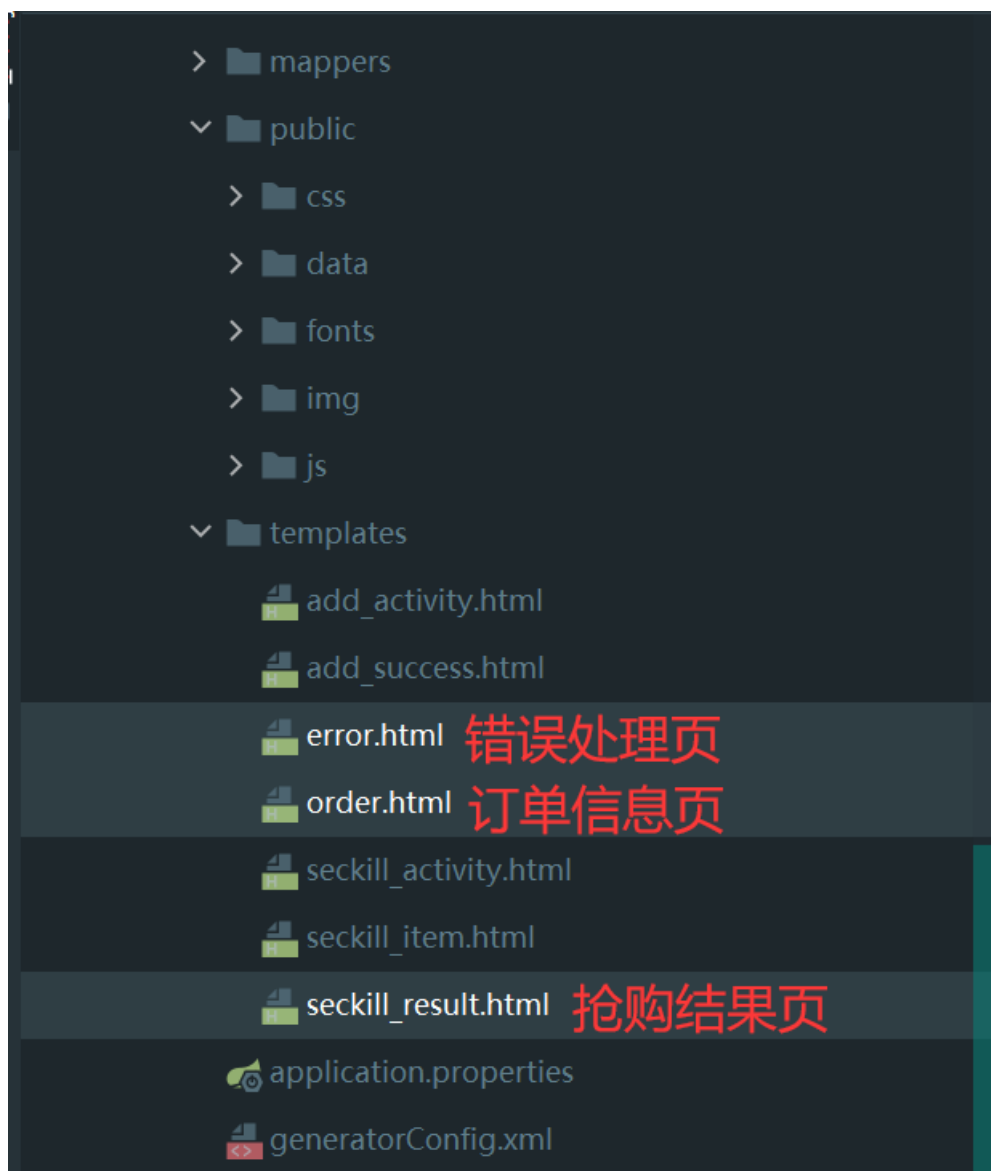


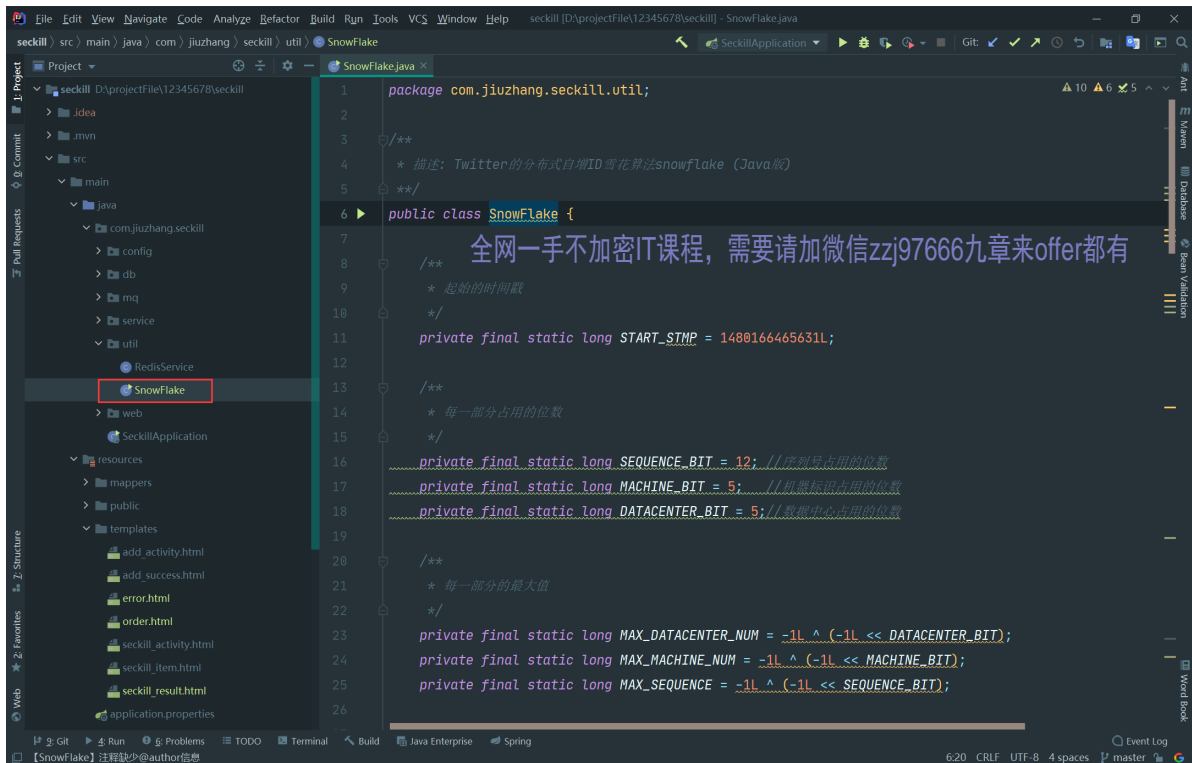
《保姆级教程》第八集 实现订单创建功能

1. 引入 3个前端页面

[页面zip下载链接](#)



2. 创建 分布式自增ID雪花算法 SnowFlake 类



```
package com.jiuzhang.seckill.util;

/**
 * 描述：Twitter的分布式自增ID雪花算法snowflake（Java版）
 */
public class SnowFlake {

    /**
     * 起始的时间戳
     */
    private final static long START_STMP = 1480166465631L;

    /**
     * 每一部分占用的位数
     */
    private final static long SEQUENCE_BIT = 12; //序列号占用的位数
    private final static long MACHINE_BIT = 5; //机器标识占用的位数
    private final static long DATACENTER_BIT = 5; //数据中心占用的位数

    /**
     * 每一部分的最大值
     */
    private final static long MAX_DATACENTER_NUM = -1L ^ (-1L << DATACENTER_BIT);
    private final static long MAX_MACHINE_NUM = -1L ^ (-1L << MACHINE_BIT);
    private final static long MAX_SEQUENCE = -1L ^ (-1L << SEQUENCE_BIT);

    /**
     * 每一部分向左的位移
     */
    private final static long MACHINE_LEFT = SEQUENCE_BIT;
    private final static long DATACENTER_LEFT = SEQUENCE_BIT + MACHINE_BIT;
    private final static long TIMESTMP_LEFT = DATACENTER_LEFT + DATACENTER_BIT;

    private long datacenterId; //数据中心
```

```

private long machineId;    //机器标识
private long sequence = 0L; //序列号
private long lastStmp = -1L; //上一次时间戳

public SnowFlake(long datacenterId, long machineId) {
