**管理端与X00服务间的请求**

目录

[1 一般概念 1](#_Toc463697930)

[2 请求分类 2](#_Toc463697931)

[3 新增数据结构 2](#_Toc463697932)

[3.1 写法说明 2](#_Toc463697933)

[3.2 从服务获得识别记录GetSrvRecogRecordT 2](#_Toc463697934)

[3.3 识别记录RecogRecordT 3](#_Toc463697935)

[3.4 从服务获得人员列表的条件GetSrvPersonT 4](#_Toc463697936)

[3.5 管理端从服务端获得的人员信息SrvPersonBaseT 4](#_Toc463697937)

[3.6 与服务当前连接的设备SrvConnectDevT 5](#_Toc463697938)

[3.7 人员与设备的绑定关系DevPersonBindT 5](#_Toc463697939)

[4 请求列表 6](#_Toc463697940)

[5 请求及回应 6](#_Toc463697941)

[5.1 一般回应格式 6](#_Toc463697942)

[5.2 获得服务现有人员列表 7](#_Toc463697943)

[5.2.1 请求 7](#_Toc463697944)

[5.2.2 回应 7](#_Toc463697945)

[5.3 获得服务识别记录 8](#_Toc463697946)

[5.3.1 请求 8](#_Toc463697947)

[5.3.2 回应 9](#_Toc463697948)

[5.4 获取当前与服务连接的设备请求 9](#_Toc463697949)

[5.4.1 请求 9](#_Toc463697950)

[5.4.2 回应 10](#_Toc463697951)

[5.5 设置人员与设备间绑定关系 10](#_Toc463697952)

[5.5.1 请求 10](#_Toc463697953)

[5.5.2 回应 11](#_Toc463697954)

# 一般概念

1. 管理端与服务端之间的请求和设备与服务端之间的请求分开。
2. 管理端与服务端连接的端口缺省是9998；设备与服务端连接端口缺省是9999。
3. 一般来说，都是由管理端在需要的时候连接服务端，并主动发送请求到服务端。
4. 管理端与服务端之间不需要保持长连接，不需要与服务端之间发送保活请求。

# 请求分类

人员相关：

~~新增、修改、删除（一期不做）~~

获取服务上现有人员；

获取指定设备上现有人员；

设备相关：

获取与服务连接的设备（包括设备详情）；

~~（一期不做）~~

~~控制设备重启~~

~~控制设备关机~~

~~修改设备IP~~

识别记录相关：

获取服务上符合要求的识别记录

绑定相关：

设置人员与设备绑定

设置人员与设备解绑

# 新增数据结构

## 写法说明示例

int32 占4个字节的整数

char[64] 固定长度为64的字符数组

UUID 占16个字节

## 从服务获得识别记录GetSrvRecogRecordT

struct GetSrvRecogRecordT

{

int32 lastRecogIndex; // 查询的识别记录表中的自增index

// index>0，则为正向查询，从index开始，包括index

// index=0，默认为1

// index<0，则为逆向查询，index=-1是从最后一个开始，包括最后一个

// 依此类推

int32 countInPage; // 一次获取的识别记录个数，最多支持2048条记录；如果这里的参数值超过2048，则返回的识别记录条数最多仍然是2048

int32 devSnValid; // 参数中的设备序列号是否有效

// 0——无效，1——有效

char[64] devSn; // 要求发送指定设备的识别记录

};

数据定长，76Bytes

|  |  |  |  |
| --- | --- | --- | --- |
| lastRecogIndex | countInPage | devSnValid | devSn |
| 4Bytes | 4Bytes | 4Bytes | 64Bytes |

