

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

1 前言	5
1.1 编写目的	5
1.2 读者对象	5
2 金融开放平台概述	5
2.1 简介	5
2.2 输出模式	5
3 接入指南	5
3.1 接入流程	5
3.2 合作方开发准备	6
3.2 联调环境测试	6
3.3 投产	6
4 接口规范	8
4.1 字符集和编码	8
4.2 通讯规范	8
4.3 业务请求报文结构	8
4.4 文件请求报文结构	9
4.5 返回报文结构	10
4.6 业务状态字	10
4.7 报文签名规则	11
4.8 报文加密规则	13
5 调用过程	14
5.1 API 模式调用	14
5.2 H5 模式调用	15
6 SDK 使用说明	15
6.1 结构说明	16
6.2 资源引用	16
6.3 签名密钥生成	17
6.4 加密密钥生成(目前一般不用)	错误！未定义书签。
6.5 网关公钥	17
6.6 网络访问权限	17
6.7 接口调用示例	18
6.7.1 API 调用对象说明	18
6.7.2 API 调用示例	18
6.7.3 通知接口调用示例	21
6.7.4 文件传输	24
6.8 返回信息说明	28
6.8.1 返回说明	28
6.8.2 返回判断示例	28
6.8.3 返回信息列表	29
9 附录	31
9.1 域名解析地址	31

9.2 Q&A.....	错误！未定义书签。
--------------	-----------

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互生产网关的公钥 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>

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	是否必填	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	

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.
4.      https://open.95559.com.cn/api/popentest/test?app_id=app2018110513
      49&msg_id=1adc3436052e4496b2afa34e1eee446f&format=json&charset=utf-8&si
      gn=TXxFR9t80R32ZHtro45IrAr0Gdjo+niy6kCcz+phTT5V6TixL+ZH9cAEDTuB/4IngWOC
      6So6DzeSL7J+bP16ky+/wzwwkeUXgwpL0+SMz07DpdPaWl1vkFL2iVFIs1LMUPcYiYdcZ7y
      VJ02mOhElcAiD0r1XmFQnqI8T0mqWrLiZ2GR7NNOHb+DaexyNt9jdTxH0ggnmBM21DKHfJ
      uS5N7WgLLwTNgNcKaGkUyRgddcaCcHpSKPRi518CghkN3JEE04mZ4W9V1zbrYDyq+FcwE5j
      soBq1rIP00tECH40w9dgspRxie4aiVcC8ScwJQ7EYxt6g5VP1f530GJmR1nx/g==&timest
      amp=2016-10-29 20:44:38&biz_content=
5.      {
6.      "cust_id":"ABC123456",
7.      "cust_name":"TEST",
8.      }

```

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	是否必填	最大长度	描述	示例值
rsp_biz_content	str	true	20	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":"ckVL6FMpSPokuqEzpaA02xrQAQZkgpn3JwJLf0Ig3smC3P2y5odhZ9I1c
      mNHD8wMCUOSKwVBFpv2gpsx+s70Xqpjsi0Z3i4ibwHUSX00BPYtuLxJ/wlTo1q9B36mRf
      j54UT6Uwr10V1s20c+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	20	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。

② 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。

③ 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。

④ 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":"ABCD AADB","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
```

