Intelligent Building Product Line	Document No.		Version	2.4	Confidential Level	Commercial Secret A

QR-RD-023 (Ver2.4)

Access Control API Guide

Prepared by:	Qian Sihao	**Date:**	November 15, 2023
Checked by:		**Date:**	
Reviewed by:		**Date:**	
Approved by:		**Date:**	

Document Revision History

No.	Revision Description	Author	Version	Date	Approved by
1	First release.	—	V1.0	December 21, 2020	
2	Modified some content.	Qian Sihao	V2.0	March 7, 2022	
3	Modified some content.	Qian Sihao	V2.1	May 21, 2022	
4	Added new content: Auto registration, event auto push, intercom, advertising function configuration, and more.	Liu Yuyao	V2.2	November 15, 2023	
5	Added new content: searching for the capabilities of second-generation access control and modifying part of the text format.	Qian Sihao	V2.3	May 24, 2024	
6	Added new content: CGI attendance export function, fingerprint collection function, partial face parameter configuration interface.	Qian Sihao	V2.4	Feb 28, 2025	

Table of Contents

1 Introduction

This document describes access control related CGI protocols. Compared with the standard Dahua CGI protocol, this document focuses on the description of access control related functions and corresponding CGI protocol formats.

2 Application

This document is applicable to ASI8 series, ASI7 series, ASI6 series, ASI4 series, ASI3 series and other face access control products.

3 Protocol Definition

3.1 Abbreviations

The following abbreviation is used throughout this document.

API	API (Application Programming Interface) in this document refers to the HTTP protocol-based programming development interface for accessing security devices.

3.2 Syntax Description

- When describing API parameters in the URL syntax, the italic text in angle brackets indicates that the actual content should be replaced with the corresponding value or string together with the angle brackets. For example, the <*server*> in the URL should be replaced with the IP or domain name of the server, such as 192.168.1.108.
- The content in the square brackets is optional, for example: "http://<*server*>/cgi-bin/snapshot.cgi[?channel=1]" is equivalent to "http://<*server*>/cgi-bin/snapshot.cgi".
- The API syntax should follow the URI standard (RFC 3986: Uniform Resource Identifiers (URI) Generic Syntax). That is, the spaces and other reserved characters (such as ":", "/", "?", "@", ";", "=", "+", ",", "$", "&") in the name-value key-value pair should be replaced with the %<ASCII hex> format. For example, the spaces should be replaced with %20.
- Symbols such as "[]" and "{}" should be used to describe a variable range. For example: "[0–100]" represents an integer that is not less than 0 and not more than 100. "{0, 1, 2, 3}" represents an integer within the range of 0, 1, 2, and 3.

- Adding "[]" after a string indicates an array, with a subscript starting from 0. For example, "Snap[*channel*]" represents "Snap[0]", "Snap[1]", and so on.
- There are different types of variables: Variables such as string, integer, bool or float. Integer means a 32-bit integer, and the Boolean value is "true" or "false".
- The "R/O" in a parameter indicates whether this parameter is required; "R" means required and "O" means optional.

3.3 API Request Format

The format of HTTP API request is as follows:

<**protocol**> ://<**server**><**abs_path**> [?**query**]

- **Protocol**: URL scheme of request. This document supports http and https protocols, so the http used in most APIs (except some RTSP APIs that use rtsp) in this document can be replaced with https.
- **Server**: Server address port information, with the format of "hostname [: port]". hostname can be the IP address or the domain name of the device. The parameter port is the service port for device listening. The default port is used if no port is provided. For the HTTP protocol, the default port is 80; for the HTTPS protocol, the default port is 443.
- **Abs_path**: The resource path of the request command. The format of resources in this protocol specification is usually as follows: "/cgi-bin/*.cgi".
- **Query**: Request parameter, which usually consists of name-value key-value pairs: p1=v1&p2=v2&…&pn=vn, for example: http://192.168.1.108/cgi-bin/snapshot.cgi?channel=1.

3.4 Format of the Server Response

The server uses the standard HTTP return format.

Return format:

HTTP/1.1 <HTTP code> <HTTP text>\r\n		
HTTP Code	**HTTP Text**	**Description**
200	OK	Requested successfully, with the response result data in the HTTP body.
400	Bad Request	The request parameter format is incorrect.
401	Unauthorized	User authentication information is not provided.
403	Forbidden	The user has no permission to perform the request operation.
404	Not Found	The requested content does not exist.

500	Internal Server Error	The request cannot be processed because an error occurred in the server.
501	Not Implemented	The server does not implement the request.

If the returned HTTP status code is 200, it means that the API request command is executed successfully. Therefore, the returned information in the HTTP body might be single response or multiport response. The format of each response can be multiple lines of key=value data, or a json value string, or a separate line of "OK".

Example: success respose with multiline key=value

```
HTTP/1.1 200 OK
Server: xxx
Content-Length: <length>

status.Focus=0.5
status.Zoom=0.5
…
```

Example: success respose with a word "OK"

```
HTTP/1.1 200 OK
Server: xxx
Content-Length: <length>

OK
```

If the returned HTTP status code is not 200, it means that the API request command failed. Therefore, the HTTP body might be null or two lines of error information; the first line is "Error", which means that an error occurred, and the second line of string describes the error details.

Example: request does not fit with syntax.

```
HTTP/1.1 404 Not Found
Server: xxxx
```

Example: Request spells wrong.

```
HTTP/1.1 400 Bad Request
Server: xxx
Content-Length: <length>

Error
Bad Request!
```

Example: If the request fits with syntax but an error occurs while the server handles it, the response would like this:

```
HTTP/1.1 500 Internal Server Error
Server: xxx
Content-Length: <length>

Error
Internal Server Error!
```

3.5 User Authentication

The device supports HTTP digest authentication. For details, see RFC 2617. If the HTTP request sent by the client does not contain the header information of "Authorization", the device will return the HTTP status code 401 and the corresponding authentication parameters. Then, the client calculates the authentication information according to the requirements in RFC 2617, and resends a request that contains the header information of "Authorization". The device does not execute the request to return the corresponding result information unless the authentication information is correct.

For example:

When the HTTP digest authentication failed, the following information is returned:

```
HTTP/1.1 401 Unauthorized
WWW-Authenticate: Digest realm="DH_00408CA5EA04",
nonce="000562fdY631973ef04f77a3ede7c1832ff48720ef95ad", stale=FALSE,
qop="auth"
```

The client calculates the digest authorization using information like username, password, nonce, HTTP method and URI with MD5, and then sends it to server.

The client uses MD5 to calculate information such as username, password, nonce, HTTP method, and URI in accordance with RFC 2617, and then resends a request to the device. See the following example of the header information of "Authorization":

```
Authorization: Digest username="admin", realm="DH_00408CA5EA04",
nc=00000001, cnonce="0a4f113b", qop="auth",
nonce="000562fdY631973ef04f77a3ede7c1832ff48720ef95ad", uri="/cgi-
bin/magicBox.cgi?action=getLanguageCaps",
response="65002de02df697e946b750590b44f8bf"
```

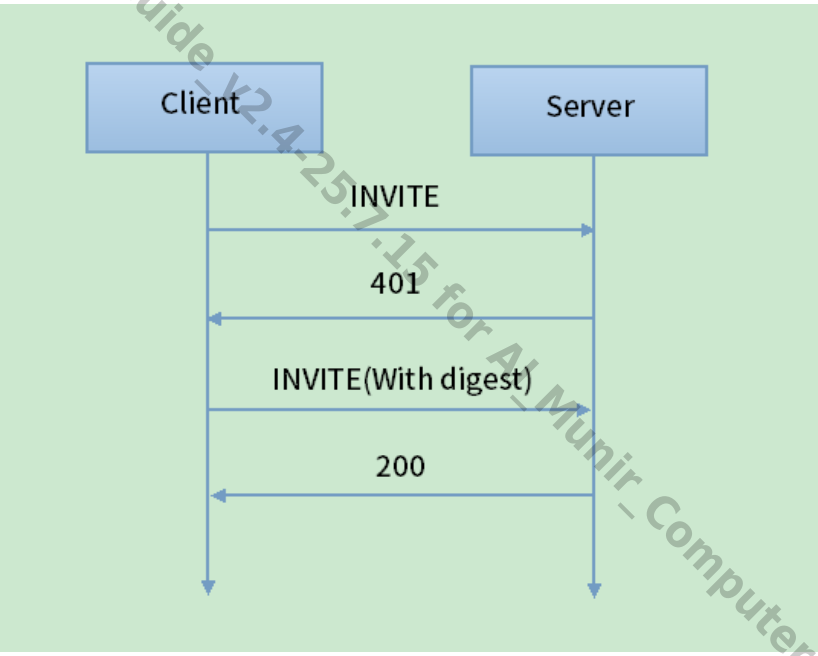
4 Description of Common Functions in Access Control

4.1 Device Authentication

Digest authentication is a simple authentication mechanism primarily developed for HTTP protocol, therefore, it is also called HTTP digest in RFC2617. It uses hash encryption as the identity authentication mechanism to avoid transmitting the user's password in clear text.

4.1.1 Digest Authentication Flow

Figure 4-1 Digest authentication flow



4.1.2 Authentication Description

User Authentication interaction is carried out when the CGI command is sent. The process is as follows:

First Interaction:

Figure 4-2 **First interaction**

```
GET /cgi-bin/configManager.cgi?action=getConfig&name=Encode HTTP/1.1
Accept: application/x-ms-application, image/jpeg, application/xaml+xml, image/
gif, image/pjpeg, application/x-ms-xbap, application/x-shockwave-flash,
application/vnd.ms-excel, application/vnd.ms-powerpoint, application/msword, */
*
Accept-Language: zh-CN
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.1; WOW64; Trident/
4.0; SLCC2; .NET CLR 2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media
Center PC 6.0; .NET4.0C; .NET4.0E; InfoPath.3)
Accept-Encoding: gzip, deflate
Host: 172.23.19.72
Connection: Keep-Alive
Cookie: username=admin; DhWebVideoPath1=C%3A%5CUsers%5C46987%5CWebDownload
%5CLiveSnapshot; DhWebVideoPath2=C%3A%5CUsers%5C46987%5CWebDownload
%5CLiveRecord; DhWebVideoPath3=C%3A%5CUsers%5C46987%5CWebDownload
%5CPlaybackSnapshot; DhWebVideoPath4=C%3A%5CUsers%5C46987%5CWebDownload
%5CPlaybackRecord; DhWebVideoPath5=C%3A%5CUsers%5C46987%5CWebDownload
%5CVideoClips; DhWebVideoPath6=C%3A%5CUsers%5C46987%5CWebDownload%5CHeatMap;
DhWebVideoPath7=C%3A%5CUsers%5C46987%5CWebDownload%5CAroudWifiSearch;
DhWebVideoPath8=C%3A%5CUsers%5C46987%5CWebDownload%5CPlaybackDownload; secure

HTTP/1.1 401 Unauthorized
WWW-Authenticate: Digest realm="Login to 172be7f93d6b9b161c8a6c43c5d3248d",
qop="auth", nonce="25105827", opaque="87505e79dd7b67a4ffa687a62a50ad2a1aa2041b"
Connection: close
Set-Cookie:secure; HttpOnly
CONTENT-LENGTH: 0
```

- For first-time request, you must send the correct URL of the CGI.
- For first-time response, fields such as realm, qop, nonce and opaque in WWW-Authenticate: Digest should be attached in the second request.

Second Interaction

Figure 4-3 **Second interaction**

```
GET /cgi-bin/configManager.cgi?action=getConfig&name=Encode HTTP/1.1
Accept: application/x-ms-application, image/jpeg, application/xaml+xml,
image/gif, image/pjpeg, application/x-ms-xbap, application/x-shockwave-flash,
application/vnd.ms-excel, application/vnd.ms-powerpoint, application/msword,
*/*
Accept-Language: zh-CN
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.1; WOW64;
Trident/4.0; SLCC2; .NET CLR 2.0.50727; .NET CLR 3.5.30729; .NET CLR
3.0.30729; Media Center PC 6.0; .NET4.0C; .NET4.0E; InfoPath.3)
Accept-Encoding: gzip, deflate
Host: 172.23.19.72
Connection: Keep-Alive
Authorization: Digest username="admin",realm="Login to
172be7f93d6b9b161c8a6c43c5d3248d",nonce="25105827",uri="/cgi-bin/
configManager.cgi?
action=getConfig&name=Encode",cnonce="5408ed2fbaad00018b02804ee53c27db",nc=00
000001,response="752fab2820c0dd90193d00d50d3a7732",qop="auth",opaque="87505e7
9dd7b67a4ffa687a62a50ad2a1aa2041b"
Cookie: secure; username=admin; DhWebVideoPath1=C%3A%5CUsers
%5C46987%5CWebDownload%5CLiveSnapshot; DhWebVideoPath2=C%3A%5CUsers
%5C46987%5CWebDownload%5CLiveRecord; DhWebVideoPath3=C%3A%5CUsers
%5C46987%5CWebDownload%5CPlaybackSnapshot; DhWebVideoPath4=C%3A%5CUsers
%5C46987%5CWebDownload%5CPlaybackRecord; DhWebVideoPath5=C%3A%5CUsers
%5C46987%5CWebDownload%5CVideoClips; DhWebVideoPath6=C%3A%5CUsers
%5C46987%5CWebDownload%5CHeatMap; DhWebVideoPath7=C%3A%5CUsers
%5C46987%5CWebDownload%5CAroudWifiSearch; DhWebVideoPath8=C%3A%5CUsers
%5C46987%5CWebDownload%5CPlaybackDownload; secure

HTTP/1.1 200 OK
X-XSS-Protection: 1;mode=block
X-Frame-Options: SAMEORIGIN
X-Content-Type-Options: nosniff
Content-Security-Policy: script-src 'self' 'unsafe-inline' 'unsafe-eval'
Strict-Transport-Security: max-age=604800; includeSubDomains
```

Second Request

- **Authorization**: Username, realm, nonce, uri, cnonce, qop, nc, response and opaque shouldbe attached following Authorization: Digest.

- **username**: Login username.
- **realm**: Consistent with the realm field in the first response.
- **nonce**: Consistent with the nonce field in the first response.
- **opaque**: Consistent with the opaque field in the first response.
- **URI**: Request URL
- **Cnonce**: The client nonce, which is an opaque string value provided by the client and used by both the client and server to avoid using clear text. This allows both parties to verify the identity of the other and provides some protection for the integrity of the message.
- **Qop**: It is auth by default.
- **Nc**: The nonce counter is a hexadecimal value that indicates the number of requests sent by the client under the same nonce. For example, in the first request of the response, the client will send "nc=00000001". The purpose of this indicator is for the server to keep a copy of this counter to identify duplicate requests.
- response: A string calculated by the user agent software to prove that the user knows the password. For example, it can be calculated through the username, realm, http method, uri, nonce, nc, qop, cnonce and password in encryption. The server will carry out the same calculation through the local password, and then compare the results with the response.

Encryption format:

sha256/md5(sha256/md5(username + ':' + realm + ':' + password) + ':' + nonce + ':' + nc + ':' + cnonce + ":auth: + sha256/md5(HTTPMethod + url))

The specific encryption format will be different depending on the product (sha256, md5 or other encryption algorithms), and it is implemented in the manager library. For details, refer to the product.

After obtaining the response filed through the above formula, the client will fill the value in the response and then send it to the server. The server will also carry out the calculation through the same formula, and then compare the result with the response. For details on the encryption process, see **RFC2617**.

4.2 Device Management

4.2.1 Getting the Device Type

Description: Get the device type

URL:http://<server>/cgi-bin/magicBox.cgi?action=getDeviceType

For details, see Protocols.

4.2.2 Getting the Hardware Version Information

Description: Get the hardware version number of the device.

URL: http://<server>/cgi-bin/magicBox.cgi?action=getHardwareVersion

For details, see Protocols.

4.2.3 Getting the Device Serial Number

URL: http://<server>/cgi-bin/magicBox.cgi?action=getSerialNo

For details, see Protocols.

4.2.4 Getting the Device Name

URL: http://<server>/cgi-bin/magicBox.cgi?action=getMachineName

For details, see Protocols.

4.2.5 Getting the Device System Information

URL: http://<server>/cgi-bin/magicBox.cgi?action=getSystemInfoNew

For details, see Protocols.

4.2.6 Getting the Software Version Information

URL: http://<server>/cgi-bin/magicBox.cgi?action=getSoftwareVersion

For details, see Protocols.

4.2.7 Restoring to the Factory Settings

URL: http://<server>/cgi-bin/magicBox.cgi?action=getSoftwareVersion

- Currently, the initialization protocol command for CGI is not available, and you need to initialize the device through other methods after restoring it to factory settings.
- For details, see Protocols.

4.2.8 Restarting the Device

URL: http://<server>/cgi-bin/magicBox.cgi?action=reboot[&delay=<paramValue>]

For details, see Protocols.

4.2.9 Turning off the Device

URL: http://<server>/cgi-bin/magicBox.cgi?action=shutdown

For details, see Protocols.

4.2.10 Logs

Note: Recorded in logs are the key information during the operation of the device, including basic information such as configuration and login, and is consistent with the system information displayed on the device web page.

4.2.10.1 Description of Postman Example

See Log Operation. Json in Postman Case List

4.2.10.2 Log Search

1. Search by conditions

   URL:http://192.168.1.108/cgi-bin/log.cgi?action=startFind&condition.StartTime=2011-1-1 12:00:00&condition.EndTime=2011-1-10 12:00:00

   Set the conditions, such as the start time, end time and event type.
   - Response token: The token returned from the search.
   - Response count: The number of returned records.
   - For details, see Protocols.
2. Global search
   - URL: http://192.168.1.108/cgi-bin/log.cgi?action=doFind&token=1&count=100
   - Set the conditions, such as the start time, end time and event type.
   - Response token: The token returned from the search.
   - Response count: The number of returned records.
   - The URL can be implemented circularly, and no more than count items can be obtained once.
   - For example, the device has 200 records altogether.
   - The first time the URL is implemented, 1 to 100 records will be obtained.
   - The second time the URL is implemented, 101 to 200 records will be obtained.
   - The third time the URL is implemented, 201 to 220 records will be obtained.
   - The fourth time the URL is implemented, 0 record will be obtained.
   - For details, see Protocols.

4.2.10.3 Log Backup

URL:http://<server>/cgi-bin/Log.backup?action=All&condition.StartTime=<startTime>&condition.EndTime=<endTime>

For details, see Protocols.

4.2.11 Time/Date Calibration

4.2.11.1 Getting the Current Time

URL: http://<server>/cgi-bin/global.cgi?action=getCurrentTime

For details, see Protocols.

4.2.11.2 Setting the Current Time

URL: http://<server>/cgi-bin/global.cgi?action=setCurrentTime&time=2011-7-3%2021:02:32

For details, see Protocols.

4.2.12 Daylight Saving Time (DST)

4.2.12.1 Getting the DST

URL: http://<server>/cgi-bin/configManager.cgi?action=getConfig&name=Locales

- For details on the configuration of Locals, see "Configuring DST Format".
- For details, see Protocols.

4.2.12.2 Setting the DST

URL: http://<server>/cgi-bin/configManager.cgi?action=setConfig&<paramName>=<paramValue>[&<paramName>=<paramValue>…]

- See "Access Time Schedule" for reference.
- For details on the configuration of Locals, see "Configuring DST Format".
- For details, see Protocols.

4.3 Getting the Unlocking Records

4.3.1 Getting the Unlocking Records

A complete unlocking record generally includes and the record entry and snapshots.

4.3.1.1 Getting the Card Swiping Records

URL:http://192.168.1.108/cgi-bin/recordFinder.cgi?action=find&name=AccessControlCardRec

- Get the card swiping records. Maximum 1024 records can be obtained each time.
- You cannot obtain all card swiping records through the protocol. To obtain all records, see "Getting the Card Swiping Records in Batches".
- For details, see Protocols.
- For response, see "Format of the Unlocking Records".

4.3.1.2 Getting the Card Swiping Records in Batches

Step 1 Use the following command to get the value of totalCount, found and the CreateTime of the last record.

URL: http://192.168.1.108/cgi-bin/recordFinder.cgi?action=find&name=AccessControlCardRec&StartTime=123456700&EndTime=153456800&condition.CardNo=12001&count=100

- TotalCount: The total number of records obtained.
- Found: The number of records returned currently.

Response:{

found=100

….

records[99].RecNo=12345

records[99].CreateTime=140556698

records[99].CardNo=12001

records[99].CardName=ZhangSan

records[99].UserID=ZhangSan

}

- StartTime, EndTime: The duration of the record, UTC time.
- CardNo: Card information to be filtered.
- If no condition is set, all records will be searched for. No more than 1024 records will be searched for.

- Found: The returned records, and its number is no more than the count.
- For details, see Protocols.
- For response, see "Format of the Unlocking Records".

Step 2 If found ≤ count, you need to fill the CreateTime value of the last record obtained in the previous step into the StartTime field, update the EndTime field value, and then continue to use the getting command.

URL: http://192.168.1.108/cgi-bin/recordFinder.cgi?action=find&name=AccessControlCardRec&StartTime=140556698&EndTime=140586698&condition.CardNo=12001&count=100

EndTime: The default value is 0xffffff.

- StartTime: Empty. The default value is 0.
- Count: Maximum number of records to be searched for this time.
- Found: The number of returned records currently. When found=0, the search ends.
- For details, see Protocols
- For response, see "Format of the Unlocking Records".

Step 3 Repeat the getting operations until found=1, that is, CreateTime=StartTime

Note: In terms of this method, multiple records exist for the same timestamp, and there might be overlapping records in two returns. Therefore, you need to perform the de-duplication operation (to remove duplicate records) every time after the client gets data.

4.3.2 Getting Linked Snapshots of Card Swiping Records

URL:http://192.168.1.108/cgi-bin/FileManager.cgi?action=downloadFile&fileName=download.jpg

- FileName: The absolute path of the file in the device. You can get the corresponding unlocking records through "Getting the Unlocking Records".
- Format of the unlocking records: "Format of the Unlocking Records".

Fields

URL	string	No	Image URL, maximum length: 127 (The video intercom device does not support this field)

4.4 Alarm Records

URL:http://192.168.1.108/cgi-bin/recordFinder.cgi?action=find&name=AccessControlAlarmRecord&StartTime=2014-8-25 00:02:32&EndTime=2014-8-25 01:02:32&count=500

- StartTime, EndTime: Duration of the alarm records
- Count: Maximum number of records to be searched for. If no value is specified. the default value is 500.
- For details, see Protocols.
- Returned value: "Format of the Alarm Records"

4.5 Access Control

4.5.1 Unlocking and Locking the Door

4.5.1.1 Unlocking

URL:http://192.168.1.108/cgi-bin/accessControl.cgi?action=openDoor&channel=1&Type=Remote

Channel: Starting from 1, which means door 0; and Channel=2, which means door 1

- For details, see Protocol.

4.5.1.2 Locking

URL:http://192.168.1.108/cgi-bin/accessControl.cgi?action=closeDoor&channel=1&Type=Remote

- Channel: Starting from 1, which means door 0; and Channel=2, which means door 1
- For details, see Protocol.

4.5.2 Getting the Door Status

URL: http://192.168.1.108/cgi-bin/accessControl.cgi?action=getDoorStatus&channel=1

- Channel: Starting from 1, which means door 0; and Channel=2, which means door 1
- For details, see Protocol.

4.6 Configuring Access Control

4.6.1 General Settings

General Access Control Configuration Instructions" in the appendix.

4.6.2 Time Period

Access is a separate configuration, with the configuration name of "AccessTimeSchedule". For the configuration and fields description, see "Time Period". For the basic operations of getting and setting, see "Getting and Setting Configurations".

Setting command example:

URL: http://172.5.2.142/cgi-bin/configManager.cgi?action=setConfig&AccessTimeSchedule[0].Enable=true&AccessTimeSchedule[0].TimeSchedule[0][0]=1 00:00:00-20:00:59&AccessTimeSchedule[0].TimeSchedule[0][1]=1 23:45:00-23:59:59

TimeSchedule[x][x]="1 00:00:00-23:59:59”

1: enable, 0: disable.

12:00:00 AM -23:00:00-23:59:59

4.6.3 Unlocking Methods

Unlocking methods can be modified according to the value of the method from **Access Control**. For basic operations on getting and settings, refer to "Getting and Setting Configurations".

Request

Template	http://<server>/cgi-bin/configManager.cgi?action=setConfig&AccessControl[channel].Method=37			
Method	GET			
Parameter Format	key=value format at URL			
Parameter	**Type**	**Required**	**Description**	**Example**
channel	integer	R	Channel number and access control number, starting from number 0	0

Method	uint8	R	Unlocking method. The order of verification types should be strictly followed. 0: By password only. 1: By swiping card only. 2: By password or swiping card. 3: Use password after swiping card. 4: Swipe card after using password. 5: Unlock by periods. It subject to the specific unlocking method under the TimeSchedule node in this method. 6: By fingerprint only. 7: By password or swiping card or fingerprint. 8: Combination of swiping card + password + fingerprint. 9: Combination of password + fingerprint. 10: Combination of swiping card + fingerprint. 16: UserID + password. 17: By face only. 18: Combination of face + password. 19: Combination of fingerprint + password. 20: Combination of fingerprint + face. 21: Combination of card + face. 22: By face or password. 23: By fingerprint or password. 24: By fingerprint or face. 25: By card or face. 26: By card or fingerprint. 27: Combination of fingerprint + face + password. 28: Combination of card + face + password.	2

			29: Combination of card + fingerprint + password. 30: Combination of card + fingerprint + face. 31: By fingerprint or face or password. 32: By card or face or password. 33: By card or fingerprint or face. 34: Combination of card + fingerprint + face + password. 35: By card or fingerprint or face or password. 36: By (ID card + person and ID card comparison) or swiping card. 37: By person and ID card comparison or swiping card (QR code). 38: (Card + password) | (fingerprint + password). 43: By multiple users. 44: By person and ID card comparison or health code, 2 by default.	
Example				
http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&AccessControl[0].Method=2				

Response

Parameter Format	OK at body			
Parameter	**Type**	**Required**	**Description**	**Example**
Example				
OK				

4.6.4 Alarm

Unlocking methods can be modified according to the value of the method from BreakInAlarmEnable, RepeatEnterAlarm, DoorNotClosedAlarmEnable, and DuressAlarmEnable. Modify the according alarms to enable the configuration.

For basic operations on getting and settings, refer to "Getting and Setting Configurations".

Request

Template	http://<server>/cgi-bin/configManager.cgi?action=setConfig&AccessControl[channel]. BreakInAlarmEnable =true			
Method	GET			
Parameter Format	key=value format at URL			
Parameter	**Type**	**Required**	**Description**	**Example**
Channel:	integer	R	Channel number and Access Control number, starting from number 0	0
BreakInAlarmEnable	Type bool.	O	Enable intrusion alarm.	true
RepeatEnterAlarm	Type bool.	O	Enable repeated entry alarm.	true
DoorNotClosedAlarmEnable	Type bool.	O	Enable alarm of not closed door.	true
DuressAlarmEnable	Type bool.	O	Enable duress alarm.	true
Example				
http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&AccessControl[0]. BreakInAlarmEnable =true& AccessControl[0]. DuressAlarmEnable =true				

Response

Parameter Format	OK at body			
Parameter	**Type**	**Required**	**Description**	**Example**
Complete Example				

OK

4.6.5 Door Status

Unlocking methods can be modified according to the value of the State on AccessControl. Modify the according door status.

For basic operations on getting and settings, refer to "Getting and Setting Configurations".

Request

Template	http://<server>/cgi-bin/configManager.cgi?action=setConfig&AccessControl[channel].State=Normal			
Method	GET			
Parameter Format	key=value format at URL			
Parameter	**Type**	**Required**	**Description**	**Example**
Channel	integer	R	Channel number and Access Control number, starting from number 0	1
Status	enumchar[32]	R	Status: Enumchar[32]{Normal: Normal CloseAlways: Normally closed. OpenAlways: Normally open /*In the normally open and normally closed status, Opendoor cannot unlock the door.*/ NoPersonNC: Normally closed when there is no person, discarded. NoPersonNO: Normally open when there is no person, discarded.	Normal
Complete Example				

http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&AccessControl[0].State=Normal

Response

Parameter Format	OK at body			
Parameter	**Type**	**Required**	**Description**	**Example**
Complete Example				
OK				

4.6.6 Door Holding Time

Unlocking methods can be modified according to the value of the **UnlockHoldInterval** from AccessControl. For basic operations on getting and settings, refer to "Getting and Setting Configurations"

Request

Template	http://<server>/cgi-bin/configManager.cgi?action=setConfig&AccessControl[channel].UnlockHoldInterval =1000			
Method	GET			
Parameter Format	key=value format at URL			
Parameter	**Type**	**Required**	**Description**	**Example**
channel	integer	R	Channel number and Access Control number, starting from number 0.	1
UnlockHoldInterval	uint	R	Unlock NC/NO output holding time (door holding time), in ms, the range is 250 ms–20000 ms.	1000
Complete Example				
http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&AccessControl[0].UnlockHoldInterval=2000				

Response

Parameter Format	OK at body			
Parameter	**Type**	**Required**	**Description**	**Example**
Complete Example				
OK				

4.6.7 Holiday

- The Holidays and Festivals function is controlled by two configurations: SpecialDaysSchedule and SpecialDayGroup.
- SpecialDaysSchedule is linked with SpecialDayGroup.
- The **SpecialDaysSchedule** array configuration specifies the periods of each day of the holiday and valid door channels corresponding to the **SpecialDayGroup**.
- The **SpecialDayGroup** array configuration specifies the start time and end time of a holiday or festival.
- The **GroupNo** field in **SpecialDaysSchedule** saves the subscript of the corresponding **SpecialDayGroup** configuration.

Procedures for configuring holidays and festivals permissions of persons:

Step 1 Configure the start time of a holiday or festival in SpecialDayGroup.

Configure SpecialDayGroup:

- URL:http://172.5.2.142/cgi-bin/configManager.cgi?action=setConfig&SpecialDayGroup[0].Name=testName&SpecialDayGroup[0].Enable=true&SpecialDayGroup[0].Days[0].StartTime=2021-10-01 10:00:00&SpecialDayGroup[0].Days[0].EndTime=2021-10-07 23:59:59&SpecialDayGroup[0].Days[0].SpecialDayName=National
- SpecialDayGroup: Supports up to 128.
- Enable the corresponding **SpecialDayGroup**, and then configure the start time of a holiday or festival.
- For details, see Protocol.

Step 2 Configure the time periods per day in a holiday or festival and the corresponding door permissions in SpecialDaysSchedule.

URL: http://172.5.2.142/cgi-bin/configManager.cgi?action=setConfig&SpecialDaysSchedule[0].Name=testName&SpecialDaysSchedule[0].Enable=true&SpecialDaysSchedule[0].Doors[0]=0&SpecialDaysSchedule[0].TimeSection[0]=1 08:00:00-16:59:59&SpecialDaysSchedule[0].GroupNo=0

- SpecialDaysSchedule: Supports up to 128.
- TimeSection: Supports 4 time periods

- 1 15:00:00-20:00:00: 1 means **SpecialDaysSchedule** will operate checking.
- Doors: This schedule is only suitable for No. X door. If Doors=255, all doors are valid by default.
- For details, see Protocol.

Step 3 Record the array subscript of SpecialDaysSchedule configuration in the SpecialDaysSchedule array field of person permissions.

First generation protocol:

URL:http://172.5.2.142/cgi-bin/recordUpdater.cgi?action=insert&name=AccessControlCard&CardName=ZhangSan&CardNo=12345&UserID=102&CardStatus=0&CardType=0&Password=123456&Doors[0]=1&Doors[1]=3&Doors[2]=5&ValidDateStart=20210111 093811&ValidDateEnd=20211222 093811&SpecialDaysSchedule[0]=0

- SpecialDaysSchedule:**SpecialDaysSchedule** for personnel
- For details, see Protocols.

Second generation protocol:

- URL: POST http://192.168.1.108/cgi-bin/AccessUser.cgi?action=insertMulti
- The protocol is only available on devices that support second-generation access control.
- For details, see Protocols.

Summary

Logic of SpecialDaysSchedule

Step 1 Check personnel, and then get the corresponding **SpecialDaysSchedule** number.

Step 2 Check the corresponding **SpecialDayGroup** from the **GroupNo** of the **SpecialDaysSchedule**.

Step 3 Check if the **SpecialDaysSchedule** is enabled or not, and then check if the current date lies within the range.

Step 4 Check the enabling of **SpecialDaysSchedule**, the door group number, and then the time period.

4.6.8 Enabling Capturing Settings

For details, see "General Access Control Configuration Instructions" in the appendix.

For the configuration format, see "Access Configuration".

Request

Template	http://<server>/cgi-bin/configManager.cgi?action=setConfig&AccessConfig.PhotoGraph=0

Method	GET			
Parameter Format	key=value format at URL			
Parameter	**Type**	**Required**	**Description**	**Example**
+PhotoGraph	uint32	R	Enable the capturing function or not. After enabling, the camera will take a snapshot of the background and save the image. 0: Not taking 1: Taking	1
Complete Example				
http://172.10.54.182/cgi-bin/configManager.cgi?action=setConfig&AccessConfig.PhotoGraph=1				

Response

Parameter Format	OK at body			
Parameter	**Type**	**Required**	**Description**	**Example**
Complete Example				
OK				

4.6.9 Enabling Administrator Password

Unlocking methods can be modified according to the value of the **CustomPasswordEnable** on AccessControl. Modify the according door status.

For basic operations on getting and settings, refer to "Getting and Setting the Configuration".

Request

Template	http://<server>/cgi-bin/configManager.cgi?action=setConfig&AccessControl[channel].State=Normal
Method	GET
Parameter Format	key=value format at URL

Parameter	Type	Required	Description	Example
channel	integer	R	Channel number and Access Control number, starting from number 0	1
+CustomPasswordEnable	bool	O	Enable the Administrator Password or not "CustomPasswordEnable: true,"	true
Complete Example				
http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&AccessControl[0].CustomPasswordEnable=true				

Response

Parameter Format	OK at body			
Parameter	**Type**	**Required**	**Description**	**Example**
Complete Example				
OK				

4.6.10 Motion Detection

For details, see "General Access Control Configuration Instructions" in the appendix.

For the configuration format, see "Motion Detection".

Request

Template	http://<server>/cgi-bin/configManager.cgi?action=setConfig&MotionDetect[0].MotionDetectWindow[0].Sensitive=54&MotionDetect[0].MotionDetectWindow[0].Threshold=66			
Method	GET			
Parameter Format	key=value format at URL			
Parameter	**Type**	**Required**	**Description**	**Example**

Threshold	uint8	O	Area threshold, with the range of [0–100].	50
Sensitive	uint8	O	Sensitivity, with the range of [0–100].	50
Region object	int	O	Motion detection region blocks: An array, in which each row of the region is represented by a 32-bit integer; each bit of the integer corresponds to a block; the left side of the screen corresponds to higher bits. Note: The correspondence between the rows and columns on the protocol and the coordinates of the input channel image blocks is as follows: Image columns: Left to right. Corresponding protocol columns (bits): Left (higher bits) to right (lower bits). Because motion detection only has 22 columns, the lower 22 bits and the higher 10 bits should be fixed as 0. Image rows: Up to down; Corresponding protocol rows: Up to down; Third-generation motion detect field. It is invalid in the first-generation motion detection, and use the full screen Region field.	[4194303, 3145728, …]
Complete Example				
http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&MotionDetect[0].MotionDetectWindow[0].Sensitive=54&MotionDetect[0].MotionDetectWindow[0].Threshold=66				

4.6.11 Configuring CGI Auto Registration

For the specific configuration method, see "General Access Control Configuration Instructions" in the appendix. Auto registration mainly relies on the VSP_CGI configuration format. For details, see "错误!未找到引用源。 错误!未找到引用源。".

Template	http://<server>/cgi-bin/configManager.cgi?action=setConfig&VSP_CGI.ServiceStart=false			
Method	GET			
Parameter Format	key=value format in URL			
Parameters	**Type**	**Required**	**Description**	**Example**
+ServiceStart	bool	O	CGI service control configuration. It is true by default.	true
+AutoRegister	object	O	Auto registration.	
Enable	bool	O	Enable.	false
++DeviceID	char[64]	O	Device ID.	""
++Servers	object[]	O	Client address. The large and small web pages can be automatically increased or decreased, and the number of it ranges from 1 to 4.	
+++Type	enumint	O	Address type. It is displayed through a drop-down list on the web page. Available values: 0: IP address; 1: Domain name.	0
+++Address	char[64]	O	IP address. When the value of Type is 0, the field is displayed on the web page.	""
+++Port	uint32	O	Port. When the value of Type is 0, the field is	80

			displayed on the web page.	
+++DoMain	char[128]	O	Domain address. When the value of Type is 1, the field is displayed on the web page.	""
+++HttpsEnable	bool	O	Enable HTTPS.	false
Example				
http://172.5.46.11/cgi-bin/configManager.cgi?action=setConfig&VSP_CGI.ServiceStart=false				

4.7 Communication Configuration

For the specific configuration methods (getting and setting), see "General Access Control Configuration Instructions".

4.7.1 Configuring Wiegand

For details, see "5.8.6 Wiegand".

Request

Template	http://<server>/cgi-bin/configManager.cgi?action=setConfig&/cgi-bin/configManager.cgi?action=setConfig&Wiegand[wiegandchannel].InputType=3			
Method	GET			
Parameter Format	key=value format in URL			
Parameter	**Type**	**Required**	**Description**	**Example**
wiegandchannel	Int	R	Wigand channel number, starting from 0.	0
Mode	integer	R	Work mode. enumint{ 0: Wigand input. 1: Wigand output. }	+Mode
OutType	integer	R	Output type. Enumint{ 0: Output ID.	1

			1: Output card number. }	
PulseStep	integer	R	Pulse interval. Unit: us. The value range depends on the connected peripheral and might vary.	1000
PulseWidth	integer	R	Pulse width. Unit: us. The value range depends on the connected peripheral and might vary.	+PulseWidth

TransferMode	integer	R	Transmission mode.
			enumint{
			0: Transmit through Wigand 34bit. 4-byte card number and 2Bit verification.
			1: Transmit through Wigand 66bit. 8-byte card number and 2Bit verification.
			2: Transmit through Wigand 26bit. 3-byte card number and 2Bit verification.
			}
				1

Complete Example				
http://192.168.0.12/cgi-bin/configManager.cgi?action=setConfig&Wiegand[0].InputType=1&Wiegand[0].Mode=0&Wiegand[0].OutType=1&Wiegand[0].PulseStep=1000&Wiegand[0].PulseWidth=200&Wiegand[0].TransferMode=0				

Response

Parameter Format	OK at body			
Parameter	**Type**	**Required**	**Description**	**Example**
Complete Example				

OK

4.7.2 Configuring Wi-Fi

4.7.2.1 Getting Wi-Fi Configuration

URL: http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=WLan

For details, see "5 Protocol Description".

4.7.2.2 Enabling Wi-Fi

URL: http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&WLan.eth2.Enable=true

For details, see "5 Protocol Description".

4.7.2.3 Searching for Wi-Fi

URL:http://192.168.1.108/cgi-bin/wlan.cgi?action=scanWlanDevices&SSID=xia_yuguo 13098 Internet

For details, see "5 Protocol Description".

4.7.3 Configuring Wired Network

4.7.3.1 Getting Wired Network Configuration

URL:http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=Network

For details, see "5 Protocol Description".

4.7.3.2 Configuring the IP Address of NIC 1

URL:http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&Network.eth0.IPAddress=192.168.1.108

For details, see "5 Protocol Description".

4.7.3.3 Configuring the Parameters for NIC 1

http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&Network.eth0.IPAddress=192.168.1.108&Network.eth0.SubnetMask=255.255.255.0&Network.eth0.DefaultGateway=192.168.1.1

Configure the IPV4 address, subnet mask, default gateway, preferred DNS server, and alternate DNS server in sequence.

For details, see "5 Protocol Description".

4.7.3.4 Configuring the IP Address of NIC 2

URL:http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&Network.eth1.IPAddress=192.168.1.108

For details, see "5 Protocol Description".

4.7.4 Configuring IPv6 Parameters

4.7.4.1 Configuring IPv6

URL:http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&IPv6.eth0.DefaultGateway=2001::1&IPv6.eth0.IPAddress=2001::24&IPv6.Enable=true

Configure the default IPV6 gateway and IP address, and enable IPV6 in sequence.

For details, see "5 Protocol Description".

4.7.4.2 Enabling IPv6

URL:http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&IPv6.Enable=true

For details, see "5 Protocol Description".

4.8 Configuring Intelligent Recognition

4.8.1 Configuring Face Recognition Threshold

Use http://192.168.1.108/cgi-bin/faceRecognitionServer.cgi?action=findGroup&groupID=1 to search for the corresponding face configuration information first.

Request

Template	http://<server>/cgi-bin/faceRecognitionServer.cgi?action=modifyGroup			
Method	GET			
Parameter Format	key=value format in URL			
Parameter	**Type**	**Required**	**Description**	**Example**
groupID	string	R	Person group ID. Up to 63 characters.	10000
groupName	string	R	Person group name. Up to 127 characters.	ASI
groupDetail	string	O	Person group remarks. Up to 255 characters.	ForTest1
Similarity	uint8	O	Similarity threshold [0,100].	90
Alive	uint8	O	Liveness threshold [0,100].	0
MaskSimilarity	uint8	O	Similarity threshold for face masks. [0,100]	0
Complete Example				
http://192.168.1.108/cgi-bin/faceRecognitionServer.cgi?action=modifyGroup&groupID=10000&groupName=ASI&Similarity=50				

Response

Parameter Format	OK at body			
Parameter	**Type**	**Required**	**Description**	**Example**
Complete Example				
OK				

4.8.2 Configuring Pupillary Distance

For the detailed configuration method, see "General Access Control Configuration Instructions".

For the configuration format, see "Video Analyse Rule".

Request

Template	http://<server>/cgi-bin/configManager.cgi?action=setConfig&VideoAnalyseRule[0][0].Config.EyesDistThreshold=100			
Method	GET			
Parameter Format	key=value format in URL			
Parameter	**Type**	**R/O**	**Description**	**Example**
EyesDistThreshold	uint32	R	The value of pupillary distance	60
Complete Example				
http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&VideoAnalyseRule[0][0].Config.EyesDistThreshold=100				

Response

Parameter Format	OK at body			
Parameter	**Type**	**R/O**	**Description**	**Example**
Complete Example				
OK				

4.8.3 Configuring Liveness Detection

For the detailed configuration method, see "General Access Control Configuration Instructions" in the appendix.

For the configuration format, see “Intelligent Rules (VideoAnalyseRule)".

Request

Template	http://<server>/cgi-bin/configManager.cgi?action=setConfig&VideoAnalyseRule[0][0].Config.FilterUnAliveEnable=false			
Method	GET			
Parameter Format	key=value format in URL			
Parameter	**Type**	**R/O**	**Description**	**Example**
FilterUnAliveEnable	bool	R	Whether to enable non-living filtering; the default value is false.	false
Complete Example				
http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&VideoAnalyseRule[0][0].Config.FilterUnAliveEnable =true				

Response

Parameter Format	OK at body			
Parameter	**Type**	**R/O**	**Description**	**Example**
Complete Example				
OK				

4.8.4 Face Recognition Interval for the Same Individual

Request

Template	http://<server>/cgi-bin/configManager.cgi?action=setConfig&FaceSnapshot[0].OptimalTime=3			
Method	GET			
Parameter Format	key=value format in URL			
Parameter	**Type**	**R/O**	**Description**	**Example**
FaceSnapshot	object[]	Empty	One-dimensional array. Each represents a video channel.	

+OptimalTime	uint16	Empty	Face priority maximum delay (in seconds)	3
Example				
http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&FaceSnapshot[0].OptimalTime=3				

Response

Parameter Format	OK at body			
Parameter	**Type**	**R/O**	**Description**	**Example**
Example				
OK				

4.8.5 Face Recognition Failure Timeout

Request

Template	http://<server>/cgi-bin/configManager.cgi?action=setConfig&FaceSnapshot[0].RecognizeTimeout=10			
Method	GET			
Parameter Format	key=value format in URL			
Parameter	**Type**	**R/O**	**Description**	**Example**
FaceSnapshot	object[]	Empty	One-dimensional array. Each represents a video channel	
+RecognizeTimeout	uint16	No	Target recognition failure timeout (in seconds)	5
Example				
http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&FaceSnapshot[0].RecognizeTimeout =5				

Response

Parameter Format	OK at body			
Parameter	**Type**	**R/O**	**Description**	**Example**
Example				

OK

4.8.6 Getting Human Faces

Request

Template	http://<server>/cgi-bin/accessControl.cgi?action=captureCmd&type=1&UserID=5&heartbeat=5&timeout=10			
Method	GET			
Parameter Format	key=value format in URL			
Parameter	**Type**	**Required**	**Description**	**Example**
type	int	R	Face registration.	1
UserID	string	O		
heartbeat	int	R	Heartbeat interval. Unit: Second.	5
timeout	int	O	Timeout period. Unit: Second	10
Complete Example				
http://192.168.1.108/cgi-bin/accessControl.cgi?action=captureCmd&type=1&UserID=5&heartbeat=5&timeout=10				

Response

Parameter Format	OK at body			
Parameter	**Type**	**R/O**	**Description**	**Example**
Complete Example				
OK				

4.9 System Settings

4.9.1 Configuring Illuminators

For the detailed configuration method, see "General Access Control Configuration Instructions" in the appendix.

For the configuration format, see “Configuring Illuminator (SignLight)”.

