云考勤

WEBSOCKET+JSON

通讯协议

修订历史记录

日期	版本	说明	作者
2016-03-25	1.0	初始定义	邹春庆
2016-04-18	1.1	1、Senduser,getuserinfo,setfp,setcard,setpwd 增加用户名字项 2、增加 deleteuserlock 和 cleanuserlock	邹春庆
2016-04-21	1.2	1、增加 setuserinfo 统一下发信息指令 2、删除 setfp,setpwd,setcard 指令 3、修改 deleteuser 中的注释，增加备份号为 12 ，13 时，删除全部指纹，和删除整个人 4、修改 getuserlist, getnewlog,getalllog，当数据为空时，返回成功的空记录	邹春庆
2016-05-17	1.3	1、增加 reboot 指令	邹春庆
2016-05-25	1.4	增加 settime 指令	邹春庆
2016-07-06	1.5	增加考勤/门禁记录定义说明	邹春庆
2017-11-06	1.7	1.增加使用禁止用户功能， 2、增加上传记录时，返回开门或者关门权限	邹春庆

目录

说明：

1、采用 WEBSOCKET 协议通讯，版本 RFC6455　13，默认端口为 7788,无 TLS 加密。
2、数据格式严格按照 JSON 标准，可以使用 javascript 快速序列化和反序列化。
3、所有键值使用小写英文，名字或者中文信息使用 UTF8 编码。

一、考勤机主动发起的请求

1. 登录注册云服务

考勤机发送登录注册格式:

```
{
  "cmd":"reg", //通讯命令字
  "sn",”ZX0006827500", //考勤机序列号，出厂时固定，全球唯一
  "devinfo",{
            "modelname":"tfs30", //考勤机型号
            “usersize”:3000, //所有用户容量 1000/3000/5000
            "fpsize":3000, //指纹容量 1000/3000/5000
            "cardsize":3000，//感应卡容量 1000/3000/5000/10000
            "pwdsize":3000, //密码容量
            "logsize":100000, //考勤记录容量
            “useduser”:1000, //已存用户数
            "usedfp":1000, //已存指纹数
            "usedcard":2000, //已存感应卡数
            "usedpwd":400, //已存密码数
            "usedlog":100000, //已存考勤记录数
            “usednewlog”:5000 ,//已存新记录数
            "fpalgo":"thbio3.0", //指纹算法版本 thbio1.0 或者 thbio3.0
            "firmware":"th200w v5.0",//考勤机固件版本
            "time":"2016-03-25 13:49:30"//考勤机当时时间

            }
}
```

服务器回复信息：
成功：

```
{
  "ret":"reg", //回复的命令字
  "result":true, //结果
  "cloudtime":"2016-03-25 13:49:30”//服务器时间
}
```

失败：

```
{
  "ret":"reg",
```

```
  "result":false,
  "reason":1
}
```

2. 上传考勤记录

考勤机发送格式：

```
{
 "cmd":"sendlog",
 "count":2,
  “sn”:”AI05_MIC_001”,
  "logindex":0,
 "record":[
                {
                "enrollid":1,
                 "name":"zzz",
                "time":"2016-03-25 13:49:30",
                "mode":0, //0 fp 1:card 2:pwd
                "inout":0, //0 in 1:out
                "event":0,
                 "verifymode":13,//just AI device support.qrcode verify
                 "image":"/9j/4AAQSkZJRgABAQAAAQABAAD”
                }
                {
                "enrollid":2,
                 "name":"xyz",
                “time ":"2016-03-25 13:49:30",
                "mode":0, //0 fp 1:card 2:pwd
                "inout":0, //0 in 1:out
                "event":1,
                 "verifymode":13,//just AI device support.qrcode verify
                 "image":"/9j/4AAQSkZJRgABAQAAAQABAAD”
                }
 ]
}
```

服务器回复信息：

成功：

```
{
  "ret":"sendlog",
  "result":true,
  "count":1,
  "logindex":0,
  "cloudtime":"2016-03-25 13:49:30",
  “access”:1   //扩展功能，用于指示这个用户是否可以进门。1，可以进门，0 不可以进门，用于实时监控并由服务器决定该用户是否有权限开门，需要把机器的服务器模式功能打开
}
```