    if (datacenterId > MAX_DATACENTER_NUM || datacenterId < 0) {
        throw new IllegalArgumentException("datacenterId can't be greater
than MAX_DATACENTER_NUM or less than 0");
    }
    if (machineId > MAX_MACHINE_NUM || machineId < 0) {
        throw new IllegalArgumentException("machineId can't be greater than
MAX_MACHINE_NUM or less than 0");
    }
    this.datacenterId = datacenterId;
    this.machineId = machineId;
}

/**
 * 产生下一个ID
 *
 * @return
 */
public synchronized long nextId() {
    long currStmp = getNewstmp();
    if (currStmp < lastStmp) {
        throw new RuntimeException("Clock moved backwards. Refusing to
generate id");
    }

    if (currStmp == lastStmp) {
        //相同毫秒内，序列号自增
        sequence = (sequence + 1) & MAX_SEQUENCE;
        //同一毫秒的序列数已经达到最大
        if (sequence == 0L) {
            currStmp = getNextMill();
        }
    } else {
        //不同毫秒内，序列号置为0
        sequence = 0L;
    }

    lastStmp = currStmp;

    return (currStmp - START_STMP) << TIMESTMP_LEFT //时间戳部分
        | datacenterId << DATACENTER_LEFT //数据中心部分
        | machineId << MACHINE_LEFT //机器标识部分
        | sequence; //序列号部分
}

private long getNextMill() {
    long mill = getNewstmp();
    while (mill <= lastStmp) {
        mill = getNewstmp();
    }
    return mill;
}

private long getNewstmp() {

