



考勤 PUSH 通讯协议

PUSH SDK

文档版本：V3.6 日期：2019年08月

push 协议版本：V2.4.1

修改记录

日期	版本	描述	修改人	备注
2019/08/02	V3.6	1. 增加异常日志协议	李仙平	
2019/05/30	V3.5	1. 关于人证协议的增加： ① 上传身份证考勤记录协议 ② 上传身份证考勤记录照片协议 ③ 身份证黑名单下发协议	曹彦明	
2018/10/08	V3.4	1. 通信加密新增2条协议： ①交换公钥协议 ②交换因子协议 2. 支持通信加密版本说明： ①考勤PUSH: 2.4.0及以上 3. 通信加密详细说明见（附录8）	阎广田	
2018/8/9	V3.3	1. TransFlag增加2位： ①11（工作号码，WORKCODE） ②12（比对照片，BioPhoto） 2. 在线登记卡 ENROLL_MF 3. 在线登记人脸，掌纹（一体化模板） ENROLL_BIO 4. 上传一体化模板增加 可见光人脸Type=9 5. 在线升级 6. 后台验证 7. 增加以下参数： ①BioPhotoFun来标识比对照片 ②BioDataFun来标识可见光人脸模板 ③VisilightFun来标识可见光设备 8.新增比对照片相关协议	阎广田 汪国冬	
2017/11/10	V3.2	1. 对序列号进行说明 2. 增加初始化请求回复内容支持BIODATA表	李仙平	
2017/9/8	初版	1. 完善错误码列表，区分通用错误码及特殊命令错误 2. 新增一体化数据协议（目前应用于手掌模板） 3. 新增推送配置信息协议（需定制开启） 4. 新增用户个人验证方式设置 5. 新增数据打包上传协议（需定制开启） 6. 拓展 PUTFILE命令支持同步数据协议 7. 修正上传操作记录协议格式	梁贤森	

目录

1 摘要.....	6
2 特点.....	6
2.1 编码.....	6
2.2 HTTP协议简介.....	6
3 定义.....	7
4 功能.....	8
5 流程.....	9
6 初始化信息交互.....	10
7 交换公钥（支持通信加密的场合）.....	15
8 交换因子（支持通信加密的场合）.....	16
9 推送配置信息.....	17
10 上传更新信息.....	18
11 上传数据.....	20
11.1上传方式.....	20
11.2 上传考勤记录.....	20
11.3 上传考勤照片.....	22
11.4 上传操作记录.....	24
11.5 上传用户信息.....	26
11.6 上传身份证信息.....	29
11.7 上传身份证考勤记录.....	34
11.8 上传身份证考勤记录照片.....	36
11.9 上传指纹模板.....	37
11.10 上传面部模板.....	41
11.11 上传指静脉模板.....	44
11.12 上传一体化模板.....	47
11.13 上传用户照片.....	51
11.14 上传数据包.....	53
11.15 上传比对照片.....	55
11.16 上传异常日志.....	58
12 获取命令.....	60

12.1 DATA命令.....	62
12.2 UPDATA 子命令.....	62
12.2.1 用户信息.....	62
12.2.2 身份证信息.....	63
12.2.3 指纹模板.....	66
12.2.4 面部模板.....	67
12.2.5 指静脉模板.....	67
12.2.6 一体化模板.....	68
12.2.7 用户照片.....	68
12.2.8 比对照片.....	69
12.2.9 短消息.....	69
12.2.10 个人短消息用户列表.....	70
12.2.11 宣传照片.....	70
12.2.12 工作代码.....	71
12.2.13 快捷键.....	71
12.2.14 门禁组.....	72
12.2.15 门禁时间表.....	73
12.2.16 门禁节假日.....	74
12.2.17 门禁多组验证.....	74
12.2.18 身份证黑名单下发.....	75
12.3 DELETE 子命令.....	75
12.3.1 用户信息.....	75
12.3.2 指纹模板.....	75
12.3.3 面部模板.....	76
12.3.4 指静脉模板.....	76
12.3.5 一体化模板.....	76
12.3.6 用户照片.....	77
12.3.7 比对照片.....	77
12.3.8 短消息.....	77
12.3.9 工作代码.....	78
12.3.10 宣传照片.....	78
12.4 QUERY 子命令.....	78

12.4.1 考勤记录.....	79
12.4.2 考勤照片.....	79
12.4.3 用户信息.....	79
12.4.4 指纹模板.....	80
12.5 CLEAR命令.....	80
12.5.1 清除考勤记录.....	80
12.5.2 清除考勤照片.....	80
12.5.3 清除全部数据.....	81
12.6 检查命令.....	81
12.6.1 检查数据更新.....	81
12.6.2 检查并传送新数据.....	82
12.6.3 考勤数据自动校对功能.....	82
12.7 配置选项命令.....	82
12.7.1 设置客户端的选项.....	82
12.7.2 客户端重新刷新选项.....	83
12.7.3 发送客户端的信息到服务器.....	83
12.8 文件命令.....	83
12.8.1 取客户端内的文件.....	83
12.8.2 发送文件到客户端.....	84
12.9 远程登记命令.....	86
12.9.1 登记用户指纹.....	86
12.9.2 登记卡号.....	87
12.9.3 登记人脸，掌纹（一体化模板）.....	87
12.10 控制命令.....	88
12.10.1 重新启动客户端.....	88
12.10.2 输出打开门锁信号.....	88
12.10.3 取消报警信号输出.....	89
12.11 其他命令.....	89
12.11.1 执行系统命令.....	89
12.11.2 在线升级.....	89
12.11.3 后台验证.....	91
13 命令回复.....	92

14 异地考勤.....	95
15 附录.....	97
15.1 附录1.....	97
15.2 附录2.....	98
15.3 附录3.....	99
15.4 附录4.....	100
15.5 附录5.....	101
15.6 附录6.....	101
15.7 附录7.....	102
15.8 附录8.....	103
15.9 附录9.....	106

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。

2 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

2.1 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

2.2 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個成功或错误码。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。

3 定义

文档中引用定义使用格式为: $\text{\$}\{\text{ServerIP}\}$

●ServerIP: 服务器IP地址

●ServerPort: 服务器端口

●XXX: 未知值

- Value1\Value2\Value3.....\Valuen: 值1\值2\值3.....值n
- Required: 必须存在
- Optional: 可选
- SerialNumber: 系列号(可以为字母、数字、字母+数字组合)
- NUL: null (\0)
- SP: 空格
- LF: 换行符 (\n)
- HT: 制表符 (\t)
- 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

4 功能

客户端的角度来描述Push协议支持的功能

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

5 流程

使用Push协议的客户端和服务端，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



6 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}表示设备当前最新的push协议版本，新开发的客户端必须支持且必须大于等于2.2.14 版本，详见（附录6）

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}${LF}Encrypt=${XXX}${LF}ServerVer=${XXX}${LF}PushProtVer=${XXX}${LF}PushOptionsFlag=${XXX}${LF}PushOptions=${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

<code>\${XXX}</code>	数据类型
ATTLOG	考勤记录
OPERLOG	操作日志
ATTPHOTO	考勤照片
BIODATA	一体化模板
IDCARD	身份证信息
ERRORLOG	异常日志

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
11	工作号码
12	比对照片

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
WORKCODE	工作号码
BioPhoto	对比照片

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 $-12 < \text{TimeZone} < 12$ 时，表示整时区，单位为小时，如 $\text{TimeZone}=4$ ，表示东4区。

当 $\text{TimeZone} > 60$ 或 $\text{TimeZone} < -60$ 时，可表示半时区、1/4时区，单位为分钟，如 $\text{TimeZone}=330$ ，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: 是否加密传送数据标识，支持通信加密的场合，该参数需设置为1。

EncryptFlag: 数据加密的标识。

举例：EncryptFlag=10000000

第几位 数据类型

1 考勤记录

目前仅本协议2.3.0版本支持，并且只支持考勤记录加密，加密方式使用的是rc4。

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。

PushProtVer: 服务端依据哪个协议版本开发的，详见（附录6）。

PushOptionsFlag: 软件是否支持设备推送配置参数请求，0不支持，1支持，未设置时默认不支持。

PushOptions: 软件需要设备推送的参数列表，格式：PushOptions=key1,key2,key3,.....,keyN，如

PushOptions=FingerFunOn,FaceFunOn

ATTPHOTOBase64: 考勤照片base64标识。1: base64编码，其他场合不base64编码。

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

7 交换公钥（支持通信加密的场合）

该功能设备推送设备公钥，接收服务器返回的服务器公钥。

客户端请求消息

POST /iclock/exchange?SN=\${SerialNumber}&type=publickey

Host: \${ServerIP}:\${ServerPort}

Content-Length: \${XXX}

.....

PublicKey=\${XXX}

注释：

HTTP请求方法使用：POST方法

URI使用：/iclock/exchange

HTTP协议版本使用：1.1

Host头域：\${Required}

其他头域：\${Optional}

PublicKey：调用加密库返回的设备公钥。

服务器正常响应消息：

HTTP/1.1 200 OK

Server: \${XXX}

Set-Cookie: \${XXX}; Path=/; HttpOnly

Content-Type: application/push;charset=UTF-8

Content-Length: \${XXX}

Date: \${XXX}

PublicKey=\${XXX}

注释：

PublicKey：服务器返回的服务器公钥。

8 交换因子（支持通信加密的场合）

该功能设备推送设备因子，接收服务器返回的服务器因子。

客户端请求消息

POST /iclock/exchange?SN=\${SerialNumber}&type=factors

Host: \${ServerIP}:\${ServerPort}

Content-Length: \${XXX}

.....

Factors=\${XXX}

注释：

HTTP请求方法使用：POST方法

URI使用：/iclock/exchange

HTTP协议版本使用：1.1

Host头域：\${Required}

其他头域：\${Optional}

Factors： 调用加密库返回的设备因子。

服务器正常响应消息：

HTTP/1.1 200 OK

Server: \${XXX}

Set-Cookie: \${XXX}; Path=/; HttpOnly

Content-Type: application/push;charset=UTF-8

Content-Length: \${XXX}

Date: \${XXX}

Factors=\${XXX}

注释：

Factors： 服务器返回的服务器因子。

9 推送配置信息

该功能设备主动推送相关的配置信息，配置信息可以设备指定，亦可服务器指定(详见"初始化信息交互"的"PushOptions"参数)，当配置信息有变化时将主动推送服务器

客户端请求消息

POST /iclock/cdata?SN=\${SerialNumber}&table=options HTTP/1.1

Host: \${ServerIP}:\${ServerPort}

Content-Length: \${XXX}

.....

\${key}=\${Value},\${key}=\${Value},\${key}=\${Value}.....,\${key}=\${Value}

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:

table=options

Host头域: \${Required}

其他头域: \${Optional}

服务器正常响应消息

HTTP/1.1 200 OK

Content-Length: \${XXX}

.....