失败：

```
{
  "ret":"sendlog",
  "result":false,
  "reason":1
}
```

说明：关于考勤记录说明：

1、 当 enrollid 不等于 0 时，记录的是某个人的考勤/出入时间，定义的事件状态。

Mode ： 0 fp 1:card 2:pwd ; 为 0 的时候，表示这个人是通过按指纹验证通过的。

为 1 的时候，表示这个人是刷卡通过的。

Inout ： 为 0 的时候，表示这个人是在主机上验证通过的。

为 1 的时候，表示这个人是在子机上（机器可以外接一个子机，实现子母机，作门禁时，一般子机安装在外面，主机安装在里面）

Event：用户自定义动作（对应某些机型上的 F1~F4），例如可以自定义按 F1 表示上班，F2 表示下班，按 F3 表示加班上班，按 F4 表示加班下班，需要结合软件处理。

2、 当 enrollid 等于 0 的时候。是作为门禁机的，指示机器的门状态信息。

```
  "mode":0, //默认为 0
  "inout":0, //默认为 0
  "event": 0~7.定义如下：
        typedef enum
  {
      UI_MGLOG_CLOSED, //门是关着的
    UI_MGLOG_OPENED, //门的开着的
    UI_MGLOG_HAND_OPEN, //开门按钮开门
    UI_MGLOG_PROG_OPEN, // 软件开门
    UI_MGLOG_PROG_CLOSE, // 软件关门
    UI_MGLOG_ILLEGAL_OPEN, // 门被非法打开（没有按验证通过，直接破门而入）
    UI_MGLOG_ILLEGAL_REMOVE, // 门禁机被非法拆除
    UI_MGLOG_ALARM, //门禁机报警（胁迫报警，消防联动报警）
  } T_UI_MGLOG_TYPE;
```

3. 上传用户信息

当有新的用户增加时发送该命令

考勤机发送格式：

指纹：

```
{
 "cmd":"senduser",
"sn":"AI05_MIC_001",
 “enrollid”:1,
“name”:”chingzou”,
 “backupnum”:0, //0~9 指纹 10：密码 11 卡 50:照片
 “admin”:0,
 “record”,”kajgksjgaglasdjgjjglksajlgjkdgjajkdjgksaj”
}
```

卡：

{
"cmd":"senduser",
“enrollid”:1,
“backupnum”:11, //0~9 指纹 10：密码 11 卡
“admin”:0,
“record”,2352253
}
密码：
{
"cmd":"senduser",
“enrollid”:1,
“backupnum”:10, //0~9 指纹;10:密码;11:卡
“admin”:0,
“record”,12345678
}
照片：
{
"cmd":"senduser",
"sn":"AI05_MIC_001",
“enrollid”:1,
“name”:”chingzou”,
“backupnum”:50, //0~9 指纹 10：密码 11 卡 50:照片
“admin”:0,
“record”,”kajgksjgaglasdjgjjglksajlgjkdgjajkdjgksaj”
}

服务器回复信息：
成功：
{
"ret":”senduser ",
"result":true,
"cloudtime":"2016-03-25 13:49:30"
}
失败：
{
"ret":”senduser ",
"result":false,
"reason":1
}

二、服务器发起的请求

1. 获取用户列表

```
{
  "cmd":"getuserlist",
  ″stn″:true //stn: 起始包 true ;应答包：false
}
```

考勤机回复信息：

成功：

```
{
  "ret":"getuserlist",
  "result":true,
   “count ":40, //1~40 每包最大为 40 个包
“from”：0,
“to”:39,
 “record ":[
            {
               “enrollid ":1,
               "admin ":0,
               "backupnum ":0
            },
            {
               “enrollid ":2,
               "admin ":1,
               "backupnum ":0
            },
            {
               “enrollid ":3,
               "admin ":0,
               "backupnum ":10 //卡
            },
            ……
  ]
}
```

服务器回复：

```
{
  "cmd":"getuserlist",
  ″stn″:false //stn: 起始包 true ;应答包：false
}
//接着上传的第二个包
{
  "ret":"getuserlist",
  "result":true,
  "count ":40, //1~40 每包最大为 40 个包
“from”：40,
“to”:79,
```

```
 “record ":[
            {
               “enrollid ":1234,
               "admin ":0,
               "backupnum ":0
            },
            {
              “enrollid ":2345,
               "admin ":1,
               "backupnum ":0
            },
            {
              “enrollid ":5677,
               "admin ":0,
               "backupnum ":10 //卡
            },
            ......
  ]
}
```

.....

