

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 2. 增加用户名字项 3. 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	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
2018-06-07	1.8	1. 为所有命令 添加 sn 2. 门禁控制器增加开门（门禁控制器有 4 个门） 3. 为 门 禁 控 制 器 添 加 setuserlock 和 getuserlock 的 timezone2 和 timezone3（门禁控制器有 4 个门）	
2019-03-27	1.9	sendlog 添加索引	
2021-2-2	2.0	增加 AI 设备照片信息（backupnum 为 50，包含日志图片和临时图片）	
2025-03-21	2.8	远程添加用户	
2025-06-11	2.9	下载用户信息（新增接口）、增加“设置门状态”接口	
2025-12-04	3.0	调整格式	

目录

一、考勤机主动发起的请求	1
1. 登录注册云服务	1
2. 上传考勤记录	2
3. 上传用户信息	3
二、服务器发起的请求	5
1. 获取用户列表	5
2. 获取用户信息	7
3. 下发用户信息	9
4. 删除用户信息	10
5. 获取用户名字	11
6. 设置用户名字	12
7. 使能用户	12
8. 禁止用户	13
9. 清空所有用户	13
10. 获取新的考勤记录	14
11. 获取全部考勤记录	16
12. 清空所有记录	17
13. 系统初始化	18
14. 重启机器	18
15. 清除所有管理员	19
16. 同步考勤机时间	19
17. 设置设备参数	20
18. 读取设备参数	21
19. 开门	21
20. 设置设备门禁参数	22
21. 读取设备门禁参数	25
22. 获取用户门禁参数	27
23. 设置用户门禁参数	27
24. 删除用户门禁参数	28
25. 清空全部用户门禁参数	29

26. 获取时间	29
27. QR 码发送	30
28. QR 码服务器回复	30
29. 远程添加用户	30
30. 获取问卷参数	31
31. 获取设备信息	32
32. 设置问卷参数	32
33. 设置用户信息	33
34. 获取用户信息	34
35. 获取假期参数	34
36. 设置假期参数	35
37. 下发用户信息（需 AI 人脸识别设备固件版本 v5.09 或 v2.09 及以上）	36
38. 设置门状态	37

说明：

1. 采用 WEBSOCKET 协议通讯，版本 RFC6455 13，默认端口为 7788。
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。
4. 关于 backupnum:
 - 0~9: 指纹
 - 10: 密码
 - 11: RFID 卡
 - 20~27: 静态人脸
 - 40~41: 掌静脉（palm vein）
 - 50: 照片（Base 64 格式）单个用户支持 10 个指纹、1 个密码、1 个 RFID 卡

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":"ZX12345678",    //考勤机序列号，出厂固定，全球唯一
  "cpush":"123456789",  //CPU 序列号，固定值
  "devinfo":{
    "modelname":"tfs30", //考勤机型号
    "usersize":3000,      //用户容量（1000/3000/5000）
    "fpsize":3000,        //指纹容量
    "cardsize":3000,      //RFID 卡容量
    "pwdsize":3000,       //密码容量
    "logsize":100000,     //考勤记录容量
    "useduser":1000,      //已注册用户数
    "usedfp":1000,        //已存指纹数
    "usedcard":2000,      //已存 RFID 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（thbio1.0 或 thbio3.0）
    "firmware":"th600w v6.1", //终端固件版本
    "time":"2016-03-25 13:49:30", //终端时间
    "mac":"00-01-A9-01-00-01" //局域网 MAC 地址
  }
}
```

服务器回复信息:

成功:

```
{
  "ret":"reg",    //回复命令字
  "result":true,  //结果
  "cloudtime":"2016-03-25 13:49:30", //服务器时间
  "nosenduser":true //是否自动发送新用户消息（true=自动发送）
}
```

失败:

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

2. 上传考勤记录

考勤机发送格式:

```
{
  "cmd":"sendlog",    // 命令字
  "sn":"ZX12345678",  // 设备序列号
  "count":2,          // 记录数量
  "logindex":0,       // 日志索引（2019-03-27 新增）
  "record":[
    {
      "enrollid":1,    // 用户注册 ID
      "name":"zzz",    // 用户姓名（UTF-8 编码）
      "time":"2016-03-25 13:49:30", // 打卡时间
      "mode":0,        // 验证方式（1: 指纹 2: 密码 3: 卡 8: 人脸）
      "inout":0,       // 进出状态（0:主机进门 1:子机出门）
      "event":0,       // 自定义事件（0~16，如 F1 键对应 1）
      "temp":36.5,     // 体温（仅测温设备支持）
      "image":"/9j/4AAQSkZJRgABAQAAQABAAQ" // 实时打卡图片（Base64）
    },
    {
      "enrollid":2,
      "name":"xyz",
      "time":"2016-03-25 13:49:30",
      "mode":0,
      "inout":0,
      "event":1,
      "temp":36.5, // 体温（仅测温设备支持）
      "image":"/9j/4AAQSkZJRgABAQAAQABAAQ" // 打卡图片（Base64，仅 AI 人脸设备支持）
    }
  ]
}
```

服务器回复信息:

成功:

