微信二维码支付技术代理商对接文档

（被扫模式）

北京易付通金服技术有限公司

邱伟

2018-03-17

V2.1

# 业务背景

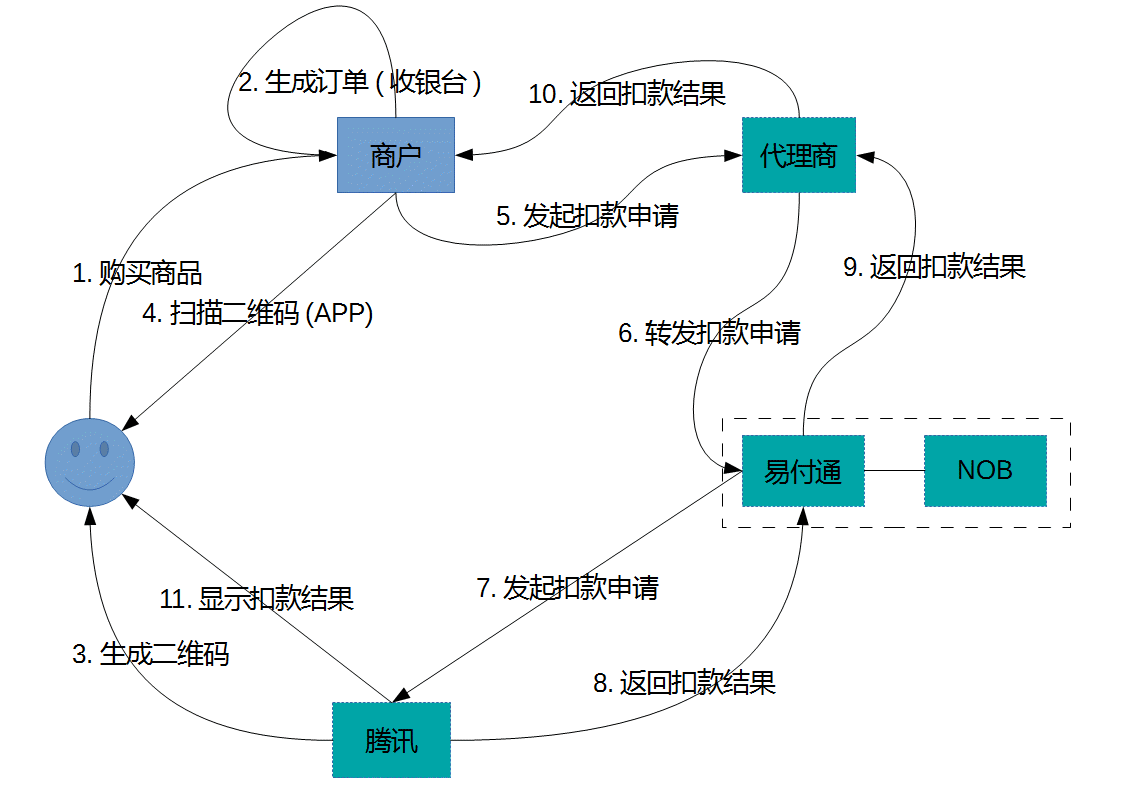
随着支付业务的增长，用户在跨境支付方面的需求也日益增加。特别是基于微信这样大众更易接受的方式，更是迫切需要在国外的落地实施。在此方面，所涉及到的用户群体，如：商务人士、旅游人士、留学生等；所涉及到的商户，如：免税店、大型商超、汽车租赁等。