```

```

        return System.currentTimeMillis();
    }

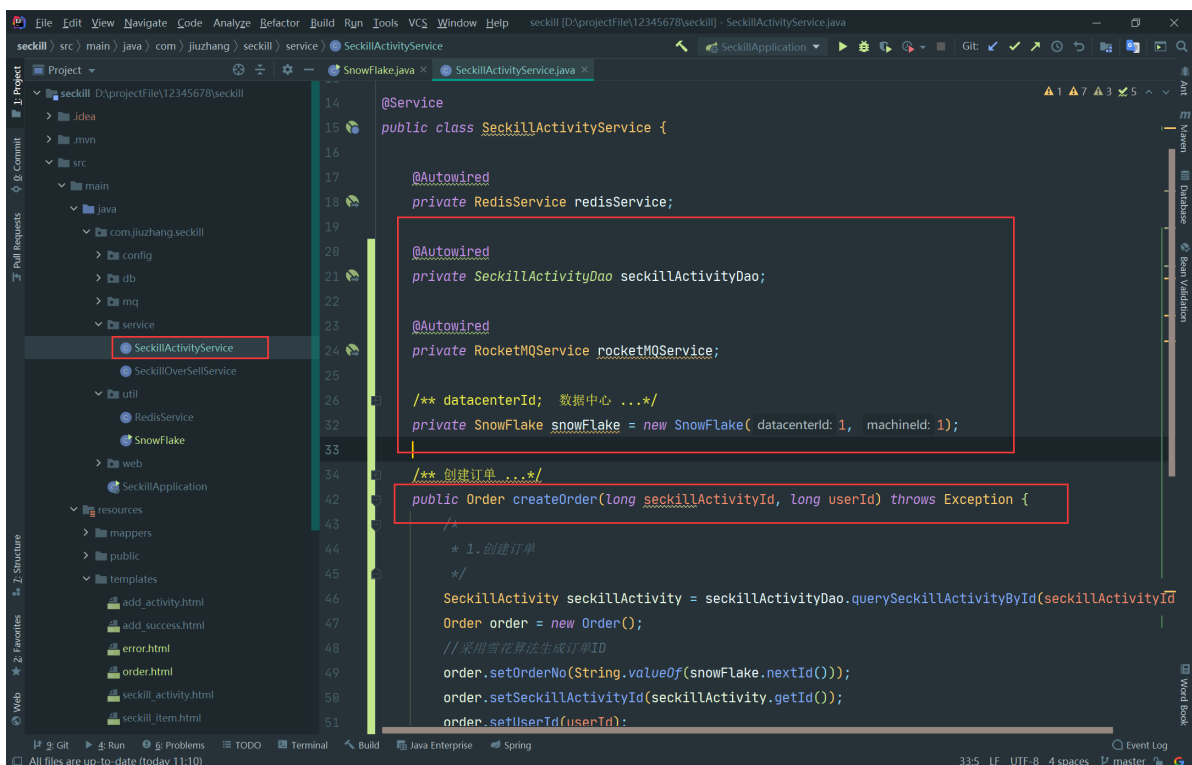
    public static void main(String[] args) {
        SnowFlake snowFlake = new SnowFlake(2, 1);

        long start = System.currentTimeMillis();
        for (int i = 0; i < 1000000; i++) {
            System.out.println(snowFlake.nextId());
        }

        System.out.println("总耗时: " + (System.currentTimeMillis() - start));
    }
}

```

3. 添加订单创建的方法 全网一手不加密IT课程，需要请加微信zzj97666九章来offer都有



```

@Autowired
private SeckillActivityDao seckillActivityDao;

@Autowired
private RocketMQService rocketMQService;

/**
 * datacenterId; 数据中心
 * machineId; 机器标识
 * 在分布式环境中可以从机器配置上读取
 * 单机开发环境中先写死
 */
private SnowFlake snowFlake = new SnowFlake(1, 1);

/**
 * 创建订单
 *
 * @param seckillActivityId

