

移动协同签名系统

接口说明手册

(V 2.0.8)

目 录

欢迎使用.....	4
版权声明.....	4
1 概述.....	5
1.1 文档用途.....	5
1.2 名词解释.....	5
2 集成说明.....	6
2.1 组件介绍.....	6
2.2 集成环境.....	6
2.3 集成流程.....	6
2.4 集成方式.....	6
2.5 接口规范.....	6
2.5.1 公共请求参数.....	7
2.5.2 公共返回参数.....	8
2.5.3 报文签名值 signature 计算方法.....	8
2.5.4 请求响应报文示例.....	10
2.6 注意事项.....	11
3 接口说明.....	11
3.1 初始化.....	11
3.2 通用接口.....	12
3.3 必要接口.....	12
3.3.1 添加用户.....	12
3.3.2 产生激活码.....	14
3.3.3 添加签名任务.....	16
3.4 可选接口.....	20
3.4.1 用户管理场景接口.....	20
3.4.1.1 产生激活码（使用姓名+身份证号核实身份）.....	20
3.4.1.2 产生激活 H5 页面.....	22
3.4.1.3 获取激活码并发送短信.....	23
3.4.1.4 查询用户信息.....	24



3.4.1.5 修改用户信息.....	28
3.4.1.6 查询设备信息.....	29
3.4.1.7 用户设备注销.....	31
3.4.2 基本签名场景接口.....	32
3.4.2.1 获取签名结果.....	32
3.4.2.2 获取签名任务明细.....	32
3.4.2.3 撤销签名任务.....	38
3.4.2.4 查询印章图片.....	39
3.4.3 自动签名场景接口.....	41
3.4.3.1 开启自动签.....	41
3.4.3.2 关闭自动签.....	44
3.4.3.3 自动签名.....	45
3.4.3.4 查询自动签时间.....	48
3.4.4 验证签名场景接口.....	48
3.4.4.1 验证数据签名.....	48
3.4.4.2 验证网页签名.....	50
3.4.5 授权场景接口.....	53
3.4.5.1 授权关系明细查询.....	53
3.4.6 证书管理场景接口.....	57
3.4.6.1 解析证书信息.....	57
3.4.6.2 查询证书.....	59
3.4.6.3 证书验证.....	61
3.4.7 集成时间戳接口调用.....	63
3.4.7.1 产生时间戳.....	63
3.4.7.2 解析时间戳.....	65
3.4.7.3 验证时间戳.....	67
3.4.8 PDF 签章相关接口.....	69
3.4.8.1 计算 PDF 文件 hash 值.....	69
3.4.8.2 合并 PDF 签章.....	75



3.4.8.3 验证 PDF 签章文件.....	77
3.4.9 辅助接口.....	80
3.4.9.1 计算指定长度的随机数.....	80
3.4.9.2 根据算法类型计算原文数据的 hash 值.....	81
3.4.9.3 验签接口（原文验签，hash 验签，web_seal 验签）.....	83
4 错误码说明.....	85
5 常见 FAQ.....	91

欢迎使用

欢迎您使用移动协同签名系统，如果本手册为您提供帮助，带来便利，我们将深感欣慰。

如果您在使用过程中，遇到了问题，或对我们产品有好的建议，可以：

☎ 致电客户服务热线：4009197888

💻 或访问公司网站：www.bjca.cn

与我们联系，对您提出的问题或建议，我们表示衷心的感谢。

版权声明

本手册著作权属北京数字认证股份有限公司所有，在未经本公司许可的情况下，任何单位或个人不得以任何方式对本手册的部分或全部内容擅自进行增删、改编、节录、翻印、改写，仅限和北京数字认证公司的项目合作方公司使用。

北京数字认证股份有限公司

©2022

1 概述

1.1 文档用途

本文的预期读者为公司研发人员、公司测试人员、公司实施人员等。

本说明书为开发人员、测试人员、客户之间提供共同的协议而建立基础，指导软件功能的实现。

本说明书做为项目结项验收文档的一部分，提供技术委员会评审通过后，做为正式文档进行归档。

1.2 名词解释

产品名词简称	解释
COSS	移动协同签名系统简称
MSSPID	用户 ID，由 COSS 为用户生成的用户身份标识，MSSPID 与姓名身份证号码一一对应
AppID	应用 ID，用于标识业务系统身份
secureCode	安全码，作为计算 HMAC 的密钥，用于计算请求报文数据签名
HMAC	是密钥相关的哈希运算消息认证码，HMAC 运算利用哈希算法，以一个密钥和一个消息为输入，生成一个消息摘要作为输出。
HMAC-SHA256	是基于 SHA256 哈希算法的 HMAC 计算方法

2 集成说明

2.1 组件介绍

基于 java 语言 springboot 框架开发，方便于业务端的调用，包括了用户管理，基本签名，自动签名，验证签名，证书管理等功能。

2.2 集成环境

项目说明	内容说明
操作支持的系统	Windows, Linux 操作系统
支持的开发语言	支持 Java 语言, JDK1.8
支持的协议	HTTP
支持的浏览器	不限制浏览器版本

2.3 集成流程

集成流程参考《移动协同签名系统集成指南手册 V2.0.8》。



移动协同签名系
系统集成指南手册

2.4 集成方式

标准 restfulAPI 接口集成方式。

2.5 接口规范

- 接口功能：

该组接口是协同签名服务器以 resfulAPI 的形式对业务系统提供用户管理、证书管理、协同签名等服务。

- 接口形式：HTTP 服务
- 请求方式：POST/GET
- 数据编码：JSON, UTF8
- POST 请求头格式：Content-Type: application/json
- 请求地址：<http://ip:port/coss/> + 接口地址
- 请求和响应参数组成结构说明：

接口请求和响应的均为 json 格式数据报文，数据报文由公共参数和业务参数两部分构成，请求报文需要由业务系统使用 HMAC 签名算法进行计算，目前支持的签名算法为 HMAC-SHA256。

```
{  
    公共参数,  
    业务参数  
}
```

2.5.1 公共请求参数

字段	类型	是否必须	说明
transId	String	否	业务流水号：业务系统生成
appId	String	是	应用 ID 是由协同签名系统生成的应用 id, 通过协同签名管理页面的应用管理中获取
signAlgo	String	是	签名算法（默认传入 HMAC），使用的算法为 HMAC-SHA256
signature	String	是	请求报文签名结果
version	String	是	请求消息版本号（1.0）



2.5.2 公共返回参数

字段	类型	是否必须	说明
status	String	是	结果码
message	String	是	结果信息
transId	String	否	业务流水号, 该业务流水号为请求报文中传入的业务流水号
data	String(响应业务 数据)	否	响应内容, { "data" : "data" }

2.5.3 报文签名值 signature 计算方法

第一步, 设所有发送或者接收到的数据为集合 M, 将集合 M 内非空参数值的参数按照参数名 ASCII 码从小到大排序 (字典序), 使用 URL 键值对的格式 (即 key1=value1&key2=value2...) 拼接成字符串 stringA。

第二步, 对 stringA 对进行 HMAC 运算, 得到 hmac 的值后再做 Base64 编码, 最后得到签名值 signature。

String signTemp = hmac(stringA, secret, hmacAlgo); //密钥 secret 是由协同签名系统生成的安全码 secureCode, 通过协同签名管理页面的应用管理中获取, 安全码要与 appId 对应

String signature = Base64(signTemp);

hmac 方法示例:

```
SecretKeySpec signKey = new SecretKeySpec(secret.getBytes(), hmacAlgo);
Mac mac = null;
byte[] signBytes = null;
try {
    mac = Mac.getInstance(hmacAlgo);
```



```
mac.init(signKey);  
signBytes = mac.doFinal(stringA.getBytes());  
} catch (Exception e) {  
    log.error("计算 HMAC 值失败: ", e);  
}
```

2.5.4 请求响应报文示例

- 接口请求示例:
- 以调用添加用户接口为例（公共参数标记红色，业务参数标记蓝色），请求报文如下:

```
{  
    "version": "1.0",  
    "appId": "APP_0157E810F6894BEEB4DEC70331720557",  
    "signAlgo": "HMAC",  
    "userName": "张三",  
    "idType": "SF",  
    "idNumber": "513436200005039279",  
    "mobile": "18745292611"  
}
```

- 计算报文签名:
- 对示例请求的参数按照 key=value 的格式,并按照参数名 ASCII 字典序排序后字符串如下:

```
stringA= "appId=APP_0157E810F6894BEEB4DEC70331720557&idNumber=513436200005039  
279&idType=SF&mobile=18745292611&signAlgo=HMAC&userName=张三&version=1.0" ;
```

- 计算签名值 signature:

```
stringSignTemp=hmac(stringA, secretkey) //计算报文的签名值
```

```
signature=Base64(stringSignTemp) //把签名结果转换成 Base64 编码
```

- 对上述待签名原文进行 HMAC-SHA256 签名的签名值结果如下:

```
NA32oYb/y50i0Wn0gy+Auo304Ta4MHXEQeK0JW3Vg+Q=
```

- 组成待发送报文如下:

```
{  
    "version": "1.0",  
    "appId": "APP_0157E810F6894BEEB4DEC70331720557",  
    "signAlgo": "HMAC",  
    "userName": "张三",
```



```
"idType": "SF",  
  
"idNumber": "513436200005039279",  
  
"mobile": "18745292611",  
  
"signature": "NA32oYb/y50iOWnOgy+Auo304Ta4MHXEQeK0JW3Vg+Q="
```

```
}
```

● 响应报文示例:

```
{  
  
  "status": 200,  
  
  "message": "SUCCESS",  
  
  "data": {  
  
    "userId":  
  
    "548a43ae04f109a2c8c812c95673579a03afd8512d7fc85f9abdc9de7066b2af"  
  
  }  
}
```

2.6 注意事项

- ◆ 参数名 ASCII 码从小到大排序（字典序）；
- ◆ 如果参数的值为空不参与签名，且请求报文中也不要带此参数；
- ◆ 参数名区分大小写。
- ◆ 通过 <https://demo-system.isignet.cn/signHelper/sign.html> 校验签名值是否正确。

3 接口说明

3.1 初始化

无

3.2 通用接口

无

3.3 必要接口

注册场景下，用户打开 PC 端页面点击注册；业务端调用[添加用户接口](#)，注册用户信息到移动协同签名系统；业务端调用[产生激活码接口](#)返回激活码、二维码信息、二维码 H5 页面，返回给 PC 端生成二维码展示；用户打开手机 APP 扫码，设置证书 pin 口令，完成证书下载。

登录场景下，用户在 PC 端点击登录；业务端产生随机数，调用[添加签名任务接口](#)向移动协同签名系统发起登录认证请求，并将返回的待签名任务二维码信息，返回给 PC 端生成二维码展示；用户打开 APP 扫码完成签名；业务端通过轮询调用[获取签名结果接口](#)或回调函数获取签名结果；业务端获取签名结果后，关闭二维码，根据签名信息进入登录页。

签名场景下，用户打开浏览器点击签名；业务端通过调用[添加签名任务接口](#)，将待签名业务数据发给移动协同签名系统，返回签名任务 ID 和待签名任务二维码信息，并在 PC 浏览器端生成二维码展示；用户打开手机 app，扫码签署；签署完成后，业务端通过轮询调用[获取签名结果接口](#)或回调函数被调，获得签名结果和证书，签署完成。

3.3.1 添加用户

- **功能说明：**实现用户的添加
- **请求地址：**http://host:port/coss/service/v1/addUser
- **请求方式：**POST
- **请求参数：**

字段	类型	是否必须	说明
userName	String	是	用户姓名
idNumber	String	是	证件号码



idType	String	是	证件类型： SF-身份证 GA-港澳通行证 HZ-护照 QT-其它
mobile	String	是	手机号
department	String	是	部门信息，最长 64 位
uniqueId	String	否	用户业务系统中的唯一 Id，最长 64 位，该值为业务系统唯一标识该用户身份的数据，可以是工号、账户名等
signImg	String	否	签章图片-base64 格式数据
requireCert	String	否	是否能下载证书，用户是否证书，有为 Y 无为 N，默认有证书

● 请求示例：

```
{  
  
  "version": "1.0",  
  
  "appId": "APP_0157E810F6894BEEB4DEC70331720557",  
  
  "signAlgo": "HMAC",  
  
  "userName": "张三",  
  
  "idType": "SF",  
  
  "idNumber": "513436200005039279",  
  
  "department": "QT",  
  
  "mobile": "18745292611",  
}
```



```
"signature": "NA32oYb/y50iOWnOgy+Au0304Ta4MHXEQeK0JW3Vg+Q="
```

```
}
```

● 返回参数:

字段	类型	是否必须	说明
userId	String	否	用户 ID (即 COSS 为用户生成的用户身份标识, 也叫 MSSPID, MSSPID 与姓名身份证号码一一对应)

● 响应示例:

```
{
```

```
  "status": 200,
```

```
  "message": "SUCCESS",
```

```
  "data": {
```

```
    "userId": "548a43ae04f109a2c8c812c95673579a03afd8512d7fc85f9abdc9de7066b2a"
```

```
  }
```

```
}
```

3.3.2 产生激活码

- 功能说明: 业务系统服务端为在协同签名服务器中已经添加的用户申请激活码 (该授权码业务系统可以通过短信或邮件方式发送给最终用户, 最终用户在移动端调用相应输入授权码激活接口完成用户激活)

- 请求地址: <http://host:port/coss/service/v1/getAuthCode>

- 请求方式: POST

● 请求参数:

字段	类型	是否必须	说明
----	----	------	----



userId	String	是	用户 ID（即 COSS 为用户生成的 MSSPID，该值与姓名身份证号码一一对应）
--------	--------	---	--

● 请求示例：

```
{  
  
  "version": "1.0",  
  
  "appId": "APP_0157E810F6894BEEB4DEC70331720557",  
  
  "signAlgo": "HMAC",  
  
  "userId": "548a43ae04f109a2c8c812c95673579a03afd8512d7fc85f9abdc9de7066b2af",  
  
  "signature": "2muiLLUb1hY6WfF6uF9NhZQbQF0bzVPPNql08h/SKdA="
```

● 返回参数：

字段		类型	是否必须	说明
status		int	是	状态码（200 成功）
message		String	是	状态信息
transId		String	否	业务流水号
data			否	返回业务数据
	authCode	String	否	激活码二维码，用于在手机端扫码发起用户身份激活操作
	code	String	否	用户激活码，用于在手机端输入激活码发起用户身份激活操作

● 响应示例：



```
{  
  
  "status": 200,  
  
  "message": "SUCCESS",  
  
  "data": {  
  
    "authCode": "{\\"id\\":\\"APP_E39718EFFD514DB293F934E649B747A9\\",\\"sUr1\\":\\"ht  
tps://coss-dev.isignet.cn:18759/coss/mobile/v1/getServiceInfo\\",\\"o\\":\\"KHEODg  
QEUHoQAgJcfTY2YA5SSVAOS0BUGndxEhM70TM3FWF1YxphcSU1HQMMHS4QChMJY2NxLQ==\\",\\"cv\  
":\\"2.0\\"}",  
  
    "code": "23d0ee0a"  
  
  }  
  
}
```

3.3.3 添加签名任务

- **功能说明：**该接口用户业务系统服务端调用，用于添加一个签名任务，支持对数据原文、hash 数据、网页三种数据格式，该接口的返回值供移动终端完成协同签名使用。（**推荐使用回调方式获取签名结果**）
- **请求地址：**http://host:port/coss/service/v1/addSignJob
- **请求方式：**POST
- **请求参数：**

字段	类型	是否必须	说明
userId	string	否	用户 id
title	string	是	标题（最长不超过 100 个字符）
dataType	string	是	原文类型（DATA: 原文, HASH: hash 数据, WEB_SEAL: 网页签章）



algo	string	是	签名算法(SM3withSM2)
description	string	否	描述（最长不超过 200 个字符）
expiryDate	string	否	签名任务有效期（截止时间，单位为分钟），不能大于4320分钟，即3天有效期（该字段为空则默认有效期为1天）
data	string	是	待签数据（必须是 base64 编码，最大不超过 1M）
callbackUrl	String	否	回调地址

● 请求示例：

```
{  
  
  "version": "1.0",  
  
  "appId": "APP_0157E810F6894BEEB4DEC70331720557",  
  
  "signAlgo": "HMAC",  
  
  "userId": "548a43ae04f109a2c8c812c95673579a03afd8512d7fc85f9abdc9de7066b2af",  
  
  "title": "测试签名",  
  
  "dataType": "DATA",  
  
  "algo": "SM3withSM2",  
  
  "description": "签名数据描述信息",  
  
  "expiryDate": "1440",  
  
  "data": "562+5ZCN5rWL6K+V5pWw5o2u",  
  
  "signature": "CGlj1xQgXuewoKUs1CK61KyL2EjwsazsIln8ae7KOMw="
```

● 响应参数：

字段	类型	是否必须	说明
----	----	------	----



signDataId	String	否	签名数据 id
qrCode	String	否	用于产生二维码的文本内容
cert	String	否	证书
signature	String	否	签名值

