

金证统一接入第三方应用技术协议接入规范

公司名称	金证科技股份有限公司	文档编号	
文档名称	金证统一接入第三方应用技术协议接入规范	文档版本	V1.0.2
起草	陈晓光	起草日期	20181125
审批		审批日期	

修订历史

版本号	日期	状态	修订人	摘要
V1.0.0	20181125	C	陈晓光	
V1.0.1	20200415	M	陈晓光	修改 C/C++ API 字段加密方式描述
V1.0.2	20200423	M	陈晓光	修改章节目录及文字描述

状态标识：C - Created A - Added M - Modified D - Deleted

目录

1 目的和意义	1
2 名词解释	1
3 接入指引	1
4 技术协议说明	1
4.1 KDGP 技术协议	1
4.1.1 协议描述	2
4.1.1.1. 报文格式	2
4.1.1.2. 通讯过程	3
4.1.1.3. 消息类型	3
4.1.1.3.1. 消息标准字段	3
4.1.1.3.2. 会话层消息	5
4.1.1.3.3. 应用层消息	6
4.1.2 协议开发实现	6
4.1.2.1. 开放式接入开发实现	6
4.1.2.1.1. 接口实现	7
4.1.2.1.2. 特殊字段处理	10
4.1.2.2. API 方式接入开发实现	10
4.1.2.2.1. C/C++	10
4.1.2.2.2. JAVA	13
4.2 KHTTP 技术协议	19
4.2.1 协议描述	19
4.2.1.1. 报文格式	19
4.2.1.2. 通讯过程	20
4.2.1.3. 消息类型	20
4.2.1.3.1. 消息标准字段	20
4.2.1.3.2. 会话层消息	22
4.2.1.3.3. 应用层消息	23
4.2.2 协议开发实现	24
4.2.2.1. 开放式接入开发实现	24
4.2.2.1.1. 接口实现	24
4.2.2.1.2. 特殊字段处理	27

1 目的和意义

本文主要为第三方应用对接金证统一接入提供详尽的应用开发指导，针对统一接入平台接入指引、所支持的技术协议以及对应的 API 使用方式进行了详细说明。

2 名词解释

KUAB：统一接入平台（Kingdom Unified Access Bus）

KDGP：统一接入平台对外标准 TCP 技术协议（Kingdom General Protocol）

KHTTP：统一接入平台对外标准 HTTP 技术协议（Kingdom General Http）

AppID：开发者 ID

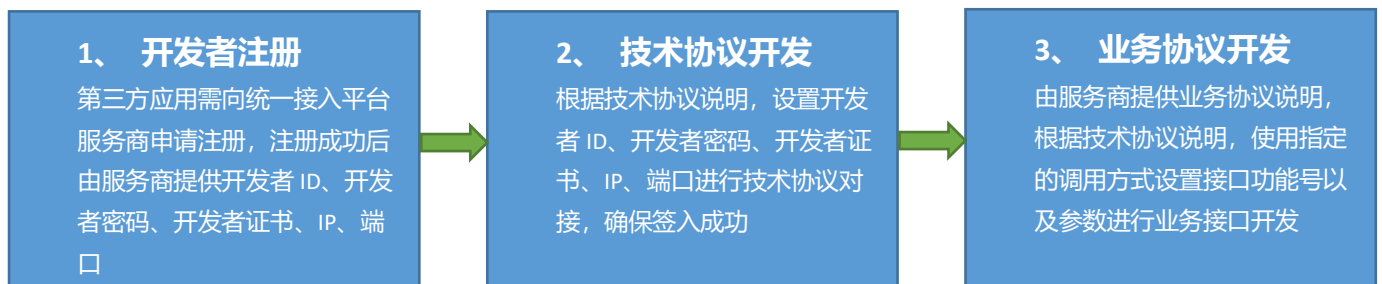
AppSecret：开发者密码

开发者证书：接入证书，用于标识开发者身份的 RSA 私钥

会话密钥：KDGP 协议签入成功以后返回的会话期间有效的密钥串

会话 Cookie：KHTTP 协议签入成功以后返回的会话期间有效的密钥串

3 接入指引



4 技术协议说明

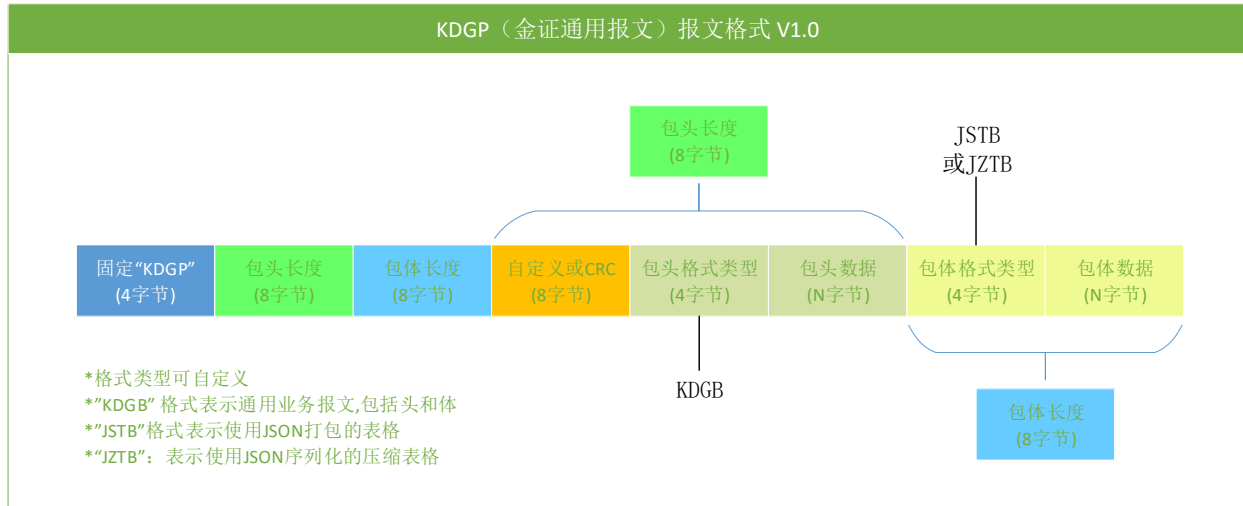
统一接入平台支持 KDGP 以及 KHTTP 两种技术协议，分别对应 TCP 应用协议和 HTTP 应用协议，接下来将对这两种技术协议的报文格式、通讯过程、消息类型以及开发实现进行详细的说明。

4.1 KDGP 技术协议

KDGP 协议是统一接入平台对外提供的标准 TCP 应用协议，金证提供了标准的 JAVA API 和 C/C++ API 给第三方应用进行应用开发。第三方应用也可以自己实现整个报文的组装、加解密、发送、接收和解析过程进行开放式接入开发实现，以下是对该协议的详细描述。

4.1.1 协议描述

4.1.1.1. 报文格式



此为通用报文格式，包括签入、业务调用和签出的入参和出参都适用。

固定“KDGP”：物理报文头部格式 “KDGP” 表明整包使用 KDGP 规范序列化（如果是加密包，使用“KGPS”标识，目前仅为保留项，不要使用这种方式）。

包头长度：计算自定义或 CRC、包头格式或类型、包头数据这三个字段的总长度，并把值写到这个字段上面。

包体长度：计算包体格式类型、包体数据这两个字段的总长度，并把值写到这个字段上面。

CRC 签名：报文必须经过动态密钥（签入报文使用 00000000）签名，签名过程如下：

1. 执行签入动作，获得统一接入平台返回的会话密钥
2. 执行业务调用或签出功能时，将会话密钥填入 KDGP 报文头部的 CRC 字段
3. 对整个报文从 CRC 字段开始（包括 CRC 字段）进行 CRC8 签名
4. 把生成的 CRC8 码填入原 KDGP 报文头部的 CRC 域，覆盖原会话密钥