```

```

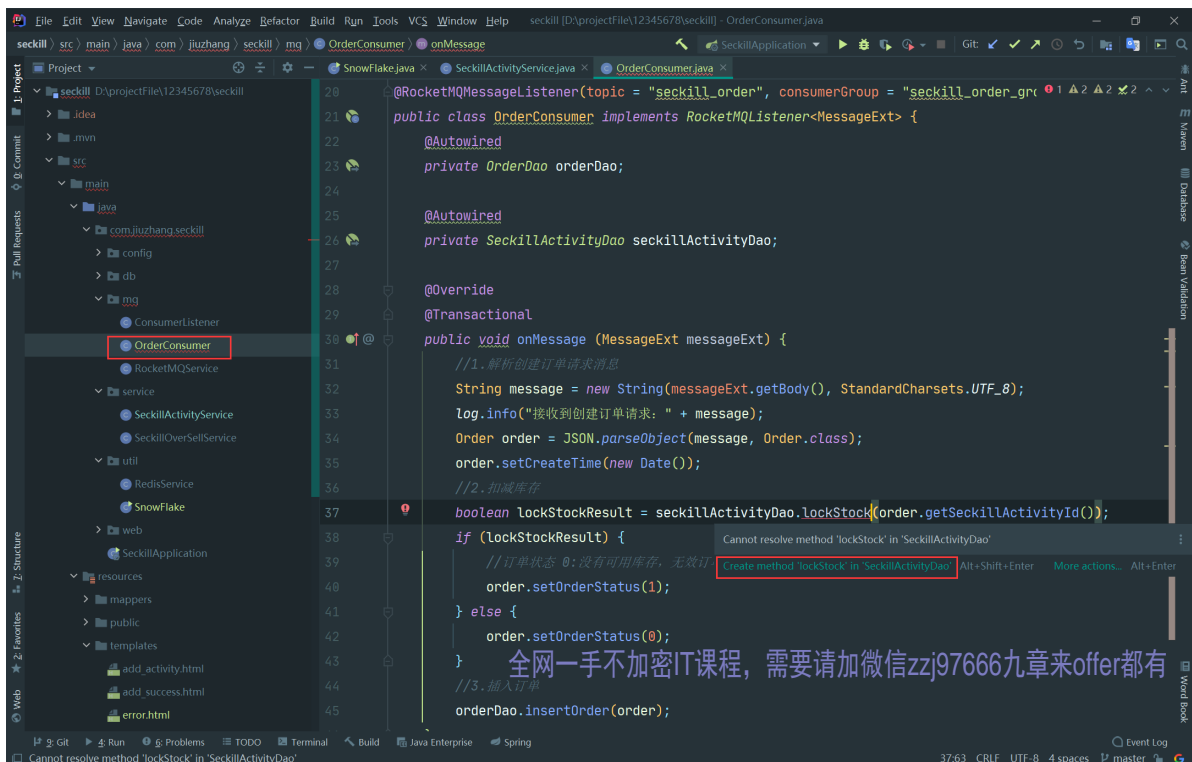
    * @param userId
    * @return
    * @throws Exception
    */
    public Order createOrder(long seckillActivityId, long userId) throws
Exception {
    /*
    * 1.创建订单
    */
    SeckillActivity seckillActivity =
seckillActivityDao.querySeckillActivityById(seckillActivityId);
    Order order = new Order();
    //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("seckill_order", JSON.toJSONString(order));

    return order;
}

```

4. 添加订单消费者类 OrderConsumer

注意：记得在类上面添加 3 个注解



```

@Slf4j
@Component
 RocketMQMessageListener(topic = "seckill_order", consumerGroup =
"seckill_order_group")
public class OrderConsumer implements RocketMQListener<MessageExt> {

```

```

@Autowired
private OrderDao orderDao;

@Autowired
private SeckillActivityDao seckillActivityDao;

@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.扣减库存
    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
        order.setOrderStatus(1);
    } else {
        order.setOrderStatus(0);
    }
    //3.插入订单
    orderDao.insertOrder(order);
}
}

```

5. 在 SeckillActivityDaoImpl 类中实现 lockStock方法

The screenshot shows the `SeckillActivityDaoImpl` class in an IDE. The class is annotated with `@Slf4j` and `@Repository`. It implements the `SeckillActivityDao` interface. The `lockStock` method is highlighted with a red box and contains a `log.error` statement. The code is as follows:

```

@Slf4j
@Repository
public class SeckillActivityDaoImpl implements SeckillActivityDao {

    @Resource
    private SeckillActivityMapper seckillActivityMapper;

    @Override
    public List<SeckillActivity> querySeckillActivitysByStatus(int activityStatus) {...}

    @Override
    public void insertSeckillActivity(SeckillActivity seckillActivity) { seckillActivityMapper.insert(s

    @Override
    public SeckillActivity querySeckillActivityById(long activityId) {...}

    @Override
    public void updateSeckillActivity(SeckillActivity seckillActivity) {...}

    @Override
    public boolean lockStock(Long seckillActivityId) {
        int result = seckillActivityMapper.lockStock(activityId);
        if (result < 1) {
            log.error("锁定库存失败");
            return false;
        }
    }
}

```

Annotations and comments in the image include:

