

API of HTTP Protocol Specification

HTTP API V3.35 – Intelbras

V3.35

2023-07-21

Cybersecurity Recommendations

Mandatory actions to be taken towards cybersecurity

1. Change Passwords and Use Strong Passwords:

The number one reason systems get "hacked" is due to having weak or default passwords. It is recommended to change default passwords immediately and choose a strong password whenever possible. A strong password should be made up of at least 8 characters and a combination of special characters, numbers, and upper and lower case letters.

2. Update Firmware

As is standard procedure in the tech-industry, we recommend keeping NVR, DVR, and IP camera firmware up-to-date to ensure the system is current with the latest security patches and fixes.

"Nice to have" recommendations to improve your network security

1. Change Passwords Regularly

Regularly change the credentials to your devices to help ensure that only authorized users are able to access the system.

2. Change Default HTTP and TCP Ports:

- Change default HTTP and TCP ports for systems. These are the two ports used to communicate and to view video feeds remotely.
- These ports can be changed to any set of numbers between 1025-65535. Avoiding the default ports reduces the risk of outsiders being able to guess which ports you are using.

3. Enable HTTPS/SSL:

Set up an SSL Certificate to enable HTTPS. This will encrypt all communication between your devices and recorder.

4. Enable IP Filter:

Enabling your IP filter will prevent everyone, except those with specified IP addresses, from accessing the system.

5. Change ONVIF Password:

On older IP Camera firmware, the ONVIF password does not change when you change the system's credentials. You will need to either update the camera's firmware to the latest revision or manually change the ONVIF password.

6. Forward Only Ports You Need:

- Only forward the HTTP and TCP ports that you need to use. Do not forward a huge range of numbers to the device. Do not DMZ the device's IP address.
- You do not need to forward any ports for individual cameras if they are all connected to a recorder on site; just the NVR is needed.

7. Limit Features of Guest Accounts:

If your system is set up for multiple users, ensure that each user only has rights to features and functions they need to use to perform their job.

8. UPnP:

- UPnP will automatically try to forward ports in your router or modem. Normally this would be a good thing. However, if your system automatically forwards the ports and you leave the credentials defaulted, you may end up with unwanted visitors.
- If you manually forwarded the HTTP and TCP ports in your router/modem, this feature should be turned off regardless. Disabling UPnP is recommended when the function is not used in real applications.

9. SNMP:

Disable SNMP if you are not using it. If you are using SNMP, you should do so only temporarily, for tracing and testing purposes only.

10. Multicast:

Multicast is used to share video streams between two recorders. Currently there are no known issues involving Multicast, but if you are not using this feature, deactivation can enhance your network security.

11. Check the Log:

If you think that someone has gained unauthorized access to your system, you can check the system log. The system log will show you which IP addresses were used to login to your system and what was accessed.

12. Physically Lock Down the Device:

Ideally, you want to prevent any unauthorized physical access to your system. The best way to achieve this is to install the recorder in a lockbox, locking server rack, or in a room that is behind a lock and key.

13. Connect IP Cameras to the PoE Ports on the Back of an NVR:

Cameras connected to the PoE ports on the back of an NVR are isolated from the outside world and cannot be accessed directly.

14. Isolate NVR and IP Camera Network

The network your NVR and IP camera resides on should not be the same network as your public computer network. This will prevent any visitors or unwanted guests from getting access to the same network the security system needs in order to function properly.

Purpose

Welcome to use API of HTTP protocol specification.

Reader

- API software development engineers
- Project managers
- Product managers

Safety Instructions

The following categorized signal words with defined meaning might appear in the Manual.

Signal Words	Meaning
 DANGER	Indicates a high potential hazard which, if not avoided, will result in death or serious injury.
 WARNING	Indicates a medium or low potential hazard which, if not avoided, could result in slight or moderate injury.
 CAUTION	Indicates a potential risk which, if not avoided, could result in property damage, data loss, lower performance, or unpredictable result.
 TIPS	Provides methods to help you solve a problem or save you time.
 NOTE	Provides additional information as the emphasis and supplement to the text.

Revision History

No.	Version	Revision Content	Release Time
1	V2.40	Based on the old version.	August 8, 2018
2	V2.60	Add find media file with TrafficCar info.	September 12, 2018
3	V2.61	Add the "16 AI APIs" chapter.	October 24, 2018
4	V2.62	Add find media files with NonMotor info	November 06, 2018
5	V2.63	Add close door and access control custom password	November 27, 2018
6	V2.64	Add access control and video talk settings.	December 12, 2018
7	V2.65	Add intelligent caps and resource usage info	January 17, 2019

No.	Version	Revision Content	Release Time
8	V2.66	Add people heat map, realtime trace, history trace info	February 13, 2019
9	V2.67	Add subscribe heat map raw data	April 18, 2019
10	V2.68	Add get traffic snap event info record	May 15, 2019
11	V2.69	Add export traffic blacklist/redlist record	May 20, 2019
12	V2.70	Add privacy masking	May 29, 2019
13	V2.71	Add get the max and min temperature values	June 3, 2019
14	V2.72	Add smart motion detection config	June 27, 2019
15	V2.73	Add encrypted download media file	July 16, 2019
16	V2.74	Adjust Camera Image, Exposure, Backlight, White Balance, Day-Night, Zoom and Focus, Lighting, Video in Options config.	July 25, 2019
17	V2.75	Add adjust angle and depth field	August 13, 2019
18	V2.76	Add subscribe vehicles distribution data.	September 10, 2019
19	V2.77	Add Seek Find Logs	October 18, 2019
20	V2.78	Add face recognition event handler config	October 30, 2019
21	V2.79	Add RemoveEx Traffic BlackList/RedList record.	December 9, 2019
22	V2.80	Add Get GPS Status	December 12, 2019
23	V2.81	Add Get Last Event Info	February 28, 2020
24	V2.82	Add Radar Adaptor	March 23, 2020
25	V2.83	Add Radar Adaptor getCapsEx Delete Radar Adaptor getCaps	May 20, 2020
26	V2.84	Add AccessControl getLockStatus	June 12, 2020
27	V2.85	Add GlobalDeviceParam SupportGlobalDeviceParam	July 2, 2020
28	V2.86	Add ObjectPlacementDetection ObjectRemovalDetection	November 5,2020
29	V2.87	Add Radar Adaptor addRadarLinkSD, delRadarLinkSD Add VideoInAnalyse getTemplateRule	November 6,2020
30	V2.88	Add LensManager, resetAngle	November 26,2020
31	V2.89	Add config to Radar Adaptor	December 3, 2020
32	V2.90	Add Get Heat Map Info	January 7, 2021
33	V2.91	Add export Encryped Log	January 12, 2021
34	V2.92	Add controller related	February 23, 2021
35	V2.93	Add config to objectdetect DirectionStats	March 1,2021
36	V2.94	Add Camera to a Specific NVR Channel	March 11, 2021
37	V2.95	Add add and remove configure	March 26, 2021
38	V2.96	Add Add WorkSuitCompareServer	April 22,2021

No.	Version	Revision Content	Release Time
39	V2.97	Add VideoAnalyseRule head. Type add ObjectPlacement and ObjectRemoval	April 25,2021
40	V2.98	Modify WorkSuitCompareServer	May 11,2021
41	V2.99	Modify WorkSuitCompareServer doFind	May 11,2021
42	V3.00	Add trafficFlowStat stopFind, trafficFlowStat doFind and trafficFlowStat startFind	May 27,2021
43	V3.01	Add configManager setConfig IntelliSchemeTour.Enable	June 3,2021
44	V3.02	Add configManager getConfig IntelliSchemeTour	June 11,2021
45	V3.03	Add method cancel upgrade	June 21, 2021
46	V3.04	doFindHistoryByPic Add Time	Sept 29, 2021
47	V3.05	Add Cloud Upgrade, add find smd data	December 22, 2021
48	V3.06	Add OpenAI	January 11,2022
49	V3.07	Add correctScene api	January 21, 2022
50	V3.08	Add SSHD config	February 16, 2022
51	V3.09	Add Water Data Analyse	February 17, 2022
52	V3.10	Add Http Uploading api	April 20, 2022
53	V3.11	Add PassedNumber, EnteredDupNumber, ... to People Count Information.	June 1, 2022
54	V3.12	Add method export account file	June 29, 2022
55	V3.13	Add method query data flux	August 3, 2022
56	V3.14	Add method about SCADA	August 23, 2022
57	V3.15	Add method get gyroscope info	September 6, 2022
58	V3.16	Add Vehicle Manager	October 11, 2022
59	V3.17	Add method getCompleteMachineVersion	October 17, 2022
60	V3.18	Add Tcp Test	November 1, 2022
61	V3.19	Modify method WorkSuitCompareServer.doFind, getFlux, queryHistoryFlux and SCADA.doFind	December 7,2022
62	V3.20	Add method AudioAnalyseManager. setClassConfig, AudioAnalyseManager. getClassConfig, AudioAnalyseManager. setConfig and AudioAnalyseManager. getConfig.	December 8,2022
63	V3.21	Add method getViewRangeStatus	December 19, 2022

No.	Version	Revision Content	Release Time
64	V3.22	Add method Get Device Info, Add method Get Camera States Add method Get Camera Recording State	January 3, 2023
65	V3.23	Add FaceAnalysis Report Data Upload	January 5, 2023
66	V3.24	Add Play the Specified Audio File	February 6, 2023
67	V3.25	Modify State of Getting the Analysis Configurations of the Specified Audio Channel	February 21, 2023
68	V3.26	Add devVideoAnalyse.disableScene, devVideoAnalyse.enableScene, devVideoAnalyse.getSceneList	February 28, 2023
69	V3.27	Add getChannelAlgVersion	March 10, 2023
70	V3.28	Add Config VideoInDayNight、LightingScheme、 Lighting_V2、Lighting and RTMP_NVR	March 13, 2023
71	V3.29	Add SceneModeManager	March 30, 2023
72	V3.30	Add set parking space light plan	April 10, 2023
73	V3.31	Modify FishEye Setting, Add field	April 20, 2023
74	V3.32	Document Format Adjusted	May 24, 2023
75	V3.33	Add ExperienceRegistry	June 28, 2023
76	V3.34	Add Remote Delete Person	July 4, 2023
77	V3.35	Add findDownload	July 21, 2023

Privacy Protection Notice

As the device user or data controller, you might collect personal data of other such as face, fingerprints, car plate number, Email address, phone number, GPS and so on. You need to be in compliance with the local privacy protection laws and regulations to protect the legitimate rights and interests of other people by implementing measures include but not limited to: providing clear and visible identification to inform data subject the existence of surveillance area and providing related contact.

About the Manual

- The Manual is for reference only. If there is inconsistency between the Manual and the actual product, the actual product shall prevail.
- We are not liable for any loss caused by the operations that do not comply with the Manual.
- The Manual would be updated according to the latest laws and regulations of related regions. For detailed information, see the paper manual, CD-ROM, QR code or our official website. If there is inconsistency between paper manual and the electronic version, the electronic version shall prevail.
- All the designs and software are subject to change without prior written notice. The product updates might cause some differences between the actual product and the Manual. Please contact the customer service for the latest program and supplementary documentation.
- There still might be deviation in technical data, functions and operations description, or errors in print. If there is any doubt or dispute, please refer to our final explanation.
- Upgrade the reader software or try other mainstream reader software if the Manual (in PDF format) cannot be opened.
- All trademarks, registered trademarks and the company names in the Manual are the properties of their respective owners.
- Please visit our website, contact the supplier or customer service if there is any problem occurred when using the device.
- If there is any uncertainty or controversy, please refer to our final explanation.

Table of Contents

Cybersecurity Recommendations	I
Foreword	III
Table of Contents	VIII
1 Overview.....	24
2 References.....	25
3 Definitions	26
3.1 Abbreviations.....	26
3.2 Syntax Convention	26
3.3 HTTP API Protocol Format	26
3.3.1 key=value format	27
3.3.2 JSON format	28
3.4 Authentication.....	31
3.5 Conventions	32
3.5.1 Channels.....	32
4 General APIs.....	33
4.1 APIs of RTSP	33
4.1.1 Get Real-Time Stream.....	33
4.1.2 Get Playback Stream.....	35
4.1.3 Get File Stream.....	35
4.1.4 Get MJPG Stream	35
4.1.5 Create RTSP Pull and Authentication Token.....	36
4.2 Configure Manager	37
4.2.1 Get and Set Configure	37
4.2.2 Add and Remove Configure	38
4.2.3 Restore the Config.....	39
4.2.4 Restore except the Config.....	39
4.3 Audio	39
4.3.1 Get Audio Input Channel Numbers.....	39
4.3.2 Get Audio Output Channel Numbers.....	40
4.3.3 Post Audio Stream	40
4.3.4 Get Audio Stream	42
4.3.5 [Config] Volume Control.....	43
4.3.6 Getting the Analysis Configurations of the Specified Audio Channel	43
4.3.7 Setting the Analysis Configuration of the Specified Audio Channel.....	50
4.3.8 Getting the Related Business Configuration of the Specified Audio Channel	52
4.3.9 Configuring the Related Business Configuration of the Specified Audio Channel	58
4.3.10 AudioAnalyseRule Config	60
4.3.11 AudioAnalyseModule Config	60
4.3.12 Play the Specified Audio in the Device.....	60
4.4 Snapshot	61
4.4.1 [Config] Snap	61
4.4.2 Get a Snapshot.....	62

