编号：

商户接入汇收付 QA 文档

版本修订管理

版本号	时间	作者	说明
V1.0	2017 年 10 月 19 日	汪令银	第一版
V1.0.1	2017 年 10 月 30 日	汪令银	添加汇收付前置 IP 地址 修改生产环境公网地址 新增 4.3.2、4.3.3
V1.0.2	2017 年 11 月 02 日	汪令银	新增 4.8、4.9、4.10
V1.1	2017 年 11 月 06 日	汪令银	修改测试环境端口
V1.2	2017 年 11 月 21 日	汪令银	添加 2.6.1.3、2.6.1.4 添加 4.9.4
V1.3	2017 年 12 月 05 日	汪令银	更新测试卡号
V1.3.1	2018 年 01 月 02 日	汪令银	修改查询结果判断逻辑
V1.4	2018 年 01 月 15 日	汪令银	添加注意事项
V1.5	2018 年 02 月 02 日	汪令银	基金接口添加两个接口：基金可用额度查询、平垫资查询；签名版本后面不再支持 1.0.1。
V1.6	2018 年 03 月 26 日	鲁惠贤	1）2.6.1.1 和 2.6.1.2 修改测试环境的端口和 ip 2）2.2 修改签名方式仅支持 RSA，敏感字段需加密 3）3.3 新增 sadk jar 包下载方式 4）新增 3.3.4、3.3.5、3.3.6、3.3.7 四个章节 5）新增 4.4.3、4.4.4 两个章节
V1.7	2018 年 05 月 29 日	鲁惠贤	更新了 2.2、2.4 和 4.5 小节 4.9.2 新增查询无此订单说明
V1.8	2018 年 06 月 19 日	鲁惠贤	1、修改 3.3 关于 sdk 中 jar 包的获取方式 2、修改 3.4 节测试卡号
V1.9	2018 年 08 月 31 日	鲁惠贤	1、修改 sdk 参考示例
V2.0	2018 年 09 月 29 日	鲁惠贤	1.新增 3.3.1 注意事项，即红色字体标注的 1 和 2 两段内容 2.新增 3.3.9 秘钥串解密方法 3.新增 3.3.10 退汇流水文件、结算户流水文件 AES 解密方法
V2.1	2018 年 11 月 16 日	鲁惠贤	1.删除原 4.3.1 小节内容 2.新增 1.1 接口文档升级的说明
V2.2	2018 年 12 月 12 日	鲁惠贤	1.新增 3.5.3 和 3.5.4 两个小节，关于对账文件下载新旧接口的说明；后续将不再支持非加密的对账文件下载接口
V2.3	2019 年 03 月 18 日	鲁惠贤	1.新增对账文件下载接口（文件内容加密）请求路径 2.批量查询交易新增批次状态 3-处理失败 3.批量查询交易，结果文件新增统计成功笔数、成功金额
V2.4	2019 年 05 月 10 日	鲁惠贤	1.细化 4.3.2 小节商户流水号规则说明
V2.5	2019 年 10 月 09 日	范卫涛	1.添加协议签约测试环境及生产环境请求 URL

1. 文档说明 1

此文档由兴业银行汇收付产品小组编写，提供给商户接入兴业银行汇收付系统时参考。文档包含三个部分，第一部分为商户接入汇收付系统步骤，第二部分为汇收付给商户提供的开发工具包，以便商户开发使用，第三部分为常见问题解答。兴业银行汇收付系统简称汇收付。

1.1关于接口文档升级说明

接口文档中部分接口升级改造，商户版接口文档从版本 C1.2.7 开始，基金版接口文档从版本 F2.2.9 开始。存量商户可以忽略此次修改，无需改动，因为本次接口升级兼容以前版本。

涉及升级改造的接口包括：

账户认证：删除 bankNo 和 bankName 字段

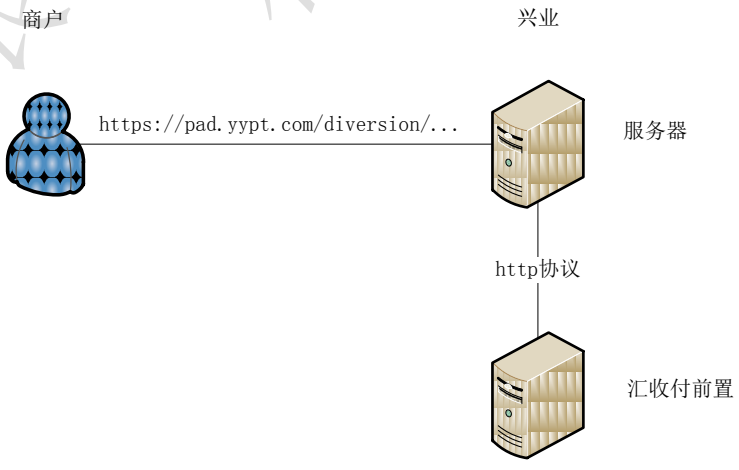
单笔代收：新增 bankCode 字段

2. 商户如何接入汇收付（接入步骤）

2.1第一步：选择通信网络

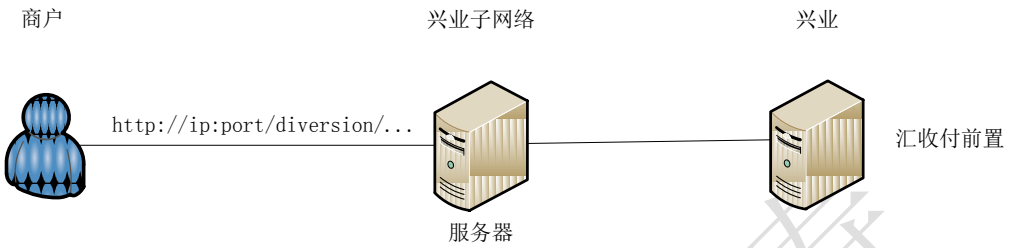
商户需要先选择通过公网还是通过专线访问汇收付系统。（选择其一，情况允许可以两者都接通）

2.1.1 公网（https 协议）



如果商户以公网方式接入兴业汇收付系统，使用 https 协议，直接访问 https://pad.yypt.com/diversion/...，后面加具体接口对应的地址。

2.1.2 专线（http 协议）



如果商户以专线方式接入兴业汇收付系统，首先需要接通兴业的子网络，子网络包括商户所在地兴业分行、兴业子公司等，该步骤需要商户和兴业子网络完成。兴业子网络需要跟兴业总公司接通网络，此步骤由兴业子网络向兴业总公司申请。

