

# 实验4 Spring与MyBatis实验

---

## (一) 实验目的

---

1. 通过本实验了解MyBatis的作用
2. 通过本实验了解什么是ORM
3. 通过本实验了解SqlSessionFactory
4. 通过本实验了解SqlSession
5. 通过本实验掌握MyBatis的配置
6. 通过本实验学会用JUnit进行单元测试

## (二) 实验内容

---

1. 新建Web工程
2. 创建applicationContext.xml 文件，引入Spring的约束
3. 分层实现service层和dao层bean
4. 将MyBatis核心包和数据库驱动jar包文件导入项目
5. 配置MyBatis框架的配置文件SqlMapConfig.xml
6. 编写mapper接口
7. 配置实体映射文件xml

## (三) 实验要求

---

1. 实验过程要求每个同学根据实验指导书认真完成
2. 每个同学认真书写实验总结，不能有雷同

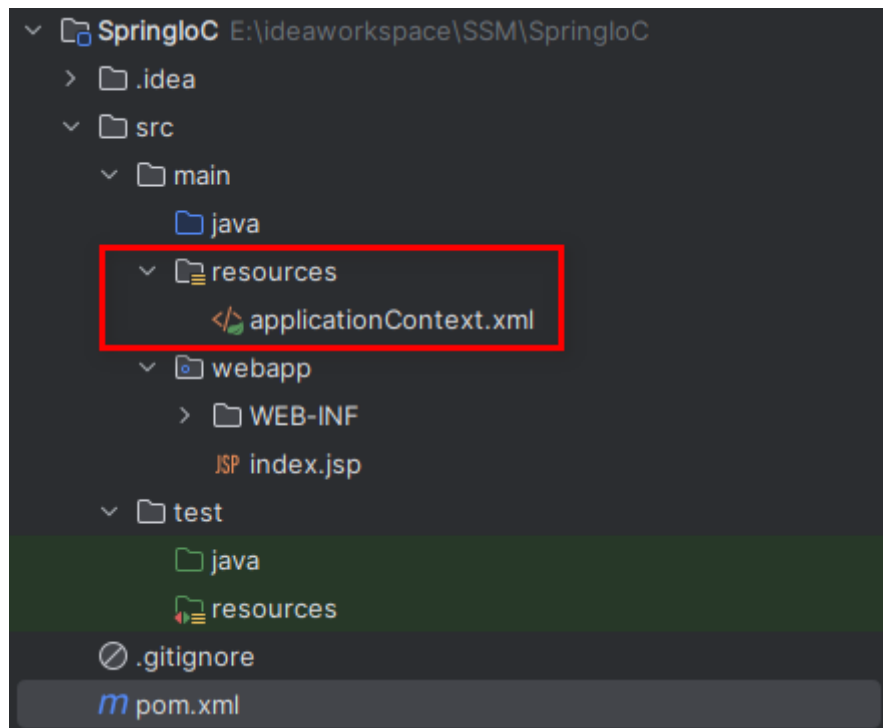
## (四) 实验步骤

---

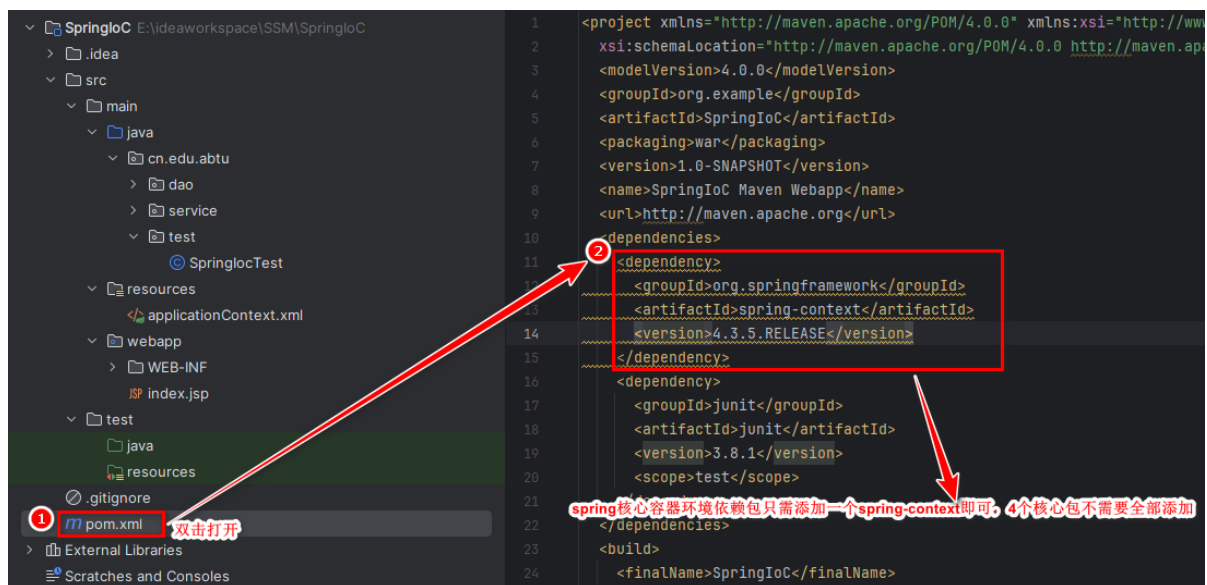
### 1. 在IDEA中创建一个Maven项目（项目名：mybatis\_test）

此步骤略，请参考 [IDEA新建Maven项目实验](#)。

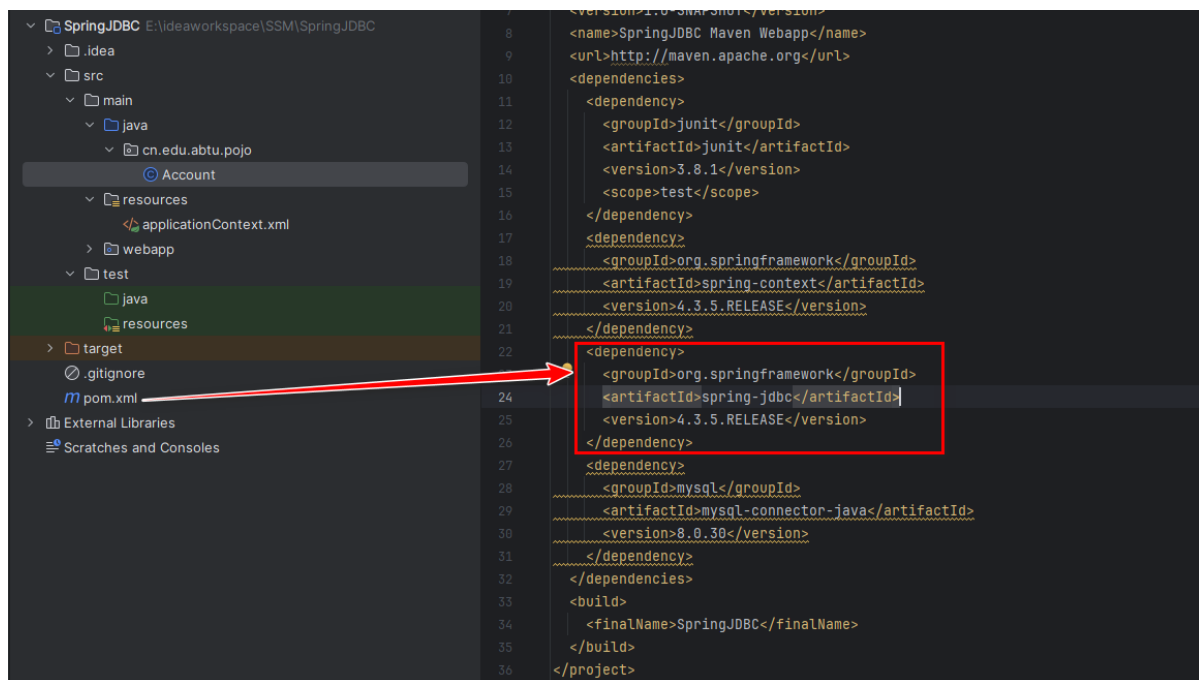
## 2. 将applicationContext.xml文件引入工程src->main->resources目录下