4.4.3 Subscribe to Snapshot	63
4.5 Video Attributes	65
4.5.1 Get Max Extra Stream Numbers	65
4.5.2 Get Encode Capability	66
4.5.3 Get Encode Config Capability	67
4.5.4 [Config] Encode of Media	71
4.5.5 [Config] Encode of Region Interested	76
4.5.6 [Config] Channel Title	77
4.5.7 Get Video Input Channels Device Supported	78
4.5.8 Get Video Output Channels Device Supported.....	78
4.5.9 Get Max Remote Input Channels	78
4.5.10 [Config] Video Standard	79
4.5.11 [Config] Video Widget	79
4.5.12 Get Video Input Capability.....	82
4.5.13 Get Coordinates of Current Window	87
4.5.14 Set Coordinates of Current Window.....	87
4.5.15 [Config] Video Out	88
4.5.16 [Config] Smart Encode	89
4.5.17 Get Decoder Caps	89
4.5.18 [Config] PrivacyMasking	90
4.5.19 Get Privacy Masking	91
4.5.20 Set Privacy Masking	93
4.5.21 Goto Privacy Masking.....	94
4.5.22 Delete Privacy Masking	94
4.5.23 Clear Privacy Masking.....	95
4.5.24 Get Privacy Masking Rect	95
4.5.25 [Config] Motion Detection Settings	96
4.5.26 [Config] LAEConfig	101
4.5.27 Enable/Disable all privacy masking covers	103
4.5.28 Get enable/disable state of all privacy masking covers.....	103
4.6 System	104
4.6.1 [Config] General.....	104
4.6.2 Get Current Time	105
4.6.3 Set Current Time.....	105
4.6.4 [Config] Locales	106
4.6.5 [Config] Holiday Management	108
4.6.6 Get Language Capability	109
4.6.7 [Config] Language	109
4.6.8 Get Device Type	109
4.6.9 Get Hardware Version	110
4.6.10 Get Serial Number of Device.....	110
4.6.11 Get Machine Name	111
4.6.12 Get System Information.....	111
4.6.13 Get Vendor Information	112
4.6.14 Get Software Information.....	112
4.6.15 Get Version of Onvif	113
4.6.16 Get Version of HTTP API.....	113

4.6.17 Get Device Class	114
4.6.18 [Config] Auto Maintain	114
4.6.19 Reboot	115
4.6.20 Shutdown	115
4.6.21 Factory Reset	116
4.6.22 Get Tracing Code of Device	116
4.6.23 Add Camera to Specified Channel	117
4.6.24 Delete Camera by Group.....	118
4.6.25 Acquiring All Available Resources	119
4.6.26 Subscribing for Device Online/Offline Status	125
4.6.27 Get Complete Machine Version.....	126
4.6.28 Connection test.....	127
4.6.29 Getting Online Status of the Channel.....	127
4.6.30 Getting the Recording Status of the Channel.....	129
4.7 User management	130
4.7.1 Get Information of a Particular User.....	130
4.7.2 Get Information of All Users.....	131
4.7.3 Get Information of All Active Users.....	132
4.7.4 Get Information of a Particular Group	132
4.7.5 Get Information of all Groups	133
4.7.6 Add a New User.....	134
4.7.7 Delete a User.....	134
4.7.8 Modify User Information	135
4.7.9 Modify User's Password	135
4.7.10 Manager Modify Common User's Password	135
4.7.11 [Config] User Login Authentication Policy.....	136
4.7.12 Export all the user info.....	错误!未定义书签。
4.8 Network	137
4.8.1 Get Network Interfaces.....	137
4.8.2 [Config] Client Access Filter.....	138
4.8.3 [Config] Network Basic Config.....	139
4.8.4 [Config] PPPoE.....	140
4.8.5 [Config] DDNS	142
4.8.6 [Config] Email	144
4.8.7 [Config] WLan	145
4.8.8 Scan Wlan Devices.....	147
4.8.9 [Config] UPnP	148
4.8.10 Get UPnP Status	149
4.8.11 [Config] NTP.....	150
4.8.12 [Config] RTSP	152
4.8.13 [Config] Alarm Server	153
4.8.14 [Config] Onvif Service Authorization.....	153
4.8.15 [Config] SSHD Config	154
4.8.16 [Config] Cellular Network Traffic Packages	154
4.8.17 Obtaining the Traffic Information of Current Month.....	155
4.8.18 Searching for History Traffic Statistics According to Specified Conditions	157
4.8.19 [Config]Configuring General Settings of RTMP	159

4.9 Event	161
4.9.1 [DataType] Event Handler	161
4.9.2 [Config] Alarm Event	167
4.9.3 [Config] Alarm Out	167
4.9.4 Get Alarm Input Channels	168
4.9.5 Get Alarm Output Channels.....	168
4.9.6 Get States of Alarm Input Channels	169
4.9.7 Get States of Alarm Output Channels	169
4.9.8 [Config] Video Blind Event.....	169
4.9.9 [Config] Video Loss Event	170
4.9.10 [Config] Login Failure Event	171
4.9.11 [Config] Storage Not Exist Event	171
4.9.12 [Config] Storage Access Failure Event.....	172
4.9.13 [Config] Storage Low Space Event	172
4.9.14 [Config] Net Abort Event	173
4.9.15 [Config] IP Conflict Event.....	173
4.9.16 Get Channels Event Happened.....	174
4.9.17 Subscribe to Event Message.....	174
4.9.18 Get Capability of Event Management.....	177
4.9.19 [Config] Net Alarm Event	180
4.9.20 Set Net Alarm State	181
4.9.21 Get Supported Events	182
4.9.22 [Config] Triger Http User List	183
4.10 Record.....	183
4.10.1 Get Capability of Recording.....	183
4.10.2 [Config] Record Config	184
4.10.3 [Config] Record Mode.....	185
4.10.4 [Config] Media Global	186
4.10.5 Find Media Files	187
4.10.6 Find media files with FaceDetection info.....	190
4.10.7 Find media files with FaceRecognition info.....	193
4.10.8 Find media files with HumanTrait info	197
4.10.9 Find media files with TrafficCar info	201
4.10.10 Find media files with IVS info	204
4.10.11 Find media files with NonMotor info	207
4.10.12 Searching for Media Files According to WorkClothesDetection	210
4.10.13 Download Media File with the File Name.....	215
4.10.14 Download Media File between Times.....	216
4.10.15 Encrypted Download Media File with the File Name	216
4.10.16 Query Total Number of Alarms.....	217
4.11 Log.....	218
4.11.1 Find Logs	218
4.11.2 Clear All the Logs	220
4.11.3 Backup Logs	220
4.11.4 Seek Find Logs	221
4.11.5 Export Encryped Log	222

4.11.6 [Config] Serial Port Log Redirection	222
4.12 Upgrader	223
4.12.1 Strat to Upgrade	223
4.12.2 Get Upgrade State	224
4.12.3 Set upgrader url	224
4.12.4 Cancel Upgrade	225
4.12.5 Checking Cloud Update Version	225
4.12.6 Performing Online Update	227
4.12.7 Canceling Online Update	228
4.13 Http Uploading	228
4.13.1 [Config] Active Image and Event Uploading	228
4.13.2 Active Image and Event Uploading	230
4.13.3 [Config] Active Event Uploading	231
4.13.4 Active Event Uploading	232
4.13.5 [Config] Active Report Data Uploading	233
4.13.6 People Counting Report Data Uploading	234
4.13.7 Video Structuring Report Data Upload	237
4.13.8 People Flow Heat Map Report Data Upload	238
4.13.9 ANPR Report Data Upload	239
4.13.10 Crowd Distribution Report Data upload	240
4.13.11 Vehicle Density Report Data Upload	241
4.13.12 FaceAnalysis Report Data Upload	242
5 Camera APIs	244
5.1 Image	244
5.1.1 [Config] Brightness, Contrast and Saturation	244
5.1.2 [Config] Sharpness	245
5.1.3 [Config] Flip, Mirror and Rotate90	245
5.2 Exposure	246
5.2.1 [Config] Exposure Config	246
5.3 Backlight	248
5.3.1 [Config] Backlight Config	248
5.4 White Balance	249
5.4.1 [Config] White Balance Config	249
5.5 Day-Night	250
5.5.1 [Config] Day-Night Config	250
5.6 Zoom and Focus	251
5.6.1 Adjust Focus	251
5.6.2 Adjust Focus Continuously	251
5.6.3 Auto Focus	252
5.6.4 Get Focus Status	252
5.6.5 [Config] Zoom Config	253
5.6.6 [Config] Focus Config	254
5.7 Lighting	255
5.7.1 [Config] Lighting Config	255
5.8 Video in Options	257
5.8.1 Change binocular camera's splice mode	257
5.8.2 [Config] Video in Options Config	257

6 Storage APIs	264
6.1 Storage Devices.....	264
6.1.1 Get Hard Disk Information.....	264
6.1.2 Get the Name of All Storage Devices.....	265
6.1.3 Get Storage Device Information	265
6.1.4 Get Storage Capability	266
6.1.5 Format Camera SD-Card	266
6.1.6 [Config] Hard Disk Recording Type	267
6.1.7 Getting Disk Information	267
6.2 NAS	275
6.2.1 [Config] NAS Information.....	275
6.3 Storage Point	276
6.3.1 [Config] Record Storage Point.....	276
6.3.2 [Config] Storage Group.....	278
6.4 SDEncrypt	279
6.4.1 Encrypt SD Card.....	279
6.4.2 Decrypt SD Card.....	279
6.4.3 Clear SD Card Password	280
6.4.4 Modify SD Card Password	280
6.4.5 Get SD Card Operate Error Policy	281
6.4.6 [Config] Storage Health Alarm Settings.....	281
7 Display APIs	283
7.1 GUI.....	283
7.1.1 [Config] GUISet.....	283
7.2 Split Screen.....	284
7.2.1 Split Screen Mode	284
7.3 Moniter Tour	285
7.3.1 [Config] Moniter Tour	285
7.3.2 Enable Tour.....	286
7.3.3 [Config] Monitor Collection	286
8 Comm APIs	288
8.1 PTZ.....	288
8.1.1 [Config] PTZ Config	288
8.1.2 Get PTZ Protocol List	289
8.1.3 Get PTZ Capability of Current Protocol	290
8.1.4 Get PTZ Status	291
8.1.5 PTZ Control	292
8.1.6 Preset.....	296
8.1.7 Tour	299
8.1.8 Scan.....	301
8.1.9 Pattern	303
8.1.10 Pan.....	304
8.1.11 [Config] PTZ Auto Movement.....	305
8.1.12 PTZ Restart	307
8.1.13 PTZ Reset.....	308
8.1.14 OSD Menu	308
8.1.15 [Config] Electronic PTZ.....	310

8.1.16 Get View Range Status	311
8.2 Wiper	313
8.2.1 Move Continuously	313
8.2.2 Stop Move.....	313
8.2.3 Move Once.....	313
8.3 Illuminator.....	314
8.3.1 [Config] Visible-light Illuminator	314
8.3.2 [Config]Configuring Lighting	314
8.3.3 [Config]Configuring Lighting V2.....	317
8.3.4 [Config]Configuring Light Schemes.....	321
8.4 Flashlight.....	322
8.4.1 [Config] Flashlight Config	322
8.5 Coaxial Control IO.....	323
8.5.1 Control White Light or Speaker	323
8.5.2 Get White Light and Speaker Status	324
8.6 Pir Alarm.....	324
8.6.1 [Config] Pir Parameter	324
8.7 SCADA.....	328
8.7.1 Searching for SCADA Attributes.....	328
8.7.2 Configuring SCADA Attributes	330
8.7.3 Obtaining Real-time Data of Monitoring Points	331
8.7.4 Configuring Monitoring Points	333
8.7.5 Starting Searching for Historical Data	334
8.7.6 Obtaining Historical Data.....	334
8.7.7 Stopping Searching for Historical Data	336
8.7.8 Obtaining IDs of External Devices Connected to the Host	336
8.8 Gyro.....	337
8.8.1 get gyroscope info	337
8.9 Other	338
8.9.1 [Config]Day/Night Settings of PTZ Module	338
9 Video Analyse APIs	342
9.1 Video Analyse Event	342
9.1.1 [Event] LeftDetection	342
9.1.2 [Event] TakenAwayDetection	343
9.1.3 [Event] WanderDetection.....	343
9.1.4 [Event] StayDetection	345
9.1.5 [Event] HumanTrait.....	346
9.1.6 [Event] CrossLineDetection	349
9.1.7 [Event] CrossRegionDetection	351
9.1.8 [Event] QueueStayDetection	352
9.1.9 [Event] QueueNumDetection.....	354
9.2 FaceRecognitionServer	355
9.2.1 Create Face Group.....	355
9.2.2 Modify Face Group	355
9.2.3 Delete Face Group	356
9.2.4 Deploy Face Group.....	356
9.2.5 Find Face Group.....	359

9.2.6 Re-Abstract Feature By Group	361
9.2.7 Add Person	361
9.2.8 Modify Person	362
9.2.9 Delete Person	363
9.2.10 Find Person	364
9.2.11 Re-Abstract Features By Person	366
9.2.12 [Config] Face Recognition AlarmOut Setting	367
9.2.13 Find Person by Picture	368
9.2.14 Find History Person by Picture	372
9.2.15 [Event] FaceDetection	376
9.2.16 [Event] FaceRecognition	377
9.2.17 [Event] FaceFeatureAbstract	379
9.2.18 [Config] Face Recognition Event Handler Setting	380
9.2.19 [Config] Face-ID Recognition Threshold	381
9.2.20 Export Face Database	382
9.2.21 Importing Face Database	382
9.3 People Counting	383
9.3.1 Get Summary	384
9.3.2 Query the Count of People	386
9.3.3 Clear the People Count Information	389
9.3.4 Subscribe the People Count Information	390
9.3.5 Clear statistics in time section	392
9.3.6 [Config] Video Widget Number Status	393
9.3.7 [Event] NumberStat	394
9.3.8 [Event] ManNumDetection	395
9.3.9 [Event] CrowdDetection	395
9.3.10 [Event] LeaveDetection	396
9.4 Heat Map	397
9.4.1 Get Heat Map Information	397
9.4.2 Get People Heat Map Information	397
9.4.3 Subscribe People Realtime Trace Information	399
9.4.4 Get People Histroy Trace Information	401
9.4.5 Subscribe Heat Map Raw Data	402
9.5 Crowd Distribute Map	403
9.5.1 Get Channel Caps	403
9.5.2 Subscribe to Realtime Crowd Stat	403
9.5.3 Get Current Crowd Stat	405
9.6 Video Analyse	407
9.6.1 Get Video Analyse Capability	407
9.6.2 [Config] Video Analyse Global	409
9.6.3 [Config] Video Analyse Rule	410
9.6.4 Get Last Event Info	414
9.6.5 [Config] GlobalDeviceParam	415
9.6.6 Get Template Rule	415
9.6.7 [Config] IntelliSchemeTourEnableSetting	416
9.6.8 [Config] Intelligent Tour Plan	417
9.6.9 Export Intelligent Diagnosis, Allowlist, and Blocklist Information	421

