

平安银行

跨行支付接口文档

(Java 版)

平安银行软件开发中心

2014 年6 月24 日

文档修改记录

| 版本 | 日期 | 说明 | 编写者 | 审核者 |
|------|------------|---|-----|-----|
| V1.0 | 2013-6-6 | 定稿 | 梁高吉 | 及晨光 |
| V1.1 | 2013-6-17 | 修改 KH0002、KH0003 返回字段和返回集合名称，增加 KH0001 返回接口字段。定义 4.1 支付流程 action 地址 | 梁高吉 | 及晨光 |
| V1.2 | 2013-10-22 | 修订商户对账接口清算日期和交易完成日期说明；新增 4.4 支付接口数据处理说明 | 梁高吉 | 及晨光 |
| V1.3 | 2013-10-29 | 增加 4.4.2 前后台通知区别说明，5.1.1 示例工程说明和 5.1.2 更新文件说明； | 梁高吉 | 及晨光 |
| V1.4 | 2014-05-23 | 优化 | 朱宁 | 及晨光 |
| V1.5 | 2014-06-24 | 增加退款功能 | 朱宁 | 及晨光 |
| V1.6 | 2014-07-02 | 增加单独 B2B、B2C、手机快捷支付入口，详见 3.1。 | 朱宁 | 及晨光 |

说明:本文档中所包含的信息属于机密信息, 如无平安银行跨行支付项目组的书面许可, 任何人都无权复制或利用。

目 录

| | |
|--|----|
| 第一章 概述..... | 5 |
| 1.1 文档说明..... | 5 |
| 1.2 版本 | 5 |
| 1.3 阅读对象..... | 5 |
| 第二章 商户端数据结构..... | 6 |
| 2.1 数据类型..... | 6 |
| 2.1.1 DataField 基本数据定义 | 6 |
| 2.1.2 KeyedCollection 结构化数据定义..... | 7 |
| 2.1.3 IndexedCollection 结构化数组定义..... | 8 |
| 2.2 数据使用开发示例..... | 9 |
| 2.2.1 组装特定的业务对象结构化数据 (KeyedCollection) | 9 |
| 2.2.2 解析返回的业务对象结构化数据 (KeyedCollection) | 9 |
| 第三章 跨行支付功能..... | 11 |
| 3.1 网上支付接口定义..... | 11 |
| 3.2 orig 组成字段 | 12 |
| 3.2.1 订单支付上送组成字段..... | 13 |
| 3.2.2 订单支付返回组成字段..... | 14 |
| 3.3 网上支付数据处理接口..... | 15 |
| 3.3.1 获取签名源数据及签名后数据..... | 15 |
| 3.3.2 解包返回签名源数据..... | 16 |
| 3.3.3 返回数据签名验证..... | 17 |
| 3.4 网上支付接口数据转码..... | 17 |
| 3.4.1 签名源数据及签名转码顺序..... | 18 |
| 3.4.2 前后台通知的接收数据区别..... | 18 |
| 3.4.3 签名源数据及签名后转码示例..... | 18 |
| 第四章 商户对账功能..... | 20 |
| 4.1 商户对账概述..... | 20 |
| 4.2 商户对账交易码(TranCode)..... | 20 |
| 4.3 商户对账交易操作..... | 20 |
| 4.4 商户对账接口定义..... | 21 |
| 4.4.1 (KH0001) 单笔订单状态查询 | 21 |
| 4.4.2 (KH0002) 订单列表信息查询 | 23 |
| 4.4.3 (KH0003) 每日对账单查询 | 24 |
| 第五章 商户退款功能..... | 27 |

| | |
|------------------------------|----|
| 5.1 商户退款功能概述..... | 27 |
| 5.2 商户退款交易码(TranCode)..... | 27 |
| 5.3 商户退款交易操作..... | 27 |
| 5.4 商户退款接口定义..... | 28 |
| 5.4.1 （KH0005）单笔订单退款..... | 28 |
| 5.4.2 （KH0006）订单退款查询接口 | 29 |
| 第六章 商户开发说明..... | 32 |
| 6.1.1 示例工程说明..... | 32 |
| 6.1.2 模拟商户说明..... | 32 |
| 6.1.3 投产需更新文件说明..... | 32 |

