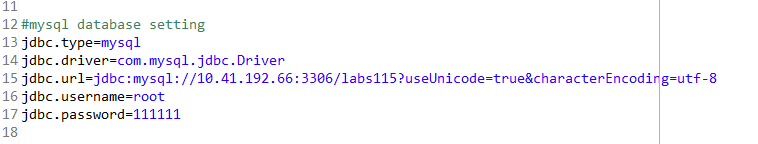
## 配置、启动框架

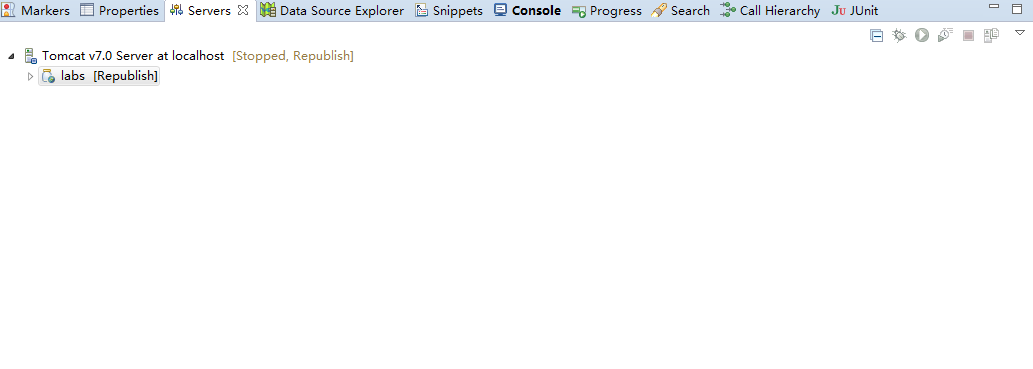
### 配置文件

labs.properties

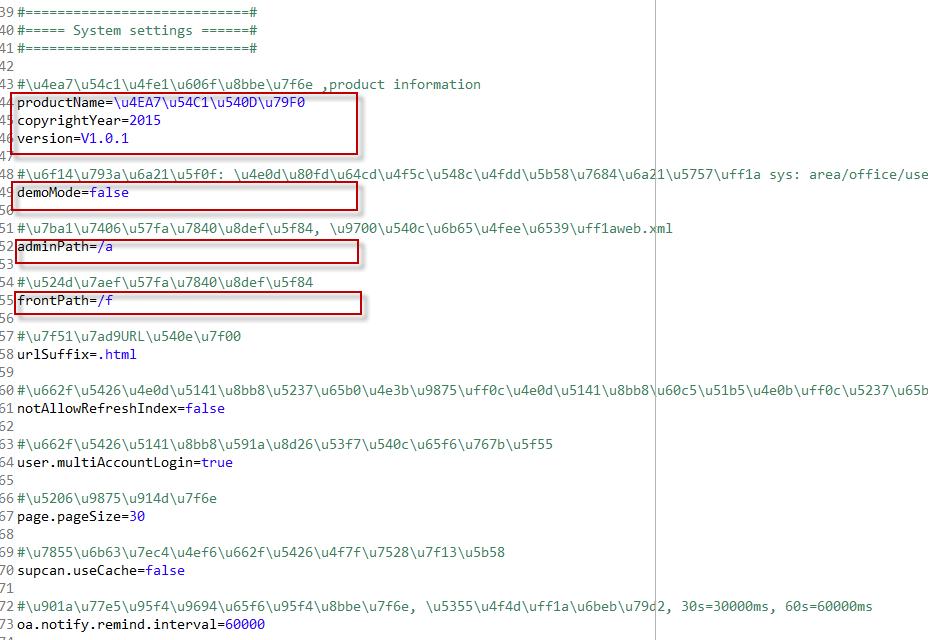
#### 数据库配置



数据库配置完成后，既可以在tomcat中运行。

