Echo 项目结构分析

原创 飞天小牛肉 [飞天小牛肉](javascript:void(0);)

**飞天小牛肉**

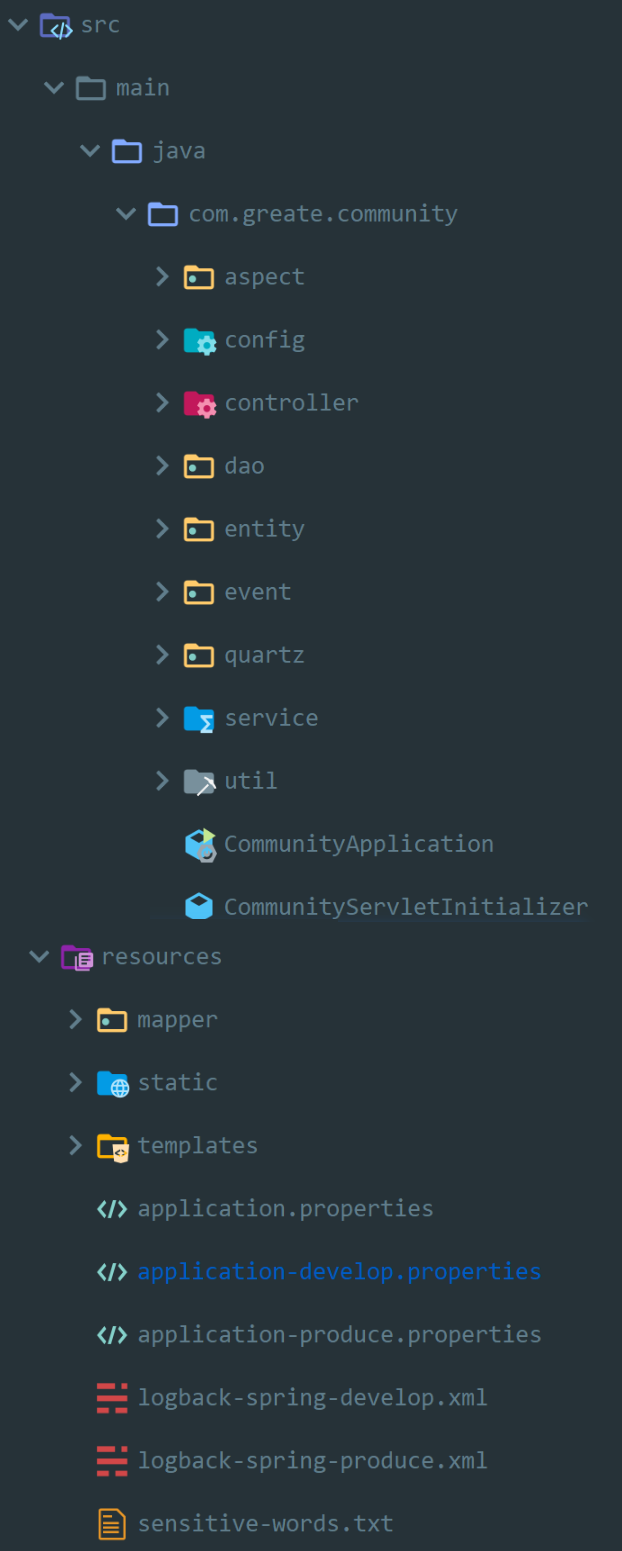
微信号 CS-Wiki

功能介绍 InfoQ & 阿里云签约作者，分享原创技术干货和成长经验

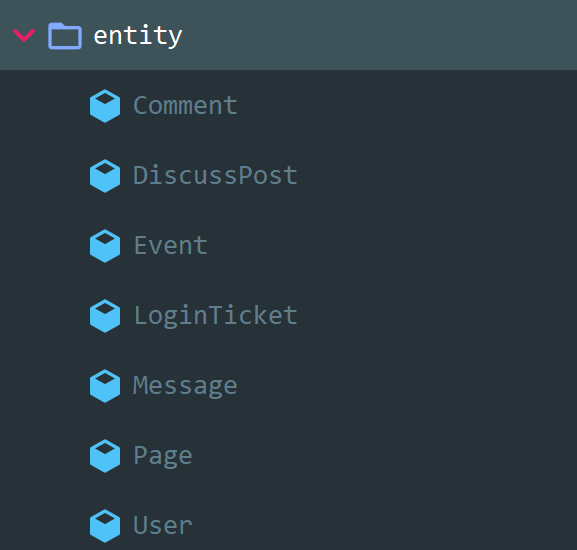
2021-02-19[原文](https://mp.weixin.qq.com/s?__biz=MzI0NDc3ODE5OQ==&mid=2247485379&idx=2&sn=3b650582bdb22a32daf067a5d0798b5e&chksm=e959d27bde2e5b6d6180e6d524d1cf0516af03570ebfbe8525b1213aab8153f5f9fbc602cde7&scene=178&cur_album_id=1744497649518493697#rd) 发表于