当数据为空时，考勤机返回：

```
{
  "ret":"getuserlist",
  "result":true,
  "count ":0,
  “from”：0,
  “to”:0,
  “record ":[]
}
```

失败：

```
{
  "ret":"getuserlist",
  "result":false,
  "reason":1
}
```

2. 获取用户信息

指纹：

```
{
  "cmd":" getuserinfo "
  "enrollid ":1,
  "backupnum ":0
}
```

考勤机回复信息：

成功：

{
"ret":" getuserinfo ",
"result":true,
“enrollid":1,
“name”:”chingzou”, //UTF8 addd in v1.1
“backupnum":0,
“admin":0,
"record":"aabbccddeeffggddssiifdjdkjfkjdsjlkjal",//长度:thbio3.0 算法 1620 以内;
thbio1.0:算法 1024 字节以内
}
失败：
{
"ret":”getuserinfo ",
"result":false,
"reason":1
}
卡：
{
"cmd":”getuserinfo "
"enrollid ":1,
"backupnum ":11
}
考勤机回复信息：
成功：
{
"ret":”getuserinfo ",
"result":true,
“enrollid":1,
“name”:”chingzou”, //UTF8 addd in v1.1
“backupnum":11,
“admin":0,
"record":23532253
}
失败：
{
"ret":”getuserinfo ",
"result":false,
"reason":1
}
密码：
{
"cmd":”getuserinfo "
"enrollid ":1,
"backupnum ":10

}
考勤机回复信息：
成功：
{
"ret":”getuserinfo ",
"result":true,
“enrollid":1,
“name”:”chingzou”, //UTF8 addd in v1.1
“backupnum":10,
“admin":0,
"record":23532253
}
失败：
{
"ret":”getuserinfo ",
"result":false,
"reason":1
}

照片：
{
"cmd":" getuserinfo "
"enrollid ":1,
"backupnum ":50
}
考勤机回复信息：
成功：
{
"ret":" getuserinfo ",
"sn":"AI05_MIC_001",
"result":true,
“enrollid":1,
“name”:”chingzou”, //UTF8 addd in v1.1
“backupnum":50,
“admin":0,
"faceflag":1,
"enable":1,
"shiftid":1,
"postid":1,
"record":"aabbccddeeffggddssiifdjdkjfkjdsjlkjal",//Base64
}
失败：
{
"ret":”getuserinfo ",

```
  "result":false,
  "reason":1
}
```

3. 下发用户信息

指纹：
服务器发送格式：

```
{
  "cmd":"setuserinfo",
  "enrollid":1,
  “name”:”chingzou”, //UTF8 addd in v1.1
  "backupnum",0, //0~9
  "admin":0,
  "record":"aabbccddeeffggddssiifdjdkjfkjdsjlkjalflsgsadg"
}
```

密码：
服务器发送格式：

```
{
  "cmd":"setuserinfo",
  "enrollid":1,
  “name”:”chingzou”, //UTF8 addd in v1.1
  "backupnum",10,
  "admin":0,
  "record":12345}
```

感应卡：
服务器发送格式：

```
{
  "cmd":"setuserinfo",
  "enrollid":1,
  “name”:”chingzou”, //UTF8 addd in v1.1
  "backupnum",11,
  "admin":0,
  "record":2352253}
```

照片：

```
{"cmd":"setuserinfo",
"enrollid":1,
"name":"1",
"backupnum":50,
"admin":0,
"record":"aabbccddeeffggddssiifdjdkjfkjdsjlkjalflsgsadg"//Base64
}
```

2、考勤机回复信息：
成功：

```
{"ret":"setuserinfo",
```

"result":true
}
失败：
{
"ret":"setuserinfo ",
"result":false,
"reason":1
}

4. 删除用户信息

服务器发送格式：
{
"cmd":"deleteuser"
"enrollid",1,
"backupnum":0 //0~9 fp; 10: password 11:card ； =》12：删除所有指纹信息 13，删除整个人
}
考勤机回复信息：
成功：
{
"ret":"deleteuser",
"result":true
}
失败：
{
"ret":"deleteuser",
"result":false,
"reason":1
}