第一章 概述

1.1 文档说明

跨行支付接口文档。

1.2 版本

本文档是 JAVA 版。

1.3 阅读对象

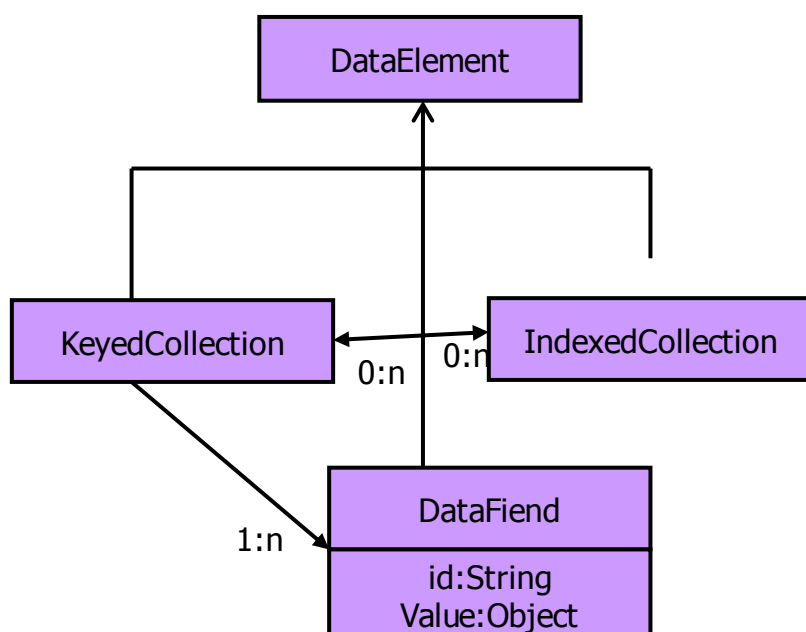
商户系统开发中心技术人员。

平安银行软件开发中心技术人员。

第二章 商户端数据结构

2.1 数据类型

商户客户端数据定义有三种类型：DataField、KeyedCollection、IndexedCollection，分别对应着单个数据、数据集合和数据列表集合三种。三种数据类型都继承于支付网关平台的抽象数据定义类：DataElement。类实现关系如下图所示：



2.1.1 DataField 基本数据定义

DataField 是指单个数据的定义，它的 XML 描述如下：

```
<field id="testField" value="test" type="String"/>
```

一个 DataField 对象有自己的 Id，value 值和类型，类型如果不定义，则缺省为 String 类型，定义时，一般将 DataField 定义在一个 KeyedCollection 结构化对象中，以便在业务逻辑处理中统一进行管理。

DataField 在编程接口上主要提供了 `getValue()` 和 `setValue(Object value)` 两个方法，分别用于获取数据值和设置数据值。

2.1.2 KeyedCollection 结构化数据定义

KeyedCollection 是指集合型的对象，它可以包含多个 DataField 或 IndexedCollection 对象。我们用 KeyedCollection 对象来描述一个特定的业务对象（商户订单查询条件、单笔订单的详情信息、订单列表的信息等等），一个业务对象如单笔订单查询条件可能包含商户 ID、订单号、订单金额等等数据域，这些数据域集中定义在一个 KeyedCollection 对象中，组合代表为一个单笔订单查询条件。KeyedCollection 对象的 XML 描述如下，其中标签 kColl 映射为 KeyedCollection 类实现定义名称。

```
<kColl id="orderQuery">
    <field id="masterId"/>
    <field id="orderId"/>
    <field id="amount"/>
    <iColl id="orderListInfo">
        <kColl>
            <field id="status"/>
            <field id="masterId"/>
        </kColl>
    </iColl>
</kColl>
```

KeyedCollection 主要提供的编程接口是对数据元素的操作，基本接口方法如下所述：

- getDataValue (String dataName)：得到一个数据值
- setDataValue (String dataName, Object value)：设置一个数据值
- getDataElement(String dataName)：得到一个数据对象
- addDataField(String dataName, Object value)：添加一个数据对象 DataFiled,（也有可能是一个 KeyedCollection 对象或者 IndexedCollection 对象）

- addDataDataField(DataField field): 添加一个数据对象
- put(Object key, Object value): 添加一个数据对象,若该对象存在,则设置该对象值;

2.1.3 IndexedCollection 结构化数组定义

IndexedCollection 对象是结构化的数据数组,包含多个重复结构的数据对象,如订单列表的描述,就只能采用 IndexedCollection 对象来进行描述。一条订单记录结构作为 IndexedCollection 中的 kColl,包含在 IndexedCollection 对象内,多条记录的 XML 定义如下:

```
<iColl id="orderListInfo">

    <kColl>

        <field id=" status "/>

        <field id=" masterId "/>

        <field id=" orderId "/>

        <field id=" amount "/>

        <field id=" currency "/>

        <field id=" paydate"/>

        <field id=" remark "/>

    </kColl>

    <kColl>

        <field id=" status "/>

        <field id=" masterId "/>

        <field id=" orderId "/>

        <field id=" amount "/>

        <field id=" currency "/>

        <field id=" paydate"/>

        <field id=" remark "/>

    </kColl>
```


</iColl>

以上描述的是一个订单列表的信息。`IndexedCollection` 对象提供的编程接口也是基于多数据对象的操作，由于它是基于数据数组模型的操作，所有有一些基于标记位的接口方法，如下：

- `getElementAt(int idx)` 取第 `idx` 条数据定义
- `addDataElement (DataElement element)` 添加一条数据

2.2 数据使用开发示例

2.2.1 组装特定的业务对象结构化数据（`KeyedCollection`）

```
//建立客户端实例
PayclientInterfaceUtil util = new PayclientInterfaceUtil ();

//创建业务对象结构化送数据
KeyedCollection inputKcoll = new KeyedCollection("inputKcoll");
//将单个 DataField 字段添加到 kcoll，(如商户 id 和订单流水号)
//具体上送字段见商户对账接口定义上送数据
inputKcoll.put("masterId ", "20000000833");
inputKcoll.put("orderId ", "20000000833201306011000000001");
```

