

HHArome授权管理接入文档

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appId、appKey、appSecret、privateKey管理员提供

国密M4加密方法，参照文末代码示例

修正历史记录：

- 1. 2025-03-21 第一次添加

获取ampe设备列表（分页）

接口地址: https://tx.hz-hanghui.com:8088/hanghui-server-platform/openapi/v1/ampe/device/page

请求方式: POST

请求数据类型: application/json

响应数据类型: */*

接口描述: 获取ampe设备分页列表

请求示例:

```
1 //国密m4加密后密文
2 {
3   "appId": "",
4   "bizContent":
5     "NI4ry+D01kDXrNf50cmAzMGaB0td9kXfa321xmIS3qw5homFdZT4xaJu8Vqid37NDivdqj1ADzd0+v",
6   "reqTimestamp": 1692693665800,
7   "sign": "签名方式见文末加密算法"
8 }
9 //国密m4加密前
10 {
11   "appId": "",
12   "bizContent": {
13     "ampeName": "",
14     "pageNumber": 1,
15     "pageSize": 10,
16     "sn": "",
17     "status": true,
18     "tenantName": ""
19   },
20   "reqTimestamp": 1692693665800,
21   "sign": "签名方式见文末加密算法"
22 }
23
```

请求参数:

参数名称	参数说明	请求类型	是否必须	数据类型	
req	req	body	true	SM4EncryptReq	
appld	API调用标识 管理员颁发		true	string	
bizContent	privateKey私钥加密后的内容		true	string	
sn	设备编号	bizContent	false	string	
tenantName	服务商平台名称	bizContent	false	string	
ampeName	ampe平台名称	bizContent	false	string	
status	授权状态 false: 未授权 true: 已授权	bizContent	false	boolean	
pageNumber	第几页 起始页1	bizContent	true		
pageSize	每页条数	bizContent	true		
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«SM4EncryptResp»

响应参数:

参数名称	参数说明	类型	schema
code	状态码	integer(int32)	integer(int32)
data	对象	SM4EncryptResp	SM4EncryptResp
appld	API调用标识 管理员颁发	string	
bizContent	privateKey私钥加密后的内容	string	
pages	总页数	integer(int64)	
total	总条数	integer(int64)	
records	分页记录	array	
ampeName	ampe平台名称	string	
tenantName	服务商平台名称	string	
sn	设备编号	string	
status	授权状态 false: 未授权 true: 已授权	boolean	
reqTimestamp	时间戳毫秒 1666881956513	integer(int64)	
sign	签名md5(appKey+appSecret+timestamp) 小写32位	string	
msg	返回消息	string	

响应示例:

1	//国密m4加密后密文
---	-------------

```

2  {
3    "code": 200,
4    "data": {
5      "appId": "",
6      "bizContent":
"NI4ry+D01kDXrNf50cmAzMGaB0td9kXfa321xmIS3qw5homFdZT4xaJu8Vqid37NDivdqj1ADzd0+v+QD",
7      "reqTimestamp": 1692693665800,
8      "sign": "签名方式见文末加密算法"
9    },
10   "msg": ""
11 }
12 //国密m4加密前
13 {
14   "code": 200,
15   "data": {
16     "appId": "",
17     "bizContent": {
18       "pages": 1,
19       "records": [{
20         "ampeName": "",
21         "sn": "",
22         "status": true,
23         "tenantName": ""
24       }],
25       "total": 1
26     },
27     "reqTimestamp": 1692693665800,
28     "sign": "签名方式见文末加密算法"
29   },
30   "msg": ""
31 }
32
33

```

获取ampe列表

接口地址: `https://tx.hz-hanghui.com:8088/hanghui-server-platform/openapi/v1/ampe/list`

请求方式: `POST`

请求数据类型: `application/json`

响应数据类型: `*/*`

接口描述: 获取ampe列表

请求示例:

```

1  //国密m4加密后密文
2  {
3    "appId": "",

```

```
4      "bizContent": "D01kDXrNf50cmAzMGaB0td9kXfa321xmIS3qw5homFdZT4xaJu8Vqid37NDivdqj1ADzd0",
5      "reqTimestamp": 1692693665800,
6      "sign": "签名方式见文末加密算法"
7  }
8
9  //国密m4加密前
10 {
11     "appId": "",
12     "bizContent": {
13         "ampeName": "xxxxx"
14     },
15     "reqTimestamp": 1692693665800,
16     "sign": "签名方式见文末加密算法"
17 }
```

请求参数:

参数名称	参数说明	请求类型	是否必须	数据类型	
req	req	body	true	SM4EncryptReq	
appId	API调用标识 管理员颁发		true	string	
bizContent	privateKey私钥加密后的内容		true	string	
ampeName	ampe平台名称		false	string	
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«SM4EncryptResp»

响应参数:

参数名称	参数说明	类型	schema
code	状态码	integer(int32)	integer(int32)
data	对象	SM4EncryptResp	SM4EncryptResp
appld	API调用标识 管理员颁发	string	
bizContent	privateKey私钥加密后的内容	string (array)	
ampeName	ampe平台名称	string	
appld	小程序AppId(miniProgram_appid)	string	
appName	小程序名称	string	
hostAppId	调用应用(ampe_appid)	string	
productId	产品ID(product_id)	string	
reqTimestamp	时间戳毫秒 1666881956513	integer(int64)	
sign	签名md5(appKey+appSecret+timestamp) 小写32位	string	
msg	返回消息	string	

响应示例:

```
1 //国密m4加密后密文
2 {
3   "code": 200,
4   "data": {
5     "appId": "",
6     "bizContent":
7     "NI4ry+D01kDXrNf50cmAzMGaB0td9kXfa321xmIS3qw5homFdZT4xaJu8Vqid37NDivdqj1ADzd0+v+QD",
8     "reqTimestamp": 1692693665800,
9     "sign": "签名方式见文末加密算法"
10  },
11  "msg": ""
12 }
13 //国密m4加密前密文
14 {
15   "code": 200,
16   "data": {
17     "appId": "",
18     "bizContent": [{
19       "ampeName": "",
20       "appId": "",
21       "appName": "",
22       "hostAppId": "",
23       "productId": ""
24     }],
25     "reqTimestamp": 1692693665800,
26     "sign": "签名方式见文末加密算法"
27  },
28  "msg": ""
29 }
30 }
```

获取服务商列表

接口地址: https://tx.hz-hanghui.com:8088/hanghui-server-platform/openapi/v1/ampe/tenant/list

请求方式: POST

请求数据类型: application/json

响应数据类型: */*

接口描述: 获取服务商列表

请求示例:

```
1 //国密m4加密后密文
2 {
3   "appId": "",
4   "bizContent": "D01kDXrNf50cmAzMGaB0td9kXfa321xmIS3qw5homFdZT4xaJu8Vqid37NDivdqj1ADzd0",
5   "reqTimestamp": 1692693665800,
6   "sign": "签名方式见文末加密算法"
7 }
8 //国密m4加密前
9 {
10  "appId": "",
11  "bizContent": {
12    "name": ""
13  },
14  "reqTimestamp": 1692693665800,
15  "sign": "签名方式见文末加密算法"
16 }
```

请求参数:

参数名称	参数说明	请求类型	是否必须	数据类型	
req	req	body	true	SM4EncryptReq	
appId	API调用标识 管理员颁发		true	string	
bizContent	privateKey私钥加密后的内容		true	string	
name	服务商名称		false	string	
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«SM4EncryptResp»

响应参数:

参数名称	参数说明	类型	schema
code	状态码	integer(int32)	integer(int32)
data	对象	SM4EncryptResp	SM4EncryptResp
appld	API调用标识 管理员颁发	string	
bizContent	privateKey私钥加密后的内容	string (array)	
machineAppId	设备appld	string	
name	服务商名称	string	
serviceAddress	商户平台地址	string	
serviceAppId	商户appld	string	
reqTimestamp	时间戳毫秒 1666881956513	integer(int64)	
sign	签名md5(appKey+appSecret+timestamp) 小写32位	string	
msg	返回消息	string	

响应示例:

```
1 //国密m4加密后密文
2 {
3   "code": 200,
4   "data": {
5     "appId": "",
6     "bizContent": "D01kDXrNf50cmAzMGaB0td9kXfa321xmIS3qw5homFdZT4xaJu8Vqid37NDivdqj1ADzd0",
7     "reqTimestamp": 1692693665800,
8     "sign": "签名方式见文末加密算法"
9   },
10  "msg": ""
11 }
12
13 //国密m4加密前明文
14 {
15   "code": 200,
16   "data": {
17     "appId": "",
18     "bizContent": [{
19       "machineAppId": "",
20       "name": "",
21       "serviceAddress": "",
22       "serviceAppId": ""
23     }],
24     "reqTimestamp": 1692693665800,
25     "sign": "签名方式见文末加密算法"
26   },
27   "msg": ""
28 }
29
```

