**抢红包项目介绍**

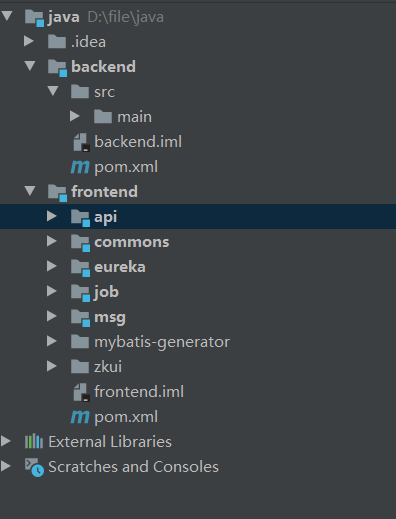
系统使用springcloud+springmvc+mybatis架构。

中间件使用：redis，zookeeper，rabbitmq,nginx。

项目主要考虑的问题有并发性，控制库存，防止超发，投放策略，中奖策略。

主要包含会员管理模块，会员等级模块，活动管理模块。

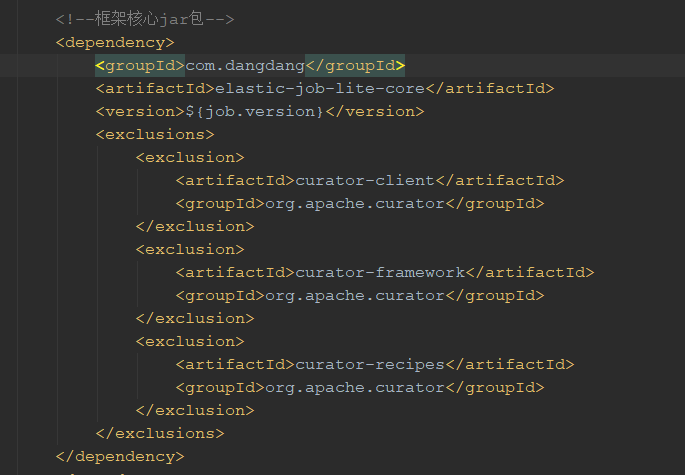
项目代码目录：

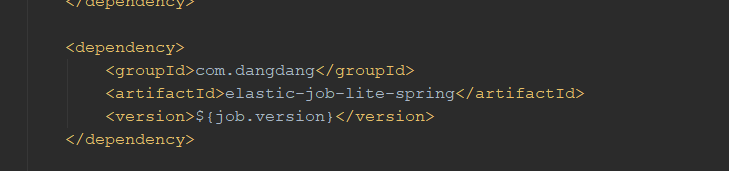


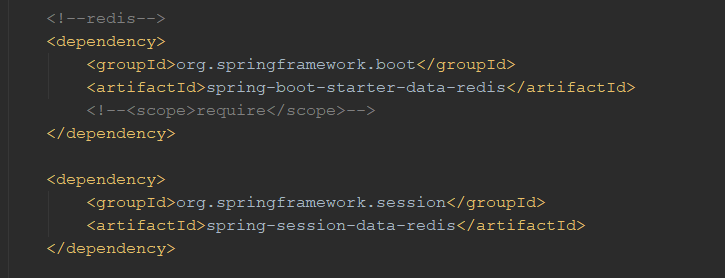
抢红包活动中，将活动提前加载进redis,以应对后续高并发会给系统带来一定的风险。

这里采用活动预热方式，使用elasticJob和redis.