包头格式类型：固定设置为“KDGB”，通用业务描述，使用 JSON 序列化，格式形如：
{key:value}

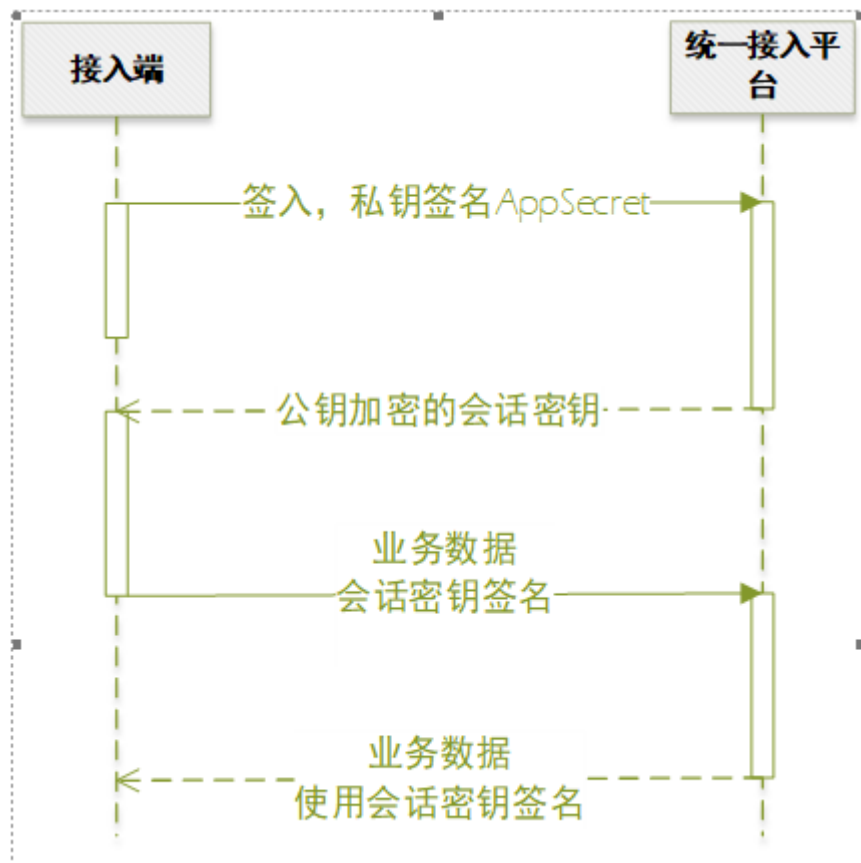
包头数据：JSON 对象，包含了消息标准字段，为平台专用字段，第三方应用也可以自行扩充，统一接入平台会原样返回，具体请参考消息类型章节描述。

包体格式类型：对于入参，固定设置为“JSTB”，表示使用 JSON 序列化的表格，请求包体格式为 {}，应答包体格式为 [{}, {}] 或 [[{}], {}], [{}, {}]]；对于出参，由统一接入平台自动确定是用“JSTB”还是“JZTB”，外围要对这个字段进行判断，如果返回的是“JZTB”，表示统一接入平台对返回结果使用 JSON 序列化的压缩表格，进行了 gzip 压缩和 base64 编码，需要进行相应的解码和解压处理。

包体数据：包括签入消息、签出消息、心跳消息、业务消息等数据

注：所有字段名和字段值区分大小写

4.1.1.2. 通讯过程



4.1.1.3. 消息类型

4.1.1.3.1. 消息标准字段

第三方应用需要在包头填充消息标准字段，以下表格中描述了消息标准字段，如非特殊说

明，则都表示请求时填充字段，KUAB 响应时会原样返回。

通讯报文头部标准字段					
名称	类型	必须	示例	缺省值	描述
MsgType	string	Y	KSPB.410510		会话层功能号或平台应用层业务功能号
SendingTime	date	Y	2018-11-25 16:07:17.339		通讯报文发送时间
Context	string	Y	UUID		客户端消息唯一标识，用于异步请求上下文关联，服务端原样返回
Protocol	string	N	STD	STD	业务协议，STD 表示标准化 FS1.0 协议
Version	string	N	2.0	1.0	技术协议版本
FLG_WantTran	string	N	Y	N	透传标识，Y-不透传、N-透传，如果使用 FS1.0 协议，该标志必须为 Y
FLG_Encrypt	string	N	1	0	加密标志，值为 1 时表示使用统一接入加密方式，为 0 时表示使用业务系统对应的加密方式。
BASE64Fields	array	N	["IMG_DATA"]		二进制字段名 JSON 数组，如果要设置二进制参数，需要在该 JSON 数组中添加二进制字段名
AppIP	string	Y			客户端 IP 地址
AppMAC	string	Y			客户端 MAC 地址
AppName	string	Y			客户端程序名称
AppSign	string	Y			客户端程序二进制文件内容 MD5 值
ReplyCode	int	C	0		返回码，非 0 表示失败 (应答字段，由 KUAB 返回)
ReplyMsg	string	C	查询市场板块信		返回错误信息 (应答字

			息成功!!		段, 由 KUAB 返回)
--	--	--	-------	--	---------------

注: 必须选项中'C'表示条件必须, 下同

4.1.1.3.2. 会话层消息

4.1.1.3.2.1. 签入消息

- 第三方应用必须先签入后才能发送其他类型的消息。

签入消息 (MsgType= session_login)					
请求字段	类型	必须	示例	缺省值	描述
消息标准字段					
AppID	string	Y			开发者标识, 签入业务时送入
AppSecret	string	Y			开发者密码, 签入业务时送入, 使用开发者证书进行加密
应答字段	类型	必须	示例	缺省值	描述
消息头部标准字段					
WORKSKEY	string	Y			会话密钥, 后续报文加密和签名用, 第三方应用需要使用开发者证书进行解密后使用, 会话密钥跟连接为——对应关系, 连接断开后, 会话也随之无效, 重连后, 之前保存的会话密钥需要重置。

4.1.1.3.2.2. 签出消息

- 用于取消当前会话密钥, 一般很少使用, 连接断开时, 会话密钥也随之无效

签出消息 (MsgType= session_logout)					
请求字段	类型	必须	示例	缺省值	描述
消息标准字段					
应答字段	类型	必须	示例	缺省值	描述
消息头部标准字段					

4.1.1.3.2.3. 心跳消息

- 长连接时使用，用于第三方应用跟 KUAB 的心跳检测，判断 KUAB 是否可用

签出消息 (MsgType= session_logout)					
请求字段	类型	必须	示例	缺省值	描述
消息标准字段					
应答字段	类型	必须	示例	缺省值	描述
消息头部标准 字段					

4.1.1.3.3. 应用层消息

4.1.1.3.3.1. 业务消息

- 第三方应用在调用业务消息时，需要参考业务接口文档，在包体中填充业务请求字段

业务消息 (MsgType=业务功能号)					
请求字段	类型	必须	示例	缺省值	描述
消息头部标准 字段					
业务请求字段					
应答字段	类型	必须	示例	缺省值	描述
消息头部标准 字段					
业务应答字段					

