

# BDS CLOUD SERVER CONFIGURATION MANUAL

---

BDS-8W, BDS-8THA, BDS-8FHA

# BDS CLOUD SERVER CONFIG

---

24/7 TECHNICAL SUPPORT AT 1.877.877.2269 OR VISIT [BLACKBOX.COM](http://BLACKBOX.COM)



**TABLE OF CONTENTS**

<b>CONTENTS</b>	
1. DYNAMIC DETECTION DISPLAY CALLBACK INTERFACE V1.1 .....	4
2. DYNAMIC DETECTION DISPLAY LAN INTERFACE V1.4 .....	6
2.1 INTERFACE SPECIFICATION .....	6
2.2 SET DEVICE PASSWORD .....	6
2.3 PARAMETER CONFIGURATION (FACE RECOGNITION, OPENING MODE, ETC.) .....	7
2.4 GET PARAMETER CONFIGURATION (FACE RECOGNITION, SWITCHING MODE, ETC.) .....	9
2.5 PERSONNEL REGISTRATION .....	10
2.6 PERSONNEL INFORMATION EDITING .....	11
2.7 DELETE PERSON .....	12
2.8 IDENTIFICATION RECORD ACQUISITION .....	13
2.9 MODIFY LOGO .....	14
2.10 ACQUISITION OF EQUIPMENT MAC ADDRESS .....	15
2.11 PARAMETER CONFIGURATION (BODY TEMPERATURE, MASK, FAN, ETC.) .....	15
2.12 ACCESS TO PHOTOS .....	17
2.13 DEVICE PARAMETER INFORMATION ACQUISITION .....	18
2.14 TAKING PICTURES .....	18
2.15 DEVICE INITIALIZATION .....	19
2.16 SYSTEM TIME SETTING .....	19
2.17 IDENTIFY THE RECORDING INTERFACE CALLBACK SETTINGS .....	20
2.18 DEVICE DOOR CONTROL .....	20
2.19 GET BODY TEMPERATURE AND MASK PARAMETERS .....	21
2.20 PERSONNEL CHECKUP .....	22
2.21 PERSONNEL PAGE QUERY .....	23
3. DYNAMIC DETECTION DISPLAY COMMUNICATION PROTOCOL USAGE .....	25
3.1 PROTOCOL DESCRIPTION .....	25
3.1.1 INTERFACE SPECIFICATION .....	25
3.1.2 PROTOCOL FIELD DESCRIPTION .....	25
3.1.3 PROTOCOL REPLY INSTRUCTIONS .....	25
3.2 PROTOCOL LIST .....	26
3.2.1 DEVICE REGISTRATION .....	26
3.2.2 HEARTBEAT .....	26
3.2.3 PARAMETER CONFIGURATION .....	27
3.2.4 PERSONNEL REGISTRATION .....	29
3.2.5 PERSONNEL INFORMATION EDITING .....	30
3.2.6 DELETE PERSONNEL .....	31
3.2.7 IDENTIFICATION RECORD UPLOAD ADDRESS SETTING .....	31
3.2.8 MODIFY LOGO .....	32
3.2.9 BODY TEMPERATURE AND MASK PARAMETER CONFIGURATION .....	32
3.2.10 DEVICE INITIALIZATION .....	33
3.2.11 VERSION UPGRADE .....	33



# TABLE OF CONTENTS

3.2.12 REMOTE DOOR OPENING .....	34
3.2.13 REMOTE RESTART .....	34
3.2.14 TEST RESULTS OF PERSONNEL PHOTOS .....	34
3.2.15 INQUIRY OF PERSONNEL INFORMATION .....	36
3.2.16 QUERY PERSONNEL INFORMATION RESULTS.....	36
3.2.17 QUERY PERSONNEL INFORMATION .....	37
3.2.18 PAGINATION QUERY RESULTS OF PERSONNEL INFORMATION.....	38
3.2.19 PARAMETER CONFIGURATION QUERY.....	39
3.2.20 PARAMETER CONFIGURATION QUERY RESULTS .....	39
3.2.21 BODY TEMPERATURE PARAMETER CONFIGURATION QUERY .....	40
3.2.22 QUERY RESULTS OF BODY TEMPERATURE PARAMETER CONFIGURATION.....	40
3.2.23 SHUTDOWN.....	40
3.2.24 APPLICATION GUARD.....	41
3.2.25 POWER ON .....	41
<b>DISCLAIMER .....</b>	<b>42</b>
<b>TRADEMARKS USED IN THIS MANUAL .....</b>	<b>42</b>

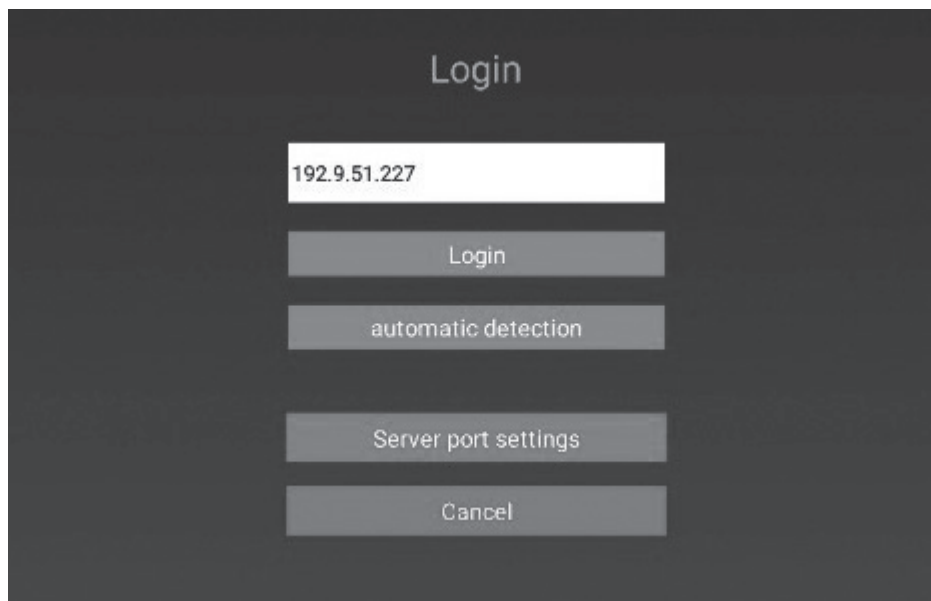


# CHAPTER 1: DYNAMIC DETECTION DISP. CALLBACK INT. V1.1

## 1. DYNAMIC DETECTION DISPLAY CALLBACK INTERFACE V1.1

Instructions: This interface is a description of the interface that the client APP pushes to other backgrounds. We recommend that you first use postman and other tools to test and verify.

After the interface development is completed, you can set it in the APP application settings, then enter the callback address to save and use.



