

---

# 智慧工地平台

## 建设领域实名制管理终端设备数据接口说明(试行)

V1.0

成都市住房和城乡建设局制

---

## 目录

1	范围.....	1
2	目的.....	1
3	方案概述.....	1
4	采集设备（https 协议）.....	2
4.1	接口描述.....	2
4.2	请求方式.....	2
4.3	测试地址.....	2
4.4	请求参数.....	2
4.5	返回值.....	2
4.6	其它.....	3
5	考勤设备（socket 协议）.....	3
5.1	接口描述.....	3
5.2	接口请求数据包格式.....	3
5.3	接口详情.....	5
5.3.1	心跳包.....	5
5.3.2	人员批量下发.....	6
5.3.3	上传识别结果.....	7
5.3.4	删除人员信息.....	8
5.4	其它.....	9
6	考勤设备（https 协议）.....	9
6.1	接口描述.....	9
6.2	人员批量下发.....	10
6.2.1	请求方式.....	10
6.2.2	测试地址.....	10
6.2.3	接口调用实例.....	10
6.2.4	请求参数.....	10
6.2.5	返回值.....	10
6.3	上传识别结果.....	11
6.3.1	请求方式.....	11
6.3.2	请求地址.....	11
6.3.3	接口调用实例.....	11
6.3.4	请求参数.....	11
6.3.5	返回值.....	12
6.4	删除人员信息.....	13
6.4.1	请求方式.....	13
6.4.2	请求地址.....	13
6.4.3	接口调用实例.....	13
6.4.4	请求参数.....	13
6.4.5	返回值.....	13
6.5	设备操作后反馈消息给服务端.....	14
6.5.1	请求方式.....	14
6.5.2	请求地址.....	14
6.5.3	接口调用实例.....	15

---

6.5.4	请求参数.....	15
6.5.5	返回值.....	15

---

## 1 范围

本规范适用于以接口方式，对接实名制注册和考勤设备，适合在成都市登记的所有设备厂商。

## 2 目的

为实现劳务人员信息、考勤信息的标准化采集和全市统一应用，制定本数据接口。

## 3 方案概述

本接口方案主要实现采集设备上传采集的劳务人员信息到实名制系统，以及考勤设备从实名制系统获取考勤人员信息、禁入人员信息，并上传考勤的数据，采集设备与系统通讯支持 https 协议，考勤设备与系统通讯支持 socket 和 https 协议。

## 4 采集设备 (https 协议)

### 4.1 接口描述

上传实名认证后的注册人员信息。

### 4.2 请求方式

Post 请求

### 4.3 测试地址

<https://118.122.92.139:6961/Regist.Asfx>

### 4.4 请求参数

详细数据, post 参数 (**form-data**) :

序号	参数	类型	描述
1	Idno	string	身份证号
2	Name	string	姓名
3	gender	int	性别 (1: 男, 2: 女)
4	nation	string	民族
5	birthday	string	出生日期 (yyyy-MM-dd)
6	address	string	住址
7	idissue	string	发证机关
8	idperiod	string	证件有效期, 格式:20010101-20110101
9	idphoto	string	身份证照片 (base64)
10	photo	string	现场人员可见光照片 (base64)
11	inf_photo	string	现场人员红外照片 (base64)
12	userType	int	注册人员类型 (1: 采集劳务人员, 2: 采集岗位人员)
13	dev_mac	string	设备序列号
14	RegType	int	注册类型 (3: 人脸采集)

### 4.5 返回值

返回值(application/json) :

序号	参数	类型	描述
1	errcode	int	成功返回 0, 失败或错误返回非 0 的值

2	errmsg	string	接口调用失败或错误时，返回相应的提示信息
---	--------	--------	----------------------

