



您的位置：[首页](#) → [网络编程](#) → [PHP编程](#) → [php实例](#) → [PHP Redis单据锁实现方法](#)

PHP实现Redis单据锁以及防止并发重复写入

更新时间：2018年04月10日 08:48:06 作者：leeon 我要评论

本篇文章给大家分享了PHP实现Redis单据锁以及如何防止并发重复写入的方法，对此有需要的朋友参考学习下。

一、写在前面：

在整个供应链系统中，会有很多种单据（采购单、入库单、到货单、运单等等），在涉及写单据数据的接口时（增删改操作），即使前端做了相关限制，还是有可能因为网络或异常操作产生并发重复调用的情况，导致对相同单据做相同的处理；

为了防止这种情况对系统造成异常影响，我们通过Redis实现了一个简单的单据锁，每个请求需先获取锁才能执行业务逻辑，执行结束后才会释放锁；保证了同一单据的并发重复操作请求只有一个请求可以获取到锁（依赖Redis的单线程），是一种悲观锁的设计；

注：Redis锁在我们的系统中一般只用于解决并发重复请求的情况，对于非并发的重复请求一般会去数据库或日志校验数据的状态，两种机制结合起来才能保证整个链路的可靠。

二、加锁机制：

主要依赖Redis setnx指令实现：

SETNX key value

将 key 的值设为 value ，当且仅当 key 不存在。

若给定的 key 已经存在，则 SETNX 不做任何动作。

SETNX 是『SET if Not eXists』（如果不存在，则 SET）的简写。

但使用setnx有一个问题，即setnx指令不支持设置过期时间，需要使用expire指令另行为key设置超时时间，这样整个加锁操作就不是一个原子性操作，有可能存在setnx加锁成功，但因程序异常退出导致未成功设置超时时间，在不及解锁的情况下，有可能会导致死锁（即使业务场景中不会出现死锁，无用的key一直常驻内存也不是很好的设计）；

这种情况可以使用Redis事务解决，把setnx与expire两条指令作为一个原子性操作执行，但这样做相对而言会比较麻烦，好在Redis 2.6.12之后版本，Redis set指令支持了nx、ex模式，并支持原子化地设置过期时间：

SET key value [EX seconds] [PX milliseconds] [NX|XX]

将字符串值 value 关联到 key 。

如果 key 已经持有其他值，SET 就覆写旧值，无视类型。

对于某个原本带有生存时间（TTL）的键来说，当 SET 命令成功在这个键上执行时，这个键原有的 TTL 将被清除。

可选参数

从 Redis 2.6.12 版本开始，SET 命令的行为可以通过一系列参数来修改：

- EX second：设置键的过期时间为 second 秒。SET key value EX second 效果等同于 SETEX key second value 。
- PX millisecond：设置键的过期时间为 millisecond 毫秒。SET key value PX millisecond 效果等同于 PSETEX key millisecond value 。
- NX：只在键不存在时，才对键进行设置操作。SET key value NX 效果等同于 SETNX key value 。
- XX：只在键已经存在时，才对键进行设置操作。

因为 SET 命令可以通过参数来实现和 SETNX 、SETEX 和 PSETEX 三个命令的效果，所以将来的 Redis 版本可能会废弃并最终移除 SETNX 、SETEX 和 PSETEX 这三个命令。

三、加锁实现（完整测试代码会贴在最后）：

```
1  /**
2   * 加单据锁
3   * @param int $intOrderId 单据ID
4   * @param int $intExpireTime 锁过期时间（秒）
5   * @return bool|int 加锁成功返回唯一锁ID，加锁失败返回false
6   */
7  public static function addLock($intOrderId, $intExpireTime = self::REDIS_LOCK
8  {
9      //参数校验
10     if (empty($intOrderId) || $intExpireTime <= 0) {
11         return false;
```

请输入关键词

站长推荐

正版 Windows 10
正版Windows 10 家庭/专
作系统限时抢购[¥4988→

站长推荐

正版 Office 软件
Microsoft Office 2016/20
正版最低价仅需[¥148元

大家感兴趣的内容

- 1 php获取数组长度的方法（有实
- 2 微信公众平台实现获取用户Op
- 3 教你如何使用php session
- 4 使用PHP生成二维码的两种方
- 5 php发送get、post请求的6种方
- 6 PHP中把stdClass Object转ar
- 7 微信公众平台网页授权获取用
- 8 Laravel框架数据库CURD操作
- 9 PHP字符串的连接简单实例
- 10 PHP删除数组中空值的方法介绍

最近更新的内容

记录Yii2框架开发微信公众号遇到
php for 循环使用的简单实例
php实现微信发红包
php生成随机数或者字符串的代码
php 伪造HTTP_REFERER页面U
ThinkPHP快速入门实例教程之数
PHP获取网页标题的3种实现方法
php微信公众号开发之快速查询
php读取qqwry.dat ip地址定位文
php实现微信支付之现金红包