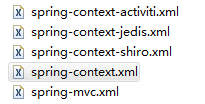


#### 系统配置



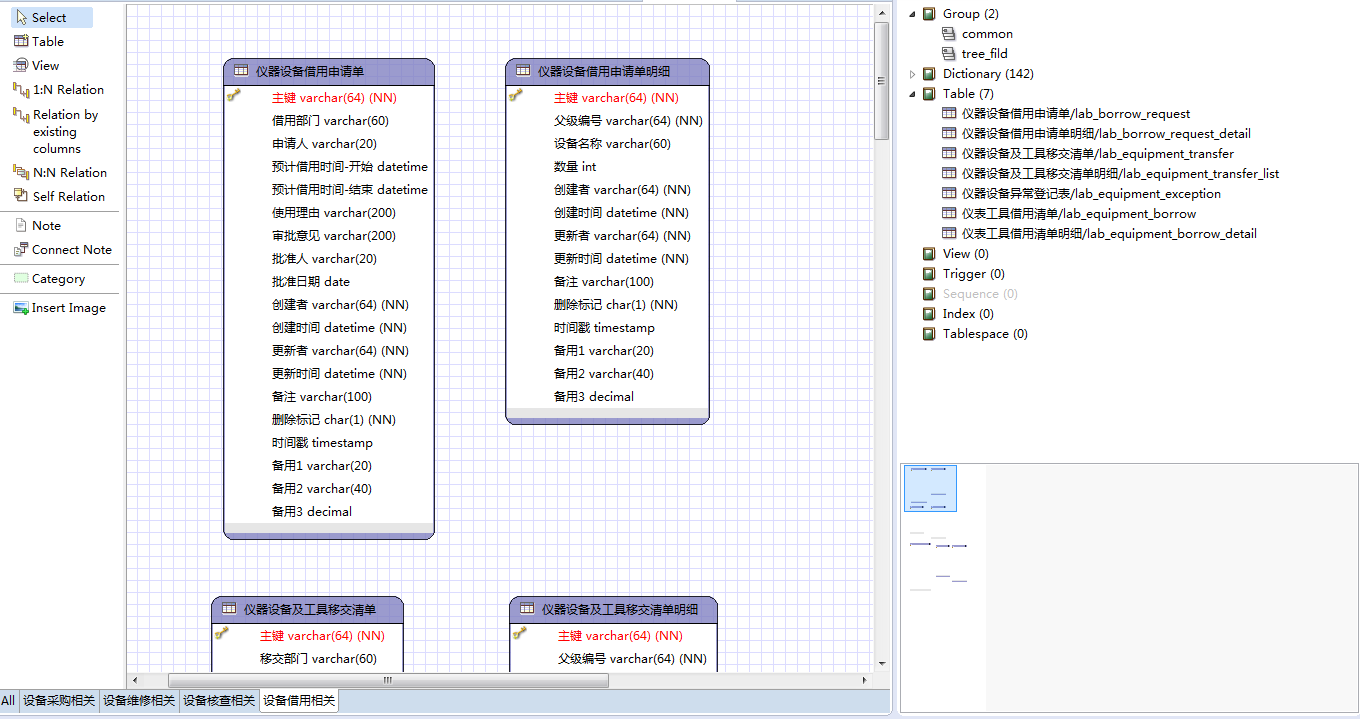
#### Spring配置

此框架用Spring来管理类（DAO/SERVICE/CONTROLER）的实例和依赖，新建功能时可能会调整的Spring配置。

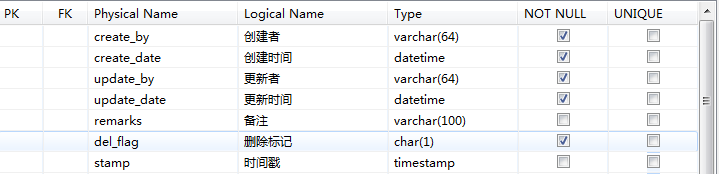


