**金融开放平台接入规范**

**金融开放平台项目组**

**2021年11月**

文档信息

| 文档版本号 | V1.3 | | |
| --- | --- | --- | --- |
| 起草人 | 金融开放平台项目组 | 起草日期 |  |
| 审核人 |  | 审核日期 |  |
| 批准人 |  | 批准日期 |  |

修订记录

| 版本 | 日期 | 作者 | 说明 |
| --- | --- | --- | --- |
| V1.0 | 20190501 | 项目组 | 初始版本 |
| V1.1 | 20200620 | 项目组 | 将fastjson替换为jackson |
| V1.2 | 20211025 | 项目组 | 新增SDK中使用国密算法的说明和代码示例 |
| V1.3 | 20211108 | 项目组 | 新增通知接口调用示例说明 |
| V1.3.1 | 20211131 | 项目组 | 更新SM2签名的描述，新增.cn域名和FAQ |
| V1.3.2 | 20211202 | 项目组 | 更新投产说明和接口规范 |
| V1.3.3 | 20211216 | 项目组 | 新增ext\_info和风控字段的描述 |

[1前言 5](#_Toc26143)

[1.1编写目的 5](#_Toc21563)

[1.2读者对象 5](#_Toc29522)

[2 金融开放平台概述 5](#_Toc12538)

[2.1 简介 5](#_Toc25007)

[2.2 输出模式 5](#_Toc9018)

[3 接入指南 5](#_Toc29063)

[3.1 接入流程 5](#_Toc25600)

[3.2 合作方开发准备 6](#_Toc29156)

[3.2 联调环境测试 6](#_Toc24574)

[3.3 投产 6](#_Toc14160)

[4接口规范 7](#_Toc20908)

[4.1字符集和编码 7](#_Toc21012)

[4.2 通讯规范 8](#_Toc4376)

[4.3业务请求报文结构 8](#_Toc18247)

[4.4文件请求报文结构 9](#_Toc28472)

[4.5 返回报文结构 9](#_Toc27482)

[4.6 业务状态字 10](#_Toc24771)

[4.7报文签名规则 11](#_Toc12810)

[4.8报文加密规则 13](#_Toc7390)

[5 调用过程 14](#_Toc9950)

[5.1 API模式调用 14](#_Toc13851)

[5.2 H5模式调用 15](#_Toc6811)

[6 SDK使用说明 15](#_Toc10404)

[6.1 结构说明 16](#_Toc13654)

[6.2 资源引用 16](#_Toc28297)

[6.3 签名密钥生成 17](#_Toc23047)

[6.4 加密密钥生成(目前一般不用） 17](#_Toc29562)

[6.5 网关公钥 17](#_Toc28621)

[6.6 网络访问权限 17](#_Toc19330)

[6.7 接口调用示例 18](#_Toc1606)

[6.7.1 API调用对象说明 18](#_Toc27778)

[6.7.2 API调用示例 18](#_Toc13731)

[6.7.3 通知接口调用示例 20](#_Toc26737)

[6.7.4文件传输 23](#_Toc24675)

[6.8 返回信息说明 25](#_Toc22191)

[6.8.1返回说明 25](#_Toc27606)

[6.8.2返回判断示例 25](#_Toc19697)

[6.8.3返回信息列表 25](#_Toc28125)

[9附录 27](#_Toc3217)

[9.1 域名解析地址 27](#_Toc383)

[9.2 Q&A 28](#_Toc28480)

# 1前言

## 1.1编写目的

本文档主要用于帮助合作方相关技术人员了解与交通银行金融开放平台对接的流程（线下）和开发规范。在金融开放平台开发者门户建设完成之后，对接流程将从线下移交到线上。

## 1.2读者对象

合作方相关技术人员

# 2 金融开放平台概述

## 2.1 简介

金融开放平台是交通银行为第三方合作伙伴提供金融服务输出的合作平台，为合作伙伴提供从入驻到运营的全流程服务。

## 2.2 输出模式

金融开放平台对外服务输出的形式主要包含API模式和H5模式。API模式为合作方直接通过请求开发平台API调用交行后台服务,返回后台数据，H5模式为合作方通过调用一次API请求交行H5链接，经浏览器打开后进入交行H5场景。

# 3 接入指南

## 3.1 接入流程

目前合作方接入交通银行金融开放平台的方式为线下方式，由交通银行对接人员和合作方线下沟通联系，确认合作方资质，明确对接的模式和API。后续进行测试联调、投产上线。流程如下：



## 3.2 合作方开发准备

在交通银行对接人员提供最新SDK包及相关说明文档给合作方之后，合作方可以开始开发，清单如下：

|  |  |  |
| --- | --- | --- |
| 物料 | 必要 | 说明 |
| boocm-api-sdk\_v1.1\_xxx.zip | 是 | Sdk包，包含接口文档和需要的jar包，demo，javabean，密钥生成工具等 |

## 3.2 联调环境测试

合作方使用交通银行提供的SDK包，测试和开放平台的联通性。SDK使用方法见前文第6章<SDK使用说明>。测试中发现的问题请参考附录<Q&A>。

个别业务，合作方需开发接收交行的通知功能。具体通知要求见第6.7节<通知接口调用示例>。

测试完成后提供验收报告给相关行方人员，由行方人员审核。具体请联系对口行方人员。

## 3.3 投产

测试验收成功后，开始投产准备。

* 第三方需用sdk包里的公私钥生成工具keygen\_sm2.bat(国密）或keygen\_rsa.bat（国际），生成一对公私钥对，把公钥发给分行对接人员进行入驻上传。保留生成的私钥。
* 第三方部署环境如果出口IP固定（一年内不更换），请提供出口IP给分行对接人员，便于商户入驻上传，后续开放平台做IP白名单控制，增强访问安全性。
* 分行对接人员通过GUIP登录合作商平台进行商户入驻，将第三方生成好的公钥上传，维护第三方提供的出口IP，入驻完成后将合作商平台生成的appid 提供给第三方，同时从合作商平台将交行生产网关的公钥bocom\_prd.pub下载下来提供给第三方。
* 第三方用上述appid、开放平台网关公钥、自己生成的私钥替换测试环境的相应参数到生产版本，并且沿用测试环境的合作方编号。
* 投产前需验证生产网络联通性，若有回调通知类接口，第三方回调地址需开放80或443端口。

具体说明：

|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 参数名称 | 参数提供方 | 说明 |
| 1 | 第三方标识 | 合作商平台 | 外部系统编号（后台系统发布API接口的时候设置协议号的字段，一般是客户号、机构号、商户号等） |
| 2 | 生产环境地址 | 开放平台 | 公网接入：  **<https://open.bankcomm.cn> (推荐）**  <https://open.bankcomm.com>（备用，将来会下线）  专线接入：https://入口地址 （如果从分行端接入，分行需要开通到182.251.184.11:8090的网络） |
| 3 | APP\_ID | 合作商平台 | 商户通过线上中心合作商平台入驻产生的APP\_ID |
| 4 | 开放平台公钥 | 合作商平台 | 合作商平台下载 |
| 5 | 第三方公钥 | 第三方 | 第三方用开放平台提供的sdk包里keygen\_sm2.bat或keygen\_rsa.bat 自动生成一对公私钥对，把公钥上传到合作商平台，保留私钥替换商户的生产版本。第三方发给商拓转发开放平台 |
| 6 | 第三方出口IP | 第三方 | 如果第三方出口IP固定，将出口IP维护到合作商平台，如果出口IP不固定，一段时间会变化，可暂不提供。第三方发给商拓转发开放平台 |

双方按照约定的投产时间投产并进行生产验证

**注意：生产私钥要严格保密。**

# 4接口规范

## 4.1字符集和编码

所有报文采用UTF-8编码方式，传输报文时，应注意报文中不能包含UTF-8编码的BOM header（即EF BB BF）。在整个开放平台的字段命名格式要求采用小写字母+下划线的方式(UnderScoreCase)进行字段命名，例如app\_id、msg\_id，不建议采用驼峰方式(CamelCase)进行字段命名。

## 4.2 通讯规范

开放平台的通讯协议为：HTTPS POST请求，HTTPS表示单向SSL请求，保证链路通讯层安全，请求URL规则为：https://域名/api/组名/操作名/版本号；请求URL示例：

范例：[https://open.bankcomm.cn/api/group/action/version](https://open.95559.com.cn/api/group/ApiName/version)