常用在线小工具

```
12     }
13
14     //获取Redis连接
15     $objRedisConn = self::getRedisConn();
16
17     //生成唯一锁ID，解锁需持有此ID
18     $intUniqueLockId = self::generateUniqueLockId();
19
20     //根据模板，结合单据ID，生成唯一Redis key（一般来说，单据ID在业务系统中唯一的）
21     $strKey = sprintf(self::REDIS_LOCK_KEY_TEMPLATE, $intOrderId);
22
23     //加锁（通过Redis setnx指令实现，从Redis 2.6.12开始，通过set指令可选参数也可以实现set
24     $bolRes = $objRedisConn->set($strKey, $intUniqueLockId, ['nx', 'ex'=>$intExp
25
26     //加锁成功返回锁ID，加锁失败返回false
27     return $bolRes ? $intUniqueLockId : $bolRes;
28 }
```

四、解锁机制：

解锁即比对加锁时的唯一lock id，如果比对成功，则删除key；需要注意的是，解锁整个过程中同样需要保证原子性，这里依赖redis的watch与事务实现：

WATCH命令可以监控一个或多个键，一旦其中有一个键被修改（或删除），之后的事务就不会执行。监控一直持续到EXEC命令（事务中的命令是在EXEC之后才执行的，所以在MULTI命令后可以修改WATCH监控的键值）

五、解锁实现（完整测试代码会贴在最后）：

```
1  /**
2   * 解单据锁
3   * @param int $intOrderId 单据ID
4   * @param int $intLockId 锁唯一ID
5   * @return bool
6   */
7  public static function releaseLock($intOrderId, $intLockId)
8  {
9      //参数校验
10     if (empty($intOrderId) || empty($intLockId)) {
11         return false;
12     }
13
14     //获取Redis连接
15     $objRedisConn = self::getRedisConn();
16
17     //生成Redis key
18     $strKey = sprintf(self::REDIS_LOCK_KEY_TEMPLATE, $intOrderId);
19
20     //监听Redis key防止在【比对lock id】与【解锁事务执行过程中】被修改或删除，提交事务后会
21     $objRedisConn->watch($strKey);
22     if ($intLockId == $objRedisConn->get($strKey)) {
23         $objRedisConn->multi()->del($strKey)->exec();
24         return true;
25     }
26     $objRedisConn->unwatch();
27     return false;
28 }
```

六、附整体测试代码（此代码仅为简易版本）

```
1  <?php
2
3  /**
4   * Class Lock_Service 单据锁服务
5   */
6  class Lock_Service
7  {
8      /**
9       * 单据锁redis key模板
10      */
11     const REDIS_LOCK_KEY_TEMPLATE = 'order_lock_%s';
12
13     /**
14      * 单据锁默认超时时间（秒）
15      */
16     const REDIS_LOCK_DEFAULT_EXPIRE_TIME = 86400;
17
18     /**
19      * 加单据锁
20      * @param int $intOrderId 单据ID
21      * @param int $intExpireTime 锁过期时间（秒）
22      * @return bool|int 加锁成功返回唯一锁ID，加锁失败返回false
23      */
24     public static function addLock($intOrderId, $intExpireTime = self::REDIS_IC
25     {
26         //参数校验
27         if (empty($intOrderId) || $intExpireTime <= 0) {
28             return false;
29         }
30
31         //获取Redis连接
32         $objRedisConn = self::getRedisConn();
33
34         //生成唯一锁ID，解锁需持有此ID
35         $intUniqueLockId = self::generateUniqueLockId();
36
37         //根据模板，结合单据ID，生成唯一Redis key（一般来说，单据ID在业务系统中唯一的）
38         $strKey = sprintf(self::REDIS_LOCK_KEY_TEMPLATE, $intOrderId);
39
40         //加锁（通过Redis setnx指令实现，从Redis 2.6.12开始，通过set指令可选参数也可以实现s
41         $bolRes = $objRedisConn->set($strKey, $intUniqueLockId, ['nx', 'ex'=>$intE
42
43         //加锁成功返回锁ID，加锁失败返回false
```








