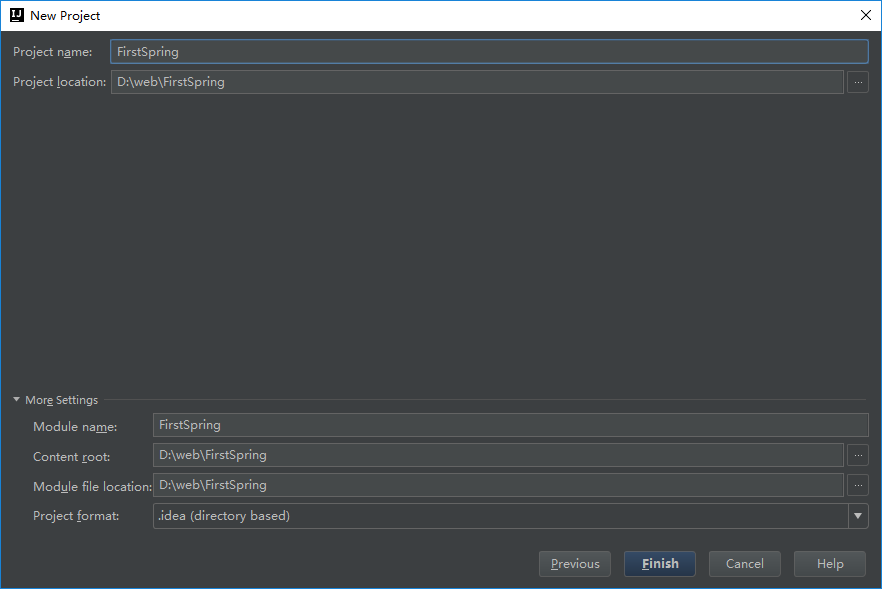
* **1、Spring安装**

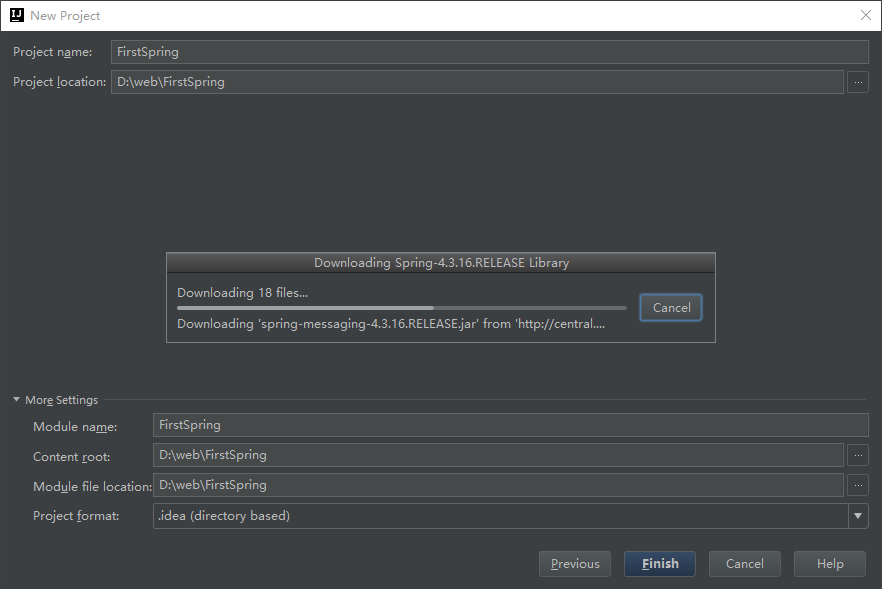
以idea为例



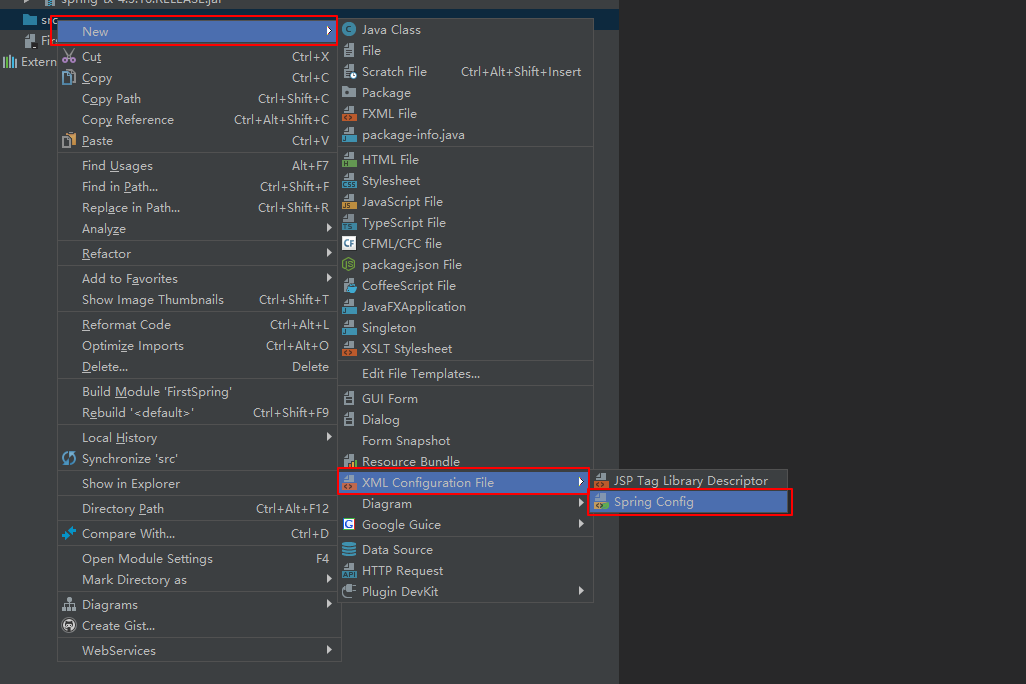
也可以在官网下载：

<http://repo.spring.io/release/org/springframework/spring>





创建xml文件



输入要创建的文件名，确定后生成相应xml

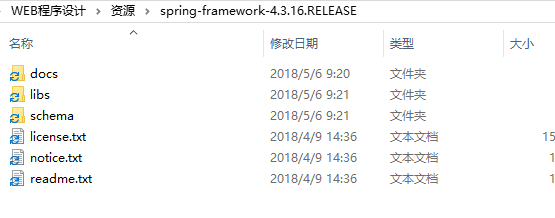
