# 支付宝SDK接入文档

目录

**[第1步： 下载支付宝sdk](#_Toc9379_WPSOffice_Level1)** **[2](#_Toc9379_WPSOffice_Level1)**

**[第2步： 客户端接入支付宝SDK](#_Toc30366_WPSOffice_Level1)** **[2](#_Toc30366_WPSOffice_Level1)**

[Android客户端接入：](#_Toc30366_WPSOffice_Level2) [2](#_Toc30366_WPSOffice_Level2)

[1. 同步通知:](#_Toc30204_WPSOffice_Level2) [9](#_Toc30204_WPSOffice_Level2)

[2. 异步通知：](#_Toc4515_WPSOffice_Level2) [10](#_Toc4515_WPSOffice_Level2)

[IOS客户端接入：](#_Toc4958_WPSOffice_Level2) [10](#_Toc4958_WPSOffice_Level2)

[第一种方式：通过 CocoaPods 导入](#_Toc27209_WPSOffice_Level2) [10](#_Toc27209_WPSOffice_Level2)

[第二种方式：手动导入](#_Toc673_WPSOffice_Level2) [11](#_Toc673_WPSOffice_Level2)

[}](#_Toc3586_WPSOffice_Level2) [13](#_Toc3586_WPSOffice_Level2)

**[第3步： 服务端接入平台支付API](#_Toc30204_WPSOffice_Level1)** **[18](#_Toc30204_WPSOffice_Level1)**

**[统一支付下单接口参数说明](#_Toc4515_WPSOffice_Level1)** **[18](#_Toc4515_WPSOffice_Level1)**

**[附件： 整个支付流程图](#_Toc4958_WPSOffice_Level1)** **[25](#_Toc4958_WPSOffice_Level1)**

**[附件：通道编码参考](#_Toc27209_WPSOffice_Level1)** **[27](#_Toc27209_WPSOffice_Level1)**

**第1步：** **下载支付宝sdk**

官方地址：<https://docs.open.alipay.com/54/104509/>。

平台地址：[http://www.starpay2019.com//Uploads/SDK/zfb.zip](http://www.starpay2019.com/Uploads/SDK/zfb.zip)