特别是以上用户群体在境外使用二维码支付方面，更是粘合用户的使用习惯，并大大增加用户的使用体验。同时，也进一步为商户节约运维成本，提升用户与商户之间的粘性。

# 适用读者

本文档适用于代理商技术人员阅读。通过本文档，技术人员能够迅速的与易付通后台进行对接，尽快打通用户通过微信二维码支付（被扫模式）的业务链条。

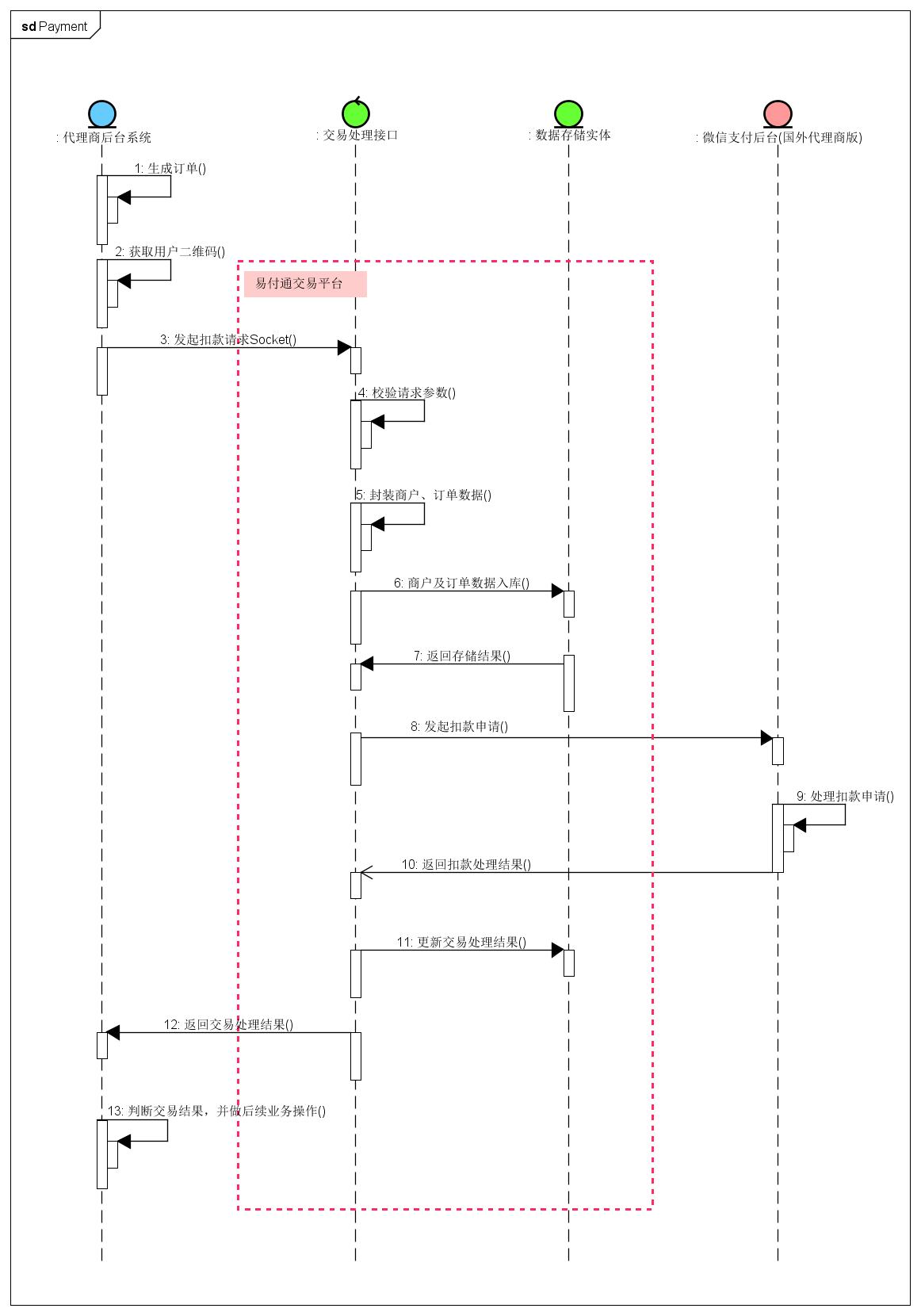
# 业务关系



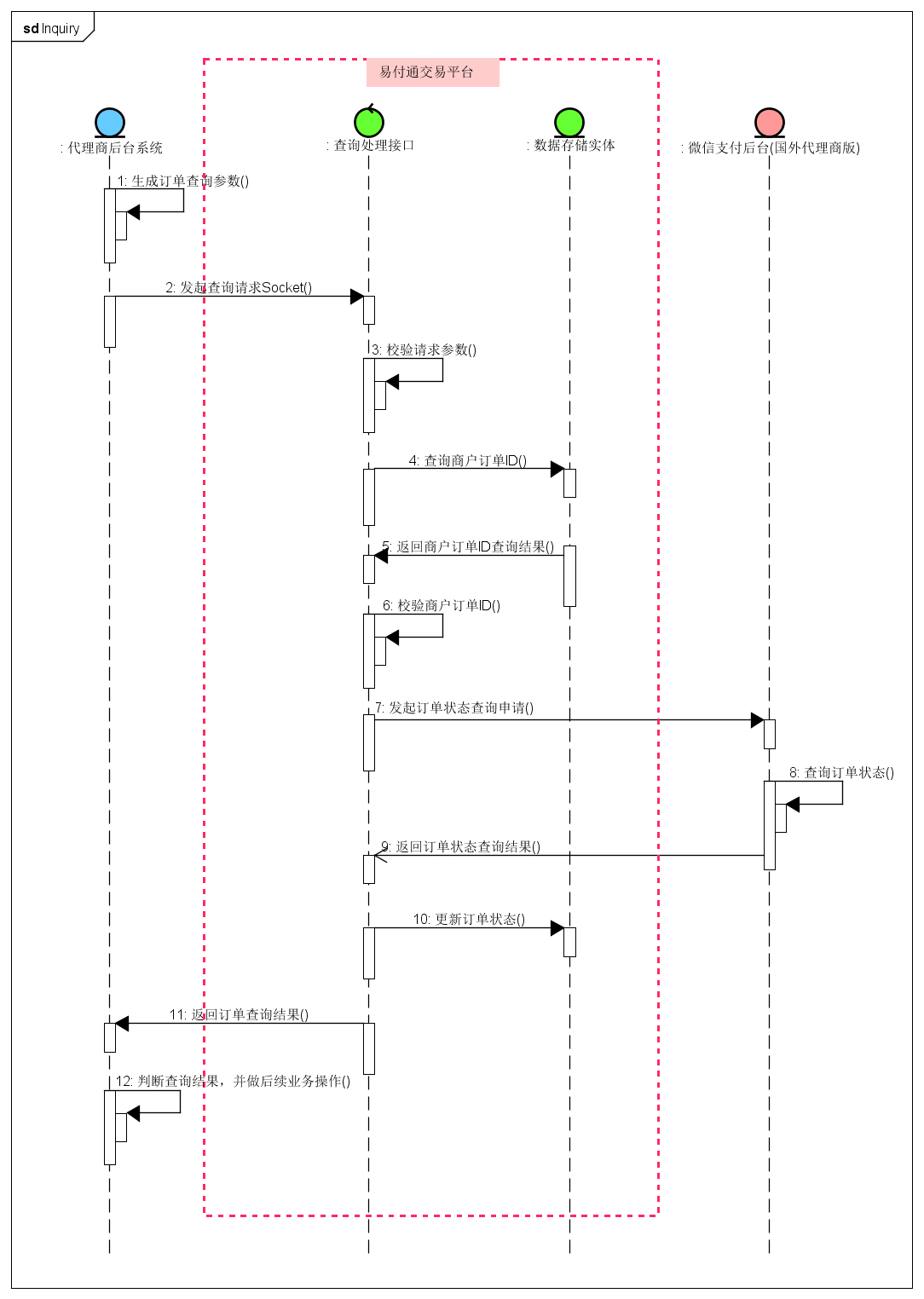