网络接通后，商户直接访问 http://ip:port/diversion/...调用汇收付服务，地址中的 ip、port 是兴业子网络给商户提供的。

兴业子网络接生产环境汇收付前置地址：IP：168.7.19.8　port：8080

兴业子网络接测试环境汇收付前置地址：IP：168.7.62.13　port：8080（UAT 环境）

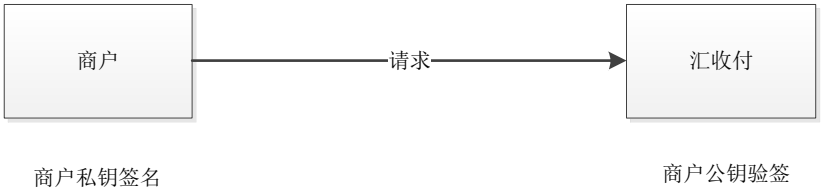
2.2第二步：签名方式及敏感字段脱敏

为了增加安全性，**要求所有商户签名方式选择 RSA（所需证书需要商户自行向 CFCA 等机构采购），敏感字段需脱敏。**

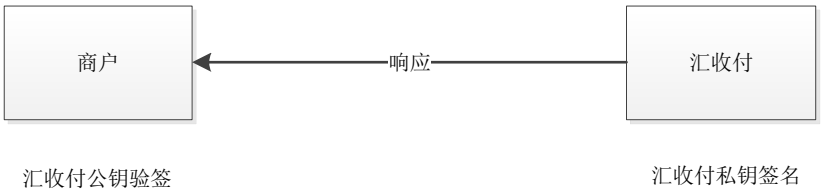
2.2.1.1 RSA 签名

RSA，需要用到商户证书。

a) 商户发送的请求信息用自己的私钥签名，汇收付收到请求后，用商户的公钥验签。

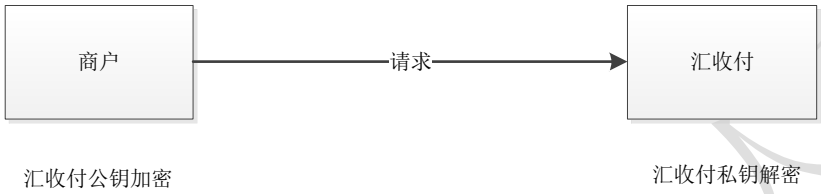


b) 汇收付给商户的响应信息用汇收付私钥签名，商户收到响应后，用汇收付的公钥验签。



2.2.1.2 字段脱敏（加密）

加密证书使用汇收付公钥。



2.3第三步：商户获得自己的证书

2.3.1 测试环境（SDK 中有提供）

merchant001-rsa.pfx（私钥）、pubMerchant001-rsa.cer（公钥）

2.3.2 生产环境

- **自行向 CFCA 等机构采购，并将公钥发送汇收付业务人员。**

2.4第四步：开通商户号及获取汇收付公钥

2.4.1 测试环境

测试申请提交后，将由汇收付联调支持人员分配测试环境商户号。

汇收付测试环境公钥证书：
在 SDK 中，/cib-yypay-sdk/src/main/java/prod/yypay/sdk/key/pubServer-rsa.cer

2.4.2 生产环境

完成商户入网申请后，将由业务人员发送正式商户开通邮件，包括商户号、公钥证书等信息。

2.5第五步：商户调用接口发送交易

2.5.1 测试环境

2.5.1.1 普通商户(公网)

账户认证	http://139.198.176.149:16004/diversion/merchant/acctAuth
短信验证	http://139.198.176.149:16004/diversion/merchant/common/smsCheck
单笔代收	http://139.198.176.149:16004/diversion/merchant/single/collect
单笔代付	http://139.198.176.149:16004/diversion/merchant/single/pay
单笔代收查询	http://139.198.176.149:16004/diversion/merchant/single/collectQuery
单笔代付查询	http://139.198.176.149:16004/diversion/merchant/single/payQuery
批量代收	http://139.198.176.149:16004/diversion/merchant/batch/collect
批量代付	http://139.198.176.149:16004/diversion/merchant/batch/pay
批量查询	http://139.198.176.149:16004/diversion/merchant/batch/queryBatchStatus
子商户列表查询	http://139.198.176.149:16004/diversion/common/childInsts
对账	http://139.198.176.149:16004/diversion/merchant/file/reconc
商户信息查询	http://139.198.176.149:16004/diversion/common/merchant
获取银行信息	http://139.198.176.149:16004/diversion/common/getBankInfo
退汇流水文件下载	http://139.198.176.149:16004/diversion/merchant/file/refund
结算户流水文件下载	http://139.198.176.149:16004/diversion/merchant/file/provisions
对账文件下载（文件内容加密）	http://139.198.176.149:16004/diversion/merchant/file/ checkFile
账户协议签约	http://139.198.176.149:16004/diversion/merchant/agreementSign

2.5.1.2 基金公司(公网)

账户认证	http://139.198.176.149:16004/diversion/merchant/acctAuth
短信验证	http://139.198.176.149:16004/diversion/merchant/common/smsCheck
单笔代收	http://139.198.176.149:16004/diversion/fund/single/collect
单笔代付	http://139.198.176.149:16004/diversion/fund/single/pay
单笔代收查询	http://139.198.176.149:16004/diversion/fund/single/collectQuery

单笔代付查询	http://139.198.176.149:16004/diversion/fund/single/payQuery
批量代收	http://139.198.176.149:16004/diversion/merchant/batch/collect
批量代付	http://139.198.176.149:16004/diversion/merchant/batch/pay
批量查询	http://139.198.176.149:16004/diversion/merchant/batch/queryBatchStatus
子商户列表查询	http://139.198.176.149:16004/diversion/common/childInsts
对账	http://139.198.176.149:16004/diversion/merchant/file/reconc
商户信息查询	http://139.198.176.149:16004/diversion/common/merchant
获取银行信息	http://139.198.176.149:16004/diversion/common/getBankInfo
基金可用额度查询	http://139.198.176.149:16004/diversion/fund/availableCreditQuery
平垫资查询	http://139.198.176.149:16004/diversion/fund/repaymentQuery
退汇流水文件下载	http://139.198.176.149:16004/diversion/merchant/file/refund
结算户流水文件下载	http://139.198.176.149:16004/diversion/merchant/file/provisions
对账文件下载（文件内容加密）	http://139.198.176.149:16004/diversion/merchant/file/ checkFile
账户协议签约	http://139.198.176.149:16004/diversion/merchant/agreementSign

2.5.1.3 普通商户(专线)

账户认证	http://ip:port/diversion/merchant/acctAuth
短信验证	http://ip:port/diversion/merchant/common/smsCheck
单笔代收	http://ip:port/diversion/merchant/single/collect
单笔代付	http://ip:port/diversion/merchant/single/pay
单笔代收查询	http://ip:port/diversion/merchant/single/collectQuery
单笔代付查询	http://ip:port/diversion/merchant/single/payQuery
批量代收	http://ip:port/diversion/merchant/batch/collect
批量代付	http://ip:port/diversion/merchant/batch/pay
批量查询	http://ip:port/diversion/merchant/batch/queryBatchStatus
子商户列表查询	http://ip:port/diversion/common/childInsts
对账	http://ip:port/diversion/merchant/file/reconc
商户信息查询	http://ip:port/diversion/common/merchant
获取银行信息	http://ip:port/diversion/common/getBankInfo
退汇流水文件下载	http://ip:port/diversion/merchant/file/refund
结算户流水文件下载	http://ip:port/diversion/merchant/file/provisions
对账文件下载（文件内容加密）	http://ip:port/diversion/merchant/file/checkFile
账户协议签约	http://ip:port/diversion/merchant/agreementSign

（请求路径中的 ip 和 port 是兴业子网络给商户提供的）

兴业子网络接测试环境汇收付前置地址：IP：168.7.62.13　port：8080（UAT 环境）

2.5.1.4 基金公司(专线)

账户认证	http://ip:port/diversion/merchant/acctAuth
短信验证	http://ip:port/diversion/merchant/common/smsCheck
单笔代收	http://ip:port/diversion/fund/single/collect
单笔代付	http://ip:port/diversion/fund/single/pay
单笔代收查询	http://ip:port/diversion/fund/single/collectQuery
单笔代付查询	http://ip:port/diversion/fund/single/payQuery
批量代收	http://ip:port/diversion/merchant/batch/collect
批量代付	http://ip:port/diversion/merchant/batch/pay
批量查询	http://ip:port/diversion/merchant/batch/queryBatchStatus
子商户列表查询	http://ip:port/diversion/common/childInsts
对账	http://ip:port/diversion/merchant/file/reconc
商户信息查询	http://ip:port/diversion/common/merchant
获取银行信息	http://ip:port/diversion/common/getBankInfo
基金可用额度查询	http://ip:port/diversion/fund/availableCreditQuery
平垫资查询	http://ip:port/diversion/fund/repaymentQuery
退汇流水文件下载	http://ip:port/diversion/merchant/file/refund
结算户流水文件下载	http://ip:port/diversion/merchant/file/provisions
对账文件下载（文件内容加密）	http://ip:port/diversion/merchant/file/checkFile
账户协议签约	http://ip:port/diversion/merchant/agreementSign

（ip 和 port 是兴业子网络给商户提供的）

兴业子网络接测试环境汇收付前置地址：IP：168.7.62.13　port：8080（UAT 环境）

2.5.2 生产环境

2.5.2.1 普通商户(公网)