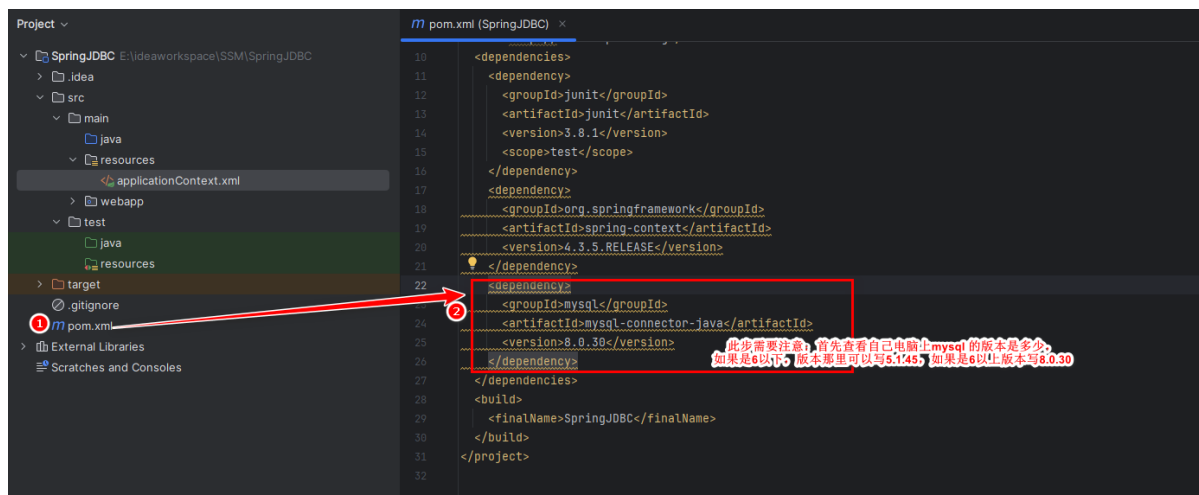
## 3. 在项目pom.xml文件中引用Spring的核心jar包



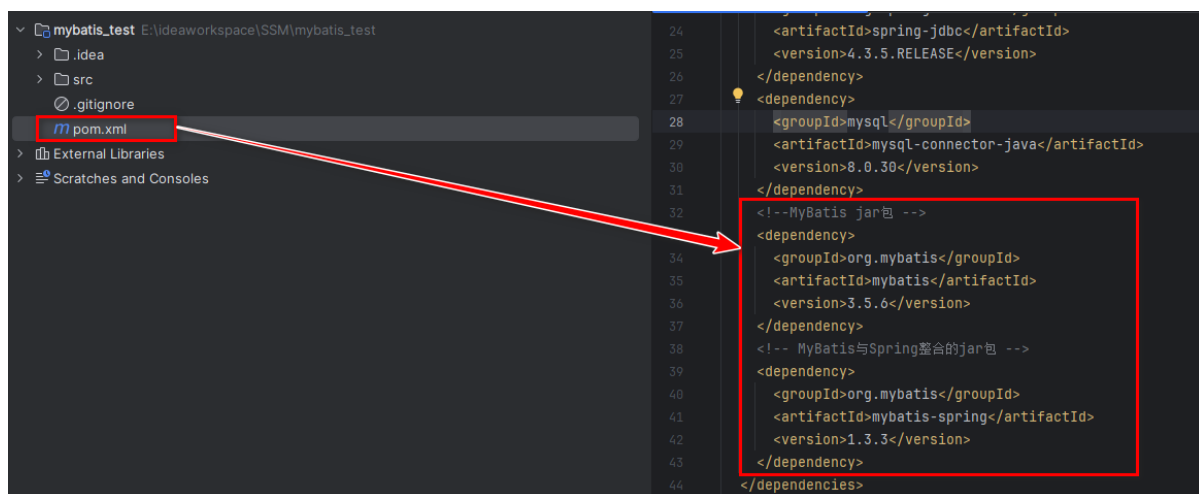
#### 4. 在项目pom.xml文件中引入Spring JDBC模块包



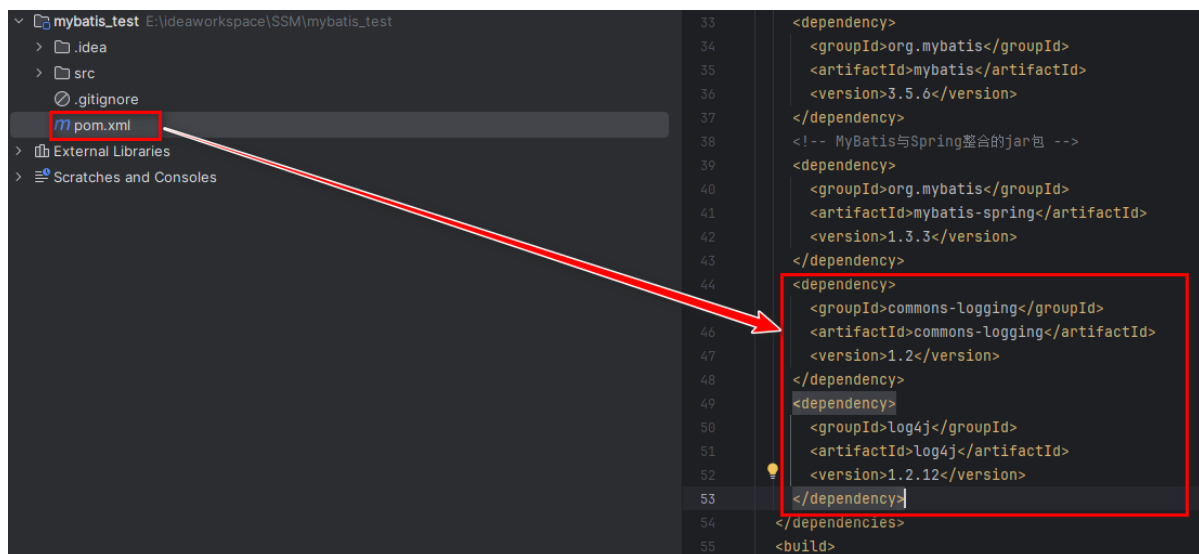
#### 5. 在项目pom.xml文件中引入mysql的驱动包



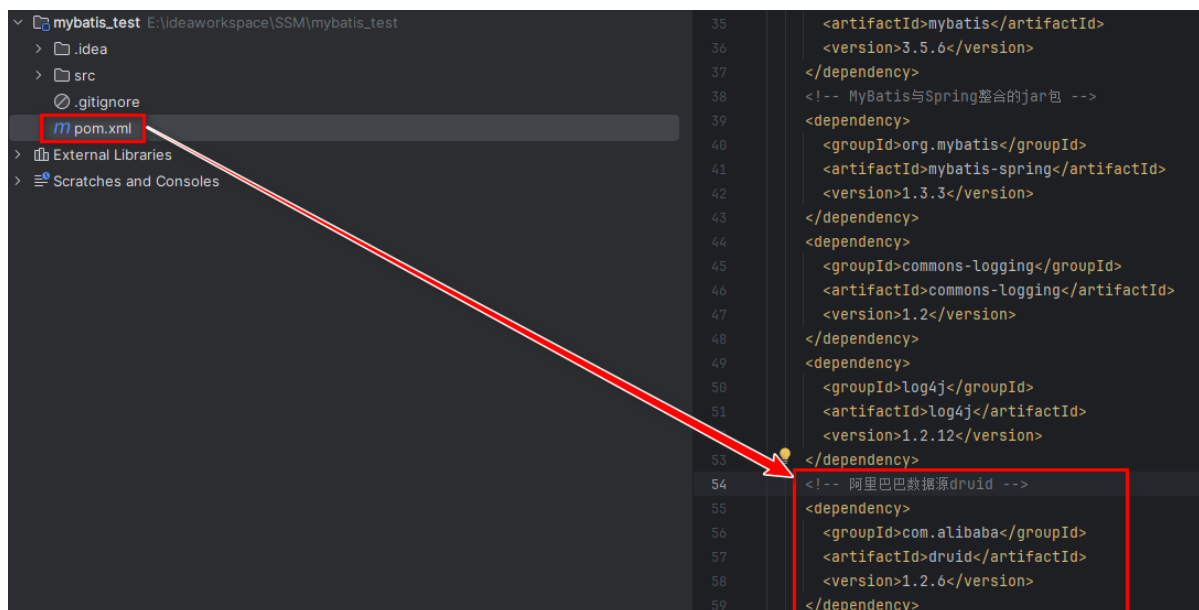
#### 6. 引入MyBatis及与Spring的整合jar包



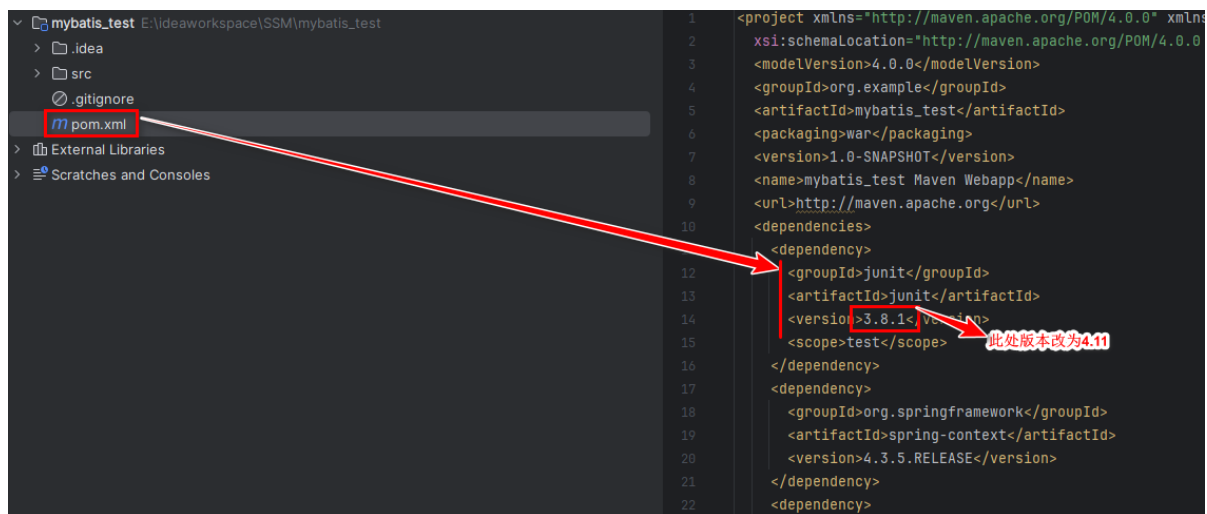
## 7. 在POM文件中引入commons-logging、log4j的相关jar包



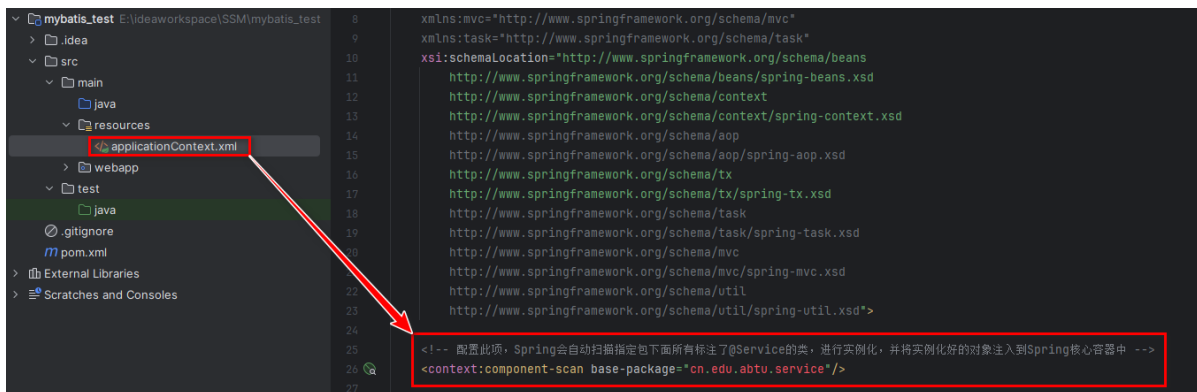
## 8. 引入阿里巴巴数据源druid



## 9. 将JUnit的版本改为4.11

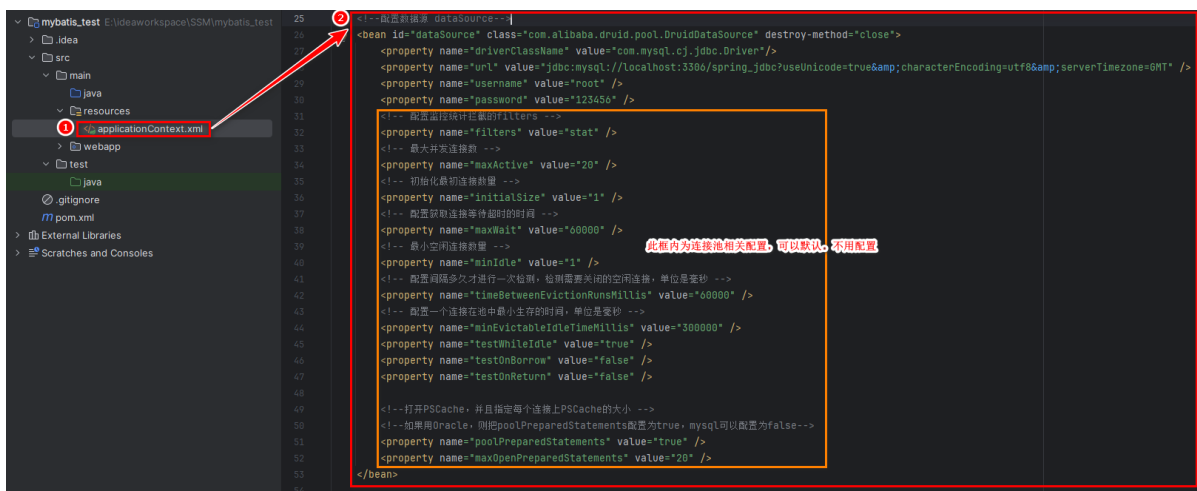


## 10. 在applicationContext.xml中开启注解扫描



此处报错先不用管，因为包还不存在。后面创建。

