



兴业银行
INDUSTRIAL BANK CO.,LTD.

收付直通车商户鉴权接口

目录

一	接入准备.....	4
1.1	兴业账号.....	4
1.2	通讯模式.....	4
1.2.1	有用户参与通讯模式.....	4
1.2.2	无用户参与通讯模式.....	5
1.3	接入与测试.....	6
1.3.1	环境地址.....	6
1.3.2	SDK 开发包.....	6
1.4	商户秘钥.....	9
1.4.1	签名/加密算法.....	9
1.4.2	算法代码示例.....	10
1.5	注意事项.....	15
二	独立鉴权.....	15
2.1	无页面独立鉴权接口.....	15
2.1.1	功能概述.....	15
2.1.2	输入参数.....	16
2.1.3	返回结果.....	17
2.1.4	接口示例.....	19
2.2	独立鉴权查询接口.....	20
2.1.5	功能概述.....	20
2.1.6	输入参数.....	20
2.1.7	返回结果.....	21
2.1.8	接口示例.....	23
三	附录.....	24
3.1	错误码说明.....	24
3.1.1	通用错误(10000-10099).....	24
3.1.2	服务分发错误(10100-10199).....	24
3.1.3	数据访问错误(10200-10299).....	24
3.1.4	后端系统交易错误(10300-10399).....	25
3.1.5	商户类错误(20000-20099).....	25
3.1.6	订单类错误(20100-20199).....	26
3.1.7	支付类错误(20200-20299).....	26
3.1.8	对账类错误(20500-20599).....	27

3.1.9	托收交易类错误(20600-20699)	27
3.1.10	账户类错误(20700-20799).....	28
3.1.11	文件下载类错误(21000-21099)	29

一 接入准备

1.1 兴业账号

网上商户只要有兴业账号(企业账号、个人借记卡号),即可和兴业银行签订收付直通车合作协议,在商户网站上实现使用各大银行账户(包括借记卡、信用卡、对公账户)对客户购买的订单进行支付。

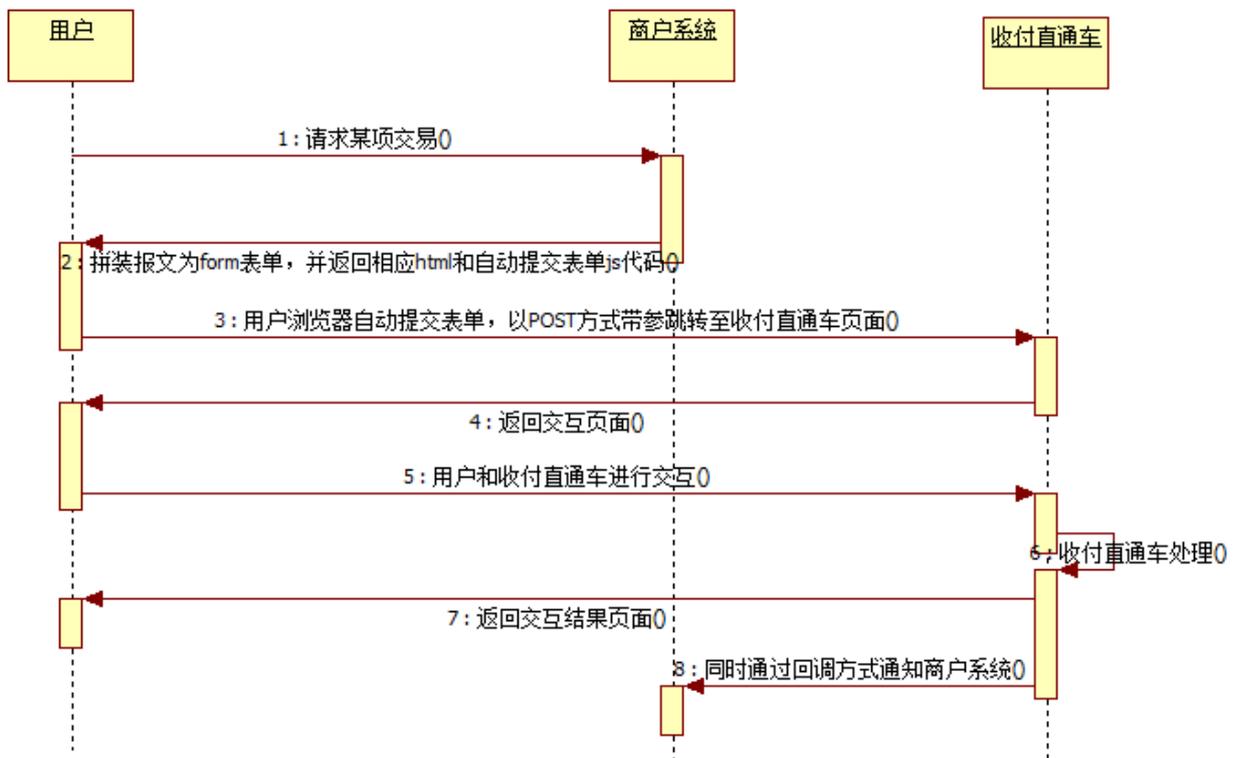
兴业银行收付直通车基于 https 协议,向签约商户提供 WEB 服务。

1.2 通讯模式

收付直通车与商户或用户间的通讯方式均为 HTTPS 协议方式,商户均采用 POST 方法发送数据,收付直通车返回给商户的数据为 JSON 格式字符串。对于下载类业务,业务成功时返回的数据类型为 ZIP 格式压缩包,业务失败时返回的类型为 JSON 格式字符串,可以根据 HTTPS 头部的 Content-Type 判断。根据是否需要用户参与交互,分为两种通讯模式:有用户参与通讯模式和无用户参与通讯模式。

