# 支付宝商家账户申请

* 申请网址 ： b.alipay.com
* 注册流程：

1. 申请商家版支付宝

2. 进行实名认证，等待审核（**3个工作日**）

3. 成为开发者

4. 签约支付功能（详见[支付宝支付能力](#_支付宝支付能力)）中的一种，可以多种

5. 审核支付功能 （**3-5个工作日**）

6. 开发者配置、Admin配置

# 支付宝概念

**支付宝应用**： 所有支付对外都功能都称之应用，应用又分为第三方应用和自研型应用。

**第三方应用**： 主要服务对象是ISV（独立软件服务商）。

**自研型应用**： 主要给商户自己用。自研型应用又分为小程序、网页&移动应用、生活号



**应用审核** ： 创建应用会收集名称和图标，需要等待审核。

**商户**： 所有支付宝的对公客户都被称为商户。

**独立软件服务商** ：即ISV，Independent Software Vendors , 特指专门从事软件的开发、生产、销售和服务的企业。 这个扮演的角色主要是帮助那些不具备软件开发的公司（比如很多企事业单位）去完成软件开发。所以支付宝选第三方应用的时候，会选择第三方代理公司，这里的代理就是ISV。

# 支付宝支付能力

目前官方https://docs.open.alipay.com/ 文档中支付能力包括：

1. 当面付

2. APP支付

3. 手机网站支付

4. 电脑网站支付

5. 花呗分期

6. 刷脸付

7. 收单资金结算到银行账户的到账消息

# 支付宝第三方开发流程

1. 需要申请商户号，详情见[支付宝商家账户申请](#_支付宝商家账户申请)，至少需要等10来天。
2. 创建应用，关于概念，详情见 [支付宝概念](#_支付宝概念)
3. 生成密钥

# 支付宝密钥生成

* **密钥生成工具地址**： <https://docs.open.alipay.com/291/105971>
* 下载文件的名字： RSA签名验签工具windows\_V1.4

注意 : 1. 双击脚本文件 “RSA签名验签工具.bat” 即运行RSA签名验签工具。

2. RSA2签名方式使用算法：SHA256withRSA

RSA签名方式使用算法：SHA1withRSA

3. 签名验签工具不要放在文件路径名存在空格的文件夹下。

* 密码生成后：

开发者就可以在应用的开发配置页面进行配置。点击“设置应用公钥”后，复制上一步生成的公钥，点击“保存”，即可完成公钥的设置。

* 签名验签，私钥的加签和公钥的验签都用下面这个包

**groupid : com.alipay.sdk**

**artifactid : alipay-sdk-java**

**Lastest Version : 3.7.26.ALL**

# Java证书管理工具keytool

**说明：和支付宝对接用的是SSLServerSocket, 我们开启服务端Socket，等待支付宝来调用。**

**Keytool生成密钥**

* 生成服务端

**keytool -genkey -alias serverkey -keyalg RSA -validity 365 -keystore d:\key\kserver.keystore**

输入密钥库口令及地区信息等（密钥库口令123456）

* 输入密钥口令（1234567）
* 建议用PKC12标准

keytool -importkeystore -srckeystore d:\key\kserver.keystore -destkeystore d:\key\kserver.keystore -deststoretype pkcs12

（此时报错：keytool 错误: java.lang.Exception: 目标 pkcs12 密钥库具有不同的 storepass 和 keypass。请在指定了 -destkeypass 时重试。）

更改命令如下：

keytool -importkeystore -srckeystore d:\key\kserver.keystore -destkeypass 1234567 -destkeystore d:\key\kserver.keystore -deststoretype pkcs12

所以尽量库密码和私钥密码一致。

* 根据私钥导出服务端证书

keytool -export -alias serverkey -keystore d:\key\kserver.keystore -file d:\key\server.crt

* 将服务证书导入到客户端Trust keystore中

keytool -import -alias serverkey -file d:\key\server.crt -keystore d:\key\tclient.keystore

* 生成客户端

keytool -genkey -alias clientkey -keyalg RSA -validity 365 -keystore d:\key\kclient.keystore

密钥库和密钥的密码都设置成abcdef

执行

keytool -importkeystore -srckeystore d:\key\kclient.keystore -destkeystore d:\key\kclient.keystore -deststoretype pkcs12

就不会再让加-destkeypass 时重试

* 根据私钥导出客户端证书

keytool -export -alias clientkey -keystore d:\key\kclient.keystore -file d:\key\client.crt

* 将客户端证书导入到服务端Trust keystore中

keytool -import -alias clientkey -file d:\key\client.crt -keystore d:\key\tserver.keystore

# 数据安全的额外补充

## 对称加密

**对称加密是一种很快速很简单的加密方式，加密和解密用相同的密钥，密钥越短，解密耗时越短，密钥越长，解密时间就越长。**所以使用对称加密需要注意两个方面：

一是如何平衡耗时和安全性。二是加解密双方需要如何传送和保管密钥，一旦密钥泄露，可是天大的事情。

## 非对称加解密（公钥和私钥对）

**非对称加密为数据的加密与解密提供了一个非常安全的方法，它使用了一对密钥，公钥（public key）和私钥（private key）。私钥只能由一方安全保管， 不能外泄，而公钥则可以发给任何请求它的人。非对称加密使用这对密钥中的一个进行加密，而解密则需要另一个密钥**。

