跨境通定时任务 - 代码逻辑

## JobGetChangeProductFromKJT

**跨境通信息变化的商品入库**

完整类名：com.hjy.selleradapter.job.JobGetChangeProductFromKJT.java

方法标识：Product.ProductIDGetQuery| RsyncConfigGetKjtProductIdByDate.java

Product.ProudctInfoBatchGet| RsyncGetKjtProductIdByDate.java line165-> |

RsyncConfigGetKjtProductById.java ->RsyncGetKjtProductById.java

Step 1：

RsyncGetKjtProductIdByDate.java 继承RsyncKjt类，执行doRsync()方法。

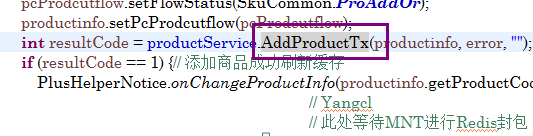
ILcRsyncKjtLogService对logcenter.lc\_rsync\_kjt\_log表进行数据插入；获取响应数据后，第一次更新这条记录，包括更新响应内容和响应时间。注意：这是第一次更新

Step 2：

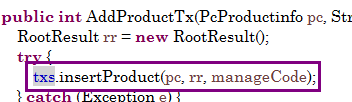
RsyncKjt.java的子类RsyncGetKjtProductIdByDate 执行doProcess( )；在执行的时候，如果请求接口时候返回的数据中tResponse.getData().getProductList()不为空，则进行循环数据处理。其中一步：MWebResult mResult = saveProductData()；这个方法中调用了另一个java类：RsyncGetKjtProductById.java 这个类也继承了RsyncKjt.java。并在saveProductData()方法中调用父类doRsync()。然后请求接口，并且调用RsyncGetKjtProductById的doProcess( )。 在这个方法中：MWebResult mResult = saveProductData(info); 执行了数据的插入或更新。(查询商品信息，并入库)

Step 3

MWebResult mResult = saveProductData(info) 若果不存在，就添加：



productService.AddProductTx()方法中ITxProductService txs;又是一个Service：



txs.insertProduct()方法插入了如下信息：

1 插入商品基本信息

IPcProductinfoDao pc\_productinfo表插入数据

2添加商品的分类信息

IUcSellercategoryProductRelationDao uc\_sellercategory\_product\_relation表插入数据

3添加商品的实类信息

IPcProductcategoryRelDao pc\_productcategory\_rel表插入数据

4添加描述信息

IPcProductdescriptionDao pc\_productdescription表插入数据

5插入商品图片信息

IPcProductpicDao pc\_productpic表插入数据

6插入sku信息

IPcSkuinfoDao pc\_skuinfo表插入数据

7插入商品sku库存

IScStoreSkunumDao sc\_store\_skunum表插入数据

8插入商品属性信息

IPcProductpropertyDao pc\_productproperty表插入数据

9插入商品扩展信息

IPcProductinfoExtDao pc\_productinfo\_ext 表插入数据

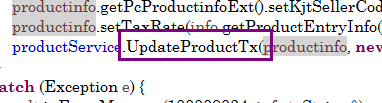
10插入商品历史流水信息

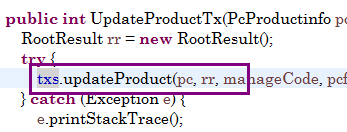
IPcProductflowDao pc\_productflow 表插入数据

11插入商品库存流水

ILcStockchangeDao lc\_stockchange表插入数据

若果存在，就更新：





txs.updateProduct()方法更新了如下信息：

1更新商品的扩展信息

IPcProductinfoExtDao pc\_productinfo\_ext 表更新数据

2 图片操作--先删除--再添加

IPcProductpicDao pc\_productpic表更新数据 先删除-再添加

3 更新商品的基本信息

IPcProductinfoDao pc\_productinfo表更新数据

4 插入或者更新sku信息

IPcSkuinfoDao pc\_skuinfo表更新数据

IScStoreSkunumDao sc\_store\_skunum表更新数据

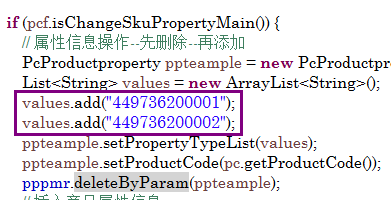
5 商品库存流水

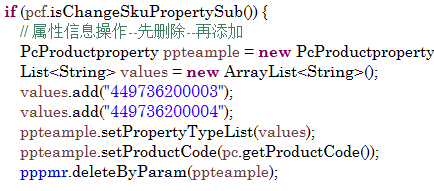
ILcStockchangeDao lc\_stockchange表更新数据

6 商品属性信息操作--先删除--再添加

IPcProductpropertyDao pc\_productproperty表先删除-再添加

这里又分为2种情况，非常蛋疼：





7 插入商品历史流水信息

IPcProductflowDao pc\_productflow 表插入数据

8 描述信息操作--先删除--再添加

IPcProductdescriptionDao pc\_productdescription表先删除-再添加

9 商品的分类信息

IUcSellercategoryProductRelationDao uc\_sellercategory\_product\_relation表更新数据

## JobForOrderSoCreate

**跨境通订单同步**

完整类名：com.hjy.selleradapter.job.JobForOrderSoCreate.java

方法标识：Order.SOCreate|RsyncConfigOrderSoCreate.java

Step 1：

首先从job\_exectimer表中读取可用数据List，数据获取依据为当前时间段1小时内，flag\_success为0且exec\_number小于20的记录集合

Step 2：

RootJobForExec.java执行execByInfo()方法，该方法由实现它的子类来实例化，即：

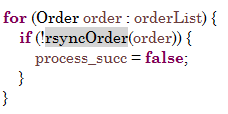
JobForOrderSoCreate.java的execByInfo()方法。

Step 3：

OrderForKJT. rsyncOrder()方法开始执行。这个方法中分为2种情况来执行。

首先从ordercenter.oc\_order\_kjt\_list表中根据order\_code取得一个List<OcOrderKjtList>集合；然后对这个集合进行判断：list.size() > 0则证明这个表中有数据；然后会执行本类中的rsyncOrder(Order order)方法，直到退出循环，不再执行下面的代码。

否则从oc\_orderinfo表中根据order\_code寻找记录；这一步中有拆单规则，会将订单拆为order\_code + "#" +1…n的新订单。目前按照金额拆单，≤2000元则不拆单，最后返回一个List<Order>集合。然后遍历这个集合，ordercenter.oc\_order\_kjt\_list、ordercenter.oc\_order\_kjt\_list\_data和ordercenter.oc\_order\_kjt\_detail三个表都会插入数据。跳出else后，会执行如下图的代码，直到退出循环。



Step 4：rsyncOrder(Order order)

1 RsyncRequestOrderSoCreate类开始组装请求报文；

2 执行父类RsyncKjt的doRsync()；

3 父类RsyncKjt在doRsync()中回调子类OrderSoCreate的doProcess()方法；

4 ordercenter.oc\_order\_kjt\_list开始更新；

5 ordercenter.oc\_orderinfo开始更新；

6 lc\_orderstatus开始更新；