✧ 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":"ckVL6FMpSPokuqEzpA02xrQAQZkgpn3JwJLf0Ig3smC3P2y5odhZ9I1c
      mNHD8wMCUOSKWvVBFpv2gpsx+s70XqpjsiOZ3i4ibWHUSX00BPYtuLxJ/wlTolq9B36mRf
      j54UT6Uwr10V1s20c+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.      }
```

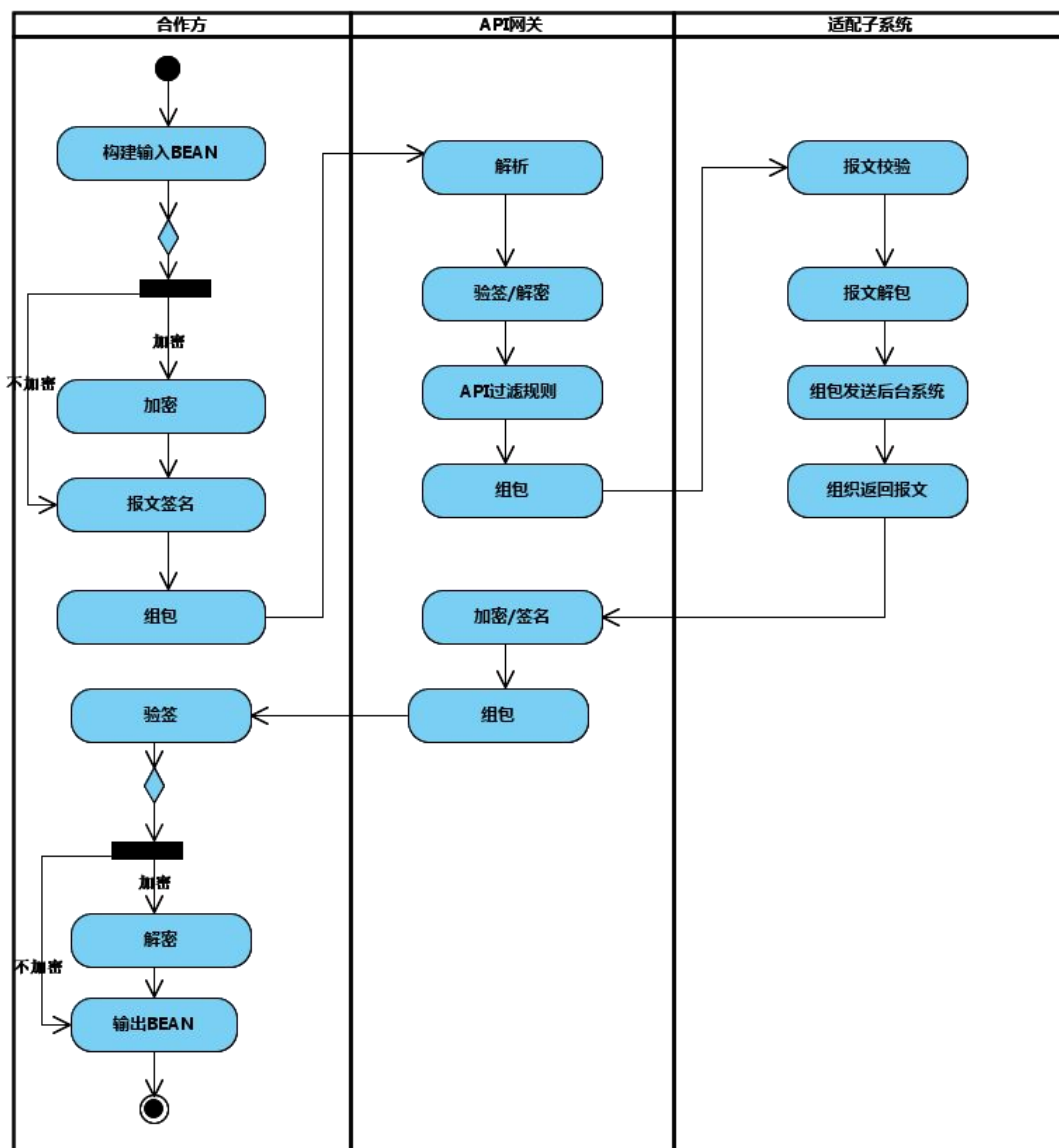
4.9 报文加密规则

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。

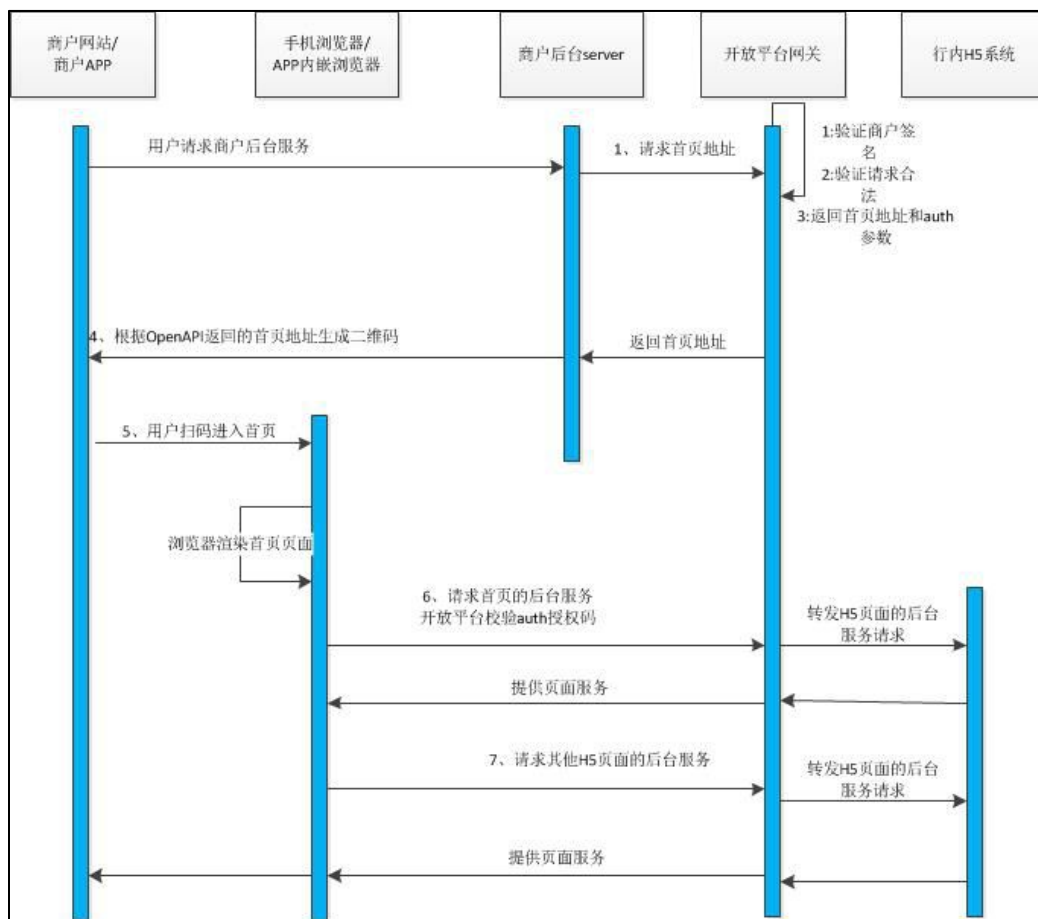
5 调用过程

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.cn> 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）：