2.2.2 解析返回的业务对象结构化数据（`KeyedCollection`）

```
//创建返回的业务对象结构化送数据（KeyedCollection ）
KeyedCollection outputKcoll = new KeyedCollection("outputKcoll");
//对返回的业务对象结构化送数据赋值
```

```
outputKcoll = util.execute(inputKcoll,businessCode);

//从结果对象中获取单个数据的值，变量名参见接口文档。

String  masterId = (String) outputKcoll.getDataValue("masterId ");

//若返回对象中有数据列，取值方法如下,变量名参见接口文档。

IndexedCollection icoll = (IndexedCollection)
outputKcoll.getDataElement("iOrderList ") ;

//循环取出各条记录值，并处理，记录总数为 icoll.size()

for( int i=0; i<icoll.size(); i++){

    //取出 index 为 i 的一条记录，结构为 KeyedCollection

    KeyedCollection kcoll = (KeyedCollection)icoll.getElementAt(i);

    //处理 index 为 i 的记录内的各字段，具体包含的字段参考接口文档

    String date = (String)kcoll.getDataValue("date ");

    String status = (String)kcoll.getDataValue("status ");

    String orderId = (String)kcoll.getDataValue("orderId ")

}

}
```

第三章 跨行支付功能

3.1 网上支付接口定义

商户在支付页面，按照跨行支付的规则处理原始订单数据，采用 https 通讯协议，通过 form 表单提交数据到跨行支付网关进行支付。

```
<form method="post" action=
“https://testebank.sdb.com.cn:461/corporbank/khPayment.do“ >

  <input type=hidden name="orig" value=" <%=orig%>">

  <input type=hidden name="sign" value=" <%= sign%>">

  <input type=hidden name="returnurl" value=" xxxx3">

  <input type=hidden name="NOTIFYURL" value=" xxxx4">

</form>
```

通用测试环境地址为：

<https://testebank.sdb.com.cn:461/corporbank/khPayment.do>;

单独 B2B 测试环境地址为：

https://testebank.sdb.com.cn:461/corporbank/khPayment_B2B.do;

单独 B2C 测试环境地址为：

https://testebank.sdb.com.cn:461/corporbank/khPayment_B2C.do;

手机快捷支付测试环境地址为：

https://testebank.sdb.com.cn:461/corporbank/khPayment_wapQuickPay.do;

通用生产环境地址为：

<https://ebank.sdb.com.cn/corporbank/khPayment.do>;

单独 B2B 生产环境地址为：

https://ebank.sdb.com.cn/corporbank/khPayment_B2B.do;

单独 B2C 生产环境地址为：

https://ebank.sdb.com.cn/corporbank/khPayment_B2C.do;

手机快捷支付生产环境地址为：

https://ebank.sdb.com.cn/corporbank/khPayment_wapQuickPay.do。

银行返回结果同样采用 https 通讯协议，通过 form 提交数据，发给商户上送数据中“returnurl”指定的接收页面。

```

<form action= "returnurl " method="post">
    <input type=hidden name="orig" value=" " <%=origData%>">
    <input type=hidden name= "sign" value=" " <%= sign %>">
</form>

```

form 组成规则及说明:

| 表单域名 | 描述 | 必填项 | 备注 |
|-----------|----------------------|-----|------------|
| orig | 商户订单数据组成的原始数据字符串 | 是 | 组成规则参考 3.2 |
| sign | 原始数据字符串的签名 | 是 | 获取方法参考 3.3 |
| returnurl | 商户接收订单支付结果的 URL | 是 | 前台页面通知 |
| NOTIFYURL | 商户接收订单支付结果服务器通知的 URL | 是 | 后台服务器通知 |

说明:

1. retrunurl 和 NOTIFYURL 的字段都是商户指定接收在银行网站在支付成功后, 银行正常交易返回结果的 URL 地址, 这两个字段的区别在于:

returnurl 是在浏览器上由用户点击“通知商户”按钮或点击“支付成功”对话框后触发, 发送 post 请求到 returnurl 地址, 所以如果用户在支付成功后, 不触发通知商户请求, 则商户接收不到返回结果, returnurl 通知以下称为前台通知;

NOTIFYURL 是银行在正常交易后由后台程序发送的一个 post 请求的接收地址, NOTIFYURL 通知以下称之为后台通知, NOTIFYURL 的通知更保险。

2. 网上支付接口中, 使用 form 表单提交的 orig、sign 字段和交易成功后通知的 orig、sign 字段, 需要使用 GBK 编码进行 Base64 转码和 URL 转码, 具体转码方法参见 3.4。

3.2 orig 组成字段

接口字段定义的类型和长度均为平安银行数据库限制长度, 实际传输字段为字符串, 所有字段组成名称区分大小写, 汉字使用 GBK 编码, 字段在组装过程中对顺序不做要求, 但文档中要求的上送字段均需要添加到对应的业务对象结构 (KeyedCollection) 中。