1.2.1 有用户参与通讯模式

当需要用户与收付直通车进行交互时,使用这种通讯方式。商户系统将报文数据拼装好后,发送至用户浏览器,用户浏览器将商户拼装的数据以 POST 方式(即提交 method=post 类型的 form 表单)跳转至收付直通车页面进行继续操作。流程图如下:



第 2 步中, html 和 js 代码示例(以网关支付为例):

1.3 接入与测试

1.3.1 环境地址

支付类型	环境类型	URL
快捷支付	生产环境	https://pay.cib.com.cn/acquire/easypay.do
	测试环境	https:// 220.250.30.210:37031/acquire/easypay.do
网关支付	生产环境	https://pay.cib.com.cn/acquire/cashier.do
	测试环境	https:// 220.250.30.210:37031/acquire/cashier.do
托收支付	生产环境	https://pay.cib.com.cn/acquire/easypay.do
	测试环境	https:// 220.250.30.210:37031/acquire/easypay.do
代付地址	生产环境	https://pay.cib.com.cn/payment/api
	测试环境	https:// 220.250.30.210:37031/payment/api

注意事项:

1. 商户秘钥可通过兴业银行收付直通车商户平台设置，**请勿与测试环境密钥一致。**
2. 商户秘钥可设置为 8-32 位，建议在 10 位以上。
3. 生产上建议定期更换商户秘钥，例如一年一换。
4. 测试环境应监管要求网络互访需要对接运营人员开通网络白名单。
5. 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。

1.3.2 SDK 开发包

收付直通车目前提供 JAVA 和 PHP 语言两种 SDK 开发包，商户可以参考 SDK 开发包进行开发联调，以加快开发效率。商户也可以根据需求直接修改 SDK 中的源代码以符合自己系统的用途。

需要特别注意的是，SDK 提供的接口对传入参数没有作检查和过滤，商户在传入参数前，需要自行对传入的参数进行检查和过滤。特别是生成跳转页面代码的接口中（如网关支付接口等），请务必对传入参数进行安全性检查，防止出现 XSS 攻击等安全问题。

SDK 相关工程由对接运营人员提供。

1.3.2.1 JAVA

JAVA 语言 SDK 开发包为 J2EE 工程，支持 JAVA 版本 1.6+，可以使用 Eclipse 直接导入工程使用。建议使用前先阅读 SDK 包中 Readme.txt 文件。

调用 API 的示例，详细示例请参见 Example 类中说明：

```
//设置参数
Configure.setAppid("A0000093"); //商户 ID, appid
Configure.setCommKey("85EDE15FE3654471B3117D9F2BC9F3D4"); //商户秘钥
```

```

Configure.setSub_mrch("SDK 测试商城");           //二级商户名称
Configure.setNeedChkSign(true);                 //应答是否验签，开发时可以设成 false
进行调试

// 调用快捷支付 API
String result_json = EPay.epPay("DD20150825KJ", "100.00", "示例订单标题", "示例订
单内容", "6222801234567888953");

```

其中所有对外交易接口均在 `com.cib.epay.sdk.EPay` 类中。

签名和验签接口在 `com.cib.epay.sdk.util.Signature` 类中。

`epay_notify.jsp` 页面为商户回调示例页面，商户需要添加自己的业务逻辑处理代码。

`epay_redirect.jsp` 该页面为跳转接口示例，即使用[有用户参与通讯模式](#)的示例。仅供参考，商户可以根据需要自行实现。同时该页面也需要加上商户自己的业务逻辑，如订单状态变更为支付中等。

所有示例在 `com.cib.epay.sdk.Example` 类中，该类请不要包含在商户工程中。

`example.jsp` 为需要用户交互的接口示例，商户可以在浏览器中打开该页面查看示例。同样，该文件不要包含在商户工程中。

需要注意的是：

1. 该工程为 UTF-8 编码，如果商户系统不是该编码，需要自行进行修改或编码转换；
2. 工程中使用的 HTTPS 通讯方式为 `HttpsURLConnection`，如果商户想使用其它方式，如 `HttpClient` 等，可以自行实现一个 `IRequestService` 接口，并在 `Configure` 类中修改 `httpsRequestClassName` 变量为你实现的类的全名；
3. 调用任何 API 之前，请先设置 `Configure` 参数类，具体设置方法可以参考 `Example` 类中的示例；
4. `com.cib.epay.sdk.Example` 类和 `example.jsp` 页面为示例，请不要包含在商户系统中。
5. 日期时间必须为北京时间，注意操作系统中的时区和时间设置。如果收付直通车提示“订单已过期”等类似提示，请检查系统时间是否正确。
6. 若提示`{"errcode":"EPAY_29098","errmsg":"[EPAY_29098]应答消息验签失败，交易未决"}`，可在开发调试时设置 `Configure.setNeedChkSign(false)`，不验证应答签名，待调试正常后再改回 `true`。生产上建议设置为 `true`。

1.3.2.2 PHP

PHP 版本 SDK 支持 PHP 版本 5.3+，可以使用 Eclipse for PHP 直接导入工程中。建议使用前阅读 SDK 包中 `Readme.txt` 文件。

调用 API 的示例，详细示例请参见 `example.php` 中说明：

```

require_once('lib/epay_core.class.php');
$epay = new EPay($epay_config); //使用参数实例化 EPay 类
$result_json = $epay -> epPay("DD20150825KJ", "100.00", "示例订单标题", "示例订
单内容", "6222801234567888953"); //调用快捷支付 API