```
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEASJrF8T/5rKB4NnwMjIxUer+ELf1PQX02GSdZ/fvuQCc10R9tB1NWL4jF0ftebeL+bvMV0J+JHm/aSes1AN8YNIDGiFUpF6aDkSCaLyrDdjK/mQTWhSNa2f00GG0+yw0BTdYUjVjVtzJ48bbyG3NSy1f1EdnBwnMpFa8qpXJXR4ELpVpMkPDC+93HBA1xEgUjhcIJlP5VdKIiudsmhE2T07qtpIQSuE5hntXP6X6GKJReCk+yek2QJITvIBc3cHPw8KDsHHCs7MaR81KI3onJRWyqFtTfVYTiKsd9EcYSxv+Gx5M0F8B/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 bocomPublicKey	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_JS
    ON, APIGW_PUBLIC_KEY, BocomConstants.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/t
est/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/t
est/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());
    }
}
```

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	是否必填	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": "HaqN9k9mbxYVCAZj10XREpx6Pp1v1T061TMLHC7CZkfQv14Lgxo1jcw3/XCwd6I2Ay88hu  
g5Mz5wyr8wUQfN3dCT2YCKVALsZoMMS7xX6j/8s2Ian0Yjgo3+kf0qEEEx0Ws4uGDdWva48zzGyVc08DmV2  
bV2utGmnMROes3FuzQp/JAp/OT1bdldAVZNcK+ZLD0uW9I2lAUg9YNFjA1HixCRdAh0vTmTZp8D0oK23P85  
hJ7x2AMwU2aDzpAoBIroMHEFi+vSYbK0TLIUltzUYy7KmVDIrXD7HiUnJmU5E511E+zZxBzmnMpJikPEv/x  
xY5cYCwDDMiJzcwa+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。

```

public class TradeCallbackV1Test {

    public static final String APIGW_PUBLIC_KEY =
"MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAXSjrF8T/5rKB4NnwMjIxUer+ELf1PQX02GSdZ/
fVuQCc10R9tB1NWL4jF0ftebeL+bvMVOJ+JHm/aSes1AN8YNIDGiFUpF6aDkSCaLynDdjK/mQTWhSNa2f00
GG0+ywOBtdYUjVjVtzJ48bbyG3NSylf1EdnBwnMpFa8qpXJXR4ELpVpMkPDC+93HBA1xEgUjhcIJ1P5VdKI
iudsmhE2T07qtpIQSuE5hntXP6X6GKJReCk+yek2QJITvIBq3cHPw8KDsHHCs7MaR81KI3onJRWyqFtTfVY
TiKsd9EcYSxv+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\",\"currenc
y\":\"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\":\"30191
0053990034\",\"req_time\":\"20190723093745\"},\"sign\":\"gw2z0qjTci5BEq8NNY1MeZoBFE
M20W0UfgEcXIIsIjKiZn0aL8WoLiJApphB7FaS3NEW3J/O9pM4xJOeuHuUQGC7cS//lBd5vkOp2I8gmcpRj
xkV5ma6GTVa5srgHk9ftF8e2Q1HaFSzGZ5foc+53Vt98/kudTBdcE0rOoJD/Drxy4AIgVgH2sSsIk7MTCQ
CEicMNOMkKc6NEY+abqWkisTBPpfDBY6tyZAcQ6e5liqQQ/yCiieG1SoQJpgz17nxh6ZoQj5364Igw7JXfx
oyDJuC8qZoeMqqHVIomS5kqtWiT/6MCoAvqfhNnKiaI0dgxYtzreFKa1mn0kze+C0mg==\"}";