Request

Template	http://<server>/cgi-bin/configManager.cgi?action=setConfig&SignLight[0].onCycle=30			
Method	GET			
Parameter Format	key=value format in URL			
Parameter	**Type**	**R/O**	**Description**	**Example**
onCycle	Uint32	R	Range: 0–100	30
Complete Example				
http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&SignLight[0].onCycle=30				

Response

Parameter Format	OK at body			
Parameter	**Type**	**R/O**	**Description**	**Example**
Complete Example				
OK				

4.9.2 Volume Control

For the detailed configuration method, see "General Access Control Configuration Instructions" in the appendix.

For the configuration format, see “Configuring Audio Output Volume”.

Request

Template	http://<server>/cgi-bin/configManager.cgi?action=setConfig&AudioOutputVolume[0]=20			
Method	GET			
Parameter Format	key=value format in URL			
Parameter	**Type**	**R/O**	**Description**	**Example**
AudioOutputVolume	Uint32[]	R	Each element of the array represents an audio channel volume, with the range of [0–100].	30

Complete Example
http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&AudioOutputVolume[0]=20

Response

Parameter Format	OK at body			
Parameter	**Type**	**R/O**	**Description**	**Example**
Complete Example				
OK				

4.9.3 Configuring Video Images

4.9.3.1 Configuring Brightness, Contrast, and Saturation

For the detailed configuration method, see "General Access Control Configuration Instructions" in the appendix.

For the configuration format, see “Configuring Video Input Colors”.

Get configuration information.

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=getConfig&name=VideoColor
Method	GET
Description	Get brightness, contrast and saturation
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=VideoColor
Success Return	***head***.Name=Day, ***head***.Brightness=50, ***head***.Contrast=50, ***head***.Saturation=50, ***head***.Hue=50, ***head***.Gamma=50, ***head***.ChromaSuppress=50, ***head***.Style=Standard, ***head***.TimeSection=1 00:00:00-24:00:00
Comment	Parameters in the URL: paramName and paramValue are as table below. In table below,

	head = table.VideoInSharpness [***ChannelNo***] [***ConfigNo***] ***ChannelNo***: integer, array index starts from 0, which means video channel (equals to video channel index -1, and so 0 means channel 1). ***ConfigNo***: array index, can be 0, 1 or 2. 0 means config for day, 1 means config for night, and 2 means config for normal scene.

Configure parameters

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=setConfig&<***paramName***>=<***paramValue***>[&<***paramName***>=<***paramValue***>…]
Method	SET
Description	Set brightness, contrast and saturation
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&VideoColor[0][0].Brightness=50
Success Return	OK
Comment	Parameters in the URL: paramName and paramValue are as table below. In table below, ***head*** = table.VideoInSharpness [***ChannelNo***] [***ConfigNo***] ***ChannelNo***: integer, array index starts from 0, which means video channel (equals to video channel index -1, and so 0 means channel 1). ***ConfigNo***: array index, can be 0, 1 or 2. 0 means config for day, 1 means config for night, and 2 means config for normal scene.

Appendix:

ParamName	ParamValue type	Description
head. Brightness	integer	Brightness, range is [0–100]
head. Contrast	integer	Contrast, range is [0–100]
head. Saturation	integer	Saturation, range is [0–100]

4.9.3.2 Configuring Day/Night Mode

For the detailed configuration method, see "General Access Control Configuration Instructions" in the appendix.

For the configuration format, see ”Configuring Day/Night Mode”.

Table 4-1 Get configuration information

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=getConfig&name=VideoInOptions
Method	GET
Description	
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=VideoInOptions
Success Return	…… table.VideoInOptions[0].DayNightColor=2 ……
Comment	***head*** = table. VideoInOptions[***ChannelNo***]: ***ChannelNo***: integer, array index starts from 0, which means video channel (equals to video channel index -1, and so 0 means channel 1).

Table 4-2 Configure parameters

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=setConfig&<***paramName***>=<***paramValue***>
Method	SET
Description	Set day-night
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&VideoInOptions[0].DayNightColor=1
Success Return	OK
Comment	Parameters in the URL: ParamName and paramValue are as table below. In table below, ***head*** = VideoInOptions[0]

Appendix:

ParamName	**ParamValue type**	**Description**
head.DayNightColor	int	Automatically switch colors at day and night: 0: Always colored; 1: Auto switch based on the brightness; 2: Always black & white.

4.10 Configuring Temperature Monitoring

For the detailed configuration method, see "General Access Control Configuration Instructions" in the appendix.

For the configuration format, see "Temperature Monitoring".

GuideModuleParam corresponds to T1 device, and WristTemperatureParam corresponds to T0 device.

Request

Template	http://<server>/cgi-bin/configManager.cgi?action=setConfig&SignLight[0].onCycle=30			
Method	GET			
Parameter Format	key=value format in URL			
Parameter	**Type**	**R/O**	**Description**	**Example**
+Enable	bool	R	Whether to enable temperature monitoring.	
Complete Example				
http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&SignLight[0].onCycle=30				

Response

Parameter Format	OK at body			
Parameter	**Type**	**R/O**	**Description**	**Example**
Complete Example				
OK				

4.11 Subscribing for Events

At present, there are two types of unlock events for access control subscriptions: general events and intelligent events.

At present, the access control subscription for the unlock events carried out through Subscribing for Intelligent Events. You can obtain real-time images of corresponding events. The smart events of access control devices only support unlock events and face-ID events.

If images are not required, you can also subscribe for unlock events and face-ID events through “Subscribing for General Events”, but you need to disable snapshot. For details, see Disabling Snapshot.

4.11.1 Subscribing for General Events

URL: &heartbeat=5http://192.168.1.108/cgi-bin/eventManager.cgi?action=attach&codes=[All]

- Codes: Enter the event name that needs to be subscribed. You can see the protocol for the specific name. ALL means subscribing to all events.
- Keepalive: Keep alive time for the client (caller)
- Heartbeat: Keep alive time for the service terminal (device)
- For details, see Protocols.

4.11.2 Subscribing for Intelligent Events

URL:http://192.168.1.108/cgi-bin/snapManager.cgi?action=attachFileProc&Flags[0]=Event&Events=[AccessControl]&heartbeat=5

- Codes: Enter the event name that needs to be subscribed. You can see the protocol for the specific name. ALL means subscribing to all events.
- Keepalive: Keep alive time for the client (caller)
- Heartbeat: Keep alive time for the service terminal (device)
- For details, see Protocols.

4.12 Managing Permissions

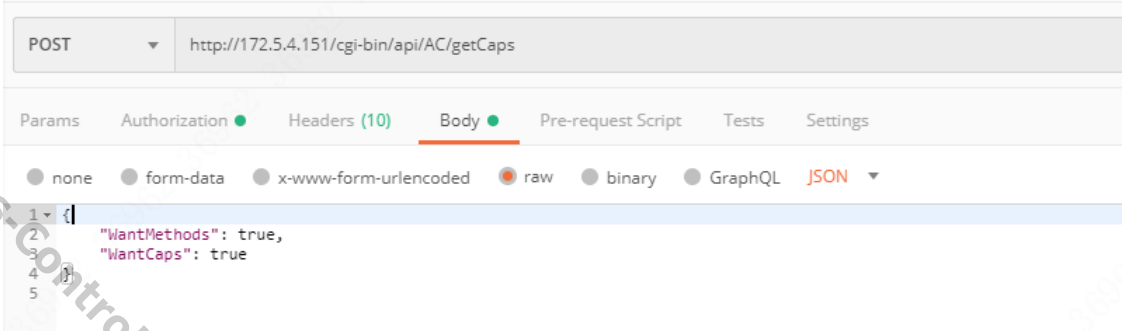
Dahua access control protocol is divided into first-generation and second-generation access control protocols.

- The permission is divided into: personnel, card, fingerprint, face, password permissions (same for the first-generation and second-generation protocols).
- In the first-generation access control protocol, personnel, card, and fingerprint binding permissions need to be sent at the same time, and face permission is sent separately.
- In the second-generation access control protocol, the above 4 permissions can be sent separately, but the personnel permission must be sent first.
- First-generation and second-generation access control protocols cannot be mixed.
- If the device supports the second-generation access control protocol, the second-generation protocol is preferred.

4.12.1 Second-generation Access Control Protocol

4.12.1.1 Searching for Second-generation Protocol Capabilities

URL:http://192.168.0.101/cgi-bin/api/AC/getCaps



Description of important return value:

- AccessUser: Personnel capability description of the second-generation protocol
- AccessCard: Card-related capability description of the second-generation protocol
- AccessFingerprint: Fingerprint-related capability description of the second-generation protocol
- AccessFace: Face-related capability description of the second-generation protocol
- If the capability can be obtained through CGI, it supports the second-generation protocol.

4.12.1.2 User Information (Second-generation Protocol)

4.12.1.2.1 Sending Postman Cases by Users

View “Second-generation protocol_personnel operations) in Postman Case List.

4.12.1.2.2 Sending User Information

URL:POST http://192.168.1.108/cgi-bin/AccessUser.cgi?action=insertMulti

{ *"UserList" :[{*

"UserID": "100013",

"UserName": "",

"UserType": 0,

"UseTime": 1,

"IsFirstEnter": true,

"FirstEnterDoors": [0, 1],

"UserStatus": 0,

"Authority": 1,

"CitizenIDNo": "123456789012345678",

"Password": "xxxxxxxxxx",

"Doors": [1,3,5,7],

"TimeSections": [1,2,3,4],

"SpecialDaysSchedule": [1,2],

"ValidFrom": "2018-01-02 00:00:00",

"ValidTo": "2018-01-02 01:00:00",

} ,…,{}]

}

- To use this protocol, make sure that the device supports the capabilities of the second-generation access control.
- To perform this action, UserID must not exist. It can be searched through person search protocol.
- Doors: Corresponds to door permissions, starting from 0. It corresponds to the subscript of the AccessControl configuration.
- TimeSections: Corresponds to period plan, and the member positions in the array are in one-to-one correspondence with the Door array.
- "Doors": [1,3,5,7],
- "TimeSections": [1,2,3,4],
- In this example, door 3 corresponds to period 2, and door 5 corresponds to period 3.
- SpecialDaysSchedule: Corresponds to holiday plan, and you can set multiple holiday plans for one person.
- ValidFrom: Valid starting time of face and fingerprint biometric authentication
- ValidTo: Valid expiration time of face and fingerprint biometric authentication
- A maximum of 10 personnel lists are supported at one time.
- For details, see the protocol.

4.12.1.2.3 Updating User Information

URL:POST http://192.168.1.108/cgi-bin/AccessUser.cgi?action= updateMulti

```
{
"UserList": [{
   "UserID": "100013",
   "UserName": "",
   "UserType": 0,
   "UseTime": 1,
   "IsFirstEnter": true,
   "FirstEnterDoors": [0, 1],
   "UserStatus": 0,
   "Authority": 1,
   "CitizenIDNo": "123456789012345678",
```

"Password": "xxxxxxxxxx",
"Doors": [1,3,5,7],
"TimeSections": [1,2,3,4],
"SpecialDaysSchedule": [1,2],
"ValidFrom": "2018-01-02 00:00:00",
"ValidTo": "2018-01-02 01:00:00",
} ,…,{}]

- To use this protocol, make sure that the device supports the second-generation access control capability.
- If the user already exists, the update action will be performed for this user (with the same UserID).
- Doors: Corresponds to door permissions, starting from 0. It corresponds to the subscript of the AccessControl configuration.
- TimeSections: Corresponds to period plan, and the member positions in the array are in one-to-one correspondence with the Door array.
- "Doors": [1,3,5,7],
- "TimeSections": [1,2,3,4],
- In this example, door 3 corresponds to period 2, and door 5 corresponds to period 3.
- SpecialDaysSchedule: Corresponds to holiday plan, and you can set multiple holiday plans for one person.
- ValidFrom: Valid starting time of face and fingerprint biometric authentication
- ValidTo: Valid expiration time of face and fingerprint biometric authentication
- A maximum of 10 personnel lists are supported at one time.
- For details, see the protocol.

4.12.1.2.4 Deleting All User Information

URL:http://192.168.1.108/cgi-bin/AccessUser.cgi?action=removeAll

- To use this protocol, make sure that the device supports the second-generation access control capability.
- For details, see the protocol.

4.12.1.2.5 Deleting Information of Multiple Users

URL:http://192.168.1.108/cgi-bin/AccessUser.cgi?action=removeMulti&UserIDList[0]=102&UserIDList [1]=102

- To use this protocol, make sure that the device supports the second-generation access control capability.
- Up to 10 users can be deleted at a time.
- UserID is the prerequisite, otherwise the deletion will fail.
- For details, see the protocol.

4.12.1.2.6 Searching for Information of Multiple Users

URL:http://192.168.1.108/cgi-bin/AccessUser.cgi?action=list&UserIDList[0]=102&UserIDList[1]=102

- To use this protocol, make sure that the device supports the second-generation access control capability.
- This method is suitable for searching for the personnel information with specific ID when knowing UserID.
- All UserIDs in the UserIDlist must be valid, otherwise the search will fail.
- Personnel information of up to 10 users can be searched at a time.
- For details, see the protocol.

4.12.1.2.7 Searching for Personnel Information by Conditions

The main purposes of this scenario: 1. Search for all personnel 2. Search for user information by filtering conditions.

① Start searching

URL:http://192.168.1.108/cgi-bin/AccessUser.cgi?action=startFind&Condition.UserID=1&Condition.ValidFrom=2018-01-02 00:00:00

- To use this protocol, make sure that the device supports the second-generation access control capability.
- Set the filter condition, and the return value is the number of people that match this condition.

  In the example, it means UserID=1 and ValidFrom=2018-01-02 00:00:00.
- If there is no condition, search all data.
- Token: The search token, used in dofind.
- Total: The number of qualified search results.
- Caps: The number of search entries supported at a time. the maximum is 1000.
- For details, see the protocol. All fields in the protocol can be used as search conditions.

② Obtain user information

URL: http://192.168.1.108/cgi-bin/AccessUser.cgi?action=doFind&Token=1234&Offset=0&Count=20

- To use this protocol, make sure that the device supports the second-generation access control capability.
- Token: The token returned by startfind.
- Offset: The offset of this search.
- Count: The amount of data to be obtained at a time. The maximum value depends on the return Caps of startfind.

- For details, see the protocol.

③ **Stop searching**

URL: http://192.168.1.108/cgi-bin/AccessUser.cgi?action=stopFind&Token=1234

- To use this protocol, make sure that the device supports the second-generation access control capability.
- Token: The token returned by startfind.
- After the token is used, you need release it. Otherwise, it will occupy memory.
- For details, see the protocol.

4.12.1.2.8 Obtaining Information of All personnel

This section introduces how to searching for all personnel by conditions.

① **Start searching**

URL: http://192.168.1.108/cgi-bin/AccessUser.cgi?action=startFind

- To use this protocol, make sure that the device supports the second-generation access control capability.
- If there is no condition, search all data.
- Token: The search token, used in dofind.
- Total: The number of qualified search results.
- For details, see the protocol. All fields in the protocol can be used as search conditions.

② **Obtain user information**

URL: http://192.168.1.108/cgi-bin/AccessUser.cgi?action=doFind&Token=1234&Offset=0&Count=20

- To use this protocol, make sure that the device supports the second-generation access control capability.
- Search for 20 items at a time. When offset=0, search for records with numbers from 0 to 19.

URL:GET http://192.168.1.108/cgi-bin/AccessUser.cgi?action=doFind&Token=1234&Offset=20&Count=20

- To use this protocol, make sure that the device supports the second-generation access control capability.
- Search for 20 items at a time. When offset=20, search for records with numbers from 20 to 39.

③ **Stop searching**

URL: http://192.168.1.108/cgi-bin/AccessUser.cgi?action=stopFind&Token=1234

- To use this protocol, make sure that the device supports the second-generation access control capability.

Summary:

- The maximum value of Count depends on the return Caps value of startfind, which can be called cyclically. The value is different depending on the offset value.
- Startfind: Start searching for and return the number of personnel, which can be used as the end condition.

4.12.1.3 Card Information (Second-generation Protocol)

4.12.1.3.1 Sending Postman Cases by Card Number

View “Second-generation protocol_card operations) in Postman Case List.

4.12.1.3.2 Sending Card Information

URL: POST http://192.168.1.108/cgi-bin/AccessCard.cgi?action=insertMulti

```
{
CardList[{
   "UserID" : "100013"
   "CardNo" : ""
   "CardType" : 0
   "CardName" : "201-John"
   "CardStatus" : 0
} ,…,{}]
   }
```

- To use this protocol, make sure that the device supports the second-generation access control capability.
- When sending card information, make sure that the personnel information of the UserID corresponding to the card number has been sent.
- The device card number is unique. If the card number conflicts, card information sending will fail.
- For details, see the protocol.

4.12.1.3.3 Updating Card Information

URL: POST http://192.168.1.108/cgi-bin/AccessCard.cgi?action=updateMulti

```
{
CardList[{
   "UserID" : "100013"
   "CardNo" : ""
   "CardType" : 0
```

"CardName" : "201-Joe"
"CardStatus" : 0
} ,…,{}]

- To use this protocol, make sure that the device supports the second-generation access control capability.
- When updating the card information, make sure that it already exists in the device
- When updating the card information, make sure the UserID corresponding to the card already exists in the device.
- You can update the information of 10 cards in CardList at a time.
- For details, see the protocol.

4.12.1.3.4 Deleting Information of All Cards

URL: http://192.168.1.108/cgi-bin/AccessCard.cgi?action=removeAll

- To use this protocol, make sure that the device supports the second-generation access control capability
- For details, see the protocol.

4.12.1.3.5 Deleting Information of Multiple Cards

URL: http://192.168.1.108/cgi-bin/AccessCard.cgi?action=removeMulti&CardNoList[0]=12345678&CardNoList[1]=12345687

- To use this protocol, make sure that the device supports the second-generation access control capability.
- You can delete information of 10 cards at a time.
- If CardNo does not exist, the deletion will fail.
- For details, see the protocol.

4.12.1.3.6 Searching for Information of Multiple Cards

URL: http://192.168.1.108/cgi-bin/AccessCard.cgi?action=list&CardNoList[0]=102&CardNoList[1]=102

- To use this protocol, make sure that the device supports the second-generation access control capability.
- This method is suitable for searching for the Card information with specific CardNo when knowing CardNo.
- All CardNo in the CardNoList must be valid, otherwise the search will fail.
- Information of up to 10 cards can be searched at a time.
- For details, see the protocol.

4.12.1.3.7 Searching for Card Information by Conditions

The main purposes of this scenario: 1. Search for all cards. 2. Search for card information by filtering conditions.

① Start searching

URL: http://192.168.1.108/cgi-bin/AccessCard.cgi?action=startFind&Condition.UserID=1& &Condition.CardType=1

- To use this protocol, make sure that the device supports the second-generation access control capability.
- Set the filter condition, and the return value is the number of cards that match this condition.

  In the example, it means UserID=1 and CardType=1.
- If there is no condition, search all data.
- Token: The search token, used in dofind.
- Total: The number of qualified search results.
- Caps: The number of search entries supported at a time. the maximum is 1000.
- For details, see the protocol. All fields in the protocol can be used as search conditions.

② Obtain card information

URL: http://192.168.1.108/cgi-bin/AccessCard.cgi?action=doFind&Token=1234&Offset=0&Count=20

- To use this protocol, make sure that the device supports the second-generation access control capability.
- Token: The token returned by startfind.
- Offset: The offset of this search.
- Count: The amount of data to be obtained at a time. The maximum value depends on the return Caps of startfind.
- For details, see the protocol.

③ Stop searching

URL: http://192.168.1.108/cgi-bin/AccessCard.cgi?action=stopFind&Token=1234

- To use this protocol, make sure that the device supports the second-generation access control capability.
- Token: The token returned by startfind.
- After the token is used, you need release it. Otherwise, it will occupy memory.
- For details, see the protocol.

4.12.1.3.8 Obtaining Information of All Cards

This section introduces how to searching for all cards by conditions.

① Start searching

URL: http://192.168.1.108/cgi-bin/AccessCard.cgi?action=startFind

- To use this protocol, make sure that the device supports the second-generation access control capability.
- If there is no condition, search all data.
- Token: The search token, used in dofind.
- Total: The number of qualified search results.
- For details, see the protocol. All fields in the protocol can be used as search conditions.

② Obtain information of all cards

URL: http://192.168.1.108/cgi-bin/AccessCard.cgi?action=doFind&Token=1234&Offset=0&Count=20

- To use this protocol, make sure that the device supports the second-generation access control capability.
- Search for 20 items at a time. When offset=0, search for records with numbers from 0 to 19.

URL: http://192.168.1.108/cgi-bin/AccessCard.cgi?action=doFind&Token=1234&Offset=20&Count=20

- To use this protocol, make sure that the device supports the second-generation access control capability.
- Search for 20 items at a time. When offset=20, search for records with numbers from 20 to 39.

③ Stop searching

URL:http://192.168.1.108/cgi-bin/AccessCard.cgi?action=stopFind&Token=1234

- To use this protocol, make sure that the device supports the second-generation access control capability.

Summary:

- The maximum value of Count depends on the return Caps value of startfind, which can be called cyclically. The value is different depending on the offset value.
- Startfind: Start searching for and return the number of cards, which can be used as the end condition.

4.12.1.4 Face Information (Second-generation Protocol)

4.12.1.4.1 Sending Postman Cases by Faces

View “Second-generation protocol_face operations) in Postman Case List.

4.12.1.4.2 Sending Face Information

URL: POST http://192.168.1.108/cgi-bin/AccessFace.cgi?action=insertMulti

{
“FaceList”:[
{
"UserID": "102",
"PhotoData": ["yyyy", "yyyy", …],
" PhotoURL": ["yyyy", "yyyy", …],
],
….
]
}

- To use this protocol, make sure that the device supports the second-generation access control capability.
- When sending face information, make sure that the personnel information of the UserID corresponding to the face has been sent.
- PhotoData and PhotoURL: When both PhotoData and PhotoURL are sent, PhotoData data shall prevail.

  If there is only PhotoURL, it will go to the URL to download data when the network connection is smooth.
- Information of up to 10 face can be sent at a time.
- Photo requirements: The photo is less than 100 KB and meet the requirements of Dahua face database.
- PhotoData: The photo must be processed by base64 and the head data must be removed.

  Example: After xxx.jpg is processed by base64, the result is shown as Figure 1.

  Remove:
```

Figure 1

```
/9j/4AAQSkZJRgABAQAAAQABAAD/2wBDAAEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEE
```
```

Figure 2

- For details, see the protocol.

4.12.1.4.3 Updating Face Information

URL: POST http://192.168.1.108/cgi-bin/AccessFace.cgi?action=updateMulti

```
{
“FaceList”:[
{
"UserID": "102",
"PhotoData": [ "yyyy", "yyyy", … ],
" PhotoURL": [ "yyyy", "yyyy", … ],
},
….
]
  }
```

- To use this protocol, make sure that the device supports the second-generation access control capability.
- When updating face information, make sure that the personnel information of the UserID corresponding to the face has been sent.
- PhotoData and PhotoURL: When both PhotoData and PhotoURL are sent, PhotoData data shall prevail.

  If there is only PhotoURL, it will go to the URL to download data when the network connection is smooth.
- Information of up to10 faces can be updated at a time.
- Photo requirements: The photo is less than 100 KB and meet the requirements of Dahua face database.
- PhotoData: The photo must be processed by base64 and the head data must be removed.

  Example: After xxx.jpg is processed by base64, the result is shown as Figure 1.

  Remove:
```

Figure 1

```
/9j/4AAQSkZJRgABAQAAAQABAAD/2wBDAAEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEE
```

Figure 2
```

- For details, see the protocol.

4.12.1.4.4 Deleting All Face Information

URL: http://192.168.1.108/cgi-bin/AccessFace.cgi?action=removeAll

- To use this protocol, make sure that the device supports the second-generation access control capability.
- For details, see the protocol.

4.12.1.4.5 Deleting Information of Multiple Faces

URL:http://192.168.1.108/cgi-bin/AccessFace.cgi?action=removeMulti&UserIDList[0]=101&UserIDList[1]=102

- To use this protocol, make sure that the device supports the second-generation access control capability.
- You can delete information of 10 cards at a time.
- UserID is the prerequisite, otherwise the deletion will fail.
- For details, see the protocol.

4.12.1.4.6 Searching for Information of Multiple Faces

URL: http://192.168.1.108/cgi-bin/AccessFace.cgi?action=list&UserIDList[0]=1&UserIDList[1]=2

- To use this protocol, make sure that the device supports the second-generation access control capability.
- You can search for information of up to 10 faces at a time.
- PhotoData: The results processed by base64
- UserID: UserID corresponding to faces
- For details, see the protocol.

4.12.1.4.7 Searching for Face Information by Conditions

Main purposes of this scene;

1. Search for all face information.

2. Search for qualified face information by filtering conditions.

3. When searching by condition, only MD5 value (unique label) of the face database can be obtained. Face database cannot be obtained.

- **Start searching**

URL: http://192.168.1.108/cgi-bin/AccessFace.cgi?action=startFind&Condition.UserID=1

- To use this protocol, make sure that the device supports the second-generation access control capability.

- Set the filter condition, and the return value is the number of faces that match this condition.

  In the example, it means UserID=1.
- If there is no condition, search all data.
- Token: The search token, used in dofind.
- Total: The number of qualified search results.
- Caps: The number of search entries supported at a time. the maximum is 1000.
- For details, see the protocol. All fields in the protocol can be used as search conditions.

  - **Find the face information**

URL http://192.168.1.108/cgi-bin/AccessFace.cgi?action=doFind&Token=1234&Offset=0&Count=20

- To use this protocol, make sure that the device supports the second-generation access control capability.
- Token: The token returned by startfind.
- Offset: The offset of this search.
- Count: The amount of data to be obtained at a time. The maximum value depends on the return Caps of startfind.
- Description of return value: The following are included in the Info.

```
[
        {
            "MD5": [ "0360B53DBEB1C61265D36CFD941D204D" ], //MD5 value of face database
            "UserID" : "5"    //UserID
        },
        ……
]
```

- For details, see the protocol.

  - **Stop searching**

URL: http://192.168.1.108/cgi-bin/AccessFace.cgi?action=stopFind&Token=1234

- To use this protocol, make sure that the device supports the second-generation access control capability.
- For details, see the protocol.

4.12.1.5 Fingerprint Information (Second-generation Protocol)

4.12.1.5.1 Sending Postman Cases by Fingerprints

View “Second-generation protocol_fingerprint operations) in Postman Case List.

4.12.1.5.2 Sending Fingerprint Information

URL:POST http://192.168.1.108/cgi-bin/AccessFingerprint.cgi?action=insertMulti

```
{
    “vecPackets“: [ "xxxx", "xxxx", … ],
“AccessFingerprints”:[
{
"UserID": "102",
    "FingerprintPacket" :
{
   "Length" : 810,
                "Count" : 3,
                "DuressIndex" : 2
}
}
]
    }
```

- To use this protocol, make sure that the device supports the second-generation access control capability.
- When sending fingerprints, make sure that the personnel information of UserID corresponding to fingerprint has been sent.
- You can only send the fingerprint data of one person at a time.
- VecPackets: Save fingerprint feature data and fingerprint packet array.
- FingerPrintPacket.Length: The length of the a single fingerprint packet
- FingerPrintPacket .DuressIndex: Specify a fingerprint number in the fingerprint packet array as duress fingerprint.

  For example, DuressIndex=1 means that the first fingerprint is a duress fingerprint.

  You can only set one fingerprint as the duress fingerprint at a time.
- For details, see the protocol.

4.12.1.5.3 Updating Fingerprint Information

URL:POST http://192.168.1.108/cgi-bin/AccessFingerprint.cgi?action=updateMulti

```
{
“vecPackets“: [ "xxxx", "xxxx", … ],
“AccessFingerprints”:[
```

```
{
"UserID": "102",
    "FingerprintPacket" :
{
  "Length" : 810,
                "Count" : 3,
                "DuressIndex" : 2
}
}
]
  }
```


- To use this protocol, make sure that the device supports the second-generation access control capability.
- Before updating fingerprint information, make sure that there is fingerprint information of UserID.
- You can only send the fingerprint data of one person at a time.
- VecPackets: Save fingerprint feature data and fingerprint packet array.
- FingerPrintPacket.Length: The length of the a single fingerprint packet
- FingerPrintPacket .DuressIndex: Specify a fingerprint number in the fingerprint packet array as duress fingerprint.

  For example, DuressIndex=1 means that the first fingerprint is a duress fingerprint.

  You can only set one fingerprint as the duress fingerprint at a time.
- For details, see the protocol.

4.12.1.5.4 Deleting All Fingerprint Information

URL: http://192.168.1.108/cgi-bin/AccessFingerprint.cgi?action=removeAll

- To use this protocol, make sure that the device supports the second-generation access control capability.
- For details, see the protocol.

4.12.1.5.5 Deleting Information of Multiple Fingerprints

URL: http://192.168.1.108/cgi-bin/AccessFingerprint.cgi?action=removeMulti&UserIDList[0]=101&UserIDList[1]=102

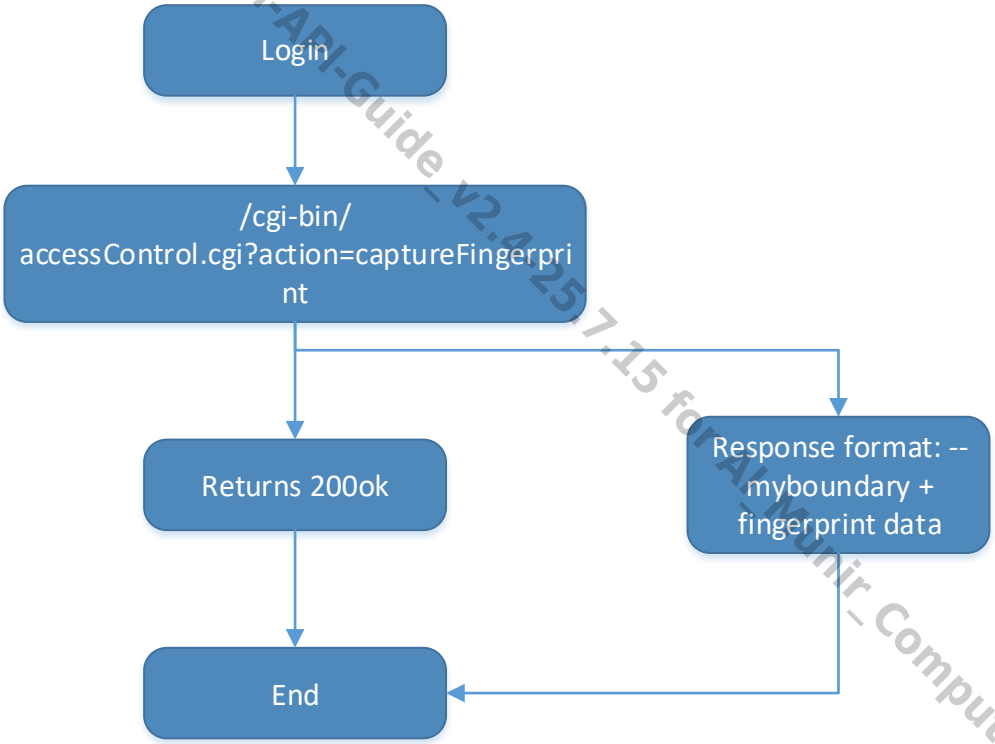
- To use this protocol, make sure that the device supports the second-generation access control capability.
- In UserIDList, UserID must exist, otherwise the deletion will fail.
- For details, see the protocol.

4.12.1.5.6 Searching for Fingerprint Information

URL: http://192.168.1.108/cgi-bin/AccessFingerprint.cgi?action=get&UserID=1

- To use this protocol, make sure that the device supports the second-generation access control capability.
- Before updating fingerprint information, make sure that there is fingerprint information of UserID.
- You can only send the fingerprint data of one person at a time.
- VecPackets: Save fingerprint feature data and fingerprint packet array.
- FingerPrintPacket.Length: The length of the a single fingerprint packet
- For details, see the protocol.

4.12.1.5.7 Fingerprint Acquisition



STEP 1: Login follows Dahua's standard CGI login procedure.

STEP 2: The client initiates fingerprint capture by accessing:
/cgi-bin/accessControl.cgi?action=captureFingerprint
and sets a timeout period.

Notes:

① If the command is successfully executed, the device will trigger the capture process and return "200 OK".

② The timeout period applies to the entire capture process (default: 4 minutes). If no fingerprint is successfully captured within this time, no fingerprint data will be reported.

STEP 3: The device enters fingerprint capture mode, requiring three scans of the same finger. The Dahua fingerprint algorithm processes the data to generate a fingerprint template (feature value), which is then reported.

Request URL	**http://<server>/cgi-bin/accessControl.cgi?action=captureFingerprint**			
Method	GET			
Request Params (key=value format in URL)				
Name	**Type**	**R/O**	**Description**	**Example**
info	object	R	Fingerprint Information	
+ReaderID	char[]	R	Card Reader ID	
heartbeat	int	R	Heartbeat Interval Unit：S	
timeout	int	R	Timeout Unit：S	
Request Example				
http://<server>/cgi-bin/accessControl.cgi?action=captureFingerprint&info.ReaderID=1&heartbeat=5&timeout=10				

Response Params (multipart , json format in body , Heartbeat in body)				
Name	**Type**	**R/O**	**Description**	**Example**
Response Example				
HTTP/1.1 200 OK **Cache-Control: no-cache** **Pragma: no-cache** **Expires: Thu, 01 Dec 2099 16:00:00 GMT** **Connection: close** **Content-Type: multipart/x-mixed-replace; boundary=myboundary** **--myboundary** **Content-Type: multipart/form-data** **Content-Length: 1272** **{** **"CollectResult" : true,** **"ErrorCode" : 0,** **"FingerprintData" : "XXX", < Fingerprint Algorithm Template Data >** **"FingerprintPacket" : {**				

```
"Count" : 1,        < Number of Fingerprints Collected >
"Length" : 810    < Fingerprint Data Length >
},
"RealUTC" : 1736393078
}
```


4.12.2 Access Control Cards and Passwords

This protocol is mainly used to open the door by password, or by both card and password.

4.12.2.1 Unlocking by Password in Second-generation Protocol

Sending User Information

- Password and UserID are sent together.

4.12.3 Administrator Passwords

There are two types of access control passwords. This chapter introduces the independent permission of administrator password. The operation record set is **AccessControlCustomPassword**, which is used to open the door with the administrator password.

4.12.3.1 Enabling Administrator Passwords

Enable administrator passwords, otherwise the administrator passwords will not be available.

For details, see 4.6.9 Enabling Administrator Password.

4.12.3.2 Adding Administrator Password

URL:http://192.168.1.108/cgi-bin/recordUpdater.cgi?action=insert&name=AccessControlCustomPassword&UserID=102&OpenDoorPassword=123456&Doors[0]=0&Doors[1]=1

- Enable administrator password, otherwise the administrator password will not be available.
- For example, Door[0]=1,and Door[1]=1. They represent personnel passages. If you do not enter the content, the passages will not be verified.

  In this example, the user can pass through channel 0 and channel 1.
- For other fields, see Protocols.

4.12.3.3 Searching for Administrator Passwords

URL: http://172.10.1.223/cgi-bin/recordFinder.cgi?action=find&name=AccessControlCustomPassword&condition.UserID=102& condition.RecNo=1&count=1

- Condition.UserID and condition.RecNo added as filter conditions can exist separately.
- condition.RecNo is the record set number, which can be obtained from the return value of Adding Administrator Password.

  During search, the recno corresponding to UserID will also be obtained.
- Count: The number of entries for one search.
- For other fields, see Protocols.

4.12.3.4 Changing Administrator Passwords

URL: http://172.10.1.223/cgi-bin/recordUpdater.cgi?action=update&name=AccessControlCustomPassword&recno=2&UserID=102&OpenDoorPassword=234567

- recno is required, which can be obtained by searching for the return value of administrator password.
- Either UserID or OpenDoorPassword parameter must exist, or the update will fail.
- For other fields, see Protocols.

4.12.3.5 Deleting Administrator Passwords

4.12.4.5.1 Deleting Administrator Passwords by Record Set Number

URL:http://172.10.1.223/cgi-bin/recordUpdater.cgi?action=remove&name=AccessControlCustomPassword&recno=4

- recno is required, which can be obtained by searching for the return value of administrator password.
- For other fields, see Protocols.

4.12.4.5.2 Deleting All Administrator Passwords

URL:http://172.10.1.223/cgi-bin/recordUpdater.cgi?action=clear&name=AccessControlCustomPassword

- For specific fields, see Protocols.

4.12.3.6 Obtaining Total Number of Administrator Password Records

URL:http://172.10.1.223/cgi-bin/recordFinder.cgi?action=getQuerySize&name=AccessControlCustomPassword

- For specific fields, see Protocols.

4.13 Intercom

4.13.1 Configuring SIP Server

4.13.1.1 Getting SIP Server Configuration

URL: http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=SIP

For details, see "5 Protocol Description".

4.13.1.2 Enabling SIP Server

URL: http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&SIP.UserType=2

For details, see "5 Protocol Description".

4.13.1.3 Configuring Server Port

URL: http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&SIP.SIPServerPort=5080

For details, see "5 Protocol Description".

4.13.1.4 Configuring SIP Server IP Address

URL:http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&SIP.SIPServer=192.168.1.111

For details, see "5 Protocol Description".

4.13.1.5 Configuring SIP Server Identity Authentication Code

URL:http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&SIP.AuthPassword=852369

For details, see "5 Protocol Description".

4.13.1.6 Configuring SIP Registration Domain

URL: http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&SIP.RegisterRealm=xxx

For details, see "5 Protocol Description".

4.13.1.7 Configuring SIP Alternative IP Address

URL:http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&SIP.SIPServerRedundancy=2156879

For details, see "5 Protocol Description".

4.13.1.8 Configuring Standby Server Password

URL:http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&SIP.SIPServerRedundancyPassWord=7894564

For details, see "5 Protocol Description".

4.13.1.9 Configuring Standby Server Username

URL:http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&SIP.SIPServerRedundancyUserName=159753

For details, see "5 Protocol Description".

4.13.1.10 Enabling Standby Server

URL: http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&SIP.IsMainVTO=1

For details, see "5 Protocol Description".

4.13.1.11 Configuring SIP Server Username

URL:http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&SIP.SIPServerLoginPWD=9512369

For details, see "5 Protocol Description".

4.13.1.12 Configuring SIP Server Password

URL:http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&SIP.SIPServerLoginUserName=admin

For details, see "5 Protocol Description".

4.13.2 Configuring Server Type

4.13.2.1 Getting Server Type

URL: http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=Registar

For details, see "5 Protocol Description".

4.13.2.2 Configuring Server Type

URL:http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&Registar[0].ServerType=H500 H500 is the private SIP server and VTO is the device.

For details, see "5 Protocol Description".

4.13.3 Configuring Local Settings

4.13.3.1 Configuring Device Type

URL: http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=VTOBasicInfo

For details, see "5 Protocol Description".

4.13.3.2 Configuring Device Type

URL: http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&VTOBasicInfo.Type=1

For details, see "5 Protocol Description".

4.13.3.3 Configuring ID

URL:http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&VTOBasicInfo.Number=8006

For details, see "5 Protocol Description".

4.13.4 Configuring VTO Floor

4.13.4.1 Getting the Building Number

URL: http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=Building

For details, see "5 Protocol Description".

4.13.4.2 Configuring Section Number

URL:http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&Building.SectionNumber=05

For details, see "5 Protocol Description".

4.13.4.3 Configuring Building Number

URL:http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&Building.BuildingNumber=899

For details, see "5 Protocol Description".

4.13.4.4 Configuring Unit Number

URL:http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&Building.BuildingUnitNumber=789For details, see "5 Protocol Description".

4.13.5 Device Management Information

4.13.5.1 Getting Devices Added to Intercom Device Management

URL: http://192.168.1.108/cgi-bin/recordFinder.cgi?action=find&name=VideoTalkContact

For details, see "5 Protocol Description".

4.13.5.2 Adding Devices

URL:http://192.168.1.108/cgi-bin/recordUpdater.cgi?action=insert&name=VideoTalkContact&VTNetAddress=127.0.0.1&VTHPassword=123456&Type=VTS&GroupNumber=-1&VTShortNumber=23657

Type field: VTS represents the main station, VTO represents the door station and VTH is the indoor monitor.

For details, see "5 Protocol Description".

4.13.5.3 Deleting Added Devices

URL:http://192.168.1.108/cgi-bin/recordUpdater.cgi?action=clear&name=VideoTalkContact

Clear all added devices.

URL:http://192.168.1.108/cgi-bin/recordUpdater.cgi?action=removeEx&name=VideoTalkContact&RecNo=2

Delete devices (RecNo is the number of records. You can view the records by using the command that is used to obtain the devices added to intercom device management. The URL is as follows:

URL:http://192.168.1.108/cgi-bin/recordFinder.cgi?action=find&name=VideoTalkContact

Obtain the devices added to intercom device management).

For details, see "5 Protocol Description".

4.13.5.4 Updating Records of Added Devices

URL:http://192.168.1.108/cgi-bin/recordUpdater.cgi?action=update&name=VideoTalkContact&recno=1&FirstName=&FamilyName=1

Update the records of the added devices (recno is the number of records, FirstName: name, FamilyName: last name).

For details, see "5 Protocol Description".

4.13.6 VTO Call Configuration Extension

4.13.6.1 Getting VTO Call Extension Configuration

URL:http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=VTOCallInfo

For details, see "5 Protocol Description".

4.13.6.2 Enabling Group Call

URL:http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&VTOCallInfo.GroupCallEnable=true

For details, see "5 Protocol Description".

4.13.6.3 Configuring Management Center Number

URL:http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&VTOCallInfo.ManagerNumber=89753

For details, see "5 Protocol Description".

4.13.6.4 Getting Transmission Mode Configuration

URL:http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=VTOCallInfo.MulticastEnable

For details, see "5 Protocol Description".

4.13.6.5 Configuring Transmission Mode

URL:http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&VTOCallInfo.MulticastEnable=false

Set the transmission mode to mode 2 (If it is set to true, mode 1 is used).

For details, see "5 Protocol Description".

4.13.7 Configuring Parameters Used When Devices are Added

URL:http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&DeviceLoginInfo[0].Password=789456&DeviceLoginInfo[0].LongNumber=3

When adding a device (VTO), set the password (LongNumber refers to the configured SIP number).

URL:http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&DeviceLoginInfo[0].Username=457897&DeviceLoginInfo[0].LongNumber=3

Set the username when adding a device (VTO) (LongNumber refers to the configured SIP number).

For details, see "5 Protocol Description".

4.13.8 Floor Extension Configuration

4.13.8.1 Getting Floor Extension Configuration

URL:http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=BuildingExternal

For details, see "5 Protocol Description".

4.13.8.2 Adding Indoor Monitors in Batches

You need to configure the following two together before you can add them in batches.

URL:http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&BuildingExternal.BeginNumber[0]=501&BuildingExternal.BeginNumber[1]=602&BuildingExternal.FloorCount=6&BuildingExternal.RoomCount=5When adding the indoor monitors in batches, set the room number on the first floor to 501, the room number on the second floor to 602, the number of unit floors to 6, and the number of rooms on each floor to 5.

URL:http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&Building.CreateRoom=true

Enable batch add.

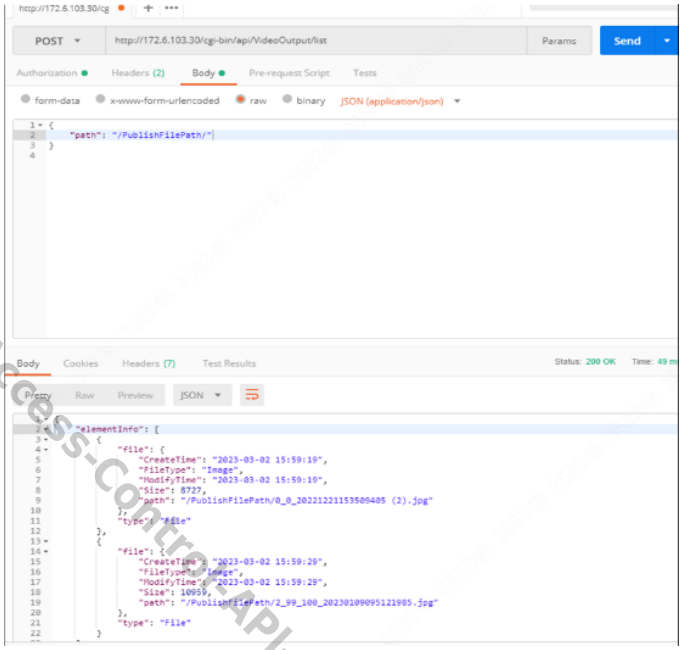
For details, see "5 Protocol Description".

4.14 Advertisement

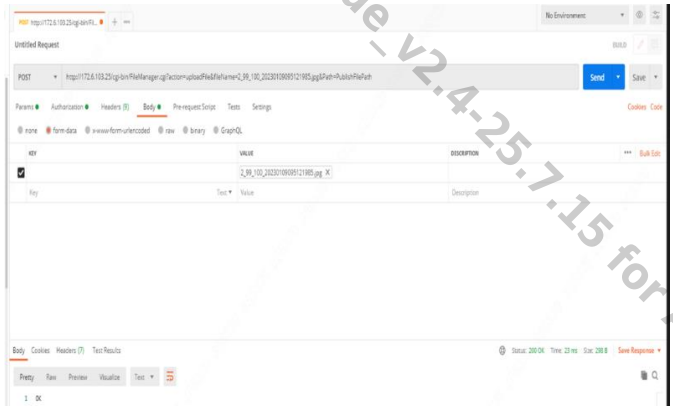
4.14.1 Getting Advertising Resource List

URL: http://192.168.1.108/cgi-bin/api/VideoOutput/list

- The request must be sent by using the postman tool. postman tool address: \\Pvs3552481713\9.13\CGI Test



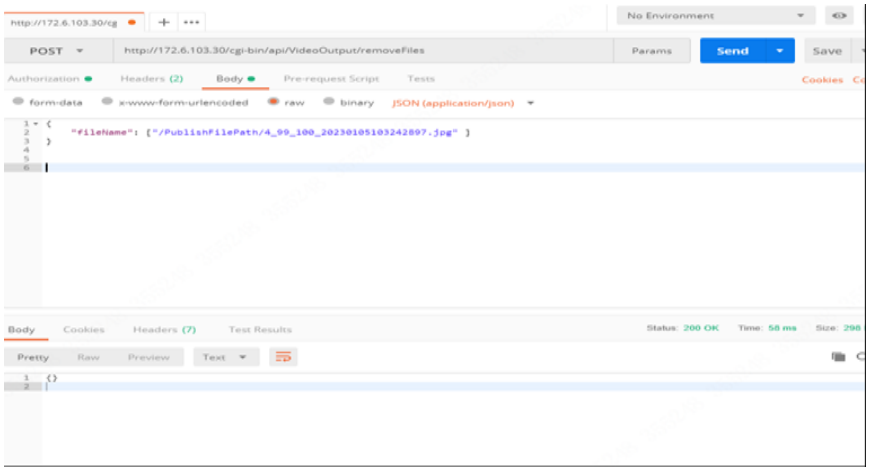
For details, see "5 Protocol Description".



