

**收付直通车**

**商户代付接口**

**V1.2**

**(文档更新地址：**

[**http://220.250.30.210:7052/cibhall/images/epay/%d0%cb%d2%b5%d2%f8%d0%d0%ca%d5%b8%b6%d6%b1%cd%a8%b3%b5%c9%cc%bb%a7%b4%fa%b8%b6%bd%d3%bf%da.docx**](http://220.250.30.210:7052/cibhall/images/epay/ÐËÒµÒøÐÐÊÕ¸¶Ö±Í¨³µÉÌ)**）**

**版本修订说明**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **编号** | **修订内容简述** | **修订日期** | **版本号** | **修订人** |
| 1 | 草稿 | 2014-09-01 | V0.1 | 叶黎明 |
| 2 | 增加商户查询接口 | 2015-04-07 | V1.1 | 郑尚书 |
| 3 | 增加批量代付接口 | 2015-12-24 | V1.2 | 谢振哲 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

目录

[1 接口安全策略 1](#_Toc438799393)

[1.1 安全组件 1](#_Toc438799394)

[1.1.1 对接口的统一说明 1](#_Toc438799395)

[1.1.2 使用说明 1](#_Toc438799396)

[1.2 数字签名 8](#_Toc438799397)

[1.2.1 签名规则 8](#_Toc438799398)

[1.2.2 签名算法示例 8](#_Toc438799399)

[2 环境说明 11](#_Toc438799400)

[2.1 代付接入地址 11](#_Toc438799401)

[2.2 接口频率限制说明 12](#_Toc438799402)

[3 接口调用 12](#_Toc438799403)

[3.1 单笔资金付款 12](#_Toc438799404)

[3.1.1 功能概述 12](#_Toc438799405)

[3.1.2 输入参数 12](#_Toc438799406)

[3.1.3 返回结果 13](#_Toc438799407)

[3.1.4 接口示例 13](#_Toc438799408)

[3.2 单笔代付查询 15](#_Toc438799409)

[3.2.1 功能概述 15](#_Toc438799410)

[3.2.2 输入参数 15](#_Toc438799411)

[3.2.3 返回结果 15](#_Toc438799412)

[3.2.4 接口示例 16](#_Toc438799413)

[3.3 批量资金付款 17](#_Toc438799414)

[3.3.1 功能概述 17](#_Toc438799415)

[3.3.2 输入参数 17](#_Toc438799416)

[3.3.3 返回结果 18](#_Toc438799417)

[3.3.4 接口示例 18](#_Toc438799418)

[3.4 批量代付查询 19](#_Toc438799419)

[3.4.1 功能概述 19](#_Toc438799420)

[3.4.2 输入参数 19](#_Toc438799421)

[3.4.3 返回结果 19](#_Toc438799422)

[3.4.4 接口示例 19](#_Toc438799423)

[3.5 商户信息查询接口 20](#_Toc438799424)

[3.5.1 功能概述 20](#_Toc438799425)

[3.5.2 输入参数 20](#_Toc438799426)

[3.5.3 返回结果 21](#_Toc438799427)

[3.5.4 接口示例 21](#_Toc438799428)

[4 对账服务 22](#_Toc438799429)

[4.1 资金对账文件下载 22](#_Toc438799430)

[4.1.1 功能概述 22](#_Toc438799431)

[4.1.2 输入参数 22](#_Toc438799432)

[4.1.3 返回结果 23](#_Toc438799433)

[4.1.4 接口示例 23](#_Toc438799434)

[5 银行代码列表 24](#_Toc438799435)

[5.1 银行代码列表文件下载 24](#_Toc438799436)

[5.1.1 功能概述 24](#_Toc438799437)

[5.1.2 输入参数 25](#_Toc438799438)

[5.1.3 返回结果 25](#_Toc438799439)

[5.1.4 接口示例 25](#_Toc438799440)

[6 错误码说明 26](#_Toc438799441)

[6.1 通用错误(10000-10099) 26](#_Toc438799442)

[6.2 服务分发错误(10100-10199) 27](#_Toc438799443)

[6.3 数据访问错误(10200-10299) 27](#_Toc438799444)

[6.4 后端系统交易错误(10300-10399) 27](#_Toc438799445)

[6.5 商户类错误(20000-20099) 27](#_Toc438799446)

[6.6 订单类错误(20100-20199) 28](#_Toc438799447)

[6.7 支付类错误(20200-20299) 28](#_Toc438799448)

[6.8 对账类错误(20500-20599) 29](#_Toc438799449)

[6.9 账户类错误(20700-20799) 29](#_Toc438799450)

[6.10 文件下载类错误(21000-21099) 30](#_Toc432759622)

# 接口安全策略

代付接口目前使用到的安全策略包括安全组件和数字签名两种方式。

# 安全组件

以下接口不使用安全组件：

资金对账文件下载接口、银行代码列表下载接口。

## 对接口的统一说明

* + - 1. 参数

接口传输的所有参数都使用UTF-8进行编码。

**兴业银行返回的应答或通知消息可能会由于升级增加参数，请验证应答签名或处理应答报文时注意允许这种情况。**

* + - 1. 调用方式

接口采用http/https连接方式进行通讯，通讯API在客户端中已编写完成，调用人员只需关注业务参数的填充工作。

* + - 1. IP限制

允许对接口的请求IP进行限制。代付商户在开通时可以向收付直通车系统提供一个或者多个固定IP用于做交易和查询。

## 使用说明

在客户端使用安全组件之前需要在客户端系统中做好如下的文件配置和依赖包的导入工作。

|  |  |
| --- | --- |
| 名称 | 说明 |
| API |  |
| appsvr.base. jar | 基础包(下载地址：  http://220.250.30.210:7052/cibhall/images/epay/jar/appsvr.base.jar) |
| appsvr.client. jar | 客户端包(下载地址：  http://220.250.30.210:7052/cibhall/images/epay/jar/appsvr.client.jar) |
| 配置文件 |  |
| conf | 配置文件，存放路径classpath:appsvr\_conf/conf |
| appsvr\_server.cert | 服务器证书（行方提供） |
| appsvr\_client.pfx | 客户端证书，根据conf配置（行方提供） |
| LIB | 依赖库（可自行下载） |
| commons\_codec-1.6.jar | 编解码器 |
| commons-io-2.2.jar | IO工具 |
| slf4j-api-1.7.10.jar | Log |
| httpclient-4.3.6.jar | http客户端 |
| httpcore-4.3.3.jar | http核心 |
| DEMO | 示例代码（可自行下载） |
| CibAppsvrDemo.7z | 客户端调用示例（下载地址：  http://220.250.30.210:7052/cibhall/images/epay/sdk/CibAppsvrDemo.zip） |

