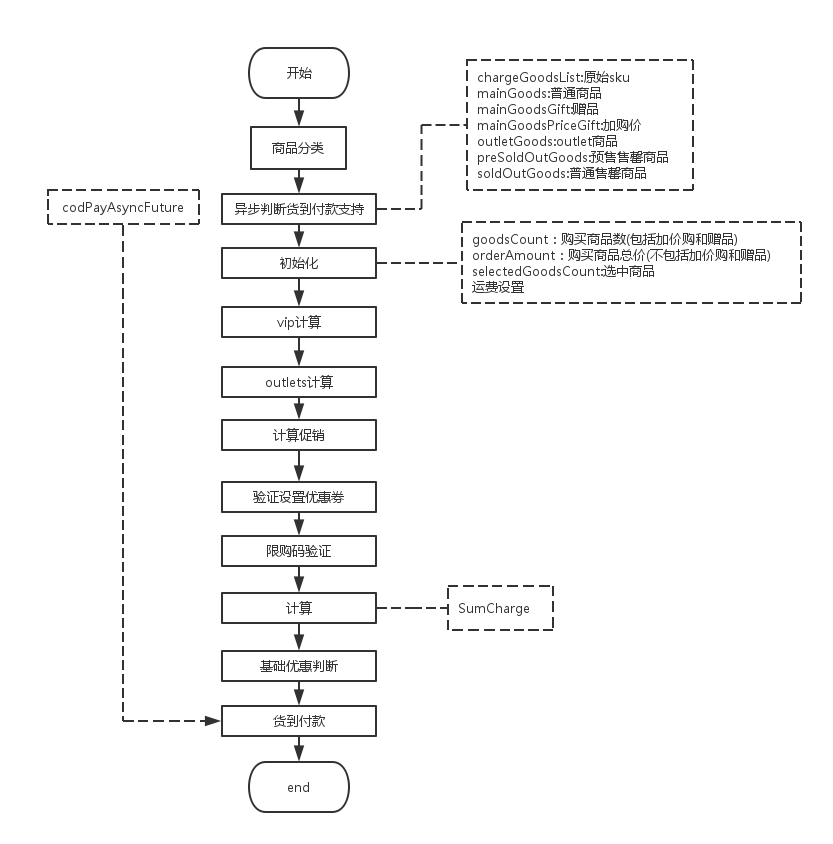
#### Shoping模块

购物车计费模块：

类封装了计费服务，其大致类图如下所示：

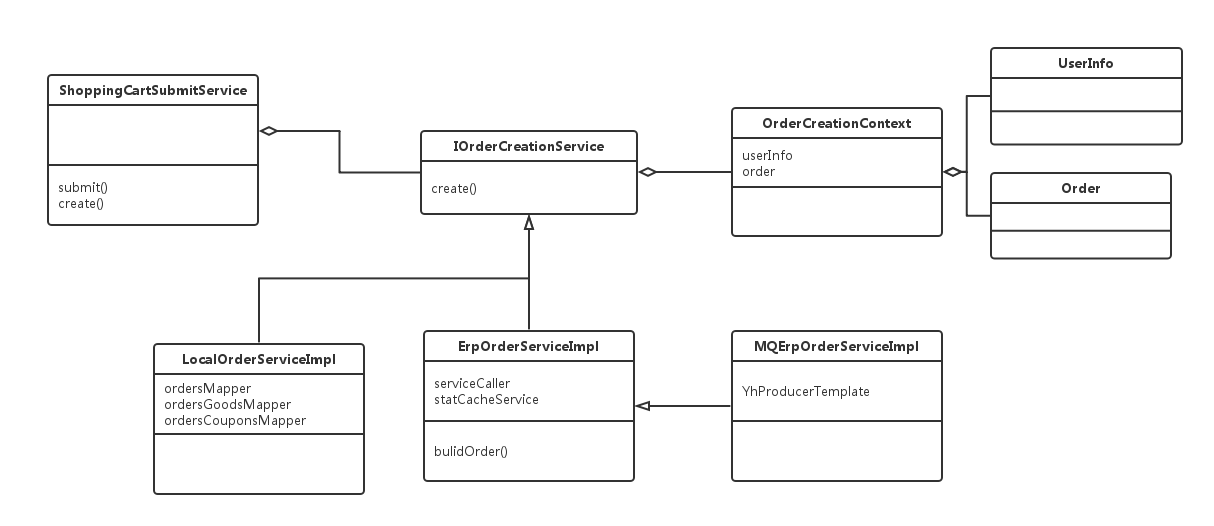


ChargerService类的charge方法向外提供计费服务，在购物车提交或查询时会调用该方法进行金额计算。ChargeContext封装了购物车计算相关信息，ChargerService会按照顺序逐步调用相关的内部成员进行总金额的计算，ChargeContext对象会作为参数传递给这些成员的charge方法。大致charge的流程如下图所示：



