aipay支付接口规范

# 概述

术语说明如下：

**M: 必需参数**

**C: 有条件的才存在的参数**

**O: 可选参数**

**M(X): 其中的X为最大长度**

**如没有特别说明，所有字段的类型都为String格式，JSON格式作为参数整体的请求和应答。**

## 技术接入流程

# 数据安全交互

* 客户端与服务器端的数据交互为请求/应答的方式
* 采用标准的HTTP/HTTPS通讯协议
* 客户端请求报文放在HTTP\_BODY中使用POST方法发送到服务器端
* 请求数据类型使用application/json，字符编码格式为utf-8
* 数据层采用MD5进行签名和验签（在**2.1和2.2章节**详细描述）
* 生产环境上下文地址: http://\*\*\*\*\*\*\*\*\*\*

## 请求创建订单签名

在创建订单的时候需要生成签名，把指定的参数和参数值，按照指定的顺序，拼接成请求串，然后进行MD5加密，然后转成大写，就是最终的签名串，以下是java代码参考：

// 生成签名并加入请求参数中

StringBuilder signSb = **new** StringBuilder();

signSb.append("merchantCode=").append("MTEST");

signSb.append("&notifyUrl=").append("http://192.168.0.2/callback");

signSb.append("&payChannel=").append("ALIPAY\_CODE");

signSb.append("&tradeNo=").append("O34343MLKLJ93434");

signSb.append("&totalAmount=").append("102.00");

signSb.append("&appSecret=").append("B0403D088F520FB2666E2E3545E4C9FB");

String sign = DigestUtils.*md5Hex*(signSb.toString()).toUpperCase());

备注：DigestUtils是apache common codec包下的一个工具类,appSecret取值是商户自己的密钥

## 平台回调验证签名

支付成功后，平台需要回调商户，这个时候需要验证签名，把指定的参数和参数值，按照指定的顺序，拼接成请求串，然后进行MD5加密，然后转成大写，就是最终的签名串，以下是java代码参考：

**// 验证签名**

**StringBuilder signSb = new StringBuilder();**

**signSb.append("code=").append(**"APO\_9384394DFJR1234"**);**

**signSb.append("&merchantCode=").append(**"MTEST"**);**

**signSb.append("&payAmount=").append(**"102.00"**);**

**signSb.append("&payChannel=").append(**"ALIPAY\_CODE"**);**

**signSb.append("&totalAmount=").append(**"102.00"**);**

**signSb.append("&appSecret=").append(**"B0403D088F520FB2666E2E3545E4C9FB"**);**

**String newSign = DigestUtils.md5Hex(signSb.toString()).toUpperCase();**

备注：DigestUtils是apache common codec包下的一个工具类,appSecret是商户自己的密钥,code是平台生成的订单唯一标识

## 查询订单是否支付签名

在查询订单是否已经支付的时候需要生成签名，把指定的参数和参数值，按照指定的顺序，拼接成请求串，然后进行MD5加密，然后转成大写，就是最终的签名串，以下是java代码参考：

// 生成查询订单签名

StringBuilder querySignSb = **new** StringBuilder();

querySignSb.append("code=").append("APO\_9384394DFJR1234");

querySignSb.append("&merchantCode=").append("MTEST");

querySignSb.append("&appSecret=").append(merchantAppSecret);

String sign = DigestUtils.*md5Hex*(querySignSb.toString()).toUpperCase());

备注：DigestUtils是apache common codec包下的一个工具类,code是平台生成的订单唯一标识

# 请求头参数说明

|  |  |  |
| --- | --- | --- |
| HTTP请求头参数定义 | | |
| 参数名 | 出现要求 | 描述 |
| Content-Type | M(128) | application/json |
| Accept-Charset | O(32) | 默认为UTF-8 |
| Accept-Language | O(16) | 默认为zh-cn，目前只会返回中文的字符提示 |

# 响应头参数说明

|  |  |  |
| --- | --- | --- |
| HTTP响应头参数定义 | | |
| 参数名 | 出现要求 | 描述 |
| Content-Type | M(128) | text/plain; charset=UTF-8; |

|  |  |  |
| --- | --- | --- |
| HTTP响应报文体为JSON文本。  每个应答报文都包含以下公共参数： | | |
| 参数名 | 出现要求 | 描述 |
| status | M(12) | 标识接口是否请求成功 |
| msg | O(256) | 返回的提示或者错误信息 |
| data | O(MAX) | 返回的具体业务数据 |

# 接口说明

## 功能描述

## 扫码支付

接口地址: 上下文+“ /order/create”

### 请求

|  |  |  |
| --- | --- | --- |
| 参数名 | 出现要求 | 描述 |
| version | O(8) | 接口版本号，可以为空，默认值目前固定为1.0.1 |
| charset | O(8) | 字符集，可以为空，默认值目前固定为“UTF-8” |
| signType | O(8) | 签名类型，取值范围参考底部的定义，目前只支持MD5签名算法 |
| sign | M(MAX) | 签名串 |
| model | M(14) | 模式：标识这条订单是测试数据还是线上生产数据，取值范围参考文档底部的定义 |
| clientIp | M(20) | 客户端端ip标识 |
| merchantCode | M(32) | 平台分配的商户号 |
| tradeNo | M(64) | 商户自己的订单唯一标识 |
| payChannel | M(32) | 支付渠道，取值范围参考底部的定义 |
| codeType | M(32) | 支付宝和微信的收款码生成类型,取值范围参考底部的定义 |
| totalAmount | M(15) | 总的交易金额，单位为元，保留两位小数 |
| subject | M(128) | 商品标题 |
| body | M(256) | 商品描述 |
| extraData | O(256) | 扩展字段，目前作为保留字段使用 |
| passbackData | O(256) | 回传字段，异步通知回调时会回传给商户 |
| notifyUrl | M(256) | 异步通知的回调地址 |
| requestDateTime | M(64) | 请求接口的客户端时间，格式为yyyy-MM-dd HH:mm:ss |
| remark | O(256) | 订单备注信息 |