```
{
  "ret":"sendlog",    // 响应命令字
  "result":true,      // 结果
  "count":1,          // 接收记录数（2019-03-27 新增）
  "logindex":0,       // 日志索引（2019-03-27 新增）
  "cloudtime":"2016-03-25 13:49:30", // 服务器时间
  "access":1,         // 开门权限（1:允许 0:禁止，扩展功能）
  "message":"设备接口信息", // AI 人脸模式返回（最大 125 字符）
  "voice":"OK",       // 语音提示（中英文）
  "fontsize":32,      // AI 人脸消息栏字体大小（可选值：10,12,...40）
}
```

```

    "text_color":[255,0,0] // 文字颜色（RGB，消息栏模式）
}
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 : 1:fp, 2:pwd, 3:card, 8:face ; 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", // 命令字
    "sn":"ZX12345678", // 设备序列号
    "enrollid":1,      // 用户注册 ID
}

```

```
"name":"chingzou", // 用户姓名（UTF-8 编码）
"backupnum":0, //信息类型(0~9:指纹 10:密码 11:RFID 卡 20-27:静态人脸 40-41:掌静脉
                    50:照片)
"admin":0, // 权限（0:普通用户 1:管理员 2:超级用户）
"record":"kajgksjgaglas" // 特征数据（THbio3.0≤1620 字符，THbio1.0≤1024 字符）
}
卡：
{
    "cmd":"senduser",
    "sn":"ZX12345678",
    "enrollid":1,
    "name":"chingzou",
    "backupnum":11, // RFID 卡
    "admin":0,
    "record":2352253 // 卡编号（数字）
}
密码：
{
    "cmd":"senduser",
    "sn":"ZX12345678",
    "enrollid":1,
    "name":"chingzou",
    "backupnum":10, // 密码
    "admin":0,
    "record":12345678 // 密码（最多 8 位）
}
掌静脉：
{
    "cmd":"senduser",
    "sn":"ZX12345678",
    "enrollid":1,
    "name":"chingzou",
    "backupnum":40, // 0~9 指纹 10: 密码 11 卡 40:掌静脉 41:掌静脉 50:照片
    "admin":0,
    "record":"kajgksjgaglasdjgijlksajlgjkdgjajkdjgksaj" // 特征数据
}
照片：
{
    "cmd":"senduser",
    "sn":"ZX12345678",
    "enrollid":1,
    "name":"chingzou",
    "backupnum":50, // 照片（Base64 格式）
    "admin":0,
```



```
"record":"aabbccddeeffgg..." // Base64 编码图片数据
}
```

服务器回复信息：

成功：

```
{
  "ret":"senduser", // 响应命令字
  "result":true, // 结果
  "cloudtime":"2016-03-25 13:49:30" // 服务器时间
}
```

失败：

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

二、服务器发起的请求

1. 获取用户列表

服务器发送格式：

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

考勤机回复信息（终端响应）：

成功（首包）：

```
{
  "ret":"getuserlist",
  "sn":"ZX12345678", // 设备序列号
  "result":true,
  "count":40, // 每包记录数（1~40，最大 40 条）
  "from":0, // 起始索引（从 0 开始）
  "to":39, // 结束索引（包含 to）
  "record":[
    {
      "enrollid":1, // 用户注册 ID
      "admin":0, // 权限（0:普通用户 1:管理员 2:超级用户）
      "backupnum":0 // 身份信息类型（0:指纹 10:密码 11:RFID 卡）
    },
    {
      "enrollid":2,
      "admin":1,
      "backupnum":0
    },
  ],
}
```

```
{
  "enrollid":3,
  "admin":0,
  "backupnum":10 // RFID 卡
}
]
```

服务器回复（请求下一包）：

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

//接着上传的第二个包

