

人脸设备 dx_vf106 MQTT接口文档

修改记录

版本	日期	修改人	涉及接口	修改内容
V1.0.0	2022.08.14	zhangxt	*	初始化接口文档
V1.0.1	2022.08.17	zhangxt	*	人脸权限下发逻辑优化
V1.0.2	2022.08.24	zhangxt	*	远程控制返回data参数优化
V1.0.3	2022.08.26	zhangxt	*	增加密码通行逻辑
V1.0.4	2023.01.09	马志祥	权限管理	取消人脸特征值概念， 改成人脸照片base64交互

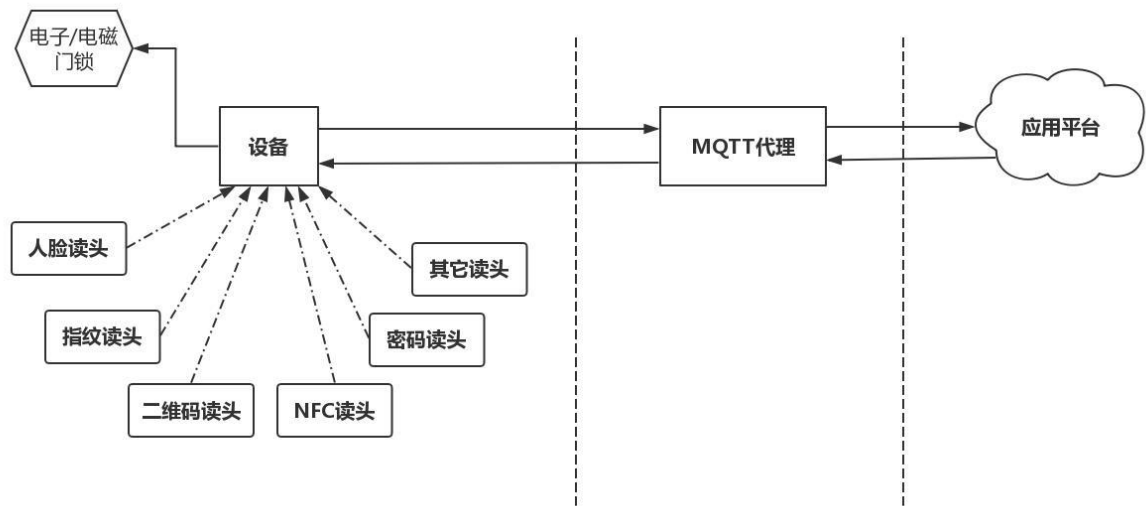
文章目录
>>

一. 概述

1.1 应用场景

本文档提供一套直接访问设备的接口，用于推送获取设备信息，使用者购买设备后，就可以直接根据接口说明进行自定义开发整套门禁或相关应用系统。接口使用 MQTT 通信协议，可以对接各种通用的 MQTT 代理，也可使用我们推荐的 PaaS 平台。

此款dx_vf106设备是Android系统，同时此接口是基于我们自带应用实现。如果您基于Android系统另行开发应用，可自行对接Android底层系统开发文档。



1.2 接口规范

接口使用 MQTT 协议，按方向分为上行与下行：

- **上行：**是指设备上发消息给后台或应用。
- **下行：**是指后台或应用下发消息给设备。

按发起方分为：指令类、事件类

- **指令类：**是由应用平台发起，先**下行**，给设备发送指令，再**上行**，设备返回给应用平台。
- **事件类：**是由设备主动发起，先**上行**，设备上报到平台应用一些数据，再**下行**，平台应用返回给设备一个回复。

文章目录>

消息格式是标准的 MQTT 报文结构，包含消息 Topic 与消息内容，格式如下：

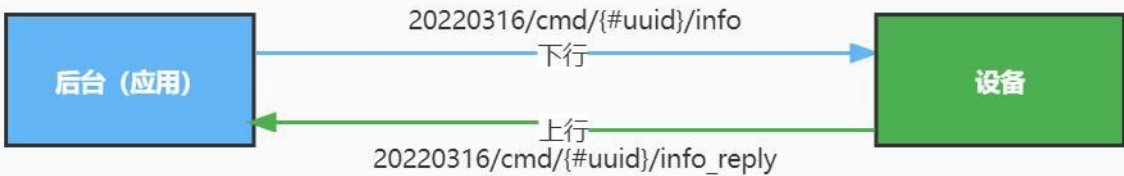
1) 消息 Topic：消息标题,不管是上行还是下行，发送指令和返回指令的 Topic 基本规则是**指令**和**指令_reply**。格式如下：

1. - 下行：20220814/cmd/{#uuid}/xxxx，对应的返回指令（上行）：
20220814/cmd/xxxx_reply

2. - 上行：20220814/event/yyyy，对应的返回指令（下行）：
20220814/event/{#uuid}/yyyy_reply

注：其中20220316是接口版本的特定标识，cmd 表示由后台或应用先主动给设备发送指令，event 表示设备主动给后台或应用发送消息或触发事件，{#uuid} 表示设备的唯一标识，示例如下图

1) 后台 (应用) 下发指令到设备



2) 设备上报数据到后台 (应用)



2) 消息内容:文本格式的消息内容，内容为Json格式
发送消息内容的示例如下：

```
1. {
2.     "serialNo": "6w8keif5g6",
3.     "uuid": "0a1b2c3d",
4.     "data": {
5.         ...
6.     },
7.     "time":1647580466,
8.     "sign":"e0k4jrjr85tje8ru4jrur499r99ii4ur"
9. }
```

返回消息内容的示例如下：

```
1. {
2.     "serialNo": "6w8keif5g6",
3.     "uuid": "0a1b2c3d",
4.     "code": "000000",
5.     "data": {
6.         ...
7.     },
8.     "time":1647580466,
9.     "sign":"e0k4jrjr85tje8ru4jrur499r99ii4ur"
10. }
```

