PUSH 通讯协议

PUSH SDK

文档版本：V3.0.1

push 协议版本：V2.2.14

日期：2016 年 2 月

目　录

1. 摘要

Push 协议是基于超文本传输协议（HTTP）的基础上定义的数据协议，建立在 TCP/IP 连接上，主要应用于中控考勤、门禁等设备与服务器的数据交互， 定义了数据（用户信息、生物识别模板、考勤记录等）的传输格式、控制设备的命令格式；目前中控支持的服务器有 WDMS、 ZKECO、 ZKNET、 ZKBioSecurity3.0、 等，第三方支持的服务器有 印度 ESSL 等

1.1 特点

- 新数据主动上传
- 断点续传
- 所有行为都由客户端发起，比如上传数据、服务器下发的命令等

1.2 编码

协议中传输的数据大部分都是 ASCII 字符，但是个别的字段也涉及到编码的问题，比如用户姓名， 所以对该类型数据做如下规定

- 为中文时，使用 GB2312 编码
- 为其他语言时，使用 UTF-8 编码

目前涉及到该编码的数据如下：

- 用户信息表的用户姓名
- 短消息表的短消息内容

1.3 HTTP 协议简介

Push 协议是基于 HTTP 协议的基础上定义的数据协议，这里简单介绍下什么是 HTTP 协议，如果已经熟悉可跳过此部分。

HTTP 协议是一种请求/响应型的协议。客户端给服务器发送请求的格式是一个请求方法（request method），URI，协议版本号，然后紧接着一个包含请求修饰符（modifiers），客户端信息，和可能的消息主体的类 MIME（MIME-like）消息。服务器对请求端发送响应的格式是以一个状态行（status line），其后跟随一个包含服务器信息、实体元信息和可能的实体主体内容的类 MIME（MIME-like）的消息。其中状态行（status line）包含消息的协议版本号和一个成功或错误码。如下例子

客户端请求：

```
GET http://113.108.97.187:8081/iclock/accounts/login/?next=/iclock/data/
iclock/ HTTP/1.1
User-Agent: Fiddler
Host: 113.108.97.187:8081
```

服务器响应：

```
HTTP/1.1 200 OK
Server: nginx/0.8.12
Date: Fri, 10 Jul 2015 03:53:16 GMT
Content-Type: text/html; charset=utf-8
Transfer-Encoding: chunked
Connection: close
Content-Language: en
Expires: Fri, 10 Jul 2015 03:53:16 GMT
Vary: Cookie, Accept-Language
Last-Modified: Fri, 10 Jul 2015 03:53:16 GMT
ETag: "c487be9e924810a8c2e293dd7f5b0ab4"
Pragma: no-cache
Cache-Control: no-store
Set-Cookie: csrftoken=60fb55cedf203c197765688ca2d7bf9e; Max-Age=31449600;
 Path=/
Set-Cookie: sessionid=06d37fdc8f36490c701af2253af79f4a; Path=/

0
```

HTTP 通信通常发生在 TCP/IP 连接上。默认端口是 TCP 80，不过其它端口也可以使用。但并不排除 HTTP 协议会在其它协议之上被实现。HTTP 仅仅期望的是一个可靠的传输（译注：HTTP 一般建立在传输层协议之上）；所以任何提供这种保证的协议都可以被使用

2. 定义

文档中引用定义使用格式为：${ServerIP}

- ServerIP：服务器 IP 地址
- ServerPort：服务器端口
- XXX：未知值
- Value123……：值 123……值 n
- Required：必须存在

- Optional：可选
- SerialNumber：系列号
- NUL：null（）
- SP：空格
- LF：换行符（）
- HT：制表符（）
- DataRecord：数据记录
- CmdRecord：命令记录
- CmdID：命令编号
- CmdDesc：命令描述
- Pin：工号
- Time：考勤时间
- Status：考勤状态
- Verify：验证方式
- Workcode：workcode 编码
- Reserved：预留字段
- OpType：操作类型
- OpWho：操作者
- OpTime：操作时间
- BinaryData：二进制数据流
- TableName：数据表名
- SystemCmd：系统命令
- Key：键
- Value：值
- FilePath：文件路径
- URL：资源位置

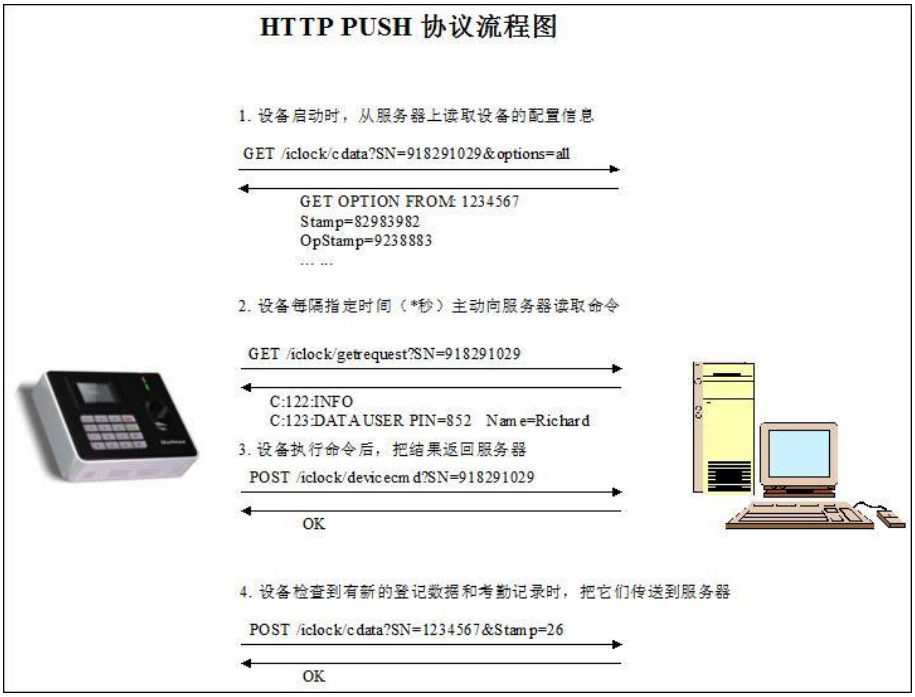
3. 功能

从客户端的角度来描述 Push 协议支持的功能

- 初始化信息交互
- 上传更新信息
- 上传数据
- 获取命令
- 命令回复
- 异地考勤

4. 流程

使用 Push 协议的客户端和服务器，必须由客户端先发起“初始化信息交互”请求成功之后，才能使用其他功能，比如上传数据、获取服务器命令、上传更新信息、回复服务器命令等，其中这些功能并没有先后顺序，取决于客户端应用程序的开发，如下图



5.初始化信息交互

客户端发起请求，将相应的配置信息发送给服务器，服务器接收到该请求，将相应的配置信息回复给客户端，只有当客户端获取到相应的配置信息，才能算交互成功；配置信息交互是按照规定好的格式进行的，具体如下

客户端请求消息

```
GET /iclock/cdata?SN=${SerialNumber}&options=all&pushver=${XXX}&language
=${XXX}&pushcommkey=${XXX} HTTP/1.1
Host: ${ServerIP}:${ServerPort}
……
```

注释：

HTTP 请求方法使用：GET 方法
URI 使用：/iclock/cdata
HTTP 协议版本使用：1.1
客户端配置信息：
SN：${Required}表示客户端的序列号
options：${Required}表示获取服务器配置参数，目前值只有 all
pushver：${Optional}表示 push 协议的版本，新开发的客户端必须支持且必须大于等于 2.2.14 版本
language：${Optional}表示客户端支持的语言，新开发的客户端最好支持，服务端可通过该参数知道目前设备是什么语言，见“附录 2”
pushcommkey：${Optional}表示客户端与服务器绑定的密文信息，软件通过此密文判断设备是否经过授权，不同设备值一般是不一样的，该参数需要服务器支持之后，客户端才需支持
Host 头域：${Required}
其他头域：${Optional}

服务器正常响应

```
HTTP/1.1 200 OK
Date: ${XXX}
Content-Length: ${XXX}
……

GET OPTION FROM: ${SerialNumber}${LF}${XXX}Stamp=${XXX}${LF}ErrorDelay=
${XXX}${LF}Delay=${XXX}${LF}TransTimes=${XXX}${LF}TransInterval=${XXX}
${LF}TransFlag=${XXX}${LF}TimeZone=${XXX}${LF}Realtime=${XXX}${LF}Encryp
t=${XXX}${LF}ServerVer=${XXX}
```

注释：

HTTP 状态行：使用标准的 HTTP 协议定义
HTTP 响应头域：

Date 头域：${Required}使用该头域来同步服务器时间，并且时间格式使用 GMT 格式，如 Date: Fri, 03 Jul 2015 06:53:01 GMT

Content-Length 头域：根据 HTTP 1.1 协议，一般使用该头域指定响应实体的数据长度，如果是在不确定响应实体的大小时，也支持 Transfer-Encoding: chunked，Content-Length 及 Transfer-Encoding 头域均是 HTTP 协议的标准定义，这里就不在详述
服务器端配置信息：

第 1 行必须为该描述：GET OPTION FROM: ${SerialNumber}并且使用${LF}间隔配置信息，

其中${SerialNumber}为客户端发起请求的系列号，配置信息是使用键值对的形式（key=value）并且不同配置之间使用${LF}间隔

${XXX}Stamp：各种数据类型的时间戳，目前支持如下

${XXX}	数据类型
ATTLOG	考勤记录
OPERLOG	操作日志
ATTPHOTO	考勤照片

时间戳标记设计目的：客户端上传数据时会同时上传相应的时间戳标记，服务器负责记录该标记，当设备重启时，客户端发起初始化信息交互请求，

服务器会将一系列的标记下发给客户端，从而达到客户端断点续传的功能

时间戳标记缺陷：由于修改时间是被允许的，并且时间产生变化的不确定性因素也是可能存在的，会造成客户端无法正确判断出哪些数据已经上传到服务器，

哪些数据未被上传，从而导致服务器数据丢失

服务器对时间戳的应用：目前服务器对时间戳标记只有 1 个应用，当服务器需要重新上传所有相应的数据时，就会将相应的时间戳标记设置为 0，功能参见“获取命令--控制命令--检查数据更新”

客户端废弃时间戳：新架构固件 Push 设计上不使用时间戳来标记数据上传截点，但是为了兼容老的服务器，对时间戳标记也做了发送，实际过程中也只是应用了当标记被设置为 0 时，

重新上传数据的功能，所以服务器并不需要区分客户端是否废弃时间戳

ErrorDelay：联网失败后客户端重新联接服务器的间隔时间（秒），建议设置 30~300 秒

Delay：正常联网时客户端联接服务器的间隔时间（秒），即客户端请求“获取命令”功能，建议设置 2~60 秒，需要快速响应时可设置小点，但是对服务器的压力会变大

TransTimes：客户端定时检查并传送新数据时间（时:分，24 小时格式），多个时间用分号分开，最多支持 10 个时间，如 TransTimes=00:00;14:00

TransInterval：客户端检查并传送新数据间隔时间（分钟），当设置为 0 时，不检查，如 TransInterval=1

TransFlag：客户端向服务器自动上传哪些数据的标识，设置的值支持两种格式

格式一：TransFlag=1111000000……，每一位代表一种数据类型，0–

表示禁止该数据类型自动上传，1–表示允许该数据类型自动上传

第几位	数据类型
1	考勤记录
2	操作日志
3	考勤照片
4	登记新指纹
5	登记新用户
6	指纹图片
7	修改用户信息
8	修改指纹
9	新登记人脸
10	用户照片

格式二：TransFlag=TransData AttLog${HT}OpLog${HT}AttPhoto……

字符串标示	数据类型
AttLog	考勤记录
OpLog	操作日志
AttPhoto	考勤照片
EnrollUser	登记新用户
ChgUser	修改用户信息
EnrollFP	登记新指纹
ChgFP	修改指纹
FPImag	指纹图片
FACE	新登记人脸
UserPic	用户照片

客户端新开发时：请同时支持两种格式，并且当服务器使用格式一下发，并且设置的值全部为 0（TransFlag=0000000000）时，表示仅支持上传考勤照片

服务端新开发时：支持格式二即可

TimeZone：指定服务器所在时区，主要为了同步服务器时间使用，参见[获取命令](#downloadcmd)的 Date 头域

取值为整数值，该值设计成支持整时区、半时区、1/4 时区

当 -12 < TimeZone < 12 时，表示整时区，单位为小时，如 TimeZone=4，表示东 4 区

当 TimeZone > 60 或 TimeZone < -60 时，可表示半时区、1/4 时区，单位为分钟，如 TimeZone=330，表示东 5 半区

Realtime：客户端是否实时传送新记录。 为 1 表示有新数据就传送到服务器，为 0 表示按照 TransTimes 和 TransInterval 规定的时间传送

Encrypt：是否加密传送数据，保留未使用

ServerVer：服务器支持的协议版本号及时间（时间格式待定），新开发的服务器必须设置为 2.2.14 以上

客户端应用：通过该参数，客户端可以知道目前对接的服务器是否支持一些新特性，当服务器不下发时，客户端默认服务器只支持最开始的协议版本 1.X.X；

目前通过该参数区分的新特性如下：

1、上传数据时同时上传该数据的时间戳标记，不同版本时间戳标记描述是不一致的。详见“上传数据”功能

示例

客户端请求：

```
GET /iclock/cdata?SN=0316144680030&options=all&pushver=2.2.14&language=83&pushcommkey=4a9594af164f2b9779b59e8554b5df26 HTTP/1.1
Host: 58.250.50.81:8011
User-Agent: iClock Proxy/1.09
Connection: close
Accept: */*
```

服务器响应：

```
HTTP/1.1 200 OK
Server: nginx/1.6.0
Date: Fri, 03 Jul 2015 06:53:01 GMT
Content-Type: text/plain
Content-Length: 190
Connection: close
Pragma: no-cache
Cache-Control: no-store

GET OPTION FROM: 0316144680030
ATTLOGStamp=None
OPERLOGStamp=9999
ATTPHOTOStamp=None
ErrorDelay=30
Delay=10
TransTimes=00:00;14:05
TransInterval=1
TransFlag=TransData AttLog  OpLog   AttPhoto    EnrollUser  ChgUser EnrollFP    ChgFP   UserPic
TimeZone=8
Realtime=1
Encrypt=None
```

6.上传更新信息

该功能复用获取命令请求，在其 URL 上加入参数，主要上传客户端的固件版本号、登记用户数、登记指纹数、考勤记录数、设备 IP 地址、指纹算法版本、人脸算法版本、注册人脸所需人脸个数、登记人脸数、设备支持功能标示信息

客户端请求消息

```
Get /iclock/getrequest?SN=${SerialNumber}&INFO=${Value1},${Value2},${Val
ue3},${Value4},${Value5},${Value6},${Value7},${Value8},${Value9},${Value
10}
Host: ${ServerIP}:${ServerPort}
……
```

注释：

HTTP 请求方法使用：GET 方法
URI 使用：/iclock/getrequest
HTTP 协议版本使用：1.1
客户端配置信息：

SN：${Required}表示客户端的序列号
${Value1}：固件版本号
${Value2}：登记用户数
${Value3}：登记指纹数
${Value4}：考勤记录数
${Value5}：设备 IP 地址
${Value6}：指纹算法版本
${Value7}：人脸算法版本
${Value8}：注册人脸所需人脸个数
${Value9}：登记人脸数
${Value10}：设备支持功能标示，格式：101，每一位代表一种功能，0–表示不支持该功能，1–表示支持该功能

第几位	功能描述
1	指纹功能
2	人脸功能
3	用户照片功能

Host 头域：${Required}
其他头域：${Optional}

服务器响应参见获取命令

示例

客户端请求：

```
GET /iclock/getrequest?SN=0316144680030&INFO=Ver%202.0.12-20150625,0,0,0,
192.168.16.27,10,7,15,0,111 HTTP/1.1
Host: 58.250.50.81:8011
User-Agent: iClock Proxy/1.09
Connection: close
Accept: */*
```

服务器响应：

```
HTTP/1.1 200 OK
```

```
Server: nginx/1.6.0
Date: Tue, 30 Jun 2015 01:24:26 GMT
Content-Type: text/plain
Content-Length: 2
Connection: close
Pragma: no-cache
Cache-Control: no-store

OK
```

7. 上传数据

具体哪些数据需要自动上传，服务器是可以控制的(详细见“初始化信息交互”的“TransFlag”参数)

7.1 上传的方式

实时上传
间隔上传
定时上传

实时，若支持实时，则间隔实时上传，设备本身默认支持，服务器是可以控制的(详细见“初始化信息交互”的“Realtime”参数)
间隔上传，具体的间隔时间服务器是可以控制的(详细见“初始化信息交互”的“TransInterval”参数)
定时上传，具体的上传时间点服务器是可以控制的(详细见“初始化信息交互”的“TransTimes”参数)

7.2 上传考勤记录

客户端请求消息

```
POST /iclock/cdata?SN=${SerialNumber}&table=ATTLOG&Stamp=${XXX} HTTP/1.1
Host: ${ServerIP}:${ServerPort}
Content-Length: ${XXX}
……

${DataRecord}
```

注释：

HTTP 请求方法使用：POST 方法
URI 使用：/iclock/cdata
HTTP 协议版本使用：1.1
客户端配置信息：
SN：${Required}表示客户端的序列号
table=ATTLOG：${Required}表示上传的数据为考勤记录
Stamp：${Optional}表示考勤记录上传到服务器的最新时间戳(详细见“初始化信息交互”的“Stamp”或者“ATTLOGStamp”参数)
Host 头域：${Required}
Content-Length 头域：${Required}
其他头域：${Optional}
请求实体：${DataRecord}，考勤记录数据，数据格式如下
${Pin}${HT}${Time}${HT}${Status}${HT}${Verify}${HT}${Workcode}${HT}${Reserved}${HT}${Reserved}
注：
${Time}：验证时间，格式为 XXXX-XX-XX XX:XX:XX，如 2015-07-29 11:11:11
多条记录之间使用${LF}连接

服务器正常响应消息

```
HTTP/1.1 200 OK
Content-Length: ${XXX}
……

OK:${XXX}
```

注释：

HTTP 状态行：使用标准的 HTTP 协议定义
HTTP 响应头域：
Content-Length 头域：根据 HTTP 1.1 协议，一般使用该头域指定响应实体的数据长度，如果是在不确定响应实体的大小时，也支持 Transfer-Encoding: chunked，Content-Length 及 Transfer-Encoding 头域均是 HTTP 协议的标准定义，这里就不在详述
响应实体：当服务器接收数据正常并处理成功时回复 OK:${XXX}，其中${XXX}表示成功处理的记录条数，当出错时，回复错误描述即可

示例

客户端请求：

```
POST /iclock/cdata?SN=0316144680030&table=ATTLOG&Stamp=9999 HTTP/1.1
Host: 58.250.50.81:8011
User-Agent: iClock Proxy/1.09
Connection: close
Accept: */*
Content-Length: 315
```

```
1452    2015-07-30 15:16:28 0   1   0   0   0
1452    2015-07-30 15:16:29 0   1   0   0   0
1452    2015-07-30 15:16:30 0   1   0   0   0
1452    2015-07-30 15:16:31 0   1   0   0   0
1452    2015-07-30 15:16:33 0   1   0   0   0
1452    2015-07-30 15:16:34 0   1   0   0   0
1452    2015-07-30 15:16:35 0   1   0   0   0
8965    2015-07-30 15:16:36 0   1   0   0   0
8965    2015-07-30 15:16:37 0   1   0   0   0
```

服务器响应：

```
HTTP/1.1 200 OK
Server: nginx/1.6.0
Date: Thu, 30 Jul 2015 07:25:38 GMT
Content-Type: text/plain
Content-Length: 4
Connection: close
Pragma: no-cache
Cache-Control: no-store

OK:9
```

7.3 上传考勤照片

初始化信息交换服务器下发的配置 ServerVer 参数大于等于 2.2.14 版本

客户端请求消息

```
POST /iclock/cdata?SN=${SerialNumber}&table=ATTPHOTO&Stamp=${XXX} HTTP/1.
1
Host: ${ServerIP}:${ServerPort}
Content-Length: ${XXX}
……

${DataRecord}
```

注释：

HTTP 请求方法使用：POST 方法
URI 使用：/iclock/fdata 或/iclock/cdata
HTTP 协议版本使用：1.1
客户端配置信息：
    SN：${Required}表示客户端的序列号

table=ATTPHOTO：${Required}

Stamp：${Optional}表示考勤照片到服务器的最新时间戳(详细见“初始化信息交互”的“ATTPHOTOStamp”参数)

Host 头域：${Required}

Content-Length 头域：${Required}

其他头域：${Optional}

请求实体：${DataRecord}，考勤照片数据，数据格式如下

PIN=${XXX}${LF}SN=${SerialNumber}${LF}size=${XXX}${LF}CMD=uploadphoto${NUL}${BinaryData}

注：

PIN=${XXX}：考勤照片的文件名，目前只支持 jpg 格式

SN=${XXX}：客户端系列号

size=${XXX}：考勤照片原始大小

${BinaryData}：原始图片二进制数据流

不支持多条记录传输

服务器正常响应消息

```
HTTP/1.1 200 OK
Content-Length: ${XXX}
……

OK
```

注释：

HTTP 状态行：使用标准的 HTTP 协议定义

HTTP 响应头域：

Content-Length 头域：根据 HTTP 1.1 协议，一般使用该头域指定响应实体的数据长度，如果是在不确定响应实体的大小时，也支持 Transfer-Encoding: chunked，Content-Length 及 Transfer-Encoding 头域均是 HTTP 协议的标准定义，这里就不在详述

响应实体：当服务器接收数据正常并处理成功时回复 OK，当出错时，回复错误描述即可

示例

客户端请求：

```
POST /iclock/cdata?SN=0316144680030&table=ATTPHOTO&Stamp=9999 HTTP/1.1
Host: 58.250.50.81:8011
User-Agent: iClock Proxy/1.09
Connection: close
Accept: */*
Content-Length: 1684

PIN=20150731103012-123.jpg SN=0316144680030 size=9512 CMD=uploadphoto${NUL}${BinaryData}
```

服务器响应：

```
HTTP/1.1 200 OK
Server: nginx/1.6.0
Date: Thu, 30 Jul 2015 07:25:38 GMT
Content-Type: text/plain
Content-Length: 2
Connection: close
Pragma: no-cache
Cache-Control: no-store

OK
```

7.4 上传操作记录

初始化信息交换服务器下发的配置 ServerVer 参数大于等于 2.2.14 版本

客户端请求消息

```
POST /iclock/cdata?SN=${SerialNumber}&table=OPERLOG&Stamp=${XXX} HTTP/1.
1
Host: ${ServerIP}:${ServerPort}
Content-Length: ${XXX}
……

${DataRecord}
```

注释：

HTTP 请求方法使用：POST 方法
URI 使用：/iclock/cdata
HTTP 协议版本使用：1.1
客户端配置信息：
    SN：${Required}表示客户端的序列号
    table=OPERLOG：${Required}
    Stamp：${Optional}表示操作记录到服务器的最新时间戳(详细见“初始化信息交互”的“OPERLOGStamp”参数)
Host 头域：${Required}
Content-Length 头域：${Required}
其他头域：${Optional}
请求实体：${DataRecord}，操作记录数据，数据格式如下
    OPLOG${SP}${OpType}${HT}${OpWho}${HT}${OpTime}${HT}${Value1}${HT}${Value2}${HT}${Value3}${HT}${Reserved}
    ${OpType}：操作代码，见附录 3
    ${Value1}、${Value2}、${Value3}、${Reserved}：操作对象 1、2、3、4，见附

录 4

注：

多条记录之间使用${LF}连接

服务器正常响应消息

```
HTTP/1.1 200 OK
Content-Length: ${XXX}
……

OK:${XXX}
```

注释：

HTTP 状态行：使用标准的 HTTP 协议定义

HTTP 响应头域：

Content-Length 头域：根据 HTTP 1.1 协议，一般使用该头域指定响应实体的数据长度，如果是在不确定响应实体的大小时，也支持 Transfer-Encoding: chunked，Content-Length 及 Transfer-Encoding 头域均是 HTTP 协议的标准定义，这里就不在详述

响应实体：当服务器接收数据正常并处理成功时回复 OK:${XXX}，其中${XXX}表示成功处理的记录条数，当出错时，回复错误描述即可

示例

客户端请求：

```
POST /iclock/cdata?SN=0316144680030&table=OPERLOG&Stamp=9999 HTTP/1.1
Host: 58.250.50.81:8011
User-Agent: iClock Proxy/1.09
Connection: close
Accept: */*
Content-Length: 166

OPLOG 4 14  2015-07-30 10:22:34 0   0   0   0
```

服务器响应：

```
HTTP/1.1 200 OK
Server: nginx/1.6.0
Date: Thu, 30 Jul 2015 07:25:38 GMT
Content-Type: text/plain
Content-Length: 3
Connection: close
Pragma: no-cache
Cache-Control: no-store

OK:1
```

7.5 上传用户信息

初始化信息交换服务器下发的配置 ServerVer 参数大于等于 2.2.14 版本

客户端请求消息

```
POST /iclock/cdata?SN=${SerialNumber}&table=OPERLOG&Stamp=${XXX} HTTP/1.
1
Host: ${ServerIP}:${ServerPort}
Content-Length: ${XXX}
……

${DataRecord}
```

注释：

HTTP 请求方法使用：POST 方法
URI 使用：/iclock/cdata
HTTP 协议版本使用：1.1
客户端配置信息：
SN：${Required}表示客户端的序列号
table=OPERLOG：${Required}
Stamp：${Optional}表示用户信息到服务器的最新时间戳(详细见“初始化信息交互”的“OPERLOGStamp”参数)
Host 头域：${Required}
Content-Length 头域：${Required}
其他头域：${Optional}
请求实体：${DataRecord}，用户信息数据，数据格式如下
USER${SP}PIN=${XXX}${HT}Name=${XXX}${HT}Pri=${XXX}${HT}Passwd=${XXX}${HT}Card=${XXX}${HT}Grp=${XXX}${HT}TZ=${XXX}
注：
Name=${XXX}：用户姓名，当设备为中文时，使用的是 GB2312 编码，其他语言时，使用 UTF-8 编码
Card=${XXX}：卡号，值支持两种格式
a、十六进制数据，格式为[%02x%02x%02x%02x]，从左到右表示第 1、2、3、4 个字节，如卡号为 123456789，则为：Card=[15CD5B07]
b、字符串数据，如卡号为 123456789，则为：Card=123456789
TZ=${XXX}：用户使用的时间段编号信息，值格式为
XXXXXXXXXXXXXXXX，1 到 4 字符描述是否使用组时间段，5 到 8 字符描述使用个人时间段 1，9 到 12 字符描述使用个人时间段 2，13 到 16 字符描述使用个人时间段 3
如：0000000000000000，表示使用组时间段
0001000200000000，表示使用个人时间段，且个人时间段 1 使用时间段编号 2 的时间信息
0001000200010000，表示使用个人时间段，且个人时间段 1 使用时间段编

号 2 的时间信息，个人时间段 2 使用时间段编号 1 的时间信息
多条记录之间使用${LF}连接

服务器正常响应消息

```
HTTP/1.1 200 OK
Content-Length: ${XXX}
……

OK:${XXX}
```

注释：

HTTP 状态行：使用标准的 HTTP 协议定义
HTTP 响应头域：
Content-Length 头域：根据 HTTP 1.1 协议，一般使用该头域指定响应实体的数据长度，如果是在不确定响应实体的大小时，也支持 Transfer-Encoding: chunked，Content-Length 及 Transfer-Encoding 头域均是 HTTP 协议的标准定义，这里就不在详述
响应实体：当服务器接收数据正常并处理成功时回复 OK:${XXX}，其中${XXX}表示成功处理的记录条数，当出错时，回复错误描述即可

示例

客户端请求：

```
POST /iclock/cdata?SN=0316144680030&table=OPERLOG&Stamp=9999 HTTP/1.1
Host: 58.250.50.81:8011
User-Agent: iClock Proxy/1.09
Connection: close
Accept: */*
Content-Length: 166

USER PIN=36234  Name=36234  Pri=0   Passwd= Card=133440 Grp=1   TZ=00010
00000000000
USER PIN=36235  Name=36235  Pri=0   Passwd= Card=133441 Grp=1   TZ=00010
00000000000
```

服务器响应：

```
HTTP/1.1 200 OK
Server: nginx/1.6.0
Date: Thu, 30 Jul 2015 07:25:38 GMT
Content-Type: text/plain
Content-Length: 4
Connection: close
Pragma: no-cache
Cache-Control: no-store

OK:2
```

7.6 上传指纹模版

初始化信息交换服务器下发的配置 ServerVer 参数大于等于 2.2.14 版本

客户端请求消息

```
POST /iclock/cdata?SN=${SerialNumber}&table=OPERLOG&Stamp=${XXX} HTTP/1.
1
Host: ${ServerIP}:${ServerPort}
Content-Length: ${XXX}
……

${DataRecord}
```

注释：

HTTP 请求方法使用：POST 方法
URI 使用：/iclock/cdata
HTTP 协议版本使用：1.1
客户端配置信息：
    SN：${Required}表示客户端的序列号
    table=OPERLOG：${Required}
    Stamp：${Optional}表示指纹模版到服务器的最新时间戳(详细见“初始化信息交互”的“OPERLOGStamp”参数)
Host 头域：${Required}
Content-Length 头域：${Required}
其他头域：${Optional}
请求实体：${DataRecord}，指纹模版数据，数据格式如下
    FP${SP}PIN=${XXX}${HT}FID=${XXX}${HT}Size=${XXX}${HT}Valid=${XXX}${HT}TMP=${XXX}
    注：
    Size=${XXX}：指纹模版 base64 编码之后的长度
    TMP=${XXX}：传输指纹模版时，需要对原始二进制指纹模版进行 base64 编码
    多条记录之间使用${LF}连接

服务器正常响应消息

```
HTTP/1.1 200 OK
Content-Length: ${XXX}
……

OK:${XXX}
```

注释：

HTTP 状态行：使用标准的 HTTP 协议定义
HTTP 响应头域：

Content-Length 头域：根据 HTTP 1.1 协议，一般使用该头域指定响应实体的数据长度，如果是在不确定响应实体的大小时，也支持 Transfer-Encoding: chunked，Content-Length 及 Transfer-Encoding 头域均是 HTTP 协议的标准定义，这里就不在详述
响应实体：当服务器接收数据正常并处理成功时回复 OK:${XXX}，其中${XXX}表示成功处理的记录条数，当出错时，回复错误描述即可

示例

客户端请求：

```
POST /iclock/cdata?SN=0316144680030&table=OPERLOG&Stamp=9999 HTTP/1.1
Host: 58.250.50.81:8011
User-Agent: iClock Proxy/1.09
Connection: close
Accept: */*
Content-Length: 4950

FP PIN=2    FID=0    Size=1124    Valid=1 TMP=SghTUzIxAAADS00ECAUHCc7QAAAn
SnkBAAAAg/YUfEsyAIEPHgH6ALFHRQBBAPkP8wBAS2UPEwBTACYPe0tYAHkIjACuAHdIeQBt
AGwEUAB1S20DhAB+AK8EXUuPAOoPJABwANVENQDCANsPZQDbSx8PbwDeACwPz0vjAJ8PdwAr
APFELAD5AMwPvQASSvMKMgAwAQkPSUE2DkcXQ0uCQ1B4AJT7GZuC3GyNySrvjoKT7X77SkYk
B9L6MQhMCV5G1PUR+T0FfPiGTqMABQHp+XgBhclzg397xf0iD5CkQAXvErv3q4PQZ940xfmX
zBb5bcher2e7PQkLAXyf8gJ78nP7iwFIQmrXcwKn31LfiwoBIDQBAjrbrAdLOBFwwv8ExVYS
tv7ABQBOE7+JCEt/GIBs/4SqDwPoJ4yHwMDEBcHDicB/DQCrLkbAwgnFcnwGAHn2g4iICQCs
NoPCnm4RS75CjJ0rgwFqwS4HADFDZmwEAwJcRjrABwA1jWTCtMX+whMAw4+JwYnC+//Aw3pC
wsK0wRIAxFKJB8PAsfzCw3WLwaENAz9besHBi4eqBwOPW4PBwsEoygBxKnvCwcLAwqx4WFIB
02WGwl86boCLw4b/ZMFx3ADXJ4FSi3X/wqt2whjD/wkAc261eMKKigQBG249lwUD0nP9QxEA
dbZwwYt5wcHCwME6wcMuBQCeeANl1gCQMWjC/v+DwL90UbQCAIF7cMLNAJo2CMHA+/xTzQCC
y2jA/pPABMXTh2ZXEQEfnkCn/8OLwWVdSQMB18tDiwsBF8s9RZFcFUvqza3C/3gAwMOMw1lx
Z8IaxeXM7cFyxMDCpLvAdCDD/n4HAKMcHsMH/gsAcNreO/z/tfxC/wsAdBvk/bf9/T5KFAAM
4KfPxMHFjP/CB8DDi8FzCgBs4p+EwhHBGwDr4qQEUcA2wsPBwMF4B8DCEML+wAQAei/wH1wB
6/Gma8AFxI2IwIPAg1gYxej46Ut8n8PBwqrBbgYJEHsO4vk8/cMkBRCIF1NpxhCGe0jBERDT
MGxri4/CrGb/FBAaMqiLccDAwMTFAMLAi8PB/8HAA9WLOgj/DxDPPqtA/8GJtcRCDBDEj55B
i/7JyZMEEdFRQ7T4
FP PIN=2    FID=1    Size=2120    Valid=1 TMP=T3dTUzIxAAAGNDsECAUHCc7QAAAe
NWkBAAAAhtlDsjQjAKIPYgDiAHo7NgA1AHAPZgBSNKQPdwBWAOAPVjRYABMPsACiADM76wB8
ALkO2QB4NFgPVQCCAP0O2TSTALoOYABeAEo7TgCdAFMPwQGmNNgNwgCiAH0PnDSpAEQPCgFu
AEI5eQCtAEcPIwCpNMoNEgGwAPkNvTSyADoP+ABxAEY/KgC1ANUPVQC/NMMP7gC8AAwNHTTA
AM0O9AAEAEc5JgDCANQO7QDKNN4OzgDOAH4OHjTQANUODgAdAMc6ZgDgAEsPNgDmNMMOKQDi
AKMOEjTlAOUOzgAhAM86QQDmAGUPRQDoNFIPaQDwAAIP6DT3AD8OBgEzALk6uAD3AMQOCAAH
NTQOPwAHAaEPDTUGAUIOtQDMAUY7oQAMAU8PEAAJNbQO8gAZAXgPxDQfATsOCgHaAcQ63AAp
ATYOZAAqNToMmAAvAY0M/TQuAcANyQD1AcY60gA2AT8OaAA/NcELOgA7ASYPlDQ8AWQLJQCE
AfQ7hQBEAd8OKwBCNb8ORABMASoPdDROAeUOnACUAco5Lv0jb0N/vw8xbVZpOl8G9+JPA0Zb
g6sF2hafgw2db/Kq3hanMJLdJ1p/HQLu48rxWbIqIj8Lee9U5wASmAL+8DIEdPjjI1MEKQj1
```

H1OD0qPInf2aGYFzANzakIW6/UoNkQruW6gFTYfW/PMF5zDAjd2O3PosZbvLGIVugNb5YXuL
NMx1+QARCTT6k7VQf7qL1IpZi2c0ufv9+NHwGXoLNOR66PcwC7wAWjsUDD0UBfargLTBFRd1
CKH4be/q65T3cYRJ+3fqD7LgB4KAmYEciic2SSdlEZmBvBIfM/iXpQJeAHuJr7rAC04HdXxw
dkBTDH5Bglr9MH1HtmT/BYXRkmT/c7TgcsH23gAPAw2o9IWBgqaEzH27xWgGLRFyFNOMc7Qw
fgGD9f4I/i+/dH0pivH+2AaHtkgGpfi9efR7vsJofaXaLA5M5Eu+oAJxBhkGHYN3sbSN2IMd
CCwHZ7FUej1fyfXU8mW17ByJhEmaQBKYMGKTkXwN/XAXnzZXgG6IPQ3n/JE4hAmKgZ4bpGpP
24wPjfRRa9Hpz8AEIEwBAungoQU0ZwUAwAwAaAcidMBEwDgKAH0IIctEQv4SAHzOEzP0/0NA
W8L9xgCfPx3/CgDDC+L/U8tKDQBnDAk4RvgMw/3AFgCNyRcwH1VYVDyfB8XQChnBVf8SADfU
/cbK/v7/O8H8BcL4y0gDAOQkLToLBgMb/TwwQQnFuCAZS1X+CwAm4vdKyf/AKwUAX+2AdiEB
US4A/yeD/VdyZWADADMztMAeNFc1APwwMDvA+WdZW//AxQTFND9ZZAgApVcwOlXG9AYAelgg
/ur/EzRUWRD+/v31wDEHTcE4GgADm+lV9Pz9wP0wKTpDxnhoEAAHaOL9O/vK/v7+/jQHxAVu
Z5X9wgUAsaw0TCYBCHLkR/4FISx1Oh8AAX7wBkcozy8wRTbC/j7CQBgCARCAVsHBABy0X40E
AOuAg3YANEWDaZ3CCMVYjwP6RjYEAD5KXI4wAUSPVlYFxUqJePxPHQAEkRL++GP8+/4j/v+O
/8fK/cFW/ykOxd+QDv5Xc8DCwJULBmufSTj//cDqBQZ/nlNBBwBQZEz4y/8yBADDpvHA+CQB
l6tDUlE+wccHwfsUAJyrhVVdyv3E///9/wX7+2MCAQevRsHDAQqbQf9SBgB4dUZD9AsAv7U3
QQXAxUQEALq2QGjdAB+Jx/7///7+0vz4HcH+/8H9/wX++TcBCL9AwQXF7cZ0/sLBCAD9AkD7
9Xb/IAACygUoxsvB/f/9+/04/fjK/v3AwP//Bf3E9MH9/v3+HcUP0PfD/MD+//86/fvP/SEr
S//AOz74MAAS3El9BMUl510xBgD/40AH/Hk/AWfmUP5VhCIANN3nQMDBV8AAOt5j/0AKAIA0
UEV0/P0GAOv6g8RAMRE9CmLAI8EQtDhCcwkQuAzyQPvJ+/8FEJwPlf5wMxHWEzf9wgT/wjcQ
FRhQwQbV0xxyY/0DEMMihcEFJNwvNMEDEBY5RvUEEJVBVvmTBRY5Tf1RChBCpwDG9MHAwMBK
AA==
FP PIN=3 FID=0 Size=1592 Valid=1 TMP=TetTUzIxAAAEqKsECAUHCc7QAAAc
qWkBAAAAhFUooagrAKAPQADpAGKnzgAuAKkPtQA1qIsPnQBGAO0PTqhKAGIPlgCrADanNwBy
AEIPngB6qDQOfgCLAGENJKiNAD0PiABTAKimdACXAB0OVgCmqKwOigCnAFgOtqi0AKsPiABy
AJimyQC3AKsPjQDEqKwPngDBAFIPrqjDAKIPaQAPABGnrwDQAIwPAQDQqJsP6wDUAF4PXKjg
AJ8PwgAhAIOm2QDrAIwPTgDrqJIPQwD8AFYPdqj/AJQP9gDBAYmnugAGAXwPrwANqZMP6QAJ
AUYOWagTAYsP+AD2AWqmVAA0AX8PUQA8qYAP6ABDAbIPKdPn+18ZTgTm3GePXwtHgjd7mih7
jAplJYui+i6h1lRzIwfYrpLTDpojnwVODhLjqxleOdv2Afg5iMsIJlU6C2uTeQgwABW4PHhh
cX2DhRNxqCgS0PMpAGTwKaDYAkEWqQB88vGuWAIdE0oIJ5kGuRof7PcJDhDuZKcUGLX4xXs/
aVLRqOyV9v326BRpuqzzQhBmApMXuSc/DZL1wezA/p1XgAbS8Jr0nANWq9IICgnK9TsCQaxM
C9kDZQxXAjeIz/Xm8gv4DABRoH8C8fiq+fcX2VDY+5L5dfoD79dIz/XX7LfvZvOHr94H6QrL
9xbvuTys+2sgRAHHpSVRDQCpBCtrlmTEpgGQBxw4/55qYK0BcQgP/zHBAIWgEScTAF0Jxvz7
V/5HwP5rZ5QMBDYIHv5KwGKRBAQSCydZBQDDzifE7wUAtQ0nYNMAQLgBwP3+QV6hUl30CgAl
J+D8O/3EVf4wFwAqKiL9+Vf/MThTZcAEwY+gAWwsj5LDQAYElSxndMINAG0tI/1j/8L/XQzF
bDYhwJ9pwnsFxdI2j1MJAHQzHoNU+64BRkprwsS2DwTmS/38//wwkMPGxgwAnEorwjv/YGjA
wYQOAEmKZ8ckksLBwsDAzwCxwSj/wVjCcdQAU8VWxMPCwcIHw/tqwMDC/UsMxTlqfif9/MD9
/qoMBPRuT8LDxMBYwWiuAZdwMFLAwwBl3UHDw8DDB8Uzcu6RwsIGADmzQ8UgCQBleCfDiWEf
qA6Fyf////T++VX+/f7/wP4F/8RowItvBgBaQDTGIAwAe4ipwjvIzW2hfhUAFo8DQfpX//39
/S7/7nwIqByPQ8LCiAV+xKABI489jMK2BAQdkDSDFQCIV6v7IMfHw8TCwgb/xtTAhQUAdZvn
w/poCgAVoD2SuWQDqLe5IHSFCsURv5jCacHB/sPGAKZrFsAGAEzE4cF+rwFsyRPAhjsGBKbL
KcLAiAnFa8q2wv/Cw1MGxcjSv8HCRg0AWyEXftZHSQQA3O3KWQ2ojvADaEf/xgBDVRLABRAn
AMxHArh3AAl+wATVvwKoQAUQKQgTAPzGrhFuCQZpwMgQ/7qHdsDCQ/4FwRW4LhYJwf/COsJU
aMNDQBAQOuYJdFfAZmb+YAjV/C7YM2QJEQMtk/77Vf//UgUQWfH9UKcR9TdwLcCeUm2vEZA7
fVJdyRDg7nvA/2jC/gY/Drg7XAPAbcI7/cb6QwALQwAADKcPCQ==

服务器响应：

```
HTTP/1.1 200 OK
Server: nginx/1.6.0
Date: Thu, 30 Jul 2015 07:25:38 GMT
Content-Type: text/plain
Content-Length: 4
Connection: close
Pragma: no-cache
Cache-Control: no-store

OK:3
```

7.7 上传面部模版

初始化信息交换服务器下发的配置 ServerVer 参数大于等于 2.2.14 版本

客户端请求消息

```
POST /iclock/cdata?SN=${SerialNumber}&table=OPERLOG&Stamp=${XXX} HTTP/1.
1
Host: ${ServerIP}:${ServerPort}
Content-Length: ${XXX}
……

${DataRecord}
```

注释：

HTTP 请求方法使用：POST 方法
URI 使用：/iclock/cdata
HTTP 协议版本使用：1.1
客户端配置信息：
    SN：${Required}表示客户端的序列号
    table=OPERLOG：${Required}
    Stamp：${Optional}表示面部模版到服务器的最新时间戳(详细见“初始化信息交互”的“OPERLOGStamp”参数)
Host 头域：${Required}
Content-Length 头域：${Required}
其他头域：${Optional}
请求实体：${DataRecord}，面部模版数据，数据格式如下
    FACE${SP}PIN=${XXX}${HT}FID=${XXX}${HT}SIZE=${XXX}${HT}VALID=${XXX}${HT}TMP=${XXX}

注：

SIZE=${XXX}：面部模版 base64 编码之后的长度

TMP=${XXX}：传输面部模版时，需要在原始二进制面部模版前加上 16 个字节（内容随意）后在进行 base64 编码

多条记录之间使用${LF}连接

服务器正常响应消息

```
HTTP/1.1 200 OK
Content-Length: ${XXX}
……

OK:${XXX}
```

注释：

HTTP 状态行：使用标准的 HTTP 协议定义

HTTP 响应头域：

Content-Length 头域：根据 HTTP 1.1 协议，一般使用该头域指定响应实体的数据长度，如果是在不确定响应实体的大小时，也支持 Transfer-Encoding: chunked，Content-Length 及 Transfer-Encoding 头域均是 HTTP 协议的标准定义，这里就不在详述

响应实体：当服务器接收数据正常并处理成功时回复 OK:${XXX}，其中${XXX}表示成功处理的记录条数，当出错时，回复错误描述即可

示例

客户端请求：

```
POST /iclock/cdata?SN=0316144680030&table=OPERLOG&Stamp=9999 HTTP/1.1
Host: 58.250.50.81:8011
User-Agent: iClock Proxy/1.09
Connection: close
Accept: */*
Content-Length: 1684

FACE PIN=306    FID=2    SIZE=1648    VALID=1 TMP=AAAAAAAAAAAAAAAAAAAAAFpL
RmlYATFLFLToAUQBQ1Mg+fgXuia23BDrNtwSfgJ8g74H3YHmXlkFpgetB5eH5yXuBvMLoa6w
Sx9HNgK7RP80v1i+LLY8nCn7PXmD7w15Bp8N1wm/A78PowejZx9jJyWnBZ88K5wVfDDcNTji
fGIvox9iD8sf1g37B70Fk4WRl5RrKq8uD2MngRexMxk5cbDiH+c3xj+CV8Zf1idaDfWbkB8R
nwt/AV8Du0SvAddBywHMQ9MVysfFkNENfZD9FJ9jrnGeBD5Kcwp7CVySfJzOE+wZxjWFVY6f
greXHBd6B4ov4BXBX+GuIZ4pazBTINiG8kf4h/DHxGaFxYe+yh+O1slCDsPcweuB/SHnA+Un
qwG3AvnA88DZg/vhmaaV4dsWzwerBn1jLcN8wu/ErITiR+YHVsc1wy2wdaC5uEmxKbwZ+AGe
B5fFt6WLVa8kq/gvqqv8LvwsBAACAgIHAQABAAEGAUxyAQkcIP9SAAohGhchAQAAEAEAAQYF
ASBPAgYGB1YKAQEMBgAACg8HFQ0DDAoEAg8CCBQDAxtLDwM40CUFEBNVSQYDDRNJJg8CAAcJ
FBMMAQQiYA8MAwhZHAcEehMCACYEAAwACQsJBQAFBAYxAAknpwQGJ6EiBAUPIxwFBwQPOgQC
BAARGz8WCAIAGUQWBggLhyMIBJQlBAIiAgUEAgMPAwUACQAFUAABE/8EBwkQEgABCRECBwQF
FEYNCAcCFiJ3YwUACTpKLgwBDn8yDApjFAIBEwAAAwACBgEAAAECAUMBAwBCAAACAQIBAAEA
BAMBAQUEAQMCBR0x/z4CAhUhSk8GAAcIDhUBNgsLDQMBAQAAAAYDAAAAAAEEAQAHBwEGQWEI
```

```
AAJe/zQBAAYY/30GAAABAAICAgAABAMAAAEFFAgAAAoEAAEUAgABAQEABwEECQMWGAIBDSIH
ASH/HQQICIlQeBgYMCgsIAwEBDSB0IBEJAjLbQgkIB245BQYKCAAEAAEDAwEDDgcKAQIBAhwA
BB+4DAQOoiAFCAgXFwMBByNDCgYBABUQGSYDAwItYhMIATG9KQwLV0oCBBkEAQgBDggHAQAD
BwJDAxAS6AIGFxYXBgMiRxQNCgMXvCMOBQIXGrxABgkGJQ4YBQEQHxUJBUgNCgggAwIBAgMK
AAAAAgMCJwADAzUAAAEAAwAAAB4BAAEAAAAAAAAAA1j/JgQGAxf/fAMBAgkLAAEEAgIAAAAE
AAEAAwEAAAABAAQAAAAAAAABAAAAARgQAQABAjP/KQMAAAIDAQEBBAIEBAoCAAoUAgMBRAYC
ACoBAQAAAAsDAgAAAgEJAAMDHgABEiwNAQQnRA8DAwc5lDICAQAIETAbAAEGM00eAQApYCIK
A/81CwU5DAIGAAEEBQQABQACKAAGG2MEAxkpFQMBCTkKAAUCFSYJAQABDByEGwIBAjUrIQUD
O/8eCwSMRAQACAQABQEABgAAAAMAAA8AAAMuAAAAAwgCBQMFCQcCBBNECgMAAAQvZhABAAhZ
/DIBACL/XQgDJiEHAgwFAQEAAAcDAAAAAQEDAAACAAAAAwEAAgEFCQEBBAEHIAIAAQELFRQE
AwICS/8rBQEDRVQUBAYODwQAAAgAAAANAAAAAAAA
```

服务器响应：

```
HTTP/1.1 200 OK
Server: nginx/1.6.0
Date: Thu, 30 Jul 2015 07:25:38 GMT
Content-Type: text/plain
Content-Length: 4
Connection: close
Pragma: no-cache
Cache-Control: no-store

OK:1
```

7.8 上传用户照片

初始化信息交换服务器下发的配置 ServerVer 参数大于等于 2.2.14 版本

客户端请求消息

```
POST /iclock/cdata?SN=${SerialNumber}&table=OPERLOG&Stamp=${XXX} HTTP/1.
1
Host: ${ServerIP}:${ServerPort}
Content-Length: ${XXX}
……

${DataRecord}
```

注释：

HTTP 请求方法使用：POST 方法
URI 使用：/iclock/cdata
HTTP 协议版本使用：1.1

客户端配置信息：

SN：${Required}表示客户端的序列号

table=OPERLOG：${Required}

Stamp：${Optional}表示用户照片到服务器的最新时间戳(详细见“初始化信息交互”的“OPERLOGStamp”参数)

Host 头域：${Required}

Content-Length 头域：${Required}

其他头域：${Optional}

请求实体：${DataRecord}，用户照片数据，数据格式如下

USERPIC${SP}PIN=${XXX}${HT}FileName=${XXX}${HT}Size=${XXX}${HT}Content=${XXX}

注：

FileName=${XXX}：用户照片的文件名，目前只支持 jpg 格式

Content=${XXX}：传输用户照片时，需要对原始二进制用户照片进行 base64 编码

Size=${XXX}：用户照片 base64 编码之后的长度

多条记录之间使用${LF}连接

服务器正常响应消息

```
HTTP/1.1 200 OK
Content-Length: ${XXX}
……

OK:${XXX}
```

注释：

HTTP 状态行：使用标准的 HTTP 协议定义

HTTP 响应头域：

Content-Length 头域：根据 HTTP 1.1 协议，一般使用该头域指定响应实体的数据长度，如果是在不确定响应实体的大小时，也支持 Transfer-Encoding: chunked，Content-Length 及 Transfer-Encoding 头域均是 HTTP 协议的标准定义，这里就不在详述

响应实体：当服务器接收数据正常并处理成功时回复 OK:${XXX}，其中${XXX}表示成功处理的记录条数，当出错时，回复错误描述即可

示例

客户端请求：

```
POST /iclock/cdata?SN=0316144680030&table=OPERLOG&Stamp=9999 HTTP/1.1
Host: 58.250.50.81:8011
User-Agent: iClock Proxy/1.09
Connection: close
Accept: */*
Content-Length: 1684

USERPIC PIN=123 FileName=123.jpg    Size=10 Content=AAAAAAAAAA……
```

服务器响应：

```
HTTP/1.1 200 OK
Server: nginx/1.6.0
Date: Thu, 30 Jul 2015 07:25:38 GMT
Content-Type: text/plain
Content-Length: 4
Connection: close
Pragma: no-cache
Cache-Control: no-store

OK:1
```

8.获取命令

如果服务器需要对设备进行操作，需先生成命令格式，等待设备发起请求时，再将命令发送到设备，对于命令的执行结果见回复命令

客户端请求消息

```
Get /iclock/getrequest?SN=${SerialNumber}
Host: ${ServerIP}:${ServerPort}
……
```

注释：

HTTP 请求方法使用：GET 方法
URI 使用：/iclock/getrequest
HTTP 协议版本使用：1.1
客户端配置信息：
    SN：${Required}表示客户端的序列号
Host 头域：${Required}
其他头域：${Optional}

服务器正常响应消息

当无命令时，回复信息如下：

```
HTTP/1.1 200 OK
Date: ${XXX}
Content-Length: 2
……

OK
```

当有命令时，回复信息如下：

```
HTTP/1.1 200 OK
Date: ${XXX}
Content-Length: ${XXX}
……

${CmdRecord}
```

注释：

HTTP 状态行：使用标准的 HTTP 协议定义

HTTP 响应头域：

Date 头域：${Required}使用该头域来同步服务器时间，并且时间格式使用 GMT 格式，如 Date: Fri, 03 Jul 2015 06:53:01 GMT

Content-Length 头域：根据 HTTP 1.1 协议，一般使用该头域指定响应实体的数据长度，如果是在不确定响应实体的大小时，也支持 Transfer-Encoding: chunked，Content-Length 及 Transfer-Encoding 头域均是 HTTP 协议的标准定义，这里就不在详述

响应实体：${CmdRecord}，下发的命令记录，数据格式如下

C:${CmdID}:${CmdDesc}${SP}${XXX}

注：

${CmdID}：该命令编号是由服务器随机生成的，支持数字、字母，长度不超过 16，客户端回复命令时需带上该命令编号，详细见下面“回复命令”功能

${CmdDesc}：命令描述分为数据命令和控制命令，数据命令统一为“DATA”描述，详细见下面“数据命令”功能，各种控制命令为各种不同的描述

多条记录之间使用${LF}连接

8.1 DATA 命令

当服务器下发的命令中${CmdDesc}为“DATA”时，就认为该命令为数据命令，可以对客户端的数据进行增、删、改、查操作，但是具体的不同的业务数据可支持的操作是不一样的，下面将详述

8.1.1 UPDATE 子命令

新增或修改数据，是新增还是修改取决于客户端是否存在相应的数据，服务器无需关心，命令格式如下

```
C:${CmdID}:DATA${SP}UPDATE${SP}${TableName}${SP}${DataRecord}
```

说明：

UPDATE：使用该描述表示新增或修改数据操作

${TableName}：不同的业务数据表名，比如用户信息为 USERINFO，具体支持的数据如下描述

${DataRecord}：业务数据记录，使用 key=value 的形式，不同业务数据，key 描述是不一样的，具体如下描述

8.1.1.1 用户信息

命令格式如下

```
C:${CmdID}:DATA${SP}UPDATE${SP}USERINFO${SP}PIN=${XXX}${HT}Name=${XXX}
${HT}Passwd=${XXX}${HT}Card=${XXX}${HT}Grp=${XXX}${HT}TZ=${XXX}${HT}Pri=
${XXX}
```

说明：

PIN=${XXX}：用户工号

Name=${XXX}：用户姓名，当设备为中文时，使用的是 GB2312 编码，其他语言时，使用 UTF-8 编码

Passwd=${XXX}：密码

Card=${XXX}：卡号，值支持两种格式

a、十六进制数据，格式为[%02x%02x%02x%02x]，从左到右表示第 1、2、3、4 个字节，如卡号为 123456789，则为：Card=[15CD5B07]

b、字符串数据，如卡号为 123456789，则为：Card=123456789

Grp=${XXX}：用户所属组，默认属于 1 组

TZ=${XXX}：用户使用的时间段编号信息，值格式为

XXXXXXXXXXXXXXXX，1 到 4 字符描述是否使用组时间段，5 到 8 字符描述使用个人时间段 1，9 到 12 字符描述使用个人时间段 2，13 到 16 字符描述使用个人时间段 3

如：0000000000000000，表示使用组时间段

0001000200000000，表示使用个人时间段，且个人时间段 1 使用时间段编号 2 的时间信息

0001000200010000，表示使用个人时间段，且个人时间段 1 使用时间段编号 2 的时间信息，个人时间段 2 使用时间段编号 1 的时间信息

Pri=${XXX}：用户权限值，值意义如下

值	描述
0	普通用户
2	登记员
6	管理员
10	用户自定义
14	超级管理员

多条记录之间使用${LF}连接

命令执行结果如何回复见回复命令功能，Return 返回值见附录 1，返回内容格式如下：

```
ID=${XXX}&Return=${XXX}&CMD=DATA
```

8.1.1.2 指纹模版

命令格式如下

```
C:${CmdID}:DATA${SP}UPDATE${SP}FINGERTMP${SP}PIN=${XXX}${HT}FID=${XXX}
${HT}Size=${XXX}${HT}Valid=${XXX}${HT}TMP=${XXX}
```

说明：

`PIN=${XXX}`：用户工号

`FID=${XXX}`：手指编号，取值为 `0` 到 `9`

`Size=${XXX}`：指纹模版二进制数据经过 `base64` 编码之后的长度

`Valid=${XXX}`：描述模版有效性及胁迫标示，值意义如下：

值	描述
0	无效模版
1	正常模版
3	胁迫模版

`TMP=${XXX}`：传输指纹模版时，需要对原始二进制指纹模版进行 `base64` 编码

多条记录之间使用`${LF}`连接

命令执行结果如何回复见回复命令功能，Return 返回值见附录 1，返回内容格式如下：

```
ID=${XXX}&Return=${XXX}&CMD=DATA
```

8.1.1.3 面部模版

命令格式如下

```
C:${CmdID}:DATA${SP}UPDATE${SP}FACE${SP}PIN=${XXX}${HT}FID=${XXX}${HT}Va
lid=${XXX}${HT}Size=${XXX}${HT}TMP=${XXX}
```

说明：

`PIN=${XXX}`：用户工号

`FID=${XXX}`：面部模版编号，取值从 `0` 开始

`Size=${XXX}`：面部模版二进制数据经过 `base64` 编码之后的长度

`Valid=${XXX}`：面部模版有效性标示，值意义如下：

值	描述
0	无效模版
1	正常模版

`TMP=${XXX}`：传输面部模版时，需要对原始二进制面部模版进行 `base64` 编码

多条记录之间使用`${LF}`连接

命令执行结果如何回复见回复命令功能，Return 返回值见附录 1，返回内容格式如下：

```
ID=${XXX}&Return=${XXX}&CMD=DATA
```

8.1.1.4 用户照片

命令格式如下

```
C:${CmdID}:DATA${SP}UPDATE${SP}USERPIC${SP}PIN=${XXX}${HT}Size=${XXX}${HT}Content=${XXX}
```

说明：

PIN=${XXX}：用户工号

Size=${XXX}：用户照片二进制数据经过 base64 编码之后的长度

Content=${XXX}：传输用户照片时，需要对原始二进制用户照片进行 base64 编码

多条记录之间使用${LF}连接

命令执行结果如何回复见回复命令功能，Return 返回值见附录 1，返回内容格式如下：

```
ID=${XXX}&Return=${XXX}&CMD=DATA
```

8.1.1.5 短消息

命令格式如下

```
C:${CmdID}:DATA${SP}UPDATE${SP}SMS${SP}MSG=${XXX}${HT}TAG=${XXX}${HT}UID=${XXX}${HT}MIN=${XXX}${HT}StartTime=${XXX}
```

说明：

MSG=${XXX}：短消息内容，最大支持 320 个字节，当设备为中文时，使用的是 GB2312 编码，其他语言时，使用 UTF-8 编码

TAG=${XXX}：短消息类型，值意义如下：

值	描述
253	公共短消息
254	用户短消息
255	预留短消息

UID=${XXX}：短消息编号，只支持整数

MIN=${XXX}：短消息有效时长，单位分钟

StartTime=${XXX}：短消息生效开始时间，格式为 XXXX-XX-XX XX:XX:XX，如 2015-07-29 00:00:00

多条记录之间使用${LF}连接

命令执行结果如何回复见回复命令功能，Return 返回值见附录 1，返回内容格式如下：

```
ID=${XXX}&Return=${XXX}&CMD=DATA
```

8.1.1.6 个人短消息用户列表

命令格式如下

C:${CmdID}:DATA${SP}UPDATE${SP}USER_SMS${SP}PIN=${XXX}${HT}UID=${XXX}
说明：
PIN=${XXX}：用户工号
UID=${XXX}：短消息编号，只支持整数
多条记录之间使用${LF}连接

命令执行结果如何回复见回复命令功能，Return 返回值见附录 1，返回内容格式如下：

ID=${XXX}&Return=${XXX}&CMD=DATA

8.1.2 DELETE 子命令

删除数据，命令格式如下

C:${CmdID}:DATA${SP}DELETE${SP}${TableName}${SP}${DataRecord}
说明：
DELETE：使用该描述表示删除数据操作
${TableName}：不同的业务数据表名，比如用户信息为 USERINFO，具体支持的数据如下描述
${DataRecord}：删除数据的条件，不同业务数据，支持的条件不一样，具体如下描述

8.1.2.1 用户信息

命令格式如下

C:${CmdID}:DATA${SP}DELETE${SP}USERINFO${SP}PIN=${XXX}
说明：
PIN=${XXX}：用户工号
删除指定的用户信息，包括指纹模版、面部模版、用户照片等相关信息

命令执行结果如何回复见回复命令功能，Return 返回值见附录 1，返回内容格式如下：

ID=${XXX}&Return=${XXX}&CMD=DATA

8.1.2.2 指纹模版

命令格式如下

C:${CmdID}:DATA${SP}DELETE${SP}FINGERTMP${SP}PIN=${XXX}${HT}FID=${XXX}
说明：

PIN=${XXX}：用户工号
FID=${XXX}：手指编号，取值为 0 到 9
删除指定的指纹模版，当只传输 PIN 信息时，删除用户的所有指纹模版

命令执行结果如何回复见回复命令功能，Return 返回值见附录 1，返回内容格式如下：

ID=${XXX}&Return=${XXX}&CMD=DATA

8.1.2.3 面部模版

命令格式如下

C:${CmdID}:DATA${SP}DELETE${SP}FACE${SP}PIN=${XXX}
说明：
PIN=${XXX}：用户工号
删除指定用户的面部模版

命令执行结果如何回复见回复命令功能，Return 返回值见附录 1，返回内容格式如下：

ID=${XXX}&Return=${XXX}&CMD=DATA

8.1.2.4 用户照片

命令格式如下

C:${CmdID}:DATA${SP}DELETE${SP}USERPIC${SP}PIN=${XXX}
说明：
PIN=${XXX}：用户工号
删除指定用户的用户照片

命令执行结果如何回复见回复命令功能，Return 返回值见附录 1，返回内容格式如下：

ID=${XXX}&Return=${XXX}&CMD=DATA

8.1.2.5 短消息

命令格式如下

C:${CmdID}:DATA${SP}DELETE${SP}SMS${SP}UID=${XXX}
说明：
UID=${XXX}：短消息编号，只支持整数
多条记录之间使用${LF}连接

命令执行结果如何回复见回复命令功能，Return 返回值见附录 1，返回内容格式如下：

ID=${XXX}&Return=${XXX}&CMD=DATA

8.1.3 QUERY 子命令

查询数据，命令格式如下

C:${CmdID}:DATA${SP}QUERY${SP}${TableName}${SP}${DataRecord}
说明：
QUERY：使用该描述表示查询数据操作
${TableName}：不同的业务数据表名，比如用户信息为 USERINFO，具体支持的数据如下描述
${DataRecord}：查询数据的条件，不同业务数据，支持的条件不一样，具体如下描述

8.1.3.1 考勤记录

命令格式如下

C:${CmdID}:DATA${SP}QUERY${SP}ATTLOG${SP}StartTime=${XXX}${HT}EndTime=${XXX}
说明：
StartTime=${XXX}：查询起始时间，格式为 XXXX-XX-XX XX:XX:XX，如 2015-07-29 00:00:00
EndTime=${XXX}：查询截止时间，格式为 XXXX-XX-XX XX:XX:XX，如 2015-07-29 23:59:59

命令执行结果如何回复见回复命令功能，Return 返回值见附录 1，返回内容格式如下：

ID=${XXX}&Return=${XXX}&CMD=DATA

查询指定时间段的考勤记录，如何上传详见“上传考勤记录”

8.1.3.2 考勤照片

命令格式如下

C:${CmdID}:DATA${SP}QUERY${SP}ATTPHOTO${SP}StartTime=${XXX}${HT}EndTime=${XXX}
说明：
StartTime=${XXX}：查询起始时间，格式为 XXXX-XX-XX XX:XX:XX，如 2015-07-29 00:00:00
EndTime=${XXX}：查询截止时间，格式为 XXXX-XX-XX XX:XX:XX，如 2015-07-29 23:59:59

命令执行结果如何回复见回复命令功能，Return 返回值见附录 1，返回内容格式如下：

```
ID=${XXX}&Return=${XXX}&CMD=DATA
```

查询指定时间段的考勤照片，如何上传详见“上传考勤照片”

8.1.3.3 用户信息

命令格式如下

```
C:${CmdID}:DATA${SP}QUERY${SP}USERINFO${SP}PIN=${XXX}
```

说明：

`PIN=${XXX}`：用户工号

命令执行结果如何回复见回复命令功能，Return 返回值见附录 1，返回内容格式如下：

```
ID=${XXX}&Return=${XXX}&CMD=DATA
```

查询指定用户的基本信息，如何上传详见“上传用户信息”

8.1.3.4 指纹模版

命令格式如下

```
C:${CmdID}:DATA${SP}QUERY${SP}FINGERTMP${SP}PIN=${XXX}${HT}FID=${XXX}
```

说明：

`PIN=${XXX}`：用户工号

`FID=${XXX}`：手指编号，取值为 `0` 到 `9`

命令执行结果如何回复见回复命令功能，Return 返回值见附录 1，返回内容格式如下：

```
ID=${XXX}&Return=${XXX}&CMD=DATA
```

查询指定用户的指纹模版信息，当只传输 PIN 信息时，查询指定用户的所有指纹模版信息，如何上传详见“上传指纹模版”

8.2 CLEAR 命令

8.2.1 清除考勤记录

清除客户端的考勤记录，命令格式如下

C:${CmdID}:CLEAR${SP}LOG
说明：
使用 CLEAR${SP}LOG 描述该命令

命令执行结果如何回复见回复命令功能，Return 返回值见附录 1，返回内容格式如下：

ID=${XXX}&Return=${XXX}&CMD=CLEAR_LOG
说明：
CMD=CLEAR_LOG：使用 CLEAR_LOG 描述该命令

8.2.2 清除考勤照片

清除客户端的考勤照片，命令格式如下

C:${CmdID}:CLEAR${SP}PHOTO
说明：
使用 CLEAR${SP}PHOTO 描述该命令

命令执行结果如何回复见回复命令功能，Return 返回值见附录 1，返回内容格式如下：

ID=${XXX}&Return=${XXX}&CMD=CLEAR_PHOTO
说明：
CMD=CLEAR_PHOTO：使用 CLEAR_PHOTO 描述该命令

8.2.3 清除全部数据

清除客户端的全部数据，命令格式如下

C:${CmdID}:CLEAR${SP}DATA
说明：
使用 CLEAR${SP}DATA 描述该命令

命令执行结果如何回复见回复命令功能，Return 返回值见附录 1，返回内容格式如下：

ID=${XXX}&Return=${XXX}&CMD=CLEAR_DATA
说明：
CMD=CLEAR_DATA：使用 CLEAR_DATA 描述该命令

8.3 检查命令

8.3.1 检查数据更新

要求客户端从服务器读取配置信息，详见"初始化信息交换"，并根据时间戳重新上传对应的数据到服务器，目前只支持服务器将时间戳重置为 0，比如设置参数 Stamp=0，客户端读取配置参数之后将重新上传考勤记录，命令格式如下

C:${CmdID}:CHECK
说明：
使用 CHECK 描述该命令

命令执行结果如何回复见回复命令功能，Return 返回值见附录 1，返回内容格式如下：

ID=${XXX}&Return=${XXX}&CMD=CHECK

8.3.2 检查并传送新数据

客户端立即检查是否有新的数据，并立即把新数据传送到服务器上，命令格式如下

C:${CmdID}:LOG
说明：
使用 LOG 描述该命令

命令执行结果如何回复见回复命令功能，Return 返回值见附录 1，返回内容格式如下：

ID=${XXX}&Return=${XXX}&CMD=LOG

8.3.3 考勤数据自动校对功能

由服务器下发校对某时间段之间的考勤记录，考勤设备上传开始时间，截止时间及记录总数，由服务器实现校对，命令格式如下

C:${CmdID}:VERIFY${SP}SUM${SP}ATTLOG${SP}StartTime=${XXX}${HT}EndTime=${XXX}
说明：
使用 VERIFY${SP}SUM 描述该命令
StartTime=${XXX}：服务器下发起始时间，格式为：XXXX-XX-XX XX:XX:XX，如 2015-07-29 00:00:00
EndTime=${XXX}：服务器下发截止时间，格式为：XXXX-XX-XX XX:XX:XX，如 2015-07-29 00:00:00

命令执行结果如何回复见回复命令功能，Return 返回值见附录 1，返回内容格式如下：

ID=${XXX}&Return=${XXX}&CMD=VERIFY${SP}SUM&StartTime=${XXX}&EndTime=${XXX}&AttlogSum=${XXX}
说明：
AttlogSum=${XXX}：在起止时间段内的考勤记录总数

8.4 配置选项命令

8.4.1 设置客户端的选项

设置客户端的配置信息，命令格式如下

C:${CmdID}:SET${SP}OPTION${SP}${Key}=${Value}
说明：
使用 SET${SP}OPTION 描述该命令
以键值对的形式设置配置信息，该命令只支持单一配置信息的设置

命令执行结果如何回复见回复命令功能，Return 返回值见附录 1，返回内容格式如下：

ID=${XXX}&Return=${XXX}&CMD=SET${SP}OPTION

8.4.2 客户端重新刷新选项

客户端重新加载配置信息，命令格式如下

C:${CmdID}:RELOAD${SP}OPTIONS
说明：
使用 RELOAD${SP}OPTIONS 描述该命令

命令执行结果如何回复见回复命令功能，Return 返回值见附录 1，返回内容格式如下：

ID=${XXX}&Return=${XXX}&CMD=RELOAD${SP}OPTIONS

8.4.3 发送客户端的信息到服务器

服务器获取客户端配置等信息，命令格式如下

C:${CmdID}:INFO
说明：
使用 INFO 描述该命令

命令执行结果如何回复见回复命令功能，Return 返回值见附录 1，返回内容格式如下：

ID=${XXX}&Return=${XXX}&CMD=INFO${LF}${Key}=${Value}${LF}${Key}=${Value}${LF}${Key}=${Value}${LF}${Key}=${Value}……

说明：
CMD=INFO 后面跟着具体的客户端配置信息，以键值对的形式组成

8.5 文件命令

8.5.1 取客户端内文件

客户端将服务器指定的文件发送到服务器上，命令格式如下

C:${CmdID}:GetFile${SP}${FilePath}
说明：
使用 GetFile 描述该命令
${FilePath}：客户端系统内的文件

命令执行结果如何回复见回复命令功能，Return 返回值见附录 1，返回内容格式如下：

ID=${XXX}${LF}SN=${SerialNumber}${LF}FILENAME=${XXX}${LF}CMD=GetFile${LF}
Return=${XXX}${LF}Content=${BinaryData}
说明：
Return=${XXX}：返回文件的大小
Content=${BinaryData}：传输文件的二进制数据流

8.5.2 发送文件到客户端

要求设备下载服务器上的文件，并保存到指定的文件中（如果是 tgz 文件，下载后将自动解压到 FilePath 指定目录，未指定目录则解压到/mnt/mtdblock 目录，其他格式文件需指定文件保存路径及文件名）。该文件必须由服务器以 HTTP 方式提供，并给出获取该文件的 URL，如果 URL 以"http://"开头，设备将把 URL 看着是完整的 URL 地址，否则，设备将把本服务器的/iclock/地址附加到指定的 URL 上，命令格式如下

C:${CmdID}:PutFile${SP}${URL}${HT}${FilePath}
说明：
使用 GetFile 描述该命令
${URL}：服务器上需要下载的文件地址
${FilePath}：文件存入客户端的目标路径
示例 1：PutFile file/fw/X938/main.tgz　main.tgz 或 PutFile file/fw/X938/main.tgz 将要求设备下载 http://server/iclock/file/fw/X938/main.tgz 并解压缩 main.tgz 到/mnt/mtdblock 文件夹中
示例 2：PutFile file/fw/X938/main.tgz　/mnt/将要求设备下载 http://server/iclock/file/fw/X938/main.tgz 并解压缩 main.tgz 到/mnt/文件夹中
示例 3：PutFile file/fw/X938/ssruser.dat /mnt/mtdblock/ssruser.dat 将要求设备下载 http://server/iclock/file/fw/X938/ssruser.dat 并保持为/mnt/mtdblock/ssruser.dat 文件

命令执行结果如何回复见回复命令功能，Return 返回值见附录 1，返回内容格式如下：

```
ID=${XXX}${LF}Return=${XXX}${LF}CMD=PutFile
```

说明：

`Return=${XXX}`：返回文件的大小

8.6 远程登记命令

8.6.1 登记用户指纹

由服务器发起，在客户端登记指纹，命令格式如下

```
C:${CmdID}:ENROLL_FP${SP}PIN=${XXX}${HT}FID=${XXX}${HT}RETRY=${XXX}${HT}
OVERWRITE=${XXX}
```

说明：

使用 `ENROLL_FP` 描述该命令

`PIN=${XXX}`：登记的工号

`FID=${XXX}`：登记的指纹编号

`RETRY=${XXX}`：若登记失败，需重试的次数

`OVERWRITE=${XXX}`：是否覆盖指纹，`0` 表示相应用户存在指纹则不覆盖指纹返回错误信息，`1` 表示相应用户存在指纹也覆盖该指纹

命令执行结果如何回复见回复命令功能，Return 返回值见附录 1，返回内容格式如下：

```
ID=${XXX}&Return=${XXX}&CMD=ENROLL_FP
```

8.7 控制命令

8.7.1 重新启动客户端

重启客户端，命令格式如下

```
C:${CmdID}:REBOOT
```

说明：

使用 `REBOOT` 描述该命令

命令执行结果如何回复见回复命令功能，Return 返回值见附录 1，返回内容格式如下：

```
ID=${XXX}&Return=${XXX}&CMD=REBOOT
```

8.7.2 输出打开门锁信号

门禁设备输出门锁打开信号，命令格式如下

C:${CmdID}:AC_UNLOCK
说明：
使用 AC_UNLOCK 描述该命令

命令执行结果如何回复见回复命令功能，Return 返回值见附录 1，返回内容格式如下：

ID=${XXX}&Return=${XXX}&CMD=AC_UNLOCK

8.7.3 取消报警信号输出

门禁设备取消报警信号输出，命令格式如下

C:${CmdID}:AC_UNALARM
说明：
使用 AC_UNALARM 描述该命令

命令执行结果如何回复见回复命令功能，Return 返回值见附录 1，返回内容格式如下：

ID=${XXX}&Return=${XXX}&CMD=AC_UNALARM

8.8 其他命令

8.8.1 执行系统命令

服务器下发客户端支持的操作系统命令，客户端将执行结果发送到服务器上，命令格式如下

C:${CmdID}:SHELL${SP}${SystemCmd}
说明：
使用 SHELL 描述该命令
${SystemCmd}：操作系统命令，比如，当客户端为 linux 系统，支持 ls 等

命令执行结果如何回复见回复命令功能，Return 返回值见附录 1，返回内容格式如下：

ID=${XXX}${LF}SN=${SerialNumber}${LF}Return=${XXX}${LF}CMD=Shell${LF}FILENAME=shellout.txt${LF}Content=${XXX}
说明：
Return=${XXX}：值为系统命令的返回值
Content=${XXX}：值为系统命令的输出内容

9.命令回复

客户端在获取到服务器下发的命令后，需要对相应的命令进行回复

客户端请求消息

```
POST /iclock/devicecmd?SN=${SerialNumber}
Host: ${ServerIP}:${ServerPort}
Content-Length: ${XXX}
……

${CmdRecord}
```

注释：

HTTP 请求方法使用：GET 方法
URI 使用：/iclock/devicecmd
HTTP 协议版本使用：1.1
客户端配置信息：
    SN：${Required}表示客户端的序列号
Host 头域：${Required}
Content-Length 头域：${Required}
其他头域：${Optional}
响应实体：${CmdRecord}，回复的命令记录，回复的内容都会包含 ID\Return\CMD 信息，含义如下
    ID：服务器下发命令的命令编号
    Return：客户端执行命令之后的返回结果
    CMD：服务器下发命令的命令描述
    少部分回复会包含其他信息，具体回复内容格式请看各个命令的说明
    多条命令回复记录之间使用${LF}连接

服务器正常响应消息

```
HTTP/1.1 200 OK
Date: ${XXX}
Content-Length: 2
……

OK
```

注释：

HTTP 状态行：使用标准的 HTTP 协议定义
HTTP 响应头域：
    Date 头域：${Required}使用该头域来同步服务器时间，并且时间格式使用 GMT 格式，如 Date: Fri, 03 Jul 2015 06:53:01 GMT

Content-Length 头域：根据 HTTP 1.1 协议，一般使用该头域指定响应实体的数据长度，如果是在不确定响应实体的大小时，也支持 Transfer-Encoding: chunked，Content-Length 及 Transfer-Encoding 头域均是 HTTP 协议的标准定义，这里就不在详述

示例

客户端请求：

```
POST /iclock/devicecmd?SN=0316144680030 HTTP/1.1
Host: 58.250.50.81:8011
User-Agent: iClock Proxy/1.09
Connection: close
Accept: */*
Content-Length: 143

ID=info8487&Return=0&CMD=DATA
ID=info8488&Return=0&CMD=DATA
ID=info8489&Return=0&CMD=DATA
ID=info7464&Return=0&CMD=DATA
ID=fp7464&Return=0&CMD=DATA
```

服务器响应：

```
HTTP/1.1 200 OK
Server: nginx/1.6.0
Date: Tue, 30 Jun 2015 01:24:48 GMT
Content-Type: text/plain
Content-Length: 2
Connection: close
Pragma: no-cache
Cache-Control: no-store

OK
```

10. 异地考勤

当用户出差到异地需要考勤时，考勤机内无该用户信息，用户可以通过异地考勤方式考勤。目前的应用场景：用户通过考勤机键盘直接输入工号，按 OK 键后，考勤机向服务器请求下发该用户全部信息（基本信息、指纹信息）。之后，用户考勤；下载用户信息后，在考勤机内保存一定时间。由参数确定保存时间，一段时间后删除该用户信息

客户端请求消息

```
GET /iclock/cdata?SN=${SerialNumber}&table=RemoteAtt&PIN=${XXX}
Host: ${ServerIP}:${ServerPort}
……
```

注释：

HTTP 请求方法使用：GET 方法
URI 使用：/iclock/cdata
HTTP 协议版本使用：1.1
客户端配置信息：
    SN：${Required}表示客户端的序列号
    Table=RemoteAtt：表示异地考勤获取用户信息
    PIN=${XXX}：需要获取的工号信息
Host 头域：${Required}
其他头域：${Optional}

服务器正常响应消息

当存在用户信息时，回复信息如下：

```
HTTP/1.1 200 OK
Date: ${XXX}
Content-Length: ${XXX}
……

DATA${SP}UPDATE${SP}USERINFO${SP}PIN=${XXX}${HT}Name=${XXX}${HT}Passwd=
${XXX}${HT}Card=${XXX}${HT}Grp=${XXX}${HT}TZ=${XXX}${HT}Pri=${XXX}
DATA${SP}UPDATE${SP}FINGERTMP${SP}PIN=${XXX}${HT}FID=${XXX}${HT}Size=${X
XX}${HT}Valid=${XXX}${HT}TMP=${XXX}
```

注释：响应实体的数据多条记录之间使用${LF}连接，具体数据格式见下发用户信息与下发指纹模板

11. 附录 1

错误码	描述
0	成功
2	登记用户指纹，对应用户的指纹已经存在
4	登记用户指纹，登记失败，通常由于指纹质量太差，或者三次按的指纹不一致

5	登记用户指纹，登记的指纹已经在指纹库中存在
6	登记用户指纹，取消登记
7	登记用户指纹，设备忙，无法登记
-1	参数错误
-2	传输用户照片数据与给定的 Size 不匹配
-3	读写错误
-9	传输的模板数据与给定的 Size 不匹配
-10	设备中不存在 PIN 所指定的用户
-11	非法的指纹模板格式
-12	非法的指纹模板
-1001	容量限制
-1002	设备不支持
-1003	命令执行超时
-1004	数据与设备配置不一致
-1005	设备忙
-1006	数据太长
-1007	内存错误

12. 附录 2

语言编号	意义
83	简体中文
69	英文
97	西班牙语
70	法语
66	阿拉伯语
80	葡萄牙语

82	俄语
71	德语
65	波斯语
76	泰语
73	印尼语
74	日本语
75	韩语
86	越南语
116	土耳其语
72	希伯来语
90	捷克语
68	荷兰语
105	意大利语
89	斯洛伐克语
103	希腊语
112	波兰语
84	繁体

13. 附录 3

操作代码	意义
0	开机
1	关机
2	验证失败
3	报警
4	进入菜单
5	更改设置

6	登记指纹
7	登记密码
8	登记 HID 卡
9	删除用户
10	删除指纹
11	删除密码
12	删除射频卡
13	清除数据
14	创建 MF 卡
15	登记 MF 卡
16	注册 MF 卡
17	删除 MF 卡注册
18	清除 MF 卡内容
19	把登记数据移到卡中
20	把卡中的数据复制到机器中
21	设置时间
22	出厂设置
23	删除进出记录
24	清除管理员权限
25	修改门禁组设置
26	修改用户门禁设置
27	修改门禁时间段
28	修改开锁组合设置
29	开锁
30	登记新用户
31	更改指纹属性
32	胁迫报警

14. 附录 4

操作代码	操作对象 1	操作对象 2	操作对象 3	操作对象 4
2	若为 1:1 验证，则为用户工号			
3	报警	报警的原因，见附录 5		
5	被修改的设置项的序号	新修改后的值		
6	用户的工号	指纹的序号	指纹模板的长度	
9	用户的工号			
10	用户的工号			
11	用户的工号			
12	用户的工号			

15. 附录 5

报警原因	意义
50	Door Close Detected
51	Door Open Detected
53	Out Door Button
54	Door Broken Accidentally
55	Machine Been Broken
58	Try Invalid Verification
65535	Alarm Cancelled

web: www.zkteco.com

author: xsen

mail: xsen@zkteco.com

date: 2015-10-09