

微众银行信贷**开放平台开发指南**

一、前言

本文目的在于指导用户获取访问令牌(access token)和票据(ticket), 为后续业务接入做好准备工作。

注: 所有场景默认采用UTF-8编码

二、网络接入

开放平台和第三方服务器之间的通信采用单向HTTPS, 接入区域分公网和专线接入。分配的参数中, 测试和生产只有域名、secret、回调加密的盐值会变, url路径保持不变。

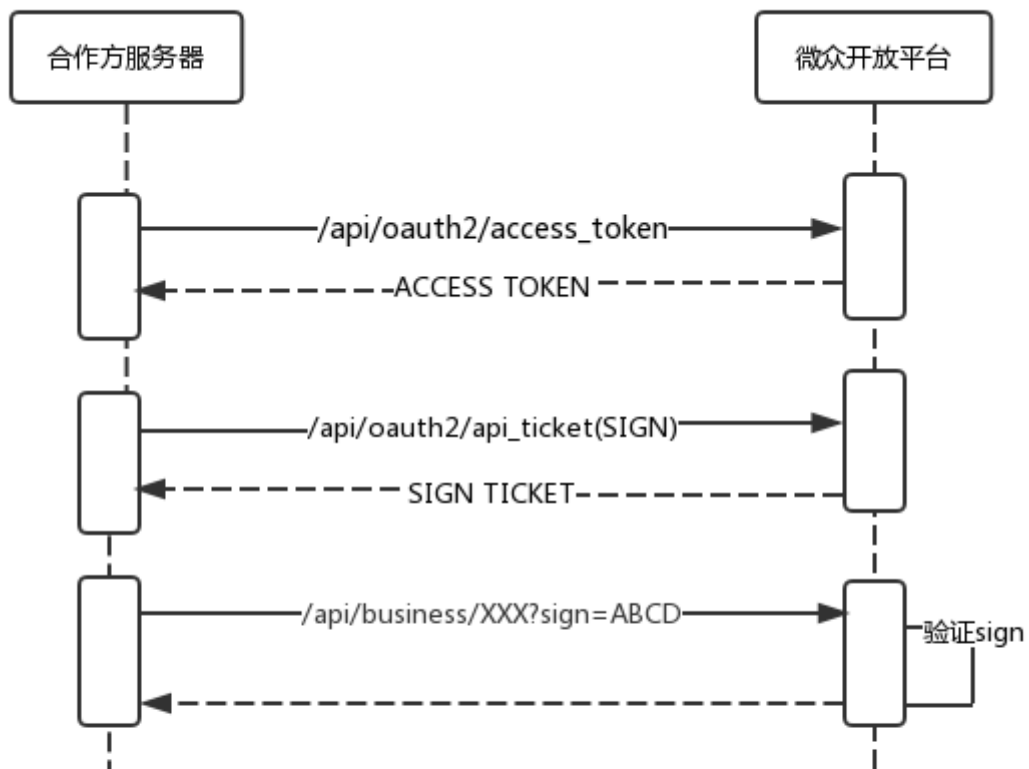
注: 微众提供的接入DEMO运行环境要求在JDK1.7及以上版本。

开墙策略由微众产品经理填写合作方ip信息递交运维处理, 添加ip白名单。

三、一般交互流程

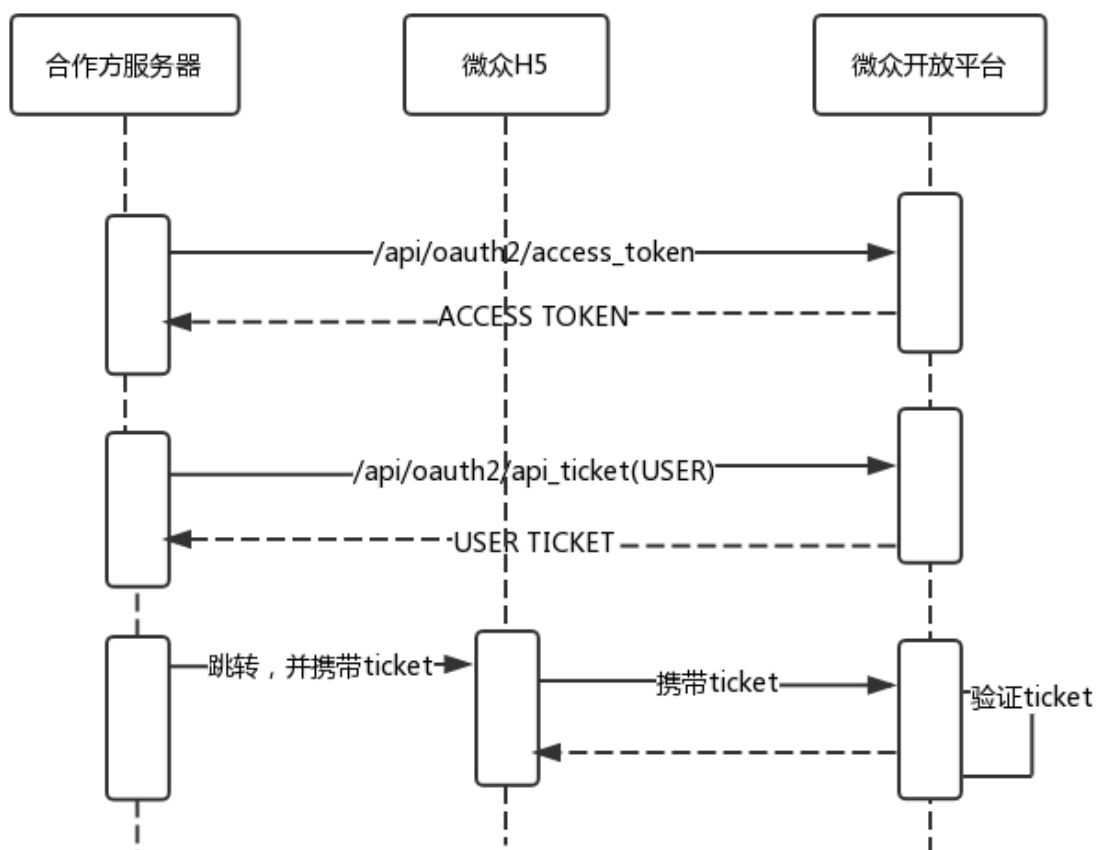
合作方与微众银行接入交互, 分为两种场景: 合作方服务器后台接入, 即平台纬度授权接入; 合作方APP或公众号前端接入, 嵌入微众H5页面, 即用户纬度授权接入。两种场景的授权验证交互流程如下:

3.1 平台纬度授权



合作方服务器和微众开放平台之间的通信使用双向HTTPS
access token有效期3600秒
sign ticket有效期3600秒
access token和sign ticket全局唯一
合作方必须再临近过期和使用中失效时重新获取

3.2 用户纬度授权



合作方服务器和微众开放平台之间的通信使用双向HTTPS
 access token与平台纬度授权一致，有效期3600秒
 user ticket有效期300秒
 access token和user ticket全局唯一
 获取user ticket需要appid、userid、mechant_jd信息
 user ticket单次有效，登录失效时需合作方重新获取

四、合作方调用微众

4.1 获取access_token