## 新建表

本框架使用Eclipse ERM插件新建表模型，也可以使用其他模型工具。



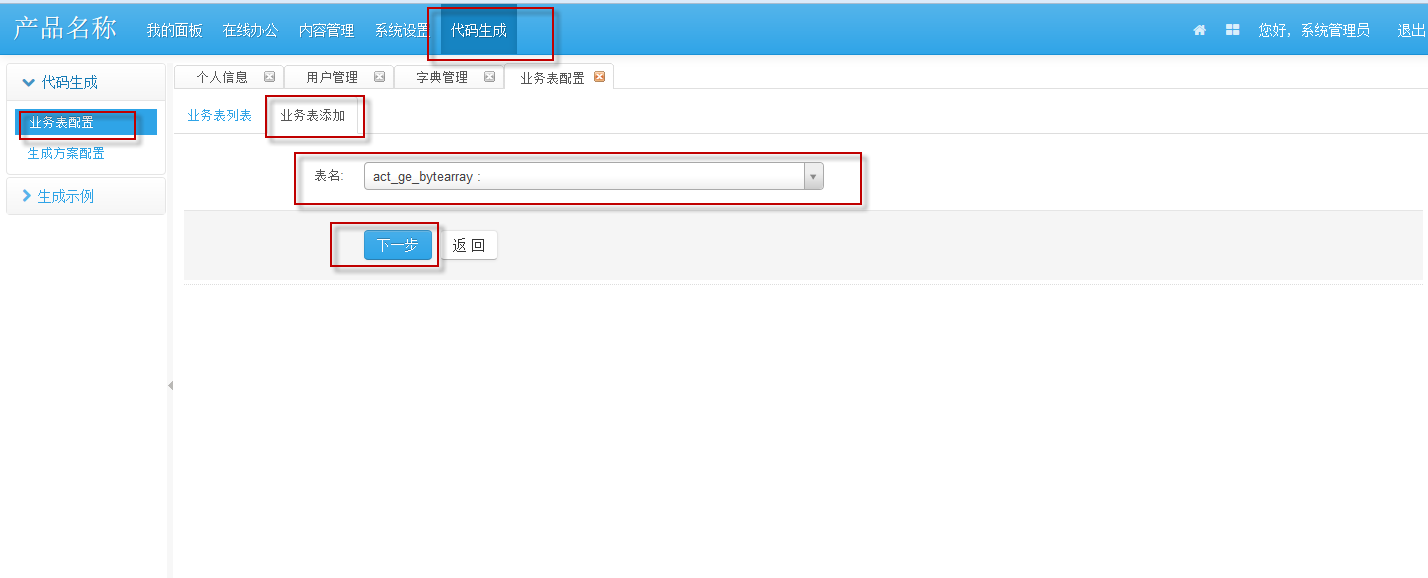
其中框架要求必须的字段

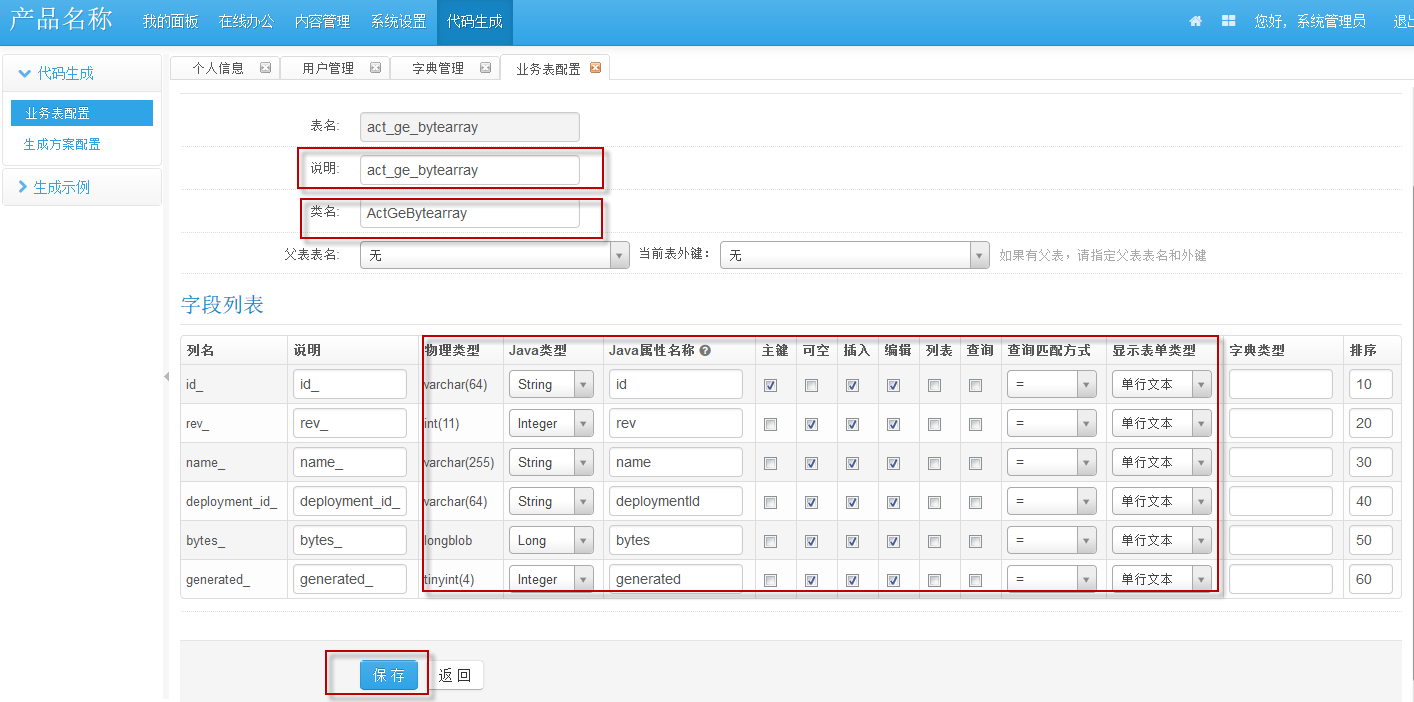


建完表模型后，根据模型生成SQL文件，在数据库中执行、建表。