官方demo下载：<https://docs.open.alipay.com/54/104509>

平台demo下载：http://www.starpay2019.com//Uploads/SDK/sdk\_demo.zip

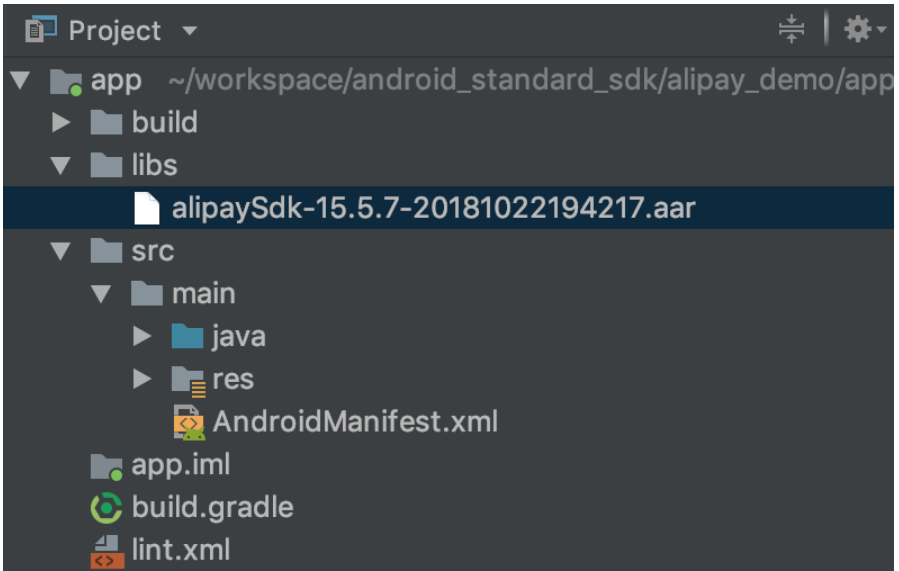
**第2步： 客户端接入支付宝SDK**

**Android客户端接入：**

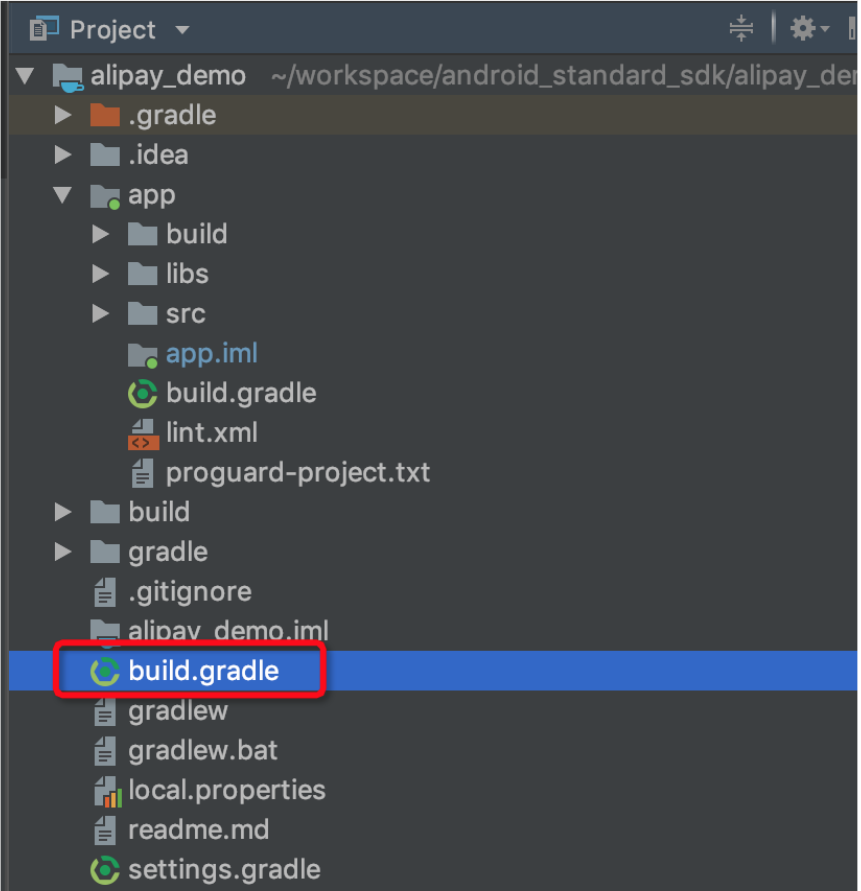
**首先客户端导入支付宝SDK:**

1. 将 alipaySdk-15.5.7-20181023110917.aar 包放在您的应用工程的 libs 目录下（图片中文件名仅做示例，请以实际 SDK 文件名为准）：

示例如下：



1. 在主项目的 build.gradle 中，添加下面的内容，将 libs 目录作为依赖仓库



在上面选择的文件中添加下面内容：

allprojects {

repositories {

// 添加下面的内容

flatDir {

dirs 'libs'