收录于合集 #Echo 学习教程 18个

Echo 项目后端采用 MVC 模式，使用现在流行的 SpringBoot 框架。SpringBoot 是基于 SpringMVC 衍生出来的框架。宗旨是较少配置，让开发者快速上手做项目。

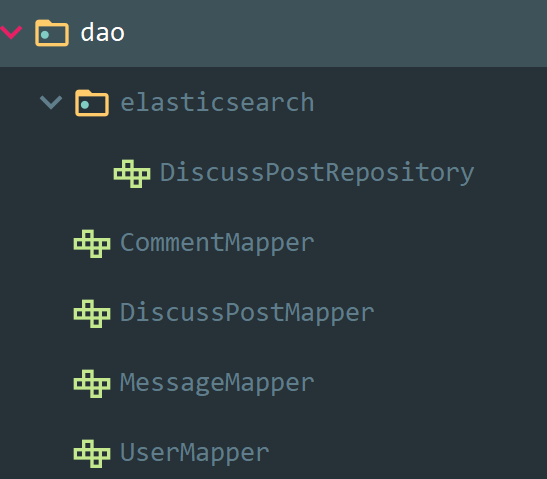


**entity**



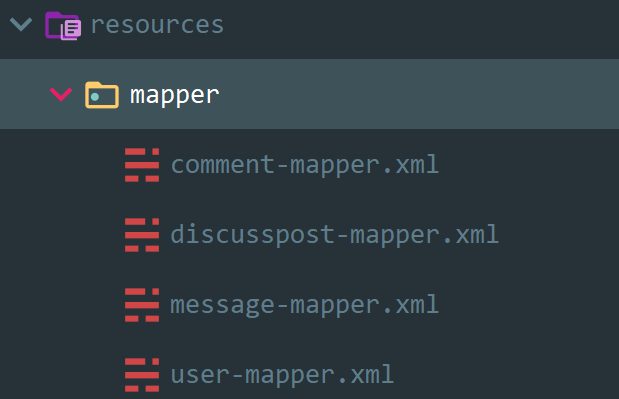
* User：数据库模型，对应表 user
* Comment：数据库模型，对应表 comment
* DiscussPost：数据库模型，对应表 discuss\_post
* Message：数据库模型，对应表 message
* Page：分页模型
* Event：事件模型（Spring Kafka）
* LoginTicket：登录凭证

