***开发文档***

日期： 2018/3/27

编写人：田杰

摘要：本文档主要帮助公司开发人员快速进入开发工作。熟悉框架各个组成部分和使用方法等。框架开发使用Spring boot + mybaits。权限使用Apache shiro。另外在框架基础上添加了通用的crud 接口。以及列表结果处理。弹出框设置等。下面文档将一一列出来。

# 项目启动

svn 地址：svn://192.168.11.41/springboot\_mybatis\_shiro\_ms3.0

初始化数据库：initsql->ms3.0.sql

开发环境jdk1.7

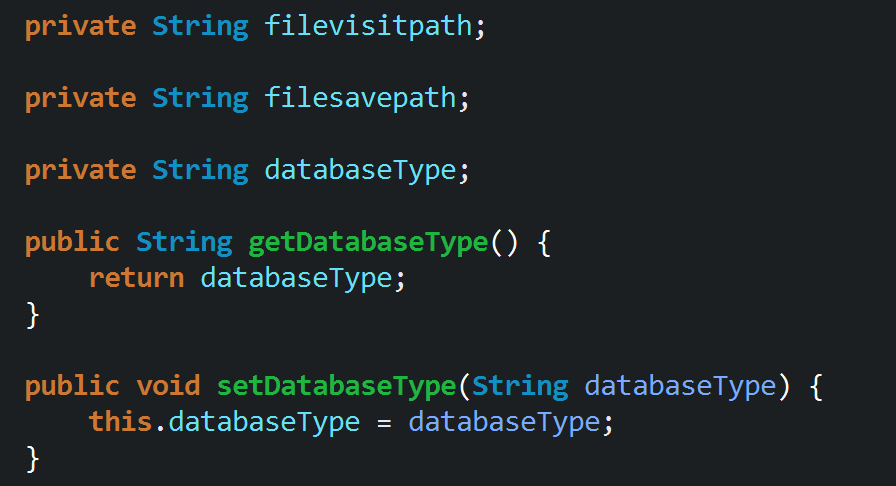
右击 **MsApplication.java – run as java application启动项目**

# 项目配置

项目配置文件为resource -》\*.yml文件。当然也同样可以是用properties文件。在项目中可以看到可以用application-xx.yml的形式设置不同环境用到的配置。目前项目中分为dev和prod两个。分别代表开发环境和生产环境。当然也可以添加test等。公司测试服务器环境等。

我们在application.yml 中设置引入的配置文件spring.profiles.active= dev 。端口及路径通过server: port: 8081 context-path: /。其他配置请查看官方文档。

对于项目自定义的配置。我们可以在YmlConfig.java 设置并接收。如文件保存路径等。



# 开发者选项

使用admin 123 登录系统。可以看到右侧已经有了系统管理和开发者选项。开发者选项只有在dev环境下才会出现。

## 系统枚举项管理

对系统中使用的枚举项管理。当然其实这里只是做了增删改查而已。表名和字段名大小写其实都没关系。都弄大写只是看起来比较好看。

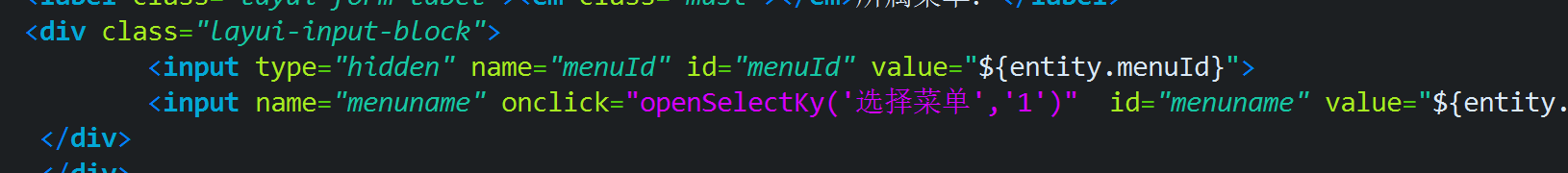


## 选择弹出框管理

在实际项目开发中，我们很多时候需要从另一张表里选择字段填写当前表的字段完成表单填写。选择弹出框正式解决这个问题。以操作按钮选择菜单为例：



填写好表单后在jsp 页面直接调用，后面的1 为选择弹出框生成的主键值。对于多选方法名为 openSelectKyd.



