

# 第五章 数据一致性处理

欧阳修



# 本章导学

全网一手不加密IT课程，需要请加微信zzj97666九章来offer都有

使用延迟消息队列关闭超时订单，解决数据一致性问题

使用延迟消息关闭超时订单



订单页面的开发与测试



限购功能的实现



付款库存扣减数据一致性处理



# 1. 使用延迟消息关闭超时订单

全网一手不加密IT课程，需要请加微信zzj97666九章来offer都有

延迟消息 (Delay Message) 的配置与处理

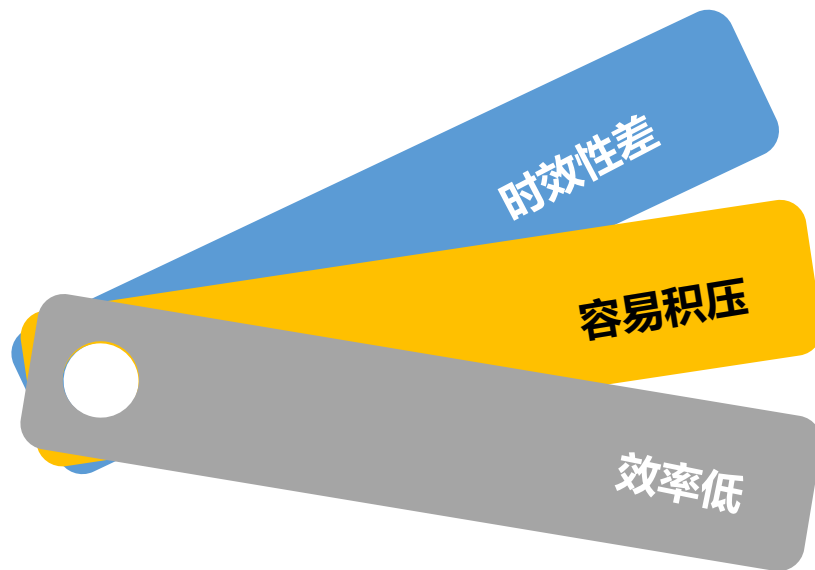
# 如何处理超时的任务

你能想到哪些可用的方案?



全网一手不加密IT课程，需要请加微信zzj97666九章来offer都有

### 轮询方案的特点



如果每分钟轮询一次，那么订单取消的最大误差就有一分钟。



如果一分钟之内有大量数据，但是一分钟内没处理完，那么下一分钟的就会顺延。



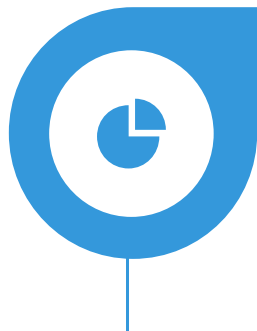
需要遍历资源，不适合拥有大量数据的项目。

## 1.2 方案对比



### 性能可靠

异常退出不丢数据



### 时效性高

时间精度高



### 低消耗

不需要遍历，系统资源低消耗

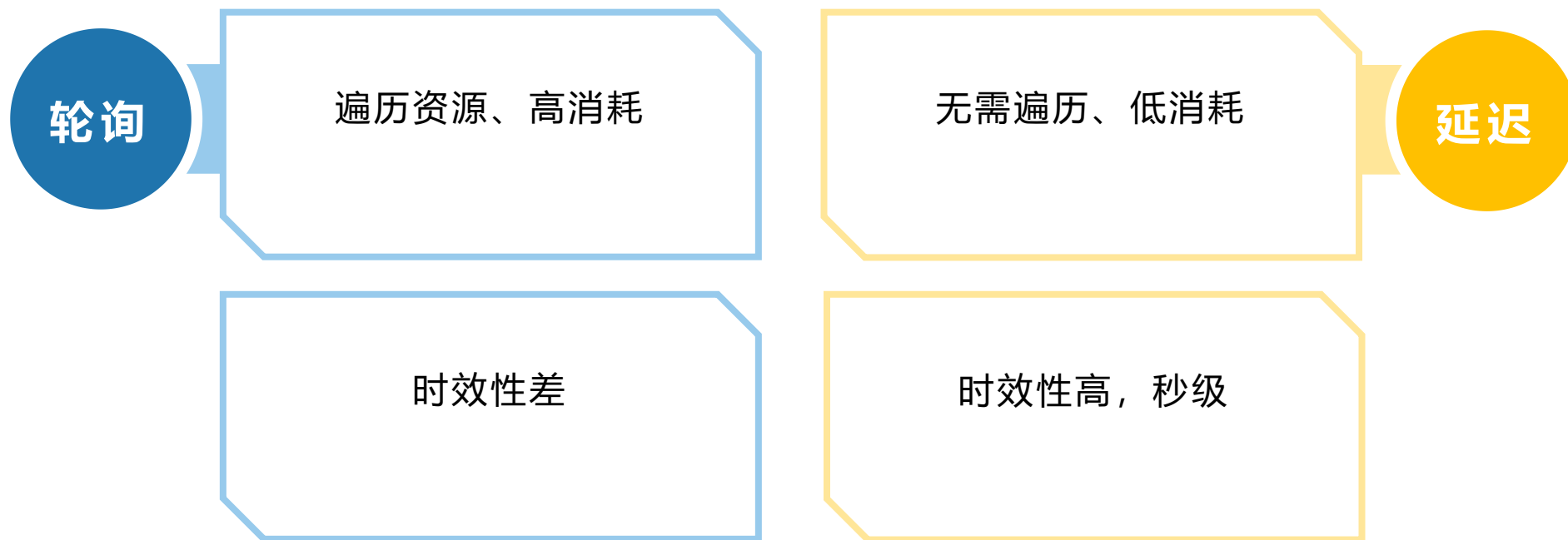
延迟消息的特点



### 两种方案的对比

各个方案的特点

全网一手不加密IT课程，需要请加微信zzj97666九章来offer都有





### 1.3.1 RocketMQ 介绍

RocketMQ 所支持的延迟任务精度



### 1.3.2 发送延迟消息

修改创建订单方法 `createOrder`

全网一手不加密IT课程，需要请加微信zzj97666九章来offer都有



// 1. 发送订单创建消息

```
rocketMQService.sendMessage("seckill_order", JSON.toJSONString(order));
```



// 2. 发送订单付款状态校验消息

```
rocketMQService.sendDelayMessage("pay_check", JSON.toJSONString(order), 5);
```

