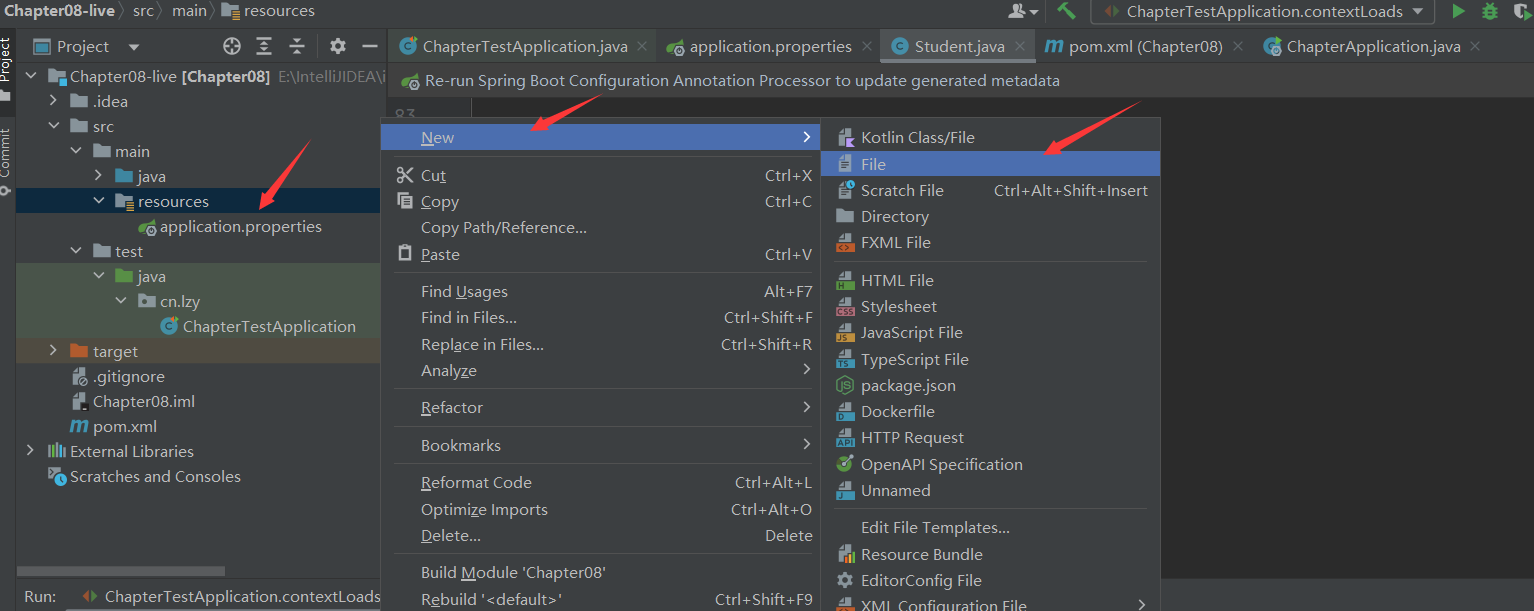
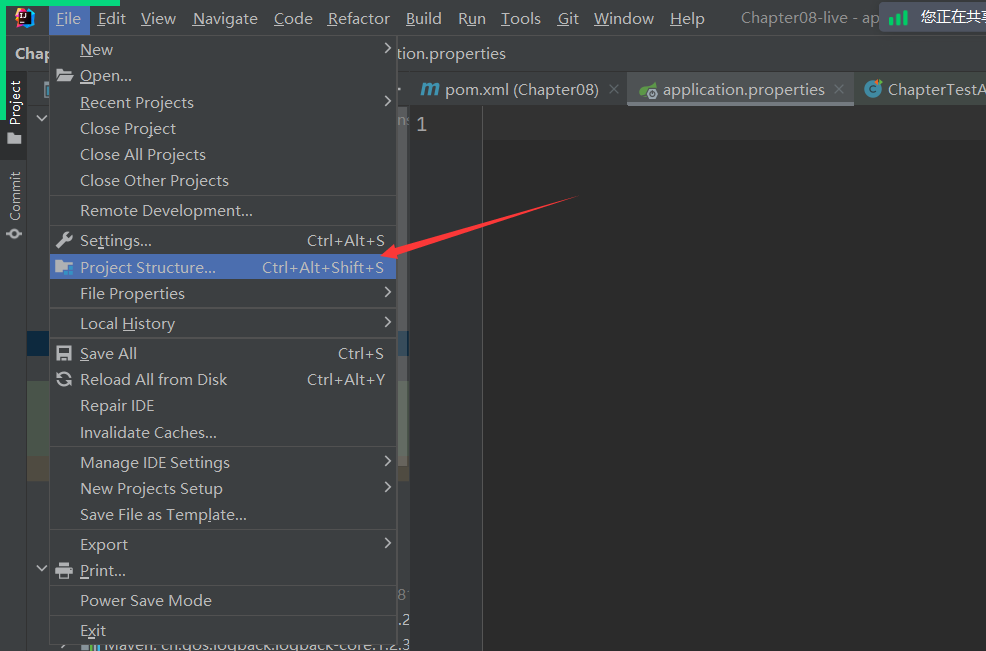
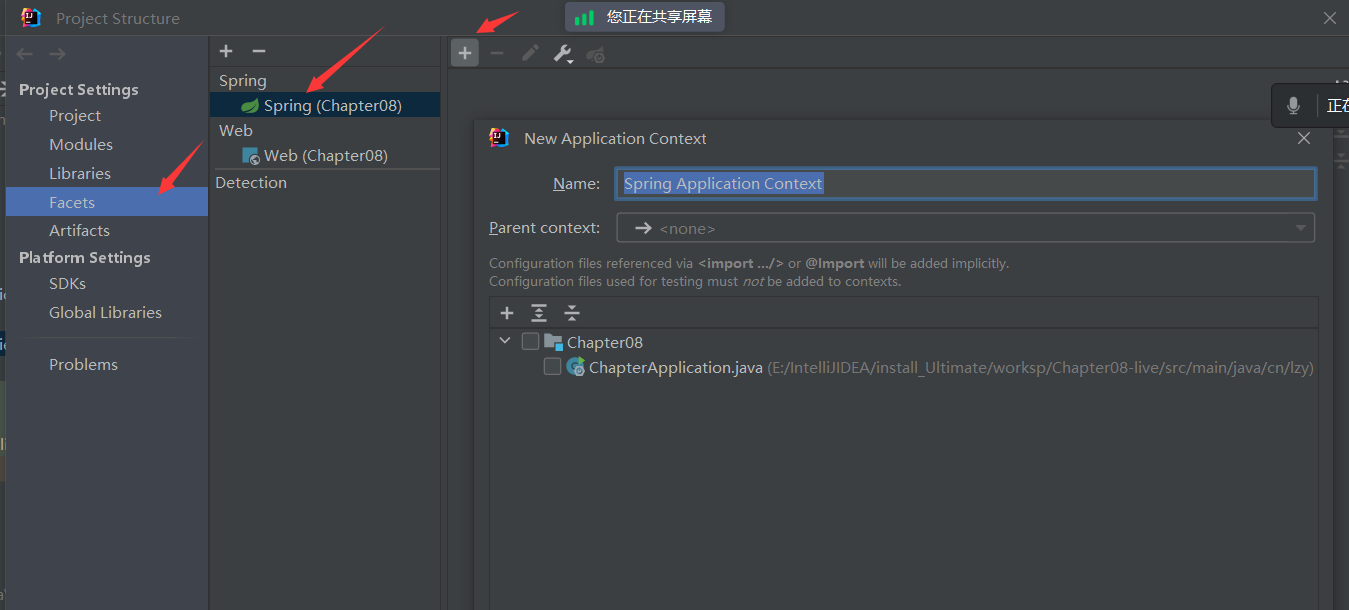
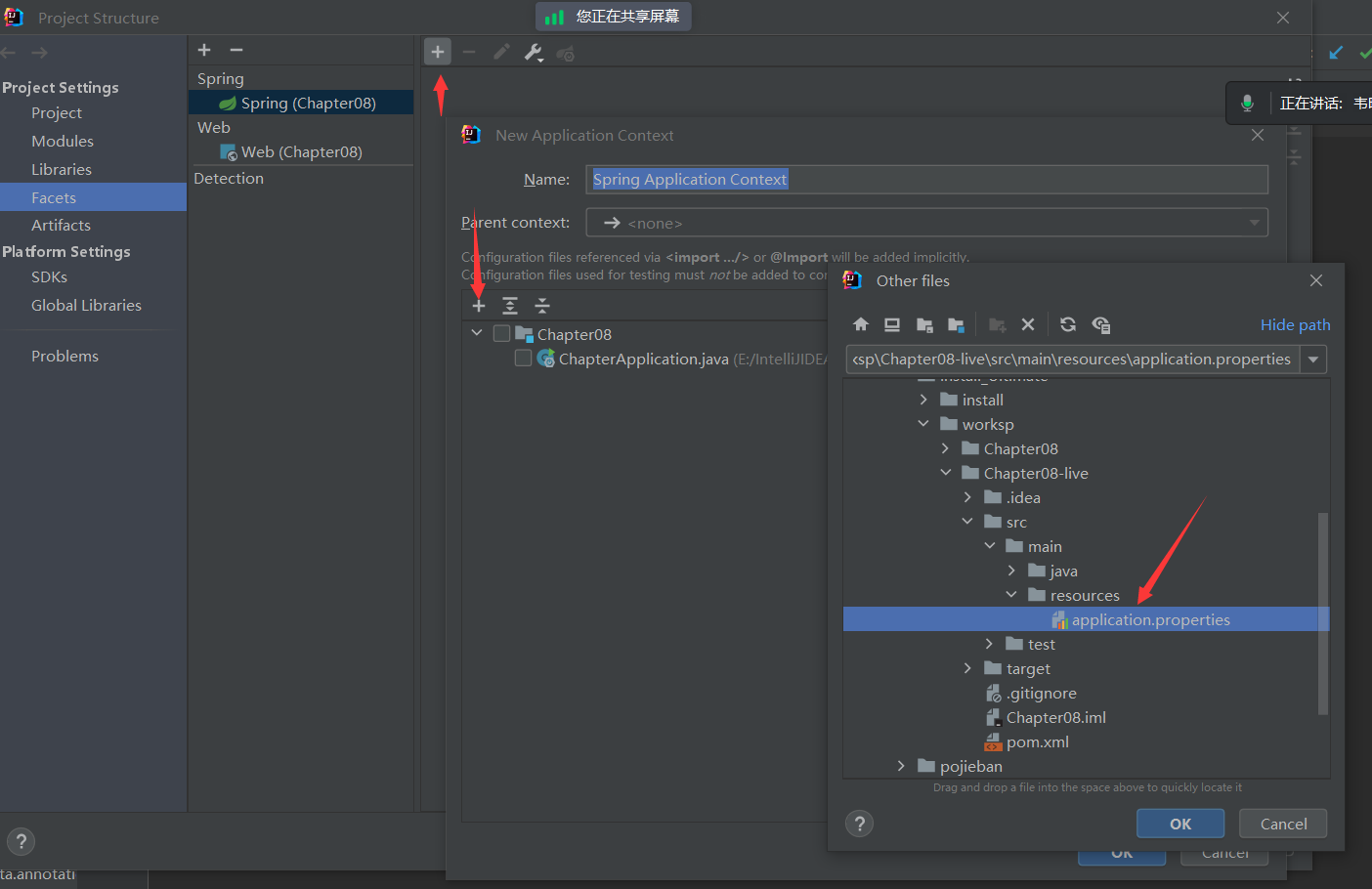
新建application.properties文件