9.6.10 Import Intelligent O&M, Allowlist, and Blocklist Information	422
9.6.11 Get Intelligent Capability.....	423
9.6.12 Subscribe Resource Usage Info.....	425
9.6.13 Export Encrypted Files	426
9.6.14 Platform intelligent control	427
9.6.18 Get Algorithm Version Of Channel	429
9.7 WorkSuitCompareServer	430
9.7.1 Add Compliance Library	430
9.7.2 Delete Compliance Library	431
9.7.3 Find Compliance Library.....	432
9.7.4 Get Compliance Library Arming Information of Channels.....	433
9.7.5 Modify Compliance Group Information.....	434
9.7.6 Deploy Compliance Library	435
9.7.7 Find Workwear Information in Compliance Library	436
9.7.8 Get Find Workwear Information Result	436
9.7.9 Stop Find Workwear Information	439
9.7.10 Delete Workwear Information	439
9.7.11 Re-extracting Features by Workwear	440
9.7.12 Stop Re-extracting Workwear Features	440
9.8 Smart Motion Detection	441
9.8.1 [Config] SmartMotionDetect	441
9.8.2 Start SMD Data Search	442
9.8.3 Get SMD Data Search Result.....	443
9.8.4 End SMD Data Search	443
9.9 Intelligent analysis tasks	444
9.9.1 Add task	444
9.9.2 Attach task result	455
9.9.3 Remove task.....	458
9.9.4 Push Picture File.....	459
9.10 SceneModeManager.....	460
9.10.1 Get Scene Mode Capabilities	460
9.10.2 Get The Current Scene Mode	460
9.10.3 Get The Default Scene Mode	461
9.10.4 Set The Scene Mode	462
10 Intelligent Traffic APIs.....	496
10.1 Intelligent Traffic Event.....	496
10.1.1 [Event] TrafficJunction	496
10.1.2 [Event] TrafficRetrograde.....	498
10.1.3 [Event] TrafficJam	498
10.1.4 [Event] TrafficUnderSpeed	499
10.1.5 [Event] TrafficOverSpeed	500
10.1.6 [Event] TrafficPedestrain	501
10.1.7 [Event] TrafficParking.....	502
10.2 Traffic Flow	504
10.2.1 [Event] TrafficFlowStat.....	504
10.2.2 Find Traffic Flow History	504
10.2.3 Start Traffic Statistics Search	506

10.2.4 Get Traffic Statistics	507
10.2.5 End Traffic Statistics Search.....	508
10.3 Traffic Record.....	509
10.3.1 Insert Traffic BlockList/AllowList Record	509
10.3.2 Update Traffic BlockList/AllowList Record.....	509
10.3.3 Remove Traffic BlockList/AllowList Record	510
10.3.4 Find Traffic BlockList/AllowList Record	511
10.3.5 RemoveEx Traffic BlockList/AllowList Record.....	512
10.3.6 Import Traffic BlockList/AllowList.....	512
10.3.7 Export Traffic BlockList/AllowList.....	513
10.3.8 Export Traffic Flow	514
10.3.9 Export Traffic Snap Event Info.....	516
10.4 Traffic Snap Operation	517
10.4.1 Open Strobe	517
10.4.2 Open/Close Unlicensed Vehicle Detection.....	518
10.4.3 Manual Snap.....	519
10.5 Traffic Parking	519
10.5.1 Get the Specific Parking Space Status	519
10.5.2 Get All Status of Parking Spaces	520
10.5.3 [Config] Parking Space Light State	521
10.5.4 Set Order State.....	524
10.5.5 Set Light State	525
10.5.6 [Config] Parking Space Access Filter Setting.....	525
10.5.7 Set OverLine State	526
10.5.8 Set Parking Control Info	526
10.5.9 Set Parking Space Lighting Plan	529
10.6 Vehicles Distribution.....	530
10.6.1 Subscribe Vehicles Distribution Data	530
10.7 Vehicle Manager	531
10.7.1 Adding Vehicle Groups	531
10.7.2 Modifying Vehicle Groups.....	532
10.7.3 Deleting Vehicle Groups	533
10.7.4 Searching for Vehicle Groups.....	533
10.7.5 Adding Vehicle Records.....	534
10.7.6 Modifying Vehicle Information.....	537
10.7.7 Deleting Vehicle Information.....	539
10.7.8 Searching for Vehicles from Registered Database	540
10.7.9 Obtaining Vehicle Search Results	542
10.7.10 Stopping Searching for Vehicles	543
11 Thermography and Radiometry APIs	545
11.1 Thermography Manager.....	545
11.1.1 Get Capability of Thermography.....	545
11.1.2 [Config] Thermography Options.....	548
11.1.3 Get ExternSystem Information.....	550

11.1.4 Get Information of Preset Mode.....	551
11.1.5 Get Optimized Region Information	552
11.1.6 Enable Shutter	552
11.1.7 Fix Focus.....	553
11.1.8 Do Flat Field Correction.....	553
11.2 Radiometry	553
11.2.1 Get Capability of Radiometry.....	553
11.2.2 [Config] Heat Image Thermometry	555
11.2.3 [Config] Thermometry Rule.....	557
11.2.4 [Config] Heat Image Temper Event.....	559
11.2.5 Get Temperature of Particular Point	560
11.2.6 Get Temperature of Particular Condition	560
11.2.7 Find Temperature Information.....	561
11.2.8 Subscribe Temperature Information.....	563
11.2.9 Subscribe Radiometry Data.....	565
11.2.10 Fetch Radiometry Data	566
11.2.11 [Config] FireWarning Config.....	567
11.2.12 [Config] FireWarningMode Config	568
11.2.13 Get Current Hot Cold Spot.....	569
11.2.14 [Config] Heat Image Temper PreAlarm Event	570
11.2.15 Get Heat Map Info.....	570
11.2.16 Get Temperature of Particular Region.....	572
11.3 TemperCustom	573
11.3.1 Set Environment Temperature	573
12 Access Control APIs	574
12.1 Access Control	574
12.1.1 Open Door	574
12.1.2 Close Door.....	574
12.1.3 Get Door Status.....	575
12.1.4 Get Lock Status	575
12.1.5 Capture Fingerprint.....	576
12.1.6 Capture Face Picture.....	577
12.1.7 Query AccessControl Record	578
12.1.8 Query Access Control Alarm Record.....	582
12.1.9 [Event] AccessControl	583
12.1.10 [Event] CitizenPictureCompare	586
12.1.11 [Event] Door Status Event	588
12.1.12 [Config] Access Control General Setting	589
12.1.13 [Config] Access Control Setting.....	590
12.1.14 [Config] Wiegand Setting.....	594
12.1.15 [Config] Access Time Schedule Setting	595
12.1.16 [Config] Special Day Group Setting.....	596
12.1.17 [Config] Special Days Schedule Setting.....	597
12.1.18 [Config] MeasureTemperature Setting	598
12.1.19 [Config] CitizenPictureCompare Setting.....	601
12.2 Access Control Manager.....	602
12.2.1 Get Access Control Capability	602

12.2.2 Add SubController.....	603
12.2.3 Modify SubController	604
12.2.4 Remove SubController	604
12.2.5 Get SubController Info.....	605
12.2.6 Get SubController States.....	606
12.2.7 Set RepeatEnter Route	606
12.2.8 Get RepeatEnter Route	607
12.2.9 Set ABlock Route.....	607
12.2.10 Get ABlock Route	608
12.2.11 Get Log Status	608
12.2.12 Sync Offline Log	609
12.2.13 Sync SubController Time.....	609
12.3 Access User Account (V1)	610
12.3.1 Add Access User Face	610
12.3.2 Modify Access User Face	611
12.3.3 Delete Access User Face	612
12.3.4 Find Access User Face.....	612
12.3.5 Add Access User Card and Fingerprint.....	613
12.3.6 Modify Access User Card and Fingerprint.....	616
12.3.7 Delete Access User Card and Fingerprint.....	619
12.3.8 Find Access User Card and Fingerprint	620
12.3.9 Get the Total Number of Records of Access User Card and Fingerprint.....	628
12.4 Access User Account (V2)	628
12.4.1 Add Access User.....	628
12.4.2 Modify Access User	630
12.4.3 Delete All Access User	631
12.4.4 Delete Multiple Access Users.....	631
12.4.5 Find Multiple Access Users	632
12.4.6 Start Find Access User Related Information	633
12.4.7 Get Find Result of Access User Related Information.....	635
12.4.8 Stop Find Access User Related Information	636
12.4.9 Add Multiple Access Cards	637
12.4.10 Modify Multiple Access Cards	637
12.4.11 Delete All Access Cards.....	638
12.4.12 Delete Multiple Access Cards.....	639
12.4.13 Find Multiple Access Cards	639
12.4.14 Start Find Access User Card Related Information	640
12.4.15 Get Find Result of Access User Card Related Information.....	641
12.4.16 Stop Find Access User Card Related Information	642
12.4.17 Add Multiple Access User Fingerprint.....	643
12.4.18 Modify Access User Fingerprint.....	644
12.4.19 Delete All Access User Fingerprint	645
12.4.20 Delete Multiple Access User Fingerprint	645
12.4.21 Find Access User Fingerprint	645
12.4.22 Add Multiple Access User Face	646
12.4.23 Update Multiple Access User Face	647
12.4.24 Delete All Access User Face	648

12.4.25 Delete Multiple Access User Face.....	648
12.4.26 Find Access User Face.....	648
12.4.27 Start Find Access User Face Related Information	649
12.4.28 Get Find Result of Access User Face Related Information	650
12.4.29 Stop Find Access User Face Related Information	651
12.4.30 Access Control Protocol Capability Query	651
12.4.31 Inserting Bluetooth Card Information.....	656
12.4.32 Searching for Information on Bluetooth Card.....	657
12.4.33 Deleting All Bluetooth Card Information	657
12.4.34 Deleting Bluetooth Card Information	658
12.4.35 Searching for Information on Access Control Bluetooth Card	658
12.4.36 Acquiring the Search Results of Bluetooth Card.....	659
12.4.37 Ending the Bluetooth Card Search.....	659
12.5 Admin Password	660
12.5.1 Add Access Control Admin Password	660
12.5.2 Modify Access Control Admin Password.....	661
12.5.3 Delete Access Control Admin Password	662
12.5.4 Find Access Control Admin Password.....	663
12.5.5 Get the Total Number of Records of Access Control Admin Password	665
13 Intelligent Building APIs	667
13.1 Video Talk.....	667
13.1.1 Subscribe Video Talk Status.....	667
13.1.2 Unsubscribe Video Talk Status.....	669
13.1.3 Invite Server on Video Talk.....	669
13.1.4 Cancel the Video Talk	671
13.1.5 Answer the Invitation	671
13.1.6 Refuse to Answer the Video Talk Invitation	672
13.1.7 Hang Up.....	673
13.2 Intelligent Building Record	673
13.2.1 Query Video Talk Log Record.....	673
13.2.2 Insert Announcement Record	674
13.2.3 Query Alarm Record	675
13.3 SIP.....	676
13.3.1 [Config] SIP Configuration	676
13.3.2 [Config] Registrar Configuration	679
13.4 Room Number Database Management.....	680
13.4.1 Adding Room Number	680
13.4.2 Getting Records by Video Talk short number.....	682
13.4.3 Getting Records by recno.....	684
13.4.4 Updating Room Number Records	686
13.4.5 Deleting Records by recno	687
13.4.6 Clearing All Room Numbers	687
13.4.7 Getting Total Quantity of Room Number	688
13.5 ElevatorFloorCounter	688
13.5.1 Set Elevator Floor Info	688
13.5.2 Get Elevator WorkInfo	688
13.5.3 Get Capability	689

14 DVR Custom APIs.....	690
14.1 File Finder	690
14.1.1 Create a File Finder	690
14.1.2 Create a Motion File Finder	691
14.1.3 Get the File Information Found by the Finder	692
14.1.4 Stop the Finder	693
14.1.5 Get Bound Files	694
14.2 BandLimit	695
14.2.1 Get Bandwidth Limit State	695
14.3 Record Files Protection.....	696
14.3.1 Add Protection	696
14.3.2 Cancel Protection	696
14.3.3 Remove Protection	697
14.3.4 DownloadFile	697
14.3.5 UploadFile.....	698
14.3.6 List all elements in the specified directory.....	698
14.4 Daylight	702
14.4.1 Get Daylight.....	702
15 Other APIs	703
15.1 Discover Devices	703
15.1.1 Discover Devices on Internet.....	703
15.2 Open Platform.....	704
15.2.1 Application Start and Stop	704
15.2.2 Install Application	705
15.2.3 Update Application and License	706
15.2.4 Uninstall Application	707
15.2.5 Download Application Log	707
15.3 GPS.....	708
15.3.1 Get Capablity	708
15.3.2 [Config] GPS config	708
15.3.3 Get GPS Status	709
15.4 Lens Function.....	710
15.4.1 Get Lens Capability	710
15.4.2 Adjust Angle Continuously	711
15.4.3 Stop Adjusting Angle.....	711
15.4.4 Adjust Depth Field.....	711
15.4.5 Adjust Depth Field Continuously	712
15.4.6 Get Depth Field Status	712
15.4.7 Auto Adjust Depth Field	713
15.4.8 Scene Correction	713
15.4.9 Reset Angle	714
15.5 FishEye	715
15.5.1 Get FishEye Capability	715
15.5.2 [Config] FishEye Setting	716
15.6 Radar Adaptor	718
15.6.1 Get Radar Capability	718
15.6.2 Get Radar Capbility (Enhanced)	719

15.6.3 Get Status	720
15.6.4 Calculate Real Size	720
15.6.5 Subscribe Alarm Point Info	721
15.6.6 Manual Locate	724
15.6.7 Start Radar Calibration	724
15.6.8 Add Radar Link SD	725
15.6.9 Del Radar Link SD	726
15.6.10 Get Link SD State	726
15.6.11 [Config] MapPara	727
15.6.12 [Config] RadarAnalyseRule	728
15.6.13 [Config] RadarCalibration	730
15.6.14 [Config] RadarGuardLine	731
15.6.15 [Config] RadarLink	732
15.6.16 [Config] RadarLinkDevice	732
15.6.17 [Config] RadarPara	733
15.6.18 [Config] RadarTrackGlobal	734
15.6.19 [Config] RemoteSDLLink	735
15.7 Water Radar	735
15.7.1 Acquire Radar Capability	735
15.7.2 Get Radar Detection Target Data	736
15.8 Water Quality Detection	736
15.8.1 Get Capability	736
15.8.2 Get Real-time Detection Data	737
15.8.3 Start Find Water Quality Report	739
15.8.4 Get Find Water Quality Report Search Results	740
15.8.5 Stop Find Water Quality Report	741
15.9 Advertisement	742
15.9.1 Changing Stay Time of Advertisement Image	错误!未定义书签。
15.9.2 Acquiring Video and Image Files Played by the Device	错误!未定义书签。
15.9.3 Modifying the Display Duration of Image Advertisements	742
15.9.4 Releasing Advertisements	742
15.9.5 Getting the List of Advertisement Resources	745
15.9.6 Searching for the Released Advertisements	746
15.9.7 Deleting Advertisement Resources	749

1

Overview

This document specifies the HTTP-based application programming interface of video products.