access_token用于验证接入方合法性。

dev域名：<https://f-capapi-v2.test.webank.com>

测试环境F1域名：<https://f1-capapi-v2.test.webank.com>

测试环境G1域名：<https://g1-capapi-v2.test.webank.com>

URL路径：/api/oauth2/access_token

方法：get

URL参数（非BODY内容）：

name	comment	value
app_id	应用ID	微众开放平台分配（生产和测试一致）
secret	应用密钥	微众开放平台分配（生产和测试不一致）
grant_type	授权类型	client_credential
version	版本号	2.0.0

响应：

```
{
  "msg": "系统处理成功",
  "access_token":
    "55350620a1eec3c6842a9cfcc01e9effa31af2b4021c49d535f65358139f8d2d",
  "code": "60850000",
  "expire_time": "1516178421000",
  "expire_in": "3600",
  "transactionTime": "20180117154021"
}
```

注：

1. code不是60850000是表示获取失败，可以根据code和msg字段定位和调试。
2. 不同的type的ticket在List中依次排列
3. expire_time为access_token失效的绝对时间戳，单位毫秒。
4. expire_in为access_token的最大生存时间，单位秒，默认为3600秒。
5. 修改secret之后，该app_id生成的access_token和ticket都失效。
6. transactionTime 生成token时间