OK

示例

客户端请求:

POST /iclock/cdata?SN=0316144680030&table=options HTTP/1.1

Host: 58.250.50.81:8011

Content-Length: 26

User-Agent: iClock Proxy/1.09

Connection: close

Accept: */*

FingerFunOn=1,FaceFunOn=1

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

10 上传更新信息

该功能复用[获取命令](#downloadcmd)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},\${Value3},\${Value4},\${Value5},\${Value6},\${Value7},\${Value8},\${Value9},\${Value10}

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
---	--------

4	比对照片功能	（支持比对照片功能，参数BioPhotoFun需要设置为1）
---	--------	--------------------------------

5	可见光人脸模板功能	（支持人脸模板该功能，参数BioDataFun需要设置为1）
---	-----------	--------------------------------

（默认推送前3位，需要设置VisilightFun为1时，推送5位）

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

11 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)

11.1 上传方式

实时上传

间隔上传

定时上传

实时\间隔\定时三种上传方式，若支持实时，则间隔\定时方式不起作用。

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)

11.2 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

11.3 上传考勤照片

[初始化信息交换]服务器下发的配置PushProtVer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}`: 客户端序列号

`size=${XXX}`: 考勤照片原始大小

`${BinaryData}`: 原始图片二进制数据流

考勤照片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
```

11.4 上传操作记录

[[初始化信息交换](#)]服务器下发的配置PushProtVer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}\${Operator}\${HT}\${OpTime}\${HT}\${OpWho}\${HT}\${Value1}\${HT}\${Value2}\${HT}\${Value3}

\${OpType}: 操作代码, 见(附录3)

\${OpWho}、\${Value1}、\${Value2}、\${Value3}: 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
```

11.5 上传用户信息

[[初始化信息交换](#)]服务器下发的配置PushProtVer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=\${XX}\${HT}Grp=\${XXX}\${HT}TZ=\${XXX}\${HT}Verify=\${XXX}\${HT}ViceCard=\${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}: 用户卡号(主卡), 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。

Verify=\${XXX}: 用户验证方式, 不包含该字段、值为空时或者设置为-1(使用组验证方式, 如果没有门禁组, 组验证方式为0), 否则见 (附录7)

ViceCard=\${XXX}: 用户卡号(副卡), 字符串数据, 如卡号为123456789, 则为: ViceCard=123456789
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=0001000000000000

USER

PIN=36235 Name=36235 Pri=0 Passwd= Card=133441 Grp=1 TZ=0001000000000000

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

11.6 上传身份证信息

[初始化信息交换]服务器下发的配置PushProtVer参数大于等于2.3.0版本

客户端请求消息

POST /iclock/cdata?SN=\${SerialNumber}&table=IDCARD&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=IDCARD: \${Required}

Stamp: \${Optional}表示身份证信息到服务器的最新时间戳(不使用)。

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

```
IDCARD${SP}PIN=${XXX}${HT}SNNum=${XXX}${HT>IDNum=${XXX}${HT}DNNum=${XXX}${HT}
Name=${XXX}${HT}Gender=${XXX}${HT}Nation=${XXX}${HT}Birthday=${XXX}${HT}ValidInfo=${X
XX}${HT}Address=${XXX}${HT}AdditionalInfo=${XXX}${HT}Issuer=${XXX}${HT}Photo=${XXX}${H
T}FPTemplate1=${XXX}${HT}FPTemplate2=${XXX}${HT}Reserve=${XXX}${HT}Notice=${XXX}
```

注:

PIN=\${XXX}: 用户工号, 如果用户信息不跟身份证绑定, 那么PIN的值为0。

SNNum=\${XXX}: 身份证物理卡号

IDNum=\${XXX}: 公民身份证号码

DNNum=\${XXX}: 居民身份证证卡序列号 (卡体管理号)

Name=\${XXX}: 身份证姓名, 使用UTF-8编码

Gender=\${XXX}: 性别代码

1,"男"

2,"女"

Nation=\${XXX}: 民族代码

0,"解码错"

1,"汉"

2,"蒙古"

3,"回"

4,"藏"

5,"维吾尔"

6,"苗"

7,"彝"

8,"壮"

9,"布依"

10,"朝鲜"

11,"满"

12,"侗"

13,"瑶"

14,"白"

15,"土家"

16,"哈尼"

- 17,"哈萨克"
- 18,"傣"
- 19,"黎"
- 20,"傈僳"
- 21,"佤"
- 22,"畲"
- 23,"高山"
- 24,"拉祜"
- 25,"水"
- 26,"东乡"
- 27,"纳西"
- 28,"景颇"
- 29,"柯尔克孜"
- 30,"土"
- 31,"达斡尔"
- 32,"仡佬"
- 33,"羌"
- 34,"布朗"
- 35,"撒拉"
- 36,"毛南"
- 37,"仡佬"
- 38,"锡伯"
- 39,"阿昌"
- 40,"普米"
- 41,"塔吉克"
- 42,"怒"
- 43,"乌孜别克"
- 44,"俄罗斯"
- 45,"鄂温克"
- 46,"德昂"
- 47,"保安"
- 48,"裕固"

- 49,"京"
- 50,"塔塔尔"
- 51,"独龙"
- 52,"鄂伦春"
- 53,"赫哲"
- 54,"门巴"
- 55,"珞巴"
- 56,"基诺"
- 57,"编码错"
- 97,"其它"
- 98,"外国血统"

BirthDay=\${XXX}: 出生日期（格式: yyyyMMdd）。

ValidInfo=\${XXX}: 有效期,开始日期和结束日期（格式: yyyyMMddyyyyMMdd）。

Address=\${XXX}: 地址，使用UTF-8编码。

AdditionalInfo=\${XXX}: 机读追加地址，使用UTF-8编码。

Issuer=\${XXX}: 签发机关，使用UTF-8编码。

Photo=\${XXX}: 身份证存储的照片数据，数据是加密的，并转化成base64数据内容进行传输。

FPTemplate1=\${XXX}: 指纹1_指纹特征数据，并转化成base64数据内容进行传输。

FPTemplate2=\${XXX}: 指纹2_指纹特征数据，并转化成base64数据内容进行传输。

Reserve=\${XXX}: 保留字段。

Notice=\${XXX}: 备注信息，使用UTF-8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=IDCARD&Stamp=9999 HTTP/1.1
Host: 58.250.50.81:8011
User-Agent: iClock Proxy/1.09
Connection: close
Accept: */*
Content-Length: 658
```

IDCARD