账户认证	https://pad.yypt.com/diversion/merchant/acctAuth
短信验证	https://pad.yypt.com/diversion/merchant/common/smsCheck
单笔代收	https://pad.yypt.com/diversion/merchant/single/collect
单笔代付	https://pad.yypt.com/diversion/merchant/single/pay
单笔代收查询	https://pad.yypt.com/diversion/merchant/single/collectQuery
单笔代付查询	https://pad.yypt.com/diversion/merchant/single/payQuery
批量代收	https://pad.yypt.com/diversion/merchant/batch/collect
批量代付	https://pad.yypt.com/diversion/merchant/batch/pay
批量查询	https://pad.yypt.com/diversion/merchant/batch/queryBatchStatus
子商户列表查询	https://pad.yypt.com/diversion/common/childInsts
对账	https://pad.yypt.com/diversion/merchant/file/reconc

商户信息查询	https://pad.yypt.com/diversion/common/merchant
获取银行信息	https://pad.yypt.com/diversion/common/getBankInfo
退汇流水文件下载	https://pad.yypt.com/diversion/merchant/file/refund
结算户流水文件下载	https://pad.yypt.com/diversion/merchant/file/provisions
对账文件下载（文件内容加密）	https://pad.yypt.com/diversion/merchant/file/checkFile
账户协议签约	https://pad.yypt.com/diversion/merchant/agreementSign

2.5.2.2 基金公司(公网)

账户认证	https://pad.yypt.com/diversion/merchant/acctAuth
短信验证	https://pad.yypt.com/diversion/merchant/common/smsCheck
单笔代收	https://pad.yypt.com/diversion/fund/single/collect
单笔代付	https://pad.yypt.com/diversion/fund/single/pay
单笔代收查询	https://pad.yypt.com/diversion/fund/single/collectQuery
单笔代付查询	https://pad.yypt.com/diversion/fund/single/payQuery
批量代收	https://pad.yypt.com/diversion/merchant/batch/collect
批量代付	https://pad.yypt.com/diversion/merchant/batch/pay
批量查询	https://pad.yypt.com/diversion/merchant/batch/queryBatchStatus
子商户列表查询	https://pad.yypt.com/diversion/common/childInsts
对账	https://pad.yypt.com/diversion/merchant/file/reconc
商户信息查询	https://pad.yypt.com/diversion/common/merchant
获取银行信息	https://pad.yypt.com/diversion/common/getBankInfo
基金可用额度查询	https://pad.yypt.com/diversion/fund/availableCreditQuery
平垫资查询	https://pad.yypt.com/diversion/fund/repaymentQuery
退汇流水文件下载	https://pad.yypt.com/diversion/merchant/file/refund
结算户流水文件下载	https://pad.yypt.com/diversion/merchant/file/provisions
对账文件下载（文件内容加密）	https://pad.yypt.com/diversion/merchant/file/checkFile
账户协议签约	https://pad.yypt.com/diversion/merchant/agreementSign

2.5.2.3 普通商户(专线)

账户认证	http://ip:port/diversion/merchant/acctAuth
短信验证	http://ip:port/diversion/merchant/common/smsCheck
单笔代收	http://ip:port/diversion/merchant/single/collect
单笔代付	http://ip:port/diversion/merchant/single/pay

单笔代收查询	http://ip:port/diversion/merchant/single/collectQuery
单笔代付查询	http://ip:port/diversion/merchant/single/payQuery
批量代收	http://ip:port/diversion/merchant/batch/collect
批量代付	http://ip:port/diversion/merchant/batch/pay
批量查询	http://ip:port/diversion/merchant/batch/queryBatchStatus
子商户列表查询	http://ip:port/diversion/common/childInsts
对账	http://ip:port/diversion/merchant/file/reconc
商户信息查询	http://ip:port/diversion/common/merchant
获取银行信息	http://ip:port/diversion/common/getBankInfo
退汇流水文件下载	http://ip:port/diversion/merchant/file/refund
结算户流水文件下载	http://ip:port/diversion/merchant/file/provisions
对账文件下载（文件内容加密）	http://ip:port/diversion/merchant/file/checkFile
账户协议签约	http://ip:port/diversion/merchant/agreementSign

（请求路径中的 ip 和 port 是兴业子网络给商户提供的）

兴业子网络接生产环境汇收付前置 IP 地址：IP：168.7.19.8　port：8080

2.5.2.4 基金公司(专线)

账户认证	http://ip:port/diversion/merchant/acctAuth
短信验证	http://ip:port/diversion/merchant/common/smsCheck
单笔代收	http://ip:port/diversion/fund/single/collect
单笔代付	http://ip:port/diversion/fund/single/pay
单笔代收查询	http://ip:port/diversion/fund/single/collectQuery
单笔代付查询	http://ip:port/diversion/fund/single/payQuery
批量代收	http://ip:port/diversion/merchant/batch/collect
批量代付	http://ip:port/diversion/merchant/batch/pay
批量查询	http://ip:port/diversion/merchant/batch/queryBatchStatus
子商户列表查询	http://ip:port/diversion/common/childInsts
对账	http://ip:port/diversion/merchant/file/reconc
商户信息查询	http://ip:port/diversion/common/merchant
获取银行信息	http://ip:port/diversion/common/getBankInfo
基金可用额度查询	http://ip:port/diversion/fund/availableCreditQuery
平垫资查询	http://ip:port/diversion/fund/repaymentQuery
退汇流水文件下载	http://ip:port/diversion/merchant/file/refund
结算户流水文件下载	http://ip:port/diversion/merchant/file/provisions
对账文件下载（文件内容加密）	http://ip:port/diversion/merchant/file/checkFile
账户协议签约	http://ip:port/diversion/merchant/agreementSign

（ip 和 port 是兴业子网络给商户提供的）

兴业子网络接生产环境汇收付前置 IP 地址：IP：168.7.19.8　port：8080

3. 开发工具包（如果缺少联系汇收付业务人员发送）

3.1商户号

商户调用汇收付交易接口时，其中 mchtId 传此商户号。
测试环境和生产环境分别开通，联系业务人员确认商户号。

3.2接口文档

接口开发参考文档（接口开发以此文档为准）

3.3sdk 压缩包

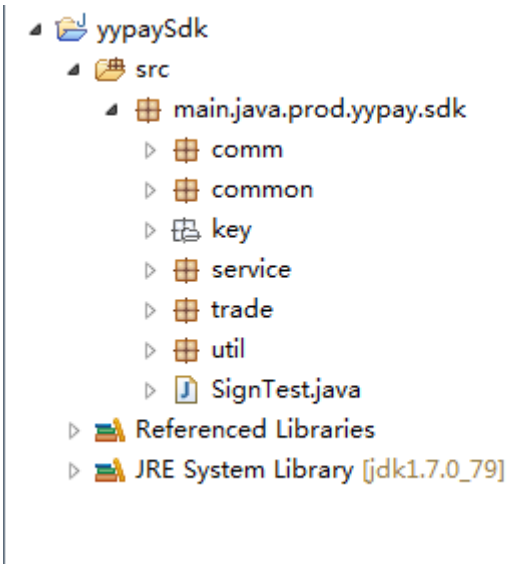
3.3.1 注意事项：

1.接收到的 sdk 版本号<=1.1 的商户：若需调用下载退汇流水文件或结算户流水文件接口，需申请最新版本的 sdk，且只需关注 3.3.9 秘钥串解密方法和 3.3.10AES 解密方法，具体代码参考最新版本的 sdk 中 Signature.java 中的 Signature.decoderByCFCA(aesKey, String, String) 方法和 Signature.AESDecode(String, String)方法；

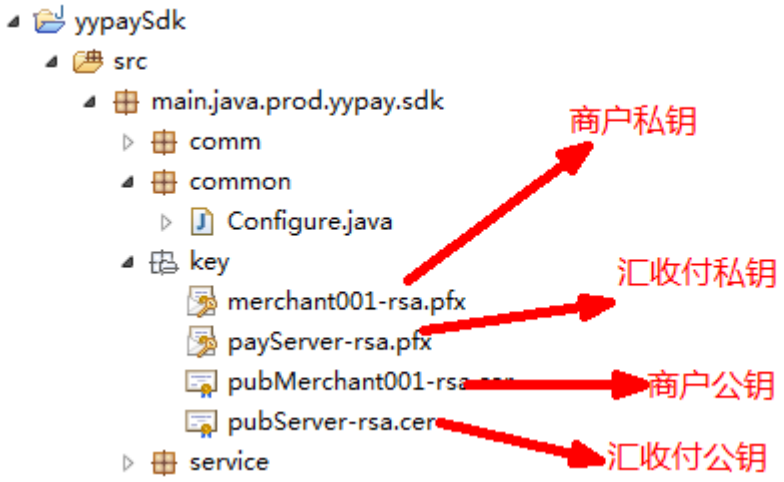
2.接收到的 sdk 版本号>=1.2 的商户：可申请最新版本的 sdk，具体代码参考最新版本的 sdk。

