

联动平台三方接入协议

接口地址: 第三方提供接口地址

请求方式: POST

请求数据类型: application/x-www-form-urlencoded, application/json

响应数据类型: */*

接口描述: 请求和响应报文参数均需要使用国密4进行加解密, 涉及的appKey、appSecret、privateKey管理员提供

请求示例:

```
//国密m4加密后密文 也就是三方接口接收到的数据
{
  "bizContent":
  "NI4ry+D0lkDXrNf5OcmAzMGaB0td9kXfa321xmIS3qw5homFdZT4xaJu8Vqid37NDivdqj1ADzd0+v+QDKNovFSB
kmOyOQt2ObJ4HW43i0dKKHAKlI3FtZA0F9URhlqXbckbpGMDlhev3XS/b9uxRw+QLOnTzhuzgoxpiOXNA9rmQs3V7
prpF9Wpaetts2pv",
  "reqTimestamp": 0,
  "sign": "签名方式见文末加密算法"
}

//国密m4加密前 也就是三方服务解密后的数据
{
  "bizContent": {
    "idNumber": "",
    "name": "",
    "phone": "",
    "qrcode": "",
    "sn": "",
    "verifyType": "",
    "taskId": "",
    "value": "",
    "action": "",
    "photo": "",
    "extInfo": {
      "idcardContent": "",
      "passRes": "",
      "smileParams": {
        "vid": "",
        "code": 1000,
        "isvResult": true,
        "imageData": "",
        "subCode": "",
        "ftoken": "",
        "subMessage": "",
        "avatarBase64": "",
        "extInfo": {
          "authCode": "",
          "checkWay": "",
```

```
        "resultFromRemote": ""
    }
}
},
"reqTimestamp": 0,
"sign": "签名方式见文末加密算法"
}
```

请求参数:

参数名称	参数说明	请求类型	是否必须	数据类型	schema
联动平台三方查询加密内容	国密m4加密	body	true	国密m4加密	国密m4加密
bizContent	加密后的内容		true	String	国密m4加密后的内容
idNumber	身份号码		false	string	国密m4解密后的内容
name	姓名		false	string	国密m4解密后的内容
phone	手机号		false	string	国密m4解密后的内容
qrcode	健康码barCode		false	string	国密m4解密后的内容
sn	设备编码		true	string	国密m4解密后的内容
verifyType	验证方式 QRCODE:二维码 IDCARD:身份证 FACE:刷脸认证 PASSPORT: 护照认证 INPUT:手动输入		true	string	国密m4解密后的内容
taskId	任务id (连续任务时必传)		false	String	国密m4解密后的内容
value	用户选择结果 (连续任务时必传)		false	String	国密m4解密后的内容
photo	图片的base64		false	String	国密m4解密后的内容
action	检查模式 预检查: check 异步核销: verify 检验并核销: checkVerify		false	String	国密m4解密后的内容
extInfo	支付宝刷脸用户的原始数据		false	Object	国密m4解密后的内容
reqTimestamp	时间戳毫秒 1666881956513		true	integer(int64)	
sign	签名md5(appKey+appSecret+timestamp) 小写32位		true	string	

响应状态:

状态码	说明	schema
200	OK	返回类«第三方接口sm4返回»
异常码		除200以外的code码均为异常码

响应参数:

参数名称	参数说明	是否必须	status状态	类型
code	状态码	true	公共参数	integer(int32)