```
PIN=2  SNNum=xxxxxxxx460088xxxxxx  IDNum=xxxxxx19911218xxxx  DNNum=  Name=张
三  Gender=1  Nation=1  Birthday=19911218  ValidInfo=2017091520270915  Address=xx省
xx市xxx县xxx村xx组xxx          AdditionalInfo=  Issuer=xxx县公安
局          Photo=V0xmAH4AMgAA/4UYUV+sjnpymK1Boqvz3UCBevbbHnYikGyH1XA7Emt2agF0H
FhDc4Bxzeg/jH0Yp8Ngl1861Y812K1AOUIRgy1Z5TEuSG1GV4MwIAB3qY0tKqWNPzyEd8Pn0EhR
sgAAjeWPxiUzLaPU1w  FPTemplate1=QwGIEgELUQAAAAAAAAAAAAAAAAAC9AAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAA4  FPTemplate2=QwGIEgEQUAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
D8  Reserve=  Notice=
```

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

11.7 上传身份证考勤记录

[初始化信息交换]服务器下发的配置PushProtVer参数大于等于2.4.0版本

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}表示上传的数据为身份证考勤记录

Stamp: \${Optional}表示身份证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}, 上传身份证考勤记录, 格式如下

\${Pin}\${HT}\${Time}\${HT}\${Status}\${HT}\${Verify}\${HT}\${Workcode}\${HT}\${Reserved1}\${HT}\${Reserved2}\${HT}\${IDNum}\${HT}\${Type}

IDNum: 身份证号

Type: 记录类型 (0表示考勤, 1表示核验)

Type值为0, 考勤记录的内容为按照考勤协议定义。

Type值为1,

STATUS:0-成功, 1-失败, 2-黑名单

VERIFY :1-人脸, 2-人脸+指纹, 3-指纹+人脸

其他内容按照标准协议定义。

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	210218199105072345	0
1452	2015-07-30 15:16:29	0	1	0	0	0	210218199103062104	0
1452	2015-07-30 15:16:30	0	1	0	0	0	210218199411212642	0
1452	2015-07-30 15:16:31	0	1	0	0	0	210218199207123075	0
1452	2015-07-30 15:16:33	0	1	0	0	0	210218199512012332	0
1452	2015-07-30 15:16:34	0	1	0	0	0	210218199011304365	0
1452	2015-07-30 15:16:35	0	1	0	0	0	210218199806068325	0
8965	2015-07-30 15:16:36	0	1	0	0	0	210218199310094316	0
8965	2015-07-30 15:16:37	0	1	0	0	0	210218199708167443	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

11.8 上传身份证考勤记录照片

[初始化信息交换]服务器下发的配置PushProtVer参数大于等于2.4.0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}表示身份证考勤照片到服务器的最新时间戳(详见“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}, 上传身份证考勤记录照片, 数据格式如下

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

说明如下

PIN=时间-照片类型-工号-身份证号.jpg

照片类型:

- 0: 用户考勤成功照片
- 1: 用户考勤失败照片
- 2: 黑名单照片
- 3: 人证核验成功照片
- 4: 人证核验失败照片

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

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=20160615093758-0-1457-210218199011304365.jpg SN=0316144680030 size=9512 CM
D=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: 4

Connection: close

Pragma: no-cache

Cache-Control: no-store

OK:9

11.9 上传指纹模板

[初始化信息交换]服务器下发的配置PushProtVer参数大于等于2.2.14版本, 支持的指纹算法版本小于等于10.0。

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=SghTUzIxAADS00ECAUHCc7QAAAnSnkBAAAAG/
YUfEsyAIEPHgH6ALFHRQBBAPkP8wBAS2UPEwBTACYPe0tYAHkljACuAHdleQBtAGwEUAB1S2
0DhAB+AK8EXUuPAOoPJABwANVENQDCANsPZQDbSx8PbwDeACwPz0vjAJ8PdwArAPFELAD
5AMwPvQASSvMKMGAwAQkPSUE2DkcXQ0uCQ1B4AJT7GZuC3GyNySrvjoKT7X77SkYkB9L6M
QhMCV5G1PUR+T0FfPiGTqMABQHp+XgBhclzg397xf0iD5CkQAXvErv3q4PQZ940xfmXzBb5bche
r2e7PQkLAXyf8gJ78nP7iwFIQmrXcwKn31LfiwoBIDQBAjrbrAdLOBFwww8ExVYStv7ABQBOE7+JC
Et/GIBs/4SqDwPoJ4yHwMDEBcHDicB/DQCrLkbAwgnFcnwGAHn2g4ilCQCcsNoPCnm4RS75CjJ0r
gwFqWS4HADFDZmwEAWJcRjrABwA1jWTCtMX+whMAw4+JwYnC+//Aw3pCwsK0wRIAXFKJB8PA
sfzCw3WLwaENAz9besHBi4eqBwOPW4PBwsEoygBxKncwCLAwqx4WFIB02WGwl86boCLw4b/Z
MFx3ADXJ4FSi3X/wqt2whjD/wkAc261eMKKigQBG249lwUD0nP9QxEadbZwwYt5wcHCwME6wcM
uBQCeeANI1gCQMwJc/v+DwL90UbQCAIF7cMLNAJo2CMHA+/xTzQCCy2jA/pPABMXTh2ZZXEQE
fnkCn/8OLwWVdSQMB18tDiwsBF8s9RZFcFuvqza3C/3gAwMOMw1lxZ8laxeXM7cFyxMDCpLvAd
CDD/n4HAKMcHsMH/gsAcNreO/z/tfxC/wsAdBvk/bf9/T5KFAAM4KfPxMHfJP/CB8DDi8FzCgBs4p+
EwhHBGwDr4qQEUCa2wsPBwMF4B8DCEML+wAQAEi/wH1wB6/Gma8AFxl2lwIPAg1gYxej46Ut8
n8PBwqrBbgYJEHsO4vk8/cMkBRCIF1NpxhCGe0jBERDTMGxri4/CrGb/FBAaMqiLccDAwMTFAML
Ai8PB/8HAA9WLOgj/DxDPPqtA/8GJtcRCDBDEj55Bi/7JyZMEEdFRQ7T4

FP

PIN=2 FID=1 Size=2120 Valid=1 TMP=T3dTUzIxAAGNDsECAUHCc7QAAAEwNBkAAAA
htlDsjQjAKIPYgDiAHo7NgA1AHAPZgBSNKQPdwBWAOPVjRYABMPsACiADM76wB8ALkO2QB4
NFgPVQCCAP002TSTALoOYABeAEo7TgCdAFMPwQGmNNgNwgCiAH0PnDspAEQPCgFuAEI5
eQCtAEcPlwCpNMONEgGwAPkNvTSyADoP+ABxAEY/KgC1ANUPVQC/NMMP7gC8AAwNHTTAA

M009AAEAEC5JgDCANQO7QDKNN4OzgDOAH4OHjTQANUODgAdAMc6ZgDgAEsPNgDmNMM
OKQDiAKMOEjTIAOUOzgAhAM86QQDmAGUPRQDoNFIPaQDwAAIP6DT3AD8OBgEzALk6uAD3
AMQOCAAHNTQOPwAHAaEPDTUGAUIOtQDMAUY7oQAMAU8PEAAJNbQO8gAZAXgPxDQfAT
sOCgHaAcQ63AApATYOZAAqNTToMmAaVAY0M/TQuAcANYQD1AcY60gA2AT8OaAA/NcELOgA7
ASYPIDQ8AWQLJQCEAfQ7hQBEAd8OKwBCNb8ORABMASoPdDROAeUOnACUAco5Lv0jb0N/v
w8xbVZpOI8G9+JPA0Zbg6sF2hafgw2db/Kq3hanMJLdJ1p/HQLu48rxWblqIj8Lee9U5wASmAL+8DI
EdPjI1MEKQj1H1OD0qPInf2aGYFzANzakIW6/UoNkQruW6gFTYfW/PMF5zDAjd2O3PosZbvLGIVu
gNb5YXuLNMx1+QARCTT6k7VQf7qL1pZi2c0ufv9+NHwGXoLNOR66PcwC7wAWjsUDD0UBfargL
TBFRd1CKH4be/q65T3cYRJ+3fqD7LgB4KAmyEciic2SSdIEZmBvBlfM/iXpQJeAHuJr7rAC04HdXx
wdkBTDH5BgI9MH1HtmT/BYXRkmT/c7TgcsH23gAPAw2o9IWBgqaEzH27xWgGLRFyFNOMc7Q
wfgGD9f4I/i+/dH0pivH+2AaHtkgGpfi9efR7vsJofaXaLA5M5Eu+oAJxBhkGHYN3sbSN2IMCCwHZ7
FUej1fyfXU8mW17ByJhEmaQBKYMGKTKxwN/XAXnzZXgG6IPQ3n/JE4hAmKgZ4bpGpP24wPjR
Ra9Hpz8AEIEwBAungoQU0ZwUAWAwAaAcidMBEwDgKAH0IltEQv4SAHzOEzP0/0NAW8L9xgCf
Px3/CgDDC+L/U8tKDQBnDAk4RvgMw/3AFgCNyRcwH1VYVDyfB8XQChnBVf8SADfU/cbK/v7/O8
H8BcL4y0gDAOQkLTOLBgMb/TwwQQnFuCAZS1X+CwAm4vdKyf/AKwUAX+2AdiEBUS4A/yeD/Vd
yZWADADMztMAeNfc1APwwMDvA+WdZW//AxQTFND9ZZAgApVcwOIXG9AYAelgg/ur/EzRUWR
D+/v31wDEHTcE4GgADm+IV9Pz9wP0wKTPDxnhoEAAHaOL9O/vK/v7+/jQHxAVuZ5X9wgUAsaw0
TCYBCHLkR/4FISx1Oh8AAX7wBkcozy8wRTbCj/7CQBgCARCAVsHBABY0X40EAOuAg3YANEW
DaZ3CCMVYjwP6RjYEAD5KXI4wAUSPVIYfXUqJePxPHQAEkRL++GP8+/4j/v+O/8fK/cFW/ykOxd+
QDv5Xc8DCwJULBmufSTj//cDqBQZ/nINBBwBQZEz4y/8yBADDpvHA+CQBI6tDUIE+wccHwfsUAJy
rhVVdyv3E///9/wX7+2MCAQevRsHDAQqbQf9SBgB4dUZD9AsAv7U3QQXAXUQEALq2QGjdAB+Jx
/7///7+0vz4HcH+/8H9/wX++TcBCL9AwQXF7cZ0/sLBCAD9AkD79Xb/IAACygUoxsvB/f/9+/04/fjK/v3
AwP//Bf3E9MH9/v3+HcUP0PfD/MD+//86/fvP/SErS//AOz74MAAS3EI9BMUI510xBgD/40AH/Hk/AWf
mUP5VhCIANN3nQMDBV8AAOt5j/0AKAIA0UEV0/P0GAOv6g8RAMRE9CmLAI8EQtdhCcwKQuAz
yQPvJ+/8FEJwPlf5wMxHWEzf9wgT/wjcQFRhQwQbV0xxyY/0DEMMihcEFJNwvNMEDEBY5RvUE
EJVBVvmTBRY5Tf1RChBCpwDG9MHAwMBKAA==

FP

PIN=3 FID=0 Size=1592 Valid=1 TMP=TetTUzlxAAAEqKsECAUHCc7QAAAcqWkBAAAAh
FUooagrAKAPQADpAGKnzgAuAKkPtQA1qlsPnQBGAO0PTqhKAGIPlgCrADanNwByAEIPngB6qD
QOfgCLAGENJKiNAD0PiABTAKimdACXAB0OVgCmqKwOigCnAFgOtqi0AKsPiABYAJimyQC3AKs
PjQDEqKwPngDBAFIPrjDAKIPaQAPABGnrwDQAiwPAQDQqJsP6wDUAF4PXKjgAJ8PwgAhAIO
m2QDrAlwPTgDrqJIPQwD8AFYPdqj/AJQP9gDBAYmnugAGAXwPrwANqZMP6QAJAUyOWagTA
YsP+AD2AWqmVAA0AX8PUQA8qYAP6ABDAbIPKdPn+18ZTgTm3GePXwtHgjd7mih7jAplJYui+i6
h1IRzlwfyRpLTDpojnW/ODhLjqxleOdv2Afg5iMsIJIU6C2uTeQgwABW4PHhncX2DhRNxqCgS0PMp
AGTwKaDYAkEWqQB88vGuWAlde0oIj5kGuRof7PcJDhDuZKcUGLX4xXs/aVLRqOyV9v326BRpu
qzzQhBmApMXuSc/DZL1wezA/p1XgAbS8Jr0nANWq9IICgnK9TsCQaxMC9kDZQxXAJelz/Xm8gv4
DABRoH8C8fiq+fcX2VDY+5L5df0D79dlz/XX7LfVzVOHr94H6QrL9xbvuTys+2sgRAHHpSVRDQCpB

CtrlmTEpgGQBxw4/55qYK0BcQgP/zHBAIWgEScTAF0Jxvz7V/5HwP5rZ5QMBDYIHv5KwGKRBAQ
SCydZBQDDzifE7wUAtQ0nYNMAQLgBwP3+QV6hUI30CgAIJ+D8O/3EVf4wFwAqKiL9+Vf/MThtZc
AEwY+gAWwsj5LDQAYEISxndMINAG0tl/1j/8L/XQzFbDYhwJ9pwnsFxdI2j1MJAHQzHoNU+64BRk
prwsS2DwTmS/38//wwwMPGxgwAnEorwv/YGjAwYQOAEKZ8ckksLBwsDAzwCxxSj/wVjCcdQA
U8VWxMPCwclHw/tqwMDC/UsMxTlqfif9/MD9/qoMBPRuT8LDxMBYwWiuAZdwMFLAwwBI3UHDw
8DDB8Uzcu6RwslGADmzQ8UgCQBleCfDiWEfqA6Fyf///T++VX+/f7/wP4F/8RowItvBgBaQDTGIAw
Ae4ipwjvlzW2hfhUAFo8DQfpX//39/S7/7nwlqByPQ8LCiAV+xKABI489jMK2BAQdkDSDFQCIV6v7IM
fHw8TCwgb/xtTAhQUAdZvnw/poCgAVoD2SuWQDqLe5IHSFCsURv5jCacHB/sPGAKZrFsAGAEzE
4cF+rwFsyRPAhjsGBKbLKcLAIAnFa8q2wv/Cw1MGxcjSv8HCRg0AWyEXftZHSQQA3O3KWQ2ojv
ADaEf/xgBDVRLABRAnAMxHArh3AAI+wATVvwKoQAUQKQgTAPzGrhFuCQZpwMgQ/7qHdsDCQ
/4FwRW4LhYJwf/COsJUaMNDQBAQOuYJdFfAZmb+YAjV/C7YM2QJEqMtk/77Vf//UgUQWfH9UKc
R9TdwLcCeUm2vEZA7fVJdyRDg7nvA/2jC/gY/Drg7XAPAbcl7/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

11.10 上传面部模板

[初始化信息交换]服务器下发的配置PushProtVer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=AAAAAAAAAAAAAAAAAAAAAFpLRmIYATFLF
LTtoAUQBQ1Mg+fgXuia23BDrNtwSfgJ8g74H3YHmXlkFpgetB5eH5yXuBvMLoa6wSx9HNgK7RP80
v1i+LLY8nCn7PXmD7w15Bp8N1wm/A78PowejZx9jJyWnBZ88K5wVfDDcNTjifGlvox9iD8sf1g37B7