* + - 1. Conf配置文件说明

|  |  |
| --- | --- |
| 名称 | 说明 |
| appId | appId应用ID（行方提供，类似Q0000001） |
| appsvr\_server\_cert | 服务端证书文件路径 |
| appsvr\_client\_pfx | 客户端证书文件路径 |
| appsvr\_client\_pfx\_alias | 客户端证书alias（默认为appsvr\_client） |
| appsvr\_client\_pfx\_pwd | 客户端证书密码（默认为123456） |
| tokenUrl | Token请求URL，例<https://pay.cib.com.cn/payment/pymtApi.do>  （参照下文示例） |
| txnUrl | Txn请求URL，例<https://pay.cib.com.cn/payment/pymtApi.do>  （参照下文示例） |

测试环境配置示例：

|  |
| --- |
| appId=Q0000001  appsvr\_server\_cert=appsvr\_conf/appsvr\_server.cer  appsvr\_client\_pfx=appsvr\_conf/appsvr\_client.pfx  appsvr\_client\_pfx\_alias=appsvr\_client  appsvr\_client\_pfx\_pwd=123456  tokenUrl=https://3gtest.cib.com.cn:37031/payment/pymtApi.do  txnUrl=https://3gtest.cib.com.cn:37031/payment/pymtApi.do |

生产环境配置示例：

|  |
| --- |
| appId=Q0000001  appsvr\_server\_cert=appsvr\_conf/appsvr\_server.cer  appsvr\_client\_pfx=appsvr\_conf/appsvr\_client.pfx  appsvr\_client\_pfx\_alias=appsvr\_client  appsvr\_client\_pfx\_pwd=123456  tokenUrl=https://pay.cib.com.cn/payment/pymtApi.do  txnUrl=https://pay.cib.com.cn/payment/pymtApi.do |

* + - 1. 客户端初始化

客户端实例仅需要初始化一次。

* + - 1. MAVEN依赖配置示例

|  |
| --- |
| <dependencies>  <dependency>  <groupId>junit</groupId>  <artifactId>junit</artifactId>  <version>4.8.2</version>  </dependency>  <dependency>  <groupId>commons-io</groupId>  <artifactId>commons-io</artifactId>  <version>2.2</version>  </dependency>  <dependency>  <groupId>org.slf4j</groupId>  <artifactId>slf4j-api</artifactId>  <version>1.7.10</version>  </dependency>  <dependency>  <groupId>commons-logging</groupId>  <artifactId>commons-logging</artifactId>  <version>1.1.1</version>  </dependency>  <dependency>  <groupId>ch.qos.logback</groupId>  <artifactId>logback-core</artifactId>  <version>1.1.2</version>  </dependency>  <dependency>  <groupId>ch.qos.logback</groupId>  <artifactId>logback-classic</artifactId>  <version>1.1.2</version>  </dependency>  <dependency>  <groupId>ch.qos.logback</groupId>  <artifactId>logback-access</artifactId>  <version>1.1.2</version>  </dependency>  <dependency>  <groupId>org.apache.commons</groupId>  <artifactId>commons-lang3</artifactId>  <version>3.4</version>  </dependency>  <dependency>  <groupId>commons-codec</groupId>  <artifactId>commons-codec</artifactId>  <version>1.6</version>  </dependency>  <dependency>  <groupId>com.alibaba</groupId>  <artifactId>fastjson</artifactId>  <version>1.1.40</version>  </dependency>  <dependency>  <groupId>org.apache.httpcomponents</groupId>  <artifactId>httpclient</artifactId>  <version>4.3.6</version>  </dependency>  <dependency>  <groupId>org.apache.httpcomponents</groupId>  <artifactId>httpcore</artifactId>  <version>4.3.3</version>  </dependency>    <dependency>  <groupId>com.cib.cap4i.appsvr</groupId>  <artifactId>appsvr.base</artifactId>  <version>1.0.12.1</version>  </dependency>  <dependency>  <groupId>com.cib.cap4i.appsvr</groupId>  <artifactId>appsvr.client</artifactId>  <version>1.0.12.1</version>  </dependency>  </dependencies> |

* + - 1. 示例代码

|  |
| --- |
| **package** com.cib.appsvr;  **import** java.io.File;  **import** java.util.HashMap;  **import** java.util.Map;  import org.apache.commons.io.FileUtils;  **import** com.cib.cap4i.appsvr.base.base.AppResponse;  **import** com.cib.cap4i.appsvr.client.AppClientService;  /\*\*  \* **@author** huzj  \*/  **public** **class** AppServiceTest {  **public** **static** **void** main(String[] args) {  AppServiceTest t = **new** AppServiceTest();  **try** {  t.txnInfTest();  } **catch** (Exception e) {  e.printStackTrace();  }  }  @SuppressWarnings("unused")  **public** **void** txnInfTest() **throws** Exception {  /\*\*  \* 初始化客户端  \*  \* 建议:使用单例的初始化方式，让客户端服务类只初始化一次  \*/  AppClientService appClientService = **new** AppClientService("appsvr\_conf/conf");  /\*\*  \* 填充报文  \*/  Map<String, String> txnContent = **new** HashMap<String, String>();  // 1 ：单笔资金代付接口示例  /\*  \* txnContent.put("appid", "P0000137");  \* txnContent.put("service", "cib.epay.payment.pay");  \* txnContent.put("ver", "01");  \* txnContent.put("timestamp", "20151224101847");  \* txnContent.put("sub\_mrch", "XXXX门店");  \* txnContent.put("order\_no", "20110110721164046917");  \* txnContent.put("to\_bank\_no","309391000011");  \* txnContent.put("to\_acct\_no", "622909159009772010");  \* txnContent.put("to\_acct\_name", "张三");  \* txnContent.put("acct\_type", "0");  \* txnContent.put("cur", "CNY");  \* txnContent.put("trans\_amt", "1234.56");  \* txnContent.put("trans\_usage", "测试");  \*/  // 2 ：单笔代付查询接口示例  /\*  \* txnContent.put("appid", "P0000137");  \* txnContent.put("service", "cib.epay.payment.get");  \* txnContent.put("ver", "01");  \* txnContent.put("timestamp", "20151224101847");  \* txnContent.put("order\_no", "20110110165455000001");  \*/  // 3 ：批量资金代付接口示例  /\*  \* txnContent.put("appid", "P0000137");  \* txnContent.put("service",  \* "cib.epay.payment.batchPayApi");  \* txnContent.put("ver", "01");  \* txnContent.put("timestamp", "20151224101847");  \* txnContent.put("order\_date", "20151223");  \* txnContent.put("file\_name", "BAT\_P0000137\_2015122101.txt");  \* //(上传模板详见/templates/BAT\_P0000137\_2015122101.txt)  \* File batchFile = new  \* File("templates/BAT\_P0000137\_2015122101.txt");  \* byte[] byteArray = FileUtils.readFileToByteArray(batchFile);  \*/  // 4 ：批量代付查询接口示例  /\*  \* txnContent.put("appid", "P0000137");  \* txnContent.put("service",  \* "cib.epay.payment.batchPayQueryApi");  \* txnContent.put("ver", "01");  \* txnContent.put("timestamp", "20151224101847");  \* txnContent.put("batId", "2015122372471000");  \*/  // 5 ：商户信息查询接口示例  txnContent.put("appid", "P0000137");  txnContent.put("service", "cib.epay.payment.getMrch");  txnContent.put("ver", "01");  txnContent.put("timestamp", "20151224101847");  /\*\*  \* 发送报文  \*/  AppResponse res = **null**;  **try** {  // 批量接口带附件请使用接口:AppClientService.txn(Map<String,  // tring> map, byte[]  // byteArray,String serviceName, String version)  res = appClientService.txn(txnContent, txnContent.get("service"), txnContent.get("ver"));  /\*\*  \* 处理返回结果  \*/  String body = res.getBody();  **byte**[] backFile = res.getByteArray();// 获取响应附件  } **catch** (Exception e) {  // **TODO** 此处抛出异常为通讯组件异常，若抛错建议查看详细日志解决  }  }  } |