# 技术架构

## 交互序列图

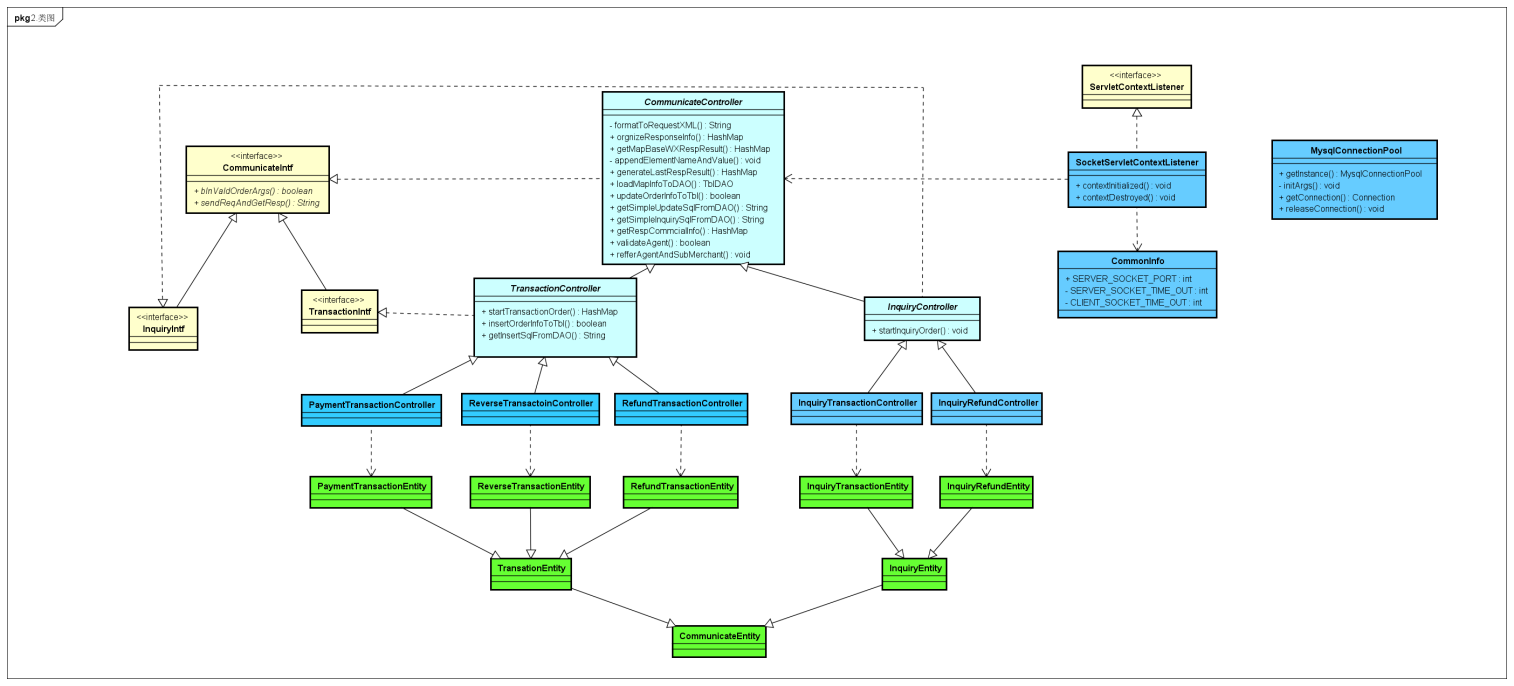
### 支付



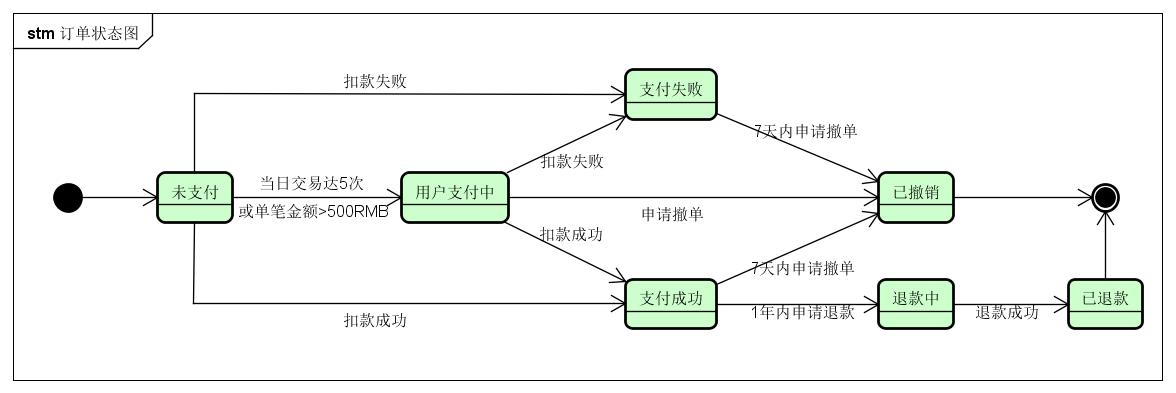
### 查询



## 实体关系图



## 状态变迁图



# 对接方式

## 相关协议

* 协议类型：Socket
* 协议端口：10086
* IP地址：\*\*\*.\*\*\*.\*\*\*.\*\*\*

## 业务类型

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 操作类型 | 操作标识 | 备注 |
| 1 | 订单支付 | PaymentTransaction | 标识此业务是将商户订单进行支付操作。 |
| 2 | 查询订单 | InquiryTransactionOrder | 标识此业务是查询已经支付的订单。 |
| 3 | 撤销订单 | ReverseTransactoin | 标识此业务是撤销已经支付的订单。 |
| 4 | 申请退款 | RefundTransaction | 标识此业务是将已经支付的订单进行退款。 |
| 5 | 查询退款单 | InquiryRefundOrder | 标识此业务是查询已经退款的订单。 |