5. 获取用户名字

服务器发送格式：
{
"cmd":"getusername",
"enrollid":1
}
考勤机回复信息：
成功：
{
"ret":"getusername",
"result":true,
"record":"chingzou”//utf8
}
失败：
{
"ret":"getusername",

```
  "result":false,
  "reason":1
}
```

6. 设置用户名字

服务器发送格式：

```
{
  "cmd":"setusername",
  “count”:50, //每次最多发送 50 个
   “record”,[
            {
              “enrollid”:1,
              “name”:”chingzou”
            },
            {
              “enrollid”:2,
              “name”:”chingzou2”
            },
            ……..
           ]
}
```

考勤机回复信息：

成功：

```
{
  "ret":"setusername",
  "result":true
}
```

失败：

```
{
  "ret":"setusername",
  "result":false,
  "reason":1
}
```

7. 使能用户

服务器发送格式：

```
{
  "cmd":"enableuser",
  “enrollid”:1,
  “enflag”:1
}
```

考勤机回复信息：

成功：

```
{
```

"ret":" enableuser ",
"result":true
}
失败：
{
"ret":" enableuser ",
"result":false,
"reason":1
}

8. 禁止用户

服务器发送格式：
{
"cmd":"enableuser",
“enrollid”:1,
“enflag”:0
}
考勤机回复信息：
成功：
{
"ret":" enableuser ",
"result":true
}
失败：
{
"ret":" enableuser ",
"result":false,
"reason":1
}

9. 清空所有用户

服务器发送格式：
{
"cmd":"cleanuser"
}
考勤机回复信息：
成功：
{
"ret":"cleanuser",
"result":true
}
失败：
{
"ret":"cleanuser",
"result":false

```
  "reason":1
}
```

10. 获取新的考勤记录

服务器发送格式：

```
{
 "cmd":"getnewlog",
 “stn”:true
}
```

考勤机回复信息：

成功：

```
{
  "ret":"getnewlog ",
  "result":true,
  "count":1000,
  “from”:0,
  “to”:49,
  "record":[
                    {
                     "enrollid":1,
                     "time":"2016-03-25 13:49:30",
                     "mode":0, //0 fp 1:card 2:pwd
                     "inout":0, //0 in 1:out
                     "event":0
                    }
                    {
                     "enrollid":2,
                     “time ":"2016-03-25 13:49:30",
                     "mode":0, //0 fp 1:card 2:pwd
                     "inout":0, //0 in 1:out
                     "event":1
                   }
                    ......
              ]
}
```

服务器应答：

```
{
 "cmd":"getnewlog",
 “stn”:false
}
```

考勤机回复第二个数据包：

```
{
  "ret":"getnewlog ",
  "result":true,
```

```
  "count":1000,
  “from”:50,
  “to”:99,
  "record":[
                    {
                     "enrollid":111,
                     "time":"2016-03-25 13:49:30",
                     "mode":0, //0 fp 1:card 2:pwd
                     "inout":0, //0 in 1:out
                     "event":0
                    }
                    {
                     "enrollid":112,
                     "time ":"2016-03-25 13:49:30",
                     "mode":0, //0 fp 1:card 2:pwd
                     "inout":0, //0 in 1:out
                     "event":1
                   }
                    ......
              ]
}
```

当数据为空时，考勤机返回：

```
{
  "ret":" getnewlog ",
  "result":true,
  "count ":0,
  “from”:0,
  “to”:0,
  “record ":[]
}
```

失败：

```
{
  "ret":"getnewlog ",
  "result":false
  "reason":1
}
```

11. 获取全部考勤记录

服务器发送格式：

```
{
 "cmd":"getalllog",
 “stn”:true
}
```

考勤机回复信息：

成功：

```
{
   "ret":"getalllog",
   "result":true,
   "count":1000,
   “from”:0,
   “to”:49,
   "record":[
                    {
                     "enrollid":1,
                     "time":"2016-03-25 13:49:30",
                     "mode":0, //0 fp 1:card 2:pwd
                     "inout":0, //0 in 1:out
                     "event":0
                    }
                    {
                     "enrollid":2,
                     “time ":"2016-03-25 13:49:30",
                     "mode":0, //0 fp 1:card 2:pwd
                     "inout":0, //0 in 1:out
                     "event":1
                   }
                    ......
              ]
}
```