修改ampe设备配置信息

接口地址: `https://tx.hz-hanghui.com:8088/hanghui-server-platform/openapi/v1/ampe/device/update`

请求方式: `POST`

请求数据类型: `application/json`

响应数据类型: `*/*`

接口描述: 修改ampe设备配置信息

请求示例:

```
1 //国密m4加密后密文
2 {
3   "appId": "",
4   "bizContent":
5     "NI4ry+D01kDXrNf50cmAzMGaB0td9kXfa321xmIS3qw5homFdZT4xaJu8Vqid37NDivdqj1ADzd0+v",
6   "reqTimestamp": 1692693665800,
7   "sign": "签名方式见文末加密算法"
8 }
9 //国密m4加密前
10 {
11   "appId": "",
12   "bizContent": {
13     "ampeName": "",
14     "appletEnterPageUrl": "",
15     "autoLaunch": true,
16     "automaticEnterApplet": true,
17     "cacheSwitch": true,
18     "expirationDate": "",
19     "forbidPullDown": true,
20     "hideNavigationBar": true,
21     "homePageUrl": "",
22     "idCardReaderBrand": 0,
23     "infraredSensingSwitch": true,
24     "isOpenMicrophone": true,
25     "isOpenPrinter": true,
26     "m1cardSwitch": true,
27     "manualInIdCard": true,
28     "manualInIdCardAuth": {
29       "appId": "",
30       "appKey": "",
31       "appSecret": "",
32       "privateKey": ""
33     },
34     "microphoneType": 0,
35     "mpassword": "",
36     "nfcSwitch": true,
37     "printerConfig": 0,
38     "qrCodeReaderBrand": 0,
```

```
39      "qrCodeReaderCommunicationMode": 0,
40      "qrCodeReaderSerialPort": "",
41      "sn": "",
42      "socialSecurityCardSwitch": true,
43      "startPageUrl": "",
44      "tenantName": "",
45      "ttsSwitch": true,
46      "useAlipayFace": true,
47      "useIdcard": true,
48      "useQrcode": true,
49      "volume": 0
50  },
51  "reqTimestamp": 1692693665800,
52  "sign": "签名方式见文末加密算法"
53  }
54
55
```

请求参数:

参数名称	参数说明	请求类型	是否必须	数据类型	
req	req	body	true	SM4EncryptReq	
appId	API调用标识 管理员颁发		true	string	
bizContent	privateKey私钥加密后的内容		true	string (object)	
sn	设备编号		true	string	
ampeName	ampe平台名称		false	string	
appletEnterPageUrl	入口页面地址		false	string	
autoLaunch	应用是否自启动 false-不启用(默认) true-启用		false	boolean	
automaticEnterApplet	是否自动进入小程序 false-不启用(默认) true-启用		false	boolean	
cacheSwitch	是否开启缓存功能 false-不启用 true-启用		false	boolean	
expirationDate	使用时间期限		false	string(date)	
forbidPullDown	禁止下拉通知栏 false-不启用(默认) true-启用		false	boolean	
hideNavigationBar	隐藏底部导航栏 false-不启用(默认) true-启用		false	boolean	
homePageUrl	应用首页图片		false	string	
idCardReaderBrand	身份证阅读器品牌		false	integer(int32)	
infraredSensingSwitch	是否开启红外感应		false	boolean	
isOpenMicrophone	是否开启麦克风 默认false		false	boolean	
isOpenPrinter	是否开启打印机 默认false		false	boolean	
m1cardSwitch	是否开启M1卡功能		false	boolean	
manuallyInIdCard	是否开启手动输入证件号 默认false		false	boolean	