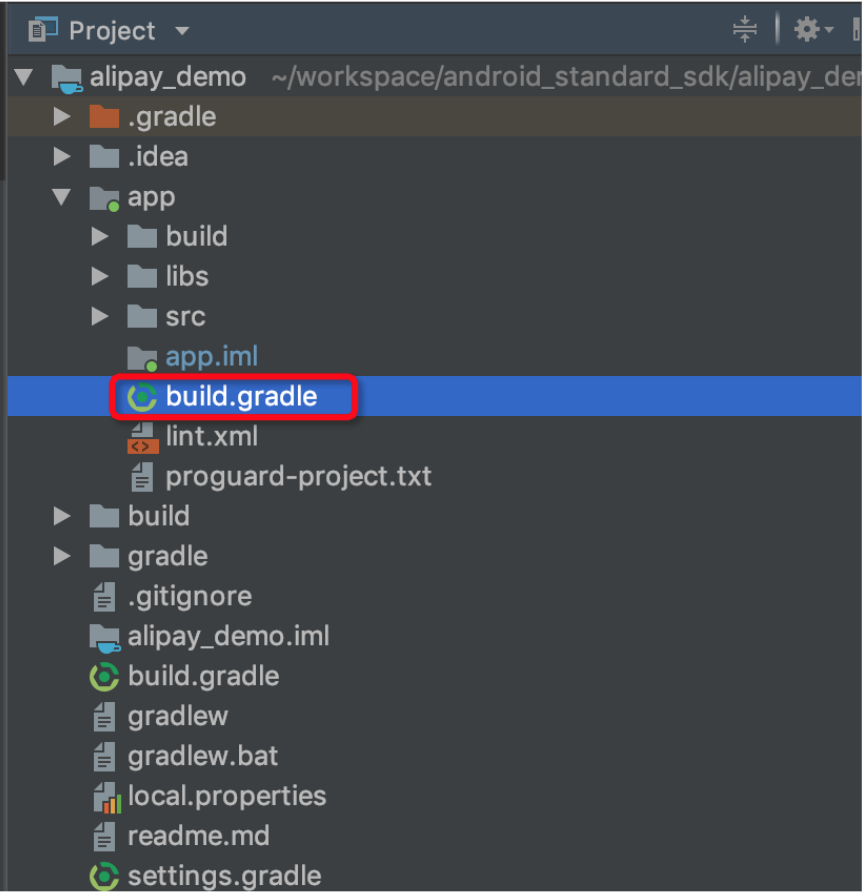
}

// ... jcenter() 等其他仓库

}

}

1. 在您 App Module 的 build.gradle 中，添加下面的内容，将支付宝 SDK 作为项目依赖：



在上面选择的文件中添加下面内容：

dependencies {

// 添加下面的内容

compile (name: 'alipaySdk-15.5.7-20181023110917', ext: 'aar')

// ... 其他依赖项

}

**至此，支付宝 SDK 开发资源导入完成。**

**然后获取运行权限:**

1. 为正常完成良好的支付流程体验，支付宝 SDK 需要使用下面这些权限：

android.permission.INTERNET

android.permission.ACCESS\_NETWORK\_STATE

android.permission.ACCESS\_WIFI\_STATE

您需要在 AndroidManifest 里配置以上 3 个权限，支付宝 SDK 在运行时需要进行网络连接，并在必要的时候判断网络连接的状态（4G/Wi-Fi）等来进行支付体验的优化。

**再然后进行支付接口调用:**

1. 需要在新线程中调用支付接口（可参考 alipay\_demo 实现）。

PayTask 对象主要为商户提供订单支付、查询功能，及获取当前开发包版本号。  
获取 PayTask 支付对象调用支付（支付行为需要在独立的非ui线程中执行），代码示例：

final String orderInfo = info; // 订单信息

Runnable payRunnable = new Runnable() {

@Override

public void run() {

PayTask alipay = new PayTask(DemoActivity.this);

    Map <String,String> result = alipay.payV2(orderInfo,true);

Message msg = new Message();

msg.what = SDK\_PAY\_FLAG;

msg.obj = result;

mHandler.sendMessage(msg);

}

};

// 必须异步调用

Thread payThread = new Thread(payRunnable);

payThread.start();

|  |  |  |
| --- | --- | --- |
| **orderInfo** | **String** | app支付请求参数字符串，主要包含商户的订单信息，key=value形式，以&连接。从支付平台通过统一下单接口获取。建议形式：商户客户端->**商户服务器**->支付平台->**返回参数给商户服务器**->客户端 |
| **isShowPayLoading** | **boolean** | 建议将该值设置为 true，优化点击付款到支付唤起支付页面的过渡过程。 |

