

会议固件Best协议接口

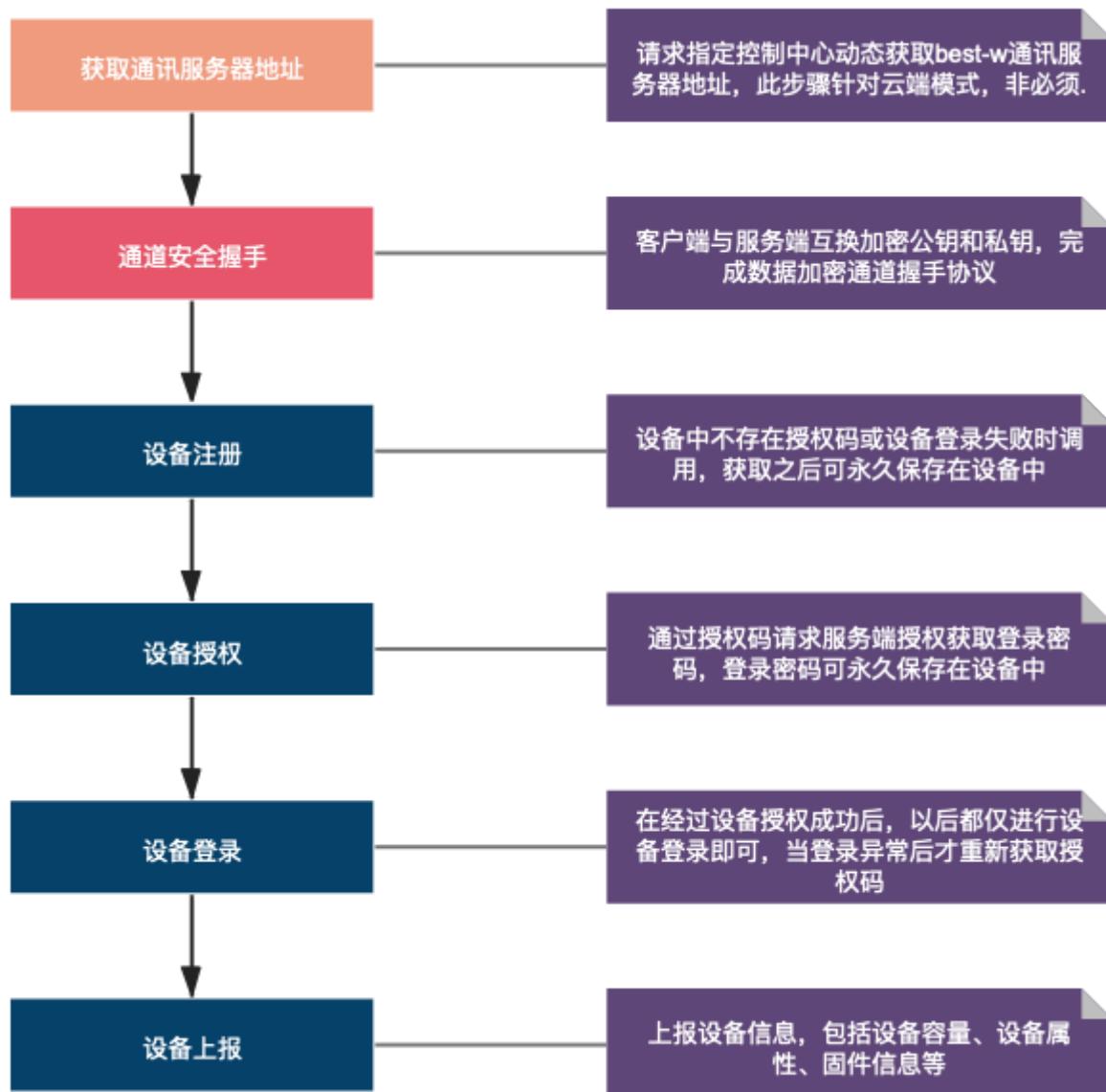
1 协议类型

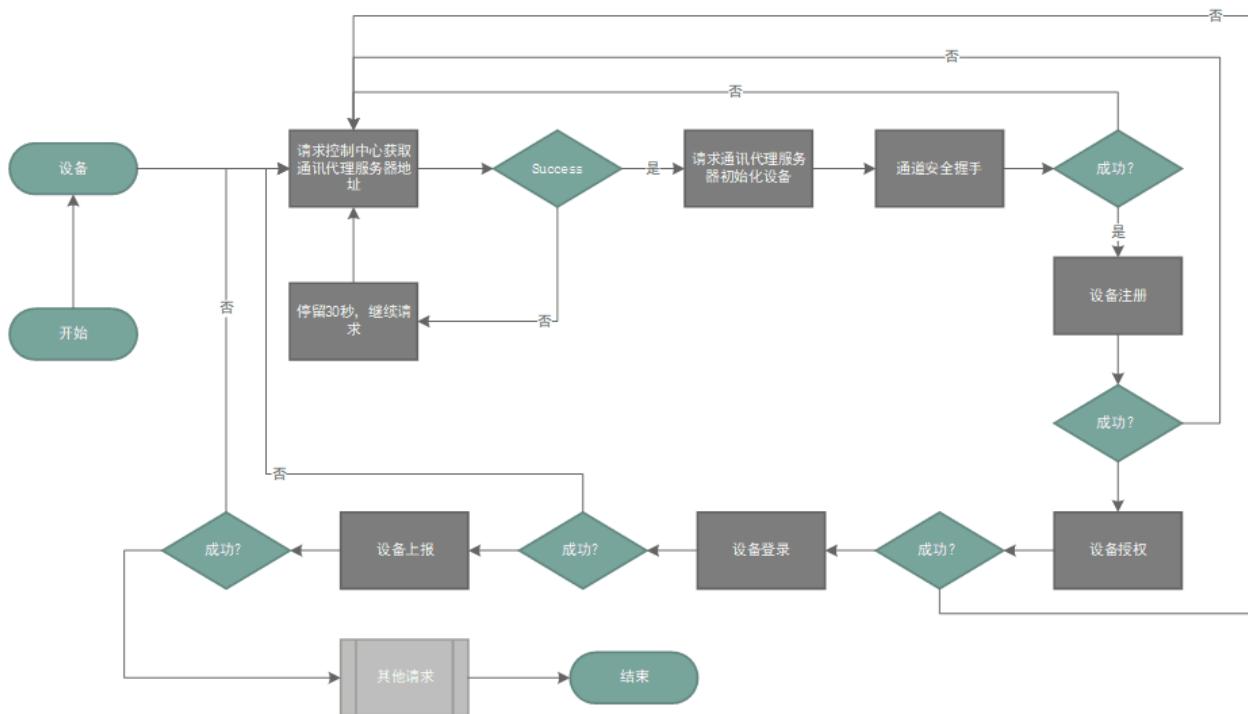
BEST 协议: 基于best-w为基准, 拓宽会议相关接口。

2 初始化流程

初始化流程步骤图解及流程图以best-w为基准进行阐述。

内容如下:





3业务协议接口

3.1 获取服务器地址

(1) 设备请求

URL:

<https://dc.zkclouds.com/v3/device/getBrokerServer?sn={SN}&type={TYPE}&nonce={NONCE}×tamp={timestamp}&sign={SIGN}>