字段脱敏需使用 CFCA 的 sadk.jar，联系分行人员获取。

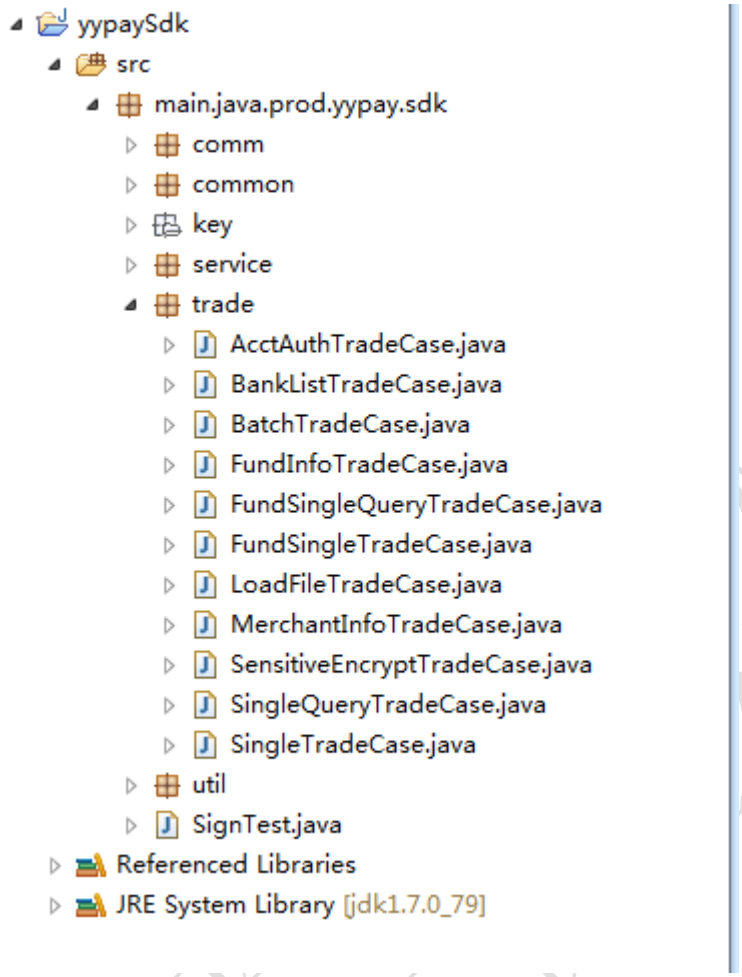
sdk 压缩包是一个 java 项目，供商户的开发人员参考，具体接口开发规范以提供的接口文档为准。
sdk 结构如下



3.3.2 公私钥在 key 文件夹中

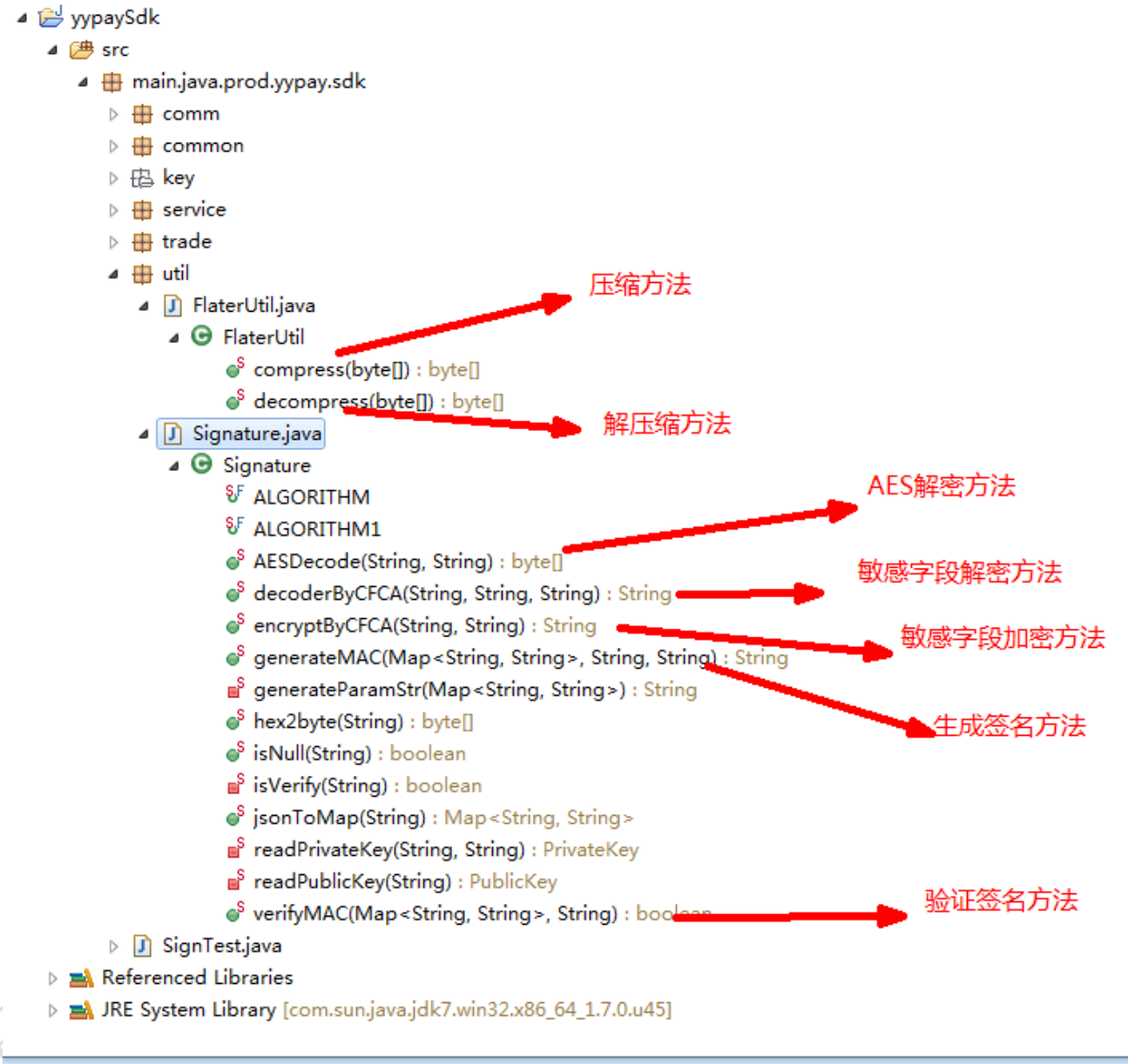


3.3.3 trade 文件夹提供了每个接口的调用示例



AcctAuthTradeCase：账户认证、短信验证接口示例
BankListTradeCase：支持银行列表文件接口示例
BatchTradeCase：批量代收、批量代付、批量查询接口示例
FundInfoTradeCase：基金可用额度查询、基金商户平垫资查询接口示例
FundSingleQueryTradeCase：基金商户单笔代收查询、基金商户单笔代付查询接口示例
FundSingleTradeCase：基金商户单笔代收、基金商户单笔代付接口示例
LoadFileTradeCase：下载对账文件、下载退汇流水文件、下载结算户流水文件、下载对账文件（文件内容加密）接口示例
MerchantInfoTradeCase：商户信息查询、子商户信息查询、银行信息查询接口示例
SensitiveEncryptTradeCase：敏感字段加密、解密方法示例
SingleQueryTradeCase：普通商户单笔代收查询、普通商户单笔代付查询接口示例
SingleTradeCase：普通商户单笔代收、普通商户单笔代付接口示例