## 根据表生成基本CRUD代码

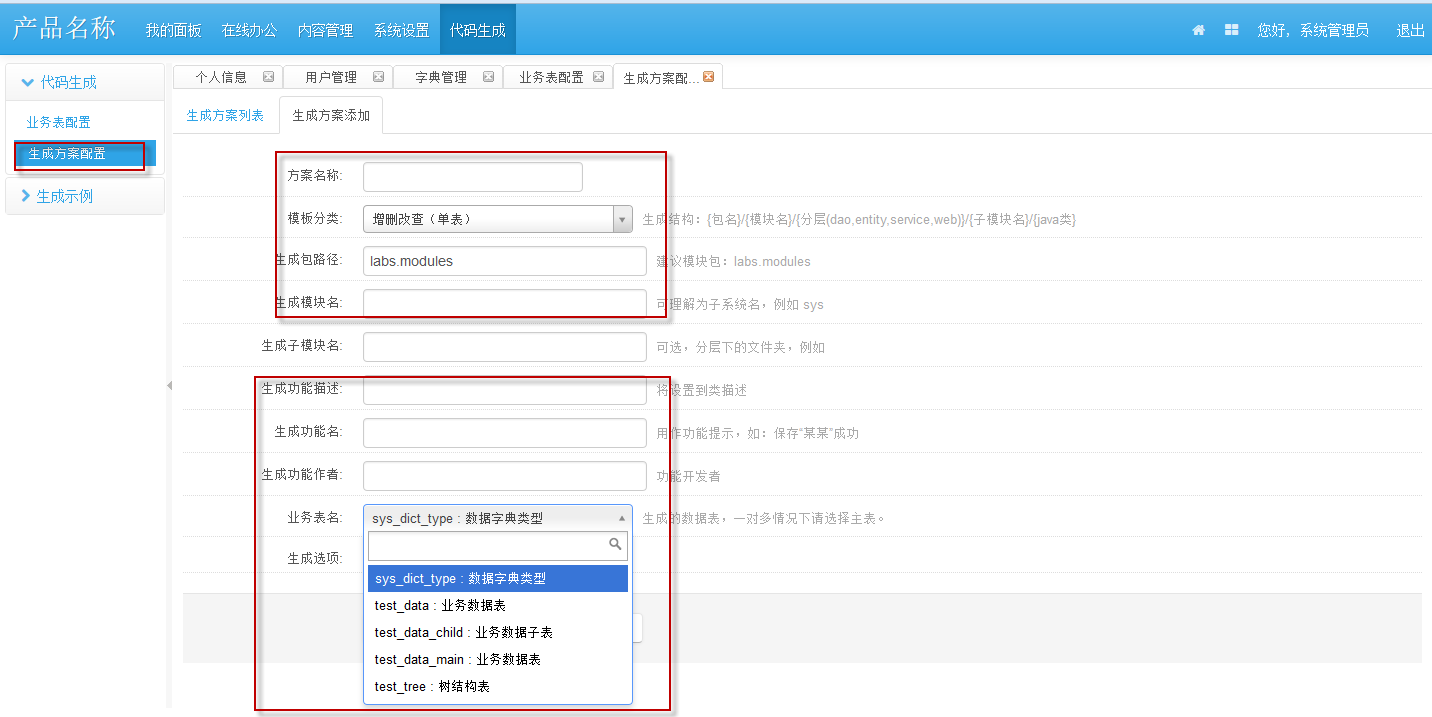
### 系统中配置需生成代码的表



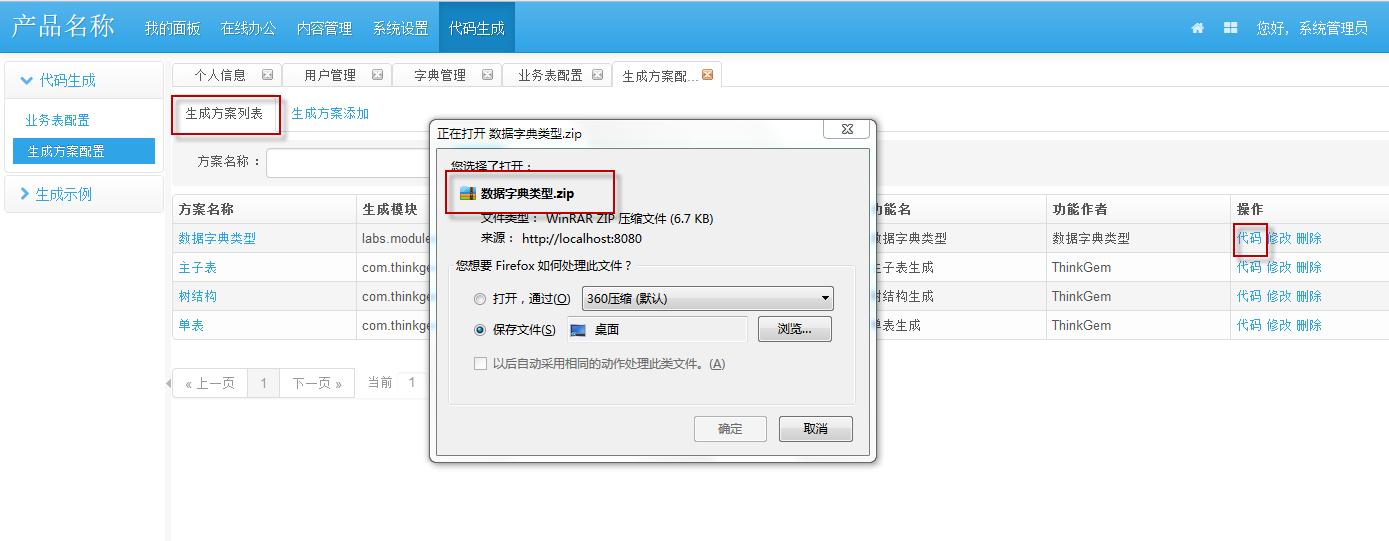


### 配置生成方案

包括模板分类、包路径、选择待生成的表等。默认java类包名labs.modules，如果改变包名，需要配置Spring 扫描路径，用来注册新的DAO/SERVICE/CONTROLLER类