Method: Get

参数	是否必须	描述
sn	是	设备序列号
type	是	通讯协议类型, 取值范围: push:表示获取push通讯地址; best-w:表示获取best over websocket 通讯服务器地址 best-t:表示获取best over tencent broker地址 best-m:表示获取best over mqtt best-a:表示获取best over aws iot core
nonce	是	随机数, 不得超过32位, 随机数不得重复
timestamp	是	时间戳, 精确到毫秒, 若时间戳与服务器时间戳误差较大(比如5分钟), 此请求就会被当作非法请求
sign	是	数字签名, {SN}+{nonce}+{timestamp}+{secretKey} 经32位小写md5摘要之后得到的签名, 如SN=abcd, nonce=123456, timestamp=1646102871000, secretKey=aabbcc, 则sign的值为字符串“abcd1234561646102871000aabbcc”的32位md5摘要内容, 即bdd58420c142a11569ff3eb8a9ec535c, 注意secretKey是设备生产时写入设备的, 不会直接明文传输

(2) 正常回复:

```
{
  "code": "00000000",
  "message": "success",
  "payload": {
    "results": {
      "url": "mqtts://xxxx:port",
      "rootCA": "string",
      "certificate": "string",
      "privateKey": "string",
      "publicKey": "string",
      "companyId": "string",
      "siteId": "string"
    }
  }
}
```

参数	必填	描述
url	是	通讯(代理)服务器地址
rootCA	否	通讯(代理)服务器CA根证书,pem格式文本
certificate	否	设备证书, 安全模式为证书模式时返回,pem格式文本
privateKey	否	私钥文件, 安全模式为证书模式时返回
publicKey	否	公钥文件, 安全模式为证书模式时返回
companyId	否	绑定该设备的公司id, BEST-A时必返回
siteId	否	设备所属区域/分公司id, BEST-A时必返回

(3) 异常回复:

设备没权限或服务异常时回复:

```
{
  "code": "其他不等于00000000的错误码",
  "message": "错误描述"
}
```

1. 通道安全握手**2.1 交换公钥****(1) 设备请求**

```
{
  "funcId": "device.exchange.publicKey",
  "mid": "xxxxxx",
  "payload": {
    "params": {
      "publicKey": "固件端公钥"
    }
  }
}

(2) 服务端回复
{
```

```
"code": "00000000",
"message": "消息内容",
"mid": "xxxxxxx",
"payload": {
  "results": {
    "publicKey": "服务端公钥"
  }
}
```

2.2 交换因子

(1) 设备请求

```
{
  "funcId": "device.exchange.factor",
  "mid": "xxxxxxx",
  "payload": {
    "params": {
      "factor": "固件端加密因子"
    }
  }
}
```

(2) 服务端回复

```
{
  "code": "00000000",
  "message": "消息内容",
  "mid": "xxxxxxx",
  "payload": {
    "results": {
      "factor": "服务端加密因子"
    }
  }
}
```

3. 2设备注册

(1) 设备请求

```
{
  "funcId": "dev.register",
  "sn": "xxxx",
  "mid": "xxxx",
  "payload": {
    "params": {
      "protType": "best-w",
      "protVer": "1.0.0",
      "productId": "xxxxxxx",
      "devKey": "xxxx",
      "devType": "acc"
    }
  }
}
```

参数描述：

参数名	是否必须	描述
sn	是	当使用发布、订阅网络协议时不需要此参数, 如best-m, 其他为必须, 如best-w
protType	是	协议名称, 取值: best-w: 基于主流websocket; best-m: 基于主流mqtt best-t: 基于腾讯IoT Hub
protVer	是	协议版本
productId	是	产品id, 由软件平台提供, 生产时写入设备, 用于一机一密安全校验。 当使用发布、订阅网络协议时不需要此参数, 如best-m, 其他为必须, 如best-w
devType	是	设备类型, 取值: att: 考勤 acc: 门禁 elev: 梯控 icc: 证卡 meet: 会议
devKey	是	设备密钥, 由软件平台提供, 生产时写入设备, 用于设备通讯一机一密安全校验, 此处传输的是加密的摘要, 摘要生成策略如下: 摘要算法: PBKDF2WithHmacSHA256, 密文长度: 32字节, 迭代次数: 1000, 盐值为 productId, 进行摘要时, 盐值转为字节数组进行摘要 摘要参考: 明文=123456, 盐值=5d9b11e1a3f448ae81f9f920e976d852, 摘要值=7f666475e1cea7d10964055343367701c2fb3a96f82e8b0adc25147992c8bb65

(2) 服务端响应

```
{
  "funcId": "cmd.dev.response",
  "code": "00000000",
  "message": "消息内容",
  "mid": "xxxxxxxx",
  "payload": {
    "results": {
      "authCode": "xxxxx"
    }
  }
}
```

参数名	是否必须	描述
authCode	是	授权码

3.3 设备授权

(1) 设备请求

```
{
  "funcId": "dev.accredit",
  "sn": "xxxx",
  "mid": "xxxx",
  "payload": {
```

```

"params": {
    "productId": "xxxxxxx",
    "devKey": "xxxx",
    "authCode": "xxxx"
}
}
}
}

```

(2) 服务端响应

```

{
    "funcId": "cmd.dev.response",
    "code": "00000000",
    "message": "消息内容",
    "mid": "xxxxxxxx",
    "payload": {
        "results": {
            "secret": "xxxxxxxx"
        }
    }
}

```

参数名	描述
productId	见 设备注册 参数说明
devKey	见 设备注册 参数说明
secret	设备登录密码

3.2 设备登录

(1) 设备请求

```

{
    "funcId": "dev.login",
    "sn": "xxxx",
    "mid": "xxxx",
    "payload": {
        "params": {
            "productId": "xxxxxxxx",
            "devKey": "xxxx",
            "authCode": "xxxx",
            "secret": "xxxx"
        }
    }
}

```

(2) 服务端回复

```

{
    "funcId": "cmd.dev.response",
    "code": "xxxxxxxx",
    "message": "消息内容",
    "mid": "xxxxxxxx"
}

```

3.5 设备上报

(1) 请求参数

```
{  
    "funcId": "dev.report",  
    "sn": "xxxx",  
    "mid": "xxxx",  
    "payload": {  
        "params": {  
            "fwVer": "ZAM170-NF-Ver1.1.21",  
            "devModel": "TDB08Z-PLUS",  
            "commType": "ethernet",  
            "ip": "192.168.213.55",  
            "netmask": "255.255.255.0",  
            "gateway": "192.168.213.1",  
            "mac": "00:17:61:12:e5:07",  
            "lockCount": "1",  
            "userCount": "50",  
            "maxUserCount": "30000",  
            "transactionCount": "115",  
            "maxTransactionCount": "50000",  
            "multiBioDataSupport": "0:1:1:0:0:0:0:0:0:0",  
            "multiBioPhotoSupport": "0:1:1:0:0:0:0:0:0:0",  
            "multiBioVer": "0:10.0:7.0:0:0:0:0:0:0:0",  
            "multiBioDataCount": "0:100:200:0:0:0:0:0:0:0",  
            "multiBioPhotoCount": "0:100:200:0:0:0:0:0:0:0",  
            "maxMultiBioDataCount": "0:10000:2000:0:0:0:0:0:0:0",  
            "maxMultiBioPhotoCount": "0:10000:2000:0:0:0:0:0:0:0",  
            "maxPackageSize": "64",  
            "qrCodeSupport": "1",  
            "qrCodeOn": "1",  
            "qrCodeDecryptFunList": "01",  
            "capturePic": "2",  
            "enableBioTest": "1",  
            "voiceOn": "1",  
            "keyPadBeepOn": "1",  
            "volume": "50",  
            "irTempDetectionOn": "1",  
            "maxTemperature": "37.00",  
            "minTemperature": "32.10",  
            "irTempUnit": "1",  
            "maskDetectionOn": "1",  
            "maskOn": "1",  
            "temperatureOn": "1",  
            "irTempDetectionFunOn": "1",  
            "cameraOn": "1",  
            "attPhotoFunOn": "1",  
            "maskDetectionFunOn": "1",  
            "maxPendingCmdCount": "20",  
            "scene": "1",  
            "verifyStyles": "100000000010",  
            "sipOn": "1",  
            "adOn": "1",  
            "funcIds": ["dev.option.upload", "dev.data.upload.user", "xxxxx"]  
        }  
    }  
}
```