返回值示例：

```
{
  "errcode": 0,
  "errmsg": "",
}
```

#### 4.6 其它

错误编码说明：

序号	错误码	描述
1	0	成功
2	不等于 0	失败或错误

### 5 考勤设备（socket 协议）

#### 5.1 接口描述

- 1) 使用的通讯协议（Socket）
- 2) 为了保证传输过程中的数据安全，接口中的 JSON 格式的包体数据设定为使用 DES 进行加密。
- 3) 字符编码格式 UTF-8
- 4) 单包数据(包头+包体+包尾)小于 10K
- 5) 在连续发送多包数据时，第一包包体为数据，后续包中为二进制数据，不需要加密处理。
- 6) 测试地址 xxx.xxx.xx.xxx，端口:xxxxxx

#### 5.2 接口请求数据包格式

##### 1)、包头

请求包包头

序号	内容	长度	备注	其它
----	----	----	----	----

1	包头标记	2 byte	0xDDCC	
2	数据包长度	2 byte	0x0000~0x2800	最大 0x2800 (包头+包体+包尾)
3	命令字	1 byte	0x00~0xFF	最大 0xFF
4	分包序号	2 byte	0x0000~0xFFFF	0x0000
5	分包总数	2 byte	0x0000~0xFFFF	0x0000
6	设备状态	1 byte	0x00~0xFF	请参考设备状态对照表
7	设备 SN 编号	16byte		

### 应答包包头

序号	内容	长度	备注	其它
1	包头标记	2 byte	0xCCDD	
2	数据包长度	2 byte	0x0000~0x2800	最大 0x2800 (包头+包体+包尾)
3	命令字	1 byte	0x00~0xFF	最大 0xFF
4	分包序号	2 byte	0x0000~0xFFFF	0x0000
5	分包总数	2 byte	0x0000~0xFFFF	0x0000
6	执行结果返回 编码或后续命 令字	1 byte	0x00~0xFF	0x00: 服务器空闲(主要是心跳包使用) 0x01: 成功 0x02: 失败 0x10~0xFF: 为后续命令字
7	设备 SN 编号	16byte		

### 2)、包体

JSON 格式数据或二进制数据

### 3)、包尾

序号	内容	长度	备注	其它
1	数据校验	1 byte	0x00~0xFF	包头+包体 Xor 校验运算 U8

				<pre> calcChecksum(unsigned char *pkg, int len) {     U8 BBC = 0;     for(int i = 0; i &lt; len; i++)     {         BBC ^= pkg[i];     }     return BBC; } </pre>
--	--	--	--	---

### 5.3 接口详情

#### 5.3.1 心跳包

设备主动发送请求到服务器

(一) 请求(设备→服务器):

1. 请求命令: 0x10
2. 请求实例: 0xDDCC 包长 0x100x00000x00000x01SN16Byte 二进制数据校验
3. 请求包体数据格式: {"sn":"", "date\_time":""}

包体数据详细说明:

序号	参数	类型	说明
1	sn	string	设备序列号
2	date_time	datetime	设备请求时间, 格式为: 年-月-日 时:分:秒

(二) 返回示例(服务器→设备):



0: 空闲（无后续命令）

返回数据: 0xCCDD 包长 0x100x00000x00000x00SN16Byte 校验

无包体

1: 成功（查看包体中的详细信息-有后续命令）

返回数据: 0xCCDD 包长 0x100x00000x00000x01SN16Byte 二进制数据校验

返回包体数据格式: {"cmd":"","params":[{"date\_time":""}]}

包体详细说明:

序号	参数	说明
1	cmd	服务器端发送给需要设备执行的命令
2	params	服务器端发送给需要设备执行的命令的参数

2: 失败（查看包体中的详细信息-无后续命令）

返回数据: 0xCCDD 包长 0x100x00000x00000x02SN16Byte 二进制数据校验

包体数据格式: {"result\_code":"","result\_desc":""}

包体详细说明:

序号	参数	类型	说明
1	result_code	int	错误编码, 请参考错误编码说明
2	result_desc	string	描述

### 5.3.2 人员批量下发

根据心跳包返回数据到设备

(一) 返回结果（服务器->设备）:

1. 返回命令: 0x25

2. 返回数据: 0xCCDD 包长 0x250x00000x00000x01SN16Byte 二进制数据校验

3. 返回数据包体格式:

{ "count" : " 2" ,

  "users" : [{

    user\_id:'' , name:'' , work\_sn:'' , id\_card:'' , face\_template:'' } ]

}

包体数据详细说明:

序号	属性	类型	名称	说明
1	user_id	string	用户 ID	用户唯一 ID
2	name	string	姓名	
3	work_sn	string	用户工号	和 user_id 一样
4	id_card	string	身份证号	
5	face_template	string	人脸图片	Base64

(二) 下发结果返回(设备→服务器):

1. 返回命令: 0x25
2. 返回数据: 0xDDCC 包长 0x250x00000x00000x01SN16Byte 加密后的二进制数据校验
3. 返回数据格式: {"result\_code":"","result\_desc":""}

包体数据详细说明:

序号	参数	类型	说明
1	result_code	int	错误编码, 请参考错误编码说明
2	result_desc	string	描述

### 5.3.3 上传识别结果

设备识别后把识别结果上传到服务器(单包数据数量小于 5 条)

(一) 请求(设备→服务器):

1. 请求命令: 0x19
2. 请求数据: 0xDDCC 包长 0x190x00000x00000x01SN16Byte 二进制数据校验
3. 请求数据格式:

```
{  
  "count": 2,  
  "logs": [  
    {"sn": "XXXX", "user_id": "XXXXX", "recog_time": "yyyy-Mm-dd HH:mm:ss"},  
    {"sn": "XXXX", "user_id": "XXXXX", "recog_time": "yyyy-Mm-dd HH:mm:ss"}  
  ]  
}
```

包体数据详细说明:

序号	属性		类型	名称	说明
1	count		int	识别记录数量	
2	logs	sn	string	设备序列号	
3		user_id	string	用户 ID	用户唯一 ID
4		recog_time	datetime	识别时间	

(二) 上传结果返回 (服务器→设备) :

1. 返回命令: 0x19
2. 返回数据: 0xCCDD 包长 0x190x0000x0000x01SN16Byte 二进制数据校验
3. 返回包体数据格式: {"result\_code":"","result\_desc":""}

包体数据详细说明:

序号	参数	类型	说明
1	result_code	int	错误编码, 请参考错误编码说明
2	result_desc	string	描述

#### 5.3.4 删除人员信息

根据心跳包返回数据到设备

(一) 返回结果 (服务器→设备) :

1. 返回命令: 0x14
2. 返回数据: 0xCCDD 包长 0x140x0000x0000x01SN16Byte 二进制数据校验
3. 返回包体数据格式: {"type":"","user\_id":"","time":""}, type 默认为: 0。

包体规范数据详细说明:

序号	属性	类型	名称	说明
1	type	int	删除类型	默认值 0
2	user_id	string	用户工号	
3	time	datetime	时间	默认为空

(二) 删除结果返回 (设备→服务器):

1. 返回命令: 0x14
2. 返回数据: 0xDDCC 包长 0x140x0000x0000x01SN16Byte 加密后的二进制数据校验
3. 返回包体数据格式: {"result\_code":"","result\_desc":""}

包体数据详细说明:

序号	参数	类型	说明
1	result_code	int	错误编码, 请参考错误编码说明
2	result_desc	string	描述

## 5.4 其它

### 1、错误编码说明

序号	编码值	说明
1	0000	成功
2	9999	失败(无效的数据)
3	9998	未知错误, 请和开发人员联系
4	9001	数据格式错误[json 解析失败]
5	9021	设置时间失败

## 6 考勤设备 (https 协议)

### 6.1 接口描述

1) 使用的通讯协议 (https)

2) 请求返回结果

```
{  
  Result:      // 返回请求结果类型  
  Content:     // 返回详细内容, json 格式  
  Msg:        // 返回消息  
}
```

3) 为了保证传输过程中的数据安全, 返回结果中 Content 包体数据设定为使用 DES 进行加密。

4) 字符编码格式 UTF-8。

---

## 6.2 人员批量下发

### 6.2.1 请求方式

Get 请求

### 6.2.2 测试地址

https://118.122.92.139:6965/httpserver/Service/DevivePacketWebSvr.assx  
/GetAddPerson