服务器应答：

```
{
 "cmd":"getalllog",
 “stn”:false
}
```

考勤机回复第二个数据包：

```
{
   "ret":"getalllog",
   "result":true,
   "count":1000,
   “from”:50,
   “to”:99,
   "record":[
                    {
                     "enrollid":111,
                     "time":"2016-03-25 13:49:30",
                     "mode":0, //0 fp 1:card 2:pwd
                     "inout":0, //0 in 1:out
                     "event":0
                    }
```

```
                    {
                      "enrollid":112,
                      “time ":"2016-03-25 13:49:30",
                      "mode":0, //0 fp 1:card 2:pwd
                      "inout":0, //0 in 1:out
                      "event":1
                    }
                     ......
                ]
}
```

当数据为空时，考勤机返回：

```
{
   "ret":" getalllog ",
   "result":true,
   "count ":0,
   “from”:0,
   “to”:0,
   “record ":[]
}
```

失败：

```
{
   "ret":"getalllog",
   "result":false
   "reason":1
}
```

12. 清空所有记录

服务器发送格式：

```
{
   "cmd":"cleanlog"
}
```

考勤机回复信息：

成功：

```
{
   "ret":"cleanlog",
   "result":true
}
```

失败：

```
{
   "ret":"cleanlog",
   "result":false
   "reason":1
}
```

13. 系统初始化

说明:系统初始化将删除所有用户信息和考勤记录.其它参数不变
服务器发送格式：

```
{
   "cmd":"initsys"
}
```

考勤机回复信息：
成功：

```
{
   "ret":" initsys",
   "result":true
}
```

失败：

```
{
   "ret":" initsys ",
   "result":false
   "reason":1
}
```

14. 重启机器

说明:将终端考勤机重新启动。没有应答信息，当考勤机收到这条指令后，马上重启。重启后将会在 10~20 秒，向服务器发送注册信息。
服务器发送格式：

```
{
   "cmd":"reboot"
}
```

15. 清除所有管理员

说明:将清除系统所有管理员.
服务器发送格式：

```
{
   "cmd":"cleanadmin"
}
```

考勤机回复信息：
成功：

```
{
   "ret":" cleanadmin",
   "result":true
}
```

失败：

```
{
   "ret":" cleanadmin",
   "result":false
   "reason":1
```

}

16. 同步考勤机时间

说明:设置考勤机时间。
服务器发送格式：

```
{
   "cmd":"settime",
   "cloudtime":"2016-03-25 13:49:30"
}
```

考勤机回复信息：
成功：

```
{
   "ret":" settime ",
   "result":true
}
```

失败：

```
{
   "ret":" settime",
   "result":false
   "reason":1
}
```

17. 设置设备参数

服务器发送格式：

```
{
   "cmd":"setdevinfo",
   “deviceid”:1, //设备机号
   “language”:0, //设备显示语言 ：如备注选项
   “volume”:0, //音量：0~10 默认 6
   “screensaver”:0 // 屏保，不操作 0~255 秒进入屏保，0：无屏保
   “verifymode”:0, //组合认证方式,如备注所示
   “sleep”: 0 , //休眠 0：不休眠，指纹头常亮 1：休眠。
   “userfpnum”:3,//每个用户多少指纹数，1~10，默认 3
   “loghint”:1000, //满记录提示，当记录离满只有 1000 条时，提示。0：不提示。
   “reverifytime”:0 //重复确认时间 0~255 分钟
}
```

考勤机回复信息：
成功：

```
{
   "ret":" setdevinfo ",
   "result":true
}
```

失败：

```
{
   "ret":" setdevinfo ",
```

```
   "result":false
   "reason":1
}
```

说明:设置考勤机所有公共参数,子项目可选,如果没有这个项目,可以不写;

例如你可以只设置屏保与重复确认时间：

```
{
   ”cmd“:”setdevinfo”,
   “screensaver”:30,
  “reverifytime”:5
}
```