### INTERFACE SPECIFICATION

Interface	uploadMipsGateRecord
address	url address
mode	POST
required parameter	JSON
return type	JSON

### FACE DETECTION RESULT DATA PUSH INTERFACE

REQUIRED PARAMETER	PARAMETER	PARAMETER MODE	REQUEST OR NOT	EXPLANATION
device ID	mac	string	Y	MAC address format, 12 bits
userId	userId	string	Y	Person ID, -1 means stranger
checkTime	checkTime	long	Y	Timestamp
name	name	string	N	Personnel name
type	type	int	N	Identity type, 1 is visitor, 2 is blacklist, 3 for employees, 4 for ID card identification

## CHAPTER 1: DYNAMIC DETECTION DISP. CALLBACK INT. V1.1

## FACE DETECTION RESULT DATA PUSH INTERFACE (CONTINUED)

REQUIRED PARAMETER	PARAMETER	PARAMETER MODE	REQUEST OR NOT	EXPLANATION
cardNo	cardNo	string	N	Card number information, stranger is empty
idCardNo	idCardNo	string	N	ID card number
temperature	temperature	string	N	temperature, e.g., 36.5
mask	mask	int	N	mask, 1-wear, 0-no wear
Pic infos	checkPic	string	N	Snap pictures, the picture format is Base64
extra	extra	string	N	Extra information, extended use
ID card information	sex	String	N	
ID CARD INFO	nation	String	N	
ID CARD INFO: birthdate	birthdate	String	N	
ID CARD INFO: address	address	String	N	
ID CARD INFO: photo	cardPhoto	String	N	From base64

## RETURN VALUE

REQUIRED PARAMETER	PARAMETER	EXPLANATION
Return result	code	0-success, 1-fail
Return information	msg	Return information

## Example:

```

{
  "mac": "8CFCA0039257",
  "userId": "",
  "checkTime": 1585361367292,
  "name": "test",
  "type": -1,
  "cardNo": "",
  "idCardNo": "440...",
  "temperature": "36.6",
  "mask": -1,
  "checkPic": "",
  "extra": "",
  "sex": "",
  "nation": "",
  "birthdate": "1986-9-2",
  "address": "...",
  "cardPhoto": ""
}return example
{
  "code": 0,
  "msg": "Push successfully"
}

```

## CHAPTER 2: DYNAMIC DETECTION DISPLAY LAN INT. V1.4

### 2. DYNAMIC DETECTION DISPLAY LAN INTERFACE V1.4

#### 2.1 INTERFACE SPECIFICATION

- Interface root address: http://device ip address: 8080/
- Interface form: provide external services through HTTP requests
- Interface security: The device password (pass) needs to be set first when calling.
- The interface for the first time, and the device password (pass) needs to be passed in as the interface security verification key for any subsequent calls to the interface.

#### Interface returns

Interface general return instructions:

```
public class ResultInfo <T> {
```

```
private Int result; // Indicates whether the interface is tuned up, 1: succeeds, 0: fails, usually as long as the device server can respond, the value is 1
```

```
private Boolean success; // Whether the operation is successful, the success is true, the failure is false private T data; // The business data returned by the interface, the type can be numeric, string or collection, etc
```

Interface list

#### 2.2 SET DEVICE PASSWORD

Request method: POST request

Request address: http://device IP:8080/setPassWord

#### DATE

FIELD	DATE	REQUEST	DESCRIPTION
oldPass ID	String	Y	Old password
newPass	String	Y	New password

Request description: For new devices or devices after resetting (restoring initialization), you need to set the initial password before calling other interfaces. OldPass and newPass can be passed the same value. When changing the password, paste in the new and old password separately.

#### Example:

```
{
  "data": "passWord:q", // The device password, also called the interface calling password, please enter it properly. If you forget your password, you need to reset the device, and the device will erase all data.
  "result": 1, // Interface tuning
  "success": true // Device password set successfully
}
```



## CHAPTER 2: DYNAMIC DETECTION DISPLAY LAN INT. V1.4

### 2.3 PARAMETER CONFIGURATION (FACE RECOGNITION, OPENING MODE, ETC.)

Request method: POST request

Request address: http://device IP: 8080/setConfig

#### DATE

FIELD	MODE	REQUEST	DESCRIPTION
pass	String	Y	Password setting
config	Json	Y	Parameter configuration Json The following are the parameters of json

#### Request description

POST request parameters are placed in the body

The config request parameter is in json format, an example follows (adjust according to the actual situation).

```
{
  "companyName": "smdt",
  "deviceId": 86,
  "displayCustom": "{name}",
  "displayMode": 1,
  "id": 1,
  "liveIdentiLevel": 1,
  "passType": false,
  "password": "123456",
  "recoDistance": "1.5",
  "recoInterval": "2000",
  "relayDelay": 5,
  "relayMode": 0,
  "serialCustomize": "#{idcardNum}#",
  "serialMode": 2,
  "similarity": 80,
  "strangerVoiceCustom": "moshengren",
  "strangerVoiceMode": 2,
  "voiceCustom": "name",
  "voiceMode": 1,
  "wg": 0
}
```

**CHAPTER 2: DYNAMIC DETECTION DISPLAY LAN INT. V1.4****FIELD**

FIELD	MODE	DESCRIPTION
companyName	String	companyName
deviceId	int	deviceId
displayCustom	String	displayCustom default as {name}
displayMode	int	displayMode, 1: name, 100: custom
id	int	
liveIdentiLevel	int	Living body recognition level, default fast mode, 0 no living body, 1 fast mode cannot reject photos, 2 can reject some photos, 3 can reject photos and adapt
passType	boolean	Access type false into true out
password	String	passcode
recoDistance	String	Recognition distance (0.5-3)
recoInterval	String	Recognition interval 2000-10000
relayDelay	Int	Relay automatically closes after a delay of x seconds (5-63)
Relay mode	Int	<p>relayMode</p> <p>0: indicates the automatic closing mode, that is, X seconds will be delayed after the relay is opened.</p> <p>Dynamic closure</p> <p>1: It means that it does not close automatically, that is, it will not close automatically after opening the relay.</p> <p>Fit</p>
serialCustomize	String	Serial port definition, default is "#{idcardNum}#", Unlimited length
similarity	Int	Similarity, default 80, 30-100 adjustment
strangerVoiceCustom	String	Stranger voice customization, default for stranger recognition, within 32 characters
strangerVoiceMode	Int	Stranger voice mode, the default stranger alarm, 1 does not require voice broadcast, 2 Stranger alarm, 100 custom
voiceCustom	String	Voice customization, default is "name", within 32 characters
voiceMode	Int	Voice mode, 1 no broadcast, 2 broadcast name, 100 custom
wg	Int	Wiegand 26 or 34 output mode, 0: output card number WG26, 1: output person Staff ID WG26, 2: output card number WG34, 3: output staff ID WG34
icNO	int	IC Card, 1: Card Swipe
idCardNo	int	ID Card 1: Close
tricolorLamp	int	Red light, 1: Off, 2: On