```

包含的文件如下：

1. `epay.config.php`: 参数示例。PHP 版本 SDK 中使用的参数为一个数组，在实例化接口类时，需

要传入该数组。具体字段请打开该文件查看。

2. `lib/epay_core.class.php`: 收付直通车全部接口类，实例化该类时请传入参数数组，示例参数数组在 `epay.config.php` 文件中。
3. `lib/epay_util.class.php`: 工具类，`epay_core.class.php` 中包含该文件。
4. `lib/ca-bundle.crt`: 可信 CA 证书。商户可以自行互联网上下载该证书列表。也可以直接使用该文件。
5. `epay_redirect.php`: 该页面为跳转接口示例，即使用[有用户参与通讯模式](#)的示例。仅供参考，商户可以根据需要自行实现。同时该页面也需要加上商户自己的业务逻辑，如订单状态变更为支付中等。
6. `epay_notify.php`: 该页面为商户回调示例页面，商户需要添加自己的业务逻辑处理代码。
7. `example.php`: 所有 API 调用示例，请直接打开该文件源代码进行查看。该文件请不要包含在商户工程中。
8. `index.php`: 需要用户交互的接口示例，商户可以在浏览器中打开该页面查看示例。同样，该文件不要包含在商户工程中。

需要注意的是：

1. 本 SDK 需要使用 `php_curl` 扩展作为 `http` 通讯使用，因此，请确保商户系统运行的 PHP 环境中包含 `php_curl` 扩展。Windows 下可以去掉 `php.ini` 中 `extension=php_curl.dll` 这行前面的注释并重启服务，即可启用 `php_curl` 扩展。
2. 若你的环境无法启用 `php_curl` 扩展，需要自行修改 `epay_util.class.php` 文件中的 `getHttpPostResponse` 方法。
3. `epay_util.class.php` 文件中上方 4 个常量为客户端失败时（如网络异常等），返回的错误提示，非服务端返回内容。因此，你可以自己修改为需要的内容，也可以直接使用原有文件中的内容。
4. 调试时，可以去掉 `epay_util.class.php` 文件 `getHttpPostResponse` 方法中 `var_dump(curl_error($curl));`和 `var_dump(curl_getinfo($curl));`两行前面的注释，来显示 `curl` 通讯过程中的信息。
5. `epay.config.php` 中的参数为示例参数，商户需要修改为自己的配置。商户系统可以根据需要在其它地方初始化这些参数，只要在实例化 `EPay` 类时，传入配置参数数组即可。
6. 日期时间必须为北京时间，注意操作系统或 `php.ini` 中的时区和时间设置。如果收付直通车提示“订单已过期”等类似提示，请检查系统时间是否正确。
7. 该工程为 UTF-8 编码，如果你的系统不是该编码，你需要自行修改或进行编码转换。
8. 若提示 `{"errcode":"EPAY_29098","errmsg":"[EPAY_29098]应答消息验签失败，交易未决"}`，可在开发调试时设置 `$epay_config['needChkSign'] = false`，不验证应答签名，待调试正常后再改回 `true`。

生产上建议设置为 true。

1.4 商户密钥

代付部分使用商户密钥包含两种：用于 MAC 校验算法使用的商户密钥，一般为 32 位十六进制字符串，如下所示：**8D0565EF811F4C61806A6301C9DF8726**

另外一种，用于 sha1WithRSAEncryption 签名算法使用的客户证书文件，一般为 pfx 文件。

注意事项：

1. 为确保订单信息真实有效不被篡改，每个商户均应在兴业银行收付直通车商户平台中设定商户商户密钥。
2. 商户密钥作为商户与银行间的信任关键，双方都应对其保密。
3. 商户密钥内容只能由数字和字母组成，其中字母区分大小写。
4. 为避免商户密钥泄漏，生产环境上请勿使用与测试环境相同的商户密钥。
5. 商户密钥应定期更换，至少应每年更换一次。
6. 对于客户证书文件，收付直通车服务器上没有保存副本，商户若遗失，只能进行重置操作。

1.4.1 签名/加密算法

目前收付直通车代付部分使用两种算法：SHA1WithRSAEncryption 签名、MAC 校验（SHA1）；鉴于安全考虑，后续将逐步淘汰掉 SHA1-MAC 校验方法。

1.4.1.1 SHA1 方式 MAC 校验

该算法适用于 sign_type=SHA1 时计算 mac 参数的值。以以下待发送参数，商户密钥 84A8251FECD84E599B7B1037579E6A4D 为例，具体说明计算步骤：

```
timestamp=20150708111509
appid=P0000014
service=cib.epay.payment.receiptFile
ver=01
trans_date=20150623
rcpt_type=1
```

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&上商户密钥。

此步需要注意的是：

编码需要使用 UTF-8；

参数中不能出现类似&等特殊字符

此时不需要对参数键、值等进行 URLEncode 操作。

```
appid=P0000014&rcpt_type=1&service=cib.epay.payment.receiptFile&timestamp=20150708111509&trans_date=20150623&ver=01&84A8251FECD84E599B7B1037579E6A4D
```

2. 计算 SHA-1 值

使用 hash 算法 SHA-1 计算上述字符串的 hash 值，得到 16 进制形式的字符串（共 40 个字母），并全部转换为英文字母大写形式。如:30F2B0CA895BB25881CEE20CB32E5EEADCB9E18D