### 1.3.2 发送延迟消息

修改创建订单方法 `createOrder`

```
public Order createOrder(long seckillActivityId, long userId) throws Exception {  
    /*  
     * 1. 创建订单  
     */  
    SeckillActivity seckillActivity = seckillActivityDao.querySeckillActivityById(seckillActivityId);  
    Order order = new Order();  
    // 采用雪花算法生成订单ID  
    order.setOrderNo(String.valueOf(snowFlake.nextId()));  
    order.setSeckillActivityId(seckillActivity.getId());  
    order.setUserId(userId);  
    order.setOrderAmount(seckillActivity.getSeckillPrice().longValue());  
    /*  
     * 2. 发送创建订单消息  
     */  
    rocketMQService.sendMessage(topic: "seckill_order", JSON.toJSONString(order));  
    /*  
     * 3. 发送订单付款状态校验消息  
     * 开源RocketMQ支持延迟消息，但是不支持秒级精度。默认支持18个level的延迟消息，这是通过broker端的messageDelayLevel配置项确定的，如下：  
     * messageDelayLevel=1s 5s 10s 30s 1m 2m 3m 4m 5m 6m 7m 8m 9m 10m 20m 30m 1h 2h  
     */  
    rocketMQService.sendDelayMessage(topic: "pay_check", JSON.toJSONString(order), delayTimeLevel: 5);  
    return order;  
}
```



### 1.3.3 处理延迟消息并关闭未支付订单

新增方法 onMessag

1

#### 读取消息

```
Order order = JSON.parseObject(message, Order.class)
```

2

#### 查询订单

```
Order orderInfo = orderDao.queryOrder(order.getOrderNo());
```

全网一手不加密IT课程，需要请加微信zzj97666九章来offer都有

3

#### 判断订单是否完成支付

```
if (orderInfo.getOrderStatus() != 2)
```

4

#### 关闭未支付订单

```
orderInfo.setOrderStatus(99);  
orderDao.updateOrder(orderInfo);
```

### 1.3.3 处理延迟消息并关闭未支付订单

#### 代码预览

```
@Slf4j
@Component
@RocketMQMessageListener(topic = "pay_check", consumerGroup = "pay_check_group")
public class PayStatusCheckListener implements RocketMQListener<MessageExt> {

    @Autowired
    private OrderDao orderDao;

    @Override
    public void onMessage(MessageExt messageExt) {
        String message = new String(messageExt.getBody(), StandardCharsets.UTF_8);
        log.info("接收到订单支付状态校验消息:" + message);
        Order order = JSON.parseObject(message, Order.class);
        //1. 查询订单
        Order orderInfo = orderDao.queryOrder(order.getOrderNo());
        //2. 判断订单是否完成支付
        if (orderInfo.getOrderStatus() != 2) {
            //3. 未完成支付关闭订单
            log.info("未完成支付关闭订单, 订单号: "+orderInfo.getOrderNo());
            orderInfo.setOrderStatus(99);
            orderDao.updateOrder(orderInfo);
        }
    }
}
```



### 访问 URL 发送秒杀请求创建订单

访问 <http://localhost:8082/seckill/buy/666888/12>

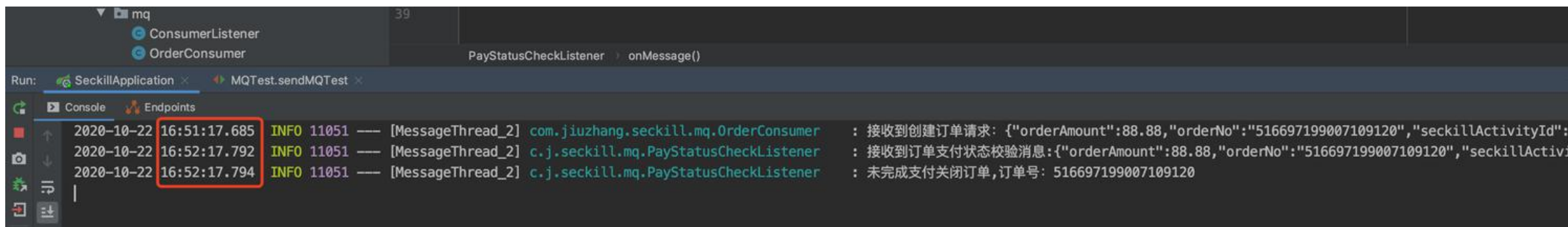
全网一手不加密IT课程，需要请加微信zzj97666九章来offer都有

← → ↻ ⓘ localhost:8082/seckill/buy/666888/12

恭喜你秒杀成功

### 在控制台查看测试结果

查看是否成功关闭订单



```
Run: SeckillApplication x MQTest.sendMQTest x
mq
  ConsumerListener
  OrderConsumer
    PayStatusCheckListener onMessage()

2020-10-22 16:51:17.685 INFO 11051 --- [MessageThread_2] com.jiuzhang.seckill.mq.OrderConsumer : 接收到创建订单请求: {"orderAmount":88.88,"orderNo":"516697199007109120","seckillActivityId":
2020-10-22 16:52:17.792 INFO 11051 --- [MessageThread_2] c.j.seckill.mq.PayStatusCheckListener : 接收到订单支付状态校验消息: {"orderAmount":88.88,"orderNo":"516697199007109120","seckillActivi
2020-10-22 16:52:17.794 INFO 11051 --- [MessageThread_2] c.j.seckill.mq.PayStatusCheckListener : 未完成支付关闭订单, 订单号: 516697199007109120
```