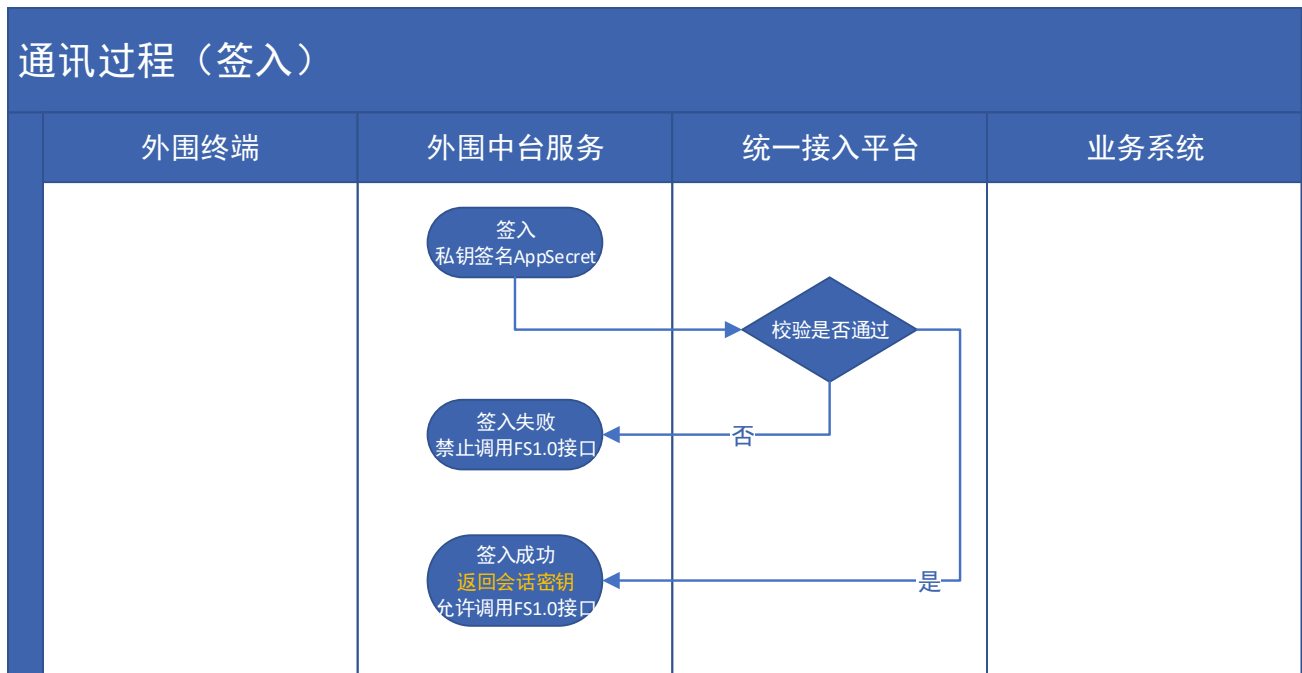
4.1.2 协议开发实现

4.1.2.1. 开放式接入开发实现

开放式接入开发实现由第三方应用自己实现整个报文的组装、加解密、发送、接收和解析过程，具体示例参见统一接入系统目录下 api\KDGP\C&C++\KUABKdgpTest

4.1.2.1.1. 接口实现

4.1.2.1.1.1. 签入



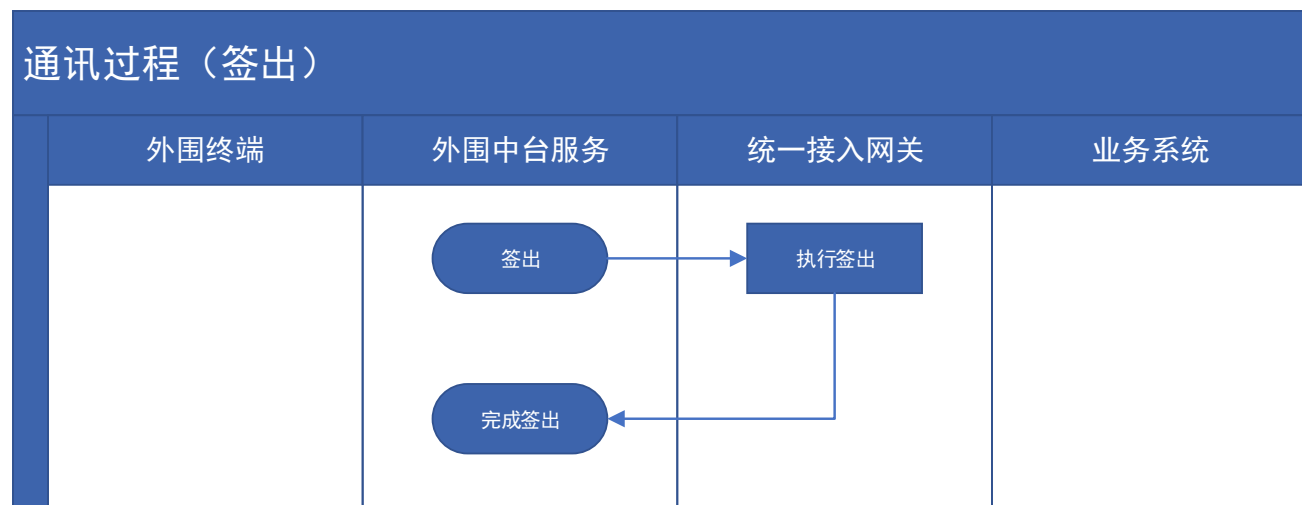
请求报文：

KDGP000001380000001700002962KDGB{"Context":"02FB0048-3","FLG_WantTran":"Y","MsgType":"**session_login**","Protocol":"STD","SendingTime":"2018-11-25 16:07:17.339"}JSTB{" AppID ":"xxxx","AppSecret":"yyyy"}

标红色的内容需要根据实际情况进行修改！

注：AppSecret 需要使用统一接入网关下发给终端的 RSA 证书进行加密，加密后的结果使用 BASE64 进行编码。

4.1.2.1.1.2. 签出



请求报文：

```
KDGP000001380000001700002962KDGB{"Context":"02FB0048-3","FLG_WantTran":"Y","MsgType":"session_logout","Protocol":"STD","SendingTime":"2018-11-25 16:07:17.339"}JSTB{}
```

标红色的内容需要根据实际情况进行修改！

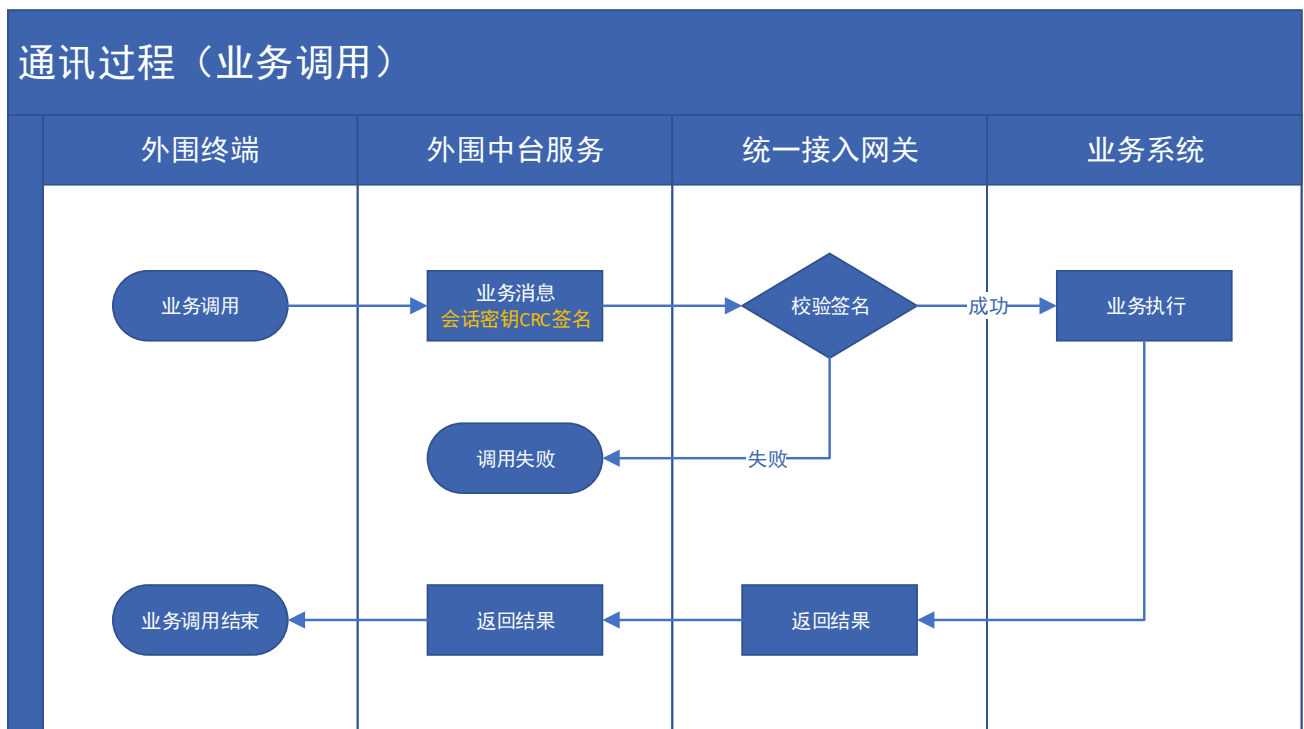
外围系统使用短连接时执行签出动作；如果使用长连接，不需要执行这个功能。

4.1.2.1.1.3. 心跳

```
KDGP000001380000001700002962KDGB{"Context":"02FB0048-3","FLG_WantTran":"Y","MsgType":"session_heartbeat","Protocol":"STD","SendingTime":"2018-11-25 16:07:17.339"}JSTB{}
```

