前提条件:你的类必须继承HibernateDaoSupport   
一：   
回调函数:

public List getList(){

return (List ) **getHibernateTemplate**().execute(

new HibernateCallback() {

public List doInHibernate(final Session s)

throws HibernateException, SQLException {

StringBuffer sb = new StringBuffer();

Query query = s.createSQLQuery(sb.toString());//[b]执行普通sql[/b]

Query query = s.createQuery(sb.toString());// [b]执行hql [/b]

List list = query.list();

return list ;

}

});

}  
采用这种做法的好处是:不用关心事务。session的创建和销毁，一切都在程序内部完成。看看名字doInHibernate就知道好处了。不好的是程序看起来比较乱！个人认为。   
二：

**getHibernateTemplate**().find（）:

List list = **getHibernateTemplate**().find（String sql ）:  
这种的前提是sql必须是hql ,原始sql 不能执行。   
三：   
自己获取session处理。   
获取session的两种方式:this.getSession();和this.**getHibernateTemplate**().getSessionFactory().openSession();   
获取session以后再createQuery操作就一样了。但是这种做法，是需要自己去手动关闭session的。所以你需要配置openSessioninview，个人认为不推荐使用！   
其他：   
**getHibernateTemplate**.delete(Object);   
**getHibernateTemplate**.save(Object);   
**getHibernateTemplate**.update(Object);   
总结:推荐使用回调函数。

**getHibernateTemplate()和getSession()的区别（转载）**(2009-05-06 15:10:39)

|  |  |
| --- | --- |
| 标签：[杂谈](http://uni.sina.com.cn/c.php?t=blog&k=%D4%D3%CC%B8&ts=bpost&stype=tag) | 分类：[J2EE](http://blog.sina.com.cn/s/articlelist_1309174121_9_1.html) |

在前一篇日志中自动生成hibernate配置文件的时候，会在dao层用到getSession()方法来操作数据库记录，但是他还有个方法**getHibernateTemplate**()，这两个方法究竟有什么区别呢？

1.使用getSession()方法你只要继承sessionFactory，而使用**getHibernateTemplate**()方法必须继承 HibernateDaoSupport当然包括sessionFactory，这点区别都不是特别重要的，下面这些区别就很重要了

2.getSession()方法是没有经过spring包装的，spring会把最原始的session给你，在使用完之后必须自己调用相应的 close方法，而且也不会对声明式事务进行相应的管理，一旦没有及时关闭连接，就会导致数据库连接池的连接数溢出，**getHibernateTemplate**()方法是经过spring封装的，例如添加相应的声明式事务管理，由spring管理相应的连接。

在实际的使用过程中发现的确**getHibernateTemplate**()比getSession()方法要好很多，但是有些方法在**getHibernateTemplate**()并没有提供，这时我们用HibernateCallback 回调的方法管理数据库.

例如如下代码：  
public List getListForPage ( final String hql , final int offset , final int length ) {

              List list = **getHibernateTemplate**().executeFind ( new HibernateCallback ( ) {  
                            public Object doInHibernate ( Session session ) throws HibernateException, SQLException {  
                                            Query query = session.createQuery ( hql ) ;  
                                            query.setFirstResult ( offset ) ;  
                                            query.setMaxResults ( length ) ;  
                                            List list = query.list ( ) ;  
                                            return list ;  
                           }  
               }) ;  
               return list ;  
}