4.14.2 Deleting Advertising Resource Files

URL: http://192.168.1.108/cgi-bin/api/VideoOutput/removeFiles

- Example of sending

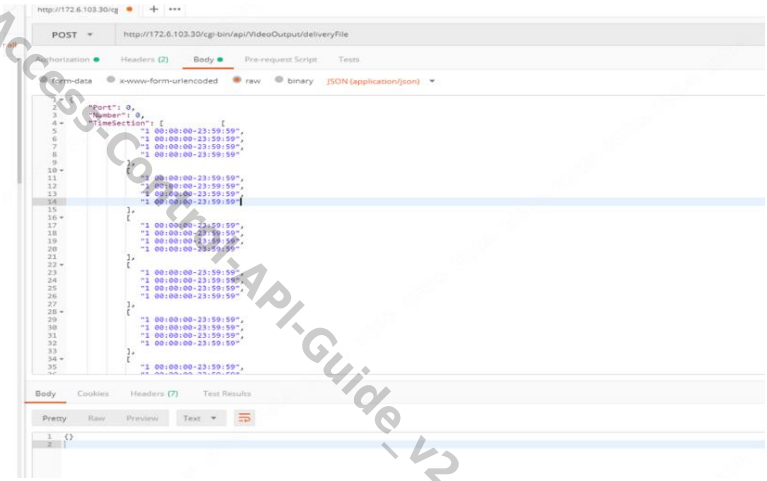


- For details, see "5 Protocol Description".

4.14.3 Sending Advertising Resources

URL: http://192.168.1.108/cgi-bin/api/VideoOutput/deliveryFile

- Example of sending:

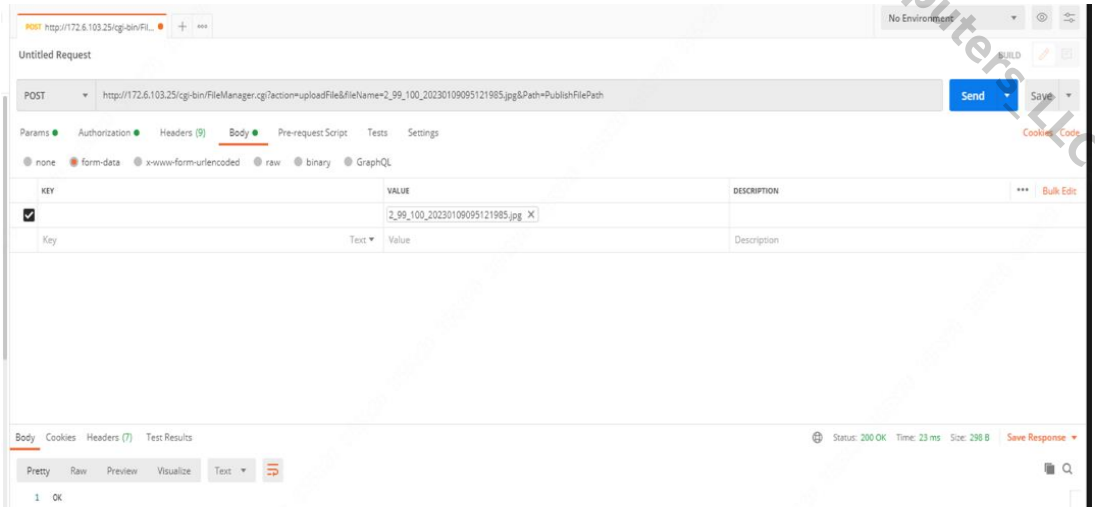


- For details, see "5 Protocol Description".

4.14.4 Uploading Advertising Resources

URL:http://192.168.1.108/cgi-bin/FileManager.cgi?action=uploadFile&fileName=2_99_1000_20230109095121985.jpg&Path=PublishFilePath

- Example of sending. fileName: The name of the image to be uploaded to the device. Path: The path for the image to be uploaded to the device. It is a relative path. It is similar to upload the image to the device's /mnt/appdata/Publish. Use the postman tool to select the local image file, that is, the file you want to upload.

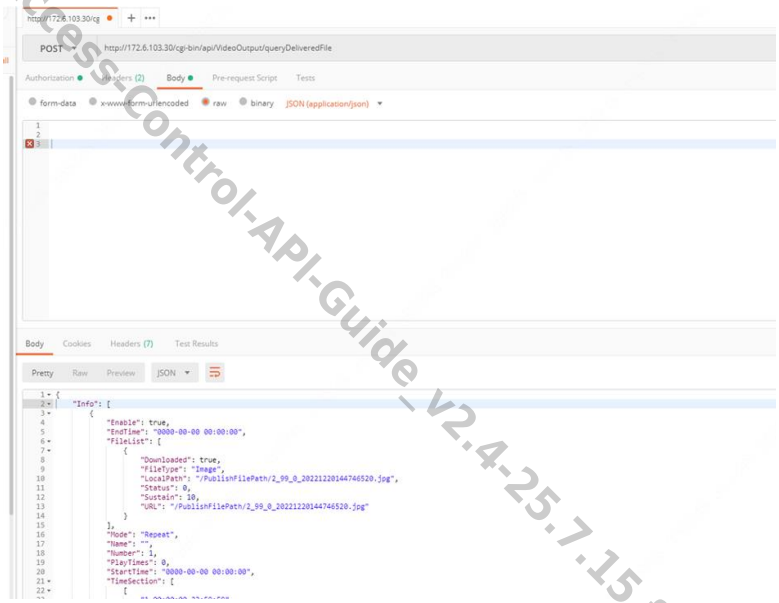


- For details, see Protocol Details.

4.14.5 Getting Advertising Files Sent to Devices

URL: http://192.168.1.108/cgi-bin/api/VideoOutput/queryDeliveredFile

- Example of sending



- For details, see "5 Protocol Description".

4.14.6 Configuring Parameters Such as Advertisement Setting Method

URL:http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&AccessDisplay.AccessDisplayObject=2&AccessDisplay.AdvertiseDisplay.AdvertisePicDisplay=1&AccessDisplay.AdvertiseDisplay.AdvertiseVideoDisplay=1&AccessDisplay.AdvertiseSplitType=1

Set the personalized content to the welcome words (AccessDisplay.AccessDisplayObject = 2. 0 represents advertisement; 2 means the welcome words. AccessDisplay.AdvertiseDisplay.AdvertiseVideoDisplay = 0: Configure how to display video advertisement in the split-screen mode. 0: Original scale; 1: Full screen. AccessDisplay.AdvertiseSplitList[0]. AdvertiseSplitID = 0: Configure how to display the image advertisements in the split-screen mode. 0: Original scale; 1: Full screen. AccessDisplay.AdvertiseSplitType = 1: Configure the advertisement split-screen mode.0: Do not split screen (Set as normal mode. Do not display advertisement); 1: Advertisement mode 1; 2. Advertisement mode 2; 3: Only uploaded advertisement images are displayed (the subject is not displayed).

For details, see "5 Protocol Description".

4.14.7 Configuring Welcome Words

4.14.7.1 Getting Welcome Words from Database

URL: http://192.168.1.108/cgi-bin/recordFinder.cgi?action=find&name=Announcement

For details, see "5 Protocol Description".

4.14.7.2 Configuring the Welcome Words

URL:http://192.168.1.108/cgi-bin/recordUpdater.cgi?action=insert&name=Announcement&Content=stringData&ExpirTime=2023-03-17 12:00:00&IssueTime=2023-03-16% 14:00:00&Title=guityuvdvhs&User=101&State=0&ReadFlag=0&BackgroundPicture=1

Configure the welcome words (insert into the database). The subtitle is stringData, the announcement expiration time is 2023-01-01%2012:00:00, the announcement release time is 2023-01-01%2012:00:00, the title is guityuvdvhs, the room number to release the announcement is 101, and the status of the announcement is {0: Initial state (not sent); 1: Already sent; 2: Expired}. Whether the announcement has been viewed {0: Unread;1: Read}. The background image is image 2. After the announcement is successfully sent, you will see that the advertisement words already exist on the webpage. You can manually click Apply.

For details, see "5 Protocol Description".

4.14.7.3 Clearing Welcome Words from Database

URL: http://192.168.1.108/cgi-bin/recordUpdater.cgi?action=clear&name=Announcement

For details, see "5 Protocol Description".

4.15 Safety Helmet

4.15.1 Getting Helmet Configuration

URL:http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=AccessControl[0].HelmetEnable

For details, see "5 Protocol Description".

4.15.2 Enabling Helmet Detection

URL:http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&AccessControl[0].HelmetEnable=true

Enable the helmet function (Only after you disable multi-face recognition, can the configuration be successful).

For details, see "5 Protocol Description".

4.16 Multi-face Recognition

4.16.1 Getting Multi-face Recognition Configuration

URL:http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=VideoAnalyseRule[0][0].Config.FaceWorkModel.FaceRecognizeModel

For details, see "5 Protocol Description".

4.16.2 Enabling Multi-face Recognition

URL:http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&VideoAnalyseRule[0][0].Config.FaceWorkModel.FaceRecognizeModel=2

For details, see "5 Protocol Description".

4.17 Doorbell

4.17.1 Getting Doorbell Configuration

URL: http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=DoorBell

For details, see "5 Protocol Description".

4.17.2 Enabling Doorbell

URL: http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&DoorBell.Enable=true

For details, see "5 Protocol Description".

4.17.3 Enabling Ringtone

URL:http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&DoorBell.RingBellEnable=true

For details, see "5 Protocol Description".

4.17.4 Configuring Playback Ringtone

URL:http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&DoorBell.RingBellConfig=1

For details, see "5 Protocol Description".

4.17.5 Configuring Alarm Linkage

URL:http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&DoorBell.AlarmLinkage=true

For details, see "5 Protocol Description".

4.17.6 Configuring Ringtone Time

URL:http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&DoorBell.RingBellTime=3

For details, see "5 Protocol Description".

4.18 Auto Upload

4.18.1 Auto Image Event Upload

URL: http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&PictureHttpUpload.Enable=true&PictureHttpUpload.Type=digest&PictureHttpUpload.UploadServerList[0].Address=192.168.1.208&PictureHttpUpload.UploadServerList[0].Port=80&PictureHttpUpload.UploadServerList[0].UserName=abc&PictureHttpUpload.UploadServerList[0].Password=123&PictureHttpUpload.UploadServerList[0].Uploadpath=/example/handlepic.php&PictureHttpUpload.UploadServerList[0].EventType[0]=AccessControl&PictureHttpUpload.UploadServerList[0].EventType[1]=FaceDetection&PictureHttpUpload.UploadServerList[0].rall=3

For specific fields, see Protocol Details.

- Currently only access control unlock event5.11.1Access Control Unlock Eventis

supported.

- Regarding response: The third-party platform must respond after receiving the uploaded data in order that the device continue to push data normally.

Response

Parameter Format	OK at body		
Parameter	**Type**	**Required**	**Description**
Name	Type	O	Param Description
Complete Example			
HTTP/1.1 200 OK Connection: keep-alive CONTENT-LENGTH: 0			

4.18.2 Auto Event Upload

URL: http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&EventHttpUpload.Enable=true&EventHttpUpload.Type=digest&EventHttpUpload.UploadServerList[0].Address=192.168.1.208&EventHttpUpload.UploadServerList[0].Port=80&EventHttpUpload.UploadServerList[0].UserName=abc&EventHttpUpload.UploadServerList[0].Password=123&EventHttpUpload.UploadServerList[0].Uploadpath=/example/handleevt.php&EventHttpUpload.UploadServerList[0].EventType[0]=AccessControl&EventHttpUpload.UploadServerList[0].EventType[1]=FaceDetection

For specific fields, see Protocol Details.

- Currently the door status event is supported and the access control event is supported when the device does not support the snapshot function.
- Regarding response: The third-party platform must respond after receiving the uploaded data in order that the device continue to push data normally.

Response

Parameter Format	OK at body		
Parameter	**Type**	**Required**	**Description**
Name	Type	O	Param Description
Complete Example			
HTTP/1.1 200 OK Connection: keep-alive CONTENT-LENGTH: 0			

4.18.3 General Information Upload

URL:http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&HttpPushGeneral.Enable=true&HttpPushGeneral.UploadServerList[0].Address=192.168.1.108&HttpPushGeneral.UploadServerList[0].AuthEnable=false&HttpPushGeneral.UploadServerList[0].Enable=false&HttpPushGeneral.UploadServerList[0].HttpsEnable=false&HttpPushGeneral.UploadServerList[0].Password=admin123&HttpPushGeneral.UploadServerList[0].Port=80&HttpPushGeneral.UploadServerList[0].Type[0].PushType=UserManagerInfor&HttpPushGeneral.UploadServerList[0].Type[0].Uploadpath=/&HttpPushGeneral.UploadServerList[0].UserName=admin

- For specific fields, see Protocol Details.
- Currently the access control only supports person information report. Deleting person information does not trigger a push notification whereas deletion of fingerprints, cards, passwords or faces will trigger a push notification.
- Personnel information upload format.
- Regarding response: The third-party platform must respond after receiving the uploaded data in order that the device continue to push data normally.

Response

Parameter Format	OK at body		
Parameter	**Type**	**Required**	**Description**
Name	Type	O	Param Description
Complete Example			
HTTP/1.1 200 OK Connection: keep-alive CONTENT-LENGTH: 0			

4.19 CGI Auto Registration

4.19.1 Auto Connection Device Interface

After the device initiates a connection to a third-party platform or software, the device regularly pushes the auto connection message to the third-party platform or software before the third-party platform or software is registered successfully. This message supports digital digest authentication. Whether the authentication is performed depends on the third party.

- For CGI auto registration configuration, see 4.6.11 Configuring CGI Auto Registration.

Request

Template	http://<server>/cgi-bin/api/autoRegist/connect		
Method	POST		
Parameter Format	Auto connection message of the device.		
Parameter	**Type**	**R/O**	**Description**
DevClass	char[64]	R	Device type
DeviceID	char[64]	R	Device ID
ServerIP	char[64]	R	Remote server IP
Complete Example			

http://192.168.1.108/ cgi-bin/api/autoRegist/connect
POST /cgi-bin/api/autoRegist/connect HTTP/1.1
Host: 172.32.0.141
Connection: keep-alive
CONTENT-LENGTH: 32
{
"DevClass": "",
"DeviceID": "",
"ServerIP": ""
}			

Response

Parameter Format	OK at body		
Parameter	**Types**	**R/O**	**Description**
Name	Type	O	Param Description
Complete Example			
HTTP/1.1 200 OK Connection: keep-alive CONTENT-LENGTH: 0			

4.19.2 Login Interface

After the device initiates a connection to a third-party platform or software, you need to log in to the third-party platform or software with the following information. Only after the login is successful, can other CGI messages be interacted. After the login is successful, the device returns the token used for subsequent authentication. This token is permanently valid during the connection. For subsequent commands, you can directly use the token returned from X-cgi-token field in the HTTP header to authenticate the data. After the authentication is successful, you do not need to authenticate other CGI messages in this connection.

Request

Template	http://<server>/cgi-bin/api/global/login

Method	POST		
Parameter Format	Login message		
Parameter	**Type**	**R/O**	**Description**
Complete Example			
http://192.168.1.108/cgi-bin/api/gloal/login POST /cgi-bin/api/global/login HTTP/1.1 Host: 172.32.0.141 Connection: keep-alive CONTENT-LENGTH: 0			

Response

Parameter Format	OK at body		
Parameter	**Type**	**Required**	**Description**
Name	Type	O	Param Description
Token	char[64]	R	Token value
Complete Example			

HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: <length>
Connection: keep-alive
{
"Token": "tGzv3JOkF0XG5Qx2TlKWIA",
}			

4.19.3 Heartbeat Interface

After the third-party platform or software successfully registers with the device, it needs to send heartbeat messages to the device regularly (30 s by default). If the device does not receive heartbeat messages for 3 times in a row, the connection is closed. The process needs to start again.

Request

Template	http://<server>/cgi-bin/api/global/keep-alive		
Method	POST		
Parameter Format	Heartbeat message		
Parameter	**Types**	**Required**	**Description**
Complete Example			
http://192.168.1.108/cgi-bin/api/global/keep-alive POST /cgi-bin/api/global/keep-alive HTTP/1.1 Host: 172.32.0.141 X-cgi-token: tGzv3JOkF0XG5Qx2TlKWIA Connection: keep-alive			

CONTENT-LENGTH: 0

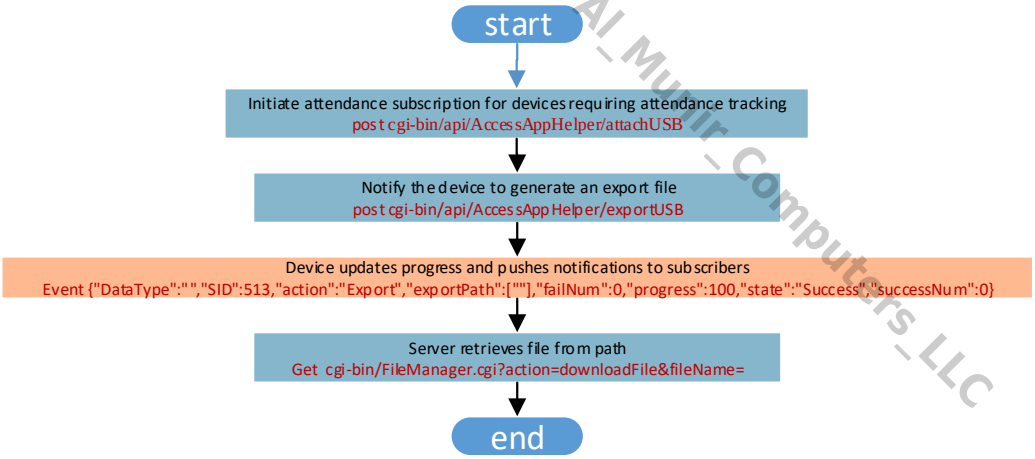
Response

Parameter Format	OK at body		
Parameter	**Type**	**Required**	**Description**
Name	Type	O	Param Description
Complete Example			
HTTP/1.1 200 OK Connection: keep-alive CONTENT-LENGTH: 0			

4.20 CGI Attendance Export

A third-party platform or software subscribes to the device (subscription must succeed to proceed with subsequent operations). After successful subscription, the device is notified to generate the file. Once the file is generated, the device returns the file path via the subscription link (the file must be retrieved within 5 minutes, otherwise it will be deleted). The third-party platform or software then obtains the generated file (multiple file paths require multiple retrievals). See the example operation below (using IP 192.168.0.1 as an example).

4.20.1 Attendance Export Process



STEP1: Server initiates attendance subscription to device: *cgi-bin/api/AccessAppHelper/attachUSB*. No parameters required). Success Response: HTTP 200 OK

STEP2: Command device to generate export file */cgi-bin/api/AccessAppHelper/exportUSB*

STEP3: The device will send a progress event to the subscription caller during the file generation process. The format is as follows.


```
{"DataType":"ShiftInfoXML","SID":513,"action":"Export","failNum":0,"progress":95,"state":"Runing","successNum":0}
```

```
{"DataType":"","SID":513,"action":"Export","exportPath":["\/ExportFilePath\/Attendance_Shift.xml"],"failNum":0,"progress":
100,"state":"Success","successNum":0}
```


Step 4: When the progress reaches 100% and the file path will be reported, call */cgi-bin/api FileManager.cgi?action=downloadFile* to download the corresponding file.

4.21.2 Subscription

URL: The http://192.168.0.1/cgi-bin/api/AccessAppHelper/attachUSB does not need to carry any parameters, and "HTTP 200OK" is returned successfully

Template	http://<server>/cgi-bin/api/AccessAppHelper/attachUSB			
Method	POST			
Parameter Format(JSON format in body)				
Name	**Type**	**R/O**	**Description**	**Example**
None				
Request Example				
{}				

Response Params (JSON format in body)				
Parameter Format	**Type**	**R/O**	**Description**	**Example**
SID	uint32	O	Returned subscription ID	123
Response Example				

{
"SID": 123
}				

4.21.3 Export File Generated Notification

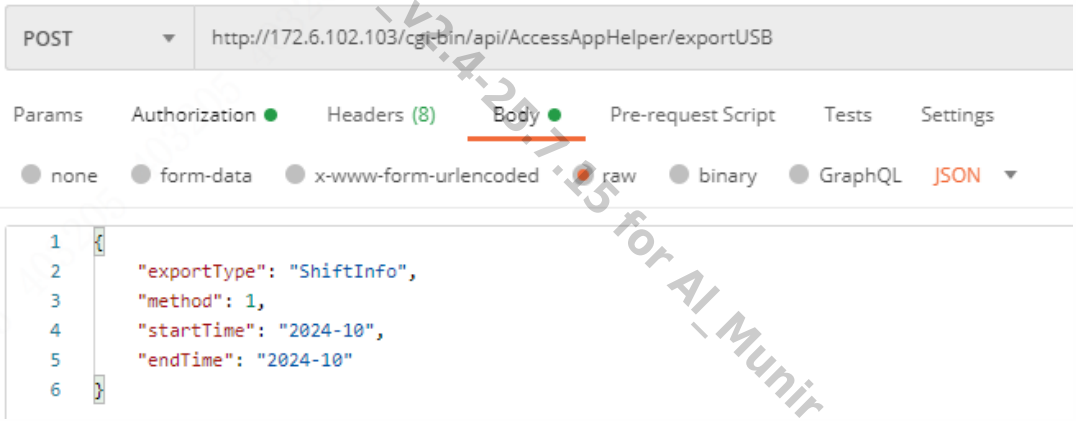
Request

Template	http://192.168.0.1/cgi-bin/api/AccessAppHelper/exportUSB		
Method	POST		
Parameter Format	JSON format in body		
Parameter	**Type**	**R/O**	**Description**

exportType	char[256]	Yes	Export Type
			Enumchar[32]{
			"ShiftInfo": Shift Data
			"UserShift": Shift Schedule Data
			"TotalAttenInfo": Monthly Attendance
			"AbnormalAttenInfo": Monthly Attendance Anomalies
			}

method	enumint	Yes	Export Method (Fixed Value=1): Subscription Export
startTime	char[20]	No	Export start time (optional)
endTime	char[20]	No	Export start time (optional)

Example



Response

Parameter Format	JSON format in body		
Parameter	**Type**	**R/O**	**Description**
taskID	uint32	No	Task ID（）
Example			
"taskID": 0			

After execution completes, wait for the attach subscription link to return exported data in the following format:

Parameter Format	JSON format in body		
Parameter	**Type**	**R/O**	**Description**

action	char[32]	No	Event Type " Export"
successNum	int32	No	Success Count
failNum	int32	No	Failure Count
progress	int32	No	Percentage Progress
state	char[32]	No	Result Status "Success/Fail/Runing"
exportPath	char[32][64]	No	Export File Path ["path1","path2","....."]

4.21.4 Downloading File Based on Export Path

URL: http://192.168.0.1/cgi-bin/FileManager.cgi?action=downloadFile&fileName=/ExportFilePath/Statistics_Report_10.xml

- FileName：Absolute file path on the device
- For detailed field descriptions, refer to the protocol specification.

4.21 Other Functions

4.21.1 Obtaining Video Streams

URL: rtsp://192.168.1.108:554/cam/realmonitor?channel=1&subtype=1

- For specific fields, see Protocols.

4.21.2 Capturing Video Images

URL: http://192.168.1.108/cgi-bin/snapshot.cgi?channel=1

- For specific fields, see Protocols.

4.21.3 Obtaining Files

URL:http://192.168.1.108/cgi-bin/FileManager.cgi?action=downloadFile&fileName=download.jpg

- FileName: The absolute path of the file in the device.
- For specific fields, see Protocols.

5 Protocol Description

For complete protocol, see *DAHUA_HTTP_API_ Protocol Standard*. The protocol contents in this document are excerpts of functions used for access control.

5.1 Device Management

5.1.1 Obtaining Device Types

Obtain the device type displayed externally (instead of the real type).

Request URL	http://<server>/cgi-bin/magicBox.cgi?action=getDeviceType			
Method	GET			
Request Params (none **)**				
Request Example				
http://192.168.1.108/cgi-bin/magicBox.cgi?action=getDeviceType				
Response Params (key=value format in body **)**				
Name	**Type**	**R/O**	**Description**	**Sample**
type	string	R	The displayed device model	DVR
Response Example				
type=DVR				

5.1.2 Obtaining the Hardware Version

Request URL	http://<server>/cgi-bin/magicBox.cgi?action=getHardwareVersion			
Method	GET			
Request Params (none **)**				
Request Example				
http://192.168.1.108/cgi-bin/magicBox.cgi?action=getHardwareVersion				
Response Params (key=value format in body **)**				
Name	**Type**	**R/O**	**Description**	**Sample**
version	string	R	The hardware version is in the format of xx.xx. Use two numbers to respectively represent the main and sub versions. If the beginning number of the main version is 0, it should be omitted.	1.00
Response Example				
version=1.00				

5.1.3 Obtaining the Device Serial Number

Request URL	http://<server>/cgi-bin/magicBox.cgi?action=getSerialNo
Method	GET
Request Params (none **)**	
Request Example	
http://192.168.1.108/cgi-bin/magicBox.cgi?action=getSerialNo	

Response Params (key=value format in body **)**				
Name	**Type**	**R/O**	**Description**	**Sample**
sn	string	R	Serial name	YZC0GZ05100020
Response Example				
sn=YZC0GZ05100020				

5.1.4 Obtaining the Device Name

Request URL	http://<server>/cgi-bin/magicBox.cgi?action=getMachineName			
Method	GET			
Request Params (none **)**				
Request Example				
http://192.168.1.108/cgi-bin/magicBox.cgi?action=getMachineName				
Response Params (key=value format in body **)**				
Name	**Type**	**R/O**	**Description**	**Sample**
name	string	O	Device name	my machine
Response Example				
name=my machine				

5.1.5 Obtaining the Device System Information

Request URL	http://<server>/cgi-bin/magicBox.cgi?action=getSystemInfoNew			
Method	GET			
Request Params (none **)**				
Request Example				
http://192.168.1.108/cgi-bin/magicBox.cgi?action=getSystemInfoNew				
Response Params (key=value format in body **)**				
Name	**Type**	**R/O**	**Description**	**Sample**
info	object	O	System information	
+SyncTime	object	O	Time synchronization parameter. hasRTC is the synchronization strategy of false. Or although RTC is true, it might also need	

			additional synchronization when battery is dead.	

++Strategy	enumchar[16]	R	Time synchronization strategy
			enumchar[16]{
			"None"
			"PerLogin": synchronize the time every time you log in
			}
				PerLogin

+2DCode	char[32]	O	Security code. 0–9; A–Z (uppercase)	123456
+TotalRunTime	uint64	O	The total running time of the device. Unit: s	12456
+cameraNum	uint8	O	The number of cameras	1
+cardReader	bool	O	Whether card swiping is supported. true: supported false: not supported	true
+flashID	uint8[8]	O	flash ID	[200,209,128,149,64,127,127,200]
+hasRTC	bool	O	Whether the RTC chip is included (for recording the system time) If the field does not exist, the default value is true, which means that the RTC is included.	true

Response Example

```
info.SyncTime.Strategy="PerLogin",
info.2DCode="123456",
info.TotalRunTime=123456
info.cameraNum=0
info.cardReader=false
info.flashID[0]=200
info.flashID[1]=209
info.flashID[2]=128
info.flashID[3]=149
info.flashID[4]=64
info.flashID[5]=127
info.flashID[6]=127
info.flashID[7]=200
info.hasRTC=true
```

5.1.6 Getting Software Version Information

Syntax	http://<*server*>/cgi-bin/magicBox.cgi?action=getSoftwareVersion
Method	GET
Description	Get software version information.
Example	http://192.168.1.108/cgi-bin/magicBox.cgi?action=getSoftwareVersion
Success Return	version=2.212.0000.0.R,build:2013-11-14
Comment	—

5.1.7 Resetting to Factory Settings

Syntax	http://<*server*>/cgi-bin/magicBox.cgi?action=resetSystemEx[&type=<***type***>]
Method	GET
Description	Reset the device to factory settings.
Example	http://192.168.1.108/cgi-bin/magicBox.cgi?action=resetSystemEx&type=0
Success Return	OK
Comment	Parameters in the URL: **Type**: Restoration type, with the value of 0 or 1. If it does not exist, the default value is 0. 0 means that all parameters are reset to the factory defaults. 1 means that the parameters except some specific parameters are reset to factory settings. The previous specific parameters vary with the device, while all of them contain information such as network configuration and user configuration. Therefore, after resetting the device to factory settings, you can still access the device with the same password and IP address.

5.1.8 Restarting the Device

Syntax	http://<*server*>/cgi-bin/magicBox.cgi?action=reboot[&delay=<*paramValue*>]
Method	GET
Description	Restart the device.
Example	http://192.168.1.108/cgi-bin/magicBox.cgi?action=reboot
Success Return	OK
Comment	If successful, return OK. If failed, return Error.

5.1.9 Shutting Down the Device

Syntax	http://<*server*>/cgi-bin/magicBox.cgi?action=shutdown
Method	GET
Description	Shut down the device.
Example	http://192.168.1.108/cgi-bin/magicBox.cgi?action=shutdown
Success Return	OK
Comment	If successful, return OK. If failed, return Error.

5.2 Log Management

5.2.1 Description of the Response Parameters

Field in Response	**Description**
found	The number of logs that are found. If the value is 0, no log has been found.
User	Username
Type	Log type
Time	Log time
RecNo	Log record number
Detail	Log details

5.2.2 Starting Searching for Logs

Syntax	http://<*server*>/cgi-bin/log.cgi?action=startFind&condition.StartTime=<***start***>&condition.EndTime=<***end***>[& condition.Type=<***type***>]
Method	GET
Description	Start searching for logs by conditions.
Example	Find log between 2011-1-1 12:00:00 and 2011-1-10 12:00:00, URL is: http://192.168.1.108/cgi-bin/log.cgi?action=startFind&condition.StartTime=2011-1-1 12:00:00&condition.EndTime=2011-1-10 12:00:00
Success Return	token=1 count=100
Comment	Parameters in the URL: ***start/end***: The start time and end time of the log, with format of: yyyy-mm-dd hh:mm:ss.

	In the field in response, token is used for acquiring the future log content. If the token is greater than 0, the log is found; otherwise, no log is found. ***Type***: Log type, data range { "System", "Config", "Event", "Storage", "Account", "Data", "File", "CourseRecord" }. ***count***: Number of logs that were found.

5.2.3 Acquiring Log Searching Results

Syntax	http://<*server*>/cgi-bin/log.cgi?action=doFind&token=<***TokenValue***>&count=<***logCount***>
Method	GET
Description	Get a certain number of logs.
Example	http://192.168.1.108/cgi-bin/log.cgi?action=doFind&token=1&count=100
Success Return	found=2 items[0].***RecNo***=789 items[0].***Time***=2011-05-20 11:59:10 items[0].***Type***=ClearLog items[0].***User***=admin items[1].***Detail***.Compression=H.264->MJPG items[1].***Detail***.Data=Encode items[1].***RecNo***=790 items[1].***Time***=2011-05-20 11:59:21 items[1].***Type***=SaveConfig items[1].***User***=System …
Comment	Parameters in the URL: ***TokenValue:*** The token value returned by calling startFind. ***logCount:*** Number of logs obtained this time, up to 100. For response parameters, see the following table.

5.2.4 Stopping Searching for Log

Syntax	http://<*server*>/cgi-bin/log.cgi?action=stopFind&token=<***TokenValue***>
Method	GET
Description	Stop searching for log.
Example	http://192.168.1.108/cgi-bin/log.cgi?action=stopFind&token=1
Success Return	OK
Comment	Parameters in the URL: ***TokenValue:*** The token value returned by calling startFind.

5.2.5 Backup Logs

Syntax	http://<*server*>/cgi-bin/Log.backup?action=All&condition.StartTime=<***startTime***>&condition.EndTime=<***endTime***>
Method	GET
Description	Download the logs in the specified periods and save them as a file, with default name of Log.Backup.
Example	http://192.168.1.108/cgi-bin/Log.backup?action=All&condition.StartTime=2014-8-25 00:02:32&condition.EndTime=2020-8-25% 01:02:32
Success Return	HTTP/1.1 200 OK CONTENT-LENGTH: 743087 CONNECTION: close Content-type: application/binarytet-stream; charset=utf-8 &w_User: default &Time: 2014-09-01 15:20:45 &Type: VideoLoss &Content: EventType: VideoLoss channel: <8> StartTime: 2014-09-01 15:20:45 …
Comment	Parameters in the URL: ***startTime/endTime***: The start time and end time of the log, with the format of: yyyy-mm-dd hh:mm:ss. Example: 2014-8-25 00:02:32 2020-8-25 01:02:32

5.3 Time Management

5.3.1 Getting the Current Time

Syntax	http://<*server*>/cgi-bin/global.cgi?action=getCurrentTime
Method	GET
Description	Get the current time.
Example	http://192.168.1.108/cgi-bin/global.cgi?action=getCurrentTime
Success Return	result = 2011-7-3 21:02:32

Comment	Time format is:"Y-M-D H-m-S". Time zone information are not included, please see SetLocalesConfig for more information.

5.3.2 Setting the Current Time

Syntax	http://<server>/cgi-bin/global.cgi?action=setCurrentTime&time=2011-7-3 21:02:32
Method	GET
Description	Set the current time.
Example	http://192.168.1.108/cgi-bin/global.cgi?action=setCurrentTime&time=2016-01-01 21:02:32
Success Return	OK
Comment	Time format is:"Y-M-D H-m-S". Time zone information are not included, please see SetLocalesConfig for more information.

5.3.3 Configuring DST Format

Appendix:

Parameter Name	Type	Description
Locales.DSTEnable	bool	Whether to enable the Daylight Saving Time (DST, daylight saving time).
Locales.DSTEnd.Day	integer	End date of DST: The value range is [0–6] or [1–31]. If Locales.DSTEnd.Week is 0, use the date of the month; otherwise, use the day of the week. [0–6]: Day of the week; 0 means Sunday and 6 means Saturday; [1–31]: Date of the month.
Locales.DSTEnd.Hour	integer	End time of DST: Hour, with the range of [0–23]
Locales.DSTEnd.Minute	integer	End time of DST: Minute, with the range of [0–59]
Locales.DSTEnd.Month	integer	End time of DST: Month, with the range of [1–12]

Parameter Name	Type	Description
Locales.DSTEnd.Week	Integer	End time of DST: Week, with the range of {1, 2, 3, 4, -1, 0}; 0 means the date of the month, instead of the day of week; [1, 2, 3, 4, -1] means week, 1 for the first week, 2 for the second week, 3 for the third week, 4 for the fourth week, and -1 for the last week.
Locales.DSTEnd.Year	Integer	End time of DST: Year, with the range of [2000–2038]
Locales.DSTStart.Day	Integer	Start time of DST, with the format similar to Locales.DSTEnd
Locales.DSTStart.Hour		
Locales.DSTStart.Minute		
Locales.DSTStart.Month		
Locales.DSTStart.Week		
Locales.DSTStart.Year		
Locales.TimeFormat	string	Defines the time format overlaid on the video, a string description. For example: *year-month-day hour*:mm:ss, and the positions of year, month, and day can be exchanged. The year format is {yy, yyyy}. Yy: Year without century. yyyy: Year with century. The month format is {M, MM, MMMM}. M = 1 means January. MM = 01 means January. MMMM = Jan means January. The date format is {d, dd}. d = 1 means the first day. dd = 01 means the first day. The time format is {H, HH, h, hh}. H = 1 and HH = 01 mean 1:00; the value range is 0–23. h = 1 and hh = 01 mean 1:00; the value range is 1–12. Example: yyyy-MM-dd HH:mm:ss or MM-dd-yyyy HH:mm:ss or dd-M-yy hh:mm:ss

5.3.4 Getting DST

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=getConfig&name=Locales
Method	GET
Description	Get locales configuration.
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=Locales
Success Return	table.Locales.DSTEnable=false table.Locales.DSTEnd.Day=1 table.Locales.DSTEnd.Hour=0 table.Locales.DSTEnd.Minute=0 table.Locales.DSTEnd.Month=1 table.Locales.DSTEnd.Week=2 table.Locales.DSTEnd.Year=2011 table.Locales.DSTStart.Day=0 table.Locales.DSTStart.Hour=0 table.Locales.DSTStart.Minute=0 table.Locales.DSTStart.Month=1 table.Locales.DSTStart.Week=1 table.Locales.DSTStart.Year=2011 table.Locales.TimeFormat=yyyy-MM-dd HH:mm:ss
Comment	—

5.3.5 Setting DST

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=setConfig&<***paramName***>=<***paramValue***>[&<***paramName***>=<***paramValue***>…]
Method	GET
Description	Set locales configuration.
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&Locales.DSTEnable=false
Success Return	OK
Comment	Set the DST according to Configuration instructions.

5.4 Event Subscription

5.4.1 General Event Subscription

Syntax	http://<*server*>/cgi-bin/eventManager.cgi?action=attach&codes=[<***eventCode***>,<***eventCode***>,…][&*keepalive* = 20][&heartbeat=<***Heartbeat***>]
Method	GET
Description	Subscribe to events
Example	http://192.168.1.108/cgi-bin/eventManager.cgi?action=attach&codes=[All]&heartbeat=5
Success Return	HTTP Code: 200 OK\r\n Cache-Control: no-cache\r\n Pragma: no-cache\r\n Expires: Thu, 01 Dec 2099 16:00:00 GMT\r\n Connection: close\r\n Content-Type: multipart/x-mixed-replace; boundary=<*boundary*>\r\n Body: --<*boundary*>\r\n Content-Type: text/plain\r\n Content-Length: <*data length*>\r\n <eventInfo>\r\n\r\n --<*boundary*>\r\n Content-Type: text/plain\r\n Content-Length: <*data length*>\r\n <eventInfo>\r\n\r\n For example: HTTP Code: 200 OK\r\n Cache-Control: no-cache\r\n Pragma: no-cache\r\n Expires: Thu, 01 Dec 2099 16:00:00 GMT\r\n Connection: close\r\n Content-Type: multipart/x-mixed-replace; boundary=myboundary\r\n\r\n Body: --myboundary\r\n Content-Type: text/plain\r\n Content-Length: 39\r\n Code=VideoMotion;action=Start;index=0\r\n\r\n --myboundary\r\n Content-Type: text/plain\r\n Content-Length: 38\r\n

	Code=VideoBlind;action=Start;index=0\r\n\r\n --myboundary\r\n Content-Type: text/plain\r\n Content-Length: 9\r\n Heartbeat\r\n\r\n --myboundary\r\n …
Comment	***eventCode***:A list of event codes, and “All” means all of the event codes. ***eventcode*** includes: VideoMotion: Motion detection events AlarmLocal: Local alarm events AccessControl: Unlocking events (optional), subscribe AI events to receive alarms. In the example, the general event data is: "Code=eventcode; action=Start; index=0", but some events will contain additional parameters "data", such as: "Code=eventcode; action=Start; index=0 ; data=datainfo" , where the format of datainfo is a JSON (JavaScript Object Notation) description. For details, see the individual description of each event. ***keepalive****:* Client keep-alive. If this parameter exists, the client needs to send keep-alive data to the device at an interval in seconds; the value range is [1–60], and the keep-alive data can be the string "keep alive". Note: It is recommended to use parameter of Heartbeat instead of parameter of keepalive. ***Heartbeat****:* Server keep-alive, integer, in seconds; the value range is [1,60]. For example, if URL comes with this parameter, and the value is 5, it means that the device should send a keep-alive message to the client every five seconds, and the keep-alive message is "Heartbeat". Note: The keep-alive message must be sent before the keepalive parameter expires.

5.4.2 Intelligent Event Subscription

Syntax	http://<*server*>/cgi-bin/snapManager.cgi?action=attachFileProc&Flags[0]=Event&Events=[<***eventCode***>,<***eventCode***>...][&channel=<***ChannelNo***>][&heartbeat=<***Heartbeat***>]
Method	GET
Description	Subscribe to snapshots. You can specify the snapshots of events in the eventcode.

Example	http://192.168.1.108/cgi-bin/snapManager.cgi?action=attachFileProc&Flags[0]=Event&Events=[AccessControl]&heartbeat=5
Success Return	--<*boundary*>\r\n Content-Type: text/plain\r\n Content-Length: <data length>\r\n Events[0].Code=TrafficJunction Events[0].CountInGroup=1 Events[0].IndexInGroup=1 Events[0].Lane=1 Events[0].Data.PTS= 42949485818.0 Events[0].TrafficCar.PlateNumber=Z A12345 Events[0].TrafficCar.DeviceAddress=Hangzhou …… Events[1].Code=TrafficJunction …… --<boundary> Content-Type: image/jpeg Content-Length:<image size> <JPEG image data> --<boundary> Content-Type: text/plain Content-Length:<data length> Heartbeat --<boundary>

Comment	***ChannelNo***: Video channel number, starting from 1. The default value 1 is used if not specified. ***Heartbeat****:* Heartbeat interval. The unit is second, the value range is [1,60], the default value is 5. When the device sends event data in the response, it will periodically send a heartbeat message to keep it alive according to the heartbeat interval. The content of the message is the string "Heartbeat". ***eventCode***: event code, including the following events: AccessControl: access control event CitizenPictureCompare: ID card comparison ***eventCode***:A list of event codes, and “All” means all of the event codes. Parameters in the response: ***GroupID***: Event group, integer, indicating the ID of the snapshot event group. ***CountInGroup***: The number of events in the event group, integer. ***IndexInGroup***: The sequence number of this event in the event group, an integer. For example, if CountInGroup is 3 and IndexInGroup is 1, it means that there are 3 events and snapshots in this event group, and this is the first event and snapshot.

5.5 File and Media

5.5.1 Getting Real-time Video Stream

The command to get real-time media stream uses the standard RTSP protocol. For details, see the RFC 2326 standard document. The default RTSP server port is 554. RTP transmission modes include RTP over UDP and RTP over RTSP. Authentic mode supports http digest, and the details of the authentic process is similar to the description in “3.5 user authentication”.

The format of the RTSP URL parameter to get real-time media stream is as follows.

URL	rtsp://<***server***>:[***port***]/cam/realmonitor			
Method	DESCRIBE, SETUP, PLAY, PAUSE, TEARDOWN, ...			
URL Params (key=value format in URL **)**				
Parameter Name	**Type**	**R/O**	**Description**	**Example**
channel	int	R	Video channel number, starting from 1.	1
subtype	int	R	Stream type: Main stream and sub stream. The number of sub streams can be obtained by the command in "4.5.1"	0

			Getting Maximum Number of Sub Streams". Optional values: 0: Main stream 1: Sub Stream 1 2: Sub stream 2	

URL Example
rtsp://192.168.1.108:554/cam/realmonitor?channel=1&subtype=0

Step 1 Obtain media description by executing the DESCRIBE command.

Request Example
DESCRIBE rtsp://192.168.1.108/cam/realmonitor?channel=1&subtype=0 RTSP/1.0 CSeq: 1 User-Agent: LibVLC/3.0.5
Response Example
RTSP/1.0 200 OK CSeq: 1 Server: Rtsp Server/3.0 Content-Base: rtsp://192.168.1.108/cam/realmonitor?channel=1&subtype=0/ Content-Length: xxx Content-Type: application/sdp v=0 o=- 2253484289 2253484289 IN IP4 0.0.0.0 s=Media Server c=IN IP4 0.0.0.0 t=0 0 a=control:* a=range:npt=now- m=video 0 RTP/AVP 98 a=control:trackID=0 a=framerate:25.000000 a=rtpmap:98 H265/90000 a=recvonly

Step 2 Establish a transmission channel for each medium by executing the SETUP command.

Take RTP over UDP as an example:

Establish the UDP socket for receiving and sending UDP packages on 63088 and 63089 interfaces.

Request Example
SETUP rtsp://192.168.1.108/cam/realmonitor?channel=1&subtype=0/trackID=0 RTSP/1.0 CSeq: 2

User-Agent: LibVLC/3.0.5
Transport: RTP/AVP;unicast;client_port=63088-63089

Response Example

RTSP/1.0 200 OK
CSeq: 2
Server: Rtsp Server/3.0
Session: 1546116282447;timeout=60
Transport: RTP/AVP/UDP;unicast;client_port=63088-63089;server_port=24764-24765;ssrc=71B0AFDC

Step 3 Execute the play command to play the media, and receive and send RTP and RTCP data through the UDP socket established in step 2.