3.2.1 订单支付上送组成字段

| 组成字段 | 类型 | 长度 | 必输项 | 含义 |
|------------|---------|------|-----|--|
| masterId | char | 10 | 是 | 商户号，注意生产环境上要替换成商户自己的生产商户号 |
| orderId | varchar | 26 | 是 | 订单号，严格遵守格式：商户号+8 位日期 YYYYMMDD+8 位流水 |
| currency | char | 3 | 是 | 币种，目前只支持 RMB |
| amount | number | 12,2 | 是 | 订单金额，12 整数，2 小数 |
| objectName | varchar | 200 | 是 | 订单款项描述（商户自定） |
| paydate | varchar | 14 | 是 | 下单时间， YYYYMMDDHHMMSS |
| validtime | number | 10 | 是 | 订单有效期(毫秒),0 不生效 |
| remark | varchar | 500 | 否 | 备注字段（商户自定） |

添加订单信息到业务对象结构（KeyedCollection）中，具体代码详见 DemoPayment.java。

```
KeyedCollection inputOrig = new KeyedCollection("inputOrig");  
//商户号，注意生产环境上要替换成商户自己的生产商户号  
inputOrig.put("masterId","2000311146");  
//订单号，严格遵守格式：商户号+8 位日期 YYYYMMDD+8 位流水  
inputOrig.put("orderId","2000311146"+datetamp+getOrderId());  
//币种，目前只支持 RMB  
inputOrig.put("currency","RMB");  
//订单金额，12 整数，2 小数  
inputOrig.put("amount",count*price);
```

```

//下单时间，YYYYMMDDHHMMSS
inputOrig.put("paydate",timestamp);

//备注字段（商户自定）
inputOrig.put("remark","2000311146");

//订单款项描述（商户自定）
inputOrig.put("objectName","KHpaygate");

//订单有效期(秒)，0 不生效
inputOrig.put("validtime","0");

```

3.2.2 订单支付返回组成字段

| 组成字段 | 类型 | 长度 | 含义 |
|------------|---------|------|---------------------------|
| status | char | 2 | 订单状态，只返回‘01’，表示成功 |
| date | varchar | 14 | 支付完成时间， YYYYMMDDHHMMSS |
| charge | number | 12,2 | 订单手续费金额， 12 整数，2 小数 |
| masterId | char | 10 | 商户号 |
| orderId | varchar | 26 | 订单号 |
| currency | char | 3 | 币种，目前只支持 RMB |
| amount | number | 12,2 | 订单金额，12 整数，2 小数 |
| objectName | varchar | 200 | 款项描述 |
| paydate | varchar | 14 | 下单时间， YYYYMMDDHHMMSS |
| validtime | number | 10 | 订单有效期(秒)，0 不生效 |
| remark | varchar | 500 | 备注字段 |

3.3 网上支付数据处理接口

商户企业端提供的方法全部为 `com.sdb.payclient.core.PayclientInterfaceUtil` 的公共方法。

3.3.1 获取签名源数据及签名后数据

KeyedCollection getSignData (KeyedCollection inputOrig) throws Exception

功能：商户根据订单支付接口约定，将订单支付上送字段组装到 `KeyedCollection` 中，调用签名方法获取签名源数据及签名后数据组成的特定业务对象。

参数：`KeyedCollection` 类，封装了业务数据。

返回值：`KeyedCollection` 类，封装了签名源数据 `orig` 和签名后数据 `sign`。

```
//建立客户端实例
PayclientInterfaceUtil util = new PayclientInterfaceUtil ();
KeyedCollection signDataput = new KeyedCollection("signDataput");
//inputOrig 的组成详见 3.2.1 订单支付上送组成字段
signDataput = util.getSignData(inputOrig);
//获取签名源数据
String orig = (String) signDataput.getDataValue("orig ");
//获取签名后数据
String sign= (String) signDataput.getDataValue("sign");
```

具体代码可以参照 `DemoPayment.java`。

3.3.2 解包返回签名源数据

KeyedCollection parseOrigData(String origData) throws Exception

功能：商户根据订单支付接口约定，将接收的签名源数据解析到特定业务对象 KeyedCollection 中。

参数：origData，银行返回的签名源数据。

返回值：KeyedCollection 类，封装了返回数据集合的数据结构。

//建立客户端实例

```
PayclientInterfaceUtil util = new PayclientInterfaceUtil ();
```

//获取银行返回签名源数据，解析到特定的业务对象中

```
String orig = request.getParameter("orig");
```

```
KeyedCollection origKcoll = new KeyedCollection("output ");
```

```
output = util.parseOrigData (orig);
```

//处理业务对象，其包含的字段详见 3.2.2 订单支付返回组成字段

```
System.out.println("---订单详细信息---"+output);
```

```
System.out.println("---订单状态---"+output.getDataValue("status"));
```

```
System.out.println("---支付完成时间---"+output.getDataValue("date"));
```

```
System.out.println("---手续费金额---"+output.getDataValue("charge"));
```

```
System.out.println("---商户号---"+output.getDataValue("masterId"));
```

```
System.out.println("---订单号---"+output.getDataValue("orderId"));
```

```
System.out.println("---币种---"+output.getDataValue("currency"));
```

```
System.out.println("---订单金额---"+output.getDataValue("amount"));
```

```
System.out.println("---下单时间---"+output.getDataValue("paydate"));
```

```
System.out.println("---商品描述---"+output.getDataValue("objectName"));
```

```
System.out.println("---订单有效期---"+output.getDataValue("validtime"));
```



```
System.out.println("---备注---"+output.getDataValue("remark"));
```