## 11. 在applicationContext.xml中配置连接Druid数据源



## 12. 在applicationContext.xml中配置SqlSessionFactoryBean

`SqlSessionFactoryBean` 是MyBatis框架中用于配置和创建 `SqlSessionFactory` 的Spring Bean。

`SqlSessionFactory` 是MyBatis的核心接口之一，用于创建会话( `SqlSession`)对象，该对象用于执行SQL语句。

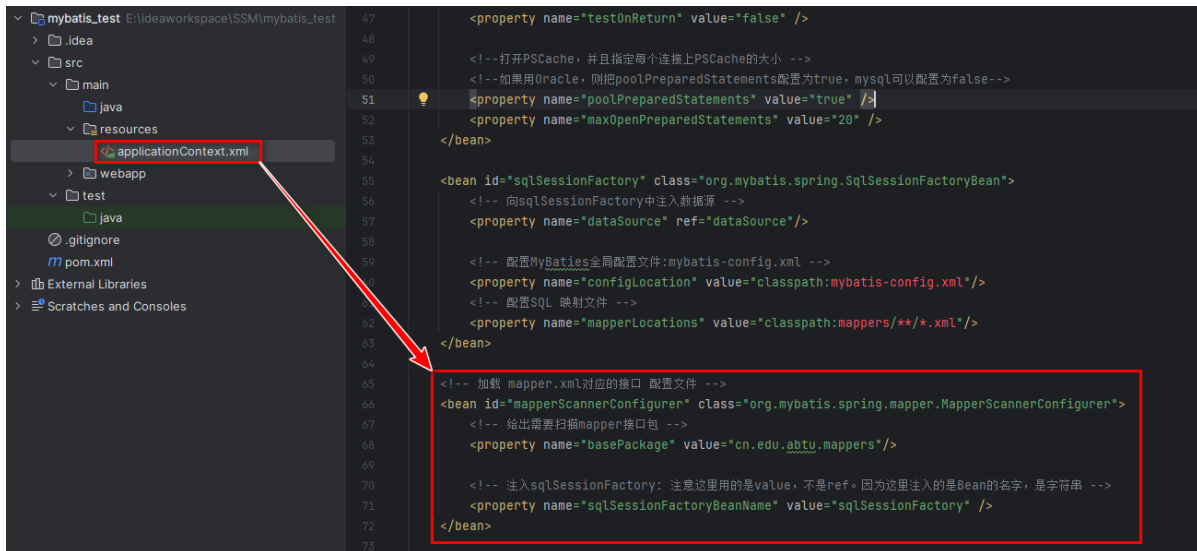
SqlSessionFactoryBean的基本的方法：

1. `setDataSource(DataSource dataSource)`：
  - 通过该方法设置数据源，指定MyBatis的数据库信息。
2. `setConfigLocation(Resource configLocation)`：
  - 设置MyBatis配置文件的位置。
3. `setMapperLocations(Resource mapperLocations)`：
  - 设置MyBatis映射文件（Mapper）的位置。