● 响应示例:

```
{
  "status": 200,
  "message": "SUCCESS",
  "data": {
    "signDataId": "SD_e7841451-8ba2-48ce-b4ef-5a7ddcf0fb86",
    "qrCode":
    "{\\"id\\":\\"APP_E39718EFFF514DB293F934E649B747A9\\",\\"sUrl\\":\\"https://coss-
    dev.isignet.cn:18759/coss/mobile/v1/getServiceInfo\\",\\"o\\":\\"KHE0DgQEUHoQY
    3VnKGRrZF5EUEntClJQCmBnazMKXQdGJVQdBF16NzczCUADEHgEEh0aOSojNU1KRyEJdX4TFG8
    lNiIcGQocYggSABZ9cS4=\\",\\"cv\\":\\"2.0\\"}"
  }
}
```

注: 对于回调地址说明, 集成方需提供回调接口。结构如下:

入参说明:

字段	类型	是否必须	说明
signJobId	String	是	签名任务 id
status	String	是	签名任务状态
msspId	String	是	用户唯一标识
signResult	String	否	签名结果
signCert	String	否	签名证书

返回参数:

返回参数:

字段	类型	是否必须	说明
----	----	------	----



status		Int	是	状态码
message		String	是	提示信息
data		String	否	响应数据
	signJobId	String	否	签名任务 id
	status	String	否	签名任务状态

请求报文示例:

```
{
  "signJobId": "SD_000697d0-4e1b-4cb7-b9cf-619e347892f8",
  "status": "FINISH",
  "signResult":
    "MEYCIQDvcy6/WHJ+8QS1S1Ja1Q9FBa6gtnboxwndyQpHBBbcaQIhAMeZjymkozFKm1MaD2xq/L4JhW3gO5XTm
    GL6S00MzZwr",
  "signCert":
    "MIIDAzCCAqagAwIBAgISIAIX9hLyA5saPDc8EKE2osMvMAwGCCqBHM9VAYN1BQAwPjELMAkGA1UEBgcwCQ
    04xDTALBgNVBAoMBEJJKQ0ExDTALBgNVBASMBEJJKQ0ExETAPBgNVBAMMCExPQ0FMU00yMB4XDTE5MDYwNDE
    xMDMzN1oXDTIOMDYwNDExMDMzN1owcDELMAkGA1UEBgcwCQ04xDTALBgNVBAMMBuW8oOS4iTFQME4GCgmSj
    omT8ixkAQEMQDU00GEOM2F1MDRmMTA5YTJjOGM4MTJjOTU2NzM1NzlhMDNhZmQ4NTEyZDdmYzg1Zj1hYmR
    jOWR1NzA2NmIyYWYwWTATBgqcqhkJOPQIBBgqgRzPVQGCLQNCAAQ1TH7W1yLLMd4honFRQJPZDAegXM7K1
    /OZ8jY2wuzwTXs55jUQZAu4C9Vj6tLp6Qem+01v7v/mAUyM7ePj/CT5o4IBTjCCAUowCwYDVR0PBAQDAgb
    AMAkGA1UdEwQCMAAwEwYDVR01BAwwCgYIKwYBBQUHAwMwgYUGA1UdHwR+MHwwOKA2oDSGMmh0dHBzOi8vY
    3JsLmlzaWduZXQuY24vY3JsL0xPQ0FMU00yL0xPQ0FMU00yXzAuY3JsMECgPqA8hjpodHRwczovL2Nybc5
    pc2lnbmV0LmNuL2Nybc9MT0NBTFNnMi9pbmMvTE9DQUxTTTJfaW5jXzAuY3JsMB0GA1UdDgQWBBR1zT2DR
    BtGU81NNfmDSMiICVjKvzAfBgNVHSMEGDAWgBQ78fsiTPugyfgIrS+tCqf3oTpz1zBTBgNVHSAETDBKMEg
    GCiqBHIbvMgYEAQEwOjA4BggrBgEFBQcCARYsaHR0cHM6Ly9jcmwuaXNpZ251dC5jb19jcHMvTE9DQUxTT
    TIvY3BzLmh0bWwwDAYIKoEcz1UBg3UFAANJADBGAiEAz3ueB71NEVYTsQ+wrckoQ+u9hi1Ls/C19TeaSrK
    +GcQCIQC/Euzxhze900FEZPQDd4PJ0tUEceZCcZvu4j9CUZ/w0w=="
}
```

响应成功报文示例:

```
{
```



```
"status": 200,
"message": "SUCCESS",
"data": {
    "signJobId": "SD_000697d0-4e1b-4cb7-b9cf-619e347892f8",
    "status": "FINISH"
}
```

3.4 可选接口

3.4.1 用户管理场景接口

实现查询用户详细信息（姓名、证件、ID 等），修改用户信息（可修改部门、邮箱、工号等），查询用户设备和用户设备注销等功能。

3.4.1.1 产生激活码（使用姓名+身份证号核实身份）

- **功能说明：**业务系统服务端为在协同签名服务器中已经添加的用户申请激活码，协同签名服务器会校验 userId、姓名、身份证号的一致性，校验通过则返回激活码，否则报错。
- **请求地址：**http://host:port/coss/service/v1/getAuthCodeWithIdenVerify
- **请求方式：**POST
- **请求参数：**

字段	类型	是否必须	说明
userId	String	是	用户 Id
idNumber	String	是	证件号码，最长 30 位
idType	String	是	证件类型： SF-身份证 GA-港澳通行证 HZ-护照 QT-其它



userName	String	是	用户姓名，最长 50 位
----------	--------	---	--------------

● 请求示例：

```
{  
  
  "version": "1.0",  
  
  "appId": "APP_0157E810F6894BEEB4DEC70331720557",  
  
  "signAlgo": "HMAC",  
  
  "userId": "548a43ae04f109a2c8c812c95673579a03afd8512d7fc85f9abdc9de7066b2af",  
  
  "userName": "张三",  
  
  "idType": "SF",  
  
  "idNumber": "513436200005039279",  
  
  "signature": " nDCs6NeG02o6cTApYE68JPb80YF6vKusPhSisKW7ZpQ="  
}
```

● 返回参数：

字段	类型	是否必须	说明
authCode	String	否	激活码二维码，用于在手机端扫码发起用户身份激活操作
code	String	否	用户激活码，用于在手机端输入激活码发起用户身份激活操作

● 响应示例：

```
{  
  
  "status": 200,  
  
  "message": "SUCCESS",  
  
  "data": {
```



```
"authCode":  
  "{ \"id\": \"APP_E39718EFFD514DB293F934E649B747A9\", \"sUrl\": \"https://coss-  
dev.isignet.cn:18759/coss/mobile/v1/getServiceInfo\", \"o\": \"KHEODgQEUHoQA  
1cOemsyZg5SSVA0SOBUGndxEhM7OTM3FWF1YxphcSU1HQMMHS4QChMJY2NxLQ==\", \"cv\": \"  
2.0\" }",  
  
"code": "2f678a6a"  
  
}  
  
}
```

3.4.1.2 产生激活 H5 页面

- **功能说明：**业务系统服务端为在协同签名服务器中已经添加的用户申请激活码，协同签名服务器会校验 userId、signature 校验通过则返回包含激活二维码的 H5 页面，否则报错。
- **请求地址：**http://host:port/coss/service/v1/getAuthCodeForH5Mini
- **请求方式：**POST
- **请求参数：**

字段	类型	是否必须	说明
userId	String	否	用户 ID，即 MSSPID

- **请求示例：**

```
{  
  
  "version": "1.0",  
  
  "appId": "APP_7B3F36A14E99410A80B37AEF332E3247",  
  
  "signAlgo": "HMAC",  
  
  "userId": "057fbd512ab6455bdeb3a34c70344fa3e2a3d162a46fa203de867873dfd08e30",  
  
  "signature": "Yrh9zFW1uerRSzEVNCc/C/dluoejt+Th+w+18JNq7R4=",  
  
}
```

- **响应示例：**



```
<!DOCTYPE html>

<html lang="en">

  <head>

    <meta charset="UTF-8">

    <title>扫码确认</title>

  </head>

  <body>

    <div id="qrCodeArea">

      //base64 格式二维码图片

    </div>

  </body>

</html>
```

3.4.1.3 获取激活码并发送短信

- **功能说明：**业务系统服务端为在协同签名服务器中已经添加的用户申请激活码，协同签名服务器会校验 userId、signature 校验通过后发短信给用户，否则报错。
- **请求地址：**http://host:port/coss/service/v1/getAuthCodeAndSendSMS
- **请求方式：**POST
- **请求参数：**

字段	类型	是否必须	说明
userId	String	否	用户 ID，即 MSSPID

- **请求示例：**

```
{

  "version": "1.0",

  "appId": "APP_7B3F36A14E99410A80B37AEF332E3247",

  "signAlgo": "HMAC",

  "userId": "057fbd512ab6455bdeb3a34c70344fa3e2a3d162a46fa203de867873dfd08e30",
```



```
"signature": "Yrh9zFW1uerRSzEVNCC/C/dluoejt+Th+w+18JNq7R4="
```

```
}
```

● 响应示例:

```
{
```

```
  "status": 200,
```

```
  "message": "短信激活码发送成功",
```

```
  "data": {
```

```
    "authCode":
```

```
    "{\n  \"cv\": \"2.0\", \"id\": \"APP_B7D04A541B2B420A9E93218967400630\", \"o\": \"KHEODgQEUHoQV1IPKGB1YQxSSVA0S0BUGndxEhM70TM3FWF1YxphcSUIHQMMHS4QChMJY2NxLQ==\", \"sUrl\": \"http://192.168.136.9:10201/coss/mobile/v1/getServiceInfo\"}"
```

```
    "code": "23224484"
```

```
  }
```

```
}
```

● 短信示例:

【北京数字认证】尊敬的用户张三，您本次下载证书操作的激活码为：23224484，有效期5分钟。本激活码仅用于移动协同签名的证书签发，请妥善保管。

3.4.1.4 查询用户信息

- 功能说明：查询用户详细信息，业务系统调用本接口即可完成通过用户 uniqueId 或证件类型、证件号查询用户的 userId（即 MSSPID）、姓名等信息。

- 请求地址：http://host:port/coss/service/v1/queryUserInfo

- 请求方式：POST

- 请求参数：

字段	类型	是否必须	说明
userId	String	否	用户 ID，即 MSSPID



idNumber	String	否	证件号码
idType	String	否	证件类型（身份证号为 SF）
uniqueId	String	否	用户业务系统中的唯一 Id，最长 64 位，该值为业务系统唯一标识该用户身份的数据，可以是工号、账户名等

说明：

第一优先级：如传入 userId 则使用 userId 作为查询条件；

第二优先级：如 userId 为空，且传入了 idNumber、idType 则以证件号查询用户信息；

第三优先级：如以上两项同时不满足，则以 uniqueId 为条件查询用户信息；

如以上参数全部为空，返回错误信息。

● 请求示例：

```
{  
  
  "version": "1.0",  
  
  "appId": "APP_0157E810F6894BEEB4DEC70331720557",  
  
  "signAlgo": "HMAC",  
  
  "userId": "548a43ae04f109a2c8c812c95673579a03afd8512d7fc85f9abdc9de7066b2af",  
  
  "idType": "SF",  
  
  "idNumber": "513436200005039279",  
  
  "signature": "gagFkAkydxIVzDc166s1t08//Q/Yv0sQg4cJJpPLQiw="  
  
}
```

● 返回参数：

字段	类型	是否必须	说明
----	----	------	----



id	String	是	用户表业务主键
userName	String	是	用户姓名
department	String	是	部门
idType	String	是	证件类型
idCard	String	是	证件号
mobilePhone	String	是	手机
telephone	String	否	固定电话
email	String	否	邮件
jobNumber	String	否	工号
faceId	String	否	人脸 id
signpicId	String	是	签章 id
msspId	String	是	用户 id
status	String	是	状态
createTime	Date	是	创建时间
updateTime	Date	是	修改时间
requireCert	String	是	用户是否有权限下载证书（有 Y，无 N）
perserve1	String	否	冗余字段作为无证书用户的登录密码
perserve2	String	是	冗余字段作为默认签章标识（默认为 false ,非默认为 true ）

● 响应示例：



```
{  
  
  "status": 200,  
  
  "message": "SUCCESS",  
  
  "data": {  
  
    "id": "468010718332071936",  
  
    "username": "测试",  
  
    "department": "开发",  
  
    "idType": "身份证(SF)",  
  
    "idCard": "610527200312320988",  
  
    "mobilePhone": "17609098778",  
  
    "telephone": "",  
  
    "email": "",  
  
    "jobNumber": "",  
  
    "faceId": "",  
  
    "signpicId": 468010719288373248,  
  
    "msspId":  
    "80c60f904c0244803bd240f358a571f86a52abd1cc7304a446166e2a5aab0b4f",  
  
    "status": "NORMAL",  
  
    "createTime": "2021-07-15 11:08:30",  
  
    "updateTime": "2021-07-15 11:08:30",  
  
    "requireCert": "Y",  
  
    "perserve1": "",  
  
    "perserve2": "FALSE"  
  
  }  
}
```



}

3.4.1.5 修改用户信息

- 功能说明：用于修改用户信息
- 请求地址：http://host:port/coss/service/v1/updateUser
- 请求方式：POST
- 请求参数：

字段	类型	是否必须	说明
userId	String	是	用户 ID
status	String	否	用户新状态（激活：ACTIVE，冻结：LOCKED，注销：DELETE）
department	String	否	部门，最长 64 位
mobile	String	否	手机号码
image	String	否	用户印章图片（签名图片）base64
uniqueId	String	否	用户唯一 Id，最长 64 位

- 请求示例：

{

"version": "1.0",

"appId": "APP_0157E810F6894BEEB4DEC70331720557",

"signAlgo": "HMAC",

"userId": "548a43ae04f109a2c8c812c95673579a03afd8512d7fc85f9abdc9de7066b2af",

"status": "ACTIVE",

"department": "测试部",

"mobile": "13745293322",



"signature": "WBCwFrp2dya/aEveHJ3a0tEgXj8YGJyDiNuI1JeF8nE="

}

● 返回参数:

公共参数

● 响应示例:

```
{
  "status": 200,
  "message": "SUCCESS"
}
```

3.4.1.6 查询设备信息

- 功能说明: 查询用户下的设备信息
- 请求地址: http://host:port/coss/service/v1/getUserDevice
- 请求方式: POST
- 请求参数:

字段	类型	是否必须	说明
userId	String	是	用户 Id

● 请求示例:

```
{
  "version": "1.0",
  "appId": "APP_0157E810F6894BEEB4DEC70331720557",
  "signAlgo": "HMAC",
  "userId": "548a43ae04f109a2c8c812c95673579a03afd8512d7fc85f9abdc9de7066b2af",
  "signature": "YmLkpqE0ieLC7qe/4s1lqr+7qjwDm284MCNEkvFiCAo="
}
```



● 返回参数：

字段	类型	是否必须	说明
deviceId	String	是	设备 id
userId	String	是	用户 ID
deviceName	String	是	设备名称
reqCertTime	Date	是	证书下载时间

● 响应示例：

```
{
  "status": 200,
  "message": "SUCCESS",
  "data": {
    "deviceInfos": [
      {
        "deviceId": "DEVICE_93c6da4d-b832-4ed4-8fa5-8c79376d6abc",
        "userId":
          "f2f14013b88bde5d7f4afa12ef99c535e95a2c36dbd978b8dbebe5ae856815d0",
        "deviceName": "TEST GROUP AUTO TEST",
        "reqCertTime": "2022-06-09 10:19:03"
      },
      {
        "deviceId": "DEVICE_93c6da4d-b832-4ed4-8fa5-8c79376d6abc",
        "userId":
          "f2f14013b88bde5d7f4afa12ef99c535e95a2c36dbd978b8dbebe5ae856815d0",
        "deviceName": "TEST GROUP AUTO TEST",
        "reqCertTime": "2022-06-09 10:19:03"
      }
    ]
  }
}
```



```
    }  
  }  
}
```

3.4.1.7 用户设备注销

- 功能说明：删除用户设备
- 请求地址：http://host:port/coss/service/v1/deleteUserDevice
- 请求方式：POST
- 请求参数：

字段	类型	是否必须	说明
userId	String	是	用户 Id
deviceId	String	是	设备 Id

- 请求示例：

```
{  
  "version": "1.0",  
  "appId": "APP_0157E810F6894BEEB4DEC70331720557",  
  "signAlgo": "HMAC",  
  "userId": "548a43ae04f109a2c8c812c95673579a03afd8512d7fc85f9abdc9de7066b2af",  
  "deviceId": "DEVICE_fae2a200-68de-4c87-8aab-ada758ae5f75",  
  "signature": "vHq+FQ7WT1QEChQTJ15MKFm8vbCZbwThb3ke07oVoqs="br/>}
```

- 返回参数：
公共参数
- 响应示例：

```
{
```



"status": 200,

"message": "用户设备注销成功"

}

3.4.2 基本签名场景接口

3.4.2.1 获取签名结果

- **功能说明：**移动端完成签名任务后，该接口可以根据签名任务 id 获取签名结果，该接口配合 3.4.1 使用（可以使用轮询的方式进行接口调用，轮询频率建议1-2s/次，轮询频率太高会导致服务器高并发）

