|  |  |
| --- | --- |
| 版 本 号 | 2016.2.2 |
| 密 级 | 普通 |

# Server SDK文档



目录

[Server SDK文档 1](#_Toc448391859)

[1 概述 5](#_Toc448391860)

[2 快速指引 5](#_Toc448391861)

[3 公共请求参数 5](#_Toc448391862)

[4 查询APP可用支付渠道列表 6](#_Toc448391863)

[4.1 请求参数 6](#_Toc448391864)

[4.2 调用示例 7](#_Toc448391865)

[4.2.1 JAVA语言 7](#_Toc448391866)

[4.2.2 Python语言 7](#_Toc448391867)

[4.2.3 NodeJs语言 8](#_Toc448391868)

[4.2.4 php语言 8](#_Toc448391869)

[4.3 响应示例 9](#_Toc448391870)

[5 预支付请求 11](#_Toc448391871)

[5.1 概述 11](#_Toc448391872)

[5.2 请求参数 12](#_Toc448391873)

[5.3 调用示例 14](#_Toc448391874)

[5.3.1 JAVA语言 14](#_Toc448391875)

[5.3.2 Python语言 14](#_Toc448391876)

[5.3.3 NodeJs语言 15](#_Toc448391877)

[5.3.4 php语言 16](#_Toc448391878)

[5.4 响应 17](#_Toc448391879)

[6 退款请求 18](#_Toc448391880)

[6.1 请求参数 18](#_Toc448391881)

[6.2 调用示例 19](#_Toc448391882)

[6.2.1 JAVA语言 19](#_Toc448391883)

[6.2.2 Python语言 19](#_Toc448391884)

[6.2.3 NodeJs语言 20](#_Toc448391885)

[6.2.4 php语言 21](#_Toc448391886)

[6.3 响应 22](#_Toc448391887)

[7 支付查询请求 23](#_Toc448391888)

[7.1 请求参数 23](#_Toc448391889)

[7.2 调用示例 23](#_Toc448391890)

[7.2.1 JAVA语言 23](#_Toc448391891)

[7.2.2 Python语言 23](#_Toc448391892)

[7.2.3 NodeJs语言 24](#_Toc448391893)

[7.2.4 php语言 25](#_Toc448391894)

[7.3 响应 25](#_Toc448391895)

[8 退款查询请求 26](#_Toc448391896)

[8.1 请求参数 26](#_Toc448391897)

[8.2 调用示例 26](#_Toc448391898)

[8.2.1 JAVA语言 26](#_Toc448391899)

[8.2.2 Python语言 27](#_Toc448391900)

[8.2.3 NodeJs语言 27](#_Toc448391901)

[8.2.4 php语言 28](#_Toc448391902)

[8.3 响应 29](#_Toc448391903)

1. 概述

魔派Server SDK 提供了API的请求封装、摘要签名、响应解悉等功能，使用SDK可以轻松完成API的调用及结果的获取。sdk基于https请求与魔派服务端通信，提供多种不同语言的sdk供使用。

1. 快速指引

首先您需要先完成以下几步：1、注册魔派用户🡪2、创建应用；

然后就可以开始开发程序了，对于使用移动终端进行支付的场景:

1、商户服务端收到用户支付请求.

2、商户服务端通过Server SDK发起预支付请求(见第5节预支付请求).

3、商户服务端将Server SDK返回的JSON串内容发送给移动终端.

4、移动终端调用Client SDK支付接口,传入返回的JSON串内容作为参数.

5、Client SDK将唤醒相应渠道支付控件引导用户完成支付.

开发程序的同时可以进行如下工作：

1、魔派官网提交企业认证、签约

2、在各支付渠道开通账户、签约相关服务

3、在魔派官网配置各支付渠道相关参数

1. 公共请求参数
   1. 请求地址：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 参数名称 | 描述 | 范例 |
| ServerRoot | 魔派SERVER SDK API请求地址 | 末尾注意不要加斜杠 | 测试:  https://220.181.25.235/yop-center  生产:  https://www.mobpex.com/yop-center |

* 1. 参数：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 参数 | 参数名称 | 类型 | 必填 | 描述 | 范例 |
| appId | 开发者的AppId | String | 是 | 分配给开发者的应用Id,可登录官司网查询 | 15122404177104893655 |
| userId | 商户注册的用户id | String | 是 | 用户登录id，同注册时使用的邮箱地址。 | test168888@datayp.com |
| liveMode | 是否生产模式 | boolean | 否 | 是否生产模式，可以不传该参数，默认值为true。 | true |
| secretKey | 密钥 | String | 是 | 密钥。请登录官网查询或修改密钥，如果使用测试模式接入请使用Test 密钥，如果使用生产LIVE模式，请使用Live密钥 | FDSAFSDAF233 |
| methodOrUri | 请求uri | String | 是 | 请求uri，是API的唯一标识,在调用MobpexClient.post方法中指定此参数，不同API的methodOrUri值参见下文各API的请求参数说明 | /rest/v1.0/pay/unifiedOrder |
| ignoreSSLCheck | 是否忽略SSL域名检查 | Boolean | 否 | 是否忽略SSL域名检查 | 默认false |
| signRet | 是否签名 | Boolean | 否 | 是否对请求和响应的数据进行签名 | 默认true |
| signAlg | 签名算法 | String | 否 | api签名算法 | 默认MD5 |

