

request(Uid,SeckillId)



checker.askForEntry(uid, id)

第一层检查，同一个用户同一个商品还在秒杀过程中还没结束，不允许再进入



checkSecKillProductConditons)

第二层检查，判断秒杀的产品是否到时间？过期？秒杀库存不足？



CheckUserCondition

第三层检查，判断用户的积分是否足够，这边会产生并发风险，检查用户是否已经参加过此次商品的秒杀，通过用户的积分使用详情表查看

获取SeckillProducLock,进入秒杀阶段



SeckillProducLock

再次检查剩余秒杀库存，是否充足在进行秒杀，不同用户线程对同一个商品秒杀引起的库存问题，所以需要锁机制

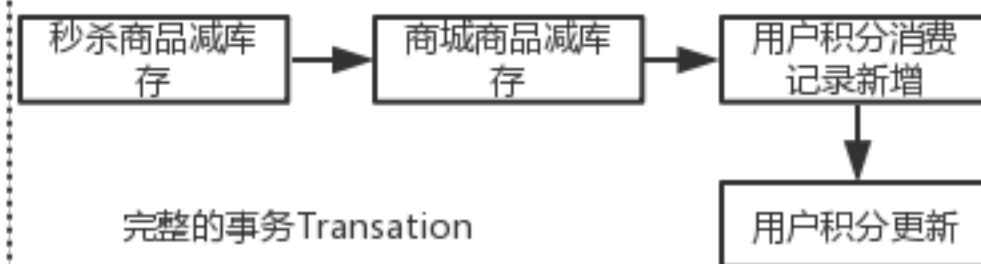
获取ProducLock



ProducLock

再次检查商品总库存，是否充足在进行秒杀，不同用户线程对同一个商品秒杀，但是最终如果商城总商品库存不足，则不能秒杀，会对商品进行竞争

执行换货操作成功



秒杀成功结束

同一个用户，同一次只能秒杀一个商品，且只能秒杀一个商品，userLock可以限制同一个用户在不同的客户端中同时执行积分兑换的操作和秒杀使用Synchronized（对象锁）