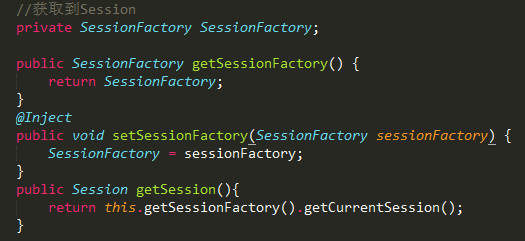
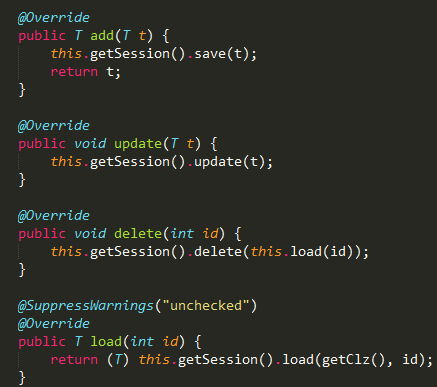
**基础dao的编写**

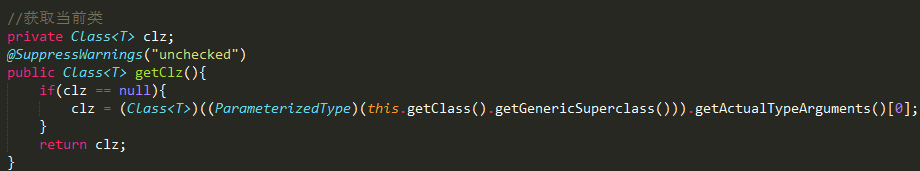
1. 首先是获取sessionFactory 和 session



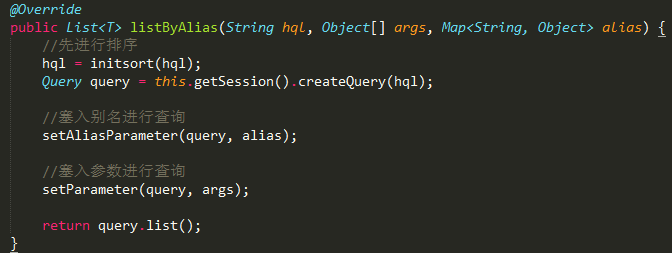
1. add、update、delete、load四个基础方法



这里牵涉到一个关于如何获取当前class的类：

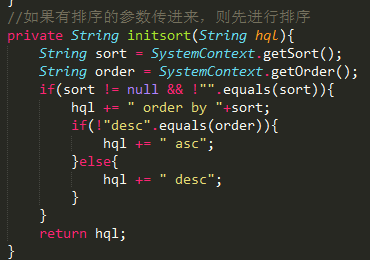


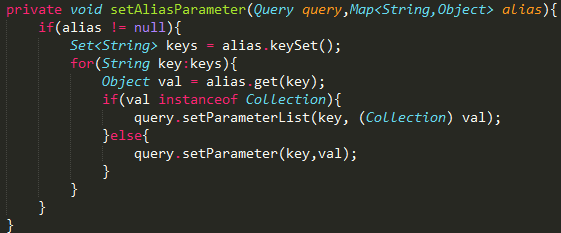
1. 下面一个方法是完成根据别名和普通参数的混合查询获取List不分页的结果集

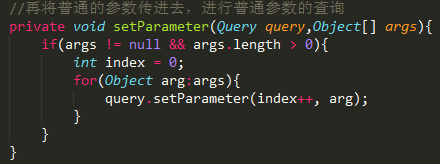


解析：这里的思路是首先根据全局变量获取排序的参数进行排序，然后先塞入别名，再塞入普通参数获取查询的结果，最后返回List