1. 查询APP可用支付渠道列表
   1. 请求参数

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | 类型 | 是否必须 | 示例 | 描述 |
| methodOrUri | String | 必须 | /rest/v1.0/query/findChannelInfoByAppId |  |

* 1. 调用示例
     1. JAVA语言

|  |
| --- |
| MobpexConfig.setServerRoot("https://220.181.25.235/yop-center");  MobpexConfig.setIgnoreSSLCheck(true);  //魔派对外提供的接口地址  MobpexConfig.*setAppId*("15120409211235616710");//企业在魔派注册的应用ID；  MobpexConfig.*setSecretKey*("fwefwe1ffdsa2");//密钥值，参见公共参数说明  MobpexRequest request = new MobpexRequest();  request.setUserId( "test168888@datayp.com");  MobpexResponse response = MobpexClient.*post*(  "/rest/v1.0/query/findChannelInfoByAppId",  request);  if(response.isValidSign()){ //判断验签是否通过  System.out.println(response.getContent());  // response.getContent()可得到响应报文  } |

* + 1. Python语言

|  |
| --- |
| #密钥值，参见公共参数说明  secretKey = 'TS212fdsfdsfdsfsfsfsdfdsfds12241421g'  appId = '15122404163671048936'  userId = 'test@test.com'  serverRoot = 'https://220.181.25.235/yop-center'  ignoreSSLCheck='true'  liveMode='false'  #获取客户端对象  client = MobpexPythonSDK.base.Client(appId,userId,serverRoot,secretKey,ignoreSSLCheck,  liveMode= liveMode)  #发起请求  req = MobpexPythonSDK.base.TopRequest('/rest/v1.0/query/findChannelInfoByAppId')  jsonObj = client.getResponse(req)  #验证签名  flag =client.validResult(jsonObj)  if flag:  logging.info('签名验证成功! ')  else:  logging.info('签名验证失败!') |

* + 1. NodeJs语言

|  |
| --- |
| //客户端对象  var client = new Client({  'appId':'15122404163671048936',  'secretKey':'TS212fdsfdsfdsfsfsfsdfdsfds12241421g',  'serverRoot':'https://220.181.25.235/yop-center',  'userId':'test@test.com'  });  //请求参数  var params={  'liveMode': 'true'  };  //发起请求  client.request("/rest/v2.0/query/findChannelInfoByAppId",  params,function (error,response) {  if(!error){  var content=response.body;  //验证签名  var flag=client.certifyResponse(content);  if(flag){  console.log("签名验证成功！");  console.log(response.body);  }  else{  console.log("签名验证失败！");  console.log(response.body);  }  }  else{  console.log(error);  }  }); |

* + 1. php语言

|  |
| --- |
| include\_once('MobpexSdk.php');  $client = new MobpexClient('https://220.181.25.235/yop-center', //Mobpex对外提供的地址  '/rest/v1.0/query/findChannelInfoByAppId', //接口对应的uri  false, //是否需要进行证书验证  '15122404366710489367', //appid  'LS\_212fdsfdsfdsfsfsfsdfdsfds12241421'); //密钥值  $client->set\_param('userId', 'long.chen-1@yeepay.com'); //设置用户id  $result = $client->execute(); //对于不需要session的api，则可以不用session参数  if($client->validSign($result)){ //返回结果的验签x  var\_dump($result);  } |