标红色的内容需要根据实际情况进行修改！

4.1.2.1.1.4. 业务调用



请求报文：

KDGP000001380000001700002962KDGB{"Context":"02FB0048-3","FLG_WantTran":"Y","MsgType":"KGOB.L0300003","Protocol":"STD","SendingTime":"2018-11-25 16:07:17.339"}JSTB{"STKEX":"0"}

标红色的内容需要根据实际情况进行修改！

其中"MsgType":"KGOB.L0300003"、JSTB{"STKEX":"0"}要根据具体的功能进行调整。
MsgType 需修改为需调用的功能号，JSTB 要修改为调用功能的固定入参+业务参数。

应答报文：

KDGP000003040000019800006db7KDGB{"Context":"02FB0048-3","FLG_WantTran":"Y","MsgType":"KGOB.L0300003","Protocol":"STD","SendingTime":"2018-11-25 16:07:17.339","AppID":"KCBPCli","ReplyMsg":"查询市场板块信息成功!!","ReplyCode":"0","RefMsgType":"KGOB.L0300003","18ABA0EE0010CF16247AD259_STD":"KGOB.L0300003","RefProtocol":"STD"}JSTB[{"STKEX_NAME":"深圳 A","STKEX":"0","STKBD":"0"},{"STKEX_NAME":"深圳 B","STKEX":"0","STKBD":"1"},{"STKEX_NAME":"三板 S","STKEX":"0","STKBD":"2"},{"STKEX_NAME":"深港通","STKEX":"0","STKBD":"3"}]]

4.1.2.1.2. 特殊字段处理

4.1.2.1.2.1. 密码字段

使用签入返回的 WORKSKEY 作为密钥对字段值进行 DES 加密之后再进行 Base64 编码，其中 DES 密钥格式为 ECB，填充方式为 zeropadding。

4.1.2.1.2.2. 二进制字段

使用 Base64 对字段值进行编码，同时在消息标准字段 BASE64Fields 中标记二进制字段名，BASE64Fields 值为 JSON 数组格式，二进制字段名需要添加到 JSON 数组中。

4.1.2.2. API 方式接入开发实现

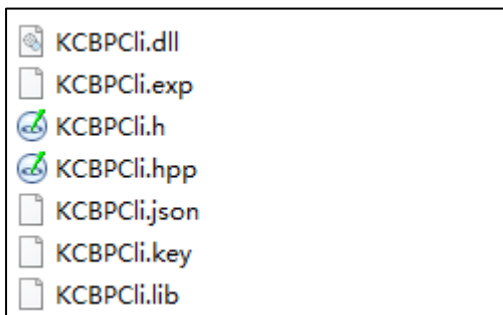
目前 KUAB 对 KDGP 协议进行了封装，支持 C/C++和 JAVA，第三方应用不需要关心底层的具体实现，只需要调用对应的 API 接口即可。

4.1.2.2.1. C/C++

4.1.2.2.1.1. API 说明



C/C++ API 包含以下文件:



KCBPcli.json 里面包含统一接入平台的地址、端口、使用密钥文件等配置信息，需要跟 dll 库文件放在同一目录下。KCBPcli.key 是开发者证书，需要由统一接入平台配置生成并提供给外围系统，跟 dll 放在同一个目录下，开发者证书名称要和开发者 ID 必须保持一致。

4.1.2.2.1.2. API 接口

C/C++ API 采用兼容 KCBPCLI 模式，具体 API 函数说明参考《KCBP 程序员手册.pdf》

具体示例参见统一接入系统目录下 api\KDGP\C&C++\KCBPCLI\example\clienttest.cpp

4.1.2.2.1.2.1. 签入

```
KCBPCLI_ConnectServer()
```

4.1.2.2.1.2.2. 签出

```
KCBPCLI_Exit()
```

4.1.2.2.1.2.3. 心跳

```
KCBPCLI_SQLExecute("session_heartbeat")
```

4.1.2.2.1.2.4. 业务调用

```
KCBPCLI_SQLExecute("业务功能号")
```

4.1.2.2.1.3. 特殊字段处理

4.1.2.2.1.3.1. 密码字段

- 1) 升级 API 版本到 2.0.2.0 以上
- 2) 在 KCBPCLI.json 里的 additional 增加 FLG_Encrypt:1 配置
- 3) 在字段名前添加@符号，调用 KCBPCLI_SetValue 接口进行字段参数设置

调用示例：

```
KCBPCLI_SetValue("@PASSWORD", "123456");
```

4.1.2.2.1.3.2. 二进制字段

- 1) 调用 KCBPCLI_SetVal 接口进行字段参数设置

调用示例：

```
KCBPCLI_SetVal("IMG_DATA", "二进制数据");
```

4.1.2.2.1.4. 配置文件说明

使用 API 之前需要先配置 KCBPCLI.json，确保开发者 ID、开发者密码、开发者证书、IP、端口跟 KUAB 提供的一致。

4.1.2.2.1.4.1. KCBPcli.json

```
{
  "name" : "统一接入 kcbpcli.dll 配置文件",
  "version": "1.0",
  "modify": "20170329",
  "AppID": "KFZID",
  "AppSecret": "63F686C5-F128-474D-9623-A102E3423003",
  "protocol": "STD",
  "tcp_nodelay": "1",
  "tcp_keepalive": "10",
  "trace": "0",
  "addition": "FLG_WantTran:Y, FLG_Encrypt:1, FLG_Version:2.0",
  "reconnect": "0",
  "connection": "127.0.0.1:22000:queue_req:queue_ans"
}
```

参数详细说明如下（带*的参数需要特别注意，是,否的配置分别对应 0,1）：

参数	名称	描述
name	描述配置文件	
version	描述配置文件版本	
modify	描述配置文件修改日期	
*AppID	开发者 ID	从 KUAB 申请
*AppSecret	开发者密码	从 KUAB 申请
*porotocol	业务协议	默认为 STD
tcp_nodelay	是否开启 nagle 算法	默认关闭
tcp_keepalive	检查心跳时间间隔	单位：秒，默认 10 秒
*trace	是否记录日志信息	测试时开启，生产时关闭
addition	额外配置信息	key:value 格式，多个用英文逗号分隔，用于扩充消息标准字段 FLG_WantTran:Y 表示是否需要在接入网关进行协议转换，Y 表示需要，N 表示不需要； FLG_Encrypt:1 表示使用统一接入提供的业务字段加密方式，为 0 时表示使用后台业务系统提供的加密方式 Version:2.0 表示使用 KDGP 协议 2.0 规范
*reconnect	是否重连	测试工具使用，开发时请自己实现重连，因为 dll 内部重连成功后会将业务进行重发，可能会引起问题

connection	KUAB 连接信息	格式：IP:端口:请求队列:应答队列，多个配置以英文逗号分隔，可以配置为空，使用 API 进行连接信息设置
------------	-----------	---