相关请求[REQ\_GET\_SRV\_RECOG\_RECORD](#_请求)。

## 识别记录RecogRecordT

服务返回的结构与服务和X00设备间数据的定义相同，但是意义稍有不同：

struct RecogRecordT

{

char[64] deviceSn; // 设备序列号，标识是在哪台设备上识别的，

// 应为唯一值，此值目前未确定

UUID featureId; // 特征ID，无效，仅占位

UUID personId; // 人员ID

char[64] name; // 人员姓名

int32 recogId; // 在服务数据库中的识别记录序列号，连续且自增长

// （服务和X00设备间数据的意义是设备数据库中识别记录序列号）

int32 eyeFlag; // 标识是哪只眼睛，无效，仅占位

char[20] idenTime; // 识别时间

};

共188Bytes

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| deviceSn | featureId | personId | name | recogId | eyeFlag | idenTime |
| 64Bytes | 16Bytes UUID | 16Bytes UUID | 64Bytes | 4Bytes | 4Bytes | 20Bytes |

相关请求[ACK\_GET\_SRV\_RECOG\_RECORD](#_回应)。

## 获得人员列表的条件GetPersonCondT

结构总长度固定，为264字节。

struct GetPersonCondT

{

int32 pageIndex; // 页面序号

int32 countInPage; // 一页记录个数

int32 irisStatus; // 注册状态

char[64] devSn; // 设备序列号

char[64] personName; // 人员姓名

char[64] workSn; // 人员工号

char[20] idCard; // 身份证号

char[20] startTime; // 起始时间

char[20] endTime; // 结束时间

};

字段详细说明（长度单位：字节）

|  |  |  |
| --- | --- | --- |
| 内容 | 长度 | 说明 |
| pageIndex | 4 | 页面序号，从0开始，如果发来的index<0，自动设置为0 |
| countInPage | 4 | 一次获取的人员记录个数，最多支持2048条记录；如果这里的参数值超过2048，则返回的记录条数最多仍然是2048 |
| irisStatus | 4 | 是否已注册虹膜，查询条件之一。1：已注册；0：未注册；-1：全部 |
| devSn | 64 | 设备序列号，查询条件之一，支持模糊查询。如果为空，则忽略此查询条件 |
| personName | 64 | 人员姓名，查询条件之一，支持模糊查询。如果为空，则忽略此查询条件 |
| workSn | 64 | 人员工号，查询条件之一，支持模糊查询。如果为空，则忽略此查询条件 |
| idCard | 20 | 身份证号，查询条件之一，支持模糊查询。如果为空，则忽略此查询条件 |
| startTime | 20 | 起始时间，查询条件之一。如果为空，则忽略此查询条件；如不为空，则条件为人员创建时间>=startTime |
| endTime | 20 | 结束时间，查询条件之一。如果为空，则忽略此查询条件；如不为空，则条件为人员创建时间<startTime |

相关请求[REQ\_GET\_SRV\_PERSON\_LIST](#_请求_1)，REQ\_GET\_DEV\_PERSON\_LIST。

## 管理端从服务端获得的人员信息SrvPersonBaseT

struct SrvPersonBaseT

{

UUID id; // 人员ID，UUID

int32 opToken; // 暂时无效，设置为0即可

int32 hasFeature; // 是否带有注册特征

char[64] name; // 人员姓名  
 char[64] workSn; // 人员工号

int32 type; // 人员类型，是否管理员，暂时无效，置0即可

int32 sex; // 性别

char[20] idCardNumber; // 身份证号

char[64] departName; // 部门名称

char[16] cardNumber; // 卡号

char[128] memo; // 备注

};

共388Bytes

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| id | opToken | hasFeature | name | workSn | type | sex | idCardNumber | departName | cardNumber | memo |
| 16Bytes | 4Bytes | 4Bytes | 64Bytes | 64Bytes | 4Bytes | 4Bytes | 20Bytes | 64Bytes | 16Bytes | 128Bytes |

