

赢商通

赢商通支付平台

HTTP 接口开发文档

版本号：1.3.1

历史记录

号码	修改日期	修改内容	版本号
1.	2015-01-01	创建该文档	V1.0.0
2.	2015-03-15	接口规范重大调整	V1.1.0
3.	2015-03-20	接口响应变动, 更换接口响应的签名机制	V1.1.1
4.	2015-03-27	修改支付订单和退款订单状态; 增加支付时间字段;	V1.1.2
5.	2017-05-09	增加扫码支付和公众号支付接口	v1.2.0
6.	2017-10-24	添加支付宝JS支付文档说明	V1.3.0
7.	2018-02-02	增加H5支付	V1.3.1

1. 引言	4
1. 1. 概述	4
1. 2. 适用范围	4
1. 3. 文档一般约定	4
2. 通信协议及方式	5
2. 1. 通信协议	5
2. 2. 消息结构说明	5
2. 3. 报文消息中约束字段的描述	5
2. 4. 报文消息中长度字段的描述	5
2. 5. 请求	5
2.5.1. 通信安全	6
2.5.2. 签名机制	6
2. 6. 响应	7
2.6.1. 通信安全	7
2.6.2. 签名机制	8
3. 接口说明	8
3.1. 接口中的约束字段描述	8
3.2. 接口中的长度字段描述	9
4. 条码支付	9
4.1. 提交支付	9
4.2. 查询订单	11
4.3. 退款接口	13
4.4. 退款查询	14
4.5. 撤单接口	16
5. 扫码支付	17
5.1. 提交支付	17
5.2. 查询订单	19
5.3. 退款接口	20
5.4. 退款查询	20
5.5. 支付结果通知	20
6. 微信公众号支付	21
6.1. 提交支付	21
6.2. 查询订单	24
6.3. 退款接口	24
6.4. 退款查询	24
6.5. 支付结果通知	24

1. 引言

本文档规定了第三方收银系统与赢商通支付平台的接口要求，是收银系统接入平台的技术规范依据。主要包括以下几方面内容：通信协议及方式、接口消息格式定义、接口定义等。

1.1. 概述

赢商通支付平台针对每个接入平台的商户都会分配一个 PRIVATE_KEY 用于接入平台时的加密私钥。

1.2. 适用范围

本文档的适用对象为接入赢商通支付平台的收银厂商技术开发人员。

1.3. 文档一般约定

本文档在数据通信方面采用了 AES，以及 MD5 两种算法进行加密处理，用以保证数据传输安全性。在不做特殊的说明情况下，文档中提到 AES、MD5 加密算法时都应该遵循以下的规范要求。

约定类型	约定说明	
AES	标准的 AES/CBC/NoPadding 模式。	私钥：商户入驻时，分配给商户的唯一私有凭证。保证在整个系统中唯一性。
MD5	MD5 加密算法	经 MD5 加密算法加密后的密文，需转换为 大写 。
Code	UTF-8	本文档中所有需要进行编码的操作均采用 UFT-8 编码。
Fee	金额以“ 分 ”为单位	接口中的支付金额，退款金额等所有涉及到金额相关的数据，都以“ 分 ”为单位。

2. 通信协议及方式

2. 1. 通信协议

赢家通微信支付平台系统提供基于 HTTP 协议通信的接入接口，以满足各类程序语言的接入要求。

2. 2. 消息结构说明

消息报文采用 XML 方式，content-type 设置为 text/xml；XML 的 encoding 设置为“UTF-8”；报文中 request_type 的值做为接口请求的唯一标识，请求的报文以_request 结尾，响应的报文以_response 结尾，如：query_request、query_response。

2. 3. 报文消息中约束字段的描述

符号	含义
?	0 或 1 项，可选项，可为空。
*	0.....n 项，可为空，也可以有多项
+	1.....n 项，不可为空，至少有 1 项，也可以有多项。
1	只能有 1 项，不可为空。

2. 4. 报文消息中长度字段的描述

符号	含义
F	固定长度，F32，固定 32 个字符
V	可变长度，V128，最长 128 个字符

2. 5. 请求

接口采用 HTTP POST 方法提交请求，请求的示例如下。

包头示例：

```
"POST /index.jsp HTTP/1.1\r\n"
"Accept-Language: zh-cn\r\n"
"Connection: Keep-Alive\r\n"
"Host: " + hostUrl + "\r\n"
"Content-Length: " + content_length + "\r\n"
```