## 功能按钮管理

主要针对列表上方区域的操作按钮进行配置管理。





其中请求地址即你在后端编写的接口地址。需要注意的是 需要ID的接口，不管是否批量，勾选checkbox ，前端都会以id为名称吧你勾选的id值传到后台（多个“,”隔开）。

权限字符串对应的你接口上添加的shiro 权限注解。这里对权限字符串的使用建议是:对于项目的一般逻辑可以不使用，对于项目的核心业务一定要添加权限字符串。

***@RequiresPermissions***("role:allotbtn")

## 菜单管理

对系统菜单进行管理



对于菜单固定参数还需要在list.jsp 页面添加一些东西。

<table class=*"layui-table"* lay-data="*{height:500, url:'*${basePath}*/crud/list/sys\_role?fixedprams=*${fixedprams }*', page:true,limit:10,limits:[10,50,100,200,500] ,id:'comontable'}*" lay-filter=*"test"*>

# 通用crud

## 查询列表

1. 配置

在list.jsp 页面中配置表头信息[url:'${basePath}/crud/list/#](url:'$%7bbasePath%7d/crud/list/#)表名#'。

在ListDAO中添加findxxxList 方法 ，如：List<Map<String,String>> findSysMenuList(Map<String,String> map);

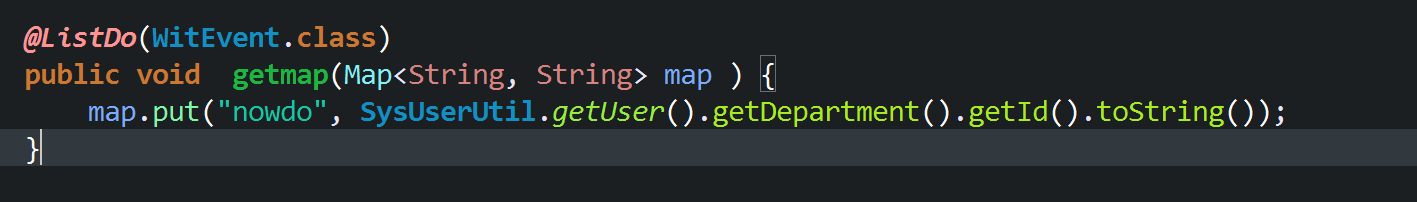
然后再mapper文件中添加对应的select 语句。或者使用@select 注解。

1. 添加动态内容

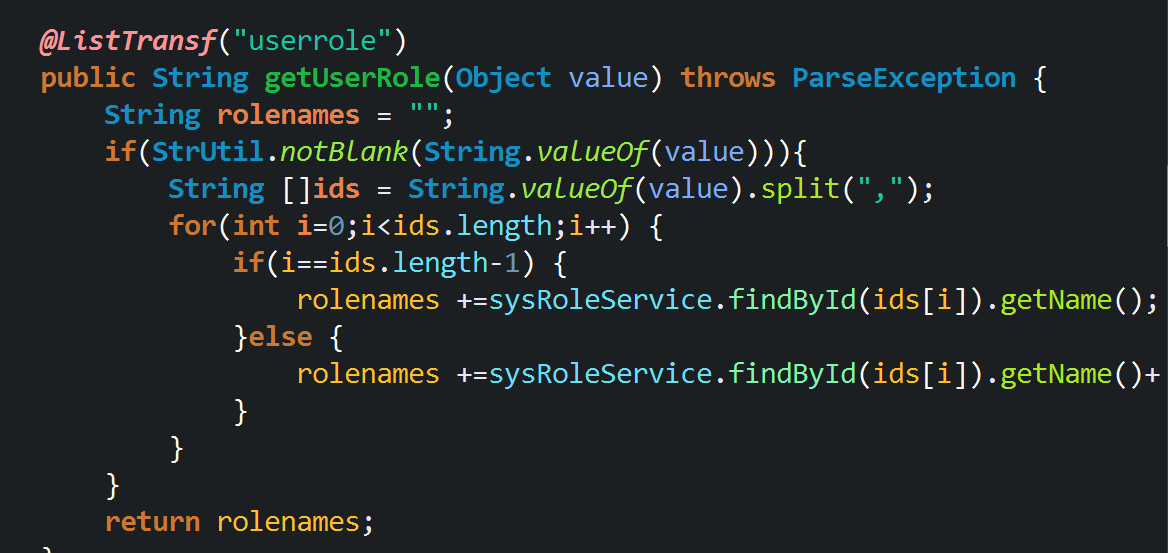
有时候我们需要添加当前登录用户的信息等到查询中。我们可以添加入下代码.