0Fk4WRI5RrKq8uD2MngRexMxk5cbDiH+c3xj+CV8Zf1idaDfWbkB8Rnwt/AV8Du0SvAddBywHMQ9
MVysfFkNENfZD9FJ9jrnGeBD5Kcwp7CVySfJzOE+wZxjWfVY6fgreXHBd6B4ov4BxBX+GulZ4paz
BTINiG8kf4h/DHxGaFxYe+yh+O1slCDsPcweuB/SHnA+UnqwG3AvnA88DZg/vhmaaV4dsWzwerBn
1jLcN8wu/ErITiR+YHVsc1wy2wdaC5uEmxKbwZ+AGeB5fFt6WLVa8kq/gvqqv8LvwsBAACAgIHAQ
ABAAEGAUxyAQkclP9SAAohGhchAQAAEAQAAQYFASBPAGYGB1YKAQEMBGAACg8HFQ0DDA
oEA8CCBQDAxtLDwM40CUFEBNVSQYDDRNJJg8CAAcJFBMMAQQiYA8MAwhZHAceehMCAC
YEAawACQsJBQAFBAYxAaknpwQGJ6EiBAUPIxwFBwQPOgQCBAARGz8WCAIAGUQWBggLhy
MIBJQIBAliAgUEAgMPAwUACQAFUAABE/8EBwkQEgABCRECBwQFFEYNCAcCFiJ3YwUACTpK
LgwBDn8yDApjFAIBEWAAAwACBgEAAAECAUMBAAwBCAAACAQIBAAEABAMBAQUEAQMCBR0
x/z4CAhUhSk8GAAcIDhUBNgSLDQMBAQAAAAYDAAAAAAEEAQAHBwEGQWEIAAJe/zQBAAYY
/30GAAABAAICAgAABAMAAAEFFAgAAAOEAAEUAgABAQEABwEECQMWGAIBDSIHASH/HQQI
CIQeBgYMCgsIAwEBDSB0IBEJAjLbQgkIB245BQYKCAAEAAEDAwEDDgCkAQIBAhwABB+4DAQ
OoiAFCAGXFwMBByNDCgYBABUQGSYDAwItYhMIATG9KQwLV0oCBBkEAQgBDggHAQADBwJ
DAS6AIGFxYXBgMiRxQNCgMXvCMOBQIXGrxABgkGJQ4YBQEQHxUJBuGNCgggAwIBAgMK
AAAAAgMCJwADAZUAAAEAAwAAAB4BAAEAAAAAAAAA1j/JgQGAXf/fAMBAGkLAAEEAgIAAAA
EAAEAawEAAAABAAQAAAAAAAAABAAAAARgQAQABAjP/KQMAAAIDAQEBBAIEBAoCAAoUAg
MBRAYCACoBAQAAAAAsDAgAAAgEJAAMDHgABEiwNAQQnRA8DAwc5IDICAQAIETAbAAEGM0
0eAQApYCIKA/81CwU5DAIGAAEEBQQABQACKAAGG2MEAxkpFQMBCTkKAAUCFSYJAQABD

ByEGwIBajUriQUDO/8eCwSMRAQACAQABQEABgAAAAMAAA8AAAMuAAAAAwgCBQMFCQcC
BBNECgMAAAQvZhABAAhZ/DIBACL/XQgDJiEHAgwFAQEAAAcDAAAAAQEDAAACAAAAAwEAA
gEFCQEBBAEHIAIAAQELFRQEAWICS/8rBQEDRVQUBAYODwQAAAgAAAAANAAAAAAA

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

11.11 上传指静脉模板

[[初始化信息交换](#)]服务器下发的配置PushProtVer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

FVEIN\${SP}Pin=\${XXX}\${HT}FID=\${XXX}\${HT}Index=\${XXX}\${HT}Size=\${XXX}\${HT}Valid=\${XXX}\${HT}Tmp=\${XXX}

注:

Pin=\${XXX}: 用户工号

FID=\${XXX}: 手指编号, (0~9)

Index=\${XXX}: 一个手指有多个指静脉模板, Index是某手指对应指静脉模板的编号(0~2)。

Size=\${XXX}: 指静脉模版二进制数据经过base64编码之后的长度

Valid=\${XXX}: 指静脉模版有效性标示, 值意义如下:

值	描述
0	无效模版
1	正常模版

Tmp=\${XXX}: 传输指静脉模版时, 需要对原始二进制指静脉模版进行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: 1698

FVEIN

Pin=306 FID=2 Index=0 Size=1648 Valid=1 Tmp=AAAAAAAAAAAAAAAAAAAAAFpLRml
YATFLFLT0AUQBQ1Mg+fgXuia23BDrNtwSfgJ8g74H3YHmXlkFpgetB5eH5yXuBvMLoa6wSx9HNg
K7RP80v1i+LLY8nCn7PXmD7w15Bp8N1wm/A78PoweJzX9jJyWnBZ88K5wVfDDcNTjifGIvox9iD8sf
1g37B70Fk4WRI5RrKq8uD2MngRexMxk5cbDiH+c3xj+CV8Zf1idaDfWbkB8Rnwt/AV8Du0SvAddBy
wHMQ9MVysfFkNENfZD9FJ9jrnGeBD5Kcwp7CVySfJzOE+wZxjWfVY6fgreXHBd6B4ov4BxBX+G
ulZ4pazBTINiG8kf4h/DHxGaFxYe+yh+O1slCDsPcweuB/SHnA+UnqwG3AvnA88DZg/vhmaaV4ds
WzwerBn1jLcN8wu/ErITiR+YHVsc1wy2wdaC5uEmxKbwZ+AGeB5fFt6WLva8kq/gvqqv8LvwsBAA
CAgIHAQABAAEGAUxyAQkclP9SAAohGhchAQAAEAEEAAQYFASBPAGYGB1YKAQEMBgAACg8
HFQ0DDA0eAg8CCBQDAxtLDwM40CUFEBNVSQYDDRNNJg8CAAcJFBMMAQQiYA8MAwhZHA
EehMCACYEAAwACQsJBQAFBAYxAAknpwQGJ6EiBAUPIxwFBwQPOgQCBAARGz8WCAIAGUQ
WBggLhyMIBJQIBAlAgUEAgMPAwUACQAFUAABE/8EBwkQEGABCRCBwQFFEYNCAcCFiJ3Y
wUACTpKLgwBDn8yDApjFAIBEWAAwACBgEAAAECAUMBawBCAAACAQIBAAEABAMBAQUE
AQMCBR0x/z4CAhUhSk8GAAClDhUBNgSLDQMBAQAAAAYDAAAAAEEAQAHBwEGQWEIAAJe
/zQBAAYY/30GAAABAAICAgAABAMAAAEFFAgAAAOEAAEUAgABAQEABwEECQMWGAIBDSIH
ASH/HQQIClQeBgYMCgsIAwEBDSB0IBEJAjLbQgkIB245BQYKCAAEAAEDAwEDDgcKAQIBahwA
BB+4DAQOOiAFCAgXFWMBByNDCgYBABUQGSYDAWltYhMIATG9KQwLV0oCBBkEAQgBDggH
AQADBwJDAXAS6AIGFxYXBgMiRxQNCgMXvCMOBQIXGrxABgkGJQ4YBQEQHxUJBUgNCgggA
wIBAgMKAaaaaAgMCJwADAZUAAAEAAwAAAB4BAAEAAAAAAAAAAAA1j/JgQGAXf/fAMBAgkLAAE
EAgIAAAAEAAEAwEAAAABAAQAAAAAAAAABAAAAARgQAQABajP/KQMAAAIDAQEBAIEBAO
CAAoUAgMBRAYCACoBAQAAAAASDAgAAAgEJAAMDHgABEiwNAQQnRA8DAwc5IDICAQAIETAb
AAEGM00eAQApYCIKA/81CwU5DAIGAAEEBQQABQACKAAGG2MEAXkpFQMBCTkKAAUCFSY
JAQABDBYEGwIBajUrIQUDO/8eCwSMRAQACAQABQEABgAAAAAMAAA8AAAMuAAAAAwgCBQ
MFCQcCBBNECgMAAAQvZhABAAhZ/DIBACL/XQgDJiEHAgwFAQEAAAcDAAAAAQEDAAACAA

AAAwEAAgEFCQEBBAEHIAIAAQELFRQEAWICS/8rBQEDRVQUBAYODwQAAAgAAAANAAAAA
AAA

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

11.12 上传一体化模板

[[初始化信息交换](#)]服务器下发的配置PushProtVer参数大于等于2.2.14版本，后续新增的生物识别模板将统一格式上传下载，使用数据中的Type来区分是何种生物识别模板，使用一体化格式的有：手掌模板等。

客户端请求消息

POST /iclock/cdata?SN=\${SerialNumber}&table=BIODATA&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=BIODATA: \${Required}

Stamp: \${Optional}表示一体化模版到服务器的最新时间戳(不使用)

Host头域: \${Required}

Content-Length头域: \${Required}

其他头域: \${Optional}

请求实体: \${DataRecord}, 一体化模版数据, 数据格式如下

BIODATA\${SP}Pin=\${XXX}\${HT}No=\${XXX}\${HT}Index=\${XXX}\${HT}Valid=\${XXX}\${HT}Dure
ss=\${XXX}\${HT}Type=\${XXX}\${HT}MajorVer=\${XXX}\${HT}MinorVer=\${XXX}\${HT}Format=\${XXX}
\${HT}Tmp=\${XXX}

注:

Pin=\${XXX}: 人员工号

No=\${XXX}: 生物具体个体编号, 默认值为0

【指纹】编号是: 0-9, 对应的手指是: 左手: 小拇指/无名指/中指/食指/拇指, 右手: 拇指/食
指/中指/无名指/小拇指;

【指静脉】 和指纹相同

【面部】 都为0

【虹膜】 0为左眼 1为右眼

【手掌】 0为左手 1为右手

Index=\${XXX}: 生物具体个体模板编号, 如1个手指存储多枚模板。从0开始计算

Valid=\${XXX}: 是否有效标示, 0: 无效, 1: 有效, 默认为1

Duress=\${XXX}: 是否胁迫标示, 0: 非胁迫, 1: 胁迫, 默认为0

Type=\${XXX}: 生物识别类型

值	意义
0	通用的
1	指纹
2	面部
3	声纹
4	虹膜
5	视网膜
6	掌纹
7	指静脉
8	手掌
9	可见光面部

MajorVer=\${XXX}: 主版本号, 如: 指纹算法版本10.3, 主版本是10, 副版本是3

【指纹】 9.0、10.3及12.0

【指静脉】 3.0

【面部】 5.0、7.0和8.0

【手掌】 1.0

MinorVer=\${XXX}: 副版本号, 如: 指纹算法版本10.3, 主版本是10, 副版本是3

【指纹】 9.0、10.3及12.0

【指静脉】 3.0

【面部】 5.0、7.0和8.0

【手掌】 1.0

Format=\${XXX}: 模板格式, 如指纹有ZK\ISO\ANSI等格式

【指纹】

值 格式

0 ZK

1 ISO

2 ANSI

【指静脉】

值 格式

0 ZK

【面部】

值 格式

0 ZK

【手掌】

值 格式

0 ZK

Tmp=\${XXX}: 模板数据, 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=BIODATA&Stamp=9999 HTTP/1.1

Host: 58.250.50.81:8011

User-Agent: iClock Proxy/1.09

Connection: close

Accept: */*

Content-Length: 1736

BIODATA

Pin=306 No=0 Index=2 Valid=1 Duress=0 Type=8 MajorVer=1 MinorVer=0 Format=0 Tmp=AAAAAAAAAAAAAAAAAAAAAFpLRmIYATFLFLTToAUQBQ1Mg+fgXuia23BDrNtwSfgJ8g74H3YHmXlkFpgetB5eH5yXuBvMLoa6wSx9HNgK7RP80v1i+LLY8nCn7PXmD7w15Bp8N1wm/A78PowejZx9jJyWnBZ88K5wVfDDcNTjifGlvox9iD8sf1g37B70Fk4WRI5RrKq8uD2MngRexMxk5cbDiH+c3xj+CV8Zf1idaDfWbkB8Rnwt/AV8Du0SvAddBywHMQ9MVysfFkNENfZD9FJ9jrmGeBD5Kcwp7CVySfJzOE+wZxjWfVY6fgreXHBd6B4ov4BXXB+GulZ4pazBTINiG8kf4h/DHxGaFxYe+yh+O1slCDsPcweuB/SHnA+UnqwG3AvnA88DZg/vhmaaV4dsWzwerBn1jLcN8wu/ErlTiR+YHVsc1wy2wdaC5uEmxKbwZ+AGeB5fFt6WLVa8kq/gvqqv8LvwsBAACAgIHAQABAAEGAUxyAQkcIP9SAAohGhchAQAAEAEAAQYFASBPAGYGB1YKAQEMBgAACg8HFQ0DDAoEAg8CCBQDAxtLDwM40CUFEBNVSQYDDRNNJg8CAACJFBMMAQQiYA8MAwhZHAcEehMCACYEAAwACQsJBQAFBAYxAAknpwQGJ6EiBAUPIxwFBwQPOgQCBAARGz8WCAIAGUQWBggLhyMIBJQIBAliAgUEAgMPAwUACQAFUAABE/8EBwkQEgABCRECBwQFFEYNCAcCFiJ3YwUACTpKLgwBDn8yDApjFAIBEwAAAACBgEAAAECAUMBAwBCAAACAQIBAAEABAMBAQUEAQMCBR0x/z4CAhUhSk8GAAcIdhUBNgsLDQMBAQAAAAYDAAAAAAEEAQAHBwEGQWEIAAJe/zQBAAYY/30GAAABAAICAgAABAMAAAEFFAgAAoEAAEUAgABAQEABwEECQMWGAIBDSIHASH/HQQICiQeBgYMCgsIAwEBDSB0IBEJAjLbQgklB245BQYKCAEAAEDAwEDDgckAQIBAhwABB+4DAQOoiAFCAGXFWMBByNDCgYBABUQGSY

DAwItYhMIATG9KQwLV0oCBBkEAQgBDggHAQADBwJDAS6AIGFxYXBgMiRxQNCgMXvCMO
BQIXGrxABgkGJQ4YBQEQHxUJBUGNCgggAwIBAgMKAAAAAgMCJwADAzUAAAEAAwAAAB4B
AAEAAAAAAAAAAAA1j/JgQGAXf/fAMBAGkLAAEEAgIAAAAEAAEAwEAAAABAAQAAAAAAAAABAA
AAARgQAQABAJp/KQMAAAIDAQEBBAIEBAoCAAoUAgMBRAYCACoBAQAAAAAsDAgAAAgEJAA
MDHgABEiwNAQQnRA8DAwc5IDICAQAIETAbAAEGM00eAQApYCIKA/81CwU5DAIGAAEEBQQA
BQACKAAGG2MEAxkpFQMBCTkKAAUCFSYJAQABDBYEGwIBAJUrIQUDO/8eCwSMRAQACAQA
BQEABgAAAAMAAA8AAAMuAAAAAwgCBQMFCQcCBBNECgMAAAQvZhABAAhZ/DIBACL/XQg
DJiEHAgwFAQEAAAcDAAAAAQEDAAACAAAAAwEAAgEFCQEBAEHIAIAAQELFRQEAWICS/8r
BQEDRVQUBAYODwQAAAQAAAAANAAAAAAAAA

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

11.13 上传用户照片

[[初始化信息交换](#)]服务器下发的配置PushProtVer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

11.14 上传数据包

[[初始化信息交换](#)]服务器下发的配置PushProtVer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&ContentType=\${Value} HTTP/1.1

Host: \${ServerIP}:\${ServerPort}

Content-Length: \${XXX}

.....

\${DataPack}

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}

ContentType: 实体数据格式，目前支持如下

\${Value} 意义

tgz 数据包压缩格式为tgz

Host头域: \${Required}

Content-Length头域: \${Required}

其他头域: \${Optional}

请求实体: \${DataPack}，数据包的数据格式参见其他数据类型的格式。

多条记录之间使用\${LF}连接，然后进行打包，比如：