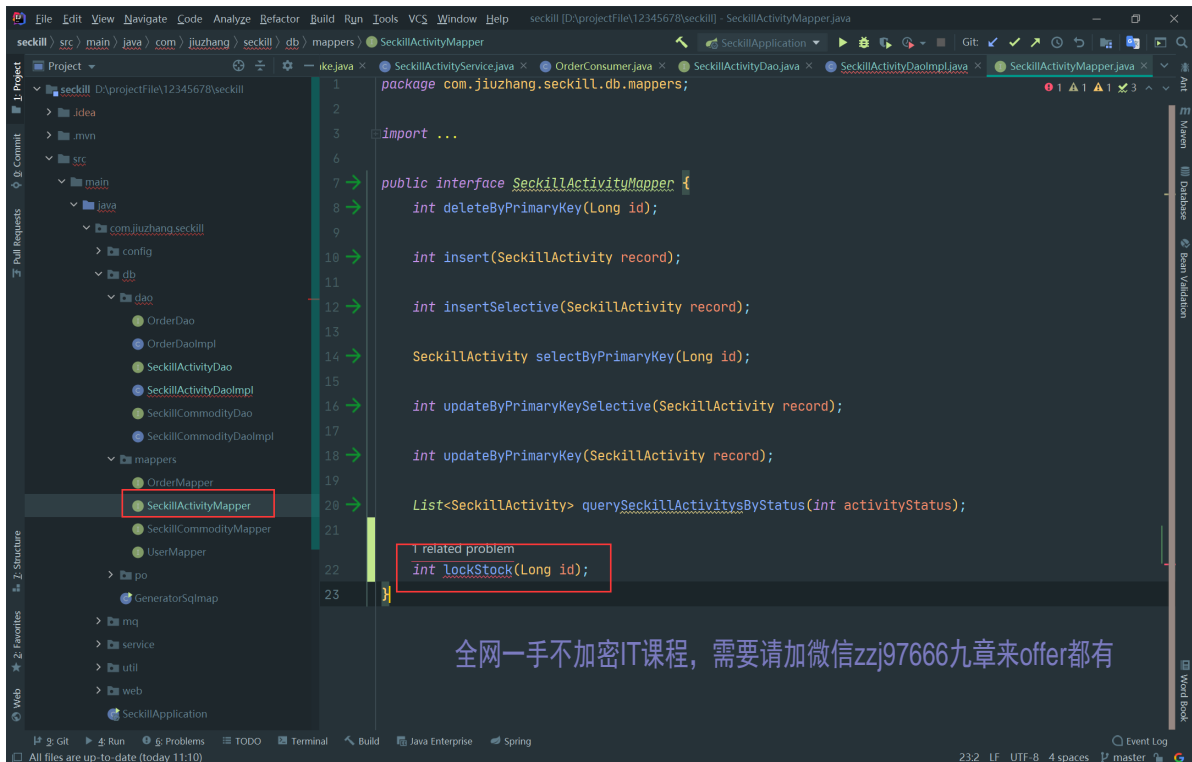
- `@Slf4j` (highlighted with a red box)
- `@Repository`
- Comment: "全网一手不加密IT课程, 需要请加微信zzj97666九章来offer都有"
- Comment: "SeckillActivity 接口中添加 lockStock 方法后, 然后在实现类中实现该方法"
- Comment: "Mapper类也需要创建 lockStock方法"
- Comment: "这里有用到 log输出日志, 所以类上面需要加上 @Slf4j 注解"

```

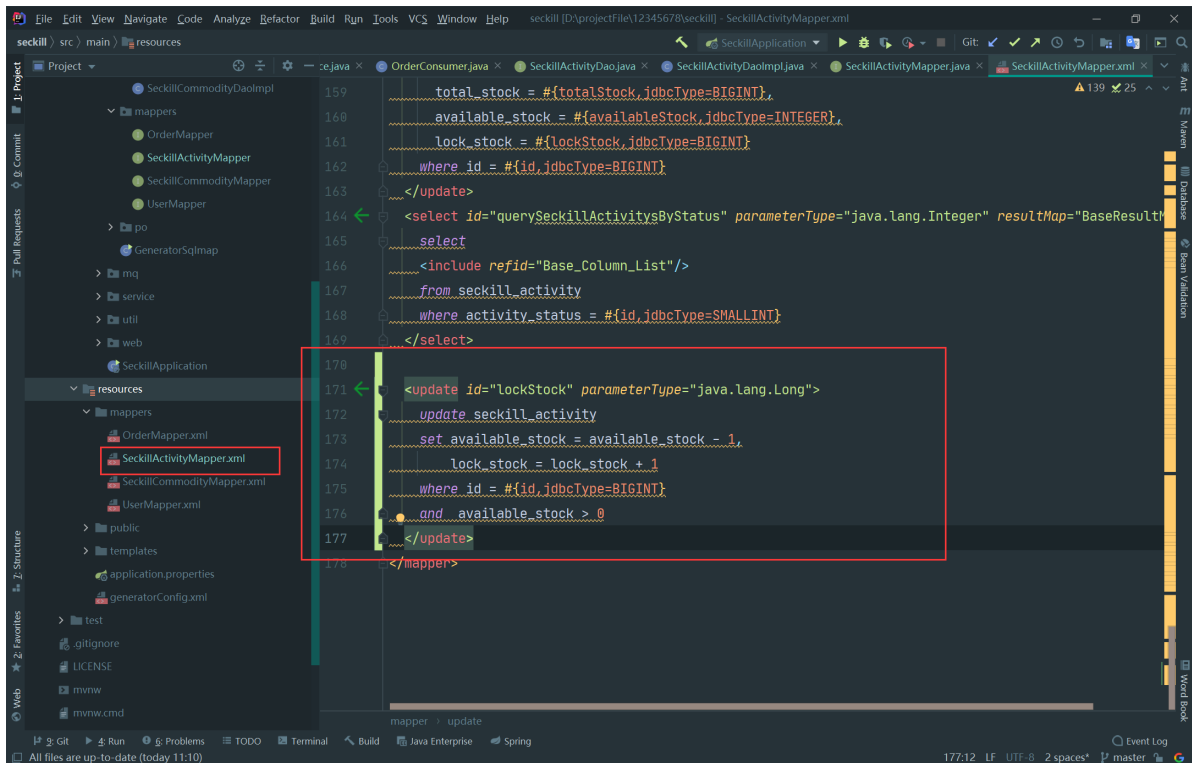
@Override
public boolean lockStock(Long seckillActivityId) {
    int result = seckillActivityMapper.lockStock(seckillActivityId);
    if (result < 1) {
        log.error("锁定库存失败");
        return false;
    }
    return true;
}