包体示例：

```
<request>
  <request_type>device_auth_request</request_type>
  <account>d0001</account>
  <dsn>669262B6A21A3B4E4610A78ACF29AE71</dsn>
  <data>xWfmmNdEG7NGw3VwD/zOp447KV$avoBv+g5+A47YBBbXn3XYTu4
WA2TF86nnlHBoY7TAnfX9Bw3Fo1H1J+1XCqylUbqrz5dcqhgyOJSArWSMMUiw
4W7p7A3hykMYVjIkG+zvNsmtsmo6wuIawEtKj8pgyqsfCkcy/QqXkDAWU0mA
XKAappRYdmk4sYuUwInpLLxdYaKdNyjV/wmAjwvcDQ==</data>
  <signature>669262B6A21A3B4E4610A78ACF29AE71</signature>
</request>
```

2.5.1. 通信安全

包体中的<data>为正式的请求内容，数据格式为XML格式，采用AES加密后的密文，AES加密使用的Key，IV即由平台生成并分发的PrivateKey。明文内容实例如下：

```
<data>
  <request_type>device_auth_request</request_type>
  <dsn>669262B6A21A3B4E4610A78ACF29AE71</dsn>
  <req_seq>1426497738440</req_seq>
  <auth_code>130356087568669625</auth_code>
  <consume>2500</consume>
</data>
```

请求中的设备接入序列号dsn在网络间传输是通过MD5加密后的密文。加密方法如下：

$dsn = MD5(acct + privateKey + md5(pass))$

字段	说明	备注
acct	终端设备登录账号	最少4个，最长20个字符，保证每个商户下的账号唯一。 参与dsn计算时，先将acct转小写再计算。
private_key	AES加密密钥。	由系统自动分配，保证每个商户唯一。
pass	终端设备登录密码。	最少6个字符，最长不做限制。

2.5.2. 签名机制

签名数据放在<signature>节点下，由MD5加密算法生成，具体的签名方法如下：

$MD5(private_key + dsn + data)$

`data`为经过AES加密后的密文内容。

2. 6. 响应

交易的结果无论处理成功还是失败，只要不是网络异常导致网络断开，都会收到服务器响应，响应的 http 报文头 status 均为 200； 报文体为 xml 格式的 UTF-8 编码字节流，报文具体内容包含在`<response></response>`标签内。

包头示例：

```
"HTTP/1.1 200 OK\r\n"
"Content-Length: " + content_length + "\r\n" "\r\n"
```

包体示例：

```
<response>
    <timestamp>1426497738440</timestamp>
    <data>xWfmmNdEG7NGw3VwD/zOp447KV$avoBv+g5+A47YBBbXn3XYTu4
WA2TF86nnlHBoY7TAnfX9Bw3Fo1H1J+1XCqylUbqrz5dcqhgyOJSArWSMMUiw
4W7p7A3hykMYVjIkG+zvNsmtsmo6wuIawEtKj8pgyqsfCkcy/QqXkDAWU0mA
XKAappRYdmk4sYuUwInpLLxdYaKdNyjV/wmAjwvcDQ==</data>
    <signature>669262B6A21A3B4E4610A78ACF29AE71</signature>
</response>
```

2.6.1. 通信安全

响应报文的实质响应内容包含在`<data></data>`标签中，通过 AES 加密算法加密而成。下面是一个具体请求的响应例子。

```
<data>
    <response_type>transaction_response</response_type>
    <req_seq>1426497738440<req_seq>
    <res_seq>1426499182103</res_seq>
    <result>Y</result>
    <trans_seq>1426499372793</trans_seq>
    <status>SUCCESS</status>
    <receipts>
        <line>----微信支付交易小票-----</line>
        <line>      【商户联】</line>
        <line>交易时间：2015-04-17 11:10:51</line>
        <line>交易流水：15041711103000029582</line>
        <line>交易金额：0.01元</line>
        <line>交易门店：金堂一店</line>
        <line>操作员：sp004</line>
        <line> </line>
        <line> </line>
```

```

<line>-----微信支付交易小票-----</line>
<line>      【用户联】</line>
<line>交易时间: 2015-04-17 11:10:51</line>
<line>交易流水: 15041711103000029582</line>
<line>交易金额: 0.01元</line>
<line>交易门店: 金堂一店</line>
<line>操作员: sp004</line>
</receipts>
<server_time></server_time>
<code></code>
<msg></msg>
</data>

```

在响应报文中，将上面的响应内容通过 AES 加密算法加密后放在报文的<data></data>节点下面。

响应报文中的 AES 加密算法中使用的 Key, IV 是 DSN 前 16 个字符。即提交请求时的计算生成的 DSN 的前 16 位。

2.6.2. 签名机制

签名数据放在<signature></signature>标签中，签名方法如下：
MD5 (dsn + timestamp + data)