3.3.4 util 文件夹中包含了签名、验签、加密、文件解压缩等算法



3.3.5 签名

注：sdk 版本号>=1.2 的签名方法做了微调，即将证书信息提取出来，作为参数传入。

1.找到 sdk 工程中的 main.java.prod.yypay.sdk 包下的 SignTest 类；
2.创建 map 对象 request，存放请求参数，如下图：

```
Map<String,String> request = new HashMap<String,String>();
// 账户认证
request.put("version", "1.0.2");// 版本号
request.put("mchtId", "ORG070000000000000000");// 商户号（具体以汇收付配置的商户号为准）
request.put("mac",
        "C65wrB4DYktilz6bAqyHDXREMBd5ArNE4P2Ng80yFnjWE0pJjGCD6y9FGxs6rSNCI"
        + "FB3baY/GevN1KchRx3KChgSyhFchAXRhJFEqrdk1OISfE9k5Qa9ySIAqA2wUeei"
        + "kHx/jOslg4VugLy31sXrUAkooCvAkMcrRR7nkLuxCak=");// 签名值
request.put("signType", "RSA");// 签名类型
request.put("serialNo", "201808211529020001");//流水号
request.put("transTime", "20180821152902");// 请求时间（选填）
request.put("acctType", "1");// 账户类型（选填）
request.put("acctName",
        "rle+HShCATnHmWzpTSklQbE7Bm9T3xepa3YgDFEBrKmkIaa33I8rVP7Ykas7QuA/+"
        + "lnHGRYMgAfENI36cgRbU7xYTVNax5FcOdPCyc864/XoO03pMvbVlZditlhqNuji"
        + "0DqffIv6xxRR+vHuYBpAbsA3RWb1kYt7S1YepMEh/Z0=");// 账户名
request.put("acctNo",
        "K0R3ll++UIOi4DhTwqbjCQ+2RjXjGAaayR7u9m2lTUKeDbq+02KMNfR9UJUM1x80b"
        + "1y5PGjgHCPnIxq88EC9KDOOI9+BwRVeFK5ity5EQMyVwEkVEFB2+VQFpZWT/gLw"
        + "d6H7EXR/b7Ukdo8wlXp2RVLHiIPEyPueq/YCq3za0GM=");// 账号
request.put("certType", "0");// 证件类型
request.put("certNo",
        "T2w+wjkqoswEYdb9FmZ6FWhwkNHcgQyKFE3KzrsDkN2sfnsFvru9wgtvHzh6YJ2TY"
        + "J8kg114fpFb7GHiMskBEnKriWf5ckEmKBjCbSDBvPNCE8F2MKRAIb0eJ5ZM1fOt"
        + "pBP0kKrLqBLH0btTQh3rQ5KGk4K/hC0MCSqihGF49v4=");// 证件号码
request.put("bankNo", "00001");// 五位联网行编号（选填）
request.put("bankName", "兴业银行");// 开户行名称（选填）
request.put("rsrvPhoneNo",
        "PmXcik5y6hoMnyMyMjsYa+mKtu55nr2iQodyeMYftRej5HMMA7713s4JnfKp4UcvF"
        + "8fxmReQYH9zUn2X4pjqoNNQNzAqpFNbIiJtZLDes4hZpMdav0govkmzKJ4CPklL"
        + "p30ZeHBem/9Nz4xLcL9wgY9aM8LfUtzZBfQTw0dmfy0=");// 银行预留手机号
request.put("cvn", "");// 安全验证码（选填）
request.put("expireDate", "");// 有效期（选填）
request.put("sendSms", "false");// 发送短信标志
request.put("pubOrPriFlag", "1");// 对公对私标志，默认对私（选填）
```

3.调用 Signature.generateMAC(request)方法，生成签名值，如下图：

```
/**
 * 版本号为1.0.2，字段为null或空字符串时不参与签名和验签
 */
// 签名方法
System.out.println("mac:" + Signature.generateMAC(request, Configure.PRI_MERCHANT_CERT,
        Configure.PRI_MERCHANT_CERT_PWD));
```

3.3.6 验签

注：sdk 版本号>=1.2 的验签方法做了微调，即将证书信息提取出来，作为参数传入。

1.找到 sdk 工程中的 main.java.prod.yypay.sdk 包下的 SignTest 类；
2.调用 Signature.verifyMAC(request)方法验签，如下图：

```
// 验签方法
boolean mac = Signature.verifyMAC(request, Configure.PUB_SERVER_CERT);
System.out.println("————————— "+mac);
```

3.3.7 文件加密

1.找到 sdk 工程中的 main.java.prod.yypay.sdk 包下的 SignTest 类；
2.指定本地交易的文件，读取文件内容并转成字节数组，如下图：

```
File file = new File("D:/data/batchPay.txt");//批量代付文件
FileInputStream fi = new FileInputStream(file);
ByteArrayOutputStream bos = new ByteArrayOutputStream();
byte[] data = null;
byte[] buff = new byte[1024];
int len=0;
while((len=fi.read(buff))!=-1){
bos.write(buff, 0, len);
}

data = bos.toByteArray();
```

3.先调用 FlaterUtil.compress(data)方法，对文件内容进行压缩，如下图：

```
//使用deflater压缩文
byte[] bb = FlaterUtil.compress(data);
```

4.再调用 Base64.encode(bb)方法，对文件内容进行编码，得到加密后的文件内容，如下图：

```
//使用base64转码
System.out.println("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~");
System.out.println((Base64.encode(bb)));
System.out.println("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~");
```

3.3.8 文件解密

1.找到 sdk 工程中的 main.java.prod.yypay.sdk 包下的 SignTest 类；
2.先调用 Base64,decode(content.toCharArray())对加密后的文件内容进行解码，如

下图：

```
//解压批量文件内容
String content = "eJyt2D2KFEEYgOHcw0h9P/X9nMBQ8A6ewA0rMdhINPECGwgKmxhu4HVG8RbW
byte[] bb1 = Base64.decode(content.toCharArray());
```

3.再调用 Flater.decompress(bb1)方法对加密后的文件内容进行解压缩，如下图：

```
byte[] dedata = FlaterUtil.decompress(bb1);
ByteArrayInputStream bais = new ByteArrayInputStream(dedata);
BufferedReader br = new BufferedReader(new InputStreamReader(bais, "UTF-8"));
String strdata = null;
int i = 0;
while ((strdata = br.readLine()) != null && !strdata.equals("")) {
    System.out.println("解压后的文件内容：" + strdata);
}
} catch (Exception e) {
e.printStackTrace();
}finally{
}
```

3.3.9 秘钥串解密

1.找到 sdk 工程中的 main.java.prod.yypay.sdk 包下的 SignTest 类；
2.将应答报文中返回的秘钥串，需使用商户的私钥证书调用 Signature.decoderByCFCA(aesKey, String, String)方法进行解密，得到解密后的秘钥串。

```
// 应答返回的秘钥串
String aesKey = "PCgFPO2aCEFSGzYGmZujqy1+ZYoaKcDFXf61W0aMwsToVQgjmu8OMTREXyXG6V0"
        + "Do0IzrbRxQLOHhzQjmVEwabokyeZ22xz3fk2LkLPq21wUyy73OJxjSDGiKanaeQ86HB3y"
        + "wSGKe0k7FTO9foft5sb+IZ1PlWuEgTym6XmkaUw=";
// 用商户私钥证书对秘钥串进行解密
String key = Signature.decoderByCFCA(aesKey,
        Configure.PRI_MERCHANT_CERT,
        Configure.PRI_MERCHANT_CERT_PWD);
System.out.println("解密后的秘钥串:" + key);
```