你有几点需要注意：

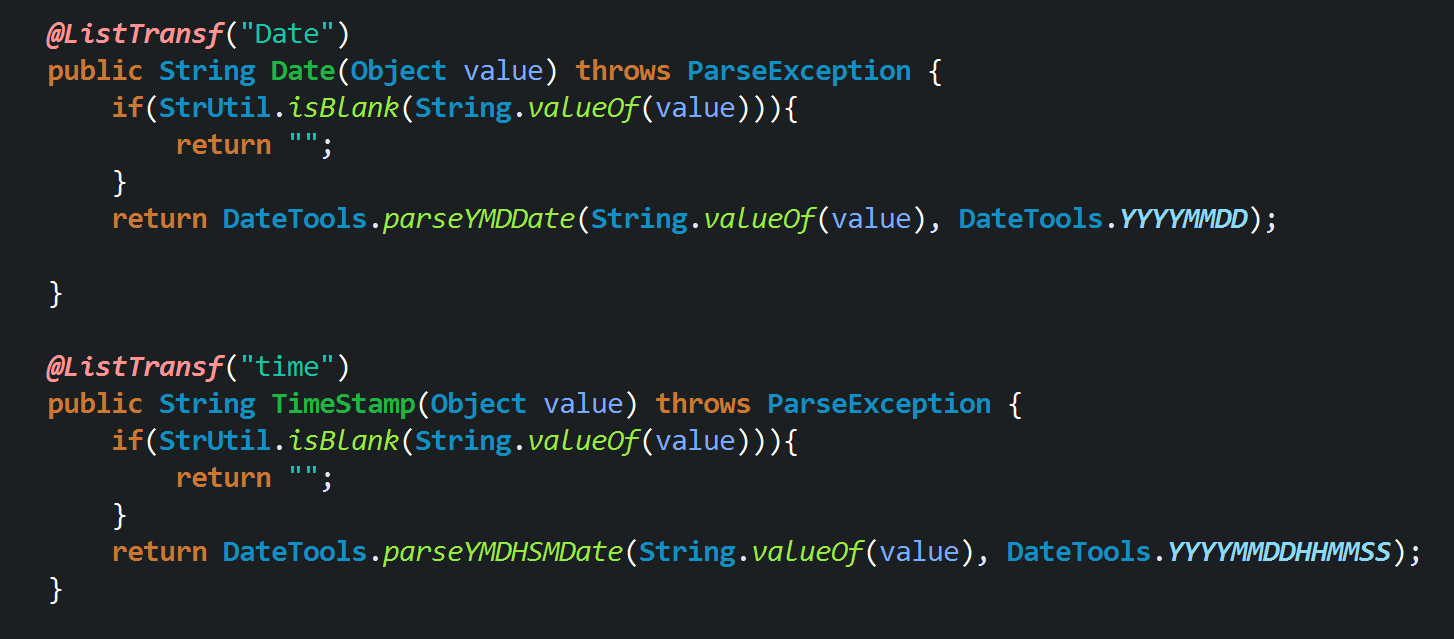
1. 必须定义在serviceimpl.java 中。
2. 同一个model 只能定义一个。
3. 参数是固定的，Map<String,String> map