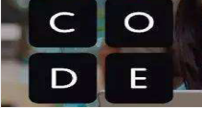


```
44     return $bolRes ? $intUniqueLockId : $bolRes;
45 }
46
47 /**
48  * 解单据锁
49  * @param int $intOrderId 单据ID
50  * @param int $intLockId 锁唯一ID
51  * @return bool
52  */
53 public static function releaseLock($intOrderId, $intLockId)
54 {
55     //参数校验
56     if (empty($intOrderId) || empty($intLockId)) {
57         return false;
58     }
59
60     //获取Redis连接
61     $objRedisConn = self::getRedisConn();
62
63     //生成Redis key
64     $strKey = sprintf(self::REDIS_LOCK_KEY_TEMPLATE, $intOrderId);
65
66     //监听Redis key防止在【比对lock id】与【解锁事务执行过程中】被修改或删除，提交事务后$
67     $objRedisConn->watch($strKey);
68     if ($intLockId == $objRedisConn->get($strKey)) {
69         $objRedisConn->multi()->del($strKey)->exec();
70         return true;
71     }
72     $objRedisConn->unwatch();
73     return false;
74 }
75
76 /**
77  * Redis配置: IP
78  */
79 const REDIS_CONFIG_HOST = '127.0.0.1';
80
81 /**
82  * Redis配置: 端口
83  */
84 const REDIS_CONFIG_PORT = 6379;
85
86 /**
87  * 获取Redis连接 (简易版本, 可用单例实现)
88  * @param string $strIp IP
89  * @param int $intPort 端口
90  * @return object Redis连接
91  */
92 public static function getRedisConn($strIp = self::REDIS_CONFIG_HOST, $intP
93 {
94     $objRedis = new Redis();
95     $objRedis->connect($strIp, $intPort);
96     return $objRedis;
97 }
98
99 /**
100  * 用于生成唯一的锁ID的redis key
101  */
102 const REDIS_LOCK_UNIQUE_ID_KEY = 'lock_unique_id';
103
104 /**
105  * 生成锁唯一ID (通过Redis incr指令实现简易版本, 可结合日期、时间戳、取余、字符串填充、
106  * @return mixed
107  */
108 public static function generateUniqueLockId()
109 {
110     return self::getRedisConn()->incr(self::REDIS_LOCK_UNIQUE_ID_KEY);
111 }
112 }
113
114 //test
115 $res1 = Lock_Service::addLock('666666');
116 var_dump($res1);//返回lock id, 加锁成功
117 $res2 = Lock_Service::addLock('666666');
118 var_dump($res2);//false, 加锁失败
119 $res3 = Lock_Service::releaseLock('666666', $res1);
120 var_dump($res3);//true, 解锁成功
121 $res4 = Lock_Service::releaseLock('666666', $res1);
122 var_dump($res4);//false, 解锁失败
```

以上就是本次给大家整理的全部内容，感谢大家对脚本之家的支持。

您可能感兴趣的文章:

- [php使用redis的有序集合zset实现延迟队列应用示例](#)
- [30个php操作redis常用方法代码例子](#)
- [php操作redis中的hash和zset类型数据的方法和代码例子](#)
- [php结合redis实现高并发下的抢购、秒杀功能的实例](#)
- [PHP操作Redis常用技巧总结](#)
- [php操作redis缓存方法分享](#)
- [redis安装、配置、使用和redis php扩展安装教程](#)
- [php中使用redis队列操作实例代码](#)
- [php使用redis的几种常见操作方式和用法示例](#)

相关文章

	<div>php layui实现前端多图上传实例</div> <div>在本篇文章里小编给大家整理的是关于php结合layui前端实现多图上传的实例内容，有需要的朋友们可以参考下。</div> <div>2019-07-07</div>
	<div>PHP实现递归目录的5种方法</div> <div>本篇文章主要介绍了PHP实现递归目录的5种方法，主要是利用一些循环来实现的，感兴趣的小伙伴们可以参考一下。</div> <div>2016-10-10</div>
	<div>PHP中Laravel 关联查询返回错误id的解决方法</div> <div>这篇文章主要介绍了Laravel 关联查询返回错误id的解决方法，非常不错，具有参考借鉴价值，需要的朋友参考下吧</div> <div>2017-04-04</div>
	<div>PHP微信刮刮卡 附微信接口</div> <div>像微信大转盘、刮刮卡这样的抽奖，最重要最核心的就是如何设置中奖的概率。本文综合了网上多个微信大转盘、刮刮卡的代码</div> <div>2016-07-07</div>
	<div>Yii2框架控制器、路由、Url生成操作示例</div> <div>这篇文章主要介绍了Yii2框架控制器、路由、Url生成操作,结合实例形式分析了Yii2框架控制器、路由、url生成跳转等相关原理及操作技巧,需要的朋友可以参考下</div> <div>2019-05-05</div>
	<div>php简单的留言板与回复功能具体实现</div> <div>留言板是在刚接触php时用来学习的一个简单的应用例子了，今天我再给初学php的朋友提供一个完整的php留言板的全部制作过程，希望对你会有帮助</div> <div>2014-02-02</div>
	<div>PHP环境搭建(PHP+Apache+mysql)</div> <div>这篇文章主要为大家详细介绍了PHP环境搭建，包括php、Apache、mysql环境安装,具有一定的参考价值，感兴趣的小伙伴们可以参考一下</div> <div>2016-11-11</div>
	<div>php 模拟 asp.net webFrom 按钮提交事件实例</div> <div>需要php方面的项目开发，php刚刚入门，在写按钮提交过程中，asp.net里的按钮事件更好些，下面有个实例，大家可以看看</div> <div>2014-10-10</div>
	<div>Symfony学习十分钟入门经典教程</div> <div>这篇文章主要介绍了Symfony学习十分钟入门教程,详细介绍了Symfony的安装配置,项目初始化,建立Bundle,设计实体,添加约束,增删改查等基本操作技巧,需要的朋友可以参考下</div> <div>2016-02-02</div>
	<div>PHP 获取远程网页内容的代码(fopen,curl已测)</div> <div>获取远程网页内容的php代码，做小偷采集程序的程序经常用得到，现在用curl的比较多。</div> <div>2011-06-06</div>
<div>最新评论</div>	