4.2 获取/更新API票据(ticket)

ticket用于对请求数据签名或加密。

dev域名：<https://f-capapi-v2.test.webank.com>

测试环境F1域名：<https://f1-capapi-v2.test.webank.com>

测试环境G1域名：<https://g1-capapi-v2.test.webank.com>

URL路径：/api/oauth2/api_ticket

方法：get

URL参数（非BODY内容）：

name	comment	value
app_id	应用ID	微众开放平台分配
access_token	app_id的访问令牌	token接口获取的值
type	ticket类型	见附录1
version	版本号	2.0.0
user_data	用户信息	type为用户时必须填json字符串
mechant_id	商户Id	type为用户时必须填

使用说明：

1. 不同的OPENAPI需要使用不同的type，一次输入多个type用英文逗号分隔。
2. 当type为sign时，使用sign ticket，主要用于后台接入，这里不需要传user_id和mechant_id。
3. 当type为用户时，使用user ticket，主要用于页面接入，user_id和mechant_id为必填参数。
4. user_data里的EXTEND_INFO字段可根据具体需求塞一些字段，格式为json字符串

user_data举例:

```
{
  "USER_ID": "平台用户ID（必填，String）",
  "ID_TYPE": "证件类型（可选，String）",
  "ID_NO": "证件号码（可选，String）",
  "USER_NAME": "姓名（可选，String）",
  "MOBILE": "手机号（可选，String）",
  "UNION_ID": "微信union id（可选，String）",
  "EXTEND_INFO": "扩展字段（可选，jsonString）"
}
```

EXTEND_INFO举例:

```
"EXTEND_INFO": {
  "DRIV_LICENSE_INFO": [{
    "PLATE_NUM": "车牌号C(12)",
    "ENGINE_NO": "发动机号C(20)",
    "VIN_CODE": "车辆识别代码C(32)",
    "OWNER_ID_NO": "行驶证身份证号C(30)"}
  ]
}
```

响应：

```
{
  "msg": "系统处理成功",
  "code": "60850000",
  "tickets": [
    {
      "expire_time": "1516178995000",
      "expire_in": 3600,
      "value":
"3e1b6bc4c77ba284ecd7ca8eb026a261c1d9fe8305c241e90771cf242633f05a"
    }
  ],
  "transactionTime": "20180117154955"}
}
```

注:

1. code不是60850000是表示获取失败，可以根据code和msg字段定位和调试。
2. expire_in为ticket的最大生存时间，单位秒，平台纬度默认为3600秒，用户纬度默认为300秒。
3. expire_time为ticket失效的绝对时间戳，单位毫秒。
4. value为 ticket 值。
5. transactionTime 生成ticket时间。
6. 为了简单方便，建议 API ticket 的刷新机制与 Access Token 定时机制原理一致，建议按照每 20 分钟和 Access Token 绑定定时刷。

4.3 后台联机业务请求

dev域名：<https://f-capapi-v2.test.webank.com>

测试环境F1域名：<https://f1-capapi-v2.test.webank.com>

测试环境G1域名：<https://g1-capapi-v2.test.webank.com>

URL路径：api/s/xxx/{service_code}（URL待分配，service_code为业务系统接口文档里具体的接口ID）

方法：只支持使用POST

body类型：application/json

URL参数（非BODY内容）：

name	comment	value
app_id	应用ID	微众开放平台分配
nonce	随机数	32位字符串（字母+数字）
sign	签名	具体参考签名算法
version	版本号	2.0.0
type	同步数据类型	可选填，可为空

使用说明：

业务接口字段全部以Json格式编码成一个字符串，并将整个字符串作为POST Body并参与签名。可参考附录3和附录4。

业务接口HTTP请求URL（拿到分配的参数之后，举例其中一种情况）：

```
https://f-capapi-v2.test.webank.com/api/s/yh/10002?
app_id=wxhiw5xt6q4qtmwcrn&nonce=489c60c97daef599efd10cfda46860a1&sign=07DC5F5033
3B99D1D99C80AEE3863701C107C6B7&version=2.0.0
```