具体代码可以参照 DemoReturn.java 和 DemoNotify.java。

3.3.3 返回数据签名验证

boolean verifyData(String origData,String signData)throws Exception

功能：验证银行端返回的签名源数据和签名后数据，返回验签结果。

参数：

- 1、 origData，银行返回的签名源数据。
- 2、 signData，银行返回的签名后数据。

返回值：boolean 值，为 true 表示验签成功，为 false 表示验签失败。

//建立客户端实例

```
PayclientInterfaceUtil util = new PayclientInterfaceUtil ();
```

//获取银行返回签名源数据及签名后数据

```
String orig = request.getParameter("orig");
```

```
String sign = request.getParameter("sign");
```

//处理返回数据，返回数据包含的字段见接口返回组成字段

```
boolean result = util.verifyData(orig,sign);
```

具体代码可以参照 DemoReturn.java 和 DemoNotify.java。

3.4 网上支付接口数据转码

为了避免网上支付交易中传输的数据出现乱码等编码问题，需要对交易签名数据进行转码和解码的相关处理。

3.4.1 签名源数据及签名转码顺序

网上支付在 form 表单提交订单数据前要对签名源数据 orig 及签名后数据 sign 字段先做 Base64Encode 转码，然后进行 URLEncoder 转码；

接收到支付成功通知后，需要对返回的数据 orig 和 sign 字段，先进行 URLDecoder 解码，然后进行 Base64Decode 解码；

3.4.2 前后台通知的接收数据区别

前台通知 POST 的支付返回字段是通过客户浏览器 URL 跳转的方式接收的，接收的通知数据已经做过了 URLEncoder 转码，所以商户端程序需要再调用 URLDecoder 方法解码，接着再进行 Base64Decode 解码；而后台通知数据直接进行 Base64Decode 解码即可。

3.4.3 签名源数据及签名后转码示例

上送数据处理，具体代码可以参照 **DemoPayment.java**：

```
//设置转码编码和建立实例；

String encoding = "GBK";

com.sdb.payclient.core.PayclientInterfaceUtil util = new
com.sdb.payclient.core.PayclientInterfaceUtil();

//签名源数据先做 Base64Encode 转码

orig = util.Base64Encode(orig,encoding);

sign = util.Base64Encode(sign,encoding);

//签名源数据及签名后数据做 URL 转码

orig = java.net.URLEncoder.encode(orig, encoding);

sign = java.net.URLEncoder.encode(sign, encoding);
```

返回数据处理，具体代码可以参照 **DemoReturn.java** 和 **DemoNotify.java**：

```
//设置转码编码和建立实例；

String encoding = "GBK";
```

```
com.sdb.payclient.core.PayclientInterfaceUtil util = new  
com.sdb.payclient.core.PayclientInterfaceUtil();  
  
//签名源数据先做 URLDecoder 解码，后台通知不用做  
  
orig = java.net.URLDecoder.decode(orig, encoding);  
sign = java.net.URLDecoder.decode(sign, encoding);  
  
//签名源数据及签名后数据做 Base64Decode 转码  
  
orig = util.Base64Decode(orig,encoding);  
sign = util.Base64Decode(sign,encoding);
```

第四章 商户对账功能

4.1 商户对账概述

商户对账功能是银行提供的供商户系统主动查询订单详情及入账相关信息的一系列相关接口。查询相关页面由商户系统自行设计，查询数据银行系统实时返回。银行系统提供的接口会随着业务需求同步保持更新，如有需要请与平安银行获取最新的接口文档。

4.2 商户对账交易码(TranCode)

采用 https 通讯协议，商户通过银行指定的 url 发送交易请求，并接受银行返回的结果数据。以下是系统提供的接口交易码（功能交易码区分大小写）及对应的功能名称：

| 交易码（TranCode） | 功能名称 |
|---------------|------------|
| KH0001 | 单笔订单信息查询 |
| KH0002 | 订单列表信息查询 |
| KH0003 | 每日对外转账信息查询 |

4.3 商户对账交易操作

商户对账操作调用的方法全部为 com.sdb.payclient.core.PayclientInterfaceUtil 的公共方法。

`KeyedCollection execute (KeyedCollection input,String businessCode)
throws CsiiException`

功能：根据商户系统的对账需求，将对应的查询条件向支付网关传送对应的交易代码及查询条件，并将查询结果数据返回 KeyedCollection 对象中。如果方

法抛出 `CsiiException`，用户通过捕捉 `CsiiException` 然后调用其实例的 `getMessage()` 方法取得错误的描述。

参数：

1、`KeyedCollection` 类：封装了用于商户对账交易申请数据的结构化数据定义。具体请求数据参见下文中接口交易描述中的上送数据；

2、`businessCode`：传送商户对账请求的接口交易码。

返回值：

`KeyedCollection` 类：封装了用于商户对账交易申请返回数据的结构化数据类型数据类。具体返回字段参见下文中接口描述中的返回数据。查询记录不存在的情况，用户可以通过对于返回 `KeyedCollection` 中的 `errorCode` 和 `errorMsg` 字段取得相应的错误代码和错误信息描述。