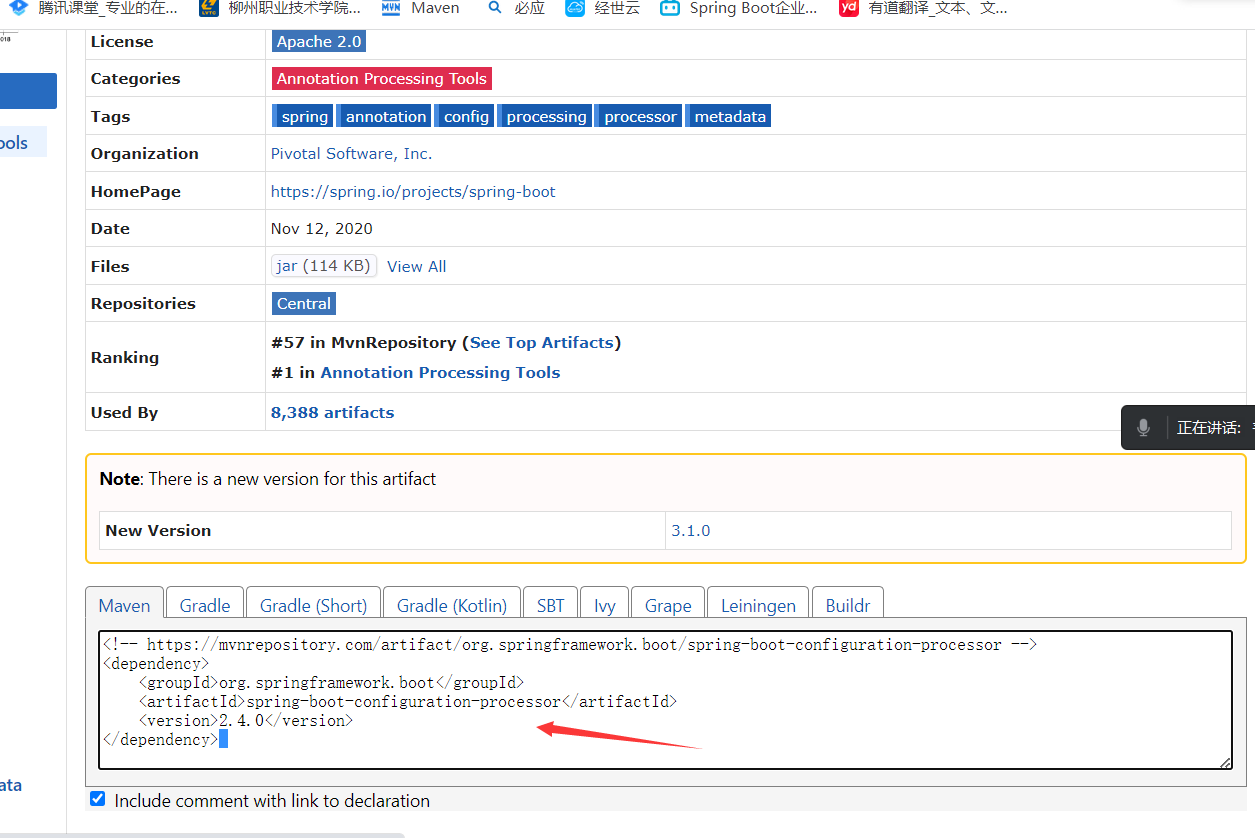
把application.properties文件变绿叶子



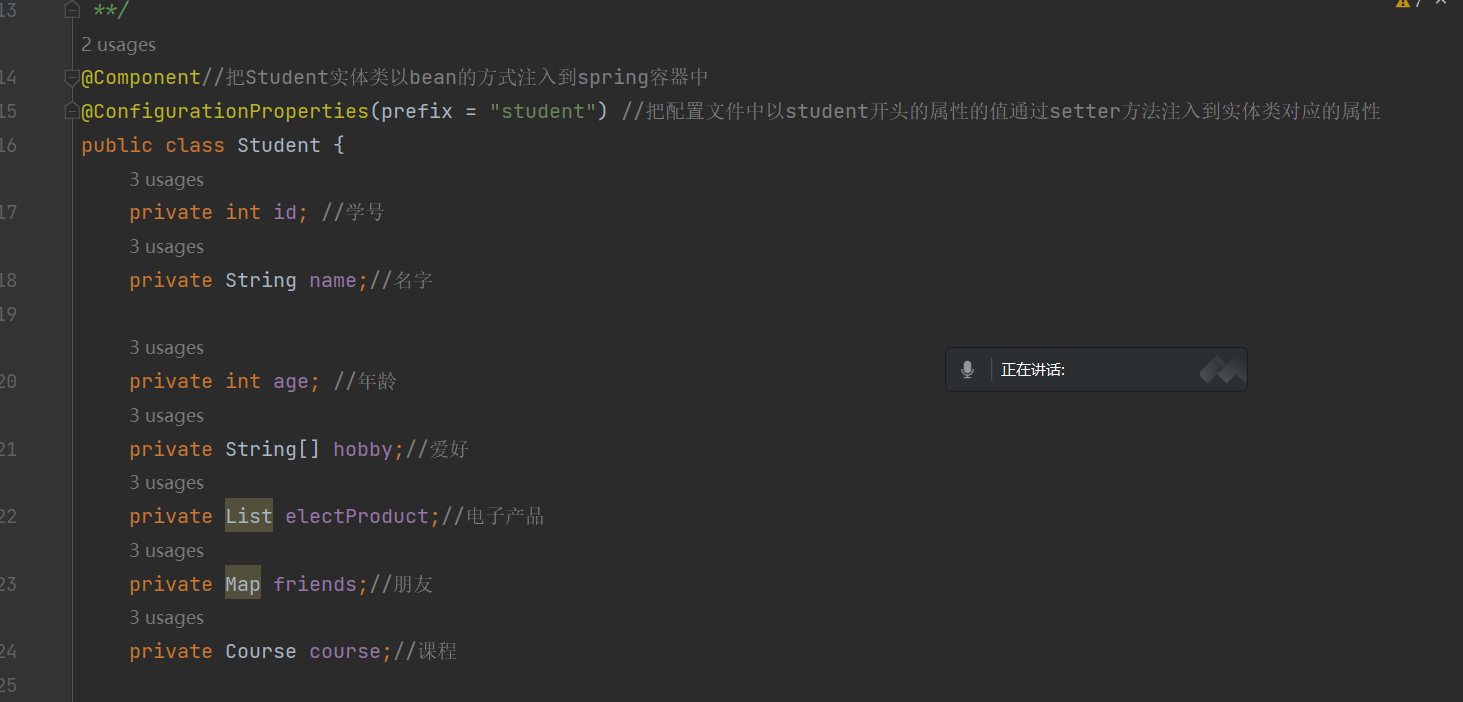


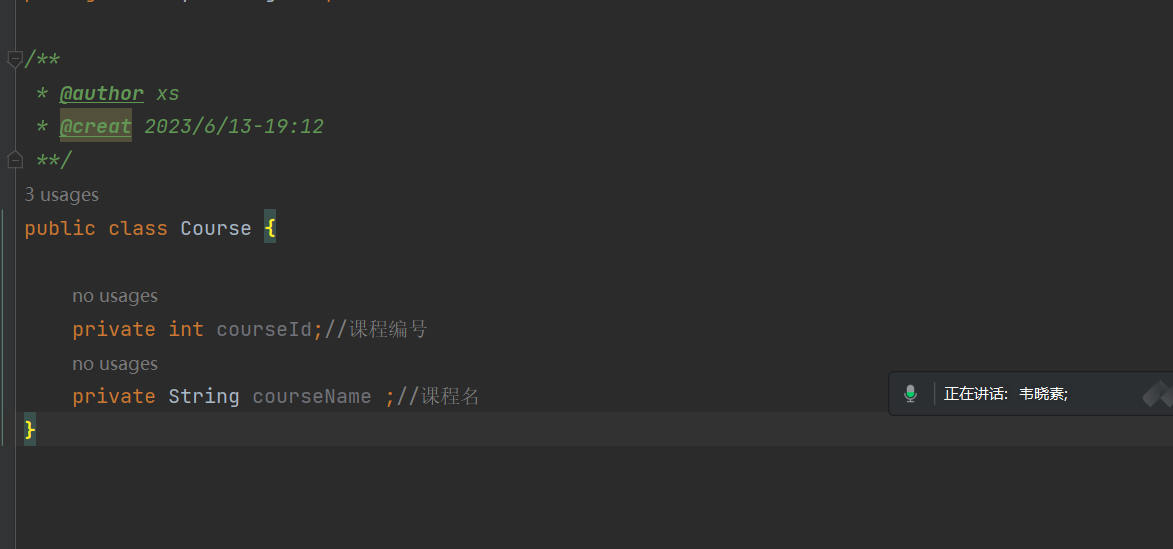


spring-boot-configuration-processor



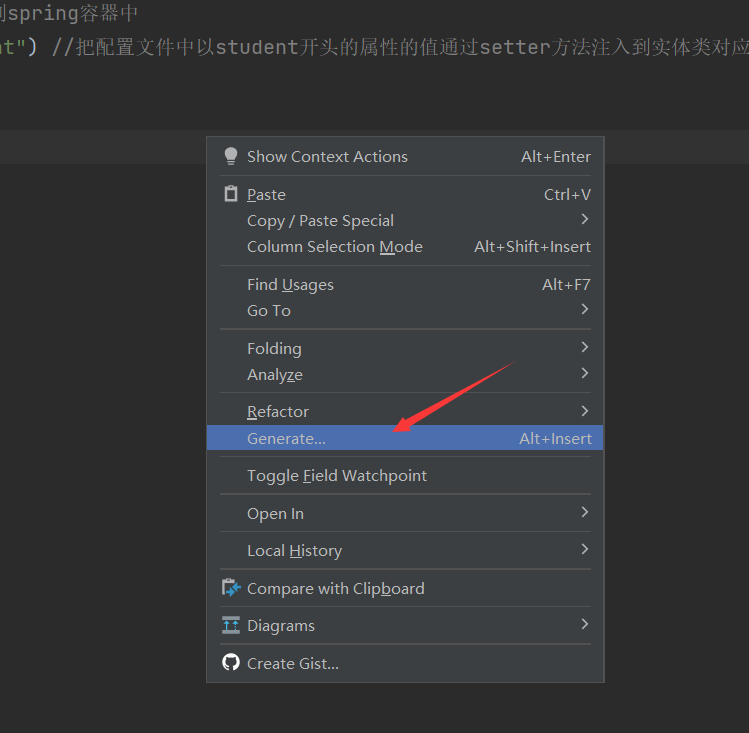
去new配置实体类Student

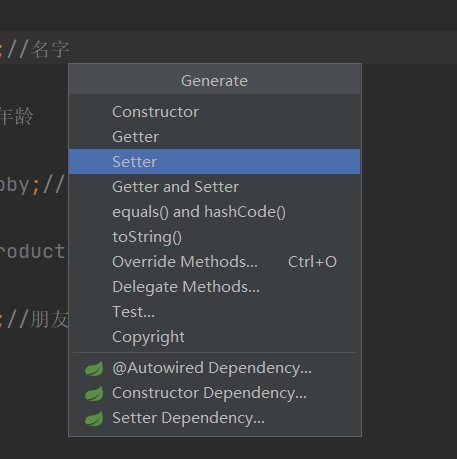




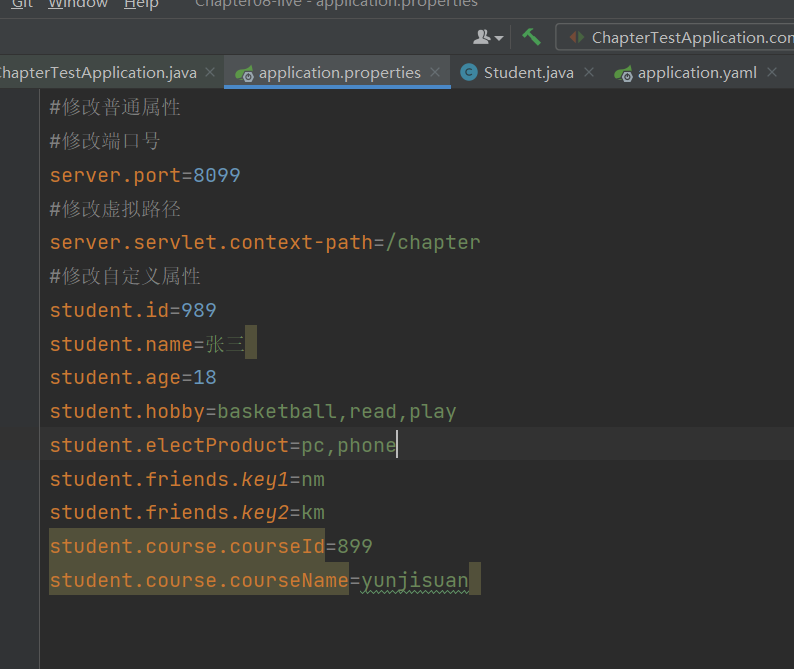
在student类里面通过alt+insert按键 或者点击右键弹出generate界面，自动生成get()、