业务接口HTTP报文（举例其中一种情况）：

```
{ "VERSION": "v1.0", "BIZ_SEQ_NO": "07275627318605619635664897877820", "SYS_SEQ_NO": "
94171967313166940672238906817817", "REQUEST_TIME": "20200226230000", "PRODUCT_CODE"
: "290001", "CHANNEL_ID": "01", "LEGAL_ENTITY": "", "TXN_BRANCH_CODE": "", "TXN_TELLER_I
D": "", "CAPITAL_SIDE": "1231315", "TXN_ID": "182063", "CTFS_APP_ID": "wb5pd6b3ux71lg35
v8", "USER_ID": "0999960200760245", "PS_CODE": "PS500101", "SERVICE_CODE": "182063", "I
NSURANCE_ID": "13213135", "ID_TYPE": "01", "ID_NO": "542221199608233947", "NBS_ORDER_N
O": "123456789", "APP_EXPIRY_DATE": "20200227", "ORI_BIZ_SEQ_NO": "072756273186056196
35664897877819", "NEREST_RATE": "0.23976", "APPLY_LIMIT_AMT": "121313.77", "APS_STATU
S": "SUCC" }
```

响应（举例其中一种情况）：

```
{
  "BizSeqNo": "23584462869882334698177803729161",
  "ConsumerSeqNo": "20170911100904727419000550615401",
  "transactionTime": "20151022044027",
  "code": "60840000",
  "msg": "系统处理成功",
  "jsonData": "{ \"CODE\": \"66590000\", \"DESC\": \"交易成功\" }"
}
```

注：

1. transactionTime 为服务器处理完成时间。
2. BizSeqNo和ConsumerSeqNo为微众流水号，供查询定位单笔交易使用。
3. code取值说明（分开放平台层返回码和业务层返回码）
 - 3.1 code为60840000时，表示开放平台层正常，可进一步取jsonData里面的业务系统返回码进行判断；
 - 3.2 当code前四位为6084，且后四位不为0时，表示开放平台接入异常，一般无jsonData；
 - 3.3 当code为60841017时，表示ticket不存在或已失效，需重新获取ticket凭据。
 - 3.4 当code为60841018时，表示参与签名的ticket不正确。
 - 3.5 当code为60841018和60841017时，建议重新获取ticket，防止定时获取出现异常影响一段时间交易。

4.3.1 签名sign生成算法

1. 生成一个 32 位的随机字符串 nonce，将app_id、version、type、ticket、nonce连同httpbody的json字符串进行字典序排序，将排序后的所有参数字符串拼接成一个字符串进行SHA1编码。
2. SHA1编码后的40位字符串作为 sign。
3. 签名算法代码（Hashing使用的是com.google.common.hash.Hashing，版本是guava-18.0）

```
public static String sign(List<String> values, String signTicket) {
    if (values == null) {
        throw new NullPointerException("values is null");
    }
    values.removeAll(Collections.singleton(null)); // remove null
    values.add(signTicket);
    java.util.Collections.sort(values);

    StringBuilder sb = new StringBuilder();
    for (String s : values) {
        sb.append(s);
    }
    return Hashing.sha1().hashString(sb,
    Charsets.UTF_8).toString().toUpperCase();
}
```

4.4 完整的请求代码示例

- 首先获取Token和ticket再用ticket对数据进行签名

```
String accessToken = client.getAccessToken();//http请求
String signTicket = client.getSignTicket(accessToken); //http请求
```

- 请求url参数和body:

```
app_id=wxhiw5xt6q4qtmwcrn
version=2.0.0
nonce=489c60c97daef599efd10cfda46860a1（随机数）
ticket=58b8ea10e5487b88e555feec27fb48201fd788c05b2d424234537b73c7282b70
type=（type为可选填，此处为空）
http body: {"REQUEST_TIME": "20190829163531", "SERVICE_CODE": "10002",
"WX_OPEN_ID": "oAUZp5LGzwcXkP1k7m8N16TKIac8", "SF_SUC_TIME": "20191124003946",
"ID_NO": "411528199110024452", "DEVICE_TYPE": "1", "DEVICE_LOGIN_COUNT": "0",
"MERCHANT_ID": "260125411000001", "WX_UNION_ID": "otko4v0iektCBqq8f4jaiifu7jzA"}
```

- 使用signTicket对url参数和body字典排序签名:


```
List values = new ArrayList();
values.add(app_id);
values.add(version);
values.add(nonce);
values.add(type);
values.add(httpBody);
String sign= SignUtils.sign(values, signTicket);
```

签名参数列表:

```
[wxhiw5xt6q4qtmwcrn, 2.0.0, 489c60c97daef599efd10cfda46860a1, null,
{"REQUEST_TIME": "20190829163531", "SERVICE_CODE": "10002", "WX_OPEN_ID":
"oAUZp5LGzWcXKP1k7m8N16TK1ac8", "SF_SUC_TIME": "20191124003946", "ID_NO":
"411528199110024452", "DEVICE_TYPE": "1", "DEVICE_LOGIN_COUNT": "0",
"MERCHANT_ID": "260125411000001", "WX_UNION_ID":
"otk04v0ieKtCBqq8f4jaiifu7jzA"}]
```