}

请求参数详解:

参数名	是否必须	描述
fwVer	是	固件版本号
Sn	是	设备sn
devModel	是	设备型号
commType	是	通讯类型, 可有以下方式: ethernet:有线通信 usb-4G-modem:4G网络通信 serial-wireless:wifi(串口和Sdio)通信
ip	是	通讯类型对应网口的ip地址
netmask	是	通讯类型对应网口的子网掩码
gateway	是	通讯类型对应网口的网关地址
mac	是	通讯类型对应网口的mac地址
lockCount	是	门的数量
userCount	是	设备当前用户数
maxUserCount	是	设备用户容量
transactionCount	是	设备当前记录数
maxTransactionCount	是	设备最大记录容量
multiBioEnrollSupport	否	设备多模态生物特征远程登记功能支持, 生物识别类型见 附录 生物特征类型描述 进行按位定义, 不同类型间用冒号隔开, 0:不支持, 1:支持, 如:0:1:1:0:0:0:0:0:0, 表示支持指纹模板支持和近红外人脸远程登记功能支持
multiBioDataSupport	是	设备多模态生物特征模板支持, 生物识别类型见 附录 生物特征类型描述 进行按位定义, 不同类型间用冒号隔开, 0:不支持, 1:支持, 如:0:1:1:0:0:0:0:0:0, 表示支持指纹模板支持和近红外人脸模板支持
multiBioPhotoSupport	否	设备多模态生物特征图片支持, 生物识别类型见 附录 生物特征类型描述 进行按位定义, 不同类型间用冒号隔开, 0:不支持, 1:支持, 如:0:1:1:0:0:0:0:0:0, 表示支持指纹图片支持和近红外人脸图片支持
multiBioVer	是	设备多模态生物特征版本, 生物识别类型见 附录 生物特征类型描述 进行按位定义, 不同类型间用冒号隔开, 如:0:10.0:7.0:0:0:0:0:0:0, 表示支持指纹算法10.0和近红外人脸算法7.0
multiBioDataCount	是	设备当前多模态生物特征模板数量, 生物识别类型见 附录 生物特征类型描述 进行按位定义, 不同类型间用冒号隔开, 如:0:100:200:0:0:0:0:0:0, 表示支持指纹模板数量100和近红外人脸模板数量200

参数名	是否必须	描述
multiBioPhotoCount	否	设备当前多模态生物特征图片数量, 生物识别类型见 附录 生物特征类型描述 进行按位定义, 不同类型间用冒号隔开, 如: 0:100:200:0:0:0:0:0:0 , 表示支持指纹图片数量 100 和近红外人脸图片数量 200
maxMultiBioDataCount	是	设备最大多模态生物特征模板数量, 生物识别类型见 附录 生物特征类型描述 进行按位定义, 不同类型间用冒号隔开, 如: 0:10000:2000:0:0:0:0:0:0 , 表示支持指纹模板最大数量 10000 和近红外人脸模板最大数量 2000
maxMultiBioPhotoCount	否	设备最大多模态生物特征图片数量, 生物识别类型见 附录 生物特征类型描述 进行按位定义, 不同类型间用冒号隔开, 如: 0:10000:2000:0:0:0:0:0:0 , 表示支持指纹图片最大数量 10000 和近红外人脸图片最大数量 2000
maxPackageSize	否	通信最大数据包的大小, 默认值为 64 , 单位 KB
qrCodeSupport	否	二维码支持情况, 取值如下: 0 :不支持二维码 1 :支持二维码
qrCodeOn	否	二维码功能开关, 若设备支持二维码, 但未开启二维码功能, 也不可使用, 取值范围 0 :关闭; 1 :开启
qrCodeDecryptFunList	否	设备支持二维码解密方式; 从左往右数每一位分别代表是否支持方案一、方案二解密, 0 不支持, 1 支持; 例如 01 , 代表设备支持方案二解密方式; 具体方案细节参考 动态二维码方案
cameraOn	否	相机打开: 0 、关闭; 1 、开启
attPhotoFunOn	否	支持考勤照片, 1 : 支持, 0 : 不支持
capturePic	否	拍照模式。 0 、不拍照, 1 、拍照不保存, 2 、拍照并保存, 3 、验证成功保存, 4 、验证失败保存
enableBioTest	否	活体检测开关, 0 :关闭; 1 :开启
voiceOn	否	语音提示开关, 0 :关闭; 1 :开启
keyPadBeepOn	否	触屏提示开关, 0 :关闭; 1 :开启
volume	否	音量: 0-100
irTempDetectionFunOn	否	支持红外体温检测功能, 1 : 支持, 0 : 不支持
irTempDetectionOn	否	红外测温检测开关, 0 :关闭; 1 :开启
maxTemperature	否	体温高报警阈值
minTemperature	否	体温低报警阈值
temperatureOn	否	体温异常不允许通行开关, 0 :关闭; 1 :开启
irTempUnit	否	温度单位, 0 :摄氏度, 1 :华氏度
maskDetectionFunOn	否	支持口罩检测功能, 1 : 支持, 0 : 不支持
maskDetectionOn	否	口罩检测开关, 0 :关闭; 1 :开启

参数名	是否必须	描述
maskOn	否	未佩戴口罩不允许通行开关, 0:关闭;1:开启
maxPendingCmdCount	否	设备端支持的单批下发最大命令数
scene	否	场景模式, 取值范围: 1:标准门禁模式 2:普通模式, 此模式下人员识别时, 不用校验时间段、权限组等信息, 如人脸识别通过即可开门, 此时下面的funcIds不应包含时间段、权限组有关的所有funcId 场景模式一般应用于把门禁设备拿去做考勤场景, 或者考勤设备直接接入门禁协议时使用
verifyStyles	是	设备支持的验证方式, 见 附录 验证方式码描述
sipOn	否	是否支持sip, 0:不支持(默认);1:支持, 当设备第一次初始化(激活)
adOn	否	是否支持广告, 0:不支持(默认);1:支持
funcIds	是	设备端支持的funcIds列表, 增加funcId交换的背景:为了解决新增功能时, 设备同时支持老服务器和新服务器(服务器同时支持老设备和新设备), 需要增加功能参数来标识是否支持对应数据或功能, 统一funcId交换, 避免一直新增功能参数来做兼容, 其中dev.register、dev.accredit、dev.login、dev.report、cmd.dev.response、dev.cmd.response默认必须支持。

(2) 服务端响应

```
{
  "funcId": "cmd.dev.response",
  "code": "xxxx",
  "message": "xxxx",
  "mid": "xxxx",
  "payload": {
    "results": {
      "timezone": "+08:00",
      "supportHeartbeat": "0",
      "heartbeatInterval": "60",
      "multiBioDataSupport": "0:1:1:0:0:0:0:0:0",
      "multiBioPhotoSupport": "0:1:1:0:0:0:0:0:0",
      "funcIds": ["xxxx", "xxxx"]
    }
  }
}
```