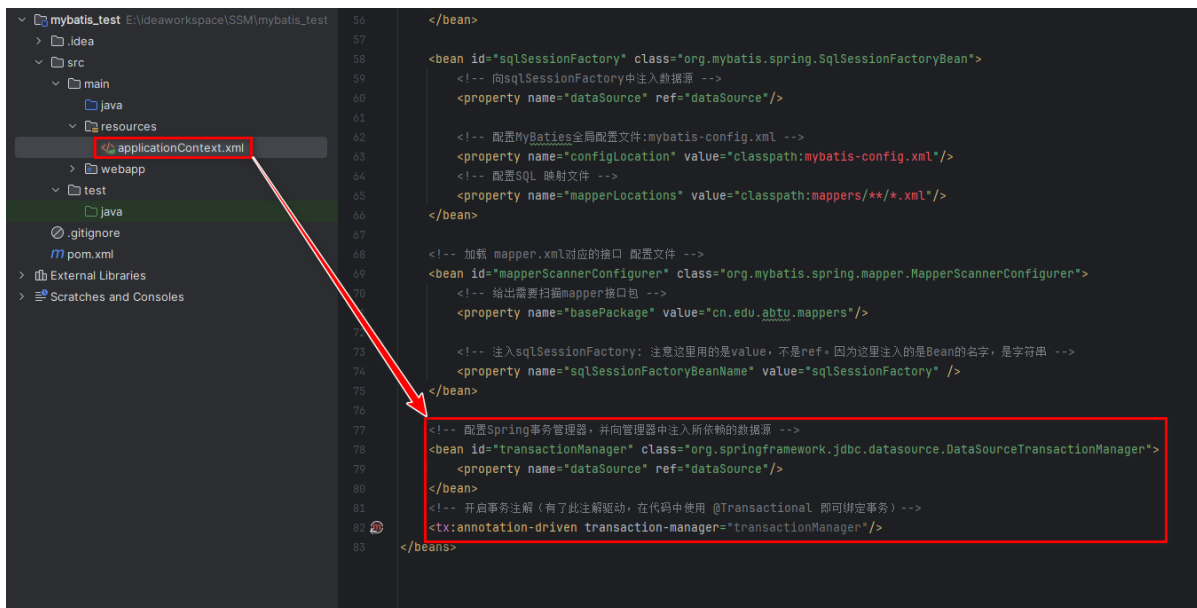


## 13. 配置MapperScannerConfigurer

*MapperScannerConfigurer*可以为指定的包下的所有的Mapper接口创建代理，并注入Spring核心容器。

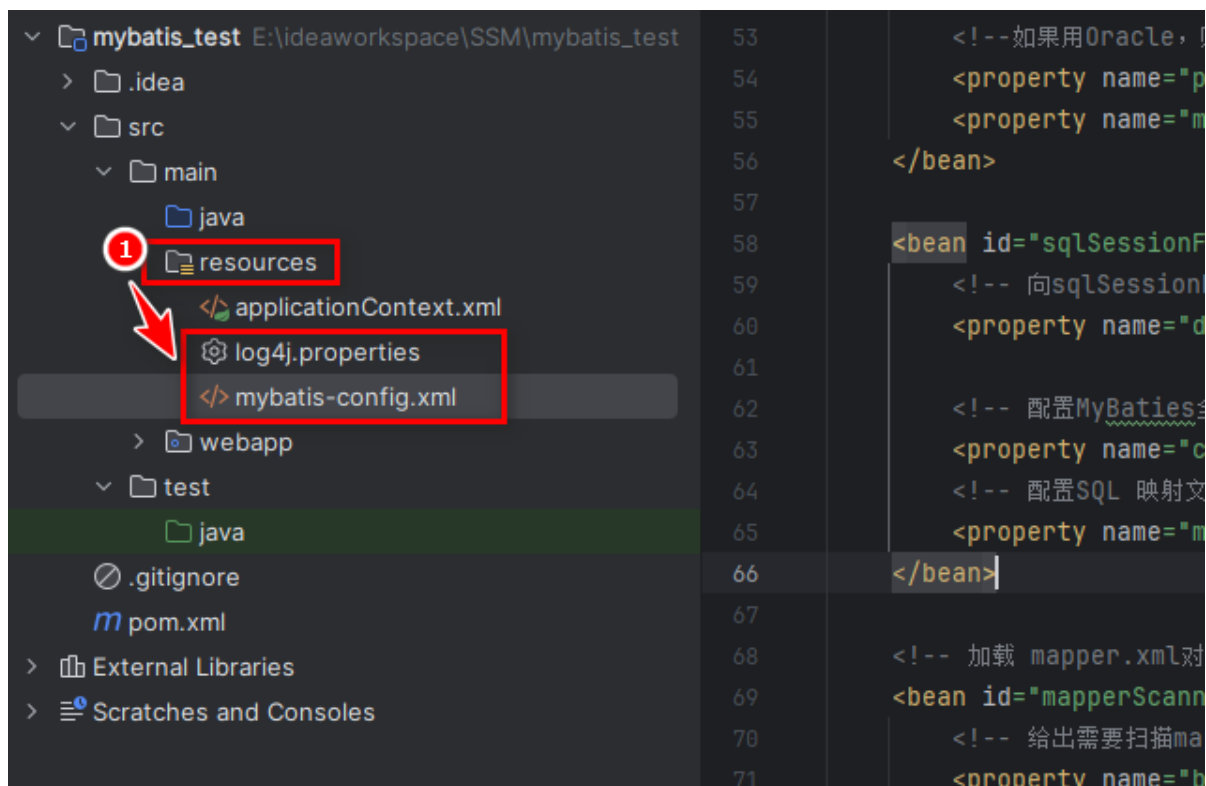


## 14. 在applicationContext.xml中配置事务，并开启事务注解

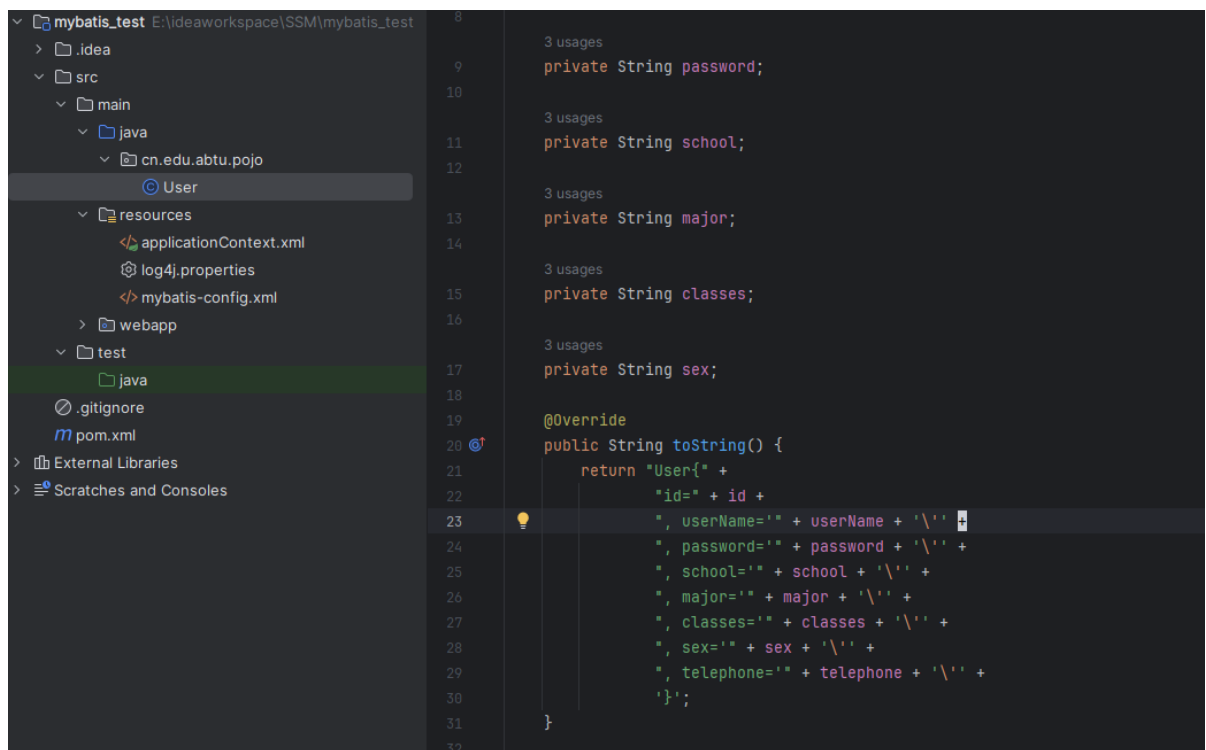


## 15. 将log4j.properties、和mybatis-config.xml拷贝到resources目录下

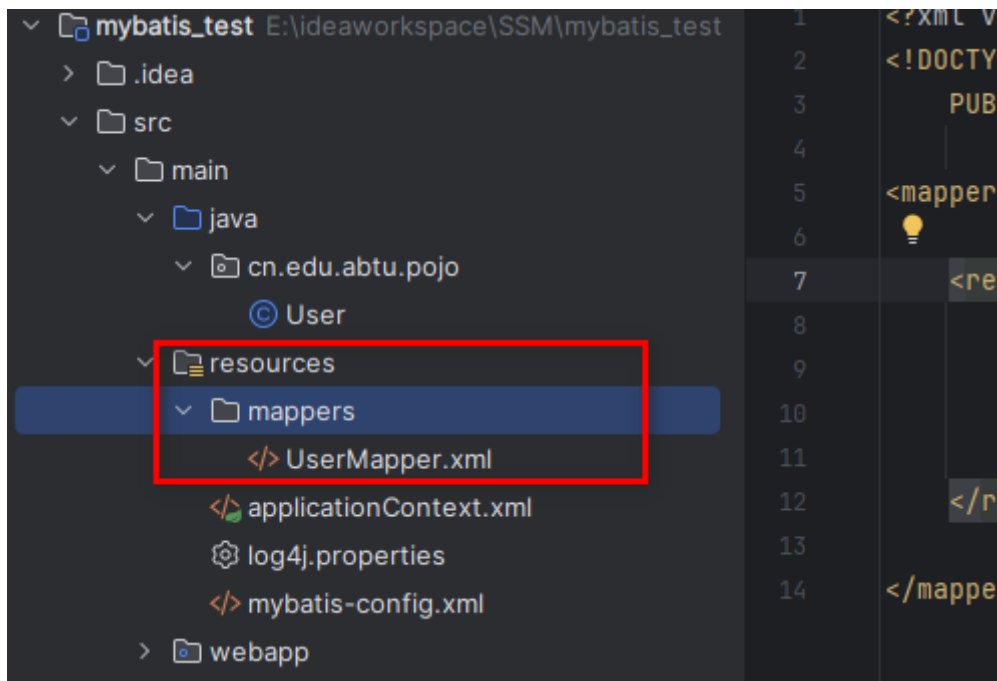