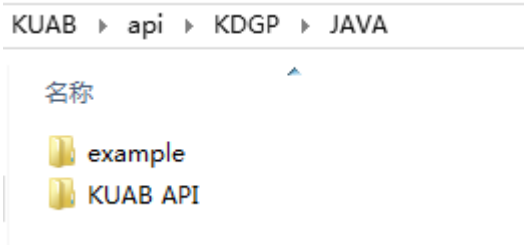
以上的参数都是可以在程序里面动态进行调整的。特别是 addition，对于某些系统需要协议转换，而某一些系统不需要的时候，就要在代码里面做动态调整。

4.1.2.2.2. JAVA

4.1.2.2.2.1. API 说明

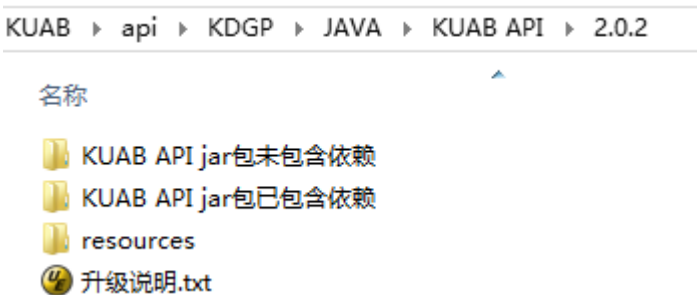


JAVA API 提供文件如下：



KUAB API 目录存放的是 jar 包及相关依赖文件，example 目录存放的是调用示例文件。

KUAB API 目录内容如下：



“KUAB API jar 包已包含依赖” 提供包含外部依赖的 jar 包

“KUAB API jar 包未包含依赖” 提供不包含外部依赖的 jar 包

“resources” 是 jar 包依赖的配置文件

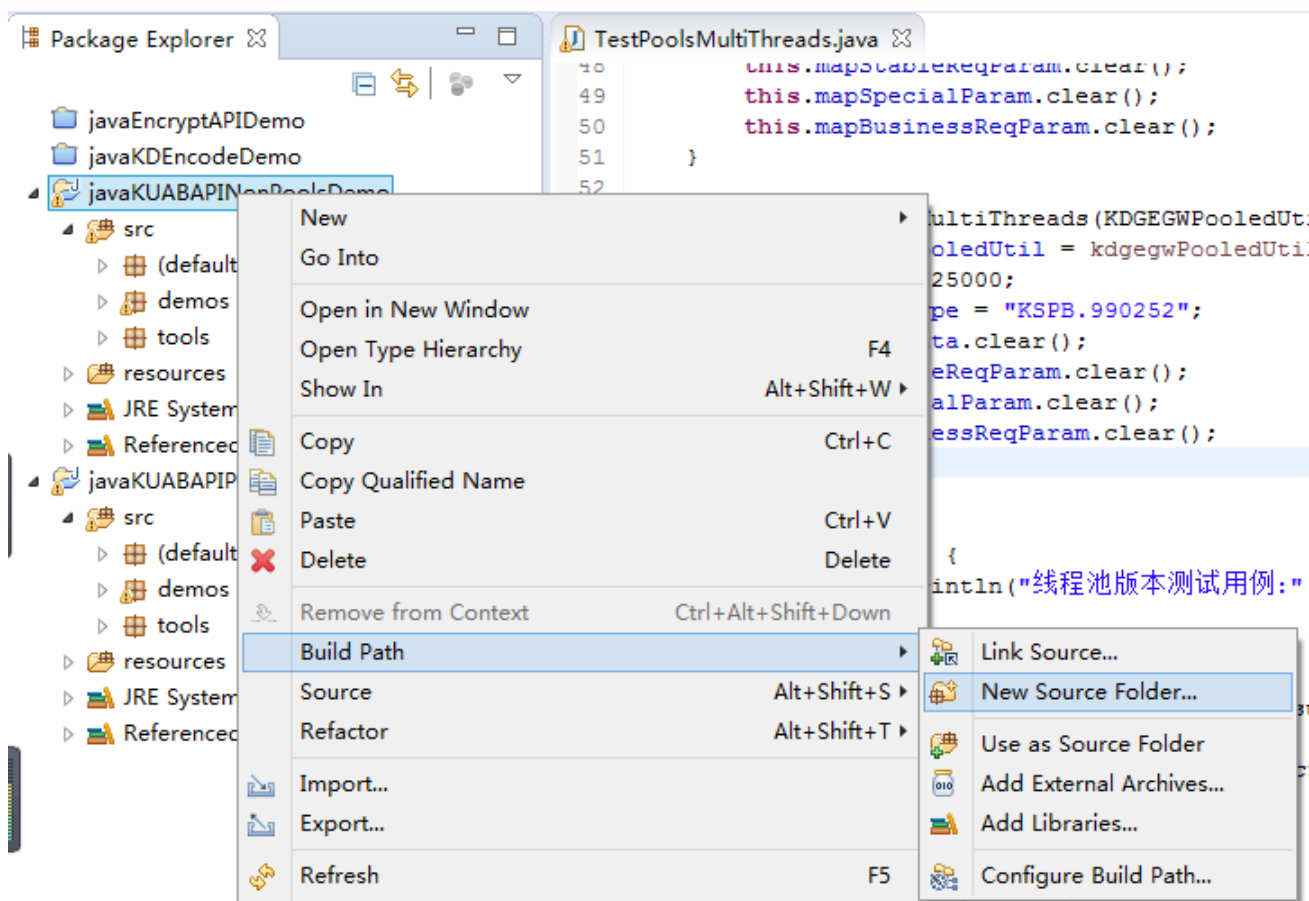
“升级说明.txt”是 jar 包相对上一个版本 jar 包的升级说明

jar 包依赖的配置文件:



- KUAB.json:
jar 包主配置文件，里面包含统一接入平台的地址、端口、使用密钥文件等配置信息，需要修改为实际的值
- KUAB.key:
开发者证书，需要由统一接入平台配置生成并提供给外围系统，证书名称需要和开发者 ID 相同
- PoolConfig.json:
连接池参数配置文件，一般不需要修改，使用默认配置即可

注意：配置文件必须放在 java 工程的 resources 目录，在 eclipse 下创建 resources 目录需要使用以下菜单进行操作



新建 resources 目录完成后，直接将相关配置文件拷贝进去即可

example 内容如下：

```
KUAB > api > KDGP > JAVA > example
```

名称
非连接池版本测试用例
连接池版本测试用例

example 下面有两个测试用例，具体使用步骤请参考 6.1.2 章节

4.1.2.2.2.2. API 接口

Jar 包提供了连接池和非连接池两种模式，下面介绍这两种模式的使用方式

4.1.2.2.2.2.1. 签入

1. 非连接池模式：

```
KdgegApi kdgegApi = KDGEgwUtil.getApi();
// 签入参数可以为空
Map<String, String> mapLoginReqParam = new HashMap<String, String>;
kdgegApi.login(mapLoginReqParam);
```

2. 连接池模式：

API 内部会自动签入，上层应用不需要关心签入逻辑

4.1.2.2.2.2.2. 签出

1. 非连接池模式：

```
// 签出参数可以为空
Map<String, String> mapLogoutReqParam = new HashMap<String, String>;
kdgegwApi.logout(mapLogoutReqParam, "", "");
```

外围系统使用短连接时执行签出动作；如果使用长连接，不需要执行这个功能。

2. 连接池模式：

API 内部会自动签出，上层应用不需要关心签出逻辑

4.1.2.2.2.2.3. 业务调用

1. 非连接池

```
kdgegwApi.submit(业务功能号, 特殊参数, 固定入参, 业务入参);
```

2. 连接池模式

```
KDGEGWPooledUtil kdgegwPoolUtil = new KDGEGWPooledUtil();
kdgegwPoolUtil.submit(业务功能号, 特殊参数, 固定入参, 业务入参);
```

