目录

[一、项目介绍 2](#_Toc536029669)

[1、项目的导入(以IDEA为例,Eclipse类似) 2](#_Toc536029670)

[2、项目结构介绍 3](#_Toc536029671)

[3、部署项目 4](#_Toc536029672)

[二、开发功能 5](#_Toc536029673)

[1、在线代码生成 5](#_Toc536029674)

[2、添加任务计划 6](#_Toc536029675)

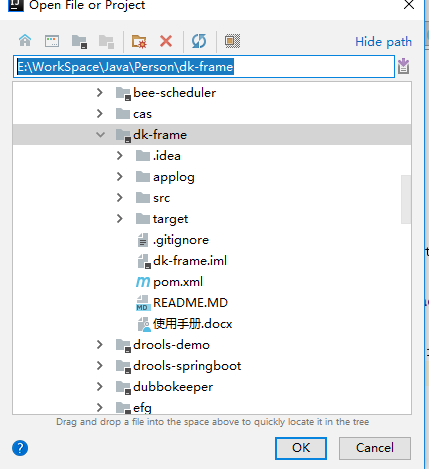
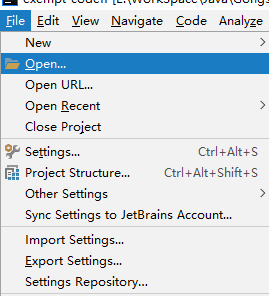
[3、添加过滤器 7](#_Toc536029676)

[4、修改认证路径 8](#_Toc536029677)

# 一、项目介绍

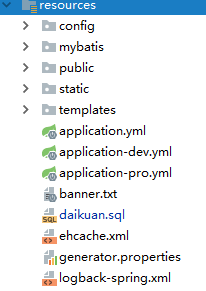
本项目以主流java技术为框架，搭建的一个简单易用的开发框架，用的技术有: Maven、JDK8、SpringBoot2.0、Shiro、Mybatis、Thymeleaf、Bootstrap、Jquery，使用的第三方中间件有: Redis、Ehcache，支持代码的一键生成…现已实现用户管理、角色管理、权限管理、部门管理、菜单管理、公告通知、数据字典、在线用户、图表展示等功能，系统方面有：系统日志、系统监控、接口测试、计划任务、SQL监控等功能，满足日常的基本管理系统的开发需求