导入依赖

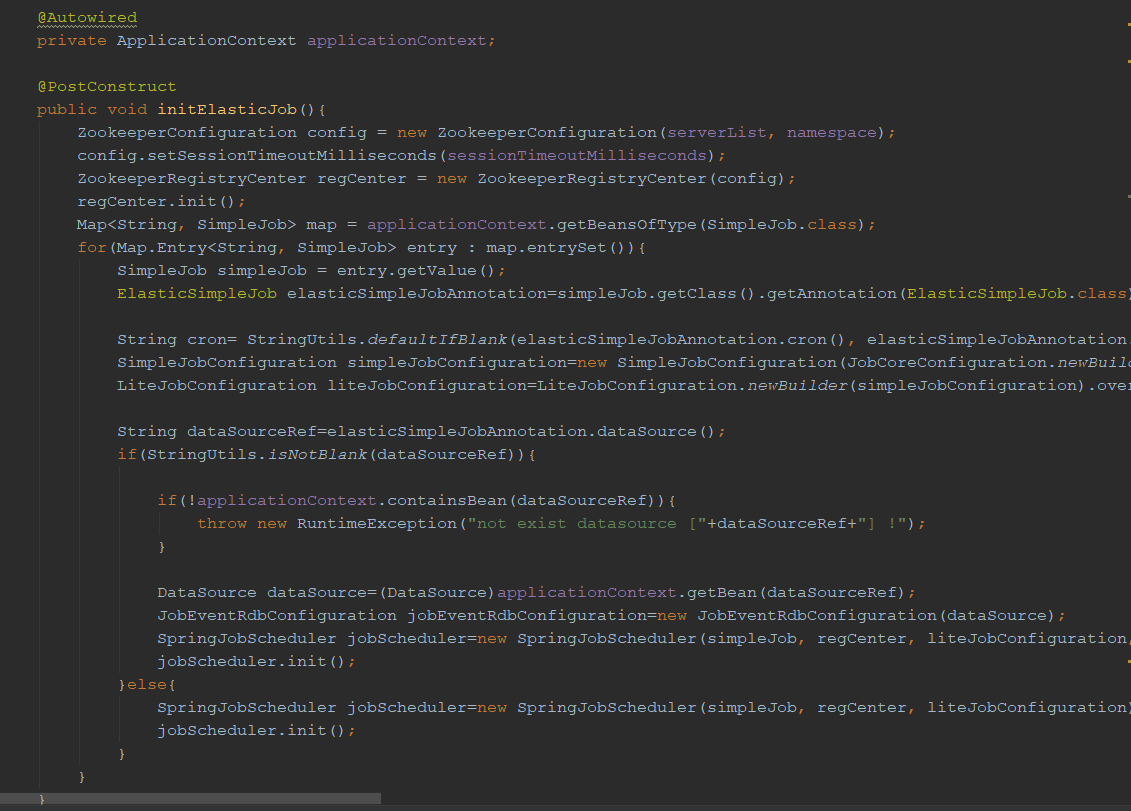




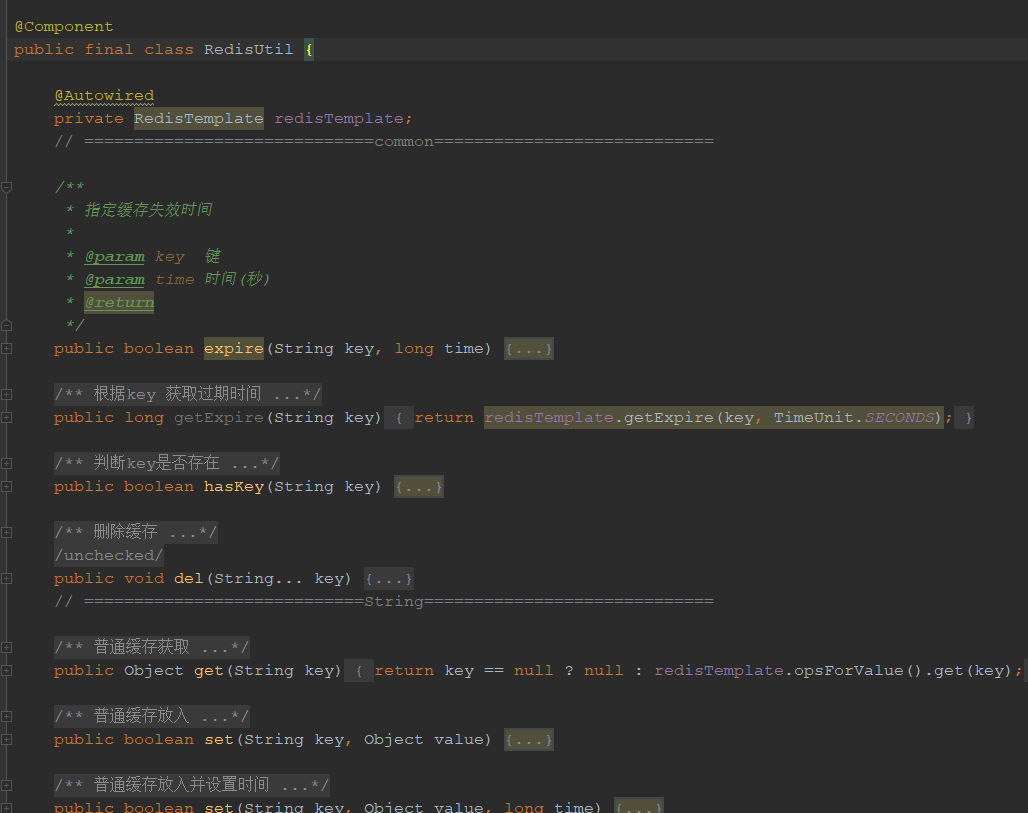