1. 结果处理
   1. 对于枚举项处理：canbatch as 'E-sys\_action-canbatch'。使用大写E开头，E-表名-字段，前台返回的将会是枚举项中设置的结果值，名称为字段的驼峰名。
   2. 对于其他转换 ：role\_ids as 'T-userrole-role\_ids'。使用大写T开头。以用户角色名称转换为例。在TransFormUtil.java添加方法。对结果进行转换,这里的ListTransf注解值不区分大小写。返回前台的名称为roleIds



在转换类中已经有了基本的时间和日期转换。

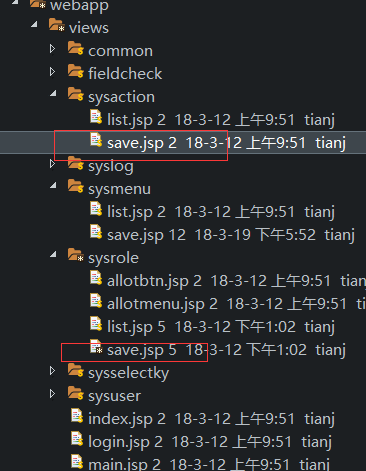


## 通用新增编辑接口

1. 功能按钮接口路径设置

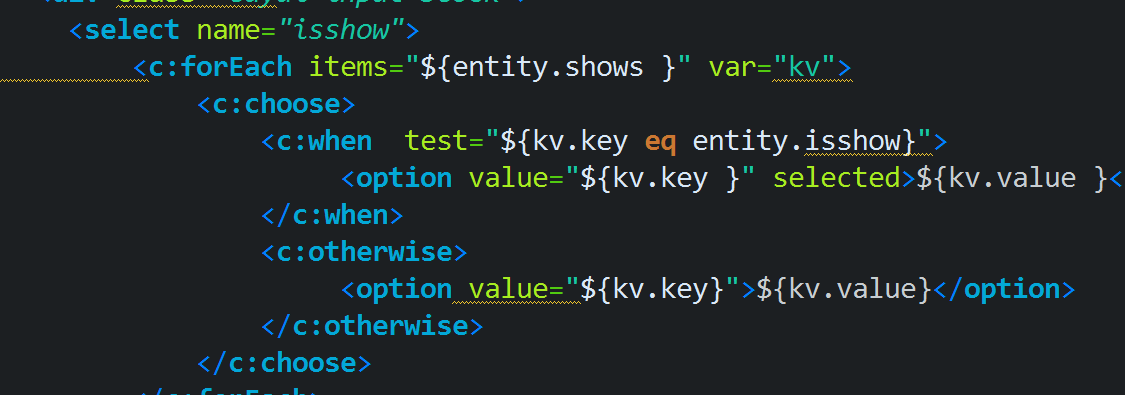
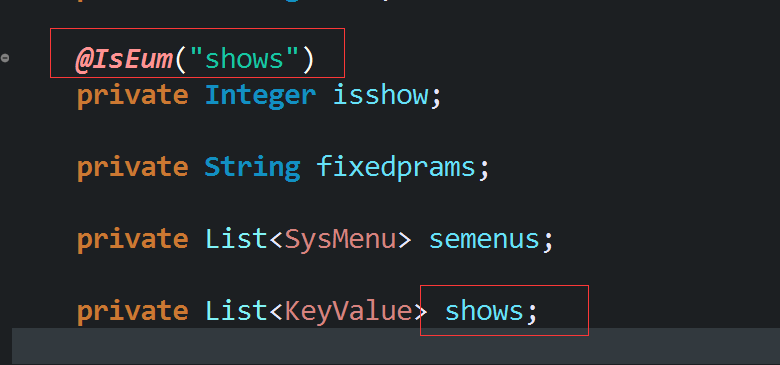
/crud/tosave/#表名#