这两个文件不用配置



16. 在src->main->java下新建cn.edu.abtu.pojo包，并在包下新建User类，在类中声明id、userName、password、school、sex、major、classes、telephone变量，并生成setter/getter方法，并重写toString()方法。

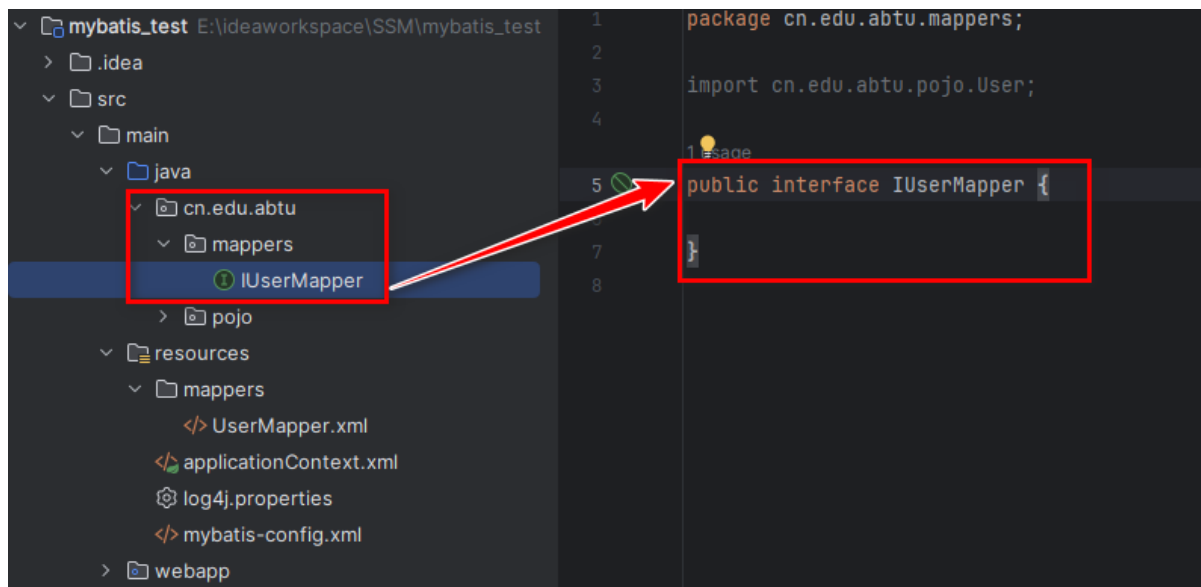


17. 在resources目录下新建mappers文件夹，并将UserMapper.xml映射文件拷贝到此目录下



18. 在src->main->java下新建cn.edu.abtu.mappers包，并在该包下新建IUserMapper接口，并在接口中声明addUser(User user)方法