响应参数详解:

参数名	是否必须	描述
timezone	否	设置设备的时区, 格式: $\pm HH:mm$, 默认值为+08:00
supportHeartbeat	否	是否支持设备业务心跳, 0:不支持, 1:支持, 默认值为1

参数名	是否必须	描述
heartbeatInterval	否	设备业务心跳间隔时间, 默认值为60, 单位为秒
multiBioDataSupport	是	设备多模态生物特征模板支持, 生物识别类型见 附录 生物特征类型描述 进行按位定义, 不同类型间用冒号隔开, 0:不支持, 1:支持, 如: 0:1:1:0:0:0:0:0:0, 表示支持指纹模板支持和近红外人脸模板支持
multiBioPhotoSupport	是	设备多模态生物特征图片支持, 生物识别类型见 附录 生物特征类型描述 进行按位定义, 不同类型间用冒号隔开, 0:不支持, 1:支持, 如: 0:1:1:0:0:0:0:0:0, 表示支持指纹图片支持和近红外人脸图片支持
funcIds	否	服务端支持的funcIds列表, 增加funcId交换的背景: 为了解决新增功能时, 设备同时支持老服务器和新服务器(服务器同时支持老设备和新设备), 需要增加功能参数来标识是否支持对应数据或功能, 统一funcId交换, 避免一直新增功能参数来做兼容, 其中dev.register、dev.accredit、dev.login、dev.report、cmd.dev.response、dev.cmd.response默认必须支持。

3.6 设备心跳

(1) 设备请求

```
{
  "funcId": "dev.heartbeat",
  "imme": "xxxx",
  "mid": "xxxx",
  "payload": {
    "params": {
      "isFree": "xxxx"
    }
  }
}
```

请求参数详解:

通用部分见[附录 协议通用描述](#)

参数名	是否必须	数据内容格式	描述
isFree	否	整型	是否处于空闲状态, 即没有在处理命令, 0:繁忙(设备有在处理命令), 1:空闲(设备没有在处理命令), 默认值为1

(2) 服务端响应

```
{
  "funcId": "cmd.dev.response",
  "code": "xxxx",
  "message": "xxxx",
  "imme": "xxxx",
  "mid": "xxxx"
}
```

3.7 上传大文件

上传开始:

(1) 请求参数

```
{
  "funcId": "dev.file.upload.start",
  "mid": "xxxx",
  "payload": {
    "params": {
      "fileId": "e78dc2cef23d495ab4f890fe3362bee9",
      "size": "1208",
      "checksum": ""
    }
  }
}
```

参数名	是否必须	描述
fileId	是	设备内文件唯一标识, 由字母和数字组成, 最高不能超过64位, 建议使用32位uuid
size	是	整包字节长度
checksum	否	总和校验码, 值为整包二进制字节md5摘要, 服务端根据checksum是否为空进行checksum校验文件完整性

(2) 服务端响应

```
{
  "funcId": "cmd.dev.response",
  "code": "xxxx",
  "message": "xxxx",
  "mid": "xxxx"
}
```

分包上传:

(1) 请求参数

```
{
  "funcId": "dev.file.upload.package",
  "mid": "xxxx",
  "payload": {
    "params": {
      "fileId": "e78dc2cef23d495ab4f890fe3362bee9",
      "offset": "215",
      "content": "xxxxxxxxxxxxxxxx"
    }
  }
}
```

参数名	是否必须	描述
fileId	是	设备内文件唯一标识, 由字母和数字组成, 最高不能超过64位, 建议使用32位uuid
offset	是	上次已传输文件偏移位置

参数名	是否必须	描述
content	是	文件分包内容, 读取二进制数据流base64加密

(2) 服务端响应

```
{
  "funcId": "cmd.dev.response",
  "code": "xxxx",
  "message": "xxxx",
  "mid": "xxxx"
}
```

上传完成:

(1) 请求参数

```
{
  "funcId": "dev.file.upload.end",
  "mid": "xxxx",
  "payload": {
    "params": {
      "fileId": "e78dc2cef23d495ab4f890fe3362bee9"
    }
  }
}
```

(2) 返回结果

```
{
  "funcId": "cmd.dev.response",
  "code": "xxxx",
  "message": "xxxx",
  "mid": "xxxx"
}
```

3.8 上传用户信息

(1) 请求参数

```
{
  "funcId": "dev.data.upload.user",
  "mid": "xxxx",
  "payload": {
    "params": {
      "users": [
        {
          "pin": "2956",
          "password": "fbfe82e76c06ea568e3301ab17a9d4677c1481b8a4f1ca137411e10153deabc3",
          "passwordSalt": "c09eb7463827cda245f3a398bd453c42",
          "group": "xxxx",
          "idNum": "xxxx",
          "startTime": "2020-01-01T00:00:00+08:00",
          "endTime": "2020-12-31T23:59:59+08:00",
          "firstName": "xxxx",
          "lastName": "xxxx",
          "funSwitch": "0",
          "disable": "0",
        }
      ]
    }
  }
}
```

```
"threatLevelGroupNum": "",  
"bioDatas": [  
    {  
        "no": "0",  
        "index": "0",  
        "valid": "1",  
        "duress": "0",  
        "type": "2",  
        "majorVer": "10",  
        "minorVer": "1",  
        "fmt": "0",  
        "tmp": "xxxxxxxxxxxxxxxxxxxxxx"  
    }]  
}]
```

请求参数详解:

参数名	是否必须	描述
pin	是	用户的工号
password	否	个人密码, 算法:PBKDF2WithHmacSHA256, 密文长度:32字节, 迭代次数:1000
passwordSalt	否	password加密的盐值
group	否	用户所属组, 用于多人组合开门
idNum	否	身份证号或者身份证物理卡号
startTime	否	用户有效期开始时间, ISO8601标准时间格式: yyyy-MM-ddTHH:mm:ss±HH:mm, (yyyy-MM-ddTHH:mm:ss):设备的本地时间, (HH:mm):偏移量
endTime	否	用户有效期结束时间, ISO8601标准时间格式: yyyy-MM-ddTHH:mm:ss±HH:mm, (yyyy-MM-ddTHH:mm:ss):设备的本地时间, (HH:mm):偏移量
firstName	是	用户姓名, 没有区分时使用firstName
lastName	否	用户姓名, 没有区分时使用firstName
funSwitch	是	功能开关, 0:没有附加功能(默认), 1:打开扩展锁驱动时长功能, 2:打开临时用户功能, 3:打开临时用户以及扩展锁驱动时长功能
disable	是	是否为禁止名单, 0:允许名单, 1:禁止名单
threatLevelGroupNum	否	威胁等级组
bioDatas	否	指纹、面部、手掌等生物信息

参数名	是否必须	描述
no	否	生物具体个体编号见 附录 生物具体个体编号 , 如果no没有明确定义, 则默认值为0
index	否	生物具体个体模板编号, 如1个手指存储多枚模板, 从0开始计算, 默认值为0
valid	否	是否有效标志, 0:无效, 1:有效, 默认值为1
duress	否	是否胁迫标志, 0:非胁迫, 1:胁迫, 默认值为0
type	否	如果bioDatas字段存在, 该字段必须存在, 生物识别类型见 附录 生物特征类型描述
majorVer	否	如果bioDatas字段存在, 该字段必须存在, 模板算法版本之主版本, 如:指纹算法版本10.3, 主版本是10, 副版本是3
minorVer	否	如果bioDatas字段存在, 该字段必须存在, 模板算法版本之副版本, 如:指纹算法版本10.3, 主版本是10, 副版本是3
fmt	否	如果bioDatas字段存在, 该字段必须存在, 模板格式, 如指纹有ZK\ISO\ANSI等格式, 0:ZK, 1:ISO, 2:ANSI
tmp	否	如果bioDatas字段存在, 该字段必须存在, 指纹、面部、手掌等生物模板