## CHAPTER 2: DYNAMIC DETECTION DISPLAY LAN INT. V1.4

Return data

```

{
  "data": { // Set parameters
    "companyName": "smdt",
    "deviceId": 86,
    "displayCustom": "{name}",
    "displayMode": 1,
    "id": 1,
    "liveIdentiLevel": 1,
    "passType": false,
    "password": "123456",
    "recoDistance": "1.5",
    "recoInterval": "2000",
    "relayDelay": 5,
    "relayMode": 0,
    "serialCustomize": "#{idcardNum}#",
    "serialMode": 2,
    "similarity": 80,
    "strangerVoiceCustom": "moshengren",
    "strangerVoiceMode": 2,
    "voiceCustom": "name",
    "voiceMode": 1,
    "wg": 0
  },
  "result": 1, // Interface access
  "success": true // Setting successful
}

```

### 2.4 GET PARAMETER CONFIGURATION (FACE RECOGNITION, SWITCHING MODE, ETC.)

Request method: POST request

Request address: http://device IP:8080/getConfig

#### REQUEST DATE

FIELD	MODE	REQUEST	DESCRIPTION
pass	String	Y	Device password

## CHAPTER 2: DYNAMIC DETECTION DISPLAY LAN INT. V1.4

Return Example:

```
{
  "data": { // Set parameters
    "companyName": "smdt",
    "deviceId": 86,
    "displayCustom": "{name}",
    "displayMode": 1,
    "id": 1,
    "liveIdentiLevel": 1,
    "passType": false,
    "password": "123456",
    "recoDistance": "1.5",
    "recoInterval": "2000",
    "relayDelay": 5,
    "relayMode": 0,
    "serialCustomize": "#{idcardNum}#",
    "serialMode": 2,
    "similarity": 80,
    "strangerVoiceCustom": "moshengren",
    "strangerVoiceMode": 2,
    "voiceCustom": "name",
    "voiceMode": 1,
    "wg": 0
  },
  "result": 1, // Interface access
  "success": true // Setting successful
}
```

### 2.5 PERSONNEL REGISTRATION

Request method: POST request

Request address: http://device ip:8080/person/create

#### REQUEST DATE

FIELD	MODE	REQUEST	DESCRIPTION
pass	String	Y	Device password
person	Json	Y	Personnel information Json The following are the parameters of json



## CHAPTER 2: DYNAMIC DETECTION DISPLAY LAN INT. V1.4

Request description

POST request parameters are placed in the body

The base64 string of the photo, without the header, such as: data: image/jpg; base64. Staff photos only support JPG and PNG.

### PERSON FIELD

FIELD	MODE	DESCRIPTION
age	int	age
imgBase64	String	imgBase64 String
name	String	name
prescription	String yyyy-MM-dd HH:mm,yyyy-MM-dd HH:mm	
sex	int	sex
type	Int	Type The default is 1, 1 for visitors, 2 blacklists, and 3 for employees.
vipID	Int	Create a person's id, find the person's editor based on the id
welcome	String	Reserved, can be left blank
icCard	String	ic card information
card	String	ID CARD no
wn	String	Reserved, can be left blank

RETURN Example:

```
{
  "data": "msg: Successful registration ",
  "result": 1,
  "success": true
}
```

Registration failed, return failure description, the corresponding error code is as follows.

### 2.6 PERSONNEL INFORMATION EDITING

Request method: POST request

Request address: http://device ip:8080/person/update

### REQUEST DATE

FIELD	MODE	REQUEST	DESCRIPTION
pass	String	Y	Device password
person	Json	Y	Personnel information Json The following are the parameters of json

## CHAPTER 2: DYNAMIC DETECTION DISPLAY LAN INT. V1.4

request

POST request parameters are placed in the body

The base64 string of the photo, without the header, such as

Staff photos only support JPG and PNG

### PERSON PARAMETER DESCRIPTION

FIELD	MODE	DESCRIPTION
age	int	age
imgBase64	String	imgBase64 String
name	String	Name
prescription	String	yyyy-MM-dd HH:mm,yyyy-MM-dd HH:mm
sex	int	sex
type	int	Type The default is 1, 1 for visitors, 2 blacklists, and 3 for employees.
vipID	Int	Create a person's id, find the person's editor based on the id
welcome	String	Reserved, can be left blank
icCard	String	ic card information
card	String	ID number
wn	String	Reserved, can be left blank

Example of returned data:

```
{  
  "data": "msg: Staff editing success ",  
  "result": 1,  
  "success": true  
}
```

### 2.7 DELETE PERSON

Request method: POST request

Request address: http://device ip:8080/person/delete



## CHAPTER 2: DYNAMIC DETECTION DISPLAY LAN INT. V1.4

## REQUEST DATE

FIELD	MODE	REQUEST	DESCRIPTION
pass	String	Y	Device password
deleteld	String	Y	Delete person id

Return Example:

```
{
  "data": "msg: Person deleted successfully
  "result": 1,
  "success": true
}
```

## 2.8 IDENTIFICATION RECORD ACQUISITION

Request method: POST request

Request address: http://device ip:8080/newFindRecords

## REQUEST DATE

FIELD	MODE	REQUEST	DESCRIPTION
pass	String	Y	Device password
startTime	String	Y	start: 2019-07-11 12:00
endTime	String	Y	finish: 2019-07-11 17:00

```
{
  "data": [
    {
      "birthDate": "", //reserved
      "currentTime": 1562819094334, // Transit time, milliseconds
      "idCardNum": "", // staff ID
      "imageFlag": 0, // The name of the photo taken by the public, if you want to get the photo, please call getRecordImg interface to get it.
      "name": "pdf", //passer name
      "type": 1, //mode
      "temperature": "36.5", // Body temperature when passing, transmitted when body temperature detection is off -1
      "mask": 1 // Whether to wear a mask: 1 wear, 0: no wear mask, -1 when mask detection is closed},
    {
      "birthDate": "",
      "currentTime": 1562824109687,
      "idCardNum": "",
      "imageFlag": 0,
      "imageName": "1_86_1562824109687.jpg",
    }
  ]
}
```

## CHAPTER 2: DYNAMIC DETECTION DISPLAY LAN INT. V1.4

```

    "name": "pdf",
    "type": 1
    "temperature": "36.5",
    "mask": 1 // Whether to wear a mask, 1: worn, 0: not worn
  },
  {
    "birthDate": "",
    "currentTime": 1562825856546,
    "idCardNum": "null",
    "imageFlag": 1,
    "imageName": "100_86_1562825856546.jpg",
    "name": "tts",
    "type": 1,
    "temperature": "36.5",
    "mask": 1 // Whether to wear a mask: 1 wearing, 0 not wearing
    "birthDate": "",
    "currentTime": 1562825856546,
    "idCardNum": "null",
    "imageFlag": 1,
    "imageName": "100-86-1562825856546.jpg",
    "name": "tts",
    "type" :1,
    "temperature": "36.5",
    "mask": 1 //Whether to wear a mask: 1 wearing; 0 not wearing
  }
],
"result": 1, // Interface transfer through
"success": true // success