            TradeCallbackResponseV1 response = (TradeCallbackResponseV1)
ApiUtils.parseNotifyJsonWithBocomSign(notifyJson, "UTF-8", APIGW_PUBLIC_KEY,
TradeCallbackResponseV1.class);

            System.out.println(response);
        } catch (Exception xcp) {
            xcp.printStackTrace();
        }
    }
}

```

● 7.3 响应报文

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，

```

8.      {
9.      "biz_state": "S",
10.     "rsp_code": "0000",
11.     "rsp_msg": "success",
12.     }

```

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+LYQVhuXQsKVYvcwXLFudXzqwi5b14vFugbX0wBhJtz9gLwEhQBMw0sKpTvuaQzN7hLx0ES9sDiThEuTA00VhQGJDG8It0fwn2qzvnvQsZvewZxSovwoRtf2PE8aD6JZkTCXwjpHqEJyr7QAPtxubwo5YwXfH1w85vRRl0scGF0107okvC9q1bUN4oL4LvEky0tKSKDaZk88Gwiu6t9l0gnreH5fovdouScRHZMzZ+R+gi5/4a5s5bp6IVxxLDoSL68S0SU3adunJjPI1VpWRD0C1mLGjBhwZMhNJ3a0QLYunr1Z54NP0UPGXNTN16aQzWk3cV2shkfzbJ3D4zT5b1Ce9LMnUw+4G+cxtYqB1b7iS8V45Lf0Y4/rfbkt2eKWVabiVMYi9qf+VSBWU0bgl9Dkk0nAD8=",
  "msg_id": "fb4729ab9b5740b2ad6f8703ec6bac24",
  "timestamp": "2020-10-04 19:00:17",
  "encrypt_key": "EhEKX32UjHrWDOuoTBN4nU8sdxixuFQEht8YqNpeBSPp6LldPucr/+pXNwu/e2EwDhf5TVpJBxvUR8HHF4jBfrwmUeG4ygCeeCJ6JL/ayK9U1oxOXCD4z9m4PMvhDATas12qH4DHu6nyjDg+zzs+k7/SqhhWxiUBfN4+TprUpT4YyaxYjTHqftPiU2eKvq8pGebZ3jNHZpaeEhbEa2h/n9u/0/HZ0BDMLbk6RvB9WtnvQzFKGDWgvFGHhTmb8qt7lodXEIOZ/GpSXpXeyrxrCOWuGOLgrx1HTzi4AD2TqNLGIQGWNeoz1iaefTtFrhwfgzmzrJmtzQUyOIm7p4gkw=",
  "sign": "EZ77fRoQYVJGyo0KETGmsjKy/7/wawmCNxmmzx4Gk6g60/Ko/2nKPnZ5iNYeMhbdftFxLnq8sX+g9qLDP7+zD1apNFsuBrpc6H14xpeD57MZggBnanM74aztrER"
}
```



```
OAZ6zc1jfpb+b7TNRmNYb63wD6ZPwclpmbh2GTHIOVrBSJnHgFIiJf4iSM7SPCTstfx02/qYq3jPVpvoysz3qpjjRJ8H2uuF7fqMol1rPolRf4lhjj+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": "nPxMcEfQqCAoRcTY/39d7LVpZglBGQPhnY1HDS03nJ1iPm/i9bPxHkiDrqpmTjuS+EmPr2TVbP7X1+LYQVhuXQsKVYvcwLxFudXzqwi5bl4vfugbX0wBhJtz9gLwEhQBMwOsKpTvuaQzN7hLx0ES9sDiThEuTA00VhQGJDG8ItofwN2qzNVQSzvwZxSovwoRtf2PE8aD6JZkTCXwjpHqEJyr7QAPtxubwo5YwXfH1w85vRRloscGF0107okvC9q1bUN4oL4LvEky0tKSKDaZk88Gwiu6t9l0gnreH5fovdouScRHZMzZ+R+gi5/4a5s5bp6IVxxLDoSL68S0SU3adunJjPI1VpWRD0C1mLgJbHwZMhN3a0QLYunr1Z54NP0UPGXNTN16aQzWk3cV2shkfzbJ3D4zT5b1Ce9LMnUw+4G+cxtYqB1b7iS8V45Lf0Y4/rfbkt2eKWVabiVMYi9qf+VSBWU0bgL9DkkOnAD8=", "msg_id": "fb4729ab9b5740b2ad6f8703ec6bac24", "timestamp": "2020-10-04 19:00:17", "encrypt_key": "EhEKX32UjHrWD0uoTBN4nU8sdxixuFQEht8YqNpeBSPp6LldPucr/+pXNwu/e2EwDhf5TVpJBxvUR8HHF4jBfrwmUeG4ygCeeCJ6JL/ayK9U1oxOXCD4z9m4PMvhDATas12qH4DHu6nyjDg+zzs+k7/SqhhWxiUBfN4+TpRUPT4YyaxYjTHqftPiU2eKvq8pGebZ3jNHZpaeHbEa2h/n9u/0/HZ0BDMLbk6RvB9WTnvQzFKGDwGvfGHHtmb8qt7lodXEIOZ/GpSXpXEyrxCOWuGOLgrx1HTzi4AD2TqNLGIQGWNeoz1iaeFTtFrhWfgzmzrJmtzQUyOI7p4gkw=="
```