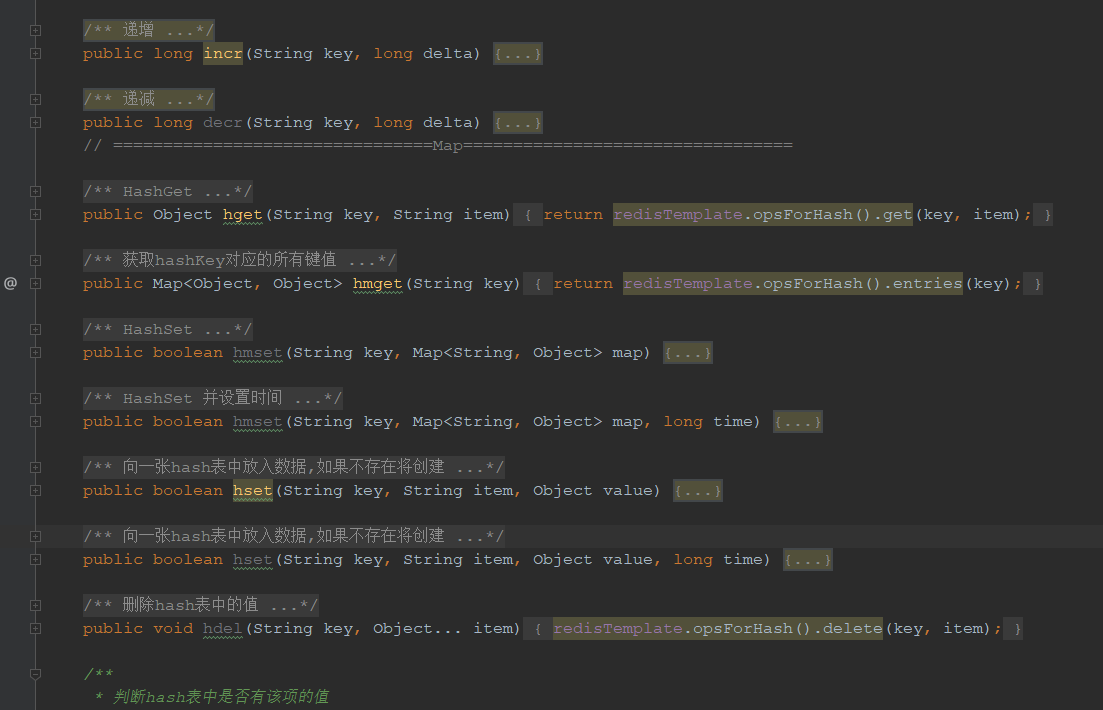
使用elasticJob作为调度器，每个一分钟进行扫描，将一分钟内将要开始的活动加载进redis。