(2) 服务端响应

```
{
  "funcId": "cmd.dev.response",
  "code": "xxxx",
  "message": "xxxx",
  "mid": "xxxx"
}
```

3.9 上传用户照片（头像）

(1) 设备请求

```
{
  "funcId": "dev.data.upload.userPhoto",
  "mid": "xxxx",
  "payload": {
    "params": {
      "userPhotos": [
        {
          "pin": "xxxx",
          "extension": "xxxx",
          "content": "xxxx",
          "contentFileId": "xxxx"
        }
      ]
    }
  }
}
```

参数名	是否必须	数据内容格式	描述
pin	是	字符串	用户的工号
extension	否	字符串	图片的扩展名, extension没有明确定义, 则默认值为jpg
content	是	字符串	图片转化成base64的数据, 与contentFileDialog不能同时为空
contentFileDialog	是	字符串	生物特征图片内容id, 当上传照片数据包超过通讯协议限制的大小时使用, 将照片预先分包上传设备, 详见 上传大文件 , 然后指定fileId, 与content不能同时为空

(2) 服务器响应

```
{
  "funcId": "cmd.dev.response",
  "code": "xxxx",
  "message": "xxxx",
  "mid": "xxxx"
}
```

3. 10 上传用户生物特征图片

(1) 设备请求

```
{
  "funcId": "dev.data.upload.bioPhoto",
  "mid": "xxxx",
  "payload": {
    "params": {
      "bioPhotos": [
        {
          "pin": "xxxx",
          "no": "xxxx",
          "index": "xxxx",
          "type": "xxxx",
          "extension": "xxxx",
          "content": "xxxx",
          "contentFileDialog": "xxxx"
        }
      ]
    }
  }
}
```

参数名	是否必须	数据内容格式	描述
pin	是	字符串	用户的工号
no	否	整型	生物具体个体编号见 附录 生物具体个体编号 , 如果no没有明确定义, 则默认值为o
index	否	整型	生物具体个体模板编号, 如1个手指存储多枚模板, 从o开始计算, index 没有明确定义, 则默认值为o
type	是	整型	生物识别类型见 附录 生物特征类型描述
extension	否	字符串	图片的扩展名, extension没有明确定义, 则默认值为jpg
content	是	字符串	图片转化成base64的数据, 与contentFileDialog不能同时为空
contentFileDialog	是	字符串	生物特征图片内容id, 当上传照片数据包超过通讯协议限制的大小时使用, 将照片预先分包上传设备, 详见 上传大文件 , 然后指定fileId, 与content不能同时为空

(2) 服务器响应

```
{
  "funcId": "cmd.dev.response",
  "code": "xxxx",
  "message": "xxxx",
  "mid": "xxxx"
}
```

3.11 上传卡信息

(1) 设备请求

```
{
  "funcId": "dev.data.upload.multiCard",
  "mid": "xxxxxxxxxxxxxxxxxx",
  "payload": {
    "params": {
      "multiCards": [
        {
          "pin": "1001",
          "card": "CC11000001xxx",
          "type": "0",
          "lossFlag": "0"
        }
      ]
    }
  }
}
```

```

    }
}
}
```

参数名	是否必须	数据内容格式	描述
pin	是	字符串	用户的工号
card	是	字符串	卡号为10进制字符串数据
type	否	整型	主卡和副卡, 0:主卡(默认), 1:副卡
lossFlag	否	整型	丢失卡标记, 0:正常使用卡(默认), 1:丢失卡

(2) 服务器返回

```

{
  "funcId": "cmd.dev.response",
  "code": "xxxx",
  "message": "xxxx",
  "mid": "xxxx"
}
```

3. 12上传验证记录

(1) 设备请求

```

{
  "funcId": "dev.data.upload.verifyLog",
  "mid": "xxxx",
  "payload": {
    "params": {
      "verifyLog": [
        {
          "pin": "1",
          "meetCode": "123",
          "verifyTime": "2024-06-26T00:00:00+08:00",
          "verifyStyle": "1",
          "status": "0",
          "category": "1",
          "logId": "1",
          "recodeType": "1"
        }
      ]
    }
  }
}
```

请求参数详解:

通用部分见[附录 协议通用描述](#)

参数名	是否必须	数据内容格式	描述
pin	是	字符串	用户pin
meetCode	是	字符串	会议室编号
verifyTime	是	字符串	用验证时间
verifyStyle	是	整型	验证方式, 见 附录 验证方式码描述
status	是	整型	验证状态: 0:在最早签到时间和最晚签到时间期间签到 1:在最早签退时间和最晚签退时间内签退 255:默认值, 不是0和1则传255
category	是	整型	用户角色, // 普通人员NORMAL(0),// 访客VISITOR(4);
recodeType	是	整型	0:签到 1:签退 255默认值
logId	是	字符串	验证记录id, 用于和验证照片进行对应

(2)服务器响应

```
{
  "funcId": "cmd.dev.response",
  "code": "xxxx",
  "message": "xxxx",
  "mid": "xxxx"
}
```

3. 13上传验证照片

(1)设备请求

```
{
  "funcId": "dev.data.upload.verifyPhoto",
  "mid": "xxxx",
  "payload": {
    "params": {
      "verifyPhotos": [
        {
          "pin": "xxxx",
          "meetCode": "123",

```

```
        "photoTime": "2024-06-26T00:00:00+08:00",
        "extension": "xxxx",
        "content": "xxxx",
        "logId": "1",
        "contentFileId": "xxxx"
    }
]
}
}
```

请求参数详解:

通用部分见附录 协议通用描述

参数名	是否必须	数据内容格式	描述
pin	是	字符串	用户的工号
photoTime	否	字符串	用户验证拍照时间
meetCode	是	字符串	会议编号
extension	否	字符串	图片的扩展名, extension没有明确定义, 则默认值为jpg
content	是	字符串	图片转化成base64的数据, 与contentFileDialog不能同时为空
contentFileDialog	是	字符串	考勤图片内容id, 当上传照片数据包超过通讯协议限制的大小时使用, 将照片预先分包上传设备, 详见 上传大文件 , 然后指定fileId, 与content不能同时为空
logId	否	字符串	验证记录id, 用于和验证照片进行对应

(2) 服务器响应

```
{  
  "funcId": "cmd.dev.response",  
  "code": "xxxx",  
  "message": "xxxx",  
  "mid": "xxxx"  
}
```

3.14更新人员信息

(1)服务器下发:

```
{
  "funcId": "cmd.data.update.user",
  "mid": "xxxx",
  "payload": {
    "params": {
      "cmdId": "xxxx",
      "users": [
        {
          "pin": "2956",
          "role": "1",
          "password": "fbfe82e76c06ea568e3301ab17a9d4677c1481b8a4f1ca137411e10153deabc3",
          "passwordSalt": "c09eb7463827cda245f3a398bd453c42",
          "startTime": "2020-01-01T00:00:00+08:00",
          "endTime": "2020-12-31T23:59:59+08:00",
          "firstName": "huiwu",
          "lastName": "lin",
          "funSwitch": "0",
          "disable": "0",
          "bioDatas": [
            {
              "no": "0",
              "index": "0",
              "valid": "1",
              "duress": "0",
              "type": "2",
              "majorVer": "10",
              "minorVer": "1",
              "fmt": "0",
              "tmp": "xxxxxxxxxxxxxxxxxxxxxx"
            }
          ],
          "ruleType": "2"
        }
      ]
    }
  }
}
```