- **请求地址：**http://host:port/coss/service/v1/getSignResult

- **请求方式：**POST

- **请求参数：**

字段	类型	是否必须	说明
signDataId	String	是	签名数据 id

- **请求示例：**

{

"version": "1.0",

"appId": "APP_0157E810F6894BEEB4DEC70331720557",

"signAlgo": "HMAC",

"signDataId": "SD_cc3782fe-6596-4c7d-885f-21e29ff64dfb",

"signature": "pMCIJXgnjd6hl0t9UaElxIIDGln7yhDvwdWIWZoIXs="

}

- **响应参数：**

字段	类型	是否必填	说明
----	----	------	----



signDataId	String	否	签名数据 id
signResult	String	否	Base64格式签名结果
signCert	String	否	Base64格式证书
jobStatus	String	否	状态（UNSIGN：待签，FINISH：已签，EXPIRE：过期， REVOKE：签名任务被服务端撤销， REFUSE：签名任务被客户端拒绝）
msspId	String	否	用户编号 id

● 响应示例：

```
{  
  "status": 200,  
  "message": "SUCCESS",  
  "data": {  
    "signDataId": "SD_18eb25f8-3d15-4971-a60c-730c1480d837",  
    "signResult":  
      "MEYCIQDvcy6/WHJ+8QS1S1Ja1Q9FBa6gtnboxwndyQpHBBbcaQIhAMeZjymkozFKm1MaD2xq/  
      L4JhW3g05XTmGL6S00MzZwr",  
    "signCert":  
      "MIIDAzCCAqagAwIBAgISIAIX9hLyA5saPDc8EKE2osMvMAwGCCqBHM9VAYN1BQAwpjELMAkGA  
      1UEBgwCQ04xDALBgNVBAoMBEJKQ0ExDTALBgNVBASMBEJKQ0ExETAPBgNVBAMMCEXPQ0FMU00  
      yMB4XDTE5MDYwNDEyMDMzN1oXDTIOMDYwNDEyMDMzN1owcDELMAkGA1UEBgwCQ04xDzANBgNVB  
      AMMBuW8oOS4iTFQME4GCgmSJomT8ixkAQEMQDU0OGE0M2F1MDRmMTA5YTJjOGM4MTJjOTU2NzM  
      1NzlhMDNhZmQ4NTEyZDdmYzgzLzJlYmRjOWR1NzA2NmIyYWYwWTATBgqcqhkJOPQIBBggqgRzPV  
      QGCLQNCAAQ1TH7W1yLLMd4honFRQJPZDAegXM7K1/OZ8jY2wuzwTXs55jUQZAu4C9Vj6tLp6Qe  
      m+O1v7v/mAUyM7ePj/CT5o4IBTjCCAOUowCwYDVR0PBAQDAgbAMAKGA1UdEwQCMAAwEwYDVR01B  
      AwwCgYIKwYBBQUHAWMwgYUGA1UdHwR+MHwwOKA2oDSGMmhOdHBzOi8vY3JsLm1zaWduZXQuY24  
      vY3JsL0xPQ0FMU00yL0xPQ0FMU00yXzAuY3JsMECgPqA8hjpodHRwczovL2Nybc5pc21nbmVOL  
      mNuL2Nybc5pMT0NBTFNNMi9pbmMvTE9DQUxTTTJfaW5jXzAuY3JsMBOGA1UdDgQWBBR1zT2DRBt
```



```
GU81NNfmDSMiICVjKvzAfBgNVHSMEGDAWgBQ78fsiTpugyfgIrS+tCqf3oTpz1zBTBgNVHSAET
DBKMEgGCiqBHIbvMgYEAQEwOjA4BggrBgEFBQcCARYsaHR0cHM6Ly9jcmwuaXNpZ25ldC5jbi9
jcHMvTE9DQUxTTTIvY3BzLmhObWwwDAYIKoEcz1UBg3UFAANJADBGAiEAz3ueB71NEVYTsQ+wr
ckoQ+u9hi1Ls/C19TeaSrK+GcQCIQC/Euzxhze900FEZPQDd4PJ0tUEceZCcZvu4j9CUZ/wOw=
=",
    "jobStatus": "FINISH"
  }
}
```

3.4.2.2 获取签名任务明细

- **功能说明：**根据查询条件，查询签名任务详情或者授权签名任务详情；检索时指定处理日期，查询当日任务，否则查询三日内处理过的任务。当根据任务 id 查询时，可以做获取任务签名结果使用。（注：做获取任务结果使用时，任务待签且已过期时不再抛异常，而是将任务设置为已过期状态“EXPIRE”返回）；可以根据“authorizeeUserId”查询某个人代签的签名任务详情；根据“userId”可以查询某人由自己发起的签名任务的详情。
- **请求地址：**http://host:port/coss/service/v1/getSignJobOrAuthorizeJobs
- **请求方式：**POST
- **请求参数：**

字段	类型	是否必须	说明
authorizeeUserId	String	否	被授权人用户 ID
signDataId	String	否	签名任务的 ID
operateTime	String	否	处理签名任务时间
userId	String	否	签名任务所属人用户 ID
currentPage	int	否	页码（默认 1）
pageSize	int	否	每页数据条数（默认 20）



● 请求示例:

```
{  
  
  "version": "1.0",  
  
  "appId": "APP_3B79535A195248FC956CB6FDF203A2C0",  
  
  "signAlgo": "HMAC",  
  
  "currentPage": 1,  
  
  "pageSize": 10,  
  
  "operateTime": "2022-07-29",  
  
  "signDataId": "SD_7eb0c4d0-464c-47a6-9fc7-12e770922a75",  
  
  "userId": "fdf1e58f063eeca589e1f9250cf447806c1a47e8982349586187caa33bb80557",  
  
  "authorizeeUserId": "5cf55a4c3a2c5265d756db9d57be9e1f2000857fc066ae02a2334d2b  
bf329357",  
  
  "signature": "v6kbxzEVSyzWMq/orNbSqX2hrW+dRwSvaaZGzH1f8c="
```

● 响应参数:

字段	类型	是否必须	说明
id	String	是	签名任务 id
userId	String	否	签名任务所属人 id
status	String	是	签名任务状态（UNSIGN：待签，FINISH：已签，EXPIRE：过期，REVOKE：签名任务被服务端撤销，REFUSE：签名任务被客户端拒绝）
resultData	String	否	签名结果
createTime	String	是	签名任务创建时间
updateTime	String	是	签名任务处理时间



authorizedUserId	String	否	被授权人用户 id
authorizedUserName	String	否	被授权人姓名
authorizedMobile	String	否	被授权人电话
authorizedJobId	String	否	被授权人工号
signCert	String	否	签名证书
expiryDate	Long	是	任务有效期截止时间（毫秒）
pageNum	int	是	当前页码
pageSize	int	是	每页数据条数
totalNum	int	是	数据总数
totalPage	int	是	总页数

● 响应示例：

```
{
  "status": 200,
  "message": "SUCCESS",
  "data": {
    "signList": [
      {
        "id": "SD_7eb0c4d0-464c-47a6-9fc7-12e770922a75",
        "userId": "fdf1e58f063eeca589e1f9250cf447806c1a47e8982349586187caa33bb80557",
        "createtime": "2022-07-22 16:47:55",
        "updatetime": "2022-07-29 10:04:41",
        "status": "FINISH",
        "authorizedUserId": "5cf55a4c3a2c5265d756db9d57be9e1f2000857fc066ae02a2334d2bbf329357",
        "authorizedUserName": "王同学",
        "authorizedMobile": "15665006990",
        "authorizedJobId": "110083",
        "resultData": "ZmJhNTFjYjUyZDc5NDAXN2JkY2Z1NmQyNTMyMWEyMjE="
      }
    ]
  }
}
```



// 在根据任务 id 检索时，会返回该字段（signCert）

```
    "signCert": "MIIEZjCCBAygAwIBAgIKGhAAAAAAMVNDAKBggqgRzPVQGDDTBSMQ  
swCQYDVQQGEwJDTjEVMBMGA1UECgwMQkpDQSBURVNUENBMRUwEwYDVQQLDxCSkNBIFRFU1Qg  
QOEFTATBgNVBAU2Nzg5MDAi.....BgoqgRyG7zICAQEXBBQMEjJAMjE1MDA5UVQwNDU2Nzg5  
MDAVBggqgRzQFAQBBAQJDACONTY3ODkwMBYGCiqBHIbvMgIBAR4ECAwGMTAwMzIyMAoGCCqBHM  
9VAYN1A0gAMEUCIQD1Apxq6BiTSRSy92qz0YAAQgj0Say/0gFHo4CQm2fQT9QIgJu4PpkG9e2BQ  
kWWWmOuj17FWt8wCSaiyc7lpPziQoIc=",  
    "expiryDate": 1658566075350  
  }  
],  
  "pageNum": 1,  
  "pageSize": 10,  
  "currentNum": 1,  
  "totalNum": 1,  
  "totalPage": 1  
}  
}
```

3.4.2.3 撤销签名任务

- 功能说明：通过业务端撤销已经添加的签名任务
- 请求地址：http://host:port/coss/service/v1/revokeSignJob
- 请求方式：POST
- 请求参数：

字段	类型	是否必须	说明
signJobId	String	是	签名任务 ID（添加签名任务接口返回的 signDataId）

- 请求示例：

```
{  
  "version": "1.0",
```



```
"appId":"APP_0157E810F6894BEEB4DEC70331720557",  
  
"signAlgo":"HMAC",  
  
"signJobId":"SD_e7841451-8ba2-48ce-b4ef-5a7ddcf0fb86",  
  
"signature":"NTrvHt0NUF7Crm3ia9daxFBMuDduNQAtZ2wah10t7SA="
```

```
}
```

- 返回参数:

公共参数

- 响应示例:

```
{  
  
  "status": 200,  
  
  "message": "SUCCESS"  
  
}
```

3.4.2.4 查询印章图片

- 功能说明: 用于查询个人的印章图片
- 请求地址: <http://host:port/coss/service/v1/queryImage>
- 请求方式: POST
- 请求参数:

字段	类型	是否必须	说明
userId	String	是	用户 id

- 请求示例:

```
{  
  
  "version":"1.0",  
  
  "appId":"APP_0157E810F6894BEEB4DEC70331720557",  
  
  "signAlgo":"HMAC",  
  
  "userId":"548a43ae04f109a2c8c812c95673579a03afd8512d7fc85f9abdc9de7066b2af",
```

```
"signature": "2muiLLUb1hY6WfF6uF9NhZQbQF0bzVPPNq1o8h/SKdA="
```

}

● 返回参数:

字段	类型	是否必须	说明
userId	String	否	用户 id
image	String	否	印章图片 base64

● 响应示例:

 $\{$

```
"status": 200,
```

```
"message": "SUCCESS",
```

```
"data": {
```

"userId":

"548a43ae04f109a2c8c812c95673579a03afd8512d7fc85f9abdc9de7066b2af".

"image":

[illegible]



```
+ /v///yH5BA1kAPAAALAAAAABQADwAAAjFAOEJHEiwoMGDCBMqXMiwoC0HECNKnEixosWLGDNq3
Mixo8ePIEOKHEmypMmTKFOqXMmypoXMGPKnEmzps2bOBOWyKmzgE+fAneyFLqRKNGgGn8qXfq
To1GCQHkqfIpU6sG1A4UetQolq9etXL0ajBp2LFS18LAmZY02o9q0cMsWpBoXrNyqde9eFRtXL
96+dsse1drXZOCJgwEffsi2MdmKTbNuXYxzst/LmDNr3sy5s+fPoEOLHk26tOnTqFOrXs16ZEA
AOw=="
}
}
```

3.4.3 自动签名场景接口

用户可通过业务端或 APP 中开启自动签名功能，业务端调用[开启自动签接口](#)，移动协同签名系统返回签名 ID，生成二维码；用户打开 app，扫码并签名；业务系统轮询调用[获取签名结果接口](#)获取到签名结果，提示签名成功，开启成功。用户在 PC 端发起自动签名，业务系统调用[自动签名接口](#)完成对数据的自动签名，自动签名成功。用户点击关闭自动签名，业务系统调用[关闭自动签接口](#)，关闭自动签名成功。自动签名关闭后，用户再次使用自动签名时需要开启。

3.4.3.1 开启自动签

- **功能说明：**通过业务端为指定的用户添加自动签名的权限，开启自动签名时需要用户进行扫码签名或基于任务 ID 签名，然后通过[3.4.3 获取签名结果接口](#)判断授权签名结果，只有用户对开启自动签名的签名任务进行授权签名完成后才能调用[3.5.3 自动签名接口](#)进行自动签名。
- **请求地址：**http://host:port/coss/service/v1/startAutoSign
- **请求方式：**POST
- **请求业务参数：**

字段	类型	是否必须	说明
userId	String	否	用户 Id
timeRegion	String	是	自动签名的过期时间，单位为秒（最长 86400 秒，即 24 小时）
callbackUr	String	否	回调地址



1			
---	--	--	--

● 请求示例:

```
{  
  
  "version": "1.0",  
  
  "appId": "APP_0157E810F6894BEEB4DEC70331720557",  
  
  "signAlgo": "HMAC",  
  
  "userId": "548a43ae04f109a2c8c812c95673579a03afd8512d7fc85f9abdc9de7066b2af",  
  
  "timeRegion": "1440",  
  
  "signature": "RQQ+PFL00J7WkI1ckkBLj1ZQ5jxTZMu+fd0DEwo3KKo="
```

● 返回参数:

字段	类型	是否必须	说明
signDataId	String	否	待签任务 ID
qrCode	String	否	二维码内容
data	String	否	原文

● 响应示例:

```
{  
  
  "status": 200,  
  
  "message": "SUCCESS",  
  
  "data": {  
  
    "signDataId": "SD_e7841451-8ba2-48ce-b4ef-5a7ddcf0fb86",  
  
    "qrCode":  
  
    "{ \"id\": \"APP_E39718EFFF514DB293F934E649B747A9\", \"sUrl\": \"https://coss-  
dev.isignet.cn:18759/coss/mobile/v1/getServiceInfo\", \"o\": \"KHEODgQEUHoQY  
3VnKGRrZF5EUEntC1JQCmBnazMKXQdGJVQdBF16NzczCUADEHgEEh0aOSojNU1KRyEJdX4TFG8
```



```
lNiIcGQocYggSABZ9cS4=\",\\"cv\\":\\"2.0\\"}"
```

```
}
```

```
}
```

注：对于回调地址说明，集成方需提供回调接口。结构如下：

入参说明：

字段	类型	是否必须	说明
signJobId	String	是	签名任务 id
status	String	是	签名任务状态
msspId	String	是	用户唯一标识
signResult	String	否	签名结果
signCert	String	否	签名证书

返回参数：

字段		类型	是否必须	说明
status		Int	是	状态码
message		String	是	提示信息
data		String	否	响应数据
	signJobId	String	否	签名任务 id
	status	String	否	签名任务状态

请求报文示例：

```
{
```

```
  "signJobId": "SD_000697d0-4e1b-4cb7-b9cf-619e347892f8",
```

```
  "status": "FINISH",
```

```
  "signResult":
```

```
"MEYCIQDvcy6/WHJ+8QS1S1Ja1Q9FBa6gtnboxwndyQpHBBbcaQIhAMeZjymkozFKm1MaD2xq/L4JhW3gO5XTm  
GL6S00MzZwr",
```

```
  "signCert":
```

```
"MIIDAzCCAqagAwIBAgISIAIX9hLyA5saPDc8EKE2osMvMAwGCCqBHM9VAYN1BQAwpJELMAkGA1UEBgcQ  
04xDtALBgNVBAoMBEJKQ0ExDTALBgNVBASMBEJKQ0ExETAPBgNVBAMMCExpQ0FMU00yMB4XDTE5MDYwNDE  
xMDMzN1oXDTE0MDYwNDExMDMzN1owcDELMAkGA1UEBgcQ04xDzANBgNVBAMMBuW8oOS4iTFQME4GCgmSJ  
omT8ixkAQEMQDU0OGE0M2F1MDRmMTA5YTJjOGM4MTJjOTU2NzM1NzlhMDNhZmQ4NTEyZDdmYzg1Zj1hYmR
```



```
jOWR1NzA2NmIyYWYwWTATBgqhkJOPQIBBgqgRzPVQGCLQNCAAQ1TH7W1yLLMd4honFRQJPZDAegXM7K1
/OZ8jY2wuzwTXs55jUQZAu4C9Vj6tLp6Qem+01v7v/mAUyM7ePj/CT5o4IBTjCCAUowCwYDVR0PBAQDAgb
AMAKGA1UdEwQCMAAwEwYDVR01BAAwCgYIKwYBBQUHAWMwgYUGA1UdHwR+MHwwOKA2oDSGMmh0dHBzOi8vY
3JsLmlzaWduZXQuY24vY3JsL0xPQ0FMU00yL0xPQ0FMU00yXzAuY3JsMECgPqA8hjpodHRwczovL2NybC5
pc2lnbmV0LmNuL2NybC9MT0NBTFNNMi9pbmMvTE9DQUxTTTJfaW5jXzAuY3JsMBOGA1UdDgQWBBR1zT2DR
BtGU81NNfmDSMiICVjKvzAfBgNVHSMEGDAWgBQ78fsiTPugyfgIrS+tCqf3oTpz1zBTBgNVHSAETDBKMEg
GCiqBHIbvMgYEAQEW0jA4BggrBgEFBQcCARYsaHR0cHM6Ly9jcmwuaXNpZ25ldC5jb25jcmVTE9DQUxTT
TivY3BzLmh0bWwwDAYIKoEcz1UBg3UFAANJADBGAIeAz3ueB71NEVYTsQ+wrckoQ+u9hi1Ls/C19TeaSrK
+GcQCIQC/Euzxhze900FEZPQDd4PJ0tUEceZCcZvu4j9CUZ/w0w=="
```

```
}
```