参数说明：

orderInfo参数结果示例**：**

app\_id=2015052600090779&biz\_content=%7B%22timeout\_express%22%3A%2230m%22%2C%22seller\_id%22%3A%22%22%2C%22product\_code%22%3A%22QUICK\_MSECURITY\_PAY%22%2C%22total\_amount%22%3A%220.02%22%2C%22subject%22%3A%221%22%2C%22body%22%3A%22%E6%88%91%E6%98%AF%E6%B5%8B%E8%AF%95%E6%95%B0%E6%8D%AE%22%2C%22out\_trade\_no%22%3A%22314VYGIAGG7ZOYY%22%7D&charset=utf-8&method=alipay.trade.app.pay&sign\_type=RSA2&timestamp=2016-08-15%2012%3A12%3A15&version=1.0&sign=MsbylYkCzlfYLy9PeRwUUIg9nZPeN9SfXPNavUCroGKR5Kqvx0nEnd3eRmKxJuthNUx4ERCXe552EV9PfwexqW%2B1wbKOdYtDIb4%2B7PL3Pc94RZL0zKaWcaY3tSL89%2FuAVUsQuFqEJdhIukuKygrXucvejOUgTCfoUdwTi7z%2BZzQ%3D

**最后支付结果获取和处理：**

调用 pay 方法支付后，将通过2种途径获得支付结果：

1. **同步通知**:

商户应用客户端通过当前调用支付的Activity的Handler对象，通过它的回调函数获取支付结果。（可参考 alipay\_demo 实现）  
代码示例：

private Handler mHandler = new Handler() {

public void handleMessage(Message msg) {

Result result = new Result((String) msg.obj);

Toast.makeText(DemoActivity.this, result.getResult(),

Toast.LENGTH\_LONG).show();

};

};

1. **异步通知**：

通过商户上传的pay\_notifyurl进行支付结果的通知（形式post）。详细见《API支付接口说明文档》->异步通知。

**获取当前开发包版本号**

调用 PayTask 对象的 getVersion() 方法查询。  
代码示例：

PayTask payTask = new PayTask(activity);

String version = payTask.getVersion();

**支付宝Android-SDK的客户端接入到此为止。**

**IOS客户端接入：**

**首先客户端导入支付宝ios-SDK:**

第一种方式：通过 CocoaPods 导入

**pod '**AlipaySDK-iOS**'**

## 第二种方式：手动导入

启动 IDE（如 Xcode），把 [iOS 包](https://docs.open.alipay.com/54/104509)中的压缩文件中以下文件拷贝到项目文件夹下，并导入到项目工程中。

AlipaySDK.bundle

AlipaySDK.framework

在 Build Phases 选项卡的 Link Binary With Libraries 中，增加以下依赖：



其中，需要注意的是：

1. 如果是 Xcode 7.0之后的版本，需要添加 libc++.tbd、libz.tbd；
2. 如果是 Xcode 7.0之前的版本，需要添加 libc++.dylib、libz.dylib（如下图所示）。



**增加头文件引用**

在需要调用 AlipaySDK 的文件中，增加头文件引用。

import <AlipaySDK/AlipaySDK.h>

**组装请求信息**

注意：获取orderInfo参数的逻辑应在商户服务端完成，商户服务端向支付平台获取 orderInfo，然后传给客户端，客户端直接传给 SDK 发起请求。官方Demo 是为了示例效果在客户端实现。

// NOTE: 获取到服务器返回的orderInfo参数，则继续执行支付

if (orderInfo!= nil) {

//应用注册scheme,在AliSDKDemo-Info.plist定义URL types

NSString \*appScheme = @"alisdkdemo";

// NOTE: 调用支付结果开始支付

[[AlipaySDK defaultService] payOrder:orderInfo fromScheme:appScheme callback:^(NSDictionary \*resultDic) {

NSLog(@"reslut = %@",resultDic);

}];

}

详细可参见 [Demo](https://docs.open.alipay.com/54/104509) 中示例文件：

AliSDKDemo\APViewController.h

AliSDKDemo\APViewController.m

AliSDKDemo\Order.h

AliSDKDemo\Order.m

**配置返回 url 处理方法**

本步骤指引开发者配置支付宝客户端返回 url 处理方法（外部存在支付宝钱包，支付宝钱包将处理结果通过 url 返回）。

如示例 AliSDKDemo\APAppDelegate.m 文件中，增加引用代码：

import <AlipaySDK/AlipaySDK.h>

将 @implementation AppDelegate 中以下代码中的 NSLog 改为实际业务处理代码：

- (BOOL)application:(UIApplication \*)application openURL:(NSURL \*)url sourceApplication:(NSString \*)sourceApplication annotation:(id)annotation