The HTTP-based interface provides the functionality for requesting snapshot and media stream, controlling camera functions (for example, PTZ and focus), and getting and setting internal parameter values.

The video products serve as a server. The client sends requests to server, and then server handles requests and returns resources accordingly.

HTTP API V3.35 - Intelbras

2

References

- [1] RFC 2326 Real Time Streaming Protocol (RTSP)
- [2] RFC 2616 Hypertext Transfer Protocol-HTTP/1.1
- [3] RFC 3986 Uniform Resource Identifiers (URI) Generic Syntax
- [4] RFC 7616 HTTP Digest Access Authentication
- [5] RFC 8259 The JavaScript Object Notation (JSON) Data Interchange Format

HTTP API V3.35 - Intelbras

3

Definitions

3.1 Abbreviations

The following abbreviations are used throughout this document.

API	Application Programming Interface. In the document, it especially presents application programming interface of video products.
-----	---

3.2 Syntax Convention

- In URL syntax and in descriptions of API parameters, text in italic within angle brackets denotes content that should be replaced with either a value or a string. When replacing the text string, the angle brackets must also be replaced. For example, <server> in URL syntax is replaced with the IP number of server, e.g., 192.168.1.108.
- String shown in bold face denotes a brief explanatory note of the string close to it.
- Name-value pair in square brackets denotes content that is optional. For example, "http ://<server>/cgi-bin/snapshot.cgi[?channel=1]" can be like this "http ://<server>/cgi-bin/snapshot.cgi".
- The API syntax must follow the standard of URI. (RFC 3986: Uniform Resource Identifiers (URI) Generic Syntax); that is, spaces and other reserved characters (e.g, ":" , "/" , "?" , "@" , ";" , "=" , "+" , "," , "\$" , "&") within a name-value pair should be replaced with %< ASCII hex>. For example, the blank should be replaced with %20.
- To describe the range of a variable, we use some symbols such as "[]" and "{ }". For example, "[0-100]" denotes an integer not less than 0 and not larger than 100. "{0, 1, 2, 3}" denotes the valid value of an integer among 0, 1, 2 and 3.
- "[]" following a string denotes an array. The index is an integer and starts from 0. For example, "Snap[channel]" may be "Snap[0]" "Snap[0]" etc.
- The variable may be different types: string, integer, bool or float. Integer is 32 bits. The range of bool is "true" and "false".
- "R/O" in parameters means this parameter is required or not, "R"means required, "O"means optional.

3.3 HTTP API Protocol Format

HTTP APIs support HTTP and HTTPS protocol. It is recommend using HTTPS to improve network security.

The request and response parameters of the HTTP APIs have two formats: key=value format and JSON format. Each HTTP API will choose one of these formats, please refer to the API detail in later chapters.

3.3.1 key=value format

If the HTTP API use key=value format for it's parameters, the URL syntax is :

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

protocol: URL scheme for the particular request. The http protocol and https protocol are both supported in this specification. So "http", as most of the APIs' default protocol except several RTSP APIs, can be replaced by "https".

server: Server could be "**hostname[:port]**". The **hostname** can be IP address or the fully qualified domain name of an IP device. The **port** is the port number of **server** listening for TCP connections. If the port is not given, the default port is assumed. For HTTP, the default port is 80. For HTTPS, the default port is 443.

abs_path: The Request-URI for the resources is abs_path. The abs_path in this specification is most often of the form "/cgi-bin/*.cgi".

query: The query field is a string of information to be interpreted by the resource. It consists of resource-related parameters. And it must be listed in name-value pair syntax (p1=v1&p2=v2&...&pn=vn).

For the HTTP API that use key=value format for it's parameters, it usually has no HTTP Body.

Request Example: HTTP request with key=value format parameters

```
GET http://192.168.1.108/cgi-bin/devVideoInput.cgi?action=getFocusStatus&channel=1 HTTP/1.1
User-Agent: client/1.0
Content-Length: 0
```

The server uses the standard HTTP status codes, the syntax of the status line of the response is:

```
HTTP/1.1 <HTTP code> <HTTP text>
```

With the following HTTP code and meanings:

HTTP code	HTTP text	Description
200	OK	The request has succeeded. The requested resource will be returned in the HTTP text.
400	Bad Request	The request had bad syntax or was inherently impossible to be satisfied.
401	Unauthorized	The request requires user authentication or the authorization has been refused.
403	Forbidden	The user does not have the right to access the service.
404	Not Found	The server has not found anything matching the request.
500	Internal Server Error	The server encountered an unexpected condition that prevented it from fulfilling the request.
501	Not Implemented	The server has not implemented the service.

If the HTTP code is 200, means the API execute success, and the response data in HTTP Body (maybe multipart) can be a multiline key=value data, or just a line with a word "OK".

Reponse Example: success response with multiline key=value

```
HTTP/1.1 200 OK
Server: Device/1.0
Content-Type: text/plain
Content-Length: <length>
```

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

Reponse Example: success response with a word "OK"

```
HTTP/1.1 200 OK
Server: Device/1.0
Content-Type: text/plain
Content-Length: <length>
```

```
OK
```

If the HTTP code is not 200, means the API execute failed, and the response data in HTTP Body maybe empty, or just two line, first line is a word "Error" to indicate error happened, the second line contain error detail.

Reponse Example: request does not fit with syntax.

```
HTTP/1.1 404 Not Found
Server: Device/1.0
```

Reponse Example: Request spells wrong.

```
HTTP/1.1 400 Bad Request
Server: Device/1.0
Content-Type: text/plain
Content-Length: <length>
```

```
Error
Bad Request!
```

Reponse 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: Device/1.0
Content-Type: text/plain
Content-Length: <length>
```

```
Error
Internal Server Error!
```

3.3.2 JSON format

If the HTTP API use JSON format for it's parameters, the URL syntax is :

```
<protocol>://<server><abs_path>
```

protocol: URL scheme for the particular request. The http and https protocols are both supported in this specification. So "http", as most of the APIs' default protocol except several RTSP APIs, can be replaced by "https".

server: Server could be "**hostname[:port]**". The **hostname** can be IP address or the fully qualified domain name of an IP device. The **port** is the port number of **server** listening for TCP connections. If the port is not given, the default port is assumed. For HTTP, the default port is 80. For HTTPS, the default port is 443.

abs_path: The Request-URI for the resources is abs_path. The abs_path in this specification is most often prefix with "/cgi-bin/api/" , and followed with resource name, for example : "/cgi-bin/api/CloudUpgrader/check" .

For the HTTP API that use JSON format for it's parameters, the parameters are put to the HTTP Body as a JSON object. If the API has no request parameter, then the HTTP Body is an empty JSON object, for example " {} ", or there is no HTTP Body.

Request Example: HTTP request with JSON format parameters

```
GET http://192.168.1.108/cgi-bin/api/CloudUpgrader/check HTTP/1.1
User-Agent: client/1.0
Content-Type: application/json
Content-Length: xxx

{
    "way" : 0,
    "proxy" : {
        "IP" : "10.1.2.3",
        "Port" : 8080
    }
}
```

The server uses the standard HTTP status codes, the syntax of the status line of the response is:

```
HTTP/1.1 <HTTP code> <HTTP text>
```

With the following HTTP code and meanings:

HTTP code	HTTP text	Description
200	OK	The request has succeeded. The requested resource will be returned in the HTTP text.
400	Bad Request	The request had bad syntax or was inherently impossible to be satisfied.
401	Unauthorized	The request requires user authentication or the authorization has been refused.
403	Forbidden	The user does not have the right to access the service.
404	Not Found	The server has not found anything matching the request.
500	Internal Server Error	The server encountered an unexpected condition that prevented it from fulfilling the request.

HTTP code	HTTP text	Description
501	Not Implemented	The server has not implemented the service.

If the HTTP code is 200, means the API execute success, and the response data in HTTP Body (maybe multipart) can be a JSON object. If the API has no response parameter, then the HTTP Body is an empty JSON object, for example " {} ", or there is no HTTP Body.

Reponse Example: success respose with JSON format Body

```
HTTP/1.1 200 OK
Server: Device/1.0
Content-Type: application/json
Content-Length: <length>

{
  "info" : {
    "State" : "None",
    "PackageType" : "all",
    "OldVersion" : "0000",
    "NewVersion" : "0004",
    "Attention" : "What is new",
    "PackageUrl" : "https://example.com/1.zip",
    "Packageld" : "1d2ee7",
    "CheckSum" : "F3D288AB",
    "BuildTime" : "08-10-2018 01:01:02"
  }
}
```

If the HTTP code is not 200, means the API execute failed, and the response data in HTTP Body maybe empty, or a JSON object to describe the error detail. The value of "ErrorCode" contains error code, and the value of "ErrorMsg" contains error detail.

Reponse Example: request does not fit with syntax.

```
HTTP/1.1 404 Not Found
Server: Device/1.0
```

Reponse Example: Request spells wrong.

```
HTTP/1.1 400 Bad Request
Server: Device/1.0
Content-Type: application/json
Content-Length: <length>

{
  "ErrorCode" : 10086,
  "ErrorMsg" : "Bad Request!"
}
```

Reponse 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: Device/1.0
Content-Type: application/json
Content-Length: <length>

{
    "ErrorCode" : 10097
    "ErrorMsg" : "Internal Server Error!"
}

```

3.4 Authentication

Video products support digest authentication, see RFC 7616 for detail. If the http request sent by client does not provide valid "Authorization" header information, video products would return HTTP status code 401 and some information for authentication, then client should calculate authentication information according to RFC 7616, and sent request again with authentication information using "Authorization" header. Video products return the required resource only if authorization information correct.

- When the client does not providing digest authentication or the client calculates digest authentication with expired "nonce" data, video product reply 401 with information for authorization.

Request Example: The client send request without authentication infomation

```

GET /cgi-bin/magicBox.cgi?action=getLanguageCaps HTTP/1.1
User-Agent: client/1.0
Content-Length: 0

```

Reponse Example: The device send 401 reponse with parameters for authentication calculation.

```

HTTP/1.1 401 Unauthorized
Server: Device/1.0
WWW-Authenticate: Digest realm="Device_00408CA5EA04",
    nonce="000562fd20ef95ad", qop="auth",
    opaque="5ccc069c403ebaf9f0171e9517f40e41"

```

- Then the client generate "nc" and "cnonce", calculates the digest authorization using "username" "password" "HTTP method" "URI" "nc" "cnonce" "realm" "nonce" , according to RFC 7616, and then sends it to video product again. If the digest authorization is correct, the video product will reply HTTP status code 200 with the response data. If digest authorization is not correct, the video product will reply HTTP status code 403 .

Request Example: The client send request again with authentication information (suppose the password of admin user is "abcd1234")

```
GET /cgi-bin/magicBox.cgi?action=getLanguageCaps HTTP/1.1
User-Agent: client/1.0
Authorization: Digest username="admin", realm="Device_00408CA5EA04",
nonce="000562fd20ef95ad", nc=00000001, cnonce="0a4f113b", qop="auth",
uri="/cgi-bin/magicBox.cgi?action=getLanguageCaps",
response="dfd0f24bed4c336d20c8f0729dd5dbc8"
opaque="5ccc069c403ebaf9f0171e9517f40e41"
Content-Length: 0
```

Reponse Example: If the authentication information is correct, the device send 200 reponse.

```
HTTP/1.1 200 OK
Server: Device/1.0
Content-Type: text/plain
Content-Length: <length>

Languages=SimpChinese,English,French
```

Reponse Example: If the authentication information is not correct, the device send 403 reponse.

```
HTTP/1.1 403 Forbidden
Server: Device/1.0
Content-Length: 0
```

3.5 Conventions

3.5.1 Channels

For requests containing the param 'channel' like :

`http://<server>/cgi-bin/mjpg/video.cgi[?channel=<ChannelNo>]`

ChannelNo: integer, must starts from 1. Default is 1 if not present.

But for responses, the '**ChannelNo**' should start from 0. In other words, the request channel 1 equals response channel 0.

The APIs specified in this section are supported by all video products.

4.1 APIs of RTSP

4.1.1 Get Real-Time Stream

Get real-time media stream APIs use RTSP protocol, please refer to RFC 2326 for detail. The rtsp service default port is 554. The IP Camera supports both TCP and UDP transmission forms. It supplies digest authentication ways. The authentication process is similar with "3.5 Authentication".