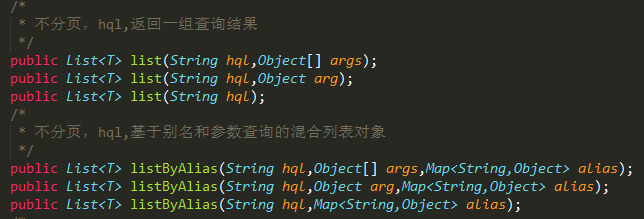
这里的排序，别名查询以及普通参数查询是一个共有方法，如下：



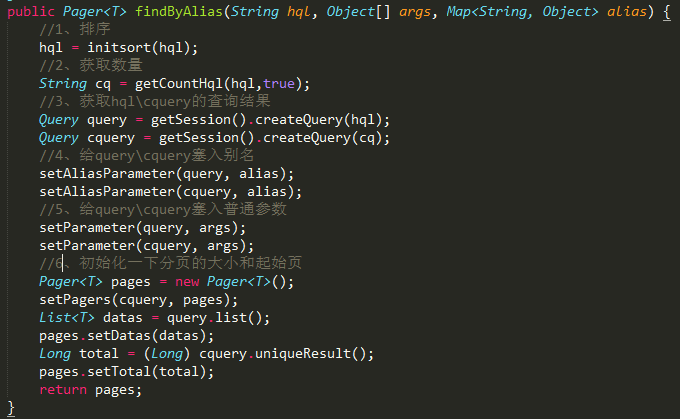




一旦这个方法完成，那么一下的方法就可以直接调用这个最复杂的方法即可。

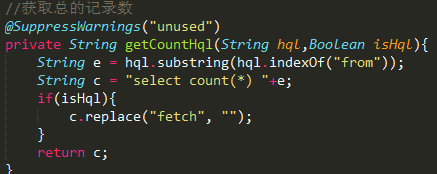


1. 下面一个方法是完成根据别名和普通参数的混合查询获取find分页的结果集

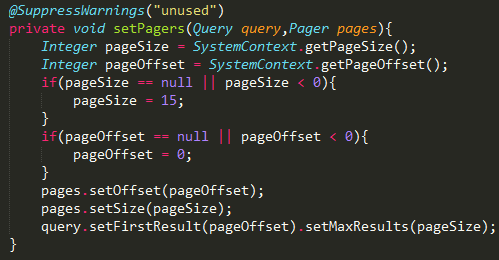


这里涉及到分页的一些处理：

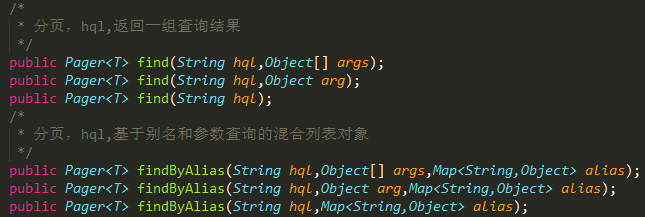
获取总的记录数:getCountHql



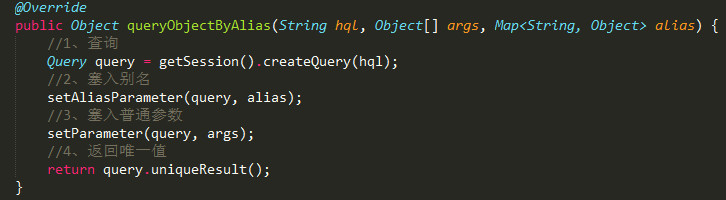
初始化分页的初始参数setPagers



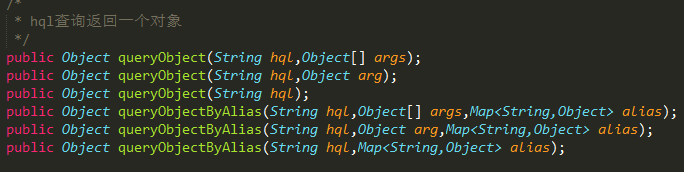