Request Example

PLAY rtsp://192.168.1.108/cam/realmonitor?channel=1&subtype=0/ RTSP/1.0
CSeq: 3
User-Agent: LibVLC/3.0.5
Session: 1546116282447
Range: npt=0.000-

Response Example

RTSP/1.0 200 OK
CSeq: 3
Server: Rtsp Server/3.0
Session: 1546116282447
Range: npt=0.000-
RTP-Info: url=trackID=0;seq=45020;rtptime=1907404764

Step 4 Execute the TEARDOWN command to stop playing the media, and then disable the UDP socket.

Request Example

TEARDOWN rtsp://192.168.1.108/cam/realmonitor?channel=1&subtype=0/ RTSP/1.0
CSeq: 4
User-Agent: LibVLC/3.0.5
Session: 1546116282447

Response Example

RTSP/1.0 200 OK
CSeq: 4
Server: Rtsp Server/3.0
Session: 1546116282447

5.5.2 Downloading Files

Request URL	http://<server>/cgi-bin/FileManager.cgi?action=downloadFile			
Method	GET			
Parameter	Type	R/O	Description	Example
fileName	string	R	filename or path	download.jpg
Description	Get the files that can be downloaded from the device			
Example	http://192.168.1.108/cgi-bin/FileManager.cgi?action=downloadFile&fileName=download.jpg			
Success Return	HTTP/1.1 200 OK Content-type: text/plain;charset=utf-8 CONNECTION: close Set-Cookie:secure; HttpOnly CONTENT-LENGTH: <length> <Binary Data>			
Comment	—			

5.5.3 Getting Snapshot Image

Get snapshots of the specified video channel.

Request URL	http://<*server*>/cgi-bin/snapshot.cgi			
Method	GET			
Request Params (key=value format in URL **)**				
Parameter Name	**Type**	**R/O**	**Description**	**Example**
channel	int	O	Video channel number, starting from 1. The default value is 1	1
Request Example				
http://192.168.1.108/cgi-bin/snapshot.cgi?channel=1				

Response Params (binary in body **)**
<binary data>: images in JPEG format
Response Example
HTTP/1.1 200 OK Server: Device/1.0 Content-Type: image/jpeg Content-Length:<image size> <jpeg data>

5.6 Getting Records

5.6.1 Format of the Unlocking Records

Parameter Format	key=value format in body			
Parameter Name	**Type**	**R/O**	**Description**	**Example**
totalCount	integer	O	Total number of records found	1000
found	integer	O	Number of the returned record	100
records	array <object>	R	Records returned	
+RecNo	integer	R	Record number	12345
+CreateTime	integer	O	Card swiping time, UTC time	123456789
+CardNo	integer	R	Card number	12001
+CardName	string	O	Card name	ZhangSan
+CardType	integer	O	Card type (Only for card unlock) 0: General card 1: VIP card 2: Guest card 3: Patrol card 4: Blocklist card 5: Duress card 0xff is mother card	0
Password	string	O	Password	123456
+UserID	string	R	User ID	Zhang San

+Type	string	O	Event type:
			Enumchar [32]{
			"Entry": Entry
			"Exit": Exit
			}
				Exit

+Status	integer	O	Card swiping result:
			Enumint{
			0: Failed;
			1: Success
			}
				1

+Method	integer	R	Unlocking method: 0: By Password 1: By Card	1

			2: Use password after swiping card. 3: Swipe card after using password. 6: By Fingerprint 15: By (local) face recognition.	
+Door	integer	O	Door number Video talk devices do not support the field	5
+ReaderID	string	O	Card reader ID Video Talk devices do not support the field	
+ErrorCode	integer	O	Unlocking failure error code, which is valid when the Status is 0.	
+URL	string	O	Image URL, with length of up to 127. Video Talk devices do not support the field	
+IsOverTemperature	bool	O	Whether it is over-temperature	true
+TemperatureUnit	integer	O	Temperature unit (0: Celsius; 1: Fahrenheit; 2: Kelvin)	0
+CurrentTemperature	float	O	Body Temperature	36.8
+CitizenIDResult	bool	O	If the similarity is larger than or equal to the threshold, the person and ID card comparison is successful.	true
+CitizenIDName	string	O	Resident name	Zhang San
+CitizenIDNo	string	O	ID card number, 18 digits	342000000000000000

+CitizenIDSex	integer	O	Gender
			enumint8{
			0: Unknown
			1: Male
			2: Female
			9: Unspecified
			}
				1

+CitizenIDEthnicity	integer	O	Ethnic (Refer to the definition of the CitizenIDCard event)	1
+CitizenIDBirth	string	O	Date of birth	1980-01-01
+CitizenIDAddress	string	O	Address:	No.1199 Bin'an Road
+CitizenIDAuthority	string	O	Issued by	Hangzhou Public Security Bureau
+CitizenIDStart	string	O	Start date of validity period	1996-01-01
+CitizenIDEnd	string	O	End date of validity period. "Endless" means long-term validity.	2006-01-01

5.6.2 Getting the Unlocking Records

Template	http://<server>/cgi-bin/recordFinder.cgi?action=find&name=<RecordName>			
Method	GET			
Parameter Format	key=value format in URL			
Parameter Name	**Type**	**R/O**	**Description**	**Example**
name	string	R	The access control card and fingerprint record name of the user is fixed as "AccessControlCardRec".	"AccessControlCardRec"
count	integer	O	Maximum number returned, 1024 by default	100
StartTime	string	O	Start time of record creation	123456700
EndTime	string	O	End time of record creation	123456800
condition	object	O	Search conditions	
+CardNo	string	O	Card number	123456
Description	Get the unlocking records			
Example	http://192.168.1.108/cgi-bin/recordFinder.cgi?action=find&name=AccessControlCardRec&Start			

	Time=123456700&EndTime=123456800&condition.CardNo=12001&count=100
Success Return	totalCount=1000 // found=100 …. records[99].RecNo=12345 records[99].CreateTime=140556698 records[99].CardNo=12001 records[99].CardName=ZhangSan records[99].UserID=ZhangSan
Comment	—

5.6.3 Format of the Alarm Records

Parameter Format	key=value format in body			
Parameter Name	**Type**	**R/O**	**Description**	**Example**
totalCount	integer	R	Total number of records	200
found	integer	R	Number of the return record	100
records	Array<object>	R	Record	
+RecNo	integer	O	Record number	1234
+CreateTime	integer	O	Alarm time: UTC seconds	12345678
+UserID	string	O	User ID	1254
+EventCode	string	O	Alarm event type: Enumchar [32]{ DoorNotClosed: //Door is not closed. BreakIn: //Intrusion RepeatEnter: //Repeated entry Duress: //Duress AlarmLocal: //Local ChassisIntruded: //Dismantlement prevention MaliciousAccessControl: //Malicious unlocking event AccessControlBlocklist:// Blocklist alarm	AlarmLocal

			}	
+DevAddrs	integer	O	Sub-control ID: 0: Centralized controller or direct-connect device itself >0: Sub-controller	1
+IndexNum	integer	O	Channel number	0
+Time	string	O	Time of event occurrence (UTC with time zone and DST deviation)	16:00:01

5.6.4 Getting Alarm Records

The interface is suitable for access control devices.

Template	http://<server>/cgi-bin/recordFinder.cgi?action=find&name=AccessControlAlarmRecord[&StartTime=<startTime>&EndTime=<endTime>&count=<countNo>]			
Method	GET			
Parameter Format	key=value format in URL			
Parameter	**Type**	**R/O**	**Description**	**Example**
name	string	R	The record name is fixed as "AccessControlAlarmRecord".	AccessControlAlarmRecord
StartTime:	string	O	Start time, with format of: 2014-8-25%2000:01:32	2014-8-25%2000:01:32
EndTime	string	O	End time, with format of: 2014-8-25%2000:02:32	2014-8-25%2000:02:32
count	integer	O	Number of the return record	500
Description	Get alarm records			
Example	http://192.168.1.108/cgi-bin/recordFinder.cgi?action=find&name=AccessControlAlarmRecord&StartTime=2014-8-25 00:02:32&EndTime=2014-8-25% 01:02:32&count=500			
Success Return	totalCount=1000 found=500 records[0].RecNo=789 records[0].CreateTime=123456789			

	records[0].UserID=10113 records[0].EventCode=DoorMagnetism records[0].DevAddrs=1 records[0].IndexNum=0 records[0].Time=2017-05-10 16:00:01 …
Comment	—

5.7 Access Control

5.7.1 Unlocking the Door

Request URL	http://<server>/cgi-bin/accessControl.cgi?action=openDoor&channel=<ChannelNo>[&UserID=<UserID>&Type=<Type>]			
Method	GET			
Parameter Fomrat	key=value format in URL			
Parameter	Type	R/O	Description	Example
channel	integer	R	Channel number and Access Control number, starting from number 1	1
UserID	integer	O	User ID	101
Type	string	O	Unlocking method, "Remote" by default	Remote
Description	Unlocking command for access control products			
Example	http://192.168.1.108/cgi-bin/accessControl.cgi?action=openDoor&channel=1&UserID=101&Type=Remote			
Success Return	OK			
Comment	—			

5.7.2 Locking the Door

Request URL	http://<server>/cgi-bin/accessControl.cgi?action=closeDoor&channel=<ChannelNo>[&UserID=<UserID>&Type=<Type>]
Method	GET

Parameter Fomrat	key=value format in URL			
Parameter	Type	R/O	Description	Example
channel	integer	R	Channel number and Access Control number, starting from number 1	1
UserID	integer	O	User ID	101
Type	string	O	Unlocking method, "Remote" by default	Remote
Description	Locking command for access control products			
Example	http://192.168.1.108/cgi-bin/accessControl.cgi?action=closeDoor&channel=1&UserID=101&Type=Remote			
Success Return	OK			
Comment	—			

5.7.3 Getting Door Status

Request

Request URL	http://<server>/cgi-bin/accessControl.cgi?action=getDoorStatus&channel=<ChannelNo>			
Method	GET			
Parameter Format	key=value format in URL			
Parameter Name	**Type**	**R/O**	**Description**	**Example**
channel	integer	R	Channel number and Access Control number, starting from number 1	1
Request Example				
http://192.168.1.108/cgi-bin/accessControl.cgi?action=getDoorStatus&channel=1				

Response

Response Params (key=value format in body **)**				
Name	**Type**	**R/O**	**Description**	**Example**
Info	object	R	Door information	
+status	string	R	Status of the door. The value range are: {Open, Close}	Open
Response Example				
Info.status=Open				

5.8 General Access Control Configuration Instructions

5.8.1 Getting and Setting the Configuration

Get the configuration content of the specified name. For details on each configuration, see **Configuration** section.

Request URL	http://<*server*>/cgi-bin/configManager.cgi?action=getConfig			
Method	GET			
Request Params (key=value format in URL **)**				
Parameter Name	**Type**	**R/O**	**Description**	**Example**
name	string	R	Configuration name	SmartEncode
Request Example				
http://<*server*>/cgi-bin/configManager.cgi?action=getConfig&name=SmartEncode				

Response Params (key=value format in body**)**				
Parameter Name	**Type**	**R/O**	**Description**	**Example**
table	object	R	Configuration data object	
+<config name>	char[32]/object/object[]	R	The field name is the configuration name.The field value is the corresponding configuration data. For configuration details on each configuration name, see configuration item description.	
Response Example				
table.SmartEncode.Enable=true table.SmartEncode.Extra[0]=true table.SmartEncode.Extra[1]=false				

Set up content of the specified name. For details on each configuration, see **Configuration** section.

Request URL	http://<*server*>/cgi-bin/configManager.cgi?action=setConfig			
Method	GET			
Request Params (key=value format in URL **)**				
Parameter Name	**Type**	**R/O**	**Description**	**Example**
<config name>	string/object/object[]	R	The field name is the configuration name.	

			The field value is the corresponding configuration data. For details on each configuration, see Configuration section.	
Request Example				
http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&SmartEncode.Enable=true&SmartEncode.Extra[0]=true&SmartEncode.Extra[1]=false				

Response Params (OK in body **)**
Response Example
OK

Note: The following configurations only list the fields related to access control settings.

5.8.2 Access Time Schedule

Parameter name	**Type**	**R/O**	**Description**	**Example**
AccessTimeSchedule	object[]	O	Access control time configuration: (The custom password also uses this configuration subscript) An array, up to 128 groups of time configurations	
+TimeSchedule	TimeSection[7][4]	O	Schedule type: A two-dimensional array, with first seven elements corresponding to the seven days of each week, with up to four periods per day (Note: Among the seven elements, the first one is Sunday, the second one is Monday, and so on...)	
+Name	char[64]	O	Custom name	"xxxx"

+Enable	bool	O	Enable period	true
Example				
table.AccessTimeSchedule[0].Enable=false table.AccessTimeSchedule[0].TimeSchedule[0][0]=1 00:00:00-23:59:59 table.AccessTimeSchedule[0].TimeSchedule[0][1]=1 00:00:00-00:00:00 table.AccessTimeSchedule[0].TimeSchedule[0][2]=1 00:00:00-00:00:00 table.AccessTimeSchedule[0].TimeSchedule[0][3]=1 00:00:00-00:00:00 table.AccessTimeSchedule[0].TimeSchedule[1][0]=1 00:00:00-23:59:59 table.AccessTimeSchedule[0].TimeSchedule[1][1]=1 00:00:00-00:00:00 table.AccessTimeSchedule[0].TimeSchedule[1][2]=1 00:00:00-00:00:00 table.AccessTimeSchedule[0].TimeSchedule[1][3]=1 00:00:00-00:00:00 table.AccessTimeSchedule[0].TimeSchedule[2][0]=1 00:00:00-23:59:59 table.AccessTimeSchedule[0].TimeSchedule[2][1]=1 00:00:00-00:00:00 table.AccessTimeSchedule[0].TimeSchedule[2][2]=1 00:00:00-00:00:00 table.AccessTimeSchedule[0].TimeSchedule[2][3]=1 00:00:00-00:00:00 table.AccessTimeSchedule[0].TimeSchedule[3][0]=1 00:00:00-23:59:59 table.AccessTimeSchedule[0].TimeSchedule[3][1]=1 00:00:00-00:00:00 table.AccessTimeSchedule[0].TimeSchedule[3][2]=1 00:00:00-00:00:00 table.AccessTimeSchedule[0].TimeSchedule[3][3]=1 00:00:00-00:00:00 table.AccessTimeSchedule[0].TimeSchedule[4][0]=1 00:00:00-23:59:59 table.AccessTimeSchedule[0].TimeSchedule[4][1]=1 00:00:00-00:00:00 table.AccessTimeSchedule[0].TimeSchedule[4][2]=1 00:00:00-00:00:00 table.AccessTimeSchedule[0].TimeSchedule[4][3]=1 00:00:00-00:00:00 table.AccessTimeSchedule[0].TimeSchedule[5][0]=1 00:00:00-23:59:59 table.AccessTimeSchedule[0].TimeSchedule[5][1]=1 00:00:00-00:00:00 table.AccessTimeSchedule[0].TimeSchedule[5][2]=1 00:00:00-00:00:00 table.AccessTimeSchedule[0].TimeSchedule[5][3]=1 00:00:00-00:00:00 table.AccessTimeSchedule[0].TimeSchedule[6][0]=1 00:00:00-23:59:59 table.AccessTimeSchedule[0].TimeSchedule[6][1]=1 00:00:00-00:00:00 table.AccessTimeSchedule[0].TimeSchedule[6][2]=1 00:00:00-00:00:00 table.AccessTimeSchedule[0].TimeSchedule[6][3]=1 00:00:00-00:00:00 table.AccessTimeSchedule[1].Enable=false				

```
......
table.AccessTimeSchedule[127].Enable=false
table.AccessTimeSchedule[127].TimeSchedule[0][0]=1 00:00:00-23:59:59
table.AccessTimeSchedule[127].TimeSchedule[0][1]=1 00:00:00-00:00:00
table.AccessTimeSchedule[127].TimeSchedule[0][2]=1 00:00:00-00:00:00
table.AccessTimeSchedule[127].TimeSchedule[0][3]=1 00:00:00-00:00:00
table.AccessTimeSchedule[127].TimeSchedule[1][0]=1 00:00:00-23:59:59
table.AccessTimeSchedule[127].TimeSchedule[1][1]=1 00:00:00-00:00:00
table.AccessTimeSchedule[127].TimeSchedule[1][2]=1 00:00:00-00:00:00
table.AccessTimeSchedule[127].TimeSchedule[1][3]=1 00:00:00-00:00:00
table.AccessTimeSchedule[127].TimeSchedule[2][0]=1 00:00:00-23:59:59
table.AccessTimeSchedule[127].TimeSchedule[2][1]=1 00:00:00-00:00:00
table.AccessTimeSchedule[127].TimeSchedule[2][2]=1 00:00:00-00:00:00
table.AccessTimeSchedule[127].TimeSchedule[2][3]=1 00:00:00-00:00:00
table.AccessTimeSchedule[127].TimeSchedule[3][0]=1 00:00:00-23:59:59
table.AccessTimeSchedule[127].TimeSchedule[3][1]=1 00:00:00-00:00:00
table.AccessTimeSchedule[127].TimeSchedule[3][2]=1 00:00:00-00:00:00
table.AccessTimeSchedule[127].TimeSchedule[3][3]=1 00:00:00-00:00:00
table.AccessTimeSchedule[127].TimeSchedule[4][0]=1 00:00:00-23:59:59
table.AccessTimeSchedule[127].TimeSchedule[4][1]=1 00:00:00-00:00:00
table.AccessTimeSchedule[127].TimeSchedule[4][2]=1 00:00:00-00:00:00
table.AccessTimeSchedule[127].TimeSchedule[4][3]=1 00:00:00-00:00:00
table.AccessTimeSchedule[127].TimeSchedule[5][0]=1 00:00:00-23:59:59
table.AccessTimeSchedule[127].TimeSchedule[5][1]=1 00:00:00-00:00:00
table.AccessTimeSchedule[127].TimeSchedule[5][2]=1 00:00:00-00:00:00
table.AccessTimeSchedule[127].TimeSchedule[5][3]=1 00:00:00-00:00:00
table.AccessTimeSchedule[127].TimeSchedule[6][0]=1 00:00:00-23:59:59
table.AccessTimeSchedule[127].TimeSchedule[6][1]=1 00:00:00-00:00:00
table.AccessTimeSchedule[127].TimeSchedule[6][2]=1 00:00:00-00:00:00
table.AccessTimeSchedule[127].TimeSchedule[6][3]=1 00:00:00-00:00:00
```

5.8.3 Access Control

Parameter Name	**Type**	**R/O**	**Description**	**Example**
AccessControl	object[]	O	Access control configurations: An array, corresponding to each access control channel	
+Enable	bool	O	Enable: Whether to enable the configuration of this channel; true means enabling and false means disabling.	true
+SN	char[32]	O	Device serial number, such as wireless (smart lock): Read-only and non-configurable at the client	"1C03E08YAZ00020"
+Name	char[32]	O	Name	"Door1"
+State	enumchar[32]	O	Status: Enumchar [32]{ Normal: Normal. CloseAlways: Normally closed. OpenAlways: Normally open /*In the normally open and normally closed status, Opendoor cannot unlock the door.*/ NoPersonNC: Normally closed when there is no person, discarded. NoPersonNO: Normally open when there is no person, discarded.	"Normal"
+LocalControlEnable	bool	O	Enable local control:	true

			true means enabling and false means disabling.	
+RemoteControlEnable	bool	O	Enable remote control: true means enabling and false means disabling.	true
+SensorDetect	object	O		
++SensorDelay	uint8	O	Sensor output delay(s): If the delay exceeds this value, it is determined that at least one person exists. Unit: Second. 0–10	5
++HumanStatusSensitivity	uint	O	Human status detection sensitivity: This setting is to make sure that the controller continues to give the status of person existing within the preset time range when a curtain sensor is used and no one in the cabin moves, thus avoiding misjudgment of human status because no person in the cabin moves within a short time. Unit: Second. 0–300.	10
++DetectSensitivity	uint8	O	Sensor detection sensitivity, unit: %, the range is 0–100.	50
+Mode	enumchar[32]	O	Mode: Enumchar[32]{ HandProtected: Anti-pinch mode. SafeRoom: Safe room mode. Other Others:	"HandProtected"

+Method	uint8	O	Unlocking method. The order of verification types should be strictly followed 0: By password only 1: By swiping card only 2: By password or swiping card 3: Use password after swiping card 4: Swipe card after using password 5: Unlock by periods. It subject to the specific unlocking method under the TimeSchedule node in this method 6: By fingerprint only 7: By password or swiping card or fingerprint 8: Combination of swiping card + password + fingerprint 9: Combination of password + fingerprint 10: Combination of swiping card + fingerprint 11: By multiple persons 12: By face comparison of person and ID card (Card record set authorization is not R/O. Door is unlocked as long as the face matches the photo on the ID card) 13: ID card + person and ID card comparison 14: By person and ID card comparison or swiping card or fingerprint	2

			15: By (ID card + person and ID card comparison) or swiping card or fingerprint 16: UserID + password 17: By face only 18: Combination of face + password 19: Combination of fingerprint + password 20: Combination of fingerprint + face 21: Combination of card + face 22: By face or password 23: By fingerprint or password 24: By fingerprint or face 25: By card or face 26: By card or fingerprint 27: Combination of fingerprint + face + password 28: Combination of card + face + password 29: Combination of card + fingerprint + password 30: Combination of card + fingerprint + face 31: By fingerprint or face or password 32: By card or face or password 33: By card or fingerprint or face 34: Combination of card + fingerprint + face + password	

			35: By card or fingerprint or face or password 36: By (ID card + person and ID card comparison) or swiping card or face 37: By person and ID card comparison or swiping card (QR code) or face 38: (Card + password) | (fingerprint + password) 39: By person and ID card comparison (photo) or face 40: By person and ID card comparison (fingerprint) 41: By person and ID card comparison (photo + fingerprint) 42: By person and ID card comparison or swiping card or fingerprint or face or password, 2 by default 43: By multiple users 44: By person and ID card comparison or health code, 2 by default	
+RemoteCheck	bool	O	Whether platform verification is required: true means that platform verification is R/O to unlock after the permission verification is passed. false means that the door can be unlocked once the permission verification is passed.	true
+RemoteDetail	object	O	Used with RemoteCheck. It indicates whether to unlock the door or not after the preset device timeout period	

			expires if there is no response for remote verification.	
++TimeOut	uint	O	Timeout period: 0 means always waiting. Other values mean the timeout period(s).	0

++TimeOutDoorStatus	enumchar[32]	O	Door status after timeout:
			Enumchar[32]{
			"Open": Open
			"Close": Close
			}
				Open

+EnableMode	enumint	O	Enable level:
			Enumint{
			0: Effective at low level (power off and start).
			1: Effective at high level (power on and start).
			}
				1

+CloseDuration	int32	O	Door closing duration(s): The duration from the time when the door starts to be closed to the time when the door is completely closed, which is used for anti-pinch mode judgment	5
+OpenAlwaysTime	uint	O	Access control working period, with the value of the AccessTimeSchedule index	1
+CloseAlwaysTime	uint	O	Access control normally closed period, with the value of AccessTimeSchedule index	1
+HolidayTime	uint	O	Access control working period during holiday, with	1

			the value of the AccessTimeSchedule index	
+HolidayTimeGroup	uint[32]	O	[Customized to ION projects in Singapore. It is not supported for other projects. For similar requirements, use SpecialDayGroup, SpecialDaysSchedule.] Periods corresponding to holidays: An array, in which each element corresponds to the HolidayNo in the record set Holiday, and the value is a period index. There are up to 32 elements in the array.	[0,1,2,3,4]
+UnlockHoldInterval	uint	O	Unlocking NC/NO output holding time: In ms, the range is 250 ms–20000 ms.	2
+UnlockReloadInterval	int32	O	Unlocking command response interval: In ms.	15000
+AccessProtocol	enumchar[32]	O	Access control protocol: Enumchar[32]{ "Local": Local switch quantity control. Dahua": Dahua access control protocol (serial port protocol). "Remote": Dahua access control udp unlocking.}	Local
+ProtocolType	enumint	O	Functions of the serial port protocol: Valid only when AccessProtocol = Dahua. Enumint{	0

			0: Not used. 1: Dahua access control 485. 2: Lift control. 3: Long-distance reader. }	
+AccessControlUDP	object	O	Valid when AccessProtocol is Remote.	
++Address	char[64]	O		"0.0.0.0"
++Port	int32	O		10001
+EntranceLockChannel	uint32	O	Sub-channels under the access controller: Valid when AccessProtocol = Remote.	1
+CloseTimeout	uint	O	Locking timeout period: An alarm will be triggered if the door is not locked after the threshold time expires. 0 means no timeout detection.	10
+Handicap	object	O	Unlocking and locking parameters for other users.	
++UnlockHoldInterval	uint	O	Unlocking NC/NO output holding time: The unlocking holding time is inconsistent when other person passes and a normal person passes. Range: 250 ms–60000 ms.	3
++CloseTimeout	uint	O	Locking timeout period	15
+BreakInAlarmEnable	bool	O	Enable intrusion alarm.	true
+RepeatEnterAlarm	bool	O	Enable repeated entry alarm.	true

+DoorNotClosedAlarmEnable	bool	O	Enable alarm of not closed door.	true
+DuressAlarmEnable	bool	O	Enable duress alarm.	true
+TimeSchedule	object [ ][]	O	Unlock by periods: Valid when the unlocking mode is unlocking by periods. A two-dimensional array, with first seven elements corresponding to seven days of each week, with up to four periods per day. (Note: Among the seven elements, the first one is Sunday, the second one is Monday, and so on...) Up to four periods can be set per day.	
++TimeSection	TimeSection	O	Period: The format is hh:mm:ss-hh:mm:ss.	"00:00:00-24:00:00"
++Method	enumint	O	Unlocking method during the period: Enumint{ 0: By password only. 1: By swiping card only. 2: By password or swiping card. 3: Use password after swiping card. 4: Swipe card after using password. 5: Unlock by periods. 6: By fingerprint only.	2

			7: By password or swiping card or fingerprint. 8: Combination of password + swiping card + fingerprint. 9: Combination of password + fingerprint. 10: Combination of swiping card + fingerprint. 11: Unlock by multiple persons. 12: By ID card and face comparison. 17: By face only. 35: By card or fingerprint or face or password. } 2 by default	
+SnapshotEnable	bool	O	Enable event linkage snapshot: The unlocking event is only linked with snapshots, and the event source dynamically builds EventHandler during the notify event.	true
+SnapshotUpload	bool	O	Whether to upload snapshots	true
+SnapUploadPos	uint32	O	Snapshot upload address: For the upload address, see the NAS configuration item of Storage Manager in the Storage Manual, and the value corresponds to the NAS subscript.	0
+LockTongueEnable	bool	O	Enable bolt.	true

+SensorEnable	bool	O	Enable door detector.	true
+CloseCheckSensor	bool	O	Whether to detect the door detector before locking: true: After the unlocking holding time expires, the locking action cannot be restored until a valid door detector signal is detected. Conversely, if no valid door detector signal is detected after the unlocking holding time expires, the door is always unlocked. false (by default): The unlocking holding and locking recovery actions are directly carried out based on the preset unlocking holding time.	true
+FirstEnter	object	O	First card unlocking: During the specified period, other users can enter by swiping card (or fingerprint) only after the user with the first card permission passes the verification.	
++Enable	bool	O	Enable: true: Enable; false: Disable.	true
++Status	enumchar[32]	O	Door status: Enumchar[32]{ KeepOpen: The door is normally open after the first card permission verification is passed. Normal: Other users can enter by swiping card (or fingerprint) only after the first	"KeepOpen"

			card permission verification is passed. }	
++Time	uint	O	Verification period: The period during which first card verification is R/O, with the value of the AccessTimeSchedule index.	1
+AutoRemoteCheck	object	O	Auto remote unlocking	
++Enable	bool	O	Enable: true: Enable; false: Disable.	false
++Time	uint	O	Auto remote unlocking period, with the value of the AccessTimeSchedule index.	2
+ABLockRoute	int	O	AB interlock route (R/O for centralized controller): Corresponding to the index of AB interlock; -1 means invalid.	0
+DoorNotClosedReaderAlarmTime	uint	O	Time of card reader alarm after door sensor timeout. After a door sensor timeout alarm is generated, the controller notifies the front-end card reader, which will trigger a buzzer alarm (in seconds, 30 s by default).	30
+CustomPasswordEnable	Type bool.	O	Whether to enable custom password: "CustomPasswordEnable": true,	true
+RepeatEnterTime	uint	O	Repeated entry time: In seconds, the range is [0–180].	0

			0 means disabling.	

+CardNoConvert	enumint	O	Card number conversion:
			Enumint{
			0: Not R/O.
			1: Invert byte value conversion.
			2: Conversion by HIDpro.
			}
				0

+MaliciousAccessControlEnable	bool	O	Enable malicious unlocking event.	true
+FakeLockedAlarmEnable	bool	O	Enable fake locking alarm.	true
+ReadCardState	uint32	O	Whether the current door is in acquisition status: 0: Normal. 1: Acquisition card.	0
+HelmetEnable	bool	O	Whether to detect safety helmet.	false
Example				
table.AccessControl[0].AccessProtocol=Local table.AccessControl[0].AutoRemoteCheck.Enable=false table.AccessControl[0].AutoRemoteCheck.Time=255 table.AccessControl[0].BreakInAlarmEnable=false table.AccessControl[0].CardNoConvert=0 table.AccessControl[0].CloseAlwaysTime=255 table.AccessControl[0].CloseTimeout=60 table.AccessControl[0].CustomPasswordEnable=false table.AccessControl[0].DoorNotClosedAlarmEnable=false table.AccessControl[0].DuressAlarmEnable=true table.AccessControl[0].Enable=true table.AccessControl[0].FirstEnter.Enable=false table.AccessControl[0].FirstEnter.Status=Normal table.AccessControl[0].FirstEnter.Time=1				

table.AccessControl[0].HelmetEnable=false

table.AccessControl[0].HolidayTime=255

table.AccessControl[0].LockMode=2

table.AccessControl[0].Method=37

table.AccessControl[0].Name=Door1

table.AccessControl[0].OpenAlwaysTime=255

table.AccessControl[0].ProtocolType=0

table.AccessControl[0].ReadCardState=0

table.AccessControl[0].RemoteCheck=false

table.AccessControl[0].RemoteDetail.TimeOut=0

table.AccessControl[0].RemoteDetail.TimeOutDoorStatus=Close

table.AccessControl[0].RepeatEnterAlarm=false

table.AccessControl[0].RepeatEnterTime=0

table.AccessControl[0].SensorEnable=false

table.AccessControl[0].State=Normal

table.AccessControl[0].TimeSchedule[0][0].Method=35

table.AccessControl[0].TimeSchedule[0][0].TimeSection=00:00:00-23:59:59

table.AccessControl[0].TimeSchedule[0][1].Method=35

table.AccessControl[0].TimeSchedule[0][1].TimeSection=00:00:00-00:00:00

table.AccessControl[0].TimeSchedule[0][2].Method=35

table.AccessControl[0].TimeSchedule[0][2].TimeSection=00:00:00-00:00:00

table.AccessControl[0].TimeSchedule[0][3].Method=35

table.AccessControl[0].TimeSchedule[0][3].TimeSection=00:00:00-00:00:00

table.AccessControl[0].TimeSchedule[1][0].Method=35

table.AccessControl[0].TimeSchedule[1][0].TimeSection=00:00:00-23:59:59

table.AccessControl[0].TimeSchedule[1][1].Method=35

table.AccessControl[0].TimeSchedule[1][1].TimeSection=00:00:00-00:00:00

table.AccessControl[0].TimeSchedule[1][2].Method=35

table.AccessControl[0].TimeSchedule[1][2].TimeSection=00:00:00-00:00:00

table.AccessControl[0].TimeSchedule[1][3].Method=35

table.AccessControl[0].TimeSchedule[1][3].TimeSection=00:00:00-00:00:00

table.AccessControl[0].TimeSchedule[2][0].Method=35

table.AccessControl[0].TimeSchedule[2][0].TimeSection=00:00:00-23:59:59

table.AccessControl[0].TimeSchedule[2][1].Method=35

table.AccessControl[0].TimeSchedule[2][1].TimeSection=00:00:00-00:00:00
table.AccessControl[0].TimeSchedule[2][2].Method=35
table.AccessControl[0].TimeSchedule[2][2].TimeSection=00:00:00-00:00:00
table.AccessControl[0].TimeSchedule[2][3].Method=35
table.AccessControl[0].TimeSchedule[2][3].TimeSection=00:00:00-00:00:00
table.AccessControl[0].TimeSchedule[3][0].Method=35
table.AccessControl[0].TimeSchedule[3][0].TimeSection=00:00:00-23:59:59
table.AccessControl[0].TimeSchedule[3][1].Method=35
table.AccessControl[0].TimeSchedule[3][1].TimeSection=00:00:00-00:00:00
table.AccessControl[0].TimeSchedule[3][2].Method=35
table.AccessControl[0].TimeSchedule[3][2].TimeSection=00:00:00-00:00:00
table.AccessControl[0].TimeSchedule[3][3].Method=35
table.AccessControl[0].TimeSchedule[3][3].TimeSection=00:00:00-00:00:00
table.AccessControl[0].TimeSchedule[4][0].Method=35
table.AccessControl[0].TimeSchedule[4][0].TimeSection=00:00:00-23:59:59
table.AccessControl[0].TimeSchedule[4][1].Method=35
table.AccessControl[0].TimeSchedule[4][1].TimeSection=00:00:00-00:00:00
table.AccessControl[0].TimeSchedule[4][2].Method=35
table.AccessControl[0].TimeSchedule[4][2].TimeSection=00:00:00-00:00:00
table.AccessControl[0].TimeSchedule[4][3].Method=35
table.AccessControl[0].TimeSchedule[4][3].TimeSection=00:00:00-00:00:00
table.AccessControl[0].TimeSchedule[5][0].Method=35
table.AccessControl[0].TimeSchedule[5][0].TimeSection=00:00:00-23:59:59
table.AccessControl[0].TimeSchedule[5][1].Method=35
table.AccessControl[0].TimeSchedule[5][1].TimeSection=00:00:00-00:00:00
table.AccessControl[0].TimeSchedule[5][2].Method=35
table.AccessControl[0].TimeSchedule[5][2].TimeSection=00:00:00-00:00:00
table.AccessControl[0].TimeSchedule[5][3].Method=35
table.AccessControl[0].TimeSchedule[5][3].TimeSection=00:00:00-00:00:00
table.AccessControl[0].TimeSchedule[6][0].Method=35
table.AccessControl[0].TimeSchedule[6][0].TimeSection=00:00:00-23:59:59
table.AccessControl[0].TimeSchedule[6][1].Method=35
table.AccessControl[0].TimeSchedule[6][1].TimeSection=00:00:00-00:00:00
table.AccessControl[0].TimeSchedule[6][2].Method=35

table.AccessControl[0].TimeSchedule[6][2].TimeSection=00:00:00-00:00:00 table.AccessControl[0].TimeSchedule[6][3].Method=35 table.AccessControl[0].TimeSchedule[6][3].TimeSection=00:00:00-00:00:00 table.AccessControl[0].UnlockHoldInterval=2000

5.8.4 Special Days Schedule

The SpecialDaysSchedule configuration format is as follows:

Parameter Name	Type	R/O	Description	Example
SpecialDaysSchedule	array <object>	R	Holidays and festivals schedule: Each element represents a holiday schedule, and the maximum number of holiday schedules supported in each access control plan is determined by MaxSpecialDaysSchedules.	
+Name	string	O	Holiday schedule name	SpecialDayGroup1
+Enable	bool	R	Enable SpecialDaysSchedule	true
+GroupNo	integer	R	Holidays and festivals group ID: SpecialDayGroup configuration subscript	1
+TimeSection	array<string>	R	Periods: Number of periods per day, which is controlled by the capability set MaxTimePeriodsPerDay.	

+Doors	array<integer>	R	Doors that take effect during this holiday period	
Example				
SpecialDaysSchedule[0].Name=SpecialDayGroup1 SpecialDaysSchedule[0].Enable=true SpecialDaysSchedule[0].GroupNo=1 SpecialDaysSchedule[0].TimeSection[0]=1 00:00:00-12:00:00 SpecialDaysSchedule[0].TimeSection[1]=1 15:00:00-20:00:00 SpecialDaysSchedule[0].Doors[0]=2 SpecialDaysSchedule[0].Doors[1]=3 …				

5.8.5 Special Day Group

The SpecialDayGroup configuration format is as follows:

Parameter Name	Type	R/O	Description	Example
SpecialDayGroup	array <object>	R	Holidays and festivals group configuration: An array; each element represents a holidays and festivals group which is controlled by the capability set MaxSpecialDayGroups.	
+Name	string	O	Name of holidays and festivals group	Holidays and festivals group
+Enable	bool	R	Enable SpecialDayGroup	true
+Days	array <object>	O	Holidays and festivals The number of holidays and festivals supported in a holidays and festivals group is	

			determined by MaxDaysInSpecialDay Group.	
++SpecialDay Name	string	O	Holiday name	National Day
++StartTime	string	O	Start time of a holiday or festival	2017-10-01 00:00:00
++EndTime	string	O	End time of a holiday or festival	2017-10-07 23:59:59
Example				
SpecialDayGroup[0].Name=SpecialDayGroup1 SpecialDayGroup[0].Enable=true SpecialDayGroup[0].Days[0].SpecialDayName=NationalDay SpecialDayGroup[0].Days[0].StartTime=2017-10-01 00:00:00 SpecialDayGroup[0].Days[0]. EndTime =2017-10-07 23:59:59 …				

5.8.6 Wiegand

Parameter Name	Type	R/O	Description	Example
Wiegand	array <object>	R	Wiegand configuration: Array (considering that multiple Wiegand interfaces might exist in the device later)	

+Mode	integer	R	Working mode
			enumint{
			0: Wiegand input.
			1: Wiegand output.
			}
				1

+PulseWidth	integer	R	Pulse width, unit: us; the value range is related to and might vary with the connected external device.	200

+PulseStep	integer	R	Pulse interval, unit: us; the value range is related to and might vary with the connected external device.	1000

+TransferMode	integer	R	Transmission mode
			enumint{
			0: Wiegand 34-bit transmission, 4-byte card number, 2Bit verification;
			1: Wiegand 66-bit transmission, 8-byte card number, 2Bit verification;
			2: Wiegand 26-bit transmission, 3-byte card number, 2Bit verification
			}
				1

+OutType	integer	R	Output type:
			Enumint{
			0: Output ID.
			1: Output card number.
			}
				1

++InputType	integer	O	Received input type. InputType is an integer value of type int32. Each bit represents the mask value of a received input; all 0 means that no input is received. Bit0: Card number input. Bit1: Password input. Bit2–Bit31: Reserved.	0

++Doors	integer	O	Door number (used by unidirectional turnstile with double doors; if there is only one 485 interface and Wiegand interface, you need to set one Wiegand to Control Door 1). 0: Number of door 1. 1: Number of door 2. n: Number of door n.	0
Example				
table.Wiegand[0].Mode=1 table.Wiegand[0].PulseWidth=200 table.Wiegand[0].PulseStep=1000 table.Wiegand[0].TransferMode=1 table.Wiegand[0].OutType=1 table.Wiegand[0].InputType=3 table.Wiegand[0].Doors=0 table.Wiegand[1].Mode=1 …				

5.8.7 Access Configuration

Parameter Name	Type	R/O	Description	Example
AccessConfig	object	R	General access control device configuration.	
+PhotoGraph	uint32	R	Whether to enable the snapshot. If enable, every time the device will snapshot a picture from the system and save it. 0: No snapshot 1: Snapshot	1

+FingerprintPic	uint32	R	Whether to display the fingerprint each time the door was open.0:Not display 1: Display	1
+PrivacyMask	uint32	R	Privacy Masking of the video image, the video will be covered after the function was enabled. 0: Blocked 1: Low covering (The transparency is -30) 2: Middle covering (The transparency is -20) 3: High covering (The transparency is -10)	1
Example				
table.AccessConfig.FingerprintPic=0 table.AccessConfig.OpenDoorPic=1 table.AccessConfig.PhotoGraph=0 table.AccessConfig.PrivacyMask=0 table.AccessConfig.ShowPic=1				

5.8.8 Citizen Picture Compare Rule

Parameter Name	**Type**	**R/O**	**Description**	**Example**
CitizenPictureCompareRule	object	O	The threshold configuration of access control person and ID Card recognition	
+Threshold	uint8	O	Person and ID card comparison threshold: The range is [1–100].	60
Example				

table.CitizenPictureCompareRule.Threshold=60

5.8.9 Video Analyse Rule

Parameter Name	**Type**	**R/O**	**Description**	**Example**
VideoAnalyseRule	object [ ][]	O	Configuration of AI rules: A two-dimensional array, in which each element corresponds to an intelligent channel. The access control has only one intelligent channel.	[0][0]
Config	object	O	Face analysis, including face detection and face recognition.	
+FilterUnAliveEnable	bool	O	Whether to enable non-living filtering: The default value is false.	false
+ERDistThreshold	uint32	O	Pupillary distance filtering threshold: Pupillary distances less than the threshold are filtered. For image technology requirements of the face recognition application, see GB/T 35678-2017 . The value range is [50–infinite], and the default value is 0, which means no filtering.	50

+HelmetEnable	bool	O	Enable safety helmet detection. When it is enabled, the face recognition result will carry the information about safety helmet detection. The default value is false.	false
+TempSwitch	uint8	O	Select temperature monitoring status: 0: Disable temperature monitoring. 1: Normal temperature monitoring. 2: Debug temperature monitoring.	1
+TempModel	uint8	O	Temperature monitoring mode: Valid when TempSwitch is not 0: 0: Auto mode. 1: Heatmap detection mode. 2: Calibration mode.	0
+TempStrategy	object	O	Face temperature calculation strategy	
++TempValueMax	double	O	Maximum value of the normal face temperature interval	45.1
++TempValueMin	double	O	Minimum value of the normal face temperature interval	35.5
++TempType	uint8	O	Strategy type of face temperature calculation: 0: Use the maximum temperature; 1: Use the average temperature.	0

++SlideNum	uint8	O	Number of buffer frames for sliding: 0 means no sliding, and up to 32 frames can be buffered.	0
++HightTemp Strategy	object	O	Maximum temperature strategy parameter: Valid when tempType = 0.	
+++TempDete ctRegion	uint8	O	The region where the maximum temperature is calculated: 0: Forehead; 1: Whole face.	0
++AverageTe mpStrategy	object	O	Average temperature strategy parameter: Valid when TempType = 1.	
+++StrategyT ype	uint8	O	Average temperature strategy type: 0: Center point; 1: High temperature point.	0
+++PointNum	uint8	O	Number of points used to calculate the average: Square of an integer such as 4, 9, 16.	9
+++EnableFilt er	bool	O	Whether to enable the calculation of average by removing a maximum temperature and a minimum temperature. false: Disable. true: Enable.	false
+FilterMaskU nAliveEnable	bool	O	Whether to enable mask anti-fake: The value is strongly related to FilterUnAliveEnable,	false

			and this field is valid only when FilterUnAliveEnable is true (enable liveness detection).	

Example

table.VideoAnalyseRule[0][0].Class=FaceAnalysis

table.VideoAnalyseRule[0][0].Config.EyesDistThreshold=60

table.VideoAnalyseRule[0][0].Config.FeatureEnable=true

table.VideoAnalyseRule[0][0].Config.FeatureFilter=true

table.VideoAnalyseRule[0][0].Config.FeatureList[0]=Age

table.VideoAnalyseRule[0][0].Config.FeatureList[1]=Sex

table.VideoAnalyseRule[0][0].Config.FeatureList[2]=Glasses

table.VideoAnalyseRule[0][0].Config.FeatureList[3]=Emotion

table.VideoAnalyseRule[0][0].Config.FilterMaskUnAliveEnable=false

table.VideoAnalyseRule[0][0].Config.FilterUnAliveEnable=false

table.VideoAnalyseRule[0][0].Config.FlThreshold=50

table.VideoAnalyseRule[0][0].Config.HelmetEnable=true

table.VideoAnalyseRule[0][0].Config.MinQuality=50

table.VideoAnalyseRule[0][0].Config.SizeFilter.CalibrateBoxs[0].CenterPoint[0]=4096

table.VideoAnalyseRule[0][0].Config.SizeFilter.CalibrateBoxs[0].CenterPoint[1]=4096

table.VideoAnalyseRule[0][0].Config.SizeFilter.CalibrateBoxs[0].Ratio=1

table.VideoAnalyseRule[0][0].Config.SizeFilter.MaxSize[0]=8191

table.VideoAnalyseRule[0][0].Config.SizeFilter.MaxSize[1]=8191

table.VideoAnalyseRule[0][0].Config.SizeFilter.MinSize[0]=700

table.VideoAnalyseRule[0][0].Config.SizeFilter.MinSize[1]=700

table.VideoAnalyseRule[0][0].Config.SizeFilter.Type=ByLength

table.VideoAnalyseRule[0][0].Config.TempModel=1

table.VideoAnalyseRule[0][0].Config.TempStrategy.AverageTempStrategy.EnableFilter=false

table.VideoAnalyseRule[0][0].Config.TempStrategy.AverageTempStrategy.PointNum=9

table.VideoAnalyseRule[0][0].Config.TempStrategy.AverageTempStrategy.StrategyType=0

table.VideoAnalyseRule[0][0].Config.TempStrategy.HightTempStrategy.TempDetectRegion=0

table.VideoAnalyseRule[0][0].Config.TempStrategy.SlideNum=0 table.VideoAnalyseRule[0][0].Config.TempStrategy.TempType=0 table.VideoAnalyseRule[0][0].Config.TempStrategy.TempValueMax=45.100000 table.VideoAnalyseRule[0][0].Config.TempStrategy.TempValueMin=35.500000 table.VideoAnalyseRule[0][0].Config.TempSwitch=0 table.VideoAnalyseRule[0][0].Enable=true .....