- The format of the RTSP URL parameter for getting 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)			
Name	Type	R/O	Description
channel	int	R	Video channel number, starting from 1.
subtype	int	R	<p>The stream type, including main stream and sub stream.</p> <p>Execute the "4.5.1 Getting Maximum Number of Sub Streams" command to get the number of sub streams.</p> <p>Optional values:</p> <ul style="list-style-type: none"> 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 media 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
```

4.1.2 Get Playback Stream

It's similar with "4.1.1 Get Real-Time Stream", except that there are parameters "starttime" and "endtime".

URL	rtsp://<server>:[port]/cam/playback			
Method	DESCRIBE, SETUP, PLAY, PAUSE, TEARDOWN, ...			
URL Params (key=value format in URL)				
Name	Type	R/O	Description	Example
channel	int	R	The video channel index which starts from 1	1
starttime	char[32]	R	The playback begin time	2012_09_15_12_37_05
endtime	char[32]	O	The playback end time	2012_09_15_18_34_14
URL Example				
rtsp://192.168.1.108:554/cam/playback?channel=1&starttime=2012_09_15_12_37_05&endtime=2012_09_15_18_34_14				

4.1.3 Get File Stream

It's similar with "4.1.1 Get Real-Time Stream". The filename with absolute path of file stream to get is place in rtsp url.

URL	rtsp://<server>:[port]/<path to filename>			
Method	DESCRIBE, SETUP, PLAY, PAUSE, TEARDOWN, ...			
URL Params (key=value format in URL)				
Name	Type	R/O	Description	Example
URL Example				
rtsp://192.168.1.108:554//mnt/sd/2015-09-16/001/dav/20/20.32.08-20.32.28[M][0@0][0].dav				

4.1.4 Get MJPG Stream

Get a video stream encoded by mjpg (motion jpeg).

Request URL	http://<server>/cgi-bin/mjpg/video.cgi
--------------------	--

Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
channel	int	R	The video channel index which starts from 1
subtype	int	R	The stream type, It can be the following value: 0-Main Stream 1-Extra Stream 1 2-Extra Stream 2
Request Example			
http://192.168.1.108/cgi-bin/mjpg/video.cgi?channel=1&subtype=0			

Response Params (multipart, binary in body)
<binary data> : JPEG image data
Response Example
HTTP/1.1 200 OK
Server: Device/1.0
Content-Type: multipart/x-mixed-replace; boundary=<boundary>
--<boundary>
Content-Type: image/jpeg
Content-Length:<image size>
<JPEG image data>
--<boundary>
Content-Type: image/jpeg
Content-Length:<image size>
<JPEG image data>
--<boundary>
.....

4.1.5 Create RTSP Pull and Authentication Token

Request URL	http://<server>/cgi-bin/api TokenNameManager/createToken		
Method	POST		
Request Params (JSON format in body)			
Name	Type	R/O	Description
Channel	uint32	R	video channel number which starts from 0
Request Example			
{			
"Channel": 0			
}			

Response Params (key=value format in body)
Name
Type
R/O
Description
Token
char[128]
R
Token
"AwDSEci0j4EUW6pU

				z5bcQ0yK8_Rbq0vaU sUwqQKmFoUpuMJjO xy9kgcV6BICty8U"
--	--	--	--	---

Response Example

```
{
    "Token": "AwDSEci0j4EUW6pUz5bcQ0yK8_Rbq0vaUsUwqQKmFoUpuMJjOxy9kgcV6BICty8U"
}
```

4.2 Configure Manager

4.2.1 Get and Set Configure

- Get Configure

Get configure detail by name. Each configure name has it's specific configure detail. Please refer to configure detail APIs in later [Config] chapters.

Request URL	http://<server>/cgi-bin/configManager.cgi?action=getConfig		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
name	string	R	The name of the configure
Request Example			
http://<server>/cgi-bin/configManager.cgi?action=getConfig&name=General			

Response Params (key=value format in body)			
Name	Type	R/O	Description
table	object	R	The configure table object.
+<config name>	string/object/object[]	R	The configure detail of the requested configure name. Each configure name has it's specific configure detail. Please refer to configure detail apis in later chapters.

Response Example

```
table.General.MachineID=20832748927
table.General.MachineName=DVR001
table.General.MachineAddress=XXX Road
...
```

- Set Configure

Set configure detail by name. Each configure name has it's specific configure detail. Please refer to configure detail APIs in later [Config] chapters.

Request URL	http://<server>/cgi-bin/configManager.cgi?action=setConfig		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description

<config name>	string/object/object[]	R	The configure detail of the configure name. Each configure name has it's specific configure detail. Please refer to configure detail apis in later chapters.	
---------------	------------------------	---	--	--

Request Example

http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&General.MachineID=20832748927&General.MachineName=DVR001&General.MachineAddress=XXX%20Road

Response Params (OK in body)

Response Example

OK

4.2.2 Add and Remove Configure

- Add configure

The configuration name is put in the URL in the form of key-value; the specific configuration field can be obtained through getConfig. Each configure name has it's specific configure detail. Please refer to configure detail APIs in later [Config] chapters.

Request URL	http://<server>/cgi-bin/configManager.cgi?action=addConfig			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Sample
<config name>	string/object/object[]	R	The configure detail of the configure name. Each configure name has it's specific configure detail. Please refer to configure detail apis in later chapters.	

Request Example

http://192.168.1.108/cgi-bin/configManager.cgi?action=addConfig&General.MachineAddress=XXX%20Road

Response Params (OK in body)

Response Example

OK

- Remove configure

Specific configuration fields can be obtained through getConfig. Only configuration fields can be deleted, not configuration items. Each configure name has it's specific configure detail. Please refer to configure detail APIs in later [Config] chapters.

Request URL	http://<server>/cgi-bin/configManager.cgi?action=removeConfig			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Sample
name	string	R	configuration param name	General.MachineAddress

Request Example

```
http://192.168.1.108/cgi-bin/configManager.cgi?action=removeConfig&name=General.MachineAddress
```

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

OK

4.2.3 Restore the Config

Restore specified configurations to the default values.

Request URL	http://<server>/cgi-bin/configManager.cgi?action=restore		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
names	char[][32]	R	Array of the configuration names that needs to be restored to the default values.

Request Example

```
http://<server>/cgi-bin/configManager.cgi?action=restore&names[0]=General&names[1]=UPnp
```

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

OK

4.2.4 Restore except the Config

Restore all other configurations except for the specified configurations to the default values.

Request URL	http://<server>/cgi-bin/configManager.cgi?action=restoreExcept		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
names	char[][32]	R	Array of the configuration names that does not need to be restored to the default values.

Request Example

```
http://<server>/cgi-bin/configManager.cgi?action=restoreExcept&names[0]=General&names[1]=UPnp
```

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

OK

4.3 Audio

4.3.1 Get Audio Input Channel Numbers

Get audio input channel number.

Request URL	http://<server>/cgi-bin/devAudioInput.cgi?action=getCollect			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
Request Example				
http://192.168.1.108/cgi-bin/devAudioInput.cgi?action=getCollect				

Response Params (key=value format in body)				
Name	Type	R/O	Description	Example
result	int	R	audio input channel number.	1
Response Example				
result=1				

4.3.2 Get Audio Output Channel Numbers

Get audio output channel number.

Request URL	http://<server>/cgi-bin/devAudioOutput.cgi?action=getCollect			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
Request Example				
http://192.168.1.108/cgi-bin/devAudioOutput.cgi?action=getCollect				

Response Params (key=value format in body)				
Name	Type	R/O	Description	Example
result	int	R	audio output channel number.	2
Response Example				
result=2				

4.3.3 Post Audio Stream

Post audio stream to device, request is very long and client continues to send audio data. If client want to stop, just close the connection.

Request URL	http://<server>/cgi-bin/audio.cgi?action=postAudio			
Method	POST			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
httptype	string	R	audio http transmit format, can be : singlepart: HTTP content is a continuous flow of audio packets multipart: HTTP content type is multipart/x-mixed-replace, and each audio packet ends with a boundary string	singlepart
channel	int	R	The audio channel index which starts from 1.	1
Request Example (singlepart)				
POST http://192.168.1.108/cgi-bin/audio.cgi?action=postAudio&httptype=singlepart&channel=1 HTTP/				

1.1

User-Agent: client/1.0
Content-Type: Audio/G.711A
Content-Length: 9999999

<Audio data>
<Audio data>
...

Request Example (multipart)

POST http://192.168.1.108/cgi-bin/audio.cgi?action=postAudio&httpstype=multipart&channel=1 HTTP/
1.1

User-Agent: client/1.0
Content-Type: multipart/x-mixed-replace; boundary=<boundary>

--<boundary>
Content-Type: Audio/G.711A
Content-Length: 800

<Audio data>
--<boundary>
Content-Type: Audio/G.711A
Content-Length: 800

<Audio data>
--<boundary>
...

HTTP API V3.35 - Intelbras

Response Params (N/A)

Name	Type	R/O	Description	Example
Response Example				
(N/A)				

Appendix A: Audio Encode Type

MIME (Content-Type)	Description
Audio/PCM	PCM
Audio/ADPCM	ADPCM
Audio/G.711A	G.711 A Law
Audio/G.711Mu	G.711 Mu Law
Audio/G.726	G.726
Audio/G.729	G.729
Audio/MPEG2	MPEG2
Audio/AMR	AMR
Audio/AAC	AAC

4.3.4 Get Audio Stream

Get audio stream from device, response is very long and client continues to recv audio data. If client want to stop, just close the connection.

Request URL	http://<server>/cgi-bin/audio.cgi?action=getAudio		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
httpstype	string	R	audio http transmit format, can be : singlepart: HTTP content is a continuous flow of audio packets multipart: HTTP content type is multipart/x-mixed-replace, and each audio packet ends with a boundary string
channel	int	R	The audio channel index which starts from 1.
Request Example			
http://192.168.1.108/cgi-bin/audio.cgi?action=getAudio&httpstype=singlepart&channel=1			

Response Params (binary in body)	
Name	
Type	
R/O	
Description	
Example	
Response Example (singlepart)	
200 OK HTTP/1.1	
Server: device/1.0	
Content-Type: Audio/G.711A	
Content-Length: 9999999	
<Audio data>	
<Audio data>	
...	
Response Example (multipart)	
200 OK HTTP/1.1	
Server: device/1.0	
Content-Type: multipart/x-mixed-replace; boundary=<boundary>	
--<boundary>	
Content-Type: Audio/G.711A	
Content-Length: 800	
<Audio data>	
--<boundary>	
Content-Type: Audio/G.711A	
Content-Length: 800	
<Audio data>	
--<boundary>	
...	

4.3.5 [Config] Volume Control

Volume control configuration parameters:

Config Data Params				
Name	Type	R/O	Description	Example
AudioOutputVolume	int[]	R	Audio output volume. Each element of the array represents the volume value of an audio output channel, and the range is [0-100].	[80,50]

Please refer to "4.2.1 Get and Set Configure" for configuration getting and setting. Specific examples are as follows :

Get Config Request Example

`http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=AudioOutputVolume`

Get Config Response Example

`table.AudioOutputVolume[0]=80`

`table.AudioOutputVolume[1]=50`

`...`

Set Config Request Example

`http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&AudioOutputVolume[0]=80&AudioOutputVolume[1]=50`

Set Config Response Example

`OK`

4.3.6 Getting the Analysis Configurations of the Specified Audio Channel

Channel

Get the analysis configurations of the specified audio channel.

Request URL	<code>http://<server>/cgi-bin/api/AudioAnalyseManager/getConfig</code>			
Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
PresetID	int32	O	Preset. The field is valid when the analysis channel is specified. When there are no fields, the default preset is 0, and the field can be empty for IPC. When the preset is -1, all the preset configurations will be returned.	0
AudioChannel	int32	O	Audio channel number. When there are no signs indicating that the audio source is from the Internet, then the AudioChannel means the local channel number. When using the	0

			number, it must be linked with logical audio channel through channel number or other fields. If it fails to link, the API will be failed to return.	
DeviceID	char[128]	O	Device ID. It is mainly used in the back-end cascade scenes, and if there are no special requirements, it will not be used in front-end devices.	"12345678"

Request Example