**dao**

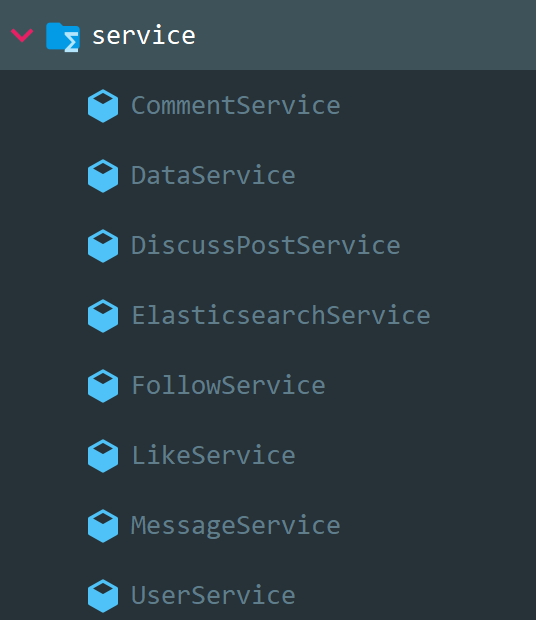


* CommentMapper：控制 Comment 的相关数据库操作
* DiscussPostMapper：控制 DiscussPost 的相关数据库操作
* MessageMapper：控制 Message 的相关数据库操作
* UserMapper：控制 User 的相关数据库操作
* DiscussPostRepository：控制 Elasticsearch 服务器上 DiscussPost 的相关操作

当然，上面这些都只是操作接口，真正操作数据库的文件放在 resources/mapper 目录下：

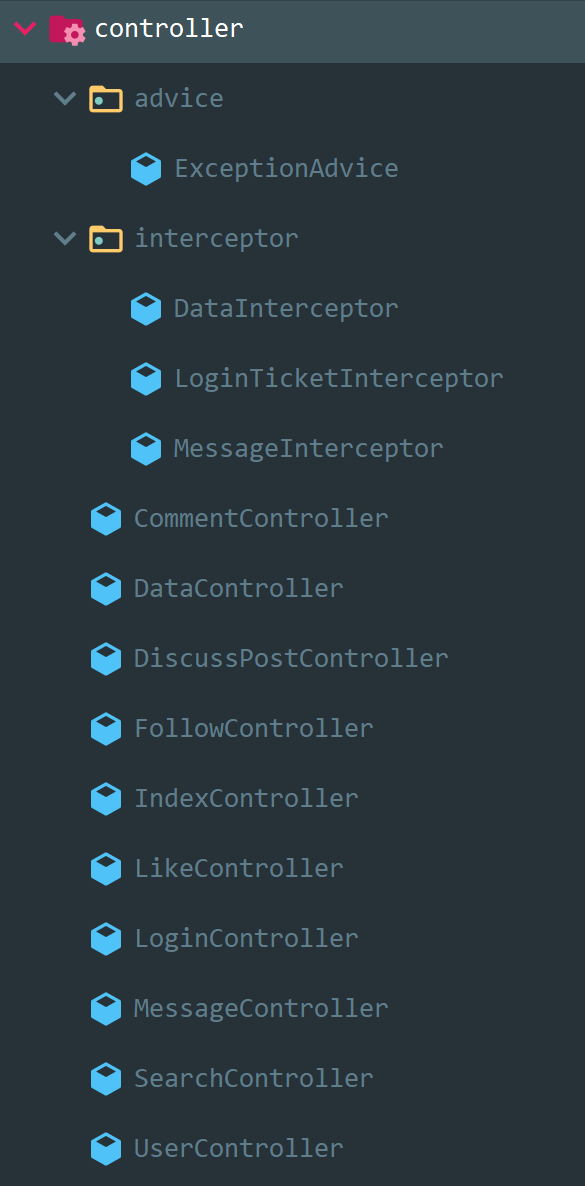


**service**



* CommentService：Comment 相关的业务逻辑
* UserService：User 相关的业务逻辑
* MessageService：Message 相关的业务逻辑
* DiscussPostService：DiscussPost 相关的业务逻辑
* FollowService：关注相关的业务逻辑
* LikeService：点赞相关的业务逻辑
* ElasticsearchService：Elasticsearch 搜索相关的业务逻辑
* DataService：数据统计相关的业务逻辑