3.3.10 退汇流水文件、结算户流水文件 AES 解密

1.找到 sdk 工程中的 main.java.prod.yypay.sdk 包下的 SignTest 类；
2.用 2.3.9 解密后的秘钥串，将应答返回的文件内容调用 Signature.AESDecode(String, String)方法进行 AES 解密；

```
/**
 * 退汇流水文件和结算户流水文件解密方法
 */
// 应答返回的文件内容
String respFileContent = "653978ce1f8a852366b63aec9caaeeb7c8e2f6e7ca7ff544851e934a"
        + "359207f178e9a53ad3c00edccc47978e693269abcc921bad88d16b266d2b108f48b04aa"
        + "a7cde2391c68b8d00eb6718aea00b2096987812334e7bbb6539a14f0377d233b35d9d05"
        + "acc648d4bcc041b3de6b025e388866955c84be9a11a00b8cb18277b50edfb21754df2f8"
        + "e655fff06b1990c8083fb5e13b60e797053a7a9aac6093ddbdf465cd4efd923b480772e"
        + "391962f7f151bfa35929f9595d26e76354dea951432a";
// 应答返回的秘钥串
String aesKey = "PCgFPO2aCEFSGzYGmZujqy1+ZYoaKcDFXf61W0aMwsToVQgjmu8OMTREXyXG6V0"
        + "Do0IzrbRxQLOHhzQjmVEwabokyeZ22xz3fk2LkLPq21wUyy73OJxjSDGiKanaeQ86HB3y"
        + "wSGKe0k7FTO9foft5sb+IZ1PlWuEgTym6XmkaUw=";
// 用商户私钥证书对秘钥串进行解密
String key = Signature.decoderByCFCA(aesKey,
        Configure.PRI_MERCHANT_CERT,
        Configure.PRI_MERCHANT_CERT_PWD);
System.out.println("解密后的秘钥串:" + key);

System.out.println("退汇流水文件和结算户流水文件内容解密start=======");
// 用AES给文件内容解密
byte[] fileContent = Signature.AESDecode(respFileContent, key);
```

3.再调用 Base64,decode(fileContent.toCharArray())对加密后的文件内容进行解码，如下图：

```
// 用Base64给文件内容转码
byte[] b = Base64.decode(new String(fileContent, "utf-8").toCharArray());
```

4.最后调用 Flater.decompress(bb1)方法对加密后的文件内容进行解压缩，如下图：

```
// 对文件内容进行解压缩
byte[] deData = FlaterUtil.decompress(b);
ByteArrayInputStream byteArrayInputStream = new ByteArrayInputStream(deData);
BufferedReader bufferReader = new BufferedReader(new InputStreamReader(byteArrayInputStream,
        "UTF-8"));
String strData = null;
while ((strData = bufferReader.readLine()) != null && !strData.equals("")) {
    System.out.println("退汇流水文件或结算户流水文件解压后的文件内容：" + strData);
}
```

3.4测试卡号（测试环境使用）

3.4.1 账户认证

账号（鉴权商户使用）	账户名	账户类型	银行编号	银行名称	手机号	证件类型	身份证号
622909049100014018	test	1	00001	兴业银行	13100049921	0	110101199607280175
622909049040205114	test	1	00001	兴业银行	13100008710	0	110101199607280108

622909049100005016	test	1	00001	兴业银行	13100047877	0	110101199607280140
622909473995042118	test	1	00001	兴业银行	13199995042	0	330195198405020427
622909473995043314	test	1	00001	兴业银行	13199995043	0	330195198405020435
622909473995044510	test	1	00001	兴业银行	13199995044	0	330195198405020443
622909473995045715	test	1	00001	兴业银行	13199995045	0	330195198405020451
6222801989022812345	通用 1	1	05004	建设银行	15601921234	0	610526198902291234
6222809876543210000	通用 2	1	05004	建设银行	13152620001	0	610526198902283010
6222809876543219810	通用 3	1	05004	建设银行	13800000004	0	350103198701019988

3.4.2 单笔代收

账号（鉴权商户使用）	账户名	账户类型	银行编号	银行名称	手机号	证件类型	身份证号
622909049100014018	test	1	00001	兴业银行	13100049921	0	110101199607280175
622909049040205114	test	1	00001	兴业银行	13100008710	0	110101199607280108
622909049100005016	test	1	00001	兴业银行	13100047877	0	110101199607280140
622909473995042118	test	1	00001	兴业银行	13199995042	0	330195198405020427
622909473995043314	test	1	00001	兴业银行	13199995043	0	330195198405020435
622909473995044510	test	1	00001	兴业银行	13199995044	0	330195198405020443
622909473995045715	test	1	00001	兴业银行	13199995045	0	330195198405020451
6222801989022812345	通用 1	1	05004	建设银行	15601921234	0	610526198902291234
6222809876543210000	通用 2	1	05004	建设银行	13152620001	0	610526198902283010
6222809876543219810	通用 3	1	05004	建设银行	13800000004	0	350103198701019988
账号（免鉴权商户使用）							
6228480000111122222	免鉴权 1	1	05002	农业银行	17300001111	0	610502198802023816
6228480000111133333	免鉴权 2	1	05002	农业银行	17300001234	0	610502201702173030
6228480000123412345	免鉴权 3	1	05002	农业银行	17389140001	0	610502201702172017
6228480000111155555	免鉴权失败	1	05002	农业银行	17300005555	0	610502198903265050
6228480000111144444	免鉴权处理中	1	05002	农业银行	17300004444	0	610502198903261220

3.4.3 单笔代付

账号	账户名	账户类型	银行编号	银行名称	手机号	证件类型	身份证号

6226090000000048	代付 1	1	05008	招商银行	18100000000	0	510265790128303
6226388000000095	代付 2	1	05020	华夏银行	18100000000	0	510265790128303
6216261000000000018	代付 3	1	05010	平安银行	13552535506	0	341126197709218366
6221558812340000	代付 4	1	05010	平安银行	13552535506	0	341126197709218366
5200831111111113	代付 5	2	05002	农行	13552535506	0	341126197709218366
6221558812340013	代付 6	1	05010	平安银行	13552535506	0	341126197709218366
6222801989022812345	通用 1	1	05004	建设银行	15601921234	0	610526198902291234
6222809876543210000	通用 2	1	05004	建设银行	13152620001	0	610526198902283010
6222809876543219810	通用 3	1	05004	建设银行	13800000004	0	350103198701019988

3.4.4 批量代收

账号（鉴权商户使用）	账户名	账户类型	银行编号	银行名称	手机号	证件类型	身份证号
6222801989022812345	通用 1	1	05004	建设银行	15601921234	0	610526198902291234
6222809876543210000	通用 2	1	05004	建设银行	13152620001	0	610526198902283010
6222809876543219810	通用 3	1	05004	建设银行	13800000004	0	350103198701019988
账户（免鉴权商户使用）							
6228480000111122222	免鉴权 1	1	05002	农业银行	17300001111	0	610502198802023816
6228480000111133333	免鉴权 2	1	05002	农业银行	17300001234	0	610502201702173030
6228480000123412345	免鉴权 3	1	05002	农业银行	17389140001	0	610502201702172017
6228480000111155555	免鉴权失败	1	05002	农业银行	17300005555	0	610502198903265050
6228480000111144444	免鉴权处理中	1	05002	农业银行	17300004444	0	610502198903261220

3.4.5 批量代付

账号	账户名	账户类型	银行编号	银行名称	手机号	证件类型	身份证号
6226090000000048	代付 1	1	05008	招商银行	18100000000	0	510265790128303
6226388000000095	代付 2	1	05020	华夏银行	18100000000	0	510265790128303
6216261000000000018	代付 3	1	05010	平安银行	13552535506	0	341126197709218366
6221558812340000	代付 4	1	05010	平安银行	13552535506	0	341126197709218366
5200831111111113	代付 5	2	05002	农行	13552535506	0	341126197709218366
6221558812340013	代付 6	1	05010	平安银行	13552535506	0	341126197709218366
6222801989022812345	通用 1	1	05004	建设银行	15601921234	0	610526198902291234

6222809876543210000	通用 2	1	05004	建设银行	13152620001	0	610526198902283010
6222809876543219810	通用 3	1	05004	建设银行	13800000004	0	350103198701019988

4. 常见问题解答

4.1http（或 https）通信常见报错

4.1.1 400 错误

- 请求对象属性多了或者少了
- 请求对象属性值不符合要求，比如 Integer 类型不能传 string 类型等…

4.1.2 405 错误

- post 请求
- content-type 设置错误 application/json,utf-8

4.2交易常见报错

4.2.1 签名不一致？

商户调用接口时，如果遇到签名不一致的报错信息，请以此检查以下几种情况：

- 证书是否用错了（使用商户私钥签名）
- 版本号是否用错了（参考接口文档，或向业务人员咨询）
- 参与签名的属性以及顺序是否不正确（参考 sdk 中 Signature.java）
- 属性赋值是否不符合规则（比如金额不是两位小数）
- 签名字段是否包含中文，编码方式是否不正确（字符串使用 UTF-8 编码）
- 签名算法跟我们提供的是否不一致（参考 sdk 中 Signature.java）