具体代码可以参照 `DemoKH0001.java`、`DemoKH0002.java` 和 `DemoKH0003.java`。

4.4 商户对账接口定义

接口字段定义的类型和长度均为平安银行数据库限制长度，实际传输字段为字符串，所有字段组成名称区分大小写，汉字使用 `GBK` 编码，字段在组装过程中对顺序不做要求，但文档中要求的上送字段均需要添加到对应的业务对象结构（`KeyedCollection`）中。

4.4.1 （KH0001）单笔订单状态查询

4.4.1.1 单笔订单查询上送数据：

| 组成字段 | 类型 | 长度 | 必填项 | 含义 |
|-----------------------|----------------------|----|-----|---|
| <code>masterId</code> | <code>char</code> | 10 | 是 | 商户号，注意生产环境上要替换成商户自己的生产商户号 |
| <code>orderId</code> | <code>varchar</code> | 26 | 是 | 订单号，严格遵守格式：商户号+8 位日期 <code>YYYYMMDD</code> +8 位流水 |

4.4.1.2 单笔订单查询返回数据：

| 组成字段 | 类型 | 长度 | 含义 |
|------------|---------|------|----------------------|
| errorCode | varchar | 8 | 错误返回相应的错误码，正常返回为空 |
| errorMsg | varchar | 100 | 错误码对应的错误说明，正常返回为空 |
| status | char | 2 | ‘01’表成功，只返回成功 |
| date | varchar | 14 | 支付清算日期，可为空 |
| charge | number | 12,2 | 订单手续费金额，12 整数，2 小数 |
| masterId | char | 10 | 收费企业 ID |
| orderId | varchar | 26 | ID+YYYYMMDD+8 位流水 |
| currency | char | 3 | 目前只支持 RMB |
| amount | number | 12,2 | 订单金额，最大 12 位整数，2 位小数 |
| objectName | varchar | 200 | 款项描述（不可含分割符） |
| paydate | varchar | 14 | YYYYMMDDHHMMSS |
| validtime | number | 10 | 订单有效期(秒)，0 不生效 |
| remark | varchar | 500 | 备注字段（不可含分割符） |
| settleflg | char | 1 | 本金清算标志‘1’已清算，‘0’待清算 |
| settletime | varchar | 14 | 本金清算时间 |
| chargeflg | char | 1 | 手续费清算标志‘1’已清算，‘0’待清算 |
| chargetime | varchar | 14 | 手续费清算时间 |

具体使用方法可以参考 DemoKH0001.java。

注：支付清算日期为用于区分商户本金的清算日，一般情况下与支付完成日期相同，但是通过支付宝支付的订单支付完成日期与清算日期存在不一致的情况；

4.4.2 （KH0002）订单列表信息查询

4.4.2.1 订单列表查询上送数据

| 组成字段 | 类型 | 长度 | 必输项 | 含义 |
|-----------|---------|----|-----|----------------------------------|
| masterId | char | 10 | 是 | 商户号, 注意生产环境上要替换成商户自己的生产商户号 |
| beginDate | varchar | 14 | 是 | 查询开始时间（支付完成时间） YYYYMMDDHHMMSS |
| endDate | varchar | 14 | 是 | 查询结束时间（支付完成时间） YYYYMMDDHHMMSS |

4.4.2.2 订单列表查询返回数据

| 组成字段 | | 类型 | 长度 | 含义 |
|-------------------------|--------|-------------------------------------|------|------------------------------------|
| errorCode | | varchar | 8 | 错误返回相应的错误码, 正常返回为空 |
| errorMsg | | varchar | 100 | 错误码对应的错误说明, 正常返回为空 |
| sumAmount | | number | 12,2 | 总金额, 最大 12 位整数, 2 位小数 |
| sumCount | | number | 8 | 交易总笔数 |
| iOrderListDetail | | 数据集合 IndexedCollection（以下字段属于集合内字段） | | |
| KeyedCollection | status | char | 2 | ‘01’表成功, 只返回成功 |
| | date | varchar | 14 | 订单结算日期（结算日期, 支付宝平台可能为空, 需要对账完成后确定） |
| | charge | number | 12,2 | 订单手续费金额, 12 整数, 2 小数 |

| | | | | |
|--|------------|---------|------|---------------------------|
| | masterId | char | 10 | 收费企业 ID |
| | orderId | varchar | 26 | ID+YYYYMMDD+8 位流水 |
| | currency | char | 3 | 目前只支持 RMB |
| | amount | number | 12,2 | 订单金额，最大 12 位整数，2 位小数 |
| | objectName | varchar | 200 | 款项描述（不可含分割符） |
| | paydate | varchar | 14 | YYYYMMDDHHMMSS |
| | validtime | number | 10 | 订单有效期(秒)，0 不生效 |
| | remark | varchar | 500 | 备注字段（不可含分割符） |
| | settleflg | char | 1 | 订单本金清算标志‘1’已清算，‘0’待清算 |
| | settletime | varchar | 14 | 本金清算时间 YYYYMMDDHHMMSS |
| | chargeflg | char | 1 | 手续费清算标志‘1’已清算，‘0’待清算 |
| | chargetime | varchar | 14 | 手续费清算时间 YYYYMMDDHHMMSS |