将如下数据进行打包作为实体数据发送即可

USER

PIN=1 Name= Pri=0 Passwd=0 Card=89776433 Grp=1 TZ=0001000100000000 Ver
ify=-1 ViceCard=123456789

FP

PIN=1 FID=1 Size=1336 Valid=1 TMP=SqFTUzIxAAAD4uUECAUHCc7QAAAb42kBAAAAG
w8hXulvAPwP0ADwAl7tjwBKAH4PIgBU4vcPfABhAMcPKOJiAOsPPACmAGbtZABIAHEPdgBq4hE
PzwB0AFUPiuJ/AAwPOABAAOvtTACYAF0PkQCi4uwPmgCiANgP1+KzACgPdWAMaFLtiADJADIP
lgDP4kwPtADTAPQPdeLbADsPZQAHAELt6QD0AC4PQAD64rsP0wAFafgPyeliAb8PvgDhAUHssQ
AxAUgOXwBH48EONwBFAYkOaeJLAa0OVgCOASrsYgBOAZcOCwvAbJqCS383/64TXp0icG8Mp
YY8kwHt+JJZh674UljJ54v3nXwJc6d9rpwlcpIOgoAaBAx2g4Ava7MH4wylEA+fuY9ehY/9S5HAd2M
ROJPzBtVBExK278PNfvjk7RabcRgxIU8Q0gOw+ilXWgz3/UX02wMzCP8WzOyyHygKKQnm9N/wu
hGqcuv/doGbeohqKBOSgJ79+X8tmo+EPleh/d8FGuuciyalpXtigDz2PyQ+tzuh+RpSZ9LsYJdVfSfc1
wx1c3JPtXyRTgPCNgECMx1NwwBp6nx1wAMAmckGw+sBhg6AdIsECQOCK/dAVMARxd8vccHBh
cLAcATCwrgHAFkwd8SXCwODMf1FQMBW1ADd1JL/w2fAeEbAUOoBikeAwJacCQNwSgAv/j0FxFY
pOn3kKAIBdAPjA/RxtFADyXZ4HwMAdwX/AhH/CQwcDz1/pwf3CJ84Ag90C/8E2O8HNAGKBdPbA
QQsARWQDHTTAwP9EBcU6ZYnDwMIEAGGscYf2AfF7I//BQllpn1v/CwCNfMb9RSJBwAwAhX1Fk
MljKXYNAIyCzDVXHUrbEQA8hSI4LSMuwP3BQhLFUZYl/v79/f/B7i9WlwkASZhiwU58BOJMnFzAh
MHVAFd/5ST+wP/+O8E1ogcAlqCXngEMA3ygFjY1/8AFwAviUaVed8FcYgCfRB3//8A4/4r/wh8HAN
W0JEwFGAPzw9dU//7AO8H+oz//PsD+/8oAVSrd/v36I/06/8Mdwf8HAE/NloNp6AFzzVOdwKPACeK

NzS0pc8A6BwO20UzBcf8DxX/X1v8HALXWLT0+w+gBdN5Aa3Q6xArie943VMB7zQBmCEJpcRgQ
MMLDPaL//v3+/P84/0Yc/EcGENQI/0DA9RE5G70+NTvA/xz9/f3///06wfwfwd/gQQyic9oAMTWCIa/wM
QwO06/eERPdc9/xfVaUZSwf4+wP3AOPv1GzRzwWwaEKNLpNPBPv7A/P4/+v0f/MD+wMDAOv/9
5hFRVwmHA9XPW5/BAAUAwe/xwA==

USER

PIN=2 Name= Pri=0 Passwd=0 Card=89776433 Grp=1 TZ=0001000100000000 Ver
ify=-1 ViceCard=123456789

USER

PIN=3 Name= Pri=0 Passwd=0 Card=89776434 Grp=1 TZ=0001000100000000 Ver
ify=-1 ViceCard=223456789

USER

PIN=4 Name= Pri=0 Passwd=0 Card=89776435 Grp=1 TZ=0001000100000000 Ver
ify=-1 ViceCard=323456789

USER

PIN=5 Name= Pri=0 Passwd=0 Card=89776436 Grp=1 TZ=0001000100000000 Ver
ify=-1 ViceCard=423456789

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。

11.15 上传比对照片

[初始化信息交换]服务器下发的配置PushProtVer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}表示比对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}, 比对照片数据, 数据格式如下

BIOPHOTO\${SP}PIN=\${XXX}\${HT}FileName=\${XXX}\${HT}Type=\${XXX}\${HT}Size=\${XXX}\${HT}Content=\${XXX}

注:

FileName=\${XXX}: 生物特征图片的文件名, 目前只支持jpg格式。

Type=\${XXX}: 生物识别类型。

值 意义

0 通用的

1 指纹

2 面部

3 声纹

4 虹膜

5 视网膜

- 6 掌纹
- 7 指静脉
- 8 手掌
- 9 可见光人脸

Size=\${XXX}: 生物特征图片base64 编码之后的长度

Content=\${XXX}: 传输时，需要对原始二进制生物特征图片进行base64 编码。

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=OPERLOG&Stamp=9999 HTTP/1.1

Host: 58.250.50.81:8011

User-Agent: iClock Proxy/1.09

Connection: close

Accept: */*

Content-Length: 1684

BIOPHOTO PIN=123 FileName=123.jpg Type=2 Size=95040 Content=AAAA.....

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

11.16 上传异常日志

[初始化信息交换]服务器下发的配置PushProtVer参数大于等于2.4.1版本

客户端请求消息

POST /iclock/cdata?SN=\${SerialNumber}&table=ERRORLOG&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=ERRORLOG: \${Required}

Stamp: \${Optional}表示异常日志到服务器的最新时间戳(详细见“初始化信息交互”的“ERRORLOGStamp”参数)

Host头域: \${Required}

Content-Length头域: \${Required}

其他头域: \${Optional}

请求实体: \${DataRecord}, 异常日志数据, 数据格式如下:

ERRORLOG

ErrCode=\${XXX}\$(HT)ErrMsg=\${XXX}\$(HT)DataOrigin=\${XXX}\$(HT)CmdId=\${XXX}\$(HT)Additional=\${XXX}

注：

ErrCode=\${XXX}：错误编码。编码说明见（[附录9](#)）

ErrMsg=\${XXX}：错误消息。

DataOrigin=\${XXX}：数据源，dev表示设备源数据，cmd表示软件下发数据

CmdId=\${XXX}：软件下发命令号。

Additional=\${XXX}：附加信息(base64数据)，原生数据格式为json。

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=ERRORLOG&Stamp=9999 HTTP/1.1

Host: 58.250.50.81:8011

User-Agent: iClock Proxy/1.09

Connection: close

Accept: */*

Content-Length: 71

ERRORLOG ErrCode=D01E0001 ErrMsg= DataOrigin=cmd CmdId=123 Additional=

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

12 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。

12.1 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

12.2 UPDATA 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

12.2.1 用户信息

命令格式如下

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

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

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XXXXXXXXXXXXXXXXX，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。

Verify=\${XXX}：用户验证方式，不包含该字段、值为空时或者设置为-1(使用组验证方式，如果没有门禁组，组验证方式为0)，否则见（附录7）

ViceCard=\${XXX}：用户卡号(副卡)，字符串数据，如卡号为123456789，则为：ViceCard=123456789
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

12.2.2 身份证信息

命令格式如下

C:\${CmdID}:DATA\$(SP)UPDATE\$(SP)IDCARD\$(SP)PIN=\${XXX}\${HT}SNNum=\${XXX}\${HT>IDNum=\${XXX}\${HT}DNNum=\${XXX}\${HT}Name=\${XXX}\${HT}Gender=\${XXX}\${HT}Nation=\${XXX}\${HT}Birthday=\${XXX}\${HT}ValidInfo=\${XXX}\${HT}Address=\${XXX}\${HT}AdditionalInfo=\${XXX}\${HT}Issuer=\${XXX}\${HT}Photo=\${XXX}\${HT}FPTemplate1=\${XXX}\${HT}FPTemplate2=\${XXX}\${HT}Reserve=\${XXX}\${HT}Notice=\${XXX}

说明：

PIN=\${XXX}：用户工号，如果用户信息不跟身份证绑定，那么PIN的值为0。

SNNum=\${XXX}：身份证物理卡号

IDNum=\${XXX}：公民身份证号码

DNNum=\${XXX}：居民身份证证卡序列号（卡体管理号）

Name=\${XXX}：身份证姓名，使用UTF-8编码

Gender=\${XXX}: 性别代码

1,"男"

2,"女"

Nation=\${XXX}: 民族代码

0,"解码错"

1,"汉"

2,"蒙古"

3,"回"

4,"藏"

5,"维吾尔"

6,"苗"

7,"彝"

8,"壮"

9,"布依"

10,"朝鲜"

11,"满"

12,"侗"

13,"瑶"

14,"白"

15,"土家"

16,"哈尼"

17,"哈萨克"

18,"傣"

19,"黎"

20,"傈僳"

21,"佤"

22,"畲"

23,"高山"

24,"拉祜"

25,"水"

26,"东乡"

27,"纳西"

28,"景颇"
29,"柯尔克孜"
30,"土"
31,"达斡尔"
32,"佤佬"
33,"羌"
34,"布朗"
35,"撒拉"
36,"毛南"
37,"仡佬"
38,"锡伯"
39,"阿昌"
40,"普米"
41,"塔吉克"
42,"怒"
43,"乌孜别克"
44,"俄罗斯"
45,"鄂温克"
46,"德昂"
47,"保安"
48,"裕固"
49,"京"
50,"塔塔尔"
51,"独龙"
52,"鄂伦春"
53,"赫哲"
54,"门巴"
55,"珞巴"
56,"基诺"
57,"编码错"
97,"其它"
98,"外国血统"

BirthDay=\${XXX}: 出生日期 (格式: yyyyMMdd)

ValidInfo=\${XXX}: 有效期,开始日期和结束日期 (格式: yyyyMMddyyyyMMdd)

Address=\${XXX}: 地址, 使用UTF-8编码

AdditionalInfo=\${XXX}: 机读追加地址, 使用UTF-8编码

Issuer=\${XXX}: 签发机关, 使用UTF-8编码

Photo=\${XXX}: 身份证存储的照片数据, 数据是加密的, 并转化成base64数据内容进行传输。

FPTemplate1=\${XXX}: 指纹1_指纹特征数据, 并转化成base64数据内容进行传输。

FPTemplate2=\${XXX}: 指纹2_指纹特征数据, 并转化成base64数据内容进行传输。

Reserve=\${XXX}: 保留字段

Notice=\${XXX}: 备注信息, 使用UTF-8编码

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

12.2.3 指纹模板

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。

注意: 该命令支持的指纹算法版本小于等于10.0。

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

12.2.4 面部模板

命令格式如下

C:\${CmdID}:DATA\${SP}UPDATE\${SP}FACE\${SP}PIN=\${XXX}\${HT}FID=\${XXX}\${HT}Valid=\${XX
X}\${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

12.2.5 指静脉模板

命令格式如下

C:\${CmdID}:DATA\${SP}UPDATE\${SP}FVEIN\${SP}Pin=\${XXX}\${HT}FID=\${XXX}\${HT}Index=\${XX
X}\${HT}Valid=\${XXX}\${HT}Size=\${XXX}\${HT}Tmp=\${XXX}

说明:

Pin=\${XXX}: 用户工号

FID=\${XXX}: 手指编号, (0~9)

Index=\${XXX}: 一个手指有多个指静脉模板, Index是某手指对应指静脉模板的编号(0~2)。

Size=\${XXX}: 指静脉模版二进制数据经过base64编码之后的长度

Valid=\${XXX}: 指静脉模版有效性标示, 值意义如下:

值	描述
0	无效模版

1 正常模版

Tmp=\${XXX}: 传输指静脉模版时, 需要对原始二进制指静脉模版进行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

12.2.6 一体化模板

后续新增的生物识别模板将统一格式上传下载, 使用数据中的Type来区分是何种生物识别模板, 使用一体化格式的有: 手掌模板等

命令格式如下

C:\${CmdID}:DATA\${SP}UPDATE\${SP}BIODATA\${SP}Pin=\${XXX}\${HT}No=\${XXX}\${HT}Index=\${XXX}\${HT}Valid=\${XXX}\${HT}Duress=\${XXX}\${HT}Type=\${XXX}\${HT}MajorVer=\${XXX}\${HT}MinorVer=\${XXX}\${HT}Format=\${XXX}\${HT}Tmp=\${XXX}

说明:

各字段说明见上传一体化模板功能

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

12.2.7 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

12.2.8 比对照片

命令格式如下

C:\${CmdID}:DATA\${SP}UPDATE\${SP}BIOPHOTO\${SP}PIN=\${XXX}\${HT}Type=\${XXX}\${HT}Size=\${XXX}\${HT}Content=\${XXX}\${HT}Format=\${XXX}\${HT}Url=\${XXX}

说明:

PIN=\${XXX}: 用户工号

Type=\${XXX}: 生物特征类型:

值 意义

0 通用的

1 指纹

2 面部（近红外）

3 声纹

4 虹膜

5 视网膜

6 掌纹

7 指静脉

8 手掌

9 可见光人脸

Size=\${XXX}: 生物特征图片二进制数据经过base64 编码之后的长度

Content=\${XXX}: 传输生物特征图片时, 需要对原始二进制生物特征图片进行base64 编码。

Url=\${XXX}: 服务器文件存放地址, 目前只支持 jpg 格式。

Format=\${XXX}: 下发方式, 0: base64方式, 1: url方式。

多条记录之间使用\${LF}连接。

注:

Url是相对路径的场合, 直接发相对路径即可。