5.8.10 Sign Light

Parameter Name	**Type**	**R/O**	**Description**	**Example**
SignLight	object [ ][]	O	Signature light An array indicates that a device has multiple signature lights. The number starts from 0.	[0][0]
+Mode	enumchar[32]	O	Light type enumchar[32]{ "Auto": automatic , automatically switched based on sensor "Timing": Timing mode "Off": Normal close "On": Normal open}	Auto
+TimeSections	char[6][20]	O	Lights on time sections, and this parameter is valid only in Timing mode. Up to six time sections are supported.	["00:00:00-24:00:00"
onCycle	Uint32	R	Range [0-100]	30
table.SignLight[0].Mode=Auto table.SignLight[0].TimeSections[0]=1 00:00:00-24:00:00 table.SignLight[0].TimeSections[1]=0 00:00:00-24:00:00				

table.SignLight[0].TimeSections[2]=0 00:00:00-24:00:00

table.SignLight[0].TimeSections[3]=0 00:00:00-24:00:00

table.SignLight[0].TimeSections[4]=0 00:00:00-24:00:00

table.SignLight[0].TimeSections[5]=0 00:00:00-24:00:00

table.SignLight[0].onCycle=30

5.8.11 Motion Detection

Parameter Name	**Type**	**Required**	**Description**	**Example**
MotionDetect	object[]	O	Motion detection configuration: The one-dimensional array, in which each element corresponds to a channel	
+Enable	bool	O	Enable motion detection: Only useful for Dahua devices, not supported by ONVIF.	true
+MotionDetectWindow	object[]	O	Video windows supported by motion detection: For third-generation motion detection, with four windows.	
++Threshold	uint8	O	Area threshold, with the range of [0–100].	50
++Sensitive	uint8	O	Sensitivity, with the range of [0–100].	50
++Region	int[]	O	Motion detection region blocks: An array, in which each row of the region is represented by a 32-bit integer; each bit of the integer corresponds to a block; the left side of the	[4194303, 3145728, …]

			screen corresponds to higher bits. Note: The correspondence between the rows and columns on the protocol and the coordinates of the input channel image blocks is as follows: Image columns: Left --> right. Corresponding protocol columns (bits): Left (higher bits) --> right (lower bits). Because motion detection only has 22 columns, the lower 22 bits and the higher 10 bits should be fixed as 0. Image rows: Up --> down; Corresponding protocol rows: Up --> down; third-generation motion detect field. It is invalid in the first-generation motion detection, and use the full screen Region field.	
++Id	int	O	Dynamic window ID: Integer, which is dynamically generated by the program and does not need to be reflected on the interface.	33
++Name	Char[256]	O	Dynamic window name: Window name.	"Region1"
Example				
..... table.MotionDetect[0].Enable=true ..... table.MotionDetect[0].MotionDetectWindow[0].Id=0 table.MotionDetect[0].MotionDetectWindow[0].Name=Region1				

table.MotionDetect[0].MotionDetectWindow[0].Region[0]=0

table.MotionDetect[0].MotionDetectWindow[0].Region[1]=0

table.MotionDetect[0].MotionDetectWindow[0].Region[2]=0

table.MotionDetect[0].MotionDetectWindow[0].Region[3]=524272

table.MotionDetect[0].MotionDetectWindow[0].Region[4]=524272

table.MotionDetect[0].MotionDetectWindow[0].Region[5]=524272

table.MotionDetect[0].MotionDetectWindow[0].Region[6]=524272

table.MotionDetect[0].MotionDetectWindow[0].Region[7]=524272

table.MotionDetect[0].MotionDetectWindow[0].Region[8]=524272

table.MotionDetect[0].MotionDetectWindow[0].Region[9]=524272

table.MotionDetect[0].MotionDetectWindow[0].Region[10]=524272

table.MotionDetect[0].MotionDetectWindow[0].Region[11]=524272

table.MotionDetect[0].MotionDetectWindow[0].Region[12]=524272

table.MotionDetect[0].MotionDetectWindow[0].Region[13]=524272

table.MotionDetect[0].MotionDetectWindow[0].Region[14]=524272

table.MotionDetect[0].MotionDetectWindow[0].Region[15]=524272

table.MotionDetect[0].MotionDetectWindow[0].Region[16]=0

table.MotionDetect[0].MotionDetectWindow[0].Region[17]=0

table.MotionDetect[0].MotionDetectWindow[0].Sensitive=54

table.MotionDetect[0].MotionDetectWindow[0].Threshold=66

table.MotionDetect[0].MotionDetectWindow[1].Id=1

table.MotionDetect[0].MotionDetectWindow[1].Name=Region2

table.MotionDetect[0].MotionDetectWindow[1].Region[0]=0

.....

table.MotionDetect[1].Enable=false

.....

table.MotionDetect[1].MotionDetectWindow[0].Id=0

table.MotionDetect[1].MotionDetectWindow[0].Name=Region1

table.MotionDetect[1].MotionDetectWindow[0].Region[0]=4194303

table.MotionDetect[1].MotionDetectWindow[0].Region[1]=4194303

table.MotionDetect[1].MotionDetectWindow[0].Region[2]=4194303

table.MotionDetect[1].MotionDetectWindow[0].Region[3]=4194303

table.MotionDetect[1].MotionDetectWindow[0].Region[4]=4194303

table.MotionDetect[1].MotionDetectWindow[0].Region[5]=4194303

table.MotionDetect[1].MotionDetectWindow[0].Region[6]=4194303
table.MotionDetect[1].MotionDetectWindow[0].Region[7]=4194303
table.MotionDetect[1].MotionDetectWindow[0].Region[8]=4194303
table.MotionDetect[1].MotionDetectWindow[0].Region[9]=4194303
table.MotionDetect[1].MotionDetectWindow[0].Region[10]=4194303
table.MotionDetect[1].MotionDetectWindow[0].Region[11]=4194303
table.MotionDetect[1].MotionDetectWindow[0].Region[12]=4194303
table.MotionDetect[1].MotionDetectWindow[0].Region[13]=4194303
table.MotionDetect[1].MotionDetectWindow[0].Region[14]=4194303
table.MotionDetect[1].MotionDetectWindow[0].Region[15]=4194303
table.MotionDetect[1].MotionDetectWindow[0].Region[16]=4194303
table.MotionDetect[1].MotionDetectWindow[0].Region[17]=4194303
table.MotionDetect[1].MotionDetectWindow[0].Sensitive=50
table.MotionDetect[1].MotionDetectWindow[0].Threshold=50
……

5.8.12 Temperature Monitoring

The configuration of Temperature Monitoring contains GuideModuleParam,

Parameter Name	Type	Required	Description	Example
MeasureTemperature	object	R	Temperature monitoring configuration	
+Enable	bool	R	Whether to enable temperature monitoring.	
+MaskOpt	uint	R	Mask mode: 0: Not detect mask. 1: Mask reminding mode. 2: Mask interception mode.	
+OnlyTemperatureMode	bool	O	Whether to enable the only Temperature monitoring mode.	false

+TemperatureDisplay	bool	R	Whether the temperature is displayed in the result prompt.	true
+TemperatureUnit	uint	O	Temperature unit 0: Celsius; 1: Fahrenheit.	0
+Type	integer	R	Temperature type 0: SCM IR. 1: Thermal imaging. 2: Amap temperature monitoring module. 3: Single-point wrist.	0
+GuideModuleParam	object	O	Parameter used in Amap Temperature monitoring module mode	
++Threshold	float	O	Temperature threshold (Celsius)	
++CalibrationModel	uint32	O	Thermal imaging calibration mode. The calibration parameters vary with the calibration mode. 0 : Indoor mode. 1 : Wall mount mode. 2 : Turnstile mode. 3 : High-end floor type mode.	
++Correct	Float	O	Temperature correction value (Celsius)	
++DebugModelEnable	Bool	O	Whether to enable the temperature monitoring debugging mode (with temperature monitoring data displayed on the top of the face box).	true

++HeatDisplayEnbale	bool	O	Whether to display the heatmap.	true
++MaxDistance	uint32	O	The allowable maximum distance for temperature monitoring (cm)	
++ProjectDebugModel	bool	O	Engineering debugging mode, which is used to start the blackbody debugging mode.	true
++RectEnable	Bool	O	Enable the display of the Temperature monitoring region box (which is displayed on the video stream interface).	
++TempRandReplaceThreshold	Float	O	Random temperature replacement threshold (A temperature lower than this threshold is randomly replaced by a valid temperature for compatibility error cases. If the threshold is 0, this function is not enabled).	
++ValidTemperatureLowerLimit	float	O	Lower limit of valid temperate: Temperatures lower than this value are invalid. (Celsius).	
+InfraredTemperatureParam	object	O	Parameter used in single-chip microcomputer IR mode.	
++Correct	float	O	Temperature correction value. (Celsius)	
++DebugModelEnable	bool	O	Whether to enable the Temperature monitoring debugging mode (with Temperature monitoring debugging data displayed on the screen).	

++MaxDistance	Uint32	O	Maximum test distance. (cm)	
++OverTemperature MaxDistance	Uint32	O	The maximum distance to report high temperature events, within which high temperature events detected should be directly reported. Distance for re-measurement (cm) means a distance within which high temperature is not detected and a prompt of getting closer for re-measurement is displayed. (Changed in order number 20200630012)	
++RectEnable	bool	O	Enable the display of the Temperature monitoring region box (which is displayed on the video stream interface).	
++RetentionTime	Uint32	O	IR temperature retention time (ms): The effective regression time for the person to get temperature from the IR device during device authentication.	
++SensorType	Uint32	O	Temperature monitoring module sensor type: "90641", "90640"	
++Threshold	Uint32	O	Temperature threshold (Celsius)	
++ValidTemperature LowerLimit	float	O	Lower limit of valid temperate: Temperatures	

			lower than this value are invalid (Celsius).	
+ThermalImagingParam	object	O	Parameter used in thermal imaging mode.	
++RetentionTime	Uint32	O	Thermal imaging temperature retention time (s): (If the face comparison fails to pass, verify the permission by swiping card or other methods; find the person base library, and then compare it with the thermal imaging device. This field indicates the temperature retention time).	
++Threshold	Uint32	O	Thermal imaging face comparison threshold.	
+WristTemperatureParam	Object	O	Parameter used in single-point wrist mode.	
++Correct	Float	O	Temperature correction value. (Celsius)	
++InvalidTemperatureDistance	Uint32	O	Invalid Temperature monitoring distance (cm): Temperatures measured at a distance larger than this value are considered as invalid and directly filtered. Meanwhile, this field is used with the ValidTemperatureDistance field. Temperatures measured between the valid and invalid distances are inaccurate and a prompt of getting closer should be displayed.	

++TemperatureTimeout	Uint32	O	Temperature monitoring timeout period (s).	
++Threshold	Float	O	Temperature threshold (Celsius).	
++ValidTemperatureDistance	Uint32	O	The measurement distance of valid temperature (cm): Temperatures measured at a distance less than this value are considered as valid.	
++ValidTemperatureLowerLimit	float	O	Lower limit of valid temperate: Temperatures lower than this value are invalid (Celsius).	36.0
+WristTemperatureParam	object	O	Parameter used in single-point wrist mode.	
++Threshold	float	O	Temperature threshold (Celsius).	37.30
++Correct	float	O	Temperature correction value (Celsius).	1.50
++ValidTemperatureLowerLimit	float	O	Lower limit of valid temperate: Temperatures lower than this value are invalid (Celsius).	36.0
++TemperatureTimeout	uint32	O	Temperature monitoring timeout period (s)	10
++ValidTemperatureDistance	uint32	O	The measurement distance of valid temperature (cm): Temperatures measured at a distance less than this value are considered as valid.	5
++InvalidTemperatureDistance	uint32	O	Invalid Temperature monitoring distance (cm): Temperatures measured at a distance larger than	30

			this value are considered as invalid and directly filtered. Meanwhile, this field is used with the ValidTemperatureDistance field. Temperatures measured between the valid and invalid distances are inaccurate and a prompt of getting closer should be displayed.	
++Compensation	float[50][2]	O	The corresponding temperature compensation value. A dimensional array, means when the environment temperature at [0u], value of [1u] need to be compensated. 50 groups of temperature can be set.	{{35，0.2}，{36，-0.5}.....}
++ModuleCompensationValue	float	O	Calibration value of module, used for the height module to do the ambient temperature calibration.	0.0

Example

table.MeasureTemperature.Enable=false

table.MeasureTemperature.ExternalReaderTemp[0]=0

table.MeasureTemperature.DHModuleParam.ValidTemperatureLowerLimit=35.000000

table.MeasureTemperature.GuideModuleParam.CalibrationModel=1

table.MeasureTemperature.GuideModuleParam.Compensation[0][0]=0.000000

….

table.MeasureTemperature.GuideModuleParam.Correct=0.000000

table.MeasureTemperature.GuideModuleParam.DebugModelEnable=false

table.MeasureTemperature.GuideModuleParam.EnvironmentTempContrast=16.000000

table.MeasureTemperature.GuideModuleParam.ErrorTempCountThreshold=3

table.MeasureTemperature.GuideModuleParam.HeatDisplayEnbale=false

table.MeasureTemperature.GuideModuleParam.MaxDistance=0

table.MeasureTemperature.GuideModuleParam.ModuleCompensationValue=0.000000

table.MeasureTemperature.GuideModuleParam.ProjectDebugModel=false

table.MeasureTemperature.GuideModuleParam.RectEnable=false

table.MeasureTemperature.GuideModuleParam.TempAdjustmentThreshold=37.300000

table.MeasureTemperature.GuideModuleParam.TempRandReplaceThreshold=0.000000

table.MeasureTemperature.GuideModuleParam.Threshold=37.300000

table.MeasureTemperature.GuideModuleParam.ValidTemperatureHighLimit=42.000000

table.MeasureTemperature.GuideModuleParam.ValidTemperatureLowerLimit=35.000000

table.MeasureTemperature.InfraredTemperatureParam.Compensation[0][0]=0.000000

table.MeasureTemperature.InfraredTemperatureParam.Compensation[0][1]=0.000000

table.MeasureTemperature.InfraredTemperatureParam.Compensation[1][0]=1.000000

table.MeasureTemperature.InfraredTemperatureParam.Compensation[1][1]=0.000000

table.MeasureTemperature.InfraredTemperatureParam.Compensation[2][0]=2.000000

table.MeasureTemperature.InfraredTemperatureParam.Compensation[2][1]=0.000000

table.MeasureTemperature.InfraredTemperatureParam.Compensation[3][0]=3.000000

table.MeasureTemperature.InfraredTemperatureParam.Compensation[3][1]=0.000000

table.MeasureTemperature.InfraredTemperatureParam.Compensation[4][0]=4.000000

table.MeasureTemperature.InfraredTemperatureParam.Compensation[4][1]=0.000000

table.MeasureTemperature.InfraredTemperatureParam.Compensation[5][0]=5.000000

table.MeasureTemperature.InfraredTemperatureParam.Compensation[5][1]=0.000000

table.MeasureTemperature.InfraredTemperatureParam.Compensation[6][0]=6.000000

table.MeasureTemperature.InfraredTemperatureParam.Compensation[6][1]=0.000000

table.MeasureTemperature.InfraredTemperatureParam.Compensation[7][0]=7.000000

table.MeasureTemperature.InfraredTemperatureParam.Compensation[7][1]=0.000000

table.MeasureTemperature.InfraredTemperatureParam.Compensation[8][0]=8.000000

table.MeasureTemperature.InfraredTemperatureParam.Compensation[8][1]=0.000000

table.MeasureTemperature.InfraredTemperatureParam.Compensation[9][0]=9.000000

table.MeasureTemperature.InfraredTemperatureParam.Compensation[9][1]=0.000000

table.MeasureTemperature.InfraredTemperatureParam.Compensation[10][0]=10.000000

table.MeasureTemperature.InfraredTemperatureParam.Compensation[10][1]=0.000000

table.MeasureTemperature.InfraredTemperatureParam.Compensation[11][0]=11.000000

table.MeasureTemperature.InfraredTemperatureParam.Compensation[11][1]=0.000000

table.MeasureTemperature.InfraredTemperatureParam.Compensation[12][0]=12.000000

table.MeasureTemperature.InfraredTemperatureParam.Compensation[12][1]=0.000000

table.MeasureTemperature.InfraredTemperatureParam.Compensation[13][0]=13.000000

table.MeasureTemperature.InfraredTemperatureParam.Compensation[13][1]=0.000000

table.MeasureTemperature.InfraredTemperatureParam.Compensation[14][0]=14.000000

table.MeasureTemperature.InfraredTemperatureParam.Compensation[14][1]=0.000000

table.MeasureTemperature.InfraredTemperatureParam.Compensation[15][0]=15.000000

table.MeasureTemperature.InfraredTemperatureParam.Compensation[15][1]=0.000000

table.MeasureTemperature.InfraredTemperatureParam.Compensation[16][0]=16.000000

table.MeasureTemperature.InfraredTemperatureParam.Compensation[16][1]=0.000000

table.MeasureTemperature.InfraredTemperatureParam.Compensation[17][0]=17.000000

table.MeasureTemperature.InfraredTemperatureParam.Compensation[17][1]=0.000000

table.MeasureTemperature.InfraredTemperatureParam.Compensation[18][0]=18.000000

table.MeasureTemperature.InfraredTemperatureParam.Compensation[18][1]=0.000000

table.MeasureTemperature.InfraredTemperatureParam.Compensation[19][0]=19.000000

table.MeasureTemperature.InfraredTemperatureParam.Compensation[19][1]=0.000000

table.MeasureTemperature.InfraredTemperatureParam.Compensation[20][0]=20.000000

table.MeasureTemperature.InfraredTemperatureParam.Compensation[20][1]=0.000000

table.MeasureTemperature.InfraredTemperatureParam.Compensation[21][0]=21.000000

table.MeasureTemperature.InfraredTemperatureParam.Compensation[21][1]=0.000000

table.MeasureTemperature.InfraredTemperatureParam.Compensation[22][0]=22.000000

table.MeasureTemperature.InfraredTemperatureParam.Compensation[22][1]=0.000000

table.MeasureTemperature.InfraredTemperatureParam.Compensation[23][0]=23.000000

table.MeasureTemperature.InfraredTemperatureParam.Compensation[23][1]=0.000000

table.MeasureTemperature.InfraredTemperatureParam.Compensation[24][0]=24.000000

table.MeasureTemperature.InfraredTemperatureParam.Compensation[24][1]=0.000000

table.MeasureTemperature.InfraredTemperatureParam.Compensation[25][0]=25.000000

table.MeasureTemperature.InfraredTemperatureParam.Compensation[25][1]=0.000000

table.MeasureTemperature.InfraredTemperatureParam.Compensation[26][0]=26.000000

table.MeasureTemperature.InfraredTemperatureParam.Compensation[26][1]=0.000000

table.MeasureTemperature.InfraredTemperatureParam.Compensation[27][0]=27.000000

table.MeasureTemperature.InfraredTemperatureParam.Compensation[27][1]=0.000000

table.MeasureTemperature.InfraredTemperatureParam.Compensation[28][0]=28.000000

table.MeasureTemperature.InfraredTemperatureParam.Compensation[28][1]=0.000000

table.MeasureTemperature.InfraredTemperatureParam.Compensation[29][0]=29.000000

table.MeasureTemperature.InfraredTemperatureParam.Compensation[29][1]=0.000000

table.MeasureTemperature.InfraredTemperatureParam.Compensation[30][0]=30.000000

table.MeasureTemperature.InfraredTemperatureParam.Compensation[30][1]=0.000000

table.MeasureTemperature.InfraredTemperatureParam.Compensation[31][0]=31.000000

table.MeasureTemperature.InfraredTemperatureParam.Compensation[31][1]=0.000000

table.MeasureTemperature.InfraredTemperatureParam.Compensation[32][0]=32.000000

table.MeasureTemperature.InfraredTemperatureParam.Compensation[32][1]=0.000000

table.MeasureTemperature.InfraredTemperatureParam.Compensation[33][0]=33.000000

table.MeasureTemperature.InfraredTemperatureParam.Compensation[33][1]=0.000000

table.MeasureTemperature.InfraredTemperatureParam.Compensation[34][0]=34.000000

table.MeasureTemperature.InfraredTemperatureParam.Compensation[34][1]=0.000000

table.MeasureTemperature.InfraredTemperatureParam.Compensation[35][0]=35.000000

table.MeasureTemperature.InfraredTemperatureParam.Compensation[35][1]=0.000000

table.MeasureTemperature.InfraredTemperatureParam.Compensation[36][0]=36.000000

table.MeasureTemperature.InfraredTemperatureParam.Compensation[36][1]=0.000000

table.MeasureTemperature.InfraredTemperatureParam.Compensation[37][0]=37.000000

table.MeasureTemperature.InfraredTemperatureParam.Compensation[37][1]=0.000000

table.MeasureTemperature.InfraredTemperatureParam.Compensation[38][0]=38.000000

table.MeasureTemperature.InfraredTemperatureParam.Compensation[38][1]=0.000000

table.MeasureTemperature.InfraredTemperatureParam.Compensation[39][0]=39.000000

table.MeasureTemperature.InfraredTemperatureParam.Compensation[39][1]=0.000000

table.MeasureTemperature.InfraredTemperatureParam.Compensation[40][0]=40.000000

table.MeasureTemperature.InfraredTemperatureParam.Compensation[40][1]=0.000000

table.MeasureTemperature.InfraredTemperatureParam.Compensation[41][0]=41.000000

table.MeasureTemperature.InfraredTemperatureParam.Compensation[41][1]=0.000000

table.MeasureTemperature.InfraredTemperatureParam.Compensation[42][0]=42.000000

table.MeasureTemperature.InfraredTemperatureParam.Compensation[42][1]=0.000000

table.MeasureTemperature.InfraredTemperatureParam.Compensation[43][0]=43.000000

table.MeasureTemperature.InfraredTemperatureParam.Compensation[43][1]=0.000000

table.MeasureTemperature.InfraredTemperatureParam.Compensation[44][0]=44.000000

table.MeasureTemperature.InfraredTemperatureParam.Compensation[44][1]=0.000000

table.MeasureTemperature.InfraredTemperatureParam.Compensation[45][0]=45.000000

table.MeasureTemperature.InfraredTemperatureParam.Compensation[45][1]=0.000000

table.MeasureTemperature.InfraredTemperatureParam.Compensation[46][0]=46.000000

table.MeasureTemperature.InfraredTemperatureParam.Compensation[46][1]=0.000000

table.MeasureTemperature.InfraredTemperatureParam.Compensation[47][0]=47.000000

table.MeasureTemperature.InfraredTemperatureParam.Compensation[47][1]=0.000000

table.MeasureTemperature.InfraredTemperatureParam.Compensation[48][0]=48.000000

table.MeasureTemperature.InfraredTemperatureParam.Compensation[48][1]=0.000000

table.MeasureTemperature.InfraredTemperatureParam.Compensation[49][0]=49.000000

table.MeasureTemperature.InfraredTemperatureParam.Compensation[49][1]=0.000000

table.MeasureTemperature.InfraredTemperatureParam.Correct=0.000000

table.MeasureTemperature.InfraredTemperatureParam.DebugModelEnable=false

table.MeasureTemperature.InfraredTemperatureParam.EnvLightThreshold[0]=78

table.MeasureTemperature.InfraredTemperatureParam.EnvLightThreshold[1]=130

table.MeasureTemperature.InfraredTemperatureParam.MaxDistance=150

table.MeasureTemperature.InfraredTemperatureParam.OverTemperatureMaxDistance=100

table.MeasureTemperature.InfraredTemperatureParam.RectEnable=false

table.MeasureTemperature.InfraredTemperatureParam.RetentionTime=500

table.MeasureTemperature.InfraredTemperatureParam.SensorType=90641

table.MeasureTemperature.InfraredTemperatureParam.TempMode=0

table.MeasureTemperature.InfraredTemperatureParam.Threshold=37.300000

table.MeasureTemperature.InfraredTemperatureParam.ValidTemperatureHighLimit=45.000000

table.MeasureTemperature.InfraredTemperatureParam.ValidTemperatureLowerLimit=35.000000

table.MeasureTemperature.MaskOpt=0

table.MeasureTemperature.OnlyTemperatureMode=false

table.MeasureTemperature.TemperatureDisplay=true

table.MeasureTemperature.TemperatureUnit=0

table.MeasureTemperature.ThermalImagingParam.RetentionTime=30

table.MeasureTemperature.ThermalImagingParam.Threshold=60

table.MeasureTemperature.Type=0

table.MeasureTemperature.WristTemperatureParam.Compensation[0][0]=0.000000

table.MeasureTemperature.WristTemperatureParam.Compensation[0][1]=0.000000

table.MeasureTemperature.WristTemperatureParam.Compensation[1][0]=1.000000

table.MeasureTemperature.WristTemperatureParam.Compensation[1][1]=0.000000

table.MeasureTemperature.WristTemperatureParam.Compensation[2][0]=2.000000

table.MeasureTemperature.WristTemperatureParam.Compensation[2][1]=0.000000

table.MeasureTemperature.WristTemperatureParam.Compensation[3][0]=3.000000
table.MeasureTemperature.WristTemperatureParam.Compensation[3][1]=0.000000
table.MeasureTemperature.WristTemperatureParam.Compensation[4][0]=4.000000
table.MeasureTemperature.WristTemperatureParam.Compensation[4][1]=0.000000
table.MeasureTemperature.WristTemperatureParam.Compensation[5][0]=5.000000
table.MeasureTemperature.WristTemperatureParam.Compensation[5][1]=0.000000
table.MeasureTemperature.WristTemperatureParam.Compensation[6][0]=6.000000
table.MeasureTemperature.WristTemperatureParam.Compensation[6][1]=0.000000
table.MeasureTemperature.WristTemperatureParam.Compensation[7][0]=7.000000
table.MeasureTemperature.WristTemperatureParam.Compensation[7][1]=0.000000
table.MeasureTemperature.WristTemperatureParam.Compensation[8][0]=8.000000
table.MeasureTemperature.WristTemperatureParam.Compensation[8][1]=0.000000
table.MeasureTemperature.WristTemperatureParam.Compensation[9][0]=9.000000
table.MeasureTemperature.WristTemperatureParam.Compensation[9][1]=0.000000
table.MeasureTemperature.WristTemperatureParam.Compensation[10][0]=10.000000
table.MeasureTemperature.WristTemperatureParam.Compensation[10][1]=0.000000
table.MeasureTemperature.WristTemperatureParam.Compensation[11][0]=11.000000
table.MeasureTemperature.WristTemperatureParam.Compensation[11][1]=0.000000
table.MeasureTemperature.WristTemperatureParam.Compensation[12][0]=12.000000
table.MeasureTemperature.WristTemperatureParam.Compensation[12][1]=0.000000
table.MeasureTemperature.WristTemperatureParam.Compensation[13][0]=13.000000
table.MeasureTemperature.WristTemperatureParam.Compensation[13][1]=0.000000
table.MeasureTemperature.WristTemperatureParam.Compensation[14][0]=14.000000
table.MeasureTemperature.WristTemperatureParam.Compensation[14][1]=0.000000
table.MeasureTemperature.WristTemperatureParam.Compensation[15][0]=15.000000
table.MeasureTemperature.WristTemperatureParam.Compensation[15][1]=0.000000
table.MeasureTemperature.WristTemperatureParam.Compensation[16][0]=16.000000
table.MeasureTemperature.WristTemperatureParam.Compensation[16][1]=0.000000
table.MeasureTemperature.WristTemperatureParam.Compensation[17][0]=17.000000
table.MeasureTemperature.WristTemperatureParam.Compensation[17][1]=0.000000
table.MeasureTemperature.WristTemperatureParam.Compensation[18][0]=18.000000
table.MeasureTemperature.WristTemperatureParam.Compensation[18][1]=0.000000
table.MeasureTemperature.WristTemperatureParam.Compensation[19][0]=19.000000
table.MeasureTemperature.WristTemperatureParam.Compensation[19][1]=0.000000

table.MeasureTemperature.WristTemperatureParam.Compensation[20][0]=20.000000
table.MeasureTemperature.WristTemperatureParam.Compensation[20][1]=0.000000
table.MeasureTemperature.WristTemperatureParam.Compensation[21][0]=21.000000
table.MeasureTemperature.WristTemperatureParam.Compensation[21][1]=0.000000
table.MeasureTemperature.WristTemperatureParam.Compensation[22][0]=22.000000
table.MeasureTemperature.WristTemperatureParam.Compensation[22][1]=0.000000
table.MeasureTemperature.WristTemperatureParam.Compensation[23][0]=23.000000
table.MeasureTemperature.WristTemperatureParam.Compensation[23][1]=0.000000
table.MeasureTemperature.WristTemperatureParam.Compensation[24][0]=24.000000
table.MeasureTemperature.WristTemperatureParam.Compensation[24][1]=0.000000
table.MeasureTemperature.WristTemperatureParam.Compensation[25][0]=25.000000
table.MeasureTemperature.WristTemperatureParam.Compensation[25][1]=0.000000
table.MeasureTemperature.WristTemperatureParam.Compensation[26][0]=26.000000
table.MeasureTemperature.WristTemperatureParam.Compensation[26][1]=0.000000
table.MeasureTemperature.WristTemperatureParam.Compensation[27][0]=27.000000
table.MeasureTemperature.WristTemperatureParam.Compensation[27][1]=0.000000
table.MeasureTemperature.WristTemperatureParam.Compensation[28][0]=28.000000
table.MeasureTemperature.WristTemperatureParam.Compensation[28][1]=0.000000
table.MeasureTemperature.WristTemperatureParam.Compensation[29][0]=29.000000
table.MeasureTemperature.WristTemperatureParam.Compensation[29][1]=0.000000
table.MeasureTemperature.WristTemperatureParam.Compensation[30][0]=30.000000
table.MeasureTemperature.WristTemperatureParam.Compensation[30][1]=0.000000
table.MeasureTemperature.WristTemperatureParam.Compensation[31][0]=31.000000
table.MeasureTemperature.WristTemperatureParam.Compensation[31][1]=0.000000
table.MeasureTemperature.WristTemperatureParam.Compensation[32][0]=32.000000
table.MeasureTemperature.WristTemperatureParam.Compensation[32][1]=0.000000
table.MeasureTemperature.WristTemperatureParam.Compensation[33][0]=33.000000
table.MeasureTemperature.WristTemperatureParam.Compensation[33][1]=0.000000
table.MeasureTemperature.WristTemperatureParam.Compensation[34][0]=34.000000
table.MeasureTemperature.WristTemperatureParam.Compensation[34][1]=0.000000
table.MeasureTemperature.WristTemperatureParam.Compensation[35][0]=35.000000
table.MeasureTemperature.WristTemperatureParam.Compensation[35][1]=0.000000
table.MeasureTemperature.WristTemperatureParam.Compensation[36][0]=36.000000
table.MeasureTemperature.WristTemperatureParam.Compensation[36][1]=0.000000

table.MeasureTemperature.WristTemperatureParam.Compensation[37][0]=37.000000
table.MeasureTemperature.WristTemperatureParam.Compensation[37][1]=0.000000
table.MeasureTemperature.WristTemperatureParam.Compensation[38][0]=38.000000
table.MeasureTemperature.WristTemperatureParam.Compensation[38][1]=0.000000
table.MeasureTemperature.WristTemperatureParam.Compensation[39][0]=39.000000
table.MeasureTemperature.WristTemperatureParam.Compensation[39][1]=0.000000
table.MeasureTemperature.WristTemperatureParam.Compensation[40][0]=40.000000
table.MeasureTemperature.WristTemperatureParam.Compensation[40][1]=0.000000
table.MeasureTemperature.WristTemperatureParam.Compensation[41][0]=41.000000
table.MeasureTemperature.WristTemperatureParam.Compensation[41][1]=0.000000
table.MeasureTemperature.WristTemperatureParam.Compensation[42][0]=42.000000
table.MeasureTemperature.WristTemperatureParam.Compensation[42][1]=0.000000
table.MeasureTemperature.WristTemperatureParam.Compensation[43][0]=43.000000
table.MeasureTemperature.WristTemperatureParam.Compensation[43][1]=0.000000
table.MeasureTemperature.WristTemperatureParam.Compensation[44][0]=44.000000
table.MeasureTemperature.WristTemperatureParam.Compensation[44][1]=0.000000
table.MeasureTemperature.WristTemperatureParam.Compensation[45][0]=45.000000
table.MeasureTemperature.WristTemperatureParam.Compensation[45][1]=0.000000
table.MeasureTemperature.WristTemperatureParam.Compensation[46][0]=46.000000
table.MeasureTemperature.WristTemperatureParam.Compensation[46][1]=0.000000
table.MeasureTemperature.WristTemperatureParam.Compensation[47][0]=47.000000
table.MeasureTemperature.WristTemperatureParam.Compensation[47][1]=0.000000
table.MeasureTemperature.WristTemperatureParam.Compensation[48][0]=48.000000
table.MeasureTemperature.WristTemperatureParam.Compensation[48][1]=0.000000
table.MeasureTemperature.WristTemperatureParam.Compensation[49][0]=49.000000
table.MeasureTemperature.WristTemperatureParam.Compensation[49][1]=0.000000
table.MeasureTemperature.WristTemperatureParam.Correct=0.000000
table.MeasureTemperature.WristTemperatureParam.InvalidTemperatureDistance=24
table.MeasureTemperature.WristTemperatureParam.TemperatureTimeout=10
table.MeasureTemperature.WristTemperatureParam.Threshold=37.300000
table.MeasureTemperature.WristTemperatureParam.ValidTemperatureDistance=5
table.MeasureTemperature.WristTemperatureParam.ValidTemperatureHighLimit=45.000000

table.MeasureTemperature.WristTemperatureParam.ValidTemperatureLowerLimit=10.000000 …

5.8.13 Audio Output Volume

Parameter Name	**Type**	**Required**	**Description**	**Example**
AudioOutputVolume	int[]	O	Audio output volume. An array, each audio output channel has a configuration.	[80, 50]
table.AudioOutputVolume[0]=20				

5.8.14 Video In Sharpness

Parameter Name	**Type**	**Required**	**Description**	**Example**
VideoInSharpness	object	O	Video sharpness setting ability	
+Brightness	uint8	O	Brightness 0~100	50
+Contrast	uint8	O	Contrast 0~100	50
+Saturation	uint8	O	Saturation 0~100	50
table.VideoColor[0][0].Brightness=50 table.VideoColor[0][0].Contrast=50 table.VideoColor[0][0].Saturation=50				

5.8.15 Day and Night Mode

Parameter Name	**Type**	**Required**	**Description**	**Example**

VideoInOptions	object	O	Front-end options for video input One-dimensional array, with one configuration for each video input channel	
+DayNightColor	int	O	Automatically switch colors at day and night: 0: Always colored; 1: Auto switch based on the brightness; 2: Always black & white; 4:Switch based on the outside I0; 5: Switch based on outside alarms.	0
table.VideoInOptions[0].DayNightColor=0				

5.8.16 Auto Registration

Parameters	Type	R/O	Description	Example
VSP_CGI	object	Yes		
+ServiceStart	bool	O	CGI service control configuration. It is true by default.	true
+AutoRegister	object	No	Auto registration	
Enable	bool	O	Enable	false
++DeviceID	char[64]	O	Device ID	""
++Servers	object[]	No	Client address. The large and small web pages can be automatically increased or decreased, and the number of it ranges from 1 to 4.	
+++Type	enumint	O	Address type. It is displayed through a drop-down list on the web page. Available values: 0: IP address; 1: Domain name	0
+++Address	char[64]	O	IP address. When the value of Type is 0, the field is displayed on the web page.	""
+++Port	uint32	O	Port. When the value of Type is 0, the field is displayed on the web page.	80

+++DoMain	char[128]	O	Domain address. When the value of Type is 1, the field is displayed on the web page.	""
+++HttpsEnable	bool	O	Enable HTTPS	false

Example

```
{
    "ServiceStart": true,
    "AutoRegister": {
        "Enable": false,
        "DeviceID": "",
        "Servers": [
            {
                "Type": 0,
                "Address": "",
                "Port": 80,
                "DoMain": "",
                "HttpsEnable": false
            }
        ]
    }
}
```


5.9 The Second Generation of Access Control

5.9.1 Searching for the Capability of Second Generation Protocol

Request

Syntax	http://<server>/cgi-bin/api/AC/getCaps			
Method	POST			
Parameter Format	key=value format at URL			
Parameter Name	**Type**	**Required**	**Description**	**Example**
WantMethods	bool	O	Whether to request to return to the Methods list, please	true

			obtain according to actual needs.	
WantCaps	bool	O	Whether to request to return to the capability item list, please obtain according to actual needs.	true

Example

```
http://<server>/cgi-bin/api/AC/getCaps
{
        "WantMethods" : true
       "WantCaps" : true
 }
```

Response

Parameter Format	key=value format at URL			
Parameter Name	**Type**	**Required**	**Description**	**Example**
Caps	object	R	Capability set	
+AC	object	O	AC capability set	
++Channels	uint32	O	Number of supported access control channels. The former is "AccessControlChannels”.	4
++HasAlarmRecord	bool	O	Supports recording access control alarm logs. The former is “AccessControlAlarmRecord” .	false
++EncryptionMethod	uint8	O	The storage method of passwords in the ACCustomPassword record set. The former is "CustomPasswordEncryption ”. 0: Plaintext, the default value is 0. 1: MD5	0

++HasFingerprintAuth	uint8	O	Whether fingerprint authentication is supported. The former is "SupportFingerprint". 0: unknown, compatible with previous one (default); 1: not supported; 2: supported	0
++HasCardAuth	bool	O	Whether card authentication is supported. false: not supported; true: supported	false
++HasFaceAuth	bool	O	Human Face identification authentication is supported. false: not supported; true: supported	false
++OnlySingleDoorAuth	uint8	O	Whether only single-door authorization (card issuing) multi-door controller is supported 0: not supported; 1: supported	0
++IsAsynAuth	uint8	O	Whether asynchronous return authentication is supported 0: not supported; 1: supported	0
++IsUserIsolate	uint8	O	Whether it is a person-card separation scheme. In the person-card separation scheme, one person can have several cards. 0: no; 1: R	0
++MaxInsertRate	uint16	O	The general maximum number of data can be sent at a time. It is similar to the default conservative set value of BIOS.	10
++ScheduleCaps	object	O	The schedule capability of the device, corresponding to the	

			former SpecialDaysSchedules	
+++Support	bool	O	Whether the new schedule is supported. false: not supported; true: supported	false
+++MaxSchedules	uint16	O	The maximum number of holiday plan capabilities supported by one access controller.	6
+++MaxTimePeriodsPerDay	uint8	O	The maximum number of time periods that can be defined in a day.	6
+++MaxSpecialDayGroups	uint16	O	The maximum number of holiday plan groups supported by the access controller.	6
+++MaxDaysInSpecialDayGroup	uint16	O	The maximum number of holidays supported by a holiday group.	16
++UnlockModes	uint16[128]	O	The combination of unlocking modes supported by the device. The element value corresponds to the former "Opening Method", namely the Method value in the AccessControl configuration.	[1, 2, ...]
++SupportBackendAI	bool	O	Whether the AI identification mode is supported, true: supported, false: not supported.	true

++SupportFastImport	enumint	O	Support fast import
			Enumint {
			0: Not supported
			1: Supported
			}
				0

			Not supported if it does not exist	

++SupportFastCheck	enumint	O	Support quick review function (only compare userID)
			Enumint {
			0: Not supported
			1: Supported
			}
			Not supported if it does not exist
				0

++SupportRapidCheck	enumint	O	Supports rapid review function
			Enumint{
			0: Not supported
			1: Supported
			}
			Not supported if it does not exist.
				0

++IncrementImport	enumint	O	Supports incremental delivery Enumint { 0: Not supported 1: Supported } Not supported if it does not exist.	0
++FingerCompareMode	uint8	O	HasFingerprintAuth supports fingerprint authentication function, this field is valid. 0: Unknown, meaningless 1: The device only supports front-end fingerprint comparison. 2: The device only supports back-end fingerprint comparison. 3: Indicates that the device supports both front-end and	2

			back-end fingerprint comparison.	
++SupportHelmet	enumint	O	Supports safety helmet function. (Taken from IsSupportHelmet) Enumint{ 0: Not supported 1: Supported } Not supported if it does not exist	0
++UserNameMaxLen	uint32	O	Supports limiting name length on the device	32
++SupportASGManager	bool	O	Supports turnstile business components (the turnstile was previously hung on the access controller as a sub-device, and the related configuration of the turnstile was written in the access control. Now divide the turnstile from the access control.) true: supported, false: not supported.	true
++SnapPicPath	char[512]	O	The local storage directory of the door-opening snapshots; the file name is stored in the door-opening record database, and the absolute path of the snapshot can be obtained by appending this directory.	"/mnt/data/userpic/"
++FaceImagePath	char[512]	O	The local storage directory of face database images.	"/mnt/data/FaceImage/"
+AccessUser	object	O	AccessUser capability set.	
++MaxInsertRate	uint16	O	Maximum number of inserts per time.	10

++MaxUsers	uint32	O	Maximum number of users that can be recorded and processed.	600
++MaxFingerPrintsPerUser	uint8	O	Maximum number of fingerprints that can be recorded per person.	5
++MaxCardsPerUser	uint8	O	Maximum number of card that can be recorded per person.	5
++MaxFacesPerUser	uint8	O	Maximum number of Face photo that can be recorded per person.	1
+AccessCard	object	O	AccessCard capability set.	
++MaxInsertRate	uint16	O	Maximum number of inserts per time.	10
++MaxCards	uint32	O	Maximum storage number of cards.	600
+AccessFingerprint	object	O		
++MaxInsertRate	uint16	O	Maximum number of inserts per time.	10
++MaxFingerprintSize	uint16	O	Maximum bytes number of single fingerprint data.	810
++MaxFingerprints	uint32	O	Maximum number of fingerprint storage.	600
++AlgorithmVendor	uint32	O	Fingerprint algorithm manufacturer; 0: Unknown;1: Dahua; 2: Brmicro	0
++AlgorithmVersion	uint32	O	Fingerprint algorithm version number; each 8 bit represents a version from high to low according to Major/Minor, for example, 1.5.2 represents as 0x0001050.	
+AccessFace	object	O	AccessFace capability set.	
++MaxInsertRate	uint16	O	Maximum number of inserts per time.	10

++MaxFaces	uint32	O	Maximum storage number of face image.	600
++RecognitionType	uint8	O	Human face recognition add mode.	1
++RecognitionAlgorithmVender	uint16	O	Face Identification algorithm provider. 0: Unknown; 1: Dahua; 2: SenseTime; 3: Yitu; 4: Hanvon; 5: Huoyan	0
++RecognitionVersion	uint32	O	Human face identify the algorithm (model) version number, if the version number has multiple digits, each 8 bit represents a version from high to low according to Major/Minor, for example, 1.5.2 represents as 0x00010502.	
++MinPhotoSize	uint16	O	Minimum size of white light face photo, KB	20
++MaxPhotoSize	uint16	O	Maximum size of white light face photo, KB	20
++MaxGetPhotoRate	uint16	O	The maximum amount of acquisitions per time by the white light Face list method Unit (number/per time)	20
++IsSupportGetPhoto	bool	O	Whether the list interface is supported to obtain white light photos.	true
++IsSupportOnlyIssueFaceEigen	bool	O	Whether only sending Face the characteristic value is supported.	true
++MultiFaceDetect	object	O		
+++Support	bool	O	Whether multi-Peopledetection identification is supported.	true

+++MaxNums	uint32	O	The maximum number of people detections supported at a time.	3
Example				

5.9.2 Sending User Information

URL	http://<server>/cgi-bin/AccessUser.cgi?action=insertMulti		
Method	POST		
Description	Insert person information.		
[Request Params] (JSON format in body)			
Parameter Name	Type	R/O	Description
UserList	**array <object>**	R	User list, with up to 10 entries
+UserID	string	R	User ID
+UserName	string	O	Username
+UserType	uint16	O	0 General user, by default; 1: Blocklist user (report the blocklist event ACBlocklist); 2: Guest user; 3: Patrol user; 4: VIP user; 5: User who need extended time
	uint16	O	Limit of pass times for guest users
+IsFirstEnter	bool	O	First user authority or not. false: O; true: R
+FirstEnterDoors	int16[]	O	-1 indicates all channels.
+UserStatus	uint16	O	0: Normal; 1: Frozen;
+Authority	uint8	O	User authority (attendance machine field). 1: Admin; 2: Normal User.
+CitizenIDNo	string	O	ID card number
+Password	string	O	The password when unlocking by card + password. The password when unlocking by UserID + password
+Doors	int16[]	O	

			Door authority. The index in the controller is used with TimeSections, and the value corresponds to the subscript of the AC configuration.
+TimeSections	uint16[]	O	The door authority corresponds to the period index. For example, door 3 corresponds to period 2. Each element corresponds to the door in Doors.
+SpecialDaysSchedule	uint32[64]	O	Holiday plan identification. The value is the subscript number configured by SpecialDaysSchedule (defined in the configuration).
+ValidFrom	string	O	"yyyy-MM-dd HH:mm:ss" start of validity period. Note: The original "ValidDateStart" is deprecated.
+ValidTo	string	O	"yyyy-MM-dd HH:mm:ss" end of validity period Note: The original "ValidDateEnd" is deprecated.
[Response Params] (OK)			
[Example]			

Request	POST http://192.168.1.108/cgi-bin/AccessUser.cgi?action=insertMulti
	
	{
	"UserList" :[{
	"UserID": "100013",
	"UserName": "",
	"UserType": 0,
	"UseTime": 1,
	"IsFirstEnter": true,
	"FirstEnterDoors": [0, 1],
	"UserStatus": 0,
	"Authority": 1,
	"CitizenIDNo": "123456789012345678",
	"Password": "xxxxxxxxxx",
	"Doors": [1,3,5,7],
	"TimeSections": [1,2,3,4],
	"alarmStatus": [1,2],		

	"ValidFrom": "2018-01-02 00:00:00", "ValidTo": "2018-01-02 01:00:00", } ,….,{}] }
Response	OK