### 应答

|  |  |  |
| --- | --- | --- |
| 参数名 | 出现要求 | 描述 |
| 以下为受理结果，在status为1时有效 | | |
| merchantCode | M(32) | 商户号 |
| tradeNo | M(64) | 商户自己的订单唯一标识 |
| code | Ｍ(64) | 平台生成的订单唯一标识 |
| payChannel | M(32) | 支付渠道 |
| totalAmount | M(15) | 总的交易金额，单位为元，保留两位小数 |
| payAmount | M(15) | 实际的支付金额，单位为元，保留两位小数 |
| codeType | M(32) | 支付宝和微信的收款码生成类型,取值范围参考底部的定义 |
| codeContent | M(256) | 生成的二维码内容,可能是base64Image格式的字符串,也可能是图片的连接地址，也可能为空,取值范围参考底部的定义 |
| createDateTime | M(32) | 订单创建时间 |
| expireDateTime | M(32) | 订单过期时间 |
| model | M(32) | 标识是测试环境还是生产环境 |

## 审核结果异步通知

支付成功后，平台会进行回调操作，平台会向商户的回调地址发送post请求，把请求数据放到body体中，商户需要获取到这些请求数据，进行相应的验签操作，然后再执行具体的支付逻辑，成功后返回”SUCCESS”的字符串，区分大小写，非”SUCCESS”的返回值平台认为回调失败，会进行相应的重试操作.

|  |  |  |
| --- | --- | --- |
| 参数名 | 出现要求 | 描述 |
| merchantCode | M(32) | 商户号 |
| tradeNo | M(64) | 商户自己的订单唯一标识 |
| code | Ｍ(64) | 平台生成的订单唯一标识 |
| payChannel | M(32) | 支付渠道 |
| totalAmount | M(15) | 总的交易金额，单位为元，保留两位小数 |
| payAmount | M(15) | 实际的支付金额，单位为元，保留两位小数 |
| payDateTime | M(32) | 订单支付成功时间 |
| signType | O(8) | 签名类型，取值范围参考底部的定义，目前只支持MD5签名算法 |
| sign | M(MAX) | 签名串 |
| model | M(32) | 标识是测试环境还是生产环境 |

以下是java参考代码：

第一步:获取回调数据

/\*\*

\* **@param** request

\* **@return**

\* **@throws** IOException

\*/

**public** String parseInputStream(HttpServletRequest request) **throws** IOException {

request.setCharacterEncoding("UTF-8");

BufferedReader br = **new** BufferedReader(

**new** InputStreamReader((ServletInputStream) request.getInputStream(), "UTF-8"));

String line = **null**;

StringBuilder sb = **new** StringBuilder();

**while** ((line = br.readLine()) != **null**) {

sb.append(line);

}

String result = sb.toString();

//返回获取到的回调数据

return result;

}

第二步:进行验签操作

JSONObject jsonObject = JSONObject.*parseObject*(result);

String merchantCode = jsonObject.getString("merchantCode");

String code = jsonObject.getString("code");

String tradeNo = jsonObject.getString("tradeNo");

BigDecimal totalAmount = jsonObject.getBigDecimal("totalAmount");

BigDecimal payAmount = jsonObject.getBigDecimal("payAmount");

String payChannel = jsonObject.getString("payChannel");

Date payDate = jsonObject.getDate("payDateTime");

String sign = jsonObject.getString("sign");

// 验证签名

参考2.2

第三步:进行具体的回调逻辑操作

//签名验证成功后，进行具体的回调逻辑处理

第四步:返回SUCCESS字符串

//回调逻辑处理完毕后,返回”SUCCESS”字符串

# 常量定义

## 支付渠道payChannel

|  |  |
| --- | --- |
| 代码 | 描述 |
| ALIPAY\_CODE | 支付宝收款码支付 |
| WECHAT\_CODE | 微信收款码支付 |

## 签名类型signType

|  |  |
| --- | --- |
| 代码 | 描述 |
| NONE | 没有采用任何签名算法 |
| MD5 | MD5签名算法 |
| SHA256 | SHA256签名算法 |

## 模式model

|  |  |
| --- | --- |
| 代码 | 描述 |
| TEST | 测试环境数据 |
| PRODUCT | 生产环境数据 |

## 响应码responseCode

|  |  |
| --- | --- |
| 代码 | 描述 |
| 0 | 系统繁忙，请稍后再试 |
| 1 | **操作成功** |
| 2 | 参数验证出错 |
| 3 | 参数映射错误 |
| 4 | 认证出错 |
| 5 | 找不到请求的接口 |
| 6 | 不支持的请求方法 |
| 7 | 不正确的context-type请求头标识 |
| 11 | 签名错误 |
| 100 | 其它错误 |
| 2001 | 商户不存在 |
| 2003 | 没有绑定支付账号 |
| 2004 | 不正确的支付渠道 |
| 2006 | 商户被禁用 |
| 2008 | 商户余额不足 |

## 收款码类型 codeType

|  |  |
| --- | --- |
| 代码 | 描述 |
| BASE64\_IMAGE | base64格式的字符串,不建议在手机h5页面中使用该类型 |
| IMAGE\_URL | 图片的链接地址,商户可以直接在h5页面上引用该地址来显示图片，目前图片的流量是免费的 |
| NONE | 平台不提供收款码图片，由商户自己提供，产生的安全问题由商户自己控制 |