字典排序后, 签名拼接字符串:

```
2.0.0489c60c97daef599efd10cfda46860a158b8ea10e5487b88e555feec27fb48201fd788c05b2
d424234537b73c7282b70wxhiw5xt6q4qtmwcrn{"REQUEST_TIME": "20190829163531",
"SERVICE_CODE": "10002", "WX_OPEN_ID": "oAUZp5LGzWcXKP1k7m8N16TK1ac8",
"SF_SUC_TIME": "20191124003946", "ID_NO": "411528199110024452", "DEVICE_TYPE":
"1", "DEVICE_LOGIN_COUNT": "0", "MERCHANT_ID": "260125411000001", "WX_UNION_ID":
"otk04v0ieKtCBqq8f4jaiifu7jzA"}
```

计算 SHA1 得到: 07DC5F50333B99D1D99C80AEE3863701C107C6B7。该字符串就是最终生成的签名。统一使用大写。

- 生成url参数

```
https://f-capapi-v2.test.webank.com/api/s/yh/10002?
app_id=wxhiw5xt6q4qtmwcrn&nonce=489c60c97daef599efd10cfda46860a1&sign=07DC5F5033
3B99D1D99C80AEE3863701C107C6B7&version=2.0.0
```

4.5 后台文件接口

使用说明:

1. cap-up 域名专门用来文件上传下载, 域名为单向认证方式, 无需加载客户端证书。
2. OAUTH 鉴权方式不变, 参考4.3.1。图片内容不参与 OAUTH 签名。

4.5.1 文件上传

dev域名: <https://f-capup.test.webank.com>

测试环境F1域名: <https://f1-capup.test.webank.com>

测试环境G1域名: <https://g1-capup.test.webank.com>

URL路径: /api/s/up/upload

方法: POST

body类型: multipart/form-data, post 内容为文件的二进制流, 表单格式提交, 文件内容不参与 Oauth sign 签名。

http请求的url参数列表:

name	comment	value
app_id	分配的app_id	
nonce	随机数	32位字符串 (字母+数字)
sign	签名	具体参考签名算法
version	版本号	1.0.0
type	同步数据类型	sign
fileType	文件类型	具体参考文件类型字典
fileFormat	文件格式	jpg/png/zip等

请求示例:

```
https://f-capup.test.webank.com/api/s/up/upload?app_id={{app_id}}&nonce={{nonce}}&sign={{sign}}&version=1.0.0&type=sign&fileType=001&fileFormat=png
```

回包示例:

```
{
  "msg": "系统处理成功",
  "BizSeqNo": "1910280FH06207170022153392801372",
  "code": "60840000",
  "jsonData": "{\"CODE\":\"60840000\",\"FILE_LIST\":\n[{\n\"HASH_VALUE\":\"028f15759359930c726a85a15c64eb72\", \"FILE_ID\":\"970000032622\n1668220060046088380101000001572253220063599\", \"FILE_NAME\":\"HX_PP_ERP_2019-\n10-28_112034.tar.gz\"}], \"DESC\":\"系统处理成功\"}\n"}
}
```

4.5.2 文件下载

dev域名: <https://f-capup.test.webank.com>

测试环境F1域名: <https://f1-capup.test.webank.com>

测试环境G1域名: <https://g1-capup.test.webank.com>

URL路径: api/s/up/download

方法: GET

http请求的url参数列表:

name	comment	value
app_id	分配的app_id	
nonce	随机数	32位字符串（字母+数字）
sign	签名	具体参考签名算法
version	版本号	1.0.0
type	同步数据类型	sign
fileId	文件ID	上传时返回的FILE_ID
hashValue	Hash值	上传时返回HASH_VALUE
contentType	HttpContentType	不传默认image/jpeg, 枚举 https://tool.oschina.net/commons/

返回参数说明：

如果下载成功，返回 Content-Type 类型为对应传参 ContentType 或 image/jpeg(默认);