请求参数详解:

通用部分见[附录 协议通用描述](#)

参数名	是否必须	数据内容格式	描述
pin	是	字符串	用户的工号
role	是	字符串	用户角色

参数名	是否必须	数据内容格式	描述
cmdId	是	字符串	仅用于命令请求时以及命令断开时参数必须存在,服务端下发命令的唯一编号
password	否	字符串	个人密码, 算法:PBKDF2WithHmacSHA256, 密文长度:32字节, 迭代次数:1000 摘要参考:明文=123456, 盐值=5d9b11e1a3f448ae81f9f920e976d852, 摘要值 =486bbe9335aca3878178a51542e4e919d1146ao1bddof2ad5a68o43e2289abf7
passwordSalt	否	字符串	password加密的盐值, 进行摘要时, 盐视为hex(16进制) 转字节数组 进行摘要
startTime	否	字符串	用户有效期开始时间, ISO标准时间格式: yyyy-MM-ddTHH:mm:ss±HH:mm, (yyyy-MM-ddTHH:mm:ss):服务端的本地时间, (HH:mm):时区偏移量
endTime	否	字符串	用户有效期结束时间, ISO+标准时间格式: yyyy-MM-ddTHH:mm:ss±HH:mm, (yyyy-MM-ddTHH:mm:ss):服务端的本地时间, (HH:mm):时区偏移量
firstName	是	字符串	用户姓名, 没有区分时使用firstName
lastName	否	字符串	用户姓名, 没有区分时使用firstName
funSwitch	是	整型	功能开关, 0:没有附加功能, 1:打开扩展锁驱动时长功能, 2:打开临时用户功能, 3:打开临时用户以及扩展锁驱动时长功能
disable	是	整型	是否为禁止名单, 0:允许名单, 1:禁止名单
bioDatas	否	/	指纹、面部、手掌等生物信息
no	否	整型	生物具体个体编号
index	否	整型	生物具体个体模板编号, 如1个手指存储多枚模板, 从0开始计算, 默认值为0
valid	否	整型	是否有效标示, 0:无效, 1:有效, 默认值为1
duress	否	整型	是否胁迫标示, 0:非胁迫, 1:胁迫, 默认值为0
type	否	整型	如果bioDatas字段存在, 该字段必须存在, 生物识别类型见 附录 生物特征类型描述

参数名	是否必须	数据内容格式	描述
majorVer	否	整型	如果bioDatas字段存在, 该字段必须存在, 模板算法版本之主版本, 如:指纹算法版本10.3, 主版本是10, 副版本是3
minorVer	否	整型	如果bioDatas字段存在, 该字段必须存在, 模板算法版本之副版本, 如:指纹算法版本10.3, 主版本是10, 副版本是3
fmt	否	整型	如果bioDatas字段存在, 该字段必须存在, 模板格式, 如指纹有ZK\ISO\ANSI等格式, 0:ZK, 1:ISO, 2:ANSI
tmp	否	字符串	如果bioDatas字段存在, 该字段必须存在, 指纹、面部、手掌等生物模板
ruleType	否	整形	规则类别, 门禁事件时需要回传此标记 取值范围:1:考勤;2:门禁(默认);3:考勤+门禁

(2) 客户端回复

```
{
  "funcId": "dev.cmd.response",
  "mid": "xxxx",
  "code": "xxxx",
  "message": "xxxx",
  "payload": {
    "params": {
      "cmdId": "xxxx"
    }
  }
}
```

3. 15更新会议信息

(1) 服务端下发

```
{
  "funcId": "cmd.data.update.meeting",
  "mid": "xxxx",
  "payload": {
    "params": {
      "cmdId": "xxxx",
      "meeting": [
        {
          "name": "会议名称",
          "startSignInTime": "2020-12-31T23:59:59+08:00",
          "lastSignInTime": "2020-12-31T23:59:59+08:00",
          "earSignOutTime": "2020-12-31T23:59:59+08:00",
          "lastSignOutTime": "2020-12-31T23:59:59+08:00",
          "id": "123",
          "startTime": "2020-12-31T23:59:59+08:00",
          "endTime": "2020-12-31T23:59:59+08:00",
          "applicant": "会议申请人"
        }
      ]
    }
  }
}
```

```
    }  
}
```

请求参数详解：

通用部分见附录 协议通用描述

参数名	是否必须	数据内容格式	描述
name	是	字符串	会议名称
cmdId	是	字符串	仅用于命令请求时以及命令回复时参数必须存在,服务端下发命令的唯一编号
startSignInTime	否	字符串	会议开始签到时间
lastSignInTime	否	字符串	会议最晚签到时间
earSignOutTime	否	字符串	会议最早签退时间
lastSignOutTime	是	字符串	会议最晚签退时间
id	是	字符串	会议id,此为会议信息唯一值
startTime	是	字符串	会议开始时间
endTime	是	字符串	会议结束时间
applicant	是	字符串	会议申请人

(2) 客户端回复

{

"funcId": "dev.cmd.response",

```

"mid": "xxxx",
"code": "xxxx",
"message": "xxxx",
"payload": {
  "params": {
    "cmdId": "xxxx"
  }
}
}
}

```

3. 16更新会议人员信息

(1) 服务端下发

```

{
  "funcId": "cmd.data.update.meetingMember",
  "mid": "xxxx",
  "payload": {
    "params": {
      "cmdId": "xxxx",
      "members": [
        {
          "meetingId": "123",
          "pins": ["xxx", "xxx"]
        }
      ]
    }
  }
}

```

请求参数详解:

通用部分见[附录 协议通用描述](#)

参数名	是否必须	数据内容格式	描述
cmdId	是	字符串	仅用于命令请求时以及命令断句时参数必须存在, 服务端下发命令的唯一编号
pins	是	数组内每个数据内容格式为字符串	用户工号集合
meetingId	是	字符串	会议id

(2) 客户端回复

```

{
  "funcId": "dev.cmd.response",
  "mid": "xxxx",
  "code": "xxxx",
  "message": "xxxx",
  "payload": {
    "params": {
      "cmdId": "xxxx"
    }
}
}

```

```

    }
}
```

3.17更新卡信息

(1) 服务端下发

```

{
  "funcId": "cmd.data.update.multiCard",
  "mid": "xxxx",
  "payload": {
    "params": {
      "cmdId": "xxxx",
      "multiCards": [
        {
          "pin": "2956",
          "card": "C100001",
          "type": "0",
          "lossFlag": "0"
        }
      ]
    }
  }
}
```

请求参数详解:

通用部分见[附录 协议通用描述](#)

参数名	是否必须	数据内容格式	描述
cmdId	是	字符串	仅用于命令请求时以及命仔回复时参数必须存在,服务端下发命令的唯一编号
pin	是	字符串	工号
card	是	字符串	卡号,卡号为10进制字符串数据
type	是	整型	主卡和副卡, 0:主卡, 1:副卡
lossFlag	是	整型	丢失卡标记, 0:正常使用卡, 1:丢失卡

(2) 客户端回复

```

{
  "funcId": "dev.cmd.response",
  "mid": "xxxx",
  "code": "xxxx",
  "message": "xxxx",
  "payload": {
    "params": {
      "cmdId": "xxxx"
    }
  }
}
```

