# 项目描述

# 框架特点

框架使用主流的Spring MVC+Hibernate4整合，基于注解的配置，基本框架都已配置完毕，不需要更改任何XML文件；

框架抽象了BaseDao/BaseService/BaseAction的功能，程序员开发一套增删改查，不需要编写大量代码，建立几个空的接口和实现类，其他全由Base包下的功能实现；

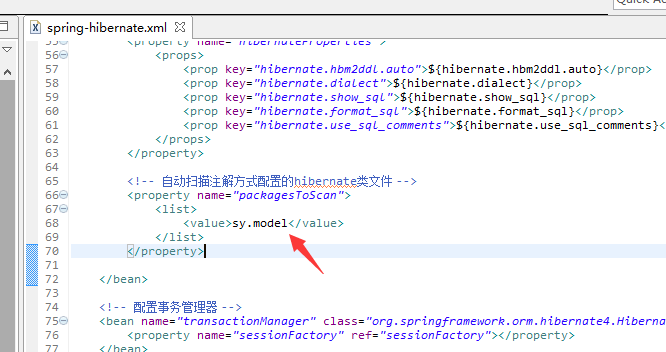
可以实现动态条件查询、分页、排序、和多表级联查询；

上手容易，程序员在此基础上二次开发，不需要配置任何配置文件，只要遵循约定，书写相应注解就可以实现项目90%的功能，针对更复杂的操作，也提供了相应的接口，具体请看BaseDao的实现。