具体实例：https://open.bankcomm.cn/api/jdong/findDepositList/v1

## 4.3业务请求报文结构

数据类型API所有参数建议均在POST的body中传送，通过form提交，form中要指明Content-Type属性，Content-Type: application/x-www-form-urlencoded，http header中禁止使用Expect参数。form请求结构如下：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 参数名 | 类型 | 是否必输 | 最大长度 | 描述 | 示例值 |
| app\_id | str | true | 50 | 应用ID标识 | appotets202104140001 |
| msg\_id | str | true | 40 | 消息通讯唯一表示 | uuid |
| fmt\_type | str | true | 5 | 请求参数格式 json | json |
| charset | str | false |  | 字符集 | UTF-8 |
| is\_encrypt | str | false | 5 | 是否加密true false  （注：false时，不传is\_encrypt节点） | true |
| timestamp | str | true | 30 | 报文时间戳 | GMT+8 北京时间 |
| biz\_content | str | true |  | 业务区数据 JSON格式 |  |
| sign | str | true |  | 报文签名 |  |
| ext\_info | str | false |  | 扩展数据JSON格式 | {"customer\_client\_info":{"customerIpAddress":"123456","deviceId":"1111"}} |

数据类型请求示例如下：

1. POST HTTP/1.1
2. Content-Type: application/x-www-form-urlencoded; charset=UTF-8
3. https://open.95559.com.cn/api/popentest/test?app\_id=app201811051349&msg\_id=1adc3436052e4496b2afa34e1eee446f&format=json&charset=utf-8&sign=TXxFR9t80R32ZHtro4SIrAr0Gdjo+niy6kCcz+phTT5V6TixL+ZH9cAEDTuB/4IngWOC6So6DzeSL7J+bP16ky+/wzwwkeUXgwpL0+SMzO7DpdPaWl1vkFL2iVFIslLMUPcYiYdcZ7yVJO2mOhElcAiD0rlXmFQnqI8TOmqWrLiZ2GR7NNOHb+DaexyNt9jdTxH0ggnmBM21DkHjFJuS5N7WgLLwTNgNcKaGkUyRgddcaCcHpSKPRi518CghkN3JEE04mZ4W9VlzbrYDyq+FcwE5jsoBq1rIP0OtECH4Ow9dgspRxie4aiVCc8ScwJQ7EYxt6g5VP1f530GJmR1nx/g==&timestamp=2016-10-29 20:44:38&biz\_content=
4. {
5. "cust\_id":"ABC123456",
6. "cust\_name":"TEST",
7. }

部分接口请求报文中要求带上风控报文，风控报文作为ext\_info的一个json键值对传输，键为customer\_client\_info，值为风控报文的json格式的报文体，风控报文体的内容如下：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 名称 | 类型 | 备注 |
| lastAuthDate | 客户端上一次的登录时间 | long |  |
| customerIpAddress | 客户端IP地址 | str |  |
| unitType | 设备类型（IOS,ANDROID,BROWSER,OTHERS） | str | 枚举字段 |
| unitModel | 设备型号 | str |  |
| systemVersion | 系统版本号 | str |  |
| userAgent | user-agent | str | 浏览器`UA`信息 |
| mac | mac地址 | str |  |
| deviceId | 设备终端号 | str |  |
| imei | 手机序列号 | str |  |
| gps | 用户位置信息 | str | longitude,latitude |

## 4.4文件请求报文结构

文件类型API所有参数建议均在POST的body中传送，通过form提交，form中要指明Content-Type属性，Content-Type: multipart/form-data，http header中禁止使用Expect参数。form请求结构如下：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 参数名 | 类型 | 是否必输 | 最大长度 | 描述 | 示例值 |
| app\_id | str | true | 50 | 应用ID标识 | appotets202104140001 |
| msg\_id | str | true | 40 | 消息通讯唯一表示 | uuid |
| fmt\_type | str | true | 5 | 请求参数格式 json | json |
| charset | str | false |  | 字符集 | UTF-8 |
| timestamp | str | true | 30 | 报文时间戳 | GMT+8 北京时间 |
| file\_hashcode | str | true |  | 文件hash值 |  |
| sign | str | true |  | 报文签名 |  |
| file\_content | stream | true |  | 文件内容 | 流 |

## 4.5 业务返回报文结构

数据类型API返回报文的Content-Type属性为Content-Type: application/json；charset=UTF-8，返回报文结构如下：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 参数名 | 类型 | 是否必输 | 最大长度 | 描述 | 示例值 |
| rsp\_biz\_content | str | true |  | 返回业务区字段标识 |  |
| --biz\_state | Str | true | 1 | 返回当前请求业务状态字 | P:处理中  S:成功  F:失败 |
| --rsp\_code | str | true | 8 | 返回码 |  |
| --rsp\_msg | str | true | 60 | 返回信息 |  |
| --ref\_msg\_id | str | true | 40 | 源请求msg\_id |  |
| sign | str | true | 40 | 报文返回签名信息 |  |
|  |  |  |  |  |  |

返回数据格式示例如下：

1. {
2. "rsp\_biz\_content":{
3. "biz\_state":"S",
4. "rsp\_code":"0000",
5. "rsp\_msg":"success",
6. "cust\_id":"your cust id",
7. "cust\_name:"your cust name"
8. },
9. "sign":"ckVL6FMpSPokuqEzpA02xrQAQZkgpn3JwJLf0Ig3smC3P2y5odhZ9IlcmNHD8wMCUOSKWvVBFpv2gpsx+s7OXqpjsiOZ3i4ibWHUSX0OPBPYtuLxJ/wlTolq9B36mRfj54UT6Uwrl0Vls20c+7RpZbz2HkEy4Ea7F2SUkqaLgXY="
10. }

## 4.6文件返回报文结构

数据类型API返回报文成功时Content-Type属性为Content-Type: application/octet-stream,报文头内容Content-Disposition为attachment; filename=文件名、file\_hashcode为文件hashcode.失败时Content-Type属性为Content-Type: application/json；charset=UTF-8，返回报文结构如下：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 参数名 | 类型 | 是否必输 | 最大长度 | 描述 | 示例值 |
| rsp\_biz\_content | str | true |  | 返回业务区字段标识 |  |
| --biz\_state | Str | true | 1 | 返回当前请求业务状态字 | P:处理中  S:成功  F:失败 |
| --rsp\_code | str | true | 8 | 返回码 |  |
| --rsp\_msg | str | true | 60 | 返回信息 |  |
| --file\_name | str | true | 200 | 文件名称 |  |
| --file\_hashcode | str | true | 200 | 文件hashcode |  |
| sign | str | true | 40 | 报文返回签名信息 |  |

## 4.7 业务状态字

业务状态字表示APP和开放平台之间当前交互业务接口的状态，在开放平台返回APP的json报文中的存在着统一的返回biz\_state字段，它包含3个状态位，P(rocessing)处理中、S(uccess)成功、F(ailed)失败，APP可根据此字段进行当前交互接口的业务状态进行判断。**如果业务体中有具体的业务状态说明字段，请以业务体中状态为准。**

如以下报文中业务状态以order\_status为准：

{

"biz\_state": "S",

"rsp\_code": "SC0000",

"rsp\_msg": null,

"merchant\_id": "0010105023999",

"merchant\_org\_id": "00101050239990001",

"merchant\_order\_id": "10007367",

"order\_status": "0",

"fail\_reason": null,

"sel\_pay\_amt\_payment\_status": "1",

"drawdown\_time": null,

"repay\_day": null

}

## 4.8报文签名规则

开放平台下载的sdk已封装签名验签方法，开发者只需要调用sdk配置业务入参即可，用sdk封装的方法发送请求到开放平台时，sdk会自动签名。如开发者不用sdk，可根据签名规则自己拼写签名方法。以下是结合开放平台业务对自主签名进行简单说明：

* **签名算法**

目前开放平台支持国际加密算法和国密加密算法，默认采用国际加密算法，若是要使用国密加密算法，使用SDK时需要在DefaultBoComClient的构造函数中将加密算法显式的指定为EncryptType.SM2\_AND\_SM4。

若采用国际加密算法，API开放平台签名算法分为RSA，对应的实际签名算法为SHA256WITHRSA的签名算法。

若采用国密加密算法，API开放平台签名算法分为SM2，对应的实际签名算法为SM3WITHSM2的签名算法。

* **请求参数签名**

1. 筛选

获取所有请求参数，不包括字节类型参数，如文件、字节流，剔除sign字段。

1. 排序

将筛选的参数按照第一个字符的键值ASCII码递增排序（字母升序排序），如果遇到相同字符则按照第二个字符的键值ASCII码递增排序，以此类推。

1. 拼接

将排序后的参数与其对应值，组合成“参数=参数值”的格式，并且把这些参数用&字符连接起来，此时生成的字符串为待签名字符串。SDK中已封装签名方法，开发者可直接调用，详见SDK说明。如自己开发，则需将待签名字符串和私钥放入SHA256WITHRSA\SM3WITHSM2算法中得出签名（sign）的值。