## 2. 订单页面的开发与测试

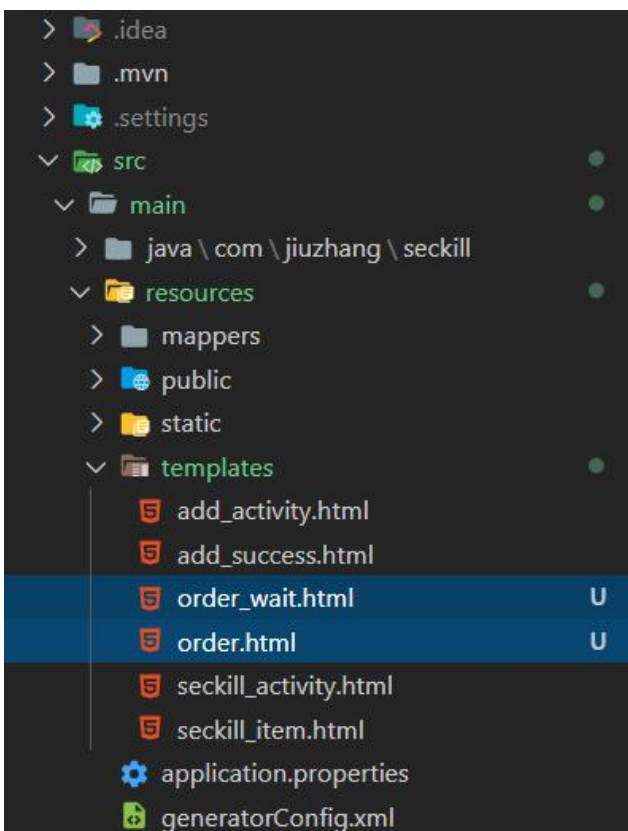
全网一手不加密IT课程，需要请加微信zzj97666九章来offer都有

实现订单页面及对应的 Controller



### 新增的模板资源文件

放置在 src/main/resources 目录下



### 如何将订单信息渲染至模板中

根据订单 ID 查询订单信息并渲染至模板中

- 1 定义路径，获取订单 ID**  
`@RequestMapping("/seckill/orderQuery/{userId}/{orderNo}")`
- 2 查询订单并判断订单是否存在**  
`Order order = orderDao.queryOrder(orderNo);  
if (order != null)`
- 3 订单存在，渲染 order 模板**  
`modelAndView.setViewName("order");`
- 4 订单不存在，渲染 order\_wait 模板**  
`modelAndView.setViewName("order_wait");`

### 如何将订单信息渲染至模板中

#### 代码预览

```
/**
 * 订单查询
 * @param orderNo
 * @return
 */
@RequestMapping("/seckill/orderQuery/{orderNo}")
public ModelAndView orderQuery(@PathVariable String orderNo) {
    Log.info("订单查询, 订单号: " + orderNo);
    Order order = orderDao.queryOrder(orderNo);
    ModelAndView modelAndView = new ModelAndView();

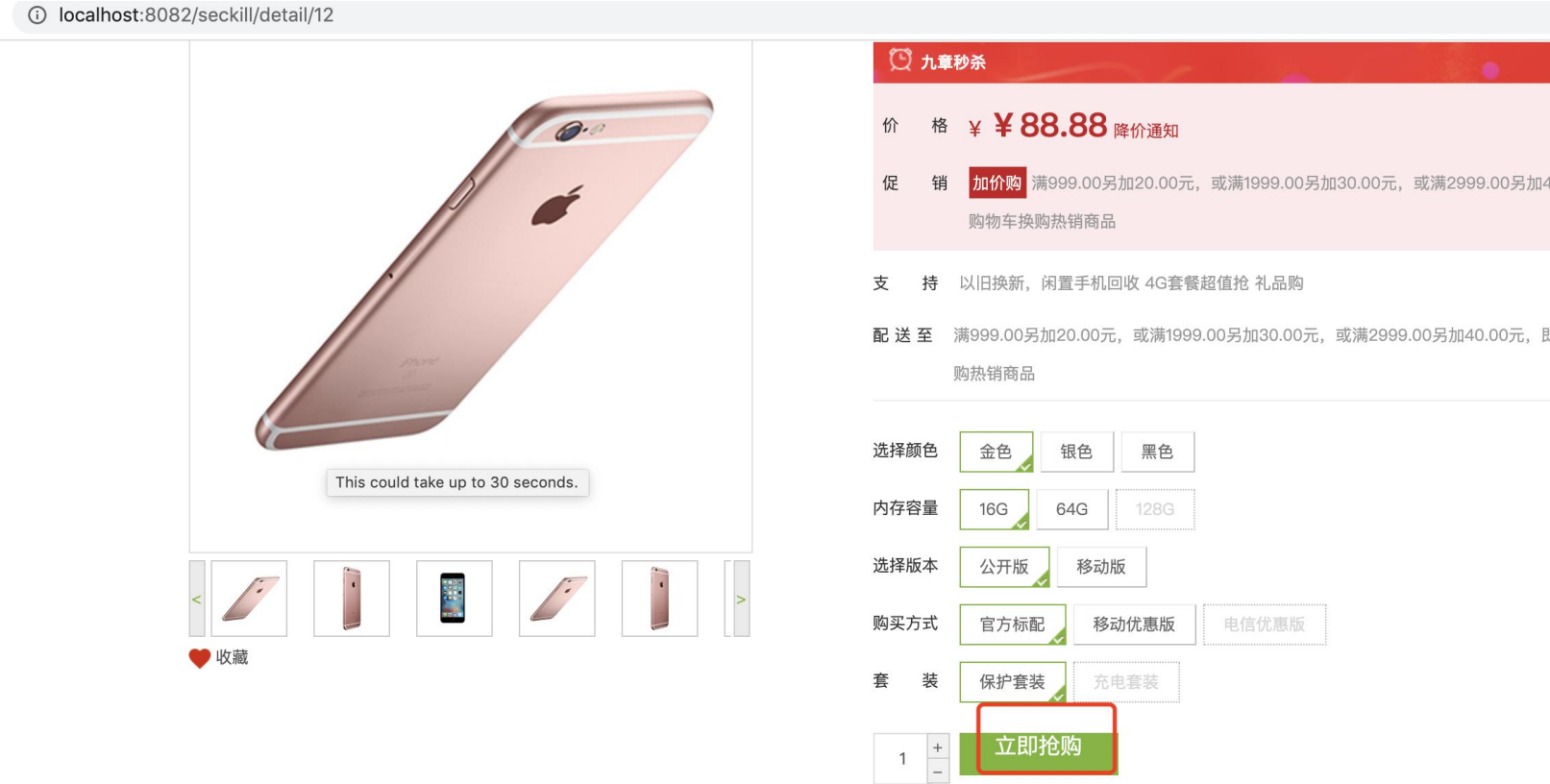
    if (order != null) {
        modelAndView.setViewName("order");
        modelAndView.addObject( attributeName: "order", order);
        SeckillActivity seckillActivity = seckillActivityDao.querySeckillActivityById(order.getSeckillActivityId());
        modelAndView.addObject( attributeName: "seckillActivity", seckillActivity);
    } else {
        modelAndView.setViewName("order_wait");
    }
    return modelAndView;
}
```



全网一手不加密IT课程，需要请加微信zzj97666九章来offer都有

### 2.3.1 秒杀抢购

访问活动页面，并点击抢购



### 2.3.2 秒杀结果

自动跳转到秒杀结果页



秒杀结果

秒杀成功，订单创建中，订单ID: 516776019382898688

查看订单信息



### 2.3.3 订单信息查询

跳转到[订单查询（支付）](#)页面

订单信息

支付方式

在线支付

货到付款

订单编号: 516766661848731648

订单状态: 已创建, 等待付款

秒杀活动信息: 超值秒杀--Apple 苹果 iPhone 11 Pro 手机 全网通 双卡双待 暗夜绿色 256GB  
This could take up to 30 seconds.