文章目录 >>

参数名	说明	类型	必须	备注
serialNo	序列号	string	是	后台或应用给设备发送消息必须传递唯一-不超过32位，设备收到消息后反馈结果的时候会使用，反之亦然。
uuid	设备唯一标识	string	是	大于8位的字符串
data	消息数据正文	object	是	为json格式，不同类型的消息正文格式，接口列表中主要是详细列出这部分数据
time	时间戳	long	是	10位长度时间戳，单位秒
sign	消息签名	string	否	如果签名启动，这个值必须有，验证数字签名的规则请参考：附录- 签名算法
code	结果码	string	否	返回的数据必须包含 code ，不同的 code 表示不同的结果标识。结果码的基本规则附录- 结果码

文章目录>>

注：要求消息内容中的每一项内容命名，都用小写字母开头，后面可以大小写结合；

二、接口列表

2.1 指令类

2.1.1、配置管理

序号	接口名称	操作	备注
2.1.1.1	配置查询	详情	查询所有或部分设备配置信息
2.1.1.2	配置修改	详情	修改设备配置信息，有部分信息不能修改，部分修改会自动触发设备重启

2.1.1.1 配置查询

1) 下行:应用平台发送查询配置指令给设备

- 说明:

- Topic: `20220814/cmd/{#uuid}/getConfig`
- 内容: 标准格式
- 内容里的 data 部分:
 - data 可以为空, 或者不传这个部分, 这个时候设备将返回所有配置数据
 - data 可以为string, 因为配置数据是 key/value 对应的 json 格式的对象, 所以如果传入一个指定的 key, 就可以返回设备里这个 key 对应的 value。支持最多2级, 比如 `key1.key2`, 表示获取设备里 `key1` 对应的数据对象, 再在这个对象里找到 `key2` 对应的value
- 请求示例

```
1. //获取所有数据
2. {
3.   "serialNo": "6w8keif5g6",
4.   "uuid": "0a1b2c3d",
5.   "time": 1647580466,
6.   "sign": ""
7. }
8. //获取网络相关数据
9. {
10.  "serialNo": "6w8keif5g6",
11.  "uuid": "0a1b2c3d",
12.  "time": 1647580466,
13.  "data": "netInfo"
14.  "sign": ""
15. }
16. //获取网络相关数据下的 ip 地址
17. {
18.  "serialNo": "6w8keif5g6",
19.  "uuid": "0a1b2c3d",
20.  "time": 1647580466,
21.  "data": "netInfo.ip"
22.  "sign": ""
23. }
```

文章目录
>>

2) 上行: 设备根据查询条件查出相应配置数据返回给应用平台

- 说明:
- Topic: `20220814/cmd/getConfig_reply`
- 内容: 标准格式
- data 部分: json 格式, key/value 对应很多属性。详情参考附件:配置数据详情
主要分以下几个大类:
 - `baseInfo`: 设备基础信息, 包括设备类型、版本、uuid 等等
 - `sysInfo`: 设备系统资源信息, 包括设备CPU、存储等等

- `netInfo`: 设备网络相关信息, 包括设备ip、wifi用户密码等等
- `mqttInfo`: mqtt 服务端相关信息, 包括服务的ip、端口等等
- `statusInfo`: 设备实时的状态信息, 包括门磁、火警等等, 只能读取
- `doorInfo`: 设备控制门相关的信息, 包括开门时长等等
- `temperatureInfo`: 设备测温相关信息, 包括测温开关等
- `faceInfo`: 设备人脸配置相关信息, 包括活体检测、口罩和人脸核验相关语音等
- `security`: 安全相关的密钥, 通常只能设置, 不能读取
- `customer`: 自定义的一些属性
- `debug`: 打开\关闭调试模式, 调试模式打开会有完整日志, 会极大降低效率, 特定情况下需要

• 返回示例:

```
1. //比如应用平台发送查询 netInfo.ip, 返回示例
2. {
3.   "serialNo": "6w8keif5g6",
4.   "uuid": "0a1b2c3d",
5.   "time": 1647580466,
6.   "code": "000000",
7.   "data": {
8.     "netInfo": {
9.       "ip": "192.168 .32 .23 "
10.    }
11.  },
12.  "sign": ""
13. }
```

文章目录>>

• 特殊结果码

类型	code	信息	说明
参数异常	221110	未找到对应的数据	传递的参数可能写错了, 注意大小写

[回到顶部](#)

2.1.1.2 配置修改

1) 下行:应用平台发送修改配置指令到设备

- 说明:
- Topic: `20220814/cmd/{#uuid}/setConfig`
- 内容: 标准格式
- 内容里的 data 部分:
 - data 可以为空, 或者不传这个部分, 这个时候设备不会做任何配置修改

- data 可以为 json 格式对象，传入一个指定的 key 和 value，就可以修改设备里这个 key 对应的 value。支持最多2级，比如 `key1.key2`，表示设置 `key1` 对应的数据对象，再在这个对象里找到 `key2` 对应的value。详情参考[附件:配置数据详情](#)，里面有部分数据不允许修改

• 请求示例

```
1. //修改网络相关数据下的 ip 地址
2. {
3.   "serialNo": "6w8keif5g6",
4.   "uuid": "0a1b2c3d",
5.   "time": 1647580466,
6.   "data": {
7.     "netInfo": {
8.       "ip": "192.168 .32 .23"
9.     }
10.  },
11.   "sign": ""
12. }
```

2) 上行：设备根据查询条件查出相应配置数据返回给应用平台

- 说明:
- Topic: `20220814/cmd/setConfig_reply`
- 内容: 标准格式
- data 部分: 为空
- 返回示例:

```
1. //比如应用平台发送查询 netInfo.ip, 返回示例
2. {
3.   "serialNo": "6w8keif5g6",
4.   "uuid": "0a1b2c3d",
5.   "time": 1647580466,
6.   "code": "000000",
7.   "sign": ""
8. }
```

• 特殊结果码

类型	code	信息	说明
参数异常	221120	未找到对应的数据	传递的参数可能写错了，注意大小写

文
章
目
录

>>

类型	code	信息	说明
参数异常	221121	要修改的数据格式不正确	比如应该是 String 类型的，但是传递了 int 类型；只能是0/1二种值，但是传递了2
其他异常	321120	这个值不允许修改	-