RSA数字签名算法，历史二十年，经历无数攻击依然屹立不倒。

公钥 ： 可以随便在网络上传输，也可以告知别人，大家都可以看得到。

私钥 ： 自己的非常重要非常私密的。

着重注意 ： 这个秘钥对中，如果使用任意一把来加密数据，那么只有使用另外一把才能解密数据。

而用法有两种：

**第一种用法：公钥加密，私钥解密。---用于加解密**

**第二种用法：私钥签名，公钥验签。---用于签名**

## 签章，也叫签名，也叫数字签名

**签章** ： 其实就是用私钥加密，对应的公钥解密的过程。为什么会出现签章这个过程？**第一种情况：**A要给B发密文，那么A肯定用B的公钥加密，然后B用自己私钥解密

就可以。但是C也可以拿B的公钥加密，冒充A发给B，B用自己私钥解密，认为是A发的，所以就出现风险问题。

**第二种情况：** A要给B发送明文，随便哪个人都可以冒充A给B发信息。所以，这两种情况B都无法认证A是否是真的A，**那么签章就是避免冒充的**。

**数字签名的过程：**

A 要给 B 发送一段明文，第一，如何避免有人冒充A？ 第二，如何避免A发送的内容没有被篡改过？

**第一步** ：明文 ---- 经过HASH算法 ---- 生成很短的消息摘要，为什么这么做？

1. hash算法不可逆。

2.消息内容不重复，哪怕明文改一个字，重新hash后生成的摘要都和之前的不一样，目的就是防止明文内容被篡改。

**第二步** ：消息摘要 ---- A的私钥进行加密 ---- 得到A的数字签名（Signature），相当于古代A按的指拇印。

**第三步** ：把数字签名附加到明文信息中，传给B。B拿到这份信件后，同样需要做两件事，第一核实是否是A发过来的，第二，核实A发过来的内容没有被篡改。

**第四步** ： B用A的公钥对A的数字签名进行解密，解密成功，便确认是A发来的，不是C或者D冒充。接着，B 把明文再进行hash运算，和解密数字签名得到的消息摘要对比，如果一致，说明信件没有被篡改。

## 数字证书

我们再来试想一个场景，A本来想把自己的公钥给B，用作他们的加解密，但是B没有拿到A的公钥前，有个别有用心的C冒充A，把C的公钥发给了B。此时B认为拿到了A的公钥，于是，他就用误认为A的公钥（实际是C的公钥）给A写信：我的银行密码是888888。C拦截到B的信件后，用自己的私钥就解密打开了。

那么，如何避免公钥冒用呢？前面说到用签名的方式，可以验证发信人的真实身份。这里要避免公钥冒用，就产生了数字证书。

**数字证书的特点：**

i. 由专门的机构签发的数字证书才安全有效。

ii、签发数字证书是收费的。

iii、不会被冒充，安全可信。

iv、数字证书有使用期限，过了使用期限，证书变为不可用。CA也可以在使用期内，对证书进行作废操作。

v、CA的公钥已经集成到操作系统中了。

**生成数字证书的流程如下：**

i、持有人将公钥以及身份信息发送给权威机构。

ii、权威机构负责对持有人的身份进行验证，确保公钥和持有人的信息准确无误。

iii、权威机构使用自己私钥对持有人公钥进行数字签名，生成数字证书。

iv、为了确保证书不被篡改，权威机构对数字证书进行hash计算（指纹算法），生成摘要（指纹），使用自己的私钥对摘要进行数字签名，放到数字证书中。

v、对持有人收费。

这样，就可以避免公钥被冒用。

综上，私钥可以通过数字签名来保证自己的唯一和真实，公钥可以用权威机构颁发的数字证书来保证不会被冒用，就大大的增加了数据在网络上传送的安全性。

# 账号信息

* 夹江支付宝账号：1707957368@qq.com

密码 ：qysc123456

支付密码：123456qysc

密码保护问题：水厂成立时间

密码保护问题的答案：2017年11月23日

* 机构码和清算单位和sftp

中电建夹江青衣水厂有限公司 WTZDJJJQYSCYXGS\_50211 机构码和清算单位

正式sftp配置如下:

1）源地址,机构访问IP和端口：112.124.212.165:3022

用户名和密码：WTZDJJJQYSCYXGS\_sftp/u5nMh\*WfkneO

默认路径：/download

正式sftp配置如下:

1）源地址,机构访问IP和端口：112.124.212.165:3022

用户名和密码：WTWYXZXXCSWYXGS\_sftp/yVkaJvtMOj3w

默认路径：/download

# --------------------分割线------------------------

# 以下是支付宝水费对接技术文献

公用事业支付宝合作

技术方案V2.2.3（缴费）

二〇一八年四月

更改履历

|  |  |  |  |
| --- | --- | --- | --- |
| 版本号 | 更改简要描述 | 更改人 | 批准人 |
| V1.0 | 初稿 | 李建 | 练建冬 |
| V1.4 | 初稿 | 李建 | 练建冬 |
| V1.5 | 初稿 | 李建 | 练建冬 |
| V1.6 | 初稿 | 喻娴 | 练建冬 |
| V1.66 | 修改月账单格式，以支持一户多表 | 许明志 | 练建冬 |
| V2.0 | 增加文档的可读性 | 许明志 | 练建冬 |
| V2.1 | 增加文档的可读性 | 许明志 | 练建冬 |
| V2.2.3 | 增加文档的可读性 | 许明志 | 练建冬 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