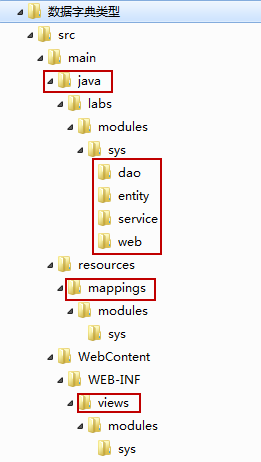


### 生成代码



## 服务端代码

### 代码结构

****

**java类**

Java**文件夹下存放的是java类，包括entity、dao、service、web controller 。**

#### Entity

自动生成的实体类继承自DataEntity，类中包含基本的校验

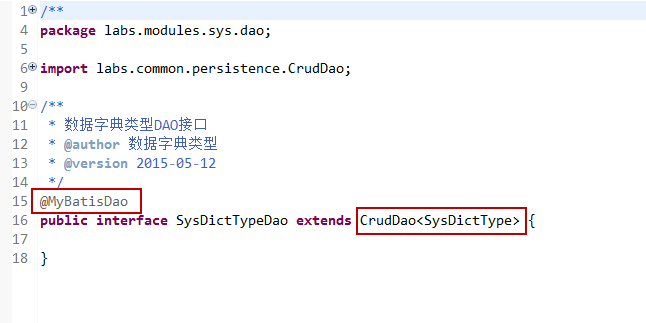
****

#### Dao

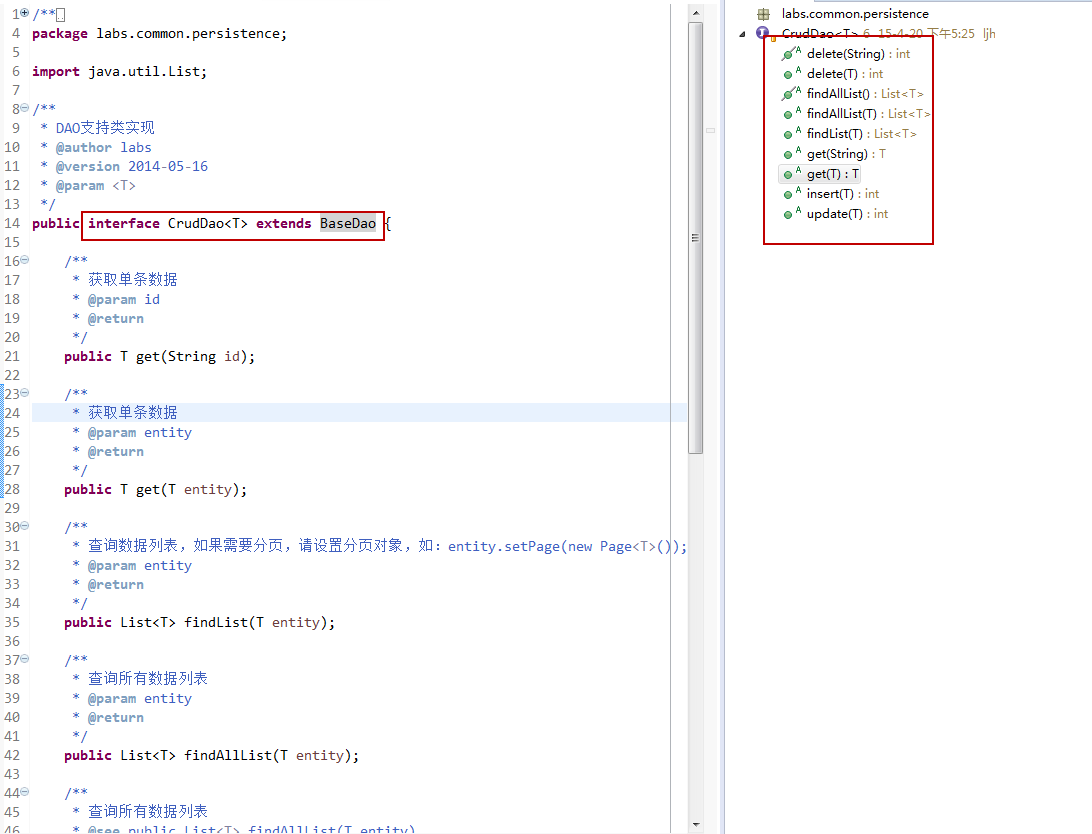
数据库操作使用了MyBatis框架，由Spring整合、配置、管理。

数据库操作类用@MyBatisDao注解，用来让Spring自动扫描DAO类，不需要手工配置DAO，并继承自CrudDao，提供了基本的CRUD操作，如果需要额外操作，在接口中定义新方法，并在MyBatis映射文件中定义id和方法名一样的SQL操作。

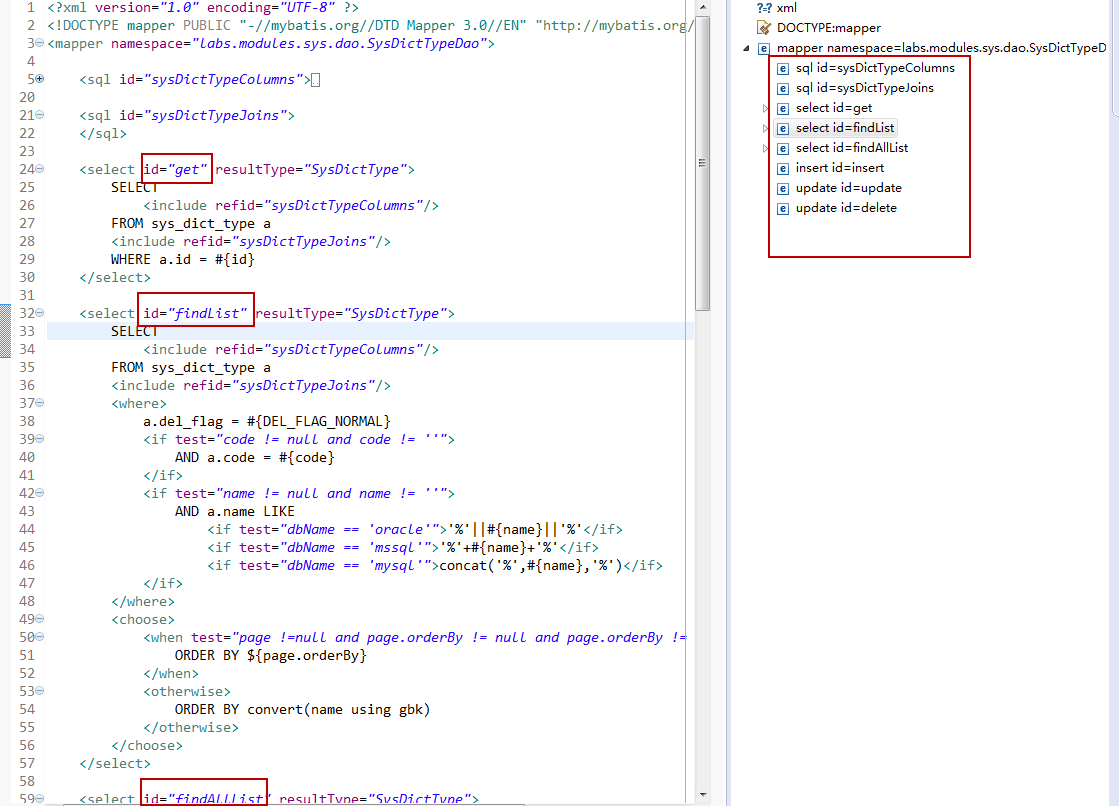
代码中可以看出，DAO操作类是一个接口，调用DAO方法时，MyBatis会根据调用的接口的方法名去匹配MyBatis映射文件中SQL 定义的ID，执行查找的SQL语句。