## 报文字段及样例

### 请求报文

#### 请求报文格式

操作标识:报文内容

【注】：”操作标识”请参照上表中”操作标识”列的内容。

#### 请求报文字段

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 字段名称 | 字段类型 | 字段说明 |
| 1 | agent\_id | String | 标识代理商的唯一标识。 |
| 2 | appid | String | 公众账号ID。 |
| 3 | mch\_id | String | 商户号ID，此处代表易付通美国(Harvest)。 |
| 4 | sub\_mch\_id | String | 子商户公众账号ID，此处代表终端商户。 |
| 5 | nonce\_str | String | 随机字符串，不长于32位。 |
| 6 | body | String | 商品或支付单的简要描述。 |
| 7 | out\_trade\_no | String | 商户系统内部订单号，要求32个字符内，只能是数字、大小写字母\_-|\*@ |
| 8 | total\_fee | String | 标价订订单总金额，单位为分，只能是整数。 |
| 9 | fee\_type | String | 符合ISO 4217标准的三位字母代码，此处为“USD”。 |
| 10 | spbill\_create\_ip | String | 调用微信支付API的机器IP。 |
| 11 | auth\_code | String | 扫码支付授权码，设备读取用户微信中的条码或二维码。 |
| 12 | app\_key | String | 商户平台设置的密钥KEY。 |
| 13 | Sign | String | 签名。 |

#### 请求报文举例

* **订单支付**

**public** **static** **void** main(String[] args) {

TransactionTester transTester = **new** TransactionTester();

String strBizReq = **null**;

String strBizType = **null**;

/\*\* 测试支付订单 \*\*/

strBizType = CommonInfo.***PAYMENT\_TRANSACTION\_BIZ***; // PaymentTransaction

strBizReq = transTester.getPamentTransRequest();

*exeSocketTest*(strBizType + ":" + strBizReq);

}

/\*\*

\* 支付交易对应的请求报文，格式: sub\_mch\_id=983477232&nonce\_str=aiadj7jsd8l

\* **@return**

\*/

**private** String getPamentTransRequest() {

String strAuthCode = "134944526081345965"; // 二维码中的用户授权码

StringBuffer sb = **new** StringBuffer();

sb.append(PaymentTransactionEntity.***AGENT\_ID*** + "=" + "1r10s84408mdj0tgp6iov2c0k54jbps9");

sb.append("&" + PaymentTransactionEntity.***SUB\_MCH\_ID*** + "=" + "12152566");

sb.append("&" + PaymentTransactionEntity.***NONCE\_STR*** + "=" + CommonTool.*getRandomString*(32));

sb.append("&" + PaymentTransactionEntity.***BODY*** + "=" + "Ipad mini 16G 白色"); // Ipad mini 16G 白色

sb.append("&" + PaymentTransactionEntity.***OUT\_TRADE\_NO*** + "=" + CommonTool.*getOutTradeNo*(**new** Date(), 18));

sb.append("&" + PaymentTransactionEntity.***TOTAL\_FEE*** + "=" + "1");

sb.append("&" + PaymentTransactionEntity.***FEE\_TYPE*** + "=" + "USD");

**try** {

sb.append("&" + PaymentTransactionEntity.***SPBILL\_CREATE\_IP*** + "=" + CommonTool.*getSpbill\_Create\_Ip*());

} **catch** (UnknownHostException e1) {

e1.printStackTrace();

}

sb.append("&" + PaymentTransactionEntity.***AUTH\_CODE*** + "=" + strAuthCode);

**return** sb.toString();

}

/\*\*

\* 通过Socket进行业务通信。

\* **@param** strData

\*/

**private** **static** **void** exeSocketTest(String strData) {

SocketConnectionManager socketConnMngr = CommonTool.SocketConnectionManager.*getInstance*();

socketConnMngr.openSocket("127.0.0.1", 10086);

socketConnMngr.writeData(strData);

String strResp = socketConnMngr.readData();

System.***out***.println("strResp = " + strResp);

socketConnMngr.closeSocket();

}

* **查询订单**

将”5.3.1 订单支付”章节中main方法内添加如下代码：

String strBizType = CommonInfo.***INQUIRY\_TRANSACTION\_BIZ***;