目 录

[1 数据通信安全 1](#_Toc28269)

[1.1 技术说明 1](#_Toc3401)

[1.2 实现案例（基于java） 1](#_Toc161)

[1.2.1 创建服务端监听 1](#_Toc21637)

[1.2.2 客户端与服务端建立连接 2](#_Toc13825)

[2 技术方案 3](#_Toc1345)

[2.1 通讯方式 3](#_Toc23251)

[2.1.1 在线交易 3](#_Toc29914)

[2.1.2 文件传输 3](#_Toc26754)

[3 接口规约说明 4](#_Toc21123)

[3.1 报文通用约定 4](#_Toc26372)

[3.1.1 字符集约定 4](#_Toc3494)

[3.1.2 参数名约定 4](#_Toc6717)

[3.1.3 基本参数约定 4](#_Toc11315)

[3.2 报文格式说明 4](#_Toc25488)

[3.2.1 报文帧格式概述 4](#_Toc30823)

[3.2.2 报文数据格式 5](#_Toc16652)

[3.2.3 报文头格式 5](#_Toc27090)

[4 名词解释 5](#_Toc24267)

[5 实时缴费业务 7](#_Toc2994)

[5.1 缴费代收 7](#_Toc5245)

[5.1.1 代收场景描述 7](#_Toc29387)

[5.1.2 ①欠费查询接口规范 8](#_Toc14402)

[5.1.3 ②代收缴费接口规范 9](#_Toc10092)

[5.2 补销账 10](#_Toc14640)

[5.2.1 补销账场景描述 10](#_Toc19528)

[5.2.2 ④销账结果查询接口规范 11](#_Toc18243)

[5.3 代收对账 12](#_Toc20797)

[5.3.1 代收对账场景描述 12](#_Toc15233)

[5.3.2 对账规则 12](#_Toc14666)

[5.3.3 文件通知接口规范 13](#_Toc19103)

[5.3.4 对账明细文件 13](#_Toc30018)

[6 附录 14](#_Toc7527)

[6.1 部分响应码说明 14](#_Toc8441)

[6.2 完整响应码参照表（详情见附件） 16](#_Toc1454)

# 数据通信安全

## 技术说明

接入机构和公共服务器平台之间采用SSL Socket通信技术。SSL Socket通信是对socket通信的扩展，在socket通信的基础上添加了一层安全性保护，提供了更高的安全性，包括身份验证、数据加密以及完整性验证。身份验证用户数字证书的发放和应用；数据加密可以防止消息传递过程中被别人监听而造成的损失，即使第三方监听到传递的消息，但是由于没有正确的密钥，其仍然无法得到正确的消息；完整性验证以防止消息在传递过程中被别人篡改。

SSL支持单向和双向认证。单向认证只认证服务端的合法性而不验证客户端的合法性；双向认证需要同时认证服务端的合法性和客户端的合法性。在我们这个应用中，需要采用双向认证。

采用SSL通信技术后，收到数据后和发送数据前的都不需要进行特殊的处理，与普通的socket通信技术没有差别。差别部分是在建立socket连接或者监听socket连接的过程。要使用SSL双向认证通信技术，必须客户端和服务端都需要有证书。

## 实现案例（基于java）

### 创建服务端监听

private static ServerSocket createSSLServerSocket(){

try {

int port = 1224; //要监听的端口

String serverPrivateKey = "D:\\kserver.keystore"; //服务端自身私钥

String serverKeyPassword = "123456"; //私钥密码

String trustKey= "D:\\tserver.keystore"; //信任的证书列表,即客户端证书

String trusttKeyPassword= "123456"; //信任的证书列表访问密码

SSLContextctx = SSLContext.getInstance("SSL");

KeyManagerFactorykmf = KeyManagerFactory.getInstance("SunX509");

TrustManagerFactorytmf = TrustManagerFactory.getInstance("SunX509");

KeyStoreks = KeyStore.getInstance("JKS");

KeyStoretks = KeyStore.getInstance("JKS");

ks.load(newFileInputStream(serverPrivateKey), serverKeyPassword.toCharArray());

tks.load(newFileInputStream(trustKey), trusttKeyPassword.toCharArray());

kmf.init(ks, serverKeyPassword.toCharArray());

tmf.init(tks);

ctx.init(kmf.getKeyManagers(), tmf.getTrustManagers(), null);

SSLServerSocketserverSocket = (SSLServerSocket) ctx.getServerSocketFactory().createServerSocket(port);

serverSocket.setNeedClientAuth(true);

returnserverSocket;

} catch (Exception e) {

e.printStackTrace();

}

return null;