3. 上步得到的字符串即作为发送参数中 mac 参数的值。

注意：若参数为空，可不发送（不加入到拼装字符串中），也无需加入计算 mac；若要发送，相应的也需加入去计算 mac。

1.4.1.2 SHA1WithRSAEncryption 签名验签

该算法适用于 sign_type=RSA 时计算 mac 参数的值。

具体做法可以参考示例代码或 SDK 中相应代码：

1. 参考 1.5.1.1 中第 1 步，将参数以 key=value 形式组装成一个数组，将该数据进行排序。注意，此时不需要再&上商户密钥。注意编码必须为 UTF-8。
2. 从客户证书文件中获取商户 RSA 私钥。行方提供的 PFX 文件默认密码为 123456，别名(alias)为 appsvr_client。
3. 使用标准的 SHA1WithRSAEncryption 签名算法对该字符串进行签名，得到的值（base64 格式字符串）即为 mac 参数的值。

1.4.2 算法代码示例

1.4.2.1 SHA1 示例

【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
StringBuilder sb = new StringBuilder();
sb.append(paramList.get(0));
for (int i = 1; i < paramList.size(); i++) {
    sb.append("&").append(paramList.get(i));
}
return sb.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) {
    StringBuffer sb = new StringBuffer(bytes.length * 2);
    for (int i = 0; i < bytes.length; i++) {
        sb.append(hexChar[(bytes[i] & 0xf0) >>> 4]);
        sb.append(hexChar[bytes[i] & 0x0f]);
    }
    return sb.toString();
}

private static char[] hexChar = { '0', '1', '2', '3', '4', '5', '6', '7', '8',
    '9', 'A', 'B', 'C', 'D', 'E', 'F' };

/**
 * 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"));
        return bytesToHexStr(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;
}
```

【.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
    ArrayList paramsList = 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
    string sb="";
    for (int i = 0; i < paramsList.Count; i++){
        sb += paramsList[i].ToString() + "&";
    }
    sb = sb.TrimEnd('&');
    return sb;
}

/**
 * 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();
        StringBuilder sb = new StringBuilder();
        for (int i = 0; i < str2.Length; i++) {
            sb.Append(str2[i].ToString("X2"));
        }
        return sb.ToString();
    } catch{
        //log.error("无法加载 SHA 算法。", e);
    }
}
```

```
        throw;
    }
}
```

【PHP 示例】

```
/**
 * 根据参数数组生成 MAC 码。
 * @return MAC 码
 */
public static function Signature($param_array, $commkey) {
    ksort($param_array);
    reset($param_array);
    $signstr = '';
    foreach ($param_array as $k => $v) {
        if(strcasecmp($k, 'mac') == 0) continue;
        $signstr .= "{$k}={$v}&";
    }
    $signstr .= $commkey;
    return strtoupper(sha1($signstr));
}
```

1.4.2.2 SHA1WithRSAEncryption 示例

【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
    StringBuilder sb = new StringBuilder();
    sb.append(paramList.get(0));
    for (int i = 1; i < paramList.size(); i++) {
        sb.append("&").append(paramList.get(i));
    }
}
```

```

    }
    return sb.toString();
}

/**
 * 生成签名 MAC 字符串
 * @param      参数列表（包含 mac 参数）
 * @param      证书路径（带文件名）
 * @param      证书密码
 * @return     MAC 字符串
 */
public static String generateMAC(Map<String, String> params, String certPath,
String certPwd) {
    try {
        PrivateKey privKey = readPrivateKey(certPath, certPwd);
        java.security.Signature signature =
        java.security.Signature.getInstance("SHA1WithRSA");
        signature.initSign(privKey);
        signature.update(generateParamStr(params).getBytes());
        byte[] signed = signature.sign();

        // 计算 base64encode(signed), 无换行。如无法使用, 请自行替换为其它
        BASE64 类库。
        @SuppressWarnings("restriction")
        String mac = new sun.misc.BASE64Encoder().encode(signed)
        .replaceAll(System.getProperty("line.separator"), "");
        return mac;
    } catch (Exception e) {
        // 读取证书等出错
        return "SIGNATURE_RSA_CERT_ERROR";
    }
}
}

```

【PHP 示例】

