微信二维码支付技术商户端对接文档

（被扫模式）

北京易付通金服技术有限公司

2018-03-13

V1.0

# 业务背景

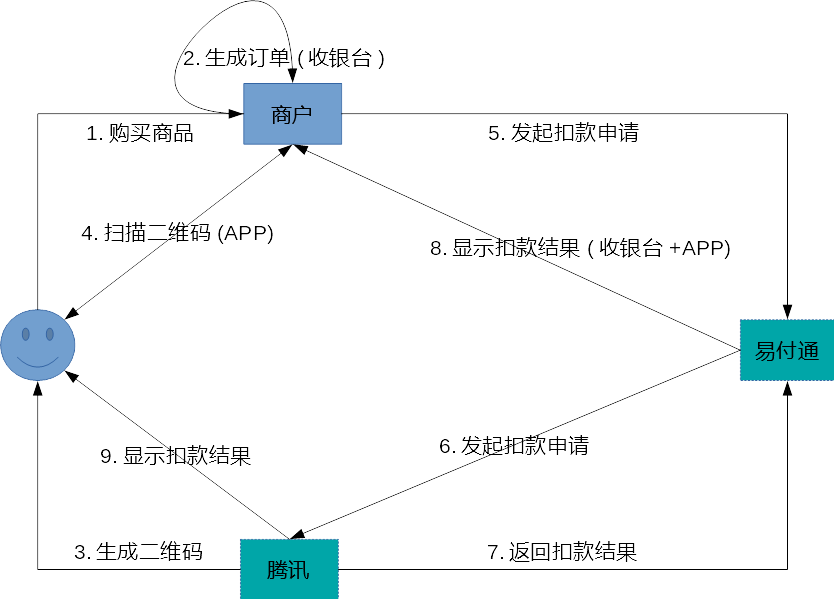
随着支付业务的增长，用户在跨境支付方面的需求也日益增加。特别是基于微信这样大众更易接受的方式，更是迫切需要在国外的落地实施。在此方面，所涉及到的用户群体，如：商务人士、旅游人士、留学生等；所涉及到的商户，如：免税店、大型商超、汽车租赁等。