}

### 客户端与服务端建立连接

private static Socket createSocket(){

try {

String serverHost = "localhost"; //服务端地址

intserverPort = 1224; //服务端监听端口

String clientPrivateKey = "D:\\kclient.keystore"; //客户端私钥

String clientKeyPassword = "123456"; //客户端私钥密码

String trustKey = "D:\\tclient.keystore"; //客户端信任证书列表，即服务端证书

String trustKeyPassword = "123456"; //客户端信任证书密码

SSLContextctx = SSLContext.getInstance("SSL");

KeyManagerFactorykmf = KeyManagerFactory.getInstance("SunX509");

TrustManagerFactorytmf = TrustManagerFactory.getInstance("SunX509");

KeyStoreks = KeyStore.getInstance("JKS");

KeyStoretks = KeyStore.getInstance("JKS");

ks.load(newFileInputStream(clientPrivateKey), clientKeyPassword.toCharArray());

tks.load(newFileInputStream(trustKey), trustKeyPassword.toCharArray());

kmf.init(ks, clientKeyPassword.toCharArray());

tmf.init(tks);

ctx.init(kmf.getKeyManagers(), tmf.getTrustManagers(), null);

return (SSLSocket) ctx.getSocketFactory().createSocket(serverHost, serverPort);

} catch (Exception e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

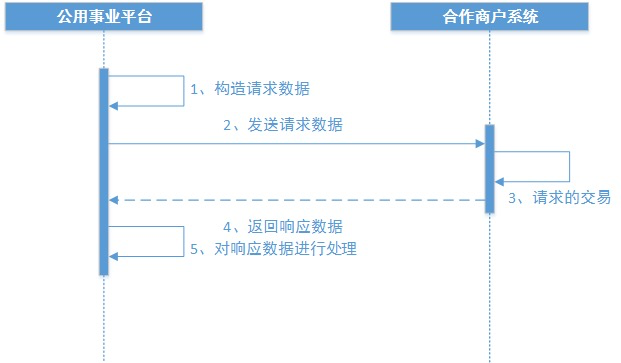
return null;

}

# 技术方案

## 通讯方式

### 在线交易



在线交易交互图

采用TCP/IP协议通讯，采用同步短连接交互模式，机构接收公用事业平台发来的交易请求，将请求处理结果发送至公用事业平台。

### 文件传输

文件传输：由公用事业平台方部署SFTP服务器，并建立相应的SFTP用户，设置该用户的工作目录，机构方可以按一定规则访问指定的文件目录。

# 接口规约说明

## 报文通用约定

### 字符集约定

所有的请求、响应、文本均使用UTF-8字符集进行编码。

### 参数名约定

公用事业平台请求与响应中所有的参数名称都由字母、数字组成。组成参数名称的除了第一个单词，所有单词第一个字母大写，其他字母小写，例如type、msgId。

### 基本参数约定

#### 时间格式

日期格式为yyyyMMddHH24miss, 例如：20090605211010 代表2009年6月5日21点10点10分（此处仅表明格式，不代表精度，如精确到天等，具体精度由各业务确定）。

#### 金额单位

所有的金额都以分为单位，精确到分，没有小数位。最大为12位。例如3元2角表示为320。

#### 文本格式

文本格式约定以竖线’|’分隔，最后以’|’结尾，必须有首行信息（根据具体业务而定，一般为多少条数据，多少金额）。

## 报文格式说明

### 报文帧格式概述

帧是传送信息的基本单元，帧的基本单元为8位字节。数据的传输顺序为低位在前，高位在后；低字节在前，高字节在后。

每帧由帧起始符标志域，数据长度，实际数据，校验域及帧结束域等5个域组成。每个域由若干字节组成。帧格式如下所示：

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 代码 | 字节数 | 说明 |
| 开始字节 | 68H | 1 | 开始 |
| 数据长度 | TRANS\_DATE\_LEN | 4 | BIN |
| 服务编码 |  | 6 | 参见各接口服务编码说明 |
| 数据 | DATA | 变长 | 根据交易不同组织的数据 |
| 结束字节 | 16H | 1 | 结束码 |

* 起始符

标识一帧信息的开始，其值为68H=01101000B

* 数据长度

为数据域的字节数。

* 数据域

JSON格式字符串数据，使用UTF-8字符集进行编码。

* 结束符

标识一帧信息的结束，其值为16H=00010110B

### 报文数据格式

{head:{tag1:value1,tag2:value2,...},body:{tag3:value3,tag4:value4, params:[{ param1:values1 },{ paramn:valuesn }],...}}

* head是报文数据头信息，记录了一系列约定数据，
* body是报文数据主体内容，承载了报文的业务数据，具体每个报文包含了哪些业务数据将在具体的业务接口中详细描述。

### 报文头格式

head:{version:’’,source:’’,desIfno:’’,servCode:’’,msgId:’’,msgTime:’’,extend:’’}

|  |  |
| --- | --- |
| 参数名称 | 参数值含义 |
| version | 报文的版本号，目前版本为：1.0.1。 |
| source | 报文来源的机构的简称。为固定值 ：ALIPAY |
| desIfno | 报文发送目的机构，常量，接入时确认。 |
| servCode | 服务编号，详见各个接口。 |
| msgId | 报文的id，对于每个报文都有一个标识，这个标识用于标识这笔报文。 |
| msgTime | 报文的发送时间。时间的格式为yyyyMMddHH24miss。例如20090603233822 |
| extend | 预留字段 |