注：官方推荐的配置文件名称为：applicationContext.xml，可以自行命名,idea会提示“Application context not configured for this file” ，点击后面的配置即可

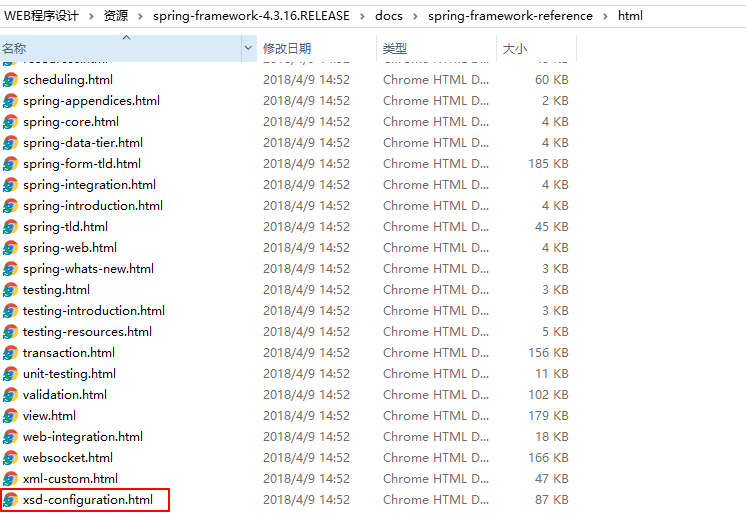
Xml格式为：

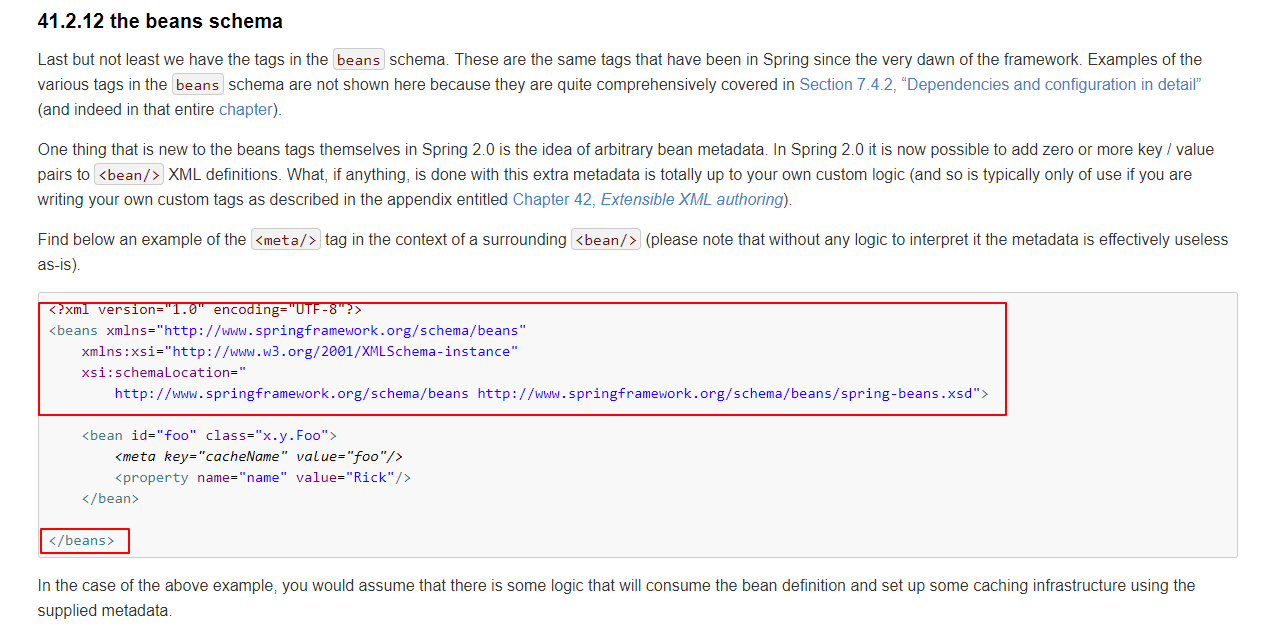
打开

\spring-framework-4.3.16.RELEASE\docs\spring-framework-reference\html\xsd-configuration.html