String strBizReq = transTester.getInquiryTransRequest();

*exeSocketTest*(strBizType + ":" + strBizReq);

并将”5.3.1 订单支付”章节中的getPamentTransRequest方法修改为如下代码：

**private** String getInquiryTransRequest() {

StringBuffer sb = **new** StringBuffer();

sb.append(InquiryTransactionEntity.***AGENT\_ID*** + "=" + "1r10s84408mdj0tgp6iov2c0k54jbps9");

sb.append("&" + InquiryTransactionEntity.***SUB\_MCH\_ID*** + "=" + "12152566");

sb.append("&" + InquiryTransactionEntity.***OUT\_TRADE\_NO*** + "=" + "20180313150955252700235704379667"); // 测试时修改此参数

sb.append("&" + InquiryTransactionEntity.***NONCE\_STR*** + "=" + CommonTool.*getRandomString*(32));

**return** sb.toString();

}

* **撤销订单**

将”5.3.1 订单支付”章节中main方法内添加如下代码：

String strBizType = CommonInfo.***REVERSE\_TRANSACTION\_BIZ***;

String strBizReq = transTester.getReverseTransRequest();

*exeSocketTest*(strBizType + ":" + strBizReq);

并将”5.3.1 订单支付”章节中的getPamentTransRequest方法修改为如下代码：

**private** String getReverseTransRequest() {

StringBuffer sb = **new** StringBuffer();

sb.append(ReverseTransactionEntity.***AGENT\_ID*** + "=" + "1r10s84408mdj0tgp6iov2c0k54jbps9");

sb.append("&" + ReverseTransactionEntity.***SUB\_MCH\_ID*** + "=" + "12152566");

sb.append("&" + ReverseTransactionEntity.***OUT\_TRADE\_NO*** + "=" + "20180313150955252700235704379667"); // 测试时修改此字段

sb.append("&" + ReverseTransactionEntity.***NONCE\_STR*** + "=" + CommonTool.*getRandomString*(32));

**return** sb.toString();

}

* **申请退款**

将”5.3.1 订单支付”章节中main方法内添加如下代码：

String strBizType = CommonInfo.***REFUND\_TRANSACTION\_BIZ***;

String strBizReq = transTester.getRefundTransRequest();

*exeSocketTest*(strBizType + ":" + strBizReq);

并将”5.3.1 订单支付”章节中的getPamentTransRequest方法修改为如下代码：

**private** String getRefundTransRequest() {

StringBuffer sb = **new** StringBuffer();

StringBuffer sb = **new** StringBuffer();

sb.append(RefundTransactionEntity.***AGENT\_ID*** + "=" + "1r10s84408mdj0tgp6iov2c0k54jbps9");

sb.append("&" + RefundTransactionEntity.***SUB\_MCH\_ID*** + "=" + "12152566");

sb.append("&" + RefundTransactionEntity.***NONCE\_STR*** + "=" + CommonTool.*getRandomString*(32));

sb.append("&" + RefundTransactionEntity.***OUT\_TRADE\_NO*** + "=" + "20180313154339401818102418248337"); // 测试时修改此字段

sb.append("&" + RefundTransactionEntity.***OUT\_REFUND\_NO*** + "=" + CommonTool.*getOutRefundNo*(**new** Date(), 18)); // 同一退款单号时记得修改此字段为固定值

sb.append("&" + RefundTransactionEntity.***TOTAL\_FEE*** + "=" + "1");

sb.append("&" + RefundTransactionEntity.***REFUND\_FEE*** + "=" + "1");

**return** sb.toString();

}

* **查询退款单**

将”5.3.1 订单支付”章节中main方法内添加如下代码：

String strBizType = CommonInfo.***INQUIRY\_REFUND\_BIZ***;

String strBizReq = transTester.getInquiryRefundRequest();

*exeSocketTest*(strBizType + ":" + strBizReq);

并将”5.3.1 订单支付”章节中的getPamentTransRequest方法修改为如下代码：

**private** String getInquiryRefundRequest() {

StringBuffer sb = **new** StringBuffer();

sb.append(InquiryRefundEntity.***AGENT\_ID*** + "=" + "1r10s84408mdj0tgp6iov2c0k54jbps9");

sb.append("&" + InquiryRefundEntity.***SUB\_MCH\_ID*** + "=" + "12152566");

sb.append("&" + InquiryRefundEntity.***NONCE\_STR*** + "=" + CommonTool.*getRandomString*(32));

sb.append("&" + InquiryRefundEntity.***OUT\_TRADE\_NO*** + "=" + "20180313154339401818102418248337"); // 测试时需修改此参数