```
{
  "PresetID": 0,
  "AudioChannel": 0,
  "DeviceID": "12345678"
}
```

Response Params (JSON format in body)

Name	Type	R/O	Description	Example
Configuration	object	O	Analysis configuration of audio channel	
+AudioChannel	uint32	O	Audio channel number. When there are no signs indicating that the audio source is from the Internet, then the AudioChannel means the local channel number.	0
+Rules	object[]	O	Analysis rule	
++Class	char[64]	O	Rule category	"<ClassType>"
++Type	char[64]	O	Rule type	"<RuleType>"
++Id	uint	O	Rule ID. The field can be 0, which means the system will automatically allocate the appropriate ID. If there are businesses that bonds the rule ID, the client will be responsible for ensuring that the code and the name are unique.	0
++Name	char[64]	O	Rule name	"..."
++Enable	bool	O	Enable the rules	true
++PresetID	uint32	O	Preset spots	0
++EventHandle r	object	O	Rule linkage items	
+++TimeSectio n	char[8][6][20]	O	Alarm period	[["1 00:00:00-08:00:00", "1 10:00:00-24:00:00",]]
+++RecordEna ble	bool	O	Enable recording, with RecordChannels. Record is enabled if	true

			it is true; start recording when the event action is start and stop recording when it is stop. Record is disabled if it is false.	
+++RecordChannels	int[]	O	Recording channel number list One-dimensional array. Each member indicates that the corresponding channel needs to record, and the channel number starts from 0.	[0, 1, 2]
+++RecordLatc h	int	O	Recording delay time (second) Range [10, 300]	10
+++AlarmOutE nable	bool	O	Enable alarm output	true
+++AlarmOutC hannels	int[]	O	Alarm output channel number list One-dimensional array. Each member indicates that the corresponding channel needs to output alarm, and the channel number starts from 0.	[1, 4]
+++AlarmOutL atch	int	O	Output delay time (second) after the alarm output stops. Range: [1, 300]	10
+++SnapshotE nable	bool	O	Enable snapshot	true
+++SnapshotC hannels	int[]	O	Snapshots channel number list One-dimensional array. Each member indicates that the corresponding channel needs to take snapshot, and the channel number starts from 0.	[2, 4]
+++MailEnable	bool	O	Send emails. If there are images, the images will be sent as email attachment.	true
+++Dejitter	int	O	Dejitter time. The field is only meaningful for part of the start or stop type events. Unit: Second (0 s-600 s). It can only be used for applications to automatically tour the detected events, such as motion detection and tampering. The similar intelligent events that reported by the algorithm database such as tripwire and face detection cannot configure dejitter time.	0
+++BeepEnabl e	bool	O	Buzzer	true
+++Delay	uint	O	Set the delay time for the event to take effect. Unit: second [0-300].	30
+++ExAlarmOu tChannels	int[]	O	Expands alarm output channel list One-dimensional array. Each member	[2, 3]

			indicates that the corresponding channel needs to output extension alarm, and the channel number starts from 0.	
+++ExAlarmOutputEnable	bool	O	Enable expand alarm output	true
+++LightingLink	object	O	PTZ illuminator linkage item	
++++Enable	bool	O	Enable	true
++++FilckerLightType	enumchar[32]	O	Flicker light type enumchar[32]{ "WhiteLight": indicates the flashing light is white light "RedBlueLight": indicates the flashing light is red and blue light }	"WhiteLight"
++++LightLinkType	enumchar[32]	O	Light linkage way enumchar[32]{ "Filcker": flicker (by default) "KeepLighting": solid on }	"Filcker"
++++FilckerInterval	float	O	Flicker interval Unit: 0.1 s, and the default value is 0.5 s.	5
++++FilckerTimes	int	O	The flicker times that can be configured. Unit: times, and the default value is 5.	5
++++LightDuration	uint	O	The duration time of the flicker or normally lighting. Unit: second.	10
++++LightBright	uint	O	Brightness of the linkage light. The range refers to the capacity of LinkLightBrightRange. If this item does not exist, the default brightness will be adopted.	50
+++LogEnable	bool	O	Whether to record alarm logs (Includes local and remote logs). The type is fixed to EventStart/EventStop/EventPulse and the original event type is recorded in the Detail.Code field in the log.	false
+++LogRemote	bool	O	Whether to record remote logs. The type is fixed as EventStart/EventStop/EventPulse and the original event type is recorded in the Detail.Code field in the log. (When LogRemote and LogEnable exist at the same time, the remote log records will	false

			use the data from LogRemote).	
+++MMSEnable	bool	O	Enable sending MMS	false
+++MatrixChannels	int[]	O	Link with video matrix channels number	[1, 6]
+++MatrixEnable	bool	O	Enable linkage video matrix	true
+++MessageEnable	bool	O	Whether to upload the message to the alarm central server. Alarm center does not need to log in to the device, and the device itself can directly send the message to alarm center.	true
+++PtzLink	object[][]	O	<p>PTZ linkage. It is a two-dimensional array, each member indicates a linkage item, and the last item indicates channel number. Each linkage item includes 5 components including linkage type, three relating parameters and the PTZ channel it linked with.</p> <p>Linkage types include:</p> <ul style="list-style-type: none"> "Preset": linkage preset; "Tour": linkage tour; "Pattern": linkage mode; "Zoom": linkage zoom "SingleScene": link with single scene. (The second linkage parameter indicates the scene number, and the third linkage parameter indicates the stay time of the scene after the alarm finished, the fourth linkage parameter indicates reserve the parameter and the fifth linkage parameter indicates the PTZ channel it linked with. "QuickFocus": the thermal imaging PTZ links with quick focus function (the second one indicates the zoom of the thermal imaging channel), and its value ranges from 0 to 24. The third parameter indicates the stay time and the fourth indicates reserve the parameter. If it is a positive value, then it means zoom in, and the fifth parameter indicates the PTZ channel number it linked with. "None" can be empty or none, which is used for compatibility. 	[["Preset", 3, 0, 0, 0], ["Tour", 4, 0, 0, 1], ["Pattern", 1, 0, 0, 2]]

+++PtzLinkEnable	bool	O	Enable PTZ linkage	true
+++SnapshotPeriod	int	O	Frame interval. It means the number of frames it takes to capture one image, and in a certain period of time, the number of images depends on the capturing frame rate. 0 indicates it takes multiple snapshots continuously.	3
+++SnapshotTimes	int	O	Times of continuous snapshot. When SnapshotEnable is true, snapshotTimes=0 means the device will continuously take snapshots until the event ends.	5
+++VoiceEnable	bool	O	Voice prompts	true
+++Voice	object	O	Linkage prompted by voice	
++++Channels	int[]	O	The channel number that the audio prompts playing.	[1, 2]
++++AudioFileName	char[256]	O	Absolute path of the linkage audio file. It can be an empty string. If it is, it means the default alarm linkage voice that specified by Sound. AlarmSound. "Null" means no audio linkage.	"/mnt/alarm.g711a"
++++PlayTimes	uint	O	Playing times of the linkage audio.	3
++++LinkTime	uint	O	The playing time of the linkage audio. Unit: second	30
++++Interval	uint	O	The playing interval of the sound, and the default value is 0. Unit: second.	30
++Config	object	O	The specific configuration of rule, and the corresponding configuration depends on the value of Type. For more information, please see AudioAnalyseRule(<RuleType>).	
+Global	object	O	Global configuration. This field has no relationship with PTZ position.	
++ClassList	char[32][64]	O	The valid intelligent business	["<ClassType1>","<ClassType2>"]
++Module	object	O	Business configuration. The field name indicates the business name. For the detailed configuration, please see AudioAnalyseModule(<ClassType>).	
+++SmartSoundDetect	object	O	Sound detection business	
++++Threshold	uint32	O	Sound intensity threshold, and it ranges from [1,100]	50

++Detail	object	O	Other universal or non-service related global configurations.	
+Scene	object[]	O	Scene configuration	
++PresetID	uint	O	Preset ID	0
++ClassList	char[32][64]	O	Valid intelligent business	[<ClassType1>,<ClassType2>]
++Module	object	O	Refer to the module under“Global”	
++Detail	object	O	Other universal or non-service related scene configurations.	
+State	uint32	O	Analysis status of audio channel 0 : not support 1 : support	1

Response Example

```
{
  "Config": {
    "AudioChannel": 0,
    "Global": {
      "ClassList": [],
      "Detail": null,
      "Module": null
    },
    "Rules": [
      {
        "Class": "SmartSoundDetect",
        "Config": {
          "Sensitive": 5,
          "TimesThreshold": 2,
          "Types": [
            "AudioCrashingGlass",
            "AudioScream",
            "AudioGunshot",
            "AudioExplosion"
          ]
        },
        "Enable": false,
        "EventHandler": {
          "TourEnable": false,
          "VoiceEnable": false,
          ...
        },
        "Id": 1,
        "Name": "SmartSoundDetect",
        "PresetID": 0,
        "Type": "SmartSoundDetect"
      }
    ],
    "Scene": [
      ...
    ]
  }
}
```

```
{
    "ClassList": [],
    "Detail": null,
    "Module": {
        "SmartSoundDetect": {
            "Threshold": 50
        }
    },
    "PresetID": 0
},
],
"State": 1
}
}
```

4.3.7 Setting the Analysis Configuration of the Specified Audio Channel

Configure the analysis configuration of the specified audio channel. We recommend you to use it in combination with getConfig. <http://<server>/cgi-bin/api/AudioAnalyseManager/getConfig>

Request URL	http://<server>/cgi-bin/api/AudioAnalyseManager/setConfig		
Method	POST		
Request Params (JSON format in body)			
Name	Type	R/O	Description
Config	object	R	For the detailed content, please see getConfig to request for the returned value. <a href="http://<server>/cgi-bin/api/AudioAnalyseManager/getConfig">http://<server>/cgi-bin/api/AudioAnalyseManager/getConfig
AudioChannel	int32	O	Audio channel number. When there are no signs indicate that the audio source is from the Internet, then the AudioChannel means the local channel number. When using the number, it must be linked with logical audio channel through channel number or other fields. If fails to link, the API will be failed to return.
DeviceID	char[64]	O	Device ID. It mainly used in the back-end cascade scenes, and if there are no special requirements, it will not be used in front-end devices.
Complete Example			
{	"AudioChannel": 0, "DeviceID": "12345678",		

```
"Config": {  
    "AudioChannel": 0,  
    "Global": {  
        "ClassList": [],  
        "Detail": null,  
        "Module": null  
    },  
    "Rules": [  
        {  
            "Class": "SmartSoundDetect",  
            "Config": {  
                "Sensitive": 6,  
                "TimesThreshold": 1,  
                "Types": [  
                    "AudioCrashingGlass",  
                    "AudioScream",  
                    "AudioGunshot",  
                    "AudioExplosion"  
                ]  
            },  
            "Enable": false,  
            "EventHandler": {  
                "TourEnable": false,  
                "VoiceEnable": false,  
                ...  
            },  
            "Id": 1,  
            "Name": "SmartSoundDetect",  
            "PresetID": 0,  
            "Type": "SmartSoundDetect"  
        }  
    ],  
    "Scene": [  
        {  
            "ClassList": [],  
            "Detail": null,  
            "Module": {  
                "SmartSoundDetect": {  
                    "Threshold": 50  
                }  
            },  
            "PresetID": 0  
        }  
    ],  
    "State": 1  
}
```

Response Params (JSON format in body)				
Name	Type	R/O	Description	Example
Response Example				
{}				

4.3.8 Getting the Related Business Configuration of the Specified Audio Channel

Get the relating business configuration of the specified audio channel.

Request URL	http://<server>/cgi-bin/api/AudioAnalyseManager/getClassConfig			
Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
Class	char[64]	R	Audio analysis business	"<ClassType>"
PresetID	int32	O	Preset. If it is empty, the default value 0 is taken and the field can be empty for IPC.	0
AudioChannel	int32	O	Audio channel number. When there are no signs indicating that the audio source is from the Internet, then the AudioChannel means the local channel number. When using the number, it must be linked with logical audio channel through channel number or other fields. If it fails to link, the API will be failed to return.	0
DeviceID	char[128]	O	Device ID. It mainly used in the back-end cascade scenes, and if there are no special requirements, it will not be used in front-end devices.	"12345678"
Complete Example				
{	"Class": "SmartSoundDetect", "PresetID": 0, "AudioChannel": 0, "DeviceID": "12345678"			
}				

Response Params (JSON format in body)				
Name	Type	R/O	Description	Example
Config	object	O		

+Rules	object[]	O		
++Class	char[64]	O	Rule category	"<ClassType>"
++Type	char[64]	O	Rule type	"<RuleType>"
++Id	uint	O	Rule ID. It can be 0, which means the system will automatically allocate the appropriate ID. If there are businesses that bonds the rule ID, the client will be responsible for ensuring that the code and the name are unique.	0
++Name	char[64]	O	Rule name	"..."
++Enable	bool	O	Enable the rules	true
++PresetID	uint32	O	Preset spots	0
++EventHandler	object	O	Rule linkage Items	
+++TimeSection	char[8][6][20]	O	Alarm period	[["1 00:00:00-08:00:00", "1 10:00:00-24:00:00",]]
+++RecordEnable	bool	O	Enable recording, with RecordChannels. Record is enabled if it is true; start recording when the event action is start and stop recording when it is stop. Record is disabled if it is false.	true
+++RecordChannels	int[]	O	Recording channel number list One-dimensional array. Each member indicates that the corresponding channel needs to record, and the channel number starts from 0.	[0, 1, 2]
+++RecordLatch	int	O	Recording delay time (second) Range [10, 300]	10
+++AlarmOutputEnable	bool	O	Enable alarm output	true
+++AlarmOutputChannels	int[]	O	Alarm output channel number list One-dimensional array. Each member indicates that the corresponding channel needs to output alarm, and the channel number starts from 0.	[1, 4]
+++AlarmOutputLatch	int	O	Output delay time (second) after the alarm output stops. Scope [1,300]	10
+++SnapshotEnable	bool	O	Enable snapshot	true
+++SnapshotChannels	int[]	O	Snapshot channel number list One-dimensional array. Each	[2, 4]

			member indicates that the corresponding channel needs to take snapshot, and the channel number starts from 0.	
+++MailEnable	bool	O	Send emails. If there are images, the images will be sent as email attachment.	true
+++Dejitter	int	O	Dejitter time. The field is only meaningful for part of the start or stop type events. Unit: Second (0 s–600 s). It can only be used for applications to automatically tour the detected events, such as motion detection and tampering. The similar intelligent events that reported by the algorithm database such as tripwire and face detection cannot configure dejitter time.	0
+++BeepEnable	bool	O	Buzzer:	true
+++Delay	uint	O	Set the delay time for the event to take effect. Unit: second [0-300].	
+++ExAlarmOutChannels	int[]	O	Expands alarm output channel list One-dimensional array. Each member indicates that the corresponding channel needs to output extension alarm, and the channel number starts from 0.	[2, 3]
+++ExAlarmOutEnable	bool	O	Enable alarm output extension	true
+++LightingLink	object	O	PTZ illuminator linkage item	
++++Enable	bool	O	Enable	true
++++FilckerLightType	enumchar[32]	O	Flicker light type enumchar[32]{ "WhiteLight": indicates the flashing light is white light "RedBlueLight": indicates the flashing light is red and blue light }	"WhiteLight"
++++LightLinkType	enumchar[32]	O	Light linkage way enumchar[32]{ "Filcker": flicker (by default) "KeepLighting": solid on }	"Filcker"
++++FilckerIntervalTime	float	O	Flicker interval Unit: 0.1s, and the default value is	5

			0.5 s.	
++++FlickerTimes	int	O	The flicker times that can be configured. Unit: times, and the default value is 5.	5
++++LightDuration	uint	O	The duration time of the flicker or normally lighting. Unit: second.	10
++++LightBrightness	uint	O	Brightness of the linked light The range refers to the capacity of LinkLightBrightRange. If this item does not exist, the default brightness will be adopted.	50
+++LogEnable	bool	O	Whether to record alarm logs. The type is fixed to EventStart/EventStop/EventPulse and the original event type is recorded in the Detail.Code field in the log.	false
+++LogRemote	bool	O	Whether to record remote logs. The type is fixed as EventStart/EventStop/EventPulse and the original event type is recorded in the Detail.Code field in the log. (When LogRemote and LogEnable exist at the same time, the remote log records will use the data from LogRemote).	false
+++MMSEnable	bool	O	Enable sending MMS	false
+++MatrixChannels	int[]	O	Link with video matrix channels number	[1, 6]
+++MatrixEnable	bool	O	Enable linkage video matrix	true
+++MessageEnable	bool	O	Whether to upload the message to the alarm central server. Alarm center does not need to log in to the device, and the device itself can directly send the message to alarm center.	true
+++PtzLink	object[][]	O	PTZ linkage. It is a two-dimensional array, each member indicates a linkage item, and the last item indicates channel number. Each linkage item includes 5 components including linkage type, three relating parameters and the PTZ channel it linked with.	[["Preset", 3, 0, 0, 0], ["Tour", 4, 0, 0, 1], ["Pattern", 1, 0, 0, 2]]

			<p>Linkage types include:</p> <p>"Preset": linkage preset;</p> <p>"Tour": linkage tour;</p> <p>"Pattern" : linkage mode.</p> <p>"Zoom" : linkage zoom</p> <p>"SingleScene": link with single scene. (The second linkage parameter indicates the scene number, and the third linkage parameter indicates the stay time of the scene after the alarm finished, the fourth linkage parameter indicates reserve the parameter and the fifth linkage parameter indicates the PTZ channel it linked with.</p> <p>"QuickFocus": the thermal imaging PTZ links with quick focus function (the second one indicates the zoom of the thermal imaging channel), and its value ranges from 0 to 24.</p> <p>The third parameter indicates the stay time and the fourth indicates reserve the parameter. If it is a positive value, then it means zoom in, and the fifth parameter indicates the PTZ channel number it linked with.</p> <p>"None" can be empty or none, which is used for compatibility.</p>	
+++PtzLinkEnable	bool	O	Enable PTZ linkage	true
+++SnapshotPeriod	int	O	<p>Frame interval. It means the number of frames it takes to capture one image, and in a certain period of time, the number of images depends on the capturing frame rate.</p> <p>0 indicates it takes multiple snapshots continuously.</p>	3
+++SnapshotTimes	int	O	Times of continuous snapshot. When SnapshotEnable is true, snapshotTimes=0 means the device will continuously take snapshots until the event ends.	5
+++VoiceEnable	bool	O	Voice prompts	true

+++Voice	object	O	Audio Linkage	
++++Channel	int[]	O	Channel	[1, 2]
++++AudioFileName	char[256]	O	Absolute path of the linkage audio file. It can be an empty string. If it is, it means the default alarm linkage voice that specified by Sound. AlarmSound. "Null" means no audio linkage.	"/mnt/alarm.g711a"
++++PlayTimes	uint	O	Play times of the linkage audio.	3
++++LinkTime	uint	O	The playing time of the linkage audio. Unit: second	30
++++Interval	uint	O	The playing interval of the sound, and the default value is 0. Unit: second.	30
++Config	object	O	The specific configuration of rule, and the corresponding configuration depends on the value of Type. For more information, please see AudioAnalyseRule(<RuleType>).	
+Module	object	O	The specific business configuration, and its value refer to the "request" section. For more information, please see AudioAnalyseModule(<ClassType>)	
+Global	object	O		
++Detail	object	O	Universal configuration of the global business	
+Scene	object	O	The business universal configuration of the current preset	
++PresetID	uint32	O	Preset ID	0
++Detail	object	O		

Response Example