5.9.3 Updating User Information

URL	http://<server>/cgi-bin/AccessUser.cgi?action=updateMulti		
Method	POST		
Description	Update the user information.		
[Request Params] (JSON format in body)			
Parameter Name	Type	R/O	Description
UserList	**array <object>**	R	User list, with up to 10 entries
+ UserID	string	R	User ID
+UserName	string	O	Username
+UserType	uint16	O	0 General user, by default; 1: Blocklist user (report the blocklist event ACBlocklist); 2: Guest user; 3: Patrol user; 4: VIP user; 5: User who need extended time
+UseTime	uint16	O	Limit of pass times for guest users
+IsFirstEnter	bool	O	First user authority or not. false: O; true: R
+FirstEnterDoors	int16[]	O	-1 indicates all channels.
+UserStatus	uint16	O	0: Normal; 1: Frozen;
+Authority	uint8	O	User authority (attendance machine field). 1: Admin; 2: Normal User
+CitizenIDNo	string	O	ID card number
+Password	string	O	The password when unlocking by card + password. The password when unlocking by UserID + password

+Doors	int16[]	O	Door authority. The index in the controller is used with TimeSections, and the value corresponds to the subscript of the AC configuration.
+TimeSections	uint16[]	O	The door authority corresponds to the period index. For example, door 3 corresponds to period 2. Each element corresponds to the door in Doors.
+SpecialDaysSchedule	uint32[64]	O	Holiday plan identification. The value is the subscript number configured by SpecialDaysSchedule (defined in the configuration).
+ValidFrom	string	O	"yyyy-MM-dd HH:mm:ss" start of validity period Note: The original "ValidDateStart" is deprecated.
+ValidTo	string	O	"yyyy-MM-dd HH:mm:ss" Period of validity. Note: The original "ValidDateEnd" is deprecated.
[Response Params] (OK)			
[Example]			

Request	POST http://192.168.1.108/cgi-bin/AccessUser.cgi?action= updateMulti
	
	{
	"UserList" :[{
	"UserID": "100013",
	"UserName": "",
	"UserType": 0,
	"UseTime": 1,
	"IsFirstEnter": true,
	"FirstEnterDoors": [0, 1],
	"UserStatus": 0,
	"Authority": 1,
	"CitizenIDNo": "123456789012345678",
	"Password": "xxxxxxxxxx",
	"Doors": [1,3,5,7],
	"TimeSections": [1,2,3,4],		

	"SpecialDaysSchedule": [1,2], "ValidFrom": "2018-01-02 00:00:00", "ValidTo": "2018-01-02 01:00:00", } ,…,{}]
Response	OK

5.9.4 Deleting the Information of All Users

URL	http://<server>/cgi-bin/AccessUser.cgi?action=removeAll		
Method	GET		
Description	Clear information of all users.		
[Request Params] (key=value format at URL)			
Parameter Name	Type	R/O	Description
[Response Params] (OK)			
[Example]			
Request	GET http://192.168.1.108/cgi-bin/AccessUser.cgi?action=removeAll		
Response	OK		

5.9.5 Deleting the Information of Multiple Users

URL	http://<server>/cgi-bin/AccessUser.cgi?action=removeMulti		
Method	GET		
Description	Delete user data.		
[Request Params] (key=value format at URL)			
Parameter Name	Type	R/O	Description
UserIDList	**array <string>**	R	User ID list, with up to 10 entries
[Response Params] (OK)			
[Example]			

Request	GET http://192.168.1.108/cgi-bin/AccessUser.cgi?action=removeMulti&UserIDList[0]=102&UserIDList[1]=102
Response	OK

5.9.6 Searching for Information of Multiple Users

URL	http://<server>/cgi-bin/AccessUser.cgi?action=list		
Method	GET		
Description	Search for user data.		
[Request Params] (key=value format at URL)			
Parameter Name	Type	R/O	Description
UserIDList	**array <string>**	R	User ID list, with up to 10 entries
[Response Params]			
Users	**array <object>**	**R**	**The records that returned.**
+ UserID	string	R	User ID
+UserName	string	O	Username
+UserType	uint16	O	0 General user, by default; 1: Blocklist user (report the blocklist event ACBlocklist); 2: Guest user; 3: Patrol user; 4: VIP user; 5: User who need extended time
+UseTime	uint16	O	Limit of pass times for guest users
+IsFirstEnter	bool	O	First user authority or not. false: O; true: R
+FirstEnterDoors	int16[]	O	-1 indicates all channels.
+UserStatus	uint16	O	0: Normal; 1: Frozen;
+Authority	uint8	O	User authority (attendance machine field). 1: Admin; 2: Normal User
+CitizenIDNo	string	O	ID card number

+Password	string	O	The password when unlocking by card + password. The password when unlocking by UserID + password
+Doors	int16[]	O	Door authority. The index in the controller is used with TimeSections, and the value corresponds to the subscript of the AC configuration.
+TimeSections	uint16[]	O	The door authority corresponds to the period index. For example, door 3 corresponds to period 2. Each element corresponds to the door in Doors.
+SpecialDaysSchedule	uint32[64]	O	Holiday plan identification. The value is the subscript number configured by SpecialDaysSchedule (defined in the configuration).
+ValidFrom	string	O	"yyyy-MM-dd HH:mm:ss" start of validity period Note: The original "ValidDateStart" is deprecated.
+ValidTo	string	O	"yyyy-MM-dd HH:mm:ss" Period of validity. Note: The original "ValidDateEnd" is deprecated.
[Example]			
Request	GET http://192.168.1.108/cgi-bin/AccessUser.cgi?action=list&UserIDList[0]=102&UserIDList[1]=102		
Response	Users[0].UserID=100013		

	Users[0].UserName=Name Users[0].UserType=1 Users[0].UseTime=1 Users[0].IsFirstEnter=ZhangSan Users[0].FirstEnterDoors=0 Users[0].UserStatus=12345678 Users[0].Authority=1 Users[0].CitizenIDNo=1 Users[0].Password=ZhangSan Users[0].Doors=0 Users[0].TimeSections=12345678 Users[0].SpecialDaysSchedule=1 Users[0].ValidFrom=1 records[0].ValidTo=ZhangSan …

5.9.7 Starting to Search for User Information (by Conditions)

URL	http://<server>/cgi-bin/AccessUser.cgi?action=startFind		
Method	GET		
Description	Start searching for related user information.		
[Request Params] (key=value format at URL)			
Parameter Name	Type	R/O	Description
Condition	**object**	R	Search conditions. Users can perform conditional search according to the user information field.
+UserID	string	O	User ID
+UserName	string	O	Username
+UserType	uint16	O	0 General user, by default; 1: Blocklist user (report the blocklist event ACBlocklist); 2: Guest user; 3: Patrol

			user; 4: VIP user; 5: User who need extended time
+UseTime	uint16	O	Limit of pass times for guest users
+IsFirstEnter	bool	O	First user authority or not. false: O; true: R
+FirstEnterDoors	int16[]	O	-1 indicates all channels.
+UserStatus	uint16	O	0: Normal; 1: Frozen;
+Authority	uint8	O	User authority (attendance machine field). 1: Admin; 2: Normal User
+CitizenIDNo	string	O	ID card number
+Password	string	O	The password when unlocking by card + password. The password when unlocking by UserID + password
+Doors	int16[]	O	Door authority. The index in the controller is used with TimeSections, and the value corresponds to the subscript of the AC configuration.
+TimeSections	uint16[]	O	The door authority corresponds to the period index. For example, door 3 corresponds to period 2. Each element corresponds to the door in Doors.
+SpecialDaysSchedule	uint32[64]	O	Holiday plan identification. The value is the subscript number configured by SpecialDaysSchedule (defined in the configuration).
+ValidFrom	string	O	"yyyy-MM-dd HH:mm:ss" start of validity period Note: The original "ValidDateStart" is deprecated.
+ValidTo	string	O	"yyyy-MM-dd HH:mm:ss" Period of validity. Note: The

			original "ValidDateEnd" is deprecated.
[Response Params]			
Token	uint32	R	Search token.
Total	uint32	R	Total number of entries found this time
Caps	uint32	R	Search capability: Maximum number of records that can be returned each time.
[Example]			
Request	GET http://192.168.1.108/cgi-bin/AccessUser.cgi?action=startFind&Condition.UserID=1		
Response	{ "Token": 1234, "Total": 20, "Caps ": 20 }		

5.9.8 Getting the User Information (by Conditions)

URL	http://<server>/cgi-bin/AccessUser.cgi?action=doFind		
Method	GET		
Description	**Get user related information.**		
[Request Params] (key=value format at URL)			
Parameter Name	Type	R/O	Description
Token	**int**	R	Search token.
Offset	**int**	R	Offset
Count	**int**	R	Number of entries obtained this time
[Response Params]			
info	array <object>	R	Person Information
[Example]			

Request	GET http://192.168.1.108/cgi-bin/AccessUser.cgi?action=doFind&Token=1234&Offset=0&Count=20
Response	**{** **"Info": [{** **"UserID": "102",** **…** **}, { … }, …]** **}**

5.9.9 Stopping Searching for the User Information (by Conditions)

URL	http://<server>/cgi-bin/AccessUser.cgi?action=stopFind		
Method	GET		
Description	Get the user information.		
[Request Params] (key=value format at URL)			
Parameter Name	Type	R/O	Description
Token	**int**	R	Search token
[Response Params] (OK)			
[Example]			
Request	GET http://192.168.1.108/cgi-bin/AccessUser.cgi?action=stopFind&Token=1234		
Response	OK		

5.9.10 Sending Card Number Information

URL	http://<server>/cgi-bin/AccessCard.cgi?action=insertMulti		
Method	POST		
Description	Insert card number information.		
[Request Params] (JSON format in body)			
Parameter Name	Type	R/O	Description
CardList	**array <object>**	R	User list, with up to 10 entries
+CardNo	string	R	Card No.
+UserID	string	R	User ID
+CardType	uint16	O	Card type Enumint{ 0: Ordinary card. 1: VIP card. 2: Guest card. 3: Patrol card. 4: Blocklist card. 5: Duress card
+CardName	string	O	Card name
+CardStatus	uint32	O	Card status. Different card status results in different person status. 0: Normal 1<<0: Reported for loss 1<<1: Canceled 1<<2: Frozen 1<<3: Arrearage 1<<4: Overdue
[Response Params] (OK)			
[Example]			

Request	POST http://192.168.1.108/cgi-bin/AccessCard.cgi?action=insertMulti
	
	{
	CardList[{
	"UserID" : "100013"
	"CardNo" : ""
	"CardType" : 0
	"CardName" : "201-Tom"
	"CardStatus" : 0
	} ,…,{}]
	}		

Response	OK

5.9.11 Updating Card Number Information

URL	http://<server>/cgi-bin/AccessCard.cgi?action=updateMulti		
Method	POST		
Description	Update card number information.		
[Request Params] (JSON format in body)			
Parameter Name	Type	R/O	Description
CardList	**array <object>**	R	User list, with up to 10 entries
+CardNo	string	R	Card No.
+UserID	string	R	User ID
+CardType	uint16	O	Card type Enumint{ 0: Ordinary card. 1: VIP card. 2: Guest card. 3: Patrol card. 4: Blocklist card. 5: Duress card
+CardName	string	O	Card name
+CardStatus	uint32	O	Card status. Different card status results in different person status. 0: Normal 1<<0: Reported for loss 1<<1: Canceled 1<<2: Frozen 1<<3: Arrearage 1<<4: Overdue
[Response Params] (OK)			
[Example]			

Request	POST http://192.168.1.108/cgi-bin/AccessCard.cgi?action=updateMulti
	
	{
	CardList[{
	"UserID" : "100013"
	"CardNo" : ""
	"CardType" : 0
	"CardName" : "201-Tom"
	"CardStatus" : 0		

	} ,…,{}] }
Respo nse	OK

5.9.12 Clearing the Information of All Card Numbers

URL	http://<server>/cgi-bin/AccessCard.cgi?action=removeAll		
Method	GET		
Description	Clear all card information.		
[Request Params] (key=value format at URL)			
Parameter Name	Type	R/O	Description
[Response Params] (OK)			
[Example]			
Request	GET http://192.168.1.108/cgi-bin/AccessCard.cgi?action=removeAll		
Response	OK		

5.9.13 Clearing the Information of Multiple Cards

URL	http://<server>/cgi-bin/AccessCard.cgi?action=updateMulti		
Method	GET		
Description	Delete card number data.		
[Request Params] (key=value format at URL)			
Parameter Name	Type	R/O	Description
CardNoList	**array <string>**	R	User card number list, with up to 10 entries
[Response Params] (OK)			
[Example]			

Request	GET http://192.168.1.108/cgi-bin/AccessCard.cgi?action=removeMulti&CardNoList[0]=12345678&CardNoList[1]=12345687
Response	OK

5.9.14 Searching for the Information of Multiple Cards

URL	http://<server>/cgi-bin/AccessCard.cgi?action=list		
Method	GET		
Description	Search for card number data.		
[Request Params] (key=value format at URL)			
Parameter Name	Type	R/O	Description
CardNoList	**array <string>**	R	User ID list, with up to 10 entries
[Response Params]			
records	**array <object>**	**R**	**The records that returned.**
+CardNo	string	R	Card No.
+UserID	string	R	User ID
+CardType	uint16	O	Card type Enumint{ 0: Ordinary card. 1: VIP card. 2: Guest card. 3: Patrol card. 4: Blocklist card. 5: Duress card
+CardName	string	O	Card name
+CardStatus	uint32	O	Card status. Different card status results in different person status. 0: Normal 1<<0: Reported for loss 1<<1: Canceled 1<<2: Frozen 1<<3: Arrearage

			1<<4: Overdue
[Example]			
Request	GET http://192.168.1.108/cgi-bin/AccessCard.cgi?action=list&CardNoList[0]=102&CardNoList[1]=102		
Response	Cards[0].CardNo=12345678 Cards[0].UserID=1 Cards[0].CardType=1 Cards[0].CardName=ZhangSan Cards[0].CardStatus=0 … Cards[1].CardNo=12345679 Cards[1].UserID=2 Cards[1].CardType=1 Cards[1].CardName=LiSi Cards[1].CardStatus=0 …		

5.9.15 Starting Searching for Card Number Information

URL	http://<server>/cgi-bin/ AccessCard.cgi?action=startFind		
Method	GET		
Description	Start searching for card number information.		
[Request Params] (key=value format at URL)			
Parameter Name	Type	R/O	Description
Condition	**object**	R	Search conditions. Users can perform conditional search according to the user information field.
+UserID	string	O	User ID
+CardNo	string	O	User card number
+CardType	uint16	O	Card type Enumint{ 0: Ordinary card.

			1: VIP card. 2: Guest card. 3: Patrol card. 4: Blocklist card. 5: Duress card
+CardName	string	O	Card name
+CardStatus	uint32	O	Card status. Different card status results in different person status. 0: Normal 1<<0: Reported for loss 1<<1: Canceled 1<<2: Frozen 1<<3: Arrearage 1<<4: Overdue
[Response Params]			
Token	uint32	R	Search token.
Total	uint32	R	Total number of entries found this time
Caps	uint32	R	Search capability: Maximum number of records that can be returned each time.
[Example]			
Request	GET http://192.168.1.108/cgi-bin/AccessCard.cgi?action=startFind&Condition.UserID=1		

Response	{
	"Token": 1234,
	"Total": 20,
	"Caps ": 20
	}		

5.9.16 Getting Related Card Number Information

URL	http://<server>/cgi-bin/AccessCard.cgi?action=doFind		
Method	GET		
Description	**Get related card number information**		
[Request Params] (key=value format at URL)			
Parameter Name	Type	R/O	Description
Token	**int**	R	Search token
Offset	**int**	R	Offset

Count	**int**	R	Number of entries obtained this time
[Response Params]			
info	array <object>	R	Person Information
[Example]			
Request	GET http://192.168.1.108/cgi-bin/AccessCard.cgi?action=doFind&Token=1234&Offset=0&Count=20		
Response	{ "Info": [{ "UserID": "102", … }, { … }, …] }		

5.9.17 Stopping Searching for Related Card Number Information

URL	http://<server>/cgi-bin/AccessCard.cgi?action=stopFind		
Method	GET		
Description	**Stop searching for related card number information**		
[Request Params] (key=value format at URL)			
Parameter Name	Type	R/O	Description
Token	**int**	R	Search token.
[Response Params] (OK)			
[Example]			
Request	GET http://192.168.1.108/cgi-bin/AccessCard.cgi?action=stopFind&Token=1234		
Response	OK		

5.9.18 Sending Face Information

URL	http://<server>/cgi-bin/AccessFace.cgi?action=insertMulti		
Method	POST		
Description	Add face information.		
[Request Params] (JSON format in body)			
Parameter Name	Type	R/O	Description

FaceList	array <object>	R	Face list, with up to 10 entries
+UserID	string	R	User ID
+FaceData	array <string>	O	Base64 of red light face templates, 8192x20. Each length contains
+PhotoData	array <string>	O	Base64 of white light face images, 100 KB × 5.
+PhotoURL	array <string>	O	Either cloud storage URL of white light face images or PhotoData. If PhotoData exists, this field is invalid when it is subject to PhotoData. Only HTTP URL is available now.
[Response Params] (OK)			
[Example]			

Request	POST http://192.168.1.108/cgi-bin/AccessFace.cgi?action=insertMulti
	
	{
	“FaceList”:[
	{
	"UserID": "102",
	"FaceData": ["xxxx", "xxxx", …],
	"PhotoData": ["yyyy", "yyyy", …],
	" PhotoURL": ["yyyy", "yyyy", …],
	},
	….
	]
	}		

Response	OK		

5.9.19 Updating Face Information

URL	http://<server>/cgi-bin/AccessFace.cgi?action=updateMulti		
Method	POST		
Description	Update face information.		
[Request Params] (JSON format in body)			
Parameter Name	Type	R/O	Description
FaceList	array <object>	R	Face list, with up to 10 entries
+UserID	string	R	User ID
+FaceData	array <string>	O	Base64 of red light face templates, 8192x20. Each length contains

+PhotoData	array <string>	O	Base64 of white light face images, 100 KB × 5.
+PhotoURL	array <string>	O	Either cloud storage URL of white light face images or PhotoData. If PhotoData exists, this field is invalid when it is subject to PhotoData. Only HTTP URL is available now.
[Response Params] (OK)			
[Example]			

Request	POST http://192.168.1.108/cgi-bin/AccessFace.cgi?action=updateMulti
	
	{
	“FaceList”:[
	{
	"UserID": "102",
	"FaceData": ["xxxx", "xxxx", …],
	"PhotoData": ["yyyy", "yyyy", …],
	" PhotoURL": ["yyyy", "yyyy", …],
	},
	….
	]
	}		

Response	OK		

5.9.20 Deleting All Face Information

URL	http://<server>/cgi-bin/AccessFace.cgi?action=removeAll		
Method	GET		
Description	Clear all face information.		
[Request Params] (key=value format at URL)			
Parameter Name	Type	R/O	Description
[Response Params] (OK)			
[Example]			
Request	GET http://192.168.1.108/cgi-bin/AccessFace.cgi?action=removeAll		
Response	OK		

5.9.21 Deleting Multiple Face Information

URL	http://<server>/cgi-bin/AccessFace.cgi?action=removeMulti		
Method	GET		
Description	Delete face data.		
[Request Params] (key=value format at URL)			
Parameter Name	Type	R/O	Description
UserIDList	**array <string>**	R	User card number list, with up to 10 entries
[Response Params] (OK)			
[Example]			
Request	GET http://192.168.1.108/cgi-bin/AccessFace.cgi?action=removeMulti&UserIDList[0]=101&UserIDList[1]=102		
Response	OK		

5.9.22 Searching for Multiple Face Information

URL	http://<server>/cgi-bin/AccessFace.cgi?action=list		
Method	GET		
Description	Search for user face data.		
[Request Params] (key=value format at URL)			
Parameter Name	Type	R/O	Description
UserIDList	**array <string>**	R	User ID list, with up to 10 entries
[Response Params]			
FaceDataList	**array <object>**	**R**	**The records that returned.**
+UserID	string	R	User ID
+FaceData	array <string>	O	Base64 of red light face templates8192x20. Each length contains (not supported, protocol reserved)

+PhotoData	array <string>	O	Base64 of white light face images, 100 KB × 5.
+PhotoURL	array <string>	O	Either cloud storage URL of white light face images or PhotoData. If PhotoData exists, this field is invalid when it is subject to PhotoData. Only HTTP URL is available now. (not supported, protocol reserved)
[Example]			
Request	GET http://192.168.1.108/cgi-bin/AccessFace.cgi?action=list&UserIDList[0]=1&UserIDList[1]=2		
Response	**Cards[0].UserID=1** FaceDataList[0].PhotoData=["xxxx", "xxxx", …], FaceDataList[0].FaceData=["xxxx", "xxxx", …], … **FaceDataList[1].UserID=2** FaceDataList[1].PhotoData=["xxxx", "xxxx", …], FaceDataList[1].FaceData=["xxxx", "xxxx", …], …		

5.9.23 Starting Searching for Face Information

URL	http://<server>/cgi-bin/AccessFace.cgi?action=startFind		
Method	GET		
Description	Start searching for face information.		
[Request Params] (key=value format at URL)			
Parameter Name	Type	R/O	Description
Condition	object	R	Search conditions. Users can perform conditional search according to the user information field.
+UserID	string	O	User ID
[Response Params]			

Token	uint32	R	Search token.
Total	uint32	R	Total number of entries found this time
Caps	uint32	R	Search capability: Maximum number of records that can be returned each time.
[Example]			
Request	GET http://192.168.1.108/cgi-bin/AccessFace.cgi?action=startFind&Condition.UserID=1		
Response	{ "Token": 1234, "Total": 20, "Caps ": 20 }		

5.9.24 Getting Face Information

URL	http://<server>/cgi-bin/AccessFace.cgi?action=doFind		
Method	GET		
Description	**Get face related information.**		
[Request Params] (key=value format at URL)			
Parameter Name	Type	R/O	Description
Token	**int**	R	Search token.
Offset	**int**	R	Offset
Count	**int**	R	Number of entries obtained this time
[Response Params]			
info	array <object>	R	Person Information
[Example]			
Request	GET http://192.168.1.108/cgi-bin/AccessFace.cgi?action=doFind&Token=1234&Offset=0&Count=20		
Response	{ "Info": [{ "UserID": "102", … }, { … }, …] }		

5.9.25 Stopping Searching for Face Information

URL	http://<server>/cgi-bin/AccessFace.cgi?action=stopFind		
Method	GET		
Description	**Stop searching for face related information.**		
[Request Params] (key=value format at URL)			
Parameter Name	Type	R/O	Description
Token	**Int**	R	Search token.
[Response Params] (OK)			
[Example]			
Request	GET http://192.168.1.108/cgi-bin/AccessFace.cgi?action=stopFind&Token=1234		
Response	OK		

5.9.26 Sending Fingerprint Information

URL	http://<server>/cgi-bin/AccessFingerprint.cgi?action=insertMulti		
Method	POST		
Description	Add fingerprint information.		
[Request Params] (JSON format in body)			
Parameter Name	Type	R/O	Description
vecPackets	array <string>	R	Fingerprint feature data
AccessFingerprints	array <object>	R	Fingerprint list, with up to 1 entry
+UserID	String	R	User ID
+FingerprintPacket	Object	R	Send fingerprint information list.
++Length	uint32	R	Length of a single fingerprint package
++Count	uint32	R	Number of fingerprint packages
++DataURL	array <string>	O	Either cloud storage URL of fingerprint data or Length. If the Length exists and is not 0, this field is invalid when it is subject to Length. Only HTTP URL is available now.
++DuressIndex	Uint8	R	Duress fingerprint number, with a value range of [1, count]. This field is invalid if the value is illegal. That is, there is no duress fingerprint. For

			example, 0 indicates no duress fingerprint.
[Response Params] (OK)			
[Example]			

Request	POST http://192.168.1.108/cgi-bin/AccessFingerprint.cgi?action=insertMulti
	
	{
	“vecPackets“: ["xxxx", "xxxx", …],
	“AccessFingerprints”:[
	{
	"UserID": "102",
	"FingerprintPacket" :
	{
	"Length" : 810,
	"Count" : 3,
	"DuressIndex" : 2
	}
	}
	]
	}		

Response	OK		

5.9.27 Updating Fingerprint Information

URL	http://<server>/cgi-bin/AccessFingerprint.cgi?action=updateMulti		
Method	POST		
Description	Update fingerprint information.		
[Request Params] (JSON format in body)			
Parameter Name	Type	R/O	Description
vecPackets	array <string>	R	Binary fingerprint data (Binary Data).
AccessFingerprints	array <object>	R	Fingerprint list, with up to 1 entry
+UserID	String	R	User ID
+FingerprintPacket	Object	R	Send fingerprint information list.
++Length	uint32	R	Length of a single fingerprint package
++Count	uint32	R	Number of fingerprint packages
++DataURL	array <string>	O	Either cloud storage URL of fingerprint data or Length. If the Length exists and is not 0, this field

			is invalid when it is subject to Length. Only HTTP URL is available now.
++DuressIndex	Uint8	R	Duress fingerprint number, with a value range of [1, count]. This field is invalid if the value is illegal. That is, there is no duress fingerprint. For example, 0 indicates no duress fingerprint.
[Response Params] (OK)			
[Example]			

Request	POST http://192.168.1.108/cgi-bin/AccessFingerprint.cgi?action=updateMulti
	
	{
	“vecPackets“: ["xxxx", "xxxx", …],
	“AccessFingerprints”:[
	{
	"UserID": "102",
	"FingerprintPacket" :
	{
	"Length" : 810,
	"Count" : 3,
	"DuressIndex" : 2
	}
	}
	]
	}		

Response	OK		

5.9.28 Deleting All Fingerprint Information

URL	http://<server>/cgi-bin/AccessCard.cgi?action=updateMulti		
Method	GET		
Description	Clear all fingerprint information.		
[Request Params] (key=value format at URL)			
Parameter Name	Type	R/O	Description
[Response Params] (OK)			
[Example]			
Request	GET http://192.168.1.108/cgi-bin/AccessFingerprint.cgi?action=removeAll		
Response	OK		

5.9.29 Deleting Fingerprint Information

URL	http://<server>/cgi-bin/AccessFingerprint.cgi?action=removeMulti		
Method	GET		
Description	Delete fingerprint data.		
[Request Params] (key=value format at URL)			
Parameter Name	Type	R/O	Description
UserIDList	**array <string>**	R	User card number list, with up to 1 entry
[Response Params] (OK)			
[Example]			
Request	GET http://192.168.1.108/cgi-bin/AccessFingerprint.cgi?action=removeMulti&UserIDList[0]=101&UserIDList[1]=102		
Response	OK		

5.9.30 Searching for Fingerprint Information

URL	http://<server>/cgi-bin/AccessFingerprint.cgi?action=get		
Method	GET		
Description	Search for user fingerprint information.		
[Request Params] (key=value format at URL)			
Parameter Name	Type	R/O	Description
UserID	string	R	User ID list, with up to 1 entry
[Response Params]			
FingerprintData	BinaryData	R	Binary fingerprint data (Binary Data).
FingerprintPacket	object	R	Description of fingerprint data
+Length	uint32	R	Length of a single fingerprint package
+Count	uint32	R	Number of fingerprint packages
+DataURL	array <string>	O	Either cloud storage URL of fingerprint data or Length. If the Length exists and is not 0, this field is invalid when it is subject to Length. Only HTTP URL is available now.
+DuressIndex	Uint8	R	Duress fingerprint number, with a value range of [1, count]. This field is invalid if the value is illegal. That is, there is no duress fingerprint. For example, 0 indicates no duress fingerprint.
[Example]			

Request	GET http://192.168.1.108/cgi-bin/AccessFingerprint.cgi?action=get&UserID=1
Response	FingerprintPacket.Length=810 FingerprintPacket.Count=3 FingerprintPacket.DuressIndex=1 FingerprintData=xxx

5.10 Administrator Password

5.10.1 Adding Administrator Password

Request

Template	http://<server>/cgi-bin/recordUpdater.cgi?action=insert&name=AccessControlCustomPassword			
Method	GET			
Parameter Format	key=value format in URL			
Parameter Name	**Type**	**R/O**	**Description**	**Example**
name	string	R	Name of custom access control password, with the value of "AccessControlCustomPassword".	AccessControlCustomPassword
UserID	string	R	User ID	101
OpenDoorPassword	string	R	Unlocking password	123456
Doors	Array <integer>	O	Number of the door that can be unlocked by the custom password (The video intercom device does not support this field)	
TimeSections	Array <integer>	O	Numbers of periods during which the door can be unlocked by the password: An array, in which each element corresponds to the door in Doors; period index.	Optional protocol

			(The video intercom device does not support this field)	
ValidDateStart	string	O	Start time of validity, with format of "yyyyMMdd hhmmss".	Optional protocol
ValidDateEnd	string	O	Expiry time, with format of "yyyyMMdd hhmmss".	Optional protocol
Sample				
GET http://192.168.1.108/cgi-bin/recordUpdater.cgi?action=insert&name=AccessControlCustomPassword&UserID=102&OpenDoorPassword=123456&Doors[0]=0&Doors[1]=1				

Parameter Format	key=value format in body			
Parameter Name	**Type**	**R/O**	**Description**	**Example**
RecNo	integer	R	Record number	1234
Sample				
RecNo=12345				

5.10.2 Searching for Administrator Password

Request

Template	http://<server>/cgi-bin/recordFinder.cgi?action=find&name=AccessControlCustomPassword			
Method	GET			
Parameter Format	key=value format in URL			
Parameter Name	**Type**	**R/O**	**Description**	**Example**
name	string	R	Name of custom access control password, with the value of "AccessControlCustomPassword".	AccessControlCustomPassword
count	integer	O	The maximum value returned, 1024 by default	1024

StartTime:	string	O	Start time of record creation	123456700
EndTime:	string	O	End time of record creation	123456800
condition	object	O	Search conditions	
+UserID	string	O	User ID	101
+recno	integer	O	Record number	1024
Sample				
http://192.168.1.108/cgi-bin/recordFinder.cgi?action=find&name=AccessControlCustomPassword&condition.UserID=103&StartTime=123456700&EndTime=123456800&count=100				

Response

Parameter Format	key=value format in body			
Parameter Name	**Type**	**R/O**	**Description**	**Example**
totalCount	integer	O	Number of all records found	2
found=3	integer	O	Number of records returned	2
records	array <object>	R	Information of records returned	
+RecNo	integer	R	Record number	1234
+CreateTime	integer	O	Record creation time	
+UserID	string	R	User ID	101
+OpenDoorPassword	string	R	Unlocking password	123456
+Doors	Array <integer>	O	Number of the door that can be unlocked by the custom password	

			(The video intercom device does not support this field)	
+TimeSections	Array <integer>	O	Numbers of periods during which the door can be unlocked by the access control card: An array, in which each element corresponds to the door in Doors; period index. (The video intercom device does not support this field)	
+ValidDateStart	string	O	Start time of validity, with format of "yyyyMMdd hhmmss".	20151022 093811
+ValidDateEnd	string	O	Expiry time, with format of "yyyyMMdd hhmmss".	20151023 093811
Sample				
totalCount=1000 found=100 records[0].RecNo=12345 records[0].CreateTime=123456789 records[0].UserID=103 records[0].OpenDoorPassword=123456 records[0].Doors[0]=1 records[0].Doors[1]=3 records[0].Doors[2]=5 … records[1].RecNo=13579 records[1].CreateTime=123456799 records[1].UserID=103 records[0].OpenDoorPassword=123456 records[1].Doors[0]=2				

records[1].Doors[1]=4 records[1].Doors[2]=6 …

5.10.3 Editing the Administrator Password

Note: You should provide at least one optional parameter for update.

Request

Template	http://<server>/cgi-bin/recordUpdater.cgi?action=update&name=AccessControlCustomPassword			
Method	GET			
Parameter Format	key=value format in URL			
Parameter Name	**Type**	**R/O**	**Description**	**Example**
name	string	R	Name of custom access control password, with the value of "AccessControlCustomPassword".	AccessControlCustomPassword
recno	integer	R	Record number	1234
UserID	string	R	User ID	101
OpenDoorPassword	string	R	Unlocking password	123456
Doors	Array <integer>	O	Number of the door that can be unlocked by the custom password (The video intercom device does not support this field)	
TimeSections	Array <integer>	O	Numbers of periods during which the door can be unlocked by the access control card:	

			An array, in which each element corresponds to the door in Doors; period index. (The video intercom device does not support this field)	
ValidDateStart	string	O	Start time of validity, with format of "yyyyMMdd hhmmss".	
ValidDateEnd	string	O	Expiry time, with format of "yyyyMMdd hhmmss".	
Sample				
http://192.168.1.108/cgi-bin/recordUpdater.cgi?action=update&name=AccessControlCustomPassword&recno=12345&UserID=102&OpenDoorPassword=123456&Doors[0]=1&Doors[1]=3&Doors[2]=5&ValidDateStart=20151022 093811&ValidDateEnd=20151222 093811				

Response

Parameter Format	OK at body			
Parameter Name	**Type**	**R/O**	**Description**	**Example**
Sample				
OK				

5.10.4 Deleting Administrator Password

Delete the custom access control password information through the record number.

Request

Template	http://<server>/cgi-bin/recordUpdater.cgi?action=remove&name=<RecordName>			
Method	GET			
Parameter Format	key=value format in URL			
Parameter Name	**Type**	**R/O**	**Description**	**Example**

name	string	R	Name of custom access control password, with the value of "AccessControlCustomPassword".	AccessControlCustomPassword
recno	integer	R	Record number	12345
Sample				
http://192.168.1.108/cgi-bin/recordUpdater.cgi?action=remove&name=AccessControlCustomPassword&recno=12345				

Response

Parameter Format	OK at body			
Parameter Name	**Type**	**R/O**	**Description**	**Example**
Sample				
OK				

5.10.5 Clearing Administrator Password

Delete the information of all the custom access control passwords.

Request

Template	http://<server>/cgi-bin/recordUpdater.cgi?action=clear&name=AccessControlCustomPassword			
Method	GET			
Parameter Format	key=value format in URL			
Parameter Name	**Type**	**R/O**	**Description**	**Example**
name	string	R	Name of custom access control password, with the value of "AccessControlCustomPassword".	AccessControlCustomPassword

Sample
http://192.168.1.108/cgi-bin/recordUpdater.cgi?action=clear&name=AccessControlCustomPassword

Response

Parameter Format	OK at body			
Parameter Name	**Type**	**(R/O)**	**Description**	**Example**
Sample				
OK				

5.10.6 Getting the Total Number of Administrator Password

Request

Template	http://<server>/cgi-bin/recordFinder.cgi?action=getQuerySize&name=AccessControlCustomPassword			
Method	GET			
Parameter Format	key=value format in URL			
Parameter Name	**Type**	**(R/O)**	**Description**	**Example**
name	string	R	Access card and Fingerprint records should be "AccessControlCustomPassword"	AccessControlCustomPassword
Sample				
http://192.168.1.108/cgi-bin/recordFinder.cgi?action=getQuerySize&name=AccessControlCustomPassword				

Response

Parameter Format	key=value format in body

Parameter Name	Type	(R/O)	Description	Example
count	string	R	Total number of records	150
Sample				
count=150				

5.11 Event Data Format

5.11.1 Access Control Unlock Event

Method	For how to subscribe to events, see the chapter"Subscribing for Events"		
Description	This event is triggered when you try to unlock the door.		
[Event Params] (key=value format)			
Parameter Name	**Type**	**Required**	**Description**
Events	array <object>	R	Event array.
+EventBaseInfo	object	R	Basic event information.
++Code	String	R	Event code, which should be "**AccessControl**".
++Action	String	R	Event action, which can be: "Start", "Stop", "Pulse".
++Index	int	O	Event related channel index.
+UTC	uint32	Null	UTC Time. Standard UTC time (without time zone and DST deviation), which is required for access control products and optional for intercom products.
+RealUTC	uint32	O	Standard UTC time (without time zone and DST deviation)
+RecNo	uint	Null	Record number. Record number in AccessControlCardRec.
+Name	char[128]	O	Access control name.
+FaceIndex	uint8	O	R/O for platform: One person and multiple faces. When unlocking by face, the matched face number is reported, with a range of 0 to 4. 0 by default when unlocking by other methods.

+Type	enumchar[32]	Null	Event type, which can be
			enumchar[32]{
			"Entry": – Entry
			"Exit": – Exit
			}

Status	int	Null	Unlocking result
			Enumint{
			0: Failed
			1: Succeeded
			}
			Unlocking succeeded by default if the field is not specified.

+CardType	enumint	Null	Card type
			Enumint{
			0: Ordinary card
			1: VIP card
			2: Guest card
			3: Patrol card
			4: Blocklist card
			5: Duress card
			6: Inspection card
			0xff: Main card
			}

+UserType	enumint	Null	User type
			Enumint{
			0: General user
			1: Blocklist user (The blocklist event AccessControlBlocklist is reported for blocklist).
			2: VIP user
			3: Guest user
			4: Patrol user
			5: Other user
			}

+CardStatus	uint	Null	Card status: 0: Normal. 1<<0: Reported for loss. 1<<1: Canceled. 1<<2: Frozen. 1<<3: Arrearage. 1<<4: Overdue. 1<<5: Pre-arrearage.
+Method	int	Null	For unlocking method, see OpenDoorMethod.

+CardNo	string	Null	Card number Hexadecimal, the value is card number for other cases.
+ReaderID	char[32]	Null	Door card reader ID: Decimal.
+UserID	char[9]	Null	Unlocking user. String.
+ErrorCode	int32	Null	Unlocking failure reason, which is valid only when the Status is 0. For available value, see AccessError.
+SnapURL	char[128]	Null	Snapshot storage address: Unlock and upload snapshots, and then notify the first snapshot address in the event. Multiple snapshots are distinguished by the image name xxx_N, and N means the serial number, starting from 1.
+Numbers	uint	Null	Number of snapshots.
+SN	char[32]	Null	Device serial number, such as smart lock: Wireless accessories require this field.

+Attendance State	enumint	Null	Attendance status:
			Enumint{
			1: Sign in.
			2: Outing.
			3: Return from outing.
			4: Sign out.
			5: Overtime sign in.
			6: Overtime sign out.
			}

+QRCode	char[512]	Null	QR code.
+Similarity	uint8	Null	Face recognition similarity: Valid when unlocking by face, 0–100.
+ImageInfo	object[]	Null	Image information extension: Up to 6 images.

++Type	enumint	Null	enumint{
			0: Local face database image.
			1: Scenario image.
			2: Face cutout.
			3: IR snapshot.
			4: Accompanying person snapshot (Capital Airport).
			5: Heatmap.
			}

++Offset	uint32	Null	Offset in the binary data block.

++Length	uint32	Null	Image size, in bytes.
++Width	uint16	Null	Image width, in pixels.
++Height	uint16	Null	Image height, in pixels.
++Bounding Box	Rect	Null	Coordinates of the target bounding box, in the 8192 coordinate system.
+CitizenIDNo	char[20]	Null	ID card number, 18 digits.
+CitizenName	char[128]	O	Legal name. Official citizen name on the ID card, household register, and passport.
+EventGroupID	uint32	Null	Event group ID, which is used for different events to link the same action.
+ReadCardState	uint32	O	Whether the current event is a collection card. 0: Card swiping for access control; 1: Collection card for access control.
+ObjectProperties	object	O	Dynamically identified structured information object, which is used to store all the structured information of targets identified by the AI.
+Mask	uint8	O	Whether to wear mask. The default value is 0. 0: Unknown 1: No mask 2: Wear mask
+ManTemperatureInfo	object	O	Personnel temperature information.
++CurrentTemperature	float	O	Body temperature.
++TemperatureUnit	uint32	O	Temperature unit (0: Celsius; 1: Fahrenheit; 2: Kelvin).
++IsOverTemperature	bool	O	Whether it is over temperature.
++TempPoint	Point	O	Position coordinates of the current temperature monitoring point, in the 8192 coordinate system.
++TempType	uint8	O	Body temperature monitoring status: 0: Low temperature. 1: Normal temperature. 2: High temperature.
+Remaining Times	uint32	O	Remaining times of use reported when unlocking, alre.ady implemented for guest users.
+CompanyName	char[200]	O	Name of unit.

+Score	uint8	O	Face quality score.
+ButtonCheck	uint8	O	Whether the door was manually unlocked by pressing the button. The default value is 0. 0- Unknown. 1- The button was not pressed 2- The button was pressed (customization request for temperature monitoring from the North America) .
+TempPassword	char[64]	O	Temporary password.
+Note	char[512]	O	Digest information.
+TrafficPlate	char[32]	O	License plate.
ErrorCodeEx	uint32[8]	O	More reasons for unlock failure, which are valid when the Status is 0. Refer to AccessError for its value.
+RecordUrl	char[128]	O	Videos for 15–30 s will be recorded when the the door is unlocked. The address of recorded videos will be notified in report events. The video will be named after its number and recording time.
+CitizenIDAddress	char[108]	O	Home address.
+CitizenIDBirth	char[12]	O	Date of birth.
+CitizenIDMinzu	uint8	O	Reserved field.
+UUID	char[256]	O	UUID of automatic network replenishment of unlock records, which ensures the unique correlation between the request and subsequent report.
+RemoteQRCodeType	int32	O	The type of QR code that is reported for the remote unlock by QR code. The type of int is easy for extension. 0- default, 1- visitor QR code, and 2- QR code in the all-in-one card .
+TicketInfo	object	O	Ticket information
++TicketType	int8	O	1- Individual ticket; 2- Team ticket. Default: 1.
++Accessible	int32	O	The event is used for verification. This field indicates the number of people passed by using the ticket.
++TicketNo	char[128]	O	Ticket number, used for the verification on the platform.

+ButtonControlInfo	object	O	Authentication information of the button control
++Operate	enumint	O	The operation type of the button. 0- Not operate, 1- Unlock, 2- Lock, 3- Reset. Default: 0.
++DoorIndex	uint32	O	The code of the door that was operated. 0x00- unknown, 0x01- door 1, 0x02- door 2, 0x03- door 1 and door 2. Default: 0.
+RoomNo	char[32]	O	Reported room number.
+TransmissionUuid	char[64]	O	The unique ID of device unlock information pass-through.
+PassResult	int32	O	Access result, which indicates if a person passed or not. 1- Person passed through the entrance barrier 2-Person passed through the exit barrier 3- No person passed through.
Sample			
Event	--<boundary> Content-Type: text/plain Content-Length: <length> Events[0].EventBaseInfo.Code=TrafficJam Events[0].EventBaseInfo.Action=Pulse Events[0].Vehicle.RelativeID=0 Events[0].RecNo=123 Events[0].Name=Door1 Events[0].Type=Entry Events[0].Status=1 Events[0].TrafficCar.IndexInGroup=1 Events[0].CardNo=09DDAABB Events[0].UserID=101 … --<boundary> Content-Type: image/jpeg Content-Length: <image size> <Jpeg image data> --<boundary>		

5.11.2 DoorStatus

Event Code	"DoorStatus"			
Event Action	Pulse			
Event Index	Door channel			
Event Paramter				
Event Data				
Parameter Name	**Type**	**R/O**	**Description**	Example
+UTC	uint32	O	Standard UTC time (without time zone and DST deviation), which is required for access control products and optional for intercom products.	6538920

+Status	enumchar[32]	O	Door status:
			Enumchar[32]{
			"Open": Open.
			"Close": Close.
			"CloseAlways": Normally closed.
			"OpenAlways": Normally open.
			"Normal": Normal.
			}
				"Open"

Sample				

```
{
   "Code" : "DoorStatus"
   "Action" : "Pulse"
   "Index" : 0,
   "Data" : {
      "UTC" : 1610417974,
       "Status" : "Open"
   }
}
```

5.11.3 VideoMotion

Event Code	"VideoMotion"			
Event Action	Start/Stop			
Event Index	Video Channel Number			
Event Paramter				
Event Data				
Parameter Name	Type	R/O	Description	Example
+Id	uint[32]	O	The No. of motion detection area corresponds to the window name Up to 32	[0, 1, …]
+RegionName	char[32][64]	O	Window Name of Motion Detection Area	["Region1", "Region2", …]

+AlarmType	enumchar[32][32]	O	Motion detection trigger type. When there is an Id field, it corresponds to the Id array. When there is no Id field, the trigger area is unknown and is not bound to the window. By default, the first array element represents the trigger type.
			enumchar[32][32]{
			"Human": human
			"Vehicle": vehicle
			"HumanAndVehicle": human and vehicle
			"PIR": PIR alarm
			}
				["Human","Vehicle", …]

+SmartMotionEnable	bool	O	Mark whether intelligent motion detection is enabled When it is determined that this parameter is enabled true, it means that the intelligent motion detection is enabled, and the client blocks this motion detection event.	true
Sample				

{
"Code" : "VideoMotion"
"Action" : "Start/Stop"
"Index" : Video Channel Number
"EventHandler" :
"Data" : {
"Id" : [0, 1, …],
"RegionName": ["Region1","Region2",…]
"AlarmType": ["Human"," Vehicle", …]
"SmartMotionEnable" : true
}
}				

5.11.4 AlarmLocal

Event Code	"AlarmLocal"			
Event Action	Start/Stop			
Event Data				
Parameter Name	**Type**	**R/O**	**Description**	**Example**
+ACK	uint32	O	Confirm ID, events with this field is confirmed by	123U

			EventRestore.ackEvent	
+UTC	uint32	O	Time of event occurrence, UTC time.	6538920
+SenseMethod	char[64]	O	Image sensor	"PassiveInfrared"
+DefenceAreaType	char[64]	O	Zone type, consistent with the configuration under the corresponding channel	"Intime"
Name	char[128]	O	Alarm Channel Name	"Door"
+ExAlarmIn	bool	O	Peripheral Alarm	true
+GPS	object	O	GPS information (mobile requirements)	
+UserID	char[32]	O	User ID	"1234"，
+UserName	char[128]	O	Login UserName	"Tom"
+SN	char[32]	O	Device SN Used to distinguish the alarm from which device	"1C03E08YAZ00020"
+Areas	int[64]	O	Area	[1,64,]
+AlarmType	enumchar[32]	O	Alarm type enumchar[32]{ "Intrusion": Intrusion	"Panic"

			"Fire":Fire "Medical": Medical "Panic":Panic "Gas":Gas "Hold-up": Alarm type of dual emergency alarm buttons }	

Sample

{

"Code" : "AlarmLocal"

"Action" : "Start/Stop"

"Index" : Channel number，history reason，different products have different definitions of products，cannot be unified

"EventHandler" :

"Data" : {

"ACK" : 123U

"UTC" : 6538920,

"SenseMethod" : "PassiveInfrared",

"DefenceAreaType" : "Intime",

"Name" : "Door",

"ExAlarmIn": true,

"GPS" :

"UserID" : "1234"，

"UserName" : "Tom",

"SN": "1C03E08YAZ00020"

"Areas": [1,64]

}

}

5.11.5 ErrorCode

Parameter Name	Description
0x00	No error.
0x10	Unauthorized
0x11	Card reported for loss or canceled
0x12	No permission for the door
0x13	Unlocking mode error
0x14	Validity period error
0x15	Anti-passback mode
0x16	Duress alarm not enabled
0x17	Door normally closed
0x18	AB interlock status
0x19	Patrol Card
0x1A	The device is in the intrusion alarm status.
0x1B	The device is locked (and recovered after the locking time expires).
0x1C	The device is in the DND mode (only the user with the top permission can unlock).
0x1D	The device is in the non-user mode status. Switch to the user mode (Non-user mode refer to disable mode. When you use a

	locally defined functional card to enter this mode, the door lock is not enabled).
0x20	Period error
0x21	Error of unlocking periods during holiday
0x22	Card arrearage
0x23	Card overdue
0x24	Card pre-arrearage
0x25	Blocklist Card
…	
0x30	Card with first card permission should be verified first.
…	
0x40	Correct card, password error.
0x41	Correct card, password timeout.
0x42	Correct card, fingerprint error.
0x43	Correct card, fingerprint timeout.
0x44	Correct fingerprint, password error.
0x45	Correct fingerprint, password timeout.
0x46	Correct UserID, password error.
0x47	Correct UserID, password timeout.