# Model编写

由于框架使用hibernate4，所以表和对象的映射需要建立一个Class来体现

Model的建立规则是：必须建立在sy.model包或者子包下面，因为我在spring-hibernate.xml中配置了

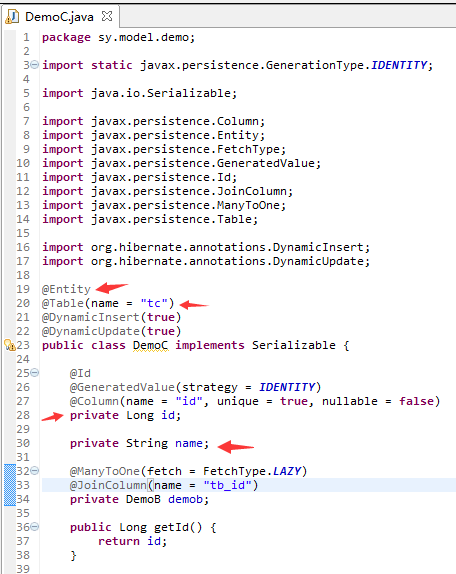


这段扫描，如果你想建立在其他包下面，请修改这个地方的配置，增加自己要扫描的包，包的扫描是包括子包的；

模型的书写规范，请参考JPA，这里举个例子：

假设有一个表，名字叫做tc，里面有个id是自增长、有个name，是字符串型

那么类的书写就是：



@Entity代表实体

@Table代表映射表，表名是tc

@Id是主键

name没有写任何注解，但是也会自动映射，如果不想name被映射到表中，请添加

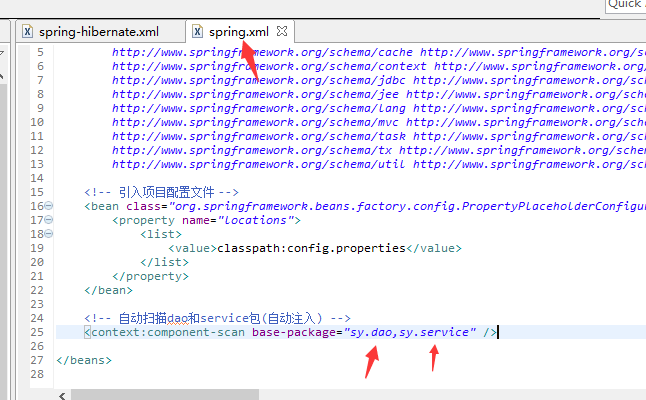
@Transient注解

至于其他注解，请参考demo包的示例类，还有疑问，请谷歌百度。

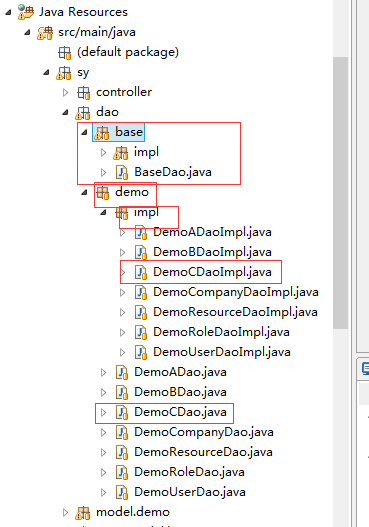
# Dao编写

项目中我抽象了BaseDao和BsaeDaoImpl实现类，实现了90%的hql操作，并提供了sql的回调方式；

Dao层的类必须建立在sy.dao包及其子包下面，因为在spring.xml里面配置了，如果想在其他包下面建立dao，请自行在这个配置修改



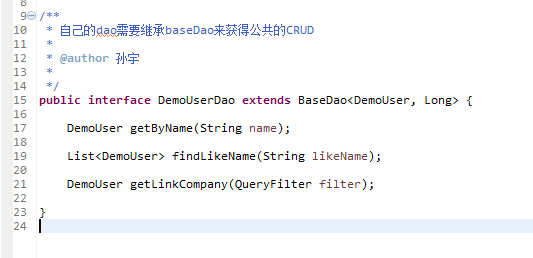
举例来说，如果我要针对上面的Tc表做操作，那么要在sy.dao包及其子包建立dao接口和impl实现类；



如上图，sy.dao.base这个包是绝对不要改动的，你可以在这里面自己建立一个包，名字随便起，只要在sy.dao包下就可以，想上面例子，建立了demo包，包下有DemoCDao和DemoCDaoImpl实现。

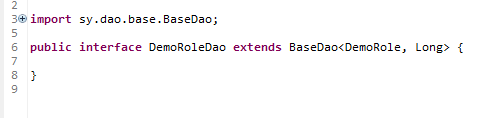
Dao的书写规范是：

首先要建立一个接口

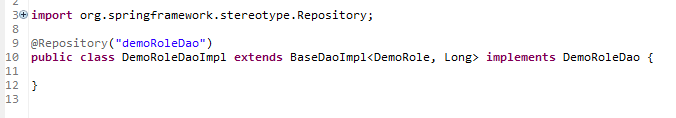


接口要继承自BaseDao，由于使用了泛型，所以要指定这个dao的model和主键类型

自己的dao里面可以没有任何接口方法，也可以自定义其他接口方法



例如上面这图，这样就行。

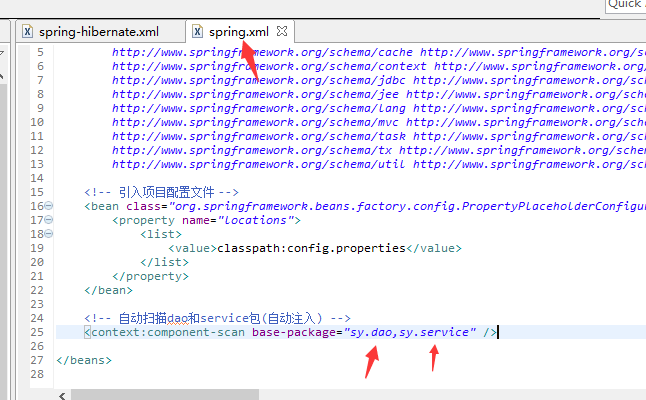


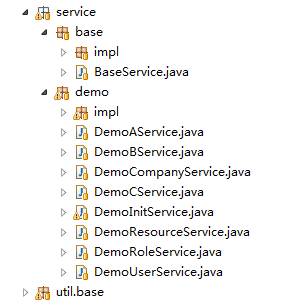
然后建立一个实现类，实现你自己的接口，但是要继承BaseDaoImpl来获得基础的增删改查、过滤、预先抓取、分页、排序等功能；