响应成功报文示例:

```
{
  "status": 200,
  "message": "SUCCESS",
  "data": {
    "signJobId": "SD_000697d0-4e1b-4cb7-b9cf-619e347892f8",
    "status": "FINISH",
    ...
  }
}
```

3.4.3.2 关闭自动签

- 功能说明: 关闭用户自动签名的权限
- 请求地址: <http://host:port/coss/service/v1/endAutoSign>
- 请求方式: POST
- 请求参数:

字段	类型	是否必须	说明
userId	String	是	用户 Id



signToken	String	否	自动签 token，开启自动签返回的 signDataId
-----------	--------	---	-------------------------------

● 请求示例：

```
{  
  
  "version": "1.0",  
  
  "appId": "APP_0157E810F6894BEEB4DEC70331720557",  
  
  "signAlgo": "HMAC",  
  
  "userId": "548a43ae04f109a2c8c812c95673579a03afd8512d7fc85f9abdc9de7066b2af",  
  
  "signature": "2muiLLUb1hY6WfF6uF9NhZQbQF0bzVPPNql08h/SKdA="
```

● 返回参数：

公共参数

● 响应示例：

```
{  
  
  "status": 200,  
  
  "message": "SUCCESS"
```

3.4.3.3 自动签名

- 功能说明：该接口用户业务系统服务端调用，对于开启自动签名权限的用户，将直接返回签名结果
- 请求地址：http://host:port/coss/service/v1/autoSign
- 请求方式：POST
- 请求参数：

字段	类型	是否必填	参数说明
userId	String	是	用户 id



title	String	否	标题（最长不超过100个字符）
dataType	String	是	原文类型（DATA：原文，HASH：hash 数据 WEB_SEAL：网页签章）
algo	String	是	签名算法(SM3withSM2)
description	String	否	描述（最长不超过200个字符）
data	String	是	待签数据（ 必须是 base64编码 ）
signToken	String	否	开启自动签返回的 signDataId

● 请求示例：

```
{  
  
  "version": "1.0",  
  
  "appId": "APP_0157E810F6894BEEB4DEC70331720557",  
  
  "signAlgo": "HMAC",  
  
  "userId": "548a43ae04f109a2c8c812c95673579a03afd8512d7fc85f9abdc9de7066b2af",  
  
  "title": "测试签名",  
  
  "dataType": "DATA",  
  
  "algo": "SM3withSM2",  
  
  "description": "签名数据描述信息",  
  
  "data": "562+5ZCN5rWL6K+V5pWw5o2u",  
  
  "signToken": "SD_e7841451-8ba2-48ce-b4ef-5a7ddcf0fb86",  
  
  "signature": "aUjWYeRnteYKmh0TPdJF5VdHqWB/dMXQ6szHx0c4jXY="
```

● 响应参数：



字段	类型	是否必填	参数说明
signDataId	String	否	签名任务 id
signResult	String	否	签名结果
signCert	String	否	证书

● 响应示例:

```
{  
  "status": 200,  
  "message": "SUCCESS",  
  "data": {  
    "signDataId": "SD_18eb25f8-3d15-4971-a60c-730c1480d837",  
    "signResult":  
    "MEYCIQDvcy6/WHJ+8QS1S1Ja1Q9FBa6gtnboxwndyQpHBBbcaQIhAMeZjymkozFKm1MaD2xq/  
L4JhW3gO5XTmGL6S00MzZwr",  
    "signCert":  
    "MIIDAzCCAqagAwIBAgISIAIX9hLyA5saPDc8EKE2osMvMAwGCCqBHM9VAYN1BQAwpJELMAkGA  
1UEBgwCQ04xDALBgNVBAoMBEJJKQ0ExDTALBgNVBASMBEJJKQ0ExETAPBgNVBAMMCEXPQ0FMU00  
yMB4XDTE5MDYwNDEwMDMzN1oXDTIOMDYwNDEwMDMzN1owcDELMAkGA1UEBgwCQ04xDzANBgNVB  
AMMBuW8oOS4iTFQME4GCgmSJomT8ixkAQEMQDU00GEOM2F1MDRmMTA5YTJjOGM4MTJjOTU2NzM  
1NzlhMDNhZmQ4NTEyZDdmYzgzZj1hYmRjOWR1NzA2NmIyYWYwWTATBgqhkhjOPQIBBgqgRzPV  
QGCLQNCAAQ1TH7W1yLLMd4honFRQJPZDAegXM7K1/OZ8jY2wuzwTXs55jUQZAu4C9Vj6tLp6Qe  
m+01v7v/mAUyM7ePj/CT5o4IBTjCCAUowCwYDVR0PBAQDAgbAMakGA1UdEwQCMAAwEwYDVR01B  
AwwCgYIKwYBBQUHAWMwgYUGA1UdHwR+MHwwOKA2oDSGMmhOdHBzOi8vY3JsLmlzaWduZXQuY24  
vY3JsL0xPQ0FMU00yL0xPQ0FMU00yXzAuY3JsMECgPqA8hjpodHRwczovL2Nybc5pc21nbmVOL  
mNuL2Nybc9MT0NBTFNNMi9pbmMvTE9DQUxTTTJfaW5jXzAuY3JsMBOGA1UdDgQWBBR1zT2DRBt  
GU81NNfmDSMiICVjKvzAfBgNVHSMEGDAWgBQ78fsiT PugyfgIrS+tCqf3oTpz1zBTBgNVHSAET  
DBKMEgGCiqBHIbvMgYEAQEwOja4BggrBgEFBQcCARYsaHR0cHM6Ly9jcmwuaXNpZ25ldC5jb29  
jCHMvTE9DQUxTTTJvY3BzLmh0bWwwDAYIKoEcz1UBg3UFAANJADBGAiEAz3ueB71NEVYTsQ+wr  
ckoQ+u9hi1Ls/C19TeaSrK+GcQCIQC/Euzxhze900FEZPQDd4PJ0tUEceZCcZvu4j9CUZ/w0w=
```



="

}

}

3.4.3.4 查询自动签时间

- **功能说明：**该接口用户业务系统服务端调用，用于查询用户自动签状态以及结束时间
- **请求地址：**http://host:port/coss/service/v1/queryAutoSignRegion
- **请求方式：**POST
- **请求参数：**

字段	类型	是否必须	说明
userId	String	否	用户 ID，即 MSSPID

- **请求示例：**

```
{  
  
  "version": "1.0",  
  
  "appId": "APP_7B3F36A14E99410A80B37AEF332E3247",  
  
  "signAlgo": "HMAC",  
  
  "userId": "057fbd512ab6455bdeb3a34c70344fa3e2a3d162a46fa203de867873dfd08e30",  
  "signature": "Yrh9zFW1uerRSzEVNCc/C/dluoejt+Th+w+18JNq7R4=",  
  
}
```

- **响应示例：**

```
{  
  
  "status": 200,  
  "message": "SUCCESS",  
  "data": {  
    "userId":  
      "057fbd512ab6455bdeb3a34c70344fa3e2a3d162a46fa203de867873dfd08e30",  
    "autoSignStatus": "STARTED",  
    "endTime": "2022-05-09 15:50"  
  }  
}
```



```
}  
  
}
```

3.4.4 验证签名场景接口

3.4.4.1 验证数据签名

- 功能说明：用于验证数据签名
- 请求地址：http://host:port/coss/service/v1/verifySign
- 请求方式：POST
- 请求参数：

字段	类型	是否必须	说明
signAlg	String	是	签名算法应为 SM3withSM2
plain	String	是	签名原文数据base64
signValue	String	是	签名值 base64
cert	String	是	证书 base64

- 请求示例：

```
{  
  
  "version": "1.0",  
  
  "appId": "APP_0157E810F6894BEEB4DEC70331720557",  
  
  "signAlgo": "HMAC",  
  
  "signAlg": "SM3withSM2",  
  
  "plain": "562+5ZCN5rWL6K+V5pWw5o2u",  
  
  "signValue":  
  "MEYCIQDvcy6/WHJ+8QS1S1Ja1Q9FBa6gtndyQpHBBbcaQIhAMeZjymkozFKm1MaD2xq/L4JhW3gO  
  5XTmGL6S00MzZwr",  
}
```



"cert":

"MIIDAzCCAqagAwIBAgISIAIX9hLyA5saPDc8EKE2osMvMAwGCCqBHM9VAYN1BQAwpjELMAkGA1UEBgcwQ04xDTALBgNVBAoMBEJkQ0ExDTALBgNVBASMBEJkQ0ExETAPBgNVBAMMCExpQ0FMU00yMB4XDTE5MDYwNDExMDMzN1oXDTIOMDYwNDExMDMzN1owcDELMAkGA1UEBgcwQ04xDTALBgNVBAMMBuW8oOS4iTFQME4GCgmSJomT8ixkAQEMQDU00GEOM2F1MDRmMTA5YTJjOGM4MTJjOTU2NzM1NzlhMDNhZmQ4NTEyZDdmYzg1Zj1hYmRjOWR1NzA2NmIyYWYwWTATBgqcqhkJOPQIBBgqgRzPVQGCLQNCAAQ1TH7W1yLLMd4honFRQJPZDAegXM7K1/OZ8jY2wuzwTXs55jUQZAu4C9Vj6tLp6Qem+01v7v/mAUyM7ePj/CT5o4IBTjCCAUowCwYDVR0PBAQDAgbAMAKGA1UdEwQCAAwEwYDVR01BAwwCgYIKwYBBQUHAWMwgYUGA1UdHwR+MHwwOKA2oDSGMmh0dHBzOi8vY3JsLmlzaWduZXQuY24vY3JsL0xPQ0FMU00yL0xPQ0FMU00yXzAuY3JsMECgPqA8hjpodHRwczovL2Nybc5pc2lnbmV0LmNuL2Nybc9MT0NBTFNNMi9pbmMvTE9DQUxTTTJfaW5jXzAuY3JsMBOGA1UdDgQWBBR1zT2DRBtGU81NNfmDSMiICVjKvzAfBgNVHSMEGDAwgbQ78fsiTPugyfgIrS+tCqf3oTpz1zBTBgNVHSAETDBKMEgGCiqBHIbvMgYEAQEwOjA4BggrBgEFBQcCARYsaHR0cHM6Ly9jcmwuaXNpZ25ldC5jbi9jcHMvTE9DQUxTTTivY3BzLmh0bWwwDAYIKoEcz1UBg3UFAANJADBGAiEAz3ueB71NEVYTsQ+wrckoQ+u9hi1Ls/C19TeaSrK+GcQCIQC/Euzxhze900FEZPQDd4PJ0tUEceZCcZvu4j9CUZ/w0w==",

"signature": "7bNni8xEv6k9UjXnzMRM3klA4M8BAAN/h/xgzMCJiQY="

}

● 返回参数:

公共响应参数

● 响应示例:

{

"status": 200,

"message": "SUCCESS",

}

3.4.4.2 验证网页签名

● 功能说明: 网页签名验签

● 请求地址: http://host:port/coss/service/v1/verifyEssSign



● 请求方式：POST

● 请求参数：

字段	类型	是否必须	说明
signAlg	String	是	签名算法应为 SM3withSM2
plain	String	是	签名数据的 base64
signValue	String	是	签名值的 base64

● 请求示例：

```
{  
  "version": "1.0",  
  "appId": "APP_0157E810F6894BEEB4DEC70331720557",  
  "signAlgo": "HMAC",  
  "signAlg": "SM3withSM2",  
  "plain": "562+5ZCN5rWL6K+V5pWw5o2u",  
  "signValue":  
    "MEUCIDG15MQWms+R9rH1pGSOLaaMwsB9rZ+ZtHEV5xyweVzTAiEApoGGEpzEcfz1QG12Eij5ddnUIpbX+RqZQ  
    SI8hunWUec=````MFkwEwYHkoZIZj0CAQYIKoEcz1UBgiODQgAEJUx+lpciyzHeIaJxUUCT2QwHoFzOytfzmfI  
    2NsLs8E170eY1EGQLuAvVY+rS6ekHpvjtb+7/5gFMj03j4/wk+Q==````MIIDAzCCAqagAwIBAgISIAIX9hLyA  
    5saPDc8EKE2osMvMAwGCCqBHM9VAYN1BQAwpjELMAkGA1UEBgcwCQ04xDTALBgNVBAoMBEJKQ0ExDTALBgNVBA  
    MBEJKQ0ExETAPBgNVBAMMCExpQ0FMU00yMB4XDTE5MDYwNDExMDMzN1oXDTIOMDYwNDExMDMzN1owcDELMAkGA  
    1UEBgcwCQ04xDzANBgNVBAMMBuW8oOS4iTFQME4GCgmSJomT8ixkAQEMQDU00GEOM2F1MDRmMTA5YTJjOGM4MTJ  
    jOTU2NzM1NzlhMDNhZmQ4NTEyZDdmYzg1Zj1hYmRjOWR1NzA2NmIyYWYwWTATBgqcqhkJOPQIBBggqgRzPVQGCL  
    QNCAAQ1TH7W1yLLMd4honFRQJPZDAegXM7K1/OZ8jY2wuzwTXs55jUQZAu4C9Vj6tLp6Qem+01v7v/mAUyM7eP  
    j/CT5o4IBTjCCAUowCwYDVR0PBAQDAgbAMAKGA1UdEwQCMAAwEwYDVR01BAwwCgYIKwYBBQUHAWMwgYUGA1UdH  
    wR+MHwwOKA2oDSGMmh0dHBzOi8vY3JsLmlzaWduZXQuY24vY3JsL0xPQ0FMU00yL0xPQ0FMU00yXzAuY3JsMEC  
    gPqA8hjpodHRwcZovL2Nybc5pc2lnbmV0LmNuL2Nybc9MT0NBTFNNMi9pbmMvTE9DQUxTTTJfaW5jXzAuY3JsM  
    BOGA1UdDgQWBBr1zT2DRBtGU81NNfmDSMiICVjKvzAfBgNVHSMEGDAWgBQ78fsiT PugyfgIrS+tCqf3oTpz1zB
```

[illegible]

```
"signature": "7bNni8xEv6k9UjXnzMRM3k1A4M8BAAN/h/xgzMC.JiQY="
```

}

- 返回参数:

字段	类型	是否必须	说明
status	int	是	状态码（200 成功）
message	String	是	状态信息
transId	String	否	业务流水号

● 响应示例:



```
{  
  
    "status": 200,  
  
    "message": "SUCCESS",  
  
}
```

3.4.5 授权场景接口

在实际工作中，由于种种原因，A 的工作可能需要 B 或者 C 以 A 的名义代替完成；此时，B 或 C 就需要预先得到 A 授予的权限。授权功能的出现，就是为了解决这类生产生活中的实际问题。目前授权功能主要由移动端进行授权的开启、关闭、查询等等，业务系统只支持了查询某个人的授权明细。

3.4.5.1 授权关系明细查询

- **功能说明：**根据给定查询条件，查询所有授权关系及每个授权关系的授权详情
- **请求地址：**http://host:port/coss/service/v1/getAuthorizedDetail
- **请求方式：**POST

请求参数：

字段	类型	是否必须	说明
userId	String	是	用户id
status	String	是	授权状态(WAIT: 等待, AUTHORIZE: 授权中, REFUSE: 拒绝, CANCEL: 取消)

- **请求示例：**

```
{  
  
    "version": "1.0",  
  
    "appId": "APP_3B79535A195248FC956CB6FDF203A2C0",  
  
}
```



```
"signAlgo": "HMAC",  
  
"userId": "fdf1e58f063eeca589e1f9250cf447806c1a47e8982349586187caa33bb80557",  
  
"status": "REFUSE",  
  
"signature": "+YXT25ddtnREZDi0d7PsHA9v+iH2Vjw+L84ZAZr95s4="
```

```
}
```

● 响应参数:

字段	类型	是否必须	说明
businessId	String	是	业务 id
authorizerId	String	是	授权人用户 id
authorizerName	String	是	授权人姓名
authorizerCardType	String	是	授权人证件类型 (SF: 身份证, HK: 户口簿, HZ: 护照, JR: 军人证, GA: 港澳台居民身份证, WJ: 武警身份证, BM: 边民出入境通行证, HX: 回乡证, QT: 其它)
authorizerCardId	String	是	授权人证件号
authorizerJobNumber	String	否	授权人工号
authorizerPhone	String	是	授权人手机号
authorizerDepartment	String	否	授权人部门名称
authorizeeId	String	是	被授权人用户 id
authorizeeName	String	是	被授权人姓名
authorizeeCardType	String	是	被授权人证件类型 (SF: 身份证, HK: 户口簿, HZ: 护照, JR: 军人证, GA: 港澳台居民身份证, WJ: 武警身份证, BM: 边民出入境通行证, HX: 回乡证, QT: 其它)
authorizeeCardId	String	是	被授权人证件号
authorizeeJobNumber	String	否	被授权人工号

authorizeePhone	String	是	被授权人电话号
authorizeeDepartmen t	String	否	被授权人部门名称
createTime	String	是	授权关系开始时间
endTime	String	是	授权关系结束时间
status	String	是	授权状态(WAIT: 等待, AUTHORIZE: 授 权中, REFUSE: 拒绝, CANCEL: 取消)