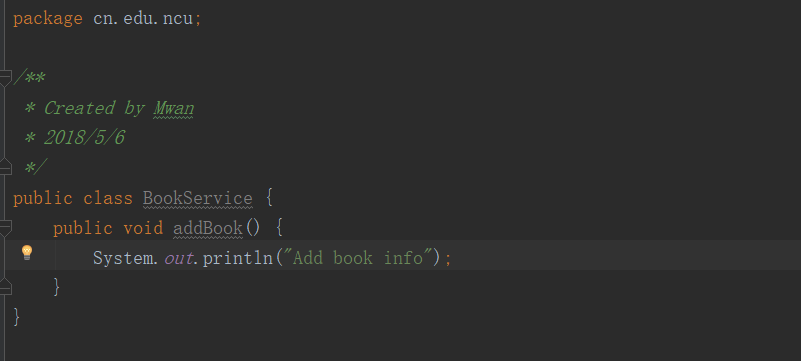
在该文档最后一段中 beans schema



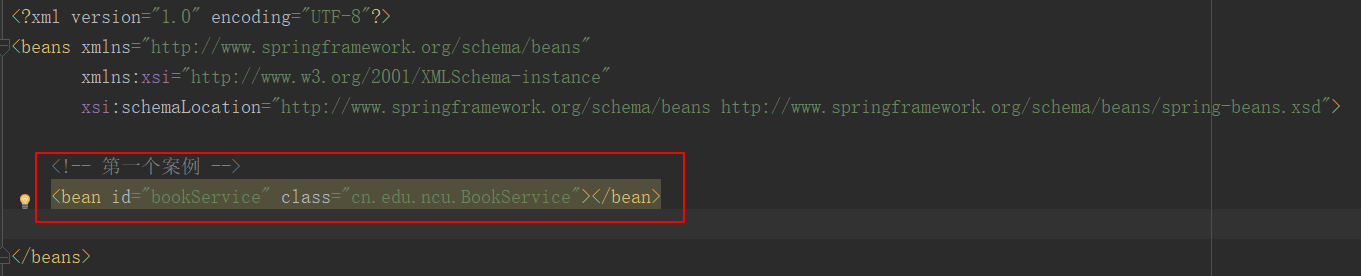




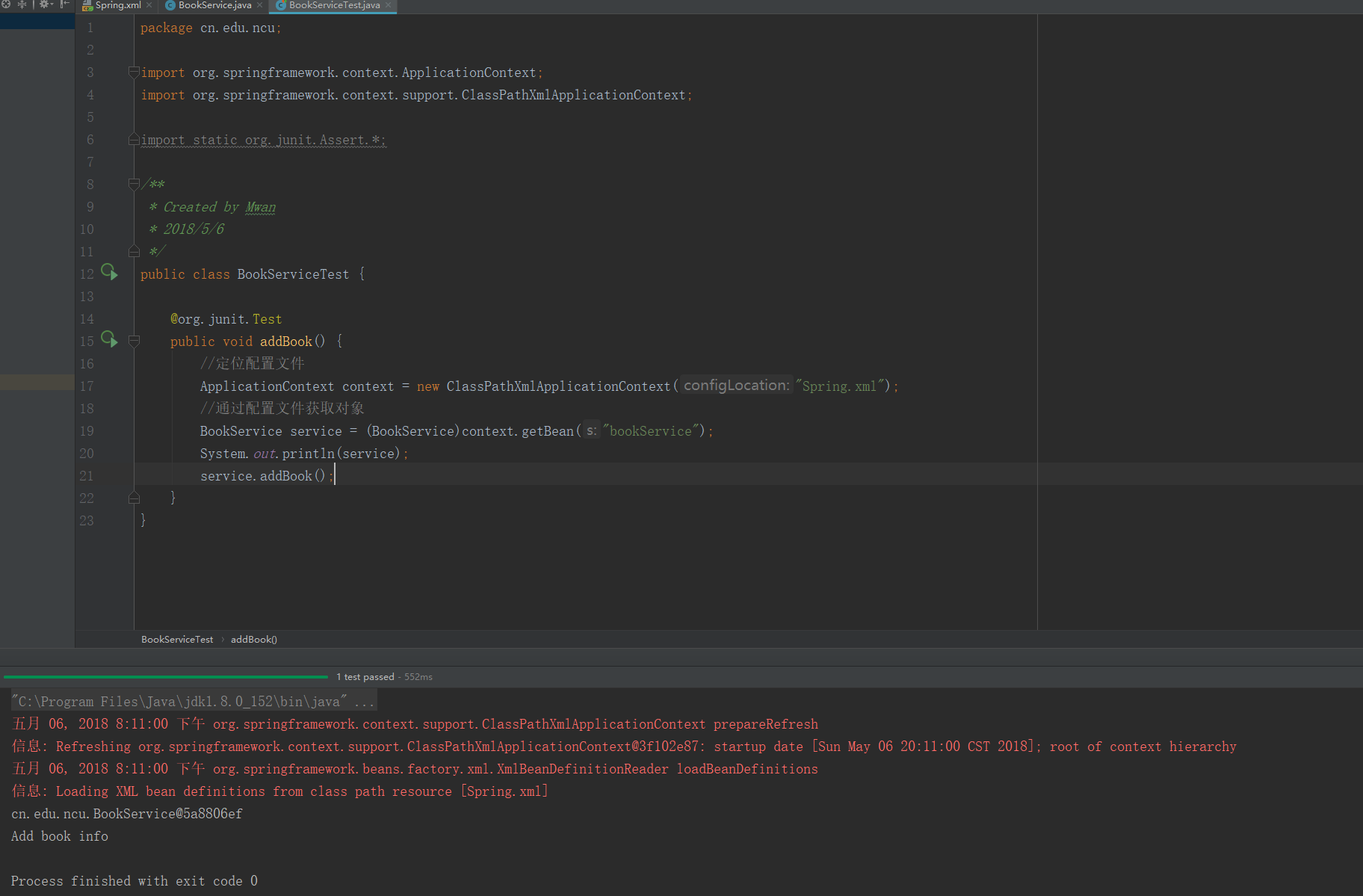
创建BookService类



修改xml文件：



创建单元测试（导入junit包）

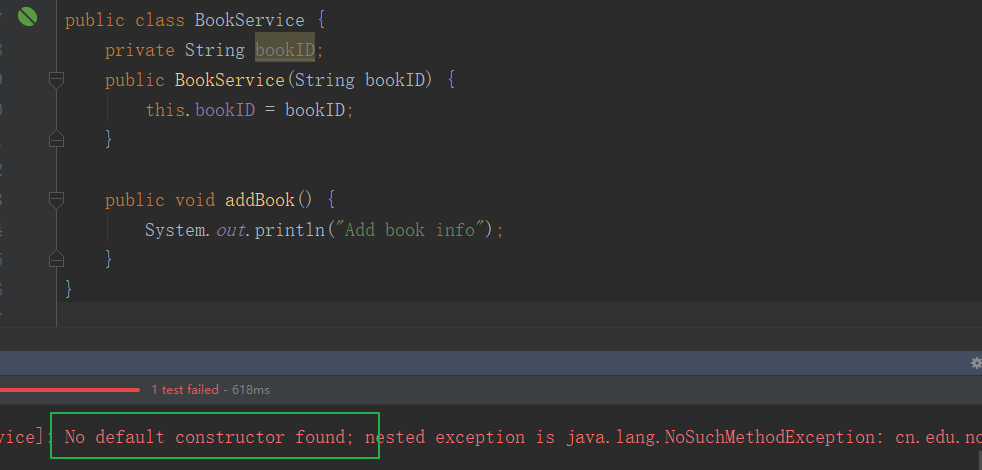


* **2、Spring Bean管理**

SpringBean三种实例化方式

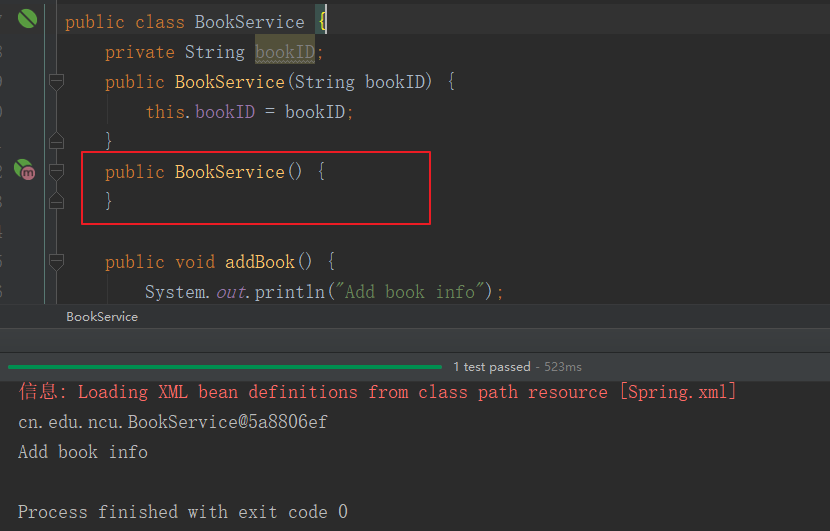
1. 使用无参数构造方法

