

java分布式面试接口如何保证幂等及概念理解

作者：有故事的人 来源：范文网 www.wtabcd.cn/fanwen/

本文原地址：<https://www.wtabcd.cn/fanwen/zuowen/c729bd06116ec1ec3aa2e326f311fdd1.html>

范文网，为你加油喝彩！

引言

稳定性设计第一篇：这一小节开始讲设计系统稳定性保证的相关设计，谁都不想自己负责的系统三天两头就出故障，也不想周六日跟女票葡萄美酒夜光杯的时候一个电话call去vpn办公，那么你就想办法让你的系统尽量稳定，我们的目标是让系统“无人值守”。

阿里新零售和阿里妈妈，美团，过去我面试这些公司都被问过接口幂等相关问题，接口幂等设计在分布式系统开发中非常常见且很重要，后来我自己做面试官也慢慢意识到幂等的重要性。

一些初学者对幂等这个概念完全不理解，更不知道如何设计，这在工作中很容易给自己惹麻烦，所以一定要会！一定要会！一定要会！

1、幂等的概念

面试官：

幂等的概念你了解吗，你设计的系统里有哪些接口使用到了幂等设计？

问题分析：

幂等的概念首先你肯定理解了，简单通俗易懂，就是无论你是 http 接口还是 rpc 接口，入参不变的情况下，无论请求多少次，结果都是一样的，请求结果不会因为请求次数不同而改变，没有副作用。

答：我参加工作的第一年，在某在线购票（电影票）app 的一家公司做后台系统开发，当时我负责积分系统，工作中接到这样一个线上活动需求。业务场景描述：用户每天使用 app 点击签到按钮参加活动，领取相应的积分，每个用户每天只能参加一次签到领积分活动，签到按钮在点击一次后会自设置灰色变为不可点击的状态，这个领积分的接口由我负责开发，提供 api 给客户端同事，上线后出现这样一个bug，当时没有完善的业务监控系统，功能上线后第二天问出于好奇系统里积分最高的人有多少积分，就在后台跑了一个sql，这一好奇，惊奇的发现有的用户积分高达几万，因为积分除了签到领取外，大多都是消费累计积分，一块钱才能累积一分，我表示怀疑，什么能人看电影能看几万块钱？

带着这个疑问，我查询了他的积分累积记录，发现大部分积分都是靠签到领积分获得的，按照活动规则，一个人一天只能参加一次签到，不可能有这么多的积分，而这个用户一天签到几百次，后来经过和前端一同检查bug发现问题所在，原因是签到按钮虽然变灰，但是请求的 url 没有在前端页面隐藏，用户通过技术手段绕过 button 变灰的前端限制重复刷新了接口，重复获得积分。

自制力事后问题分析：

这个bug最大问题还在我这里，因为我的接口没有做幂等设计，正确的逻辑应该是根据系统当前日期做幂等，幂等后无论用户发起多少次请求，最后的结果都是一样的，积分只累加一次。好在这个bug没有被黑产发现，只有几个用户发现损失可控。

因为我缺少设计经验，不懂幂等设计，领导也没提醒我，所以出现这种bug，经历更多和钱相关的系统开发后，我明白一个道理，任何系统设计，都要考虑业务的安全性，内部系统可以为了节省人力，适当简化设计，做到防君子不防小人，假设你的同事都是君子，对c端用户的系统，不光要防君子，还要防小人，风险防范不能全指望风控系统，有时bug可能会来自系统内部，比如用户并没有恶意盗刷之意，只是网络不好，用户等了两秒钟还没加载完就多点了几次签到按钮，我的接口没有做幂等设计，只要收到请求就会多给用户加积分，这个时候能怪用户吗？很显然是开发者的责任。

关于这个接口的幂等设计

我是这样解决的：

积分接口后台根据用户手机号 + userid + 系统当前日期拼接后生成唯一流水号，根据流水号后保存，如果用户重复发起请求，先根据唯一流水号校验在后台做校验，如果流水号存在直接返回上一次请求结果，考虑到并发的情况下，状态判断使用了锁处理。