3. 接口说明

测试接口地址: <http://ut.ystpay.cn:8088/ystpay/service/unitive.api>

正式接口地址在接口调试，以及功能调试完成后，正式上线前提供。

3.1. 接口中的约束字段描述

符号	含义
?	0 或 1，可选项，可为空。
*	0.....n 项，可为空，也可以有多项
+	1.....n 项，不可为空，至少有 1 项，也可以有多项。
1	只能有 1 项，不可为空。

3.2. 接口中的长度字段描述

符号	含义
F	固定长度, F32, 固定 32 个字符
V	可变长度, V128, 最长 128 个字符

4. 条码支付

4.1. 提交支付

用于提交条码支付请求。终端软件收集用户的支付金额以及支付授权码后，通过该接口提交给动态库，由动态库请求服务器进行支付。

接口请求定义：

节点名	必填	类型	描述	取值说明
request_type	是	String(V50)	请求类型	unitive_pay_request
dsn	1	String(F32)	设备序列号	
req_seq	是	String(V20)	请求流水号	客户端支付订单号，必须保证唯一
total_fee	是	Number(V10)	支付金额	单位:分
auth_code	是	String(V32)	支付条码	用户客户端中的条码，或者二维码所包含的文本内容。
channel	否	String(F1)	支付渠道	需要指定支付方式时传入 W-微支付，A-支付宝，Q-手 Q 支付
prod_details	否	String(V8192)	商品或订单明 细列表	详细的商品列表，json 格式

接口具体请求数据的明文：data

```
<data>
    <request_type>unitive_pay_request</request_type>
    <dsn>669262B6A21A3B4E4610A78ACF29AE71</dsn>
    <req_seq>20152423432532532525</req_seq>
    <total_fee>10000</total_fee>
    <prod_details>[{"goodsId": "2900002400915", "goodsName": " 奶 油 曲 奇 ", "price": "17.60", "quantity": "1", "goodsFee": "17.60"}, {"goodsId": "2900002400922", "goodsName": " 小 石 头 ", "price": "25.60", "quantity": "1", "goodsFee": "25.60"}]</prod_details>
    <auth_code>1232456487987985646</auth_code>
```

```
</data>
```

商品详情 json 格式：

节点名	必填	类型	描述	取值说明
goodsId	是	String(V50)	商品条码	
goodsName	是	String(V200)	商品名称	
price	是	Number(V10)	商品单价	单位：元
quantity	是	Number(V10)	商品数量	
goodsFee	否	Number(V10)	商品金额	该商品需要支付的金额， 单位：元

```
[  
 {  
     "goodsId": "2900002400915",  
     "goodsName": "奶油曲奇",  
     "price": "17.60",  
     "quantity": "1",  
     "goodsFee": "17.60",  
 },  
 {  
     "goodsId": "2900002400922",  
     "goodsName": "小石头",  
     "price": "25.60",  
     "quantity": "1",  
     "goodsFee": "25.60",  
 }  
 ]
```

接口响应定义：

节点名	必填	类型	描述	取值说明
req_seq	是	Number(V20)	请求流水号	调用接口时，本机生成的流水号，服务器返回该数据可用作校验。
result	1	String(F1)	响应结果	Y: 请求成功 N: 请求失败
code	是	String(F6)	响应代码	
msg	是	String(V100)	响应信息	服务对本次请求的文字描述，可通过查看 msg 了解本次请求成功、失败的具体原因。
channel	否	String(F1)	支付渠道	W-微支付

				A-支付宝 Q-手 Q 支付 Y-翼支付
channel_name	否	String(V10)	渠道名称	
trans_seq	否	Number(V20)	交易流水号	本地需要存储的交易流水号，即订单号，如果需要做退款处理，交易流水是唯一的退款凭证。
total_fee	否	Number(V10)	支付金额	单位:分
cash_fee	否	Number(V6)	现金金额	
coupon_fee	否	Number(V6)	优惠金额	
receipts	否	String(V500)	小票内容	
promotion_detail	否	String(V500)	优惠详情	详细字段参见下面
pay_time	否	String(F19)	支付时间	以此时间做为对账时间
server_time	是	String(F19)	服务器时间	

接口响应数据示例：

```

<data>
  <code>E11000</code>
  <msg>支付成功</msg>
  <server_time>2015-06-09 11:17:48:751</server_time>
  <trans_seq>15060911165638124989</trans_seq>
  <total_fee>1</total_fee>
  <cash_fee>1</cash_fee>
  <coupon_fee>0</coupon_fee>
  <pay_time>2015-06-09 11:17:18</pay_time>
  <channel>W</channel>
  <status>S</status>
</data>