```

## 2.9 MODIFY LOGO

Request method: POST request

Request address: http://device ip:8080/changeLogo

## REQUEST DATE

FIELD	MODE	REQUEST	DESCRIPTION
pass	String	Y	device
imgBase64	String	Y	Logo pic Base64, only supports png, jpg

## CHAPTER 2: DYNAMIC DETECTION DISPLAY LAN INT. V1.4

Request:

POST request parameters are placed in the body

The base64 string of the photo, without the header, such as: data:image/jpg;base65

Staff photos only support JPG and PNG

Example:

```
{
  "data": "msg: logo modify successfully",
  "result": 1,
  "success": true
}
```

### 2.10 ACQUISITION OF EQUIPMENT MAC ADDRESS

Request method: POST request

Request address: http://device ip:8080/getDeviceMac

#### REQUEST DATE

FIELD	MODE	REQUEST	DESCRIPTION
pass	String	Y	Device password

Example:

```
{
  "data": "8CFCA0000023",
  "result": 1,
  "success": true
}
```

### 2.11 PARAMETER CONFIGURATION (BODY TEMPERATURE, MASK, FAN, ETC.)

Request method: POST request

Request address: http://device ip:8080/tempAndMaskSetting

#### REQUEST DATE

FIELD	MODE	REQUEST	DESCRIPTION
pass	String	Y	Device password
config	Json	Y	Personnel information Json. The following are the parameters of json

## CHAPTER 2: DYNAMIC DETECTION DISPLAY LAN INT. V1.4

### Request description

POST request parameters are placed in the body.

The config request parameter is in json format; an example follows (adjust according to the actual situation).

```
{
  "isBodyTempAlarm": 1,
  "isBodyTempStart": 1,
  "isHighFeverAdopt": 0,
  "isLowFeverAdopt": 0,
  "isLowTempAdopt": 0,
  "isStandardTempAdopt": 1,
  "isWearingMask": 0,
  "standardBodyTemp": "37.3",
  "isStrangerRecord":0,
  "isFan":0
}
```

### config description

#### REQUEST DATE

FIELD	MODE	REQUEST
isBodyTempAlarm	int	1: Turn on body temperature alarm; 0: Turn off body temperature alarm
isBodyTempStart	Int	1: Turn on body temperature detection; 0: Turn off body temperature detection
standardBodyTemp	String	Body temperature threshold 37.3 (accurate to one decimal place), open body temperature detection and body temperature alarm. After detecting that the body temperature exceeds the threshold, a sound alarm is played.
isHighFeverAdopt	int	Whether high fever passed (0: not passed; 1: passed) is used to open the door (range 38.5–43.0)
isLowFeverAdopt	int	Whether high fever passed (0: not passed 1: passed) is used to open the door (range 38.5–43.0)
isLowTempAdopt	int	Whether the low temperature is passed (0: not passed; 1: passed) for door opening (range 30.0 and below)
isStandardTempAdopt	int	Whether the normal body temperature is passed (0: not passed 1: passed) for opening the door (range 36.1–37.2)
isWearingMask	int	Wear mask detection switch (0: No; 1: Yes)
isStrangerRecord	int	Stranger identification record storage switch (0: No; 1: Yes)
isFan	int	Fan switch (0: No; 1: Yes)
tempCompensation	float	Temperature compensation, range (-1–1), positive number means positive compensation, negative number means negative compensation

## CHAPTER 2: DYNAMIC DETECTION DISPLAY LAN INT. V1.4

Example:

```
{
  "data": {
    "deviceId": 88,
    "id": 10,
    "isBodyTempAlarm": 1,
    "isBodyTempStart": 1,
    "isHighFeverAdopt": 0,
    "isLowFeverAdopt": 0,
    "isLowTempAdopt": 0,
    "isStandardTempAdopt": 1,
    "isWearingMask": 0,
    "standardBodyTemp": "37.3",
    "isStrangerRecord": 0,
    "isFan": 0
  },
  "result": 1,
  "success": true
}
```

## 2.12 ACCESS TO PHOTOS

Request method: POST request

Request address: http://device ip: 8080/getRecordImg

## REQUEST DATA

FIELD	MODE	REQUEST	DESCRIPTION
pass	String	Y	Device
imgName	String	Y	Photo name (obtained by the identification record interface)

Example:

```
{
  "data": "/9j/4AAQSkZJRgABAQAAQABAAD/2wB", // Image's base64 string
  "result": 1, // 1 Interface call 0 Interface abnormal
  "success": true // true Get success false Get failure;
}
```

## CHAPTER 2: DYNAMIC DETECTION DISPLAY LAN INT. V1.4

### 2.13 DEVICE PARAMETER INFORMATION ACQUISITION

Request method: POST request

Request address: http://device ip: 8080/getDeviceInfo

Request data: /getDeviceInfo

Request data:

#### REQUEST DATA

FIELD	MODE	REQUEST	DESCRIPTION
pass	String	Y	Device password

Example:

```
{
  "data": {"freeSpace": "3.61GB", // freeSpace: Remaining storage space
  "ip": "192.9.51.45", // ip: ip address
  "mac": "8CFCA0036225", // mac address
  "time": 1584518232928, // System time (ms) long type
  "version": "1.5.0.22.0.01" // app Current version number
},
  "result": 1,
  "success": true
}
```

### 2.14 TAKING PICTURES

Request method: POST request

Request address: http://equipment ip: 8080/photograph

#### REQUEST DATA

FIELD	MODE	REQUEST	DESCRIPTION
pass	String	Y	Device password

Example:

```
{
  "data": "/9j/4AAQSkZJRgABAQAA ...", // base64 String, need to be converted to suffix png pictures
  "result": 1,
  "success": true
}
```



## CHAPTER 2: DYNAMIC DETECTION DISPLAY LAN INT. V1.4

### 2.15 DEVICE INITIALIZATION

Request method: POST request

Request address: http://device ip: 8080/initialization

Request data:

#### REQUEST DATA

FIELD	MODE	REQUEST	DESCRIPTION
pass	String	Y	Device password

Request description:

- Delete all identification records, personnel data, characteristics, and other data on the device, and clear all databases.
- Delete the attributes set through the device configuration interface.