开发业务监控系统，采用定时任务每天生成系统里 top100
积分增长最多名单，运营或者术人员每天观察有没有异常。

经过这次bug反思，学习到两点：

理解幂等设计的重要性，凡是和钱相关的功能请谨慎。

监控系统的重要性，这里的监控说的是业务类监控，如果那天我没有好奇系统里谁的积分最高，这个bug会什么时候发现？

面试官：嚯，有点意思，你还真的是写了个大bug，弄懂了吸取教训就好，可别进了我的项目组后拿我们的系统写这bug。

深入分析：

在编程中一个幂等操作的特点是其任意多次执行所产生的影响均与一次执行的影响相同。幂等函数，或幂等方法，是指可以使用相同参数重复执行，并能获得相同结果的函数。这些函数不会影响系统状态，也不用担心重复执行会对系统造成改变。例如，“settrue()”函数就是一个幂等函数，无论多次执行，其结果都是一样的。更复杂的操作幂等保证是利用唯一交易号(流水号)实现。

—— 百度百科

如果你了解 restful 风格接口，相信你对 get / post / delete 几个动词不陌生，在一次面试锤子科技的过程中，面试官问我是否了解 rest 接口，我balabala回答了这几常用的动词，面试官又问我：那你除了知道 get 是从服务器获取资源，还有别的理解吗？当时我没答上来，出了公司以后才想起，get 动作的设计应该是幂等的。同理 delete 也是幂等的，如果你设计的接口 get / delete 不是幂等的，那么你可能要重新思考一下了。

2、工作中常见的幂等设计场景

如果你做的功能和钱相关，或者是能还钱的，那么你就要小心了，每一个接口都要先考虑下是否需要幂等设计，下面是两种常见的需求场景。

发券/积分接口，通常通过 orderid userid 做幂等校验。

支付/退款接口，我们不希望用户发起多次支付都收到用户的钱，用户会投诉，还要把钱退还给用户，对系统还是客服人员来说都是无用功，支付系统非常复杂，想做好支付系统，还有很多东西需要学习，要考虑网络延迟，服务异常，订单中心回掉超时等各种不稳定的因素，通常采用前端控制，逻辑层状态的控制，数据层唯一索引的控制，以及分布式锁的控制，在幂等篇不过过多讨论。

3、幂等接口常见设计方案

客户端按钮提交限制，每次提交一个请求战狼2百度云网盘时，按钮置为不可用。

后台系统逻辑层处理，生成保存唯一id（流水号），每次请求先校验流水号是否已经存在，存在则表示重复操作，直接返回上一次操作结果。

token校验机制，客户端请求前先申请token，同一个token只处理一次，无token或者相同token不做处理。

分布式锁，如引入 redis 分布式锁，防止其他请求重复操作。

请求队列，引入 mq 排队的方式让请求有编年体断代史序处理，关于异步操作的应用会在后面的章节讲解。

每一种方案都有自己的优缺点，比如客服端按钮提交限制，实现简单，但是不能从根本上解决问题，后台生成唯一id，判断存在状态必须要保证原子操作，可以采用多种方案组合的方式解决幂等问题咸丰帝，我们的目标是，用最容易维护的方法解决问题。

总结

在过去的工作经历中，我招进来一个工作三年的同事，场景是开发一个退款接口，review代码的时候，我发现退款的功能是做完了，钱确实能退，但是并没有做幂等设计，我俩讨论了下，我说：如果同一个订单被请求了两次退款，那这钱是不是要退两次，这很危险呀？当时这个同事并没有意识到这一点，因为没有相关经验，连概念都不知道，作为一个三年qq空间视频经验的实在不应该，和钱相关的功能一定要慎重，做幂等设计就是为了系统能防君子，也要防小人。

以上就是java分布式面试接口如何保证幂等及概念的详细内容，更多关于java分布式面试接口幂等保证的资料请关注www.887551.com其它相关文章！

更多 作文 请访问 https://www.wtabcd.cn/fanwen/list/92_0.html

文章生成doc功能，由[范文网](http://www.wtabcd.cn/fanwen/)开发