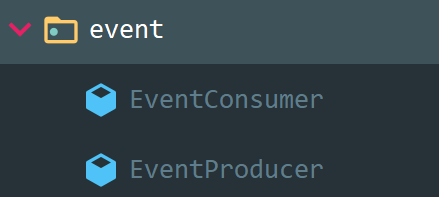
**controller**



* UserController：控制 User 相关的业务流程
* CommentController：控制 Comment 相关的业务流程，比如添加评论等
* DiscussPostController：控制 DiscussPost相关的业务流程，比如添加帖子、进入帖子详情页、置顶帖子、加精帖子等
* MesaageController：控制 Message 相关的业务流程，比如进行私信列表、发送私信、发送系统通知等
* LikeController：控制点赞相关的业务流程，比如点赞、取消点赞等
* FollowController：控制关注相关的业务流程，比如关注、取消关注、进入某个用户的关注列表等
* SearchController：控制搜索相关的业务流程，比如进入搜索界面等
* DataController：控制数据统计相关的业务流程，比如进入统计界面、统计网站 DAU、统计网站 uv 等
* IndexController：控制首页相关的业务流程，比如进入首页、进入 500/404 错误界面等
* LoginController：控制登录注册相关的业务流程，比如登录、登出、注册、激活用户等
* advice：
  + ExceptionAdvice：处理服务端异常（500）
* interceptor：拦截器
  + MessageInterceptor：获取未读私信/系统通知的数量
  + LoginTicketInterceptor：检查凭证状态，若凭证有效则在本次请求中持有该用户信息
  + DataInterceptor：统计网站 UV 和 DAU

**event**

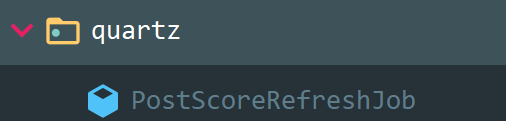
这个包主要是提供给 Spring Kafka 使用的：



* EventProducer：事件的生产者（将事件发布到指定的主题）
* EventConsumer：事件的消费者

**quartz**

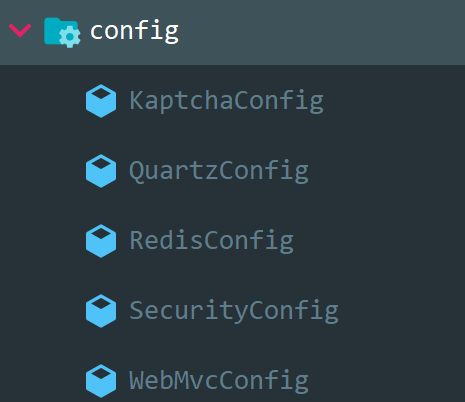
这个包主要是提供给 Spring Quartz 使用的：



* PostScoreRefreshJob：每隔一段时间刷新帖子分数（热度）

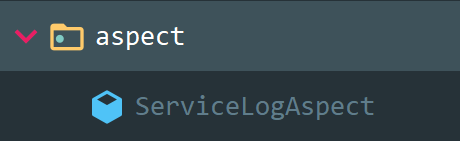
**config**

各种配置类：



* KaptchaConfig：Kaptcha（验证码）相关配置
* QuartzConfig：Spring Quartz 相关配置
* RedisConfig：Redis 相关配置
* SecurityConfig：Spring Security 相关配置
* WebMvcConfig：拦截器相关配置