# 数字签名

仅有以下接口使用数字签名：

资金对账文件下载接口、银行代码列表下载接口。

为防止数据在传输过程中被篡改，或其他人伪造支付指令，商户与银行间通讯信息均应进行数字签名。

## 签名规则

* + - 1. 拼装消息内容

将键值对以key=value形式组装成一个数组，将该数据进行排序后使用&进行拼装，最后加入商户令牌（token）。例如

timestamp=20150708111509

appid=P0000014

service=cib.epay.payment.receiptFile

ver=01

trans\_date=20150623

rcpt\_type=1

得到的拼装字符串为：

**appid=P0000014&rcpt\_type=1&service=cib.epay.payment.receiptFile&timestamp=20150708111509&trans\_date=20150623&ver=01&84A8251FECD84E599B7B1037579E6A4D**

**说明：若参数为空，可不上送（不加入到拼装字符串中），也无需加入计算mac；若要上送，相应的也需加入去计算mac。**

* + - 1. 使用UTF-8转码后生成签名(mac)

对拼装好的消息内容进行UTF-8转码后，用SHA-1算法进行加密，并将得到的数据转换为16进制，例如计算出mac为（仅为示例值）：

**30F2B0CA895BB25881CEE20CB32E5EEADCB9E18D**

## 签名算法示例

* + - 1. Java示例

/\*\*

\* 生成用于MAC计算的参数字符串。<br>

\*

\* @return 模式为key=value&key=value

\*/

public static String generateParamStr(Map<String, String>params) {

// 取所有非空字段内容（除mac以外），塞入列表

List<String>paramList = new ArrayList<String>();

for (String key : params.keySet()) {

if (“mac”.equals(key)) {

continue;

}

String val = params.get(key);

paramList.add(key + "=" + val);

}

// 防护

if (paramList.size() == 0) {

return null;

}

// 对列表进行排序

Collections.sort(paramList);

// 以&符分割拼装成字符串

StringBuildersb = new StringBuilder();

sb.append(paramList.get(0));

for (inti = 1; i<paramList.size(); i++) {

sb.append("&").append(paramList.get(i));

}

returnsb.toString();

}

/\*\*

\* 将byte数组转换为16进制格式的字符串

\*

\* @param bytes待转换数组

\* @return 16进制格式的字符串

\*/

public static String bytesToHexStr(byte[] bytes) {

StringBuffersb = new StringBuffer(bytes.length \* 2);

for (inti = 0; i<bytes.length; i++) {

sb.append(hexChar[(bytes[i] & 0xf0) >>> 4]);

sb.append(hexChar[bytes[i] & 0x0f]);

}

returnsb.toString();

}

/\*\*

\* SHA摘要算法，输入内容将被UTF-8编码

\*

\* @param content输入明文

\* @return 内容摘要，40位16进制字符串

\*/

public static String encryptBySHA(String content) {

if (content == null)

return null;

try {

MessageDigest md = MessageDigest.getInstance("SHA-1");

byte[] output = md.digest(content.getBytes("UTF-8"));

returnbytesToHexStr(output);

} catch (NoSuchAlgorithmException e) {

log.error("无法加载SHA算法。", e);

} catch (UnsupportedEncodingException e) {

log.error("无法将输入字符串转换到utf-8编码。", e);

}

return null;

}

* + - 1. .Net示例

/\*\*

\* 生成用于MAC计算的参数字符串。

\* @return 模式为key=value&key=value

\*/

public static string generateParamStr(Dictionary<string,string> \_params) {

// 取所有非空字段内容（除mac以外），塞入列表

ArrayListparamsList = new ArrayList();

foreach (KeyValuePair<string,string> key in \_params){

if (key.Key.Equals("mac")) {

continue;

}

paramsList.Add(key.Key + "=" + key.Value);

}

// 防护

if (paramsList.Count<= 0){

return null;

}

// 对列表进行排序

paramsList.Sort();

// 以&符分割拼装成字符串

stringsb="";

for (inti = 0; i<paramsList.Count; i++){

sb += paramsList[i].ToString() + "&";

}

sb = sb.TrimEnd('&');

returnsb;

}

/\*\*

\* SHA摘要算法，输入内容将被UTF-8编码

\* @param content 输入明文

\* @return 内容摘要，位进制字符串

\*/

public static string encryptBySHA(string content) {

if (content == null)

return null;

try {

SHA1CryptoServiceProvider sha = new SHA1CryptoServiceProvider();

byte[] str2 = sha.ComputeHash(Encoding.UTF8.GetBytes(content));

sha.Clear();

(sha as IDisposable).Dispose();

return Convert.ToBase64String(str2);

} catch {

//log.error("无法加载SHA算法。", e);

throw;

