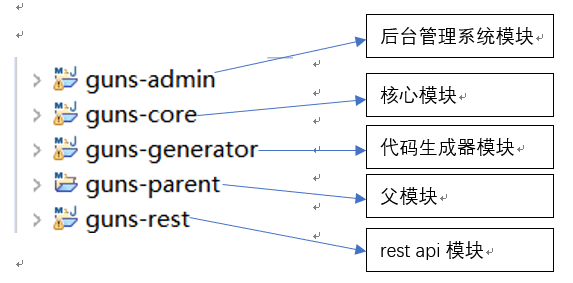
# guns技术文档

## 关于guns

基于SpringBoot，整合springmvc + shiro + mybatis-plus + beetl + flowable

### guns模块结构



1. guns-admin 模块为后台管理系统模块，包括管理系统的业务代码，前端页面，项目的配置信息等；
2. guns-core 模块为抽象出的核心（通用）模块，以供其他模块调用，此模块主要封装了一些通用的工具类，公共枚举，常量，配置等；
3. guns-generator 为代码生成模块，其中代码生成模块整合了mybatis-plus的代码生成器和guns独有的代码生成器，可以一键生成entity，dao，service，html，js等代码；
4. guns-parent 模块为其他所有模块的父模块，主要功能是管理项目中所有用到的jar，以

及聚合其他模块；

1. guns-rest 为专门提供restful api的模块，该模块中主要实现了jwt鉴权和传输数据签名的机制。

### 包结构

## 开发

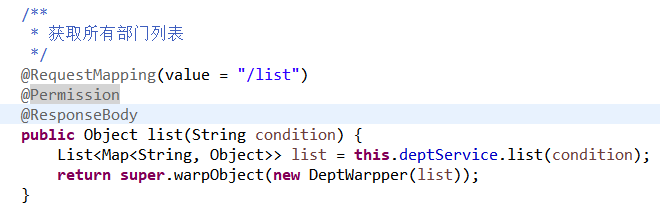
### 用户，角色和资源

用户、角色和资源（或者说权限），这三者的关系是用户对应角色， 角色对应资源，菜单和所有的按钮都可以看做是资源(或权限)，把某一个角色赋予相应的资源，那么该角色就会有访问该资源的权限，否则，该角色访问这些被管控的资源就会被服务器返回403，没有权限，当角色绑定资源后还需要给用户赋予角色才可以让登录的用户访问相关服务器接口。

一句话概括: 用户对应角色，角色对应资源

### 资源的权限控制

通过在控制器Controller上添加@Permission 注解进行权限校验

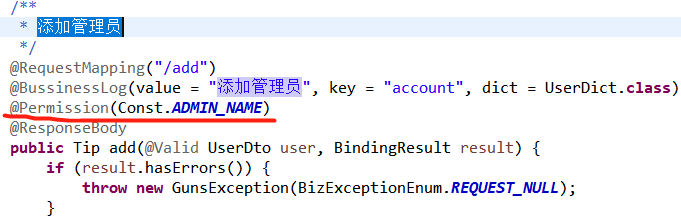


通过查找用户对应的角色，并查找角色对应的资源，可以找到当前用户是否有该资源

的权限。

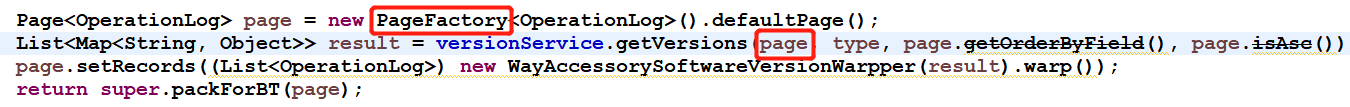
@Permission 注解中可以带一个String数组类型的参数，加上该参数，则接口被限制为