#### 订单模块

订单创建主要在shopping模块的IOrderCreationService服务类中定义。其大致类图如下



订单的创建主要由购物车提交服务进行发起，IorderCreationService接口为订单创建接口，其有三个实现类，定义了具体的订单创建方式。其中ErpOrderServiceImpl为RPC方式创建订单，MQErpOrderServiceImpl为MQ方式创建订单，LocalOrderServiceImpl为创建订单的本地操作。

疑问：

1.RPC方式调用接口如下，其服务内部实现？

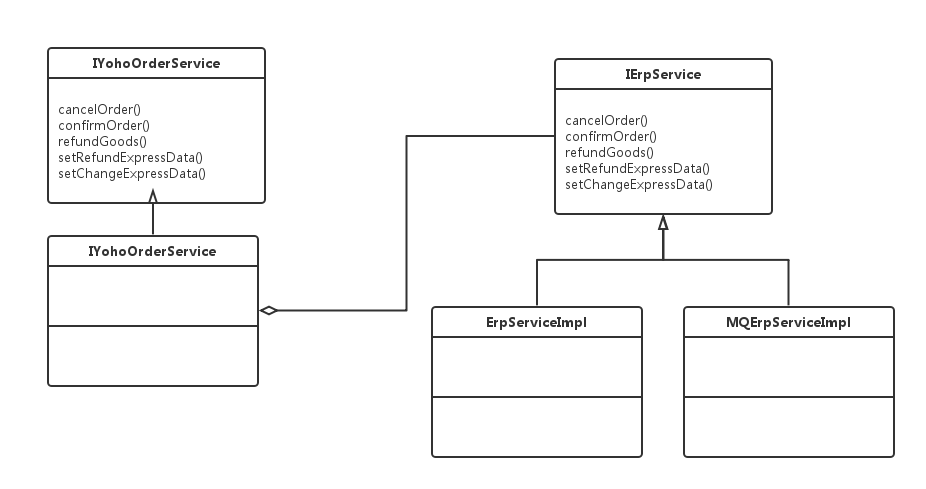
#创建erp订单接口地址

erp.order.create.url=http://portal.admin.yohobuy.com/api/orderform/create

2.两种创建订单方式均需要调用LocalOrderServiceImpl进行订单创建，其内部主要完成什么内容？

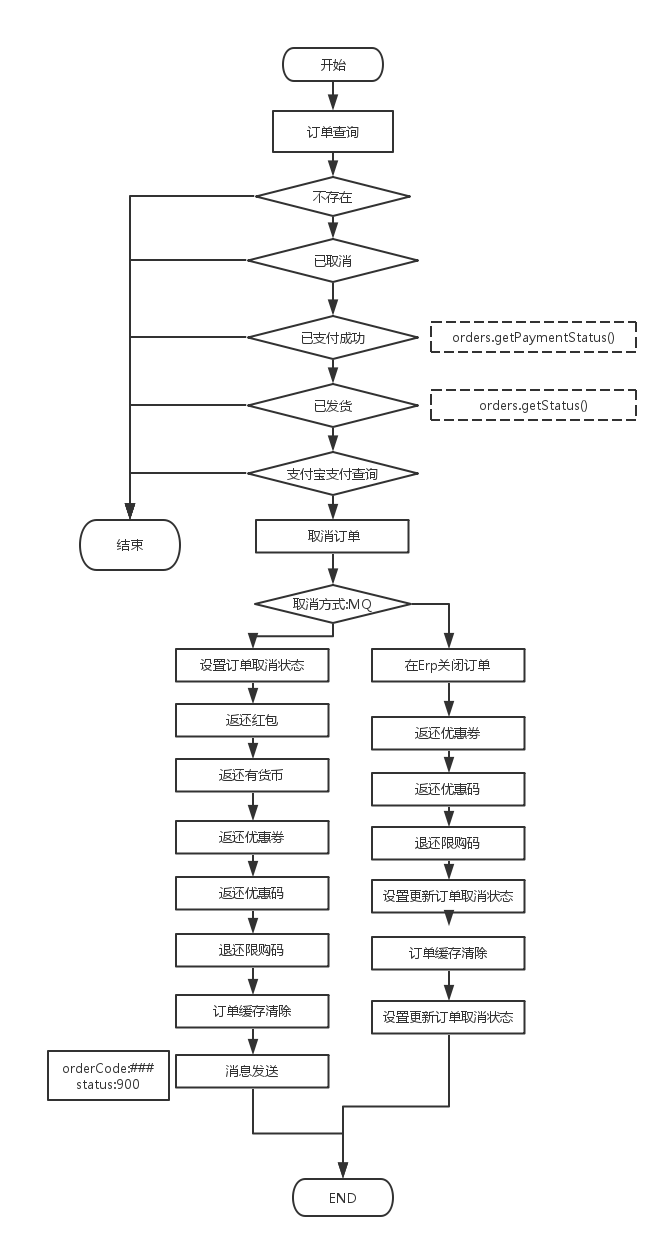
3.MQ方式的消费者在哪?