订单创建时间: 2020-10-22 21:27:19

订单金额:  
¥88.88

支付订单金额

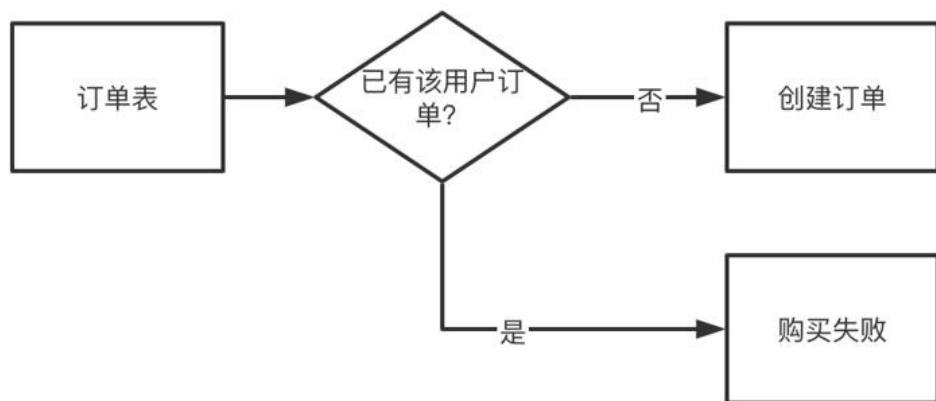
全网一手不加密IT课程, 需要请加微信zzj97666九章来offer都有

## 3. 限购功能的实现

限购功能有哪些技术要点？



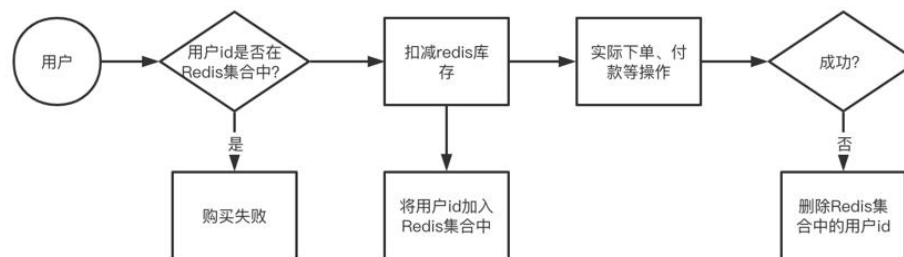
### MySQL 数据校验



### Redis 数据校验

全网一手不加密IT课程，需要请加微信zzj97666九章来offer都有

使用 Redis 提供的集合数据结构，将扣减 Redis 库存的用户 ID 写入。



语法: SADD KEY VALUE1.....VALUEN

作用: 将一个或多个成员元素加入到集合中，已经存在于集合的成员元素将被忽略。

`SADD seckill:28:commodity:189:user uid1`

语法: SISMEMBER KEY VALUE

作用: 判断成员元素是否是集合的成员。

`SISMEMBER seckill:28:commodity:189:user uid1`



### 1. SADD 命令介绍

Set Add 添加命令

SADD 集合(set)命令，在redis中主要功能是：  
向集合添加一个或多个成员，已经存在于集合的成员元素将被忽略。

#### 实例

```
redis 127.0.0.1:6379> SADD KEY_NAME VALUE1..VALUEN
```

```
redis 127.0.0.1:6379> SADD myset "hello"
```

```
(integer) 1
```

```
redis 127.0.0.1:6379> SADD myset "foo"
```

```
(integer) 1
```

```
redis 127.0.0.1:6379> SADD myset "hello"
```

```
(integer) 0
```

```
redis 127.0.0.1:6379> SMEMBERS myset
```

```
1) "hello"
```

```
2) "foo"
```



### 2. SISMEMBER

检查元素是否存在于 Set 中

全网一手不加密IT课程，需要请加微信zzj97666九章来offer都有

SISMEMBER 集合(set)命令，在redis中主要功能是：

检测一个元素是否存在于集合(set)中。

```
redis 127.0.0.1:6379> SISMEMBER KEY VALUE
```

返回值：

如果成员元素是集合的成员，返回 1 。

如果成员元素不是集合的成员，或 key 不存在，返回 0 。

```
redis 127.0.0.1:6379> SADD myset1 "hello"
(integer) 1
redis 127.0.0.1:6379> SISMEMBER myset1 "hello"
(integer) 1
redis 127.0.0.1:6379> SISMEMBER myset1 "world"
(integer) 0
```

### 添加限购名单

RedisService.addLimitMember

```
/**
 * 添加限购名单
 * @param activityId
 * @param userId
 */
public void addLimitMember(long activityId, long userId) {
    Jedis jedisClient = jedisPool.getResource();
    jedisClient.sadd("seckillActivity_users:"+activityId,String.valueOf(userId));
}
```



### 判断是否在限购名单中

`RedisService.isInLimitMember`

```
/**
 * 判断是否在限购名单中
 * @param activityId
 * @param userId
 * @return
 */
public boolean isInLimitMember(long activityId, long userId){
    Jedis jedisClient = jedisPool.getResource();
    boolean sismember = jedisClient.sismember("seckillActivity_users:"+activityId, String.valueOf(userId));
    log.info("userId:{} activityId:{} 在已购名单中:{}", activityId, userId, sismember);
    return sismember;
}
```

全网一手不加密IT课程，需要请加微信zzj97666九章来offer都有

### 移除限购名单

RedisService.removeLimitMember

```
/**
 * 移除限购名单
 * @param activityId
 * @param userId
 */
public void removeLimitMember(long activityId, long userId) {
    Jedis jedisClient = jedisPool.getResource();
    jedisClient.srem("seckillActivity_users:"+activityId,String.valueOf(userId));
}
```