```
{
  "Config": {
    "Global": {
      "Detail": null
    },
    "Module": {
      "Threshold": 50
    },
    "Rules": [
      {
        "Class": "SmartSoundDetect",
        "Condition": {
          "Type": "SmartSound"
        }
      }
    ]
  }
}
```

```

    "Config": {
        "Sensitive": 6,
        "TimesThreshold": 1,
        "Types": [
            "AudioCrashingGlass",
            "AudioScream",
            "AudioGunshot",
            "AudioExplosion"
        ]
    },
    "Enable": false,
    "EventHandler": {
        "TourEnable": false,
        "VoiceEnable": false,
        ...
    },
    "Id": 1,
    "Name": "SmartSoundDetect",
    "PresetID": 0,
    "Type": "SmartSoundDetect"
}
],
"Scene": {
    "Detail": null
}
}
}

```

4.3.9 Configuring the Related Business Configuration of the Specified Audio Channel

Configure the analysis configuration of the specified audio channel. We recommend you to use it in combination with getConfig. <http://<server>/cgi-bin/api/AudioAnalyseManager/getConfig>

Request URL	<a href="http://<server>/cgi-bin/api/AudioAnalyseManager/setClassConfig">http://<server>/cgi-bin/api/AudioAnalyseManager/setClassConfig		
Method	POST		
Request Params (JSON format in body)			
Name	Type	R/O	Description
Configuration	object	R	For the detailed content, please see getConfig to request for the returned value. <a href="http://<server>/cgi-bin/api/AudioAnalyseManager/getConfig">http://<server>/cgi-bin/api/AudioAnalyseManager/getConfig
Class	char[64]	R	Audio analysis business
AudioChannel	int32	O	Audio channel number. When there are no signs indicate that the audio source is from the Internet, then the

			AudioChannel means the local channel number. When using the number, it must be linked with logical audio channel through channel number or other fields. If fails to link, the API will be failed to return.	
DeviceID	char[128]	O	Device ID. It is mainly used in the back-end cascade scenes, and if there are no special requirements, it will not be used in front-end devices.	"12345678"
PresetID	int32	O	Preset. When there are no fields, the default preset is 0, and the field can be empty for IPC.	0

Request Example

```
{
  "Config": {
    "Global": {
      "Detail": null
    },
    "Module": {
      "Threshold": 50
    },
    "Rules": [
      {
        "Class": "SmartSoundDetect",
        "Config": {
          "Sensitive": 6,
          "TimesThreshold": 1,
          "Types": [
            "AudioCrashingGlass",
            "AudioScream",
            "AudioGunshot",
            "AudioExplosion"
          ]
        },
        "Enable": false,
        "EventHandler": {
          "TourEnable": false,
          "VoiceEnable": false,
          ...
        },
        "Id": 1,
        "Name": "SmartSoundDetect",
        "PresetID": 0,
        "Type": "SmartSoundDetect"
      }
    ],
  }
}
```

```

    "Scene": {
        "Detail": null
    }
},
"Class": "SmartSoundDetect",
"AudioChannel": 0,
"DeviceID": "12345678",
"PresetID": 0
}

```

Response Params (JSON format in body)

Name	Type	R/O	Description	Example
Response Example				
	{}			

4.3.10 AudioAnalyseRule Config

4.3.10.1 AudioAnalyseRule(SmartSoundDetect)

Name	Type	R/O	Description	Example
Types	char[32][32]	R	Sound Type enumchar[] [32]{ "AudioCry" "AudioAlarm" "AudioGunshot" "AudioExplosion" "AudioScream" "AudioCrashingGlass" }	["AudioCry"]
Sensitive	uint32	O	Detection sensitivity, range [1,10]	5
TimesThreshold	uint32	O	Times threshold, range [1, 5]	2

4.3.11 AudioAnalyseModule Config

4.3.11.1 AudioAnalyseModule(SmartSoundDetect)

Name	Type	R/O	Description	Example
Threshold	uint32	O	Sound intensity threshold, range [1, 100]	50

4.3.12 Play the Specified Audio in the Device

Request URL	http://<server>/cgi-bin/api/Speak/startPlayByTime
Method	POST

Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
Path	string	R	Audio file path	"/Audio_Path"
FileName	string	R	Audio file name	"alarm.wav"
PlayTimes	int32	O	times of broadcasts, default once	1

Request Example

```
{
  "Path": "/Audio_Path",
  "FileName": "alarm.wav",
  "PlayTimes": 1
}
```

Response Params (JSON format in body)				
Name	Type	R/O	Description	Example
Status	enumchar r[32]	R	enumchar[32]{ "OK": indicates that the operation is successful and returns before the end of the broadcast, which is different from API "PostAudio" "Busy": Indicates that the device is broadcasting "Error": Broadcast error }	"OK"

Response Example

```
{
  "Status": "OK"
}
```

4.4 Snapshot

4.4.1 [Config] Snap

Snapshot configuration parameters:

Config Data Params				
Name	Type	R/O	Description	Example
Snap	object[]	R	One element for each channel, and the array index represents the channel number which starts from 0.	[80,50]
+HolidayEnable	bool	O	Whether holiday setting is supported. It is not supported by default.	false
TimeSection	char[8][24][32]	O	Configure the period of capturing snapshots. Two-dimensional array in strings. The first dimension of the array is the day of the week, and the range is [0-7]	[["6 00:00:00-23:59:59", "0 00:00:00-23:59:59", "0 00:00-23:59:59", "0 00:00:00-23:59:59", "0 00:00:00-23:59:59", "0 00:00:00-23:59:59", "0 00:00:00-23:59:59"]]

		<p>(Sunday–Saturday). The last item is for holiday.</p> <p>The second dimension is the period index. One day is divided into multiple periods, and the range is [0–23].</p> <p>Each period is set as a string in the format of:mask hh: mm: ss-hh: mm: ss</p> <p>Mask: The range is [0–4294967295], and each bit carries a special meaning. See the description as follows:</p> <ul style="list-style-type: none"> hh: Hour; range: [0–24] mm: Minute; range: [0–59] ss: Description; ranges [0 to 59] Mask: record type. Meaning of each bit: Bit0: Normal snapshot Bit1: Motion detection snapshot Bit2: Alarm snapshot Bit3: Card-swiping snapshot Bit6: POS snapshot Bit4, Bit5, Bit7–Bit31: Reserved. 	9:59", ...]
--	--	---	--------------

Please refer to "4.2.1 Get and Set Configure" for configuration getting and setting. Specific examples are as follows :

Get Config Request Example

```
http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=Snap
```

Get Config Response Example

```
table.Snap[0].HolidayEnable=true
table.Snap[0].TimeSection[0][0]=6 00:00:00-23:59:59
table.Snap[0].TimeSection[0][1]=0 00:00:00-23:59:59
...
table.Snap[1].HolidayEnable=false
table.Snap[1].TimeSection[0][1]=0 00:00:00-23:59:59
table.Snap[1].TimeSection[0][1]=0 00:00:00-23:59:59
...
```

Set Config Request Example

```
http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&Snap[0].HolidayEnable=true&Snap[0].TimeSection[0][0]=6 00:00:00-23:59:59&Snap[0].TimeSection[0][1]=0 00:00:00-23:59:59&Snap[1].HolidayEnable=false&Snap[1].TimeSection[0][1]=0 00:00:00-23:59:59&Snap[1].TimeSection[0][1]=0 00:00:00-23:59:59
```

Set Config Response Example

```
OK
```

4.4.2 Get a Snapshot

Get snapshots of a specified video channel

Request URL	http://<server>/cgi-bin/snapshot.cgi		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
channel	int	O	Video channel number starting from 1, and the default value is 1.
type	uint32	O	0: Acquire the snapshot from the front end 1: Represent capture of local real-time video secondary decoding encoding Default 0

Request Example

http://192.168.1.108/cgi-bin/snapshot.cgi?channel=1&type=0

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>

4.4.3 Subscribe to Snapshot

Subscribe to the events and event snapshots. For details on the event name and parameters, see the corresponding [Event] sections.

Request URL	http://<server>/cgi-bin/snapManager.cgi?action=attachFileProc		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
channel	int	O	Video channel number, starting from 1. If not specified, the default value is 1. -1 means subscribing to all channels.
heartbeat	int	O	Heartbeat interval. Unit: s Range: [1–60], and the default value is 5. While sending the events, the device regularly sends a heartbeat message to keep itself alive. The message is a string named "heartbeat".
Flags	char[][32]	R	Subscription parameter. The value "Event" should be included.
Events	char[][32]	R	The list of event codes Select All to subscribe all events Common event codes are as follows:

		<p>VideoMotion: Motion detection VideoLoss: Video loss VideoBlind: Video Tampering AlarmLocal: Local alarms TrafficJunction: ANPR TrafficRetrograde: Wrong-way driving TrafficParking: Illegal parking TrafficJam: Traffic congestion TrafficThrow: Throwing out of the vehicle TrafficPedestrian: A pedestrian walks onto a motor vehicle lane TrafficParkingSpaceParking: A vehicle is in the parking space TrafficParkingSpaceNoParking: No vehicle is in the parking space TrafficParkingSpaceOverLine: A vehicle crosses the line of a parking space TrafficManualSnap: Manual snapshot capturing FaceRecognition: Face recognition TrafficParkingStatistics: Parking space counting AccessControl: Access control unlocking ... </p>	
--	--	---	--

Request Example

```
http://192.168.1.108/cgi-bin/snapManager.cgi?action=attachFileProc&channel=1&heartbeat=5&Flags[0]=Event&Events=[VideoMotion%2CVideoLoss]
```

Response Params (multipart in body)				
Name	Type	R/O	Description	Example
Events	object[]	R	Event detail array	
Channel	int	O	The video channel number of the generated picture, starting from 0.	0
+EventBaseInfo	object	R	Shared basic information for all events	
++Code	char[32]	R	Event code See the description on Events	""FaceRecognition""
++Action	char[16]	R	Event action Value: "Start", "Stop" and "Pulse"	"Pulse"
++Index	int	R	The channel number where the event happens, starting from 0	0
+...	Unique parameters of each event For details on each event, see the corresponding section.	

Response Example

```
HTTP/1.1 200 OK
Server: Device/1.0
Content-Type: multipart/x-mixed-replace; boundary=<boundary>
```