## 1、项目的导入(以IDEA为例,Eclipse类似)



## 2、项目结构介绍





**config**: ehcach配置、quartz配置

**mybatis**: mybatis访问数据的sql，按照模块进行分目录，blog、common、oa、system

**public**: 公共资源

**static**: 静态资源

**templates**: 网页模板，同样按照模块进行分目录，blog、common、oa、system

**application.yml**：项目通用配置，端口、session、mybatis、页面模板

**application-\*.xml**：数据库、redis、日志等的配置

**dev**: 开发环境

**pro**: 生产环境

**daikuan.sql**: 默认的数据表结构及初始化数据

**ehcache.xml**: ehcache缓存的配置

**logback-spring.xml**：日志路径配置

## 3、部署项目

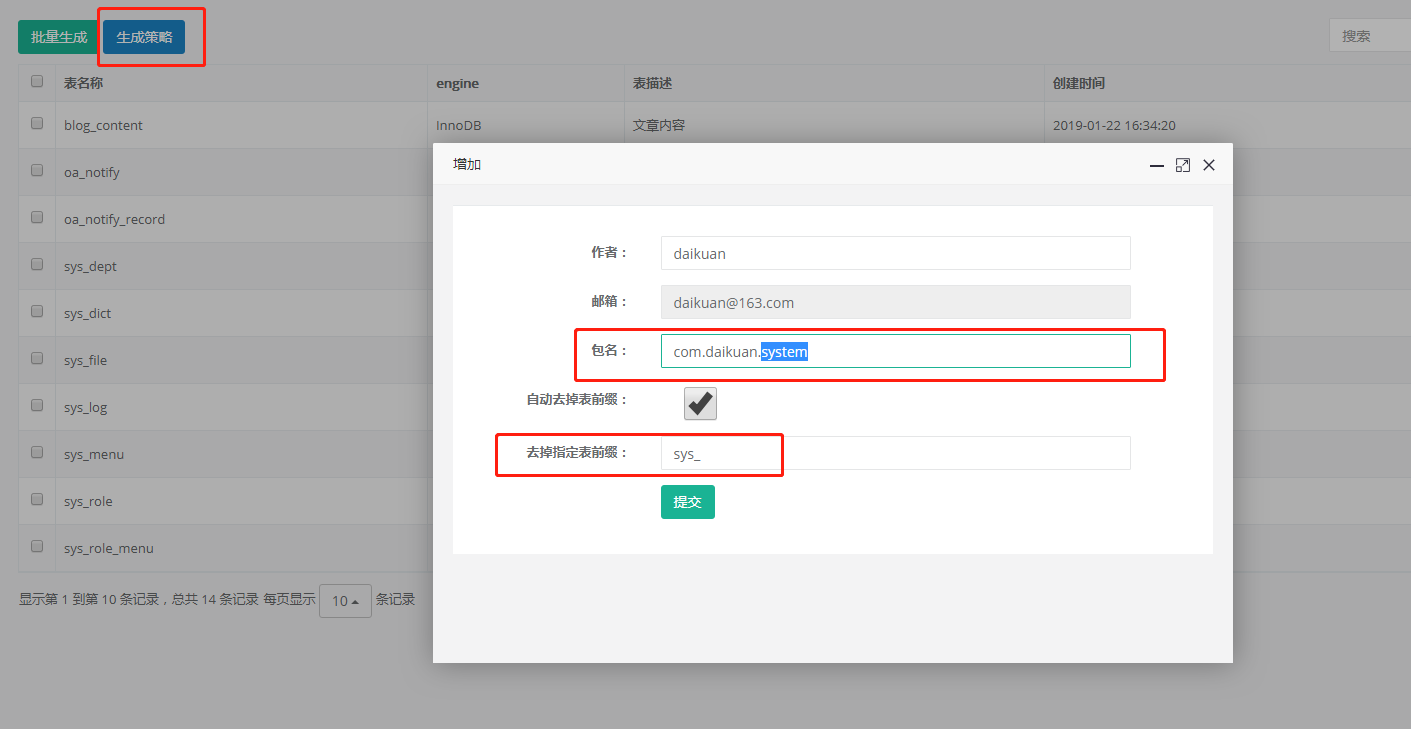
在pom.xml的同级目录下

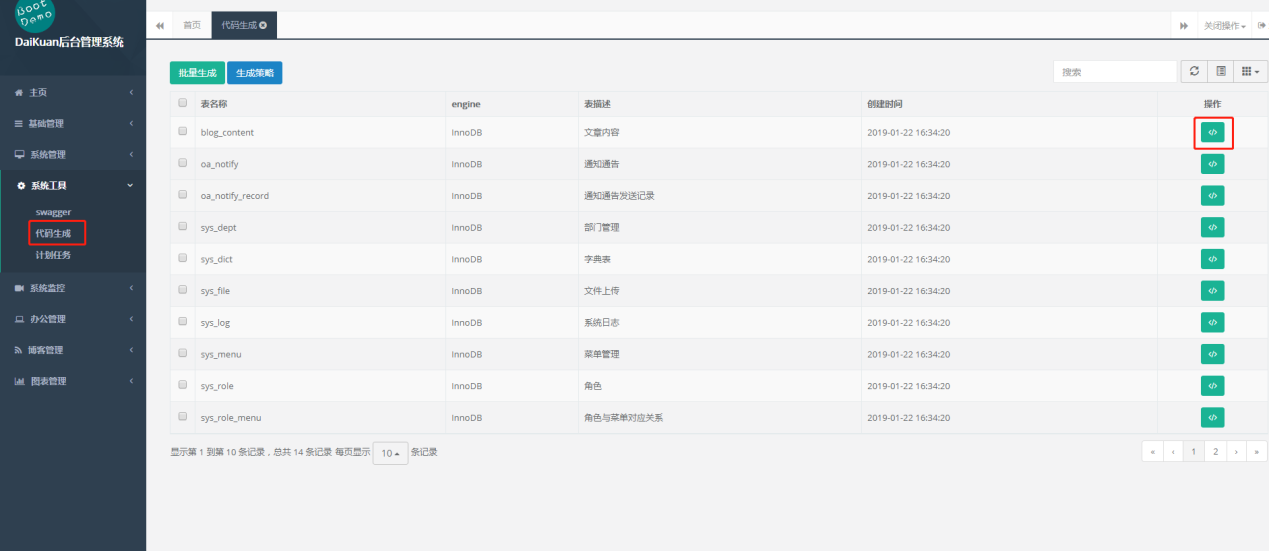
1. #执行daikuan.sql /mysql -u账号 -p密码 -D数据库名 < sql文件绝对路径
2. mysql -uroot -p123456 -Dtest<./daikuan.sql
3. ## 打包，跳过测诿/span>
4. mvn clean **package** -DskipTests=**true**
6. #启动项目 --server.port指定启动端口  --spring.profiles.active激活运行环境¿/span>
7. #nohup java -jar \*.jar --server.port=端口 --spring.profiles.active=激活环境¿gt;catalina.log &
8. nohup java -jar dk-frame-1.0.0-SNAPSHOT.jar --server.port=8080 --spring.profiles.active=dev >catalina.log &