{

if ([url.host isEqualToString:@"safepay"]) {

//跳转支付宝钱包进行支付，处理支付结果

[[AlipaySDK defaultService] processOrderWithPaymentResult:url standbyCallback:^(NSDictionary \*resultDic) {

NSLog(@"result = %@",resultDic);

}];

}

return YES;

}

// NOTE: 9.0以后使用新API接口

- (BOOL)application:(UIApplication \*)app openURL:(NSURL \*)url options:(NSDictionary<NSString\*, id> \*)options

{

if ([url.host isEqualToString:@"safepay"]) {

//跳转支付宝钱包进行支付，处理支付结果

[[AlipaySDK defaultService] processOrderWithPaymentResult:url standbyCallback:^(NSDictionary \*resultDic) {

NSLog(@"result = %@",resultDic);

}];

}

return YES;

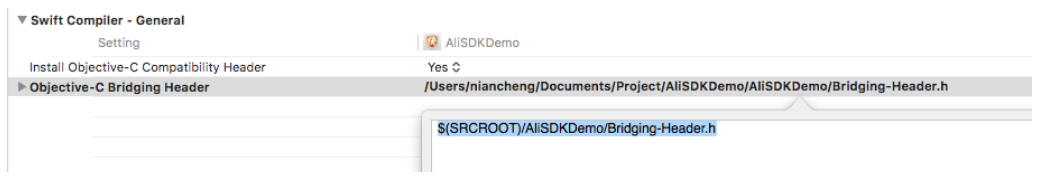
}

**针对 Swift 接入 AlipaySDK 的问题**

如果项目使用 Swift 为开发语言，需要添加桥接文件，如 Bridging-Header.h。

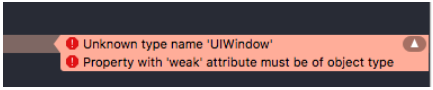


同时，在项目 Build Settings 中设置桥接文件的位置（如下图所示）。



添加成功后，在桥接文件中写入 #import <AlipaySDK/AlipaySDK.h>。  
如此，在需要调用 AlipaySDK 的地方，即可直接使用，具体调用方式参照 OC。

注意：  
运行时如果发生以下报错，



则在桥接文件中，写入#import <UIKit/UIKit.h>。

**iOS调用说明**

接口名称：AlipaySDK

接口描述：提供支付功能。

Alipay 接口主要为商户提供订单支付功能。接口所提供的方法，如下表所示：

|  |  |
| --- | --- |
| 方法名称 | 方法描述 |
| +(Alipay \*)defaultService; | 获取服务实例。 |
| -(void)payOrder:(NSString\*)orderInfo  fromScheme:(NSString\*)schemeStr callback:(CompletionBlock)completionBlock; | 支付并通过回调返回结果。 |
| -(void)processOrderWithPaymentResult:(NSURL\*)resultUrl standbyCallback:(CompletionBlock)completionBlock; | 处理支付宝客户端返回的url（在app被杀模式下，通过这个方法获取支付结果）。 |

**订单支付 iOS接口-payOrder**

方法名称：pay方法

方法原型：(void)payOrder:(NSString )orderInfo fromScheme:(NSString )schemeStr callback:(CompletionBlock)completionBlock;

方法功能：提供给商户快捷订单支付功能。

|  |  |
| --- | --- |
| 参数名称 | 参数描述 |
| NSString\* scheme | 商户程序注册的URL protocol，供支付完成后回调商户程序使用 |
| (CompletionBlock)completionBlock | 支付开发包同步回调函数，返回免登、支付结果。本地未安装支付宝客户端，或未成功调用支付宝客户端进行支付的情况下（走H5收银台），会通过该completionBlock返回支付结果 |
| NSString\* orderInfo | |  | | --- | | app支付请求参数字符串，主要包含商户的订单信息，key=value形式，以&连接。从支付平台通过统一下单接口获取。建议形式：商户客户端->**商户服务器**->支付平台->**返回参数给商户服务器**->客户端 | |