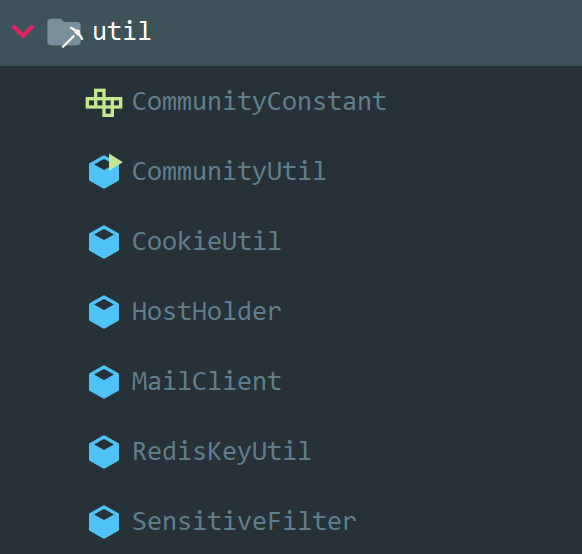
**aspect**



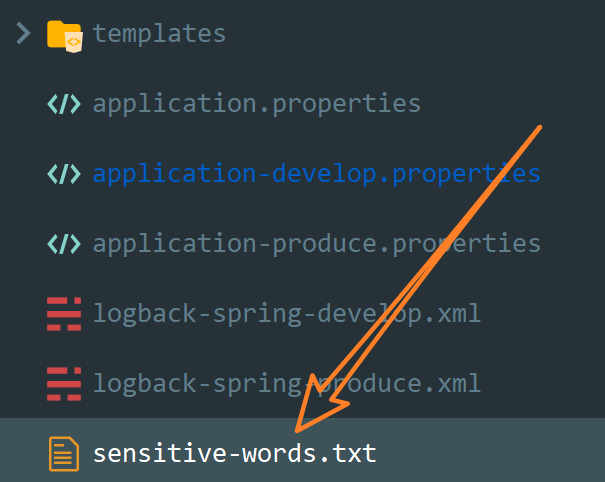
* ServiceLogAspect：使用 AOP实现统一日志记录

**util**

各种工具类：

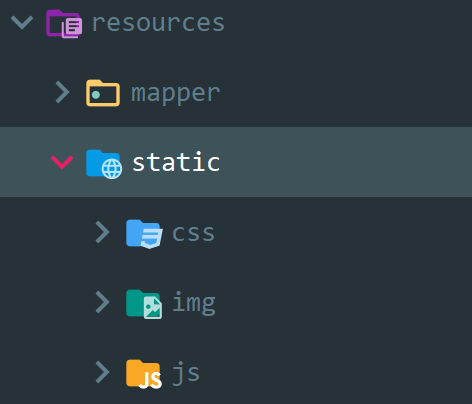


* CommunityUtil：通用工具类。比如生成随机字符串、md5 加密、将服务端返回的消息封装成 JSON 格式的字符串等
* CookieUtil：从 request 中获取指定 name 的 cookie
* HostHolder：使用 ThreadLocal 持有用户信息（多线程），用于代替 session 对象，实现 session any where
* MailClient：发送激活邮件（用于注册）
* RedisKeyUtil：生成 Redis 的 key
* SensitiveFilter：敏感词过滤器。对应的敏感词文本文件就是 sensitive-words.txt