在BookService类中添加一个带参数的构造方法，执行单元测试



错误提示为：找不到默认的构造函数

然后添加一个无参的构造方法，再执行单元测试方法

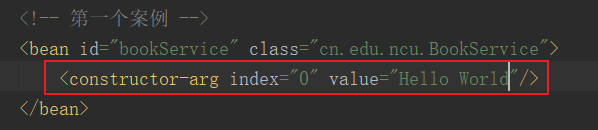


使用这种方法:

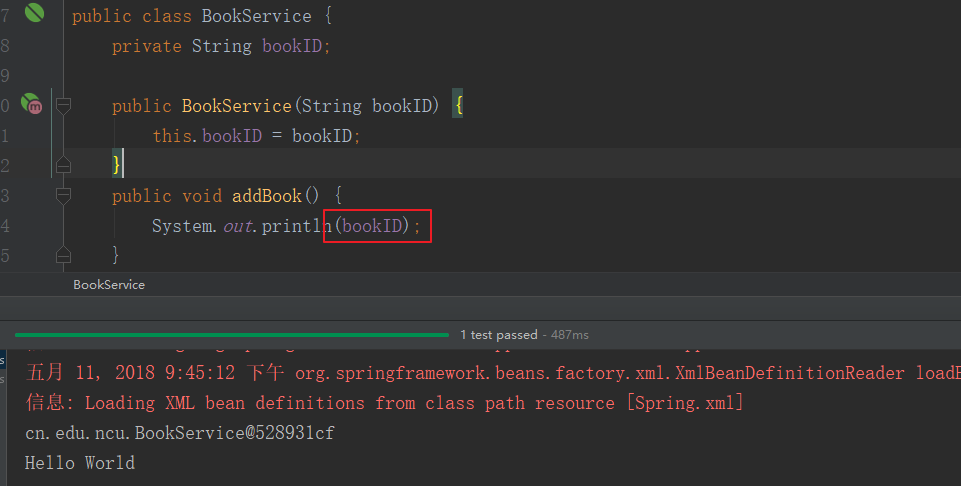
*要实例化的类中如果有构造器的话，一定要有一个无参的构造器。*

1.1有参构造

修改xml内容，指定构造函数参数



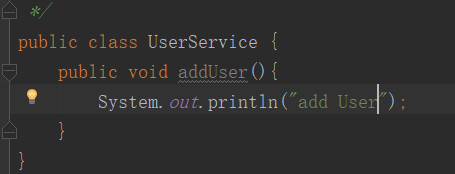
修改完xml文件之后，修改下BookService中的addBook方法，使其显示bookID的值;