Set()、tostring（）方法

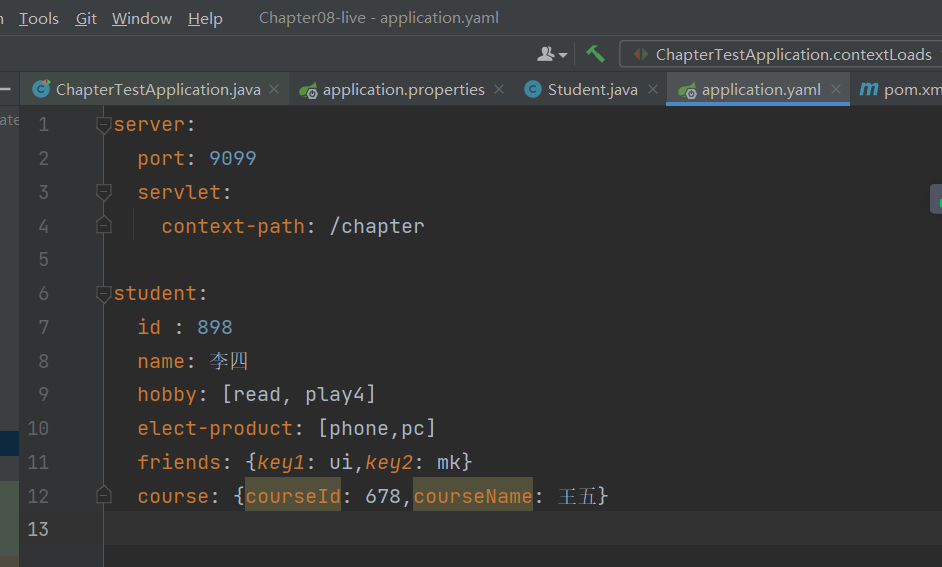




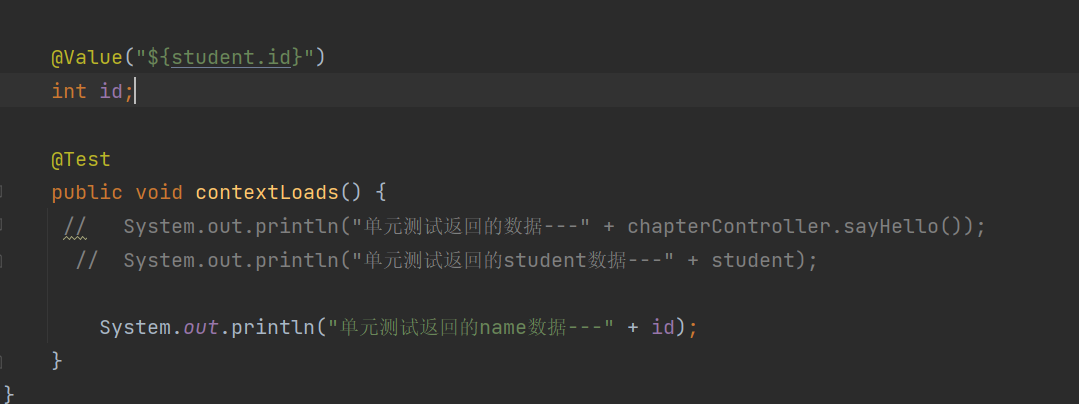
去application.properties文件里面配置student实体类的数据