0x50	(Multi-user) combination unlocking order error
0x51	Multi-user combination unlocking should continue to be verified.
0x52	Single-user combination unlocking should continue to be verified.
…	
0x60	Verification passed but unauthorized by the console.
0x61	Correct card, face error.
0x62	Correct card, face timeout.
0x63	Repeated entry
0x64	Unauthorized and need to be recognized by the back-end platform (customized for Capital Airport).
0x65	Hight body temperature
0x66	No mask
0x67	Failed to get health code
0x68	No passing for yellow code
0x69	No passing for red code
0x6a	Invalid health code
0x6b	Verification passing for green code
…	

0x70	Get health code information (The platform should return health code information corresponding to the ID card).
0x71	Check ID card information (The platform should return a check result corresponding to the ID card).
…	
0xA0	Custom password error
0xA1	The user already exists.
0xA2	The error of Single-user combination unlocking order
0xA3	Multi-person unlocking users not in the group
0xA4	User status error (frozen)
0xA5	Maximum times of guest users reached
0xA6	Lift control waiting timeout
0xA7	Beyond the limit of combination unlocking password input errors
0xA8	Not wearing safety helmet (customized and added for Shanghai Intex Exhibition Co.,Ltd.)
0xA9	Illegal Card Exceeding Time

5.12 Configuring Network

5.12.1 Network Configuration

Read Permission	No permission

Write Permission	AuthNetCfg			
Parameter	**Type**	**Required**	**Description**	**Example**
Network	object	Null	Configuration of the network port: Index by network port name, with up to 32 network adapters.	
+Hostname	char[128]	Null	The host name, which forms a network address with the domain name.	"badak"
+Domain	char[128]	Null	Domain	"dahua"
+DefaultInterface	char[32]	Null	Default network adapter configured by the user. Both IPv4 and IPv6 use this configuration. This field only represents user settings. It does not mean that the network adapter is available. For example, if eth0 is set but added to bond0, the IP of eth0 is invalid. Use netApp.getDefaultEthInfo to get the default network adapter that actually works.	"eth0"
+eth0	object	Null	The network port configuration. Each network port corresponds to a configuration.	
++PhysicalAddress	char[18]	Null	MAC address, colon + uppercase letter.	"11:2D:A3:4C:5F:66"
++MTU	uint	Null	Maximum network transmission unit.	1500
++NetMode	enumchar[32]	Null	Network transmission mode. enumchar[32]{	"adapt"

			"adapt": Self-adaptive (default value) "half10M": 10M half-duplex "full10M": 10M full-duplex "half100M": 100M half-duplex "full100M": 100M full-duplex "full1000M": 1000M full-duplex "longPoE10M": Long distance PoE 10M "longPoE100M": Long distance PoE 100M }	

++Type	enumchar[32]	Null	Network port type. (EVS7024 customization requirement)
			enumchar[32]{
			Standard: Standard network port.
			Manager: Management network port.
			Extend: Expansion network port}
			}
				"Standard"

++IPAddress	char[40]	Null	IP address.	"192.168.0.108"
++SubnetMask	char[40]	Null	Subnet mask.	"255.255.0.0"
++DefaultGateway	char[40]	Null	Default gateway.	"192.168.0.1"
++DhcpEnable	bool	Null	Enable DHCP or not.	false
++EnableDhcpReservedIP	bool	Null	Use the reserved IP address (169.254.X.X) or not when DHCP fails. Continue to send DHCP requests when the reserved IP address is used.	false
++DnsAutoGet	bool	Null	DNS acquisition	false

			method, which can be set to true when DHCP is enabled, and can be obtained through DHCP.	
++DnsServers	char[2][40]	Null	DNS server address. Different DNS addresses can be configured for each NIC, but only the DNS of the default NIC is saved to the system.	["221.123.33.228", "221.12.1.228"]
+eth1	object	Null	Other network port configurations. eth1, eth2, eth3, eth4, eth5, eth6, eth7, eth17	
+bond0	object	Null	Virtual network port binding configuration. The name must be in the format of bond + number.	
++Bonding	bool	Null	Bind the virtual network port or not. Only when the network adapter name is bondxx, can the Bonding field be used. Other network adapters cannot be used. true: The network adapter is bound and the physical network port is unavailable. false-The network adapter is unbound (multi-address mode) to make the network adapter in Members available.	true
++Mode	enumchar[32]	Null	NIC binding mode. enumchar[32]{ "BalanceRR":	“BalanceRR"

			RoundRobin load balancing (corresponding to second-generation load balancing). "BalanceXOR": XOR load balancing. "BalanceTLB": Self-adaptive transmission load balancing. "BalanceALB": NIC virtualization load balancing. "ActiveBackup": Active/standby mode (because of historical version, the device uses this value as fault tolerance mode. For compatibility reasons, which is used as fault tolerance mode in implementation). "Broadcast": Fault tolerance mode (to maintain compatibility, this value cannot be used). "802.3ad": Dynamic link aggregation. "Bridge": Bridge (layer 2 switch, in bond format)}	
++Members	char[][16]	Null	Physical network port member.	["eth0", "eth1"]
++IPAddress	char[40]	Null	IP address.	"192.168.0.108"
++Params	object	Null	Dynamic link aggregation parameters.	
+++LACP	enumchar[32]	Null	802.3ad link aggregation control mode.	"MAC"

			enumchar[32]{ "MAC": Based on MAC address. "IPPort": Based on IP address and port. "IPMAC": Based on IP address and MAC address. "IP": Based on IP address. "Port": Based on port.}	
++Name	char[32]	Null	Alias, used for interface display (configurable).	"bridge1"
++DnsServers	char[2][40]	O	DNS server address. Different DNS addresses can be configured for each NIC, but only the DNS of the default NIC is saved to the system. Address.	["221.123.33.228", "221.12.1.228"]
++PhysicalAddress	char[18]	O	MAC address, colon + uppercase letter.	"11:2D:A3:4C:5F:66"
++MTU	uint	O	Maximum network transmission unit.	1500
++SubnetMask	char[40]	O	Subnet mask.	"255.255.0.0"

++Type	char[40]	O	Network port type. enumchar[32]{
			Standard: Standard network port.
			Manager: Management network port.
			Extend: Expansion network port.
			}
				Extend

++DhcpEnable	bool	O	Enable DHCP or not.	false
++EnableDhcpReservedIP	bool	O	Use the reserved IP address (169.254.X.X) or not when DHCP fails. Continue to send	true

			DHCP requests when the reserved IP address is used.	
++DefaultGateway	char[40]	O	Default gateway.	"192.168.0.1"
+bond1	object	Null	Other virtual network port configurations. bond1, bond2, bond3	
+br0	object	Null	Bridge configuration, the name must be in the format of br + number. Use method: When there are two NICs on camera A, one of which is connected to the client, and the other is connected to camera B. Make two NICs into a bridge, and the client accesses cameras A and B through the bridge.	
++Enable	bool	Null	The switch used to enable the function. Indicates whether the bridge configuration takes effect. The default value is false.	false
++Members	char[][16]	Null	Physical sub NICs forming the bridge.	["eth0", "eth1"]
++IPAddress	char[40]	Null	IP address. Bridge working IP.	"192.168.0.108"
++SubnetMask	char[40]	Null	Subnet mask. Bridge subnet mask.	"255.255.0.0"
++DefaultGateway	char[40]	Null	Default gateway. Bridge default gateway.	"192.168.0.1"
++MTU	uint	Null	Maximum network transmission unit.	1500
++DnsServers	char[2][16]	Null	DNS server address.	["8.8.8.8","8.8.5.5"]
++DhcpEnable	bool	O	Enable DHCP or not.	false

++EnableDhcpReservedIP	bool	O	Use the reserved IP address (169.254.X.X) or not when DHCP fails. Continue to send DHCP requests when the reserved IP address is used.	true
++DnsAutoGet	bool	O	DNS acquisition method, which can be set to true when DHCP is enabled, and can be obtained through DHCP.	false

Complete Example


```
{
    "Hostname" : "badak",
    "Domain" : "dahua",
    "DefaultInterface" : "eth0",
    "eth0" : {
        "PhysicalAddress" : "11:2D:A3:4C:5F:66"
        "MTU" : 1500,
        "NetMode" : "adapt",
        "Type" : "Standard"
        "IPAddress" : "192.168.0.108",
        "SubnetMask" : "255.255.0.0",
        "DefaultGateway" : "192.168.0.1",
        "DhcpEnable" : false,
        "EnableDhcpReservedIP" : true,
        "DnsAutoGet" : false,
        "DnsServers" : ["221.123.33.228", "221.12.1.228"],
    },
    "eth1" : {},
    "bond0": {
        "Bonding" : true,
        "Mode" : "BalanceRR",
        "Members" : ["eth0", "eth1"],
        "IPAddress" : "192.168.0.108",
        "Params" : {
            "LACP" : "MAC"
        },
        "Name" : "bridge1"
        "DnsServers": ["221.123.33.228", "221.12.1.228"],
        "PhysicalAddress": "11:2D:A3:4C:5F:66",
```

```
            "MTU": 1500,
            "SubnetMask": "255.255.0.0",
            "DhcpEnable": false,
            "DefaultGateway": "192.168.0.1"
            …,
        },
        "bond1" : {},
        "br0" : {
            "Enable" : false,
            "Members" : ["eth0", "eth1"],
            "IPAddress" : "192.168.0.108",
            "SubnetMask" : "255.255.0.0",
            "DefaultGateway" : "192.168.0.1",
            "MTU" : 1500,
            "DnsServers" : ["8.8.8.8","8.8.5.5"]
        }
}
```


5.12.2 Configuring Wi-Fi

Permission	AuthNetCfg			
Parameter	**Type**	**Required**	**Description**	**Example**
WLan	object	Null	All wireless NIC settings. Use the network port name for indexing.	
+wlan0	object	Null	The network port configuration. Each network port corresponds to a configuration.	
++Enable	bool	Null	The switch used to enable the NIC Wi-Fi.	true
++SSID	char[32]	Null	Network name (SSID)	"dahua"
++BSSID	char[18]	Null	Device MAC address.	"00:aa:0a:a0:11:23"
++ConnectEnable	bool	Null	Manual connection switch. true: Manual connection. false: Manually disconnect from hotspots.	true

++LinkEnable	bool	Null	Automatic connection switch. (The actual meaning of true/false is different from the literal meaning). true: Do not automatically connect. False: Automatically connect to the hotspot. Note: This option does not apply to IPC.	true

++LinkMode	enumchar[32]	Null	Connection mode.
			enumchar[32]{
			"Auto": Automatically select the appropriate mode.
			"Ad-hoc": It is a special application mode of wireless network. When a group of computers are connected to a wireless network adapter, they can connect to each other and share resources without access point.
			“Infrastructure": It is an application mode that integrates the network architecture of wired and wireless LAN. Network resources can be shared with this architecture. Access point is needed in this mode.
			}
				"Auto"

++Encryption	enumchar[32]	Null	Encryption mode, which is the general	"Off"

			name of the two fields Authentication and DataEncryption. This field is used in Netapp.
			enumchar[32]{
			"Off": Turn off.
			"On": Turn on.
			}
			For the mapping relationship between Authentication and DataEncryption or Encryption, see	

++KeyType	enumchar[32]	Null	WEP password type.
			
			enumchar[32]{
			"Hex": hexadecimal password.
			"ASCII": ASCII password
			}
				"Hex"

++KeyID	int	Null	WEP key index. Value range: 0–3.	0
++Keys	char[4][128]	Null	WEP password array. If you use an ASCII password, 64-bit encryption uses 5 letters or numbers, and 128-bit encryption uses 13 letters or numbers (0-9, a-z, and A-Z). If you use a hexadecimal password, 64-digit encryption uses 10 letters or numbers, and the 128-digit encryption uses 26 letters or numbers (0-9, A-F).	["password1", "password2", "password3", "password4"]
++KeyFlag	bool	Null	Whether the password has been	false

			set. The compatibility with the second-generation configuration is reserved.	
++EAP	object	Null		

+++Method	enumchar[32]	Null	EAP method.
			enumchar[32]{
			"PEAP"
			"TLS"
			"TTLS"
			}
				"TLS"

+++AuthType	enumchar[32]	Null	EAP authentication method
			enumchar[32]{
			"NONE"
			"PAP"
			"MSCHAP"
			"MSCHAPV2"
			"GTC"
			}
				"EAP_NONE"

+++Identity	char[64]	Null	Identity	"admin"
+++AnonymousID	char[64]	Null	Anonymous identity.	"admin2"
+++Password	char[64]	Null	Password.	"admin"
+++CaCert	char[512]	Null	CA certificate.	"abc"
+++UserCert	char[512]	Null	User certificate.	"def"
+++Privatekey	char[512]	Null	Client private key path.	"/etc/cert/pk.prv"
+++PrivateKeyPassword	char[64]	Null	Client private key decryption password.	"admin"
++Network	object	Null	Network value after wireless network adapter is connected. Only the following 5 fields are required.	
+++IPAddress	char[40]	Null	IP address. Device working IP.	"192.168.0.108"
+++SubnetMask	char[40]	Null	Subnet mask.	"255.255.0.0"
+++DefaultGateway	char[40]	Null	Default gateway.	"192.168.0.1"
+++DhcpEnable	bool	Null	Enable DHCP or not.	false
+++DnsServers	char[2][40]	Null	DNS server.	["221.123.33.228", "221.12.1.228"]

++Pri5GRssiThreshold	int8	O	5G optimization threshold: If the device supports dual-band and the connected SSID has 2.4G or 5G, the threshold is used to implement the connection policy. If RSSI is higher than this value, 5G connection is preferred. RSSI is usually a negative value. Value range: -100<RssiTher5g<0. If this parameter is set to 0, the new 5G optimization policy is not executed.	-80
++WAPI	object	O		
+++ApCert	char[512]	O	WAPI-CERT certificate authentication method. After the AP router certificate path is specified, the certificate is provided by the AS authentication server.	"/var/cert/ap.cer"
+++StaCert	char[512]	O	WAPI-CERT certificate authentication method. After the STA certificate path is specified, the certificate is provided by the AS certificate server.	"/var/cert/sta.cer"
+wlan1	object	Null	Other wireless NICs.	
+eth2	object	Null	eth2 is used to indicate a wireless network adapter for front-end devices.	

Complete Example

```
{
    "wlan0" : {
        "Enable" : true,
        "SSID" : "dahua",
        "BSSID" : "00:aa:0a:a0:11:23",
        "ConnectEnable" : true,
        "LinkEnable" : true,
        "LinkMode" : "Auto",
        "Encryption" : "Off",
        "KeyType" : "Hex",
        "KeyID" : 0,
        "Keys" : ["password1", "password2", "password3", "password4"],
        "KeyFlag" : false,
        "EAP" : {
            "Method" : "TLS",
            "AuthType" : "EAP_NONE",
            "Identity" : "admin",
            "AnonymousID" : "admin2",
            "Password":"admin",
            "CaCert" : "abc",
            "UserCert" : "def"
            "Privatekey":"/etc/cert/pk.prv"
            "PrivateKeyPassword":"admin"
        }//End of EAP
        "Network" : {
            "IPAddress" : "192.168.0.108",
            "SubnetMask" : "255.255.0.0",
            "DefaultGateway" : "192.168.0.1",
            "DhcpEnable" : false,
            "DnsServers" : [ "221.123.33.228", "221.12.1.228"]
        },
         "Pri5GRssiThreshold": -80,
         "WAPI": {
              "ApCert": "/var/cert/ap.cer",
              "StaCert": "/var/cert/sta.cer"
         }
    }, //End of wlan0
    "wlan1" : {},
    "eth2" : {}
}
```

5.12.3 Configuring Cellular Network

Permission	admin permissions			
Parameter	**Type**	**Required**	**Description**	**Example**
Wireless	object	Null	Cellular network connection settings. Use the connection name for indexing.	
+3G	object	Null	Cellular network connection configuration. Supports 4 cellular modules. Single module: 3G; multiple modules: 3G, 3G1, 3G2, and 3G3. One module corresponds to a physical entity, such as Quectel RM500Q and Fibocom FM160. Note: Considering the compatibility, the 3G naming is reserved, which can actually represent 4G network adapters.	
++Enable	bool	Null	The switch used to enable the cellular network.	true
++Index	int	Null	Module index. It is read-only, which means the index is obtained by the PAL layer.	0
++IMSEnable	bool	O	The switch used to enable the cellular module IMS. Note: As for modules that do not support China Telecom 2G and 3G, if IMS is turned off, phone call	true

			and SMS services cannot be used with China Telecom SIM cards.	
++KeepAlive	uint32	Null	Keep-alive duration. When the application detects that the connection is not in use for a period of time, it changes the Activate flag to false and sets the configuration. Unit: second. 0 indicates that the connection is continuous and does not disconnect automatically.	30

++APN	enumchar[32]	Null	Access network.
			SIM1 dialing parameter: Access network. The default value of a single card is SIM1.
			Enumchar[32]{
			"CTNET": China Telecom 2G and 3G.
			"CTLTE": China Telecom 4G.
			"CMNET": China Mobile.
			"UNINET": China Unicom.
			"3GNET": China Unicom. It is the same as UNINET
			}
				“CTNET"

++AuthMode	enumchar[32]	Null	SIM1 dialing parameter: Authentication mode. The default value of a single card is SIM1. enumchar[32]{	"No"

			"No": Authentication is not required. "PAP": PAP authentication. CHAP: CHAP authentication. }	
++UserName	char[64]	Null	SIM1 dialing parameter: username. The default value of a single card is SIM1.	"card"
++Password	char[64]	Null	SIM1 dialing parameter: password. The default value of a single card is SIM1.	"card"
++SIMCfg	uint8	R	The SIM card number used by the current module. It is 1 by default.	1
++SIM2	object	O	The dialing parameter of the second SIM card. Refer to the SIM1 dialing parameter. If there is a third card, it is named SIM3.	++SIM2

+++APN	enumchar[32]	O	SIM2 dialing parameter: Access network.
			enumchar[32]{
			"CTNET": China Telecom 2G and 3G.
			"CTLTE": China Telecom 4G.
			“CMNET": China Mobile.
			“UNINET": China Unicom.
			“3GNET": China Unicom. It is the same as UNINET.
			}
				"CTNET"

+++AuthMode	enumchar[32]	O	SIM2 dialing	"No"

			parameter: Authentication mode. enumchar[32]{ “No": Authentication is not required. “PAP": PAP authentication. “CHAP”: CHAP authentication. }	
+++UserName	char[64]	O	SIM2 dialing parameter: Username.	"card"
+++Password	char[64]	O	SIM2 dialing parameter: Password.	"123"
+++DailNumber	char[16]	O	SIM2 dialing parameter: Dialing number. The dialing parameter of the public network SIM card of the three major domestic operators are as follows: *98*1# China Mobile *99# China Unicom China Telecom #777 China Telecom	"#777"
+++PIN	char[16]	O	SIM2 PIN code for unlock.	"1234"
+++RoamingEnable	bool	O	Enable cellular network roaming. true: Supports roaming; false: Does not support roaming.	true
++AutoDial	bool	Null	The switch used to enable auto dial-up by time. It is true by default. If it is enabled, the dial-up	true

			period uses TimeSection. If it is false, the TimeSection configuration is invalid.	
++TimeSection	systemtime[]	Null	Auto dial-up period. It is a two-dimensional array. 7 days a week and 6 time periods a day. The time period is represented by a string. The number 1 at the beginning of the string means the time period is valid. 0 means the time period is invalid. When it is time within the validity period, dialing is enabled. When the time is out of the validity period, dialing is turned off.	[["1 00:00:00-24:00:00", "0 00:00:00-24:00:00", "0 00:00:00-24:00:00" "0 00:00:00-24:00:00" "0 00:00:00-24:00:00" "0 00:00:00-24:00:00"], …, []]

++3GFluxTactic	enumint	Null	Traffic usage policy
			enumint{
			0: Monthly data traffic plan.
			1: Monthly duration plan.
			2: Unlimited data traffic.
			}
				0

++3GFluxUp	uint	Null	Data traffic limit. [0, 65535] MB or minutes.	30000
++3GFlux	uint	Null	Actual traffic usage. [0, 65535] MB or minutes.	0
++Day3GFluxTactic	enumchar[32]	Null	Daily traffic control policy. enumchar[32]{ "ByFlux" "ByTime"	"ByFlux"

			}	
++Day3GFluxUp	uint	Null	Daily traffic usage limit. [0, 65535] MB or minutes.	100
++Day3GFluxUse	uint	Null	Used traffic on the current day. [0, 65535] MB or minutes.	100

++Day3GFluxAction	enumchar[32]	Null	Traffic warning policy.
			enumchar[32]{
			"Nothing": No action.
			"3GNetDown": Indicates that 3G is offline, which is the action triggered when the maximum daily traffic is reached.
			}
				"3GnetDown"

++3GFluxTacticEnable	bool	O	Whether the data plan mode is enabled. For example, if it is a monthly plan, it indicates whether the monthly plan mode is enabled.	true

++3GFluxType	enumchar[32]	O	Select the data plan type.
			enumchar[32]{
			"ByDay": Daily plan.
			"ByMonth": Monthly plan.
			"ByYear": Yearly plan.
			}
				"ByDay"

++Month3GFluxTactic	enumchar[32]	O	Monthly traffic control policy.
			enumchar[32]{
			"ByFlux"
			"ByTime"
			}
				"ByFlux"

++Month3GFluxUp	uint32	O	Monthly data traffic limit. [0, 65535] MB or minutes.	30000

++Month3GFluxUse	uint32	O	Used traffic in the current month. [0, 65535] MB or minutes.	100
++Month3GFluxStartDay	uint32	O	The digit indicates which day the plan starts each month.	1
++MonthAvgDayUp	uint32	O	Daily average data limit in the month plan mode. [0,65535] MB.	100
++Month3GFluxEnable	bool	O	Whether the daily data limit of the current month in monthly plan mode is enabled.	true
++Year3GFluxUp	uint32	O	Yearly data traffic limit. [0,65535] MB.	36000
++Year3GFluxUse	uint32	O	Used traffic of the year. [0, 65535] MB or minutes.	3000
++Year3GFluxStartMonth	uint32	O	The digit indicates which month the plan starts each year.	1
++Year3GFluxStartDay	uint32	O	The digit indicates which day the plan starts each month.	1
++YearAvgMonthUp	uint32	O	Monthly average data limit in the yearly plan mode. [0,65535] MB.	3000
++Year3GFluxEnable	bool	O	Whether the monthly data limit of the current year in yearly plan mode is enabled.	true
++WorkMode	char[32]	Null	Wireless working mode. Value reference: WirelessNetMode.	"WCDMA"
++DailNumber	char[16]	O	SIM1 dialing parameter: Dialing number.	"#777"

			The dialing parameter of the public network SIM card of the three major domestic operators are as follows: *98*1# China Mobile *99# China Unicom China Telecom #777 China Telecom	
++Activate	bool	Null	Indicates whether it was activated by voice or SMS. Start the connection when it was activated and enabled, otherwise, the connection is closed.	true
++CardNum	char[32]	Null	Card number.	"18655667788"
++IMEI	char[16]	Null	15-digit IMEI code.	"357030026314449"
++PIN	char[16]	Null	SIM1 PIN code for unlock.	"12345667"
++ICCID	char[32]	O	Integrated circuit card identification code, that is, the SIM card number consisting of 20 numbers.	"89860116836014532534"
++AntNumber	uint8	O	The number of antennas that can be set. 0 indicates that the antenna of the current module cannot be configured. An integer greater than 0 indicates the number of antennas supported by the current module of the device. Read-only client.	0
++AntMode	enumchar[32]	O	Antenna mode.	"Omnidirectional"

			Enumchar[32]{
			“Omnidirectional": Omnidirectional mode.
			“Directional": Directional mode.
			}	

++MTU	int	O	Set the MTU value of the cellular network adapter.	1500

++AntSwitchMode	enumchar[16]	O	Set the antenna working module.
			Enumchar[16]{
			“Manual": Manual switching mode.
			“Auto": Automatic switching mode.
			}
				"Manual"

++CellularRssiReport	object	O	Configure auto reporting of cellular network signals.	
+++Enable	bool	O	Enable auto reporting of cellular network signals.	false
+++Threshold	uint8	O	Reporting threshold of the signal change. When the change reaches this threshold, it will be reported. Unit: dBm.	5
++SmartSwitchSimCard	object	O	Intelligently switch the SIM card.	
+++Enable	bool	O	Enable SIM card intelligent switch.	true
++RoamingEnable	bool	O	Enable cellular network roaming. true: Supports roaming; false: Does not support roaming.	true
Complete Example				

{
"3G" : {
"Enable" : true,
"Index" : 0,				

```
            "KeepAlive" : 30,
            "APN" : "CTNET",
            "AuthMode" : "No",
            "UserName" : "card",
            "Password" : "card",
            "AutoDial" : true,
            "TimeSection" : [
["1 00:00:00-24:00:00",
"0 00:00:00-24:00:00",
"0 00:00:00-24:00:00"
"0 00:00:00-24:00:00"
"0 00:00:00-24:00:00"
"0 00:00:00-24:00:00"
], …, []],
            "3GFluxTactic" : 0,
            "3GFluxUp" : 30000,
            "3GFlux" : 0,
            "Day3GFluxTactic" : "ByFlux",
            "Day3GFluxUp" : 100,
            "Day3GFluxUse" : 100,
            "Day3GFluxAction" : "3GNetDown",
            "WorkMode" : "WCDMA",
            "DailNumber" : "#777",
            "Activate" : true,
            "CardNum" : "18655667788",
            "IMEI" : "357030026314449",
            "PIN" : "12345667",
            "ICCID": "89860116836014532534",
            "MTU": 1500,
            "AntSwitchMode": "Manual",
             "CellularRssiReport": {
                  "Enable": false,
                  "Threshold": 5
             },
             "SmartSwitchSimCard": {
                  "Enable": true
             }
      }
}
```

5.12.4 Configuring IPv6

Permission	AuthNetCfg			
Parameter	**Type**	**Required**	**Description**	**Example**
IPv6	object	Null	Configure all network interfaces, and use network port names for indexing.	
+Enable	bool	Null	Enable IPv6.	true
+eth0	object	Null	The network port configuration. Each network port corresponds to a configuration.	
++LinkLocalAddress	char[64]	Null	IPv6 local link address automatically allocated by the system. Read-only. The address can only be directly connected. Gateway is not needed.	"fe80:215:c5ff:fe5f:b39b/64"
++GlobalAddress	char[128]	Null	If the network to which the device is connected has a router that supports IPv6 and the router is configured with auto allocation of stateless address, the system automatically configures a global address for the NIC. Read-only.	"2001:250:3000:3ca0:215:f2ff:fe5d:23fc/64"
++IPAddress	char[40]	Null	IP address.	"2001:250:3000:1::1:2"
++Prefix	uint8	Null	Network prefix. Range: [1–128]	112
++DhcpEnable	bool	Null	Enable DHCPv6 to automatically obtain the address or not.	false
++DefaultGatewa	char[40]	Null	Default gateway.	"2001:250:3000:1::1:1"

y			Required to access the GlobalAddress.	
++DnsServerEnable	bool	Null	Use DNSv6 service or not.	false
++DnsServers	char[2][40]	Null	DNSv6 server address. Different DNS addresses can be configured for each NIC, but only the DNS of the default NIC is saved to the system.	["2001:da8:2000:2017::33", "2001:da8:2000:2193::33"]
+eth1	object	Null	Other network port configurations.	
+bond0	object	Null	Virtual network port configuration. The format is the same as that of bond0 in Network.	
+lte0	object	O	Cellular network port configuration. The format is the same as that of eth0.	

Complete Example


```
{
    "Enable" : true,
    "eth0" : {
        "LinkLocalAddress" : "fe80:215:c5ff:fe5f:b39b/64",
        "GlobalAddress" : "2001:250:3000:3ca0:215:f2ff:fe5d:23fc/64",
        "IPAddress" : "2001:250:3000:1::1:2",
        "Prefix" : 112,
        "DhcpEnable" : false,
        "DefaultGateway" : "2001:250:3000:1::1:1",
        "DnsServerEnable" : false,
        "DnsServers" : ["2001:da8:2000:2017::33", "2001:da8:2000:2193::33"]
    },
    "eth1" : {},
    "bond0" : {}
}
```

5.13 Configuring Advertisement

5.13.1 Advertising Resource List

Request URL	http://<server>/cgi-bin/api/VideoOutput/list			
Method	POST			
Request Params (JSON format in body)				
Name	**Type**	**R/O**	**Description**	**Example**
path	char[260]	R	Path	"/PublishFilePath/2010/8/11/dav"
Request Example				
{ "path": "/PublishFilePath/2010/8/11/dav" }				

Response Params (JSON format in body)				
Name	**Type**	**R/O**	**Description**	**Example**
elementInfo	object[]	O	File element information.	
+type	enumchar[16]	O	File element type. enumchar[16]{ "File": File. "Directory": Directory. }	"File"
+file	object	O	File information (valid when Type= “File”).	
++FileType	char[64]	O	File node type. “Wireshark/tcpdump". See Wireshark packet capture file type.	"Wireshark/tcpdump"
++CreateTime	char[20]	O	Time when the file was created.	"2010-4-15 9:58:32"
++ModifyTime	char[20]	O	Time when the file was modified.	"2010-4-15 9:58:32"
++Size	double	O	File size. The decimal part is meaningless. Unit: Byte.	1873.0
++path	char[260]	O	Relative path.	"/PublishFilePath/2010/8/11/dav.jpg"
++Desc	char[128]	O	Customized file description.	"xxxxxxx"
+directory	object	O	Directory information (valid when Type = “Directory”).	

++CreateTime	char[20]	O	Time when the directory was created. Format: “Y-M-D H-M-S”.	"2010-4-15 9:58:32"
++path	char[260]	O	Relative path.	"/PublishFilePath/ 2010/8/11/dav"

Response Example


```
{
    "elementInfo": [{
        "type": "File",
        "file": {
            "FileType": "Wireshark/tcpdump",
            "CreateTime": "2010-4-15 9:58:32",
            "ModifyTime": "2010-4-15 9:58:32",
            "Size": 1873.0,
            "path": " /PublishFilePath/2010/8/11/dav.jpg",
            "Desc": "xxxxxxx"
        },
        "directory": {
            "CreateTime": "2010-4-15 9:58:32",
            "path": " /PublishFilePath/2010/8/11/dav"
        }
    },…{}]
}
```


5.13.2 Deleting Advertising Resources

Request URL	http://<server>/cgi-bin/api/VideoOutput/removeFiles			
Method	POST			
Request Params (JSON format in body)				
Name	**Type**	**R/O**	**Description**	**Example**
fileName	char[32][128]	R	File name.	["a.dav", "b.dav"]
Request Example				


```
{
    "fileName": ["a.dav", "b.dav" ]
}
```


Response Params (JSON format in body)				
Name	**Type**	**R/O**	**Description**	**Example**
Response Example				


```
{}
```

5.13.3 Releasing Advertising Files

Request URL	http://<server>/cgi-bin/api/VideoOutput/deliveryFile			
Method	POST			
Request Params (JSON format in body)				
Name	**Type**	**R/O**	**Description**	**Example**
Port	int	O	The video output port to which the advertising file is released. The value represents the subscript of the VideoOut configuration.	0
Number	int	O	The current advertisement plan number. The caller can use this number to set up different advertisement plans.	0
TimeSection	TimeSchedule	O	Advertisement playing period. It is a two-dimensional array in which the first 7 elements correspond to 7 days of a week, and the eighth element corresponds to holidays, with a maximum of 6 periods per day. The eighth element can be left blank or entered as null, indicating that the holiday period is not supported.	[["1 00:00:00-07:00:00", "2 09:00:00-17:30:00", "3 17:30:00-23:59:59"], …, [ ],]
Enable	bool	O	Enable play function.	true
Name	char[128]	O	Advertisement name.	“Happly holliday”
StartTime	char[20]	O	Play start time.	"8/10/2016 10:00:00 AM"
EndTime	char[20]	O	Play end time. It is only valid for loop playback.	"8/10/2016 12:00:00 PM"
Mode	char[16]	O	Video play mode. Once: Play each file in the list once. Repeat: Play the file in	"Once"

			the list repeatedly until the EndTime ends. Alone: Cut-in (play alone)	
Type	char[16]	R	Operation type. If the value is not entered, perform Replace by default. Send the advertisement again after Replace is cleared. add: Add. Remove: Delete. Based on the ID field of FileList. Clear: Clear.	"Replace"
FileList	object[]	O	Video file list.	

+FileType	enumchar[16]	O	File type.
			enumchar[16]{
			“Video”: Video files.
			“Image”: Image files.
			“Audio”: Audio files.
			}
				"Video"

+PlayWithMode	int	O	The default value is 1. File mode. 1: Recognition mode. 2: Information release mode. 3: Home page mode. The file can only be played in the corresponding working mode. The recognition mode is the standard function of the program, which will automatically preform face recognition when detecting a human face. The information release mode requires human-device interaction before recognition.	1
+URL	char[128]	O	File resource address.	"ftp://192.168.1.108/1.dav"
+URLEx	char[512]	O	File extension resource address: The original	"http://10.35.81.187:8927/1952fdd7-48e8-

			URL field is too short, and not enough for the supporting ICC platform (currently it has reached 160 bytes), which needs to be increased. The maximum length of URLEx is 512 bytes. Compatibility logic of platform: When the Support field of the device capability interface VideoOutput.getAdvertisementCaps returns true, the real download address is filled in the URLEx field, and the URL field is filled with an empty string. Compatibility logic of device: If the URL is not an empty string, the URL is used directly; only when the URL field is an empty string and URLEx is not an empty string, the URLEx field is used.	11eb-b015-08eded28a344/20210315/1/cdce2e98-855d-11eb- ac3f-08eded28a344.jpg?token=fe65ac56-ff42-4597-80f4-33e571deab22"
+Sustain	int	O	Stay time of each image, which is valid only when FileType is Image. Unit: Second.	5
+TimeSection	TimeSchedule	O	Advertisement playing period. It is a two-dimensional array in which the first 7 elements correspond to 7 days of a week, and the eighth element corresponds to holidays, with a maximum of 6 periods per day. The eighth element can be left blank or entered as null,	[["1 00:00:00-07:00:00", "1 09:00:00-17:30:00", "1 17:30:00-23:59:59"], …, [ ],]

			indicating that the holiday period is not supported.	
+Size	int	O	Files size. Unit: Byte. It is convenient to verify whether the file can be downloaded successfully before downloading. The actual download might fail, because the space limit of the device folder (used to save advertisements) has been reached If there is no such field, no verification is performed.	102400
+ID	int	O	File number.	3

Request Example


```
{
    "Port": 0,
    "Number": 0,
    "TimeSection": [ ["1 00:00:00-07:00:00", "2 09:00:00-17:30:00", "3 17:30:00-23:59:59" ], …, [ ], ],
    "Enable": true,
    “Name": "Happy holiday",
    "StartTime": "2016-08-10 10:00:00",
    "EndTime": "2016-08-10 12:00:00",
    "Mode": "Once",
    "Type": "Replace",
    "FileList": [{
        "FileType": "Video",
        "PlayWithMode": 1,
        "URL": "ftp://192.168.1.108/1.dav",
        "URLEx": "http://10.35.81.187:8927/1952fdd7-48e8-11eb-b015-08eded28a344/20210315/1/cdce2e98-855d-11eb- ac3f-08 eded28a344.jpg?token=fe65ac56-ff42-4597-80f4-33e571deab22",
        "Sustain": 5,
        "TimeSection": [ ["1 00:00:00-07:00:00", "1 09:00:00-17:30:00", "1 17:30:00-23:59:59" ], …, [ ], ],
        "Size": 102400,
        "ID": 3
    },…{}]
}
```


Response Params (JSON format in body)				
Name	**Type**	**R/O**	**Description**	**Example**

Response Example
{}

5.13.4 Uploading Advertising Resources

Request URL	http://<server>/cgi-bin/FileManager.cgi?action=uploadFile			
Method	GET			
Request Params (key=value format in url)				
Name	Type	R/O	Description	Example
fileName	char[]	R	Uploaded file name.	xxxxxx.bmp
Path	char[]	R	File upload path.	/upload_pic
Request Example				
POST http://192.168.1.108/cgi-bin/FileManager.cgi?action=uploadFile&fileName=xxxxxx.bmp&Path=/upload_pic HTTP/1.1 Host: 192.168.1.108 Connection: keep-alive Content-Type: multipart/form-data;boundary=----------8655433224198 Content-Length: xxxxxxxxx ----------8655433224198 Content-Disposition:form-data;name="upload"; filename="xxxxxx.bmp" Content-Type: image/jpeg or application/x-MS-bmp photo data…. ----------8655433224198--				

5.13.5 Searching for Advertising Files Sent to Devices

Request URL	http://<server>/cgi-bin/api/VideoOutput/queryDeliveredFile			
Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
Request Example				
{}				

Response Params (JSON format in body)				
Name	Type	R/O	Description	Example
Info	Object[]	R	Transmitted file information.	
+Enable	bool	O	Enable play function.	true
+Number	int	R	The current advertisement plan	0

			number. The caller can use this number to set up different advertisement plans.	
+TimeSection	TimeSchedule	R	Advertisement playing period. It is a two-dimensional array in which the first 7 elements correspond to 7 days of a week, and the eighth element corresponds to holidays, with a maximum of 6 periods per day. The eighth element can be left blank or entered as null, indicating that the holiday period is not supported.	[["1 00:00:00-07:00:00", "2 09:00:00-17:30:00", "3 17:30:00-23:59:59"], …, [ ],]
+Name	char[128]	O	Advertisement name.	“Happy holiday”
+StartTime	char[20]	O	Play start time.	"8/10/2016 10:00:00 AM"
+EndTime	char[20]	O	Play end time.	"8/10/2016 12:00:00 PM"
+Mode	char[16]	O	Video play mode. Once: Play each file in the list once. Repeat: Play the file in the list repeatedly until the EndTime ends. Alone: Cut-in (play alone)	"Once"
+PlayTimes	int	O	File play times. It is valid only when the mode is Once.	0
+FileList	object[]	O	Video file list. Up to 20 file lists are supported.	
++Downloaded	bool	O	Whether the file has been downloaded to the device.	true
++FileType	enumchar[16]	O	File type. enumchar[16]{ “Video”: Video files. “Image”: Image files. “Audio”: Audio files.	"Video"

			}	
++LocalPath	char[128]	O	The path where the file is downloaded to the device.	"/PublishFilePath/1.dav"
++Sustain	int	O	Stay time of each image, which is valid only when FileType is Image. Unit: Second.	5
++URL	char[512]	O	The resource address of the file, and the maximum length is 512 bytes. It has the same length as the VideoOutput.deliveryFile interface (used by video intercom and delivered at the same time).	"ftp://192.168.1.108/1.dav"
++TimeSection	TimeSchedule	O	Advertisement playing period. It is a two-dimensional array in which the first 7 elements correspond to 7 days of a week, and the eighth element corresponds to holidays, with a maximum of 6 periods per day. The eighth element can be left blank or entered as null, indicating that the holiday period is not supported.	[["1 00:00:00-07:00:00", "1 09:00:00-17:30:00", "1 17:30:00-23:59:59"], …, [ ],]
++Status	int	O	File status. 0 means that the file is normal; 1 means that the file is not supported; 2 means that the file format is incorrect; 3 means that the file is damaged; 4 means that the file is too large; 5 means that the file is too small; 6 means that the file has been deleted; 7 means that the file is being deleted; 8	1

			means that the file has been saved; 9 means that the file is being saved; 10 means that the file is being edited.	
++Size	int	O	Files size. Unit: Byte.	102400
++ID	int	O	File number.	3

Response Example


```
{
    "Info":[{
        "Enable": true,
        "Number": 0,
        "TimeSection": [ ["1 00:00:00-07:00:00", "2 09:00:00-17:30:00", "3 17:30:00-23:59:59" ], …, [ ], ],
        “Name": "Happy holiday",
        "StartTime": "2016-08-10 10:00:00",
        "EndTime": "2016-08-10 12:00:00",
        "Mode": "Once",
        "FileList": [{
            "Downloaded": true,
            "FileType": "Video",
            "LocalPath": "/PublishFilePath/1.dav",
            "Sustain": 5,
            "URL": "ftp://192.168.1.108/1.dav",
            "TimeSection": [ ["1 00:00:00-07:00:00", "1 09:00:00-17:30:00", "1 17:30:00-23:59:59" ], …, [ ], ],
            "Size": 102400,
            "ID": 3
        },…{}]
    },…{}]
}
```


5.13.6 Configuring Advertisement Display for Access Control Devices

Software specification	Media/display			
Permission	admin permissions			
Parameter	**Type**	**Required**	**Description**	**Example**
AccessDisplay	Object	R	Access control device display configuration	
+AdvertiseSplitTy	uint32	O	Advertisement split	1

pe			screen mode: 0: Do not split screen. 1: Display the advertisement in the front (left or top). 2: Display the advertisement on the back (right or bottom) 3. Preview mode. Only uploaded advertisement images are displayed.	
+AccessDisplayObject	uint32	O	Access control display subject. 0: Display the advertisement content. 2: Display the text of announcement.	0
+AdvertiseDisplay	object	O	Device advertisement display configuration in advertising split screen mode.	
++AdvertiseVideoDisplay	uint32	O	How to display video advertisements in the split-screen mode. 0: Original scale; 1: Full screen.	0
++AdvertisePicDisplay	uint32	O	How to display the image advertisements in the split screen mode. 0: Original scale; 1: Full screen.	0
+AdvertiseSplitList	object[]	O	List of advertisement split screen modes. Up to 32 is supported.	
++AdvertiseSplitID	uint32	O	Advertising split screen mode. 0: Do not split screen. 1: Display the advertisement in the front (left or top). 2:	0

			Display the advertisement in the back (right or bottom). 3. Preview mode. Only display the uploaded advertisement images.	
++AdvertiseSplitName	char[256]	O	The name of the advertisement split screen mode. The device and webpage display the type of the advertisement mode. --- We recommend that you use mapping by ID as needed. The name might not be translated.	“Preview mode"

Complete Example

```
{
    "HomePage": {
        "ButtonMainMenuVisible" : false,
        "ButtonPasswordVisible" : true,
        "ButtonQRCodeVisible" : true,
        "ButtonCallVisible" : true,
        "ButtonCallType" : 1
    }
}
```


5.13.7 Advertisement Welcome Words Database

Request URL	http://<server>/cgi-bin/recordUpdater.cgi?action=insert&name=Announcement			
Method	GET			
Request Params (key=value format in URL **)**				
Name	**Type**	**R/O**	**Description**	**Example**
Content	string	O	Content	stringData
ExpirTime	string	O	Announcement expiration time.	2012-01-01%2012:00:00
IssueTime	string	O	Announcement release time.	2012-01-01%2012:00:00
Title	string	O	Title	Anounce1
User	string	O	The room number to	101

			which the announcement will be released to.	

State	int	O	Announcement status.
			Enumint{
			0: Initial state (not sent);
			1: Already sent;
			2: Expired.
			}
				1

ReadFlag	int	O	Whether the announcement has been viewed.
			Enumint{
			0: Unread;
			1: Read.
			}
				0

BackgroundPicture	uint32	O	You can select a background image for access control announcements (a specific image is bound to a device). 0: Image 1; 1: Image 2; 2: Image 3.	0
Request Example				
http://192.168.1.108/cgi-bin/recordUpdater.cgi?action=insert&name=Announcement&Content=stringData&ExpirTime=2012-01-01%2012:00:00&IssueTime=2012-01-01%2012:00:00&Title=Anounce1&User=101&State=0&ReadFlag=0				

Response Params (key=value format in body **)**				
Name	**Type**	**R/O**	**Description**	**Example**
recno	int	R	New record index.	232
Response Example				
recno=232				

5.14 Configuring Intercom

5.14.1 Configuring SIP

Permission	AuthNetCfg			
Parameter	**Type**	**R/O**	**Description**	**Example**
SIP	object	Null	SIP protocol configuration.	