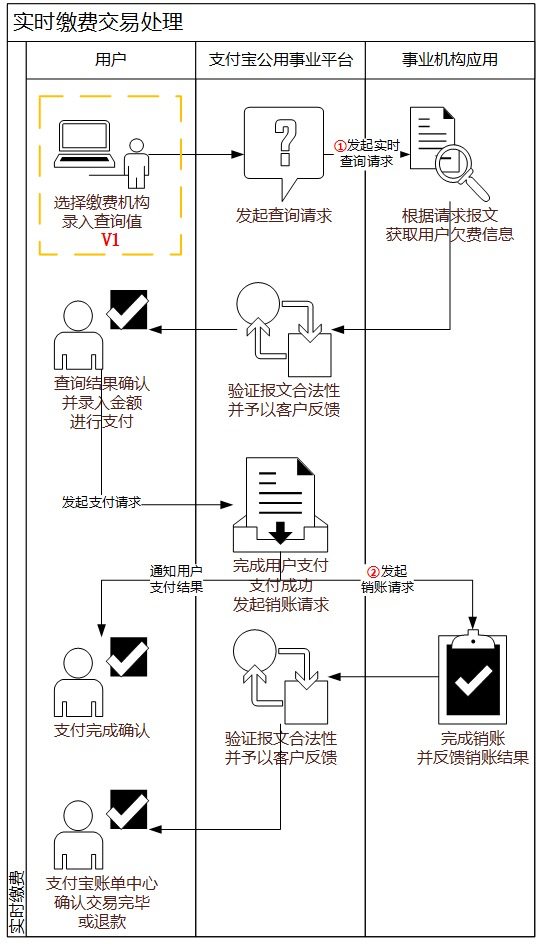
# 名词解释

|  |  |
| --- | --- |
| 名词定义 | 名词解释 |
| 机构 | 服务对象的简称，通常指水、电、燃、广电、电信等企事业单位 |
| 销帐机构 | 支付宝为每个机构设定的编码； |
| 用户单位 | 通常指用户所属机构的管理单位，即：通常指用户归属于机构下设的组织、营业网点、子机构、分支机构等； |
| 单位编号 | 指用户单位对应的编号，可由双方约定； |
| 清算单位 | 指机构下属财务独立核算、清算的组织、营业网点、子机构、分支机构等；即：对应支付宝资金清算时，款项划拨对象，与机构打款账号对应； |
| 出账机构 | 支付宝为每一个清算单位设定的编码； |
| 清算单位编号 | 指清算单位对应的编号，可由双方约定； |
| 客户 | 指机构所服务的具体自然人、法人等； |
| 客户编号 | 指有机构给客户分配的编号； |
| 用户 | 指机构的客户在使用机构具体产品时，由机构所分配的特定对象。如：客户某个住宅的用水、用气、机顶盒等计费对象； |
| 用户编号 | 指机构给用户分配的编号、服务账号、智能卡号等； |
| 账单 | 指用户消费的明细构成 |
| 应收金额 | 指用户每期消费费用合计 |
| 实收金额 | 指用户每期消费已缴纳部分费用合计 |
| 欠费金额 | 欠费金额=应收金额-实收金额+本期缴费滞纳金 |