如果下载失败，返回 Content-Type 类型为 application/json。

请求示例：

```
https://f-capup.test.webbank.com/api/s/up/download?app_id={{app_id}}&nonce={{nonce}}&sign={{sign}}&version=1.0.0&type=sign&fileId=70000003257189256668301660401326000000015204237666836800&hashValue=1e574f8e9a5e335d3ac5079ce5719e39&contentType=image/jpeg
```

4.5.3 文件上传带通知接口

URL路径：其中{bizPath}根据需求分配

1. /up/s/{bizPath}/uploadJson （application/json格式提交，只支持单文件，base64文件）
2. /up/s/{bizPath}/uploadForm （form-data格式提交，支持多文件，二进制流文件）

方法：POST

Content-Type：multipart/form-data、application/json

http请求的url参数列表：

name	comment	value
app_id	分配的app_id	
nonce	随机数	32位字符串（字母+数字）
sign	签名	具体参考签名算法
version	版本号	2.0.0
type	同步数据类型	sign
fileType	文件类型	具体参考文件类型字典
fileFormat	文件格式	jgp/png/zip等

form-data格式提交 字段说明：

Key	Value	DESCRIPTION	出现要求	备注
file	二进制流文件内容	文件名	R	上传二进制流文件，可以多文件
bizContent	通知报文内容		O	由具体业务系统定义内容

application/json格式提交 字段说明：

字段名	字段释义	类型	出现要求	备注
file	文件内容	String	R	Base64编码字符串
fileName	文件名	String	R	
bizContent	通知报文	String	O	JSON格式字符串，由具体业务系统定义内容

响应参数说明：

序号	参数名	参数说明	类型	备注
1	code	错误码	String	60840000成功
2	msg	错误描述	String	
3	fileList	文件列表	Array	
3.1	fileName	文件名称	String	
3.2	fileId	文件ID	String	
3.3	hashValue	Hash值	String	

4.6 接入其他说明

1. Access Token和API ticket全局唯一，不同系统重复获取会导致互踢，影响业务请求。
2. API ticket失效的问题，更新app secret,access token等票据，会导致它派发的票据链失效。实例：app secret->access token->any type ticket。
3. 避免频繁获取，建议设计全局唯一定时任务（保证只有一个实例在刷新防止互踢），按照每 20 分钟绑定定时刷Access Token和API ticket。

五、微众调用合作方

5.1 基本要求

默认采取以下传输标准：

- a. 合作方和微众银行之间要求双方后台使用单向HTTPS（公网接入要求HTTPS）或HTTP连接(专线接入)。
- b. 由合作方提供反向通知地址，多个接口统一为一个URL地址，以transName区分不同交易。
- c. 采用POST方式，传输内容为全部字段以Json 格式编码为一个字符串，整个字符串作为 POST Body。
- d. 双方报文均由分配的盐值签名和验证签名。

5.2 HTTP报文字段定义

1. transName字段标示接口ID，由微众业务系统与合作方约定。
2. Plain字段表示业务报文（微众的业务请求内容或合作方的响应内容）
3. Timestamp字段标示签名时取的时间戳。
4. SeqNo字段标示请求的唯一流水编号。
5. Signature字段标示签名串，签名串是对Plain字段的签名。

5.3 HTTP报文签名算法

Plain字段产生步骤：

1. 将业务报文（可以包含集合或数组）转成标准JSON字符串；
2. 将JSON字符串用URL Encode编码。

Signature = SHA256(Plain + 分配盐值 + 时间戳):

```
String signStr = Plain + salt + Timestamp;  
String Signature = Hashing.sha256().hashString(signStr,  
Charsets.UTF_8).toString();
```

5.4 合作方响应回包要求

请求报文必须通过POST以JSON流（Application/JSON，非表单）发送至合作方，合作方对Plain和Signature验证签名，保证消息无篡改。返回报文也必须通过JSON流响应，同时用【5.3 HTTP报文签名算法】对返回报文签名。合作方返回消息后，微众同时对Plain和Signature验证签名，保证消息无篡改。

签名验证算法示例：

```
String transName = (String) reqMap.get("transName");  
String plainContent = (String) reqMap.get("Plain");  
String signature = (String) reqMap.get("Signature");  
String timestamp = (String) reqMap.get("Timestamp");  
String tmpSignStr = plainContent + certSalt + timestamp;  
String tmpSign = Hashing.sha256().hashString(tmpSignStr,  
Charsets.UTF_8).toString();  
if (plainContent == null || signature == null) {  
    logger.info("缺少Plain或Signature, transName:{}, Plain: {}, Signature: {}",  
transName, plainContent, signature);  
} else if (!tmpSign.equals(signature)) {  
    //验证签名 signature  
    logger.error("签名验证失败, plainContent: {},signature: {}", plainContent,  
signature);  
    logger.info("Correct Signature: " + tmpSign);  
} else {  
    //解析业务报文内容  
    String plainStr = URLDecoder.decode(plainContent, "utf-8");  
    return plainStr;  
}
```