1. 新增页位置



1. 枚举项展示

使用IsEum在实体类中对字段进行设置。如下。页面会接收到entity.shows 。



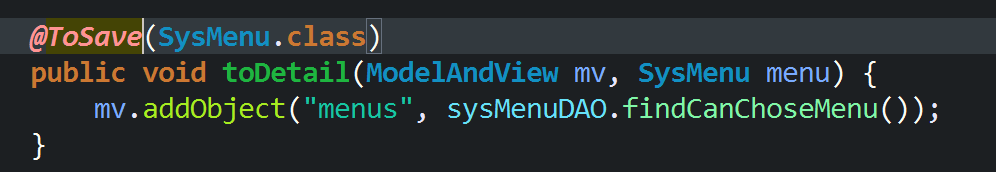
1. 个性化内容

以菜单选择父菜单为例。在SysMenuServiceImpl.java 中添加入下代码.对应个性化定义你有几点需要注意：

a必须定义在serviceimpl.java 中。

b同一个model 只能定义一个。

c参数是固定的，第一个为modelAndView ，第二个为你的model



1. 表单提交路径

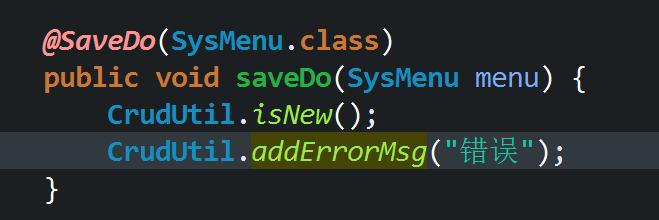
action="${basePath }/crud/save/#表名#" method="post"

1. 保存时的个性化

在保存时我们可能有如下几个问题需要解决

1. 额外的生成副标内容
2. 因为新增和修改是同一个接口。需要判断当前是否是新增而做不同的业务处理
3. 添加个性化校验，我需要返回错误信息。

处理方法如下：



代码CrudUtil.isNew 用来判断当前操作是否是新增。

CrudUtil.addErrorMsg 用来添加自己的错误信息。前端会返回通用的错误页面。

当然同样的设置SaveDo 注解需要注意几点：

a必须定义在serviceimpl.java 中。

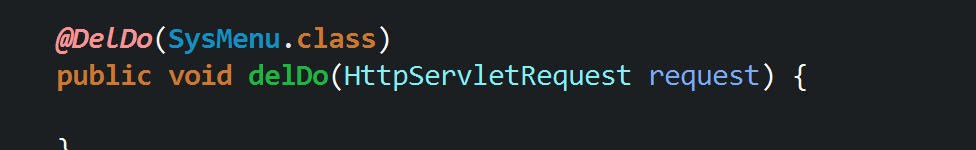
b同一个model 只能定义一个。

c参数是固定的，第二个为你的model

## 通用删除接口

1. 逻辑删除接口 /crud/ logicald/#表名#。
2. 物理删除接口 /crud/ physicald/#表名#, 默认表拥有status 字段，且为int 型。
3. 个性化操作

在servimpl.java 中添加如下代码编写个性化操作



## 通用详情接口

1. 接口地址

/crud/todetail/#表名#

1. Jsp页面定义

在表名对应文件夹下的detail.jsp

1. 枚举项及关联表
   1. 枚举项

使用上文定义的IsEum 可以将枚举项返回页面

* 1. 关联表

使用RltsField 注解



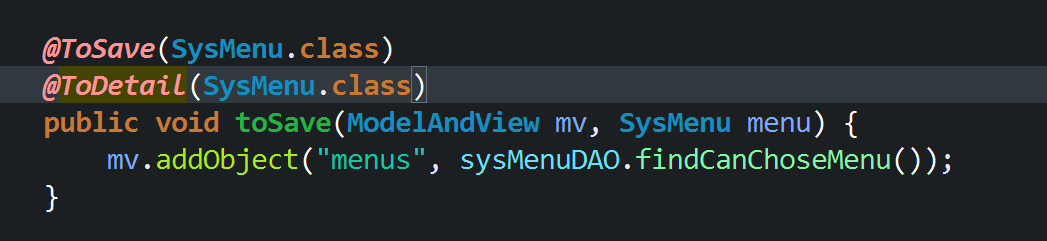
Value 为关联表model.

Otm 代表是否一对多

Result\_container 为model中存放内容名称。

1. 个性化

需要添加更为复杂数据的情况下。可以使用ToDetail 注解。如



# 结束语

在使用中大家遇到什么问题、bug或者又更好的想法可以随时跟我沟通。目前只是开发了一点点东西。后续可能会集成更多内容。