业务功能号：金证 FS 协议功能号

特殊参数：一般情况下不需要设置

固定入参：金证 FS 协议的固定入参部分

业务入参：金证 FS 协议的业务入参部分

4.1.2.2.2.3. 特殊字段处理

4.1.2.2.2.3.1. 密码字段

1) 升级 API 版本到 1.1.6 以上

2) 在 KUAB.json 里的 additional 增加 FLG_Encrypt:1 配置

3) 字段名称前加@符号

调用示例：

```
// 业务入参部分
Map<String, Object> mapBusinessReqParam = new HashMap<String, Object>();
```

```
mapBussinessReqParam.put("@PASSWORD", "123321");
```

4.1.2.2.3.2. 二进制字段

1) 在字段名前加\$符号

调用示例:

```
// 业务入参部分
Map<String, Object> mapBussinessReqParam = new HashMap<String, Object>();

mapBussinessReqParam.put("$IMG_DATA", "二进制数据");
```

4.1.2.2.2.4. 配置文件说明

使用 API 之前需要先配置 KUAB.json, 确保开发者 ID、开发者密码、开发者证书、IP、端口跟 KUAB 提供的一致。

4.1.2.2.2.4.1. KUAB.json

```
{
  "name":"KUAB JAR",
  "version":"1.0",
  "modify":"20150915",
  "AppID":"KUAB",
  "AppSecret":"DC5F7A47-20FC-4120-B333-5CF98D04512B",
  "protocol":"STD",
  "addition":"FLG_WantTran:Y,FLG_Encrypt:1, FLG_Version:2.0",
  "tcp_nodelay":"1",
  "tcp_keepalive":"1",
  "tcp_timeout":"5000",
  "reconnect":"1",
  "connection":"127.0.0.1:22000",
  "trace":"0"
}
```

AppID 开发者 ID, AppSecret 开发者密码, 需要统一接入平台提供。

Connection 参数设置为统一接入平台的接入地址和端口。

参数详细说明如下 (带'*'的参数需要特别注意, 是,否的配置分别对应 0,1):

参数	名称	描述
name	描述配置文件	
version	版本	
modify	修改日期	
*AppID	开发者 ID	从 KUAB 申请
*AppSecret	开发者密码	从 KUAB 申请

*porotocol	业务协议	默认为 STD
tcp_nodelay	是否开启 nagle 算法	默认关闭
tcp_keeplive	是否检查心跳时间间隔	0:关闭 1:开启 默认开启, 连接池模式由于内部自动做了应用层心跳, 建议选择关闭
*trace	是否记录日志信息	测试时开启, 生产时关闭
addition	额外配置信息	key:value 格式, 多个用英文逗号分隔, 用于扩充消息标准字段 FLG_WantTran:Y 表示是否需要在 KUAB 进行协议转换, Y 表示需要, N 表示不需要; FLG_Encrypt:1 表示使用统一接入提供的业务字段加密方式, 为 0 时表示使用后台业务系统提供的加密方式 Version:2.0 表示使用 KDGP 协议 2.0 规范
tcp_timeout	超时时间	单位: 毫秒
*reconnect	是否重连	建议配置为 1
connection	KUAB 连接信息	格式: IP:端口, 多个配置以英文逗号分隔

以上的参数都是可以在程序里面动态进行调整的。特别是 addition, 对于某些系统需要协议转换, 而某一些系统不需要的时候, 就要在代码里面做动态调整, addition 配置内容对应 submit 的特殊参数。

4.1.2.2.2.4.2. PoolConfig.json

```
{
  "maxTotal":"100",
  "maxIdle":"50",
  "maxWait":"5000",
  "testWhileIdle":"true",
  "timeBetweenEvictionRunsMillis":"30000"
}
```

参数	名称	描述
maxTotal	最大连接个数	maxTotal=100, 表示连接池最多与统一接入网关建立 100 个连接
maxIdle	最大空闲连接个数	当超过阈值时, 多余的空闲对象会被销毁, 一般设置为 maxTotal 的一半
maxWait	最大等待可用连接时间	当超过阈值时, 抛出"对象池已无可

		使用连接"异常信息
testWhileIdle	空闲连接有效性检查	false 表示不检查, true 表示检查, 这个参数与 "timeBetweenEvictionRunsMillis" 共同作用
timeBetweenEvictionRunsMillis	空闲连接有效性检查周期	当 testWhileIdle 为 true 时, 启用异步检查, 定期向网关发送应用层心跳

4.2 KHTTP 技术协议

KDGP 协议是统一接入平台对外提供的标准 HTTP 应用协议, 以下是对该协议的详细描述。

4.2.1 协议描述

4.2.1.1. 报文格式

请求包体: 请求参数字段使用 application/x-www-form-urlencoded 格式, 即 key=value 的形式, 多个键值对采用&进行连接, 消息标准字段前面需要加 "__" (两个下划线)前缀进行区分, 非消息标准字段前面不能出现 "__"。

应答包体: 采用 JSON 格式。

报文签名: 除了签入报文以外, 所有的物理报文必须经过签名。

签名过程如下:

1. 将 key 按照字典序从小到大排列, 如果 key 对应的 value 是 JSON 格式, JSON 内部也需要按 key 进行排序, 然后将排序后的 key 和 value 循环进行累加;
2. 将累加后的值使用 MD5 签名, 生成后的值赋给 Sign, 一起作为请求参数。

签名示例:

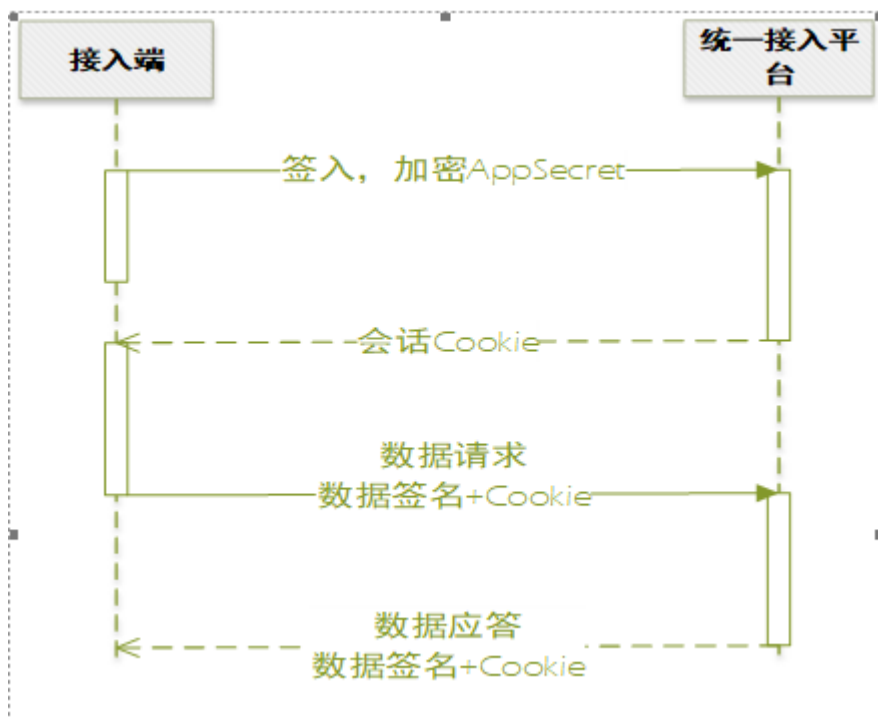
KV 格式签名:

请求参数采用 KV 模式, 假设请求参数为 key2=value2&key1=value1, 则排序后累加的值 key1value1key2value2;

JSON 格式签名:

返回结果采用 JSON 模式，假设返回结果为{"key2":[{"b":"2","a":"1"}],"key1":"value1"}, 则排序累加后的值为 key1value1key2[{"a":"2","b":"1"}]

4.2.1.2. 通讯过程



4.2.1.3. 消息类型

4.2.1.3.1. 消息标准字段

第三方应用需要在包头填充消息标准字段，以下表格中描述了消息标准字段，如非特殊说明，则表示请求时填充字段，KUAB 响应时会原样返回，KHTP 协议的消息标准字段前面需要加 “__” (两个下划线)前缀进行区分。