```

4.2. 查询订单

查询接口用于查询支付订单的支付结果。

接口请求定义：

节点名	必填	类型	描述	取值说明
request_type	是	String(V50)	请求类型	unitive_query_pay_request
req_seq	是	String(V20)	请求流水号	
dsn	1	String(F32)	设备序列号	
auth_code	否	String(V32)	支付条码	支付时传入的条码

ori_seq	是	Number (V20)	原请求流水号	请求支付时, 使用的请求流水号。
---------	---	--------------	--------	------------------

接口请求示例:

```
<data>
  <request_type>unitive_query_pay_request</request_type>
  <dsn>669262B6A21A3B4E4610A78ACF29AE71</dsn>
  <req_seq>923735276822399045444</req_seq>
  <auth_code>92373527682239904364</auth_code>
  <ori_seq>92373527682239904364</ori_seq>
</data>
```

接口响应定义:

节点名	必填	类型	描述	取值说明
req_seq	是	Number (V20)	请求流水号	调用接口时, 本机生成的流水号, 服务器返回该数据可用作校验。
result	1	String (F1)	响应结果	Y: 请求成功 N: 请求失败
code	是	String (F6)	响应代码	
msg	是	String (V100)	响应信息	服务对本次请求的文字描述, 可通过查看 msg 了解本次请求成功、失败的具体原因。
channel1	否	String (F1)	支付渠道	W-微支付 A-支付宝 Q-手 Q 支付 Y-翼支付
channel_name	否	String (V10)	渠道名称	
trans_seq	否	Number (V20)	交易流水号	本地需要存储的交易流水号, 即订单号, 如果需要做退款处理, 交易流水是唯一的退款凭证。
total_fee	否	Number (V10)	支付金额	单位:分
cash_fee	否	Number (V6)	现金金额	
coupon_fee	否	Number (V6)	优惠金额	
receipts	否	String (V500)	小票内容	
promotion_detail	否	String (V500)	优惠详情	详细字段参见下面
pay_time	否	String (F19)	支付时间	以此时间做为对账时间
server_time	是	String (F19)	服务器时间	

接口响应数据示例:

```

<data>
  <code>E11000</code>
  <msg>支付成功</msg>
  <server_time>2015-06-09 11:17:48:751</server_time>
  <trans_seq>15060911165638124989</trans_seq>
  <total_fee>1</total_fee>
  <cash_fee>1</cash_fee>
  <coupon_fee>0</coupon_fee>
  <pay_time>2015-06-09 11:17:18</pay_time>
  <channel>W</channel>
  <status>S</status>
</data>

```

4.3. 退款接口

接口请求定义:

节点名	必填	类型	描述	取值说明
request_type	是	String(V50)	请求类型	unitive_refund_request
dsn	1	String(F32)	设备序列号	
req_seq	是	String(V20)	请求流水号	客户端退款单号, 必须保证唯一
trans_seq	是	String(V20)	原交易流水号	交易时, 由服务器下发的交易流水
refund_fee	是	Number(V10)	退款金额	单位:分
reason	是	String(V255)	退款理由	

接口请求示例:

```

<data>
  <request_type>refund_fee_request</request_type>
  <dsn>669262B6A21A3B4E4610A78ACF29AE71</dsn>
  <req_seq>15042118053535326597</req_seq>
  <trans_seq>15042118055639386597</trans_seq>
  <refund_fee>1</refund_fee>
  <reason>用户对商品或者服务不满意</reason>
</data>

```

接口响应定义:

节点名	必填	类型	描述	取值说明
req_seq	是	Number(V20)	请求流水号	
result	1	String(F1)	响应结果	Y: 请求成功 N: 请求失败

code	是	String(F6)	错误代码	
msg	是	String(V100)	错误信息	
channel	否	String(F1)	支付渠道	W-微支付 A-支付宝 Q-手 Q 支付 Y-翼支付
channel_name	否	String(V10)	渠道名称	
refund_seq	否	Number(V20)	退款流水号	
total_fee	否	Number(V10)	支付金额	单位:分
refund_fee	否	Number(V10)	已退款金额	单位:分
cash_fee	否	Number(V6)	退回的现金金额	
coupon_fee	否	Number(V6)	退回的优惠券金额	
receipts	否	String(V500)	退款小票内容	
server_time	是	String(F19)	服务器时间	

接口响应示例:

```
<data>
<req_seq>92373527853748439041</req_seq>
<code>E13000</code>
<msg>退款申请成功</msg>
<server_time>2015-04-21 18:07:01:979</server_time>
<refund_seq>15042118070122094495</refund_seq>
<total_fee>1</total_fee>
<refund_fee>1</refund_fee>
<channel>W</channel>
<status>S</status>
</data>
```

4.4. 退款查询

接口请求定义:

节点名	必填	类型	描述	取值说明
request_type	是	String(V50)	请求类型	unitive_query_refund_request
dsn	1	String(F32)	设备序列号	
req_seq	是	String(V20)	请求流水号	
trans_seq	是	String(V20)	原交易流水号	交易时,由服务器下发的交易流水

ori_seq	是	String(V20)	原退款请求流水号	
---------	---	-------------	----------	--

接口请求示例:

```
<data>
  <request_type>unitive_query_refund_request</request_type>
  <dsn>669262B6A21A3B4E4610A78ACF29AE71</dsn>
  <req_seq>923735276822399045444</req_seq>
  <trans_seq>15042118055639386597</trans_seq>
  <ori_seq>15042118055639386597</ori_seq>
</data>
```

接口响应定义:

节点名	必填	类型	描述	取值说明
req_seq	是	Number(V20)	请求流水号	
result	1	String(F1)	响应结果	Y: 请求成功 N: 请求失败
code	是	String(F6)	错误代码	
msg	是	String(V100)	错误信息	
channel	否	String(F1)	支付渠道	W-微支付 A-支付宝 Q-手Q支付 Y-翼支付
channel_name	否	String(V10)	渠道名称	
refund_seq	否	Number(V20)	退款流水号	
total_fee	否	Number(V10)	支付金额	单位:分
refund_fee	否	Number(V10)	已退款金额	单位:分
cash_fee	否	Number(V6)	退回的现金金额	
coupon_fee	否	Number(V6)	退回的优惠券金额	
receipts	否	String(V500)	退款小票内容	
server_time	是	String(F19)	服务器时间	

```
<data>
  <req_seq>92373527853748439041</req_seq>
  <code>E13000</code>
  <msg>退款申请成功</msg>
  <server_time>2015-04-21 18:07:01:979</server_time>
  <refund_seq>15042118070122094495</refund_seq>
  <total_fee>1</total_fee>
```

```

<refund_fee>1</refund_fee>
<channel>W</channel>
<status>S</status>
</data>

```

4.5. 撤单接口

当客户端未成功收到服务器响应无法确认用户是否支付成功的情况下，可以调用撤单接口，如果用户已扣款，则将退款给用户。

接口请求定义：

节点名	必填	类型	描述	取值说明
request_type	是	String(V50)	请求类型	unitive_revoke_request
dsn	1	String(F32)	设备序列号	
req_seq	是	String(V20)	请求流水号	
ori_seq	是	String(V20)	支付请求流水号	客户端支付请求流水号
channel	否	String(F1)	支付渠道	W-微支付 A-支付宝 Q-手Q支付 Y-翼支付
auth_code	否	String(V32)	支付条码	用户支付客户端中的条码，或者二维码所包含的文本内容。 channel 和 auth_code 必须 2 选 1

接口请求示例：

```

<data>
  <request_type>unitive_revoke_request</request_type>
  <dsn>669262B6A21A3B4E4610A78ACF29AE71</dsn>
  <req_seq>15042118053535326597</req_seq>
  <ori_seq>15042118053535326598</ori_seq>
</data>

```

接口响应定义：

节点名	必填	类型	描述	取值说明
req_seq	是	Number(V20)	请求流水号	
result	是	String(F1)	请求结果	Y 请求成功/N 请求失败
result	1	String(F1)	响应结果	Y: 请求成功 N: 请求失败
code	是	String(F6)	错误代码	
msg	是	String(V100)	错误信息	
channel	否	String(F1)	支付渠道	W-微支付

				A-支付宝 Q-手 Q 支付 Y-翼支付
channel1_name	否	String(V10)	渠道名称	
server_time	是	String(F19)	服务器时间	

接口响应示例:

```
<data>
<req_seq>92373527853748439041</req_seq>
<result>Y</result>
<code>E13000</code>
<msg>撤单成功</msg>
</data>
```

5. 扫码支付

5.1. 提交支付

用于提交扫码支付请求。终端软件收集用户的支付金额后，通过该接口提交给动态库，由动态库请求服务器进行支付。

接口请求定义:

节点名	必填	类型	描述	取值说明
request_type	是	String(V50)	请求类型	unitive_native_pay_request
dsn	1	String(F32)	设备序列号	
req_seq	是	String(V20)	请求流水号	客户端支付订单号，必须保证唯一
total_fee	是	Number(V10)	支付金额	单位:分
expire_time	否	Number(V10)	过期时间	单位:秒 最小 300 秒 (5 分钟) 最大 1800 秒 (30 分钟)
channel1	是	String(F1)	支付渠道	W-微支付 A-支付宝