备注：

设备显示语言：

```
 enum
{
      UILANG_EN, //==0 英文
      UILANG_SC, //==1 中文简体
      UILANG_TC, //==2 中文繁体
      UILANG_JAPAN, //==3 日本语
      UILANG_NKR, //==4 朝鲜语
      UILANG_SKR, //==5 韩国语
      UILANG_THAI, //==6 泰语
      UILANG_INDONESIA, //==7 印尼语
      UILANG_VIETNAM, //==8 越南语
      UILANG_SPA, //==9 西班牙语
      UILANG_FAN, //==10 法语
      UILANG_POR, //==11 葡萄牙语
      UILANG_GEN, //==12 德语
      UILANG_RUSSIA, //==13 俄语
      UILANG_TUR, //==14 土耳其语
      UILANG_ITALIAN, //==15 意大利语
      UILANG_CZECH, //==16 捷克语
      UILANG_ALB, //==17 阿拉伯语
      UILANG_PARSI//==18 波斯语
} T_UI_LANG;
```

组合认证方式：

```
enum
{
      VERIFY_KIND_FP_CARD_PWD, //==0 卡或者指纹或者密码
      VERIFY_KIND_CARD_ADD_FP, //==1 卡加指纹
      VERIFY_KIND_PWD_ADD_FP, //==2 密码加指纹
      VERIFY_KIND_CARD_ADD_FP_ADD_PWD, //==3 卡加指纹加密码
       VERIFY_KIND_CARD_ADD_PWD, //==4 卡加密码
};
```

18. 读取设备参数

说明:读取考勤机所有公共参数;
服务器发送格式
{
“cmd”:”getdevinfo”
}
考勤机回复信息：
成功
{
"ret":"getdevinfo",
“result”:true,
“deviceid”:1, //设备机号
“language”:0, //设备显示语言 ：如备注选项
“volume”:0, //音量：0~10 默认 6
“screensaver”:0 // 屏保，不操作 0~255 秒进入屏保，0：无屏保
“verifymode”:0, //认证组合：
“sleep”: 0 , //休眠 0：不休眠，指纹头常亮 1：休眠。
“userfpnum”:3,//每个用户多少指纹数，1~10，默认 3
“loghint”:1000, //满记录提示，当记录离满只有 1000 条时，提示。0：不提示。
“reverifytime”:0 //重复确认时间 0~255 分钟
}
失败：
{
"ret":" getdevinfo ",
"result":false
"reason":1
}

19. 开门

说明:控制门禁机打开门
服务器发送格式：
{
"cmd":"opendoor"
}
考勤机回复信息：
成功：
{
"ret":" opendoor",
"result":true
}
失败：
{
"ret":" opendoor",
"result":false

```
  "reason":1
}
```

20. 设置设备门禁参数

服务器发送格式：

```
{
  "cmd":"setdevlock",
  “opendelay”:5, //开门延时
  “doorsensor”:0, //门磁类型:0 禁止, 1 常闭型门磁, 2 常开型门磁
  “alarmdelay”:0, //门磁报警:0 禁止 1~255 :开门延时超过分钟数,门禁机报警
  “threat”:0, //胁迫报警: 0 禁止  1、开门加报警  2、报警 3、开门
  “InputAlarm”:0, //消防输入:0 禁止 1,明报警 2,暗报警
  “antpass”: 0 ,//反潜回:0 禁止 1,主机在内 2,主机在外
  “interlock”:0 ,//0 禁止 1,双门互锁打开
“mutiopen”:0, //多人开门: 0,禁止 1~4  同时 1~4 个人验证才能开门
“tryalarm”:0, //试错报警：0，禁止，1~10 ，试错 1~10 次后，门禁机报警。
“tamper”:0, //防拆报警： 0，禁止， 1：开启
“wgformat”:0, //韦根输出格式 0 :26 1:34
“wgoutput”: 0, //韦根输出：0,：id 号 1:1+id 号 2：机号+ID 号
“cardoutput”:0, //当该用记有注册卡时，输出卡号 0，禁止 1：启用
“dayzone”:[    //天时段 ，最多 8 组，每组表示一天，每天最多 5 个段。可以选择只下发 1 组或者几组
            {
              “day”: [
                    {“section”:”06:00~07:00”},
                    {“section”:”08:30~12:00”},
                    {“section”:”13:00~17:00”},
                    {“section”:”18:00~21:00”},
                    {“section”:”22:00~23:30”},
                  ]
              },
              {
               “day”: [
                    {“section”:”00:01~23:59”},
                  ]
              },
              ……..
            ],
“weekzone”:[  //周时段，最多 8 组，每组表示一周。可以选择只下发 1 组或者多组
              {“week”:[
                    {“day”:1}, //周一
                    {“day”:1}, //周二
                    {“day”:1}, //周三
                    {“day”:1}, //周四
                    {“day”:1}, //周五
```