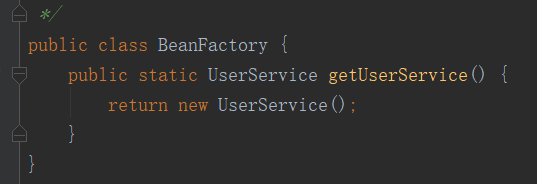


调用测试方法

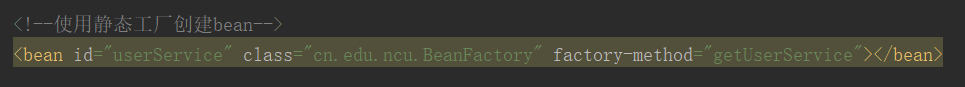
1. 使用静态工厂创建

新建UserService类和BeanFactory类，在BeanFactory中创建静态方法getUserService返回UserService对象

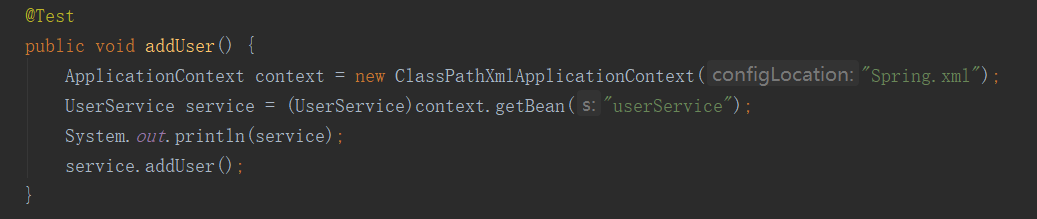


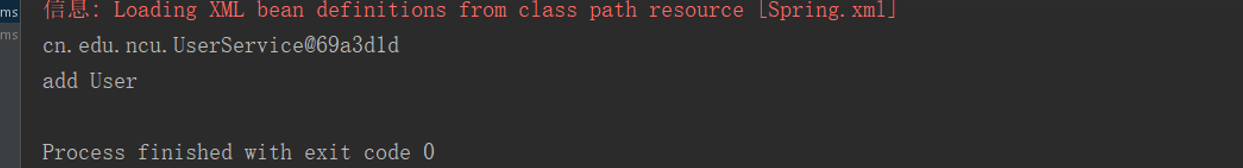


修改配置xml文件



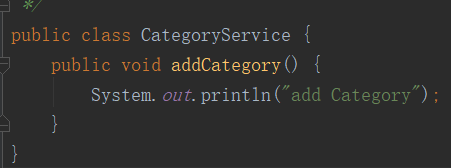
创建单元测试

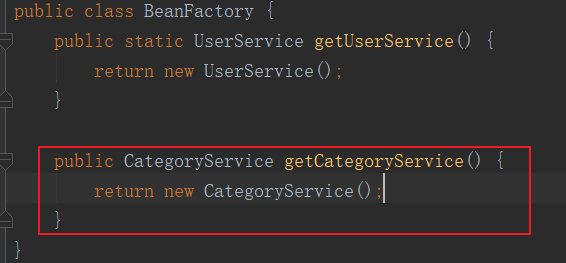




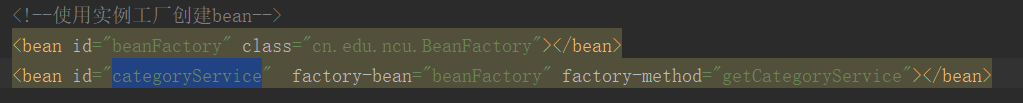
1. 使用实例工厂创建

新建CategoryService类，在BeanFactory中创建方法get CategoryService返回CategoryService对象