那么一下的方法也很简单地解决了：



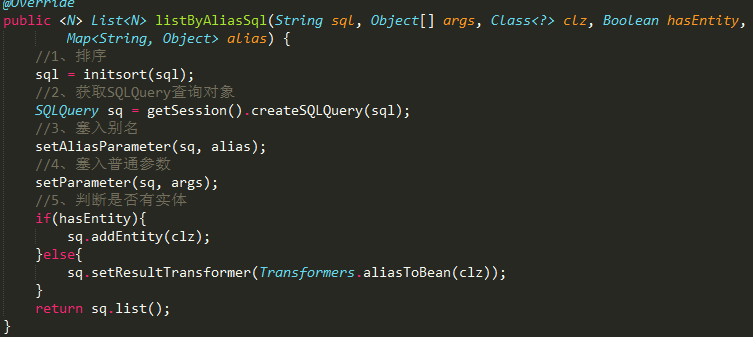
1. hql查询返回一个对象



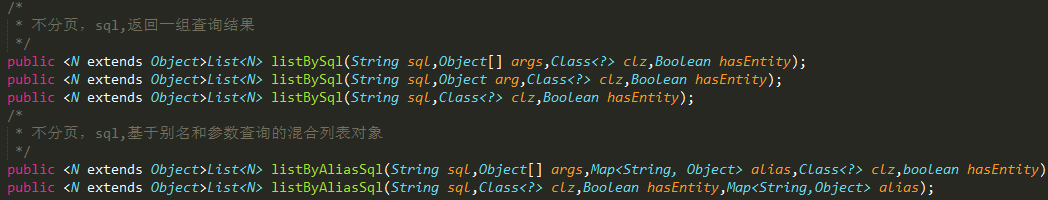
那么一下方法也随之解决：



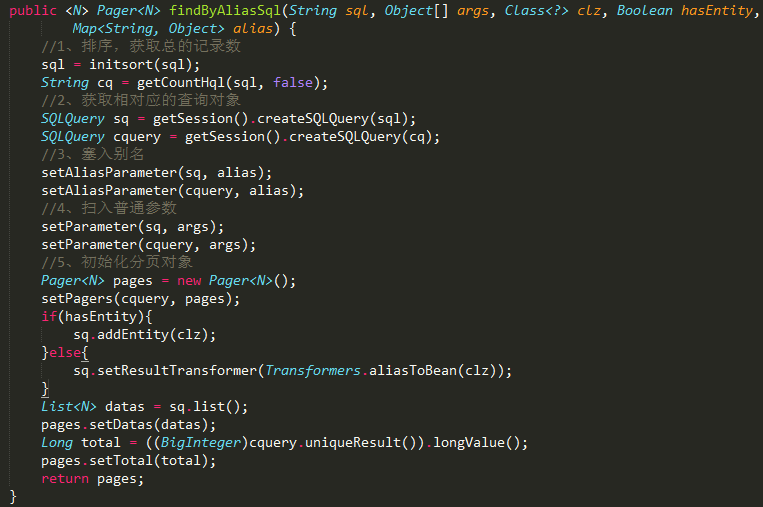
1. 不分页，sql,基于别名和参数查询的混合列表对象



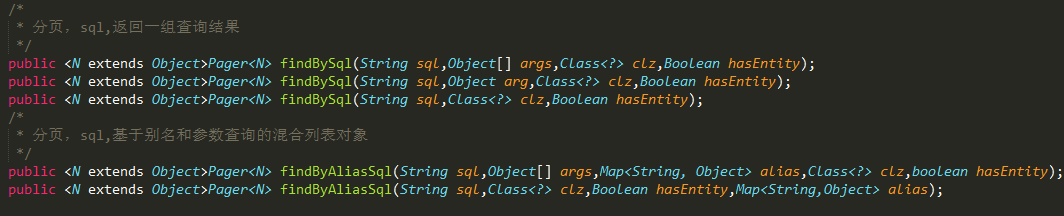
一下方法也解决



1. 分页，sql,基于别名和参数查询的混合列表对象



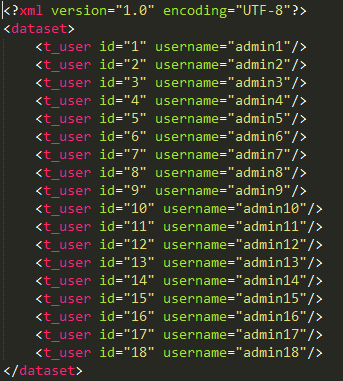
那么以下方法也解决：



到此为止，已经实现了所有所可能要用到的通用的baseDao的方法。

1. 测试

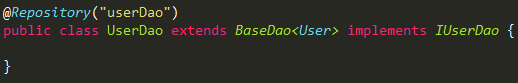
8.1、将beans.xml和t\_user.xml放到src/text/resources文件夹下