```
{
  "ret":"getuserlist",
  "sn":"ZX12345678",
  "result":true,
  "count":40,
  "from":40, // 下一包起始索引
  "to":79, // 下一包结束索引
  "record":[
    {
      "enrollid":1234,
      "admin":0,
      "backupnum":0
    },
    {
      "enrollid":2345,
      "admin":1,
      "backupnum":0
    },
    {
      "enrollid":5677,
      "admin":0,
      "backupnum":10 // RFID 卡
    }
  ]
}
```

.....

当数据为空时，考勤机返回：

```
{
  "ret":"getuserlist",
  "sn":"ZX12345678",
  "result":true,
```

```
"count":0,
"from":0,
"to":0,
"record":[]    // 空数组
}
失败:
{
  "ret":"getuserlist",
  "sn":"ZX12345678",
  "result":false,
  "reason":1    // 错误码（如 1:参数错误）
}
```

2. 获取用户信息

服务器发送格式:

指纹:

```
{
  "cmd":"getuserinfo", // 命令字
  "sn":"zx12345678", // 设备序列号
  "enrollid":1,      // 用户注册 ID
  "backupnum":0      // 信息类型（0:指纹 10:密码 11:RFID 卡 50:照片）
}
```

考勤机回复信息（终端响应）:

成功:

```
{
  "ret":"getuserinfo",
  "sn":"ZX12345678",
  "result":true,
  "enrollid":1,
  "name":"chingzou", // 用户姓名（UTF-8）
  "backupnum":0,
  "admin":0,         // 权限（0:普通用户 1:管理员）
  "record":"aabbccddeeffggddssiifdjkjfkjdsjlkjal" // 指纹特征数据（THbio3.0≤1620 字符）
}
```

失败:

```
{
  "ret":"getuserinfo",
  "sn":"ZX12345678",
  "result":false,
  "reason":1
}
```

卡:

```
{
  "cmd":"getuserinfo",
```

```
"enrollid":1,  
"backupnum":11  
}
```

考勤机回复信息:

成功:

```
{  
  "ret":"getuserinfo",  
  "sn":"ZX12345678",  
  "result":true,  
  "enrollid":1,  
  "name":"chingzou",  
  "backupnum":11,  
  "admin":0,  
  "record":23532253 // 卡编号 (数字)  
}
```

失败:

```
{  
  "ret":"getuserinfo",  
  "sn":"ZX12345678",  
  "result":false,  
  "reason":1  
}
```

密码:

```
{  
  "cmd":"getuserinfo",  
  "enrollid":1,  
  "backupnum":10  
}
```

考勤机回复信息:

成功:

```
{  
  "ret":"getuserinfo",  
  "sn":"ZX12345678",  
  "result":true,  
  "enrollid":1,  
  "name":"chingzou",  
  "backupnum":10,  
  "admin":0,  
  "record":12345678 //密码 (最多 8 位)  
}
```

失败:

```
{  
  "ret":"getuserinfo",  
  "sn":"ZX12345678",
```

```

    "result":false,
    "reason":1
}
照片:
{
    "cmd":"getuserinfo",
    "enrollid":1,
    "backupnum":50
}
考勤机回复信息:
成功:
{
    "ret":"getuserinfo",
    "sn":"ZX12345678",
    "result":true,
    "enrollid":1,
    "name":"chingzou",
    "backupnum":50,
    "admin":0,
    "record":"aabbccddeeffggddssiifdjkjfkjdsjlkjal" // Base64 编码照片数据
}
失败:
{
    "ret":"getuserinfo",
    "sn":"ZX12345678",
    "result":false,
    "reason":1
}

```

3. 下发用户信息

1、首先，服务器发送消息：

指纹信息下发：

服务器发送格式：

```

{
    "cmd":"setuserinfo",    // 命令字
    "enrollid":1,          // 用户注册 ID
    "name":"chingzou",     // 用户名（UTF-8 编码）
    "backupnum":0,         // 信息类型（0: 指纹）
    "admin":0,             // 权限等级（0: 普通用户）
    "record":"aabbccddeeffggddssiifdjkjfkjdsjlkjalflsgsadg" // 指纹特征数据
}

```

照片信息下发：

服务器发送格式：

```

{

```

```

"cmd":"setuserinfo",
"enrollid":1,
"name":"chingzou",
"backupnum":50,           // 信息类型（50:照片）
"admin":0,
"record":"aabbccddeeffggddssiifdjdkjfkjdsjlkjalflsgsadg" // Base64 编码照片数据
}

```

密码信息下发:

服务器发送格式:

```

{
  "cmd":"setuserinfo",
  "enrollid":1,
  "name":"邹春庆",
  "backupnum":10,         // 信息类型（10:密码）
  "admin":0,
  "record":12345678       // 密码（最多 8 位数字）
}

```

RFID 卡信息下发:

服务器发送格式:

```

{
  "cmd":"setuserinfo",
  "enrollid":1,
  "name":"chingzou",
  "backupnum":11,         // 信息类型（11:RFID 卡）
  "admin":0,
  "record":2352253        // 卡编号（数字）
}

```

2、随后，考勤机回复信息（终端响应）:

成功:

```

{
  "ret":"setuserinfo",     // 响应命令字
  "sn":"ZX12345678",
  "result":true           // 操作结果
}

```

失败:

```

{
  "ret":"setuserinfo",
  "sn":"ZX12345678",
  "result":false,
  "reason":1              // 错误码（1:参数错误/用户不存在）
}

```

4. 删除用户信息

服务器发送格式:

```
{
  "cmd":"deleteuser", // 命令字
  "enrollid":1,        // 用户注册 ID
  "backupnum":0        // 这里 0 是删除第一个指纹。删除类型有以下（0~9:单个指纹 10:密码 11:RFID 卡 12:所有指纹 13:所有信息）
}
```

考勤机回复信息（终端响应）：

成功：

```
{
  "ret":"deleteuser", // 响应命令字
  "sn":"ZX12345678",
  "result":true,       // 结果
  "sn":"ZX12345678"   // 设备序列号
}
```

失败：

```
{
  "ret":"deleteuser",
  "sn":"ZX12345678",
  "result":false,
  "reason":1           // 错误码（如 1:用户不存在）
}
```

5. 获取用户名字

服务器发送格式：

```
{
  "cmd":"getusername",
  "sn":"ZX12345678",
  "enrollid":1
}
```

考勤机回复信息（终端响应）：

成功：

```
{
  "ret":"getusername",
  "sn":"ZX12345678",
  "result":true,
  "record":"chingzou" //utf8 或 ascii
}
```

失败：

```
{
  "ret":"getusername",
  "sn":"ZX12345678",
  "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",
  "sn": "ZX12345678",
  "result": true
}
```

失败:

```
{
  "ret": "setusername",
  "sn": "ZX12345678",
  "result": false,
  "reason": 1
}
```

7. 使能用户

服务器发送格式:

```
{
  "cmd": "enableuser",
  "enrollid": 1,
  "enflag": 1    //1: 使能用户, 0: 禁止用户
}
```

考勤机回复信息:

成功:

```
{
  "ret": "enableuser",
```



```
"sn":"ZX12345678",
"result":true
}
```

失败:

```
{
  "ret":"enableuser",
  "sn":"ZX12345678",
  "result":false,
  "reason":1
}
```

8. 禁止用户

服务器发送格式:

```
{
  "cmd":"enableuser",
  "enrollid":1,
  "enflag":0    //0 代表禁止用户
}
```

考勤机回复信息:

成功:

```
{
  "ret":"enableuser",
  "sn":"ZX12345678",
  "result":true
}
```

失败:

```
{
  "ret":"enableuser",
  "sn":"ZX12345678",
  "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",
  "sn":"ZX12345678",
  "result":true
}
```

```
}
```

失败:

```
{
  "ret":"cleanuser",
  "sn":"ZX12345678",
  "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",
  "sn":"ZX12345678",
  "result":true,
  "count":1000,
  "from":0,
  "to":49,
  "record":[
    {
      "enrollid":1,
      "time":"2016-03-25 13:49:30",
      "mode":0,    //1: 指纹 2: 密码 3: 卡
      "inout":0,  //0: 主机进门 1: 子机出门 t
      "event":0
    },
    {
      "enrollid":2,
      "time":"2016-03-25 13:49:30",
      "mode":0,    //1: 指纹 2: 密码 3: 卡
      "inout":0,  //0: 主机进门 1: 子机出门
      "event":1
    }
    .....
  ]
}
```

服务器应答:

```
{
```

```
"cmd":"getnewlog",
"stn":false
}
```

考勤机回复第二个数据包:

```
{
  "ret":"getnewlog",
  "sn":"ZX12345678",
  "result":true,
  "count":1000,
  "from":50,
  "to":99,
  "record":[
    {
      "enrollid":111,
      "time":"2016-03-25 13:49:30",
      "mode":0,    //1: 指纹 2: 密码 3: 卡
      "inout":0,  //0: 主机进门 1: 子机出门
      "event":0
    },
    {
      "enrollid":112,
      "time":"2016-03-25 13:49:30",
      "mode":0,    //1: 指纹 2: 密码 3: 卡
      "inout":0,  //0: 主机进门 1: 子机出门
      "event":1
    }
  ]
  .....
}
```

当数据为空时，考勤机返回:

```
{
  "ret":"getnewlog",
  "sn":"ZX12345678",
  "result":true,
  "count":0,
  "from":0,
  "to":0,
  "record":[]
}
```

失败:

```
{
  "ret":"getnewlog",
  "sn":"ZX12345678",
  "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",
  "sn":"ZX12345678",
  "result":true,
  "count":1000,
  "from":0,
  "to":49,
  "record":[
    {
      "enrollid":1,
      "time":"2016-03-25 13:49:30",
      "mode":0,    //1: 指纹 2: 密码 3: 卡
      "inout":0,  //0: 主机进门 1: 子机出门
      "event":0
    },
    {
      "enrollid":2,
      "time":"2016-03-25 13:49:30",
      "mode":0,    //1: 指纹 2: 密码 3: 卡
      "inout":0,  //0: 主机进门 1: 子机出门
      "event":1
    }
    .....
  ]
}
```

服务器应答:

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

考勤机回复第二个数据包:

```
{
  "ret":"getalllog",
```

```
"sn":"ZX12345678",
"result":true,
"count":1000,
"from":50,
"to":99,
"record":[
  {
    "enrollid":111,
    "time":"2016-03-25 13:49:30",
    "mode":0,    //1: 指纹 2: 密码 3: 卡
    "inout":0,   //0: 主机进门 1: 子机出门
    "event":0
  },
  {
    "enrollid":112,
    "time":"2016-03-25 13:49:30",
    "mode":0,    //1: 指纹 2: 密码 3: 卡
    "inout":0,   //0: 主机进门 1: 子机出门
    "event":1
  }
  .....
]
```

当数据为空时，考勤机返回：

```
{
  "ret":"getalllog",
  "sn":"ZX12345678",
  "result":true,
  "count":0,
  "from":0,
  "to":0,
  "record":[]
}
```

失败：

```
{
  "ret":"getalllog",
  "sn":"ZX12345678",
  "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",
    "sn": "ZX12345678",
    "result": true
  }
  失败:
  {
    "ret": "cleanlog",
    "sn": "ZX12345678",
    "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",
    "sn": "ZX12345678",
    "result": true
  }
  失败:
  {
    "ret": "initsys",
    "sn": "ZX12345678",
    "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",  
  "sn":"ZX12345678",  
  "result":true  
}
```

失败：

```
{  
  "ret":"cleanadmin",  
  "sn":"ZX12345678",  
  "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",  
  "sn":"ZX12345678",  
  "result":true  
}
```

失败：

```
{  
  "ret":"settime",  
  "sn":"ZX12345678",  
  "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 8
  "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, 如备注说明所示
  "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。
  "reverifypm":0          //重复确认时间 0~255 分钟
}
```

考勤机回复信息:

成功:

```
{
  "ret":"setdevinfo",
  "sn":"ZX12345678",
  "result":true
}
```

失败:

```
{
  "ret":"setdevinfo",
  "sn":"ZX12345678",
  "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,
  "reverifypm":5
}
```

备注:

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",
    "sn":"ZX12345678",
    "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。
    "reverifitime":0     //重复确认时间 0~255 分钟
}

```

失败：

```

{
    "ret":"getdevinfo",
    "sn":"ZX12345678",
    "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",
    "doornum":1           //这只适用于门禁控制器（1~4），因为门禁控制器最多有 4 个门。
                          //如果删除该项目，将打开所有门。普通门禁或考勤机不需要此项。
}

```

```
}
```

考勤机回复信息:

成功:

```
{
  "ret": "opendoor",
  "sn": "ZX12345678",
  "result": true
}
```

失败:

```
{
  "ret": "opendoor",
  "sn": "ZX12345678",
  "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, //当该用户有注册卡时, 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",

