## 网关退款设计

### 一．退款场景

1. 各种业务原因，财务后台操作某笔交易退款
2. 用户超付系统自动进行退款

### 二．退款接口以及退款明细表

* **退款接口方法名**：refundOrderRecord

**接口入参**

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 是否必填 | 描述 |
| serviceId | String | 是 | 业务类型 |
| orderId | String | 是 | 订单号 |
| channelSerialNum | String | 是 | 支付交易号 |
| refundAmt | String | 是 | 退款金额 |
| financialOrderId | String | 是 | 融资单号 |

**接口出参**

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 描述 |
| / | Void | / |

* **退款明细表：pay\_order\_record\_refund**

|  |  |
| --- | --- |
| Field | Comment |
| id | / |
| created\_at | / |
| created\_by | / |
| deleted | / |
| updated\_at | / |
| updated\_by | / |
| version | 版本号 |
| channel\_refund\_serial\_num | 订单退款交易流水号 |
| channel\_serial\_num | 订单支付交易流水号 |
| payr\_order\_record\_item\_id | 支付明细单Id |
| refund\_status | 01-退款申请中,02-已退款，03-退款失败 |
| refund\_amt | 退款金额 |
| financial\_order\_id | 融资单号 |

* **支付明细表：pay\_order\_record\_item （增加当前订单已退款金额字段）**

|  |  |
| --- | --- |
| has\_refund\_amt | 已退款金额 |

### 三．退款流程

* **概述**

1. 支付网关(不考虑超付)对每一笔订单只存在一个支付明细单，一个支付流水单号.则网关退款对订单**部分退款**会记录退款的融资单号以及退款金额，由**支付明细单ID、订单支付流水号、融资单号**可确定唯一退款单，避免订单**部分退款**发生重复退款.
2. 网关与支付宝的退款交易，支付宝的**部分退款**请求报文需要加入**out\_request\_no** 参数，标识一次退款请求，同一笔交易多次退款需要保证唯一；此参数在网关可以使用**融资单号**保证唯一.
3. 银联退款交易状态确定需要依赖退款后台通知，而银联后台通知退款返回报文可以获取**支付明细单ID、支付流水号**，无法确定具体退款单.**解决方案**：利用银联退款请报文的**保留域**，将融资单号放入保留域中，则退款后台通知可以获取到保留域中**融资单号**，从而确定要更新的退款单.
4. 支付明细单会记录退款总金额，用于判断部分退款金额是否超出剩余退款额度.

* **财务后台退款**

对于业务后台对某笔订单发起退款，支付网关定义了退款接口com.zhucai.payment.service.dubbo.PayOrderRecordService.refundOrderRecord ,业务端后台调用接口即可完成退款，若退款无异常，则退款成功.

**具体实现：**

* 1. 在paymentPlatform的com.zhucai.payment.service.dubbo.impl包PayOrderRecordServiceImpl类中实现了refundOrderRecord退款接口。在方法中首先根据订单号、业务id以及支付流水号查询成功支付的支付明细单.判断支付明细单是否存在，若不存在则抛出异常；

*存在支付明细单，则判断部分退款金额是否超出剩余退款额度；若未超出，则根据****支付明细单ID、以及融资单号****查询确定是否存在退款成功或正在退款的退款单， 若不存在则获取支付明细单PayMethodCode调用相应渠道的退款实现类进行退款.*

* 1. 在com.zhucai.payment.service.RefundService 类中定义了refundOrder方法，其有两个实现类分别为支付宝（AlipayRefundServiceImpl）与银联（UnionpayRefundServiceImpl）第三方退款实现类，方法入参为支付明细单ID、支付流水号、退款金额、融资单号，退款成功则返回退款单（PayRecordRefundEntity）.

1. AlipayRefundServiceImpl：请求支付宝退款实现类，判断退款响应结果是否成功.若退款成功，则将相应数据封装到退款单对象返回.
2. UnionpayRefundServiceImpl：请求银联退款实现类，判断银联退款响应的respCode，根据respCode判断退款交易是否成功受理。

若退款返回respCode为00，表明银联退款交易成功受理，则返回退款单对象，定义退款状态为**退款中**(01).

若退款返回respCode为03、04、05、12，则需发起银联交易状态查询交易确定交易状态， **交易状态查询结果返回**的respCode为00时且被查询交易的应答码origRespCode=00\A6，则退款交易成功,返回退款状态为**退款成功(02)**的退款单;若respCode为00，且origRespCode = 03\04\05，需后续再次发起交易状态查询确定交易是否成功，回退款状态为**退款中(01)**的退款单；

返回**退款其他状态码**，则表明退款异常，并抛出异常.

* 1. ***退款成功****则返回退款单，调用支付网关内部接口保存或更新****退款单****.并更新****支付明细单****退款金额。*
  2. ***银联退款后台通知****，根据返回报文，从保留域中取出融资单号ID，退款成功则更新对应退款单状态为退款成功，以及支付明细单退款金额；退款失败，则更新退款单状态为退款失败！*
* **超付系统订单自动退款**

系统对超付订单都在支付明细单（pay\_order\_record\_item）中is\_excess\_payment字段记录当前支付明细单是否超付，由此可以查到所有超付的明细单数据。

**具体实现：**

在paymentPlatform的com.zhucai.payment.service.cycletask 中RefundOrderService类定义了两个方法完成超付订单退款。其中私有方法getRefundOrder 查询系统超付订单以及退款失败的订单，*然后遍历超付订单依次调用相应渠道的退款接口，完成超付订单退款*；而refundOrderTask则循环任务调度方法，每隔10分钟循环调用getRefundOrder方法，从而保证系统**超付订单**能自动退款以及**退款失败订单**退款成功.

问题：