接口具体请求数据的明文: data

```
<data>
<request_type>unitive_native_pay_request</request_type>
<dsn>669262B6A21A3B4E4610A78ACF29AE71</dsn>
<req_seq>20152423432532532525</req_seq>
<total_fee>10000</total_fee>
```

```

<channel>W</channel>
</data>

```

接口响应定义：

节点名	必填	类型	描述	取值说明
req_seq	是	Number (V20)	请求流水号	调用接口时, 本机生成的流水号, 服务器返回该数据可用作校验。
code	是	String (F6)	响应代码	
msg	是	String (V100)	响应信息	服务对本次请求的文字描述, 可通过查看 msg 了解本次请求成功、失败的具体原因。
trans_seq	否	Number (V20)	交易流水号	本地需要存储的交易流水号, 即订单号, 如果需要做退款处理, 交易流水是唯一的退款凭证。
cash_fee	否	Number (V6)	现金金额	
coupon_fee	否	Number (V6)	优惠金额	
receipts	否	String (V500)	小票内容	
status	?	String (V20)	状态	W—等待支付
pay_url	?	String (V2000)	支付链接	客户端将此链接生成二维码
server_time	是	String (F19)	服务器时间	

接口响应数据示例：

```

<data>
  <code>E11000</code>
  <msg>支付成功</msg>
  <server_time>2015-06-09 11:17:48:751</server_time>
  <trans_seq>15060911165638124989</trans_seq>
  <total_fee>1</total_fee>
  <cash_fee>1</cash_fee>
  <coupon_fee>0</coupon_fee>
  <status>W</status>
  <pay_url>weixin://wxpay/bizpayurl?pr=jLbHdkp</pay_url>
</data>

```

5.2. 查询订单

查询接口用于查询支付订单的支付结果。

接口请求定义：

节点名	必填	类型	描述	取值说明
request_type	是	String(V50)	请求类型	unitive_native_query_pay_request
dsn	1	String(F32)	设备序列号	
req_seq	是	String(V20)	请求流水号	
channel	否	String(F1)	支付渠道	W-微支付 A-支付宝
ori_seq	是	Number(V20)	原请求流水号	请求支付时，使用的请求流水号。

接口请求示例：

```
<data>
    <request_type>unitive_native_query_pay_request</request_type>
    <dsn>669262B6A21A3B4E4610A78ACF29AE71</dsn>
    <req_seq>923735276822399045444</req_seq>
    <channel>W</channel>
    <ori_seq>92373527682239904364</ori_seq>
</data>
```

接口响应定义：

节点名	必填	类型	描述	取值说明
req_seq	是	Number(V20)	请求流水号	服务器返回的原请求流水，以便客户端做请求-响应数据校验。
code	是	String(F6)	错误代码	
msg	是	String(V100)	错误信息	
trans_seq	否	Number(V20)	交易流水号	如果查询到订单支付成功，返回的支付流水号。即订单号。
total_fee	否	Number(V10)	支付金额	单位：分
cash_fee	否	Number(V6)	现金金额	
coupon_fee	否	Number(V6)	优惠金额	
receipts	否	String(V500)	小票内容	
status	?	String(V20)	状态	W—等待支付 S—支付成功 F--支付失败

pay_url	?	String(V2000)	支付链接	客户端将此链接生成二维码
pay_time	?	String(F19)	支付时间	以此时间做为对账时间
server_time	是	String(F19)	服务器时间	

接口响应数据示例：

```

<data>
<code>E11000</code>
<msg>支付成功</msg>
<server_time>2015-06-09 11:17:48:751</server_time>
<trans_seq>15060911165638124989</trans_seq>
<total_fee>1</total_fee>
<cash_fee>1</cash_fee>
<coupon_fee>0</coupon_fee>
<status>S</status>
<pay_time>2015-06-09 11:17:18</pay_time>
</data>

```

5.3. 退款接口

参见 4.3。

5.4. 退款查询

参见 4.4。

5.5. 支付结果通知

支付完成后，平台会把相关支付结果发送到商户系统指定的接收 URL，商户系统需要接收处理，并返回应答。

对后台通知交互时，如果平台收到的应答不是成功或超时，平台认为通知失败，会通过一定的策略（如 15 分钟共 6 次）定期重新发起通知，尽可能提高通知的成功率，但微信不保证通知最终能成功。（通知间隔为 0/30/30/60/300/900，单位：秒）

注意：同样的通知可能会多次发送给商户系统。商户系统必须能够正确处理重复的通知。

推荐的做法是，当收到通知进行处理时，首先检查对应业务数据的状态，判断该通知是否已经处理过，如果没有处理过再进行处理，如果处理过直接返回结果成功。在对业务数据进行状态检查和处理之前，要采用数据锁进行并发控制，以避免函数重入造成的数据混乱。