相关请求[ACK\_GET\_SRV\_PERSON\_LIST](#_响应获得服务现有人员列表)。

## 获得当前连接设备条件SrvConnectDevCondT

struct SrvConnectDevT

{

int32 pageIndex; // 页面序号

int32 countInPage; // 一页记录个数

}

|  |  |  |
| --- | --- | --- |
| 内容 | 长度 | 说明 |
| pageIndex | 4 | 页面序号，从0开始，如果发来的index<0，自动设置为0 |
| countInPage | 4 | 一次获取的设备个数，最多支持2048条记录；如果这里的参数值超过2048，则返回的记录条数最多仍然是2048 |

## 设备信息SrvConnectDevT

struct SrvConnectDevT

{

char[16] ip; // 设备IP，仅为辅助标识，不一定唯一

char[64] deviceSn; // 设备序列号，作为设备的唯一标识

int32 workMode; // 工作模式

int32 workStatus; // 工作状态

};

数据定长80Bytes

|  |  |
| --- | --- |
| deviceSn | ip |
| 64Bytes | 16Bytes |

相关请求[ACK\_GET\_CONNECT\_DEVICE](#_回应获取当前与服务连接的设备)。

## 人员与设备的绑定关系DevPersonBindT

struct DevPersonBindT

{

int binding; // 绑定(1)或解绑(0)

char[64] deviceSn; // 设备序列号，作为设备的唯一标识

UUID personId; // 人员ID，作为人员的唯一标识

};

数据定长84Bytes

|  |  |  |
| --- | --- | --- |
| binding | deviceSn | personId |
| 4Bytes | 64Bytes | 16Bytes(UUID) |

相关请求[REQ\_BIND\_PERSON\_DEVICE](#_请求_2)。

# 请求列表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 从服务获得人员列表 | REQ\_GET\_SRV\_PERSON\_LIST | 0x0231 | 获得服务端人员列表 | 管理端->服务器端 |
| 响应从服务获得人员列表 | ACK\_GET\_SRV\_PERSON\_LIST | 0x4231 | 响应获得服务端人员列表 | 服务器端->管理端 |
|  |  |  |  |  |
| 获取服务识别记录 | REQ\_GET\_SRV\_RECOG\_RECORD | 0x0451 | 要求获得服务保存的识别记录 | 管理端->服务器端 |
| 响应获取服务识别记录 | ACK\_GET\_SRV\_RECOG\_RECORD | 0x4451 | 响应获得服务保存的识别记录请求 | 服务器端->管理端 |
|  |  |  |  |  |
| 获取当前与服务连接的设备 | REQ\_GET\_CONNECT\_DEVICE | 0x0b01 | 客户端要求获取当前与服务处于连接状态的设备 | 客户端->服务器端 |
| 响应获取当前与服务连接的设备 | ACK\_GET\_CONNECT\_DEVICE | 0x4b01 | 回应当前与服务端连接的设备列表 | 服务器端->客户端 |
|  |  |  |  |  |
| 设置人员与设备间的绑定关系 | REQ\_BIND\_PERSON\_DEVICE | 0x0b02 | 客户端向服务器端设置人员与设备的绑定关系 | 客户端->服务器端 |
| 回应设置人员与设备间的绑定关系 | ACK\_BIND\_PERSON\_DEVICE | 0x4b02 | 回应人员与设备的绑定关系 | 服务器端->客户端 |

# 请求及回应

## 一般回应格式

大部分请求的回应帧格式如没有特殊说明，则均按照如下形式。

请求帧：

如果错误码=0，表示对应的请求成功，格式为

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 同步头 | 请求码 | 请求子码 | 实际数据长度 | 错误码 | 传送的数据 | 校验 |
| 0x5353  (2 Bytes) | (2 Bytes) | 0xaaaa  (2 Bytes) | 4  (4 Bytes) | 0  (4Bytes) | （无数据） | (2 Bytes) |

请求帧字节长度固定为：16Bytes。

如果错误码<0，表示对应的请求失败，并给出错误描述，格式为

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 同步头 | 请求码 | 请求子码 | 实际数据长度 | 错误码 | 传送的数据 | 校验 |
| 0x5353  (2 Bytes) | (2 Bytes) | 0xaaaa  (2 Bytes) | 264  (4 Bytes) | 实际值  (4Bytes) | ErrorString  (260Bytes) | (2 Bytes) |