Example:

```
{
  "data": " Initialization successful ",
  "result": 1,
  "success": true
}
```

### 2.16 SYSTEM TIME SETTING

Request method: POST request

Request address: http://device ip: 8080/setDeviceTime

#### REQUEST DATA

FIELD	MODE	REQUEST	DESCRIPTION
pass	String	Y	Device password
year	String	Y	year
month	String	Y	month
day	String	Y	day
hours	String	Y	hours
minute	String	Y	minute

Request description:

After successful configuration, the device time is changed to the currently set time. If the device is connected to the public network, the device itself has a network time calibration mechanism, and the system adjusts the device time to the public network time. For the device to display the time manually set, the device must be in the local area network. If connected to the public network, the device uses the public network time by default when refreshing its time.

## CHAPTER 2: DYNAMIC DETECTION DISPLAY LAN INT. V1.4

Example:

```
{
  "data": "2020-03-15 22:10:50", // Device current time
  "result": 1,
  "success": true
}
```

### 2.17 IDENTIFY THE RECORDING INTERFACE CALLBACK SETTINGS

Request method: POST request

Request address: http://device ip: 8080/setIdentifyCallback

#### REQUEST DATA

FIELD	MODE	REQUEST	DESCRIPTION
pass	String	Y	Device password
callbackUrl	String	Y	Identify the url interface for record upload

Request description:

The callback can be set in other settings of the APP terminal settings.

Example:

```
{
  "data": "http://192.9.51.45:8080/setIdentifyCallback",
  "result": 1, //1 Set successfully -1 No callbackUrl field -2 callbackUrl Field is empty
  "success": true //true Set successfully, false set failed
}
```

### 2.18 DEVICE DOOR CONTROL

Request method: POST request

Request address: http://device ip: 8080/device/openDoorControl

#### REQUEST DATA

FIELD	MODE	REQUEST	DESCRIPTION
pass	String	Y	Device password



**CHAPTER 2: DYNAMIC DETECTION DISPLAY LAN INT. V1.4**

Example:

```
{
  "data": "open the door success",
  "result": 1,
  "success": true
}
```

**Device restart**

Request method: POST request

Request address: http://device ip: 8080/restartDevice

**REQUEST DATA**

FIELD	MODE	REQUEST	DESCRIPTION
pass	String	Y	Device password

Example:

```
{
  "data": " Restart the success ",
  "result": 1,
  "success": true
}
```

**2.19 GET BODY TEMPERATURE AND MASK PARAMETERS**

Request method: POST request

Request address: http://device ip: 8080/getTempAndMaskSetting

**REQUEST DATA**

FIELD	MODE	REQUEST	DESCRIPTION
pass	String	Y	Device password
config	Json	Y	Parameter configuration Json. The following are the parameters of json

## CHAPTER 2: DYNAMIC DETECTION DISPLAY LAN INT. V1.4

Return data (for example, refer to the description of temperature and mask parameter configuration for details):

```
{
  "data": {
    "isBodyTempAlarm ":1,"
    isBodyTempStart ":1,"
    isHighFeverAdopt ":0,"
    isLowFeverAdopt ":0,"
    isLowTempAdopt ":0,"
    isStandardTempAdopt ":1,"
    isStrangerRecord ":0,"
    isWearingMask ":0,"
    standardBodyTemp ":37.3 ","
    tempCompensation ":0.3,"
    tempCompensationParam ":1
  },
  "result": 1,
  "success": true
}
```

### 2.20 PERSONNEL CHECKUP

Request method: POST request

Request address: <http://device ip:8080/person/find>

#### REQUEST DATA

FIELD	MODE	REQUEST	DESCRIPTION
pass	String	Y	Device password
config	Json	Y	Person ID specifies that multiple persons to be queried are separated by commas. For example, "100,101,102"; the maximum number is 50.



## CHAPTER 2: DYNAMIC DETECTION DISPLAY LAN INT. V1.4

Return Data:

Except for the facelD field, refer to interface 4 for the detailed field description.

```

{
  "data": "[
    {
      "prescription": "2020-03-21 00:00,2030-12-13 00:00",
      "age": 0,
      "card": "",
      "facelD": 57358,
      "icCard": "",
      "name": "Lee",
      "sex": 1,
      "type": 3,
      "vipID": 57358 //Less than 0 means that the picture verification failed, greater than or equal to 0 means success, the reason for
      failure. Refer to the interface 4, Instructions for returning personnel registration results.
    }
  ],
  "result": 1,
  "success": true
}

```

### 2.21 PERSONNEL PAGE QUERY

Request method: POST request

Request address: <http://device ip:8080/person/findByPage>

#### REQUEST DATA

FIELD	MODE	REQUEST	DESCRIPTION
pass	String	Y	Device password
index	int	Y	Page number, starting from 0
length	Int	Y	Maximum number per page, <= 50, if > 50, only 50 can be found

## CHAPTER 2: DYNAMIC DETECTION DISPLAY LAN INT. V1.4

Return Data:

Except for the `faceID` field, refer to interface 4 for the detailed field description.

```
{
  "data": "{
    "pageInfo": {
      "index": 0,
      "length": 1,
      "size": 1,
      "total": 2
    },
    "records": [
      {
        "age": 0,
        "card": "",
        "faceID": 66614,
        "icCard": "",
        "name": "fff",
        "sex": 0,
        "type": 3,
        "prescription": "2020-03-21 00:00,2030-12-13 00:00",
        "vipID": 66614 //Less than 0 means that the picture verification failed, greater than or equal to 0 means success,
        the reason for failure. Please refer to the interface 4, Instructions for returning personnel registration results.
      }
    ]
  }",
  "result": 1,
  "success": true
}
```



## CHAPTER 3: COMMUNICATION PROTOCOL USAGE

### 3. DYNAMIC DETECTION DISPLAY COMMUNICATION PROTOCOL USAGE

#### 3.1 PROTOCOL DESCRIPTION

##### 3.1.1 INTERFACE SPECIFICATION

---

- Use the netty framework to establish socket connections to maintain communication and long connections
- Provide services by means of general requests

##### 3.1.2 PROTOCOL FIELD DESCRIPTION

---

General instructions for using the interface:

```
public class Command {  
    private String method;//Method name  
    private Long timestamp;//Timestamp for this operation  
    private Object body;//The business data of the protocol method, can be numeric type, string or set, etc.}
```

For the interface return examples mentioned in the documentation, the return data of individual interfaces will be slightly adjusted, and the actual return result will prevail.