{“day”:2}, //周六
{“day”:2}, //周日
]
},
{“week”:[//第二个周时段
{“day”:1}, //周一
{“day”:1}, //周二
{“day”:1}, //周三
{“day”:1}, //周四
{“day”:1}, //周五
{“day”:2}, //周六
{“day”:2}, //周日
]
},
……..
],
“lockgroup”:[//锁组合功能
{“group”:1234},
{“group”:126},
{“group”:348},
{“group”:139},
{“group”:15}
]
}

考勤机回复信息：

成功：

```
{
   "ret":" setdevlock ",
   "result":true
}
```

失败：

```
{
   "ret":" setdevlock ",
   "result":false
   "reason":1
}
```

说明:1、设置门禁机的所有公共参数,子项目可选,如果没有这个项目,可以不写;好复杂，哈哈

例如：如果你只想设置 antpass 和 tamper 你可发这样写：

```
{
   "cmd":" setdevlock ",
   "antpass":1,
   “tamper”:true
}
```

或者反过来写：

{
"cmd":" setdevlock ",
“tamper”:true, //写 1 或者 写成 true 都可以
"antpass":1
}
或者你只想设置一个天时段，天时段里只有一个时间段和一个周时段，周时段里，只设置周一的天时段，也可以这样写，只要严格按 JSON 的格式来写就行：

{“cmd”:”setdevlock”,
“dayzone”:[{“day”,[{“section”:”07:00~18:00”}]}],
“weekzone”:[{“week”,[{“day”:1}]}]
}

2、关于周时段，天时段的关系：

某个用户门禁参数（周时段=3）-》周时段 3-》周一（1）-》天时段（1）-》时间段
-》周二（2）-》天时段（2）-》时间段

例如：天时段（1）中有个时间段：”00:01~23:59”
天时段（2）中有个时间段：”00:00~00:00”

这个人按了指纹，首先找到对应的周时段 3， 然后今天是周一的话，找到周时段 3 的周一对应的是天时段 1，天时段,1，是全天能行，则按指纹开门。

如果今天是周二，找到周时段 3 的周二对应的是天时段 2，天时段 2，是全天禁止能行，则按指纹，不能开门。

3、关于锁功能组合。

首先在用户门禁参数里，设置这个人属于某一组。

假如：一个公司有三个部门：

财务部：张一，张二，张三：所有人设置门禁参数的组号为 1
销售部：李一，李二，李三：所有人设置门禁参数的组号为 2
仓管部：王一，王二，王三：所有人设置门禁参数的组号为 9

如果锁组合功能里面，有一个参数为：129.

则：张一，李三，王二，同时按指纹，可以打开门。

如果参数为：119

则：张一，张三，王三，同时按指纹，可以打开门。

21. 读取设备门禁参数

说明:同上
服务器发送格式：
{
"cmd":" getdevlock "
}

门禁机回复格式：
成功
{
"ret":"getdevlock",
“result”:true
“opendelay”:5, //开门延时
“doorsensor”:0, //门磁类型:0 禁止, 1 常闭型门磁, 2 常开型门磁