● 响应示例:

```

{

    "status": 200,

    "message": "SUCCESS",

    "data": {

        // 查询人有证书代表发出的授权集合, 无证书代表收到授权的集合

        "dataList": [

            {

                "businessId": "AUTHORIZ_eed317663a824e7ea656a0eb7a7dd28f",

                "authorizerId": "fdf1e58f063eeca589e1f9250cf447806c1a47e8982349586187  
caa33bb80557",

                "authorizerName": "埃弗拉",

                "authorizerPhone": "15662176389",

                "authorizerDepartment": "开发部",

                "authorizerCardType": "其它(QT)",

                "authorizerCardId": "4567890",

                "authorizerJobNumber": "",

                "authorizeeCardType": "其它(QT)",
            }
        ]
    }
}
    
```



```
"authorizeeCardId": "3333333333",

"authorizeeJobNumber": "",

"authorizeeId": "5cf55a4c3a2c5265d756db9d57be9e1f2000857fc066ae02a233
4d2bbf329357",

"authorizeeName": "王同学",

"authorizeePhone": "15662176380",

"authorizeeDepartment": "维修部",

"createTime": "2022-06-13 16:47:31",

"endTime": "2022-07-29 17:00:00",

"status": "AUTHORIZE"

}

],

// 有证书的查询人收到的授权集合

"othersList": [

{

"businessId": "AUTHORIZ_de402155b1d643abb8b60765c359cbf7",

"authorizerId": "9604d453d03687b087ae0e5faa3ad801735116c82356cd7c30d4
7c35736a94ef",

"authorizerName": "华文行楷22",

"authorizerPhone": "18512556987",

"authorizerDepartment": "347865432",

"authorizerCardType": "其它(QT)",

"authorizerCardId": "345678654",

"authorizerJobNumber": "",
```



```
"authorizeCardType": "其它(QT)",

"authorizeCardId": "4567890",

"authorizeJobNumber": "",

"authorizeId": "fdf1e58f063eeca589e1f9250cf447806c1a47e8982349586187
caa33bb80557",

"authorizeName": "埃弗拉",

"authorizePhone": "15662176389",

"authorizeDepartment": "开发部",

"createTime": "2022-06-13 16:47:31",

"endTime": "2022-07-30 16:00:00",

"status": "AUTHORIZE"

}

]

}

}
```

3.4.6 证书管理场景接口

用于证书管理，查询证书、获取证书信息、验证证书真实性。

3.4.6.1 解析证书信息

- 功能说明：解析证书信息
- 请求地址：<http://host:port/coss/service/v1/getCertInfo>
- 请求方式：POST
- 请求参数：

字段	类型	是否必须	说明
----	----	------	----



cert	String	是	证书的 base64
oid	String	否	OID 项

● 请求示例:

```
{  
  
  "version": "1.0",  
  
  "appId": "APP_0157E810F6894BEEB4DEC70331720557",  
  
  "signAlgo": "HMAC",  
  
  "cert":  
    "MIIDAzCCAqagAwIBAgISIAIX9hLyA5saPDc8EKE2osMvMAwGCCqBHM9VAYN1BQAwPjELMAkGA1UEBgcwQ04xDTALBgNVBAoMBEJKQ0ExDTALBgNVBAsMBEJKQ0ExETAPBgNVBAMMCEExPQ0FMU00yMB4XDTE5MDYwNDExMDMzN1oXDTE0MDYwNDExMDMzN1owcDELMAkGA1UEBgcwQ04xDTALBgNVBAMMBuW8oOS4iTfQME4GCgmSJOmT8ixkAQEMQDU0OGE0M2F1MDRmMTA5YTJjOGM4MTJjOTU2NzM1NzlhMDNhZmQ4NTEyZDdmYzg1Zj1hYmRjOWR1NzA2NmIyYWYwWTATBgqcqhkJOPQIBBggqRzPVQGCLQNCAAQ1TH7W1yLLMd4honFRQJPZDAegXM7K1/OZ8jY2wuzwTXs55jUQZAu4C9Vj6tLp6Qem+01v7v/mAUyM7ePj/CT5o4IBTjCCAUowCwYDVR0PBAQDAgbAMAKGA1UdEwQCAAwEwYDVR01BAAwCgYIKwYBBQUHAWMwgYUGA1UdHwR+MHwwOKA2oDSGMmhOdHBzOi8vY3JsLmlzaWduZXQuY24vY3JsL0xPQ0FMU00yL0xPQ0FMU00yXzAuY3JsMECgPqA8hjpodHRwczovL2Nybc5pc2lnbmV0LmNuL2Nybc5MT0NBTFNNMi9pbmMvTE9DQUxTTTJfaW5jXzAuY3JsMBOGA1UdDgQWBBR1zT2DRBtGU81NNfmDSMiICVjKvzAfBgNVHSMEGDAWgBQ78fsiTfPugyfgIrS+tCqf3oTpz1zBTBgNVHSAETDBKMEgGCiqBHIbvMgYEAQEwOjA4BggrBgEFBQcCARYsaHR0cHM6Ly9jcmwuaXNpZ25ldC5jb29jcmVTE9DQUxTTTivY3BzLmh0bWwwDAYIKoEcz1UBg3UFAANJADBGAIeAz3ueB71NEVYTsQ+wrckoQ+u9hi1Ls/C19TeaSrK+GcQCIQC/Euzxhze900FEZPQDd4PJ0tUEceZCcZvu4j9CUZ/w0w==",  
  
  "signature": "qG/IAIjBUjjKJoJdFjts9p6/OPQbh9Ge5sWyeIIlKZA="
```

● 返回参数:

字段	类型	是否必须	说明
userId	String	否	用户 ID



certCN	String	否	证书 CN 项
certDN	String	否	证书 DN 项
serialNumber	String	否	证书序列号
sigAlgName	String	否	签名算法
oidResult	String	否	OID 项

● 响应示例:

```
{
  "status": 200,
  "message": "SUCCESS",
  "data": {
    "userId":
      "548a43ae04f109a2c8c812c95673579a03afd8512d7fc85f9abdc9de7066b2af",
    "certCN": "张三",
    "certDN": "C=CN, CN=张三, UID=548a43ae04f109a2c8c812c95673579a03afd8512d7
      fc85f9abdc9de7066b2af",
    "serialNumber": "200217F612F2039B1A3C373C10A136A2C32F",
    "sigAlgName": "SM3withSM2"
  }
}
```

3.4.6.2 查询证书

- 功能说明: 查询证书详细信息
- 请求地址: `http://host:port/coss/service/v1/queryCert`
- 请求方式: POST
- 请求参数:

字段	类型	是否必须	说明
----	----	------	----



userId	String	是	用户 Id
deviceId	String	是	设备 id

● 请求示例:

```
{  
  
  "version": "1.0",  
  
  "appId": "APP_0157E810F6894BEEB4DEC70331720557",  
  
  "signAlgo": "HMAC",  
  
  "userId": "548a43ae04f109a2c8c812c95673579a03afd8512d7fc85f9abdc9de7066b2af",  
  
  "deviceId": "DEVICE_8d99871c-bf36-4a64-acee-a3b648cb08c5",  
  
  "signature": "qG/IAIjBUjjKJoJdFjts9p6/OPQbh9Ge5sWyeIIlKZA="
```

● 返回参数:

字段	类型	是否必须	说明
deviceId	String	是	设备 id
algoPolicy	String	是	证书算法, RSA/SM2
usage	String	是	密钥用途: SIGN: 签名证书 AUTH: 认证证书
cert	String	是	证书

响应示例:

```
{  
  
  "status": 200,  
  
  "message": "SUCCESS",  
  
  "data": {
```



```

    "certInfos": [{
    "deviceId": "DEVICE_fae2a200-68de-4c87-8aab-ada758ae5f75",
    "algoPolicy": "SM2",
    "usage": "SIGN",
    "cert": "MIIDAzCCAqagAwIBAgISIAIX9hLyA5saPDc8EKE2osMvMAwGCCqBHM9VAYN1BQ
    AwPjELMAkGA1UEBgcwQ04xDTALBgNVBAoMBEJKQ0ExDTALBgNVBAcMBEJKQ0E
    xETAPBgNVBAMCEXPQ0FMU00yMB4XDTE5MDYwNDExMDMzN1oXDTIOMDYwNDEx
    MDMzN1owcDELMAkGA1UEBgcwQ04xDzANBgNVBAMMBuW8oOS4iTfQME4GCgmSJ
    omT8ixkAQEMQDU00GEOM2F1MDRmMTA5YTJjOGM4MTJjOTU2NzM1NzlhMDNhZm
    Q4NTEyZDdmYzg1Zj1hYmRjOWR1NzA2NmIyYWYwWTATBgqhkJOPQIBBggqRz
    PVQGCLQNCAAQ1TH7WlyLLMd4honFRQJPZDAegXM7K1/OZ8jY2wuzwTXs55jUQ
    ZAu4C9Vj6tLp6Qem+01v7v/mAUyM7ePj/CT5o4IBTjCCAUowCwYDVR0PBAQDA
    gbAMakGA1UdEwQCMAAwEwYDVR01BAwwCgYIKwYBBQUHAWMwgYUGA1UdHwR+MH
    wwOKA2oDSGMmh0dHBzOi8vY3JsLmlzaWduZXQuY24vY3JsL0xPQ0FMU00yL0x
    PQ0FMU00yXzAuY3JsMECgPqA8hjpodHRwczovL2Nybc5pc2lnbmV0LmNuL2Ny
    bc9MT0NBTFNNMi9pbmMvTE9DQUxTTTJfaW5jXzAuY3JsMBOGA1UdDgQWBBR1z
    T2DRBtGU81NNfmDSMiICVjKvzAfBgNVHSMEGDAwGBQ78fsiTfPugyfgIrS+tCq
    f3oTpzlzBTBgNVHSAETDBKMEgGCiqBHIbvMgYEAQEwOja4BggrBgEFBQcCARY
    saHR0cHM6Ly9jcmwuaXNpZ25ldC5jb19jcmMvTE9DQUxTTTlY3BzLmh0bWww
    DAYIKoEcz1UBg3UFAANJADBGAiEAz3ueB71NEVYTsQ+wrckoQ+u9hi1Ls/C19
    TeaSrk+GcQCIQC/Euzxhze900FEZPQDd4PJ0tUEceZCcZvu4j9CUZ/w0w=="
    }]
  }
}

```

3.4.6.3 证书验证

- 功能说明：验证证书有效性及有效期
- 请求地址：<http://host:port/coss/service/v1/validationCert>
- 请求方式：POST
- 请求参数：

字段	类型	是否必须	说明
cert	String	是	base64 格式的证书

● 请求示例:

```
{  
    "version": "1.0",  
  
    "appId": "APP_0157E810F6894BEEB4DEC70331720557",  
  
    "signAlgo": "HMAC",  
  
    "cert":  
  
        "MIIFQTCCBCmgAwIBAgIKGzAAAAAAAFvqLzANBgkqhkiG9wOBAQUFADBSMQswCQYDVQQGEwJDTjENMAsGA  
1UECGwEQkpDQTEYMBYGAIUECWwPUHVibGljIFRydXNOIENBMRowGAYDVQQDDBFqdWJsawMGVHJlc3QgQOE  
tMTAeFw0yMDA5MTMxNjAwMDBBaFw0yMTA5MTQxNTU5NTlaAMIGTMQswCQYDVQQGEwJDTjEtMCsGA1UECgwK5  
YyX5Lqs5pWw5a2X6K6k6K+B6IKh5Lu95pyJ6ZmQ5YW55+4MRIwEAYDVQQDDAnlijok6zli4MxJDaiBkg  
qhkiG9wOBCEWFwwxpdxBlbmdbiOBiamNhLm9yZy5jbjeEbMBkGCgmSJomT8ixkASkMcZE4NTE5NTAyODYZM  
IGfMAOGCSqGSIsb3dQEBAQUAAAGNADCBiqKBgQDtX8ts21N05vVKYW5cvbtBOoqiOCpnCdDkg5ksvczzEZy  
m/YaV5IAJkp+WNaKBxGI5+XlYbbulj/I4rXKn2haP7BCa4SiXlSYcghfd09IzMndblWS+A+DsOu3bkxpHo  
Ov4ZPtgsvrZJvvgvHZGBWZvySGkodoo,Jv+7g7Ar6JSwIDAQABo4ICWTCCA1UwHwyDVROjBBgwFoAUrdv  
srwjUA7vr6+OT2w729FXOkwhQYDVR00BBYEfkK5mnM+9BDs+irRJ4ZEs45xljDeMASGA1UdDwQEAWIgW  
DCBrwyDVROfbIGnMIGkMG2ga6BppGcwZTELMAKGAIUEbhMCQ04xDALBGNVBAOMBEJKQOEExGDABGNVBAs  
MD1BIymxpYyBUcnVzdCBDQTEAMBGA1UEAwRUHVibGljIFRydXNOIENBLTEXETAPBgNVBAMTCGNhM2Nyby  
DMxMOgMaAvhilodHRwOi8vbGRhcC5iamNhLm9yZy5jbi9jcmwvcHRjYS9jYTdjcmwzMS5jcmlwczQYDVRO  
TBAlwADARBgllghkgBhvCAQEEBAMCAP8WHQYFK1YLWEEFFNGNDexMTIZmtk5MJAxMTY4NTEEXMBOGBSpWC  
wcIBBRTRjqxmTEYMze50TIwmTE2ODUXMTAgBgghghkgBhvHEagQUU0YOMTexmjmxOTkyMDExnjglMEWGwY  
IKlaGSAGBMAEEDzEWmJA4MDAwNzkONzg2MZalBgoqqRYG7zICAQQBBBBcyQ0BTRjqxmTEYMze50TIwmTE2O  
DUxMTAQbgtaghkgtBZQMATAJCgQbaHR0cdovL2JqY2Euub3JnlmNuL2JqY2EuY3JOMA8GBSspWFQEBBAYxMDA  
wODAwaQAYDVROgbDKwnza1BgkkqRG7zICAgewKDAmBggrBgEFBQCcARYaaHR0cdovL3d3dy5iamNhLm9yZy  
y5jbi9jcHMweWyKKoeChu8yAgeBHgQFDAMIMjaWDQYYJKozIHVCNAQEFBQADggEBABqpqbktOOMjokekDD/  
/S4WOr395zknvwBuEUqrHYNLKBfddaxoxIRRAab+z31GHq+8j5HIPnvHu5k91lxh9uvlwQGpt44f8dT3  
wtVQ34tTrly4ujPGjsWyOn7FacSmxuGqtRyoec2eoXv8Sig/+ZX34ERclBFBti4Kxi9yiEC.JsynV72Q
```



```
2CumXwMyFqeHtBmFL+tmGvpkmzfW8mGTfv5yHYP1jwQ/fwLR03v/kfMK2YgbrzWfvZ5X+m7eWrTtMRjE8
jK+eKEIpiG5zSbcj492Elrs9YaOMqhiJlIjgyljeDsqr7eDJjrNMxOZP6l6mdv6FHWsLR90hpiypOM=",
"signature": "qG/IAIjBUjjKJoJdFjts9p6/OPQbh9Ge5sWyeIIlKZA="
}
```

● 返回参数:

公共响应参数

● 响应示例:

```
{
  "status": 200,
  "message": "SUCCESS"
}
```

3.4.7集成时间戳接口调用

可通过调用时间戳功能保证签名时间准确性

3.4.7.1产生时间戳

- 功能说明: 产生时间戳
- 请求地址: <http://host:port/coss/service/v1/createAndGetTssInfo>
- 请求方式: POST
- 请求参数:

字段	类型	是否必须	说明
oriData	String	是	数据原文
attachCert	String	是	最终产生的时间戳是否带证书 (true/false)

● 请求示例:



```
{  
    "version": "1.0",  
    "appId": "APP_0157E810F6894BEEB4DEC70331720557",  
    "signAlgo": "HMAC",  
    "oriData": "test",  
    "attachCert": "true",  
    "signature": "qG/IAIjBUjjKJoJdFjts9p6/OPQbh9Ge5sWyeIIlKZA="
```

● 返回参数:

字段	类型	是否必须	说明
tsResp	String	是	时间戳

● 响应示例:

```
{  
    "status": 200,  
    "message": "SUCCESS",  
    "data": {  
        "tsResp": "MIIGkAYJKoZIhvcNAQcCoIIIGgTCCBnOCAQMxDzANBgglghkgBZQMEAgEFADCBswYL  
KoZIhvcNAQkQAQSGgaMEgaAwgZ0CAQEGASowMTANBgglghkgBZQMEAgEFAAQgn4bQgYhMfWWaL+qgxVrQFaO/Tx  
srC4Is0VlsFbDwCggCBAX14QEYDzIwMjEwODA5MDgONDM1WgIGAXsqFcgpOEokQTA/MQswCQYDVQQGEwJDTjEN  
MAsGA1UECgwEQkpDQTEPMAOGA1UECwwGMTExMTEwMRwDgYDVQQDDAdyc2ExMDI0oIEBjCCBAIwggLqoAMCAQ  
ICChtAAAAAANodAwDQYJKoZIhvcNAQELBQAuWjELMAkGA1UEBhMCQ04xDTALBgNVBAoMBEJKQOEExGDAWBgNV  
BAsMD1B1YmxyYyBUcnVzdCBDQTEaMBGGA1UEAwRUHVibG1jIFRydXNOIENBLTIwHhcNMjAxMjE1MTYwMDAwWh  
cNMzAxMjE1MTU1OTU5WjA/MQswCQYDVQQGEwJDTjENMAsGA1UECgwEQkpDQTEPMAOGA1UECwwGMTExMTEwMRwD  
gYDVQQDDAdyc2ExMDI0MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQCPTqz8+FGLb1nmX4uNxf13xzq8DM  
SKzuPpjpP6tYe60DHjK2in00H2iDm0gwbxW5tZB7br3H1k7sLMrt4108j40jGy960VtQm4xn+vg650oPbUtwQR  
q5ZcNvwgDHkc8Adf/bs0znUnBqbZ3mrVOE9ZCGRsnrg7f3iy71051Z51uwIDAQABo4IBbzCCAwwHwYDVR0jBB  
gwFoAU+7fUVhdYjCN91fhCadTtd5tX6+kwga0GA1UdHwSBpTCBojBsoGqgaKRmMGQxCzAJBgNVBAYTANOMQ0w  
CwYDVQQKDARCSkNBMRgwFgYDVQQQLDA9QdWJsaWMgVHJlc3QgQ0ExGjA1YBgNVBAMMEVB1YmxyYyBUcnVzdCBDQS  
OyMRAwDgYDVQQDEwdjYTRjcmw0MDKGMKAuhixodHRwOi8vbGRhcC5iamNhLm9yZy5jb3I9jcmwvchrjYS9jYTRj  
cmw0LmNybdAJBgNVHRMEAjaAMBEGCWCsAGG+EIBAQQEAwIA/zAWBgNVHSUBaf8EDDAKBgggrBgEFBQcDCDBABg  
NVHSAEOTA3MDUGCSqBHIbvMgICATAoMCYGCCsGAQUFBwIBFhpodHRwOi8vd3d3LmJqY2Eub3JnLmNuL2NwcZAL  
BgNVHQ8EBAMCA/gwEwYKKoEchu8yAgEBHgQFDAM2NTQwDQYJKoZIhvcNAQELBQADggEBADQn82uvXWHpF1/7fv  
bIMNp05p4yQZwjE80mYE9/ULy7BXP1qmSmaCnxve9RUS8+tTXNDY8w6jc4E5b3Y1MeLsGEVW+DB2mFfv8WHbTK  
/2q8QivIgQLlzyrD56Tljiqri0lr3ShZYiVX5yuOFgulx2DOM1JU/10PoEQatYC6EAdPKJcaoj9C4+zShr/irG
```



```
6GhXodreYUcmazDVLzLHfQaf2k/E3QcA/x7hVwtEMcmmG8N7Zfvprms9UjlvEIofI/AEAB3t+41EyUs0zFqvYO
TvhczIMb76uc4kqCUZbTzJ9d6dw5+EmGmmpXEs6QB2XzcCTo3xvTPedbh0qBHjqLu30xggG1MIIBoQIBATBgMF
Ix CzA JBgNVBAYTAkNOMQ0wCwYDVQQKDARCSkNBMRgwFgYDVQQLEDA9QdWJsaWMgVHJlc3QgQ0ExGjAYBgNVBAMM
EVB1Ym xpyYyBUcnVzdCBDQS0yAgobQAAAAAADdHQMAOGCWCgsAF1AwQCAQUAoIGYMBogCSqGSIb3DQEJAzENBg
sqhkiG9w0BCRABBDACBgkqhkiG9w0BCQUxDxcNMjEwODA5MDg0NDM1WjArBgqhkiG9w0BCRACDDEcMBowGDAW
BBTCdKpud+ZFDHNAbg3RN8rHg49GdTAvgkqhkiG9w0BCQQxIgQgT/zZFRRmr9N9zSQhXDBcZDetmfCJmEblUm
S0zDTgHYkwDQYJKoZIhvcNAQEBBQAEgYA9KB+eTuLTkRw5t2AVYG/Zbo9u0cex+DBruXOBvNk5Rb21fzqhfgUd
UcT/Bd+TT419akq58Hj1WVK6h+zeFkkoe7QvwUMr13YW0j7I9Fm4T8Ct0R8s/AcUfA1LcxQM1L5mqbmGWS913Y
r/malvqTkOUFHxN/AV7vxibXTHJlsIhg=="
```

```
}
```

```
}
```

3.4.7.2 解析时间戳

- 功能说明：解析时间戳
- 请求地址：http://host:port/coss/service/v1/getTSInfo
- 请求方式：POST
- 请求参数：

字段	类型	是否必须	说明
tsResp	String	是	待解析时间戳数据
type	Int	是	解析时间戳类型 (type=1 返回时间, type=2 返回原文的 hash 值, type=3 返回证书, type=4 返回服务器证书序列号)

- 请求示例：

```
{
  "version": "1.0",
  "appId": "APP_0157E810F6894BEEB4DEC70331720557",
  "signAlgo": "HMAC",
  "tsResp": "MIIGkAYJKoZIhvcNAQcCoIIGgTCCBnOCAQMxDzANBg1ghkgBZQMEAgEFADCBswYLKoZ
IhvcNAQkQAQSggaMEgaAwgZOCAQEGASowMTANBg1ghkgBZQMEAgEFAAQgn4bQgYhMfWWaL+qgxVrQFaO/T
xsR4Is0V1sFbDwCggCBAX14QEYDzIwMjEwNzAxMDkwMzIwWgIGAXphTuvJoE0kQTA/MQswCQYDVQQGEwJ
```



DTjENMAsGA1UECgwEQkpDQTEPMAOGA1UECwwGMTEExMTEwMRAdGjYDVQQDDAdyc2ExMDI0oIIIEBjCCBAIwglLqoAMCAQICChAAAAAANOdAwDQYJKoZIhvcNAQELBQAwUjELMAkGA1UEBhMCQ04xDTALBgNVBAoMBEJKQ0ExGDAWBgNVBAsMD1B1YmxpYyBUcnVzdCBDQTEaMBGGA1UEAwRUHVibG1jIFRydXN0IENBLTIwHhcNMjAxMjE1MTYwMDAwWhcNMzAxMjE2MTU1OTU5WjA/MQswCQYDVQQGEwJDTjENMAsGA1UECgwEQkpDQTEPMAOGA1UECwwGMTEExMTEwMRAdGjYDVQQDDAdyc2ExMDI0MIGfMAOGCSqGSIb3DQEBAQUAA4GNADCBiQKBgQCPTqz8+FGLb1nmX4uNxfl3xzq8DMSKzuPpjpP6tYe60DHjK2in00H2iDm0gwbxW5tZB7br3Hlk7sLMrt4108j40jGy960VtQm4xn+vg650oPbUtwQRq5ZcNvwgDHkc8ADf/bs0znUnBqbZ3mrVOE9ZCGRsnrg7f3iy71051Z51uwIDAQABo4IBbzCCAWswHwYDVR0jBBgwFoAU+7fUVhdYjCN91fhCAdTtd5tX6+kwga0GA1UdHwSBpTCBojBsoGqgaKRmMGQxCzAJBgNVBAYTAkNOMQ0wCwYDVQQKDARCSkNBMRgwFgYDVQQLDA9QdWJsaWMgVHJlc3QgQ0ExGjAYBgNVBAMMEVB1YmxpYyBUcnVzdCBDQSOyMRAwDgYDVQQDEwdjYTRjcmw0MDKgMKAuhixodHRwOi8vbGRhcC5iamNhLm9yZy5jb29jcmwvcHRjYS9jYTRjcmw0LmNybdAJBgNVHRMEAjaAMBEGCWGSAGG+EIBAQQEAwIA/zAWBgNVHSUBAf8EDDAKBggrBgEFBQcDCDBABgNVHSAEOTA3MDUGCSqBHIbvMgICATAoMCYGCCsGAQUFBwIBFhpodHRwOi8vd3d3LmJqY2Eub3JnLmNuL2NwczALBgNVHQ8EBAMCA/gwEwYKKoEchu8yAgEBHgQFDAM2NTQwDQYJKoZIhvcNAQELBQADggEBADQn82uvXWHpF1/7fVbIMNp05p4yQZwje80mYE9/ULy7BXPlqmSmaCnxve9RUS8+tTXNDY8w6jc4E5b3YlMeLsGEVW+DB2mFfv8WHbTK/2q8QivIgQLlzyrD56T1j iqri01r3ShZYiVX5yuOFgu1x2DOM1JU/10PoEQatYC6EAdPKJcaoj9C4+zShr/irG6GhXodreYUcmazDVLzLHfQaf2k/E3QcA/x7hVwtEMcmmG8N7Zfvprms9UjlvEIOfi/AEAB3t+41EyUs0zFqvYOTvhczIMb76uc4kqCUZbTzJ9d6dw5+EmGmmpXEs6QB2XzcCTo3xvTPedbh0qBHjqLu30xggG1MIIBOQIBATBgMFIxCzAJBgNVBAYTAkNOMQ0wCwYDVQQKDARCSkNBMRgwFgYDVQQLDA9QdWJsaWMgVHJlc3QgQ0ExGjAYBgNVBAMMEVB1YmxpYyBUcnVzdCBDQSOyAgobQAAAAAADdHQAOGCSqGSIb3DQEJAzENBgsgqhkiG9w0BCRABBDACBgqhkiG9w0BCQUxDxcNMjEwNzAxMDkwMzIwWjArBgsgqhkiG9w0BCRACDDEcMBowGDAWBBTCdKpud+ZFDHNAbg3RN8rHg49GdTAvBgqhkiG9w0BCQXxIgQgXkBUkAkk1V1rJvkfs3fHoUTaKnHSmpbFYTHd3vmra58wDQYJKoZIhvcNAQEBBQAEgYAXYCOio5cHuwIx/VOBMjVJA9uylgHGUmwJ+KGRBMEwqxldrn8zVrPkH1WACpJrA4pv7bdWynrgFHQU2ZC0t5x71t33J4fbmvphqsDecd7h6N3xwxE/n6btQPf6W/KSt+iR78z9CPsTq4IaTDW7R18Nch4YhjamAzCkwWf7MyvI7w==",

"type": "1",

"signature": "qG/IAIjBUjjKJoJdFjts9p6/OPQbh9Ge5sWyeIIlKZA="

}

● 返回参数:

字段	类型	是否必须	说明
tsInfo	String	是	指定类型的解析数据

● 响应示例:

```
{
  "status": 200,
  "message": "SUCCESS",
  "data": {
```



```

    "tsInfo": "20210701170320"
  }
}

```

3.4.7.3 验证时间戳

- 功能说明：验证时间戳
- 请求地址：http://host:port/coss/service/v1/verifyTS
- 请求方式：POST
- 请求参数：

字段	类型	是否必须	说明
oriData	String	是	数据原文
tsResp	String	是	待验签时间戳信息

- 请求示例：

```

{
  "version": "1.0",
  "appId": "APP_0157E810F6894BEEB4DEC70331720557",
  "signAlgo": "HMAC",
  "oriData": "test",

  "tsResp": "MIIGkAYJKoZIhvcNAQcCoIIGgTCCBnOCAQMxDzANBg1ghkgBZQMEAgEFADCBswYLKoZIhvcNAQkQAQSGgaMEgaAwgZOCAQEGASowMTANBg1ghkgBZQMEAgEFAAQgn4bQgYhMfWWaL+qgxVrQFa0/Txs rC4Is0V1sFbDwCggCBAX14QEYDzIwMjEwNzAxMDkwMzIwWgIGAXphTuvJoE0kQTA/MQswCQYDVQQGEwJDT jENMAsgA1UECgwEQkpDQTEPMAOGA1UECwwGMTExMTEwMRAdDgYDVQQDDAdyc2ExMDI0oIIEBjCCBAIwggL qoAMCAQICChtAAAAAAAAANodAwDQYJKoZIhvcNAQELBQAwUjELMAkGA1UEBhMCQ04xDTALBgNVBAoMBEJKQ OExGDAWBgNVBAsMD1B1YmxpYyBUcnVzdCBDQTEaMBGGA1UEAwRUHVibG1jIFRydXNOIENBLTIwHhcNMjAx MjE1MTYwMDAwWhcNMzAxMjE2MTU1OTU5WjA/MQswCQYDVQQGEwJDTjENMAsgA1UECgwEQkpDQTEPMAOGA 1UECwwGMTExMTEwMRAdDgYDVQQDDAdyc2ExMDI0MIGfMAOGCSqGSIb3DQEBAQUAA4GNADCBiQKBgQCPTqz 8+FGLb1nmX4uNxf13xzq8DMSKzuPpjpP6tYe60DHjK2in00H2iDm0gwbxW5tZB7br3H1k7sLMrt4108j40 jGy960VtQm4xn+vq650oPbUtwQRq5ZcNvwgDHkc8ADf/bs0znUnBqbZ3mrVOE9ZCGRsnrg7f3iy7l05lZ5 1uwIDAQABo4IBbzCCAwwHwYDVR0jBBgwFoAU+7fUVhdYjcn91fhCAAdTtd5tX6+kwga0GA1UdHwSBpTCBo jBsoGqgaKRmMGQxCzAJBgNVBAYTAkNOMQowCwYDVQQKDARCSkNBMRgwFgYDVQQQLDA9QdWJsaWMgVHJlc3Q gQOEExGjAYBgNVBAMMEVB1YmxpYyBUcnVzdCBDQ0yMRAdDgYDVQQDEwdjYTRjcmw0MDkGMKAuhixodHRwO i8vbGRhC5iamNhLm9yZy5jb19jcmwvCHRjYS9jYTRjcmw0LmNyYDAJBgNVHRMEAjaAMBEGCWGSAGG+EI BAQQEAWIA/zAWBgNVHSAf8EDDAKBggrBgEFBQcDCDBABgNVHSAEOTA3MDUGCSqBHIbvMgICATAoMCYGC

```



```
CsGAQUFBwIBFhpodHRwOi8vd3d3LmJqY2Eub3JnLmNuL2NwczALBgNVHQ8EBAMCA/gwEwYKKoEchu8yAgE
BHgQFDAM2NTQwDQYJKoZIhvcNAQELBQADggEBADQn82uvXWHpF1/7fvbIMNp05p4yQZwjE80mYE9/ULy7B
XPlqmSmaCnxve9RUS8+tTXNDY8w6jc4E5b3YlMeLsGEVW+DB2mFfv8WHbTK/2q8QivIgQLlzyrD56Tljiq
ri01r3ShZYiVX5yuOFgu1x2DOM1JU/10PoEQatYC6EAdPKJcaoj9C4+zShr/irG6GhXodreYUcmazDVLzL
HfQaf2k/E3QcA/x7hVwtEMcmmG8N7Zfvprms9UjlvEIOfi/AEAB3t+41EyUs0zFqvYOTvhczIMb76uc4kq
CUZbTzJ9d6dw5+EmGmmpXEs6QB2XzcCTo3xvTPedbh0qBHjqLu30xggG1MIIBoQIBATBgMFIxCzAJBgNVB
AYTAkNOMQowCwYDVQQKDARCSkNBMRgwFgYDVQQLDA9QdWJsaWMgVHJ1c3QgQ0ExGjAYBgNVBAMMEVB1Ymx
pYyBUcnVzdCBDQS0yAgobQAAAAAADdHQAOGCWCgsAF1AwQCAQUAoIGYMBogCSqGSIb3DQEJAzENBgsqh
kiG9wOBCRABBDACBgkqhkiG9wOBCQUxDxcNMjEwNzAxMDkwMzIwWjArBgqhkiG9wOBCRACDDEcMBowGDA
WBBTCdKpud+ZFDHNAbg3RN8rHg49GdTAvBgkqhkiG9wOBCQQxIgQgkxkBUkAkk1V1rJvkfs3fHoUTaKnHSm
pbFYTHd3vmra58wDQYJKoZIhvcNAQEBBQAEgYAxYCOio5cHuwIx/VOBMjVJA9uylgHGUmwJ+KGRBMewqx1
drn8zVrPkH1WACpJrA4pv7bdWynrgFHQU2ZC0t5x71t33J4fbmvphqsDecd7h6N3xwxE/n6btQPf6W/KSt
+iR78z9CPsTq4IaTDW7R18Nch4YhjamAzCwkWf7MyvI7w==",
```

"signature": "qG/IAIjBUjjKJoJdFjts9p6/OPQbh9Ge5sWyeIIlKZA="

}

● 返回参数:

字段	类型	是否必须	说明
verifyRes	String	是	验证结果: 1 为时间戳有效, -1 为时间戳验证不通过, -2 为原文验证不通过, -3 为不是所信任的根, -4 证书未生效, -5 查询不到此证书, -6 为签发时间戳时服务器证书过期

● 响应示例:

```
{
  "status": 200,
  "message": "SUCCESS",
  "data": {
    "verifyRes": "1"
  }
}
```