1. 调用签名函数

现将拼接后的参数，按照编码类型处理为byte数组，若使用国际算法，使用各自语言对应的RSA签名函数利用商户私钥对待签名字符串进行签名，并将签名后结果进行Base64编码；若使用国密算法，使用各自语言对应的SM2签名函数利用商户私钥对待签名字符串进行签名，并将签名后结果进行Hex编码。

注:生成签名时，签名原文不要做URLEncode。

API签名原文构造参考：

/api/opentest/test?app\_id=app201811051349&biz\_content={"send\_date":"20181119","trans\_code":"2701","tran\_date":"20181116","tran\_time":"100113","sys\_trace\_no":"201811190100439","proxy\_code":"20005","opertype":"1","third\_bank\_acct":"6229181000179846","orgno":"01310320999","idtype":"0","idno":"420583198303141939","custname":"中国","answer\_string":"ABCDAADB","sex":"0","contact\_tel":"010-1102110","contact\_mobile":"13786865858","contact\_address":"湖北省宜昌市","zipcode":"","citizenship":"CHN","occupation":"001","id\_expiry\_date":"2027-10-25","remark":"","msgend":"YYPTSEND"}&charset=UTF-8&fmt\_type=json&msg\_id=1adc3436052e4496b2afa34e1eee446f&timestamp=2019-01-07 15:55:45

若请求报文带有ext\_info，API签名原文构造参考：

/api/opentest/test?app\_id=app201811051349&biz\_content={"send\_date":"20181119","trans\_code":"2701","tran\_date":"20181116","tran\_time":"100113","sys\_trace\_no":"201811190100439","proxy\_code":"20005","opertype":"1","third\_bank\_acct":"6229181000179846","orgno":"01310320999","idtype":"0","idno":"420583198303141939","custname":"中国","answer\_string":"ABCDAADB","sex":"0","contact\_tel":"010-1102110","contact\_mobile":"13786865858","contact\_address":"湖北省宜昌市","zipcode":"","citizenship":"CHN","occupation":"001","id\_expiry\_date":"2027-10-25","remark":"","msgend":"YYPTSEND"}&charset=UTF-8&ext\_info={"customer\_client\_info":{"customerIpAddress":"123456","deviceId":"1111"}}&fmt\_type=json&msg\_id=1adc3436052e4496b2afa34e1eee446f&timestamp=2019-01-07 15:55:45

* **返回参数验证签名**

开发者只对API平台返回的json中rsp\_biz\_content的值做验签。rsp\_biz\_content的Json值内容，如为json则需要包含首尾的“{”和“}”两个尖括号，如为字符串则需包括前后引号，如为数组，则需包 含首位的“[”和“]”，作为验签整体。

返回参数签名验证示例

1. {
2. "rsp\_biz\_content":{
3. "biz\_state":"S",
4. "rsp\_code":"0000",
5. "rsp\_msg":"success",
6. "cust\_id":"your cust id",
7. "cust\_name":"your cust name"
8. },
9. "sign":"ckVL6FMpSPokuqEzpA02xrQAQZkgpn3JwJLf0Ig3smC3P2y5odhZ9IlcmNHD8wMCUOSKWvVBFpv2gpsx+s7OXqpjsiOZ3i4ibWHUSX0OPBPYtuLxJ/wlTolq9B36mRfj54UT6Uwrl0Vls20c+7RpZbz2HkEy4Ea7F2SUkqaLgXY="
10. }

返回待签名数据：

1. {
2. "biz\_state":"S",
3. "return\_code":"0000",
4. "return\_msg":"success",
5. "cust\_id":"your cust id",
6. "class\_name:"your cust name"
7. }

## 4.9报文加密规则

如果第三方对于报文传输的信息需要进行严格的保护，可以联系交行产品服务方在发布API接口的时候选择相应的报文字段进行字段加密，目前字段加密默认是AES加密（256位）。

但是对于对外通知的接口，目前默认是全报文加密。

# 5 调用过程

## 5.1 API模式调用



合作方使用我行提供的开发工具包（SDK）实例化输入bean并发起调用，API网关完成交易验签、权限控制、限流、解密（可选），服务路由完成协议转换，服务方接收到RPC调用请求，并进一步通过接口类生成bean的实例（可选）。 服务方完成交易后，通过输出bean的实例（可选）生成json报文，返回给API网关，API网关完成签名后发送给合作方，合作方验签通过后将响应报文反序列化成输出bean。

## 5.2 H5模式调用



页面API与非页面API调用过程区别体现在以下几点：

①API网关负责商户接入。商户APP调用API网关请求首页地址，API网关验签通过后，返回首页地址，并附带API网关授权参数。

②用户浏览器进入H5。商户端收到首页地址后，以URL链接或二维码方式展示，经浏览器打开后进入交行H5场景，后续用户操作在交行H5页面上进行。

# 6 SDK使用说明

SDK是为调用方简化API开放平台调用专门提供的开发工具包，包括密钥生成、服务调用、返回结果解析等功能。目前**仅提供Java**版本实现，如果合作方未使用Java语言开发，请参照SDK的接口文档里的说明规范拼装报文，具体代码也可以反编译SDK包里的bocom-openapi-sdk-1.1.jar ，并参照其写法。

本文档所述基于SDK的Java版本实现。SDK开发运行环境：JDK1.6及以上。

每个已注册的应用会自动分配一个应用ID，调用方利用SDK提供的密钥生成功能，生成公私钥对，并将公钥提供给API开放平台。API开放平台维护应用ID与公钥的对应关系。

调用方采用私钥对系统时间加密，并将明文和密文一并发送到API开放平台验签。API开放平台收到请求后提供服务，并以Json格式返回响应结果，调用方收到返回信息后使用网关公钥对报文进行验签。使用SDK的过程中，请求信息的签名、组装及响应信息的验签、解析对调用方透明。

## 6.1 结构说明

|---JAVA-SDK

|----lib

|---bocom-openapi-sdk-1.x.jar (基础JAVA-SDK集成jar，不可更改)

|---jackson-all-2.8.9.jar (json组包组件)

|----doc

|---交通银行金融开放平台-xxxx对外输出文档\_V.x.doc

|---金融开放平台接入规范(商户版)V1.1.docx

|----bin

|---keygen\_rsa.bat, keygen\_rsa.sh, keygen\_aes.sh, keygen\_aes.bat, keygen\_sm2.sh, keygen\_sm2.bat, keygen\_sm4.sh, keygen\_sm4.bat

|----example

|----com\bocom\api\example\xxx

|---xxxxTest.java ...

|----com\bocom\api\request\xxx

|---xxxxRequestV1.java ...

|----com\bocom\api\response\xxx

|---xxxxResponseV1.java ...

## 6.2 资源引用

下载SDK后，在工程中直接引用，并引用lib文件夹下的所有jar包。

bocom-openapi-sdk-x.x.x.jar为必需集成jar

其他依赖jar，需要一起加入到项目中。**如果项目中已存在相同jar包可不替换**(注：jackson-all-2.8.9.jar包中包含jackson-annotations-2.8.9.jar、jackson-core-2.8.9.jar、jackson-databind-2.8.9.jar)；如版本不一致，请在不影响原系统使用的情况下更换成上述版本的jar包。如有冲突请联系开放平台项目组。

SDK包中已经提供测试demo，可以在工程中引入example目录下的代码，解决好代码路径并编译通过后，可以直接运行demo（xxxTest.java)进行联通性测试。**Demo里面已经提供测试环境的公私钥对和默认appid，测试时不用改动。**投产前需要替换demo里的MY\_PRIVATE\_KEY（第三方私钥），APIGW\_PUBLIC\_KEY（交行公钥），APP\_ID(交行分配)，APIGW\_URL\_ADDRESS（公网：[https://open.bankcomm.c](https://open.bankcomm.com)n 专线：182.251.184.11:8090）。

## 6.3 签名密钥生成

* Windows环境

若生成RSA密钥，进入到bin目录，双击keygen\_rsa.bat文件，生成一对RSA公钥和私钥。

若生成SM2密钥，进入到bin目录，双击keygen\_sm2.bat文件，生成一对SM2公钥和私钥。

* Linux环境

若生成RSA密钥，切换到bin目录，运行 ./keygen\_rsa.sh命令，生成一对RSA公钥和私钥。

若生成SM2密钥，切换到bin目录，运行 ./keygen\_sm2.sh命令，生成一对SM2公钥和私钥。

## 6.4 网关公钥

测试环境公钥（RSA）：

MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAxSJrF8T/5rKB4NnwMjIxUer+ELf1PQXO2GSdZ/fvuQCclOR9tBlNWL4jFOftebeL+bvMVOJ+JHm/aSes1AN8YNIDGiFUpF6aDkSCaLynDdjK/mQTWhSNa2fO0GGO+ywOBTdYUjVjVtzJ48bbyG3NSylf1EdnBWnMpFa8qpXJXR4ELpVpMkPDC+93HBAlxEgUjhcIJlP5VdKIiudsmhE2T07qtpIQSuE5hntXP6X6GKJReCk+yek2QJITvIBq3cHPw8KDsHHCs7MaR81KI3onJRWyqFtTfVYTiKsd9EcYSxv+Gx5MOF8B/P4iJCD8uzx0FrqoB3k5OYGcz4tXs+h+9wIDAQAB

测试环境公钥（SM2）：

04A33690CEC47663B70CBFF54E073AB135E04133A0E100E3F5B8E9D5C2C1E1D66E9326026A283900584CDF87BAC9740F565229FCB60C8F420FDA036A2AACD03422

## 6.5 网络访问权限

第三方开发者需要确保其生产环境能访问open.bankcomm.cn和测试环境能访问以下地址。

https://117.184.192.242:9443

## 6.6 接口调用示例

### 6.6.1 API调用对象说明

数据类型API使用SDK时，通过默认调用实现类DefaultBocomClient构造调用对象client，构造函数参数说明如下：

|  |  |
| --- | --- |
| **参数定义** | **参数说明** |
| String appId | APP的编号,应用在合作商入驻的时候生成 |
| String privateKey | 第三方应用私钥 |
| String bocomPulicKey | 交行网关公钥 |

数据类型API调用对象构造示例：

|  |
| --- |
| // 使用缺省值构造，只需传入APP编号，应用私钥和网关公钥  DefaultBocomClient client = **new** DefaultBocomClient(***APP\_ID***,  ***MY\_PRIVATE\_KEY***, ***APIGW\_PUBLIC\_KEY***);  // 传入全部参数构造client对象  DefaultBocomClient client = **new** DefaultBocomClient(***APP\_ID***,  ***MY\_PRIVATE\_KEY,***BocomConstants.***CHARSET\_UTF8,***BocomConstants.***FORMAT\_JSON, APIGW\_PUBLIC\_KEY***, BocomcConstants.***ENCRYPT\_TYPE\_AES***, ***ENCRYPT\_KEY***, null); |

### 6.6.2 API调用示例

以使用缺省的RSA签名为例，数据类型API调用示例如下：

|  |
| --- |
| // 构造client对象  DefaultBocomClient client = **new** DefaultBocomClient(***APP\_ID***, ***MY\_PRIVATE\_KEY***, ***APIGW\_PUBLIC\_KEY***);  //可设置代理服务器  //client.setProxy("182.119.1.100", 8090);  //可设置建立连接时间和响应超时时间  client.setConnectTimeout(5000);  client.setReadTimeout(5000);  //在握手期间，如果URL的主机名和服务器的标准主机名不匹配，则验证机制可以回调此接口来确定是否允许此连接  //测试环境需加上，来忽略主机名和服务器的标准主机名不匹配，生产环境不可忽略  client.ignoreSSLHostnameVerifier()  // 设置请求对象request  OpenApiSdkTestRequestV1 request = **new** OpenApiSdkTestRequestV1();  // 设置请求路径 request.setServiceUrl("https://open.bankcomm.com/api/opentest/test/V1");  // 设置业务参数，每个Request请求实现类都有一个RequestBiz内部类用来设置业务参数  OpenApiSdkTestRequestV1.OpenApiSdkTestRequestV1Biz bizContent = **new** OpenApiSdkTestRequestV1.OpenApiSdkTestRequestV1Biz();  bizContent.setClient\_id("123456");  bizContent.setOld\_phone\_no("12345678901");  bizContent.setNew\_phone\_no("12345678901");  bizContent.setSms\_id("1");  bizContent.setSms\_identify\_no("1");  bizContent.setSms\_key("中文测试");  bizContent.setSms\_no("1");  request.setBizContent(bizContent);  // 组装完成，开始发起调用  OpenApiSdkTestResponseV1 response = client.execute(request, UUID.*randomUUID*().toString().replace("-", ""));  // 判断调用是否成功，进行后续业务处理  **if** (response.isSuccess()) {  System.***out***.println(response.getMsgInfo());  } **else** {  System.***out***.println(response.getRspCode());  System.***out***.println(response.getRspMsg());  } |

若采用国密算法，API调用示例如下：

|  |
| --- |
| // 构造client对象，在构造方法中通过EncryptType.SM2\_AND\_SM4指定调用加密算法为国密  DefaultBocomClient client = **new** DefaultBocomClient(***APP\_ID***, ***MY\_PRIVATE\_KEY***, ***APIGW\_PUBLIC\_KEY***, ***EncryptType.SM2\_AND\_SM4***  );  //可设置代理服务器  //client.setProxy("182.119.1.100", 8090);  //可设置建立连接时间和响应超时时间  client.setConnectTimeout(5000);  client.setReadTimeout(5000);  //在握手期间，如果URL的主机名和服务器的标准主机名不匹配，则验证机制可以回调此接口来确定是否允许此连接  //测试环境需加上，来忽略主机名和服务器的标准主机名不匹配，生产环境不可忽略  client.ignoreSSLHostnameVerifier()  // 设置请求对象request  OpenApiSdkTestRequestV1 request = **new** OpenApiSdkTestRequestV1();  // 设置请求路径 request.setServiceUrl("https://open.bankcomm.com/api/opentest/test/V1");  // 设置业务参数，每个Request请求实现类都有一个RequestBiz内部类用来设置业务参数  OpenApiSdkTestRequestV1.OpenApiSdkTestRequestV1Biz bizContent = **new** OpenApiSdkTestRequestV1.OpenApiSdkTestRequestV1Biz();  bizContent.setClient\_id("123456");  bizContent.setOld\_phone\_no("12345678901");  bizContent.setNew\_phone\_no("12345678901");  bizContent.setSms\_id("1");  bizContent.setSms\_identify\_no("1");  bizContent.setSms\_key("中文测试");  bizContent.setSms\_no("1");  request.setBizContent(bizContent);  // 组装完成，开始发起调用  OpenApiSdkTestResponseV1 response = client.execute(request, UUID.*randomUUID*().toString().replace("-", ""));  // 判断调用是否成功，进行后续业务处理  **if** (response.isSuccess()) {  System.***out***.println(response.getMsgInfo());  } **else** {  System.***out***.println(response.getRspCode());  System.***out***.println(response.getRspMsg());  } |

若在请求报文中传送风控字段，调用示例如下：

|  |
| --- |
| // 构造client对象  DefaultBocomClient client = **new** DefaultBocomClient(***APP\_ID***, ***MY\_PRIVATE\_KEY***, ***APIGW\_PUBLIC\_KEY***);  //可设置代理服务器  //client.setProxy("182.119.1.100", 8090);  //可设置建立连接时间和响应超时时间  client.setConnectTimeout(5000);  client.setReadTimeout(5000);  //在握手期间，如果URL的主机名和服务器的标准主机名不匹配，则验证机制可以回调此接口来确定是否允许此连接  //测试环境需加上，来忽略主机名和服务器的标准主机名不匹配，生产环境不可忽略  client.ignoreSSLHostnameVerifier()  // 设置请求对象request  OpenApiSdkTestRequestV1 request = **new** OpenApiSdkTestRequestV1();  // 设置请求路径 request.setServiceUrl("https://open.bankcomm.com/api/opentest/test/V1");  // 设置业务参数，每个Request请求实现类都有一个RequestBiz内部类用来设置业务参数  OpenApiSdkTestRequestV1.OpenApiSdkTestRequestV1Biz bizContent = **new** OpenApiSdkTestRequestV1.OpenApiSdkTestRequestV1Biz();  bizContent.setClient\_id("123456");  bizContent.setOld\_phone\_no("12345678901");  bizContent.setNew\_phone\_no("12345678901");  bizContent.setSms\_id("1");  bizContent.setSms\_identify\_no("1");  bizContent.setSms\_key("中文测试");  bizContent.setSms\_no("1");  request.setBizContent(bizContent);  //开始风控相关配置  CustomerClientInfo customerClientInfo = new CustomerClientInfo();  customerClientInfo.setImei("123");  request.setExtraParameters(customerClientInfo);  //风控相关配置结束  // 组装完成，开始发起调用  OpenApiSdkTestResponseV1 response = client.execute(request, UUID.*randomUUID*().toString().replace("-", ""));  // 判断调用是否成功，进行后续业务处理  **if** (response.isSuccess()) {  System.***out***.println(response.getMsgInfo());  } **else** {  System.***out***.println(response.getRspCode());  System.***out***.println(response.getRspMsg());  } |