data	对象	true	公共参数	第三方接口sm4返回
bizContent	加密前的内容	true	公共参数	第三方接口返回
delay	继续的请求延时多少毫秒发起 默认： 0	false	公共参数	integer
isOpen	开门信号 0:开门 1:关门	true	公共参数	integer
end	是否还要继续请求 默认： true	false	公共参数	boolean
openTimes	次数	true	公共参数	integer
openVoiceMsg	语音播报信息	true	公共参数	string
showPageType	终端显示页面类型 common 默认普通 / scenic 景区	true	公共参数	string
aBackgroundUrl	背景图地址	false	closing	string
signalNo	二进制码	false	公共参数 signalType为RS232时必须	string
signalType	开门方式 default / RS232	true	公共参数 和signalNo配合使用	string
name	用户姓名（返回二维码/手输证件号查出的姓名让设备做1:1）	false	扫码，输证件号需要一比一的时必须	string
idNumber	用户身份证号（返回二维码查出的身份证号让设备做1:1）	false	扫码，输证件号需要一比一的时必须	string
photo	用户人像照片（可选择性返回）	false	扫码，输证件号需要一比一的时可选	string
status	显示类型 过渡页/loading 单选页/pending 多选页/pendingMultiple 多选结果页/closingPendingMultiple 结果页/closing 结果页-大图版/closingMaxPicture	true	公共参数	string
taskId	任务id	true	公共参数	string
aLineOneLabel	左-标题	true	closing	string
aLineOneValue	左-内容	true	closing	string
aLineThrLabel	右一标题	true	closing	string
aLineThrValue	右一内容	true	closing	string
aLineTwoValue	左二内容	true	closing	string
aLineTwoLabel	左二标题	true	closing	string
aMainIcon	大图标 1-成功 2-警告 3-失败	true	closing	integer
aMainLine	大图标显示内容	true	closing	string
aSubLineFirst	中间内容一	true	closing	string
aSubLineSecond	中间内容二	true	closing	string
advertise	图片地址	false	closing	string
bChoiseList	选择列表	true	pending	array
deep	是否深色	true	pending	string
label	显示内容	true	pending	string
value	选项值	true	pending	string
bBackgroundUrl	小背景图地址	fasle	pending	string
bMainIcon	小图标 1-成功 2-警告 3-失败	true	pending	integer
bSubLine	小图标下方显示内容	true	pending	string
bMainLine	小图标显示内容	true	pending	string

cBackgroundUrl	背景图地址	false	pendingMultiple	string
cButtonText	确定按钮文字	true	pendingMultiple	
cButtonSubText	选择结果辅助文字	true	pendingMultiple	
cChoiseList	多选列表	true	pendingMultiple	array
list	选择项	true	pendingMultiple	array
label	显示内容	true	pendingMultiple	string
selected	是否选中	true	pendingMultiple	string
value	选项值	true	pendingMultiple	string
topTitle	顶部标题	false	pendingMultiple	
bottomLeftTitle	底部左标题	false	pendingMultiple	
bottomRightTitle	底部右标题	false	pendingMultiple	
subTitle	副标题	true	pendingMultiple	string
title	标题	true	pendingMultiple	string
dBackgroundUrl	背景图地址	false	loading	string
dMainLine	处理中... (0/1)	true	loading	string
mainLineLeft	左副框上层	true	旧版UI模式	string
mainLine	顶部标题	true	旧版UI模式	string
mainLineRight	右副框上层	true	旧版UI模式	string
qrcodeText	二维码码值	true	旧版UI模式	string
subLine	通行结果	true	旧版UI模式	string
subLineLeft	左副框下层	true	旧版UI模式	string
subLineRight	右副框下层	true	旧版UI模式	string
reqTimestamp	时间戳毫秒 1666881956513	true	公共参数	integer(int64)
sign	签名md5(appKey+appSecret+timestamp) 小写32位	true	公共参数	string
msg	返回消息	false	公共参数	string

响应示例:

```
//三方接口返回数据 加密后的报文
{
  "code": 200,
  "data": {
    "bizContent":
    "NI4ry+D01kDXrNf50cmAzMGaB0td9kXfa321xmIS3qw5homFdZT4xaJu8Vqid37NDivdqj1ADzd0+v+QDKNovFSB
    kmOyOQt20bJ4HW43i0dKKHAKlI3FtZA0F9URhlqXbckbpGMD1hev3XS/b9uxRw+QLOnTzhuzgoxpiOXNA9rmQs3V7
    prpF9Wpaetts2pv",
    "reqTimestamp": 0,
    "sign": "签名"
  },
  "msg": ""
}

//三方接口返回数据 加密前的报文
```