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

12.2.9 短消息

命令格式如下

C:\${CmdID}:DATA\${SP}UPDATE\${SP}SMS\${SP}MSG=\${XXX}\${HT}TAG=\${XXX}\${HT}UID=\${XX

X){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

12.2.10 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

12.2.11 宣传照片

命令格式如下

C:\${CmdID}:DATA\${SP}UPDATE\${SP}ADPIC\${SP}Index=\${XXX}{HT}Size=\${XXX}{HT}Extension=\${XXX}{HT}Content=\${XXX}

说明：

Index=\${XXX}： 图片索引

Size=\${XXX}： 图片大小

Extension=\${XXX}： 图片扩展名

Content=\${XXX}： 图片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

12.2.12 工作代码

命令格式如下

C:\${CmdID}:DATA\${SP}UPDATE\${SP}WORKCODE\${SP}PIN=\${XXX}\${HT}CODE=\${XXX}\${HT}NAME=\${XXX}

说明：

PIN=\${XXX}： 工作代码索引。

CODE=\${XXX}： 工作代码编码。

NAME=\${XXX}： 工作代码名称。

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

12.2.13 快捷键

命令格式如下

C:\${CmdID}:DATA\${SP}UPDATE\${SP}ShortcutKey\${SP}KeyID=\${XXX}\${HT}KeyFun=\${XXX}\${HT}StatusCode==\${XXX}\${HT}ShowName=\${XXX}\${HT}AutoState=\${XXX}\${HT}AutoTime=\${XXX}\${HT}Sun=\${XXX}\${HT}Mon=\${XXX}\${HT}Tue=\${XXX}\${HT}Wed=\${XXX}\${HT}Thu=\${XXX}\${HT}Fri=\${XXX}\${HT}Sat=\${XXX}

说明：

KeyID： 快捷键ID

值	对应键
---	-----

1	F1
---	----

- 2 F2
- 3 F3
- 4 F4
- 5 F5
- 6 F6
- 7 F7
- 8 F8

KeyFun: 快捷键功能

- | 值 | 对应功能 |
|---|----------|
| 0 | 未定义 |
| 1 | 状态键 |
| 2 | 工作号码 |
| 3 | 短消息 |
| 4 | 按键求助 |
| 5 | 查下考勤记录 |
| 6 | 查询最后考勤记录 |

StatusCode: 考勤状态

ShowName: 状态名称

AutoState: 自动切换

AutoTime: 周一到周日自动切换时间, 08:00;09:00;10:00;11:00;12:00;13:00;14:00

Sun: 周日是否切换

Mon: 周一是否切换

Tue: 周二是否切换

Wed: 周三是否切换

Thu: 周四是否切换

Fri: 周五是否切换

Sat: 周六是否切换

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

12.2.14 门禁组

命令格式如下

C:{\$CmdID}:DATA{\$SP}UPDATE{\$SP}AccGroup{\$SP}ID={\$XXX}{\$HT}Verify={\$XXX}{\$HT}ValidHoliday={\$XXX}{\$HT}TZ={\$XXX}

说明:

ID: 门禁组编号

Verify: 组验证方式, 默认值为0, 见(附录7)

validholiday: 节假日是否有效: 取值范围 0-1

TZ格式: 比如: TZ=1;1;0;0: 第1个数字表示是否使用组时间段, 第2个数字表示时间段1, 第3个参数表示时间段2, 第4个参数表示时间段3

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

12.2.15 门禁时间表

命令格式如下

C:{\$CmdID}:DATA{\$SP}UPDATE{\$SP}AccTimeZone{\$SP}UID={\$XXX}{\$HT}SunStart={\$XXX}{\$HT}SunEnd={\$XXX}{\$HT}MonStart={\$XXX}{\$HT}MonEnd={\$XXX}{\$HT}TuesStart={\$XXX}{\$HT}TuesEnd={\$XXX}{\$HT}WedStart={\$XXX}{\$HT}WedEnd={\$XXX}{\$HT}ThursStart={\$XXX}{\$HT}ThursEnd={\$XXX}{\$HT}FriStart={\$XXX}{\$HT}FriEnd={\$XXX}{\$HT}SatStart={\$XXX}{\$HT}SatEnd={\$XXX}

说明:

UID: 时间段编号

SunStart: 星期天开始时间, 1159表示 11点 59 分

SunEnd: 星期天结束时间, 2359表示 23点 59 分

MonStart: 星期一开始时间

MonEnd: 星期一结束时间

TuesStart: 星期二开始时间

TuesEnd: 星期二结束时间

WedStart: 星期三开始时间

WedEnd: 星期三结束时间

ThursStart: 星期四开始时间

ThursEnd: 星期四结束时间

FriStart: 星期五开始时间

FriEnd: 星期五结束时间

SatStart: 星期六开始时间

SatEnd: 星期六结束时间

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

12.2.16 门禁节假日

命令格式如下

C:\${CmdID}:DATA\${SP}UPDATE\${SP}AccHoliday\${SP}UID=\${XXX}\${HT}HolidayName=\${XXX}\${HT}StartDate=\${XXX}\${HT}EndDate=\${XXX}\${HT}TimeZone=\${XXX}

说明：

UID: 节假日编号

HolidayName: 节假日名称。

StartDate: 1123表示 11月23日

EndDate: 1125表示 11月25

TimeZone: 时间段编号

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

12.2.17 门禁多组验证

命令格式如下

C:\${CmdID}:DATA\${SP}UPDATE\${SP}AccUnLockComb\${SP}UID=\${XXX}\${HT}Group1=\${XXX}\${HT}Group2=\${XXX}\${HT}Group3=\${XXX}\${HT}Group4=\${XXX}\${HT}Group5=\${XXX}

说明：

UID: 组验证编号

Group1: 人员分組組號。人員信息中的組號

Group2: 人員分組組號。人員信息中的組號

Group3: 人員分組組號。人員信息中的組號

Group4: 人員分組組號。人員信息中的組號

Group5: 人員分組組號。人員信息中的組號

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

12.2.18 身份证黑名单下发

格式如下

C: \${CmdID}:DATA\${SP}UPDATE\${SP}Blacklist\${SP}IDNum=\${XXX}

说明如下

IDNum: 身份证号

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

12.3 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

12.3.1 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

12.3.2 指纹模板

令格式如下

C: \${CmdID}:DATA\${SP}DELETE\${SP}FINGERTMP\${SP}PIN=\${XXX}

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

12.3.3 面部模板

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

12.3.4 指静脉模板

命令格式如下

C:\${CmdID}:DATA\${SP}DELETE\${SP}FVEIN\${SP}Pin=\${XXX}

C:\${CmdID}:DATA\${SP}DELETE\${SP}FVEIN\${SP}Pin=\${XXX}\${HT}FID=\${XXX}

说明:

Pin=\${XXX}: 用户工号

FID=\${XXX}: 手指编号, (0~9)

删除指定用户的指静脉模版

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

12.3.5 一体化模板

命令格式如下

C:\${CmdID}:DATA\${SP}DELETE\${SP}BIODATA\${SP}Pin=\${XXX}

C:\${CmdID}:DATA\${SP}DELETE\${SP}BIODATA\${SP}Pin=\${XXX}{HT}Type=\${XXX}

C:\${CmdID}:DATA\${SP}DELETE\${SP}BIODATA\${SP}Pin=\${XXX}{HT}Type=\${XXX}{HT}No=\${XXX}

说明:

各字段说明见上传一体化模板功能

删除指定用户的一体化模版

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

12.3.6 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

12.3.7 比对照片

命令格式如下

C:\${CmdID}:DATA\${SP}DELETE\${SP}BIOPHOTO\${SP}PIN=\${XXX}

说明:

PIN=\${XXX}: 用户工号。

删除指定用户的比对照片。

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

12.3.8 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

命令执行结果如何回复见[回复命令], Return返回值见(附录1), 返回内容格式如下:

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

12.3.9 工作代码

命令格式如下

C:\${CmdID}:DATA\${SP}DELETE\${SP}WORKCODE\${SP}CODE=\${XXX}

说明:

CODE=\${XXX}: 工作代码编码。

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

12.3.10 宣传照片

命令格式如下

C:\${CmdID}:DATA\${SP}DELETE\${SP}ADPIC\${SP}Index=\${XXX}

说明:

Index=\${XXX}: 图片索引

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

12.4 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

12.4.1 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](#)”

12.4.2 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](#)”

12.4.3 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](#)”

12.4.4 指纹模板

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](#)”

12.5 CLEAR命令

12.5.1 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

12.5.2 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

12.5.3 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

12.6 检查命令

12.6.1 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

12.6.2 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

12.6.3 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](#replycmd)功能，Return返回值见（附录1）(#appendix1)，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

12.7 配置选项命令

12.7.1 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

12.7.2 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

12.7.3 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。

12.8 文件命令

12.8.1 取客户端内的文件

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

12.8.2 发送文件到客户端

功能1:

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PutFile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

功能2:

C:\${CmdID}:PutFile\${SP}\${URL}\${HT}\${FilePath}\${HT}Action=\${Value}

说明:

使用PutFile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

Action: 描述下载完该文件后需做何种操作，目前支持如下:

Action=SyncData: 表示下载的文件需要设备将同样的数据类型同步成文件中的数据，即舍弃设备中原有的数据，该行为需要附加两个参数**TableName**和**RecordCount**，完整命令如下

C:\${CmdID}:PutFile\${SP}\${URL}\${HT}\${FilePath}\${HT}Action=\${Value}\${HT}TableName=\${Value}\${HT}RecordCount=\${Value}

TableName: 表示数据类型，支持如下

\${Value} 数据类型

USERINFO 用户数据

FINGERTMP 指纹数据

FACE 面部数据

RecordCount: 数据包中的记录数。

Action=AppendData: 表示下载的文件的的需要追加到设备中，完整命令如下

C:\${CmdID}:PutFile\${SP}\${URL}\${HT}\${FilePath}\${HT}Action=AppendData

压缩包中的内容格式同数据命令下发格式，如

C:123:DATA UPDATE USERINFO PIN=1 Name=1 Pri=0 Passwd=1 Grp=1

C:124:DATA UPDATE FINGERTMP

PIN=1 FID=11 SIZE=28 VALID=1 TMP=c2FmZHNhd3Jyd3JlcmVyZXJlcnc=

C:125:DATA UPDATE FACE

PIN=1 FID=0 SIZE=28 VALID=1 TMP=c2FmZHNhd3Jyd3JlcmVyZXJlcnc=

C:126:DATA UPDATE FACE

PIN=1 FID=1 SIZE=28 VALID=1 TMP=c2FmZHNhd3Jyd3JlcmVyZXJlcnc=

C:127:DATA UPDATE FACE

PIN=1 FID=2 SIZE=28 VALID=1 TMP=c2FmZHNhd3Jyd3JlcmVyZXJlcnc=

C:128:DATA UPDATE FACE

PIN=1 FID=3 SIZE=28 VALID=1 TMP=c2FmZHNhd3Jyd3JlcmVyZXJlcnc=

C:129:DATA UPDATE FACE

PIN=1 FID=4 SIZE=28 VALID=1 TMP=c2FmZHNhd3Jyd3JlcmVyZXJlcnc=

C:130:DATA UPDATE FACE