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

本文档适用于商户端技术人员阅读。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。

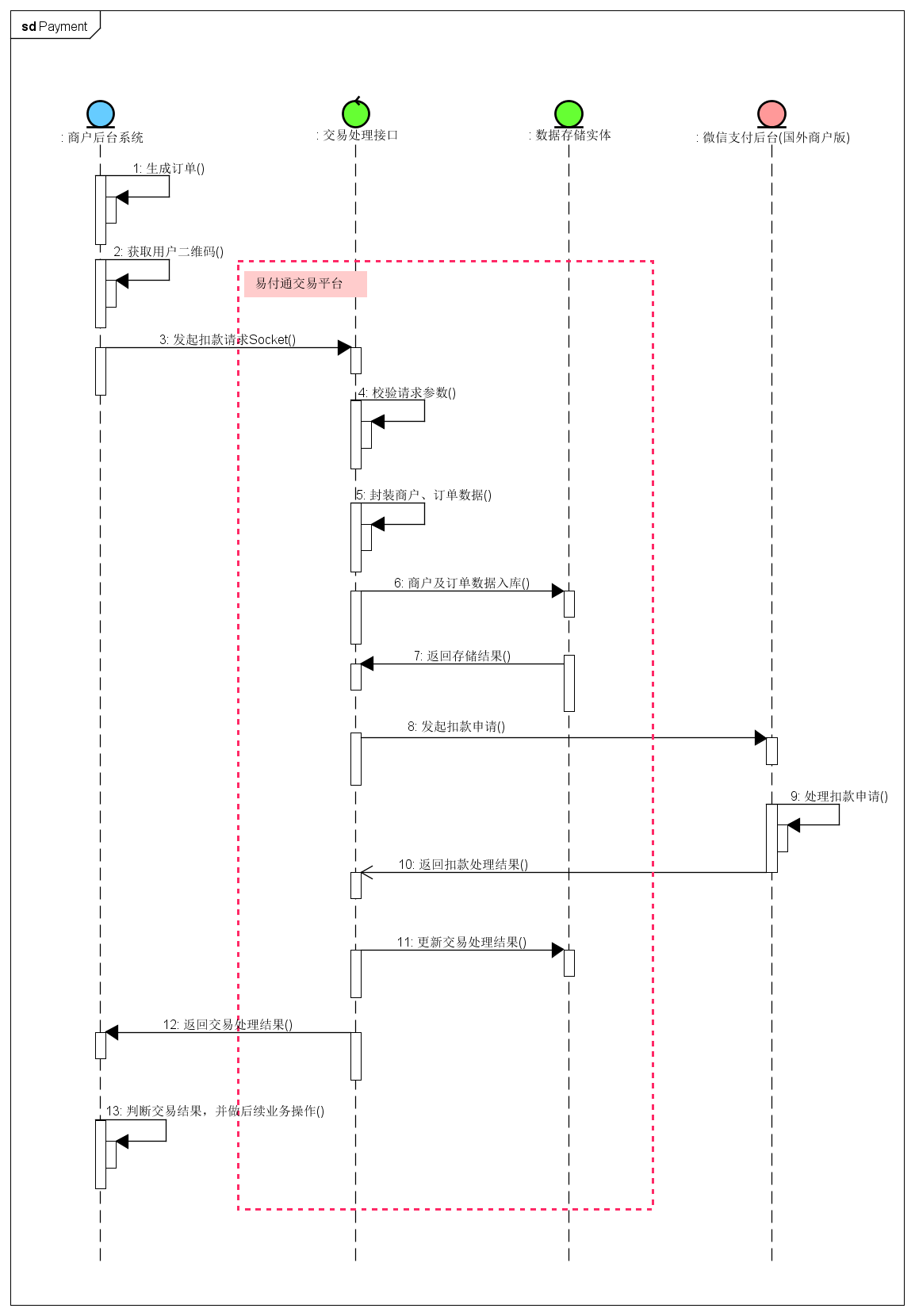
# 业务关系



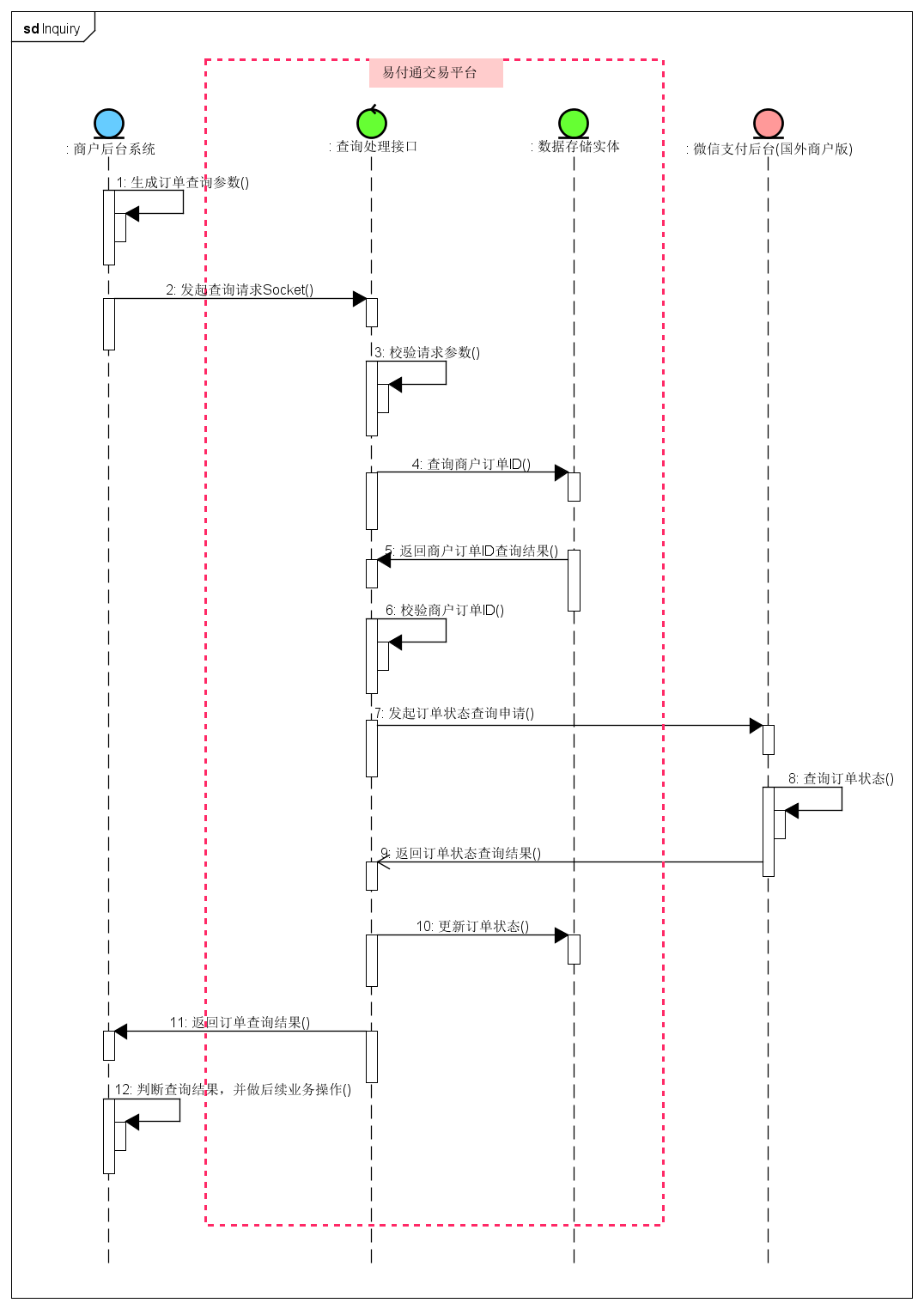
# 技术架构

## 交互序列图

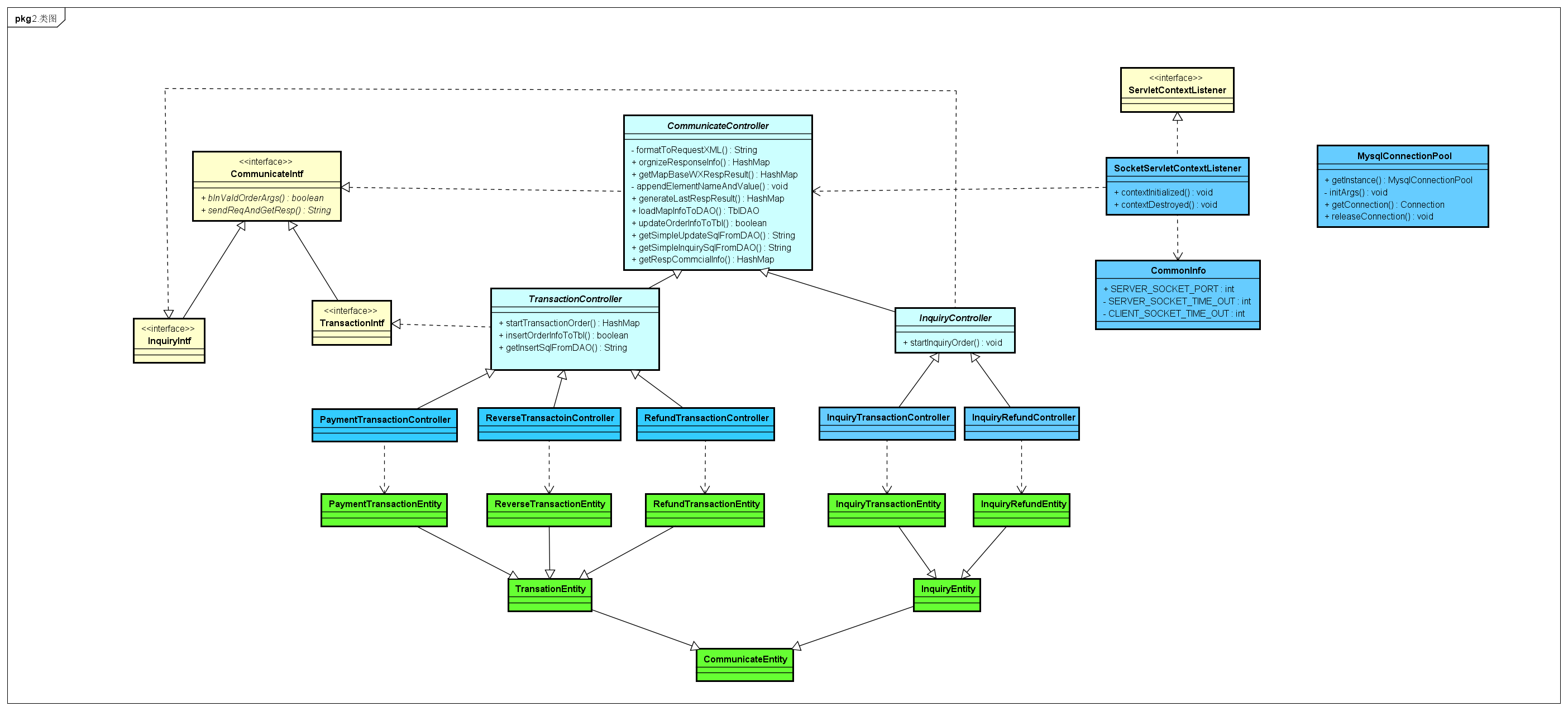
### 支付



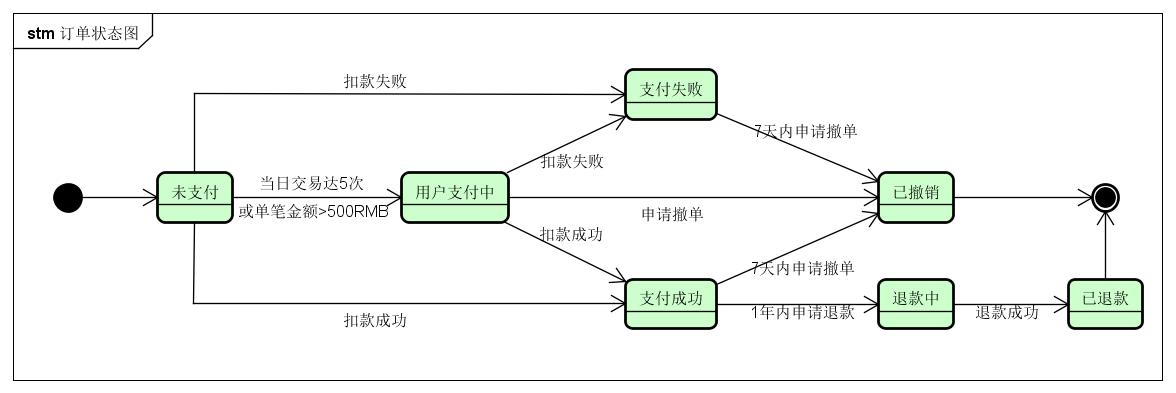
### 查询



## 实体关系图



## 状态变迁图



# 对接方式

## 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。 |

## 相关协议

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

## 接口样例

### 订单支付

**public** **class** TransactionTester {

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

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

/\*\* 测试支付订单 \*\*/

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

String strBizReq = transTester.getPamentTransRequest();

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

}

**private** String getPamentTransRequest() {

String strAuthCode = "134965823249730636"; // 二维码中的用户授权码,此部分需商户端扫码提供

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

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);

sb.append("&" + PaymentTransactionEntity.***APP\_KEY*** + "=" + "024edfffae32c829b012c98a61686f3b");

**return** sb.toString();

}

**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.***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));

sb.append("&" + PaymentTransactionEntity.***APP\_KEY*** + "=" + "024edfffae32c829b012c98a61686f3b");

**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.***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));

sb.append("&" + ReverseTransactionEntity.***APP\_KEY*** + "=" + "024edfffae32c829b012c98a61686f3b");

**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();

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");

sb.append("&" + RefundTransactionEntity.***APP\_KEY*** + "=" + "024edfffae32c829b012c98a61686f3b");

**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.***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

sb.append("&" + InquiryRefundEntity.***APP\_KEY*** + "=" + "024edfffae32c829b012c98a61686f3b");

**return** sb.toString();

}

# 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 |  |
|  |  |  |  |  |  |