}

}

# 环境说明

## 代付接入地址

|  |  |  |  |
| --- | --- | --- | --- |
| **类型** | | **环境类型** | **URL** |
| **商户申请地址** | | 生产环境 | https://pay.cib.com.cn/merchant/ |
| 测试环境 | https://3gtest.cib.com.cn:37031/merchant/ |
| **代付接入地址** | **安全组件配置** | 生产环境 | https://pay.cib.com.cn/payment/pymtApi.do |
| 测试环境 | https://3gtest.cib.com.cn:37031/payment/pymtApi.do |
| **对账文件下载** | 生产环境 | https://pay.cib.com.cn/payment/api |
| 测试环境 | https://3gtest.cib.com.cn:37031/payment/api |
| **银行列表下载** | 生产环境 | <https://pay.cib.com.cn/acquire/cashier>.do |
| 测试环境 | https://3gtest.cib.com.cn:37031/acquire/cashier.do |

## 接口频率限制说明

|  |  |  |
| --- | --- | --- |
| **接口名称** | **服务名称** | **每日限额** |
| 单笔资金代付 | cib.epay.payment.pay | 无 |
| 单笔代付查询 | cib.epay.payment.get | 无 |
| 批量资金代付 | cib.epay.payment.batchPayApi | 无 |
| 批量代付查询 | cib.epay.payment.batchPayQueryApi | 无 |
| 商户信息查询 | cib.epay.payment.getMrch | 无 |

# 接口调用

## 单笔资金付款

### 功能概述

方向：商户→银行。

输入商户号、订单号、对方银行名称、对方银行卡号、对方方户名、对方账户类型、付款金额，申请进行资金支付。

交易结果判断标准：

交易成功：正常返回接口中的transStatus为1；

交易失败：正常返回接口中的transStatus为2；

其它情况（包括正常返回接口中的transStatus为3以及异常返回）建议都当作未决，建议隔5分钟后调用单笔代付查询接口查询具体的交易结果。异常码含义详见[5.异常码说明](#_错误码说明)。

### 输入参数

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **参数名称** | **类型** | **必输** | **示例值** | **描述** |
| timestamp | char(14) | √ | 20140831212747 | 时间戳，如果与当前时间差距超过30分钟，请求会被拒绝 |
| appid | char(8) | √ | P0000001 | 商户代号，定长8位 |
| service | varchar(100) | √ | cib.epay.payment.pay | 服务名称 |
| ver | char(2) | √ | 01 | 接口版本号，当前版本为01。 |
| sub\_mrch | varchar(20) |  | XXXX门店 | 二级商户名称，指签约商户下属二级商户名称，可选参数。 |
| order\_no | varchar(32) | √ | 20110110165455000001 | 订单号，同一商户号下，所有订单号必须唯一，用于对账。 |
| to\_bank\_no | char(14) | √ | 309391000011 | 人行体系支付行号 |
| to\_acct\_no | varchar(32) | √ | 622909159009772010 | 对方银行卡号 |
| to\_acct\_name | varchar(60) | √ | 张三 | 对方账户户名 |
| acct\_type | char(1) | √ | 0 | 对方账户类型：  0-储蓄卡;1-信用卡;2-企业账户 |
| cur | char(3) | √ | CNY | 订单币种，使用ISO4217标准货币代码，目前支持：CNY |
| trans\_amt | dec(17,2) | √ | 1234.56 | 付款金额，单位为元，小数位2位 |
| trans\_usage | varchar(60) | √ | 退款，原订单号xxxxx | 转账用途 |

【参数说明】

1. 同一商户号下订单号必须保持唯一，不可重复发起。

### 返回结果

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **名称** | **类型** | **非空** | **示例值** | **描述** |
| sno | char(16) | √ | 2014090100000001 | 银行系统流水号(Serial NO) |
| mrchNo | char(8) | √ | P0000001 | 商户代号 |
| subMrchName | varchar(60) |  | XXX门店 | 二级商户名 |
| orderNo | varchar(32) | √ | 20140901165455000001 | 订单编号 |
| transDate | char(8) | √ | 20140901 | 付款日期，格式为yyyymmdd |
| transTime | char(6) | √ | 112039 | 付款时间，格式为hhmmss |
| transStatus | char(1) | √ | 1 | 交易状态，1-成功；2-失败；3-未决 |
| transAmt | dec(17,2) | √ | 1234.56 | 付款金额 |
| transFee | dec(17,2) | √ | 1.23 | 交易手续费 |
| toAcctNo | varchar(32) | √ | 622909888888888888 | 收款账号 |
| toAcctName | varchar(60) | √ | 张三 | 收款户名 |
| transUsage | varchar(60) | √ | 退款，原订单号xxxxx | 转账用途 |
| remark | varchar(255) |  |  | 备注信息，如果状态为失败，则该字段为失败原因 |

### 接口示例

详见1.1.2.4示例单笔资金付款接口

* + - 1. 输入示例(HTTP POST)

timestamp=20140901113544

appid=P0000001

service=cib.epay.payment.pay

ver=01

sub\_mrch=陆家嘴分店

ord\_no=20110331201103312011033120110331

to\_bank\_no=309391000011

to\_acct\_no=622909159009772010

to\_acct\_name=张三

acct\_type=0

cur=CNY

trans\_amt=1234.56

trans\_usage=退款

acct\_no=622909159009772010

* + - 1. 输出示例(JSON)

**【成功结果】**

{

"sno": "2014090100000001",

"mrchNo": "P0000001",

"subMrchName": "陆家嘴分店",

"orderNo": "20140901165455000001",

"transDate": "20140901",

"transTime": "112039",

"transStatus": "1",

"transAmount": 1234.56,

"transFee": 1.23,

"toAcctNo": "622909888888888888",

"toAcctName": "张三",

"transUsage": "退款",

"remark": "代付成功"

}

**【失败结果】**

{

"sno": "2014090100000001",

"mrchNo": "P0000001",

"subMrchName": "陆家嘴分店",

"orderNo": "20140901165455000001",

"transDate": "20140901",

"transTime": "112039",

"transStatus": "2",

"transAmount": 1234.56,

"transFee": 1.23,

"toAcctNo": "622909888888888888",

"toAcctName": "张三",

"transUsage": "退款",

"remark": "代付失败"

}

## 单笔代付查询

### 功能概述

方向：商户→银行。

输入商户代号、订单号完成单笔资金代付结果查询。

交易结果判断标准：

在调用单笔资金付款接口后，建议隔5分钟以后再调用本接口，否则可能由于收付直通车未生成订单而返回订单不存在的结果。

除transStatus为2或返回errcode=”EPAY\_20102”（无匹配代付订单）外，其它结果建议继续把查询的订单当作交易未决处理。

### 输入参数

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **参数名称** | **类型** | **必输** | **示例值** | **描述** |
| timestamp | char(14) | √ | 20140831212747 | 时间戳，如果与当前时间差距超过30分钟，请求会被拒绝 |
| appid | char(8) | √ | P0000001 | 商户代号，定长8位 |
| service | varchar(100) | √ | cib.epay.payment.get | 服务名称 |
| ver | char(2) | √ | 01 | 接口版本号，固定01。 |
| order\_no | varchar(32) | √ | 20110110165455000001 | 订单号，同一商户号下，所有订单号必须唯一，用于对账。 |