```

6. SeckillActivityMapper 中添加 lockStock 方法 & SeckillActivityMapper.xml 中添加 代码



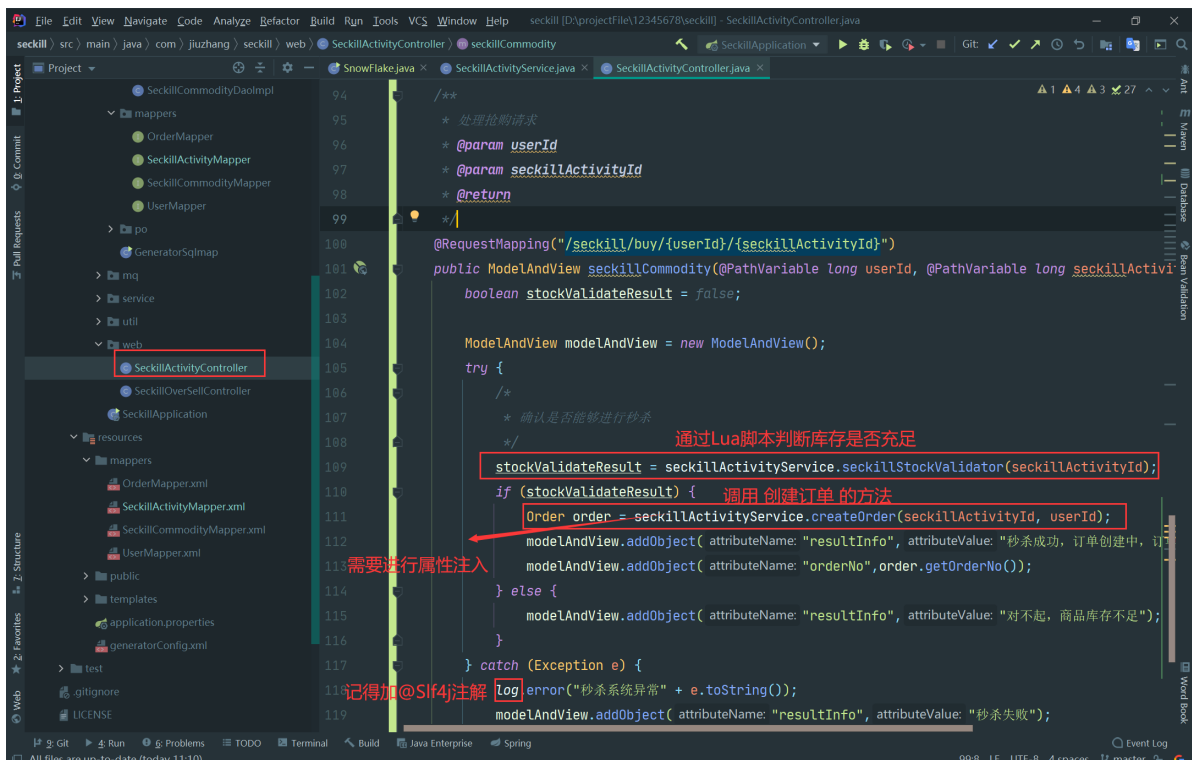
全网一手不加密IT课程，需要请加微信zzj97666九章来offer都有



```
<update id="lockStock" parameterType="java.lang.Long">
    update seckill_activity
    set available_stock = available_stock - 1,
        lock_stock = lock_stock + 1
    where id = #{id,jdbcType=BIGINT}
    and available_stock > 0
</update>
```