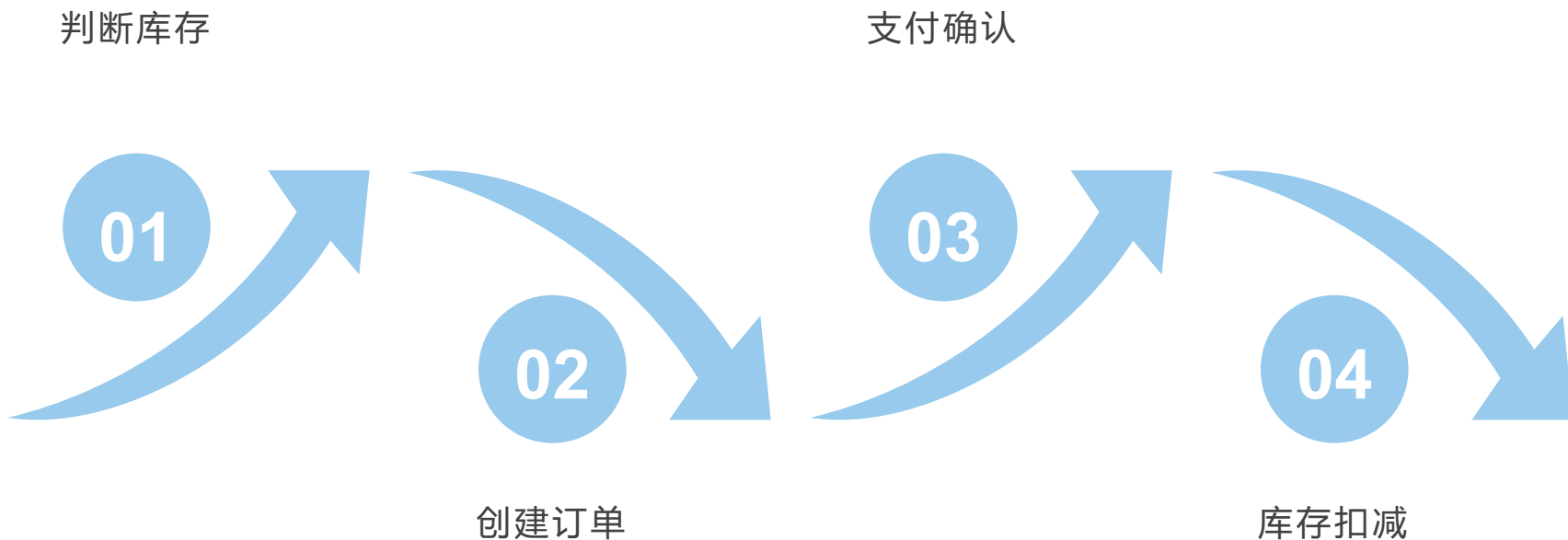
# 4. 付款、库存扣减数据一致性处理

全网一手不加密IT课程，需要请加微信zzj97666九章来offer都有

库存扣减的方案与流程

### 从订单创建到库存扣减的全流程

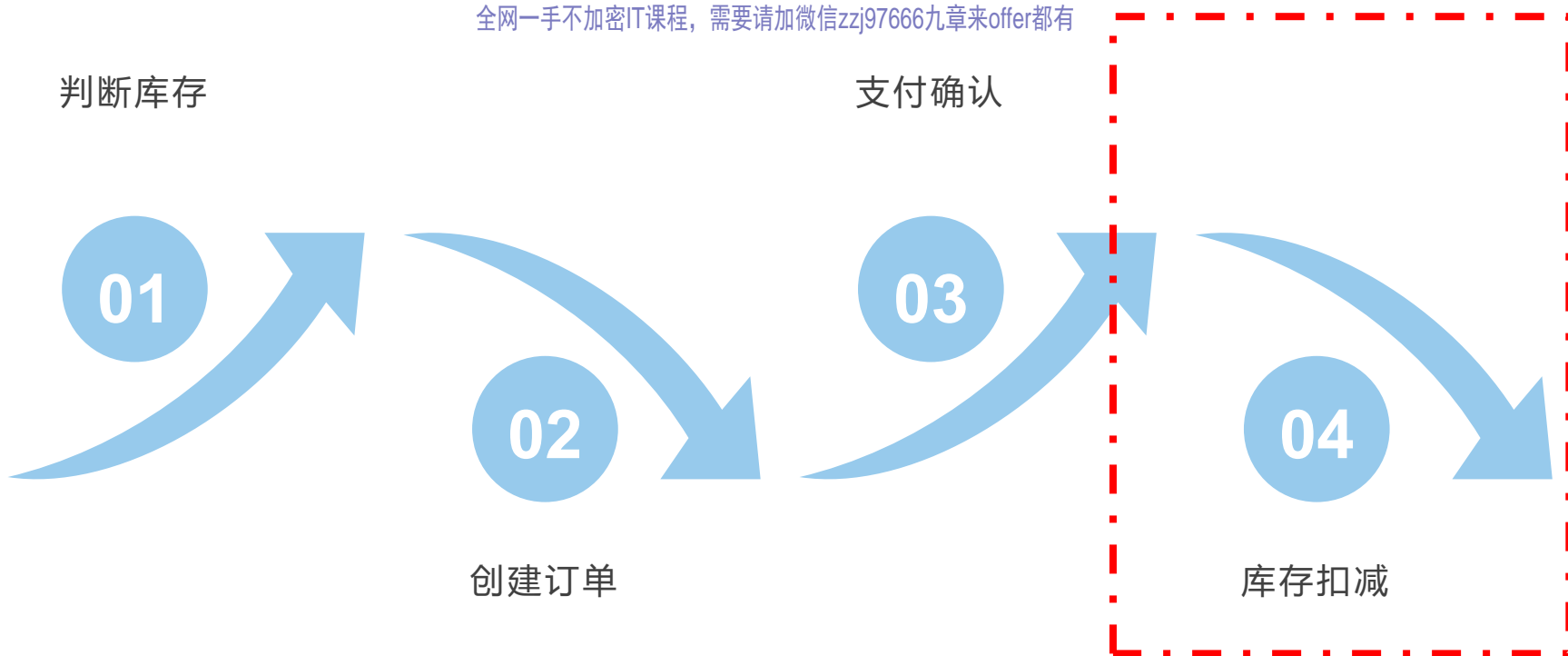
你能想到几种方案？



### 从订单创建到库存扣减的全流程

你能想到几种方案？

全网一手不加密IT课程，需要请加微信zzj97666九章来offer都有

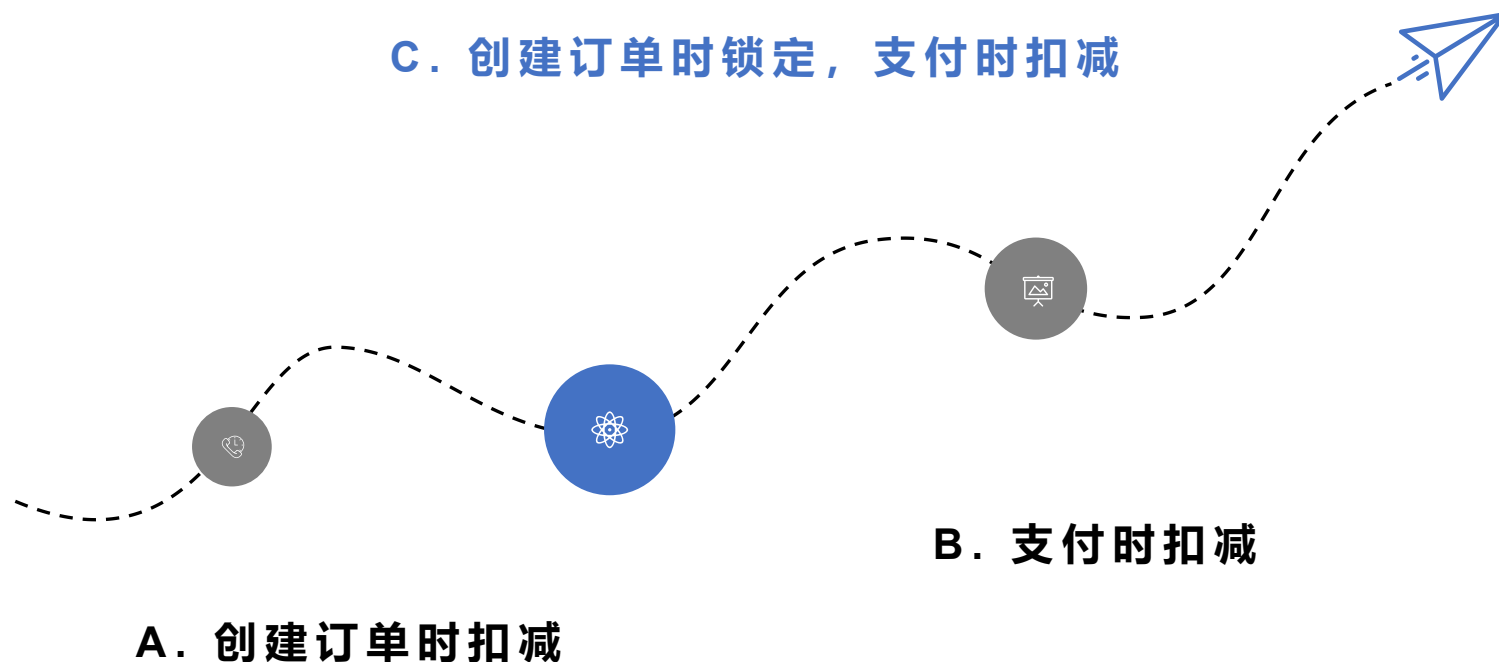


应该在什么时候扣减库存？



### 库存扣减可以发生在哪些阶段？

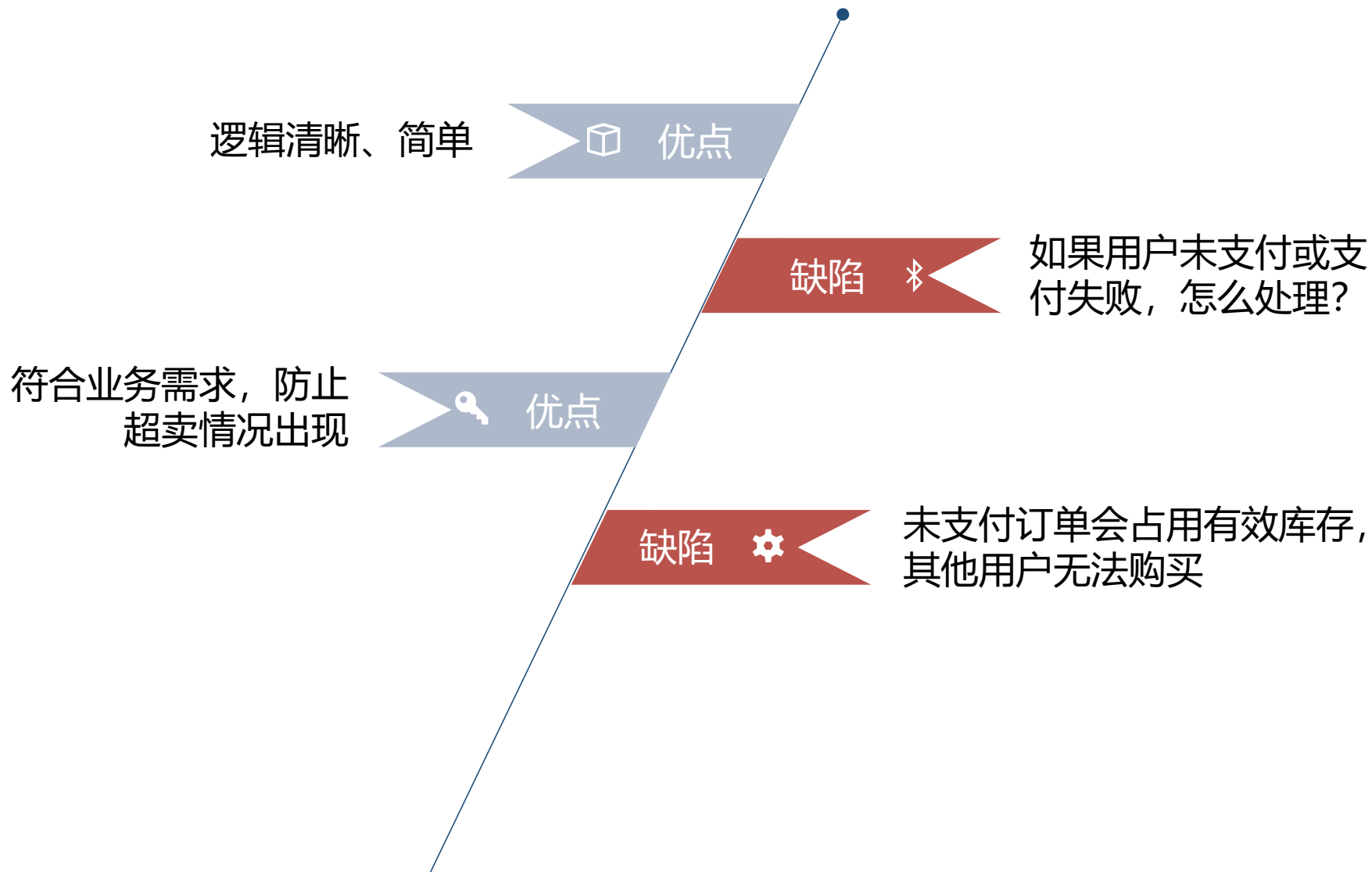
各种方案有什么利弊？



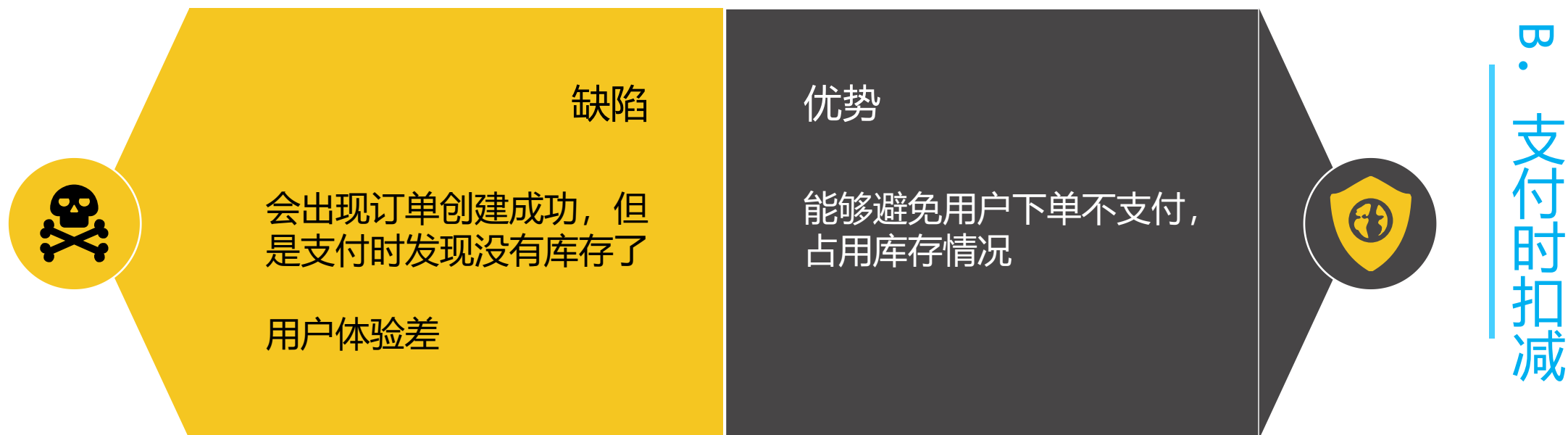