### 返回结果

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **名称** | **类型** | **非空** | **示例值** | **描述** |
| sno | char(16) | √ | 2014090100000001 | 银行系统流水号(Serial NO) |
| mrchNo | char(8) | √ | P0000001 | 商户代号 |
| subMrchName | varchar(60) |  | XXX门店 | 二级商户名 |
| orderNo | varchar(32) | √ | 20140901165455000001 | 订单编号 |
| transDate | char(8) | √ | 20140901 | 付款日期，格式为yyyymmdd |
| transTime | char(6) | √ | 112039 | 付款时间，格式为hhmmss |
| transStatus | char(1) | √ | 1 | 交易状态，1-成功；2-失败；3-未决 |
| transAmt | dec(17,2) | √ | 1234.56 | 付款金额 |
| transFee | dec(17,2) | √ | 1.23 | 交易手续费 |
| toAcctNo | varchar(32) | √ | 622909888888888888 | 收款账号 |
| toAcctName | varchar(60) | √ | 张三 | 收款户名 |
| transUsage | varchar(60) | √ | 退款，原订单号xxxxx | 转账用途 |
| remark | varchar(255) |  |  | 其他备注信息 |

### 接口示例

详见1.1.2.4示例单笔代付查询接口。

* + - 1. 输入示例(HTTP POST)

timestamp=20140901113544

appid=P0000001

service=cib.epay.payment.get

ver=01

ord\_no=20140901165455000001

* + - 1. 输出示例(JSON)

**【成功结果】**

{

"sno": "2014090100000001",

"mrchNo": "P0000001",

"subMrch": "陆家嘴分店",

"orderNo": "20140901165455000001",

"transDate": "20140901",

"transTime": "112039",

"transStatus": "1",

"transAmount": 1234.56,

"transFee": 1.23,

"toAcctNo": "622909888888888888",

"toAcctName": "张三",

"transUsage": "退款",

"remark": ""

}

**【失败结果】**

{

"errcode": "10301",

"errmsg": "1627(1627)余额不足,请确认账户是否有冻结控制"

}

## 批量资金付款

### 功能概述

方向：商户→银行。

输入订单日期，批量付款文件，申请进行批量资金支付。

### 输入参数

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **参数名称** | **类型** | **必输** | **示例值** | **描述** |
| timestamp | char(14) | √ | 20140831212747 | 时间戳，如果与当前时间差距超过30分钟，请求会被拒绝 |
| appid | char(8) | √ | P0000001 | 商户代号，定长8位 |
| service | varchar(100) | √ | cib.epay.payment.batchPayApi | 服务名称 |
| ver | char(2) | √ | 01 | 接口版本号，当前版本为01。 |
| order\_date | char(8) | √ | 20151223 | 商户订单日期 |
| file\_name | varchar(50) | √ | BATCH\_PAY\_20151201.txt | 批量代付文件名 |
| **以下为请求附件** | | | | |
| byteArray | byte[] | √ |  | 批量代付附件（将批量文件转为  byte数组作为参数传入  AppClientService.txn(Map  <String, String> map,  byte[] byteArray, String  serviceName, String  version)方法即可） |

【参数说明】

1. 同一商户号下订单号必须保持唯一，不可重复发起。
2. 批量代付附件格式如下：

输入文件格式：

订单号|二级商户名称(可为空)|对方银行卡号|对方账户户名|人行体系支付行号|对方账户类型|付款金额|用途|

### 返回结果

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **名称** | **类型** | **非空** | **示例值** | **描述** |
| bat\_id | char(16) | √ | 2015122272423048 | 批次流水号 |
| bat\_status | char(1) | √ | 0 | 批次状态:0-文件接收成功；1-入批次明细成功；2-交易成功；3-交易失败；4-交易部分成功；5回盘文件成功 |
| bat\_status\_label | char(10) |  | 文件接收成功 | 批次状态描述 |

### 接口示例

详见1.1.2.4示例批量资金付款接口

* + - 1. 输入示例(HTTP POST)

timestamp=20151223140551

appid=P0000001

service=cib.epay.payment.batchPayApi

ver=01

order\_date=20151223

file\_name= BATCH\_PAY\_20151201.txt

* + - 1. 输出示例(JSON)

**【成功结果】**

{

"bat\_id":"2015122372453002",

"bat\_status":"0",

"bat\_status\_label":"文件接收成功"

}

**【失败结果】**

{

"errcode":"EPAY\_20251",

"errmsg":"第1条数据订单号超长"

}

## 批量代付查询

### 功能概述

方向：商户→银行。

输入批次号完成批量资金代付结果查询。

在调用批量资金付款接口后，建议隔15分钟以后再调用本接口，否则可能由于收付直通车未生成订单而返回订单不存在的结果。

### 输入参数

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **参数名称** | **类型** | **必输** | **示例值** | **描述** |
| timestamp | char(14) | √ | 20140831212747 | 时间戳，如果与当前时间差距超过30分钟，请求会被拒绝 |
| appid | char(8) | √ | P0000001 | 商户代号，定长8位 |
| service | varchar(100) | √ | cib.epay.payment.batchPayQueryApi | 服务名称 |
| ver | char(2) | √ | 01 | 接口版本号，固定01。 |
| batId | char(16) | √ | 2015122272423048 | 批次编号 |

【参数说明】

1. 批量代付查询回盘文件格式如下：

输出文件格式：

订单号|二级商户|付款日期|付款时间|转出账户|转出账户名称|对方银行卡号|对方账户户名|对方账户类型|付款金额|用途|手续费|交易状态|描述|

### 返回结果

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **名称** | **类型** | **非空** | **示例值** | **描述** |
| batId | char(16) | √ | 2015122272423048 | 批次编号 |
| orderDate | char(8) | √ | 20151223 | 订单日期 |
| batStatus | char(1) | √ | 5 | 批次状态:0-文件接收成功；1-入批次明细成功；2-交易成功；3-交易失败；4-交易部分成功；5回盘文件成功 |
| batStatusLabel | char(10) | √ | 回盘文件成功 | 批次状态描述 |
| fileName | varchar(50) | √ |  | 回盘文件名称 |
| **以下为响应附件** | | | | |
| byteArray | byte[] | √ |  | 批量代付查询附件（从响应对象中直接获取该属性即可得到回盘文件的byte数值：  byte[] backFile = res.getByteArray();） |