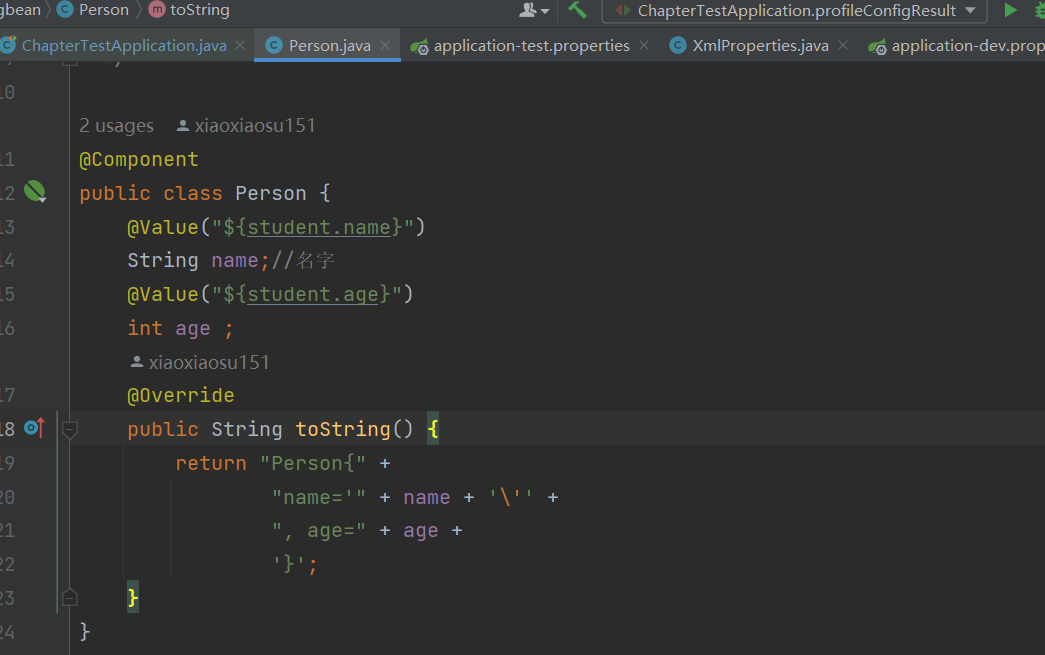


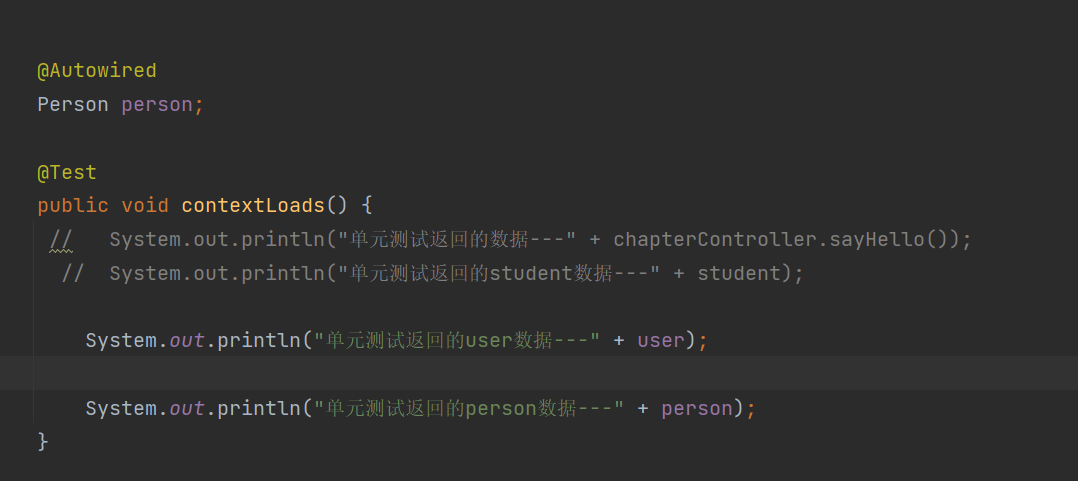
通过yaml文件给student实体类赋值



通过@Value方式去给属性赋值：







作业：

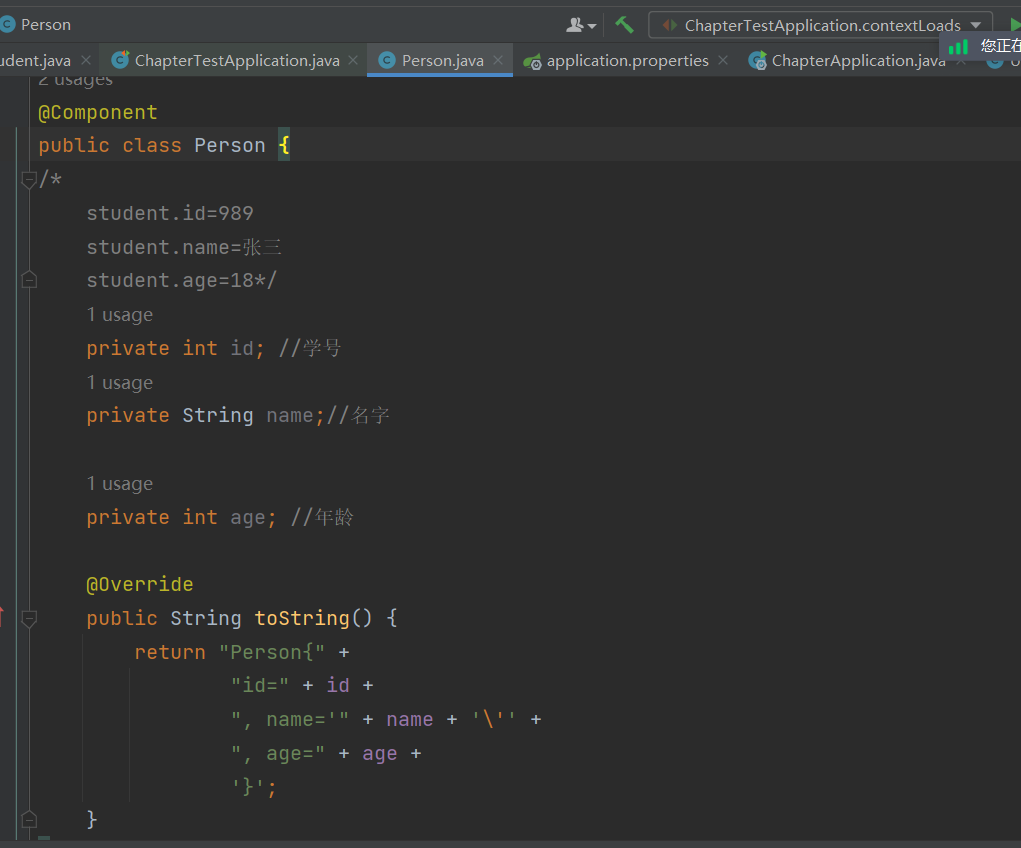
把student里面的

student.id=989

student.name=张三

student.age=18

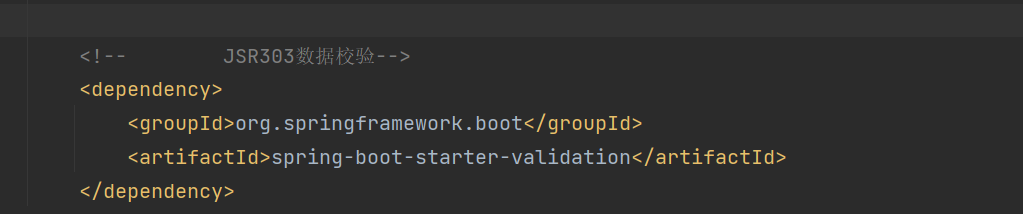
通过@value的方式赋值到person对象并输出person对象值



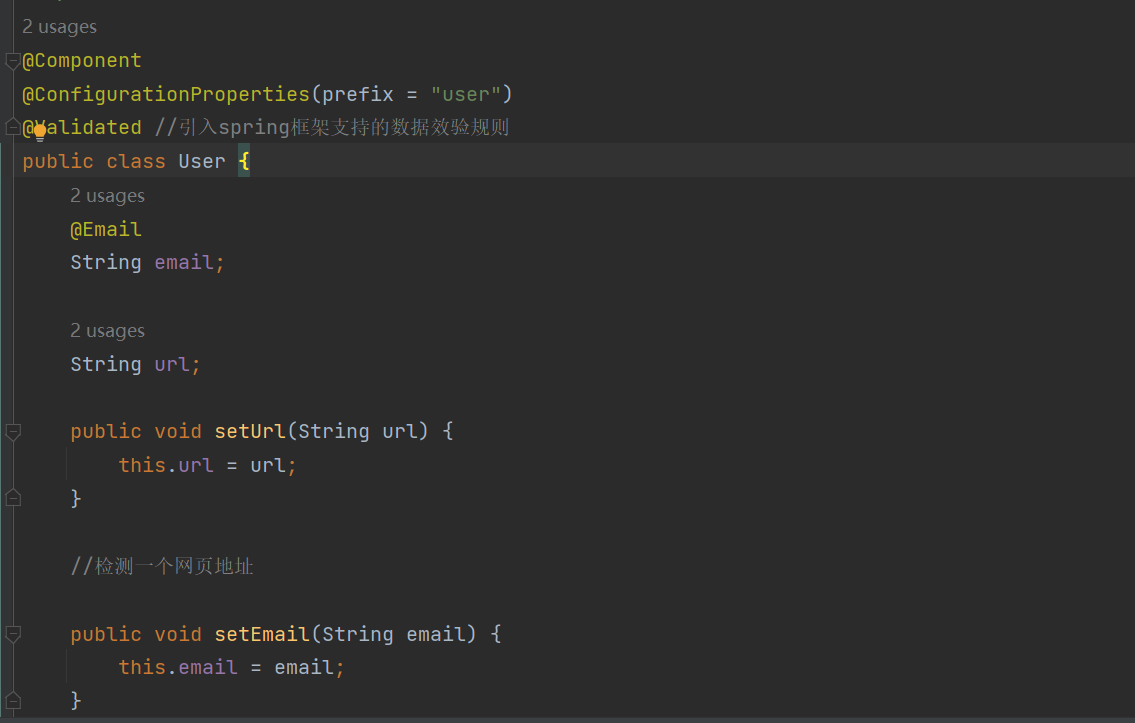