具体使用方法可以参考 DemoKH0002.java。

4.4.3 （KH0003）每日对账单查询

4.4.3.1 每日对账单查询上送数据

| 组成字段 | 类型 | 长度 | 含义 |
|----------|---------|----|------------------|
| masterId | char | 10 | 收费企业 ID |
| date | varchar | 8 | 对账日期，格式：YYYYMMDD |

4.4.3.2 每日对账单查询返回数据

| 组成字段 | | 类型 | 长度 | 含义 |
|------------------|------------|--------------------------------------|------|------------------------|
| errorCode | | varchar | 8 | 错误返回相应的错误码, 正常返回为空 |
| errorMsg | | varchar | 100 | 错误码对应的错误说明, 正常返回为空, |
| sumAmount | | number | 12,2 | 总金额, 最大 12 位整数, 2 位小数 |
| sumCount | | number | 8 | 交易总笔数 |
| iOrderListDetail | | 数据集合 IndexedCollection (以下字段属于集合内字段) | | |
| KeyedCollection | status | char | 2 | ‘01’表成功, 只返回成功 |
| | date | varchar | 14 | 支付完成日期 (结算日期) |
| | charge | number | 12,2 | 订单手续费金额, 12 整数, 2 小数 |
| | masterId | char | 10 | 收费企业 ID |
| | orderId | varchar | 26 | ID+YYYYMMDD+8 位流水 |
| | currency | char | 3 | 目前只支持 RMB |
| | amount | number | 12,2 | 订单金额, 最大 12 位整数, 2 位小数 |
| | objectName | varchar | 200 | 款项描述 |
| | paydate | varchar | 14 | YYYYMMDDHHMMSS |
| | validtime | number | 10 | 订单有效期(秒), 0 不生效 |
| | remark | varchar | 500 | 备注字段 (不可含分割符) |
| | settleflg | char | 1 | 订单本金清算 标志‘1’已清算, ‘0’待 |

| | | | | |
|--|------------|---------|----|---------------------------|
| | | | | 清算 |
| | settletime | varchar | 14 | 本金清算时间 YYYYMMDDHHMMSS |
| | chargeflg | char | 1 | 手续费清算标志‘1’已清算，‘0’待清算 |
| | chargetime | varchar | 14 | 手续费清算时间 YYYYMMDDHHMMSS |

具体使用方法可以参考 DemoKH0003.java。

第五章 商户退款功能

5.1 商户退款功能概述

商户退款功能是银行提供给供商户系统的对已经支付订单发起的原路退款功能和退款状态查询及通知的接口或者服务。银行系统提供的接口会随着业务需求同步保持更新，如有需要请与平安银行获取最新的接口文档，**注意：目前只有PC端B2C交易才具备退款功能，而且只能进行单笔全额退款，请知悉。**

5.2 商户退款交易码(TranCode)

采用 https 通讯协议，商户通过银行指定的 url 发送交易请求，并接受银行返回的结果数据。以下是系统提供的接口交易码（**功能交易码区分大小写**）及对应的功能名称：

| 交易码（TranCode） | 功能名称 |
|---------------|----------|
| KH0005 | 单笔订单退款申请 |
| KH0006 | 退款订单查询 |

5.3 商户退款交易操作

商户退款操作调用的方法全部为 `com.sdb.payclient.core.PayclientInterfaceUtil` 的公共方法。

```
KeyedCollection execute (KeyedCollection input,String businessCode)
throws CsiiException
```

功能：根据商户系统的退款需求，将对应的查询条件向支付网关传送对应的交易代码及查询条件，并将查询结果数据返回 `KeyedCollection` 对象中。如果方法抛出 `CsiiException`，用户通过捕捉 `CsiiException` 然后调用其实例的 `getMessage()` 方法取得错误地描述。

参数：

1、KeyedCollection 类：封装了用于商户对账交易申请数据的结构化数据定义。具体请求数据参见下文中接口交易描述中的上送数据；

2、businessCode：传送商户对账请求的接口交易码。

返回值：

KeyedCollection 类：封装了用于商户对账交易申请返回数据的结构化数据类型数据类。具体返回字段参见下文中接口描述中的返回数据。查询记录不存在的情况，用户可以通过对于返回 KeyedCollection 中的 errorCode 和 errorMsg 字段取得相应的错误代码和错误信息描述。

具体代码可以参照 DemoKH0005.java 和 DemoKH0006.java。

5.4 商户退款接口定义

接口字段定义的类型和长度均为平安银行数据库限制长度，实际传输字段为字符串，所有字段组成名称区分大小写，汉字使用 GBK 编码，字段在组装过程中对顺序不做要求，但文档中要求的上送字段均需要添加到对应的业务对象结构（KeyedCollection）中。