回到顶部

2.1.2、权限管理

序号	接口名称	操作	备注
2.1.2.1	查询权限	详情	查询所有权限或按条件查部分权限
2.1.2.2	添加权限	详情	插入权限
2.1.2.3	删除权限	详情	按权限 id 删除权限
2.1.2.4	清空权限	详情	删除这个设备下所有权限

文章目录

>>

2.1.2.1 查询权限

1) 下行:应用平台发送查询权限指令给设备

- 说明:
- Topic: `20220814/cmd/{#uuid}/getPermission`
- 内容: [标准格式](#)
- 内容里的 data 部分:

参数名	说明	类型	必须	备注
size	每页数量	Int	是	每页数量，最大 200(人脸类型只限1条)
page	页码	Int	是	第几页，从0开始
code	凭证号	String	否	凭证号
type	凭证类型	Int	否	目前支持100-透传二维码，101-静态二维码，103-动态二维码，202-卡，303-人脸，501-通行密码
id	权限id	String	否	-

- 请求示例

```
1. //查询所有权限
2. {
3.   "serialNo": "6w8keif5g6",
4.   "uuid": "0a1b2c3d",
5.   "data": {
6.     "page": 0,
7.     "size": 200
8.   },
9.   "time": 1647580466,
10.  "sign": ""
11. }
12. //按code和type查询
13. {
14.   "serialNo": "6w8keif5g6",
15.   "uuid": "0a1b2c3d",
16.   "data": {
17.     "page": 0,
18.     "size": 200,
19.     "code": "2kifk39k40",
20.     "type": 202
21.   },
22.   "time": 1647580466,
23.   "sign": ""
24. }
25. //按id查询，虽然最多只有一条记录，为了统一，size和page也加上
26. {
27.   "serialNo": "6w8keif5g6",
28.   "uuid": "0a1b2c3d",
29.   "data": {
30.     "page": 0,
31.     "size": 200,
32.     "id": "2kifk39k40"
33.   },
34.   "time": 1647580466,
35.   "sign": ""
36. }
```

文章目录

>>

2) 上行：设备根据查询条件查出相应权限数据返回给应用平台

- 说明:
- Topic: 20220814/cmd/getPermission_reply
- 内容: 标准格式
- data 部分: json 对象格式

参数名	说明	类型	必须	备注
page	当前页	Int	是	当前页（0开始）
size	每页大小	Int	是	当前页大小

参数名	说明	类型	必须	备注
total	总数	Int	是	白名单总数
totalPage	总页数	Int	是	总页数
count	当前页实际白名单数	Int	是	当前页实际白名单数
content	权限列表	Array	是	权限列表
— id	权限Id	String	是	权限的唯一标识，添加和删除时根据这个标识来判定，建议传32位随机值
— type	权限类型	Int	是	目前支持100-透传二维码，101-静态二维码，103-动态二维码，202-卡，303-人脸，501-通行密码
—code	权限值	String	是	权限类型为303时，这个值对应base64
—extra	额外参数	Object	否	可扩展，人脸设备会传递人员名称等信息，参数详见权限添加时的 extra参数表
—time	时间区间	Object	是	查看时间区间

文章目录 >>

• 返回示例：

```
1. {
2.   "serialNo": "6w8keif5g6",
3.   "uuid": "0a1b2c3d",
4.   "code": "000000",
5.   "data": {
6.     "content": [{
7.       "id": "20190101120101",
8.       "type": 101,
9.       "code": "001002",
10.      "extra": {
11.      },
12.      "time": {
13.        "type": 0
14.      }
15.    }],
16.    "page": 1,
```

```
17.         "size": 20,
18.         "total": 100,
19.         "totalPage": 5,
20.         "count": 20
21.     },
22.     "time": 1647580466,
23.     "sign": ""
24. }
```

• 特殊结果码

类型	code	信息	说明
参数异常	221210	未找到对应的数据	传递的参数可能写错了，注意大小写
参数异常	221211	超出页数或页数小于0	超出页数限制或页数小于0
参数异常	221212	超出每页最大限额	超出200，如果是人脸则超出1

[回到顶部](#)

2.1.2.2 添加权限

1) 下行:应用平台发送添加权限指令给设备

- 说明:
- Topic: 20220814/cmd/{#uuid}/insertPermission
- 内容: 标准格式
- 内容里的 data 部分: json array 数组对象，每个元素的参数如下:

参数名	必填	类型	说明	备注
id	权限Id	String	是	权限的唯一标识，添加和删除时根据这个标识来判定，建议传32位随机值
type	权限类型	Int	是	目前支持100-透传二维码，101-静态二维码，103-动态二维码，202-卡，303-人脸，501-通行密码
code	权限值	String	是	说明：如果权限类型为303，此值对应base64；如果type=501时，此值对应通行密码

文章目录

>>

参数名	必填	类型	说明	备注
extra	额外参数	Object	否	可扩展，人脸设备会传递人员名称等信息，参数详见权限添加时的 extra参数表
time	时间区间	Object	是	查看时间区间
account	通行次数	Int	否	通行次数，即通行权限按次计数

extra参数表

参数名	说明	类型	必须	备注
name	人员名称	String	是	name 参数必传且内容不可为空。长度、内容无限制。
accessMode	权限模式	Int	是	0: 仅权限通行模式，1: 人员识别+权限通行模式；默认：0
sex	性别	Int	否	0: 男，1: 女
country	国家	String	否	
groupName	分组名称	String	否	
idCardNumber	卡号	String	否	若人脸权限下发时填写了卡号，可直接刷对应卡号的卡进行识别，屏幕也会显示与该卡号对应的人员的名字。
mobile	手机号	String	否	人员手机号
remarks	备注	String	否	用于填写说明信息
photo	屏显照片	String	否	人员的Base64图片