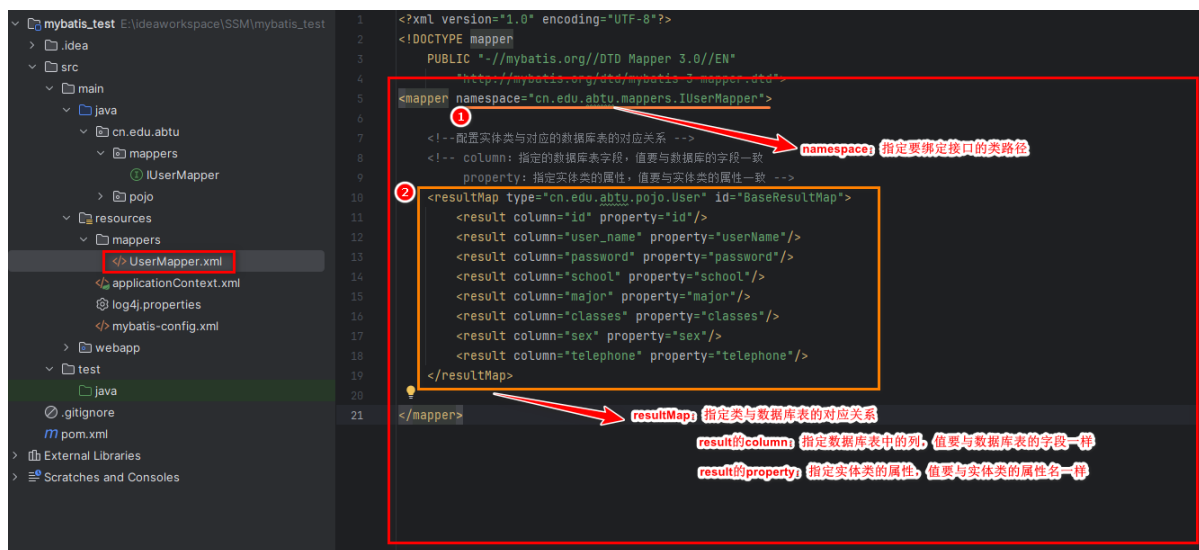
此层相当于以前的dao，区别在于：在此层只有接口，不再有实现类。



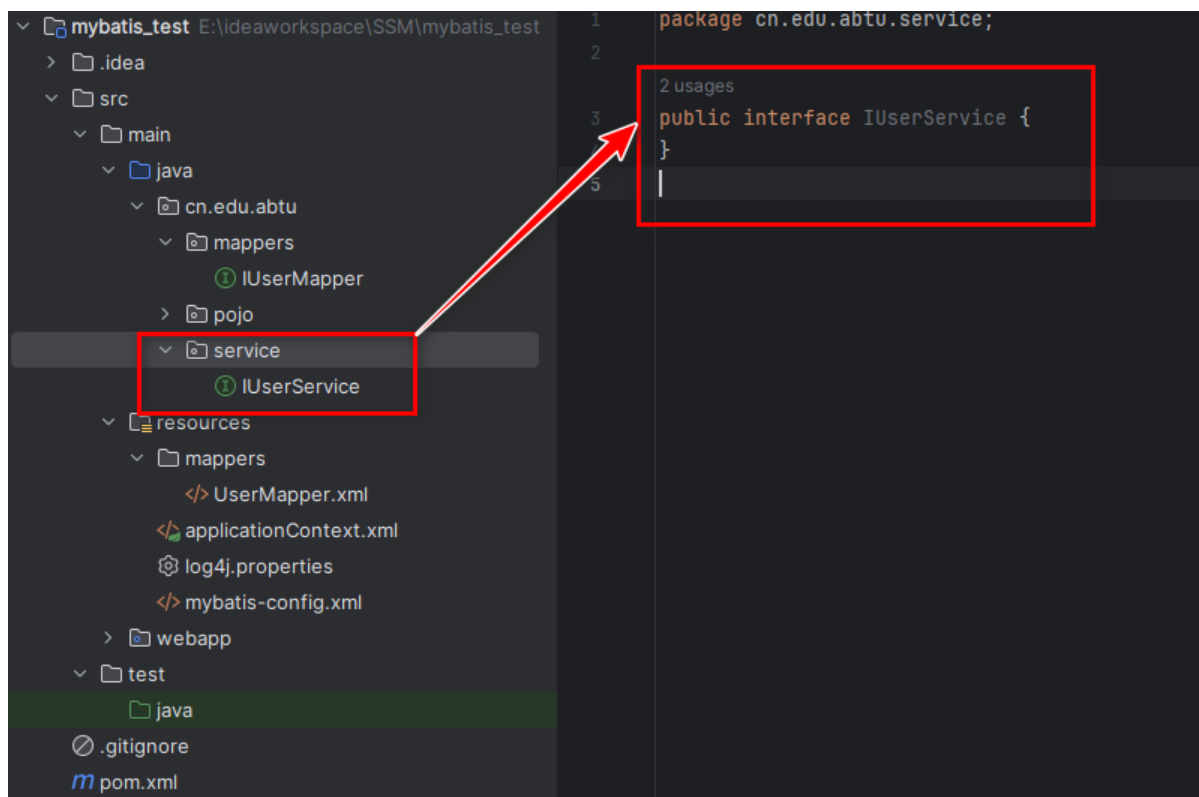
19. 配置UserMapper.xml文件的映射关系，在此文件中主要配置实体类与数据库表的映射关系及相应的数据库操作语句。

一个实体类对应一个xml映射文件



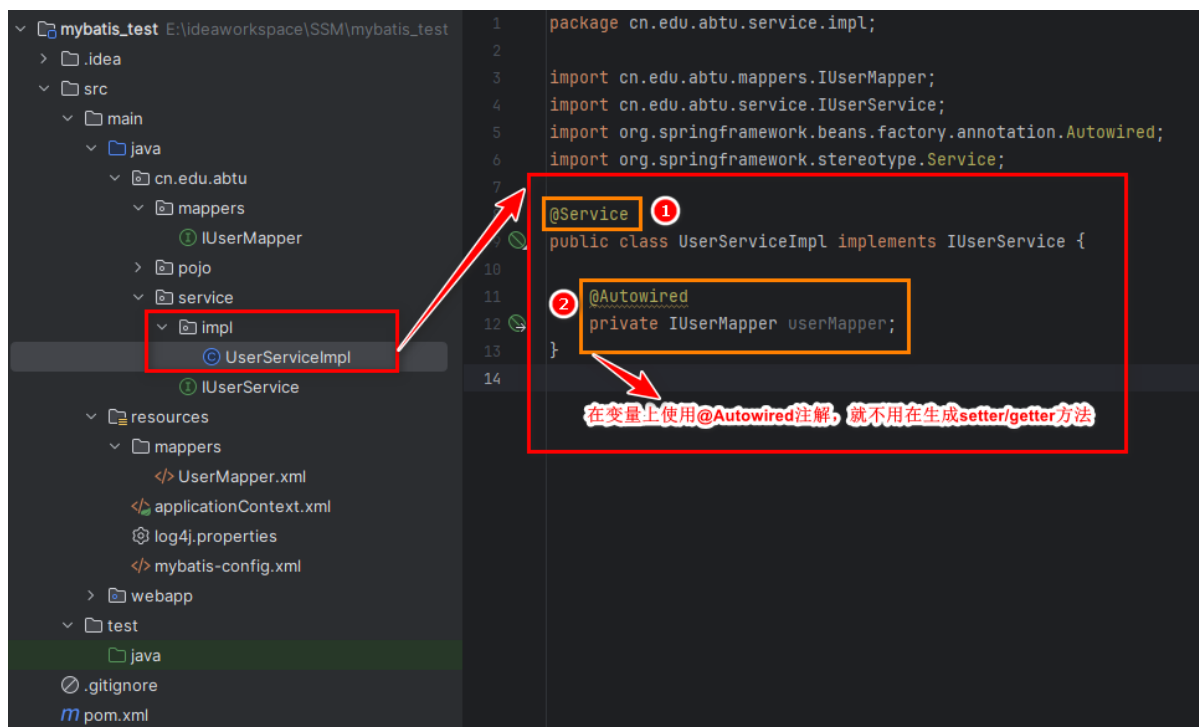


20. 在src->main->java下新建cn.edu.abtu.service包, 并新建IUserService接口。



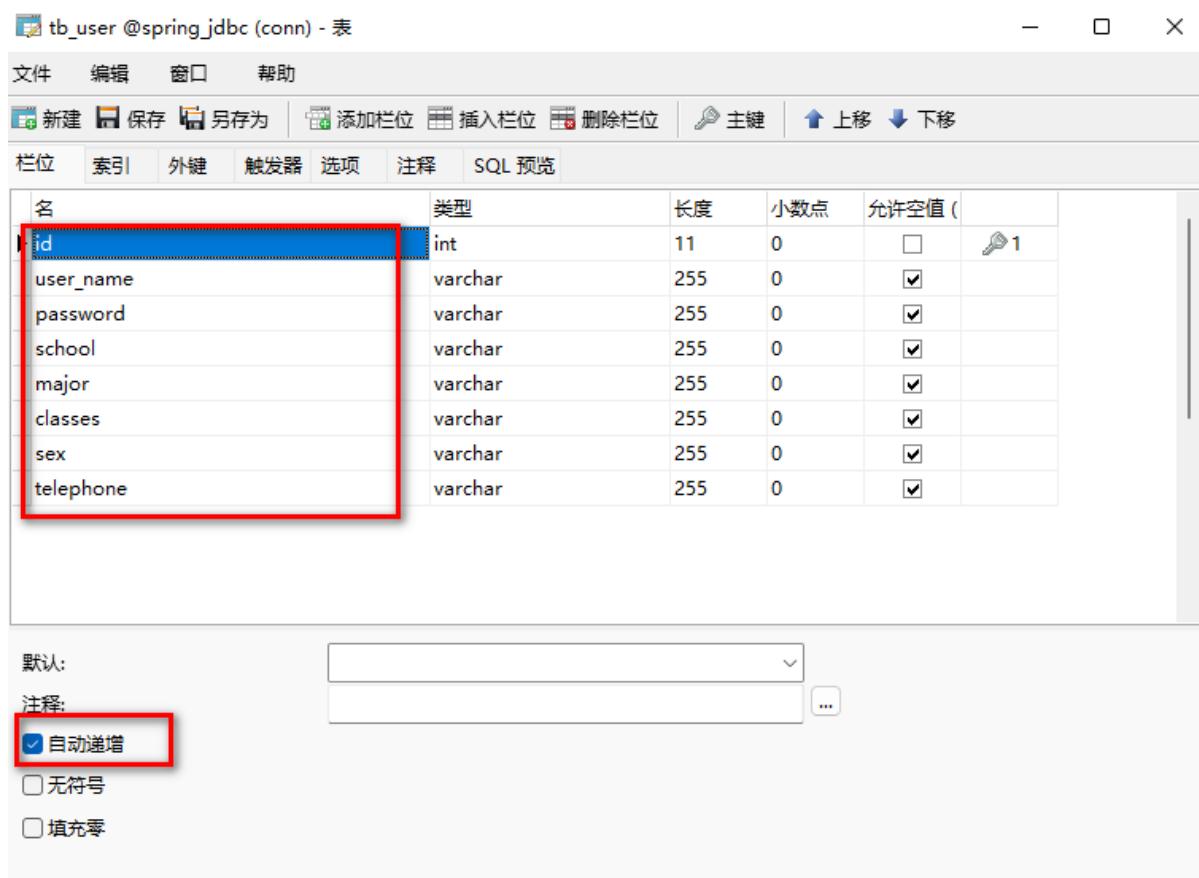
21. 在src->main->java下新建cn.edu.abtu.service.impl包, 新建 UserServiceImpl类, 并实现IUserService接口。并在UserServiceImpl类上使用@Service注解。在该方法完成Mapper层的依赖声明。