****

**CrudDao**

****

**MyBatis映射文件**

****

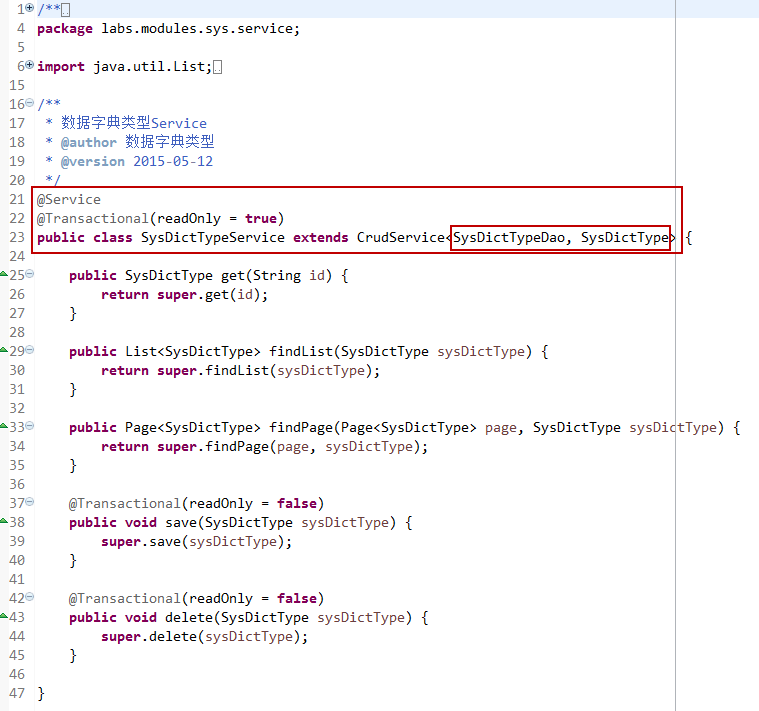
#### Service

自动生成的服务组件类，继承自CrudService，需要两个泛型参数（DAO和ENTITY），包含默认的操作方法，也可以自定义新的服务方法。

服务组件由@Service注解，让Spring自动注册服务组件。

服务组件@Transactional(readOnly = true)注解表示：默认情况下，此类方法的数据库事务是只读的。

用@Transactional(readOnly = false)注解某个方法，表示此方法的事务不是只读的（忽略类的事务注解），Spring事务管理器会在方法开始前自动开启一个事务，并在方法结束后自动提交/回滚事务。

****

#### Controller

控制层使用了SpringMVC框架。

控制层组件继承自BaseController，由@Controller注解，Spring会自动注册控制层组件。

* @RequestMapping(value = "${adminPath}/sys/sysDictType")注解定义了控制层组件的相对URL，${adminPath}是变量由BaseController定义，从配置文件中读取。
* Controller对Service的依赖由Spring自动注入，比如

@Autowired

private SysDictTypeService sysDictTypeService;

Spring自动注入Controller对SysDictTypeService 的依赖。

* @ModelAttribute注解的方法，每次其他Controller方法被调用之前都会被调用。
* @RequestParam(required=false) String id

定义了HttpServletRequest请求参数

* @RequiresPermissions("sys:sysDictType:view")

Shiro注解，表示调用此方法需要的权限标识。权限标识在菜单中定义，通过授权授予用户。

此框架使用Shiro框架管理权限（登录、登出、认证、授权、验权），Shiro配置文件spring-context-shiro.xml，比较复杂。

* @RequestMapping(value = {"list", ""})

标注了访问此方法的相对路径，比如Controller的URL为a/sys/sysDictType，访问此方法的URL为a/sys/sysDictType/ list或者a/sys/sysDictType

* model.addAttribute("page", page);