松散绑定：



引入JSR303数据较验插件



New一个user实体类来验证输入的邮箱名格式是否正确



作业：

在properties.properties文件里面 输入一个网页地址，判断是否合法

合法的地址：

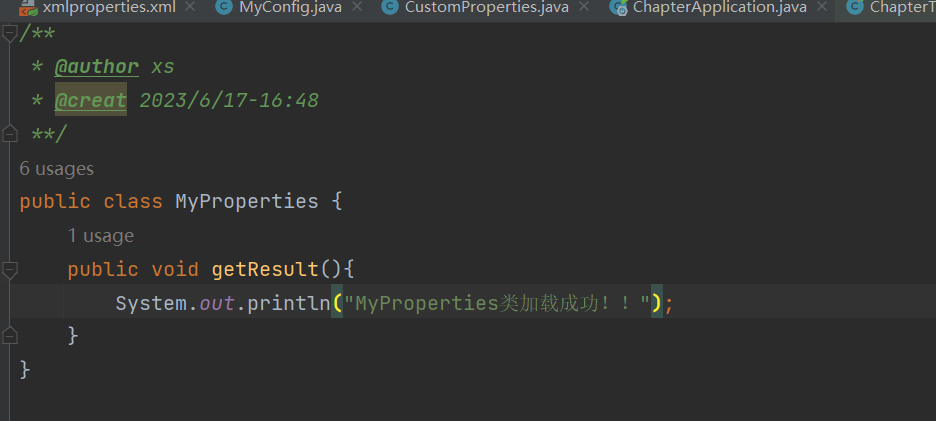
https://blog.csdn.net/wangmx1993328/article/details/81005170

不合法地址：

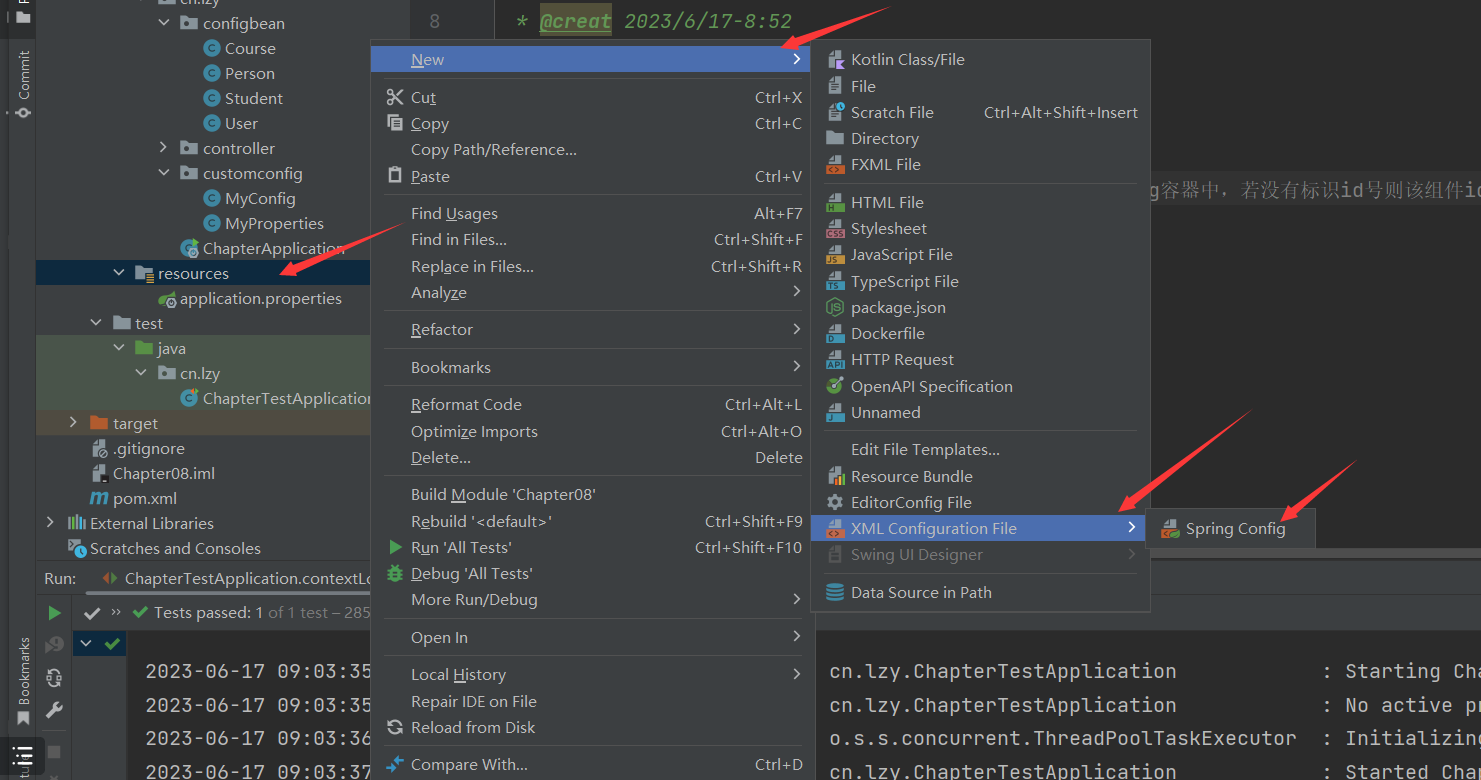
//blog.csdn.net/wangmx1993328/article/details/81005170