##### 3.1.3 PROTOCOL REPLY INSTRUCTIONS

---

To ensure that the protocol is communicated correctly, define the following reply instructions:

The reply field includes the "method" and "timestamp" issued by the protocol, and the "body" field contains "mac" and whether it is successfully marked.

```
"body": {  
    "mac": " AADDD1254HH ", //mac address  
    "success": true //Whether the operation was successful, success is true, failure is false}
```

Examples of replies are as follows:

```
{  
    "method": "setConfig", //Reply to received "method"  
    "timestamp": 1584518232928,  
    "body": {  
        "mac": " AADDD1254HH ", //mac address  
        "success": true //Whether the operation was successful, success is true, failure is false}
```

## CHAPTER 3: COMMUNICATION PROTOCOL USAGE

### 3.2 PROTOCOL LIST

#### 3.2.1 DEVICE REGISTRATION

---

Method: deviceReg

Method: client->server

Sample Data

```
{
  "method": "deviceReg",
  "timestamp": 1584518232928,
  "body": {
    "ip": "192.9.51.45",
    "mac": "8CFCA0036225",
    "time": 1584518232928,
    "version": "1.5.0.22.0.01"
  }
}
```

#### 3.2.2 HEARTBEAT

---

Method: heartBeat

Method: client->server

Description: send heartbeat every 10 s

Sample data:

```
{
  "body": {
    "ip": "192.9.51.214",
    "mac": "8CFCA0036138",
    "time": 1585207755586,
    "version": "1.6.0.7"
  },
  "method": "heartBeat",
  "timestamp": 1585207755588
}
```



## CHAPTER 3: COMMUNICATION PROTOCOL USAGE

### 3.2.3 PARAMETER CONFIGURATION

---

Method: setConfig

Method: server -> client

Data Description

“body” request parameter is in json format, an example follows (adjust according to the actual situation)

```
{
  "method": "setConfig",
  "timestamp": 1584518232928,
  "body": {
    "tricolorLamp": 2,
    "idCardNo": 2,
    "icNo": 2, "deviceId": 86,
    "displayCustom":
    "{name}", "displayMode":
    1,
    "id": 1,
    "liveIdentiLevel": 1,
    "passType": false,
    "password": "123456",
    "recoDistance": "1.5",
    "recoInterval": "2000",
    "relayDelay": 5,
    "relayMode": 0,
    "serialCustomize": "#{idcardNum}#",
    "serialMode": 2,
    "similarity": 80,
    "strangerVoiceCustom": "moshengren",
    "strangerVoiceMode": 2,
    "voiceCustom": "name",
    "voiceMode": 1,
    "wg": 0
  }
}
```

## CHAPTER 3: COMMUNICATION PROTOCOL USAGE

### FIELD DESCRIPTION

FIELD	TYPE	DESCRIPTION
companyName	String	Company Name
deviceId	int	Associated equipment (id)
displayCustom	String	Display customization, the default is {name}
displayMode	int	Display mode, 1: display name, 100: custom
id	int	
livelidentiLevel	int	Living body recognition level, default fast mode, 0: no living body, 1: fast mode cannot reject photos, 2: can reject some photos, 3: can reject photos and adapt
passType	boolean	Access Type: false: in, true:out
password	String	Passcode
recoDistance	String	Recognition distance (0.5–3)
recoInterval	String	Recognition interval (2000–10000)
relayDelay	Int	Relay automatically closes after a delay of x seconds (5–63)
relayMode	Int	Relay mode: 0: indicates the automatic closing mode, that is, it will automatically close after a delay of X seconds after opening the relay; 1: indicates that the mode is not automatically closed, that is, it will not automatically close after opening the relay
serialCustomize	String	Serial output definition, the default is "# {idcardNum} #", unlimited length
serialMode	int	Serial port mode: 1: Open the door, 2: No output, 3: Output personnel ID, 4: Output ID/IC card number, 100: Custom output
similarity	Int	Similarity, default: 80, 30–100: adjustment
strangerVoiceCustom	String	Stranger voice customization, default for stranger recognition, within 32 characters
strangerVoiceMode	Int	Stranger voice mode, the default stranger alarm, 1: does not require voice broadcast, 2: stranger alarm, 100: custom
voiceCustom	String	Voice customization, default is "name", within 32 characters
voiceMode	Int	Voice mode, 1: no broadcast, 2: broadcast name, 100: custom
wg	Int	Wiegand 26 or 34 output mode, 0: output card number WG26, 1: output personnel ID WG26, 2: output card number WG34, 3: output personnel ID WG34
icNO	int	IC Card: 1: Card Swipe 2: Human and card comparison
idCardNo	int	ID Card: 1: Close, 2: Human and card comparison, 3: Human and card comparison (Visitor mode)
tricolorLamp	int	Red light, 1: Off; 2: On



## CHAPTER 3: COMMUNICATION PROTOCOL USAGE

### 3.2.4 PERSONNEL REGISTRATION

Method: person/create

Method: server -> client

#### Interface Description

- ◆ Staff photos only support JPG and PNG
- ◆ Photo address only supports pictures stored in http protocol
- ◆ Can be placed in "body" as batches, in the form of an array

#### FIELD DESCRIPTION

FIELD	TYPE	DESCRIPTION
age	int	Age
imgUrl	String	Photo download address
name	String	Name
prescription	String	Pass period (start time and end time separated by comma) yyyy-MM-dd HH:mm,yyyy-MM-dd HH:mm
sex	int	sex
type	int	Types: The default is 1, 1 as visitors, 2 as blacklists, and 3 as employees
vipID	int	Create a person's id, find the person's editor based on the id
welcome	String	Reserved, can be left blank
icCard	String	ic card information
card	String	Id number
wn	String	Reserved, can be left blank

```
{
  "method": "person/create",
  "timestamp": 1585207755588,
  "body": [{
    "age": 40,
    "name": "hyc",
    "imgUrl": "http://.....png",
    "prescription": "2020-03-21 00:00,2030-12-13 00:00",
    "sex": 1,
    "type": 1,
    "vipID": 1,
    "welCome": "helloworld"
  ]
}
```

## CHAPTER 3: COMMUNICATION PROTOCOL USAGE

### 3.2.5 PERSONNEL INFORMATION EDITING

Method: person/update

Method: server -> client

Interface Description:

Staff photos only support JPG and PNG

Photo address only supports pictures stored in http protocol

#### FIELD DESCRIPTION

FIELD	TYPE	DESCRIPTION
age	int	Age
imgUrl	String	Photo download address
name	String	Name
prescription	String	Pass period (start time and end time separated by comma) yyyy-MM-dd HH:mm,yyyy-MM-dd HH:mm
sex	int	sex
type	int	Types: The default is 1, 1 as visitors, 2 as blacklists, and 3 as employees
vipID	int	Create a person's id, find the person's editor based on the id
welcome	String	Reserved, can be left blank
icCard	String	ic card information
card	String	Id number
wn	String	Reserved, can be left blank

```
{
  "method": "person/ update",
  "timestamp": 1585207755588,
  "body": [{
    "age": 40,
    "name": "hyc1",
    "imgUrl": "http://.....png",
    "prescription": "2020-03-21 00:00,2030-12-13 00:00",
    "sex": 1,
    "type": 1,
    "vipID": 1,
    "welCome": "helloworld"
  ]
}
```