当然，要注意，你自己的dao实现类要加上@Repository注解，并且指定dao的名字

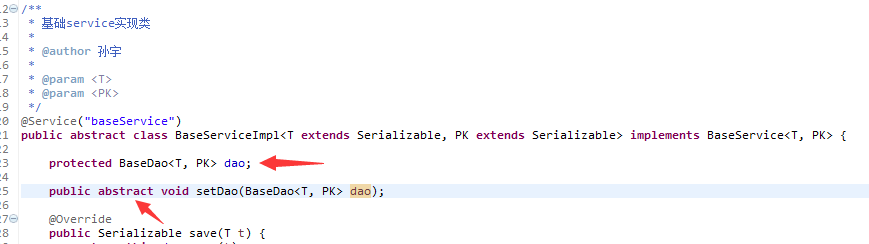
# Service编写

Service层也抽象出来了一个BseService，类的路径规则也是，必须放到sy.service包及其子包下面



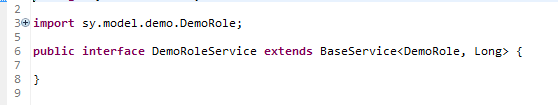


base包千万不要动，这里是我抽象好的方法，基本上够用了，不要将dao中的所有方法暴漏给service层。不要在service层面书写hql或者sql语句。

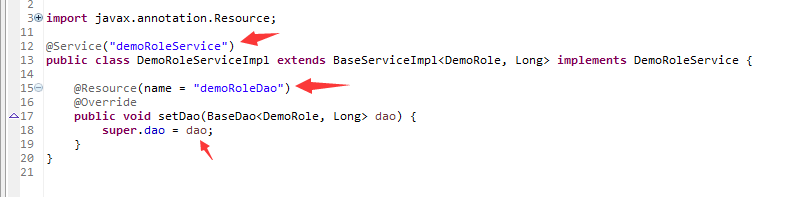


BaseServiceImpl实现类我添加了一个抽象方法，所以你的service实现类继承BsaeServiceImpl的时候，要显示的提供dao的注入，让基类的service知道调用哪个dao来操作哪个表。

书写规范：



你自己建立一个接口，当然一定要继承BaseService，并且指定model和主键类型



然后建立一个实现类，这里，setDao是重点，一定要把你自己建立的dao注入到supro的dao中，否则基类不知道你要操作哪张表。

你自己的service实现类要添加@Service注解，并且指定一个名称

@Resource(name = "demoRoleDao")//这个dao就是你自己建立的，要注入给super的dao属性中

@Override

**public** **void** setDao(BaseDao<DemoRole, Long> dao) {

**super**.dao = dao;

}

@Resource里面的名字指定就是dao的@Repository的名字

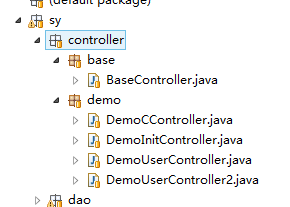
# Controller编写

Controller就相当于Struts的Action；

要求你的Controller必须要建立在sy.controller包及其子包下面，因为我在



sprin-mvc.xml中指定了controller的扫描路径，如果你想在其他地方建立controller，请修改这地方的配置，增加自己的路径就行



上图是我的例子结构。

Base包还是老规矩，不要改动，你要建立自己的controller可以自己建立一个包，我这里就建立了demo包。

Controller编写规范：



举例来说，就是，你的controller必须要有@Controller注解

如果你想加命名空间，请在类上面使用@RequestMapping，这个注解在类或者方法上都可以加

Controller要使用哪些service，请自行注入，参考实际例子就行。

@ResponseBody注解的意思就是讲对象转成JSON格式的字符串返回到前台页面

其他spring mvc的使用，请参考例子，或者自行谷歌百度。。。

# 功能开发顺序

你想要开发一个功能，首先你肯定有表，那么顺序就是

1. 先写Model，将表和对象进行映射(注意@Entity和@Table注解的使用)
2. 编写Dao接口和DaoImpl实现(注意继承BaseDao和实现BaseDaoImpl，并且要增加@Repository注解)
3. 编写Service接口和ServiceImpl实现(注意继承BaseService和实现BaseServiceImpl，并且要增加@Service的注解，里面需要注入你自己的dao)
4. 编写Controller(注意添加@Controller，便于程序自动扫描为控制器)