# 二、开发功能

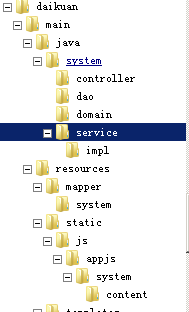
## 1、在线代码生成

首先在数据库里面创建需要的标，登录系统，找到“系统工具> 代码生成”，点击生成策略，修改生成后的包名，选择刚创建的标，点击做右侧的生成按钮即可。





生成完后的代码是以jar包的形式下载，目录如下：

，按生成的目录结构导入的项目中即可。

批量生成，勾选需要生成的代码的表，点击左上角的“批量生成”，下载导入代码即可。

**注意：此功能切勿在生成环境使用，影响性能，可以把此功能的权限给屏蔽掉。**

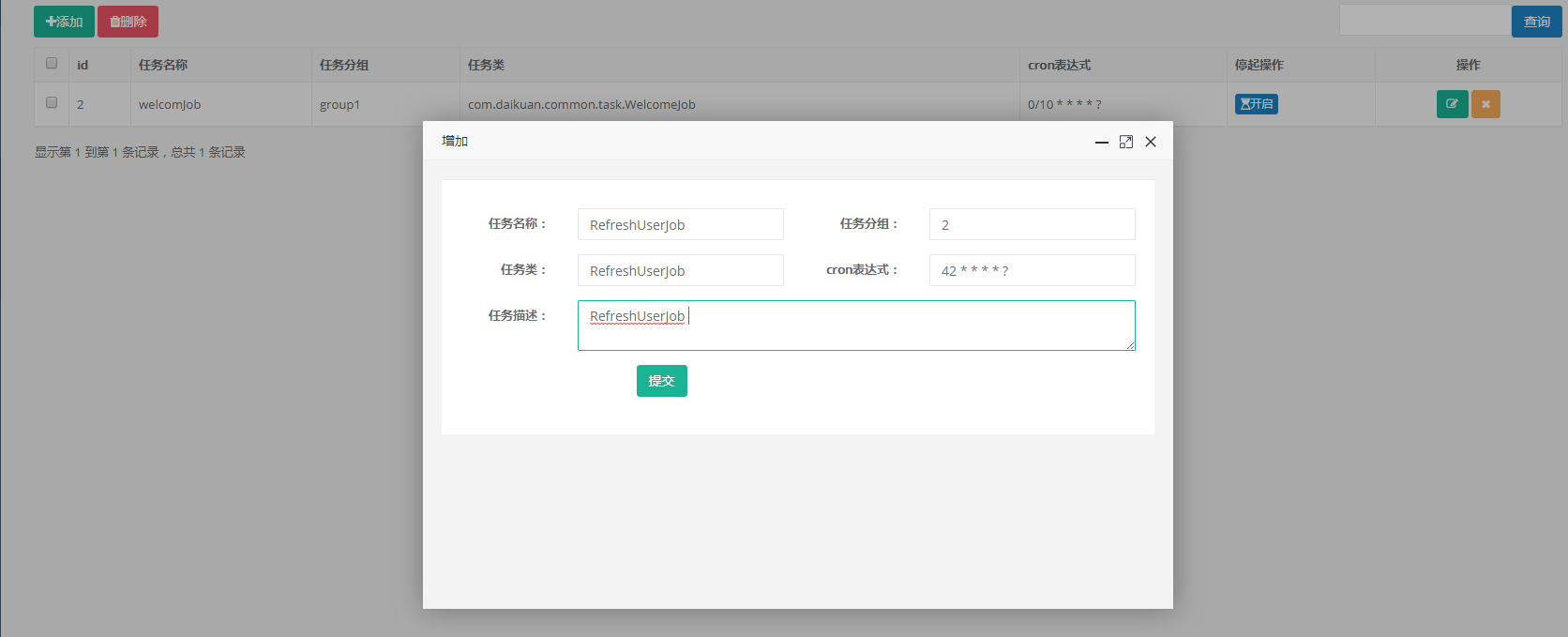
## 2、添加任务计划

任务计划，即是定时任务，在**com.daikuan.common.task**下开发

首先新的定时任务实现**org.quartz**包下**Job**接口，实现public void execute(JobExecutionContext context) 方法，在实现类上添加**@Component**注解，在注入所需要的接口接口，eg:

1. @Component
2. **public** **class** RefreshUserJob **implements** Job {
4. @Autowired
5. **private** UserService userService;
7. @Override
8. **public** **void** execute(JobExecutionContext context) **throws** JobExecutionException {
9. Set<String> set = userService.listRoles(111L);
11. }
12. }

然后在控制台上添加此任务，“系统工具 > 计划任务”



1、任务名称，及Job的名称

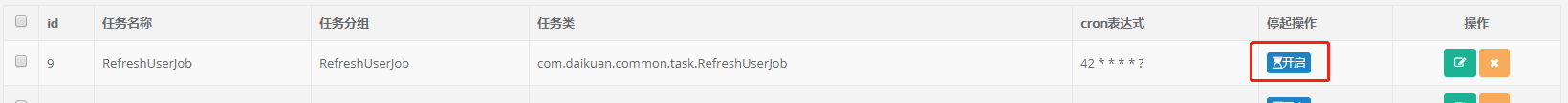
2、任务分组：任务所属的组

3、任务类：添加任务的类的全路径，新添加的那个任务为：com.daikuan.common.task.RefreshUserJob

4、cron表达式：任务的执行时点，可以在此配置生成: http://cron.qqe2.com/

5、任务描述

6、提交保存后的任务默认是关闭的，则需要手动开启



**注意：目前的部署方式不支持任务的集群，如果需要修改为集群模式，在com.daikuan.common.quartz下做相应的修改。**

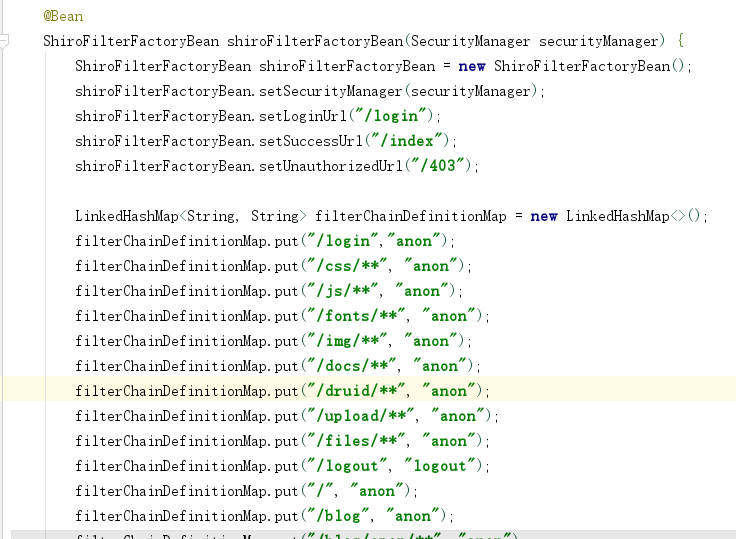
## 3、添加过滤器

首先实现**javax.servlet**包下的**Filter**，重写Filter的方法，在**doFilter**中添加需要处理的业务，然后配置新增的Filter，eg:

1. @Configuration  //这个很重要
2. **public** **class** XssConfig{
3. @Bean  //这个也能很重要
4. **public** FilterRegistrationBean xssFilterRegistrationBean() {
5. FilterRegistrationBean filterRegistrationBean = **new** FilterRegistrationBean();
6. filterRegistrationBean.setFilter(**new** XssFilter());  //配置创建的Filter
7. filterRegistrationBean.setOrder(1);
8. filterRegistrationBean.setEnabled(**false**);
9. filterRegistrationBean.addUrlPatterns("/\*");  //拦截的路径
10. Map<String, String> initParameters = Maps.newHashMap();
11. initParameters.put("excludes", "/favicon.ico,/img/\*,/js/\*,/css/\*");  //不拦截的
12. initParameters.put("isIncludeRichText", "true");
13. filterRegistrationBean.setInitParameters(initParameters);
14. **return** filterRegistrationBean;
15. }
16. }

## 4、修改认证路径

比如新增一个/news开头的请求路径，是允许不登录即可访问的，则需要在com.daikuan.system.config.ShiroConfig里面进行修改，



anon : 表示匿名可以访问，即没有登录就可以访问 ，/a/\*\*

authc：表示需要认证后才能访问，/a/\*\*\*

## 5、添加访问日志

在需要记录访问日志的的方法上添加**@Log(名称)** 即可

6、