### 接口示例

详见1.1.2.4示例批量代付查询接口。

* + - 1. 输入示例(HTTP POST)

timestamp=20151223144223

appid=P0000001

service=cib.epay.payment.batchPayQueryApi

ver=01

batId=2015122272423048

* + - 1. 输出示例(JSON)

**【成功结果】**

{

"batId":"2015122272423048",

"batStatus":"5",

"batStatusLabel":"回盘文件成功",

"fileName":"Q0000241\_2015122272423048.out",

"orderDate":"20151222"

}

**【失败结果】**

{

"errcode":"EPAY\_20254",

"errmsg":"批次不存在，请检查批次号是否正确"

}

## 商户信息查询接口

### 功能概述

方向：商户→银行。

输入商户代号完成商户信息查询。

### 输入参数

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **参数名称** | **类型** | **必输** | **示例值** | **描述** |
| timestamp | char(14) | √ | 20140831212747 | 时间戳，如果与当前时间差距超过30分钟，请求会被拒绝 |
| appid | char(8) | √ | P0000001 | 商户代号，定长8位 |
| service | varchar(100) | √ | cib.epay.payment.getMrch | 服务名称 |
| ver | char(2) | √ | 01 | 接口版本号，固定01。 |

### 返回结果

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **名称** | **类型** | **非空** | **示例值** | **描述** |
| mrchNo | char(8) | √ | P0000001 | 商户代号 |
| mrchName | varchar(20) | √ | XXX | 商户名 |
| mrchType | char(1) | √ | 1 | 商户类型,:1-智能代收商户；2-智能代付商户； |
| corpId | char(10) | √ | 3500506511 | 企业客户号 |
| settleAccount | char(18) | √ | 356980100100349183 | 商户结算账号 |
| settleAccountName | varchar(60) | √ | YYYY企业户 | 商户结算账户名称 |
| mrchStatus | char(1) | √ | 1 | 商户当前状态码:0-无效；1-有效；2-冻结 |
| mrchStatusLabel | varchar(60) | √ | 有效 | 商户当前状态描述 |

### 接口示例

详见1.1.2.4示例商户查询接口。

* + - 1. 输入示例(HTTP POST)

使用https协议将参数post到支付网关接入地址。

timestamp=20150409105658

appid=P0000001

service=cib.epay.payment.getMrch

ver=01

* + - 1. 输出示例(JSON)

**【成功结果】**

{

"corpId":"1100019649",

"mrchName":"美菱集团",

"mrchNo":"P0000001",

"mrchStatus":"1",

"mrchStatusLabel":"有效",

"mrchType":"2",

"settleAccount":"117010101400122860",

"settleAcctName":"美菱集团控股有限公司半角字符"

}

**【失败结果】**

{

"errcode":"EPAY\_20205",

"errmsg":"MAC校验失败"

}

# 对账服务

## 资金对账文件下载

### 功能概述

该功能适用于商户系统与银行系统间T+1日对明细账。对账文件的查询有效期为90天，如需回补有效期范围外的资金对账文件，需要向银行业务部门另外申请。

### 输入参数

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **参数名称** | **类型** | **必输** | **示例值** | **描述** |
| timestamp | char(14) | √ | 20150708103434 | 时间戳。格式：yyyyMMddHHmmss |
| appid | varchar(64) | √ | P0000014 | 由系统分配外围系统应用编号 |
| service | varchar(100) | √ | cib.epay.payment.receiptFile | 服务名称。 |
| ver | char(2) | √ | 01 | 接口版本号，固定01。 |
| **以下为请求内容正文** | | | | |
| trans\_date | char(8) | √ | 20150623 | 交易日期，格式为  yyyyMMdd |
| rcpt\_type | char(1) | √ | 1 | 回单类型：  0-入账回单；1-出账回单；2-手续费回单 |
| **以下为签名相关信息** | | | | |
| mac | varchar(40) | √ | 30F2B0CA895BB25881CEE20CB32E5EEADCB9E18D | 消息校验码 |

### 返回结果

**【成功结果】**

Zip压缩文件，包含指定的商户资金对账文件。

**【失败结果】**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **参数名称** | **类型** | **必输** | **示例值** | **描述** |
| errcode | varchar(10) | √ | 10003 | 错误代码，10003对账文件不存在！ |
| errmsg | varchar(255) | √ | 错误信息 | 发送错误时返回的错误信息 |

**【资金对账文件】**

1. 资金对账文件为txt格式，文件名：资金对账文件名前缀+商户号+交易日期(yyyyMMdd)+回单类型(0/1).txt

如：NPRCPTP0000014201506231.txt。

1. 资金对账文件格式为：**订单日期|流水号|商户代号|订单编号|交易帐号(后4位)|支付金额|退款金额|手续费|订单状态|交易时间|结算日期|**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **名称** | **类型** | **非空** | **示例值** | **描述** |
| transDate | char(8) | √ | 20150623 | 订单交易日期 |
| sno | char(16) | √ | 2015062370499026 | 订单流水号 |
| mrchNo | char(8) | √ | P0000014 | 商户号 |
| orderNo | varchar(32) | √ | 20110110623105358687 | 订单编号 |
| maskAcctNo | char(6) | √ | \*0789\* | 屏蔽的银行账号 |
| transAmt | dec(17,2) | √ | 10.01 | 支付金额 |
|  |  |  |  | 退款金额，暂为空 |
| transFee | dec(17,2) | √ | 0.10 | 手续费 |
|  |  |  |  | 订单状态，暂为空 |
| transTime | char(6) | √ | 105411 | 订单交易发生时间 |
| settleDate | char(8) | √ | 20150623 | 资金结算日期 |

### 接口示例

详见1.2.2示例。

**输入示例**

使用https协议将参数post到支付网关接入地址。

timestamp=20150708111509

appid=P0000014

service=cib.epay.payment.receiptFile

ver=01

trans\_date=20150623

rcpt\_type=1

mac=EB1F4EFA06C47E95473F8AE40B8216BE77B17008

**输出示例**

**【成功结果】**

{

zip\_20150708010400.zip压缩文件

}

**【失败结果】**

{

"errcode": "10003",

"errmsg": "对账文件不存在"

}

# 银行代码列表

## 银行代码列表文件下载

### 功能概述

|  |  |
| --- | --- |
| **服务名称** | **cib.epay.acquire.download** |
| **接口版本** | 01 |
| **通讯方向** | 商户→银行 |
| **通讯模式** | [无用户参与通讯模式](#_无用户参与通讯模式) |
| **SDK示例** | 暂无 |
| **服务用途** | 该功能适用于商户系统每日下载更新银行代码列表 |
| **接口返回** | 成功时返回ZIP格式压缩包，失败时返回JSON格式字符串 |
| **回调方式** | 无 |