+AccoutName	char[64]	Null	Account name.	"dahua"
+SIPServer	char[64]	Null	SIP server. IP address or domain name.	"mysipsercer.com"
+SIPServerPort	uint32	Null	SIP server port number.	5060
+OutboundProxy	char[40]	Null	Proxy server. IP address or domain name.	"10.12.9.41"
+OutboundProxyID	char[16]	Null	Proxy server ID.	"123"
+OutboundProxyPort	uint32	Null	Proxy server port number.	5060
+UserID	char[16]	Null	User account ID. Generally a phone number.	"24204301"
+UserType	int	Null	User type.	0
+AuthID	char[64]	Null	Identity authentication ID.	"24204301"
+AuthPassword	char[64]	Null	Identity authentication password.	"1234"
+STUNServer	char[64]	Null	STUN (Simple Traversal of UDP over NATs) server. IP address or domain name.	"10.12.9.40"
+RegisterRealm	char[16]	Null	Registration domain.	"dahua"
+RegExpiration	uint	Null	Registration interval, in seconds.	3600
+LocalSIPPort	uint16	Null	Local SIP port. 0-65535.	5060
+LocalRTPPort	uint16	Null	Local RTP port. 0-65535.	5004
+UnregisterOnReboot	bool	Null	Restart to delete registration information. true: Delete. false: Do not delete.	true
+DefaultCallNumber	char[16]	Null	Default call number.	"12345678"
+MediaDetail	object	Null	Media configuration.	

++VideoStream	enumchar[32]	Null	Video stream.
			enumchar[32]{
			"Main": Main stream.
			"Extra1": Sub stream 1.
			"Extra2": Sub stream 2.
			"Extra3": Sub stream 3
			}
				"Main"

++AudioStream	enumchar[32]	Null	Audio stream.
			enumchar[32]{
			"Main": Main stream.
			"Extra1": Sub stream 1.
			"Extra2": Sub stream 2.
			"Extra3": Sub stream 3.
			}
				"Main"

+RouteEnable	bool	Null	Enable SIP cross-router or not.	true

+Route	char[][128]	Null	Router address, which can be IP address or domain name.	["", " Ir", …]
+SIPServerLoginUserName	char[64]	Null	Username used to log in to VTNC. For intelligent building only.	"admin"
+SIPServerLoginPWD	char[64]	Null	Password used to log in to VTNC. For intelligent building only.	"admin"
+IsMainVTO	int	Null	Whether the door station is a standby server. For intelligent building only.	1
+SIPServerRedundancy	char[40]	Null	Standby server IP address. For intelligent building only.	"127.0.0.1"
+SIPServerRedundancyUserName	char[64]	Null	Standby server login username. For intelligent building only.	"admin"
+SIPServerRedundancyPassWord	char[64]	Null	Standby server login password. For intelligent building only.	"admin"
+AnalogNumberStart	char[64]	Null	The start number of the analog indoor monitor supported in the analog system. For intelligent building only.	"100"
+AnalogNumberEnd	char[64]	Null	The end number of the analog indoor monitor supported in the analog system. For intelligent building only.	"105"
+UserEnable	bool	Null	Enable registration. For intelligent building only. true: Register to the SIP server. false: Do not register to the SIP server.	true
+SIPServerID	char[16]	O	SIP protocol stack server ID, which cannot be the same as UserID.	"8000"

Complete Example


```
{
    "AccoutName" : "dahua",
    "SIPServer" : "mysipsercer.com",
    "SIPServerPort" : 5060,
    "OutboundProxy" : "10.12.9.41",
    "OutboundProxyID" : "123",
    "OutboundProxyPort" : 5060,
    "UserID" : "24204301",
    "UserType" : 0,
    "AuthID" : "24204301",
```

```
        "AuthPassword" : "1234",
        "STUNServer" : "10.12.9.40",
        "RegisterRealm" : "dahua",
        "RegExpiration" : 3600,
        "LocalSIPPort" : 5060,
        "LocalRTPPort" : 5004,
        "UnregisterOnReboot" : true,
        "DefaultCallNumber" : "12345678",
        "MediaDetail" : {
            "VideoStream" : "Main",
            "AudioStream" : "Main"
        }//End of MediaDetail
        "RouteEnable" : true,
        "Route" : ["", " lr", …]
        "SIPServerLoginUserName" : "admin",
        "SIPServerLoginPWD":"admin"
        "IsMainVTO":1
        "SIPServerRedundancy":"127.0.0.1"
        "SIPServerRedundancyUserName":"admin"
        "SIPServerRedundancyPassWord":"admin"
        "AnalogNumberStart":"100"
        "AnalogNumberEnd":"105"
        "UserEnable" : true,
}//End of SIP
```

5.14.2 Server Type

Permission	admin permissions			
Parameter	**Type**	**Required**	**Description**	**Example**
Registar	object[]	Null	One-dimensional array. Each subscript represents a registration server.	
+RegistarName	char[]	Null	Registration server name. The name of different registration servers must be unique and cannot be repeated. (Note: If it is VTH, also take the IP	"VTS"

			address and port in this configuration to connect to VTO).	
+Enable	bool	Null	Register to the registration server or not.	true
+ServerType	enumchar[32]	Null	Server type. It is only used during SIP intercom, but not used during VT intercom.(Black protocol becomes positive). Enumchar[32]{ "VTO" "H500" "VTNC" "ZYCOO" "ThirdParty" "3CXSystem" "Asterisk"}	"H500"
+GeneralServerInfo	object	Null		
++Address	char[40]	Null	IP address.	"10.22.5.254"
++Port	uint	Null	Registration port number.	12801
++Password	char[64]	Null	Password registered to the registration server, which will be used when registration authentication is required.	"BFCB43AABBBA2594CA9197D36"

Complete Example

```
[ {
      "RegistarName" : "VTS",
      "Enable" : true,
      "ServerType" : "H500",
      "GeneralServerInfo" : {
            "Address" : "10.22.5.254",
            "Port" : 12801,
            "Password" : "BFCB43AABBBA2594CA9197D36"
      }
},…,{}]
```

5.14.3 VTO Basic Information

Permission	admin permissions			
Parameter	**Type**	**Required**	**Description**	**Example**
VTOBasicInfo	object	Null	Basic VTO information.	
+Number	char[16]	Null	Door station number.	"6901"

+DeviceType	enumint	Null	Device type:
			Enumint{
			1: Villa station.
			2: Door station.
			3: Lift control station.
			4: Modular door station.
			5: Second confirmation station.
			6: Face registration device.
			7: Controller.
			9: VTA device
			}
				1

+Type	enumint	Null	Door station type. It is valid only when DeviceType = 2.
			Enumint{
			1: Unit door station.
			2: Fence station.
			3: Face capture device.
			4: Visitor access controller.
			}
				1

+AnalogVersion	char[]	Null	Analog system version.	"1.0"
+FaceDetect	bool	Null	Enable face target recognition. true: Enable.	false
+Position	int32	Null	The location of the	-1

			VTO floor. Compatible with old programs 0: Invalid. 1: The first floor above the ground. Other positive n: Floor n-1 under the ground. Negative number -n: Floor n + 1 above the ground.	
+IsCustomAuthID	bool	Null	Whether SIP registration AuthID can be customized. true: Enable customization. false: Disable customization. It is false by default.	false
Complete Example				
{ "Number" : "6901", "DeviceType" : 1, "Type" : 1, "AnalogVersion" : "1.0", "FaceDetect" : false, "Position" : -1, "IsCustomAuthID" : false }				

5.14.4 Configuring VTO Floor

Permission	admin permissions			
Parameter	**Type**	**Required**	**Description**	**Example**
Building	object	Null	VTO floor configuration.	
+IssueNumber	char[8]	Null	Phase number. Fixed length: 2 digits.	"11"
+EnableSection	bool	Null	Enable section number.	false
+SectionNumBit	uint8	O	Section length.	2

+SectionNumber	char[8]	Null	Section number. It is 2 digits by default and can be expandable.	"10"
+BuildingNumBit	int	Null	Length of building number.	2
+BuildingNumber	char[8]	Null	Building number. Fixed length: 3 bits.	"101"
+BuildingUnitNumber	char[8]	Null	Unit number of the building. Fixed length: 1 bit.	"6"
+SectionUnitNumber	char[8]	Null	Community unit number. Fixed length: 3 bits.	"101"
+UnitFloorNumber	char[8]	Null	Number of floors of the unit. Fixed length: 2 bits.	20
+FloorPerRoomNumber	char[8]	Null	Number of rooms on each floor. Fixed length: 2 bits.	10
+TotalRoomNumber	uint	Null	Total number of rooms. The maximum value is 9999.	200
+CreateRoom	bool	Null	Enable the function to create room numbers.	false

+SystemType	enumint	Null	System type.
			Enumint{
			1: Digital system.
			2: Analog system.
			}
				1

+BuildingName	char[64]	Null	Door station alias.	"Park"
+BuildingUnitNumberBit	uint16	O	Unit number length.	2

Complete Example


```
{
    "IssueNumber" : "11",
    "EnableSection": false,
    "SectionNumber" : "10",
    "BuildingNumBit" : 2,
    "BuildingNumber" : "101",
    "BuildingUnitNumber" : "6",
    "SectionUnitNumber" : "101",
```

```
    "UnitFloorNumber" : 20,
    "FloorPerRoomNumber" : 10,
    "TotalRoomNumber" : 200,
    "CreateRoom" : false,
    "SystemType" : 1,
    "BuildingName" : "Park"
}
```


5.14.5 VideoTalkContact Database

Permission	admin permissions			
Record Name	"VideoTalkContact"			
Primary Key	"VTLongNumber"			
Parameter	**Type**	**Required**	**Description**	**Example**
RecNo	uint	Null	Record number. Read-only	1234
CreateTime	uint	Null	Start time. UTC seconds, read-only	123456789
FirstName	char[]	Null	First name.	"Dafei"
FamilyName	char[]	Null	Last name.	"Wang"
VTShortNumber	char[16]	Null	Short number for video intercom.	"0101"
VTMiddleNumber	char[16]	Null	Middle number for video intercom.	"11010101"
VTLongNumber	char[32]	Null	Long number for video intercom or serial number of analog indoor monitor.	"330103001101010151"
VTNetAddress	char[40]	Null	Network address for video intercom.	"127.0.0.1"
MacAdress	char[40]	Null	MAC address.	"0A:3E:FF:2A:50:41"
VTOPosition	char[16]	Null	Door number linked with indoor monitor.	"01018001"
VTSlaveBindMode	enumint	Null	Mode when accessing to the analog indoor monitor for video intercom. Enumint{ 0: Use SlaveAddress 1: Use SlaveID +	0

			SlavePort }	
VTSlaveId	uint32	Null	Allocator address when accessing to the analog indoor monitor for video intercom. (Change string to uint32 for consistency)	1258421
VTSlavePort	uint32	Null	Allocator port when accessing to the analog indoor monitor for video intercom. (Change string to uint32 for consistency)	1258421
VTSlaveAddress	char[40]	Null	Address of the analog indoor monitor for video intercom.	"04:b3:01:f7"
NickName	char[32]	Null	Nickname.	"Nick"
Notes	char[32]	Null	Remarks	"Friend"

Type	enumchar[32]	Null	User type.
			Enumchar[32]{
			"VTH": Indoor monitor.
			"VTO": Door station.
			}
				"VTH"

RegisterType	enumchar[32]	Null	Registration method.
			Enumchar[32]{
			"public"
			"local"
			}
				"public"

VTHPassword	char[64]	Null	Registration password	"123456"
VTOBuilding	char[64]	Null	Building number.	"01"
VTOUnit	char[16]	Null	Unit number.	"01"
GroupNumber	char[16]	Null	Group.	"301"
Channel	uint32	Null	Channel number, based on which the mobile phone subscribes to the call notification message.	1
Floors	char[256][4]	O	Floor number (lift	["1","2"…]

			control requests). Up to 256 characters.	
LiftControlByVTH	bool	O	Lift control is triggered by the indoor monitor opening the door.	true
MemberNames	char[10][15]	O	Name of people in the room.	["Zhang San", "Li Si", "Wang Wu"]

Complete Example


```
{
    "RecNo" : 1234,
    "CreateTime" : 123456789,
    "FirstName" : "Dafei",
    "FamilyName" : "Wang",
    "VTShortNumber" : "0101",
    "VTMiddleNumber" : "11010101",
    "VTLongNumber" :    "330103001101010151",
    "VTNetAddress" : "127.0.0.1",
    "MacAdress" : "0A:3E:FF:2A:50:41",
    "VTOPosition" : "01018001",
    "VTSlaveBindMode" : 0,
    "VTSlaveId" : 1258421,
    "VTSlavePort" : 1258421,
    "VTSlaveAddress" : "04:b3:01:f7",
    "NickName" : "Nick",
    "Notes" : "Friend",
    "Type" : "VTH",
    "RegisterType" : "public",
    "VTHPassword" : "123456",
    "VTOBuilding" : "01",
    "VTOUnit" : "01",
    "GroupNumber" : "301",
    "Channel": 1
}
```


5.14.6 Configuring VTO Call Extension

Permission	admin permissions			
Parameter	**Type**	**Required**	**Description**	**Example**
VTOCallInfo	object	Null	VTO call configuration	

			extension.	
+MainVtoIP	char[40]	Null	Main VTO IP.	"10.22.5.189"
+GroupCallEnable	bool	Null	Enable group call.	false
+ManagerNumber	char[]	Null	Main station number.	"94"
+UrgentNumber	char[]	Null	Emergency call number.	"193"
+CallVTSEnable	object	Null	Enable VTS call.	
++TimeSection	TimeSection	Null	VTS call period. Valid means calling VTS. Invalid means calling VTH.	
+MaxExtensionIndex	uint	Null	The maximum indoor unit extension number that can be set on the door station. The number starts from 1.	5
+RoomRule	enumchar[32]	Null	The door station supports two rules: Continuous room number call and discontinuous room number call. Enumchar[32]{ Serial: Continuous room numbers. Some regions use 1 to100 for the room number. Noserial: Discontinuous room numbers, such as 301 and 502.	"Serial"
+Username	char[64]	Null	Third-generation username.	"admin"
+Password	char[64]	Null	Third-generation password.	"xxxx"
+MulticastEnable	bool	Null	When it is enabled, the door monitor sends streams to the multicast address,	true

			and the indoor monitor is added to the multicast address. When it is turned off, the indoor monitor actively pulls streams from the door station.	
+CallTimeSection	char[7][][20]	Null	It is an array that represents 7 days a week. It includes the time periods every day in every month. CallTimeSection is a two-dimensional array. The format of TimeSection is "Enable hour: minute: second-hour: minute: second". There is a space between enabling status and the time information. the enabling status character 1 indicates that the time period is valid, and 0 indicates that the time period is invalid. The first hour: minute: second is the start time, and the second one is the end time. [["1 01:11:00-02:22:00"], ["1 13:13:00-22:22:00"],…[]]	[["0 00:00:00-24:00:00"],[],…]
Complete Example				
{ "MainVtoIP" : "10.22.5.189", "GroupCallEnable" : false, "ManagerNumber" : "94", "UrgentNumber" : "193",				

```
    "CallVTSEnable" : {
        "TimeSection" : ,
    },
    "MaxExtensionIndex" : 5,
    "RoomRule" : "Serial",
    "Username": "admin",
    "Password":"xxxx"

    "MulticastEnable":  true

    "CallTimeSection":[["0 00:00:00-24:00:00"],[],…]
}
```

5.14.7 Configuring Sub Door Stations When Logged in to Main Door Station

Permission	admin permissions			
Parameter	**Type**	**Required**	**Description**	**Example**
DeviceLoginInfo	object[]	Null	Configure sub door stations when you are logged in to the main door station. It is an array. Each element is a sub door station configuration.	
+Address	char[40]	Null	IP address.	"0.0.0.0"
+LongNumber	char[16]	Null	Long number.	"8001"
+Username	char[32]	Null	Username.	"admin"
+Password	char[32]	Null	Password.	"admin"
Complete Example				

```
[{
    "Address" : "0.0.0.0",
    "LongNumber" : "8001",
    "Username" : "admin",
    "Password" : "admin"
}, {}]
```

5.14.8 Configuring VTO Floor Extension

Permission	admin permissions

Parameter	Type	Required	Description	Example
BuildingExternal	object	Null	VTO floor extension configuration.	
+FloorCount	char[]	Null	Total numbers of floors of the unit.	"11"
+RoomCount	char[]	Null	Number of rooms per floor.	"10"
+BeginNumber	char[][16]	Null	Start room number. Start room number of the first floor and the second floor respectively.	["101", "201"]
+TotalBuildingNumber	char[]	Null	Total number of buildings in the compound. Fixed length: 2 bits.	"8"
+BuildingPerUintNumber	char[]	Null	Number of units per building. Fixed length: 1 bit.	"3"

Complete Example

```
{
    "FloorCount" : "11",
    "RoomCount" : "10",
    "BeginNumber" : ["101", "201"],
    "TotalBuildingNumber" : "8",
    "BuildingPerUintNumber" : "3"
}
```

5.15 Configuring Auto Image Event Upload

Config Data Params				
Name	**Type**	**R/O**	**Description**	**Example**
PictureHttpUpload	object	R	Configure the parameters for automatic image event upload.	

+Enable	bool	R	Enable the function or not.	true
+Type	char[16]	O	Authentication type. “basic": Use http basic authentication. “digest": Use http digest authentication.	"digest"
+UploadServerList	object[]	O	List of servers that receives upload information.	
++Address	char[128]	O	Server IP address or domain.	"192.168.1.208"
++Port	int	O	Server port.	80
++UserName	char[32]	O	Username.	"abc"
++Password	char[128]	O	Password.	"123"
++Uploadpath	char[128]	O	Upload path.	"/example/handlepic.php"
++EventType	char[][32]	O	List of uploaded event codes.	["CrossLineDetection","FaceDetection"]
++HttpsEnable	bool	O	Enable https.	true
++AuthEnable	bool	O	Enable authentication.	true
++rall	int	O	Keep-alive period, in seconds.	3

			The http reporting function allows you to configure the heartbeat packet mechanism between the device and the customer server to verify whether the server and the device can be connected.	
+uuid	char[128]	O	Device ID, which can be customized and reported as an extended field in the http upload function.	"djsaiodfjadfng1234454"

5.16 Configuring Auto Event Upload

Config Data Params				
Name	**Type**	**R/O**	**Description**	**Example**
EventHttpUpload	object	R	Configure parameters for automatic event	

			reporting in http mode.	
+Enable	bool	R	Enable the function or not.	true
+Type	char[32]	O	Authentication type. “basic": Use http basic authentication. “digest": Use http digest authentication.	"digest"
+UploadServerList	object[]	O	The list of servers that receives upload information, which can be added and deleted.	
++Address	char[128]	O	Server IP address or domain.	"192.168.1.208"
++Port	int16	O	Server port.	80
++UserName	char[32]	O	Account name.	"abc"
++Password	char[128]	O	Password.	"123"
++Uploadpath	char[128]	O	Upload path.	"/example/handleevt.php"
++EventType	char[128][40]	O	Upload the corresponding event type (currently	["CrossLineDetection","FaceDetection"]

			only used by the front-end devices).	
++HttpsEnable	bool	O	Enable https.	true
++AuthEnable	bool	O	Enable authenticati on.	true
++rall	int	O	Keep-alive period, in seconds. The http reporting function allows you to configure the heartbeat packet mechanism between the device and the customer server to verify whether the server and the device can be connected.	300
+uuid	char[128]	O	Device ID, which can be customized and reported as an extended field in the	"djiasf12324"

			http upload function.	

5.17 Configuring General Information Upload

Parameter	Type	Required	Description	Example
HttpPushGeneral	object	R	General configuration of auto information push.	
+Enable	bool	R	Enable the function or not.	true
+UploadServerList	object[8]	O	Server information.	
++Enable	bool	O	Enable the function or not.	true
++Type	object[10]	O	Push type.	
+++PushType	enumchar[128]	O	Type of pushed data. enumchar[128] { "UserManagerInfor" }	"UserManagerInfor"
+++Uploadpath	char[128]	O	Upload path.	"/test"
++Address	char[128]	O	Upload server IP address or domain.	"192.168.1.108"
++Port	uint16	O	Server port.	80
++UserName	char[32]	O	Username, linked with authentication.	"admin"

++Password	char[128]	O	Password, linked with authentication.	"123456"
++HttpsEnable	bool	O	Enable https.	true
++AuthEnable	bool	O	Enable authentication.	true

5.18 Description of Personnel Information Upload Protocol

Params	Object	R		
+UserID	char[64]	R	User ID	"123456"
+UserName	char[512]	O	Username	"1234"
+Password	char[32]	O	Password	"123456"
+IDCardNo	char[20]	O	ID card number	"123456"
+BinaryDataInfo	object[]	O	Binary data information, up to 20 items	
++Type	int32	O	Data type should be consistent with the online collection type. 0-Iris image (left) 1-Iris features (left) 2-Iris image (right) 3-Iris features (right)	1

			4-Face image 5-Face features 6-Card number (Base64) 7-ID card number (Base64) 8-ID card original image 9-ID card captured image 10-Fingerprint original image 11-Fingerprint features 12-Palmprint white light image (left) 13-Palmprint white light features (left) 14-Palmprint white light image (right) 15-Palmprint white light features (right) 16-Palmprint red light image (left) 17-Palmprint red light features (left) 18-Palmprint red light image (right) 19-Palmprint red light	

			features (right)	
++Offset	int32	O	Offset	100
++Length	int32	O	Length	100
+CardNo	char[5][32]	O	Card number	["12345678","87654321"]
+FingerPrintData	char[10][4096]	O	Fingerprint data	["xxxxxxxxxxxxxx"]
+uuid	char[64]	R	Unique ID	"abc123"
Binary Data	Binary data	R	The binary data of person information is explained in the BinaryDataInfo field in JSON. If the BinaryDataInfo does not exist, the length of the binary data is 0	

6 CGI Common Error Codes

6.1 401 Unauthorized

In the use of CGI, when HTTP digest authentication is used, when the authentication is not successful, 401 Unauthorized will be returned, and a series of information will also be returned, requiring the next request to attach this information. See above about the specific HTTP digest authentication process.

It can be seen from the above that it is normal for the 401 Unauthorized error code to appear for the first time when using CGI. If the error code still appears in the second request, please refer to the HTTP digest authentication process below to reply.

6.2 400 Bad Request

In the coding of WebApp, the 400 Bad Request error code is a general error code, which means that the underlying component or the RPC layer returns false. Therefore, there are many possible situations for the 400 Bad Request error code, such as the device does not support, the message format is incorrect, the message execution fails, and so on.

When the 400 Bad Request error code appears, judge whether the device supports the functions used by CGI in advance. If this function is supported, further check whether the CGI URL format is correct and whether there are unnecessary
spaces. In addition, it is also necessary to check whether the CGI action is correct, for example, whether there is a query event that is not triggered, or there is no such kind of event.

If none of the above conditions exist, please contact technical support for help.

6.3 501 Not Implemented

In the CGI reply, 501 Not Implemented means that the device does not support this function, which is most likely because the device does not have the functions required by CGI.

6.4 404 Not Found

In rare cases, CGI will return 404 Not Found, which means that the URL is not registered into WebSvr. There is a high probability that the entire CGI function is clipped in the packaging environment. You can first confirm whether the device supports CGI.

6.5 403 Forbidden

Means that the account used for CGI authentication is locked. Please make sure that the account used for authentication is correct and the account status is normal.

7 Tools

7.1 Verifying the API on the Browser

This section uses Chrome V92 as the example.

7.1.1 Visiting API with Parameters in the URL

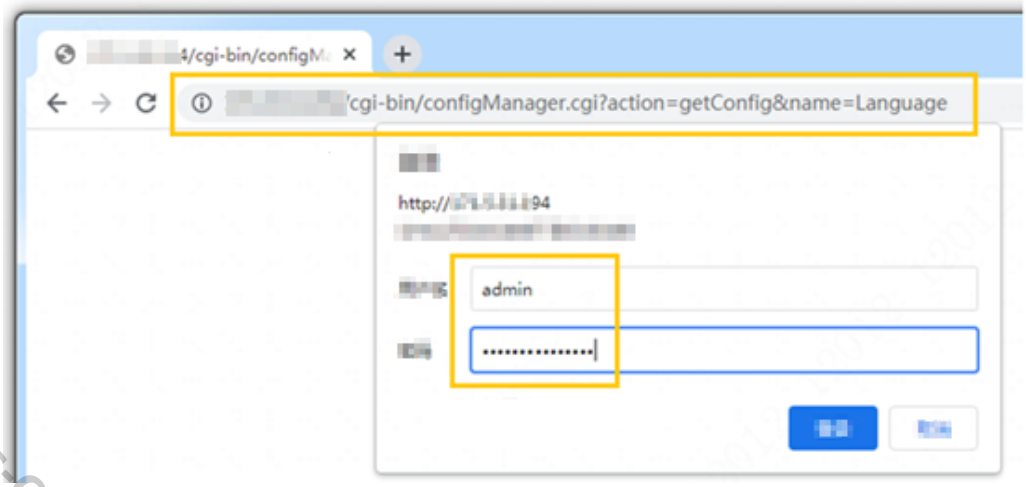
For API with parameters in the URL (for example, the command whose request params is key=value format in URL), you can enter the entire URL to the browser address bar, and then the response (json or key=value in multiple rows) will be displayed on the browser.

CAUTION

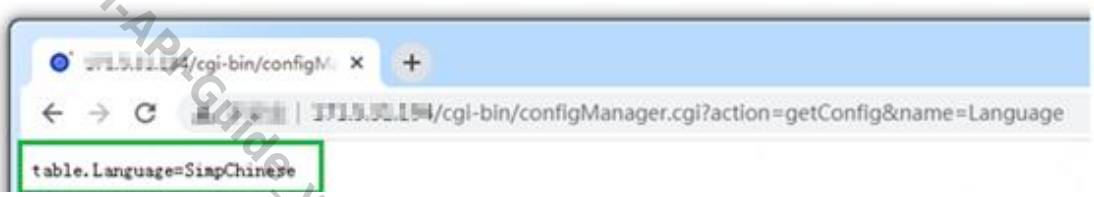
Modifying the language configuration may affect the use of the device. Before modifying the language configuration, query the current language configuration of the device. After modifying the language configuration for testing, you can modify the original language setting again to restore the original state.

For supported languages of the device, run the command of magicBox.cgi?action=getLanguageCaps.

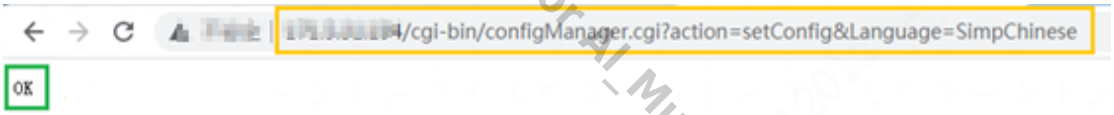
For example, to search for the current language configuration of the device, directly enter the relevant command URL into the browser address bar, and then the login box will appear when you access it for the first time:



After entering the username and password, if the device executes the command successfully, it will return the result, which will be displayed on the page:



To set the current language configuration of the device, directly enter the relevant command URL into the browser address bar. If you have executed the API command of the same device before and entered the user name and password, you will not be prompted to enter the user name and password, and return directly. The corresponding results are displayed on the page:



7.1.2 Visiting API with Parameters in the Body

For the API whose parameters are in the Body, for example, the command whose [Request Params] is [json format in Body] in the protocol document can be accessed directly through the browser and needs to be accessed through a simple html page. However, due to cross-domain access, it is necessary to Turn off your browser's security check.

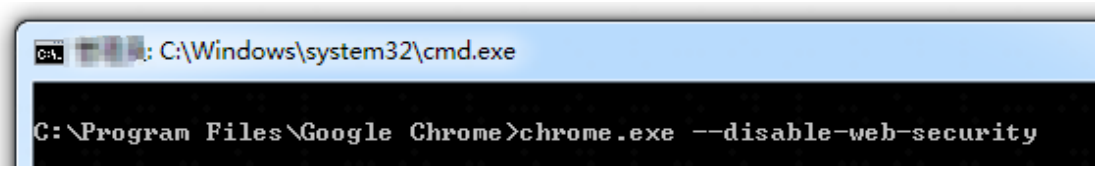
DANGER

Closing the browser's security check may bring security risks. At this time, the browser should only be used in the development environment for API testing and verification, and should not be used in the official production environment, nor should it be used to access other websites. The browser's security check should be re-enabled as soon as possible.

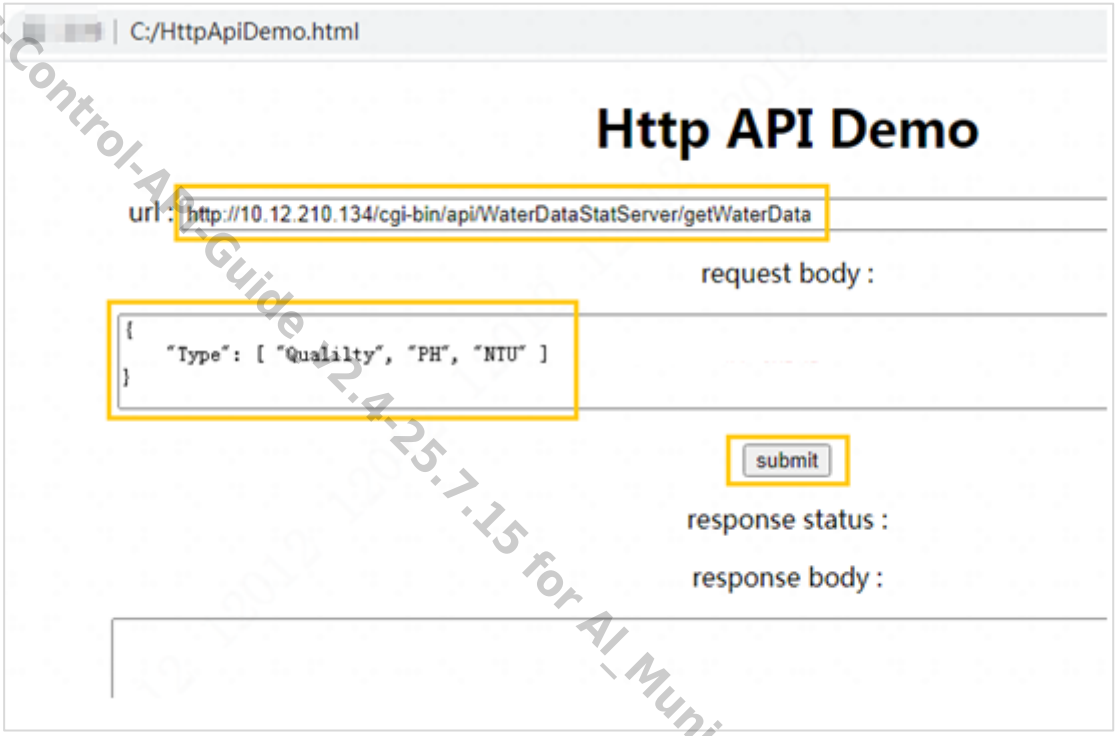
First save the following content as an html file, for example named HttpApiDemo.html and put it in the root directory of the C drive:

```
<!DOCTYPE html>
<html>
<head>
<title>Http API Demo</title>
<script type="text/javascript" >
function do_http_api_request()
{
    w_status = document.getElementById("status")
    w_response = document.getElementById( "replybody" )
    w_status.innerHTML = "requesting..."
    w_response.value = ""
    fetch( document.getElementById("url").value,
        {   method: "POST", credentials: "include",
            body: document.getElementById("reqbody").value } )
    .then( response => {
        return response.text()
    })
    .then( data => {
        w_status.innerHTML = "success"
        w_response.value = data
    })
    .catch( error => { w_status.innerHTML = "error" } )
}
</script>
</head>
<body>
<center>
<h1>Http API Demo</h1>
<form>
<p><span>url : </span><input type="text" id="url" size="110" value="http://192.168.1.108/cgi-bin/magicBox.cgi?action=getSystemInfo" />
<p><span>request body : </span><p><textarea id="reqbody" cols="120" rows="5" ></textarea>
<p><input type="button" id="submit" value="submit" onClick="do_http_api_request()" />
<p><span>response status : </span><span id="status"></span>
<p><span>response body : </span><p><textarea id="replybody" cols="120" rows="10" readonly=true ></textarea>
</form>
</center>
</body>
</html>
```

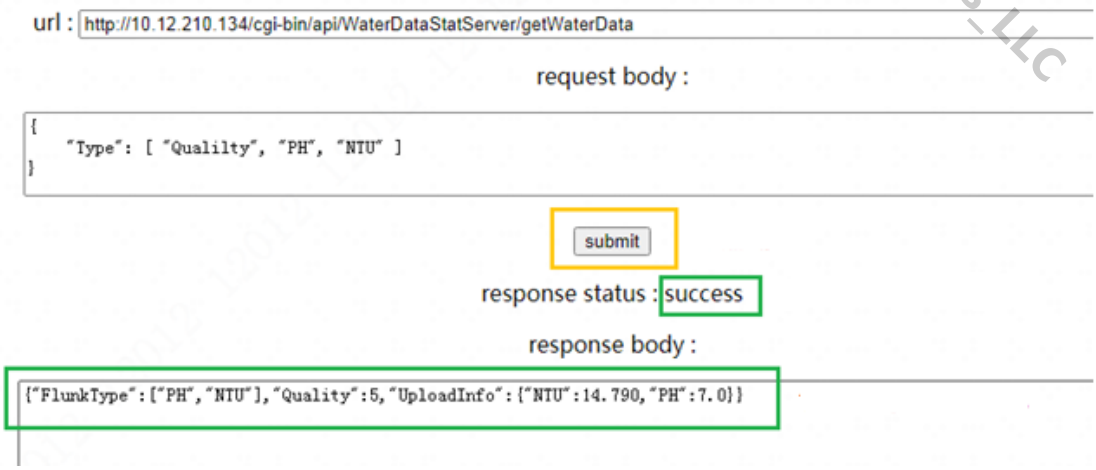
Then open a chrome browser with security check not enabled by adding the startup parameter (--disable-web-security).



Then drag and drop HttpApiDemo.html into the browser, or enter the local address of the page file file:///C:/HttpApiDemo.html in the address bar of the browser, and then fill in the URL address of the API in the url input box on the page, Fill in the json request body in the request body:



Then click the submit button to initiate a request, the status of the response status will be set to requesting..., and then wait until the device successfully executes the return result, the status of the response status will be set to success, and the response body will be filled with the returned json response result:



7.2 Postman Visiting API

7.2.1 Visiting API with Parameters in the URL

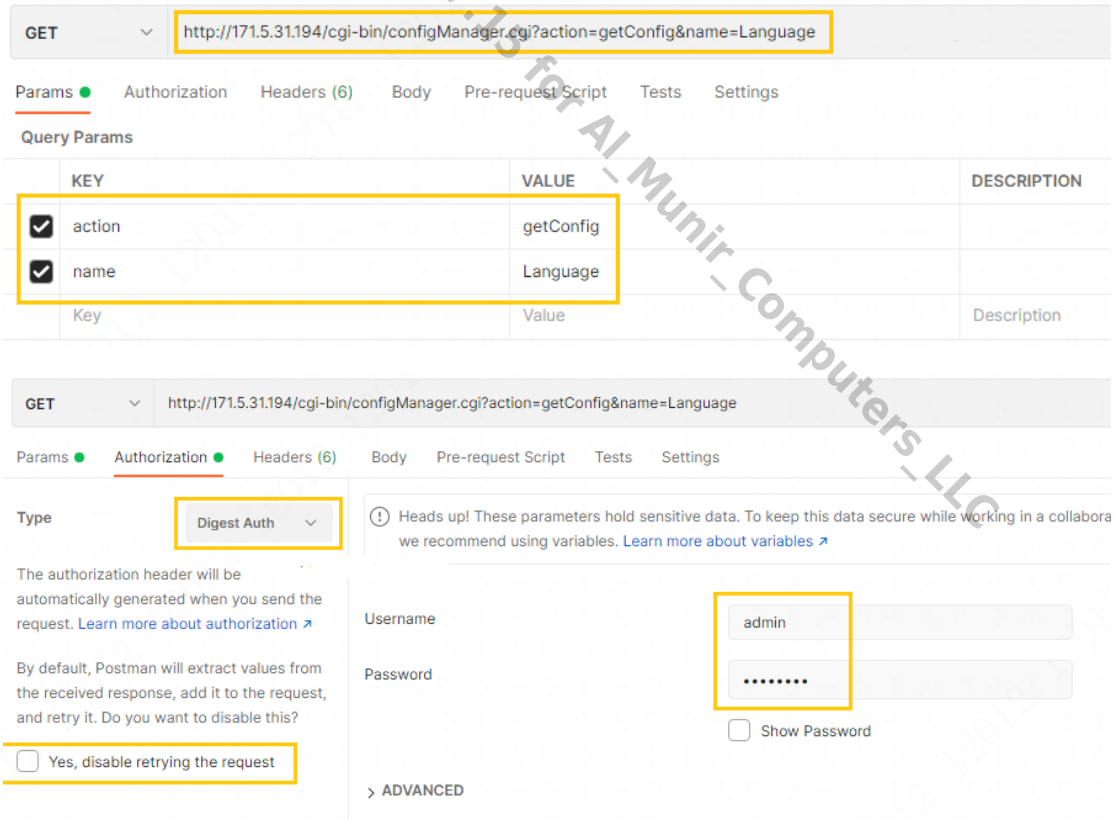
For APIs whose parameters are in the URL, for example, the [Request Params] command in the protocol document is [key=value format in URL]. When using postman, the parameters are filled in under params.

CAUTION

Modifying the language configuration may affect the use of the device. Before modifying the language configuration, query the current language configuration of the device. After modifying the language configuration for testing, you can modify the original language setting again to restore the original state.

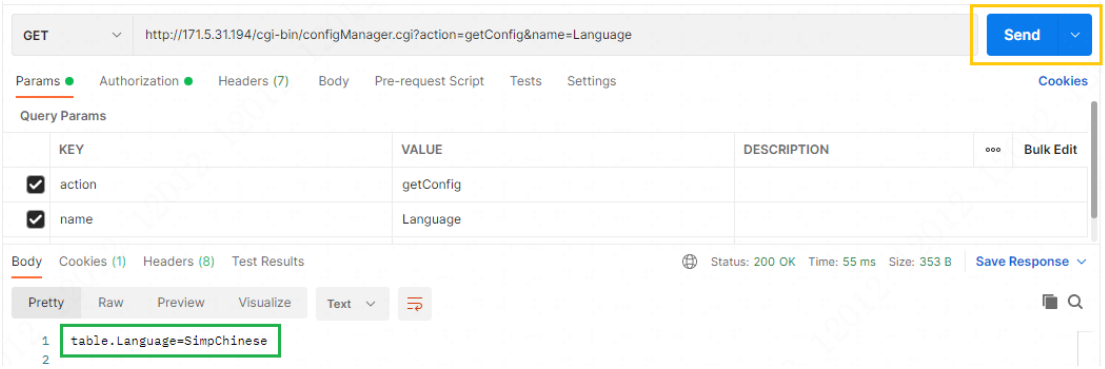
The language supported by the device can be queried through the magicBox.cgi?action=getLanguageCaps command.

For example, to search for the current language configuration of the device, directly enter the relevant command URL into the address input box of postman. The query parameters can be entered directly in the address input box or in Params, and postman will automatically synchronize:

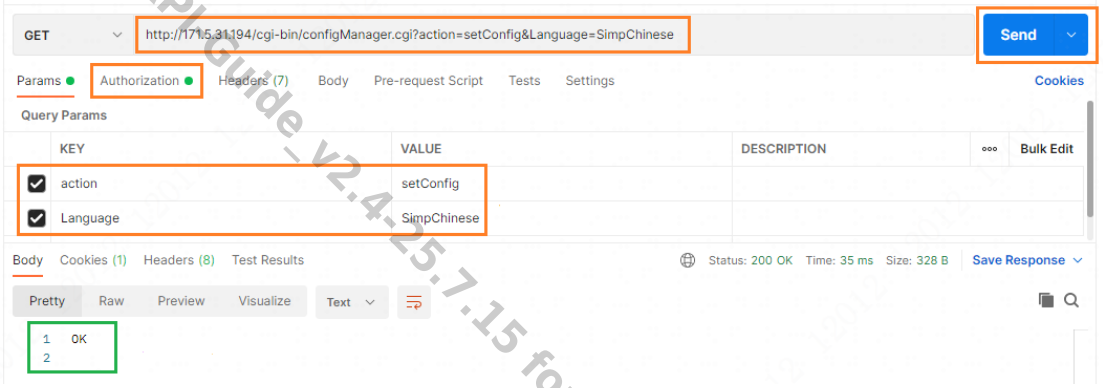


Then switch to the Authorization page and enter the login information:

Then click Send to send the request. If the device executes successfully, it will return the corresponding result:



If you want to set the current language configuration of the device, you can fill in the URL address, query parameters, and authentication information in a similar way, and then click Send to send the command. If the device executes successfully, the corresponding result will be returned for display:

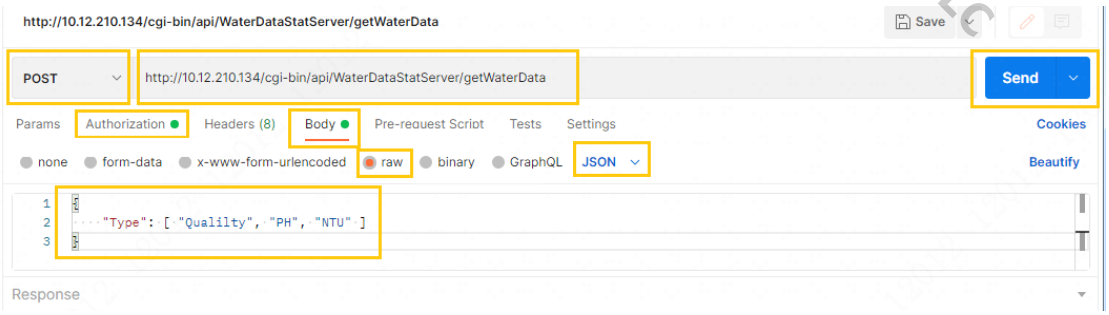


7.2.2 Visiting API with Parameters in the Body

API with parameters in the Body

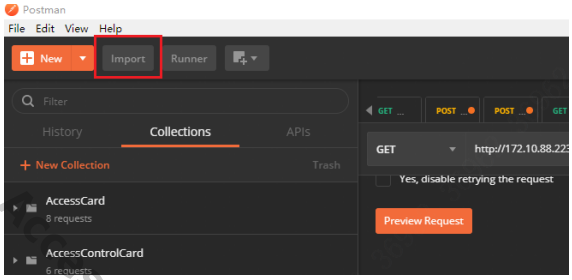
Set the URL and authentication information in a similar way, and increase the settings of the Body:

Then click Send to send the request. If the device executes successfully, it will return the corresponding result:

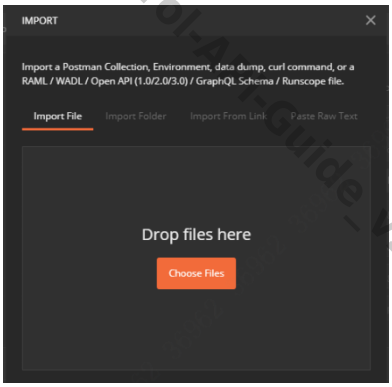


7.3 Postman Json Document Operation

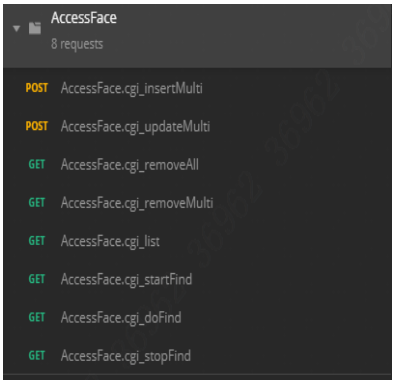
① Click on import in the upper left corner of postman



②Select the Json file to import

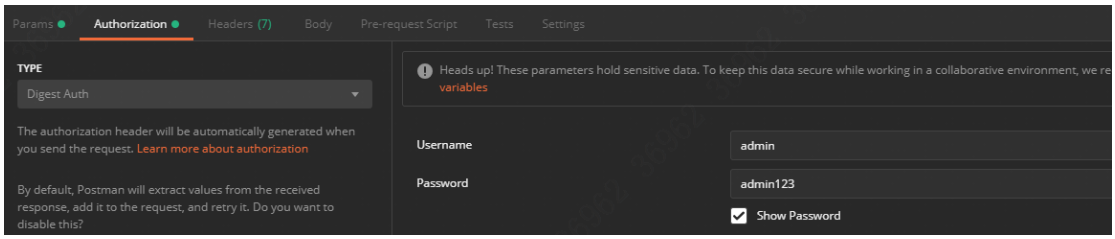


③The generated effect is as follows, click the corresponding method to execute



④Subsequent case modification:

- The IP address corresponding to the case is modified to the on-site IP
- Authorization: Modify username, password



7.4 Postman Case List

AccessControlCard.postman_collection.json

Log Operation.json

FaceInfoManager.postman_collection.json

Second-generation protocol_card operations.json

Second-generation protocol_face operations.json

Second-generation protocol_fingerprint operations.json

Second-generation protocol_personnel operations.json