3.18更新人员生物特征照片

(1) 服务端下发

```
{
}
```

```

"funcId": "cmd.data.update.bioPhoto",
"mid": "xxxx",
"payload": {
    "params": {
        "cmdId": "xxxx",
        "bioPhotos": [
            {
                "pin": "2956",
                "no": "0",
                "index": "0",
                "type": "9",
                "url": "https://xxxxxxxxxx.jpg"
            }
        ]
    }
}
}

```

请求参数详解:

通用部分见[附录 协议通用描述](#)

参数名	是否必须	数据内容格式	描述
cmdId	是	字符串	仅用于命令请求时以及命令回复时参数必须存在, 服务端下发命令的唯一编号
pin	是	字符串	用户的工号
no	否	整型	生物具体个体编号见 附录 生物具体个体编号 , 如果no没有明确定义, 则默认值为0
index	否	整型	生物具体个体模板编号, 如1个手指存储多枚模板, 从0开始计算, 默认值为0
type	是	整型	生物识别类型见 附录 生物特征类型描述
url	是	字符串	指纹、面部、手掌图片地址, 图片格式jpg

(2) 客户端回复

```

{
    "funcId": "dev.cmd.response",
    "mid": "xxxx",
    "code": "xxxx",
    "message": "xxxx",
    "payload": {
        "params": {
            "cmdId": "xxxx"
        }
    }
}

```

3.19 删除用户信息

(1) 服务端下发

```

{
    "funcId": "cmd.data.delete.user",
    "mid": "xxxx",
    "payload": {

```

```

"params": {
    "cmdId": "xxxx",
    "pins": ["xxxx", "xxxx", "xxxx"]
}
}
}
}

```

请求参数详解:

通用部分见[附录 协议通用描述](#)

参数名	是否必须	数据内容格式	描述
cmdId	是	字符串	仅用于命令请求时以及命令断开时参数必须存在, 服务端下发命令的唯一编号
pins	是	数组内每个数据内容格式为字符串	用户工号集合

(2) 客户端响应

```

{
    "funcId": "dev.cmd.response",
    "mid": "xxxx",
    "code": "xxxx",
    "message": "xxxx",
    "payload": {
        "params": {
            "cmdId": "xxxx"
        }
    }
}

```

3. 20删除用户生物特征图片

(1) 服务端下发

```

{
    "funcId": "cmd.data.delete.bioPhoto",
    "mid": "xxxx",
    "payload": {
        "params": {
            "cmdId": "xxxx",
            "bioPhotos": [
                {
                    "type": "xxxx",
                    "pin": "xxxx",
                    "no": "xxxx"
                }
            ]
        }
    }
}

```

请求参数详解:

通用部分见[附录 协议通用描述](#)

参数名	是否必须	数据内容格式	描述
cmdId	是	字符串	仅用于命令请求时以及命令断回复时参数必须存在, 服务端下发命令的唯一编号
type	否	整型	生物识别类型见 附录 生物特征类型描述 没有类别, 默认是全部
pin	否	字符串	用户工号, 没有工号默认是所有的工号
no	否	整型	生物具体个体编号见 附录 生物具体个体编号 , 如果no没有明确定义, 默认值为0

(2) 客户端响应

```
{
  "funcId": "dev.cmd.response",
  "mid": "xxxx",
  "code": "xxxx",
  "message": "xxxx",
  "payload": {
    "params": {
      "cmdId": "xxxx"
    }
  }
}
```

3.21删除卡信息

(1) 服务端下发

```
{
  "funcId": "cmd.data.delete.multiCard",
  "mid": "xxxx",
  "payload": {
    "params": {
      "cmdId": "xxxx",
      "pins": ["xxxx", "xxxx", "xxxx"]
    }
  }
}
```

请求参数详解:

通用部分见[附录 协议通用描述](#)

参数名	是否必须	数据内容格式	描述
cmdId	是	字符串	仅用于命令请求时以及命令断回复时参数必须存在, 服务端下发命令的唯一编号
pins	是	数组内每个数据内容格式为字符串	用户工号集合

(2) 客户端响应

{

```

"funcId": "dev.cmd.response",
"mid": "xxxx",
"code": "xxxx",
"message": "xxxx",
"payload": {
    "params": {
        "cmdId": "xxxx"
    }
}
}

```

3. 22删除会议信息

(1) 服务器下发

```

{
    "funcId": "cmd.data.delete.meeting",
    "mid": "xxxx",
    "payload": {
        "params": {
            "cmdId": "xxxx",
            "meetings": ["xxxx", "xxxx", "xxxx"]
        }
    }
}

```

请求参数详解:

通用部分见[附录 协议通用描述](#)

参数名	是否必须	数据内容格式	描述
cmdId	是	字符串	仅用于命令请求时以及命仔回复时参数必须存在, 服务端下发命令的唯一编号
meetings	是	数组内每个数据内容格式为字符串	会议id集合

(2) 客户端响应

```

{
    "funcId": "dev.cmd.response",
    "mid": "xxxx",
    "code": "xxxx",
    "message": "xxxx",
    "payload": {
        "params": {
            "cmdId": "xxxx"
        }
    }
}

```

3. 23删除会议人员

(1) 服务端下发

```

{
    "funcId": "cmd.data.delete.meetingMember",
}

```

```

"mid": "xxxx",
"payload": {
  "params": {
    "cmdId": "xxxx",
    "meetingId": "xxxx",
    "pins": ["xxxx", "xxxx", "xxxx"]
  }
}
}

```

请求参数详解:

通用部分见[附录 协议通用描述](#)

参数名	是否必须	数据内容格式	描述
cmdId	是	字符串	仅用于命令请求时以及命仔回复时参数必须存在,服务端下发命令的唯一编号
meetingId	是	字符串	会议id
pins	是	数组内每个数据内容格式为字符串	人员pin集合

(2) 客户端响应

```

{
  "funcId": "dev.cmd.response",
  "mid": "xxxx",
  "code": "xxxx",
  "message": "xxxx",
  "payload": {
    "params": {
      "cmdId": "xxxx"
    }
  }
}

```

3.24删除用户照片（头像）

(1) 服务端下发

```

{
  "funcId": "cmd.data.delete.userPhoto",
  "mid": "xxxx",
  "payload": {
    "params": {
      "cmdId": "xxxx",
      "pins": ["xxxx", "xxxx", "xxxx"]
    }
  }
}

```

请求参数详解:

通用部分见[附录 协议通用描述](#)

参数名	是否必须	数据内容格式	描述
cmdId	是	字符串	仅用于命令请求时以及命令断句时参数必须存在, 服务端下发命令的唯一编号
pins	是	数组内每个数据内容格式为字符串	人员pin集合

(2) 客户端响应

```
{
  "funcId": "dev.cmd.response",
  "mid": "xxxx",
  "code": "xxxx",
  "message": "xxxx",
  "payload": {
    "params": {
      "cmdId": "xxxx"
    }
  }
}
```

3.25 查询用户信息

(1) 服务端下发

```
{
  "funcId": "cmd.data.query.user",
  "imme": "xxxx",
  "mid": "xxxx",
  "payload": {
    "params": {
      "cmdId": "xxxx",
      "pin": "xxxx"
    }
  }
}
```

请求参数详解：

参数名	是否必须	数据内容格式	描述
cmdId	是	字符串	仅用于命令请求时以及命令断句时参数必须存在, 服务端下发命令的唯一编号
pin	否	字符串	人员工号, 为空时默认查询所有人员