全网一手不加密IT课程，需要请加微信zzj97666九章来offer都有

7. 添加处理抢购请求的方法




```
@Autowired
seckillActivityService seckillActivityService;
```

```
/**
 * 处理抢购请求
 * @param userId
 * @param seckillActivityId
 * @return
 */
@RequestMapping("/seckill/buy/{userId}/{seckillActivityId}")
public ModelAndView seckillCommodity(@PathVariable long userId,
@PathVariable long seckillActivityId) {
    boolean stockValidateResult = false;

    ModelAndView modelAndView = new ModelAndView();
    try {
        /*
         * 确认是否能够进行秒杀
         */
        stockValidateResult =
seckillActivityService.seckillStockValidator(seckillActivityId);
        if (stockValidateResult) {
            Order order =
seckillActivityService.createOrder(seckillActivityId, userId);
            modelAndView.addObject("resultInfo", "秒杀成功，订单创建中，订单ID: "
+ order.getOrderNo());
            modelAndView.addObject("orderNo", order.getOrderNo());
        } else {
            modelAndView.addObject("resultInfo", "对不起，商品库存不足");
        }
    } catch (Exception e) {
        log.error("秒杀系统异常" + e.toString());
        modelAndView.addObject("resultInfo", "秒杀失败");
    }
    modelAndView.setViewName("seckill_result");
    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

8. 启动项目测试

首先先确保数据库中商品有 可用库存哦， 库存为 0 的话，大家手动改一下就好了

seckill_activity @seckill_jiuzhang (localhost:MySQL) - 表 - Navicat for MySQL

文件 编辑 查看 表 收藏夹 工具 窗口 帮助

连接 新建查询 表 视图 函数 用户 其它 查询 备份 自动运行 模型 图表

对象 seckill_activity @seckill_jiuzhang (lo...

开始事务 文本 筛选 排序 导入 导出

id	name	commodity_id	old_price	seckill_price	activity_status	start_time	end_time	total_stock	available_stock	lock_stock