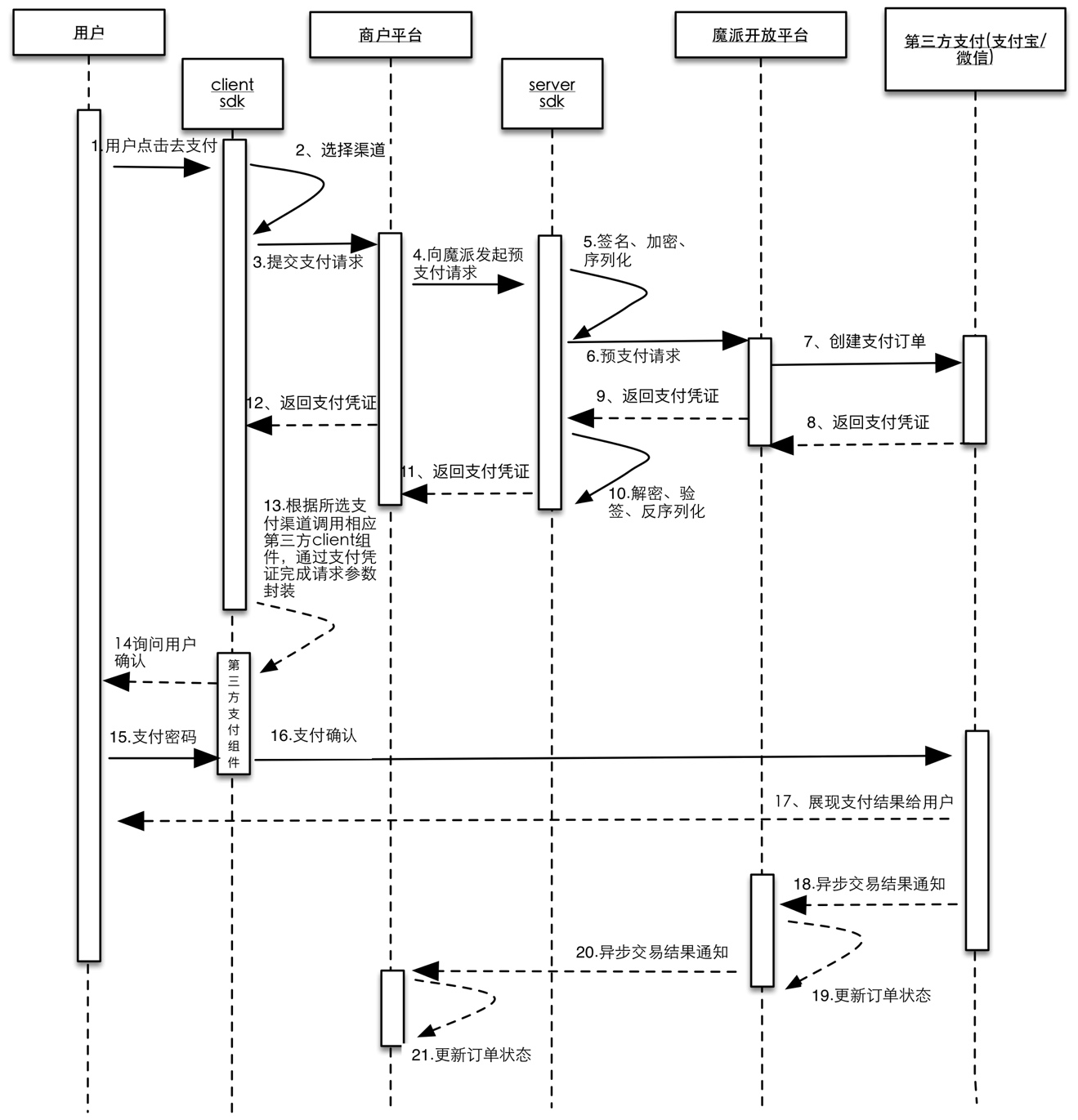
* 1. 响应示例

|  |
| --- |
| {  "state": "SUCCESS",//操作是否成功，如果为SUCCESS则为成功  "result": [ //result的内容为具体业务的响应结果  {  "createdDate": "2015-12-08 10:27:55", //创建日期  "lastModifiedDate": "2016-01-15 09:28:19",//最后修改时间  "id": 1,//时间  "channelCode": "ALIPAY",//支付渠道编码  "channelName": "支付宝移动支付",//支付渠道名称  "needPrepayRequest": false,// 通道方是否需要生成预支付订单  "type": "APP",// 支付方式 目前有APP、WEB、WAP、PUB几种  //PUB是微信特有的支付方式，表示的是公众号，最新支付方式以官网开发文档页面为准  "scene": "", //应用场景  "applyCharge": "10000",//申请费用  "applyCycle": "180",//申请周期  "belongWith": "支付宝",//归属公司  "detailsPage": "http://www.qq.com",//详情页面地址  "currency": "CNY",//默认币种  "available": true,// 通道可用性  "remark": "qweqwe"//备注  },  {  "createdDate": "2015-12-08 10:28:56",  "lastModifiedDate": "2015-12-08 10:28:56",  "id": 2,  "channelCode": "WECHAT",  "channelName": "微信APP支付",  "needPrepayRequest": false,  "type": "WAP",  "scene": "",  "belongWith": "微信",  "currency": "CNY",  "available": true,  "remark": "qweqwe"  },  {  "createdDate": "2015-12-08 16:38:38",  "lastModifiedDate": "2015-12-15 20:25:12",  "id": 3,  "channelCode": "YEEPAY",  "channelName": "一键支付",  "needPrepayRequest": false,  "type": "WEB",  "scene": "",  "belongWith": "易宝支付",  "currency": "CNY",  "available": true,  "remark": "qweqweasd"  },  {  "createdDate": "2015-12-15 20:50:18",  "lastModifiedDate": "2015-12-17 13:58:59",  "id": 4,  "channelCode": "UPACP",  "channelName": "银联移动支付",  "needPrepayRequest": false,  "type": "APP",  "scene": "",  "applyCharge": "qweqw",  "applyCycle": "qweqw",  "belongWith": "中国银联",  "currency": "CNY",  "available": true,  "remark": "asdad"  }  ],  "ts": 1455519531701,  "format": "json",  "validSign": true  } |
|  |

1. 预支付请求
   1. 概述

用户点击某个支付渠道进行支付后，商户后台系统组织参数发起预支付请求，通过魔派获得支付凭证，响应报文中的paymentParams的内容是第三方支付渠道返回的支付凭证。获得正确的支付凭证后，商户系统将获得的支付凭证返回给手机客户端，手机客户端调用Client sdk相关组件来唤醒第三方支付渠道UI控件向用户进行支付确认。

交易流程时序图



* 1. 请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类型 | 是否必须 | 示例 |
| methodOrUri | String | 必须 | /rest/v1.0/pay/unifiedOrder |
| prePayRequest | JSON字符串 | 必须 | {  "productName": "维多利亚的秘密",  "productDescription": "G cup",  "payType": "APP",  "payChannel": "WECHAT",  "payCurrency": "CNY",  "amount": "200",  "deviceId": "1183224081",  "deviceType": "MOBILE",  "requestCustomerId": "",  "tradeNo": "1134376595",  "requestIp": "127.0.0.1",  "requestIdentification": "iPhone 7s plus pro 128T",  "appId": "15122404177104893671",  "liveMode": false  } |

参数名：prePayRequest

参数值示例（JSON字符串）：

prePayRequest的JSON串的构成：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | 类型 | 是否必须 | 示例 | 描述 |
| productName | String | 是 | 维多利亚的秘密 | 本次购买商品或服务名称 |
| productDescription | String | 是 | G cup | 描述 |
| payType | String | 是 | APP | 支付类型 |
| payChannel | String | 是 | WECHAT | 支付渠道。商户已开通的渠道可通过APP可用渠道列表API获得，此处的传值要和APP渠道列表中相应支付渠道的channelCode值一致 |
| notifyUrl | String | 否 |  | Mobpex在订单状态发生变化后会通过这个地址后台回调通知商户，如果不填则使用商户在Mobpex后台设定的Webhook url进行回调通知。具体协议参见：http://weixin.mobpex.com/mashup-mobpex/callBackSdk |
| returnUrl | String | 是 |  | 在用户完成支付、取消支付、支付失败后将跳转的地址。具体协议参见：http://weixin.mobpex.com/mashup-mobpex/callBackSdk |
| payCurrency | String | 是 | CNY | 货币单位 |
| amount | String | 是 | 200.00 | 合计金额，精确到小数点后2位 |
| extra | String | 微信公众号必须 | {"openid":" ogiZrwN21-GeLECkfdX6ELpy97NY"} | 扩展参数，在使用微信公众号下单时需要传入openid，具体openid的获取请参照：https://pay.weixin.qq.com/wiki/doc/api/jsapi.php?chapter=9\_1 |
| metadata | String | 否 | {"key": "value", "key2": "value2"} | 商户扩展参数，业务正常返回时会原样传回给商户 |
| deviceId | String | 是 | 1183224081 | 终端唯一编号 |
| deviceType | String | 是 | MOBILE | 终端类型 |
| requestCustomerId | String | 是 |  | 验签时该参数不做任何处理 |
| tradeNo | String | 是 | 1134376595 | 商户系统的支付请求流水号 |
| requestIp | String | 是 | 18.210.2.53 | 请求IP |
| requestIdentification | String | 是 | iPhone 7s plus pro 128T |  |

* 1. 调用示例
     1. JAVA语言

|  |
| --- |
| MobpexConfig.setAppId("151224043667104893");//商户在Mobpex的应用ID  MobpexConfig.setSecretKey("key值");//密钥。请登录官网查询或修改密钥  //如果使用测试模式接入请使用Test 密钥  //如果使用生产LIVE模式，请使用Live密钥 请参见公共参数说明  MobpexConfig.setServerRoot("https://220.181.25.235/yop-center");  MobpexConfig.setIgnoreSSLCheck(true);  MobpexRequest mobpexRequest = new MobpexRequest();  mobpexRequest.setUserId( "test168888@datayp.com");  PrePayRequestDTO prePayRequestDTO = new PrePayRequestDTO();  prePayRequestDTO.setTradeNo(""+(new Random().nextInt(Integer.MAX\_VALUE)));//此处传商户系统支付请求流水，支付请求流水应该与商户订单号区分开来，一般的一笔订单可产生多笔支付流水。  prePayRequestDTO.setPayChannel("WECHAT"); //见请求参数说明  prePayRequestDTO.setPayType("PUB");//见请求参数说明 PUB表示公众号  if(prePayRequestDTO.getPayChannel().eqauls("WECHAT")  && prePayRequestDTO.getPayType().eqauls("PUB ")){  //微信公众号需要增加一个扩展参数 openid 具体openid的获取请参照： https://pay.weixin.qq.com/wiki/doc/api/jsapi.php?chapter=9\_1  prePayRequestDTO. getExtra().put("openid","xxffdxfds1242");//示例  }    prePayRequestDTO.setProductName("维多利亚的秘密");//见请求参数说明  prePayRequestDTO.setProductDescription("G cup");//见请求参数说明  prePayRequestDTO.setAmount("1.00"); //见请求参数说明  prePayRequestDTO.setRequestIp("127.0.0.1");//见请求参数说明  mobpexRequest.addParam(  "prePayRequest",  new Gson().toJson(prePayRequestDTO));  MobpexResponse mobpexResponse = MobpexClient.post(  "/rest/v1.0/pay/unifiedOrder",  mobpexRequest);  if(mobpexResponse.isValidSign()){ //判断验签是否通过  System.out.println( mobpexResponse.getContent());  //mobpexResponse.getContent()可得到响应报文  } |
|  |

* + 1. Python语言

|  |
| --- |
| #密钥值，参见公共参数说明  secretKey = 'TS212fdsfdsfdsfsfsfsdfdsfds12241421g'  appId = '15122404163671048936'  userId = 'test@test.com'  serverRoot = 'https://220.181.25.235/yop-center'  ignoreSSLCheck='true'  liveMode='false'  #传递的参数  prePayRequest ={  "tradeNo":"834153959836",  "payChannel":"ALIPAY",  "payType":"APP",  "productName":"维多利亚的秘密",  "productDescription":"apple",  "amount":"1.00",  "requestIp":"127.0.0.1"  }  try:  prePayRequestJson = json.dumps(prePayRequest)  #获取客户端对象  client = MobpexPythonSDK.base.Client(appId,userId,serverRoot,secretKey,  ignoreSSLCheck,prePayRequest= prePayRequestJson,liveMode=liveMode)  req = MobpexPythonSDK.base.TopRequest('/rest/v1.0/pay/unifiedOrder')  jsonObj = client.getResponse(req)  #验证签名  flag =client.validResult(jsonObj)  if flag:  logging.info('签名验证成功 ')  else:  logging.info('签名验证失败！')  except Exception as e :  print(e) |

* + 1. NodeJs语言

|  |
| --- |
| //客户端对象  var client = new Client({  'appId':'15122404163671048936',  'secretKey':'TS212fdsfdsfdsfsfsfsdfdsfds12241421g',  'serverRoot':'https://220.181.25.235/yop-center',  'userId':'test@test.com'  });  var prePayRequest={  "tradeNo":"83415395983564",  "payChannel":"ALIPAY",  "payType":"APP",  "productName":"33",  "productDescription":"apple",  "amount":"1.00",  "requestIp":"127.0.0.1" };  //请求参数  var params={  "prePayRequest":prePayRequest,  "liveMode":'true'  };  client.request("/rest/v1.0/rest/v1.0/pay/unifiedOrder",  params,function (error,response) {  if(!error){  var content=response.body;  //验证签名  var flag=client.certifyResponse(content);  if(flag){  console.log("签名验证成功！");  console.log(response.body);  }  else{  console.log("签名验证失败！");  console.log(response.body);  }  }  else{  console.log(error);  }  }); |

* + 1. php语言

|  |
| --- |
| include\_once('MobpexSdk.php');  $client = new MobpexClient('https://220.181.25.235/yop-center',  //Mobpex对外提供的地址  '/rest/v1.0/pay/unifiedOrder', //接口对应的uri  false, //是否需要进行证书验证  '15122404366710489367', //appid  'LS\_212fdsfdsfdsfsfsfsdfdsfds12241421'); //密钥值  $prePayRequest = array('productName' => "123456",  'productDescription' => "Gcup",  'payType' => "APP",  'payChannel' => "WECHAT",  'payCurrency' => "CNY",  'amount' => "200.1",  'deviceId' => "1183224081",  'deviceType' => "MOBILE",  'tradeNo' => "1134376515",  'requestIp' => "127.0.0.1",  'requestIdentification' => "iPhone 7s");  $client->set\_param('prePayRequest',json\_encode($prePayRequest, JSON\_UNESCAPED\_UNICODE | JSON\_UNESCAPED\_SLASHES));  $client->set\_param('userId', 'long.chen-1@yeepay.com');  $result = $client->execute(); //对于不需要session的api，则可以不用session参数  if($client->validSign($result)){ //返回结果的验签  var\_dump($result);  } |

* 1. 响应

|  |  |  |
| --- | --- | --- |
| 类型 | 是否必须 | 示例 |
| JSON字符串 | 必须 | {  "state":"SUCCESS",//操作是否成功，如果为SUCCESS则为成功  "result":{ //result的内容为业务的响应结果  "appId":"15122404177104893671",//应用id  "liveMode":false,//是否Live模式  "errCode":"1",//错误码 为1表示成功  "errMsg":"成功",//描述  //魔派订单号  "orderId":"PO21b9gq09AFBWkQVJt4mdh23a6t3r6j80",  "tradeNo":"1098836026",//商户系统支付请求流水号  "paymentParams":{//第三方支付系统支付凭证  "appId":"wx\_appId\_10010010",  "noncestr":"1452836638609",  "package":"Sign=WXPay",  "partnerid":"wx\_mchid\_10010010",  "prepayid":1.452836638609E12,  "sign":"signature\_test\_100100010",  "timestamp":"1452836638"  }  },  "ts":1452836639116,//时间戳  //必要验证签名正确才能认为信息有效  "sign":"2be8437156718320f50dd3407d07585f", //签名  "format":"json",//格式  "validSign":true//SDK对签名进行验证的结果  } |

1. 退款请求
   1. 请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类型 | 是否必须 | 示例 |
| methodOrUri | String | 必须 | /rest/v1.0/pay/refund |
| refundRequest | JSON字符串 | 必须 | {  "tradeNo": "1687906461",  "amount": "3.00",  "description": "我要退款~~~",  "userId": "test168888@datayp.com",  "deviceId": "136178326",  "deviceType": "MOBILE",  "requestCustomerId": "",  "refundNo": "2048522827",  "requestIp": "127.0.0.1",  "requestIdentification": "apple",  "appId": "15122404177104893671",  "liveMode": false  } |

参数名：refundRequest

参数值示例（JSON字符串）：

refundRequest的JSON串的构成：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | 类型 | 是否必须 | 示例 | 描述 |
| tradeNo | String | 必须 | 维多利亚的秘密 | 商户系统支付请求流水 |
| description | String | 必须 | 我要退款~~~ | 描述 |
| amount | String | 必须 | 200.00 | 合计金额，精确到小数点后2位 |
| deviceId | String | 必须 | 1183224081 | 终端唯一编号 |
| deviceType | String | 必须 | MOBILE | 终端类型 |
| requestCustomerId | String | 必须 |  | 验签时该参数不做任何处理 |
| refundNo | String | 必须 | 1134376595 | 商户系统的退款请求流水号 |
| requestIp | String | 必须 | 18.210.2.53 | 请求IP |
| requestIdentification | String | 必须 | apple iPhone 7s plus pro 128T |  |

* 1. 调用示例
     1. JAVA语言

|  |
| --- |
| MobpexConfig.setAppId("15122404366710489367");//Mobpex应用  MobpexConfig.setSecretKey("key值");//密钥 请参见公共参数说明  MobpexConfig.setServerRoot("https://220.181.25.235/yop-center");  MobpexConfig.setIgnoreSSLCheck(true);  MobpexRequest request = new MobpexRequest();  Random r = new Random();  request.setUserId( "test168888@datayp.com");  RefundRequestDTO refundRequest = new RefundRequestDTO();  refundRequest.setRefundNo(""+r.nextInt(Integer.MAX\_VALUE));  //（商户退款流水号）  refundRequest.setDeviceId(""+r.nextInt(Integer.MAX\_VALUE)); //设备标识  refundRequest.setDeviceType("MOBILE"); //设备类型  refundRequest.setRequestIp("127.0.0.1"); //请求者终端IP  refundRequest.setRequestIdentification("apple"); //请求者标识  refundRequest.setTradeNo("428198158791"); //商户支付请求流水号  refundRequest.setAmount("0.01"); //退款金额，单位元保留两位小数  refundRequest.setDescription("我要退款~~~");  request.addParam("refundRequest", new Gson().toJson(refundRequest));  System.out.println(request.getParam("refundRequest"));  MobpexResponse response =MobpexClient.post("/rest/v1.0/pay/refund",request);  if(response.isValidSign()){ //判断验签是否通过  System.out.println(response.getContent());  // response.getContent()可得到响应报文  } |

* + 1. Python语言

|  |
| --- |
| #密钥值，参见公共参数说明  secretKey = 'TS212fdsfdsfdsfsfsfsdfdsfds12241421g'  appId = '15122404163671048936'  userId = 'test@test.com'  serverRoot = 'https://220.181.25.235/yop-center'  ignoreSSLCheck='true'  liveMode='false'  #请求参数  refundRequest={  "tradeNo":"834153959836",  "deviceId":"834153959836",  "deviceType":"MOBILE",  "requestIp":"127.0.0.1",  "requestIdentification":"apple",  "refundNo":"333",  "amount":"0.01",  "description":"我要退款~~~"  }  try:  refundRequestJson = json.dumps(refundRequest)  client = MobpexPythonSDK.base.Client(appId,userId,serverRoot,secretKey,ignoreSSLCheck,  refundRequest= refundRequestJson,liveMode=liveMode)  req = MobpexPythonSDK.base.TopRequest('/rest/v1.0/pay/refund')  jsonObj = client.getResponse(req)  print(jsonObj)  #验证签名  flag =client.validResultSign(jsonObj)  if flag:  logging.info('签名验证成功 ')  else:  logging.info('签名验证失败！')  except Exception as e :  print(e) |

* + 1. NodeJs语言

|  |
| --- |
| //客户端对象  var client = new Client({  'appId':'15122404163671048936',  'secretKey':'TS212fdsfdsfdsfsfsfsdfdsfds12241421g',  'serverRoot':'https://220.181.25.235/yop-center',  'userId':'test@test.com'  });  var refundRequest={  "tradeNo":"834153959836",  "deviceId":"834153959836",  "deviceType":"MOBILE",  "requestIp":"127.0.0.1",  "requestIdentification":"apple",  "refundNo":"33333",  "amount":"0.01",  "description":"我要退款~~~",  };  //请求参数  var refundParams={  "refundRequest":refundRequest,  "liveMode":'true'  };  client.request("/rest/v1.0/pay/refund",  refundParams,function (error,response) {  if(!error){  var content=response.body;  //验证签名  var flag=client.certifyResponse(content);  if(flag){  console.log("签名验证成功！");  console.log(response.body);  }  else{  console.log("签名验证失败！");  console.log(response.body);  }  }  else{  console.log(error);  }  }); |

* + 1. php语言

|  |
| --- |
| include\_once('MobpexSdk.php');  $client = new MobpexClient('https://220.181.25.235/yop-center', //Mobpex对外提供的地址  '/rest/v1.0/pay/refund', //接口对应的uri  false, //是否需要进行证书验证  '15122404366710489367', //appid  'LS\_212fdsfdsfdsfsfsfsdfdsfds12241421'); //密钥值  $prePayRequest = array("tradeNo" => "656113769613",//商户系统的支付请求流水号  "amount" => "3.00",//合计金额，精确到小数点后2位  "description"=> "我要退款~~~",//描述  "deviceId"=> "1183224081",//终端唯一编号  "deviceType"=> "MOBILE",//终端类型  "requestCustomerId"=> "",  "refundNo"=> "20485228279",//商户系统的退款请求流水号  "requestIp"=> "127.0.0.1",//请求IP  "requestIdentification"=> "iPhone 7s");  $client->set\_param('userId', 'long.chen-1@yeepay.com');  $client->set\_param('refundRequest',json\_encode($prePayRequest, JSON\_UNESCAPED\_UNICODE| JSON\_UNESCAPED\_SLASHES));  $result = $client->execute(); // 对于不需要session的api，则可以不用session参数  if($client->validSign($result)){ //返回结果的验签  var\_dump($result);  } |