使用@Service注解后, 就不用在applicationContext.xml中配置bean了, Spring会扫描@Service类, 实例化该类, 并将实例化好的实例放入Spring核心容器。



22. 打开mysql中的spring\_jdbc数据库, 并在数据库中新建表tb\_user, 该表的字段有: id (int) , user\_name (varchar) 、 password (varchar) 、 school (varchar) 、 major (varchar) 、 classes (varchar) 、 sex (varchar) 、 telephone (varchar) 。

id为主键, 并自增。

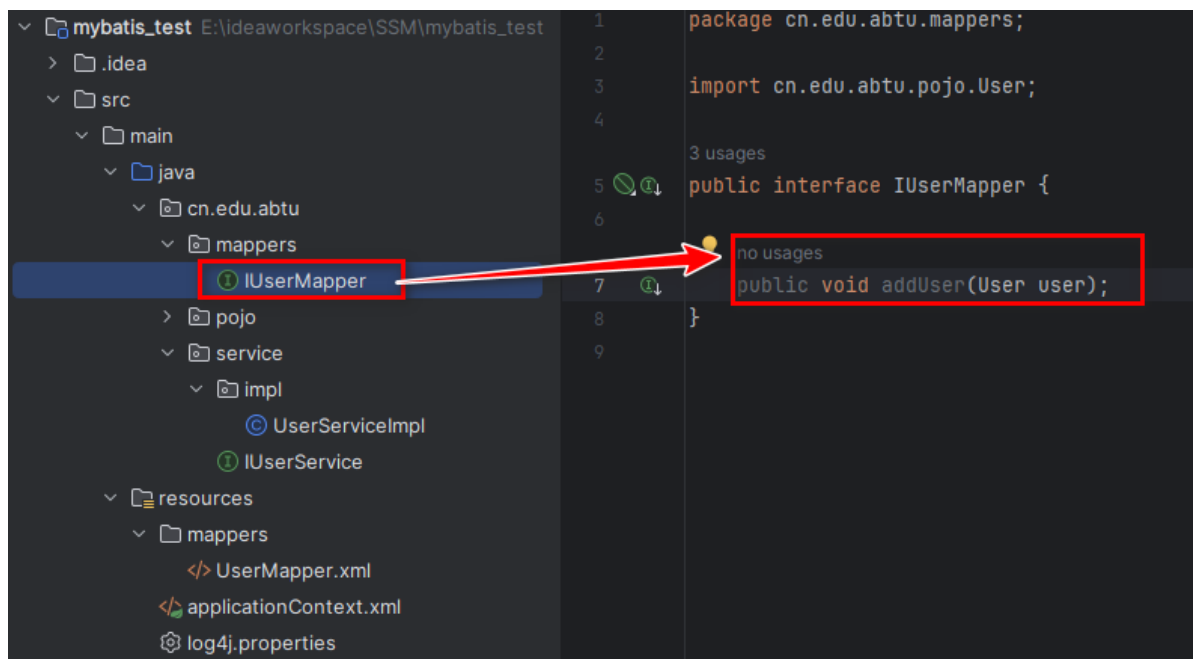


至此, 项目架构搭建完毕。

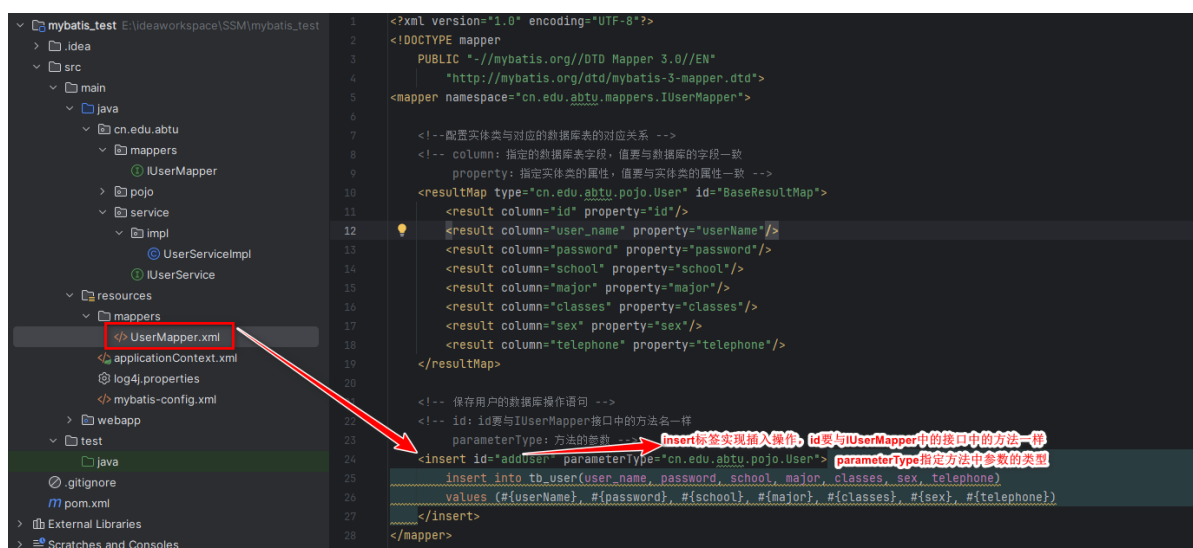
# 案例

## 1. 添加用户

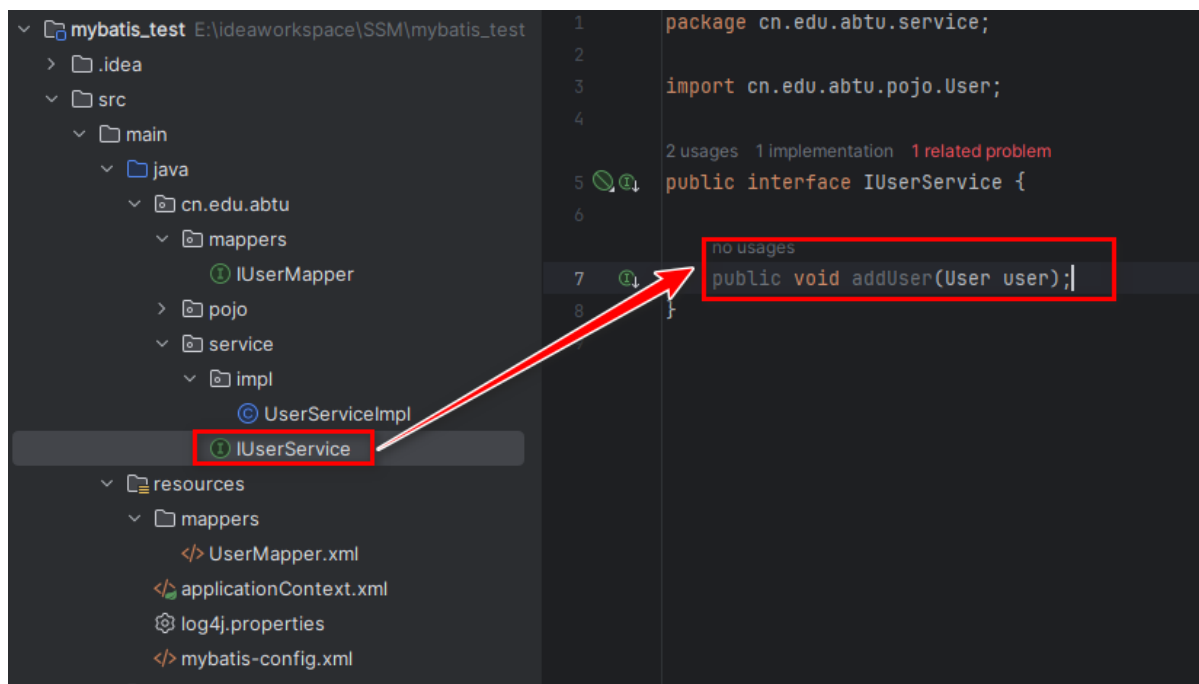
#### a. 打开IUserMapper接口，并添加方法addUser(User user)方法