### 6.6.3 通知接口调用示例（旧）

某些业务场景中，合作方需要开发接收交行的通知功能，当调用交行API时，及时接收交行通知。通知地址在交行API接口上送。**注意：合作方的通知地址需要开放80或443端口。**

通知的交互形式为我行服务器后台请求商户服务器。以收单业务中的被扫支付交易为例，当被扫交易发生时，交行支付系统主动向合作方的通知地址发送付款后的最终结果。商户端用RSASignData参数接收银行端返回的报文，得到支付结果信息。

合作方在接收到后台通知信息后，须返回给交行响应成功信息，交行在收到合作方响应成功信息后，则认为该笔订单的主动通知行为结束。合作方如果收到支付失败或未决的情况，可正向发起查询交易，确定最终交易结果。

* **通知报文结构**

回调通知接口返回给商户端的json报文中不包含biz\_state字段，它的返回格式如下所示：

返回报文的Content-Type属性为**Content-Type: application/json**；charset=UTF-8，返回报文结构如下：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 参数名 | 类型 | 是否必输 | 最大长度 | 描述 | 示例值 |
| notify\_biz\_content | str | true | 250 | 返回业务区字段标识 |  |
| sign | str | true | 40 | 报文返回签名信息 |  |

示例返回报文：

{

"notify\_biz\_content": {

"merchant\_id": "0010103026999",

"merchant\_org\_id": "00101030269990001",

"merchant\_order\_id": "13",

"order\_status": "0",

"fail\_reason": null,

"fail\_reason\_num": null,

"tran\_time": "2020-08-26 11:07:23",

"notify\_url": "http://notifyUrl",

"repay\_day": null

},

"sign": "HaqN9k9mbxYVCAZj10XREpx6PplvlT061TMLHC7CZkfQv14Lgxo1jcwb3/XCwd6I2Ay88hug5Mz5wyrm8wUQfN3dCT2YCKVALsZoMMS7xX6j/8s2Ian0Yjgo3+kfOqEExOWs4uGDdWva48zzGyVcO8DmV2bV2utGmnMROes3FuzQp/JAp/OTlbdldAVZNcK+ZLD0uW9I2lAUg9YNFjAlHixCRdAh0vTmTZp8D0oK23P85hJ7x2AMwU2aDzpAoBIroMHEFi+vSYbK0TLIULtzUYy7KmVDIrXD7HiUnJmU5E5llE+zZxBzmnMpJikPEv/xxY5cYCwDDMijzcwa+Nb+SA=="

}

* **通知报文验签**

合作方获取到以上通知报文后，需要用开放平台提供的公钥验证签名，具体验签方法示例如下，也可在SDK里的example\com\bocom\api\example\xxx 目录下找到xxxCallbackV1Test.java参考。

* **7.3响应报文**

public class TradeCallbackV1Test {

public static final String APIGW\_PUBLIC\_KEY = "MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAxSJrF8T/5rKB4NnwMjIxUer+ELf1PQXO2GSdZ/fvuQCclOR9tBlNWL4jFOftebeL+bvMVOJ+JHm/aSes1AN8YNIDGiFUpF6aDkSCaLynDdjK/mQTWhSNa2fO0GGO+ywOBTdYUjVjVtzJ48bbyG3NSylf1EdnBWnMpFa8qpXJXR4ELpVpMkPDC+93HBAlxEgUjhcIJlP5VdKIiudsmhE2T07qtpIQSuE5hntXP6X6GKJReCk+yek2QJITvIBq3cHPw8KDsHHCs7MaR81KI3onJRWyqFtTfVYTiKsd9EcYSxv+Gx5MOF8B/P4iJCD8uzx0FrqoB3k5OYGcz4tXs+h+9wIDAQAB";

public static void main(String[] args) {

try {

/\*\*

\* notifyJson为银行通知给商户的JSON报文，商户通过 上送的notify\_url地址，银行回调该地址

\*/

String notifyJson = "{\"notify\_biz\_content\":{\"tran\_type\":\"PAY\",\"mer\_tran\_no\":\"ZSW201907230001\",\"tran\_state\":\"SUCCESS\",\"tran\_state\_code\":\"\",\"tran\_state\_msg\":\"\",\"batch\_no\":\"20190723\",\"final\_time\":\"20190723093745\",\"amount\":\"1.10\",\"currency\":\"CNY\",\"tran\_content\":\"微信线下34\",\"mer\_memo\":\"微信线下34\",\"bank\_tran\_no\":\"0102201907230936440039860133\",\"third\_party\":\"微信\",\"notify\_url\":\"aaa\",\"partner\_id\":\"301140880629503\",\"mer\_ptc\_id\":\"301910053990034\",\"req\_time\":\"20190723093745\"},\"sign\":\"gw2z0qjTci5BEq8NNY1MeZoBFEM2OW0UfgEcXIsIJiKIZn0aL8WoLiJApphB7FaS3NEW3J/O9pM4xJOeuHuUQGC7cS//lBd5vkOp2I8gmcpRjxkV5ma6GTVA5srgHk9ftF8e2QlHaFSzGZ5foc+53Vt98/kudTBdcE0rOoJD/Drxy4AIGgVgH2sSsIk7MTCQCEicMNOmMkc6NEY+abqWkisTBPpfDBy6tyZAcQ6e5liqQQ/yCiieGlSoQJpgzl7nxh6ZoQj5364IgW7JXFxoyDJuC8qZoeMqqHVIomS5kqtWiT/6MCoAvqfhNnKIaI0dgxYtzreFKa1mnOkze+C0mg==\"}";

TradeCallbackResponseV1 response = (TradeCallbackResponseV1) ApiUtils.parseNotifyJsonWithBocomSign(notifyJson, "UTF-8", APIGW\_PUBLIC\_KEY, TradeCallbackResponseV1.class);

System.out.println(response);

} catch (Exception xcp) {

xcp.printStackTrace();

}

}

}

响应报文中分别用biz\_state、rsp\_code、rsp\_msg字段表示业务状态字、返回码和返回信息。biz\_state为S时表示成功，P数表示处理中（超时），F表示失败。返回不用签名，

1. {
2. "biz\_state":"S",
3. "rsp\_code":"0000",
4. "rsp\_msg":"success",
5. }

### 通知接口调用示例（新）

某些业务场景中，合作方需要开发接收交行的通知功能，当调用交行API时，及时接收交行通知。通知地址在交行API接口上送。**注意：合作方的通知地址需要开放80或443端口。**

通知的交互形式为我行服务器后台请求商户服务器。以收单业务中的被扫支付交易为例，当被扫交易发生时，交行支付系统主动向合作方的通知地址发送付款后的最终结果。

合作方在接收到后台通知信息后，须返回给交行响应成功信息，交行在收到合作方响应成功信息后，则认为该笔订单的主动通知行为结束。合作方如果收到支付失败或未决的情况，可正向发起查询交易，确定最终交易结果。

* **通知报文结构**

回调通知接口返回给商户端的json报文中不包含biz\_state字段，它的返回格式如下所示：

返回报文的Content-Type属性为**Content-Type: application/json**；charset=UTF-8，返回报文结构如下：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 描述 | 示例值 |
| biz\_content | str | 返回业务区字段标识 |  |
| msg\_id | str | 报文返回签名信息 |  |
| timestamp | str | 时间戳 | 2020-10-04 19:00:17 |
| encrypt\_key | str | 对称密钥 |  |
| sign | str | 报文返回签名信息 |  |