### 输入参数

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **参数名称** | **类型** | **必输** | **示例值** | **描述** |
| timestamp | char(14) | √ | 20150817135914 | 时间戳。格式：yyyyMMddHHmmss |
| appid | varchar(64) | √ | A0000012 | 由系统分配外围系统应用编号 |
| service | varchar(100) | √ | cib.epay.acquire.download | 服务名称。 |
| ver | char(2) | √ | 01 | 接口版本号，固定01。 |
| **以下为请求内容正文** | | | | |
| download\_type | char(2) | √ | 01 | 01行号下载 |
| **以下为签名相关信息** | | | | |
| mac | varchar(40) | √ | 0CD2273CC22142AB28EC900403DC7AE6B75ACF61 | 消息校验码 |

### 返回结果

**【成功结果】**

Zip压缩文件，包含当日的银行代码列表文件。

**【失败结果】**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **参数名称** | **类型** | **必输** | **示例值** | **描述** |
| errcode | varchar(10) | √ | 10003 | 错误代码，10003文件不存在！ |
| errmsg | varchar(255) | √ | 错误信息 | 发送错误时返回的错误信息 |

**【银行代码列表文件】**

1. 银行列表文件为txt格式，文件名：BANK\_NO\_文件日期(yyyyMMdd).txt。

如：BANK\_NO\_20150817.txt。

1. 银行列表文件格式为：

每行各字段使用竖线进行分割。格式如下：

**银行行号|银行类型|银行名称|银行简称**

### 接口示例

**输入示例**

使用https协议将参数post到支付网关接入地址。

timestamp=20150817135914

appid= A0000012

service=cib.epay.acquire.download

ver=01

download\_type =01

mac=0CD2273CC22142AB28EC900403DC7AE6B75ACF61

**输出示例**

**【成功结果】**

BANK\_NO\_20150817.zip压缩文件

**【失败结果】**

{

"errcode": "10003",

"errmsg": "文件不存在"

}

# 错误码说明

## 通用错误(10000-10099)

|  |  |  |
| --- | --- | --- |
| **错误码** | **错误描述** | **错误原因或解决方案** |
| 10000 | 未知错误 | 系统发生未预见的错误，请联系项目组进行排查。**订单状态未知**，**建议商户稍后发起查询确定订单状态。** |
| 10001 | 必输要素不能为空，请检查输入要素。 | 请参见相应的接口说明，确保必输字段都有赋值。 |
| 10002 | 输入内容格式不正确，请检查输入格式或数据。 | 请确保输入内容格式符合系统需求。 |
| 10003 | 系统获取不到相关文件 | 请检查指定的文件是否在相应的位置，或是否送错文件路径 |
| 10004 | 会话过期 | 请检查后端系统连通性 |
| 10005 | 发送HTTP请求失败 | 请检查网络状态 |
| 10006 | 文件生成失败 | 请检查文件目录权限、可用剩余空间等 |

## 服务分发错误(10100-10199)

|  |  |  |
| --- | --- | --- |
| **错误码** | **错误描述** | **错误原因或解决方案** |
| 10100 | 无效的AppID | 将送入的AppID告知项目组，项目组联系业务部门为其增加配置 |
| 10101 | 无效的版本号 | 请按相关接口规范，调整接口版本号 |
| 10102 | 无效的服务名 | [请按相关接口规范，调整接口服务名称](http://168.3.63.26:8888/index.php/ç½ä¸æ¯ä) |
| 10103 | 无效的输出格式 | [请按相关接口规范，调整指定的输出格式](http://168.3.63.26:8888/index.php/ç½ä¸æ¯ä) |
| 10104 | 无效的发起IP | 将IP地址告知项目组，项目组联系业务部门增加参数设置 |
| 10105 | 此项服务未对您的系统开放权限 | 将送入的AppID告知项目组，项目组联系业务部门为其设置服务权限 |

## 数据访问错误(10200-10299)

|  |  |  |
| --- | --- | --- |
| **错误码** | **错误描述** | **错误原因或解决方案** |
| 10200 | 查询日期范围跨度过大，请缩小查询范围。 | 缩小查询日期范围。不同服务有不同的查询日期范围上限，请参见相应的接口说明。 |
| 10201 | 对不起，未查询到符合条件的记录。 | 注：建议对于批量查询，查询结果为空应返回空列表，而不抛异常 |
| 10202 | 对不起，服务异常，请稍后再试。 | 数据库写入失败，请立即联系项目组，排查系统问题 |

## 后端系统交易错误(10300-10399)

|  |  |  |
| --- | --- | --- |
| **错误码** | **错误描述** | **错误原因或解决方案** |
| 10300 | 交易超时，请于30分钟后查询交易结果。 | 系统将对这笔交易进行监控，可过30分钟后再查询这笔交易的结果。 |
| 10301 | 服务异常，错误信息：... | 其他交易异常，建议截图或记录错误信息，并将调用的服务和错误信息反馈给项目组。 |
| 10302 | 服务异常，错误信息：... | 向下游系统发送文件失败，联系项目组排查错误原因 |
| 10303 | 服务异常，错误信息：... | 从下游系统获取文件失败，联系项目组排查错误原因 |

## 商户类错误(20000-20099)

|  |  |  |
| --- | --- | --- |
| **错误码** | **错误描述** | **错误原因或解决方案** |
| 20000 | 对不起，商户不存在。 | 请检查送入的商户号是否正确 |
| 20001 | 对不起，商户无效。 | 请检查送入的商户号是否正确 |
| 20002 | 对不起，商户被冻结，请联系开户行。 | 联系相关业务人员，了解具体情况 |
| 20003 | 对不起，商户配置信息有误，请联系开户行。 | 请检查输入的商户配置信息是否符合接口定义格式规范。 |
| 20004 | 对不起，商户未开通任何支付模式，请联系开户行。 | 联系相关业务人员，了解具体情况 |
| 20005 | 对不起，通讯密钥不符合要求。 | 请按照[[商户接口文档中通讯密钥要求重新输入 |
| 20006 | 对不起，交易来源地址格式必须符合URL格式。 | 请按照交易来源地址格式要求重新输入 |
| 20007 | 对不起，交易来源地址数目已达上限。 | 请先删除一些来源交易地址，再增加 |
| 20008 | 商户重复签约 | 商户已成功签约，请检查输入的商户号。 |
| 20009 | 商户重复解约 | 商户已成功解约，请检查输入的商户号。 |
| 20010 | 对不起，验证项类型有误 | 请联系项目组，检查是否验证项类型有误。 |
| 20011 | 对不起，无法连接商户主机，无法继续支付 | 联系商户，了解其监听地址是否正常工作中 |
| 20012 | 不允许解约 | 商户当前状态不允许解约 |
| 20013 | 商户类型错误 |  |
| 20014 | 商户类型与签约类型不符 | 商户类型与签约类型不符 |
| 20015 | 商户类型与结算模式不符 | 商户类型与结算模式不符 |
| 20016 | 商户类型与手续费扣收模式不符 | 商户类型与手续费扣收模式不符 |
| 20017 | 非开户机构进行商户登记、修改和删除操作 | 非开户机构进行商户登记、修改和删除操作 |