### 6.2.3 接口调用实例

https://118.122.92.139:6965/httpserver/Service/DevivePacketWebSvr.assx/GetAddPerson?sn=设备序列号

### 6.2.4 请求参数

序号	参数	类型	说明
1	sn	string	设备序列号

### 6.2.5 返回值

格式:

```
{  
    Result:    // 返回请求类型  
    Content:   // 返回内容  
    Msg:       // 返回消息  
}
```

说明:

序号	参数	类型	说明
1	Result	int	返回类型: -2 (设备序列号不存在), -1 (服务器异常-失败), 0 (空)

			闲-成功)																														
2	Content	string	<p>返回内容: Content 里面的内容用 DES 加密 deskey:对接的时候提供加密前格式:</p> <pre>{   [user_id:'' , name:'' , work_sn:'' , id:'' , id_card:'' ,   face_template:'' ] }</pre> <p>数据说明:</p> <table border="1"> <thead> <tr> <th>序号</th> <th>属性</th> <th>类型</th> <th>名称</th> <th>说明</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>user_id</td> <td>string</td> <td>用户 ID</td> <td>用户唯一 ID</td> </tr> <tr> <td>2</td> <td>name</td> <td>string</td> <td>姓名</td> <td></td> </tr> <tr> <td>3</td> <td>work_sn</td> <td>string</td> <td>用户工号</td> <td>和 user_id 一样</td> </tr> <tr> <td>4</td> <td>id_card</td> <td>string</td> <td>身份证号</td> <td></td> </tr> <tr> <td>5</td> <td>face_template</td> <td>string</td> <td>人脸图片</td> <td>Base64</td> </tr> </tbody> </table>	序号	属性	类型	名称	说明	1	user_id	string	用户 ID	用户唯一 ID	2	name	string	姓名		3	work_sn	string	用户工号	和 user_id 一样	4	id_card	string	身份证号		5	face_template	string	人脸图片	Base64
序号	属性	类型	名称	说明																													
1	user_id	string	用户 ID	用户唯一 ID																													
2	name	string	姓名																														
3	work_sn	string	用户工号	和 user_id 一样																													
4	id_card	string	身份证号																														
5	face_template	string	人脸图片	Base64																													
3	Msg	string	返回消息																														

### 6.3 上传识别结果

#### 6.3.1 请求方式

Post 请求

#### 6.3.2 请求地址

https://118.122.92.139:6965/httpservice/Service/DevivePacketWebSvr.assx/UploadAttendance

#### 6.3.3 接口调用实例

https://118.122.92.139:6965/httpservice/Service/DevivePacketWebSvr.assx/UploadAttendance?sn=设备序列号& content=考勤数据(des 加密后)

#### 6.3.4 请求参数

序号	参数	类型	说明
1	sn	string	设备序列号

2	content	string	<p>请求内容：          json 数组 Content 里面的内容用 des 加密 deskey: 对接的时候提供          加密前格式：  <pre>{   "count": 2,   "logs": [     {"sn": "XXXX", "user_id": "XXXXX", "recog_time": "yyyy-MM-dd HH:mm:ss"},     {"sn": "XXXX", "user_id": "XXXXX", "recog_time": "yyyy-MM-dd HH:mm:ss"}   ] }</pre> </p> <p>数据说明：          Count 数量小于 5 条</p> <table border="1"> <thead> <tr> <th>序号</th> <th>属性</th> <th>类型</th> <th>名称</th> <th>说明</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>count</td> <td>int</td> <td>识别记录数量</td> <td></td> </tr> <tr> <td>2</td> <td rowspan="3">log</td> <td>sn</td> <td>设备序列号</td> <td></td> </tr> <tr> <td>3</td> <td>user_id</td> <td>用户 ID</td> <td>用户唯一 ID</td> </tr> <tr> <td>4</td> <td>recog_time</td> <td>识别时间</td> <td></td> </tr> </tbody> </table>	序号	属性	类型	名称	说明	1	count	int	识别记录数量		2	log	sn	设备序列号		3	user_id	用户 ID	用户唯一 ID	4	recog_time	识别时间	
序号	属性	类型	名称	说明																						
1	count	int	识别记录数量																							
2	log	sn	设备序列号																							
3		user_id	用户 ID	用户唯一 ID																						
4		recog_time	识别时间																							