orderInfo参数结果示例**：**

app\_id=2015052600090779&biz\_content=%7B%22timeout\_express%22%3A%2230m%22%2C%22seller\_id%22%3A%22%22%2C%22product\_code%22%3A%22QUICK\_MSECURITY\_PAY%22%2C%22total\_amount%22%3A%220.02%22%2C%22subject%22%3A%221%22%2C%22body%22%3A%22%E6%88%91%E6%98%AF%E6%B5%8B%E8%AF%95%E6%95%B0%E6%8D%AE%22%2C%22out\_trade\_no%22%3A%22314VYGIAGG7ZOYY%22%7D&charset=utf-8&method=alipay.trade.app.pay&sign\_type=RSA2&timestamp=2016-08-15%2012%3A12%3A15&version=1.0&sign=MsbylYkCzlfYLy9PeRwUUIg9nZPeN9SfXPNavUCroGKR5Kqvx0nEnd3eRmKxJuthNUx4ERCXe552EV9PfwexqW%2B1wbKOdYtDIb4%2B7PL3Pc94RZL0zKaWcaY3tSL89%2FuAVUsQuFqEJdhIukuKygrXucvejOUgTCfoUdwTi7z%2BZzQ%3D

**处理客户端同步返回 url接口-completionBlock**

方法名称：处理客户端同步返回方法

方法原型：-(void)processOrderWithPaymentResult:(NSURL\*)resultUrl standbyCallback:(CompletionBlock)completionBlock;

方法功能：设备已安装支付宝客户端情况下，处理支付宝客户端返回的url。

|  |  |
| --- | --- |
| 参数名称 | 参数描述 |
| NSURL \*resultUrl | 支付宝app客户端回传的url |
| (CompletionBlock)completionBlock | 本地安装了支付宝客户端，且成功调用支付宝客户端进行支付的情况下，会通过该completionBlock返回支付结果 |

备注：

1.请在APPDelegate的 -(BOOL)application:(UIApplication\*)application openURL:(NSURL\*)url

sourceApplication:(NSString\*)sourceApplication annotation:(id)annotation

中调用该方法，iOS9.0及以上请在 - (BOOL)application:(UIApplication \*)app openURL:(NSURL \*)url options:(NSDictionary<NSString\*, id> \*)options 中调用该方法，具体可参见 [Demo](https://docs.open.alipay.com/54/104509" \t "https://docs.open.alipay.com/204/105295/_blank)。

2.在支付过程结束后，会通过 callbackBlock 同步返回支付结果（callbackBlock 是调用支付同步的回调）。支付结果中参数的提取，必须通过 CompletionBlock 获取，禁止开发者私自解析支付结果返回的 URL。

**支付宝IOS-SDK的客户端接入到此为止。**

# 注释： 如果在使用运行支付宝官方SDK-demo的过程中遇到问题，请查看官方文档中的《**针对Demo的运行注意**》：

地址：<https://docs.open.alipay.com/204/105295/>

**第3步： 服务端接入平台支付API**

1. **首先服务器端不区分Android和IOS，统一实现。**
2. **接入统一下单接口API，通过接口获取客户端官方SDK支付需要的orderInfo参数。实际使用方式为：客户端发起请求->到商户服务器获取参数，然后商户服务器通过统一下单接口->到支付平台获取orderInfo并返回给客户端，客户端使用orderInfo调用SDK发起支付。**
3. **支付完成后，平台通知支付结果。通知地址为调用统一下单接口时上传的pay\_notifyurl的地址。**