## CHAPTER 3: COMMUNICATION PROTOCOL USAGE

### 3.2.6 DELETE PERSONNEL

---

Method: person/delete

Method: server -> client

#### DATA

FIELD	TYPE	REQUIRED	DESCRIPTION
deleted	String	Y	Deleted person id, multiple persons are separated by (,)

```
{  
  "method": "person/delete",  
  "timestamp": 1584518232928,  
  "body": {  
    "deleteld": "1,2"  
  }  
}
```

### 3.2.7 IDENTIFICATION RECORD UPLOAD ADDRESS SETTING

---

Method: setIdentifyCallback

Method: server -> client

#### DATA

FIELD	TYPE	REQUIRED	DESCRIPTION
url	String	Y	Identify the callback address and implement field analysis

```
{  
  "method": "setIdentifyCallback",  
  "timestamp": 1585212192898,  
  "body": {  
    "callbackUrl": ""  
  }  
}
```

## CHAPTER 3: COMMUNICATION PROTOCOL USAGE

### 3.2.8 MODIFY LOGO

Method: changeLogo

Method: server -> client

#### DATA

FIELD	TYPE	REQUIRED	DESCRIPTION
logoUrl	String	Y	Logo icon download address, only supports png, jpg

Change to url download, deprecate base64

Data description: Staff photos only support JPG and PNG

```
{
  "method": "changeLogo",
  "timestamp": 1585212192898,
  "body": {
    "logoUrl": ""
  }
}
```

### 3.2.9 BODY TEMPERATURE AND MASK PARAMETER CONFIGURATION

Method: tempAndMaskSetting

Method: server -> client

The protocol parameters are in json format, examples follow (adjust according to the actual situation).

```
{
  "method": "tempAndMaskSetting",
  "timestamp": 1584518232928,
  "body": {
    "isBodyTempAlarm": 1,
    "isBodyTempStart": 1,
    "isHighFeverAdopt": 0,
    "isLowFeverAdopt": 0,
    "isLowTempAdopt": 0,
    "isStandardTempAdopt": 1,
    "isWearingMask": 0,
    "standardBodyTemp": "37.3",
    "isStrangerRecord": 0,
    "tempCompensation": 0.3,
    "isFan": 0
  }
}
```



## CHAPTER 3: COMMUNICATION PROTOCOL USAGE

## CONFIG FIELD DESCRIPTION

FIELD	TYPE	DESCRIPTION
isBodyTempAlarm	int	1: Turn on the body temperature alarm 0: Turn off body temperature alarm
isBodyTempStart	int	1: Turn on body temperature detection 0: Turn off body temperature detection
standardBodyTemp	String	Body temperature threshold 37.3 (accurate to one decimal place), after opening the body temperature detection and body temperature alarm, it detects that the body temperature exceeds the threshold and plays a sound alarm.
isHighFeverAdopt	int	Whether high fever passes (0: not passed 1: passed) is used to open the door (range 38.5–43.0)
isLowFeverAdopt	int	Whether low fever passes (0: not passed 1: passed) for opening the door (range 37.3–38.5)
isLowTempAdopt	int	Whether the low temperature is passes (0: not passed 1: passed) for door opening (range 30.0-below)
isStandardTempAdopt	int	Whether the normal body temperature is passed (0: not passed 1: passed) for opening the door (range 36.1–37.2)
isWearingMask	int	Wear mask detection switch (0: No; 1: Yes)
isStrangerRecord	int	Stranger identification record storage switch (0: No, 1: Yes)
isFan	int	Fan switch (0: No, 1: Yes)
tempCompensation	float	Temperature compensation, range (-1–1), positive number means upward compensation, negative number downward compensation

## 3.2.10 DEVICE INITIALIZATION

Method: initialization

Method: server -> client

Description:

Delete all identification records, personnel data, characteristics, and other data on the device, and clear all databases.

Delete the attributes set through the device configuration interface.

## 3.2.11 VERSION UPGRADE

Method: updateApp

Method: server -> client

## DATA

FIELD	TYPE	REQUIRED	DESCRIPTION
apkUrl	String	Y	Apk download address
md5	String	Y	Apk file md5 value

## CHAPTER 3: COMMUNICATION PROTOCOL USAGE

```
{  
  "method": "updateApp",  
  "timestamp": 1584518232928,  
  "body": {  
    "apkUrl": "",  
    "md5": ""  
  }  
}
```

### 3.2.12 REMOTE DOOR OPENING

---

Method: remoteOpendoor

Method: server -> client

### 3.2.13 REMOTE RESTART

---

Method: reboot

Method: server -> client

### 3.2.14 TEST RESULTS OF PERSONNEL PHOTOS

---

Method: person/picVerify

Method: client -> server

### DATA

FIELD	TYPE	DESCRIPTION
vipID	Int	Create a person's id, find the person's editor based on the id
mac	String	Terminal mac address
result	Int	Photo verification result, 0 means success, <0 means failure, see the verification instructions below for details
message	String	Wrong description



## CHAPTER 3: COMMUNICATION PROTOCOL USAGE

### PICTURE VERIFICATION FAILURE DESCRIPTION

FIELD	TYPE	DESCRIPTION
0	Int	Picture verification successful
-1	int	Failed to get image bitmap
-2	Int	SDK extraction feature tool anomaly
-3	int	The set vipId parameter is less than 0
-4	int	Image conversion failed
-5	int	The resolution of the picture exceeds the requirement 4096 * 2688
-6	int	SDK Face Detector Abnormal
-7	int	No face detected
-8	int	Face detection failed
-9	int	Add face operation failed
-10	int	Maximum number of faces exceeded 20000
-11	int	Face size is smaller than the preset pixel threshold
-12	int	Face score is too low
-13	int	Face tilt angle exceeds threshold
-14	int	Face horizontal angle exceeds threshold
-15	int	Face pitch angle exceeds threshold
-201	int	Image download link is invalid

Return example:

```
{
  "method": "person/picVerify",
  "timestamp": 1584518232928,
  "body": {
    "message": "Successfully added to face library,"
    "mac": "8CFCA0038A46",
    "result": 0,
    "vipID": 1
  }
}
```

## CHAPTER 3: COMMUNICATION PROTOCOL USAGE

### 3.2.15 INQUIRY OF PERSONNEL INFORMATION

---

Method: person/find

Method: server -> client

#### DATA

FIELD	TYPE	REQUIRED	DESCRIPTION
vipld	String	Y	Person ID specifies that multiple persons to be queried are separated by commas. For example, "100,101,102", the maximum number is 50.

Find instructions according to vipld:

```
{
  "body": {
    "vipld": "100,101,102"
  },
  "method": "person/find",
  "timestamp": 1585212192898
}
```

### 3.2.16 QUERY PERSONNEL INFORMATION RESULTS

---

Method: person/find/result

Method: client -> server

Return data description:

In addition to the mac and facelD fields, refer to Interface 4 Personnel Registration for other detailed field descriptions.