请求帧字节长度固定为：276Bytes。

## 获得服务现有人员列表

### 请求

请求方向：管理端→服务器端。

此请求用于管理端向服务器端获取当前服务数据库中已有的人员信息。

请求名称：REQ\_GET\_SRV\_PERSON\_LIST (0x0231)

请求帧：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 同步头 | 请求码 | 请求子码 | 实际数据长度 | 传送的数据 | 校验 |
| 0x5353  (2 Bytes) | 0x0231  (2 Bytes) | 0xaaaa  (2 Bytes) | 8  (4 Bytes) | GetSrvPersonT  8Bytes | (2 Bytes) |

请求帧字节长度固定为：20Bytes。

其中从服务获得人员列表条件的数据结构参见[GetSrvPersonT](#_从服务获得人员列表的条件GetSrvPersonT)。

### 回应

请求方向：服务器端→管理端。

此请求用于服务器端响应管理端要求获取当前服务数据中已有的人员信息。

请求名称：ACK\_GET\_SRV\_PERSON\_LIST (0x4231)

请求帧：

当错误码为0时（表示请求成功执行，回应中包括人员的信息）

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 同步头 | 请求码 | 请求子码 | 实际数据长度 | 错误码 | 传送的数据 | 校验 |
| 0x5353  (2 Bytes) | 0x4231  (2 Bytes) | 0xaaaa  (2 Bytes) | 4+传送的数据长度  (4 Bytes) | 0  (4Bytes) | SrvPersonBaseT数组  (4+388Bytes\*个数) | (2 Bytes) |

请求帧字节长度不是固定值，以传送的实际数据长度为准。如果查询到的结果为0条，则返回的错误码仍为0（没有错误），但是传送的数据中，数组个数为0。

**实际传送的数据结构**为：

typedef list<SrvPersonBaseT> lstPerson;

|  |  |  |  |
| --- | --- | --- | --- |
| len | SrvPersonBaseT | SrvPersonBaseT | …… |
| 4Bytes | 388Bytes | 388Bytes | …… |

其中，len表示后面的人员数据长度，即有多少个人员的数据。

其中人员信息的数据结构参见[SrvPersonBaseT](#_管理端从服务端获得的人员信息SrvPersonBaseT)。

当错误码不等于0，表示请求未成功执行，请求帧格式参见“一般回应格式”。

注意：由于每次获取的人员数量有限（2048个），所以如果要获取全部人员，可能需要经过多次请求，每次请求修改请求参数中的lastPersonIndex。直至最后获取返回成功，但是无数据了，表明人员获取已完成。

## 获得服务识别记录

### 请求

请求方向：管理端→服务器端

此请求用于管理端向服务器端要求获得指定的识别记录。

请求名称：REQ\_GET\_SRV\_RECOG\_RECORD (0x0451)

请求帧：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 同步头 | 请求码 | 请求子码 | 实际数据长度 | 传送的数据 | 校验 |
| 0x5353  (2 Bytes) | 0x0451  (2 Bytes) | 0xaaaa  (2 Bytes) | 76  (4 Bytes) | GeSrvRecogRecordT  (76Bytes) | (2 Bytes) |

请求帧字节长度固定为84Bytes。

从服务获得指定识别记录请求的数据结构参见[GetSrvRecogRecordT](#_从服务获得识别记录GetSrvRecogRecordT)。

### 回应

请求方向：服务器端→管理端

此请求用于服务器端响应管理端要求获得服务端数据库中保存的指定识别记录的请求，可能成功或失败。如果成功，则回应中包含符合要求的识别记录（个数可能为0）。

请求名称：ACK\_GET\_SRV\_RECOG\_RECORD (0x4451)