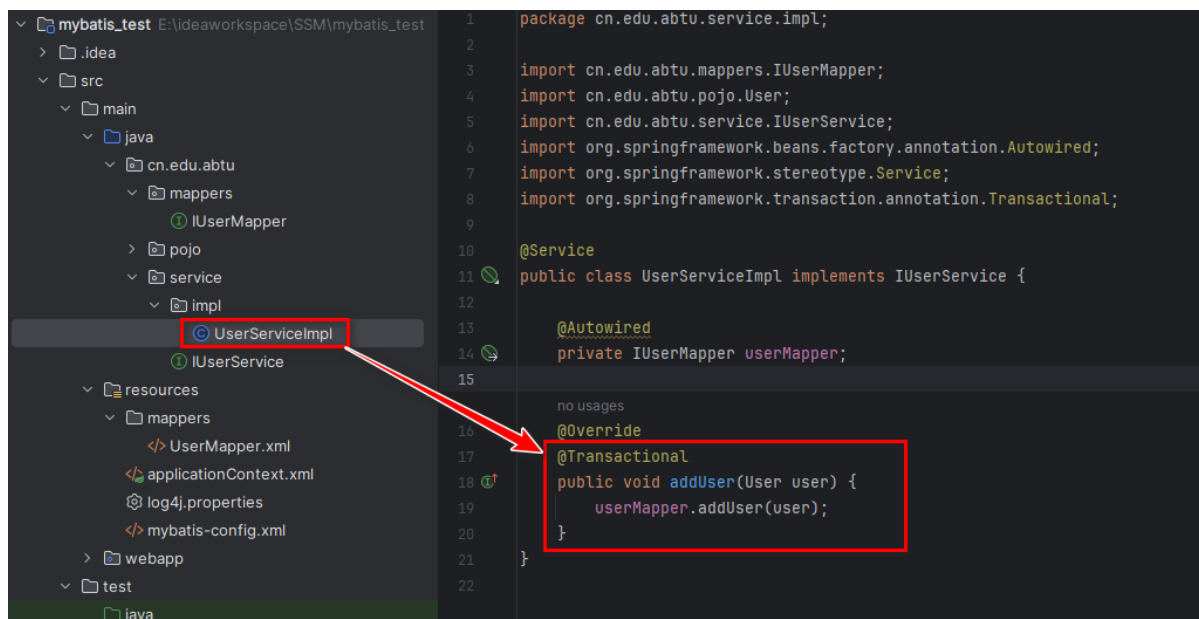
b. 打开UserMapper.xml文件，在该文件实现addUser的数据库操作



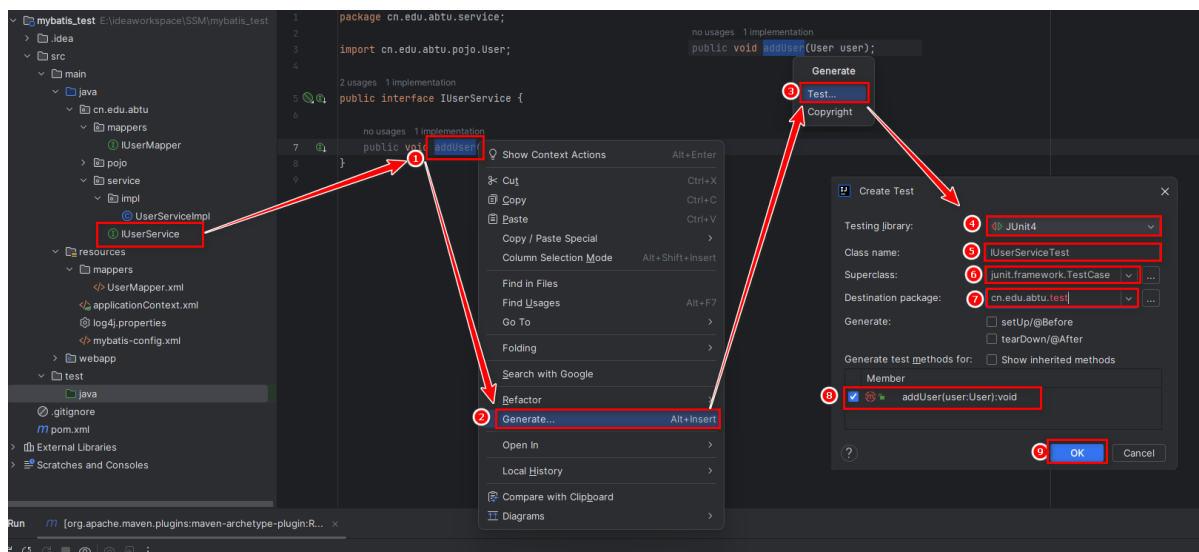
#### d. 在IUserService中添加addUser(User user)方法



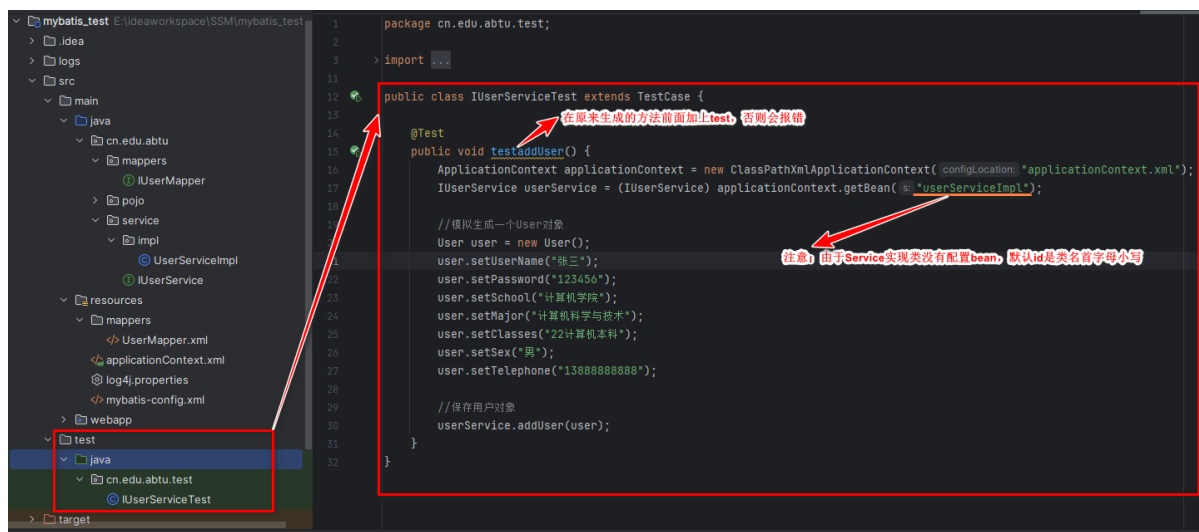
#### e. 在UserServiceImpl类中实现IUserService接口中的addUser(User user)方法，该方法实现事务，并在该方法中调用Mapper层的addUser()方法。



#### d. 依次按以下步骤生成测试类



#### e. 在测试类的addUser方法中编写测试代码



#### f. 观察实验结果

tb\_user @spring\_jdbc (conn) - 表

id	user_name	password	school	major	classes	sex	telephone
1	张三	123456	计算机学院	计算机科学	22计算机本科	男	13888888888

### (五) 实验总结

通过观察输出结果和自己实验整个过程总结一下自己的学习所得