文章目录 >>

• 请求示例

```
1. //插入权限
2. {
3.   "serialNo": "6w8keif5g6",
4.   "uuid": "0a1b2c3d",
5.   "data": [{
6.     "id": "20190101120101",
7.     "type": 101,
8.     "code": "001002",
9.     "extra": {},
10.    "time": {
```

```
11.         "type": 0
12.     }
13. },
14. {
15.     "id": "20190101120102",
16.     "type": 101,
17.     "code": "003004",
18.     "extra": {},
19.     "time": {
20.         "type": 0
21.     }
22. }
23. ],
24. "time": 1647580466,
25. "sign": ""
26. }
```

2) 上行：设备插入相应权限数据返回成功或失败给应用平台

- 说明: 要么都成功，要么全部失败，不会部分成功,如果插入的权限在设备里已经存在（id一样），则会忽略，但是还是返回成功
- Topic: 20220814/cmd/insertPermission_reply
- 内容: 标准格式
- data 部分: json array 数组对象，每个元素都是成功插入的权限id
- 返回示例:

文章目录

>>

```
1. {
2.   "serialNo": "6w8keif5g6",
3.   "uuid": "0a1b2c3d",
4.   "code": "000000",
5.   "data": ["20190101120101", "20190101120102"],
6.   "time": 1647580466,
7.   "sign": ""
8. }
```

- 特殊结果码

类型	code	信息	说明
参数异常	221220	缺少必须要有的参数	缺少必须要有的参数

回到顶部

2.1.2.3 删除权限

1) 下行:应用平台发送删除权限指令给设备

- 说明:
- Topic: 20220814/cmd/{#uuid}/delPermission
- 内容: 标准格式
- 内容里的 data 部分: json array 数组对象, 每个元素都是权限id:
- 请求示例

```
1. //删除权限
2. {
3.   "serialNo": "6w8keif5g6",
4.   "uuid": "0a1b2c3d",
5.   "data": ["20190101120101","20190101120102"],
6.   "time": 1647580466,
7.   "sign": ""
8. }
```

2) 上行: 设备删除相应权限数据返回成功或失败给应用平台

- 说明: 要么都成功, 要么全部失败, 不会部分成功,如果删除的权限在设备里没有存在则会忽略, 但是还是返回成功
- Topic: 20220814/cmd/delPermission_reply
- 内容: 标准格式
- data 部分: json array 数组对象, 每个元素都是成功删除的权限id
- 返回示例:

文章目录

>>

```
1. {
2.   "serialNo": "6w8keif5g6",
3.   "uuid": "0a1b2c3d",
4.   "code": "000000",
5.   "data": ["20190101120101","20190101120102"],
6.   "time": 1647580466,
7.   "sign": ""
8. }
```

- 特殊结果码

类型	code	信息	说明
参数异常	221230	缺少必须要有的参数	缺少必须要有的参数

回到顶部

2.1.2.4 清空权限

1) 下行:应用平台发送清空所有权限指令给设备

- 说明:
- Topic: 20220814/cmd/{#uuid}/clearPermission
- 内容: 标准格式
- 内容里的 data 部分: 无
- 请求示例

```
1. //清空权限
2. {
3.   "serialNo": "6w8keif5g6",
4.   "uuid": "0a1b2c3d",
5.   "time": 1647580466,
6.   "sign": ""
7. }
```

2) 上行: 设备清空所有权限数据返回成功或失败给应用平台

- 说明: 不会返回失败
- Topic: 20220814/cmd/clearPermission_reply
- 内容: 标准格式
- data 部分: 无
- 返回示例:

文章目录

>>

```
1. {
2.   "serialNo": "6w8keif5g6",
3.   "uuid": "0a1b2c3d",
4.   "code": "000000",
5.   "time": 1647580466,
6.   "sign": ""
7. }
```

回到顶部

2.1.3、远程控制

序号	接口名称	操作	备注
2.1.3.1	远程控制	详情	远程控制设备，触发设备的相应动作，包括重启、远程开门、设备启用、设备禁用、设备重置等

2.1.3.1 远程控制

1) 下行:应用平台发送控制指令给设备

- 说明:
- Topic: 20220814/cmd/{#uuid}/control
- 内容: 标准格式
- 内容里的 data 部分: json object 对象, 参数如下:

参数名	说明	类型	必须	备注
command	命令	Int	是	0: 重启, 1: 远程开门, 2: 设备启用, 3: 设备禁用, 4: 设备重置

- 请求示例

```
1. //远程重启
2. {
3.   "serialNo": "6w8keif5g6",
4.   "uuid": "0a1b2c3d",
5.   "data": {
6.     "command": 0
7.   },
8.   "time": 1647580466,
9.   "sign": ""
10. }
```

文章目录
>>

2) 上行: 设备接收到控制指令后做某些操作后返回成功或失败给应用平台

- 说明:
- Topic: 20220814/cmd/control_reply
- 内容: 标准格式
- data 部分: 同下行内容
- 返回示例:

```
1. {
2.   "serialNo": "6w8keif5g6",
3.   "uuid": "0a1b2c3d",
4.   "data": {
5.     "command": 0
6.   },
7.   "code": "000000",
8.   "time": 1647580466,
9.   "sign": ""
10. }
```

- 特殊结果码

类型	code	信息	说明
----	------	----	----

类型	code	信息	说明
参数异常	221310	缺少必须要有的参数	缺少必须要有的参数
参数异常	221311	不支持的 command 值	command 值错误

回到顶部

2.1.4、固件升级

序号	接口名称	操作	备注
2.1.4.1	应用升级	详情	升级固件里的应用部分或系统部分, 绝大部分情况只需要升级应用

2.1.4.1 升级固件

1) 下行:应用平台发送升级固件指令给设备

- 说明: 设备收到升级固件后, 先下载固件文件, 下载成功后返回结果给应用平台, 然后重启
- Topic: `20220814/cmd/{#uuid}/upgradeFirmware`
- 内容: [标准格式](#)
- 内容里的 data 部分: json object 对象, 参数如下:

文章目录

>>

参数名	说明	类型	必须	备注
type	升级包类型	Int	是	type值强校验, 必须为0或1 0: 系统升级, 文件为xxx.pkg 1: 应用升级升级包为xxx.tar.xz
url	升级包下载地址, http地址	String	是	调用接口时访问该url, 若url能访问到, 则设备开始执行下载文件; 若 url 访问失败, 则接口返回提示 “url 访问失败, 设备无法下载”
md5	升级包MD5值	String	是	升级包进行MD5验证