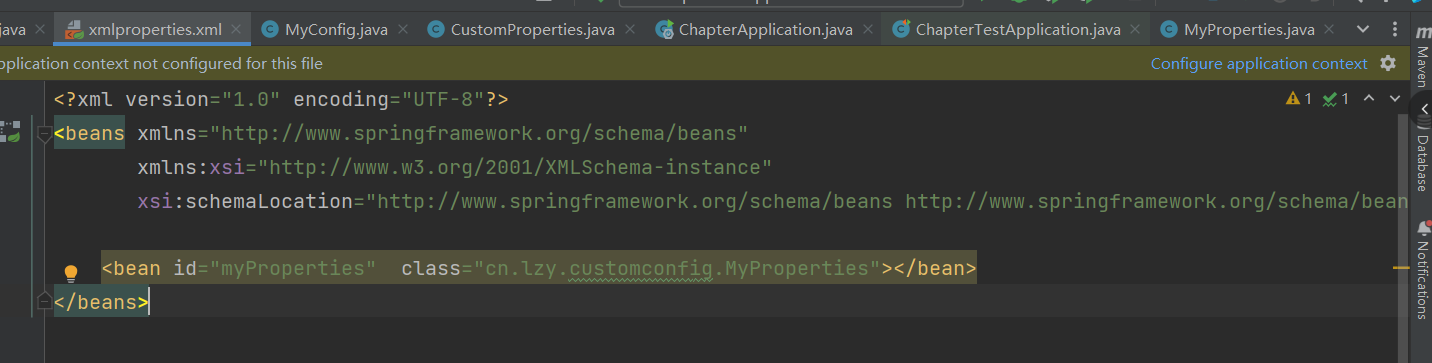
通过xml配置文件方式把目标类MyProperties注入到spring容器中

新建要注入spring容器的目标类MyProperties

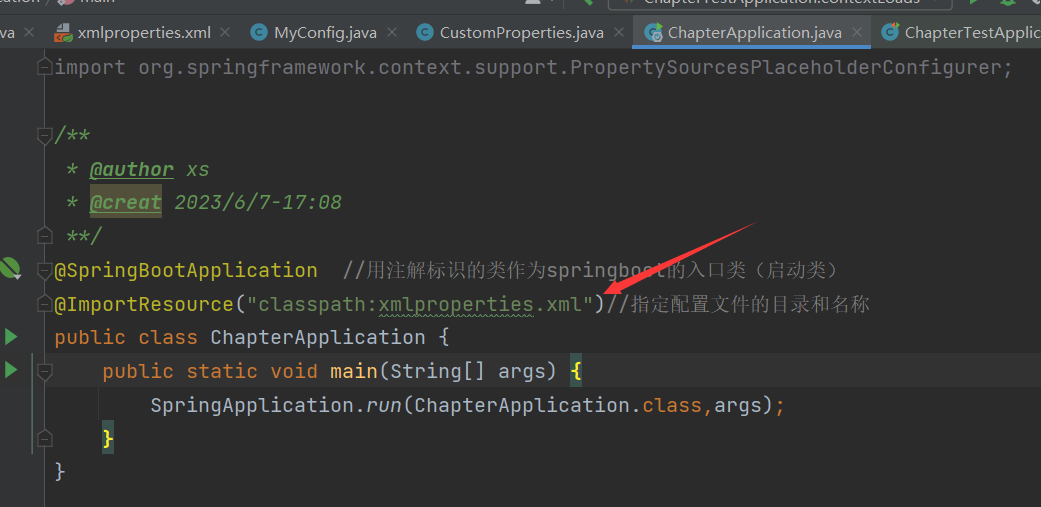


新建一个配置xml文件xmlproperties.xml

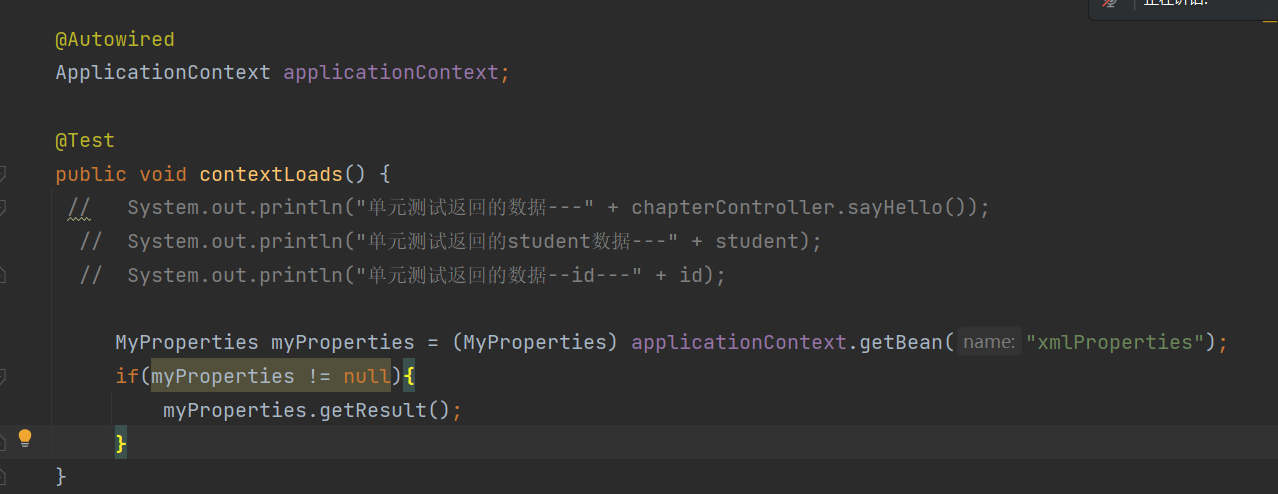




在入口类里面指定配置文件xmlproperties.xml目录和名称



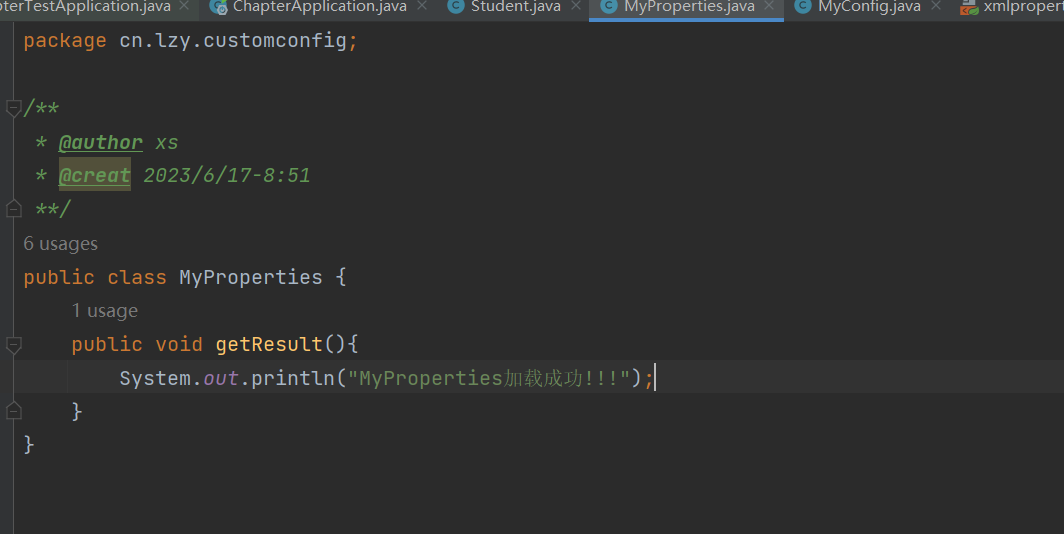
去测试类里面测试目标类MyProperties是否已经注入成功