初始化调度器

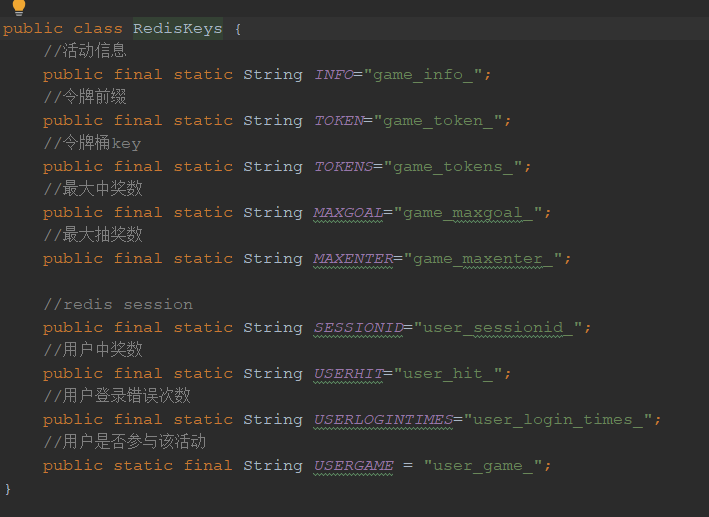


对redis调用方法的封装：

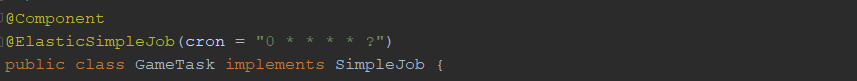


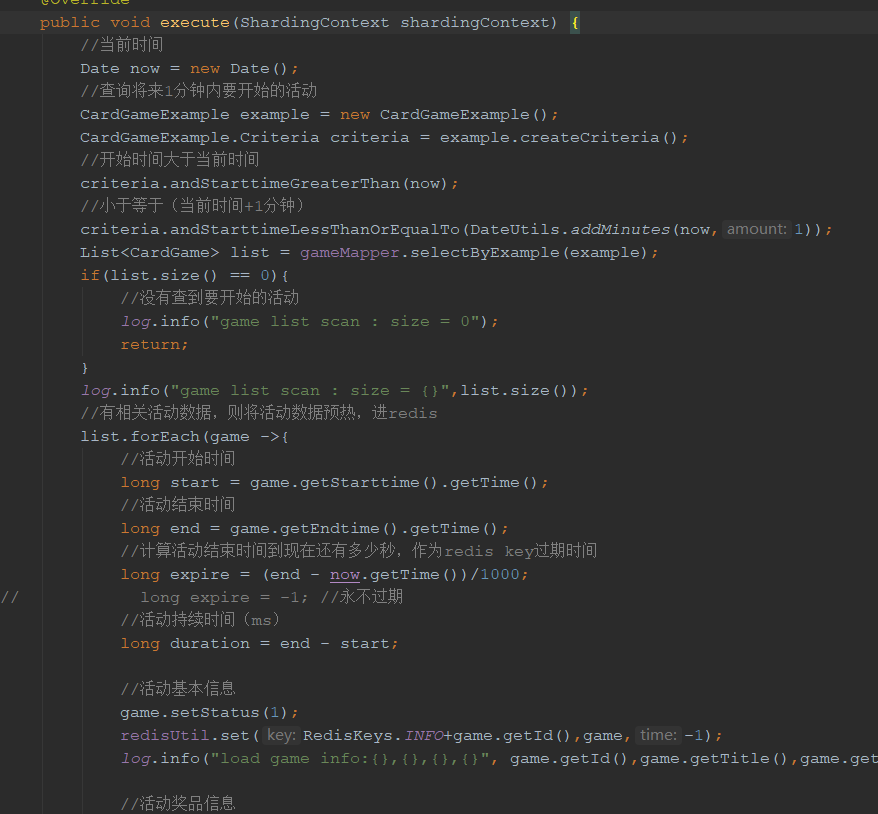


定义rediskey:

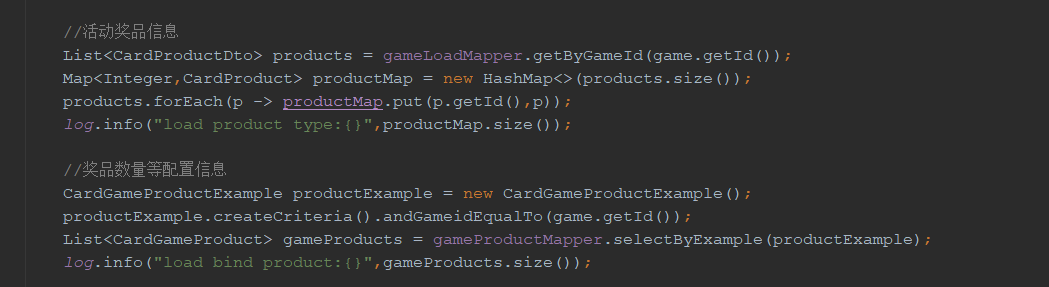


调度器里将一分钟内将要开始的活动的基本信息加载进redis。





加载活动的奖品信息



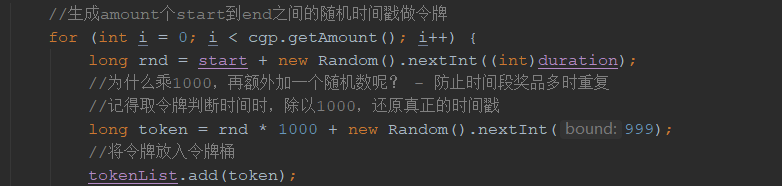
令牌生成策略：

1.创建一个Long类型的list，用于后续存放每一个令牌。

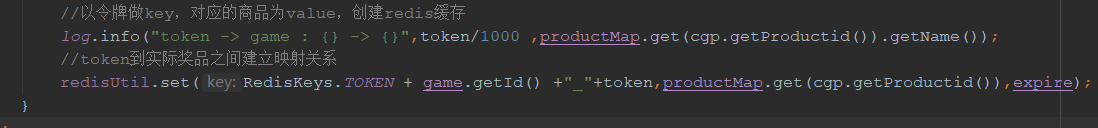
1637199104(1)