“alarmdelay”:0, //门磁报警:0 禁止 1~255 :开门延时超过分钟数,门禁机报警
“threat”:0, //胁迫报警: 0 禁止 1、开门加报警 2、报警 3、开门
“InputAlarm”:0, //消防输入:0 禁止 1,明报警 2,暗报警
“antpass”: 0 ,//反潜回:0 禁止 1,主机在内 2,主机在外
“interlock”:0 ,//0 禁止 1,双门互锁打开
“mutiopen”:0, //多人开门: 0,禁止 1~4 同时 1~4 个人验证才能开门
“tryalarm”:0, //试错报警：0，禁止，1~10 ，试错 1~10 次后，门禁机报警。
“tamper”:0, //防拆报警： 0，禁止， 1：开启
“wgformat”:0, //韦根输出格式 0 :26 1:34
“wgoutput”: 0, //韦根输出：0,：id 号 1:1+id 号 2：机号+ID 号
“cardoutput”:0, //当该用记有注册卡时，输出卡号 0，禁止 1：启用
“dayzone”:[//天时段 ，最多 8 组，每组表示一天，每天最多 5 个段。可以选择只下发 1 组或者几组

```
{
  “day”: [
        {“section”:”06:00~07:00”},
        {“section”:”08:30~12:00”},
        {“section”:”13:00~17:00”},
        {“section”:”18:00~21:00”},
        {“section”:”22:00~23:30”},
      ]
  },
  {
   “day”: [
        {“section”:”00:01~23:59”},
      ]
  },
  ……..
],
```

“weekzone”:[//周时段，最多 8 组，每组表示一周。可以选择只下发 1 组或者多组

```
{“week”:[
        {“day”:1}, //周一
        {“day”:1}, //周二
        {“day”:1}, //周三
        {“day”:1}, //周四
        {“day”:1}, //周五
        {“day”:2}, //周六
        {“day”:2}, //周日
      ]
  },
  {“week”:[  //第二个周时段
        {“day”:1}, //周一
        {“day”:1}, //周二
        {“day”:1}, //周三
```

{“day”:1}, //周四
{“day”:1}, //周五
{“day”:2}, //周六
{“day”:2}, //周日
]
},
……..
],
“lockgroup”:[//锁组合功能
{“group”:1234},
{“group”:126},
{“group”:348},
{“group”:139},
{“group”:15}
]
}
失败：
{
"ret":" setdevlock ",
"result":false
"reason":1
}

22. 获取用户门禁参数

说明:获取单个用户的门禁参数
服务器发送格式：
{
"cmd":"getuserlock",
“enrollid”:1
}
考勤机回复信息：
成功：
{
"ret":" getuserlock ",
"result":true,
“enrollid”:1,
“weekzone”:1, //对应以上设置的周时段，
“group”:1, //用于锁组合开门的所属分组，0：没有分组， 1~9：将检查锁组合
“starttime”:”2016-03-25 01:00:00”, //用户有效期开始 ，只有在有效期范围内，才能开门。不设置，默认都可以开门
“endtime”: ”2099-03-25 23:59:00”, //用户有效期结束
}
失败：
{
"ret":" getuserlock ",

```
  "result":false
  "reason":1
}
```

23. 设置用户门禁参数

说明:设置多个用户的门禁参数
服务器发送格式：

```
{
  "cmd":"setuserlock",
  “count”:40,
  “record”:[
            {
            “enrollid”:1，
            “weekzone”:1,
            “group”:1,
            “starttime”:”2016-03-25 01:00:00”,
            “endtime”: ”2099-03-25 23:59:00”
            },
            {
            “enrollid”:2，
            “weekzone”:1,
            “group”:1,
            “starttime”:”2016-03-25 01:00:00”,
            “endtime”: ”2099-03-25 23:59:00”
            },
            ……..
           ]
}
```

考勤机回复信息：
成功：

```
{
  "ret":" setuserlock ",
  "result":true
}
```

失败：

```
{
  "ret":" setuserlock ",
  "result":false
  "reason":1
}
```

24. 删除用户门禁参数

服务器发送格式：

```
{
  "cmd":"deleteuserlock",
```

```
  “enrollid”:1
}
```

考勤机回复信息：

成功：

```
{
  "ret":" deleteuserlock ",
  "result":true
}
```

失败：

```
{
  "ret":" deleteuserlock ",
  "result":false
  "reason":1
}
```

25. 清空全部用户门禁参数

服务器发送格式：

```
{
  "cmd":"cleanuserlock"
}
```

考勤机回复信息：

成功：

```
{
  "ret":" cleanuserlock ",
  "result":true
}
```

失败：

```
{
  "ret":" cleanuserlock ",
  "result":false
  "reason":1
}
```