# 实时缴费业务

## 缴费代收

### 代收场景描述



### ①欠费查询接口规范

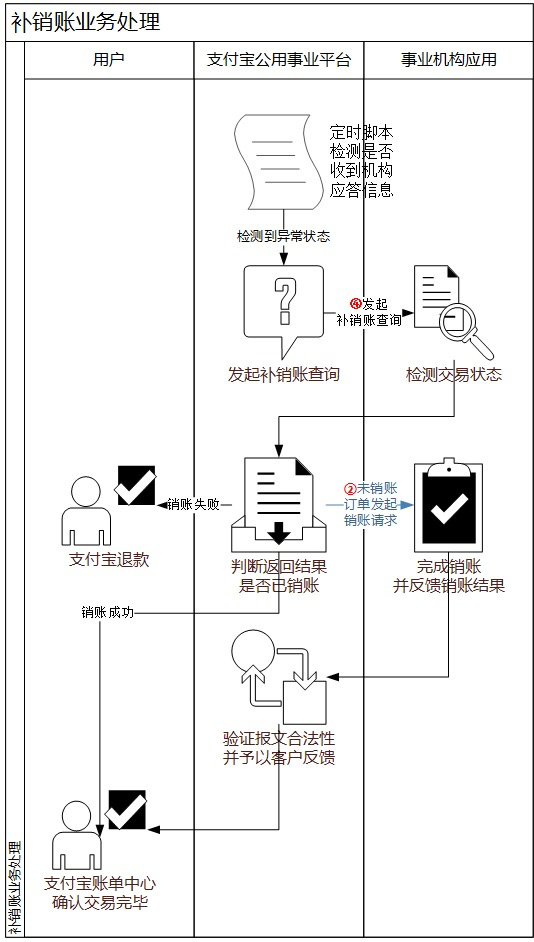
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 服务提供者 | 事业机构 | | | | | |
| 通讯规约 | SOCKET\_01/Socket&String | | | | | |
| 调用说明 | 公用事业平台进行欠费查询时调用此接口。 | | | | | |
| 服务编码 | 200001 | | | | | |
| 入参 | | | | | |  |
| 参数名称 | | 参数格式 | | | 含义 | 可空 |
| acctOrgNo | | VARCHAR2(16) | | | 清算单位，与机构打款账号对应，判断用户打款号正确性的标志。（见名词解释） | N |
| queryType | | VARCHAR2(8) | | | 查询条件的类别：0－按用户编号查询，1－按手机号码查询，2－按身份证号码查询，3－按原户号查询；（默认值：0） | N |
| queryValue | | VARCHAR2(64) | | | 对应查询类别的查询条件。 | N |
| bgnYm | | VARCHAR2(16) | | | 开始年月（可以缺省，目前暂无用） | Y |
| endYm | | VARCHAR2(16) | | | 结束年月（可以缺省，目前暂无用） | Y |
| busiType | | VARCHAR2(8) | | | 费用类型，默认值：11,表示水电燃，热力等主营费用 | N |
| extend | | VARCHAR2(256) | | | 预留 | Y |
| 出参 | | | | | |  |
| 参数名称 | | 参数格式 | | | 含义 | 可空 |
| rtnCode | | VARCHAR2(6) | | | 结果代码（[见附录响应码](#_响应码说明)） | N |
| rtnMsg | | VARCHAR2(128) | | | 结果描述 | N |
| consNo | | VARCHAR2(16) | | | 用户编号 | N |
| consName | | VARCHAR2(64) | | | 用户名称 | N |
| addr | | VARCHAR2(128) | | | 用户地址 | N |
| orgNo | | VARCHAR2(16) | | | 单位编码，用户所属营业厅，即该用户由哪个营业厅的抄表员负责抄表， | Y |
| orgName | | VARCHAR2(64) | | | 单位名称 | Y |
| acctOrgNo | | VARCHAR2(16) | | | 清算单位 | N |
| capitalNo | | VARCHAR2(32) | | | 资金编号：（可缺省，机构存在特定资金划拨需求时使用） | Y |
| consType | | VARCHAR2(8) | | | 用户分类（居民、非居民等用户分类） | Y |
| prepayAmt | | VARCHAR2(32) | | | 账户余额 | N |
| totalOweAmt | | VARCHAR2(32) | | | 合计欠费金额 | N |
| totalRcvblAmt | | VARCHAR2(32) | | | 合计应收金额 | N |
| totalRcvedAmt | | VARCHAR2(32) | | | 合计实收金额 | N |
| totalPenalty | | VARCHAR2(32) | | | 合计违约金（滞纳金） | N |
| recordCount | | VARCHAR2(8) | | | 明细记录条数 | N |
| rcvblDet 明细记录项目（可以为多条） | | | | | |  |
| rcvblAmtId | | | VARCHAR2(16) | 应收标识 | | Y |
| consNo | | | VARCHAR2(16) | 用户户号 | | N |
| consName | | | VARCHAR2(64) | 户名 | | N |
| orgNo | | | VARCHAR2(16) | 单位编码 | | Y |
| orgName | | | VARCHAR2(64) | 单位名称 | | Y |
| acctOrgNo | | | VARCHAR2(16) | 清算单位 | | N |
| rcvblYm | | | VARCHAR2(6) | 应收年月 | | N |
| tPq | | | VARCHAR2(32) | 电量、水量（吨）等：用于计费的计量结果 | | Y |
| rcvblAmt | | | VARCHAR2(32) | 应收金费 | | N |
| rcvedAmt | | | VARCHAR2(32) | 实收金费 | | N |
| rcvblPenalty | | | VARCHAR2(32) | 应收违约金（滞纳金） | | N |
| oweAmt | | | VARCHAR2(32) | 欠费小计 | | N |
| extend | | | VARCHAR2(256) | 预留 | | Y |

