**缓存优化**

# 数据库问题

数据库的压测显示，数据库的并发能力有限，优化的空间不大，毕竟是IO操作，速度天生就是劣势，是扛不住上万的QPS并发的，能不能从架构的层面想办法解决呢？？

目的：让请求尽量少的到达数据库，减少数据库访问压力。

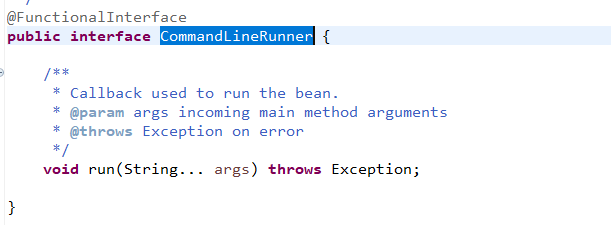
如何做呢？？？哪些业务可以用缓存？？

秒杀商品库存查询量很大，可以缓存。

秒杀前做缓存预热！！

# 缓存预热

## 启动



## Redis

|  |
| --- |
| **package** com.cjl.skill.config;  **import** org.springframework.context.annotation.Bean;  **import** org.springframework.context.annotation.Configuration;  **import** org.springframework.data.redis.connection.RedisConnectionFactory;  **import** org.springframework.data.redis.core.RedisTemplate;  **import** org.springframework.data.redis.core.StringRedisTemplate;  **import** org.springframework.data.redis.serializer.Jackson2JsonRedisSerializer;  **import** org.springframework.data.redis.serializer.StringRedisSerializer;  @Configuration  **public** **class** RedisConfig {  @Bean  **public** RedisTemplate<String,Object> redisTemplateJDK(RedisConnectionFactory factory){  RedisTemplate<String, Object> template = **new** RedisTemplate<>();  //配置连接工厂  template.setConnectionFactory(factory);  **return** template;  }    @Bean  **public** RedisTemplate<String,Object> redisTemplateJackson(RedisConnectionFactory factory){  RedisTemplate<String, Object> template = **new** RedisTemplate<>();  //配置连接工厂  template.setConnectionFactory(factory);  //使用StringRedisSerializer来序列化和反序列化redis的key值  template.setKeySerializer(**new** StringRedisSerializer());    //使用Jackson2JsonRedisSerializer来序列化和反序列化redis的value值（默认使用JDK的序列化方式）  template.setValueSerializer(**new** Jackson2JsonRedisSerializer<Object>(Object.**class**));    **return** template;  }    @Bean  **public** StringRedisTemplate stringRedisTemplate(RedisConnectionFactory factory){  **return** **new** StringRedisTemplate(factory);  }  } |

# 缓存预减

## 并发问题

有吗？？还是没有？？

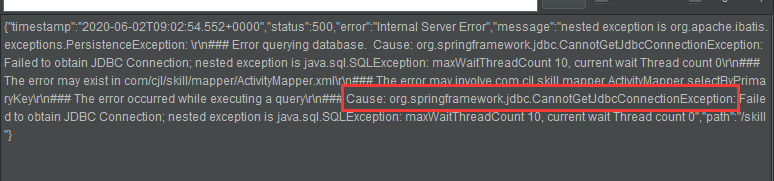
## 异常退单

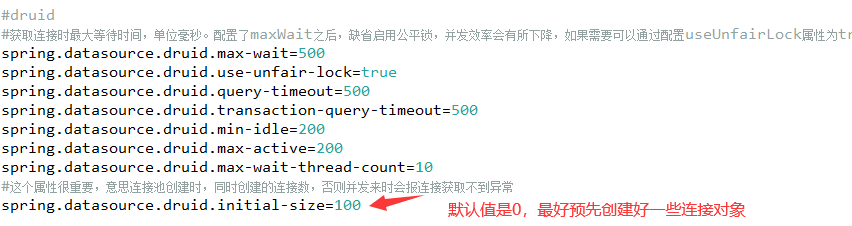
如何保证退单的库存还原正确？？

# 压测

## 连接预热

出现数据库连接池错误





**否则首次访问响应迟缓现象，甚至报错！！！！！**

Redis的连接池也是一样的