manualInIdCardAuth	开启手动输入证件号的密钥信息		false	object	
appld	接口调用凭证		false	string	
appKey	加解密签名参数		false	string	
appSecret	加解密签名参数		false	string	
privateKey	加解密参数		false	string	
microphoneType	麦克风类型 1-原生音频流 2-数科asr		false	integer(int32)	
mpassword	设备管理员密码		false	string	
nfcSwitch	是否开启NFC功能		false	boolean	
printerConfig	打印机型号 1-YX02内置打印机 2-Mla80打印机 3-MlaFCDD215（A4热敏打印机）		false	integer(int32)	
qrCodeReaderBrand	二维码读头品牌 1-天波TPS980内置二维码 2-天波D2内置二维码 3-天波K8内置二维码 4-东信老款二维码读头 5-东信新款二维码读头 6-东信多合一一体式二维码 7-新大陆 8-东信M4500高速二维码读头 9-EST-M20东信手持机二维码读头 99-其他		false	integer(int32)	
qrCodeReaderCommunicationMode	二维码读头通信模式 1-虚拟串口 2-串口serialPort 3-虚拟键盘		false	integer(int32)	
qrCodeReaderSerialPort	二维码读头串口地址serialPort		false	string	
sn	设备编号		true	string	
socialSecurityCardSwitch	是否开启社保卡功能		false	boolean	
startPageUrl	启动页地址		false	string	
tenantName	服务商平台名称		false	string	
ttsSwitch	是否开启TTS		false	boolean	
useAlipayFace	启用支付宝人脸能力 false-不启用 true-启用(默认)		false	boolean	
uselddcard	启用身份证功能 false-不启用 true-启用(默认)		false	boolean	
useQrcode	启用二维码功能 false-不启用 true-启用(默认)		false	boolean	
volume	设备语音音量 0-100 默认值80		false	integer(int32)	
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«boolean»

响应参数:

参数名称	参数说明	类型	schema
code	状态码	integer(int32)	integer(int32)
data	对象	boolean	
msg	返回消息	string	

响应示例:

```
1 //加密后密文
2 {
3   "code": 200,
4   "data": {
5     "appId": "",
6     "bizContent": "",
7     "reqTimestamp": 1692693665800,
8     "sign": "签名方式见文末加密算法"
9   },
10  "msg": ""
11 }
12
13 //加密前明文
14 {
15   "code": 200,
16   "data": {
17     "appId": "",
18     "bizContent": true,
19     "reqTimestamp": 1692693665800,
20     "sign": "签名方式见文末加密算法"
21   },
22   "msg": ""
23 }
```

国密4加解密算法及签名认证

```
1 // pom.xml中引入
2 <dependency>
3   <groupId>cn.hutool</groupId>
4   <artifactId>hutool-all</artifactId>
5   <version>5.7.0</version>
6 </dependency>
```

```
1  import cn.hutool.core.util.StrUtil;
2  import cn.hutool.crypto.SecureUtil;
3  import cn.hutool.crypto.SmUtil;
4  import cn.hutool.crypto.symmetric.SymmetricCrypto;
5
6  @Slf4j
7  public class Sm4Util {
8
9      /**
10       * 加密
11       *
12       * @param privateKey 管理员提供
13       * @param str
14       * @return
15       */
16     public static String encrypt(final String privateKey, final String str) {
17         if (StrUtil.isBlank((CharSequence) str)) {
18             return "";
19         }
20
21         try {
22             SymmetricCrypto sm4 = new SymmetricCrypto("SM4/ECB/PKCS5Padding",
privateKey.getBytes());
23             return sm4.encryptHex(str, Charset.forName("UTF-8"));
24         } catch (Exception e) {
25             log.error("加密失败", e);
26             return null;
27         }
28     }
29
30     /**
31      * 解密
32      *
33      * @param privateKey 管理员提供
34      * @param str
35      * @return
36      */
37     public static String decrypt(final String privateKey, final String str) {
38         if (StrUtil.isBlank((CharSequence) str)) {
39             return null;
40         }
41         try {
42             SymmetricCrypto sm4 = new SymmetricCrypto("SM4/ECB/PKCS5Padding",
privateKey.getBytes());
43             return sm4.decryptStr(str, Charset.forName("UTF-8"));
44         } catch (Exception e) {
45             log.error("解密失败", e);
46             return null;
47         }
48     }
49
50     /**
```

```
51     * 验证签名
52     *
53     * @param appKey
54     * @param appSecret
55     * @param sign
56     * @param timestamp
57     * @return
58     */
59     public static Boolean checkSign(String appKey, String appSecret, String sign, String
timestamp) {
60         String dbSign = SecureUtil.md5(new StringBuilder().append(appKey)
61             .append(appSecret)
62             .append(timestamp).toString()).toLowerCase();
63         if (!dbSign.equals(sign)) {
64             return Boolean.FALSE;
65         }
66         return Boolean.TRUE;
67     }
68
69     /**
70     * 获取签名
71     *
72     * @param appKey
73     * @param appSecret
74     * @param timestamp
75     * @return
76     */
77     public static String getSign(String appKey, String appSecret, String timestamp) {
78         String sign = SecureUtil.md5(new StringBuilder().append(appKey)
79             .append(appSecret)
80             .append(timestamp).toString()).toLowerCase();
81         return sign;
82     }
83
84 }
85
```