* 1. 响应

|  |  |  |
| --- | --- | --- |
| 类型 | 是否必须 | 示例 |
| JSON字符串 | 必须 | {  "state": "FAILURE", //操作是否成功，如果为SUCCESS则为成功  "ts": 1452762109989, //时间戳  //必须验证签名正确才能认为信息有效  "sign": "e1556f28a2199c073dffff0c82169d85", //签名  "result":{//result的内容为具体业务的响应结果  "appId":"15122404366710489367",//应用id  "liveMode":true,//是否live模式  "errCode":"0012",//错误码 为1表示成功  "errMsg":"支付订单为1575729200的记录不存在.",  "tradeNo":"1575729200",//商户系统交易流水号  "refundNo":"192154112619"//商户系统退款流水号  }  "error": {//错误信息，state不为SUCCESS才会有此错误信息  "code": "99999000", //错误码  "message": "应用(15122404177104893671)系统异常，请稍后重试",  "subErrors": [ ]  },  "format": "json", //格式  "validSign": true//SDK对签名进行验证的结果  } |

1. 支付查询请求
   1. 请求参数

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 名称 | 类型 | 是否必须 | 示例 | 描述 | 描述 |
| methodOrUri | 字符串 | 必须 | /rest/v1.0/pay/queryPaymentOrder | 本请求的接口URI地址，是API的唯一标识。 |
| tradeNo | 字符串 | 必须 | 1134376595 | 商户支付请求流水号。 |

* 1. 调用示例
     1. JAVA语言

|  |
| --- |
| MobpexConfig.setAppId("15122404366710489367");//应用ID  MobpexConfig.setSecretKey("key值");  //应用密钥  MobpexConfig.setServerRoot("https://220.181.25.235/yop-center");  MobpexConfig.setIgnoreSSLCheck(true);  MobpexRequest request = new MobpexRequest();  request.setUserId( "test168888@datayp.com");  request.addParam("tradeNo", "669251114954");  MobpexResponse response =  MobpexClient.post("/rest/v1.0/pay/queryPaymentOrder", request);  if(response.isValidSign()){ //判断验签是否通过  System.out.println(response.getContent());  // response.getContent()可得到响应报文  } |
|  |

* + 1. Python语言

|  |
| --- |
| #密钥值，参见公共参数说明  secretKey = 'TS212fdsfdsfdsfsfsfsdfdsfds12241421g'  appId = '15122404163671048936'  userId = 'test@test.com'  serverRoot = 'https://220.181.25.235/yop-center'  ignoreSSLCheck='true'  liveMode='false'  tradeNo ="669251114954"  try:  client = MobpexPythonSDK.base.Client(appId,userId,serverRoot,secretKey,  ignoreSSLCheck, tradeNo = tradeNo,liveMode=liveMode)  #请求  req = MobpexPythonSDK.base.TopRequest('/rest/v1.0/pay/queryPaymentOrder')  jsonObj = client.getResponse(req)  #验证签名  flag =client.validResult(jsonObj)  if flag:  logging.info('签名验证成功 ')  else:  logging.info('签名验证失败！')  except Exception as e :  print(e) |

* + 1. NodeJs语言

|  |
| --- |
| //客户端对象  var client = new Client({  'appId':'15122404163671048936',  'secretKey':'TS212fdsfdsfdsfsfsfsdfdsfds12241421g',  'serverRoot':'https://220.181.25.235/yop-center',  'userId':'test@test.com'  });  //请求参数  var params={  tradeNo:"8341539598356455",  liveMode:'true'  };  //发起请求  client.request("/rest/v1.0/pay/queryPaymentOrder",  params,function (error,response) {  if(!error){  var content=response.body;  //验证签名  var flag=client.certifyResponse(content);  if(flag){  console.log("签名验证成功！");  console.log(response.body);  }  else{  console.log("签名验证失败！");  console.log(response.body);  }  }  else{  console.log(error);  }  }); |

* + 1. php语言

|  |
| --- |
| nclude\_once('MobpexSdk.php');  $client = new MobpexClient('https://220.181.25.235/yop-center',  //Mobpex对外提供的地址  '/rest/v1.0/pay/queryPaymentOrder', //接口对应的uri  false, //是否需要进行证书验证  '15122404366710489367', //appid  'LS\_212fdsfdsfdsfsfsfsdfdsfds12241421'); //密钥值  $client->set\_param('tradeNo','1134376519');  $client->set\_param('userId', 'long.chen-1@yeepay.com');  $result = $client->execute(); // 对于不需要session的api，则可以不用session参数  if($client->validSign($result)){ //返回结果的验签  var\_dump($result);  } |

* 1. 响应

|  |  |  |
| --- | --- | --- |
| 类型 | 是否必须 | 示例 |
| JSON字符串 | 必须 | {  "state" : "SUCCESS",//操作是否成功，如果为SUCCESS则为成功  "ts": 1452762109989, //时间戳  //必须验证签名正确才能认为信息有效  "sign": "e1556f28a2199c073dffff0c82169d85", //签名  "result" : {//result的内容为具体业务的响应结果  "appId" : "15122404177104893671",//应用id  "liveMode" : true, //是否live模式  "errCode" : "1",//错误码 为1表示成功  "errMsg" : "成功",  "tradeNo" : "1687906461",//商户系统交易流水号  "productName" : "维多利亚的秘密",//商品名称  "productDescription" : "G cup",//商品描述  "payType" : "APP",// APP、WEB、WAP、PUBLIC几种之一  "payChannel" : "ALIPAY",//渠道编号  "payCurrency" : "CNY",//币种  "paidAmount" : 0.00, //金额  "payStatus" : "INIT",//状态  "createdDate" : "2016-01-05 13:18:02"  },  "ts" : 1452839342567  } |