```

PIN=1  FID=5  SIZE=28  VALID=1  TMP=c2FmZHNhd3Jyd3JlcmVyZXJlcnc=
      C:131:DATA UPDATE FACE
PIN=1  FID=6  SIZE=28  VALID=1  TMP=c2FmZHNhd3Jyd3JlcmVyZXJlcnc=
      C:132:DATA UPDATE FACE
PIN=1  FID=7  SIZE=28  VALID=1  TMP=c2FmZHNhd3Jyd3JlcmVyZXJlcnc=
      C:133:DATA UPDATE FACE
PIN=1  FID=8  SIZE=28  VALID=1  TMP=c2FmZHNhd3Jyd3JlcmVyZXJlcnc=
      C:134:DATA UPDATE FACE
PIN=1  FID=9  SIZE=28  VALID=1  TMP=c2FmZHNhd3Jyd3JlcmVyZXJlcnc=
      C:135:DATA UPDATE FACE
PIN=1  FID=10  SIZE=28  VALID=1  TMP=c2FmZHNhd3Jyd3JlcmVyZXJlcnc=
      C:136:DATA UPDATE FACE
PIN=1  FID=11  SIZE=28  VALID=1  TMP=c2FmZHNhd3Jyd3JlcmVyZXJlcnc=
      .....

```

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

12.9 远程登记命令

12.9.1 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

12.9.2 登记卡号

由服务器发起，在客户端登记指纹，命令格式如下

C:XXX:ENROLL_MF PIN=%s\tRETRY=%d

如： C:123:ENROLL_MF PIN=408\tRETRY=3

说明：

PIN— 用户号

RETRY— 重试次数

返回值：

- 0 命令执行成功
- 1 参数错误
- 3 存取错误
- 4 登记失败重试次数
- 5 登记超时退出
- 6 按 Esc 退出登记界面

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_MF

12.9.3 登记人脸，掌纹（一体化模板）

由服务器发起，在客户端登记指纹，命令格式如下

ENROLL_BIO TYPE=%?\tPIN=%?\tCardNo=%?\tRETRY=%?\tOVERWRITE=%?

TYPE：

- 0 /**< 通用模板*/
- 1 /**< 指纹*/
- 2 /**< 面部*/
- 3 /**< 语音*/
- 4 /**< 虹膜*/
- 5 /**< 视网膜*/
- 6 /**< 掌静脉*/

7 /**< 指静脉*/

8 /**< 掌纹*/

9 /**< 可见光面部*/

PIN : 登记的工号

CardNo : 登记的卡号

RETRY : 若登记失败, 需重试的次数

OVERWRITE : 是否覆盖面部, 0 表示相应用户存在面部则不覆盖面部返回错误信息, 1 表示相应用户存在面部也覆盖该面部。

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_BIO

12.10 控制命令

12.10.1 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

12.10.2 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

12.10.3 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

12.11 其他命令

12.11.1 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=shello
ut.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

12.11.2 在线升级

应用场景：用于从服务器软件上远程升级客户端设备的固件。

方式一：远程升级客户端固件，兼容控制器和新架构一体机，升级文件需要服务器转格式后下发到客户端。

格式：

服务器下发命令：

C:\${CmdID}:UPGRADE\$(SP)checksum=\${XXX},url=\$(URL),size=\${XXX}

客户端从命令中带有的 URL 下载升级包:

```
GET /iclock/file?SN=${SerialNumber}&url=${URL} HTTP/1.1
```

.....

客户端上传执行结果:

```
ID=${CmdID}&Return=${XXX}&CMD=UPGRADE
```

注释:

Checksum: 表示 md5 校验值

url: 表示升级文件下载资源地址, 升级文件名称为 emfw.cfg

size 表示原始文件大小

特别说明:

本方法中, 固件升级文件在下发时, 由服务器转化为 **base64** 格式数据。客户端接收到后需要转化为二进制格式且名称为 **emfw.cfg** 的文件

示例:

服务器下发升级固件命令:

```
C:384:UPGRADE
```

```
checksum=a5bf4dcd6020f408589224274aab132d,url=http*//localhost*8088\fireware\F20\admin\emfw.cfg,size=2312
```

客户端请求下载升级包:

```
GET /iclock/file?SN=3383154200002&url=http://192.168.213.17:8088/fireware/F20/admin/emfw.cfg HTTP/1.1
```

```
Cookie: token=af65a75608cf5b80fbb3b48f0b4df95a
```

```
Host: 192.168.213.17:8088
```

.....

客户端上传执行成功结果:

```
ID=384&Return=0&CMD=UPGRADE
```

方式二: 远程升级客户端固件, 直接获取文件, 不需要转格式, 客户端直接获取文件格式:

服务器下发命令:

```
C:${CmdID}:UPGRADE$(SP)type=1,checksum=${XXX},size=${XXX},url=${URL}
```

客户端请求下载升级包:

```
GET /iclock/file?SN=${SerialNumber}&url=${URL} HTTP/1.1
```

```
Cookie: token=af65a75608cf5b80fbb3b48f0b4df95a
```

Host: 192.168.213.17:8088

.....

客户端上传执行结果:

ID=\${CmdID}&Return=\${XXX}&CMD=UPGRADE

注释:

type: 为 1 表示从 url 获取升级文件, 暂时只支持 1。

Checksum: 表示 md5 校验值

url: 表示升级文件下载资源地址, 升级文件名称为 emfw.cfg

size 表示升级包大小

特别说明:

本方法中, 客户端直接获取到的就是固件升级文件, 无需再转格式。

示例:

服务器下发命令:

C:123:UPGRADE

type=1,checksum=oqoier9883kjankdefi894eu,size=6558,url=http://192.168.0.13:89/data/emfw.cfg

客户端请求下载升级包:

GET /iclock/file?SN=3383154200002&url=http://192.168.0.13:89/data/emfw.cfg HTTP/1.1

Cookie: token=af65a75608cf5b80fbb3b48f0b4df95a

Host: 192.168.0.13:89

.....

客户端上传执行成功结果:

ID=384&Return=0&CMD=UPGRADE

12.11.3 后台验证

应用场景: 在考勤设备上验证指纹/人脸成功之后将该人员编号通过push上传到后端系统, 后端系统收到人员编号进行逻辑判断后会返回一个结果 (是否允许验证通过) 给考勤设备。

格式:

客户端数据发送

POST /iclock/cdata?SN=\${SerialNumber}&type=PostVerifyData HTTP/1.1

Host: \${ServerIP}:\${ServerPort}

.....

\${PostData} //上传的数据

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

type=PostRecordData表示上传记录数据

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}

.....

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。

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。

参数配置：PostSelfDefineDataType=PostVerifyData

13 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

14 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} 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

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=\${XXX}\${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](#)

15 附录

15.1 附录1

通用错误码	描述
-----	----
0	成功
-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
-1008	获取服务器数据失败

Enroll_FP/Enroll_BIO错误码	描述
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

PutFile（Action=SyncData）错误码	描述
n > 0	同步数据，成功处理n条指令

15.2 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

15.3 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	

15.4 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			

15.5 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

15.6 附录6

协议版本规则

- 已发布的协议版本:

2.2.14

2.3.0

2.4.0

2.4.1

加密协议版本: 2.4.0及以上

- 设备端:

设备将当前push使用的协议版本通过下面协议推送给服务端

GET

/iclock/cdata?SN=\${SerialNumber}&options=all&pushver=\${XXX}&language=\${XXX}&pushcommkey=\${XXX}

服务端针对这个请求返回服务端使用哪个发布协议版本开发, 将协议版本返回给设备。

PushProtVer=xxx, 没返回这个参数的话, 设备默认服务器使用的协议版本为2.2.14

设备根据当前push使用的协议版本与服务端返回的协议版本比较，使用较低的那个版本交互。

• 服务端：

服务端根据下面的请求获得设备端push使用的协议版本，如果没有pushver字段，那么默认设备使用的是2.2.14协议版本。

GET

/iclock/cdata?SN=\${SerialNumber}&options=all&pushver=\${XXX}&language=\${XXX}&pushcommkey=\${XXX}

服务端需要返回软件使用哪个发布协议版本：

PushProtVer=xxx

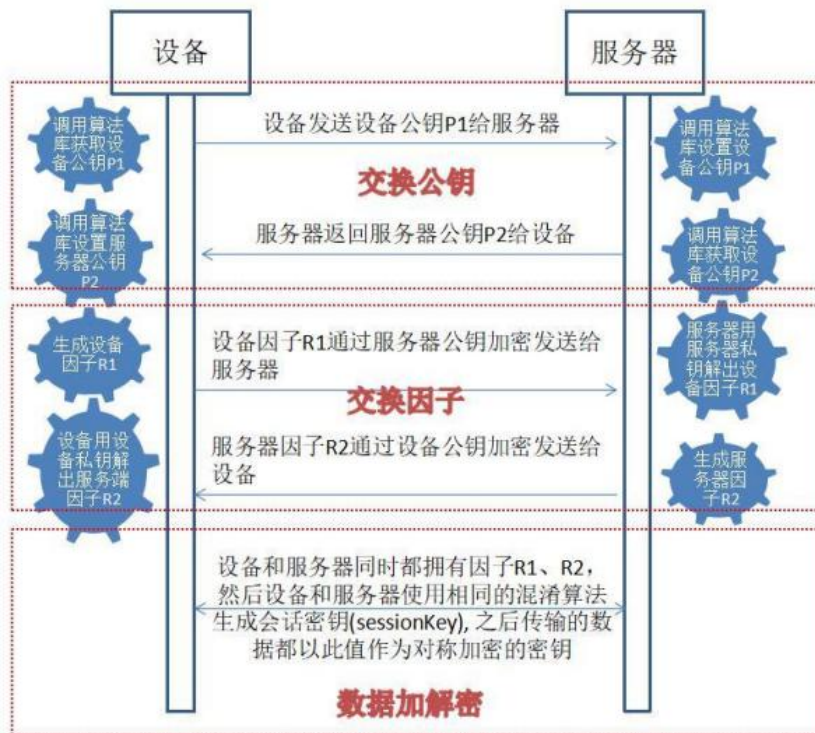
服务端根据软件使用的协议版本与设备端上传的协议版本比较，使用较低的那个版本交互。

15.7 附录7

验证方式	描述
0	指静脉或人脸或指纹或卡或密码（自动识别）
1	仅指纹
2	工号验证
3	仅密码
4	仅卡
5	指纹或密码
6	指纹或卡
7	卡或密码
8	工号加指纹
9	指纹加密码
10	卡加指纹
11	卡加密码
12	指纹加密码加卡
13	工号加指纹加密码
14	工号加指纹 或 卡加指纹
15	人脸
16	人脸加指纹
17	人脸加密码
18	人脸加卡
19	人脸加指纹加卡
20	人脸加指纹加密码
21	指静脉
22	指静脉加密码
23	指静脉加卡
24	指静脉加密码加卡
25	掌纹
26	掌纹加卡
27	掌纹加面部
28	掌纹加指纹

15.8 附录8

数据加密密钥交换方案

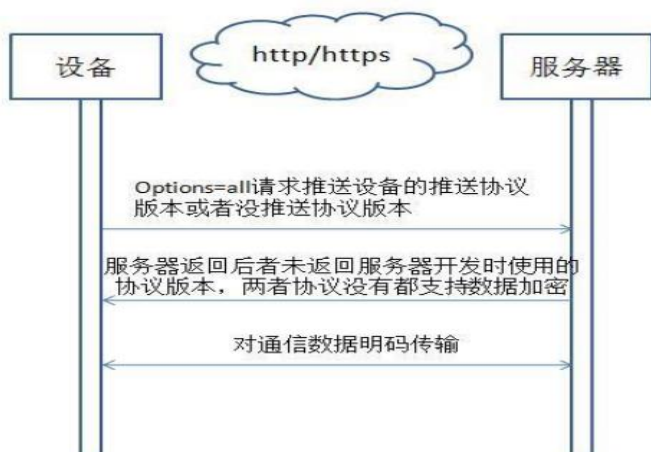


- 算法：加密算法库将统一进行封装，设备使用的算法库为静态库。
- 方案：
 - a) 设备和服务器重连的时候初始化非对称加密的公私钥。
 - b) 设备和服务器交换公钥：
 - 设备发送设备公钥 $P1$ 给服务器。
 - 服务器返回服务器公钥 $P2$ 给设备。
 - 完成公钥交换。设备和服务器同时都拥有公钥 $P1$ 、 $P2$ 。
 - c) 设备和服务器交换因子：
 - 设备生成因子 $R1$ ，并通过服务器公钥加密发送给服务端。
 - 服务器用服务器私钥解出设备因子 $R1$ 。
 - 服务器生成因子 $R2$ ，并通过设备公钥加密发送给设备。
 - 设备用设备私钥解出服务端因子 $R2$ 。
 - 完成因子交换。设备和服务器同时都拥有因子 $R1$ 、 $R2$ 。
 - d) 设备和服务器同时都拥有因子 $R1$ 、 $R2$ ，然后设备和服务器使用相同的混淆算法生成会话密钥(sessionKey)，之后传输的数据都以此值作为对称加密的密钥。

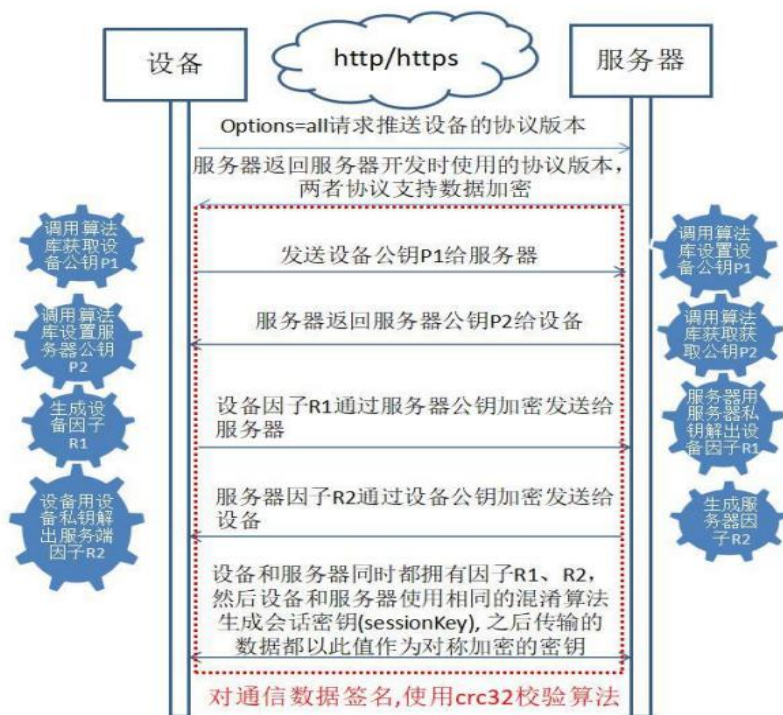
兼容方案

根据设备和服务器使用的协议版本实现兼容，情况如下：

• 情况一



• 情况二



注释:

- 设备根据设置的服务器地址来判断使用https还是http传输。
- 设备现有的第一个请求协议头中增加pushver字段为设备当前通信协议版本号，软件返回的数据内容中增加PushProtVer表示软件是基于哪个协议版本开发的。设备和服务器两个协议版本比较取最低的，按最低协议版本进行通信。

情况一： 当服务器和设备的协议版本不都支持，则使用对数据通信进行明码传输。

情况二： 设定某个协议版本是支持数据加密的，当服务器和设备的协议版本都支持，则使用数据加密方案。

交互顺序如下：

- 新增协议对设备和服务器的公钥 P1、P2 进行交换。

- 新增协议对设备和服务器的因子 R1、R2 进行交换。
- 对通信数据签名进行 crc32 校验，设备和服务器同时都拥有因子 R1、R2，然后设备和服务器使用相同的混淆算法生成会话密钥(sessionKey)，之后传输的数据都以此值作为对称加密的密钥。

15.9 附录9

错误码	描述
00000000	成功
D01E0001	探测人脸失败
D01E0002	人脸遮挡
D01E0003	清晰度不够
D01E0004	人脸角度太大
D01E0005	活体检测失败
D01E0006	提取模板失败

依据错误码产生端+模块+类型+错误值定义。

错误产生端(第一位)

D: 设备端返回的错误码

S: 软件段返回的错误码

模块(第二位~第三位)

设备端:

01: PUSH通信模块

02: 模板处理模块

03: 硬件交互模块

04: PULL通信模块

05: 脱机通信模块

06: 数据中转模块

07: 许可服务模块

软件端:

待定

类型(第四位)

E: ERROR

错误值(第五位~第八位)

整型数据

全国免费技术咨询热线：4006-900-999

广东省东莞市塘厦平山188工业大道26号中控智慧产业园

广东省深圳市龙岗区坂田五和大道北中控智慧大厦

厦门市集美区软件园三期B02栋20层

www.zkteco.com