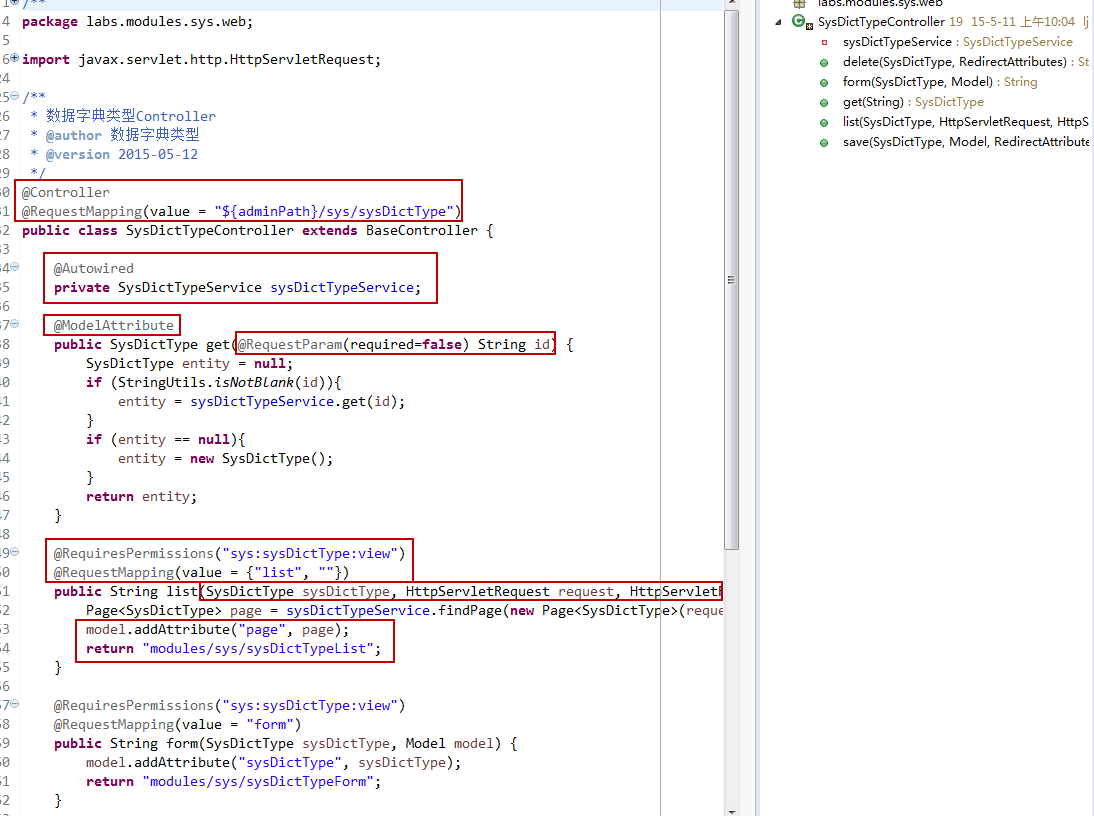
model是对应jsp页面的数据模型，jsp页面中可以通过表达式获取Controller放入model中的数据，Controller也可以从中获取到jsp提交的数据。

* return "modules/sys/sysDictTypeList";

返回路径，浏览器将显示此地址，默认的扩展名是.jsp在配置文件中配置，即浏览器将显示sysDictTypeList.jsp页面。有些方法会重定向到目的地址。

* public String form(SysDictType sysDictType, Model model)

SpringMVC会根据jsp请求中的参数构造SysDictType的实例sysDictType，将form表单中的数据转换成Java对象，这是默认的转换方式。

****

#### MyBatis映射文件

resources\mappings存放的是MyBatis映射文件，上面已经提到。

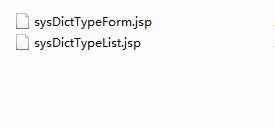
### 日志

日志由日志拦截器统一处理，dao/service层不处理异常，统一抛到controller层，由日志拦截器处理。目前拦截器保存日志有些问题，无法记录异常情况。

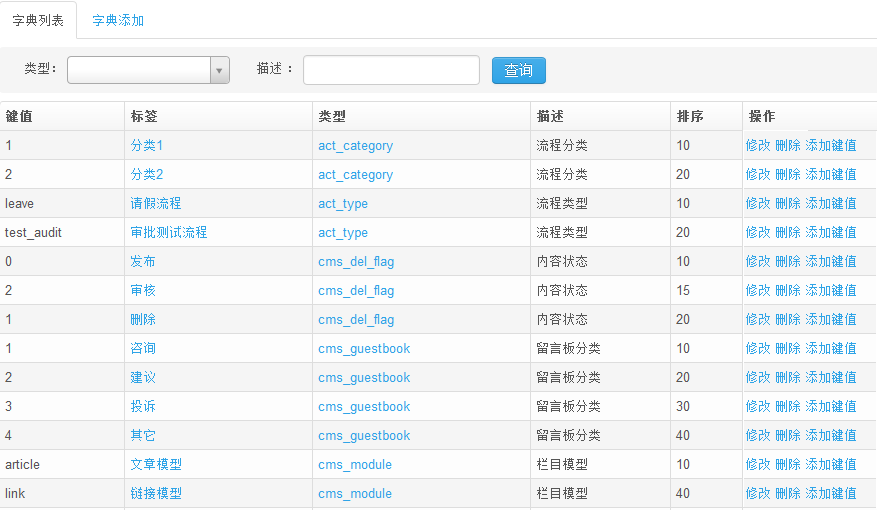
另外，如果在service层处理了异常并且没有抛出新异常，则事务将不会回滚。事务回滚依赖方法抛出异常。

## 前端

WebContent\WEB-INF\views中存放的是前台jsp文件，自动生成的jsp文件包含两个文件，xxxForm.jsp和xxxList.jsp，分别是form和list的样式。form用来新增、修改、查看一条数据，list用来显示列表数据，并提供操作链接。