1. 生成与奖品数量对应个数的令牌，每个令牌是活动开始时间到活动

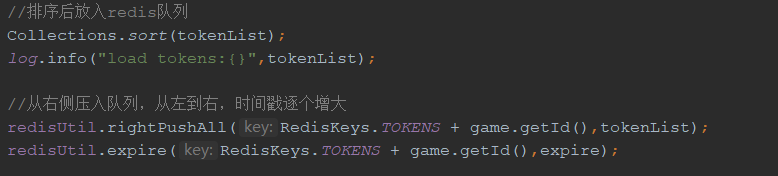
结束时间内的随机时间，并将令牌放入令牌桶。



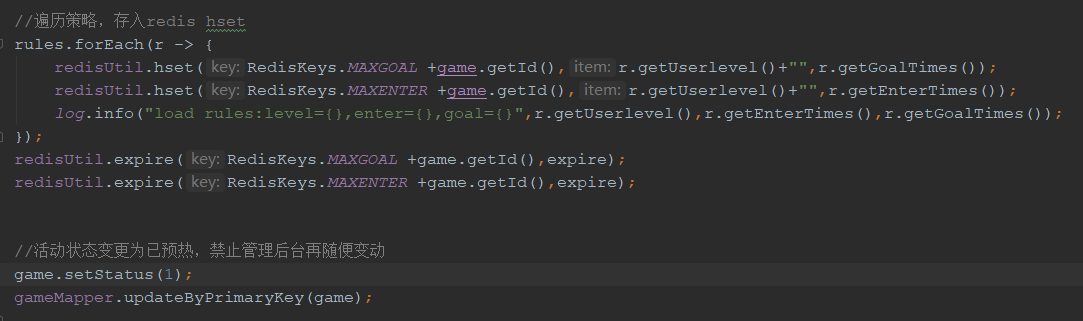
3.以令牌为key,对应商品为value,放入缓存中，设置过期时间。



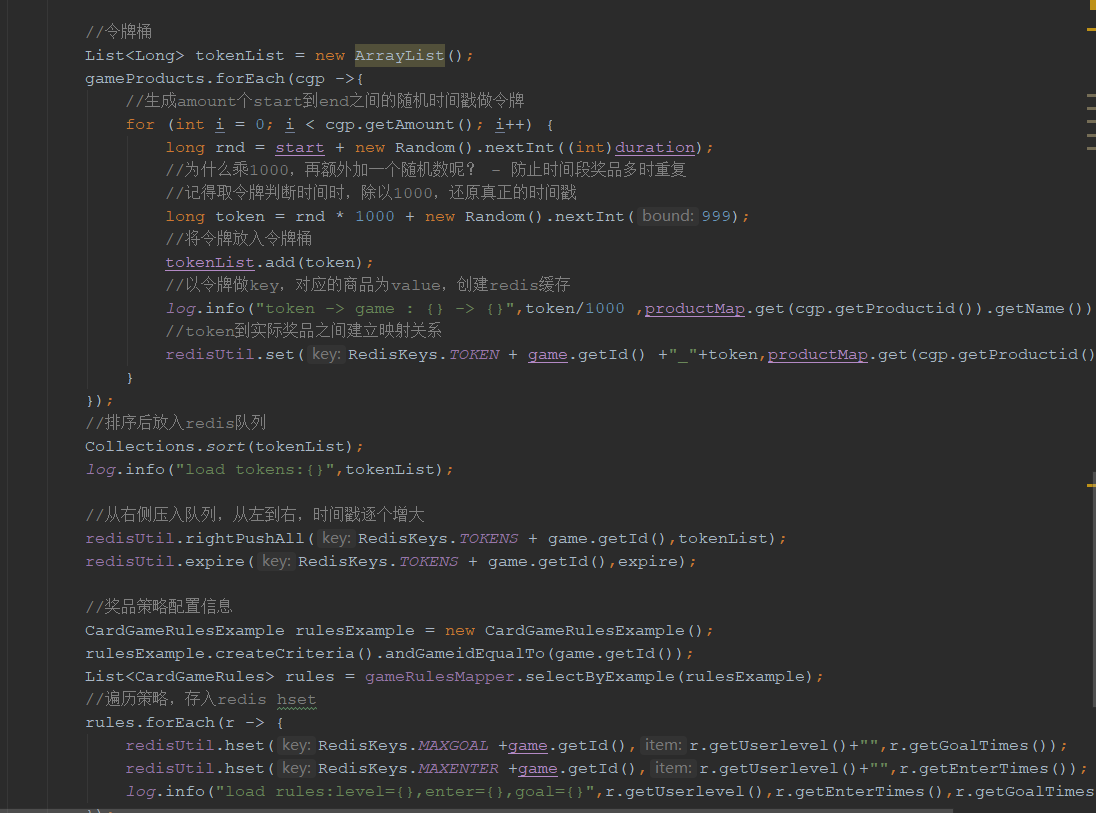
1. 将令牌桶中的令牌进行时间排序，并且从右侧压入redis，从左往右，时间戳逐个增大,并设置过期时间。