接口定义：

节点名	约束	类型	描述	取值说明
request_type	1	String(V50)	请求类型	native_order_notic

				e
req_seq	1	String(V20)	请求流水号	
channel	1	String(V1)	支付渠道	W-微支付 A-支付宝
trans_seq	1	String(V20)	交易流水号	赢家通订单号
total_fee	?	Number(V10)	支付金额	单位:分
pay_time	1	String(V20)	交易时间	
status	?	String(V20)	状态	S—支付成功

```
<data>
  <request_type>native_order_notice</request_type>
  <req>20153535325325325325325</req>
  <server_time>2015-06-09 11:17:48:751</server_time>
  <channel>W</channel>
  <trans_seq>15060911165638124989</trans_seq>
  <ext_trans_id>20153532525325325325325325325325325</ext_trans_id>
  <total_fee>1</total_fee>
  <status>S</status>
  <pay_time>2015-06-09 11:17:18</pay_time>
</data>
```

接口响应:

节点名	必填	类型	描述	取值说明
result	是	String(F6)	状态	Y-成功, N-失败
message	是	String(V100)	描述信息	

```
<data>
    <result>Y</result>
    <message>成功</message>
</data>
```

6. JS 支付

6.1. 提交支付

用于提交微信,支付宝公众号 JS 支付请求。

接口请求定义：

节点名	必填	类型	描述	取值说明
request_type	是	String(V50)	请求类型	unitive_js_pay_request

dsn	1	String(F32)	设备序列号	
req_seq	是	String(V20)	请求流水号	客户端支付订单号，必须保证唯一
total_fee	是	Number(V10)	支付金额	单位:分
channel	是	String(F1)	支付渠道	W-微支付，A-支付宝
subopen_id	是	String(V32)	用户 ID	微支付：传入 APPID 对应的用户 openID 支付宝：传入用户的 userID

接口具体请求数据的明文：data

```

<data>
  <request_type>unitive_js_pay_request</request_type>
  <dsn>669262B6A21A3B4E4610A78ACF29AE71</dsn>
  <req_seq>20152423432532532532525</req_seq>
  <total_fee>10000</total_fee>
  <channel>W</channel>
  <subopen_id>0z8tfs4224lgn3535353gdg</subopen_id>
</data>

```

接口响应定义：

节点名	必填	类型	描述	取值说明
req_seq	是	Number(V20)	请求流水号	调用接口时，本机生成的流水号，服务器返回该数据可用作校验。
result	是	String(F1)	请求结果	Y 请求成功/N 请求失败
code	是	String(F6)	响应代码	
msg	是	String(V100)	响应信息	服务对本次请求的文字描述，可通过查看 msg 了解本次请求成功、失败的具体原因。
trans_seq	否	Number(V20)	交易流水号	本地需要存储的交易流水号，即订单号，如果需要做退款处理，交易流水是唯一的退款凭证。
status	否	String(V20)	状态	W—等待支付
pay_url	否	String(V100)	预支付 ID	微信:prepay_id 支付宝:tradeNo
signature_info	否	String(V2000)	签名信息	json 格式，调用微支付、支付宝的 JS 支付 API 时，传入该内容
server_time	是	String(F19)	服务器时间	

接口响应数据示例：

```
<data>
<response_type>jspay_response</response_type>
<req_seq>01a7120ef21e4cafab3775a32f0781da</req_seq>
<res_seq>01703281332040246642</res_seq>
<result>Y</result>
<code>E11101</code>
<msg>预支付订单请求成功</msg>
<server_time>2017-03-28 13:32:04:595</server_time>
<dsn>1B9432BF29EF44CACF31C3538BAEAA95</dsn>
<channel>W</channel>
<channel_name>微支付</channel_name>
<trans_seq>11703281332040246643</trans_seq>
<status>W</status>
<pay_url>wx20170328133204411b2815c005343797</pay_url>

<signature_info>{"appId": "wxe132a10234235454", "nonceStr": "1d54621695eb4b8ca9533c430274ddc0", "package": "prepay_id=wx20170328133204411b2815c005343797", "paySign": "E0E74C0B6D8F2D9A7B6BD94EE8C806ED", "signType": "MD5", "timeStamp": "1490679124595"}</signature_info>
</data>
```

将响应 xml 中的 signature_info 值取出后传值前端页面，转换成 JSON 对象后按如下方式调用微信公众号或支付宝 JS 支付 SDK 接口，详情请参照微信或支付宝官方 JS 文档。