//设备行为参数+指定设备显示内容的参数两者构成一次响应内容

//显示参数的渲染规则：传了内容则渲染，不传则不显示

```
{
  "code": 200,
  "data": {
    "bizContent": {
      //指定设备的行为参数
      "delay": 0,
      "end": true,
      "isOpen": 0,
      "openTimes": 1,
      "openVoiceMsg": "请通行",
      "qrcodeText": "",
      "showPageType": "scenic",
      "signalNo": "",
      "signalType": "default",
      "status": "closing",
      "taskId": "",
      "countDown": 3,
      "name": "张三",
      "idNumber": "330821199603150209",
      "photo": "",
      //指定显示结果页（传 a 类参数）-showPageType为scenic, status为closing时
      "aBackgroundUrl": "",
      "aLineOneLabel": "票名",
      "aLineOneValue": "年卡（季卡）",
      "aLineThrLabel": "年卡照片",
      "aLineThrValue":
        "https://wx1.sinaimg.cn/mw690/a34412431y1g559drgon5j21jk1j54qp.jpg",
      "aLineTwoLabel": "有效期",
      "aLineTwoValue": "2023-03-13",
      "aMainIcon": 1,
      "aMainLine": "请通行",
      "aSubLineFirst": "张*三",
      "aSubLineSecond": "330821*****0212",
      //指定显示单选页（传 b 类参数）-showPageType为scenic, status为pending时
      "bChoiseList": [
        {
          "deep": "true",
          "label": "航汇测试按钮",
          "value": "1"
        }
      ],
      "bBackgroundUrl": "",
      "bMainIcon": 1,
      "bMainLine": "等待二次核验",
      "bSubLine": "请选择票种进行核销",
      //指定显示多选页（传 c 类参数）-showPageType为scenic, status为pendingMultiple时
      "cBackgroundUrl": "",
      "cButtonText": "确定核销",
      "cButtonSubText": "已选择$count张票",
      "cChoiseList": [
```

```

{
    "list": [
        {
            "label": "航汇测试票种产品1",
            "selected": "true",
            "value": "1",
            "topTitle": "订单号: 12345678912345324234",
            "bottomLeftTitle": "【门票】",
            "bottomRightTitle": "使用人数: 2"
        }
    ],
    "subTitle": "330821*****0212",
    "title": "张三",
}

//指定显示过渡页（传 d 类参数）-showPageType为scenic, status为loading时
"dBackgroundUrl": "",
"dMainLine": "请等待...",
//（即将淘汰）旧版UI模式 showPageType为common, status为closing时响应参数
"mainLine": "",
"mainLineLeft": "",
"mainLineRight": "",
"subLine": "",
"subLineLeft": "",
"subLineRight": "",
"advertise": "",
},
"reqTimestamp": 0,
"sign": "签名"
},
"msg": ""
}

```

国密4加解密算法及签名认证

```

// pom.xml中引入
<dependency>
    <groupId>cn.hutool</groupId>
    <artifactId>hutool-all</artifactId>
    <version>5.7.0</version>
</dependency>

```

```

import cn.hutool.core.util.StrUtil;
import cn.hutool.crypto.SecureUtil;
import cn.hutool.crypto.SmUtil;
import cn.hutool.crypto.symmetric.SymmetricCrypto;

@Slf4j
public class Sm4Util {

    /**

```

```

    * 加密
    *
    * @param privateKey 管理员提供
    * @param str
    * @return
    */
    public static String encrypt(final String privateKey, final String str) {
        if (StringUtil.isBlank((CharSequence) str)) {
            return "";
        }

        try {
            SymmetricCrypto sm4 = new SymmetricCrypto("SM4/ECB/PKCS5Padding",
privateKey.getBytes());
            return sm4.encryptHex(str, Charset.forName("UTF-8"));
        } catch (Exception e) {
            log.error("加密失败", e);
            return null;
        }
    }

    /**
    * 解密
    *
    * @param privateKey 管理员提供
    * @param str
    * @return
    */
    public static String decrypt(final String privateKey, final String str) {
        if (StringUtil.isBlank((CharSequence) str)) {
            return null;
        }
        try {
            SymmetricCrypto sm4 = new SymmetricCrypto("SM4/ECB/PKCS5Padding",
privateKey.getBytes());
            return sm4.decryptStr(str, Charset.forName("UTF-8"));
        } catch (Exception e) {
            log.error("解密失败", e);
            return null;
        }
    }

    /**
    * 验证签名
    *
    * @param appKey
    * @param appSecret
    * @param sign
    * @param timestamp
    * @return
    */

```

```

    public static Boolean checkSign(String appKey, String appSecret, String sign, String
timestamp) {
        String dbSign = SecureUtil.md5(new StringBuilder().append(appKey)
            .append(appSecret)
            .append(timestamp).toString()).toLowerCase();
        if (!dbSign.equals(sign)) {
            return Boolean.FALSE;
        }
        return Boolean.TRUE;
    }

    /**
     * 获取签名
     *
     * @param appKey
     * @param appSecret
     * @param timestamp
     * @return
     */
    public static String getSign(String appKey, String appSecret, String timestamp) {
        String sign = SecureUtil.md5(new StringBuilder().append(appKey)
            .append(appSecret)
            .append(timestamp).toString()).toLowerCase();
        return sign;
    }
}

```

.net调用方式

加密:

解密: