目录

---

1. 库存预加载到Redis中是怎么实现的?

我是通过实现 `InitializingBean接口` ，重写其中 `afterPropertiesSet()方法` ，实现的预加载

1.1 之后主动添加秒杀商品的话，怎么添加?

通过后台管理进行添加，修改redis缓存和数据库中的值

---

2. 在Redis中扣减库存的时候，是怎么保证线程安全，防止超卖的?

redis中有一个 `decr()` 方法，它实现的是递减操作，而且能够 保证原子性

---

3. 如果出现Redis缓存雪崩、穿透，怎么解决?

雪崩就是缓存中我存储的值全部都失效了，请求直接打到数据库上，请求过大，数据库扛不住。可以用设置这些热点数据永不失效，或者是设置一个随机的过期时间，这样来避免它同时失效。

缓存穿透是缓存和数据库中都没有的数据，如果有人利用这些数据高并发的访问的话，对数据库压力也很大。可以对数据比如它的id值进行一个校验，避免这些不存在的值对数据库进行访问或者是使用布隆过滤器，它的原理是通过高效的数据结构查询数据库中是否存在这个值，不存在的时候，就直接返回，存在的话才会访问到数据库。

---

4. 限流防刷是怎么实现的?

限流防刷我是通过拦截器来实现的，我自定义了一个注解，它实现的功能就是标记在方法上，规定它单位时间内的访问次数，如果超过要求的话，就会被拦截。

拦截器我是继承的HandlerInterceptorAdapter，重写的是preHandle方法，在该方法中，将访问次数同步到Redis中，这个键值对是存在有效期的。最后还要把拦截器配置到项目中，继承WebMvcConfigurerAdapter，重写 `addInterceptors()` 方法

---

5. 对于用户的恶意下单，他知道了你的URL地址，不停的刷，怎么办?

我是通过隐藏URL地址来避免这种问题的，当访问秒杀接口的时候，会先从后端生成一个随机的字符串，然后保存到redis中，并且拼接到URL地址上，这样再去访问秒杀的接口，通过RestFul风格的地址，获取其中的随机字符串，与redis中的进行比对，一致的话，才能继续向下访问

---

6. 秒杀成功后是怎么同步到数据库中的?

通过两步，一步是减少商品库存，第二步是创建秒杀订单。

6.1 减库存成功，创建秒杀订单失败了怎么办?

这两步过程在一个事务中执行，然后先减少库存，它有一个成功的标志，减少库存成功了，才去执行创建订单的操作

6.2 Spring默认的事务隔离级别

默认情况下Spring使用的是数据库设置的默认隔离级别，应该是 可重复读

---

7. RabbitMQ怎么提高消息的高可用?

我在创建队列实例的时候，将其创建为可持久化的，它有一个durable属性设置为true，这样，RabbitMQ服务重启的情况下，也不会丢失消息。

---

8. 说说volatile关键字儿

它最重要的一点就是保证了变量的可见性。我想先说说JMM（java内存模型），每个线程有自己的工作内存，另外还存在一个主内存，线程从主内存中获取值存储在自己的工作内存中，当对变量进行修改，它不会立即将其同步到主内中，这个时候若有其他线程来从主内存中获取该变量的时候，就会发生脏读的现象，若被volatile标记的话，就能保证变量的可见性，当变量被修改的时候他就会将其立即同步到主内存中。

---

9. TCP和UDP的区别

1. TCP是需要通过三次握手建立连接的；UDP是无连接的
2. TCP提供的可靠性高；UDP的不保证可靠性，一般用于直播或者是语音通话
3. TCP是基于字节流的传输层协议，它比较慢；UDP比较快

---

10. ArrayList

- 底层是数组，查询快，增删慢
- 它的默认大小是10，添加值的时候会先对当前数组大小和总大小进行判断，若出现超过最大容量的话，就要进行扩容，扩容的大小是原来大小的1.5倍（右移运算符，右移1位），再将之前的数据复制到新的数组里边。