1. 退款查询请求
   1. 请求参数

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 名称 | 类型 | 是否必须 | 示例 | 描述 | 描述 |
| methodOrUri | 字符串 | 必须 | /rest/v1.0/pay/queryRefundOrder | 本请求的接口URI地址，是API的唯一标识。 |
| tradeNo | 字符串 | 必须 | 1134376595 | 商户订单编号，和发起支付请求中的requestFlowId一致。 |
| refundNo | 字符串 | 必须 | 15122404177104893671 | 商户退款流水 |

* 1. 调用示例
     1. JAVA语言

|  |
| --- |
| MobpexConfig.setAppId("15122404366710489367");//应用  MobpexConfig.setSecretKey("key值");  //应用密钥，参见公共参数说明  MobpexConfig.setServerRoot("https://220.181.25.235/yop-center");  MobpexConfig.setIgnoreSSLCheck(true);  MobpexRequest request = new MobpexRequest();  request.setUserId( "test168888@datayp.com");  request.addParam("tradeNo", "669251114954");  request.addParam("refundNo", "201601261053502690728");  MobpexResponse response = MobpexClient.post("/rest/v1.0/pay/queryRefundOrder", request);  if(response.isValidSign()){ //判断验签是否通过  System.out.println(response.getContent());  // response.getContent()可得到响应报文  } |
|  |
|  |

* + 1. Python语言

|  |
| --- |
| #密钥值，参见公共参数说明  secretKey = 'TS212fdsfdsfdsfsfsfsdfdsfds12241421g'  appId = '15122404163671048936'  userId = 'test@test.com'  serverRoot = 'https://220.181.25.235/yop-center'  ignoreSSLCheck='true'  liveMode='false'  tradeNo ="669251114954",  refundNo ="201601261053502690728"  try:  client = MobpexPythonSDK.base.Client(appId,userId,serverRoot,  secretKey,ignoreSSLCheck, tradeNo = tradeNo,  refundNo = refundNo,liveMode= liveMode )  req = MobpexPythonSDK.base.TopRequest('/rest/v1.0/pay/queryRefundOrder')  jsonObj = client.getResponse(req)  #验证签名  flag =client.validResult(jsonObj)  if flag:  logging.info('签名验证成功 ')  else:  logging.info('签名验证失败！')  except Exception as e :  print(e) |

* + 1. NodeJs语言

|  |
| --- |
| //客户端对象  var client = new Client({  'appId':'15122404163671048936',  'secretKey':'TS212fdsfdsfdsfsfsfsdfdsfds12241421g',  'serverRoot':'https://220.181.25.235/yop-center',  'userId':'test@test.com'  });  //请求参数  var queryFundParamas={  " tradeNo ":"669251114954",  " refundNo ":"201601261053502690728",  "liveMode":'true'  };  //发起请求  client.request("/rest/v1.0/pay/queryRefundOrder",  queryFundParamas,function (error,response) {  if(!error){  var content=response.body;  //验证签名  var flag=client.certifyResponse(content);  if(flag){  console.log("签名验证成功！");  console.log(response.body);  }  else{  console.log("签名验证失败！");  console.log(response.body);  }  }  else{  console.log(error);  }  }); |

* + 1. php语言

|  |
| --- |
| include\_once('MobpexSdk.php');  $client = new MobpexClient('https://220.181.25.235/yop-center',  //Mobpex对外提供的地址  '/rest/v1.0/pay/queryRefundOrder', //接口对应的uri  false, //是否需要进行证书验证  '15122404366710489367', //appid  'LS\_212fdsfdsfdsfsfsfsdfdsfds12241421'); //密钥值  $client->set\_param('tradeNo','656113769613');  $client->set\_param('refundNo','20485228279');  $client->set\_param('userId', 'long.chen-1@yeepay.com');  $result = $client->execute(); // 对于不需要session的api，则可以不用session参数  if($client->validSign($result)){ //返回结果的验签  var\_dump($result);  } |

* 1. 响应

|  |  |  |
| --- | --- | --- |
| 类型 | 是否必须 | 示例 |
| JSON字符串 | 必须 | {  "state": "SUCCESS",//操作是否成功，如果为SUCCESS则为成功  "result": {//result的内容为具体业务的响应结果  "appId": "15122404366710489367",//应用id  "liveMode": true, //是否live模式  "errCode": "1",//错误码 为1表示成功  "errMsg": "成功",  "tradeNo": "669251114954",  "refundNo": "12455111",  "prepayId": "201601261053502690728",  "productName": "维多利亚的秘密",//商品名称  "productDescription": "G cup",//商品描述  "payType": "APP",// APP、WEB、WAP、PUBLIC几种  "payChannel": "UPACP",//渠道编号  "payCurrency": "CNY",//币种  "paidAmount": 0.01,//金额  "payStatus": "REFUND",//状态  "createdDate": "2016-01-26 10:45:47",//支付请求创建时间  "paidDatetime": "1970-01-26 10:53:50"//支付完成时间  },  "ts": 1453857377561,  "format": "json",  "validSign": true  } |