**如下为SDK支付的API说明，更多接口请查看《API支付接口说明文档》。**

# 统一支付下单接口参数说明

Post支付网关地址： 平台网关/Pay\_Index.html

注意，提交方式是 POST ，请不要用 GET 方式提交

#### 请求参数：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **参数名称** | **参数类型** | **参数含义** | **是否必填** | **参与签名** | **参数说明** |
| pay\_memberid | String | 商户号 | 是 | 是 | 平台分配商户号 |
| pay\_orderid | String | 订单号 | 是 | 是 | 上送订单号唯一, 字符长度不超过26位 |
| pay\_userid | String | 支付用户ID | 是 | 是 | 请传您发起支付的应用内用户的userid，最好是正式ID，也可以是任意可以唯一标识该用户的ID，如果全为相同ID会触发限制(字符最大长度20，如随意上传导致平台资金账号被封。经核查后,会冻结相应资金) |
| pay\_applydate | String | 提交时间 | 是 | 是 | 时间格式：2016-12-26 18:18:18 |
| pay\_bankcode | String | 通道编码 | 是 | 是 | 参考后续说明 |
| pay\_notifyurl | String | 服务端通知 | 是 | 是 | 服务端返回地址.（POST返回数据） |
| pay\_callbackurl | String | 页面跳转通知 | 是 | 是 | 页面跳转返回地址（POST返回数据） |
| pay\_amount | String | 订单金额 | 是 | 是 | 单位：元。例如：1元为1.00 |
| pay\_md5sign | String | MD5签名 | 是 | 否 | 请看MD5签名字段格式 |
| pay\_productname | String | 商品名称 | 是 | 否 |  |
| pay\_attach | String | 附加字段 | 否 | 否 | 此字段在返回时按原样返回 (中文需要url编码) |
| pay\_productnum | String | 商户品数量 | 否 | 否 |  |
| pay\_productdesc | String | 商品描述 | 否 | 否 |  |
| pay\_producturl | String | 商户链接地址 | 否 | 否 |  |

Response返回值(返回数据格式为json)：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **参数名称** | **参数类型** | **参数含义** | **参与签名** | **参数说明** |
| status | String | 请求状态 | 否 | success表示成功，其他为失败 |
| msg | String | 状态描述 | 否 | status=success时为”成功”，其他为响应的错误说明 |
| data | String | 返回数据 | 否 | 详见下表。 |

data数据说明:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **参数名称** | **参数类型** | **参数含义** | **参与签名** | **参数说明** |
| payurl | String | **非SDK支付**，发起地址 | 是（存在则参与签名） | 【status=success，且为**非SDK支付**时返回】，请浏览器打开该链接发起支付。切记不可用跳转方式实现，否则微信容易验证不通过。 |
| orderInfo | String | **SDK支付**，支付订单信息 | 是（存在则参与签名） | 【status=success，且为**SDK支付**时返回】，请客户端app调用对应官方sdk支付的时候，使用该参数返回的数据。详细见《SDK支付接口说明文档》   1. 支付宝，直接使用。 2. 微信，通过json解析后赋值对应参数使用。 |
| sign | String | MD5签名 | 是 | 请看MD5签名字段格式 |

支付类型说明：

**SDK支付**：通过API下单，然后调用支付SDK进行支付的方式。

**非SDK支付**：通过API下单，然后调用浏览器进行支付的方式。

示例（Json）：

{

"status":"success",

"msg":"成功",

"data":{

“orderInfo”:”payid=10000”,

"payurl":"http://www.starpay2019.com"

}

}

#### 然后通过浏览器访问payurl地址，完成支付。

#### 签名算法：

签名生成的通用步骤如下：  
第一步，设所有发送或者接收到的数据为集合M，将集合M内非空参数值的参数按照参数名ASCII码从小到大排序（a-z字典序），使用URL键值对的格式（即key1=value1&key2=value2…）拼接成字符串stringA。  
第二步，在stringA最后拼接上ApiKey得到stringSignTemp字符串，并对stringSignTemp进行MD5运算，再将得到的字符串所有字符转换为大写，得到sign值signValue。

1. stringSignTemp="pay\_amount=pay\_amount&pay\_applydate=pay\_applydate&pay\_bankcode=pay\_bankcode&pay\_callbackurl=pay\_callbackurl&pay\_memberid=pay\_memberid&pay\_notifyurl=pay\_notifyurl&pay\_orderid=pay\_orderid&pay\_userid=pay\_userid&key=ApiKey"
2. sign=MD5(stringSignTemp).toUpperCase()