## 订单类错误(20100-20199)

|  |  |  |
| --- | --- | --- |
| **错误码** | **错误描述** | **错误原因或解决方案** |
| 20100 | 对不起，订单不存在(或已过期) | 请检查送入的商户号、订单号是否正确。对于B2B支付，订单若过期将被清理，请重新申请支付。 |
| 20101 | 订单号已存在，无法再次支付 | 请检查送入的商户号、订单号是否正确 |
| 20102 | 对不起，当前订单无法操作，请检查订单状态 | 请检查输入的商户和订单号是否有误 |
| 20103 | 对不起，无符合条件的订单信息 | 与NOT\_FOUND区别在于，NOT\_MATCH是能查到订单，但是不符合客户的过滤条件（如“可退款的”） |
| 20104 | 订单日期超限 | 订单日期超过了允许支付的时间限制 |

## 支付类错误(20200-20299)

|  |  |  |
| --- | --- | --- |
| **错误码** | **错误描述** | **错误原因或解决方案** |
| 20200 | 对不起，支付凭证不存在 | 支付之前，需先生成支付凭证 |
| 20201 | 支付金额与订单金额不匹配 | 请检查支付金额是否与订单金额一致 |
| 20202 | 该商户未开通此类支付模式或系统尚不支持的支付模式 | 请检查商户代号或支付模式是否正确 |
| 20203 | 错误的手续费率或手续费模式 | 请检查商户签约信息中手续费模式是否正确 |
| 20204 | 支付账号类型与支付模式不匹配 | 请检查支付账户类型与订单支付模式是否匹配 |
| 20205 | 校验码不正确 | 请根据商户接口文档中的签名算法检查校验码计算过程 |
| 20206 | 短信口令错误超限 | 短信口令验证错误次数超限，请重新获取新的短信口令 |
| 20207 | 对不起，页面已失效，请从商户网站重新发起支付。 | 会话超时，请从商户网站重新发起支付 |
| 20208 | 支付请求已提交，请勿重复刷新页面，可拨打95561查询支付结果! | 客户重复提交了页面或当前会话已超时。 |
| 20209 | 对不起，您输入的验证码不正确。 | 输入错误的验证码，返回重新输入 |
| 20210 | 对不起，您输入的卡总号不正确。 | 输入错误的卡总号，返回重新输入 |
| 20211 | 对不起，您输入的支付卡和充值手机号非绑定关系。 | 输入的支付卡和充值手机非绑定关系，返回重新输入 |
| 20212 | 生成流水文件失败 | 生成支付流水文件失败，请联系项目组 |
| 20213 | 支付额度超限 | 支付额度超过了用户或商户定义的限额 |
| 20216 | 网上支付签约状态无效 | 网上支付功能被冻结或已关闭 |
| 20218 | 账户余额不足 | 账户活期可用余额不足 |
| 20299 | 系统错误,支付未决 | **后台系统返回超时，订单状态未知,建议商户稍后发起查询确定订单状态。** |

## 对账类错误(20500-20599)

|  |  |  |
| --- | --- | --- |
| **错误码** | **错误描述** | **错误原因或解决方案** |
| 20500 | 对不起，日结回单尚未生成 | 若超时未生成，需检查商户结算账户上资金是否充足。同时，日结回单未生成，则无法生成资金对账文件。 |
| 20501 | 对不起，资金对账文件已超过90天的有效下载期限 | 资金对账文件的有效下载期限为90天，如果超出期限需要获取，需要走数据提取流程。 |
| 20502 | 日结回单汇总信息和对账文件不匹配。 | 该问题一般在生产对账文件时出错，会导致商户该日回单无法生成，需联系项目组。 |
| 20503 | 无法生成资金对账文件（支付明细文件）。 | 系统磁盘空间满或无写入权限导致，需联系项目组。 |

## 账户类错误(20700-20799)

|  |  |  |
| --- | --- | --- |
| **错误码** | **错误描述** | **错误原因或解决方案** |
| 20700 | 对不起，该账户不是有效的兴业借记卡。 | 请检查输入的卡号是否有误 |
| 20701 | 对不起，该账户不是有效的兴业信用卡号。 | 请检查输入的卡号是否有误 |
| 20702 | 对不起，请确认您是否已开通短信口令，或手机号是否有误。 | 借记卡请检查是否已开通短信口令，信用卡请检查客户登记手机号是否有误。 |
| 20703 | 您已开通证书服务，请确认是否已安装证书。 | 检查本机IE环境中证书是否已正常安装。 |
| 20704 | 对不起，您选择的证书有误。 | 检查是否选择了正确的证书。 |
| 20705 | 对不起，您已开通证书服务，无法通过手机支付。 | 受证书保护的账户无法在手机支付网关上支付。 |
| 20706 | 对不起，客户端证书签名验证失败。 | 检查是否选择了正确的证书。 |
| 20707 | 对不起，该账户状态异常。 | 检查卡片的状态是否存在异常。 |
| 20708 | 无法获取账户保护信息 | 检查卡片是否开通短信或证书保护。 |
| 20709 | 该账户不是对公人民币账户 | 检查对公账户是否正确 |
| 20717 | 账户已冻结或开户暂封 | 客户账户异常（如挂失、冻结等） |
| 20718 | 对应卡账户不可进行支付 | 客户账户状态不允许进行网上支付 |

## 文件下载类错误(21000-21099)

|  |  |  |
| --- | --- | --- |
| **错误码** | **错误描述** | **错误原因或解决方案** |
| 21001 | 令牌失效 | 下载令牌已失效，重新发起下载 |
| 21002 | 文件不存在 | 没有找到要下载的文件，请检查输入 |
| 21003 | 任务已在运行 | 已有下载任务在运行中，请稍后发起下载请求 |
| 21004 | 文件压缩失败 | 生成供下载的压缩文件失败，请与项目组联系 |