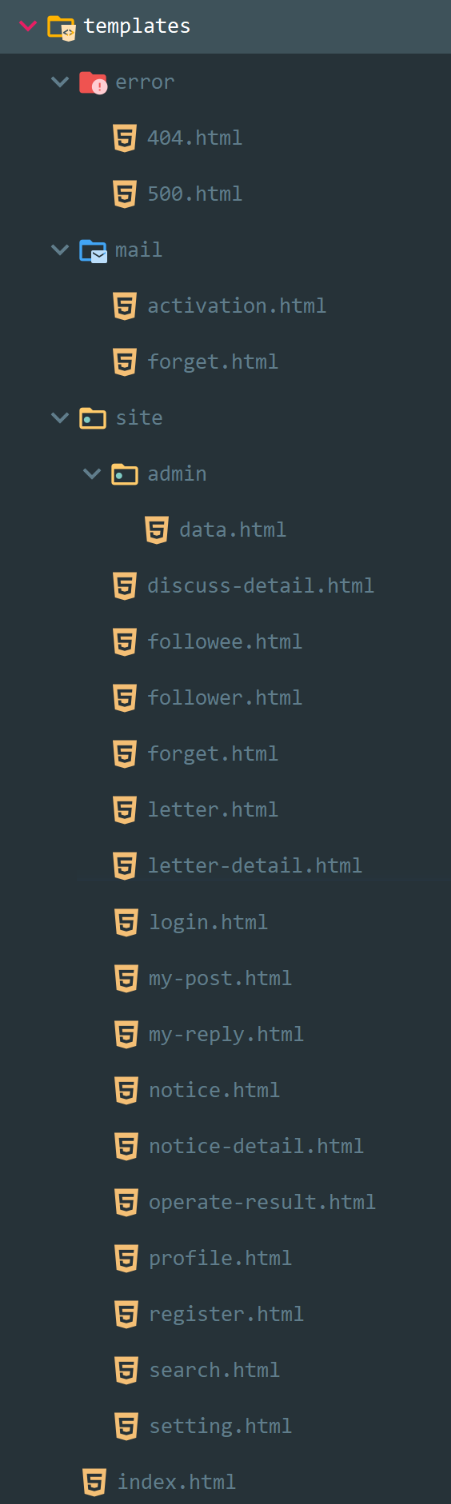
**静态资源**

resources/static：静态资源存放目录



**前台页面模板**

resources/templates：前台页面模板，包括邮件模板。

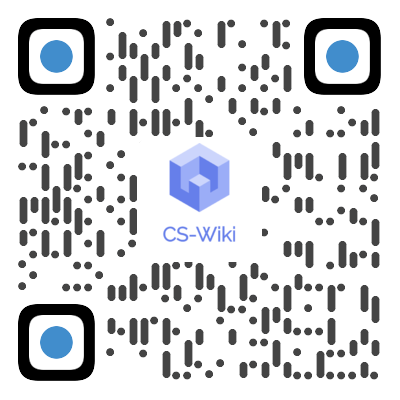


* error：错误界面模板
* mail：邮件模板
  + activation.html：激活邮件模板
  + forget.html：找回密码邮件模板（暂未使用）
* forget.html：忘记密码页面模板（暂未使用）
* operate-result.html：操作结果页模板
* login.html：登录页模板
* register.html：注册页模板
* index.html：首页（帖子列表页模板）
* discuss-detail.html：帖子详情页模板
* profile.html：个人主页模板
* my-post.html：我的帖子模板
* my-reply.html：我的回复页模板
* followee-detail.html：关注列表模板
* follower-detail.html：粉丝模板
* letter.html：私信列表模板
* letter-detail.html：私信详情页模板
* notice.html：系统通知页模板
* notice-detail.html：系统通知详情页模板
* search.html：搜索结果页模板
* setting.html：账号设置页模板
* data.html：数据统计模板



😁 点击下方卡片关注***公众号*「飞天小牛肉」**（专注于分享计算机基础、Java 基础和面试指南的相关原创技术好文，帮助读者快速掌握高频重点知识，有的放矢），与小牛肉一起成长、共同进步

🎉 并向大家强烈推荐我维护的 ***Gitee 仓库*** **「CS-Wiki」**（Gitee 推荐项目，目前已 0.9k star。面向全栈，致力于构建完善的知识体系：数据结构、计算机网络、操作系统、算法、数据库、设计模式、Java 技术栈、机器学习、深度学习、强化学习等），相比公众号，该仓库拥有更健全的知识体系，欢迎前来 star，仓库地址 https://gitee.com/veal98/CS-Wiki。也可直接下方扫码访问



原创不易，读完有收获不妨**点赞|分享|在看**支持

### 精选留言

暂无...

[阅读全文](https://github.com/Veal98/Echo)