4.2.2 限额问题

- 机构单日笔数或金额超限
- 客户单日金额超限
- 交易金额不符合路由表配置的最低最高范围
- 渠道支持的银行单日交易金额超限

4.2.3 无渠道支持此交易

商户如果连接的是汇收付的测试环境。首先检查是否用的是汇收付提供的交易对应的测试账号，有些银行在测试环境中渠道是不支持的。如果测试账号使用正确，联系汇收付系统业务人员查看渠道配置是否正确。

4.3接口参数常见问题

4.3.1 单笔代收接口里面为什么有持卡人的证件类型、证件号码、手机号字段

由于部分银行在代收的时候需要收款人的证件类型、证件号码、手机号，所以接口中加了这三个字段。这三个字段为非必填字段，商户需要跟业务人员沟通是否需要传值。

4.3.2 请求报文中的流水号 serialNo

必须唯一。参考接口文档 3.4.2 商户流水号生成算法。

注：单笔交易：请求报文中的流水号要保持唯一

批量交易：1）文件头中的批次号（即请求报文中的流水号）要保持唯一

2）文件体中的流水号也要保持唯一

3）不同批次中的流水号也要保持唯一

（建议单笔交易和批量交易的流水号也保持全局唯一性）

4.4字段脱敏（加密）

4.4.1 涉及加密的接口和字段

接口名称	参与加密的参数名称			
身份认证	账户名称（acctName）	账号（acctNo）	银行预留手机号（rsrvPhoneNo）	证件号码（certNo）
短信验证	手机号（phoneNo）			
单笔代收	账户名称（acctName）	账号（acctNo）		
单笔代付	账户名称（acctName）	账号（acctNo）		

4.4.2 加密方式

接口的请求参数，先用汇收付提供的公钥对需要加密的字段进行加密，再对请求参数签名。后发送请求。

接口的响应参数，暂时没有加密，接收到响应参数不需要解密。

4.4.3 敏感字段加密

注：sdk 版本号>=1.2 的敏感字段加密方法做了微调，即将证书信息提取出来，作为参数传入。

1.找到 sdk 中 prod.yypay.sdk.trade.SensitiveEncryptTradeCase；
2.对于明文 name，调用 Signature.encryptByCFCA(name, Configure.PUB_SERVER_CERT)方法进行加密，如下图：

```
//敏感字段加密
String name = "代付1";
String mi = Signature.encryptByCFCA(name, Configure.PUB_SERVER_CERT);
if (mi == null) {
    System.out.println("加密失败");
} else {
    System.out.println(name + "密文：" + mi);
    System.out.println(mi.length());
}
```

4.4.4 敏感字段解密

注：sdk 版本号>=1.2 的敏感字段解密方法做了微调，即将证书信息提取出来，作为参数传入。

1.找到 sdk 中 prod.yypay.sdk.trade.SensitiveEncryptTradeCase；
2.对于密文 mi，调用 Signature.decoderByCFCA(mi, PRI_SERVER_CERT, Configure.PRI_MERCHANT_CERT_PWD)方法进行解密，如下图：

```
lic class SensitiveEncryptTradeCase {
  // 敏感字段解密--测试环境用测试环境汇收付私钥证书，生产环境无需使用
  public static final String PRI_SERVER_CERT = "prod/yypay/sdk/key/payServer-rsa.pfx";

  public static void main(String[] args) throws KeyManagementException, NoSuchAlgorithmException,
      //敏感字段加密
      String name = "代付1";
      String mi = Signature.encryptByCFCA(name, Configure.PUB_SERVER_CERT);
      if (mi == null) {
          System.out.println("加密失败");
      } else {
          System.out.println(name + "密文：" + mi);
          System.out.println(mi.length());
      }
      name = "6226090000000048";
      mi = Signature.encryptByCFCA(name, Configure.PUB_SERVER_CERT);
      if (mi == null) {
          System.out.println("加密失败");
      } else {
          System.out.println(name + "密文：" + mi);
          System.out.println(mi.length());
      }
      // 敏感字段解密
      mi = "ZrvjrjXZckzgtQQI934bJG0MROQy+lDBm6f5Y2DXtHTiFavBW5CiccOzwOXhZnYnsO25byIdfDt2uykUISvM(
      String ming = Signature.decoderByCFCA(mi,PRI_SERVER_CERT,Configure.PRI_MERCHANT_CERT_PWD);
      if (ming == null) {
          System.out.println("解密失败");
      } else {
          System.out.println("明文：" + ming);
      }
  }
```

4.5对账文件

4.5.1 D+1 对账文件

商户在汇收付系统每日（D 日）做的交易，在第二日（D+1 日）10 点对账。一般在第二日 10:30 后可以取对账文件。

例如：商户在 10 月 19 日 10 点 30 分可以调用对账文件下载接口（checkType 传 1）下载对账文件。

注意：10 月 19 日的 D+1 对账文件一般是 10 月 18 日 0 点到 24 点成功的交易，但由于渠道日切时间不一致等问题，可能会存在少量 10 月 18 日交易在 10 月 19 日 D+1 对账单中未体现，出现在 10 月 20 日对账单中。因此，建议商户进行对账文件与交易流水比对时，采用 2 日对账逻辑，即 D 日交易流水若 D+1 日对账单、D+2 日对账单中均不存在，则认为交易存在差错。

4.5.2 D+0 对账文件

该对账文件一般用于基金行业对账，每日（D 日）根据商户配置的日切时间点生成对账文件，包括 D-1 日日切点~D 日日切点的所有成功交易订单。

例如：商户在汇收付配置的日切时间为 15:00:00，汇收付系统在 10 月 19 日 15 点 10 分后调用对账文件下载接口（checkType 传 0）下载对账文件。此文件是 10 月 18 日 15 点到 10 月 19 日 15 点成功的交易流水。

备注：对账文件中只有成功的交易订单

4.5.3 下载对账文件（旧接口）

注：新接入机构请参考接口文档 4.4.4，后续不再支持非加密的对账文件下载接口

对账文件内容解密方法：

1. 找到 sdk 工程中的 main.java.prod.yypay.sdk 包下的 SignTest 类；
2. 先调用 Base64,decode(content.toCharArray())对加密后的文件内容进行解码，如下图：

```
//解压批量文件内容
String content = "eJyt2D2KFEEYgOHcw0h9P/X9nMBQ8A6ewA0rMdhINPECGwgKmxhu4HVG8RbV
byte[] bb1 = Base64.decode(content.toCharArray());
```

3. 再调用 Flater.decompress(bb1)方法对加密后的文件内容进行解压缩，如下图：

```
byte[] dedata = FlaterUtil.decompress(bb1);
ByteArrayInputStream bais = new ByteArrayInputStream(dedata);
BufferedReader br = new BufferedReader(new InputStreamReader(bais, "UTF-8"));
String strdata = null;
int i = 0;
while ((strdata = br.readLine()) != null && !strdata.equals("")) {
    System.out.println("解压后的文件内容：" + strdata);
}
} catch (Exception e) {
e.printStackTrace();
}finally{
}
```

4.5.4 下载对账文件（新接口）（文件内容加密）

对账文件内容 AES 解密：

1.找到 sdk 工程中 main.java.prod.yypay.sdk.trade 包下的 LoadFileTradeCase 类，找到如下图的方法：

```
// 下载对账文件(文件内容加密)请求交易接口
String response3 = loadEncryptCheckFile();
System.out.println("下载对账文件(文件内容加密)请求返回结果：" + response3);
//解压对账文件内容(文件内容加密)
obtainEncryptCheckFileContent();
```