### 前台使用的技术

Jsp、el表达式、JavaScript、css、HTML、HTML DOM

涉及到的框架有

SpringMVC、Bootstrap、Jquery、jQuery-validation、jquery-ztree、sitemesh、shiro

### SPRING MVC标签用法

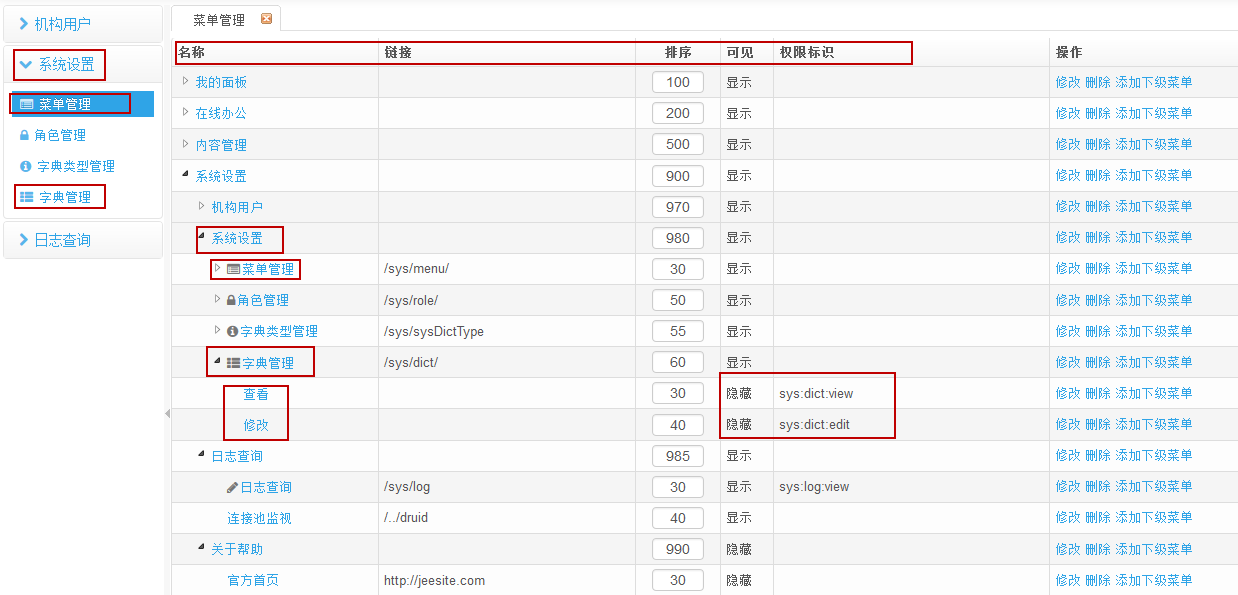


* <form:form>springmvc form标签，modelAttribute对应controller中model参数的key，action 表单提交对应的URL，method 提交方式 post/get等；
* form:hidden 隐藏域；
* sys:message自定义的显示controller返回消息的标签；
* form:input springmvc文本域，path对应java属性，或者httpservletrequest参数名称；

## 菜单和授权

### 创建菜单

菜单分为4级，顶级菜单（横向菜单）、二级菜单（左侧菜单组）、三级菜单（左侧菜单项）、四级菜单（权限标识/按钮级权限，不显示）。

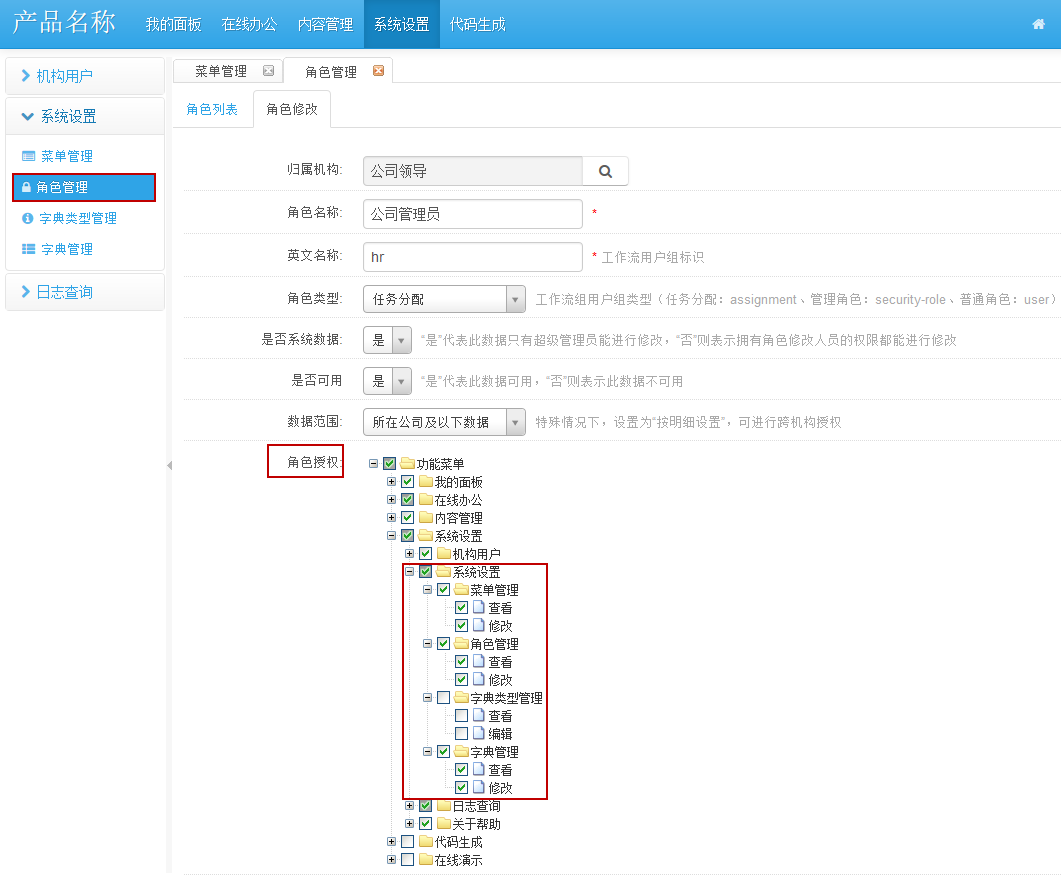




权限标识用于对controller方法的权限控制，jsp页面通过权限标识可以根据不同权限显示不同元素。



### 授权



授权后需要被授权用户重新登录，才能看见新功能点。