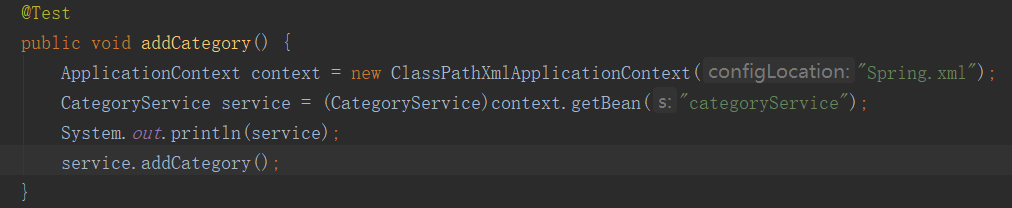


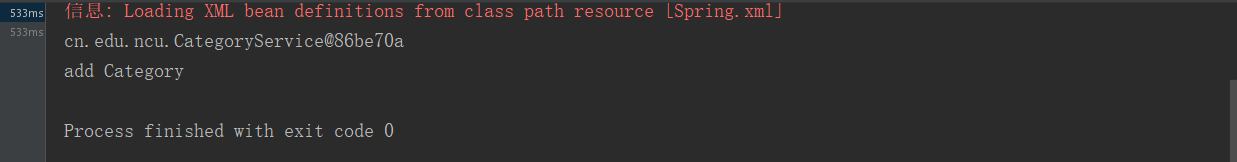


* 修改配置xml文件



* 创建单元测试





其中factory-bean要设置为工厂

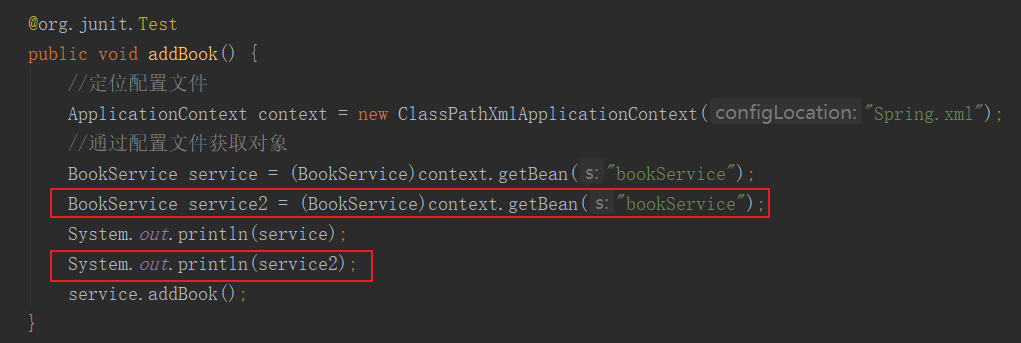
bean，factory-method方法就为实际创建这个bean的方法

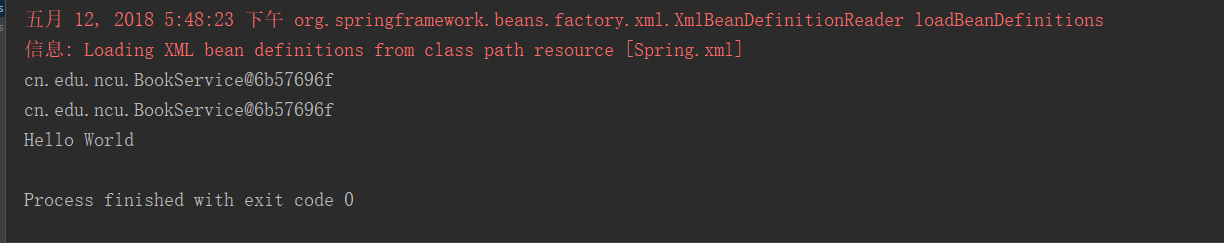
* **3、Bean的作用范围**

1. Singleton(单例)

修改BookServiceTest测试方法中的addBook方法，增加一个service的实例；

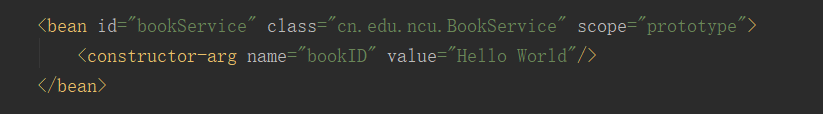
打印了两个类的hashcode，可以看到是一致的，表明这两个实例实际是一个。



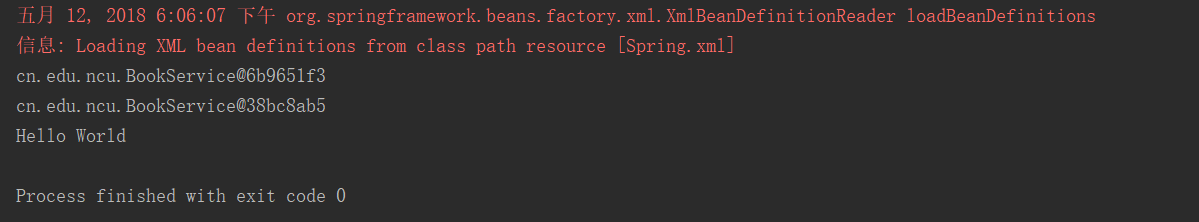


1. prototype（多例）

修改xml中BookService的scope为prototype