```

/**
 * 根据参数数组生成 MAC 码（sha1WithRSAEncryption 方式）。
 * @return MAC 码
 */
public static function Signature($param_array, $commkey) {
    ksort($param_array);
    reset($param_array);
    $signstr = '';
    foreach ($param_array as $k => $v) {
        if(strcasecmp($k, 'mac') == 0) continue;
        $signstr .= "{$k}={$v}&";
    }
    $signstr = substr($signstr, 0, strlen($signstr) - 1);
    if (false !== ($keystore = file_get_contents($cert)) &&
        openssl_pkcs12_read($keystore, $cert_info, $cert_pwd) &&
        openssl_sign($signstr, $sign, $cert_info['pkey'],
        'sha1WithRSAEncryption')) {
        return base64_encode($sign);
    } else{
        return 'SIGNATURE_RSA_CERT_ERROR';
    }
}
}

```

1.5 注意事项

- 1 输入数据都通过 key-value 的参数列表形式，使用 http post 提交到服务接入地址。
- 2 数据中不能包含"&"、"="、"?"等特殊字符。
- 3 通讯报文使用 UTF-8 编码。
- 4 商户提交数据中的空格将被认为是有效字符被接收，请提交时注意对多余空格的控制。
- 5 测试环境中快捷认证的手机号请输入格式正确的手机号，测试环境并不会真正的发送短信验证码，短信验证码请输入任意 6 位数字即可。
- 6 测试环境中，根据邮路不同，快捷支付返回的支付结果可能是随机结果，如果需要成功支付的结果，可以尝试支付多次。
- 7 快捷支付身份认证，每张银行卡每天连续认证失败 6 次后当天将无法继续进行快捷支付身份认证交易，请第二天再尝试或更换银行卡。
- 8 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。
- 9 鉴于安全考虑，后续将逐步淘汰掉 SHA1-MAC 校验方法。
- 10 判断交易是否正确，应以正常返回交易状态（trans_status）进行判断；除非特别说明，否则异常返回都应当成“支付中”、“处理中”的中间状态来进行处理。

二 独立鉴权

2.1 无页面独立鉴权接口

2.1.1 功能概述

服务名称	cib.epay.acquire.singleauth.quickSingleAuth
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
SDK 示例	暂无
服务用途	发起独立鉴权交易，无需用户与收付直通车交互。
接口返回	JSON 格式字符串
回调方式	支持后台回调，详见 账户认证通知

方向：商户→银行。

输入系统跟踪号、银行卡号、发卡行银行代码、姓名、证件类型、证件号码、银行预留手机号码申请进行账户认证，若是信用卡认证需增加输入信用卡的有效期、信用卡背面末三位安全码。

2.1.2 输入参数

参数名称	类型	必输	示例值	描述
timestamp	char(14)	√	20171124103640	时间戳。格式: yyyyMMddHHmmss
appid	char(8)	√	Q0001299	兴业银行分配给商户的 AppID
service	varchar(100)	√	cib.epay.acquire.singleauth.quickSingleAuth	服务名称
ver	char(2)	√	01	接口版本号，固定 01。
以下为账户信息				
trac_no	varchar(32)	√	20171124103640445	系统跟踪号
card_no	varchar(32)	√	6222801234567888953	卡号
bank_no	varchar(14)	√	105100000017	银行代码（测试时请从下面地址下载获取相应银行的银行代码： http://220.250.30.210:7052/cibhall/images/bank.unl ）
acct_type	char(1)	√	0	银行账户类型：0-储蓄卡;1-信用卡
user_name	char(60)	√	李四	姓名
cert_type	char(1)	√	0	证件类型 0-身份证(目前仅支持身份证)
cert_no	char(20)	√	340102198212062039	证件号
card_phone	char(11)	√	13800000004	银行预留手机号码
expireDate	char(4)		1019	信用卡有效期(信用卡认证必填)
cvn	char(3)		480	信用卡背面末三位安全码(信用卡认证必填)
以下为签名相关信息				
mac	varchar(344)	√	Q2mfo/T/CdqZPXxVmHJ5cc1aOF/xY7+OMUI5A5kfCPrUraKs9aBW552uJsA1TrwZrr5QEjuo+oP1xzjGTLA	消息校验码

			LNJ7x1c4DB8KwP0 AMI+ENM4IST5gfWq urlF5WCzHVBhDzf3 5HDm4Yd8tw6++oD 6vqOPVFhjbZAIUVQ RTJiJCJAK4eYmrHT Jk/UI8II7kgH7ETr56 U7vnyVJMP49Ntuxa JfINi0Uj6GlftrTvMjN6 EFZbjlvxrhSccAyMP OcnIXNrJ2JH5p7jiqN 76ZyExryYDfZA56Jc Sx3c4KgfdRQ0FU9t LxjMh/senxPvbUcRa 201eNb43tffovKQHkl VsRD9z0w==	
--	--	--	---	--

2.1.3 返回结果

请求超时当作认证失败处理；成功返回，根据返回接口中的状态判断认证状态；异常返回，20404-账户绑定失败。支持后台回调，通知地址为贵司提供的商户回调 URL，详见[账户认证通知](#)。

【成功结果】

参数名称	类型	必输	示例值	描述
appid	char(8)	√	Q0001299	兴业银行分配给商户的 AppID
trac_no	varchar(32)	√	20171124103640445	下游系统跟踪号
auth_status	char(1)	√	1	0-待认证；1-认证成功；2-认证失败；3-认证处理中
remark	varchar(255)		账户绑定失败	错误信息
sno	char(16)		2017112404708593	收付直通车系统流水号
sign_type	varchar(4)	√	RSA	签名算法
mac	varchar(344)	√	FBcITSuCl2YgrdeN78jHbey DaGZ3R4pL29aVU+FbBmj	应答签名/消息校验码

			dyDtKgAkdn0Uw5Zrj3Nn1s s7aXcwfndvUFavR4IHmELi XsOXHo1hNWHH06pvav1a 6aDmPn/9zcNZamImATiuV ywTe0XOyJjQ1XI4KN1YCP vL6U1luUKBcVfKlPnvUqYy keQX/5H4V86gyEq+AKbjJ NMgUrmfN9meAS2Sh7E2F PQgnZB1MAkygldGmPFB XCv8UnUeNaVA3BONOjDI 2x26sabs7Y1N6q0Mzg2mi Y6ScD2HuaGVNPvvhSR1/ FY97EyKVww/3D6sBYaN75 UvR5UeyJqixUDkJJe8Tkl+f+ +jxk5Q==	
--	--	--	--	--

【失败结果】

参数名称	类型	必输	示例值	描述
errcode	varchar(10)	√	EPAY_20404	错误代码
errmsg	varchar(255)	√	账户绑定失败	发送错误时返回的错误信息
sign_type	varchar(4)	√	RSA	签名算法
mac	varchar(344)	√	mcQZfKHD6fRM6f3Ro0 LdIXBTokcJTX2hhVArMI Jryr9rA3c8327yjHZAeDp oOyNmqnCOsGpErY31 K82Or3QLZINWbwXAzz 80bZ11XAALjRZW1LQp O/3WiC3Me+JlhvTljode EF7FvYOBBqKubnamvX XZ8j8QUNWndT9zyfA+J utaeFoy4inJSzgQ+jWSy pX90ldL7L97exwy+Nm/r 5UopWV3jDOrUjyqrG/qL Rh32BJHBow5rDiw4MS	应答签名/消息校验码

			vYwbPyZhAGiunCWk0J 5GJBCgyaxcZx7dojxgNf v4hd+doudRm6gS3YAP u68l00ImYFxyPwZ7hV pHEAKsI4NQUse4B1uz zA==	
--	--	--	--	--

2.1.4 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。