## 4.2 库存扣减时机分析

全网一手不加密IT课程，需要请加微信zzj97666九章来offer都有

### A. 创建订单时扣减



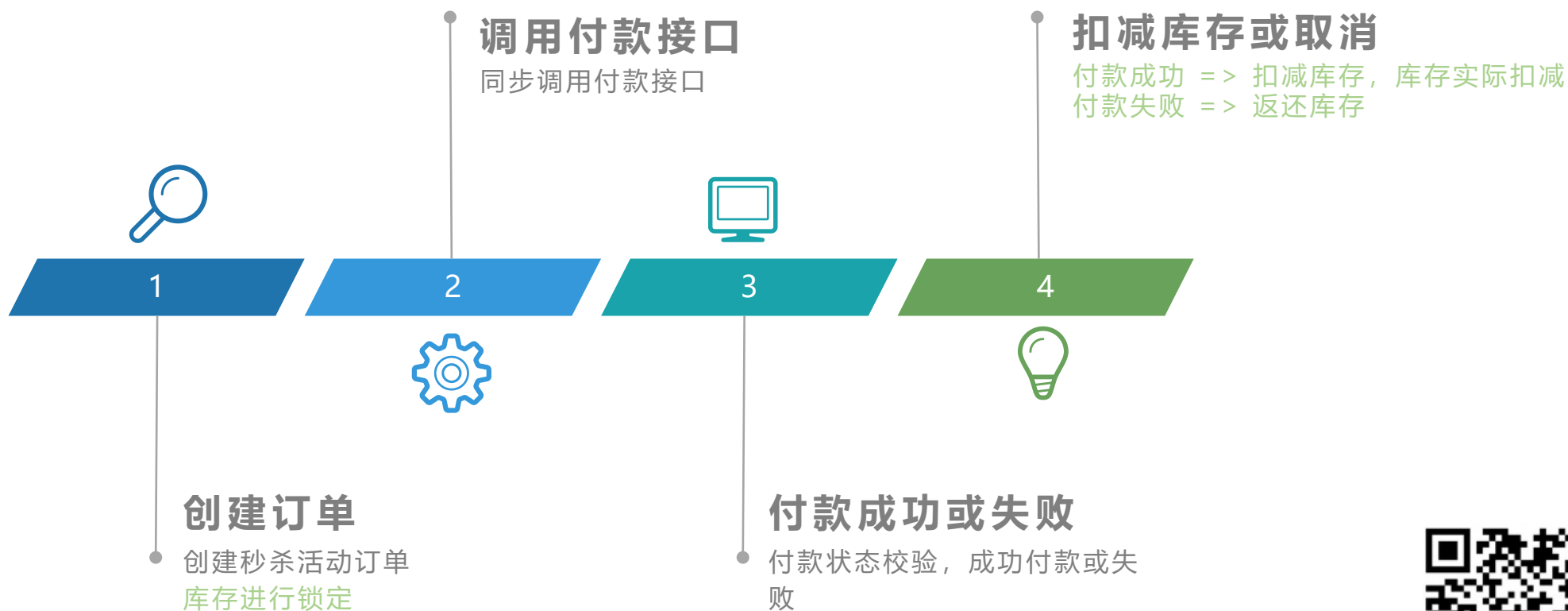
## 4.2 库存扣减时机分析



# C. 创建订单时锁定，支付时扣减

分段状态有什么优劣？

### 库存扣减流程



### 1. 创建订单锁定库存

分段锁定与扣减

```
@Override
@Transactional
public void onMessage(MessageExt messageExt) {
    //1.解析创建订单请求消息
    String message = new String(messageExt.getBody(),
StandardCharsets.UTF_8);
    log.info("接收到创建订单请求: " + message);
    Order order = JSON.parseObject(message, Order.class);
    order.setCreateTime(new Date());
    //2.锁定库存
    boolean lockStockResult =
seckillActivityDao.lockStock(order.getSeckillActivityId());
    if (lockStockResult) {
        //订单状态 0:没有可用库存,无效订单 1:已创建等待付款 ,2:支付完成
        order.setOrderStatus(1);
    } else {
        order.setOrderStatus(0);
    }
    //3.插入订单
    orderDao.insertOrder(order);
}
```