5.5 反向调用请求示例

1. TLMerchantQuery接口要传输的报文体为

```
{"merchantno":"00000006","product_cd":"230001","req_nbr":"tbt001"}
```

2. 消息体经过URLEncode后得到

```
%7B%22merchantno%22%3A%2200000006%22%2C%22product_cd%22%3A%22230001%22%2C%22req_  
nbr%22%3A%22tbt001%22%7D
```

3. 用盐值签名后得到Signature

```
1cb54c406c06033967b4790bc31c6e60
```

4. 响应报文定义

注意：如果合作方和微众业务系统有约定其它响应码，可以自行约定。如无特殊需求，要求合作方的相应报文定义如下：

返回码	返回信息
0000	系统处理成功
0001	报文格式错误
0002	未满足接口输入要求
0103	报文解析失败
0010	签名验证失败
0012	时间戳过期
0016	SSL认证失败

1. 合作方响应结果：{"CODE": "0000", "DESC": "系统处理成功"}

2. 消息体经过URLEncode后得到Plain字段值：

```
%7B%22CODE%22%3A%2260840000%22%2C%22DESC%22%3A%22E7%B3%BB%E7%BB%9F%E5%A4%84%E7%90%86%E6%88%90%E5%8A%9F%22%7D
```

5. Http body示例

微众调用合作方请求的http body:

```
{"Plain": "%7B%22merchantno%22%3A%22000000006%22%2C%22product_cd%22%3A%2230001%22%2C%22req_nbr%22%3A%22tbt001%22%7D", "Signature": "1cb54c406c06033967b4790bc31c6e60", "Timestamp": "1519700632680", "transName": "TLMerchantQuery", "SeqNo": "1519700632680112122"}
```

合作方响应的的http body:

```
{"Plain": "%7B%22CODE%22%3A%2260840000%22%2C%22DESC%22%3A%22E7%B3%BB%E7%BB%9F%E5%A4%84%E7%90%86%E6%88%90%E5%8A%9F%22%7D", "Signature": "e89021df529e09eb8cbd0f99e64c5bc9", "Timestamp": "1519700632680", "transName": "TLMerchantQuery", "SeqNo": "1519700632680112122"}
```

六、常见问题解决办法

6.1 token和ticket如何判断失效

token和ticket的有效期均为3600秒；
出现60841017说明ticket不存在，过期或者未获取成功；
出现60841018，在参数正确的情况下，有可能是系统多次获取ticket互踢导致；
如果token和ticket是定时获取的，如果出现异常导致定时获取失败，判断出现60841017(token不存在)和60841018(签名不正确)错误，需要重新获取，不要只依赖定时任务，否则会影响一段时间服务；
获取新的ticket后，上一个ticket五分钟后失效。

6.2 为什么出现“未满足接口要求”

先检查参数是否完整；
注意app_id、version等参数是放在url里面，jsonData代表业务接口字段，不要加jsonData:{}这样的格式，业务接口字段作为整体放在Post Body里面，格式为json字符串，json字符串，UTF-8编码；

6.3 为什么出现“签名不正确”

检查签名前拼接的字符串是否和微众拼接签名字符串一致，一般原因为参与ticket不是最新的

6.4 为什么出现“交易超时”

交易超时，一般是开放平台调用业务系统超时，询问业务对接开发人员系统服务是否正常

6.5 为什么出现“无效appid”