**return** sb.toString();

}

### 应答报文

#### 应答报文格式

<XML>

<业务字段>

<![CDATA[字段数值]]>

</return\_code>

<cash\_fee>字段数值</cash\_fee>

<… … …>

</XML>

#### 应答报文字段

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 字段名称 | 字段类型 | 字段说明 |
| 1 | return\_code | String | 此字段是通信标识，而非交易标识；交易标识是否成功要看result\_code来判断。 |
| 2 | result\_code | String | 业务处理结果。 |
| 3 | err\_code | String | 错误代码。 |
| 4 | openid | String | 用户在商户appid下的唯一标识。 |
| 5 | is\_subscribe | String | 用户是否关注公众账号。 |
| 6 | trade\_type | String | 交易类型，MICROPAY即为扫码支付。 |
| 7 | bank\_type | String | 银行类型，采用字符串标识的银行标识。 |
| 8 | fee\_type | String | 标价币种，此处默认为USD。 |
| 9 | total\_fee | String | 订单总金额，单位为分，只能为整数。 |
| 10 | cash\_fee | String | 订单现金支付金额。 |
| 11 | transaction\_id | String | 微信端生成的交易订单号。 |
| 12 | time\_end | String | 订单生成时间，格式为：yyyyMMddHHmmss |
| 13 | rate | String | 标价币种与支付币种的总换比例乘以10的8次方即为此值，例如美元兑换人民币的比例为6.5，则rate=650000000 |

【注】：

1. 部分字段的说明已经在“请求报文”中说明，故在上表中未再进行说明。
2. err\_code错误码对应的不同业务类型及含义，请参照如下链接：<https://pay.weixin.qq.com/wiki/doc/api/external/micropay_sl.php?chapter=9_10&index=1> 中的“错误码”部分。

#### 应答报文举例

<xml>

<return\_code>

<![CDATA[SUCCESS]]>

</return\_code>

<return\_msg>

<![CDATA[OK]]>

</return\_msg>

<appid>

<![CDATA[wxb91c84b6c4d2e07b]]>

</appid>

<mch\_id>

<![CDATA[1900014621]]>

</mch\_id>

<sub\_mch\_id>

<![CDATA[12152566]]>

</sub\_mch\_id>

<nonce\_str>

<![CDATA[PuWB1nAZaIJ2Ts9O]]>

</nonce\_str>

<sign>

<![CDATA[B8E9D0FD73C7F6A0785E168CB10B43B9]]>

</sign>

<result\_code>

<![CDATA[SUCCESS]]>

</result\_code>

<openid>

<![CDATA[oAOWGszickfzY8G-PW5GqaSOOTfo]]>

</openid>

<is\_subscribe>

<![CDATA[N]]>

</is\_subscribe>

<trade\_type>

<![CDATA[MICROPAY]]>

</trade\_type>

<bank\_type>

<![CDATA[ABC\_DEBIT]]>

</bank\_type>

<total\_fee>1</total\_fee>

<fee\_type>

<![CDATA[USD]]>

</fee\_type>

<transaction\_id>

<![CDATA[4200000073201803170563942109]]>

</transaction\_id>

<out\_trade\_no>

<![CDATA[20180317201500467656238118218599]]>

</out\_trade\_no>

<attach>

<![CDATA[]]>

</attach>

<time\_end>

<![CDATA[20180317201543]]>

</time\_end>

<trade\_state>

<![CDATA[SUCCESS]]>

</trade\_state>

<cash\_fee>6</cash\_fee>

<cash\_fee\_type>

<![CDATA[CNY]]>

</cash\_fee\_type>

<rate>

<![CDATA[635060000]]>

</rate>

</xml>

# 版本控制

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 版本号 | 修改者 | 修改日期 | 修改内容 | 备注 |
| 1 | 1.0 | 邱伟 | 2018-03-13 | 创建文档 |  |
| 2 | 2.0 | 邱伟 | 2018-03-17 | 1. 更新部分业务流程； 2. 添加关键字段详细说明； |  |
| 3 | 2.1 | 邱伟 | 2018-03-18 | 1. 修改业务流程图中的部分内容。 |  |