```

```

"sn":"ZX12345678",
"result":true
}
失败:
{
  "ret":"setdevlock",
  "sn":"ZX12345678",
  "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：

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"

该用户按了指纹，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",
  "sn":"ZX12345678",
  "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,  //当该用户有注册卡时，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": "getdevlock",
  "sn": "ZX12345678",
  "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",
  "sn": "ZX12345678",
  "result": true,
  "enrollid": 1,
  "weekzone": 1, //门禁控制器门 1 上方设置的周时段
  "weekzone2": 1, //仅用于门禁控制器门 2
  "weekzone3": 1, //仅适用于门禁控制器门 3
  "weekzone4": 1, //仅适用于门禁控制器门 4
  "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",
  "sn": "ZX12345678",
  "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, //仅用于门禁控制器门 1
    "weekzone2":1, //仅用于门禁控制器门 2
    "weekzone3":1, //仅用于门禁控制器门 3
    "weekzone4":1, //仅用于门禁控制器门 4
    "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",
  "sn":"ZX12345678",
  "result":true
}
```

失败:

```
{
  "ret":"setuserlock",
  "sn":"ZX12345678",
  "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",
  "sn":"ZX12345678",
  "result":true
}
```

失败:

```
{
  "ret":"deleteuserlock",
  "sn":"ZX12345678",
  "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",
  "sn":"ZX12345678",
  "result":true
}
```

失败:

```
{
  "ret":"cleanuserlock",
  "sn":"ZX12345678",
  "result":false,
  "reason":1
}
```

26. 获取时间

说明: 获取考勤机的日期和时间。

服务器发送格式:

```
{
  "cmd":"gettime"
}
```

终端响应消息:

成功:

```
{
  "ret":"gettime",
  "sn":"ZX12345678",
```

```
"time":"2025-06-20 15:30:00"
}
```

27. QR 码发送

考勤机发送消息（终端响应）：

```
{
  "cmd":"sendqrcode",
  "sn":"ZX12345678",
  "record":"123456"
}
```

28. QR 码服务器回复

说明：服务器接收终端发送的 QR 码验证请求后，返回验证结果及相关操作指令。

服务器响应消息：

```
{
  "ret":"sendqrcode",    // 响应命令字，与请求命令对应
  "sn":"ZX12345678",    // 设备序列号，标识请求来源终端
  "result":true,         // 验证结果（true：成功；false：失败）
  "access":1,           // 开门权限（1：允许开门；0：禁止开门）
  "enrollid":10,        // 用户注册 ID，匹配验证通过的用户
  "username":"tom",     // 用户姓名（UTF-8 编码）
  "message":"ok",       // 提示信息（如验证状态说明）
  "voice":"ok",         // 语音提示内容（支持中英文）
  "fontsize":32,        // 显示字体大小（仅消息栏模式支持，可选值：10,12,...,40）
  "text_color":[255,0,0] // 文字颜色（RGB 格式，消息栏模式有效）
}
```

29. 远程添加用户

服务器发送格式：

```
{
  "cmd":"adduser",      // 命令字（远程添加用户）
  "enrollid":1,         // 用户注册 ID（唯一标识）
  "backupnum":50,       // 初始信息类型（50：照片，0：指纹）
  "admin":0,           // 权限（0：普通用户，1：管理员）
  "name":"TEST",       // 用户名（UTF-8）
  "flag":10            // 注册标志（10：自动注册，终端生成特征）
}
```

自动注册（添加带照片的用户）：

```
{
  "cmd":"adduser",
  "enrollid":1001,
  "backupnum":50,      // 初始信息为照片
  "name":"张三",
  "flag":10           // 自动注册
}
```

```
}
```

考勤机回复消息（终端响应）：

成功：

```
{
  "ret": "adduser",          // 响应命令字
  "sn": "ZX12345678",       // 设备序列号
  "enrollid": 1001,          // 处理的用户 ID
  "result": true              // 操作结果
}
```

失败：

```
{
  "ret": "adduser",
  "sn": "ZX12345678",
  "enrollid": 1001,
  "result": false,
  "reason": 2                 // 错误码（如 2: ID 已存在）
}
```

30. 获取问卷参数

服务器发送格式：

```
{
  "cmd": "getquestionnaire", // 命令字
  "stn": true
}
```

考勤机回复信息：

成功：

```
{
  "ret": "getquestionnaire", // 命令字
  "sn": "ZX12345678",       // 序列号
  "result": true,             // 结果
  "title": "考勤事件选择",    // 问卷标题（显示顶部）
  "voice": "请选择考勤事件", // 语音提示（中英文）
  "errmsg": "请选择必填项",   // 错误提示（未选时显示）
  "radio": true,              // 单选模式（true: 单选，false: 多选）
  "optionflag": 0,            // 必填项标识（多选时使用）
  "usequestion": false,        // 启用问卷功能（true: 启用 false: 禁用）
  "useschedule": false,        // 启用时段控制（true: 启用 false: 禁用）
  "card": 0,                  // 刷卡启动问卷（0: 禁用，1: 启用）
  "items": ["上班", "下班", "加班上班", "加班下班"], // 问卷选项
  "schedules": ["00:01-11:12*1", "11:30-12:30*3", "13:00-19:00*4", "00:00-00:00*0", "00:00-00:00*0", "00:00-00:00*0", "00:00-00:00*0", "00:00-00:00*0", "00:00-00:00*0", "00:00-00:00*0"]
}
```