出现CODE为6084xxxx的信息，说明验证网络证书无问题，请求已经到达前置应用，但因为没分配appid等接入测试信息，导致权限校验失败

6.6 微众调用合作方的transName是什么？

transName是微众业务系统和合作方自行约定的接口名称，使用同一个url，用transName区分不同请求接口。

6.7 合作方请求微众银行url里的{txn_id}是什么？

{txn_id}是指微众业务系统具体的接口ID，需要对应的业务系统开发提供。

附录

附录1、Ticket类型列表

sign：用于签名请求数据

user：用于获取用户维度ticket，用于登录态同步

附录2、平台层响应码列表

该响应码列表仅包含开放平台层错误码，业务层错误码在接口协议中具体定义，从jsonData字段里面获取。

2.1 请求凭据阶段（获取access_token和获取ticket）：

```
(60850000, "系统处理成功"),
(60856999, "未知异常信息"),
(60846999, "未知异常信息"),
(60851001, "Type类型不存在"),
(60851007, "无效的请求调用"),
(60851009, "无效的appid"),
(60851002, "无效的access_token"),
(60851003, "无效的ticket"),
(60851004, "access_token已过期"),
(60851005, "ticket已过期"),
(60851010, "ticket获取失败"),
(60851013, "access_token获取失败"),
```

2.2 正常业务请求阶段：

```
(60840000, "系统处理成功"),
(60846999, "未知异常信息"),
(60840002, "未满足接口输入要求"),
(60840003, "交易超时"),
(60840110, "请求路径不合法"),
(60841017, "ticket验证失败"),
(60841018, "接入签名不正确"),
```

附录3、业务接入请求接口

业务请求统一为Https请求，请求接口形式如下：

序号	栏位项目名称	中文说明	长度	出现要求	内容说明
输入					
1	app_id	webankappid		R	由webank指定
2	version	接口版本号		R	2.0.0
3	nonce	随机数		R	32位随机串(字母+数字组成的随机数)
4	sign	签名		R	对url参数和body的签名
5	type	同步数据类型		O	根据不同的接口填不同的值,具体查看每个接口
http request body 以下是 json 格式对象,对象的属性如下					
后端字段全部以 Json 格式编码为一个字符串,整个字符串作为 POST Body 并参与签名					
输出					
1	code	响应码	C(9)	R	处理的响应码
2	msg	响应信息	C(120)	R	响应码的中文含义
3	jsonData	json串		O	字符串,内容是json格式,json对象的属性
后端返回字段全部以 Json 格式编码为一个字符串作为 JsonData 这个 Json Key 的 Value					

附录4、环境域名信息

接入方根据测试要求选择不同测试环境,微众提供F、F1、G1环境供测试接入。生产的域名端口,由微众sa邮件发出。

后台联机:

F域名: f-capapi-v2.test.webank.com
F1域名: f1-capapi-v2.test.webank.com
G1域名: g1-capapi-v2.test.webank.com

文件上传下载:

F域名: f-capup.test.webank.com
F1域名: f1-capup.test.webank.com
G1域名: g1-capup.test.webank.com

如需切换环境,测试地址更换域名即可。

附录5、HTTP请求过程示例

第一步: 获取Token GET请求

```
https://f1-capapi-v2.test.webank.com:443/api/oauth2/access_token?
app_id=wbvyujilf9xwp&secret=MdedexuPN8METi5utrZqXDPjyJ9s0XxugzqeaTSRlKPW2bcGTUV
OZ0wyapgxHE&grant_type=client_credential&version=2.0.0
```

第二步: 获取Ticket GET请求

```
https:// f1-capapi-v2.test.webbank.com:443/api/oauth2/api_ticket?
app_id=wb9ec54543ca246&access_token=3b6289eac94a40ca0d80ebd0c011891221aef7330b3c
e99c56d5811455745066&type=sign&version=2.0.0
```

第三步：计算sign值进行业务请求 POST请求

```
https:// f1-capapi-v2.test.webbank.com:443/api/s/cgy/xxx?
app_id=wb9ec54543ca246&nonce=f6YZjdk7Ks1cxVoSFs9fs2eTaba06pHr&version=2.0.0&sign
=69CD8760DCF2002B309B7FB156C8F03CB2B6A9A7
```

注意：最新的接口版本号version为2.0.0。

附录6、在线测试工具

1. [开放平台HTTP测试工具](#)

附录7、文件类型数据字典

fileType	文件类型
001	身份证
002	营业执照
003	开户许可证
004	企业授权函
005	发票
006	合同
007	资金证明文件
008	压缩文件
999	未知类型

附录8、开发阶段网络测试专用账号

```
app_id = wbvyujjilf9ws6uxwp
secret = 9FctqAlHUt0t4mJxq1LYeMNIhBAyXC6tPtrYqdtbHIymg2g4UX908kKKnLErg64r
url = https://f-capapi-v2.test.webbank.com/api/s/test/10011
```