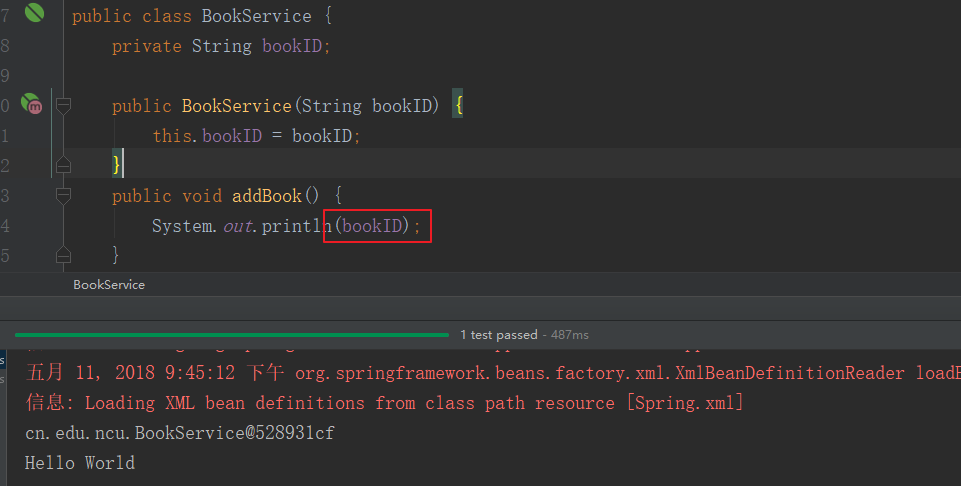
同样执行之前的测试方法



* **4、Spring的属性注入**

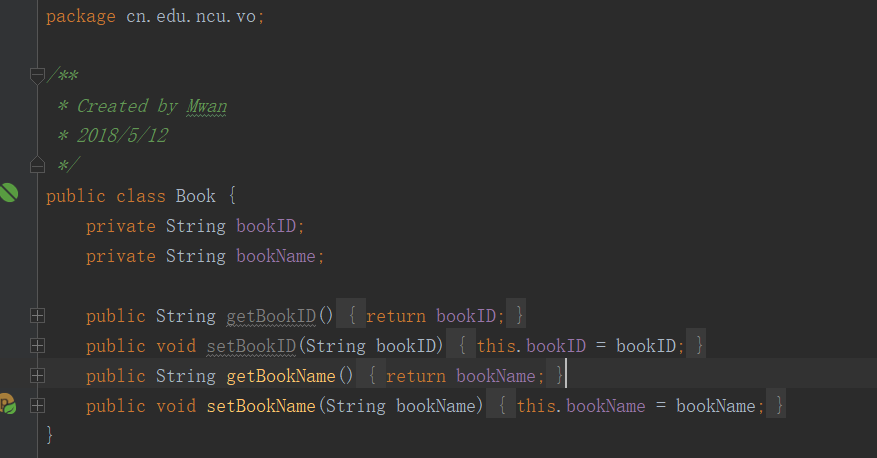
1. 有参构造

通过xml指定构造函数参数

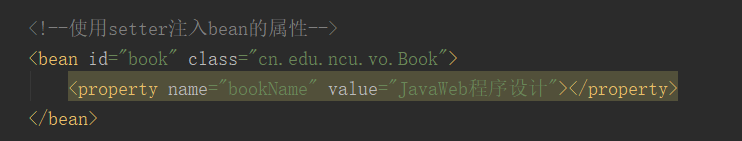


1. Setter方法

新建一个book的实体bean



* 修改xml配置



增加单元测试



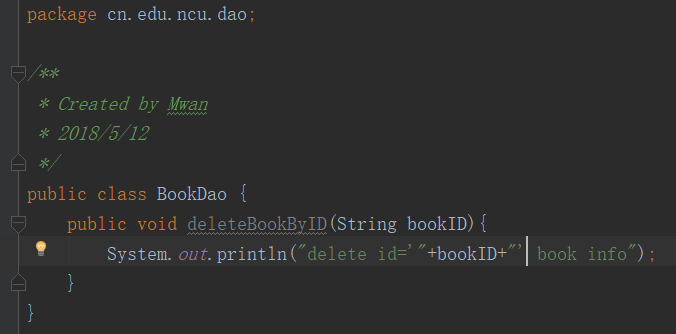
Xml的配置和之前的构造器注入的方式很像，但这里需要<property>的属性，因为我们是注入bean的属性，Name是对应bean中的属性名称，value是要注入的值

* **5、Spring Bean注入**

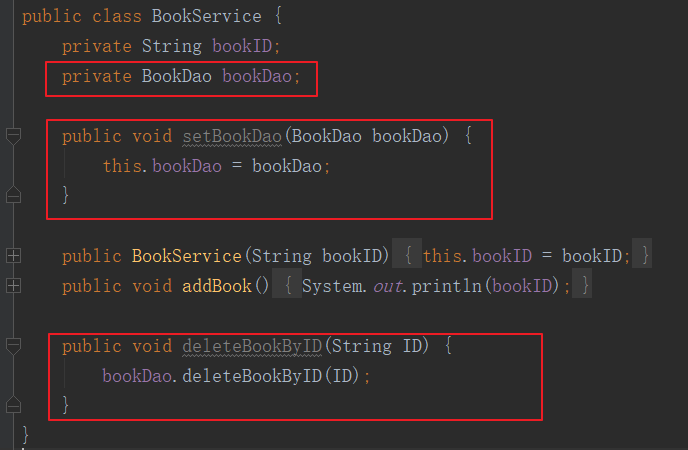
实际项目中Controller调用Service，Service调用Dao等等，在代码中需要注入相应的bean，实现类之间的解耦。