8.2、在src/test/java中，首先写一个User的Model

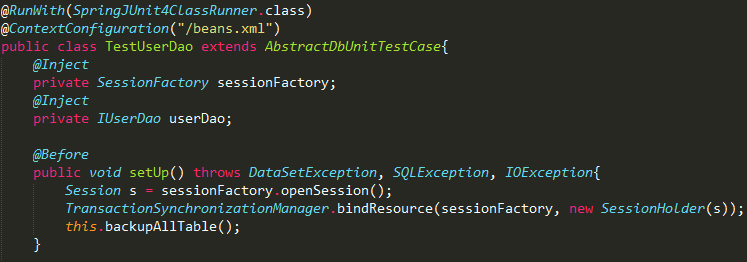


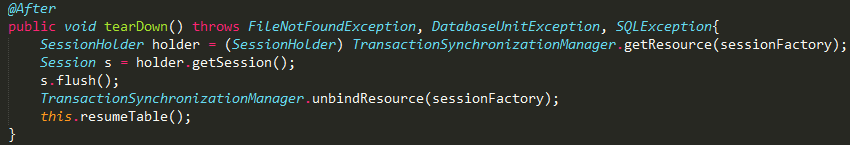
8.3、在src/test/java中，写入IUserDao和UserDao以及TestUserDao一个接口和两个类



8.4、创建数据库，名字叫cms\_test

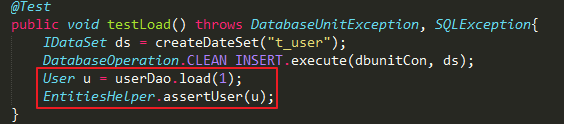
8.5、TestUserDao引入dao和初始化



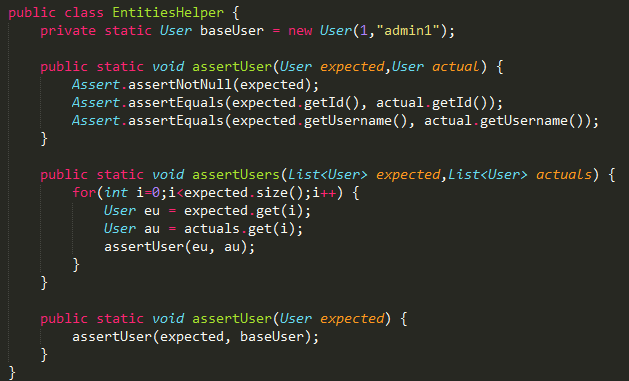


8.6、进行测试

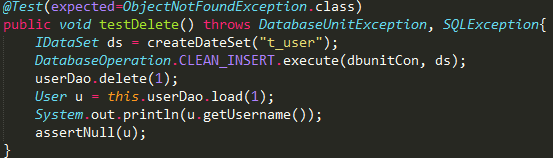
Load方法

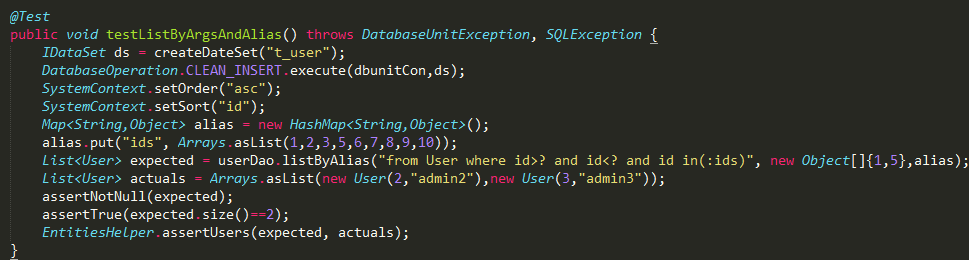


其中EntitiesHelper.assertUser(u);



Delete方法:先删除，再加载，预期为空，出现异常



testListByArgsAndAlias