- 请求示例

```
1. {
2.   "serialNo": "6w8keif5g6",
3.   "uuid": "0a1b2c3d",
4.   "data": {
```

```
5.     "type": 1,
6.     "url": "http://host/aio.tar.xz",
7.     "md5": "521c2bdc835d4f13b5f9d6db164f0881"
8. },
9.     "time": 1647580466,
10.    "sign": ""
11. }
```

2) 上行：设备接收到升级固件指令后返回成功或失败给应用平台

- 说明:
- Topic: 20220814/cmd/upgradeFirmware_reply
- 内容: 标准格式
- data 部分: 无
- 返回示例:

```
1. {
2.   "serialNo": "6w8keif5g6",
3.   "uuid": "0a1b2c3d",
4.   "code": "000000",
5.   "time": 1647580466,
6.   "sign": ""
7. }
```

文章目录

>>

- 特殊结果码

类型	code	信息	说明
参数异常	221410	缺少必须要有的参数	缺少必须要有的参数
参数异常	221411	对应的固件下载地址下载不了	对应的固件下载地址
参数异常	221412	对应的固件下载后对应的md5校验失败	md5校验失败

回到顶部

2.1.5、同步密钥

序号	接口名称	操作	备注
2.1.5.1	同步密钥	详情	103动态二维码加解密需要使用

2.1.5.1 同步密钥

1) 下行:应用平台发送同步密钥指令给设备

- 说明: 设备有同步密钥才能正常使用动态二维码，不然无法识别不可正常使用。
- Topic: 20220814/cmd/{#uuid}/secretSync
- 内容: 标准格式
- 内容里的 data 部分: json object 对象，参数如下：

参数名	说明	类型	必须	备注
syncFlag	同步标识	Int	是	1-全量，2-增量
secretKeys	密钥数组	Array	是	详见secretKeys表

- secretKeys表

参数名	说明	类型	必须	备注
secretKeyType	密钥类型	String	是	密钥类型，如RSA, AES等
secretKeyCode	密钥编码	String	是	密钥编码，对应密钥编码，如vguang
secretKeyValue	密钥值	String	是	
secretKeyStartTime	开始时间	Int	是	密钥开始时间，时间戳（秒）
secretKeyExpirytime	过期时间	Int	是	密钥过期时间，时间戳（秒）
syncType	同步类型	Int	是	当SyncFlag=2时有效，1-增加，2-修改，3-删除，删除时SecretKeyValue、SecretKeyStartTime、SecretKeyExpirytime为可选，SecretKeyStartTime存在删除大于等于SecretKeyStartTime，SecretKeyExpirytime存在删除小于等于SecretKeyExpirytime

文章目录>>

- 请求示例

```
1. {
2.   "serialNo": "6w8keif5g6",
3.   "uuid": "0a1b2c3d",
```

```
4.  "data":{
5.      "syncFlag": 1,
6.      "secretKeys": [
7.          {
8.              "secretKeyType": "RSA",
9.              "secretKeyCode": "vguang",
10.             "secretKeyValue":
11.                 "MFwwDQYJKoZIhvcNAQEBBQADSwAwSAJBAlhGA5XLhPR22MRf7ms4R3NeUyV4UvnUiu2Y
12.                 "secretKeyStartTime": 1563456783,
13.                 "secretKeyExpirytime": 1963456783,
14.                 "syncType": 1
15.             }
16.         ],
17.         "time": 1647580466,
18.         "sign": ""
19.     }
```

2) 上行：设备接收到同步密钥指令后返回成功或失败给应用平台

- 说明:
- Topic: 20220814/cmd/secretSync_reply
- 内容: 标准格式
- data 部分:
- 返回示例:

文章目录

>>

```
1.  {
2.      "serialNo": "6w8keif5g6",
3.      "uuid": "0a1b2c3d",
4.      "code": "000000",
5.      "time": 1647580466,
6.      "sign": ""
7.  }
```

回到顶部

2.2 事件类

序号	接口名称	操作	备注
2.2.1	连接上报	详情	设备连上 MQTT 代理后会主动推送基本信息
2.2.2	告警上报	详情	各种门禁相关事件都会触发数据告警上报
2.2.3	通行上报	详情	门禁通行成功或失败会触发通行日志上报

序号	接口名称	操作	备注
2.2.4	设备上报	操作	重要的特别是错误的运行日志会触发运行日志上报，打开调试模式后，会上传更多的运行调试日志

2.2.1 连接上报

1) 上行:设备一旦连接到 MQTT 服务器后，立刻主动上报设备当前状态

- 说明: 只上报一次
- Topic: 20220814/event/connect
- 内容: 标准格式
- 内容里的 data 部分: json object 对象，参数如下:

参数名	说明	类型	必须	备注
time	设备当前时间	Long	是	时间
sysVersion	系统版本号	String	是	系统版本号
appVersion	应用版本号	String	是	应用版本号
createTime	应用发布日期	Long	是	应用发布日期
uuid	设备地址 (MAC)	String	是	设备地址 (MAC)
name	设备名称	String	是	设备名称
model	设备类型	String	是	设备类型

文章目录>>

- 请求示例

```
1. {
2.   "serialNo": "6w8keif5g6",
3.   "uuid": "0a1b2c3d",
4.   "data": {
5.     "time": 1647580466,
6.     "sysVersion": "v1.0.3.4050",
7.     "appVersion": "v3.204.234.5",
8.     "createTime": 1647550466,
9.     "uuid": "0a1b2c3d",
10.    "name": "#12-B01",
11.    "model": "dx_cc101"
12.  },
13.  "time": 1647580466,
14.  "sign": ""
15. }
```

2) 下行: 应用平台收到后反馈给设备

- 说明:
- Topic: 20220814/event/{#uuid}/connect_reply
- 内容: 标准格式
- data 部分: 无
- 返回示例:

```
1. {
2.   "serialNo": "6w8keif5g6",
3.   "uuid": "0a1b2c3d",
4.   "code": "000000",
5.   "time": 1647580466,
6.   "sign": ""
7. }
```