Sign

```
"sign": "EZ77fRoQYVJGyo0KETGmsjKy/7/wawmCNxmxz4Gk6g60/Ko/2nKPnZ5iNYeMhbDftFxLnq8sX+g9qLDP7+zDlapNFsuBrpc6H14xpeD57MZggBnanM74aztrEROAZ6zc1jfpb+b7TNRmNYb63wD6ZPwclpmbh2GTHIOVrBSJnHgFIiJf4iSM7SPCTstfx02/qYq3jPVpvoysz3qpjjRJ8H2uuF7fqMol1rPolRf4lhjj+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/vpXj3aT1Lo033UhSb/LoF8akPjo3r3UajR9x/8NS1KWq4YPX73ZjhLCECaJ1m26Yysb1S5d5bC00bDS6Gb4jvKogysLomuGyuaCoBb+yXi2JwwxRvKKkSySCPHEzigvUf1d9r90yezAoLFaQD0WtxJWwhoLsUf2sHf6LLvPFS5ilumThgh4Hh00NSz3yuNFmaw/A/pf+QnMOo0IFpzgq/6gfFApMwq03MYQB9rfqhwwq0/bxhZE+4ZOEJD0gNQcikbswt3gV0vD8Uk1ncdQ6WeD0BciSGoDm91AJh4obYenHM/4f1H2LBA=="}
```

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.ISVCheckFileDownloadRe  
questV1Biz fileDownloadRequestV1Biz = new  
ISVCheckFileDownloadRequestV1.ISVCheckFileDownloadReques  
tV1Biz();  
  
    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 <code>ContentType</code>
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	<code>session</code> 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 <code>IP</code>
OAPICK0023	请求或者二维码超时，请重试
OH5TK0001	<code>TokenId</code> 创建失败
OH5TK0002	<code>TokenId</code> 已失效
OH5TK0003	会话已失效
OH5TK0004	无法获取客户端 <code>TokenId</code>
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	<code>Encode</code> 请求报文发生异常
ADPTSY0008	<code>Decode</code> 响应报文发生异常
ADPTSY0009	<code>schema</code> 检查异常, 请检查报文格式
ADPTSY0010	关键字 <code>biz_state</code> 重复
ADPTSY0011	二维码参数生成失败
ADPTSY0012	获取请求报文异常
ADPTSY0013	<code>JsonSchema</code> 中 <code>Func</code> 未定义
ADPTSY0014	<code>JsonSchema</code> 中 <code>Func</code> 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：

- a) 检查调用的 API 是否正确，url 路径是否正确；
- b) 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。

2. 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

```
C:\Users\Administrator>java -version
java version "1.8.0_152"
Java(TM) SE Runtime Environment (build 1.8.0_152-b16)
Java HotSpot(TM) 64-Bit Server VM (build 25.152-b16, mixed mode)
```

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）

```
C:\Users\kuanglei>java -version
'java' 不是内部或外部命令，也不是可运行的程序
或批处理文件。
```

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，或者在换代；

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阈值，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。