## CHAPTER 3: COMMUNICATION PROTOCOL USAGE

Example:

```
{
  "body": {
    "mac": " AADDD1254HH ", //mac address
    "data": [{
      "age": 40,
      "name": "hyc",
      "prescription": "2020-03-21 00:00,2030-12-13 00:00",
      "sex": 1,
      "type": 1,
      "vipID": 1,
      "faceID": 1, //If it is less than 0, it means that the picture inspection failed, and if it is greater than or equal
      to 0, it means success. For the reason of failure, please refer to the interface 14 personnel test result return
      instructions.
      "icCard": "", //ic card no.
      "card": "", //id no.
    }]
  },
  "method": "person/find/ result",
  "timestamp": 1585207755588,
}
```

### 3.2.17 QUERY PERSONNEL INFORMATION

Method: person/findByPage

Method: server -> client

#### DATA

FIELD	TYPE	REQUIRED	DESCRIPTION
index	int	Y	Page number, starting from 0
length	Int	Y	Maximum number per page, <= 50, if> 50, only 50 can be found

Pagination search instructions:

```
{
  "body": {
    "index": 0,
    "length": 10
  },
  "method": "person/findByPage",
  "timestamp": 1585212192898
}
```

## CHAPTER 3: COMMUNICATION PROTOCOL USAGE

### 3.2.18 PAGINATION QUERY RESULTS OF PERSONNEL INFORMATION

---

Method: person/findByPage/result

Method: client -> server

Data description:

In addition to the mac, faceID, and pageInfo fields, refer to Interface 4 for the detailed field descriptions.

Return Example:

```
{
  "body": {
    "data": [
      {
        "name": "bbb",
        "prescription": "2020-03-21 00:00,2030-12-13 00:00",
        "age": 0,
        "card": "",
        "faceID": 57367, //If it is less than 0, it means that the picture inspection failed, and if it is greater than or equal
        //to 0, it means success. For the reason of failure, please refer to the interface 14 personnel test result return
        //instructions.
        "icCard": "",
        "sex": 1,
        "type": 3,
        "vipID": 57367
      }
    ],
    "mac": "8CFCA0064F9C",
    "pageInfo": {
      "index": 2,
      "length": 5,
      "size": 5,
      "total": 79
    }
  }
  //Page number
  //Number per page
  //Number of searches
  //Total number of equipment personnel
},
"method": "person/findByPage/result",
"timestamp": 1586403020783
}
```



## CHAPTER 3: COMMUNICATION PROTOCOL USAGE

### 3.2.19 PARAMETER CONFIGURATION QUERY

---

Method: getConfig

Method: server -> client

### 3.2.20 PARAMETER CONFIGURATION QUERY RESULTS

---

Method: getConfigresult

Method: client -> server

Return data description:

In addition to the mac field, refer to the interface 3 parameter configuration for other detailed field descriptions.

Return data:

```
{
  "body": {
    "tricolorLamp": 1,
    "idCardNo": 2,
    "icNO": 2,
    "companyName": "smdt",
    "deviceId": 86,
    "displayCustom": "{name}",
    "displayMode": 1,
    "id": 1,
    "liveIdentiLevel": 1,
    "mac": " AADDD1254HH ",//Terminal mac address
    "passType": false,
    "password": "123456",
    "recoDistance": "1.5",
    "recoInterval": "2000",
    "relayDelay": 5,
    "relayMode": 0,
    "serialCustomize": "#{idcardNum}#",
    "serialMode": 2,
    "similarity": 80,
    "strangerVoiceCustom": "moshengren",
    "strangerVoiceMode": 2,
    "voiceCustom": "name",
    "voiceMode": 1,
    "wg": 0
  },
  "method": "getConfig/result ",
  "timestamp": 1584518232928
}
```

## CHAPTER 3: COMMUNICATION PROTOCOL USAGE

### 3.2.21 BODY TEMPERATURE PARAMETER CONFIGURATION QUERY

---

Method: getTempConfig

Method: server -> client

### 3.2.22 QUERY RESULTS OF BODY TEMPERATURE PARAMETER CONFIGURATION

---

Method: getTempConfig/result

Method: client -> server

Return data description:

In addition to the mac field, refer to Interface 9 Body Temperature Parameter Settings for other detailed field descriptions.

Return data:

```
{
  "body": {
    "tempCompensation": 0.1,
    "isBodyTempAlarm": 1, //1: Turn on body temperature alarm,
                        0: Turn off body temperature alarm
    "isBodyTempStart": 1, //1: Turn on body temperature detection
                       0: Turn off body temperature detection
    "isFan": 0, //1: Turn on the fan, 0: Turn off the fan
    "isHighFeverAdopt": 0,
    "isLowFeverAdopt": 0,
    "isLowTempAdopt": 0,
    "isStandardTempAdopt": 1,
    "isStrangerRecord": 0, //1: Open stranger record, 0: close
    "isWearingMask": 0,
    "mac": "8CFCA0064F9C", //Terminal mac address
    "standardBodyTemp": "37.3" //Alarm threshold
  },
  "method": "getTempConfig/result",
  "timestamp": 1586336014026
}
```

### 3.2.23 SHUTDOWN

---

Method: shutdown

Method: server -> client



## CHAPTER 3: COMMUNICATION PROTOCOL USAGE

### 3.2.24 APPLICATION GUARD

---

Method: application/guardian

Method: server -> client

#### DATA

FIELD	TYPE	REQUIRED	DESCRIPTION
onoff	int	Y	1: on; 0: off

Data description:

```
{  
  "body": {  
    "onoff": 1 //1: On 0: Off  
  },  
  "method": "application/guardian",  
  "timestamp": 1585212192898  
}
```

### 3.2.25 POWER ON

---

Method: application/boot

Method: server -> client

#### DATA

FIELD	TYPE	REQUIRED	DESCRIPTION
onoff	int	Y	1: on; 0: off

Data Description:

```
{  
  "body": {  
    "onoff": 1 //1: On 0: Off  
  },  
  "method": "application/boot",  
  "timestamp": 1585212192898  
}
```

## DISCLAIMER/TRADEMARKS

### DISCLAIMER

Black Box Corporation shall not be liable for damages of any kind, including, but not limited to, punitive, consequential or cost of cover damages, resulting from any errors in the product information or specifications set forth in this document and Black Box Corporation may revise this document at any time without notice.

### TRADEMARKS USED IN THIS MANUAL

Black Box and the Black Box logo type and mark are registered trademarks of Black Box Corporation.

Any other trademarks mentioned in this manual are acknowledged to be the property of the trademark owners.



**NOTES**

NEED HELP?  
LEAVE THE TECH TO US

**LIVE 24/7  
TECHNICAL  
SUPPORT**

1.877.877.2269



**NEED HELP?  
LEAVE THE TECH TO US**

---

**LIVE 24/7  
TECHNICAL  
SUPPORT**

---

**1.877.877.2269**