示例返回报文：

{"biz\_content":"nPxMcEfq0qCAoRcTY/39d7LVpZglBGQPhnY1HDs03nJ1iPm/i9bPxHkiDrqpmTjuS+EmPr2TVbP7X1+LYQVhuXQsKVyvcwxLFudXzqwi5bl4vfugbXOwBhJtz9gLwEhQBMwOsKpTvuaQzN7hLx0ES9sDiThEuTA0OVhQGJDG8Itofwn2qznvQSzvewZxSovwoRtf2PE8aD6JZkTCXwjpHqEJyr7QAPtxubwo5YwXfH1w85vRRloscGFOlO7okvC9q1bUN4oL4LvEkyOtKSKDaZk88GWiu6t9l0gnreH5fovdouSCrHZMzZ+R+gi5/4a5s5bp6IVxxLDoSL68SOSU3adunJjPIlVpwRDOClmLGjBhwZMhNJ3a0QLYunr1Z54NP0UPGXNTNl6aQzWk3cV2shkfzbJ3D4zT5blCe9LMnUw+4G+cxtYqBlb7iS8V45Lf0Y4/rfbkt2eKWVabiVMyi9qf+VSBWUObgL9DkkOnAD8=","msg\_id":"fb4729ab9b5740b2ad6f8703ec6bac24","timestamp":"2020-10-04 19:00:17","encrypt\_key":"EhEKX32UjHrWDOuoTBN4nU8sdxiXuFQEHt8YqNpeBSPp6LldPucr/+pXNwu/e2EwDhf5TVpJBxvUR8HHF4jBFrwmUeG4ygCeeCJ6JL/ayK9U1oxOXCD4z9m4PMvhDATas12qH4DHu6nyjDg+zzs+k7/SqhhWxiUBfN4+TpRUpT4YyaxYjTHqftPiU2eKvq8pGebZ3jNHZpaeeHbEa2h/n9u/O/HZ0BDMLbk6RvB9WTnvQzFKGDwGvfGHhTmb8qt7lodXEIOZ/GpSXpXEyrxrCOWuGOLgrx1HTzi4AD2TqNLGIQGKWNeoz1iaeFTtFrhWfgzmzrJmtzQUyOIm7p4gkw==","sign":"EZ77fRoQYVJGyo0KETGmsjKy/7/wawmCNxmmzx4Gk6g6O/Ko/2nKPnZ5iNYeMhbDftFxLnq8sX+g9qLDP7+zDlapNFsuBrpc6H14xpeD57MZggBnanM74aztrEROAZ6zcljfpb+b7TNRmNYb63wD6ZPwclpmbh2GTHIOVrBSJnHgFIiJf4iSM7SPCTstfx02/qYq3jPVpvoykz3qpjjRJ8H2uuF7fqMollrPoLrF4lhjj+WoLCErhDKIauIJGb+CLgZMBfYDefDOXBUtomN3K20EglqzgUOI6LkFgUotyG2Jmb8OwH1SF7mm0y5TAG6taS3Ty7cht3Kyg6930X42EA=="}

* **通知报文验签**

合作方获取到以上通知报文后，需要用开放平台提供的公钥验证签名，具体验签方法示例如下，也可在SDK里的example\com\bocom\api\目录下找到SimulateServerExample.java参考。

将通知报文分为待验签字符串（biz\_content,msg\_id,timestamp,encrypt\_key） 以及签名(sign)两部分

bizContentStr

"biz\_content":"nPxMcEfq0qCAoRcTY/39d7LVpZglBGQPhnY1HDs03nJ1iPm/i9bPxHkiDrqpmTjuS+EmPr2TVbP7X1+LYQVhuXQsKVyvcwxLFudXzqwi5bl4vfugbXOwBhJtz9gLwEhQBMwOsKpTvuaQzN7hLx0ES9sDiThEuTA0OVhQGJDG8Itofwn2qznvQSzvewZxSovwoRtf2PE8aD6JZkTCXwjpHqEJyr7QAPtxubwo5YwXfH1w85vRRloscGFOlO7okvC9q1bUN4oL4LvEkyOtKSKDaZk88GWiu6t9l0gnreH5fovdouSCrHZMzZ+R+gi5/4a5s5bp6IVxxLDoSL68SOSU3adunJjPIlVpwRDOClmLGjBhwZMhNJ3a0QLYunr1Z54NP0UPGXNTNl6aQzWk3cV2shkfzbJ3D4zT5blCe9LMnUw+4G+cxtYqBlb7iS8V45Lf0Y4/rfbkt2eKWVabiVMyi9qf+VSBWUObgL9DkkOnAD8=","msg\_id":"fb4729ab9b5740b2ad6f8703ec6bac24","timestamp":"2020-10-04 19:00:17","encrypt\_key":"EhEKX32UjHrWDOuoTBN4nU8sdxiXuFQEHt8YqNpeBSPp6LldPucr/+pXNwu/e2EwDhf5TVpJBxvUR8HHF4jBFrwmUeG4ygCeeCJ6JL/ayK9U1oxOXCD4z9m4PMvhDATas12qH4DHu6nyjDg+zzs+k7/SqhhWxiUBfN4+TpRUpT4YyaxYjTHqftPiU2eKvq8pGebZ3jNHZpaeeHbEa2h/n9u/O/HZ0BDMLbk6RvB9WTnvQzFKGDwGvfGHhTmb8qt7lodXEIOZ/GpSXpXEyrxrCOWuGOLgrx1HTzi4AD2TqNLGIQGKWNeoz1iaeFTtFrhWfgzmzrJmtzQUyOIm7p4gkw=="

Sign

"sign":"EZ77fRoQYVJGyo0KETGmsjKy/7/wawmCNxmmzx4Gk6g6O/Ko/2nKPnZ5iNYeMhbDftFxLnq8sX+g9qLDP7+zDlapNFsuBrpc6H14xpeD57MZggBnanM74aztrEROAZ6zcljfpb+b7TNRmNYb63wD6ZPwclpmbh2GTHIOVrBSJnHgFIiJf4iSM7SPCTstfx02/qYq3jPVpvoykz3qpjjRJ8H2uuF7fqMollrPoLrF4lhjj+WoLCErhDKIauIJGb+CLgZMBfYDefDOXBUtomN3K20EglqzgUOI6LkFgUotyG2Jmb8OwH1SF7mm0y5TAG6taS3Ty7cht3Kyg6930X42EA=="

1.调用 utils/BocomSignature里的方法Verify进行验签

2.验签成功后，调用BocomEncrypt里的方法AsymmetricalDecrypt对对称密钥encrypt\_key进行解密

3.调用BocomEncrypt里的方法SymmetricalDecrypt，用encrypt\_key对bizContent进行解密

* **响应报文**

根据接口文档进行响应

例如响应明文{"token":"456"}，调用BocomEncrypt里的方法SymmetricalEncrypt 进行加密，然后用调用 utils/BocomSignature里的方法Sign进行签名

{"rsp\_biz\_content":"hKIaAMeeT1wZ3+PmsOBZhg==","sign":"YAd5xedLoOeuucTYXMGv//BFC1VAh2i/vpXj3aTlLo033UhSb/LoF8akPjo3r3UajR9x/8NSlKWq4YPX73ZjhLCECaJ1m26Yysb1S5d5bC00bDS6Gb4jvKogysLomuGyuaCoBb+yXi2JwwxRvKKkSySCPHEzigvUf1d9r90yezAoLFaQD0WtxJWwhoLsUf2sHf6LLvPFS5ilumThgh4HhOONSz3yuNFmaw/A/pf+QnMOo0IFpzgq/6gfFApMwqO3MYQB9rfqhwqw0/bxhZE+4ZOEJDOgNQcikbswT3gV0vD8Uk1ncdQ6WeD0BciSGoDm91AjH4obYenHM/4flH2LBA=="}

### 6.6.4文件传输

金融开放平台可提供文件上传和文件下载功能。

* 文件上传

文件上传为标准接口，用户调用文件上传API接口，送到我行内部文件系统，返回上传文件的DOC\_ID。在后续业务API调用中，上送该DOC\_ID，以便和具体业务关联。目前由于带宽限制，我行提供的上传文件只支持单个文件上传，可以是压缩文件，一次请求的文件大小不超过20M。具体调用方式和接口详细信息见接口文档和SDK 示例。

|  |
| --- |
| **try** {  DefaultBocomClient client = **new** DefaultBocomClient(***APP\_ID***, ***MY\_PRIVATE\_KEY***, BocomConstants.***CHARSET\_UTF8***,  BocomConstants.***FMT\_FILE***, ***APIGW\_PUBLIC\_KEY***, **null**, **null**, **null**);  FileUploadRequestV1 request = **new** FileUploadRequestV1();  request.setServiceUrl(***APIGW\_URL\_ADDRESS*** + "/api/common/fileUpload/v1");  request.setFileItem(**new** FileItem(**new** File("C:\\Users\\Administrator\\Desktop\\新建文件夹\\ecm.jpg")));  FileUploadResponseV1 response = client.execute(request, UUID.*randomUUID*().toString().replace("-", ""));  **if** (response.isSuccess()) {  System.***out***.println("success");  System.***out***.println(response.getDocumentId());  } **else** {  System.***out***.println(response.getRspCode());  System.***out***.println(response.getRspMsg());  }  } **catch** (Exception xcp) {  xcp.printStackTrace();  } |