31. 获取设备信息

服务器发送格式:

```
{
  "cmd": "getdevcap"      // 命令字（获取设备容量信息）
}
```

考勤机回复信息:

成功:

```
{
  "ret": "getdevcap",      // 响应命令字
  "sn": "ZX12345678",     // 设备序列号
  "result": true,          // 结果
  "usersize": 15000,       // 用户容量（最大注册数）
  "facesize": 5000,        // 人脸容量（AI 设备）
  "fpsize": 10000,         // 指纹容量
  "palmsize": 0,           // 掌静脉容量（未支持则为 0）
  "cardsize": 15000,       // RFID 卡容量
  "pwdsize": 15000,        // 密码容量
  "logsize": 500000,       // 日志容量
  "useduser": 746,         // 已注册用户数
  "usedface": 12,          // 已注册人脸数
  "usedfp": 17,            // 已注册指纹数
  "usedpalm": 0,           // 已注册掌静脉数
  "usedcard": 703,         // 已注册 RFID 卡数
  "usedpwd": 9,            // 已注册密码数
  "usedlog": 135,          // 已存日志数
  "usednewlog": 135,       // 新日志数（未同步）
  "usedrtlog": 0           // 实时日志数
}
```

32. 设置问卷参数

服务器发送格式:

```
{
  "cmd": "setquestionnaire", // 命令字
  "title": "考勤事件选择",   // 问卷标题（显示顶部）
  "voice": "请选择考勤事件", // 语音提示（中英文）
  "errmsg": "请选择必填项",  // 错误提示（未选时显示）
  "radio": true,              // 单选模式（true: 单选, false: 多选）
  "optionflag": 0,            // 必填项标识（多选时使用）
  "usequestion": true,        // 启用问卷功能（true: 启用）
  "useschedule": true,        // 启用时段控制（true: 启用）
  "card": 0,                  // 刷卡启动问卷（0: 禁用, 1: 启用）
  "items": ["上班", "下班", "加班上班", "加班下班"], // 选项（最多 8 个多选, 16 个单选）
  "schedules": ["08:00-18:00*1", "00:00-00:00*0"] // 有效时段（*0 表示全天, *1-7 对应周一
```

至周日)

}

单选问卷示例（考勤事件）：

```
{
  "cmd":"setquestionnaire",
  "title":"请选择考勤类型",
  "radio":true,
  "items":["签到","签退","外出","返回"]
}
```

时间段示例：

```
{
  "useschedule":true,
  "schedules":["08:00-18:00*1", "08:00-18:00*2", "08:00-18:00*3", "08:00-18:00*4",
"08:00-18:00*5"]
}
```

考勤机回复消息：

成功：

```
{
  "ret":"setquestionnaire",
  "sn":"ZX12345678",
  "result":true
}
```

失败：

```
{
  "ret":"setquestionnaire",
  "sn":"ZX12345678",
  "result":false,
  "reason":1
}
```

33. 设置用户信息

服务器发送格式：

```
{
  "cmd":"setuserprofile",
  "enrollid":1,
  "profile":"message" （最大 200 字节）
}
```

// "enrollid":0：公共信息，

// "enrollid":1, 2、3.....：个人信息

考勤机回复消息：

```
{
  "ret":"setuserprofile",
  "sn":"ZX12345678",
  "enrollid":1,
```

```
"result":true
}
```

34. 获取用户信息

服务器发送格式:

```
{
  "cmd":"getuserprofile",
  "enrollid":1
}
```

考勤机回复消息:

成功:

```
{
  "ret":"getuserprofile",
  "sn":"ZX12345678",
  "enrollid":1,
  "record":"message",
  "result":true
}
```

35. 获取假期参数

服务器发送格式:

```
{
  "cmd":"getholiday",    // 命令字
  "stn":true             // 首包标识 (true: 开始获取, false: 继续获取)
}
```

考勤机回复信息:

成功:

```
{
  "ret":"getholiday",    // 响应命令字
  "sn":"ZX12345678",    // 设备序列号
  "result":true,         // 结果
  "holidays":[
    {
      "name":"元旦",    // 假期名称
      "startday":"01-01", // 开始日期 (月-日)
      "endday":"01-01",  // 结束日期
      "shift":0,         // 考勤班次 (0: 默认班次)
      "dayzone":0        // 关联天时段 ID (0: 默认时段)
    },
    {
      "name":"春节",
      "startday":"02-01",
      "endday":"02-07",
      "shift":0,
    }
  ]
}
```

```
        "dayzone":0
    }
]
}
```

36. 设置假期参数

服务器发送格式:

```
{
  "cmd":"setholiday",      // 命令字
  "holidays":[
    {
      "name":"国庆节",    // 假期名称
      "startday":"10-01",  // 开始日期
      "endday":"10-07",    // 结束日期
      "shift":1,           // 班次 ID (1: 早班)
      "dayzone":1          // 天时段 ID (1: 假期特殊时段)
    },
    {
      "name":"劳动节",
      "startday":"05-01",
      "endday":"05-03",
      "shift":1,
      "dayzone":1
    }
  ]
}
```

考勤机回复信息:

成功:

```
{
  "ret":"setholiday",
  "sn":"ZX12345678",
  "result":true
}
```

失败:

```
{
  "ret":"setholiday",
  "sn":"ZX12345678",
  "result":false,
  "reason":1
}
```

37. 下发用户信息（需 AI 人脸识别设备固件版本 v5.09 或 v2.09 及以上）

说明：服务器向终端下发完整用户信息，支持多因素认证配置即权限管理。

服务器发送格式：

```
{
  "cmd":"setuserinfo",          // 命令字
  "enrollid":1,                // 用户注册 ID（唯一标识）
  "name":"测试用户",          // 用户名（UTF-8 编码）
  "verifymode":8,              // 验证方式（枚举值说明如下）
  "department":"人力资源部",    // 部门名称
  "admin":0,                    // 权限等级（0: 普通用户 1: 管理员 2: 超级用户）
  "card":12345,                 // RFID 卡编号
  "pwd":123,                    // 密码（最多 8 位）
  "enable":1,                   // 启用状态（1: 启用 0: 禁用）
  "shiftid":0,                  // 自助考勤班次 ID
  "zoneid":0,                   // 周访问控制时段 ID
  "groupid":0,                  // 访问组 ID（用于组合开门）
  "access_times":0,             // 允许访问次数（0: 无限制）
  "postid":0,                   // 部门 ID（存在 department 时可忽略）
  "starttime":"2025-03-01 00:00:00", // 有效期开始时间
  "endtime":"2055-05-27 23:59:59", // 有效期结束时间
  "backupnum":0,                // 生物特征类型（0~9: 指纹 20~27: 静态人脸 40~41: 掌静脉）
  "record":"kajgksjgaglas"     // 特征数据（THbio3.0≤1620 字符，THbio1.0≤1024 字符）
}
```

verifymode 枚举值说明：

0: 设备默认验证模式（指纹 / 卡 / 密码任选）

1: 指纹验证

2: 密码验证

3: 卡验证

4: 指纹 + 密码组合验证

5: 指纹 + 卡组合验证

8: 人脸识别验证

9: 人脸 + 密码组合验证

10: 人脸 + 卡组合验证

11: 卡 + 密码组合验证

12: 指纹 + 人脸组合验证

13: 二维码验证

14: 人脸 +（卡或密码）组合验证

15: 身份证验证

16: 掌静脉验证

19: 1:N 快速验证

- 20: 任意验证方式
- 21: 掌静脉 + 人脸组合验证
- 22: 指纹 + 掌静脉组合验证
- 23: 卡 + 掌静脉组合验证
- 24: 掌静脉 + 密码组合验证

考勤机回复消息:

成功:

```
{
  "ret": "setuserinfo",      // 响应命令字
  "enrollid": 1,           // 处理的用户 ID
  "sn": "ZX12345678",      // 设备序列号
  "result": true           // 操作结果
}
```

38. 设置门状态

服务器发送格式:

```
{
  "cmd": "lockctrl",       // 命令字
  "fuc": 1                 // 功能码 (1: 强制常开 2: 强制常闭 3: 软件开门 4: 继电器复位 5: 机器重启 6: 取消报警)
}
```

考勤机回复消息:

成功:

```
{
  "ret": "lockctrl",       // 响应命令字
  "sn": "ZX12345678",      // 设备序列号
  "result": true           // 操作结果
}
```