### ②代收缴费接口规范

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 服务提供者 | 事业机构 | | | |
| 通讯规约 | SOCKET\_01/Socket&String | | | |
| 调用说明 | 公用事业平台进行缴费时调用此接口。 | | | |
| 服务编码 | 200002 | | | |
| 入参 | | | |  |
| 参数名称 | 参数格式 | | 含义 | 可空 |
| acctOrgNo | VARCHAR2(16) | | 清算单位，与机构打款账号对应，判断用户打款号正确性的标志。 | N |
| bankSerial | VARCHAR2(32) | | 缴费流水号，由支付宝生成，对账维度之一 | N |
| bankDate | VARCHAR2(8) | | 缴费日期，支付宝传入，对账维度之一 | N |
| capitalNo | VARCHAR2(32) | | 资金编号，取值于欠费查询接口出参。 | Y |
| payMode | VARCHAR2(16) | | 缴费方式，用于区分不同的支付宝缴费（如普通代收，实时代扣等） | Y |
| rcvAmt | VARCHAR2(32) | | 交易金额 | N |
| bankDateTime | VARCHAR2(32) | | 精确时间，预留 | N |
| chargeCnt | VARHCAR2(8) | | 记录数 | N |
| extend | VARCHAR2(256) | | 预留 | Y |
| rcvDet明细记录项目（多条重复） | | | |  |
| consNo | VARCHAR2(16) | 用户户号 | | N |
| rcvblAmtId | VARCHAR2(16) | 应收标识，当有多条欠费时，用户在前台选择的明细项。 | | Y |
| rcvblYm | VARCHAR2(16) | 应收年月 | | Y |
| extendDet | VARCHAR2(256) | 预留 | | Y |
| 出参 | | | |  |
| 参数名称 | 参数格式 | 含义 | |  |
| rtnCode | VARCHAR2(6) | 结果代码（[见附录响应码](#_响应码说明)） | | N |
| rtnMsg | VARCHAR2(128) | 结果描述 | | N |
| extend | VARCHAR2(256) | 预留 | | Y |

## 补销账

### 补销账场景描述

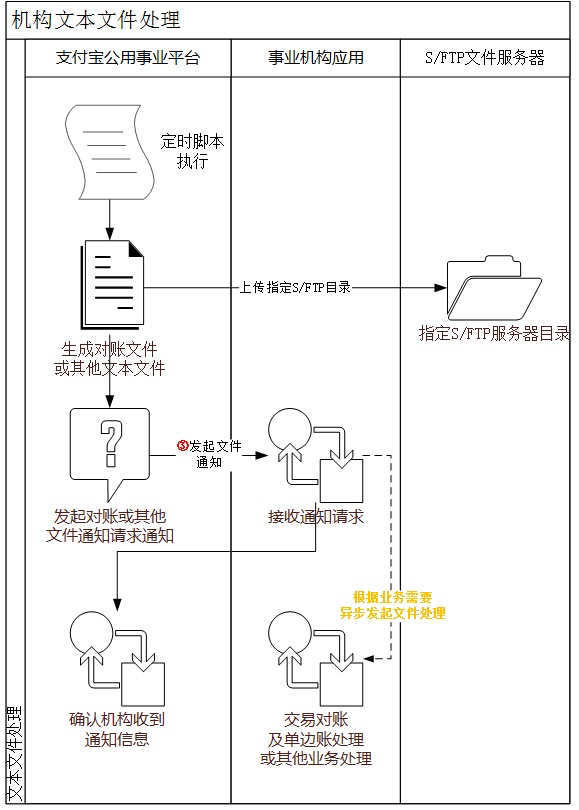


### ④销账结果查询接口规范

|  |  |  |  |
| --- | --- | --- | --- |
| 服务提供者 | 事业机构 | | |
| 通讯规约 | SOCKET\_01/Socket&String | | |
| 调用说明 | 公用事业平台发起文本推送时调用此接口 | | |
| 服务编码 | 200012 | | |
| 入参 | | |  |
| 参数名称 | 参数格式 | 含义 | 可空 |
| acctOrgNo | VARCHAR2(16) | 清算单位，与机构打款账号对应，判断用户打款号正确性的标志。 | N |
| bankSerial | VARCHAR2(32) | 缴费流水号：同代收缴费接口bankSerial | N |
| bankDate | VARCHAR2(16) | 缴费时间 | N |
| extend | VARCHAR2(256) | 预留 | Y |
| 出参 | | |  |
| 参数名称 | 参数格式 | 含义 | 可空 |
| rtnCode | VARCHAR2(6) | 结果代码（[见附录响应码](#_响应码说明)） | N |
| rtnMsg | VARCHAR2(128) | 结果描述 | N |
| instSerial | VARCHAR2(32) | 机构处理流水号 | N |
| extend | VARCHAR2(256) | 预留 | Y |

## 代收对账

### 代收对账场景描述



### 对账规则

1、对账文件或其他文本文件上传到指定S/FTP服务器后，发起文件上传通知，机构收到通知后予以应答；

2、因对账需要占用一定的时间，建议机构采用异步交易自动对账或其他文件处理，规避出现超时交易等异常

3、代收对账以支付宝对账文本文件为准，机构进行对账及单边帐处理；

单边帐处理规则：

支付宝多收，机构作补收或预收处理；

机构端多收，机构作交易冲正处理；

注：考虑到机构交易冲正权限开放存在一定交易风险，支付宝代收服务中不提供机构交易冲正服务接入；

4、机构无法进行单边处理的交易记录，可通过线下退款方式，由支付宝退款；

### 文件通知接口规范

|  |  |  |  |
| --- | --- | --- | --- |
| 服务提供者 | 事业机构 | | |
| 通讯规约 | SOCKET\_01/Socket&String | | |
| 调用说明 | 公用事业平台发起文本推送时调用此接口 | | |
| 服务编码 | 200011 | | |
| 入参 | | |  |
| 参数名称 | 参数格式 | 含义 | 可空 |
| acctOrgNo | VARCHAR2(16) | 清算单位，与机构打款账号对应，判断用户打款号正确性的标志。 | N |
| filePath | VARCHAR2(32) | 文件存放相对路径 | N |
| fileType | VARCHAR2(8) | 文件类型:  DSDZ：代收对账文本；REDK：代扣扣款反馈文本； | N |
| fileName | VARCHAR2(256) | 文件名，公共事业通知代收方时传入 | N |
| extend | VARCHAR2(256) | 预留 | Y |
| 出参 | | |  |
| 参数名称 | 参数格式 | 含义 | 可空 |
| rtnCode | VARCHAR2(6) | 结果代码（[见附录响应码](#_响应码说明)） | N |
| rtnMsg | VARCHAR2(128) | 结果描述 | N |
| extend | VARCHAR2(256) | 预留 | Y |