```
appid=Q0001299&service=cib.epay.acquire.singleauth.quickSingleAuth&ver=01&trac_no=20171124103640445&card_no=6222801234567888953&user_name=%E6%9D%8E%E5%9B%9B&cert_type=0&cert_no=340102198212062039&card_phone=13800000004&bank_no=105100000017&sign_type=RSA&timestamp=20171124103640&mac=Q2mfo%2FT%2FCdqZPXxVmHJ5cc1a0F%2Fxy7%2BOMU15A5kfCPrUraKs9aBW552uJsA1TrwZrr5QEjuo%2BoP1xzjGTLALNJ7x1c4DB8KwP0AM1%2BENM41ST5gfWqurlF5WCzHVBhdzF35HDM4Yd8tw6%2B%2BoD6vqOPVFhjbZAlUVQRTJiJCJA4eYmrHTJk%2FUI8II7kgH7ETr56U7vnyVJMP49NtuxaJf1Ni0Uj6G1ftrTvMjN6EFZbjlvxrhSccAyMPOcnIXNrJ2JH5p7jiqN76ZyExryYDfZA56JcSx3c4KgfdRQ0FU9tLxjMh%2FsenxPvbUcRa201eNb43tffovKQHKlVSRD9z0w%3D%3D&acct_type=0
```

输出示例 (JSON)

【业务成功结果】

```
{"trac_no": "20171124103640445", "sno": "2017112404708593", "appid": "Q0001299", "auth_status": "1", "sign_type": "RSA", "mac": "FBcITSuCl2YgrdeN78jHbeyDaGZ3R4pL29aVU+FbBmjdyDtKgAkdn0Uw5Zrj3Nn1ss7aXcwfndvUFavR4IHmELiXsOXHo1hNWHH06pvav1a6aDmPn/9zcNZamImAtiuVywTe0X0yJjQ1Xl4KN1YCPvL6U1luUKBcVfKlPnvUqYykeQX/5H4V86gyEq+AKbjJNMgUrmfN9meAS2Sh7E2FPQgnZB1MAkygIdGmPFBXcv8UnUeNaVA3BON0jD12x26sabb7Y1N6q0Mzg2miY6ScD2HuaGVNPvvhSR1/FY97EyKVvw/3D6sBYaN75UvR5UeyJqixUDkJe8Tk1+f++jxk5Q=="}
```

【业务失败结果】

```
{"errcode": "EPAY_20404", "errmsg": " 账 户 绑 定 失 败", "sign_type": "RSA", "mac": "mcQZfKHD6fRM6f3Ro0Ld1XBTokcJTX2hhVArMIJryr9rA3c8327yJHZAEdp0yNmqnC0sGpErY31K820r3QLZINWbWXAzz80bZ11XAALjRZW1LQp0/3WiC3Me+JlhvTIjodeEF7FvY0BbqKubnamvXXZ8j8QUNWndT9zyfA+JutaeFoy4inJSzgQ+jWSypX90IdL7L97exwy+Nm/r5UopWV3jDOrUjyqrG/qLRh32BJHBow5rDiw4MSvYwbPyZhAGiunCWk0J5GJBCgyaxcZx7dojxgNfv4hd+doudRm6gS3YAPu68l00ImYFxyPwZ7hVpHEAKsI4NQUse4B1uzzA=="}
```

【异常失败结果】