### 6.3.5 返回值

格式：

```
{
  Result: // 返回请求类型
  Content: // 返回内容
  Msg: // 返回消息
}
```

说明:

序号	参数	类型	说明
1	Result	int	返回类型: -3 (考勤数据为空), -2 (设备序列号不存在), -1 (服务器异常-失败), 0 (空闲-成功)
2	Content	string	返回内容: 空
3	Msg	string	返回消息

## 6.4 删除人员信息

### 6.4.1 请求方式

Get 请求

### 6.4.2 请求地址

https://118.122.92.139:6965/httpservice/Service/DevivePacketWebSvr.assx/GetDelPerson

### 6.4.3 接口调用实例

https://118.122.92.139:6965/httpservice/Service/DevivePacketWebSvr.assx/GetDelPerson?sn=设备序列号

### 6.4.4 请求参数

序号	参数	类型	说明
1	sn	string	设备序列号

### 6.4.5 返回值

格式:

```
{  
    Result:    // 返回请求类型  
    Content:  // 返回内容  
    Msg:      // 返回消息
```



}

说明:

序号	参数	类型	说明																				
1	Result	int	返回类型: -2 (设备序列号不存在), -1 (服务器异常-失败), 0 (空闲-成功)																				
2	Content	string	返回内容: Content 里面的内容用 DES 加密 deskey: 对接的时候提供 加密前格式: { [type:' 0', user_id:' ', time:' ' ] } 数据说明: <table border="1"><thead><tr><th>序号</th><th>属性</th><th>类型</th><th>名称</th><th>说明</th></tr></thead><tbody><tr><td>1</td><td>type</td><td>int</td><td>类型</td><td>默认: 0</td></tr><tr><td>2</td><td>user_id</td><td>string</td><td>用户 ID</td><td>用户唯一 ID</td></tr><tr><td>3</td><td>time</td><td>datetime</td><td>时间</td><td>当前时间</td></tr></tbody></table>	序号	属性	类型	名称	说明	1	type	int	类型	默认: 0	2	user_id	string	用户 ID	用户唯一 ID	3	time	datetime	时间	当前时间
序号	属性	类型	名称	说明																			
1	type	int	类型	默认: 0																			
2	user_id	string	用户 ID	用户唯一 ID																			
3	time	datetime	时间	当前时间																			
3	Msg	string	返回消息																				

## 6.5 设备操作后反馈消息给服务端

考勤设备在执行人员批量下发和删除人员操作后, 均需调用该接口与服务器之间进行确认。

### 6.5.1 请求方式

Post 请求

### 6.5.2 请求地址

<https://118.122.92.139:6965/httpserver/Service/DevivePacketWebSvr.assx/FeedBack>

### 6.5.3 接口调用实例

https://118.122.92.139:6965/httpserver/Service/DevivePacketWebSvr.assx/FeedBack?  
type=反馈类型&sn=设备序列号& msg=反馈消息

### 6.5.4 请求参数

序号	参数	类型	说明
1	type	int	反馈类型： 0-人员删除成功， 1-人员删除失败， 2-人员下发成功， 3-人员下发失败
2	sn	string	设备序列号
3	msg	string	反馈消息内容

### 6.5.5 返回值

格式：

```
{  
    Result:    // 返回请求类型  
    Content:   // 返回内容  
    Msg:       // 返回消息  
}
```

说明：

序号	参数	类型	说明
1	Result	int	返回类型：-3（反馈类型不能为空），-2（设备序列号不存在），-1（服务器异常-失败），0（空闲-成功）
2	Content	string	返回内容：空
3	Msg	string	返回消息