### 对账明细文件

#### 推送频率

按日推动；

#### 推送方向

由公用事业平台推送给机构信息系统；

#### 文件命名规则

文件名由销帐机构名称、出账机构名称、文件类别、清算单位、交易日期五部分组成，具体格式为：销帐机构名称（定值：ALIPAY）\_出账机构名称\_文件类型（定值：DSDZ）\_清算单位\_交易日期.txt

例如:ALIPAY\_出账机构\_DSDZ\_清算单位\_交易日期.txt

#### 对账文件格式

1、文件内容首行：

缴费总金额|缴费总笔数|，须与文件内容明细汇总一致。

2、文件明细内容格式：

每行字段之间以“|“线分割，首尾均为“|“线。

3、文本内容：

首行信息：

缴费总金额|缴费总笔数|

明细信息：

户号|缴费流水号|缴费日期|交易金额|记录数|用户户号$应收标识$应收年月$#<多条时重复>

# 附录

## 部分响应码说明

|  |  |  |
| --- | --- | --- |
| 成功代码 | 描述 | 解释 |
| 9999 | 交易成功 |  |
| 错误代码 | 错误描述 | 解释 |
| 程序类错误 | | |
| 0000 | 报文格式错误 | 报文的格式和标准的报文格式不一致。 |
| 0001 | 机构标识不合法 | instId中填写的接收报文的机构的简称和约定的不一致 |
| 0002 | 协议的版本号错误 | version版本号不正确 |
| 0003 | 参数格式错误 | 传递过去的参数格式错误 |
| 0004 | 关键数据为空 | 传递的关键的参数为空 |
| 0005 | 签名无效 | 验证签名失败 |
| 0006 | 解密失败 | 解密报文时错误，目前系统暂时不支持加密 |
| 0007 | 密钥id不存在 | 传递的certId参数不正确 |
| 0008 | 业务未开通 | 缴费的项目在销账机构中未开通。 |
| 0009 | 系统异常 | 销账机构处理的时候出现异常。 |
| 0010 | 与第三方通讯失败 | 销账机构和第三方单位通讯异常。 |
| 欠费查询错误 | | |
| 1000 | 缴费单已经支付 | 查询缴费单时，这表缴费单已经缴纳过 |
| 1001 | 缴费单未出账 | 有固定出账日的公共事业项目，查询的月份的费用未出账。 |
| 1002 | 查询的号码不合法 | 查询的号码不合法 |
| 1003 | 未欠费 | 查询的缴费信息用户没有欠费 |
| 1004 | 缴费号码超过受理期，请至收费单位缴费。 | 缴费号码超过受理期，请至收费单位缴费。 |
| 1005 | 暂时无法缴费或超过限定缴费金额，请咨询事业单位 | 暂时无法缴费或超过限定缴费金额，请咨询事业单位 |
| 1006 | 银行代扣期间禁止缴费 | 银行代扣期间禁止缴费 |
| 缴费类错误 | | |
| 2001 | 缴费单已经销账 | 缴费单已经在交纳过 |
| 2002 | 缴费金额不等 | 报文中的金额和销账机构中需要缴纳的金额不相等 |
| 2003 | 缴费号码超过受理期，请至收费单位缴费。 | 缴费号码超过受理期，请至收费单位缴费。 |
| 2004 | 超过限定缴费金额 | 超过限定缴费金额 |
| 2005 | 业务状态异常 | 业务状态异常，暂时无法缴费 |
| 查询确认错误 | | |
| 3000 | 没有该条记录 | 销账机构没有处理过这个缴费纪录。 |
| 3001 | 未知异常 | 销账机构处理这个缴费纪录的时候处理发生异常 |
| 3002 | 处理中 | 销账仍然处理中，可能在对账的时候来同步两边的缴费记录状态。 |
| 3003 | 处理状态失败 | 销账机构处理这个缴费纪录的时候处理失败，销账明确失败。（将实时退款） |
| 文件类错误 | | |
| 4000 | 文件格式错误 | 文件格式错误 |
| 4001 | 文件业务日期不正确 | 文件业务日期不正确 |
| 4002 | 文件摘要不正确 | 文件摘要不正确 |
| 4003 | 文件不存在 | 文件不存在 |
| 4004 | 文件已经处理 | 文件已经处理 |
| 4005 | 对账报文发送成功 | 成功 |

备注：

程序类错误、欠费查询错误、缴费类错误三大类错误代码中标红的错误代码支付宝不予退款，需谨慎使用。

## 完整响应码参照表（详情见附件）