```
{"errcode": "EPAY_20307", "errmsg": " 未 签 署 商 户 合 约 或 合 约 无 效", "sign_type": "RSA", "mac": "fGdrNMhyh1yvGej3Fze30TRE1IFbSfixZkVeknLrPPP/MOTdeR393RYRixIEu1qIP8o40MAswNuDpW/k1qREwa3L9aCiC0uZ0QjgX/V7JLLQHSzbenZJYHBe0trgCKCUyP0ld0IHIO+Oijyoelz46gRBMmPSAGGortAUG5m39btEQJXUdrsv3pj1D1dBd1pRDVKco5nqMZpJxMN1Iov97aA9qZzHfmXs2SHFWNT0UQwTg+xFWUvK1oQhs1p9NwjX265fIHlrQuKPAmr+V8D3rD/+yOeZ YJQhoHAhUJw0uyvtRySyBXlgK/46AKiuIvExwvWU6uRd+00Wc7BMikwrQ=="}
```

2.2 独立鉴权查询接口

2.1.5 功能概述

服务名称	cib.epay.acquire.singleauth.query
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
SDK 示例	暂无
服务用途	发起独立鉴权查询交易，无需用户与收付直通车交互。
接口返回	JSON 格式字符串
回调方式	无

方向：商户→银行。

输入系统跟踪号进行认证查询。

2.1.6 输入参数

参数名称	类型	必输	示例值	描述
timestamp	char(14)	√	20171124111941	时间戳。格式：yyyyMMddHHmmss
appid	char(8)	√	Q0001299	兴业银行分配给商户的 AppID
service	varchar(100)	√	cib.epay.acquire.singleauth.query	服务名称
ver	char(2)	√	01	接口版本号，固定 01。
sign_type	varchar(4)		RSA	签名算法
以下为账户信息				
trac_no	varchar(32)	√	20171124103640445	下游系统跟踪号
以下为签名相关信息				
mac	varchar(344)	√	MZ2bkLwonhP1pwc CXCGH75rKnLYEW hhipDIgRP8t3M+uXp Npmrwx9YKSv+30 UdU5x+46sAoCJAc1 GJhDFDMYZ8IZGiA 7Gu/lux91YDaZOBu 5X/reLle+a6OyfJCUg CqwXZN11J8mRkpm bXW37dYnF8TipRV	消息校验码

			MYB1ehBQhno1mk7 EV7VugogwhKr6iCW d5x4xHnBD4IHFswN PPVKJj6XM4wEAI2 cJiEINZQ6ULkyq5x5 bNuuedksya90CmhS 2tU3NDh1nFP8uY0T DBqC/s1KRLwJvIO0 WFqQrL9S4Ryf17u mqC9MHSZOomb0e p1XqXGBoHVOrysX w54bBeekVCFQ==	
--	--	--	---	--

2.1.7 返回结果

请求超时当作认证未决处理；成功返回，根据返回接口中的状态判断认证状态；异常返回，10201-不存在认证记录，可以当做失败，其他不能当做失败。

【成功结果】

参数名称	类型	必输	示例值	描述
appid	char(8)	√	Q0001299	兴业银行分配给商户的AppID
trac_no	varchar(32)	√	20171124103640445	下游系统跟踪号
auth_status	Char(1)	√	2	0-待认证；1-认证成功；2-认证失败；3-认证处理中
remark	varchar(255)		账户绑定失败	错误信息
sno	char(16)		2017112404708593	收付直通车系统流水号
sign_type	varchar(4)	√	RSA	签名算法
mac	varchar(344)	√	FBcITSuCl2YgrdeN78jHbey DaGZ3R4pL29aVU+FbBmj dyDtKgAkdn0Uw5Zrj3Nn1s s7aXcwfndvUFavR4IHmELi XsOXHo1hNWHH06pvav1a 6aDmPn/9zcNZamImATiuV ywTe0XOyJjQ1XI4KN1YCP	消息校验码

			vL6U1luUKBcVfKIpnvUqYy keQX/5H4V86gyEq+AKbjJ NMgUrmfN9meAS2Sh7E2F PQgnZB1MAkygldGmPFB XCv8UnUeNaVA3BONOjDI 2x26sabox7Y1N6q0Mzg2mi Y6ScD2HuaGVNPvvhSR1/ FY97EyKVvw/3D6sBYaN75 UvR5UeyJqixUDkJe8Tkl+f+ +jxk5Q==	
--	--	--	--	--

【失败结果】

参数名称	类型	必输	示例值	描述
errcode	varchar(10)	√	EPAY_10201	错误代码
errmsg	varchar(255)	√	不存在认证记录!	发送错误时返回的错误信息
sign_type	varchar(4)	√	RSA	签名算法
mac	varchar(344)	√	gK1OHHYviuOE7JNDIx moRk+AdBvxGnjO8kB1 yujxbZraP4r07ppyTzT2n ZT0G7MYAdmiZXbvgw GGE8k4GQfOIWOa+eV 2vOuRzjkiqTioFd3nXLkt nhlg7Oos8JITE1nsHAQ tsQcyD1pir4w74l+s36Gx 6Htuxb18z05wDaBRbS NzEJzOmiZrznEvA0y5A H8IXLNA5MtpHymxtbN MeUerAygK3gkbl2Yuuf3 HpK5+R1YmzAJT00JkD TPkml3JmGQHL06+40g F3ZCAA4bYK1WbSvFgl x6/GI7GG13RbwOoLI+X u7yx6m4MXGx53olgj2+ +Gz2us6X8YCCvu+9Lo	消息校验码

		ZmSOQ==	
--	--	---------	--

2.1.8 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。

```
appid=Q0001299&service=cib.epay.acquire.singleauth.query&ver=01&trac_no=20171124103640445&sign_type=RSA&timestamp=20171124111941&mac=MZ2bkLwonhP1pwcCXCXGH75rKnLYEWhhipDIgRP8t3M%2BuXpNpmrwxS9YKSv%2B30UdU5x%2B46sAoCJAc1GJhDFDMYZ8lZGiA7Gu%2FIux91YDaZ0Bu5X%2FreLIe%2Ba60yfJCUgCqwXZN11J8mRkpmbXW37dYnF8TipRVMYB1ehBQhno1mk7EV7VugogwhKr6iCwd5x4xHnBD4lHFswNPPVKJj6XM4wEAlS2cJiElNZQ6ULkyq5x5bNuuedksya90CmhS2tU3NDh1nFP8uY0TDBqC%2Fs1KRLw1JvI00WFqQrL9S4Ryf17umqC9MHSZ0omb0ep1XqXGBoHV0rysXw54bBeekVCFQ%3D%3D
```

输出示例 (JSON)

【业务成功结果】