5.4.1 （KH0005）单笔订单退款

5.4.1.1 单笔订单退款上送数据：

| 组成字段 | 类型 | 长度 | 必输项 | 含义 |
|------------|---------|------|-----|-----------------------------------|
| masterId | char | 10 | 是 | 商户号，注意生产环境上要替换成商户自己的生产商户号 |
| refundNo | varchar | 26 | 是 | 退款单号，严格遵守格式：商户号+8位日期YYYYMMDD+8位流水 |
| orderId | varchar | 26 | 是 | 原订单号，严格遵守格式：商户号+8位日期YYYYMMDD+8位流水 |
| currency | char | 3 | 是 | 币种，目前只支持 RMB |
| refundAmt | number | 12,2 | 是 | 退款金额，12 整数，2 小数，必须等于原订单金额 |
| objectName | varchar | 200 | 否 | 退款原因（商户自定） |
| remark | varchar | 500 | 否 | 备注字段（商户自定） |

| | | | | |
|-----------|---------|-----|---|--|
| NOTIFYURL | varchar | 100 | 是 | 商户接收订单退款结果服务器异步通知的 url，支付平台通知银行，则银行通知商户，否则不通知。 |
|-----------|---------|-----|---|--|

5.4.1.2 单笔订单退款返回数据：

| 组成字段 | 类型 | 长度 | 含义 |
|-----------|---------|------|--------------------------------|
| errorCode | varchar | 8 | 错误返回相应的错误码，正常返回为空 |
| errorMsg | varchar | 100 | 错误码对应的错误说明，正常返回为空 |
| status | char | 2 | 01：接收成功，退款是否成功请以异步通知结果为准；02：失败 |
| masterId | char | 10 | 退费企业 ID |
| refundNo | varchar | 26 | 退款单号 |
| orderId | varchar | 26 | 原订单号 |
| currency | char | 3 | 币种 |
| refundAmt | number | 12,2 | 退款金额 |

具体使用方法可以参考 DemoKH0005.java。

注：退款申请是接收商户退款的申请，具体到账日期依不同银行情况而定；

5.4.2 （KH0006）订单退款查询接口

5.4.2.1 订单退款查询接口上送数据：

| 组成字段 | 类型 | 长度 | 必填项 | 含义 |
|-----------|---------|----|-----|--------------------------------|
| masterId | char | 10 | 是 | 商户号，注意生产环境上要替换成商户自己的生产商户号 |
| beginDate | varchar | 14 | 是 | 查询开始时间（退款请求接收时间）YYYYMMDDHHMMSS |
| endDate | varchar | 14 | 是 | 查询结束时间（退款请求接收时间）YYYYMMDDHHMMSS |

5.4.2.2 订单退款查询接口返回数据：

| 组成字段 | | 类型 | 长度 | 含义 |
|-------------------------|--------------|-------------------------------------|------|----------------------------------|
| errorCode | | varchar | 8 | 错误返回相应的错误码，正常返回为空 |
| errorMsg | | varchar | 100 | 错误码对应的错误说明，正常返回为空 |
| sumAmount | | number | 12,2 | 总金额，最大 12 位整数，2 位小数 |
| sumCount | | number | 8 | 交易总笔数 |
| iOrderListDetail | | 数据集合 IndexedCollection（以下字段属于集合内字段） | | |
| KeyedCollection | masterId | char | 10 | 退费企业 ID |
| | refundNo | varchar | 26 | 退款单号 |
| | orderId | varchar | 26 | 原订单号 |
| | refundTime | varchar | 14 | 退款请求接收时间 |
| | currency | char | 3 | 币种 |
| | refundAmt | number | 12,2 | 退款金额 |
| | refundStatus | char | 2 | 退款状态： 01 成功；02 失败；00 处理中 |
| | settleflg | char | 1 | 退款本金扣收标志： 0：未扣款；1：已扣款；2：扣款中 |
| | refundFlag | varchar | 14 | 退款失败本金退还标志： 0：未退款；1：已退款；2：处理中 |
| | errorCode | varchar | 8 | 错误返回相应的错误码，正常返回为空 |

| | | | | |
|--|----------|---------|-----|-------------------|
| | errorMsg | varchar | 100 | 错误码对应的错误说明，正常返回为空 |
|--|----------|---------|-----|-------------------|

具体使用方法可以参考 DemoKH0006.java。

第六章 商户开发说明

6.1.1 示例工程说明

开发测试工程 DemoPayClient 中提供了商户接入跨行支付网关需要引用的 jar 包（在 lib 下）、测试商户证书 merchant.jks（商户号为”2000311146”）、配置文件 payclient.properties（修改 cafile 为 merchant.jks 的绝对路径）以及所有接口的示例程序。

6.1.2 模拟商户说明

商户可以参考模拟商户 khPayment_merchant_demo 程序，并结合自己的业务需求，开发商户网站。

6.1.3 投产需更新文件说明

1、投产时，商户必须联系我行发放下载生产证书所需的参考号和授权码（在开通业务时，银行应该提前给予，只能使用一次，所以可以多申请几个备用，浏览器最好用 ie6），商户自行下载生产公私钥，公钥给予我行，私钥用于制作生产用的 merchant.jks，替换测试用的 merchant.jks，具体的操作指引可以参考[投产重要步骤文件夹](#)；

2、投产时，修改 payclient.properties 文件中的 khPaymentUrl=https://ebank.sdb.com.cn/corporbank/