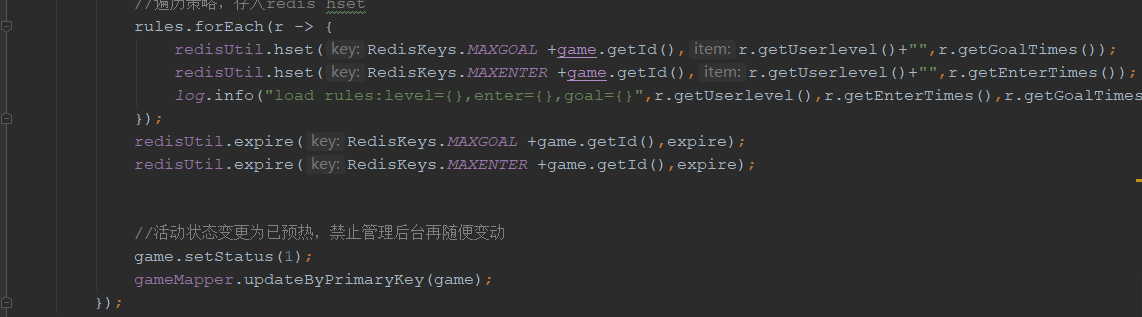


1. 将奖品策略配置信息(最大中奖数和最大抽奖数)，不同的用户等级，设置的中奖次数和抽奖次数都不同，存入redis，并设置过期时间。



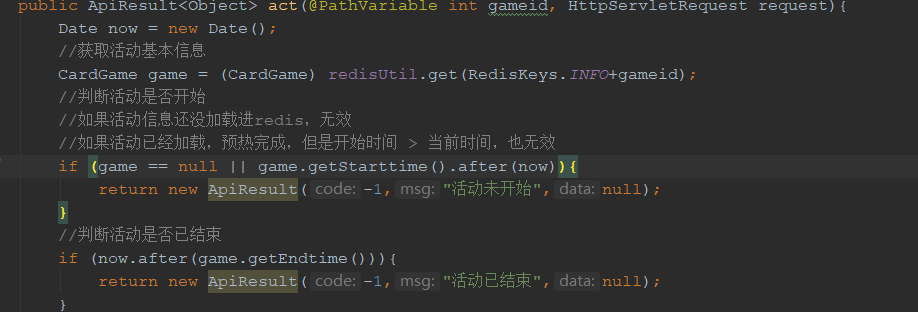
令牌桶完整代码



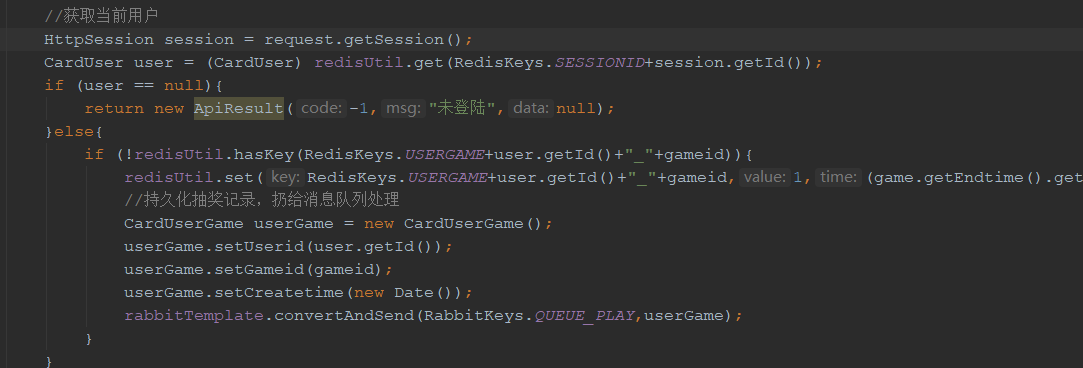


抽奖模块：

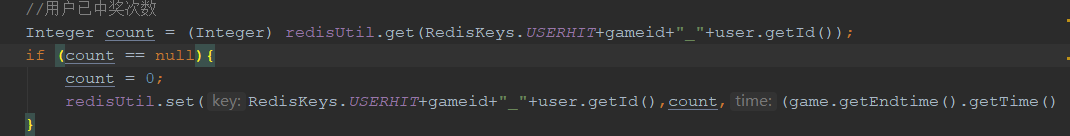
1. 抽奖业务的校验，判断活动是否正在进行：



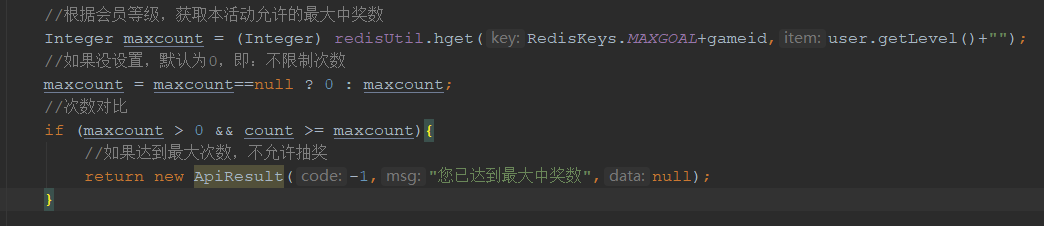
1. 判断用户是否登录，使用redis缓存用户信息，并判断用户是否是第一次参与该活动，如果不是则将用户id与活动id存入redis,并且将用于参与的抽奖记录发送到消息队列：