* 文件下载

文件下载接口需根据业务实际需求由交行开发，并提供SDK和接口文档。由于带宽限制，一次请求的文件大小不超过20M。

|  |
| --- |
| **try** {  DefaultBocomClient client = **new** DefaultBocomClient(***APP\_ID***, ***MY\_PRIVATE\_KEY***, ***APIGW\_PUBLIC\_KEY***);  ISVCheckFileDownloadRequestV1 request = **new** ISVCheckFileDownloadRequestV1();  request.setServiceUrl(***APIGW\_URL\_ADDRESS*** + "/api/download/ISVCheckFileDownload/v1");  request.setDownloadPath(***DOWNLOAD\_FILE\_PATH***);  ISVCheckFileDownloadRequestV1.ISVCheckFileDownloadRequestV1Biz fileDownloadRequestV1Biz = **new** ISVCheckFileDownloadRequestV1.ISVCheckFileDownloadRequestV1Biz();  fileDownloadRequestV1Biz.setPartnerId("201906251011");  fileDownloadRequestV1Biz.setQueryDate("2019-06-25");  fileDownloadRequestV1Biz.setOptType("00001");  request.setBizContent(fileDownloadRequestV1Biz);  FileDownloadResponseV1 response = client.execute(request, UUID.*randomUUID*().toString().replace("-", ""));  response.setFilePath(***DOWNLOAD\_FILE\_PATH***);  **if** (response.isSuccess()) {  System.***out***.println("success");  System.***out***.println(response.toString());  } **else** {  System.***out***.println(response.getRspMsg());  }  } **catch** (Exception xcp) {  xcp.printStackTrace();  } |

## 6.7 返回信息说明

### 6.7.1返回说明

响应报文中分别用biz\_state、rsp\_code、rsp\_msg字段表示业务状态字、返回码和返回信息。biz\_state为S时表示成功，P数表示处理中（超时），F表示失败。

|  |
| --- |
| {  "biz\_state": "S",  "rsp\_code": "0000",  "rsp\_msg": "success"  } |

### 6.7.2返回判断示例

|  |
| --- |
| // response对象中获取返回码和返回信息  **String** bizState = response.*getBizState*();  **String** rspMsg = response.*getRspMsg*();  **if**("S".equals(bizState)){  //调用成功处理逻辑  System.***out***.println("调用成功：" + rspMsg);  }**else** **if**("F".equals(bizState)){  //调用失败处理逻辑  System.***out***.println("调用失败：" + rspMsg);  }**else** {  //调用疑帐（或者叫超时、未知）处理逻辑  System.***out***.println("调用超时" + rspMsg);  } |

### 6.7.3返回信息列表

|  |  |
| --- | --- |
| **返回码** | **返回说明** |
| OAPISY0001 | 系统异常 |
| OAPISY0002 | 系统通讯异常 |
| OAPISY0003 | 关键字biz\_state重复 |
| OAPIAU0001 | 获取应用ID异常或者非法应用ID |
| OAPIAU0002 | 该应用ID非在线状态 |
| OAPIAU0003 | 该应用ID无该接口访问权限 |
| OAPIAU0004 | 访问接口未处于上线状态 |
| OAPIRL0004 | 后台繁忙，请稍候尝试 |
| OAPIRL0005 | 获取限流策略异常 |
| OAPIAU0006 | 验证签名失败 |
| OAPIAU0007 | 不支持的解密类型 |
| OAPIAU0008 | 获取解密密钥异常 |
| OAPIAU0009 | 不支持的编码格式 |
| OAPIAU0010 | 非256位AES密钥 |
| OAPICK0001 | app\_id字段不能为空 |
| OAPICK0002 | msg\_id字段不能为空 |
| OAPICK0003 | fmt\_type字段设置错误 |
| OAPICK0004 | is\_encrypt字段设置错误 |
| OAPICK0005 | sign字段不能为空 |
| OAPICK0006 | timestamp字段不能为空 |
| OAPICK0007 | biz\_content字段不能为空 |
| OAPICK0008 | timestamp字段格式错误,请按照yyyy-MM-ddHH:mm:ss格式化 |
| OAPICK0009 | 文件hash值检查错误 |
| OAPICK0010 | 重复访问 |
| OAPICK0011 | 非法访问请求类型，非POST请求 |
| OAPICK0012 | 非法请求数据 |
| OAPICK0013 | auth值为空 |
| OAPICK0014 | 非法请求类型 |
| OAPICK0015 | 非法请求ContentType |
| OAPICK0016 | 不支持的请求浏览器，请使用交通银行手机银行或微信小程序扫码 |
| OAPICK0017 | 请求次数过多，请稍候尝试 |
| OAPICK0018 | session限流关键字为空 |
| OAPICK0019 | 解析授权码失败 |
| OAPICK0020 | 二维码已失效 |
| OAPICK0021 | 文件下载文件大小超限 |
| OAPICK0022 | 非法请求IP |
| OAPICK0023 | 请求或者二维码超时，请重试 |
| OH5TK0001 | TokenId创建失败 |
| OH5TK0002 | TokenId已失效 |
| OH5TK0003 | 会话已失效 |
| OH5TK0004 | 无法获取客户端TokenId |
| OAPIIPT0001 | 禁止重复提交 |
| ADPTSY0001 | 系统异常 |
| ADPTSY0002 | 系统通讯异常 |
| ADPTSY0003 | 与外围系统建立连接失败 |
| ADPTSY0004 | 获取外围系统响应超时 |
| ADPTSY0005 | 与外围系统交互异常 |
| ADPTSY0006 | 与外围系统交互异常 |
| ADPTSY0007 | Encode请求报文发生异常 |
| ADPTSY0008 | Decode响应报文发生异常 |
| ADPTSY0009 | schema检查异常,请检查报文格式 |
| ADPTSY0010 | 关键字biz\_state重复 |
| ADPTSY0011 | 二维码参数生成失败 |
| ADPTSY0012 | 获取请求报文异常 |
| ADPTSY0013 | JsonSchema中Func未定义 |
| ADPTSY0014 | JsonSchema中Func校验失败 |
| ADPTSY0019 | 解析授权码失败 |
| ADPTSY0020 | 二维码已失效 |

# 7附录

## 7.1 域名解析地址

交通银行目前有两个生产域名https://open.bankcomm.cn 和 https://open.bankcomm.com

.com域名将来会下线，还有一个备用域名https://open.95559.com.cn。

交通银行金融开放平台生产域名解析地址如下：

|  |  |  |  |
| --- | --- | --- | --- |
| 域名 | 记录类型 | 对应地址 | 备注 |
| open.bankcomm.com. | A | 27.115.110.154 | 张江联通 |
| open.bankcomm.com. | A | 117.131.72.18 | 漕河泾移动 |
| open.bankcomm.com. | A | 117.184.193.67 | 张江移动 |
| open.bankcomm.com. | A | 116.228.92.240 | 张江电信 |
| open.bankcomm.com. | A | 210.13.123.172 | 漕河泾联通 |
| open.bankcomm.com. | A | 124.74.244.117 | 漕河泾电信 |
| open.bankcomm.cn | A | 210.13.120.81 |  |
| open.bankcomm.cn | A | 58.247.1.111 |  |
| open.bankcomm.cn | A | 117.131.72.60 |  |
| open.bankcomm.cn | A | 117.184.193.102 |  |
| open.bankcomm.cn | A | 58.33.93.210 |  |
| open.bankcomm.cn | A | 58.33.94.210 |  |

金融开放平台出口地址如下：

测试环境出口地址：117.184.192.252

生产环境出口地址：124.74.240.146，备用出口地址124.74.244.13, 180.166.218.66,180.166.223.10, 58.33.94.194, 27.115.110.145, 210.13.92.215