```
{"trac_no":"20171124103640445","sno":"2017112404708593","appid":"Q0001299","auth_status":"1","sign_type":"RSA","mac":"FBcITSuCl2YgrdeN78jHbeyDaGZ3R4pL29aVU+FbBmjdyDtKgAkdn0Uw5Zrj3Nn1ss7aXcwfndvUFavR4IHmELiXsOXHo1hNWHH06pvav1a6aDmPn/9zcNZamImATiuVywTe0X0yJjQ1Xl4KN1YCPvL6U1luUKBcVfKIpnvUqYykeQX/5H4V86gyEq+AKbjJNMgUrmfN9meAS2Sh7E2FPQgnZB1MAkygIdGmPFBXCv8UnUeNaVA3BONOjDl2x26sax7Y1N6q0Mzg2miY6ScD2HuaGVNPvvhSR1/FY97EyKVvw/3D6sBYaN75UvR5UeyJqixUDkJe8Tk1+f++jxk5Q=="}
```

【业务失败结果】

```
{"trac_no":"20171130165421956","sno":"2017113005653306","appid":"Q0001299","remark":"          账          户          绑          定          失          败","auth_status":"2","sign_type":"RSA","mac":"ayQGemSwwD7pAtGe7YdiRxfai74L6GYQSad50GzW+Zn7DlrZki8aKSoxX8DVAPSjr/gBacd1LKh1gbGC5DabcSpS0e95eH/Ak4bKVQyQUIw8J6KcN8LmZgiUlaeCHA6Z9gNfFD5soFpWHS1v8xjyyEYKDikk+U3jdYYvIY5ZV5SL2ReArIkgy4rQx1aPcPLFi7YorMI58mS7QeYMdmC70HaRRStVuWHrdN/Ey2PZPOshojRudUfGN8H770LFiQIEIGSNkpkRjp3Uo/+5JtIM5rMf8YxNSNS5PAgCBYUaaG4j4XVRjjsfaMAMx/ewIsp0eRvbArDAFw/bffjVnXyc6Q=="}
```

【异常失败结果】

```
{"errcode":"EPAY_10201","errmsg":"          不          存          在          认          证          记          录          !","sign_type":"RSA","mac":"gK10HHYvuiOE7JNDIxmoRk+AdBvxGnj08kB1yujxbZraP4r07pyTzT2nZT0G7MYAdmiZXbvgwGGE8k4GQf0iW0a+eV2vOuRzjkiqTioFd3nXLktnhlg700s8JITE1nsHAQtsQcyD1pir4w74I+s36Gx6Htuxb18z05wDaBRbSNzEJz0miZrznEvA0y5AH8IXLNA5MtpHymxtbNMeUerAygK3gkb12Yuuf3HpK5+R1YmzAJT00JkDTPkm13JmGQHL06+40gF3ZCAA4bYK1WbSvFglx6/GI7GG13RbwOoLl+Xu7yx6m4MXGx53o1gj2++Gz2us6X8YccVu+9LoZmSOQ=="}
```

三 附录

3.1 错误码说明

3.1.1 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

3.1.2 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
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
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

3.1.3 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

3.1.4 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

3.1.5 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
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

3.1.6 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

3.1.7 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。

3.1.8 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。

3.1.9 托收交易类错误(20600-20699)

错误码	错误描述	错误原因或解决方案
20601	托收支付成功	托收订单成功
20602	托收订单失败	托收订单直接失败
20603	托收订单处理中	托收订单处理中，待查询
20605	托收认证失败	托收认证失败，检查托收认证要素是否正确，再重新进行认证
20606	失败次数超限	同张银行卡，交易支付失败超过次数限制。不允许再进行交易
20610	可用余额不足	账户活期可用余额不足
20611	原始金额不正确	金额输入不正确
20612	金额超过单笔限额	金额超过账户单笔交易限额
20613	超出交易次数限制	超过银行卡交易次数
20614	金额超过单日累计限额	交易金额累计超过银行卡当日累计限额
20615	余额不足次数超限	交易金额不足或者交易次数超过限制
20616	金额低于最小限额	交易金额低于最小金额限制
20620	输入有误	输入的交易信息有误
20621	没有交易权限	没有代扣权限，需要先添加托收白名单
20622	无效日期	输入日期有误
20623	无效日期范围	输入的范围日期有误
20624	无效证件类型	证件类型无效
20640	无此账户	账户不存在
20641	账号户名不符	账户和户名不符
20642	该账户已被冻结	账户已被冻结
20643	账户已挂失	账户已挂失
20644	账户已注销	账户已注销

20645	账户状态不正确	账户状态不正确，无法进行交易
20646	证件号码不符	传入的证件号有误
20647	手机号码不符	传入的手机号有误
20651	参数配置获取异常	获取渠道配置参数异常
20656	外部系统处理失败	渠道交易处理错误
20658	银行卡信息有误	传入的银行卡信息有误
20659	身份信息或手机号有误	身份信息或手机号有误
20660	暂不支持该卡或卡号输入有误	该银行卡不支持交易或者卡号有误
20661	银行卡过期或有效期输入有误	银行卡过期或者有效期有误，无法进行交易
20670	退款正处理中	托收退款中
20671	订单号不存在	渠道方未收到交易请求，交易失败，请重新发起
20676	交易待查询	交易已提交，待查询
20677	交易已提交至发卡行处理	交易已提交，待查询
20681	未签约账户	账户未签约
20684	重复业务	系统重复向渠道发同样的请求
20685	白名单校验不通过	白名单校验不通过，传入信息有误
20686	无效货币	货币类型无效
20687	无效账户类型	账户类型无效

3.1.10 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

3.1.11 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