通用部分见[附录 协议通用描述](#)

(2) 客户端响应

```
{
  "funcId": "dev.cmd.response",
  "mid": "xxxx",
  "code": "xxxx",
  "message": "xxxx",
  "payload": {
```

```
"params": {  
    "cmdId": "xxxx"  
}  
}  
}
```

2. 清除用户信息

(1) 服务端下发

```
{  
    "funcId": "cmd.data.clear.user",  
    "mid": "xxxx",  
    "payload": {  
        "params": {  
            "cmdId": "xxxx"  
        }  
    }  
}  
  
(2) 客户端响应  
{  
    "funcId": "dev.cmd.response",  
    "mid": "xxxx",  
    "code": "xxxx",  
    "message": "xxxx",  
    "payload": {  
        "params": {  
            "cmdId": "xxxx"  
        }  
    }  
}
```

3. 26清除卡信息

(1) 服务端下发

```
{  
    "funcId": "cmd.data.clear.multiCard",  
    "mid": "xxxx",  
    "payload": {  
        "params": {  
            "cmdId": "xxxx"  
        }  
    }  
}
```

(2) 客户端响应

```
{  
    "funcId": "dev.cmd.response",  
    "mid": "xxxx",  
    "code": "xxxx",  
    "message": "xxxx",  
    "payload": {  
        "params": {  
            "cmdId": "xxxx"  
        }  
    }  
}
```

3.27 清除全部数据

(1) 服务端下发

```
{
  "funcId": "cmd.data.clear.all",
  "mid": "xxxx",
  "payload": {
    "params": {
      "cmdId": "xxxx"
    }
  }
}
```

(2) 客户端响应

```
{
  "funcId": "dev.cmd.response",
  "mid": "xxxx",
  "code": "xxxx",
  "message": "xxxx",
  "payload": {
    "params": {
      "cmdId": "xxxx"
    }
  }
}
```

3.28 设置设备参数（会议室相关）

(1) 服务端下发

```
{
  "funcId": "cmd.option.set",
  "imme": "xxxx",
  "mid": "xxxx",
  "payload": {
    "params": {
      "cmdId": "xxxx",
      "option": {
        "meetRoomName": "1",
        "meetRoomManager": "2",
        "meetRoomLocation": "3",
        "meetRoomCapacity": "4",
        "meetRoomManagerPin": "123"
      }
    }
  }
}
```

参数名	是否必须	数据内容格式	描述
imme	否	整型	是否紧急, 1:紧急, 0:不紧急, 参数不存在/参数值为空/参数值为0时, 则消息是不紧急消息, 否则为紧急消息

参数名	是否必须	数据内容格式	描述
mid	是	字符串	消息id, 消息唯一标识, 保障同一台设备不重复即可, 最长为32位
meetRoomName	是	字符串	会议室名称
meetRoomManager	否	字符串	会议室负责人
meetRoomLocation	否	字符串	会议室位置
meetRoomCapacity	是	字符串	会议室容量

(2) 客户端响应

```
{
  "funcId": "dev.cmd.response",
  "mid": "xxxx",
  "code": "xxxx",
  "message": "xxxx",
  "payload": {
    "params": {
      "cmdId": "xxxx"
    }
  }
}
```

3.29 重启设备

服务端请求:

```
{
  "funcId": "cmd.control.reboot",
  "imme": "1",
  "mid": "xxxx",
  "payload": {
    "params": {
      "cmdId": "1001",
      "operator": "luther"
    }
  }
}
```

3.30 同步时间

服务端下发同步时间命令后, 设备收到此命令, 再调用新的请求去获取服务器时间

服务端下发:

```
{
  "funcId": "cmd.system.sync.time",
  "imme": "xxxx",
  "mid": "xxxx",
  "payload": {
```

```

  "params": {
    "cmdId": "xxxx"
  }
}
}
}

```

设备端请求:

```

{
  "funcId": "dev.system.sync.time",
  "imme": "xxxx",
  "mid": "xxxx"
}

```

附录:**4. 协议通用描述**

为了协议统一, 约定如下:

1. 所有字段值都是以字符串表示
2. 已存在字段名不能删除, 只能扩展
3. 字段名以及funcId名, 驼峰式的, 首字母小写
4. 字段为数组的情况下, 字段名需要为复数, 加字母s

4. 1 请求业务功能格式

绑定流程请求以及心跳请求格式固定, 其他请求数据格式要求如下:

funcId: xxxx.xxxx.xxxx.xxxx

- 第一个xxxx, 请求类型, cmd表示命令下发, dev表示设备请求
- 第二个xxxx, 数据大类, data表示基础数据, option表示参数数据, firmware表示固件数据, system表示系统数据, cmd表示命令数据, control表示控制数据,
- 第三个xxxx, 操作类型, delete表示删除, upload表示上传, update表示更新, get表示获取, set表示设置, upgrade表示升级, sync表示同步, notify表示通知,
- 第四个xxxx, 数据小类, user表示用户数据, info表示信息, time表示时间,

4. 2 请求协议**4. 2. 1 服务端请求****协议格式**

```
{
  "funcId": "cmd.xxxx",      "imme": "1",      "mid": "xxxx",      "payload": {          "params": {
    "cmdId": "xxxx",           ...     }      }  }}
```

字段说明

参数名	是否必须	数据内容格式	描述
funcId	是	字符串	请求业务功能唯一标识
imme	否	整型	是否紧急, 1:紧急, 0:不紧急, 参数不存在/参数值为空/参数值为0时, 则消息是不紧急消息, 否则为紧急消息
mid	是	字符串	消息id, 消息唯一标识, 保障同一台设备不重复即可, 最长为32位

参数名	是否必须	数据内容格式	描述
cmdId	是	字符串	仅用于命令请求时以及命裔回复时参数必须存在,服务端下发命令的唯一编号
params	是	JSON	请求内容;设备要允许json中存在冗余参数,只处理对应业务所需的参数即可

4. 2. 2设备请求(设备主动请求以及命裔回复请求)

协议格式

```
{
  "funcId": "dev.xxxx",      "imme": "1",      "sn": "xxxx",      "mid": "xxxx",      "payload": {
  "params": {                "cmdId": "xxxx",           ... ...     }   }}
```

字段说明

参数名	是否必须	数据内容格式	描述
funcId	是	字符串	请求业务功能唯一标识
imme	否	整型	是否紧急, 1:紧急, 0:不紧急, 参数不存在/参数值为空/参数值为0时, 则消息是不紧急消息, 否则为紧急消息
sn	否	字符串	仅用于登入过程请求时参数必须存在, 设备序列号
mid	是	字符串	消息id, 消息唯一标识, 保障同一台设备不重复即可, 最长为32位
cmdId	否	字符串	仅用于命令请求时以及命裔回复时参数必须存在, 服务端下发命令的唯一编号

4. 3回复协议格式

4. 3. 1服务端回复

协议格式

```
{
  "funcId": "cmd.dev.response",      "code": "xxxx",      "message": "xxxx",      "imme": "xxxx",      "mid": "xxxx",
  "payload": {                      "results": {           ... ...       }}}
```

字段说明

参数名	是否必须	数据内容格式	描述
code	是	字符串	返回错误码
message	是	字符串	返回错误码对应的消息内容
imme	否	整型	是否紧急, 1:紧急, 0:不紧急, 参数不存在/参数值为空/参数值为0时, 则消息是不紧急消息, 否则为紧急消息
mid	是	字符串	消息id, 消息唯一标识, 保障同一台设备不重复即可, 最长为32位

