

详情

返回 芋道快速开发平台 Boot + Cloud



芋道源码

2023-08-09 08:03

【组件设计】基于 Redis 实现高性能、低延迟的延时消息的方案演进

目录：

- 1、业务场景
- 2、技术方案
- 3、方案一：过期事件监听
- 4、方案二：zset
- 5、方案三：Redisson
- 6、编程练习 for 面试
- 7、拓展资料

作者：尼布斯  
审稿：芋芳










1、业务场景

延时消息来源于早期计算机对异步处理的需求，随着互联网的发展，越来越多的业务场景需要使用延队列。比如：


任务调度：延时队列可以用于任务调度，将需要在未来某个特定时刻执行的任务放入队列中。...

【组件设计】基于 Redis 实现高性能、低延迟的延时消息的方案演进

 架构设计



等17人赞过



芋道源码：

置顶

如果放在 ruoyi-vue-pro 项目里，会考虑使用 redisson 实现延迟消息。

问: ruoyi-vue-pro 如果使用 rocketmq, 会不会使用 rocketmq 还是 redisson 作为延迟消息

答: rocketmq 不支持任意刻度, 所以基于它做延迟消息需要改造。

1. 如果 rocketmq 的时间刻度可以满足, 就使用 rocketmq
2. 如果无法满足, 那就先发至 redisson, 之后可以直接延迟消费, 或者在转发到 rocketmq 再消费

ruoyi-vue-pro 这些业务适, 可以考虑接入延迟消息

1. 商城-订单: 支付超时、评价超时登封
2. 支付-回调: 基于延迟消息, 实现多轮时间间隔的回调
3. 短信: 支持发送指定时间的短信。例如说, 一些短信不要半夜发给用户, 而是早上 9 点

2023-08-09 09:30



大刘哥: 这就完了? 怎么没有结合 ruoyi pro 进行讲解应用实战部分呢?

2023-08-09 08:07

芋道源码 回复 大刘哥: 我补了一条评论哈

2023-08-09 09:30



昔年: 生产端

```
RDelayedQueue<String> delayedQueue =  
redissonClient.getDelayedQueue(redissonClient.getQueue("myBlockingQueue"));  
delayedQueue.offer("Redisson Delayed Item 1", 1, TimeUnit.MINUTES);  
delayedQueue.offer("Redisson Delayed Item 2", 2, TimeUnit.MINUTES);
```

消费端

```
import org.redisson.api.RBlockingQueue;  
  
import org.redisson.api.RedissonClient;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.boot.ApplicationArguments;  
import org.springframework.boot.ApplicationRunner;  
import org.springframework.stereotype.Component;  
  
@Component  
public class QueueListenerInitializer implements ApplicationRunner {  
  
    @Autowired  
    private RedissonClient redissonClient;  
  
    @Override  
    public void run(ApplicationArguments args) throws Exception {  
        RBlockingQueue<String> blockingQueue =  
            redissonClient.getBlockingQueue("myBlockingQueue");  
  
        while (true) {  
            try {  
                String item = blockingQueue.take();  

```

```
    } catch (InterruptedException e) {  
        Thread.currentThread().interrupt();  
        break;  
    }
```

2023-08-09 09:41

回复

芋道源码 回复 昔年：写的不错呀

2023-08-09 22:06

昔年 回复 芋道源码：😊

2023-08-09 22:29



十年砍柴：期待大佬实际应用在pro项目上

2023-08-16 23:15

芋道源码 回复 十年砍柴：好滴哦

2023-08-17 14:04