3.4.8PDF 签章相关接口

3.4.8.1计算 PDF 文件 hash 值

- 功能说明：计算待签名的 **PDF** 文件的 **hash** 值
- 请求地址：http://ip:port/coss/service/v1/getPdfHash
- 请求方式：POST
- 请求参数：

字段	类型	是否必须	说明
userId	String	是	用户 ID
signSealType	String	否	签章类型，默认自有 SM2 签章 BJCA_SM2
pdfFileStr	String	是	pdf 文件 base64 数据
configureSealInfo	ConfigureSealInfo	是	签章属性，包括用途，省份，姓名（详细字段见表后说明）
coordOfRuleInfo	CoordOfRuleInfo	否	签名坐标规则，包括签名页数，页数为 0 代表全部（详细字段见表后说明）
keywordsOfRuleInfo	KeywordsOfRuleInfo	否	关键字签名规则（详细字段见表后说明）
signWidth	Float	否	签章字体大小宽
signHeight	Float	否	签章字体大小高
opacityOfEffect	int		透明度
coverTextOfEffect	boolean	否	是否覆盖文字
appName	String	否	应用名称，可通过管理界面中应



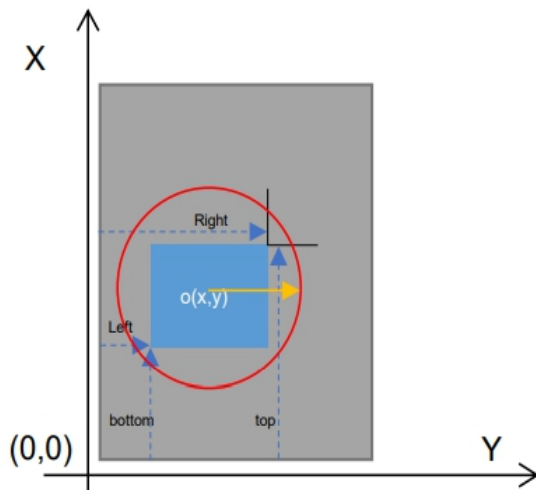
			用管理获得（国密类型或者国办类型的签章该字段必传，并且该应用与服务器证书（制章者证书绑定））
--	--	--	--

ConfigureSealInfo 对象参数如下：

字段	类型	是否必须	说明
signType	String	是	类型，个人签章或企业签章
signProvince	String	是	地理位置
signName	String	是	联系人

CoordOfRuleInfo 对象参数如下：

包含四个值分别是，left，right，top，bottom 值确定的中心点 o(x,y)位置，如图：



字段	类型	是否必须	说明
top	int	否	坐标 top 印章图片顶部区域位置 pt
right	int	否	坐标 right 印章图片右部区域位置 pt
bottom	int	否	坐标 bottom 印章图片底部区域位置 pt



left	int	否	坐标 left 印章图片左部区域位置 pt
pageNo	int	否	签章页码

KeywordsOfRuleInfo 对象参数如下：

字段	类型	是否必须	说明
keyWord	String	否	关键字
signKeywordsPosition	Int	否	关键字查找起始位置： 1---CENTER, 2---BOTTOM, 3---RIGHT 相对位置， 支持 CENTER、RIGHT、BOTTOM
signKeywordsSearch	String	否	关键字查找顺序：0, 1, 2 检索顺序： 0 ALL 全签 1 DESC 倒序查找 2 ASC 正序
keywordNum		否	关键字索引，第几个关键字， （全部关键字需传 1， 且 signKeywordsSearch 传 0）
coordinateX		否	图片偏移量，单位 pt，非负值 距离关键字坐标 X
coordinateY		否	图片偏移量，单位 pt，非负值 距离关键字坐标 Y



● 请求示例:

```
{  
  "appId": "APP_7B3F36A14E99410A80B37AEF332E3247",  
  "signAlgo": "HMACSHA256"  
  "signature": "57oDmdUD9kmGuy1L/M3fxY2cU+We5Cq0WV9XLJCr5A0=",  
  
  "signSealType": "BJCA_SM2",  
  
  "userId": "51627394e66c1f764d6211d5aab4cdc09026a7b450929fc4834b179ad90389f0",  
  
  "pdfFileStr":  
  "JVBERi0xLjUNJeLjz9MNCjMxIDAgb2JqDTw8L0xpbmVhcm16ZWQgMS9MIDEwMzg2Mi9PIDMzLOUgOTgOMTg  
vTiAxL1QgMTAzNTU3L0ggWyA0OgNOBU4BtgFbtCst0Fm7qgQnAVu0q0awGdikXeWCjdolV4zVBu0aJFgPrMP  
xtTi3BlitXR7BKxfCawAlgONQ.....EWzxGRTZCMzcxNkMzMDI3NzQxOTdGMERDQTZDOERFNjAORT48ME  
UONzUzNjI2MUM0OUEOMzg5MTQ1RUVDRTe4RjVGQUI+XS9JbmZvIDMwIDAgUi9MZW5ndGggNTAvUm9vdCAzM  
AwIFIvU2l6ZSAzMS9UeXB1L1hSZWYvV1sxIDMgMV0+PnN0cmVhbQ0KaN5iYgABJkbGhiIGJiCDCUQKmlBIxj  
4wGQCU/Z9TDWIDeQNPMr0CkYwMAAEGAJwuBUUNCmVuZHN0cmVhbQ11bmRvYmoNc3RhcnR4cmVmDQoxMTYNCi  
U1RU9GDQo=",  
  
  "configureSealInfo": {  
    "signType": "个人签章",  
    "signProvince": "北京",  
    "signName": "张浩"  
  },  
  
  "coordOfRuleInfo": {  
    "top": 300,  
    "right": 600,  
    "bottom": 200,  
    "left": 350,  
    "pageNo": 1
```



```
},  
  
"appName": "协同签名 APPID",  
  
"signWidth": 150,  
  
"signHeight": 120,  
  
"opacityOfEffect": 40,  
  
"coverTextOfEffect": false,  
  
"version": "2.0.7"  
  
}
```

● 返回参数:

字段	类型	是否必须	说明
pdfBytes	String	是	处理后的 PDF 数据 Base64 编码，不能直接打开，服务端需要自己存储，后面合章的时候需要再使用
hashBytes	String	是	签名原文的杂凑值 Base64 编码数据
hashSetBytes	String	是	P7 结构体 Base64 编码数据，第三方和客户端直接使用此属性值作为原文进行签名，后面合章的时候需要再使用
hashSetHashBytes	String	是	P7 结构体 Base64 编码数据，第三方和客户端可直接使用此属性值作为原文 HASH 进行签名
rectangleBean	RectangleBean	是	回调图片坐标信息单位（详细字段见表后说明）

RectangleBean 参数说明:

字段	类型	是否必须	说明
top	int	否	坐标 top 印章图片顶部区域位置 pt



right	int	否	坐标 right 印章图片右部区域位置 pt
bottom	int	否	坐标 bottom 印章图片底部区域位置 pt
left	int	否	坐标 left 印章图片左部区域位置 pt
pageNo	int	否	签章页码

● 响应示例：

```
{
  "status": 200,
  "message": "SUCCESS",
  "data": {
    "hashSetBytes": "AhxG1631xWk6pF/J45KPZWN4FeyL2h5vmWoW49aFKmE=",
    "hashSetHashBytes": "AhxG1631xWk6pF/J45KPZWN4FeyL2h5vmWoW49aFKmE=",
    "hashBytes": "PAhxG1631xWk6pF/J45KPZWN4FeyL2h5vmWoW49aFKmE=",
    "pdfBytes":
      "JVBERi0xLjUNJeLjz9MNCjMxIDAgb2JqDTw8L0xpbmVhcm16ZWQgMS9MIDEwMzg2Mi9PIDMz. icY2BgYPj/n5
      HZsoSBgZHZQhBEemYzMDAxM. ....HgzsDIyTl8EEjFVB5HG6b6HizGCShZF6naQuBkziDQXhMoC2V5pUDYjmA
      QSIFsA2fwM2gplbmRzdHJlYW0KZW5kb2JqCiVFU1MtUERGCnNOYXJ0eHJlZgoyMTYxMzIKJSVFT0YK",
    "rectangleBean": {
      "left": 430.0,
      "top": 306.16,
      "right": 520.0,
      "bottom": 193.84,
      "pageNo": 1,
      "width": 90.0,
      "height": 112.32001
    }
  }
}
```

3.4.8.2 合并 PDF 签章

- 功能说明：用于合并 pdf 签章
- 请求地址：http://host:port/coss/service/v1/composePdfSeal
- 请求方式：POST
- 请求参数：

字段	类型	是否必须	说明
userId	String	是	用户id
signSealType	String	否	签章类型，默认自有SM2签章BJCA_SM2
signValue	String	是	第三方签名值
pdfBytesStr	String	是	处理后的 PDF 数据Base64编码
hashSetBytesStr	String	是	P7 结构体Base64编码数据
tssSignValue	String	否	时间戳签名值

- 请求示例：

```
{  
  "appId": "APP_7B3F36A14E99410A80B37AEF332E3247",  
  "version": "2.0.7",  
  "signAlgo": "HMACSHA256",  
  "userId": "51627394e66c1f764d6211d5aab4cdc09026a7b450929fc4834b179ad90389f0",  
  "signSealType": "BJCA_SM2",  
  "signValue":  
    "MEUCIQDDqtkp0PVodB4KxUEp2SKBePxeVWjimTvrCWB9CNrCgQIgPyQAJwyDrvsDo/v/jW4TqsDG4e3uyczCQ  
    hbF+HIs8vA=",  
  "pdfBytesStr":
```



```
"JVBERi0xLjUNJeLjz9MNCjMxIDAgb2JqDTw8L0xpbmVhcm16ZWQgMS9MIDEwMzg2Mi9PIMDEwMTADicY2BgYPj/n5HZsoKBgZHZ.....QhREeuYzMDAxMHgzSDIyTl8EEjG3BJGm/6HiLGCSmZFx6naQuM1HEGn8ESoLZHt1QdmMYBJIgGwBAC1fDtoKZW5kc3RyZWFTcmVuZG9iagolRVNTLVBERgpzdGFydHhyZWYKMjE2MTM4CiU1RU9GCg=="
```

"hashSetBytesStr":

```
"MIIRCQIBAJCCDZAwwgkXMAOWAkVTAgeBFgRCSkNBFiRmNWM1MTR1YS1kNmZmLTQxZTk0OGQ5My11ZmY5YmVmZTgxOTkwggRgAgEDDA5HbeeUteWtkOWNsOeroDCCBBwEggQYMIIEFDC. c5YWQ5MDM4OWY.....wMAoGCCqBHM9VAYN1A0gAMEUCIQD/mDGS59x7Pj92FCEyCafQkW+fIuhhW6jKIoOzCkLGkAIgWZSLol7Tgf4TtrIN7LuocVFcEFERrD3syi/Ha2yrtwIGCCqBHM9VAYN1",
```

"signature": "W3laUwd9+mwqZuBawh547B+jqUp6xeD7AbSYmSqqlIY="

}

● 返回参数:

字段	类型	是否必须	说明
data	String	是	合成的 pdf 文件 base64 数据

● 响应示例:

```
{
  "status": 200,
  "message": "SUCCESS",
  "data":
    "JVBERi0xLjUNJeLjz9MNCjMxIDAgb2JqDTw8L0xpbmVhcm16ZWQgMS9MIDEwMzg2Mi9PIMDEwMTADicY2BgYPj/n5HZsoKBgZHZ.....QhREeuYzMDAxMHgzSDIyTl8EEjG3BJGm/6HiLGCSmZFx6naQuM1HEGn8ESoLZHt1QdmMYBJIgGwBAC1fDtoKZW5kc3RyZWFTcmVuZG9iagolRVNTLVBERgpzdGFydHhyZWYKMjE2MTM4CiU1RU9GCg=="
}
```



3.4.8.3 验证 PDF 签章文件

- 功能说明：用于验证 PDF 签章文件
- 请求地址：http://host:port/coss/service/v1/verifyPdfSeal
- 请求方式：POST
- 请求参数：

字段	类型	是否必须	说明
pdfFileStr	String	是	待验证 pdf 签章文件 base64 数据
signSealType	String	否	签章类型，默认自有 SM2 签章 BJCA_SM2
certChainStr	String	否	base64 的证书链数据

- 请求示例：

```
{  
  
  "appId": "APP_7B3F36A14E99410A80B37AEF332E3247",  
  
  "signature": "erscws535sfdgfdfsdftgsfsv",  
  
  "userId": "51627394e66c1f764d6211d5aab4cdc09026a7b450929fc4834b179ad90389f0",  
  
  "signSealType": "BJCA_SM2",  
  
  "pdfFileStr":  
  "JVBERi0xLjUNJeLjz9MNCjMxIDAgb2JqDTw8L0xpbmVhcm16ZWQgMS9MIDEwMzg2Mi9PI/n5HZsoKbgZHZQhR  
EeuYzMDAxMHgzs.....m/6HiLGCSmZFx6naQuMlHEGn8ESoLZHt1QdmMYBJIgGwBAC1fDtoKZW5kc3RyZWFTc  
mVuZG9iagolRVNTLVBERgpzdGFydHhyZWYKMjE2MTM4CiU1RU9GCg=="  
}
```

- 返回参数：

字段	类型	是否必须	说明
statusCode	String	是	200 统一验证结果编码，成功返



			回 200
message	List<Object>	是	详细信息列表

● Message 详细信息如下:

字段	类型	是否必须	说明
statusCode	String	是	验证结果编码, 成功返回 200
fieldName	String	否	签名区域属性名称
errInfos	String []	否	错误信息
signAlg	String	是	签名算法
sigValue	String	否	签名值
signDate	String	否	
signCerts	String []	否	签名证书
commonName	String	否	证书通用属性名称
certDNs	String []	否	证书 DN 项
certSN	String	否	证书序列号
extendsData	String	否	扩展项
isTss	boolean	否	是否时间戳
timeStampDate	String	否	时间戳签名时间戳
pageNo	Integer	是	页码
sealImg	String	否	签章图片

● 响应示例:

```

{
  "status": 200,
  "message": "SUCCESS",
  "data": {
    "statusCode": "200",
    "message": [{
      "statusCode": "200",
      "signDate": "#030E00323131313131313135323732385A",
      "signCerts": [
        "MIIDFzCCAr2gAwIBAgIKGhAAAAAAAwFazAKBgqgRzPVQGDdTBEMQswCQYDVQQGEwJDTjENMAsGA1UECgWEQkpDQTENMAsGA1UECwwEQkpDQTEXMBUGA1UEAwwOQmVpamluZyBTTTIgQ0EwHhcNMjExMDI5MTU1OTU5WjA1MQswCQYDVQQGEwJDTjESMBAGA1UECgwJ6ZmI5pmT5pmoMRIwEAYDVQQDDAnpmYjmmZPmmagwWTATBgqhkJOPQIBBgqgRzPVQGCLQNCAASF6YON6jwa/qE00AVrMYKY7QP8d75MTQ9J0j1SdweZGE8v8YgBb3HuRlfz+GM9zslC+NKk0avSrpcor+c0ZnKTo4IBpDCCAAAwHwYDVR0jBBgwFoAUH+bP1I/FliqXSimKFecWyZIOxLYwHQYDVR00BBYEFKYwyx9Mx5cCkXAYlafpoQGS0GBBMIGiBgNVHR8EgZowgZcwYKBeoFykWjBYMQswCQYDVQQGEwJDTjENMAsGA1UECgwEQkpDQTEXMBUGA1UEAwwOQmVpamluZyBTTTIgQ0ExEjAQBgNVBAMTCWNhMjFjcmywOTAzODGgLA4YtaHR0cDovLzExMS4yMDcuMTc3LjE4OTQ0MDAzL2Nybc9jYTIxY3JsmJkuY3JsmB8GCiqBHIbvMgIBAQA4EEQwPOTk4MDAwMTAwNTI4MTM3MCAGCGCGSAGG+EQCBBRTRjYzMjMyMzE5MDYwNTI2ODkyNDAUBgoqgRyG7zICAQEeBAYMBDI4NTQwDgYDVR0PAQH/BAQDAgbAMFAGCiqBHIbvMgIBCGIEQgxANTE2MjczOTR1NjZjMwY3NjRkNjI1MWQ1YWFingNkYzA5MDI2YTdiNDUwOTI5ZmMOODMOYjE3OWFkOTAzODlmMDAKBgqgRzPVQGDdQNIADBFAiEA/5gxkufcez4/dhQhMgmn0JFvnyLoYVuoYiKDswpCxpACIFmUi6Je04H+E7ayDey7qHFRXBBREaw97Movx2tsq7cC"
      ],
      "certDNs": [
        "C=CN, O=陈晓晨, CN=陈晓晨"
      ],
      "signAlg": "SM3WithSM2",
      "certSN": "1A1000000000000C056B",
      "extendsData": ""
    ]
  }
}

```

"pageNo": 1,

```
"commonName": "陈晓晨",
```

"sealImg":

[illegible]

}]