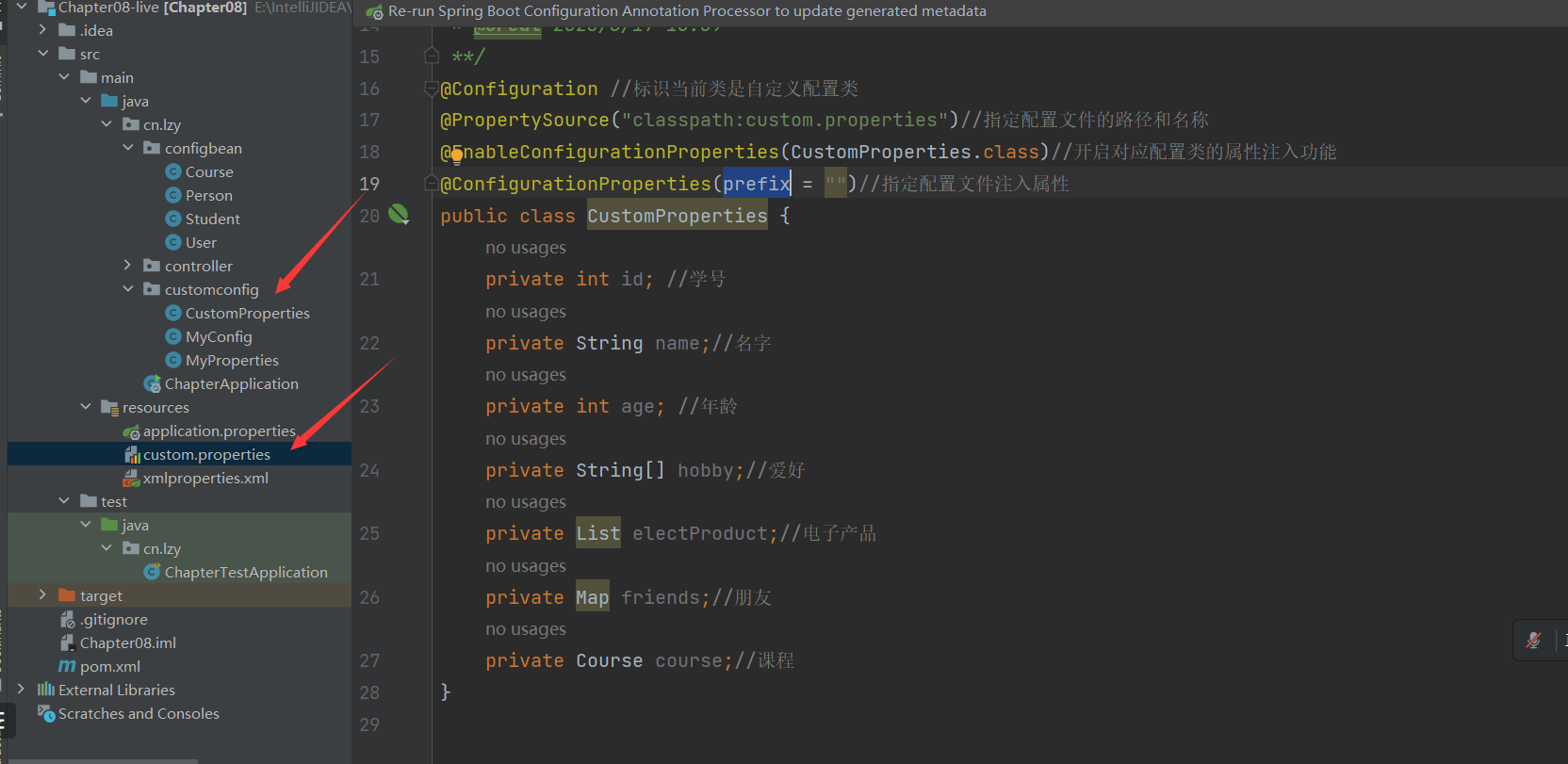
通过工具类方式把MyProperties注入到spring容器中

新建工具类MyConfig

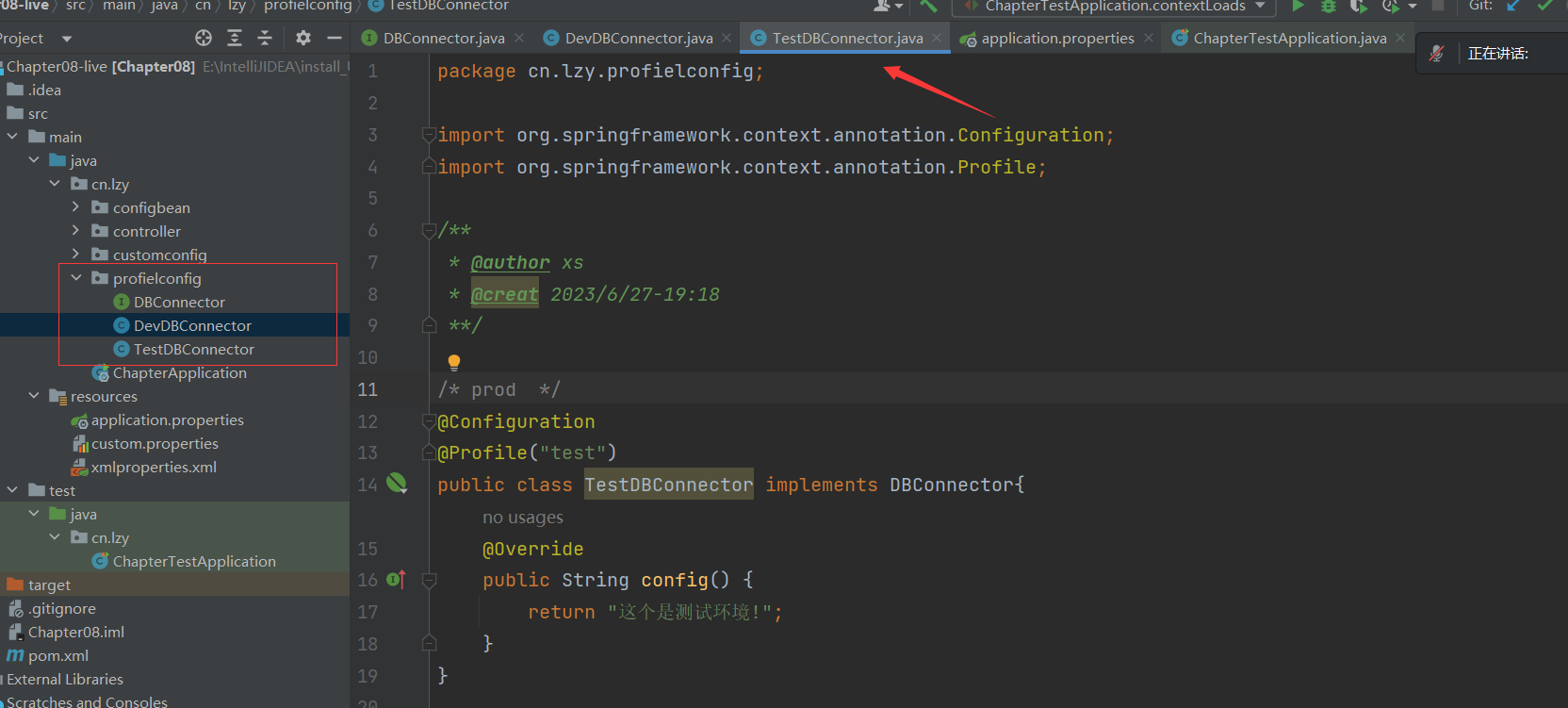


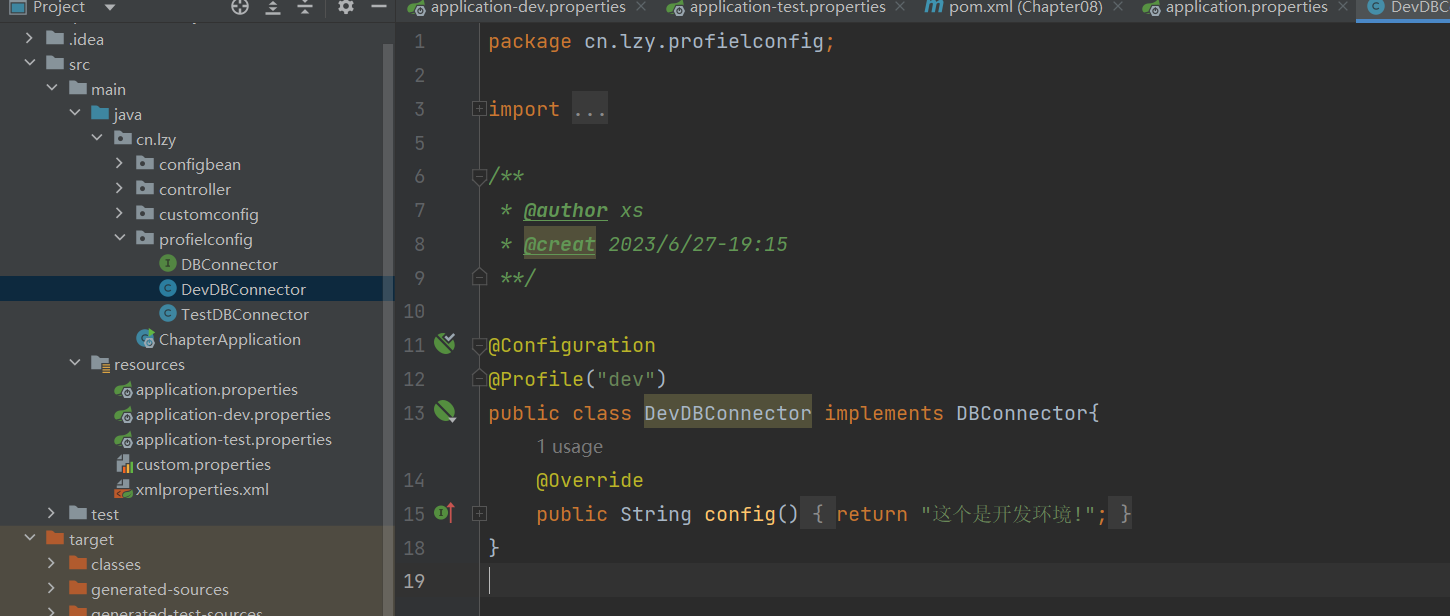


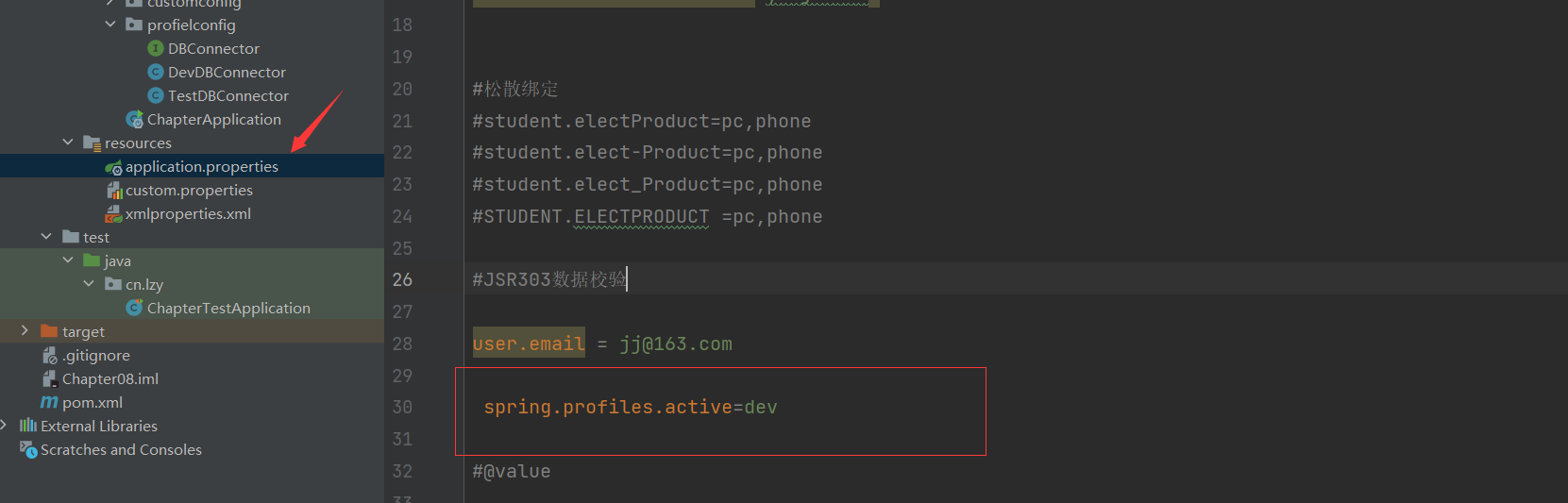
自定义一个配置文件custom.properties 和自定义实体类CustomProperties，通过配置文件把数据赋值给到对应的实体类

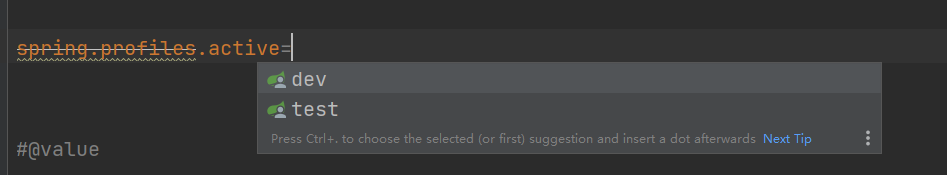


Profile配置环境：



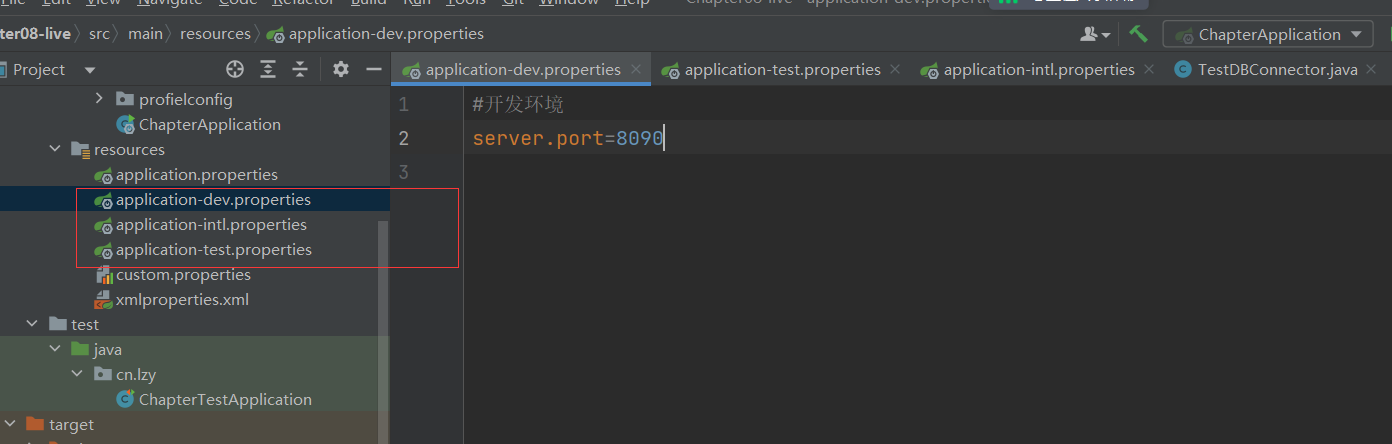






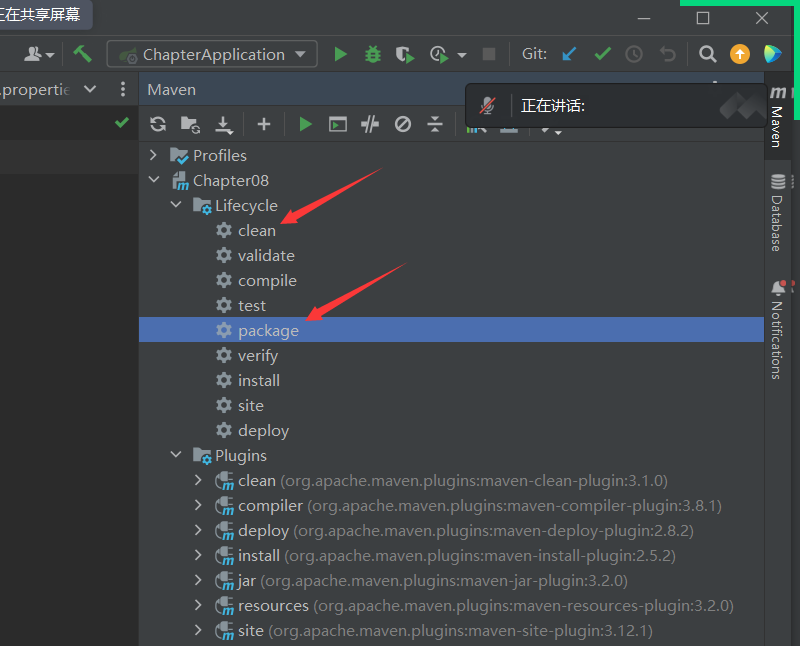
作业：

1. 测试类中打印出你配置的环境 （比如 spring.profiles.active=dev 那测试类中就输出“这个是开发环境” spring.profiles.active=test 那测试类中就输出“这个是测试环境”）
2. 添加生产环境 prod

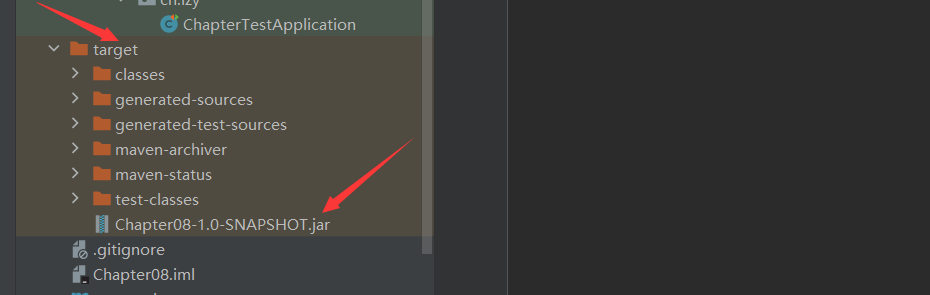


打包：

1. 双击clean
2. 双击package



生成包目录：



输入命令来切换环境：

cd target

java -jar Chapter08-1.0-SNAPSHOT.jar --spring.profiles.active=dev