[回到顶部](#)

2.2.2 告警上报

1) 上行:设备运行期间，如果有告警，主动上报给应用平台

- 说明:
- Topic: 20220814/event/alarm
- 内容: 标准格式
- 内容里的 data 部分: json object 对象，参数如下:

参数名	说明	类型	必须	备注
type	状态类型	Int	是	参考附录-设备状态码表-type列
value	状态值	Int	否	参考附录-设备状态码表，如果是“ 无” ，就不需要传值

- 请求示例

```
1. {
2.   "serialNo": "6w8keif5g6",
3.   "uuid": "0a1b2c3d",
4.   "data": {
5.     "type": 1,
6.     "value": 1
7.   },
8.   "time": 1647580466,
9.   "sign": ""
10. }
```

文章目录

>>

2) 下行：应用平台收到后反馈给设备

- 说明:
- Topic: 20220814/event/{#uuid}/alarm_reply
- 内容: 标准格式
- data 部分: 无
- 返回示例:

```
1. {
2.   "serialNo": "6w8keif5g6",
3.   "uuid": "0a1b2c3d",
4.   "code": "000000",
5.   "time": 1647580466,
6.   "sign": ""
7. }
```

[回到顶部](#)

2.2.3 通行上报

1) 上行:设备运行期间，如果人通行门禁，通行结果主动上报给应用平台

- 说明:
- Topic: 20220814/event/access
- 内容: 标准格式
- 内容里的 data 部分: json array 数组对象，参数如下:

参数名	说明	类型	必须	备注
id	权限id	String	是	如果权限存在设备的白名单就返回对应的权限 不存在就返回" -1"
code	权限凭证	String	是	如果是人脸则是特征值
type	权限类型	Int	是	目前支持100-透传二维码，101-静态二维码， 103-动态二维码，202-卡，303-人脸，501-通行密码
time	通行时间	Long	是	时间戳单位秒
result	通行结果	Int	是	1表示通过 0表示未通过
extra	额外参数	Object	否	可扩展，人脸设备会传递人员名称、 人员实时照片、体温等信息，如果id为" -1" extra的值为识别出来的原始数据

文章目录

>>

参数名	说明	类型	必填	备注

• 请求示例

```
1. {
2.   "serialNo": "6w8keif5g6",
3.   "uuid": "0a1b2c3d",
4.   "data": [{
5.     "id": "20190101120101",
6.     "code": "s345463434gg",
7.     "type": 101,
8.     "time": 1640917147,
9.     "result": 0,
10.    "extra": ""
11.  }],
12.  "time": 1647580466,
13.  "sign": ""
14. }
```

2) 下行: 应用平台收到后反馈给设备

- 说明:
- Topic: 20220814/event/{#uuid}/access_reply
- 内容: 标准格式
- data 部分: 无
- 返回示例:

```
1. {
2.   "serialNo": "6w8keif5g6",
3.   "uuid": "0a1b2c3d",
4.   "code": "000000",
5.   "time": 1647580466,
6.   "sign": ""
7. }
```

回到顶部

2.2.4 日志上报

1) 上行:设备运行期间，如果有严重错误日志，主动上报给应用平台

- 说明: 如果开放了调试模式，所有日志都会上传到应用平台
- Topic: 20220814/event/log
- 内容: 标准格式
- 内容里的 data 部分: json object 对象，参数如下:

文
章
目
录

>>

参数名	说明	类型	必须	备注
level	日志级别	Int	是	0表示trace, 1表示info, 2表示debug, 3表示warn, 4表示error, 5表示fatal
time	日志时间	Long	是	10位时间戳 (秒)
log	日志内容	String	是	-

• 请求示例

```
1. {
2.   "serialNo": "6w8keif5g6",
3.   "uuid": "0a1b2c3d",
4.   "data": {
5.     "level": 4,
6.     "time":1647583466
7.     "log":"数据库错误xxxx"
8.   },
9.   "time": 1647580466,
10.  "sign": ""
11. }
```

文章目录
>>

2) 下行：应用平台收到后反馈给设备

- 说明:
- Topic: 20220814/event/{#uuid}/log_reply
- 内容: 标准格式
- data 部分: 无
- 返回示例:

```
1. {
2.   "serialNo": "6w8keif5g6",
3.   "uuid": "0a1b2c3d",
4.   "code": "000000",
5.   "time": 1647580466,
6.   "sign": ""
7. }
```

[回到顶部](#)

三. 附录

3.1 结果码

类型	code	信息	说明
成功	000000	成功执行指令	无
通用报错	100000	未知错误	无
通用报错	100001	设备已被禁用	设备禁用后很多指令不允许执行
通用报错	100002	设备正忙，请稍后再试	设备正在升级、 设置网络或其他任务进程中
通用报错	100003	签名检验失败	无
参数异常	200000- 299999	参数异常	不同的指令对应不同的参数规范， 参考每个接口后的参数异常描述
其它异常	300000- 399999	其它已知原因导致的异常	不同的指令对应不同的异常， 参考每个接口后的参数异常描述

[回到顶部](#)

3.2 设备告警类型表

类型	type	value
门磁状态	0	0：门磁关、1：门磁开
火警状态	1	0：火警关、1：火警开
非法开门	2	无
开门超时	3	无
关门超时	4	无
升级汇报	5	0：升级失败、1：升级成功完成
看门狗导致重启	6	1：看门狗导致重启

[回到顶部](#)

3.3 时间区间

支持下面四种时间区域设置：

- 1.缺省模式，不限时间，一直有效

1. {

文章目录

>>

```
2.   "type": 0
3. }
```

- 2.通常模式，从开始到结束时间内有效

```
1. {
2.   "type": 1,
3.   "beginTime": 1640917147,
4.   "endTime": 1640917147
5. }
```

- 3.每日模式，每日特定时间到特定时间内有效

```
1. {
2.   "type": 2,
3.   "beginTime": "08:30:00",
4.   "endTime": "18:30:00"
5. }
```