取消订单与退货在order模块中，在IYohoOrderService服务类中定义了相关的方法。其大致类图如下：



同样IerpService有RPC和MQ两种实现方式。

退货流程大致如下:



换货流程：待续

#### Platform模块批量导入导出接口

上周使用了Platform模块导出接口，在此做下总结

导入导出接口的使用很简单，

1.实现IbusinessImportService导入或IbusinessExportService导出接口

2.在spring配置文件中batchImportBusiness或batchExportBusiness的map中注入导入导出接口实现类，定义key以便后续前端请求使用。

3.1 对于导入定义导入bean，注解导入字段@BatchImportFieldd(index = #)

3.2 对于导出定义导出bean，注解导出字段@BatchExportField(name = "列名")

实现原理：

BatchImportController与BatchExportController定义了统一的导入导出请求路径，统一请求方式有利于前端与后台的开发

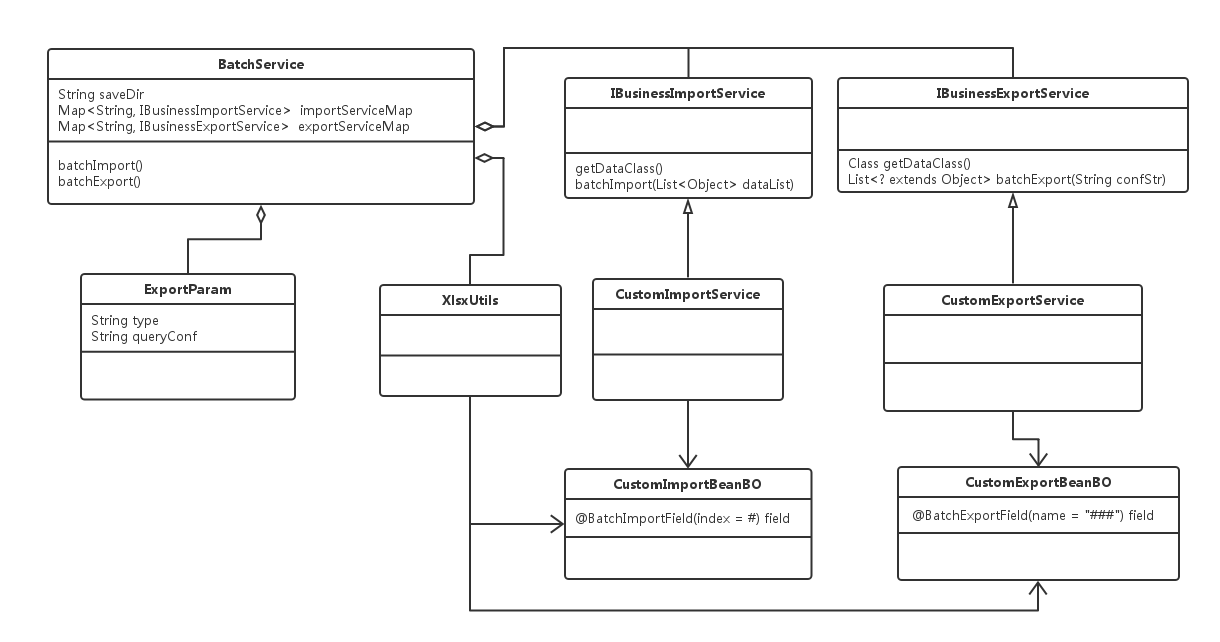
导入 /batch/import

导出 /batch/export

下载 /batch/download

导入导出控制器内部调用BatchService类的具体导入导出实现方法。

BatchService的大致类图如下所示:



BatchService的两个map由spring注入具体的导入导出服务，在spring配置文件中注入到map容器中，如下所示：



BatchService在调用batchImport或者batchExport时，首先第一步都会根据type获取具体的导入导出服务实现类，导入导出控制器接收前端的参数type，并传递给BatchService

导入实现流程

1. createLocalFile

2. XlsxUtils解析导入文件->List<Object>

3.调用具体导入实现类的batchImport进行业务处理。

导出实现流程

1. createLocalFile

2.调用具体导出实现类的batchExport进行业务处理，生成数据list。

3. XlsxUtils将需要返回的list数据转换成xls返回

XlsxUtils为list与xls的一个转换工具，导入导出对象的注解与xls的列具有对应关系，XlsxUtils正是基于这映射关系进行转换，同时使用了apache poi的excel库。

#### Yoho-error模块

ServiceGlobalExceptionHandler：全局服务异常处理

ServiceException不需要捕获，会由该handler统一处理。Handler会在httpResponse消息头内设置错误码和错误消息返回前端.

错误码定义如下：

在抛出ServiceException时，需要设置对应的错误码和消息

