

SpringBoot项目重构思路

![1711345926870.jpg](https://p9-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/ba20d98a1bf1433bb2672abf691e993f~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=1017&h=657&s=1319958&e=png&b=5d6165)

时间过得真快，一眨眼已经过去三个月了，今年过完年回来就忙成狗，一直忙着重构一个公司的老项目，这个项目之前是其他同事在负责，不过前段时间，部门大佬把我们都召集起来，和我们说了重构这件事，简单概括，就是项目实在维护不下去了，就目前的数据结构和项目框架，继续在这个老项目的基础上开发新需求只会耗费更大的成本，至于老系统的问题主要是以下几点：

1. **数据库混乱**：字段千奇百怪，有中文缩写，英文缩写，混合缩写、重复的字段、没有注释
2. **代码质量较低**：在编码风格上，百花齐放，比如工具类重复、返回到前端的参数有些使用Map，有些用实体类，同一种错误抛出不同的异常码、基本上都没有用swagger
3. **性能低下**：数据库单表数量太大，导致查询会很慢，因为数据量增长的确实有点快
4. **硬编码泛滥**：在特定的一些业务中，需要根据不同入参，来调用不同的数据源进行sql查询，这些方法定义完全一致，只是sql和数据源不同，老项目中是通过硬编码来实现
5. **任务耗时长**：定时器中的任务处理过慢，可能一个任务会消耗一天时间

重构的一些想法

重构后的项目依然是单体项目，为什么没有采用分布式、微服务呢，第一是考虑到项目复杂度和时间压力，第二是因为这个项目的并发量不高，主要问题是会产生大量的数据读取和计算比较，就算上线后撑不住，可以考虑水平扩展；

目前重心还是在解决上面几个问题，所以对于增删改查的体力活，就直接根据数据库表结构生成了，代码生成主要是包括实体类、控制层、业务层、持久层和相关的swagger注解。

暂定的一些方案

目前重构工作已经完成了一半左右，采用的基础框架也是比较常规，比如SpringBoot、MybatisPlus、Mysql、Oracle、swagger、Shiro、Redis

Cluster、RocketMQ、ShadingShpere等，还有一些技术方案是待定的，这里我会将现有的做法和想法列出，也希望社区小伙伴能够给出更加合理的意见，再次感谢了，如果方案设计不合理，也请大家指出。

1. **重新设计数据库表结构**：针对数据库结构混乱，其实就是重新设计，目前就是参照数据库设计原则和阿里巴巴开发手册，对字段命名、索引选择、冗余字段删除等等
2. **引入自动代码生成工具**：基础的增删改查代码由代码生成工具自动生成，提高开发效率,统一代码风格
3. **使用分库分表中间件**：数据库单表数据过大，目前的想法是使用中间件，比如mycat/ShadingJDBC/ShadingShpere，不过mycat好像不大活跃，官网都没有维护了，只有群还在零星的交流。此外还了解了分布式数据库（比如TiDB），当然这种改动略大，大概率只是参考参考。
4. **采用设计模式,注解,反射**：针对老系统的硬编码，新系统采用了一些设计模式（策略模式、工厂模式、模板方法模式）、新建注解、反射和动态数据源来处理。
5. **使用消息队列**：对于任务处理特别耗时，现在同事是增加了线程池来处理，但是我在想可以通过mq的方式，多部署几台服务器专门处理任务
6. **缓存服务(Redis Cluster)**：进一步优化查询性能

最后

接下来的一段时间，我会分享在这次重构中的实践经验，

1. 数据库重构与分库分表实践
2. 设计模式、注解、反射等知识如何应用到实际场景
3. 使用消息队列异步化处理耗时任务
4. 其他性能优化点探讨
5. 重构过程中的坑与反思

这次的重构，在框架的选型上就是普通的单体架构，主要优化还是专注在具体的代码设计上，针对不同的场景，选择哪种方式会更合理，希望这个系列文章能对您有所帮助。让我们一起探讨,不断学习,共同进步!

原文链接: <https://juejin.cn/post/7349940323025895465>