1	测试1	999	2.88	99.00	0	2020-11-01 19:21:20	(Null)	0	0	0
2	测试2	999	3.88	99.00	0	2020-11-01 19:21:20	(Null)	0	0	0
3	测试3	999	8.99	99.00	0	2020-11-01 19:21:20	(Null)	0	0	0
4	测试4	999	0.00	99.00	0	2020-11-01 19:21:20	(Null)	0	0	0
5	测试5	999	0.00	99.00	0	2020-11-01 19:21:20	(Null)	0	0	0
6	测试6	999	0.00	99.00	0	2020-11-01 19:21:20	(Null)	0	0	0
7	测试	999	0.00	99.00	0	2020-11-01 19:21:20	(Null)	0	0	0
8	测试	999	0.00	99.00	0	2020-11-01 19:21:20	(Null)	0	0	0
9	你好好红红火火	999	99.99	88.88	1	2020-11-01 19:21:20	(Null)	0	0	0
10	超值秒杀-Apple 苹果 iPhone 11 Pro 手机	999	8769.00	7769.00	1	2020-11-01 19:21:20	(Null)	0	0	0
11	超值秒杀-Apple 苹果 iPhone 11 Pro 手机	999	8769.00	7769.00	1	2020-11-01 19:21:20	2020-11-20	10	0	0
12	超值秒杀-Apple 苹果 iPhone 11 Pro 手机	999	99.99	88.88	1	2020-11-01 19:21:20	2020-11-18	10	0	0
19	iPhone12 Pro 手机	1001	7888.00	5888.00	1	2020-11-05 08:39:24	2020-11-05	10	7	0

SELECT * FROM `seckill_jiuzhang`.`seckill_activity` LIMIT 0,1000

第 1 条记录 (共 13 条) 于第 1 页

九章秒杀-正品秒杀!

localhost:8082/seckill



你好好红红火火

¥88.88 ~~¥99.99~~

已售999

剩余 999件

立即抢购



超值秒杀-Apple 苹果 iPhone 11 Pro 手机

全网通 双卡双待 暗夜绿色 256GB

¥7769.00 ~~¥8769.00~~

已售999

剩余 999件

立即抢购



超值秒杀-Apple 苹果 iPhone 11 Pro 手机

全网通 双卡双待 暗夜绿色 256GB

¥7769.00 ~~¥8769.00~~

已售999

剩余 999件

立即抢购



超值秒杀-Apple 苹果 iPhone 11 Pro 手机

全网通 双卡双待 暗夜绿色 256GB

¥88.88 ~~¥99.99~~

已售999

剩余 999件

立即抢购



iPhone12 Pro 秒杀抢购活动

¥5888.00 ~~¥7888.00~~

已售999

剩余 999件

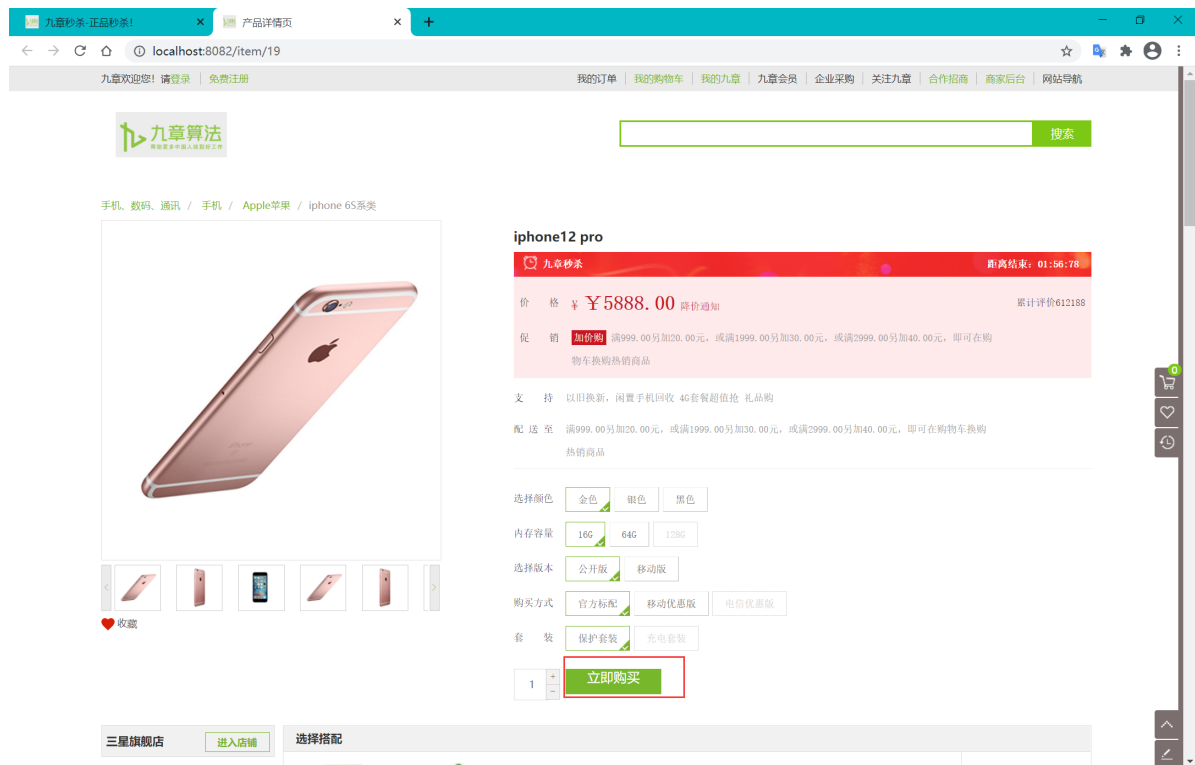
立即抢购

全网一手不加密IT课程, 需要请加微信zzj97666九章来offer都有

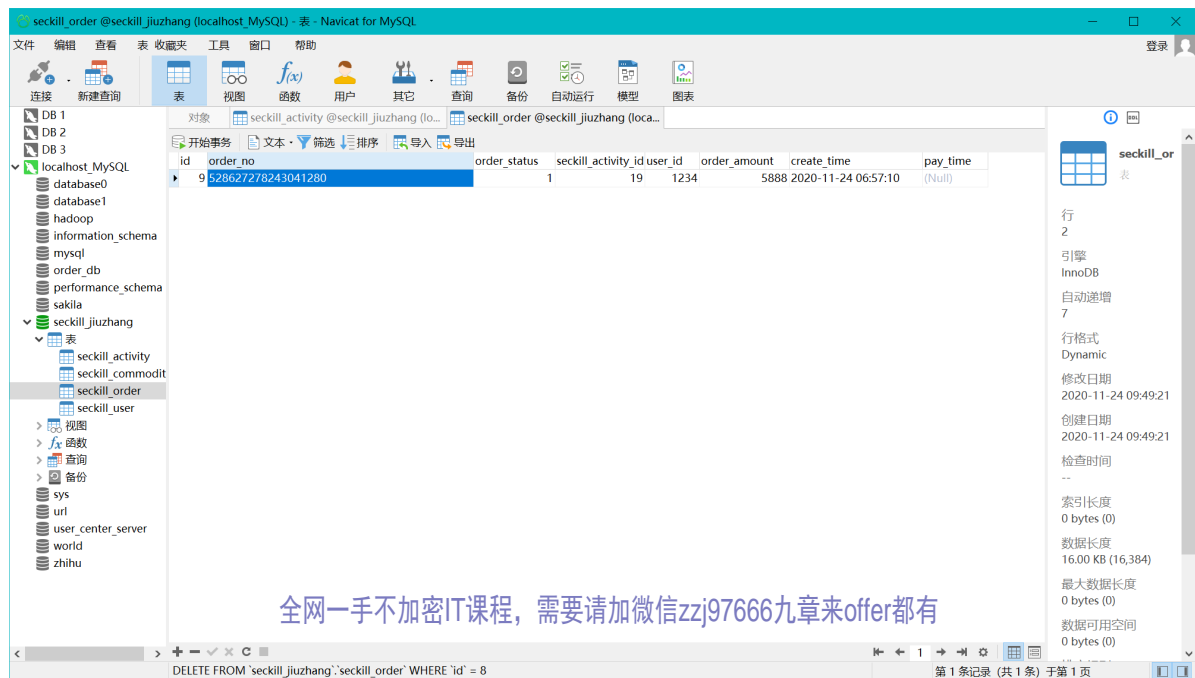
点你有库存的商品, 助教的是 19 号商品有库存

TOP

localhost:8082/item/19



在检查一下你的数据库，看看有没有这个订单。



全网一手不加密IT课程，需要请加微信zzj97666九章来offer都有

好的，订单创建部分就到这里了。