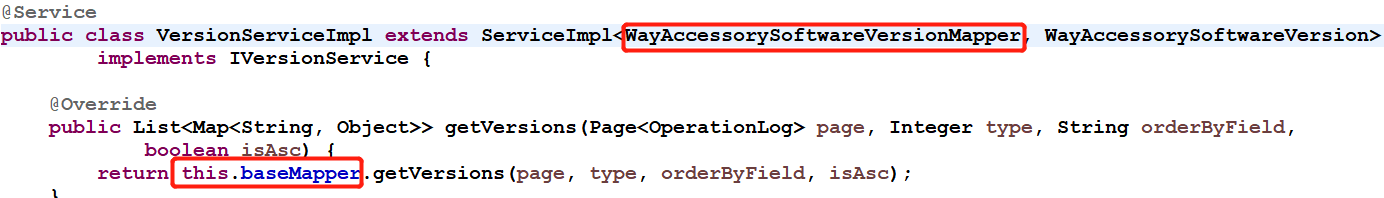
只有某个或某些角色才可访问



权限的检查是通过AOP 拦截@Permission 注解完成的，当访问受权限控制的资源时， AOP 对当前请求的servletPath 和数据库中sys\_menu 表的url 字段进行匹配，如果当前用户所拥有的权限包含当前请求的servletPath ，则访问这个接口成功

### 2.3如何分页





传递了Page 对象，对应的XXMapper的第一个参数设置为Page 对象。

当mybatis执行此方法的时候，会被mybatis-plus的分页插件自动拦截到，并且把分页查询的结果返回到这个Page 对象中!

### 2.4统一异常拦截

@ControllerAdvice注解。

GlobalExceptionHandler 类中做统一异常拦截处理，GlobalExceptionHandler 类中可以拦截所有控制器执行过程中抛出的异常

@Exception注解：指定拦截何种异常；

@ResponseStatus:指定拦截到异常后的http状态码；

@ResponseBody:拦截到异常后返回json或者页面。

### 2.5异常日志记录

数据库记录日志：

Guns的日志记录工具类：

LogManager.me().executeLog();

Guns系统有统一的异常拦截器（2.4），一般程序的报错，不管是业务异常还是未知的

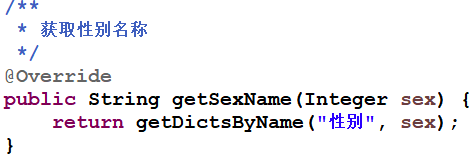
RuntimeException都会拦截并记录到数据库（sys\_operation\_log），

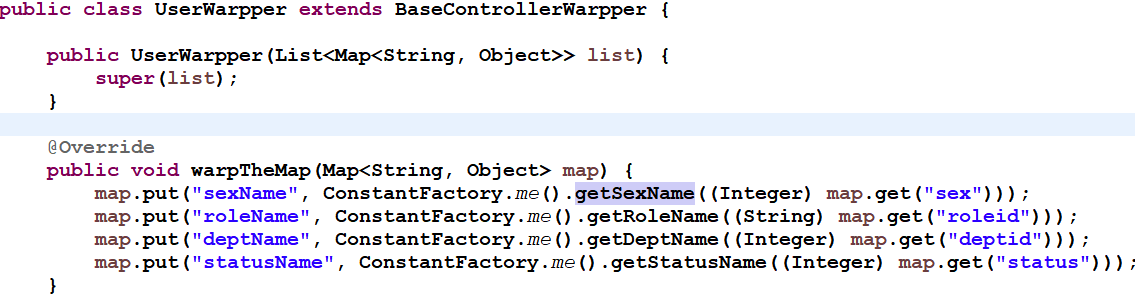
### 2.6结果包装类—XXWarpper

数据库查询时，某些表中字段存储的是数字，但前端展现时需要展现数字所对应的值（如1-男，2-女），此时需要用到结果包装类：

XXWarpper 类继承BaseControllerWarpper 类，主要是实现warpTheMap() 方法，也就是具体的包装过程，warpTheMap() 方法的参数map就是被包装的原始数据的每个条目，我们可以在这每个条目中增加一些字段也就是被包装字段的中文名称

ConstantFactory：常量的生产工厂，数据库与包装类的桥梁，将数字与对应的值转换(字典功能)的工具。





### 2.7关于${ctxPath}

页面中引入的${ctxPath}, ctxPath变量是在哪里定义的?

Beebl自带: ctxPath: 表示Web应用ContextPath

### 2018-08-10

删除项目根目录logs/dairy文件夹