2.用 2.3.9 解密后的秘钥串，将应答返回的文件内容调用 Signature.AESDecode(String，String)方法进行 AES 解密；

```
// 应答返回的加密对账文件内容
String encryptCheckFileContent = "89c13a68a922600c4ade779ccf3c2d8a5d20a1379782fb482
// 应答返回的秘钥串
String aesKey = "HnDFNe1o3+AgXoRCUg3F+8ohFTSrQG7CraQKaJfcoax3YrTOQ+qZMUB3nrLolUXm5B
// 用商户私钥证书对秘钥串进行解密
String key = Signature.decoderByCFCA(aesKey,
        Configure.PRI_MERCHANT_CERT,
        Configure.PRI_MERCHANT_CERT_PWD);
System.out.println("解密后的秘钥串:" + key);

System.out.println("加密对账文件内容解密start=======");
// 用AES给文件内容解密
byte[] fileContent = Signature.AESDecode(encryptCheckFileContent, key);
```

3.再调用 Base64,decode(fileContent.toCharArray())对加密后的文件内容进行解码，如下图：

```
// 用Base64给文件内容转码
byte[] b = Base64.decode(new String(fileContent, "UTF-8").toCharArray());
```

5.最后调用 Flater.decompress(bb1)方法对加密后的文件内容进行解压缩，如下图：

```
// 对文件内容进行解压缩
byte[] deData = FlaterUtil.decompress(b);
ByteArrayInputStream byteArrayInputStream = new ByteArrayInputStream(deData);
BufferedReader bufferReader = new BufferedReader(new InputStreamReader(byteArrayInputStream,
        "UTF-8"));
String strData = null;
while ((strData = bufferReader.readLine()) != null && !strData.equals("")) {
    System.out.println("加密对账文件内容解压后的文件内容：" + strData);
}
```

4.6测试环境 Q&A

4.6.1 测试环境的商户号如何开通？

第一时间联系兴业银行相关产品业务人员，申请开通即可，原则上当天开通后即可使用。见 3.1。

4.6.2 非基金公司测试环境接口访问路径是什么？

见 2.6.1.1。

4.6.3 基金公司测试环境接口访问路径是什么？

见 2.6.1.2。

4.6.4 测试环境银行卡信息是什么？

见 3.4。

4.6.5 测试环境接口 demo 样例有吗？

第一时间联系兴业银行相关产品业务人员，申请即可，原则上当天申请后即可使用。
见 3.3.2。

4.6.6 测试环境的商户公私钥是什么？

详请参阅 SDK 压缩包文件，文件路径：SDK\java\prod\yypay\sdk\key。见 3.3.2。

4.6.7 测试环境有没有 IP 白名单限制？

暂无。

4.6.8 测试环境的网络访问支持公网访问吗？

支持。

4.6.9 接口的签名算法和签名要求是什么？

详请参阅 SDK 压缩包文件，文件路径：SDK\java\prod\yypay\sdk\SignTest.java。见 3.3.4。

4.6.10 签名算法支持哪些方式？

签名算法目前仅支持 RSA 签名方式。见 2.2.1.1。

4.6.11 签名具体步骤是怎么样的？

将请求或者响应中的键值对以 key=value 形式组装成一个数组，将该数据按照参数名 ASCII 码从小大到大排序后使用 URL 键值对的格式&进行拼装（keyl=valuel&key2=value2)。
需要注意的：
（1）：[接口版本 1.0.2]可选字段（参数值为空、null、undefine 等）不参与签名；
（2）：签名字段（即 mac)不参与签名。
（3）：编码使用 UTF-8,参数中不能出现类似&等特殊字符；
（4）：不需要对参数键、值等进行 URLEncode 操作。
详情见接口文档。

4.6.12 公共报文头里的“接口版本”字段，具体如何上送？

版本号为 1.0.2，字段为 null 或空字符串时不参与签名和验签

4.6.13 接口带“*”字段信息属于敏感字段需脱敏处理，具体如何操作？

用 pubServer-rsa.cer 公钥证书，参考 sdk 里 Signature.encryptByCFCA()方法。注意：先脱敏，后签名。见 4.4 或者参阅接口文档“3.4.5 字段脱敏”。

4.7生产环境 Q&A

4.7.1 非基金公司生产环境接口访问路径是什么？

公网见 2.6.2.1 专线见 2.6.2.3。

4.7.2 基金公司生产环境接口访问路径是什么？

公网见 2.6.2.2 专线见 2.6.2.4。

4.7.3 上生产环境时，合作商户除基础协议还需要提供什么？

- 商户公钥（双方交换）
- IP 白名单（根据商户自身的业务业务诉求可配置）

4.7.4 上生产环境时，合作商户商户号是什么？

兴业银行根据商户签约的协议走流程配置基本信息，完成后发送给《开通表》内联系人邮箱。

4.7.5 生产环境的网络访问，是什么方式？

二选一：
公网；
专线：因行内需要走流程申请，请有需要的合作商户至少提前 1-2 个月提出来；
详情见 2.1。

4.8账户认证常见问题

4.8.1 账户认证支持哪些账户类型

支持银行卡认证。特殊卡种请联系业务人员确认。

4.8.2 同一个账户是否可以重复认证

可以。无验证次数限制，但按笔收费。

4.8.3 发送短信标志 sendSms 有何用途？

卡鉴权是否需要发短信，我行系统可实现由兴业银行发或者不发，具体取决于贵司业务诉求。
具体可上送：true 或者 false。

4.8.4 如何根据响应码判断交易结果

- 认证成功：　　E0000（认证成功）
- 认证失败：　　E05**（除了 E0508，其他 05 开头都是业务类失败，例如 E0501）
  　　E01**（缺少认证要素类失败，例如 E0101）
- 认证结果未明：　E0508（认证处理中）
  　　E00**（除了 E0000，其他都是异常类错误、未知状态，例如 E0001）

4.8.5 账户认证有没有查询接口

账户认证没有查询接口。账户认证结果未明的，请对此账户重新调用账户认证接口进行认证。

4.9查询类接口常见问题

4.9.1 如何查询交易未明的订单

请求参数中 serialNo 传需要查询的单笔订单的流水号，或者批量交易的批次号，进行查询。

4.9.2 如何判断查询结果

4.9.2.1 单笔交易查询

响应信息中 tranStatus 表示所查询的订单的状态。

1) tranStatus 为 null 时，
   a. 如果 respCode 为 E0523（订单信息不存在），需再次查询，若 15 分钟后仍为“E0523”，则该笔订单为失败订单，
   b. 其他情况，为查询失败，respCode 查询交易的错误码、respMsg 查询交易的错误信息；
2) tranStatus 为 1 时，所查询订单为交易成功（respCode 为 E0000）；
3) tranStatus 为 2 时，所查询订单为交易处理中（respCode 为 E0508）；
4) tranStatus 为 0 时，所查询订单为交易失败
（respCode 订单的错误码、respMsg 订单的错误信息）。

4.9.2.2 批量交易查询

判断响应码（respCode）：

1) E0000--查询成功，再判断批次状态（batchStatus）
   - 1，处理完成（只有批次状态为 1-处理完成时，才会返回明细）
   - 2，处理中
   - 3，处理失败
2) E0529--订单不存在，需再次查询，若 30 分钟后仍为“E0529”，则该笔订单为失败订单，
3) 其他响应码，继续查询

4.9.3 交易未明的订单什么时候来查询状态

交易未明的订单，15 分钟查询一次，直到查到订单的明确结果结束查询。

4.9.4 批量查询文件头成功笔数、成功金额

批量查询结果文件新增批量文件的成功笔数、成功金额统计。

4.10 交易类接口常见问题

4.10.1 如何根据响应码判断交易结果（同账户认证）

- 交易成功：E0000（交易成功）
- 交易失败：E05**（除了 E0508，其他 05 开头都是业务类失败，例如 E0501）
  E01**（关键交易参数为空，例如 E0101）
- 交易结果未明：E0508（交易处理中）
  E00**（除了 E0000，其他都是异常类错误、未知状态，例如 E0001）

4.10.2 交易未明什么时候有明确结果

一般未明的交易 15 分钟后即可查询交易结果。如果未查到明确结果，请 15 分钟查询一次。

4.10.3 交易是否限制并发？并发数为多少？

限制并发，目前生产环境默认配置的支持并发为 TPS 80 笔/秒。如果贵司有业务需求，可提前跟我们反馈。

4.10.4 交易限额（单笔、单日、单月）

取决于后端支付渠道银行本身的限额，详情请咨询业务人员。