1. 将用户的已中奖次数存入缓存



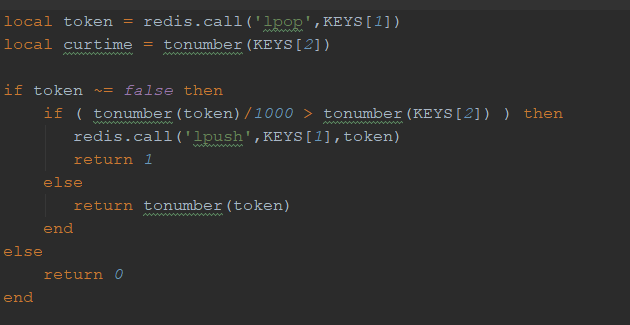
1. 判断该用户允许的最大中奖次数：



1. 判断完以后开始抽奖，判断是否还有奖品，并且判断该用户的抽奖时间大于该奖品时间戳，如果小于，则表示未到奖品发放时间，该用户未中奖：

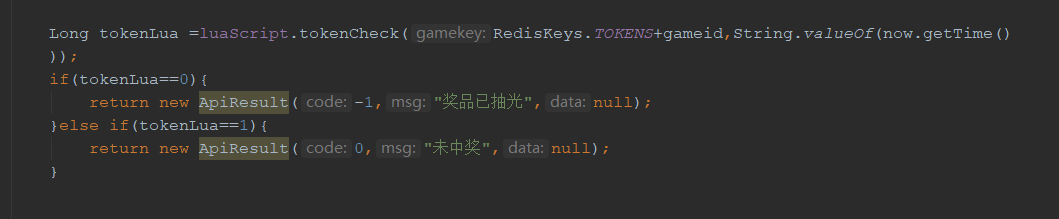


1. 在用户取完令牌以后，到未中奖，放回令牌的这段时间，为了防止下一位抽奖者取第二个令牌，将令牌给打乱了，这里时间lua脚本将取令牌和放令牌的整个过程打成原子操作。

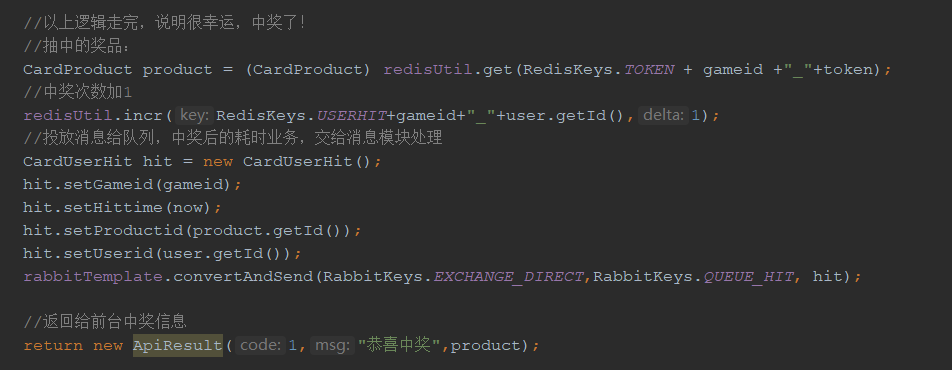




使用lua的方法，将取令牌和放令牌打包成原子操作

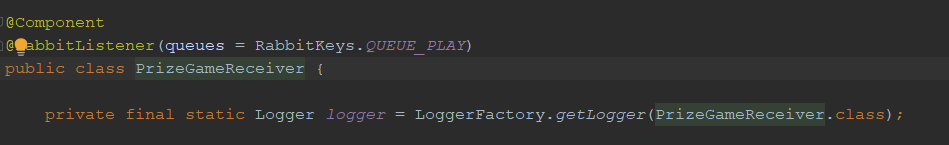


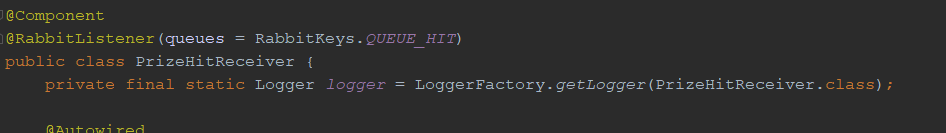
7.中奖以后，在redis中将该用户的中奖信息加1，并且将信息写入队列，交给消息队列处理。



1. 使用rabbitmq消息中心的消费，持久化到数据库中。

定义消费者监听特定队列





并将接收到的信息进行入库做持久化