通讯报文头部标准字段					
名称	类型	必须	示例	缺省值	描述
MsgType	string	Y	KSPB.410510		会话层功能号或平台应用层业务功能号
SendingTime	date	Y	2018-11-25 16:07:17.339		通讯报文发送时间
Context	string	Y	UUID		客户端消息唯一标识，用于异步请求上下文关联，服务

					端原样返回
Protocol	string	N	STD	STD	业务协议，STD 表示标准化 FS1.0 协议
Version	string	N	2.0	1.0	技术协议版本
FLG_WantTran	string	N	Y	N	透传标识，Y-不透传、N-透传，如果使用 FS1.0 协议，该标志必须为 Y
FLG_Encrypt	string	N	1	0	加密标志，值为 1 时表示使用统一接入加密方式，为 0 时表示使用业务系统对应的加密方式。
RandomNo	string	Y			自定义随机值，加密 AppSecret, Cookie 的 DES 密钥，自身使用发送时间 SendingTime 加密
Sign	string	Y			数据签名
Cookie	string	C			会话 Cookie，签入成功后返回，后续业务请求必填
BASE64Fields	array	N	["IMG_DATA"]		二进制字段名 JSON 数组，如果要设置二进制参数，需要在该 JSON 数组中添加二进制字段名
AppIP	string	Y			客户端 IP 地址
AppMAC	string	Y			客户端 MAC 地址
AppName	string	Y			客户端程序名称
AppSign	string	Y			客户端程序二进制文件内容 MD5 值
Code	int	C	0		返回码，非 0 表示失败 (应答字段，由 KUAB 返回)
Msg	string	C	查询市场板块信息成功!!		返回错误信息 (应答字段，由 KUAB 返回)
Data	string	C			返回数据 (应答字段，由

					KUAB 返回)
--	--	--	--	--	----------

注：必须选项中‘C’表示条件必须，下同

4.2.1.3.2. 会话层消息

4.2.1.3.2.1. 签入消息

- 第三方应用必须先签入后才能发送其他类型的消息。

签入消息 (MsgType= session_logon)					
请求字段	类型	必须	示例	缺省值	描述
消息头部标准 字段					
AppID	string	Y			开发者标识，签入业务时送入
AppSecret	string	Y			开发者密码，签入业务时送入，签入时 AppSecret 的加密使用 DES 加密，密钥格式为 ECB，填充方式为 zeropadding，然后再进行 BASE64 编码，DES 密钥为签入时传入的 RandomNo 值，其中 RandomNo 为任意值(建议为可读字符)，自身采用发送时间 SendingTime 作为 key 进行 DES 加密；签入后，第三方应用需要对 Cookie 进行解密后保存使用，Cookie 的加密方式和 AppSecret 相同
应答字段	类型	必须	示例	缺省值	描述
消息标准字段					
Cookie	string	Y			会话密钥，后续报文加密和签名用，第三方应用需要使用开发者证书进行解密后使用，会话密钥跟连接为——对应关系，连接断开后，会话也随之无效，重连后，之前保

					存的会话密钥需要重置。
--	--	--	--	--	-------------

4.2.1.3.2.2. 签出消息

- 用于取消当前会话密钥，一般很少使用，连接断开时，会话密钥也随之无效

签出消息 (MsgType= session_logout)					
请求字段	类型	必须	示例	缺省值	描述
消息标准字段					
应答字段	类型	必须	示例	缺省值	描述
消息标准字段					

4.2.1.3.2.3. 心跳消息

- 长连接时使用，用于第三方应用跟 KUAB 的心跳检测，判断 KUAB 是否可用

签出消息 (MsgType= session_logout)					
请求字段	类型	必须	示例	缺省值	描述
消息标准字段					
应答字段	类型	必须	示例	缺省值	描述
消息标准字段					

4.2.1.3.3. 应用层消息

4.2.1.3.3.1. 业务消息

- 第三方应用在调用业务消息时，需要参考业务接口文档，在包体中填充业务请求字段

业务消息 (MsgType=业务功能号)					
请求字段	类型	必须	示例	缺省值	描述
消息头部标准字段					
业务请求字段					开发者标识，请求时送入
应答字段	类型	必须	示例	缺省值	描述
消息头部标准字段					
业务应答字段					

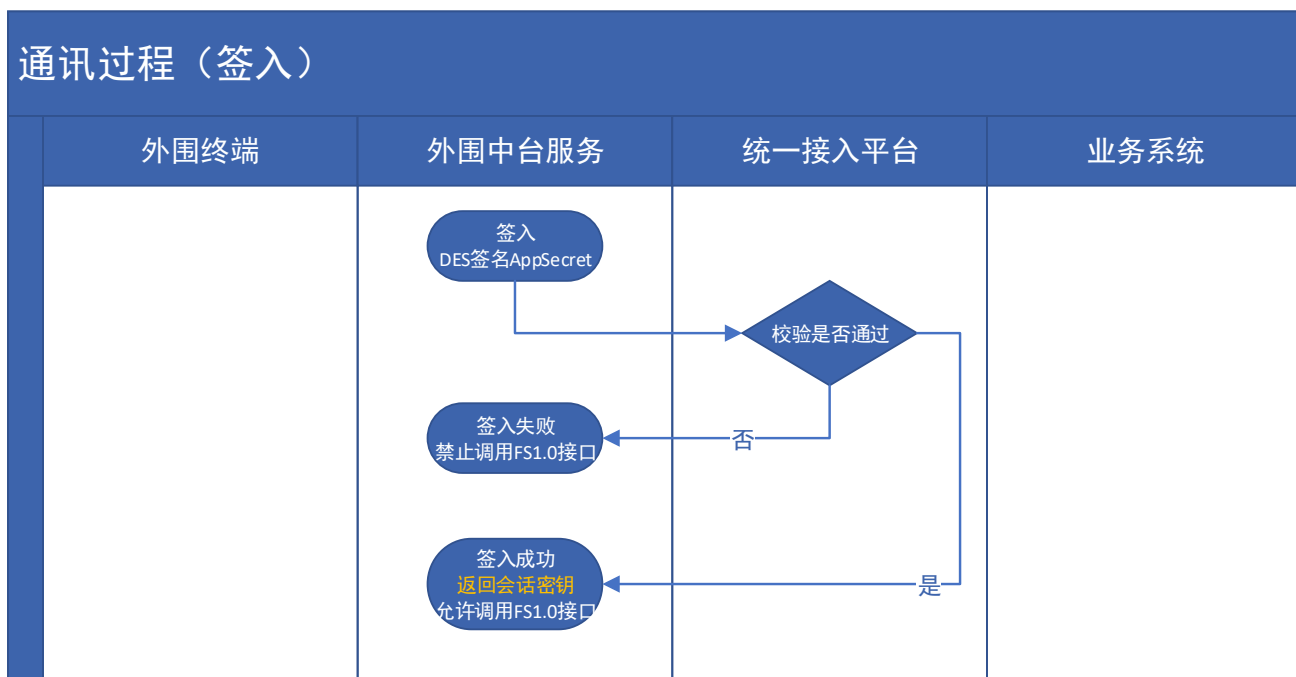
4.2.2 协议开发实现

4.2.2.1. 开放式接入开发实现

4.2.2.1.1. 接口实现

开放式接入开发实现由第三方应用自己实现整个报文的组装、加解密、发送、接收和解析过程，具体示例参见统一接入系统目录下 api\HTTP\C&C++\ KUABWebserviceTest