1、setter方法注入

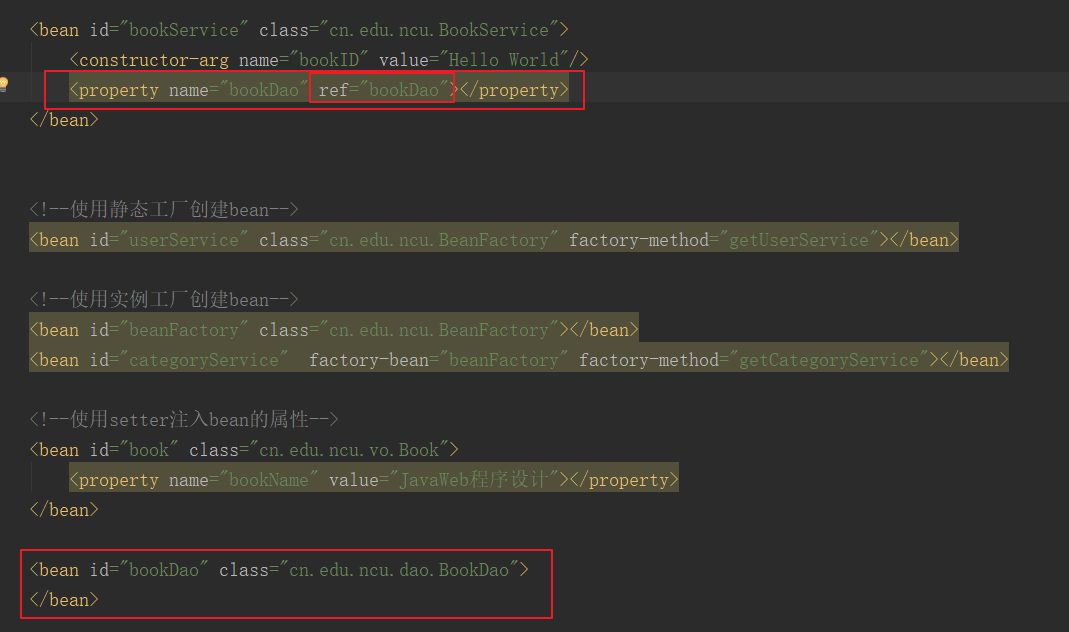
先创建一个bookDao ，为了方便只是做了一个控制台输出。



在之前的BookService中添加一个使用该Dao的方法



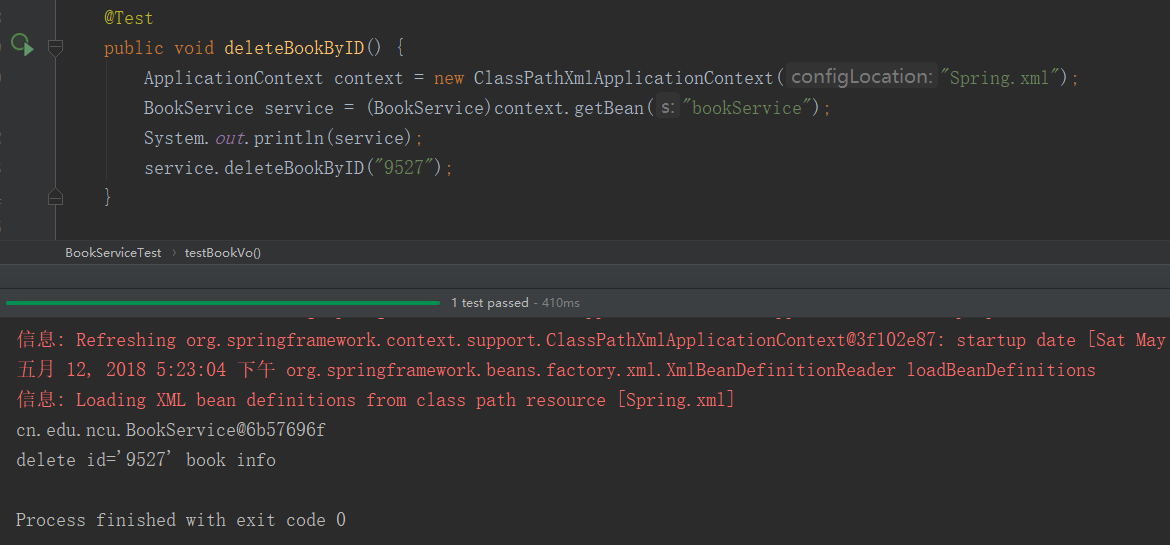
修改xml配置文件



先创建下面的bookDao的配置描述，再上方bookService中添加属性

注意：这里不再是一个字符串，而是对象，需要使用ref来指向一个对象

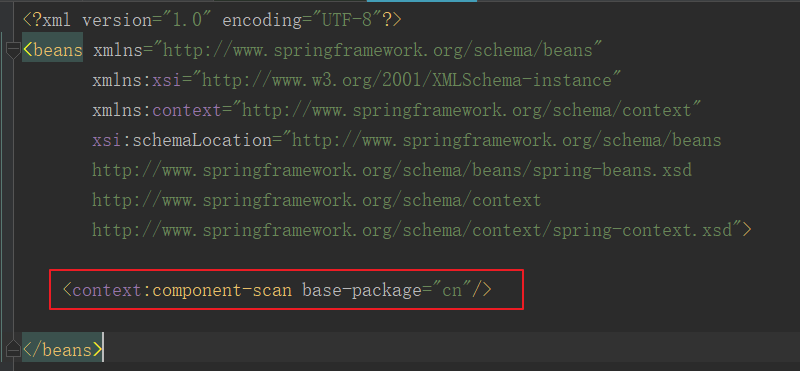
创建单元测试代码



2、注解方法注入

新建一个spring项目

在xml文件中打开自动注解扫描

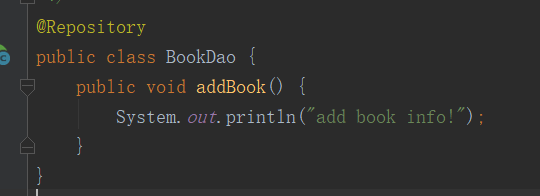


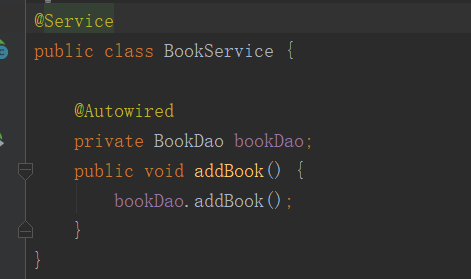
两种方式开启注解功能

<context:annotation-config> 开启注解功能

<context:component-scan>除了具有<context:annotation-config>的功能之外，<context:component-scan>还可以在指定的package下扫描以及注册javabean 。

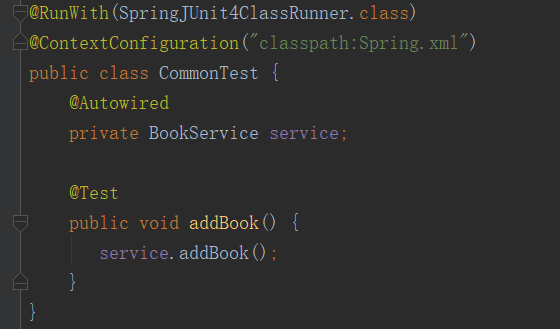
新建BookDao和BookService





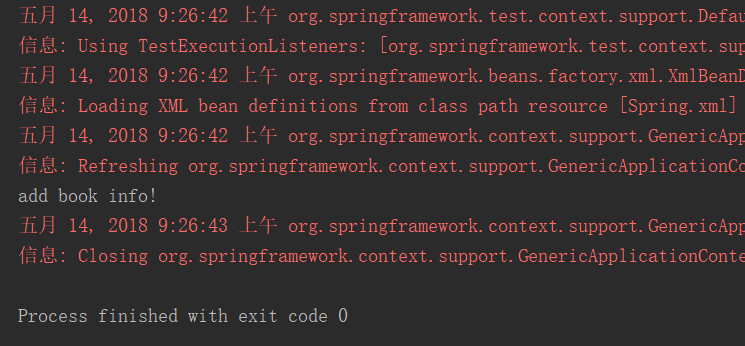
使用@Autowired自动装配bean，在属性上用这个注解，可以不用写setter方法

创建单元测试方法



使用SpringJUnit4Class进行单元测试

需要指定配置文件路径



### 三、思考题

1、修改实验7中的项目使用spring框架，实现多个模块之间的解耦。