}

}

3.4.9 辅助接口

3.4.9.1 计算指定长度的随机数

- **功能说明：**用于获取随机数
- **请求地址：**`http://host:port/coss/service/v1/genRandom`



- 请求方式：POST

- 请求参数：

字段	类型	是否必须	说明
len	Int	是	随机数长度

- 请求示例：

```
{  
  "appId": "APP_7B3F36A14E99410A80B37AEF332E3247",  
  "len": 16,  
  "version": "1.0",  
  "signAlgo": "HMACSHA256",  
  "signature": "KR2jqHdCemY0V/810f0vBp+j0f/3V4ZN4Rwj2zyQbZs="
```

- 返回参数：

公共响应参数

- 响应示例：

```
{  
  "status": 200,  
  "message": "yFYL+qmo6uVUuh9fEG5x+Q=="
```

3.4.9.2 根据算法类型计算原文数据的 hash 值

- 功能说明：用于计算原文数据的 hash 值
- 请求地址：http://host:port/coss/service/v1/getHashData
- 请求方式：POST
- 请求参数：

字段	类型	是否必须	说明
----	----	------	----

hashAlgo	String	是	Hash 算法类型 SHA1 SHA256 SM3
inData	String	是	Base64 格式的原文数据
containPubKey	String	否	摘要算法为 sm3 的情况选择公钥是否参与运算
cert	String	否	在摘要算法为 sm3 的情况并且选择公钥参与运算时，该参数必传

● 请求示例：

```
{
  "appId": "APP_7B3F36A14E99410A80B37AEF332E3247",
  "version": "1.0",
  "hashAlgo": "SM3",
  "inData": "6L+Z5piv5LiA5Liq5aSa57605Li95Y+I6YGX5oa+55qE5LiW55WM",
  "containPubKey": "true",
  "cert": "MIIDFzCCAr2gAwIBAgIKGhAAAAAAAAAwFazAKBggqgRzPVQGDdTbEMQswCQYDVQQGEwJD
TjENMAsGA1UECgwEQkpDQTENMAsGA1UECwwEQkpDQTEXMBUGA1UEAwwOQmVpamluZyBTTTIGQ0EwHhcNMj
ExMDI4MTYwMDAwWhcNMjIxMDI5MTU1OTU5WjA1MQswCQYDVQQGEwJDTjESMBAGA1UECgwJ6ZmI5pmT5pmo
MRIwEAYDVQQDDAnpmYjmmZPmmagwWTATBgqhkhjOPQIBBggqgRzPVQGCLQNCAASF6YON6jwa/qE00AVrMY
KY7QP8d75MTQ9J0j1SdweZGE8v8YgBb3HuRLfz+GM9zslC+NKk0avSrpCor+c0ZnKTo4IBpDCCAaAwHwYD
VR0jBBgwFoAUH+bP1I/FIiqXSimKFecWyZIOxLYwHQYDVR00BBYEFKYwyx9Mx5cCkXAY1afpoQGS0GBBMI
GiBgNVHR8EgZowgZcwYKBeoFykWjBYMQswCQYDVQQGEwJDTjENMAsGA1UECgwEQkpDQTENMAsGA1UECwwE
QkpDQTEXMBUGA1UEAwwOQmVpamluZyBTTTIGQ0ExEjAQBgNVBAMTCWNhMjFjcmywOTAzoDGg4YtaHR0cD
ovLzExMS4yMDcuMTc3LjE4OTQ0MDAzL2Nybc9jYTIxY3JsMjkuY3JsMB8GCiqBHIbvMgIBAQ4EEQwPOTk4
MDAwMTAwNTI4MTM3MCAGCGCGSAGG+EQCBBRTRjYzMjMyMzE5MDYwNTI2ODkyNDAUBgoqgRyG7zICAQEeBA
YMBDI4NTQwDgYDVROPAQH/BAQDAgBAMFAGCiqBHIbvMgIBCGIEQgxANTE2MjcZOTR1NjZjMWY3NjRkNjIx
MWQ1YWFfINGNkYzA5MDI2YTdiNDUwOTI5ZmMOODMOYjE3OWFkOTAzODlmMDAKBggqgRzPVQGDdQNIADBFAi
EA/5gxkufceZ4/dhQhMgmn0JFvnyLoYVuoyiKDswpCxpACIFmUi6Je04H+E7ayDey7qHFRXBBREaw97Mov
x2tsq7cC",
  "signAlgo": "HMACSHA256",
```



```
"signature": "3MuNd1CNj1VNBA1vu7csydWXAzs0T4oQUHmxahhfkOE="
}
```

● 返回参数:

公共响应参数

● 响应示例:

```
{
  "status": 200,
  "message": "SUCCESS",
  "data": "4FqCsS14q114ai23F9qi9/CnMA4+M50D5EzCtd7rx0="
}
```

3.4.9.3 验签接口（原文验签，hash 验签，web_seal 验签）

- 功能说明: 用于接口验签
- 请求地址: <http://host:port/coss/service/v1/verifySignData>
- 请求方式: POST
- 请求参数:

字段	类型	是否必须	说明
inDataType	String	是	数据类型 DATA HASH WEB_SEAL
inData	String	是	Base64 格式的原文数据
signAlg	String	是	签名算法: SHA1withRSA SHA256withRSA SM3withSM2
signValue	String	是	签名值
cert	String	是	签名证书

● 请求示例:

```
{
  "appId": "APP_7B3F36A14E99410A80B37AEF332E3247",

```

```

    "inDataType": "DATA",
    "inData": "MIIN5QIBBDCCDZQwggkgMAOWAkVTAgeEEFgRCSkNBFiRjNTh1MzRjYi02OWIzLT
RkNmMtYjAxNi0zZTE4ZmkqleXQ.....RQIgFD1imoI2WNb1j3VGSmSVBdCzAkTFgAOoQv28glAG6vUCIQ
CWjS9Q8HydsTTuyE6MhMThwVcFEXrjvhzNh1LmPG7OnBgPMjAyMTExMDgwNjUzMTZaAyEAPJim3J2VM2tS
huva6X1BCVz3v29PbHnIZ3BtCfHk5vcWFDAsMTAOMzc1LDIwOTYOMSw2ODQx",
    "signAlg": "SM3withSM2",
    "signValue": "MEQCIBGPAAfDH3PEoiBB6uOSsXngmenKun19ihdq9N+GCTPfAiAhc1AaWYOhW
rhL8G/QG5ZRuE/p38jkEseU2jl3RvP8g==",
    "cert": "MIIDFzCCAr2gAwIBAgIKGhAAAAAAAAAwFazAKBggqgRzPVQGDDTBEMQswCQYDVQ
QGEwJDTjENMAsGA1UECgwEQkpDQTENMAsGA1UECwwEQkpDQTEXMBUGA1UEAwwOQmVpamluZyBTTTIgQ0Ew
HhcNMjExMDI4MTYwMDAwWhcNMjExMDI5MTU1OTU5WjA1MQswCQYDVQQGEwJDTjESMBAGA1UECgwJ6ZmI5p
mT5pmoMRIwEAYDVQQDDAnpmYjmmZPmmagwWTATBgqhkhjOPQIBggqgRzPVQGCLNCAASF6YON6jwa/qEO
OAVrMYKY7QP8d75MTQ9J0j1SdweZGE8v8YgBb3HuRLfz+GM9zs1C+NKk0avSrpcor+c0ZnKT04IBpDCCAA
AwHwYDVR0jBBgwFoAUH+bP1I/FliqXSimKFecWyZIOxLYwHQYDVR00BBYEFKYwyx9Mx5cCkXAY1afpoQGS
OGBBMIGiBgNVHR8EgZowgZcwYKBeoFykWjBYMQswCQYDVQQGEwJDTjENMAsGA1UECgwEQkpDQTENMAsGA1
UECwwEQkpDQTEXMBUGA1UEAwwOQmVpamluZyBTTTIgQ0ExEjAQBgNVBAMTCWNhMjFjcmywOTAzODGgL4Yt
aHR0cDovLzExMS4yMDcuMjc3LjE4OT04MDAzL2Nybc9jYTIxY3JsMjkuY3JsMB8GCiqBHIbvMgIBAQ4EEQ
wPOTk4MDAwMTAwNTI4MTM3MCAGCGCGSAGG+EQCBBRTRjYzMjMyMzE5MDYwNTI2ODkyNDAUBgoqgRyG7zIC
AQEeBAYMBDI4NTQwDgYDVR0PAQH/BAQDAgbAMFAGCiqBHIbvMgIBCGIEQgxANTE2MjczOTR1NjZjMwY3Nj
RkNjIjEwMQ1YWFingNkYzA5MDI2YTdiNDUwOTI5ZmMOODMOYjE3OWFkOTAzODlmMDAKBggqgRzPVQGDDQNI
ADBFAiEA/5gxkufcez4/dhQhMgmn0JFvnyLoYVuoyiKDswpCxpACIFmUi6Je04H+E7ayDey7qHFRXBBREa
w97Movx2tsq7cC",
    "version": "1.0",
    "signAlgo": "HMACSHA256",
    "signature": "UlwZ+wqbuIQZVN3Giq5208V6/Y3+IisiiTCUPAdMkY="
}

```

● 返回参数:

公共响应参数

● 响应示例:

```
{
```



```
"status": 200,  
"message": "SUCCESS",  
"data": true  
}
```

4 错误码说明

错误码	说明	备注
200	成功	
89000001	请求参数错误	
89000002	操作数据库异常	
89000003	接入方 ID 不存在	
89000004	获得接入方公钥错误	
89000005	验证请求签名错误	
89000006	操作成功，无结果集	
89000007	服务端异常	
89000008	证件号码为空	
89000009	证件号码脱密处理失败	
89001006	证书密钥 id 与协同签名密钥 id 不匹配	
89001008	请求参数-证书格式错误	
89001010	签证书失败	
89001013	公钥与协同签名公钥不匹配	
89001014	SOURCE-请求参数错误	
89001015	身份证号格式错误	
89001016	p10 生成失败	
89001017	签证书模式错误	
89001018	证书状态为空	
89001020	证书导入失败	



89001021	读取文件流转化 base64 错误	
89001024	证书 id 不存在	
89001025	证书为空	
89001026	更新用户证书表信息失败	
89001027	数据格式不正确	
89001033	获取 P10 中公钥失败	
89001034	云端签发证书过程出现异常	
89001035	离线签发证书过程出现异常	
89002001	当前 KEK 不存在	
89002002	EKEK 不存在	
89002008	kekId 不存在	
89002009	数据库中服务端私钥因子不存在	
89002010	数据库中公钥不存在	
89002011	解析证书失败	
89002012	证书中公钥与数据库公钥不匹配	
89002015	生成服务端私钥因子失败	
89002016	生成服务端公钥因子失败	
89002017	生成公钥对象失败	
89002019	服务端签名错误	
89002020	证书已过期	
89002021	查询密钥信息失败	
89002022	查询系统密钥配置信息失败	
89002023	服务端私钥因子为空	
89002024	系统配置错误	
89002025	密钥运算出现错误	
89002027	没有对应的算法错误	
89002031	对称解密失败	
89002032	服务端签名验签过程出现异常	
89002033	KEK 对加解密钥加密过程出现异常	
89003001	签名数据超出允许的长度	



89003002	签名任务 ID 错误	
89003003	签名任务未签署	
89003004	请传入 base64 待签数据	
89003005	查询证书结果为空	
89003006	用户不存在	
89003007	印章图片不存在	
89003008	签名任务过期	
89003009	该任务已签	
89003010	当前设备未激活	
89003011	计算 hash 失败	
89003012	密钥模块异常，请联系管理员	
89003013	查询印章图片失败	
89003014	签名失败	
89003015	用户表不存在该用户	
89003016	日志录入失败	
89003017	证书文件错误	
89003018	证书和算法不匹配	
89003019	验签失败	
89003020	参数非 BASE64 编码	
89003021	证书解析错误	
89003022	查询 SM2 密钥 ID 结果为空	
89003023	生成二维码失败	
89003024	已经开启自动签	
89003026	证书已吊销	
89003027	找不到可信的根证书	
89003028	请为根证书配置 crl 文件	
89003029	CRL 验证失败	
89003030	校证书有效性失败	
89003031	无效证书, 该证书已过期	
89003032	过期时间超出证书有效期范围外	



89003033	获取 Centos 版本失败	
89003034	获取用户 id 失败	
89003035	证书未生效	
89003036	签名图片超出允许的长度	
89003037	用户姓名前后不允许有空格	
89004001	已存在该证件号的用户	
89004002	用户已存在	
89004003	用户不存在	
89004004	日期格式错误	
89004005	添加用户错误	
89004008	获取激活码错误	
89004010	身份证格式错误	
89004011	手机号格式错误	
89004012	用户状态码错误 (ACTIVE, LOCKED, DELETE)	
89004014	起止日期不能超过一年	
89004015	查询应用信息错误	
89004019	用户唯一 ID 重复	
89004020	不存在该用户唯一 ID 的用户	
89004022	不存在该用户 ID 的用户	
89004023	不存在该证件信息的用户, 请重新输入查询条件	
89004025	自动签名失败	
89004026	开启自动签名失败	
89004027	查询证书链信息失败	
89004028	拉取 crl 文件失败	
89004029	crl 内容格式错误	
89004030	证书链验证失败	
89004031	自动签名证书未导入	
89004032	服务配置项格式错误	
89004033	更新服务配置信息错误	



89004034	自动签配置数据错误, 自动签配置数据 (true, false)	
89004035	证书签发模式数据错误, 证书签发模式数据 (云端, 离线, 代理)	
89004036	查询服务配置项信息失败	
89004037	查询业务日志失败	
89004038	查询签名任务数据失败	
89004039	查询日志统计信息失败	
89004040	更新日志统计信息失败	
89004041	用户登录失败, 证书过期不可更新	
89004042	用户登录失败, 证书过期可更新	
89004043	数据库服务检测失败	
89004044	查询客户应用初始基本信息失败	
89004045	查询服务基本信息服务唯一 ID 失败	
89004046	查询服务基本信息服务地址失败	
89004047	查询自动签配置失败	
89004048	查询证书签发模式配置失败	
89004049	查询证书签名值配置失败	
89004050	查询签发证书应用 ID 配置失败	
89004051	查询签发证书网关设备 ID 配置失败	
89004052	查询 Api 网关访问地址配置错误	
89004053	查询业务日志数据归档时间配置失败	
89004054	查询签名数据归档时间配置失败	
89004055	生成 KEK 失败	
89004056	解密 KEK 失败	
89004057	计算签名值失败	
89004058	读取 license 文件失败	
89004059	读取 license 文件信息为空	
89004060	获取 license 信息失败	
89004062	未查询到该应用 id	



89004063	调用 API 网关测试接口失败	
89004064	调用 API 网关测试接口连接失败(4011)	
89004065	查询 API 网关测试接口配置为空	
89004066	服务地址参数错误，无法连接服务器	
89004067	授权信息错误，无法通过授权(4011)	
89004069	证书签发方式不是在线	
89004071	自动签未开启或已过期	
89004072	自动签时间已设置，无需重新开启	
89005001	激活码无效	
89005002	激活码已经过期，请重新生成	
89005003	用户信息不存在	
89005004	用户已被注销，请联系管理员	
89005005	用户已被冻结，请联系管理员	
89005006	用户设备信息有误，请重新下载设备证书	
89005007	用户设备已被注销，请重新下载设备证书	
89005008	用户设备尚未激活，请重新下载设备证书	
89005009	用户 token 过期	
89005010	PIN 口令错误	
89005011	PIN 口令错误次数过多，已被冻结	
89005012	用户设备不具备所需证书，请重新下载设备证书	
89005013	签名过程出现错误	
89005014	密钥 ID 为空	
89005015	挑战 PIN 查询错误	
89005016	开启自动签日志查询失败	
89005017	申请托管密钥失败	
89005018	时间日期格式错误	
89005019	生成 P10 失败	
89005020	生成 hash 失败	
89005021	未开启自动签	



89005022/89005023	设备状态有误	
89005024	该任务已签	
89005025	该任务已取消	
89005026	生成托管证书失败	
89005027	签名图片超出允许的长度（1M）	
89005028	数据转换失败	
89005029	未查询到签章图片	
89005030	用户身份认证无效	
89005031	用户身份认证已失效	
89005032	授权人身份信息错误	
89005099	身份验证不一致	

5 常见 FAQ

1. 关于协同签名中的各种时效问题

- 激活码时效：默认 10 分钟；
- 签名任务时效：单位分钟，默认时效是 1440 分钟，支持传入的最大时效是 4320 分钟；
- 开启自动签时效：单位秒，支持传入的最大时效是 86400 秒，该时效就是自动签的过期时间；
- APP 登录 token 时效：默认 30 分钟。

2. 验证请求签名错误问题

- 检查参数排序问题：是否按照 ASCII 码从小到大排序；
- 检查参数编码问题（只支持 UTF8 编码）；
- 检查参与运算的参数是否有空字符串；
- 检查请求参数中是否有空字符串。

3. 两个产生激活码接口的区别问题

- 3.3.2 产生激活码接口是通过用户 ID 产生激活码，没有校验用户信息；
- 3.3.3 产生激活码（使用姓名+身份证号核实身份）通过用户姓名身份证号产生激活码，并



校验用户信息。

4. 获取签名结果调用频率问题

- 如果采用轮询方式获取签名结果，轮询频率最好限制在 **1-2s** 每次。