- 4.工作日模式，周一到周五特定时间到特定时间内有效，其它时间无效

```
1. {
2.   "type": 3,
3.   "beginTime": "08:30:00",
4.   "endTime": "18:30:00"
5. }
```

文章目录
>>

- 5.可用时区列表

可用时区
Asia/Shanghai 中国标准时间 (北京)
Asia/Tokyo 日本时间 (东京)
Europe/London 格林尼治标准时间 (伦敦)
Europe/Paris 英国时间 (巴黎)
Europe/Berlin 德国时间 (柏林)
America/New_York 美国东部时间 (纽约)

[回到顶部](#)

3.4 配置项

- `netInfo`：关于网络配置的信息体

参数名	说明	类型	可修改	备注
type	网络类型	Int	是	网络类型(0:以太网1:wifi)
ssid	Wifi名	String	是	wifi名
pwd	Wifi密码	String	是	wifi密码
dhcp	是否为DHCP模式	Int	是	是否为DHCP(1:静态, 2:动态)
ip	ip地址	String	是	ip地址
gateway	网关	String	是	网关
dns	DNS服务器	String	是	DNS服务器
subnetMask	子网掩码	String	是	子网掩码

- `mqttInfo` : 关于 mqtt 服务器配置的信息体

参数名	说明	类型	可修改	备注
mqttAddr	Mqtt服务器ip	String	是	Mqtt地址 (ip:端口)
port	Mqtt端口	String	是	Mqtt地址 (ip:端口)
clientId	客户端id, 与设备地址一致	String	是	客户端id, 与设备地址一致
mqttName	mqtt用户名	String	是	mqtt用户名
password	mqtt密码	String	是	mqtt密码

文章目录 >>

- `baseInfo` : 设备基础配置信息体

参数名	说明	类型	可修改	备注
time	时间	Long	是	时间
sysVersion	系统版本号	String	否	系统版本号
appVersion	应用版本号	String	否	应用版本号
createTime	应用发布日期	String	否	应用发布日期

参数名	说明	类型	可修改	备注
uuid	设备地址 (MAC)	String	否	设备地址 (MAC)
deviceName	设备名称	String	是	设备名称
model	设备类型	String	否	设备类型
status	设备基本状态	String	否	0-未激活, 1-正常, 2-禁用
operationMode	运行模式	String	否	设备运行模式 (不同模式切换时会清除之前模式下的所有数据记录, 包含权限白名单), 0: 互联网模式, 1: 局域网模式, 2: 单机版模式, 默认: 0
tamperSwitch	防拆开关	String	是	0: 关闭,1: 打开, 默认: 1
alarmSwitch	报警开关	String	是	0: 关闭, 1: 打开, 默认: 1
autoRbootTime	自动重启时间	Int	是	自动重启时间, 单位: 秒, 默认10s (即应用操作无响应, 等待时间超过10s自动重启)

文章目录 >>

- sysInfo : 设备系统资源信息体

参数名	说明	类型	可修改	备注
disk	设备剩余空间	Int	否	设备剩余空间 单位是 kb
memroy	内存剩余空间	Int	否	内存剩余空间 单位是 kb
cpu	cpu 占用比率	Int	否	CPU 占用比率 %

- doorInfo : 设备开门相关配置信息

参数名	说明	类型	可修改	备注
-----	----	----	-----	----

参数名	说明	类型	可修改	备注
-mode	设备开门模式	String	是	设备开门模式，0-正常开模式，2-常闭模式；默认：0
-openTime	开门时长，单位秒	String	是	开门时长，单位秒
-relayDelay	继电器延时时间，单位秒	Int	是	默认：2秒
-recMode	识别模式	Int	是	识别模式（1：人/卡人/码合一，3：人/卡码任一）
-scannerWorkMode	扫描器工作模式	Int	是	工作模式（针对刷卡透传模式，2：协议框开发模式，4：本地白名单模式）；当为开发模式时，faceWorkMode亦会开发模式；默认：>>
-faceWorkMode	人脸工作模式	Int	是	工作模式（针对刷脸透传模式，2：协议框开发模式，4：本地白名单模式）；注当为开发模式时，faceWorkMode亦会开发模式；默认：4
-qrEn	扫码开关	Int	是	0：关闭，1：打开；
-qrCheckInterval	扫码间隔	Int	是	扫码间隔 默认1000ms
-qrAffixesFt	前后缀格式	Int	是	二维码前后缀格式 1：字符串格式，2：十六进制格式；默认
-qrPrefix	前缀	String	是	二维码前缀，默认：（空）
-qrPostfix	后缀	String	是	二维码后缀，默认：（空）

参数名	说明开关	类型	可修改	备注关闭， 1： 打开；
-nfcFt	格式转换	Int	是	NFC格式转换， 0： 不转换； 1： 转16进制字符格式； 2： 转10进制字符输出； 默认： 0
-nfcOrd	正反序	Int	是	NFC数据正反序 0： 不反序； 默认： 0
-nfcAffixesFt	前后缀格式	Int	是	NFC前后缀格式， 1： 字符串格式， 2： 十六进制格式； 默认： 0
-nfcPrefix	前缀	String	是	NFC前缀， 默认： ""
-nfcPostfix	后缀	String	是	NFC后缀， 默认： ""
-nfcReadIdNum	身份证号	Int	是	0： 读取身份证号码； 1： 读取身份证号； 默认： 0
-doorSensorDelay	报警延时	Int	是	报警延时， 报警信号出发多久后报警； 单位秒， 默认： 30s
-doorSensor	触发类型	Int	是	报警信号触发类型， 0： 与地断开触发， 1： 与地短接触发， 2： 不触发（默认）
-alarmOutputSwitch	输出开关	Int	是	报警触发输出开关， 1： 报警时输出报警信号（默认）， 0： 关闭
-alarmRelay	报警继电器	Int	是	alarmOutputSwitch 1： 输出继电器， 0： 不输出继电器