微信示例：

```
function exeWx(e) {
  ("undefined" == typeof WeixinJSBridge ? document.addEventListener ?
  document.addEventListener("WeixinJSBridgeReady", e, false) : document.attachEvent &&
  (document.attachEvent("WeixinJSBridgeReady", e), document.attachEvent("onWeixinJSBridgeReady", e)) :
  e());
};

exeWx(function(){
  WeixinJSBridge.invoke("getBrandWCPayRequest", signature_info, function(e) {
    switch (e.err_msg) {
      case "get_brand_wcpay_request:ok": alert("支付成功");return;
      case "get_brand_wcpay_request:cancel": break;
      default: alert("唤起微信支付失败, 请稍后再试(" + (e.err_code||-1) +")");break;
    };
  });
});
```

支付宝示例：

```
function exeZfb(e) {
```

```
( "undefined" == typeof AlipayJSBridge ? document.addEventListener ?
document.addEventListener("AlipayJSBridgeReady", e, false) : document.attachEvent &&
(document.attachEvent("AlipayJSBridgeReady", e), document.attachEvent("onAlipayJSBridgeReady", e)) :
e());
};

exeZfb(function(){
    AlipayJSBridge.call("tradePay", signature_info, function(e) {
        switch (e.resultCode) {
            case "9000": alert("支付成功");return;
            case "6001": break;
            default: alert("唤起支付宝失败, 请稍后再试("+ (e.resultCode||-1) +")");break;
        };
    });
});
```

6.2. 查询订单

参见 5.2 接口。

6.3. 退款接口

参见 5.3 接口。

6.4. 退款查询

参见 5.4 接口。

6.5. 支付结果通知

参见 5.5 接口。

7. H5 支付

7.1. 支付下单

用于提交第三方浏览器H 5 支付请求。

接口请求定义：

节点名	必填	类型	描述	取值说明
request_type	是	String(V50)	请求类型	unitive_h5_pay_request
dsn	1	String(F32)	设备序列号	
req_seq	是	String(V20)	请求流水号	
total_fee	是	Number(V10)	支付金额	单位:分
channel	是	String(F1)	支付渠道	W-微支付, A-支付宝
ip	是	String(V15)	用户公网 IP	传入用户所在公网 IP 注:channel 为 A 可不传入
redirect_url	是	String(V256)	用户中途取消或支付完成后用户端浏览器回调地址	注:channel 为 W 时该字段必须和支付网页域名一致

接口具体请求数据的明文： data

```
<data>
  <request_type>unitive_h5_pay_request</request_type>
  <dsn>669262B6A21A3B4E4610A78ACF29AE71</dsn>
  <req_seq>20152423432532532532525</req_seq>
  <total_fee>10000</total_fee>
  <channel>W</channel>
  <ip>125.71.132.197</ip>
  <redirect_url>http://xxx.com/call.html?xxx=xxxx</redirect_url>
</data>
```

接口响应定义：

节点名	必填	类型	描述	取值说明
req_seq	是	Number(V20)	请求流水号	调用接口时, 本机生成的流水号, 服务器返回该数据可用作校验。
result	是	String(F1)	请求结果	Y 请求成功/N 请求失败
code	是	String(F6)	响应代码	
msg	是	String(V100)	响应信息	服务对本次请求的文字描述, 可通过查看 msg 了解本次请求成功、失败的具体原因。
trans_seq	否	Number(V20)	交易流水号	本地需要存储的交易流水号, 即订单号, 如果需要做退款处理, 交易流水是唯一的退款凭证。

status	否	String(V20)	状态	W—等待支付
pay_info	否	String(V2000)	唤起支付URL	将用户浏览器指定到该地址即可拉起支付
server_time	是	String(F19)	服务器时间	

接口响应数据示例：

```
<data>
<response_type>h5pay_response</response_type>
<req_seq>17112210345837363773</req_seq>
<res_seq>01711221034580113745</res_seq>
<result>Y</result>
<code>E11101</code>
<msg>预支付订单请求成功</msg>
<server_time>2017-11-22 10:35:00:277</server_time>
<dsn>94414200128B2DE397416AC9EDF9236E</dsn>
<channel>W</channel>
<channel_name>微支付</channel_name>
<trans_seq>11711221034590166784</trans_seq>
<status>W</status>
<pay_info>https://wx.tenpay.com/cgi-bin/mmpayweb-bin/checkmweb?prepay_id=wx20171122103456f64260ef020009468000&package=1630946716&redirect_url=http%3A%2F%2Fxxx.com%2Fcall.html</pay_info>
</data>
```

7.2. 查询订单

参见 6.2 接口。

7.3. 退款接口

参见 5.3 接口。

7.4. 退款查询

参见 5.4 接口。

7.5. 支付结果通知

参见 6.5 接口。