请求帧：

如果响应成功，错误码为0，发送数据为符合要求的识别记录数组

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 同步头 | 请求码 | 请求子码 | 实际数据长度 | 错误码 | 传送的数据 | 校验 |
| 0x5353  (2 Bytes) | 0x4451  (2 Bytes) | 0xaaaa  (2 Bytes) | 4+传送的数据长度  (4 Bytes) | 0  (4Bytes) | RecogRecordT数组  (4+120Bytes\*个数) | (2 Bytes) |

请求帧字节长度不是固定值，以传送的实际数据长度为准。

单独一条识别记录的数据结构参见[RecogRecordT](#_识别记录RecogRecordT)。

如果响应不成功，错误码不等于0，请求帧格式参见“一般回应格式”。

## 获取当前与服务连接的设备请求

### 请求

请求方向：管理端→服务器端

此请求用于管理端向服务要求获取当前与服务处于连接状态的设备。

请求名称：REQ\_GET\_CONNECT\_DEVICE (0x0b01)

请求帧：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 同步头 | 请求码 | 请求子码 | 实际数据长度 | 传送的数据 | 校验 |
| 0x5353  (2 Bytes) | 0x0b01  (2 Bytes) | 0xaaaa  (2 Bytes) | 0  (4 Bytes) | 无  (0Bytes) | (2 Bytes) |

请求帧字节长度固定为：12Bytes。

### 回应

响应方向：服务器端→管理端

此响应为服务器端回应管理端要求获取当前与服务处于连接状态的设备的请求。

请求名称：ACK\_GET\_CONNECT\_DEVICE (0x4b01)

请求帧：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 同步头 | 请求码 | 请求子码 | 实际数据长度 | 错误码 | 传送的数据 | 校验 |
| 0x5353  (2 Bytes) | 0x4b01  (2 Bytes) | 0xaaaa  (2 Bytes) | 4+传送的数据长度  (4 Bytes) | 0  (4Bytes) | 设备信息数组  (4Bytes+80Bytes\*个数) | (2 Bytes) |

请求帧字节长度不是固定值，以传送的实际数据长度为准。

**实际传送的数据结构**为：

typedef list<SrvConnectDevT> lstDevice;

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| len | SrvConnectDevT | SrvConnectDevT | SrvConnectDevT | …… |
| 4Bytes | 80Bytes | 80Bytes | 80Bytes | …… |

其中，len表示后面的设备数组长度，即有多少台设备的数据。

单个设备的数据结构参见[SrvConnectDevT](#_与服务当前连接的设备SrvConnectDevT)。

## 设置人员与设备间绑定关系

### 请求

请求方向：管理端→服务器端

此请求用于管理端向服务端设置人员与设备间的绑定关系，可以绑定也可以解绑。

请求名称：REQ\_BIND\_PERSON\_DEVICE (0x0b02)

请求帧：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 同步头 | 请求码 | 请求子码 | 实际数据长度 | 传送的数据 | 校验 |
| 0x5353  (2 Bytes) | 0x0b02  (2 Bytes) | 0xaaaa  (2 Bytes) | 传送的数据长度  (4 Bytes) | 绑定信息数组  (4Bytes+84Bytes\*个数) | (2 Bytes) |

请求帧字节长度不是固定值，以传送的实际数据长度为准。

单个绑定信息的数据结构参见[DevPersonBindT](#_人员与设备的绑定关系DevPersonBindT)。

**实际传送的数据结构**为：

typedef list<DevPersonBindT> lstBinding;

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| len | DevPersonBindT | DevPersonBindT | DevPersonBindT | …… |
| 4Bytes | 84Bytes | 84Bytes | 84Bytes | …… |

其中，len表示后面的绑定信息数组长度，即有多少个绑定的数据。

### 回应

请求方向：服务器端→管理端

此请求用于服务端回应管理端设置人员与设备间的绑定关系。

请求名称：ACK\_BIND\_PERSON\_DEVICE (0x4b02)

请求帧格式参见“一般回应格式”。