- statusInfo：设备实时状态相关信息

参数名	说明	类型	可修改	备注
-door	门磁状态	Int	否	0： 门磁关、1： 门磁开

参数名	说明	类型	可修改	备注
-fire	火警状态	Int	否	0：火警关、1：火警开

- `temperatureInfo`：设备测温配置相关信息
| 参数名 | 说明 | 类型 | 可修改 | 备注 |
| --- | --- | --- | --- | --- |
| -alarmTemperature | 报警温度 | Float | 是 | 默认37.3 |
| -bodyTemperatureDetect | 测温开关 | String | 是 | 0：关闭，1：打开测温服务模块；默认：0 |
| -infraredEnable | 红外预览图 | String | 是 | 红外图像，0：关闭，1：开启；默认：1 |
| -temperatureCompensation | 温度补偿 | String | 是 | 温度补偿，默认：0 |
- `faceInfo`：设备人脸配置相关信息

参数名	说明	类型	可修改	备注
-recgThreshold	识别阈值	String	是	人脸相似度阈值，(0~100)；默认：80
-registerThreshold	权限阈值	String	是	权限阈值，人脸当人员相似度达
-living	活体检测	String	是	0：关闭活检，1：开启活检

-livingThreshold	活体阈值	String	是	活体检测阈值（0-100），活检算法每次判断并给出一个分数，就认为当前是活体活体检测阈值越大非活体攻击难度越大
-exposureValues	曝光补偿	String	是	曝光补偿，默认：0
-maxTrackFaces	人脸识别数	String	是	多人识别，人脸识别最大人数
-faceRecgEnable	识别开关	String	是	人脸识别：0：关闭，1：开启
-maskDetectEnable	口罩检测	String	是	口罩检测：0：关闭，1：开启

参数名	说明	类型	可修改	备注
-maskEnable	口罩识别	String	是	识别：0：关闭
-maskThresholdValue	口罩阈值	String	是	口罩阈值（0~100）
-recgDistance	识别距离	String	是	识别距离（50~150）
-recgInterval	识别间隔	String	是	重复识别的时间间隔需大于recgTimeout
-recgTimeout	识别超时	String	是	识别超时单位毫秒
-repeatRecgEnable	重复识别	String	是	重复识别开关，0：关闭
-strangerVoice	识别语音	String	是	陌生人语音对应比播放请先注册，bi
-alarmVoiceData	报警语音	String	是	alarmOutputSwi播放默认语音，"
-faceCheckSucVoice	语音类型	String	是	人脸核验成功语音播放默认语音，"
-faceCheckSucVoiceData	语音数据	String	是	人脸核验成功自定义语音数据
-faceCheckSucRelay	刷脸继电器	String	是	人脸核验成功继电器开关
-faceCheckFailedVoice	语音类型	String	是	人脸核验失败语音播放默认语音，2
-faceCheckFailedVoiceData	语音数据	String	是	人脸核验失败自定义语音数据

-cardCheckSucVoice	语音类型	String	是	刷卡核验成功语音播放默认语音，2
-cardCheckSucVoiceData	语音数据	String	是	刷卡核验成功自定义语音数据
-cardCheckSucRelay	继电器开关	String	是	刷卡核验成功继电器开关
-cardCheckFailedVoice	语音类型	String	是	刷卡核验失败语音播放默认语音，2
-cardCheckFailedVoiceData	语音数据	String	是	刷卡核验失败自定义语音数据
-qrCheckSucVoice	语音类型	String	是	扫码核验成功语音播放默认语音，2

参数名	说明	类型	可修改	备注
-qrCheckSucVoiceData	语音数据	String	是	扫码核验成功语音
-qrCheckSucRelay	继电器	String	是	扫码核验成功继电
-qrCheckFailedVoice	语音类型	String	是	扫码核验失败语音 播放默认语音， 2
-qrCheckFailedVoiceData	语音数据	String	是	扫码核验失败语音
-saveLargePic	全景图片	String	是	通行记录全景图片
-saveSmallPic	人脸图片	String	是	通行记录人脸图片
-saveStrangerRecord	陌生人记录	String	是	陌生人记录： 1： 1
-avoidAttendanceStep	保存间隔	String	是	通行记录保存间隔
-voiceMode	语音模式	String	是	语音模式对应比特 bit1播放问候语
-greetings	问候语	String	是	问候语对应比特 bit1： 播放请通 播放时间问候语
-infraredLightSet	补光灯	String	是	补灯光模式， 0
-lightSensitive	光敏控制	String	是	光敏控制（暂时没 0： 关闭
-lightHighBrightness	最高亮度	String	是	环境光高于此值关
-lightLowBrightness	最低亮度	String	是	环境光低于此值巨

-lightDetInterval	检测间隔	String	是	环境光检测间隔，
-lightDelayTime	补光灯延时	String	是	自动模式下补光灯 15s
-lightBrightness	亮度	String	是	补光灯亮度(0~10
-language	语言	String	是	0： 中文， 1： 英
-volume	音量	String	是	音量 （0-100） ，
-autoSetTime	时间更新	String	是	自动更新时间， 0
-autoSetTimezone	时区更新	String	是	自动设置时区， 0
				时区 默认： " A.

参数名	说明	类型	可修改	备注
-deviceName	设备名	String	是	默认：大门，任意
-deviceDirection	设备位置	String	是	设备位置，1：进
-nameShow	姓名	String	是	显示姓名，0：不
-snShow	设备SN	String	是	显示SN，0：不显
-ipShow	设备IP	String	是	显示IP，0：不显
-autoDisplayBrightness	屏幕亮度调节	String	是	自动调节屏幕亮度
-displayBrightness	屏幕亮度	String	是	屏幕亮度（0-100
-autoScreenOff	自动息屏	String	是	自动息屏，0：关
-autoScreenOffTime	息屏时间	String	是	自动息屏时间，默
-autoScreenSaverTime	自动屏保	String	是	自动屏保时间，默

- `security`：安全相关的密钥
可以自己设置 key, value,只可写不可读
- `custom`：自定义的一些属性
可以自己设置 key, value,可写可读
- `debug`：打开\关闭调试模式

参数名	说明	类型	可修改	备注
-enable	调试模式	Int	是	0-不打开，1-打开

[回到顶部](#)

文章目录>>