全网一手不加密IT课程, 需要请加微信zzj97666九章来offer都有

### 2.付款成功库存扣减

#### step1:付款完成发送付款完成消息

分段锁定与扣减

```
public void payOrderProcess(String orderNo) throws Exception {
    log.info("完成支付订单 订单号: " + orderNo);
    Order order = orderDao.queryOrder(orderNo);
    /*
     * 1.判断订单是否存在
     */
    if (order == null) {
        log.error("订单号对应订单不存在: " + orderNo);
        return;
    }
    /*
     * 2.订单支付完成
     */
    order.setPayTime(new Date());
    //订单状态 0:没有可用库存, 无效订单 1:已创建等待付款 ,2:支付完成
    order.setOrderStatus(2);
    orderDao.updateOrder(order);
    /*
     *3.发送订单付款成功消息
     */
    rocketMQService.sendMessage("pay_done", JSON.toJSONString(order));
}
```



## 4.2 库存扣减时机分析

全网一手不加密IT课程，需要请加微信zzj97666九章来offer都有

### step2:处理付款完成消息存

分段锁定与扣减

```
@Slf4j
@Component
@RocketMQMessageListener(topic = "pay_done", consumerGroup = "pay_check_group")
public class PayDoneConsumer implements RocketMQListener<MessageExt> {
    @Autowired
    private OrderDao orderDao;

    @Autowired
    private SeckillActivityDao seckillActivityDao;

    @Override
    public void onMessage(MessageExt messageExt) {
        //1.解析创建订单请求消息
        String message = new String(messageExt.getBody(), StandardCharsets.UTF_8);
        log.info("接收到创建订单请求: " + message);
        Order order = JSON.parseObject(message, Order.class);
        //2.扣减库存
        seckillActivityDao.deductStock(order.getSeckillActivityId());
    }
}
```



### 3. 订单关闭冻结库存回补

分段锁定与扣减

```
@Slf4j
@Component
@RocketMQMessageListener(topic = "pay_check", consumerGroup = "pay_check_group")
public class PayStatusCheckListener implements RocketMQListener<MessageExt> {
    @Autowired
    private OrderDao orderDao;

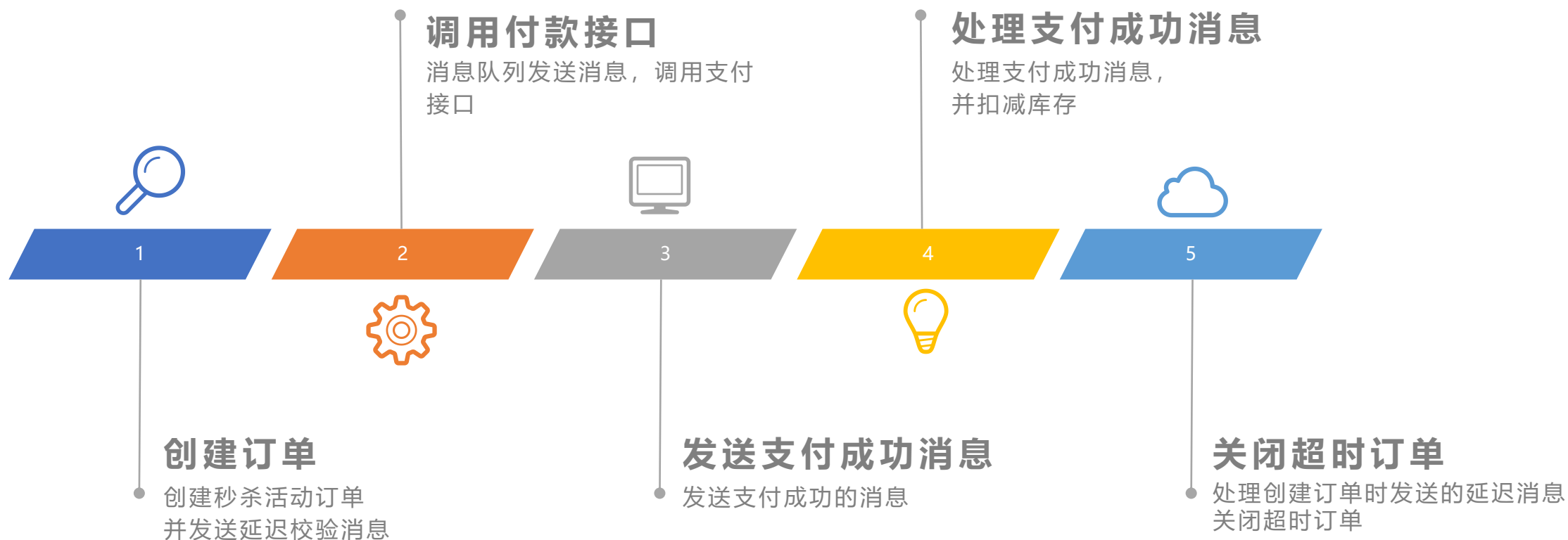
    @Autowired
    private SeckillActivityDao seckillActivityDao;

    @Override
    public void onMessage(MessageExt messageExt) {
        String message = new String(messageExt.getBody(), StandardCharsets.UTF_8);
        log.info("接收到订单支付状态校验消息:" + message);
        Order order = JSON.parseObject(message, Order.class);
        //1. 查询订单
        Order orderInfo = orderDao.queryOrder(order.getOrderNo());
        //2. 判读订单是否完成支付
        if (orderInfo.getOrderStatus() != 2) {
            //3. 未完成支付关闭订单
            log.info("未完成支付关闭订单, 订单号: " + orderInfo.getOrderNo());
            orderInfo.setOrderStatus(99);
            orderDao.updateOrder(orderInfo);
            //4. 回滚冻结的库存
            seckillActivityDao.revertStock(order.getSeckillActivityId());
        }
    }
}
```



## 库存扣减完整流程

### 逻辑流程



# 回顾与总结

回顾并总结本节主要的知识点



使用延迟消息关闭超时订单

全网一手不加密IT课程，需要请加微信zzj97666九章来offer都有

订单页面的开发与测试

限购功能的实现

付款库存扣减数据一致性处理

# Q & A

自由问答时间