## 7.2 Q&A

**问题描述一:** 测试环境需要产生新的公私钥对测试吗？

不用，SDK包中已经提供测试demo，可以在工程中引入example目录下的代码，解决好代码路径并编译通过后，可以直接运行demo（xxxTest.java)进行联通性测试。Demo里面已经提供测试环境的公私钥对和默认appid，测试时不用改动。投产前需要替换demo里的MY\_PRIVATE\_KEY（第三方私钥），APIGW\_PUBLIC\_KEY（交行公钥），APP\_ID(交行分配)，APIGW\_URL\_ADDRESS

**问题描述 二:** ADPTSY00014 JsonSchema 中Func校验失败[\*\*\*与协议编号不匹配]

该问题是调用接口的时候，\*\*\*该参数字段没有在开放平台测试环境的白名单列表里。请用SDK的测试demo里的参数值测试，如果需要更改其他测试值，请联系开放平台人员配置。

**问题描述三**: 接口调用返回报OAPIAU0006 验证签名失败或者sdk报com.bocom.api.BocomApiException: sign verify exception.

测试环境报错，一般是第三方改动了demo里的参数 -- MY\_PRIVATE\_KEY和APIGW\_PUBLIC\_KEY 。请不要改动。

如未改动，请检查是否存在以下问题：

1) 请求参数拼接时没有按照顺序拼接；

2) 请求参数拼接时漏拼接URL；

3) mgs\_id传入的值加签前后不一致；

4) sign参数未若未做URLEncode，+(加号传输）之后会变成空格；

5)私钥是否与银行提供的一致（测试环境）；

6)加签报文和发送报文是否一致；

如果生产环境报错，请检查下配置的参数是否已经替换成生产参数--私钥用自己生成的替换，公钥用开放平台提供的生产公钥，APP\_ID 用开放平台提供的appid。其中OAPIAU0006为服务器端报错，交行验证报文签名失败，最可能是客户私钥和交行端的公钥不匹配 ；om.bocom.api.BocomApiException: sign verify exception 为客户端报错，客户验证交行返回报文的签名失败，最可能是客户没有用开放平台生产公钥替换本地参数APIGW\_PUBLIC\_KEY。

**问题描述四:** ADPTSY0009 schema检查异常,请检查报文格式

上传字段不符合文档要求，请注意参照文档中规定的格式和字段长度上传字段。

**问题描述五:** OAPIAU0001 获取应用ID异常

合作方上送错误的appId，与开放平台数据库里面的appId不一致，还有一种情况是交行内部流程还没完全审批通过，appid状态未生效。

处理建议：1、合作方检查下是否用交行给的appid进行了替换；2、联系交行对接的业务老师确认下上线流程是否完成。

**问题描述六:** OAPIAU0003 该应用ID无该接口访问权限

异常原因：

1）api地址错误

2）api地址正确，但app无该api访问权限（未开通该api对应的产品权限）

处理建议：

1. 检查调用的API是否正确，url路径是否正确；
2. 联系交行业务老师确认下所选的API对应产品是否正确，入驻流程是否完成。

**问题描述七:** 调用接口报500错误, 或者报请求地址不对

测试环境以**https://117.184.192.242:9443 为准**

如果DEMO的测试示例代码里面的访问地址是https://117.184.192.242:9443 ，请加入client.ignoreSSLHostnameVerifier(); 这句。生产上把这句话给注释掉。

**问题描述 八**：****生产环境调用接口报500错误, 或者报请求地址不对

生产环境为https://open.bankcomm.com 、https://open.bankcomm.cn

1. 商户端超时，可能商户段超时时间设置比较短（默认5S），而接口时间超过5s
2. 商户端请求未能发出，可能原因包括商户出口有代理服务器，需要通过代理服务器访问公网，需设置代理服务器；商户出口有ip白名单限制，需加上我行域名解析地址；商户如果是集群直连公网，是否某台服务器异常导致部分交易超时，发不出去。

**问题描述 九：**调用接口报错类型区分

接口报错分开放平台报错和业务系统报错两种。开放银行报错的错误码以：“OAPICK”，

“ADPTSY”，”OAPISY”,”OAPIAU” 开头，其他报错码如“CAIP”开头的为后台业务系统报错。测试人员可以找相关人员询问报错原因。

**问题描述 十：**是否有其他版本的SDK

目前因条件限制，只提供java版本的SDK，里面包含接口的bean和调用例子。除java版本外，提供php和C#的一个接口的调用示例，第三方可以先用调用示例跑通接口，然后自行开发其他接口的调用代码。 除以上三种语言外的调用示例，暂时不支持，需要第三方按照java版本的调用示例自行编写调用代码。

**问题描述十一：**具体字段送值询问

涉及业务场景，如字段需要送什么枚举值等问题，可以询问业务老师或后台业务系统技术老师。

**问题描述十二：**交行推送合作方的通知接口，合作方没有收到

1.合作方提供的通知地址不是80或443，导致交行出口网络无法出去；

2.合作方提供的通知地址或域名解析地址是182或139 开头的，和交行内部网络地址冲突，导致交行出口网络无法出去。

3.交行网络可以出去，但合作方网络有限制，或者接受通知的程序有问题，没有打印收到日志，此时要由合作方抓取原始日志查看。

4.还有一种情况，合作方提供的通知地址解析后是CDN地址，CDN不能转发请求到合作方服务器。

**问题描述十三：**商户无法通过bin目录的文件生成公私钥

1. 必须将SDK包解压成文件夹，再点击keygen\_rsa.bat 即可生成。
2. 每次点击keygen\_rsa.bat，都会生成新的一对公钥（yourname.pub） 和 私钥（yourname.pri），检查下生成时间。
3. 若还是点击没有反应，需要检查下windows系统是否按照JDK，可以在cmd中输入 java -version ，出现以下返回说明是安装了java的



若出现以下截图，说明没有安装java，需要安装JDK ，或者登陆到Linux系统，执行keygen\_rsa.sh （Linux系统也是需要安装JDK的）



**问题描述十四：**报错OAPICK0023请求或者二维码超时，请重试

1. 如果是扫二维码报错，那是因为生成的二维码时间超过2小时，目前开放平台生成的H5链接或者二维码有效时间只有2小时；
2. 如果是API接口报错，可能是上送的报文中的timestamp字段的值，为过去的时间，而不是当前时间。

**问题描述十五：**接口返回信息： ADPTSY0002 系统通讯异常、ADPTSY0003 与外围系统建立连接失败、ADPTSY0004 获取外围系统响应超时、ADPTSY0006 与外围系统交互异常。

一般报这种错有以下原因：

1：后台产品服务系统服务未启动，或者在换版；

2：若是SOA服务，可能是后台产品系统在GSP上的WSDL版本和开放银行发布的WSDL不一致

3：若是调用的是分行服务，可能是分行环境问题，分行是TRL环境，开放银行需要配置TRL环境的nginx参数

**问题描述十六：**返回报错“后台繁忙，请稍候尝试”，或者“请求次数过多，请稍候尝试”是什么原因？

该页面或者接口请求达到我行流控的阀值，再次请求会报这个错误，一般情况属于正常报错，就是开放银行设置的限流策略。 若需要调整策略，可以联系线上中心协调。

**问题描述十七：**接口返回报错“timestamp字段格式错误,请按照yyyy-MM-ddHH:mm:ss格式化”是什么原因？

合作方因系统变更或其他原因导致接口变动，上送内容不完整；或者就是timestamp字段不符合格式。另外如果是分行的服务或者SOA服务，在发布接口的时候，还需要关注选择时间函数的校验。

**问题描述十八：**接口返回信息：OAPICK0001 app\_id字段不能为、OAPICK0002 msg\_id字段不能为空、OAPICK0003 fmt\_type字段设置错误、OAPICK0004 is\_encrypt字段设置错误、OAPICK0005 sign字段不能为空、OAPICK0006 timestamp字段不能为空，怎么回事？

很大可能是报文未根据银行文档规范拼接上送，请仔细阅读《金融开放平台接入规范(商户版)V1.1.》自查，建议用银行提供的sdk里面的demo先自行调用，如果需要自己拼接报文，请反编译sdk，按照示例代码编写。

**问题描述十九**：接口返回信息： ADPTSY0002 系统通讯异常、ADPTSY0003 与外围系统建立连接失败、ADPTSY0004 获取外围系统响应超时、ADPTSY0006 与外围系统交互异常，是不是上送参数错误或者程序问题？

可能是由于测试环境存在偶发的通讯异常或者服务重启动等等，遇到这些报错可以隔几分钟再试几次，若问题持续时间较长，请联系银行对接人员支持。