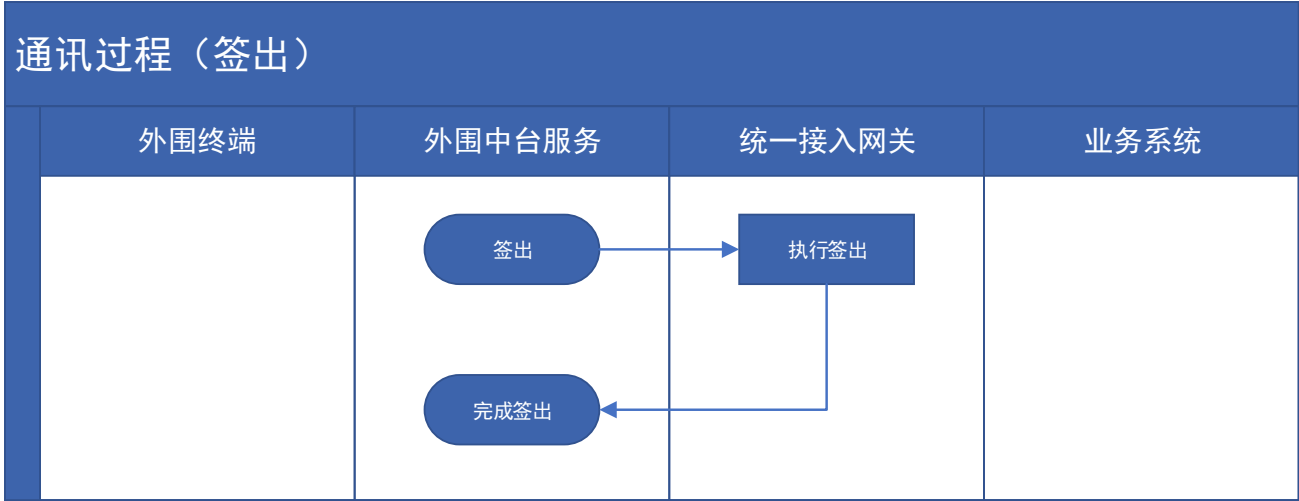
4.2.2.1.1.1. 签入



请求报文：

__AppID=KCBPCli&__AppSecret=mo95ZzCvd/5JlN4Eh/pKY4fygjmBGoZr/gMWVZQeKSf3sK
 FT1yOrJA==&__MsgType=session_logon&__RandomNo=MO1ub5D9g+o=&__SendingTime
 =2016-09-07 16:06:26.247&__Sign=e63a2c2e6ea791ab014faafb1615b955

4.2.2.1.1.2. 签出



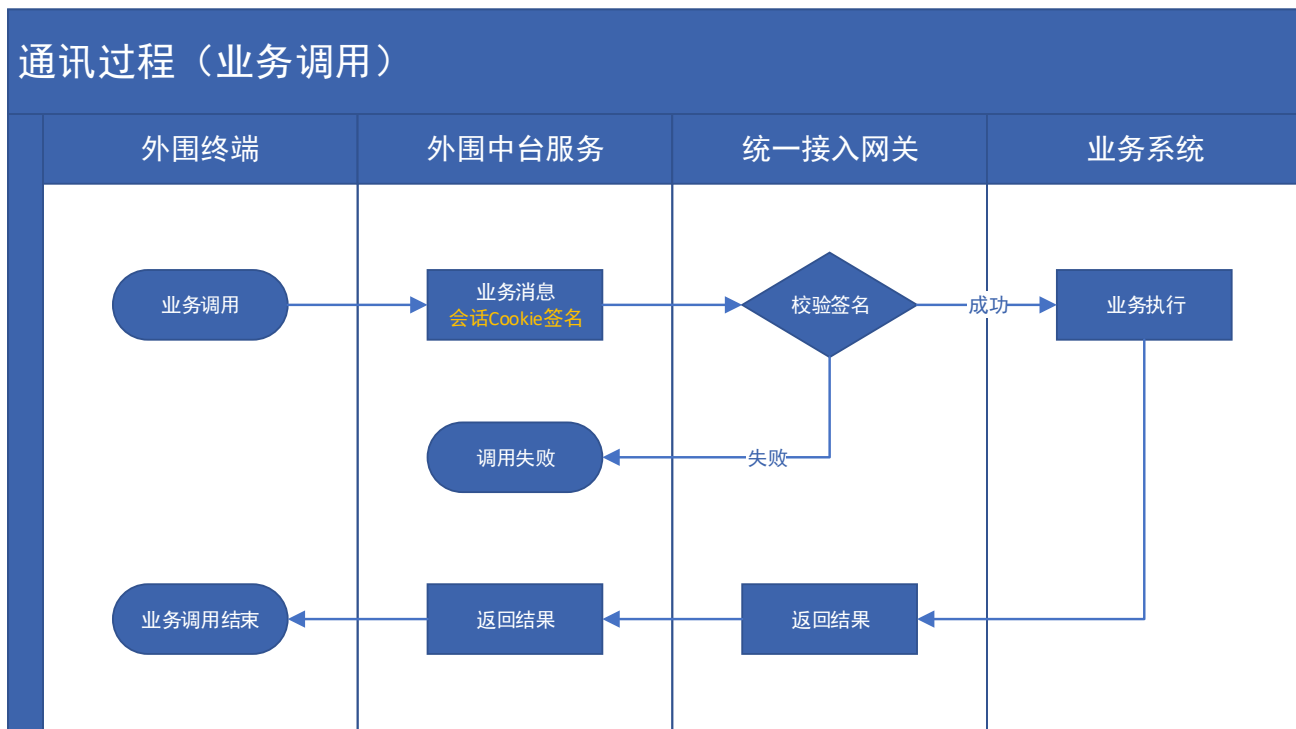
请求报文：

__AppID=KCBPCli&__Cookie=37LBUhccBZlLgQFFZvJKlzh1E7qbN8IIRpS76UWzASaH9+oqMAc8Qcvxnailna7L5kM4BFpuTDuG7nvLltcMKiXpNdEIM+r&__MsgType=session_logout&__SendingTime=2016-09-07 16:09:07.708&__Sign=353b6a9a6e85211025a203e9ee558358

外围系统使用短连接时执行签出动作；如果使用长连接，不需要执行这个功能。

4.2.2.1.1.3. 心跳

4.2.2.1.1.4. 业务调用



请求报文：

__AppID=KCBPCli&__Context=00007FF6CFE99CB8-15&__Cookie=LUP9FjolAi%2F1m0VT1lcHxbeZYKq9X2CeGTMJCWroQBO%2FuqWmIEplfHHxttviW%2BQK3OVAJB6xNB%2BaA6a2k%2BK1MJTpqtY%2FeAW2a4XbzmCvn25NvpGNNYYQ8saJzM6%2FEEExN&__FLG_Encrypt=1&__FLG_WantTran=N&__MsgType=410510&__RandomNo=sKn4arg81WU%3D&__SendingTime=2018-11-26+14%3A34%3A06.164&__Sign=952a2bb54b0ff1dd6eddf002984b5179&custid=800100000002

应答报文：

```

{"Context":"00007FF6CFE99CB8-15","Code":"0","Msg":"请求成功",
,"Data":[{"custid":"800100000002","fundid":"10100008422","orgid":"800","moneytype":"0","fundbal":"123456789.0","fundavl":"123456789.0","marketvalue":"N","fund":"123456789","stkvalue":"123456789","fundseq":"0","fundloan":"0"},{"custid":"800100000002","fundid":"10100008422","orgid":"800","moneytype":"0","fundbal":"123456789.0","fundavl":"123456789.0","marketvalue":"N","fund":"123456789","stkvalue":"123456789","fundseq":"0","fundloan":"0"},{"custid":"800100000002","fundid":"10100008422","orgid":"800","moneytype":"0","fundbal":"123456789.0","fundavl":"123456789.0","marketvalue":"N","fund":"123456789","stkvalue":"123456789","fundseq":"0","fundloan":"0"}],"Sign":"3baf3d7902030f3b1b2abdd31959b300","SendingTime":"2018-11-26 14:35:05.865"}
    
```


4.2.2.1.2. 特殊字段处理

4.2.2.1.2.1. 密码字段

- 1) 在请求参数中增加__FLG_Encrypt=1
- 2) 使用 RandomNo 对密码字段进行加密

4.2.2.1.2.2. 二进制字段

使用 Base64 对字段值进行编码，同时在消息标准字段__BASE64Fields 中标记二进制字段名，__BASE64Fields 对应的值应为 JSON 数组格式。