#### 异步通知（返回正式支付结果）（Post形式，数据为form表单, 通知地址：支付时提交的pay\_notifyurl）：

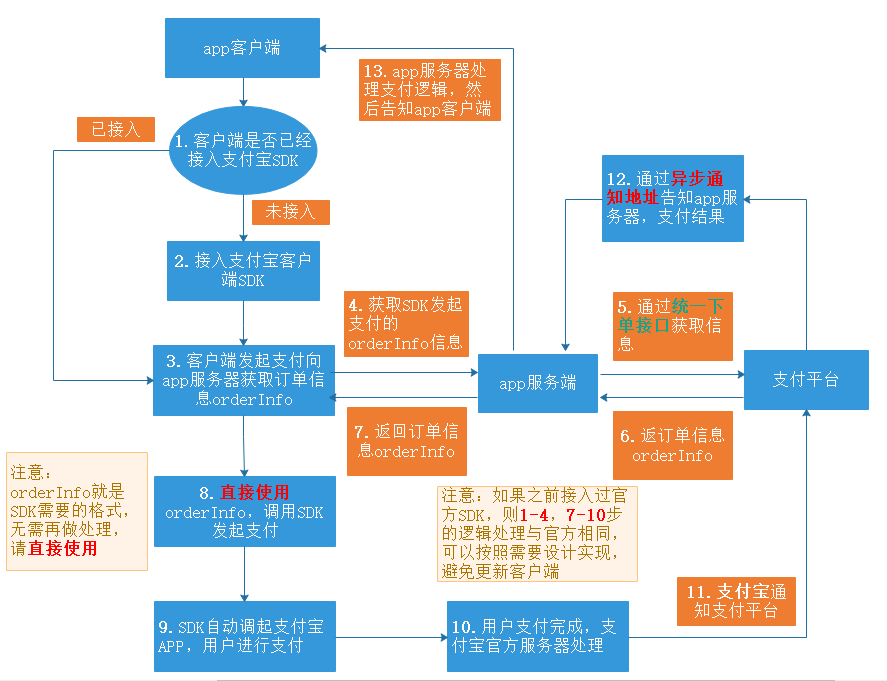
如果接收到服务器点对点通讯时，在页面输出“OK”（没有双引号，OK两个字母大写）,否则会重复5次发送点对点通知. 请务必进行结果的sign验证，确保支付的正确性。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **参数名称** | **参数类型** | **参数含义** | **参与签名** | **参数说明** |
| memberid | String | 商户编号 | 是 |  |
| orderid | String | 订单号 | 是 |  |
| amount | String | 订单金额 | 是 | 单位：元 |
| transaction\_id | String | 交易流水号 | 是 |  |
| datetime | String | 交易时间 | 是 |  |
| returncode | String | 交易状态 | 是 | “00” 为成功，其他为失败 |
| attach | String | 扩展返回 | 否 | 商户附加数据返回 |
| sign | String | 签名 | 否 | 请看验证签名字段格式 |

*[danger]注：签名见签名算法。*

returncode参数说明： 00：表示成功 01：通道返回验证签名不成功 02：订单金额异常 03：订单状态修改失败 04：结算资金保存失败 05：商户数据修改失败 06：资金变动记录添加失败 07：投诉保证金添加失败

**附件： 整个支付流程图**



**附件：通道编码参考**

附：此编码仅供参考。最新通道编码请前往后台查看。商户中心->API管理->查看通道费率->通道的编号 即是通道编码 通道编码随时有变动的可能，请以商户后台显示的为准。

|  |  |
| --- | --- |
| **银行编码** | **银行名称** |
| 901 | 微信H5 |
| 902 | 微信扫码支付 |
| 903 | 支付宝扫码支付 |
| 904 | 支付宝手机 |
| 905 | QQ手机支付 |
| 907 | 网银支付 |
| 908 | QQ扫码支付 |
| 909 | 百度钱包 |
| 910 | 京东支付 |
| 911 | 快捷支付 |
| 912 | 银联H5 |
| 916 | 快捷支付H5 |
| 918 | 银联直连 |
| 919 | 微信JSP |
| 920 | 微信H5【包户】 |
| 921 | 支付宝H5【包户】 |
| 922 | 支付宝APP |
| 923 | 微信APP |

*可以在支付产品添加自定义的支付产品，具体以自己的后台的通道编码为准。*