```
Connection: closed
```

```
--<boundary>
```

```
Content-Type: text/plain
```

```
Content-Length: <data length>
```

```
Events[0].Channel=0
```

```
Events[0].EventBaseInfo.Code=TrafficJunction
```

```
Events[0].EventBaseInfo.Action=Pulse
```

```
Events[0].EventBaseInfo.Index=0
```

```
Events[0].CountInGroup=3
```

```
Events[0].IndexInGroup=1
```

```
Events[0].Lane=1
```

```
Events[0].PTS=42949485818.0
```

```
Events[0].TrafficCar.PlateNumber=ZZZ12345
```

```
Events[0].TrafficCar.DeviceAddress=XXRoad
```

```
.....
```

```
Events[1].Channel=1
```

```
Events[1].EventBaseInfo.Code=TrafficJunction
```

```
Events[1].EventBaseInfo.Action=Pulse
```

```
Events[1].EventBaseInfo.Index=1
```

```
Events[1].CountInGroup=3
```

```
Events[1].IndexInGroup=2
```

```
Events[1].Lane=1
```

```
.....
```

```
--<boundary>
```

```
Content-Type: image/jpeg
```

```
Content-Length:<image size>
```

```
<JPEG image data>
```

```
--<boundary>
```

```
Content-Type: text/plain
```

```
Content-Length:<data length>
```

```
Heartbeat
```

```
--<boundary>
```

```
.....
```

4.5 Video Attributes

4.5.1 Get Max Extra Stream Numbers

Get the maximum number of sub streams

Request URL	http://<server>/cgi-bin/magicBox.cgi?action=getProductDefinition
Method	GET
Request Params (key=value format in URL)	

Name	Type	R/O	Description	Example
name	char[]	R	Name Fixed value: MaxExtraStream	"MaxExtraStream"

Request Example

http://192.168.1.108/cgi-bin/magicBox.cgi?action=getProductDefinition&name=MaxExtraStream

Response Params (key=value format in body)				
Name	Type	R/O	Description	Example
table	object	R	Return content	
+MaxExtraStream	int	R	The maximum number of sub streams Values: 1, 2 and 3.	1

Response Example

table.MaxExtraStream=1

4.5.2 Get Encode Capability

Get encoding capabilities.

Request URL	http://<server>/cgi-bin/encode.cgi?action=getCaps			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
Request Example				
http://192.168.1.108/cgi-bin/encode.cgi?action=getCaps				

Response Params (key=value format in body)				
Name	Type	R/O	Description	Example
caps	object	R	Encoding capability	
+MaxExtraStream	int	O	Maximum number of sub stream channels	3
+PreviewSplitNumList	int[]	O	The live view page can be split to multiple windows	[1, 4, 9, 16]
+PlaybackCompressedSplitNumList	int[]	O	Secondary compressed playback supports being split to multiple windows.	[1, 4, 9, 16]
+PreviewMode	char[16]	O	Multi-window live view mode. Values: SplitEncode: Split encoding mode SnapShot: Fast snapshot mode SplitSnap: Split fast snapshot mode If the field does not exist or is null, it means the device does not support multi-window live view.	"SplitEncode"
+VideoEncodeDevices	object[]	O	Encoding capability Each element of the array represents a channel.	
++SupportIndividualResolution	bool	O	Whether independent resolution for snapshots is supported	true
++RecordIndividualResolution	bool	O	Whether independent resolutions for recordings are supported	true

alResolution			motion detection and alarm videos are supported.	
++MaxCIFPFrame Size	int	O	CIF maximum value of the P frame. Unit: Kbits	40
++MinCIFPFrame Size	int	O	CIF minimum value of the P frame. Unit: Kbits	7
+CoriaDisplay	object[]	O	Display capability of mixed exposure. Each element of the array represents a channel.	
++CoriaDisplaySw itchs	int[]	O	Supported display mode: 0: Default 1: Vehicle mode 2: Person mode	[0,1,2]

Response Example

```

caps.MaxExtraStream=3
caps.PreviewSplitNumList[0]=1
caps.PreviewSplitNumList[1]=4
caps.PreviewSplitNumList[2]=9
caps.PlaybackCompressSplitNumList[0]=1
caps.PlaybackCompressSplitNumList[1]=4
caps.PlaybackCompressSplitNumList[2]=9
caps.PreviewMode=SplitSnap
caps.VideoEncodeDevices[0].SupportIndividualResolution=true
caps.VideoEncodeDevices[0].RecordIndividualResolution=true
caps.VideoEncodeDevices[0].MaxCIFPFrameSize=40
caps.VideoEncodeDevices[0].MinCIFPFrameSize=7
caps.CoriaDisplay[0].CoriaDisplaySwitchs[0]=0
caps.CoriaDisplay[0].CoriaDisplaySwitchs[1]=1
caps.CoriaDisplay[0].CoriaDisplaySwitchs[2]=2

```

4.5.3 Get Encode Config Capability

Get encoding configuration capabilities

Request URL	http://<server>/cgi-bin/encode.cgi?action=getConfigCaps			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
channel	int	O	Video channel number. It an integer starting from 1. When the channel does not exist, the default number is 1.	
Encode	object[]	O	The encoding configurations set before will influence the encoding configurations of other channels and streams.	
+MainFormat	object[]	O	Main stream. Each element of the array represents a recording type.	

			0: General encoding 1: Motion detection encoding 2: Alarm encoding	
++Video	object	O	Video format	
+++Compression	char[16]	O	Video encoding mode For example, "MPEG4", "H.264", "H.265"	"H.264"
+++Width	int	O	Video width Unit: pixels	1920
+++Height	int	O	Video height Unit: pixels	1080
+++BitRate	int	O	Video stream Unit: kbps	2048
+ExtraFormat	object[]	O	Sub stream Each element of the array represents a sub stream. 0: Sub stream 1 1: Sub stream 2 2: Sub stream 3	
++Video	object	O	Video format	
+++Compression	char[16]	O	Video encoding mode For example, "MPEG4", "H.264", "H.265"	"H.264"
+++Width	int	O	Video width Unit: pixels	720
+++Height	int	O	Video height Unit: pixels	576
+++BitRate	int	O	Video stream Unit: kbps	512

Request Example

http://192.168.1.108/cgi-bin/encode.cgi?action=getConfigCaps&channel=1&Encode[0].MainFormat[0].Video.Compression=H.264&Encode[0].MainFormat[0].Video.Width=1920&Encode[0].MainFormat[0].Video.Height=1080

Response Params (key=value format in body)				
Name	Type	R/O	Description	Example
caps	object	R	Encoding configuration capability	
+MainFormat	object[]	O	Main stream. Each element of the array represents a recording type. 0: General encoding 1: Motion detection encoding 2: Alarm encoding	
++Audio	object	O	Audio encoding capability	
+++CompressionTypes	char[]	O	Supported audio encoding formats: PCM, ADPCM, G.711A, G.711Mu, G.726, G.729, MPEG2, AMR, and SVAC.	G.711A, AMR
++Video	object	O	Video encoding capability	
+++CompressionT	char[]	O	Supported video encoding formats:	MPEG4, H.264,

ypes			MPEG4, MPEG2, MPEG1, MJPG, H.263, H.264 and H.265.	H.265
+++ResolutionTypes	char[]	O	Supported video resolutions. See the table below for the optional values.	D1, 1920x1080
+++BitRateOptions	char[]	O	Stream capability Two numbers are divided by a comma. The former number represents the minimal stream, and the latter represents the maximum stream. Unit: kbps	80,448
+++FPSMax	int	O	Maximum frame rate	25
+ExtraFormat	object[]	O	Sub stream. Each element of the array represents a sub stream. 0: Sub stream 1 1: Sub stream 2 2: Sub stream 3	
Audio In	object	O	Audio encoding capabilities	
+++CompressionTypes	char[]	O	Supported audio encoding formats: PCM, ADPCM, G.711A, G.711Mu, G.726, G.729, MPEG2, AMR, and SVAC.	G.711A, AMR
++Video	object	O	Video encoding capability	
+++CompressionTypes	char[]	O	Supported video encoding formats: MPEG4, MPEG2, MPEG1, MJPG, H.263, H.264 and H.265.	MPEG4, H.264, H.265
+++ResolutionTypes	char[]	O	Supported video resolutions. See the table below for the options.	D1, 1280x720
+++BitRateOptions	char[]	O	Stream capability Two numbers are divided by a comma. The former number represents the minimal stream, and the latter represents the maximum stream. Unit: kbps	80,448
+++FPSMax	int	O	Maximum frame rate:	25
+SnapFormat	object[]	O	Snapshot encoding capability Each element of the array represents a snapshot capturing type. 0: General snapshot 1: Motion detection snapshot 2: Alarm snapshot	
++Video	object	O	Snapshot format	
+++CompressionTypes	char[16]	O	Supported encoding formats For example, "JPEG"	JPEG
+++ResolutionTypes	int	O	Supported video resolutions See the table below for the options.	D1, 1280x720
+++FPSMax	int	O	Maximum frame rate	2
Response Example				

```

caps.MainFormat[0].Audio.CompressionTypes=G.711A, AMR
caps.MainFormat[0].Video.CompressionTypes=H.264,MJPEG
caps.MainFormat[0].Video.ResolutionTypes=2048 x 1536,1080,SXGA, 1280 x 960,720,D1,CIF
caps.MainFormat[0].Video.BitRateOptions=448,2560
caps.MainFormat[0].Video.FPSMax=25
caps.ExtraFormat[0].Audio.CompressionTypes=G.711A, AMR
caps.ExtraFormat[0].Video.CompressionTypes=H.264,MJPEG
caps.ExtraFormat[0].Video.ResolutionTypes=2048 x 1536,1080,SXGA, 1280 x 960,720,D1,CIF
caps.ExtraFormat[0].Video.BitRateOptions=448,2560
caps.ExtraFormat[0].Video.FPSMax=25
caps.SnapFormat[0].Video.CompressionTypes=JPEG
caps.SnapFormat[0].Video.ResolutionTypes=2048 x 1536,1080,SXGA, 1280 x 960,720,D1,CIF
caps.SnapFormat[0].Video.FPSMax=2

```

Appendix A: Video Resolution

Fixed Resolution Name	Size in PAL	Size in NTSC
"D1"	704 x 576	704 x 480
"HD1"	352 x 576	352 x 480
"BCIF"/"2CIF"	704 x 288	704 x 240
"CIF"	352 x 288	352 x 240
"QCIF"	176 x 144	176 x 120
"NHD"	640 x 360	—
"VGA"	640 x 480	—
"QVGA"	320 x 240	—
"SVCD"	480 x 480	—
"QQVGA"	160 x 128	—
"SVGA"	800 x 592	—
"SVGA1"	800 x 600	—
"WVGA"	800 x 480	—
"FWVGA"	854 x 480	—
"DVGA"	960 x 640	—
"XVGA"	1024 x 768	—
"WXGA"	1280 x 800	—
"WXGA2"	1280 x 768	—
"WXGA3"	1280 x 854	—
"WXGA4"	1366 x 768	—
"SXGA"	1280 x 1024	—
"SXGA+"	1400 x 1050	—
"WSXGA"	1600 x 1024	—
"UXGA"	1600 x 1200	—
"WUXGA"	1920 x 1200	—
"ND1"	240 x 192	—
"720P"	1280 x 720	—
"1080P"	1920 x 1080	—
"QFHD"	3840 x 2160	—
"1_3M", "1280x960"	1280 x 960 (1.3 Mega Pixels)	—

Fixed Resolution Name	Size in PAL	Size in NTSC
"2_5M", "1872x1408"	1872 x 1408 (2.5 Mega Pixels)	—
"5M", "3744x1408"	3744 x 1408 (5 Mega Pixels)	—
"3M", "2048x1536"	2048 x 1536 (3 Mega Pixels)	—
"5_0M", "2432x2048"	2432 x 2048 (5 Mega Pixels)	—
"1_2M", "1216x1024"	1216 x 1024 (1.2 Mega Pixels)	—
"1408x1024"	1408 x 1024 (1.5 Mega Pixels)	—
"3296x2472"	3296 x 2472 (8 Mega Pixels)	—
"5_1M", "2560x1920"	2560 x 1920 (5 Mega Pixels)	—
"960H",	960 x 576	960 x 480
"DV720P"	960 x 720	—
"2560x1600"	2560 x 1600 (4 Mega Pixels)	—
"2336x1752"	2336 x 1752 (4 Mega Pixels)	—
"2592x2048"	2592 x 2048	—
"2448x2048"	2448 x 2048	—
"1920x1440"	1920x1440	—
"2752x2208"	2752x2208	—
"3840x2160"	3840x2160	—
"4096x2160"	4096x2160	—
"3072x2048"	3072x2048	—

4.5.4 [Config] Encode of Media

Config Data Params				
Name	Type	R/O	Description	Example
Encode	object[]	O	encode config. array index starts from 0, which means video channel (equals to video channel index -1, and so 0 means channel 1)	
+MainFormat	object[]	O	main format array index starts from 0, which means : 0 = regular encode 1 = motion detection encode 2 = alarm encode 3= emergency encode	
++AudioEnable	bool	O	Enable/Disable audio	true
++Audio	object	O	audio	
+++Bitrate	int	O	Unit is kbps	64
+++Compression	char[16]	O	Range depends on capacity in GetAudioConfigCaps	"PCM"
+++Frequency	uint	O	Audio sampling frequency 8K~192K	44100

+++Depth	uint8	O	Audio sampling depth	16
+++PacketPeriod	int	O	Packaging cycle, Unit is ms, where 0 indicates using the default cycle. Range[10, 250]	40
+++Mode	uint8	O	code rate Different encoding types have different definitions: "AMR" code rate 0: 4.75kbps, 1: 5.15 kbps, 2: 5.9 kbps, 3: 6.7 kbps, 4: 7.4 kbps, 5: 7.95 kbps, 6: 10.2 kbps, 7: 12.2 kbps, G726 code rate 0: 40kbps(default) 1: 16kbps 2: 24kbps 3: 32kbps	0
+++Pack	char[16]	O	"DHAV", "PS"	"PS"
+++Channels	uint[8]	O	List of accompanying audio channels for this video channel	[0, 1]
++VideoEnable	bool	O	True: enable video	true
++Video	object	O	video	
+++Compression	char[16]	O	Range is {MPEG4, MPEG2, MPEG1, MJPG, H.263, H.264, H.265, SVAC2} Depends on capacity in GetVideoConfigCaps	"H.264"
+++Width	int	O	Video Width	720
+++Height	int	O	Video height	576
+++CustomResolutionName	char[16]	O	Resolution Name	"D1"
+++BitRateControl	char[16]	O	Range is {CBR, VBR} CBR: constant bitrates VBR: variable bitrates	"CBR"
+++BitRate	int	O	Unit is Kbps Range depends on capability in GetVideoConfigCaps	384
+++FPS	double	O	Frames per second. <1.0: several seconds/frame, FPS=0.3333: 3 seconds per frame. >1.0: several frames/second. FPS=3: 3 frames per second. Range depends on capability	25.0

			in GetVideoConfigCaps	
+++GOP	int	O	Range is [1—100]. Group of picture, it's the interval of I Frame, Example: GOP=50, means there is one I frame every 49 P or B frames	50
+++QualityRange	uint8	O	quality range 6~100, default 6	6
+++SVCLayer	int	O	Number of SVC-T layers The number of SVC-T layers is only valid in H.264 encoding mode, 0 is not used, and N represents the number of layers 1-N in the SVC. The higher the level, the less important it is. When N is 1, the effect is the same as not opening, that is, 0 and 1 have the same effect. Default 0	0
+++Quality	int	O	Range is [1—QualityRange]. Image Quality, available when Video.BitRateControl=VBR 1: worst quality QualityRange: best quality	4
+++Priority	int	O	Range is 0 or 1 0 for Image Quality first 1 for Bitstream first	0
+++Pack	char[16]	O	"DHAV","PS"	"DHAV"
+++Profile	char[16]	O	when video compression is H.264, range is { Baseline, Main , Extended , High } when video compression is H.265, range is {"Main","Main10"}	"Baseline"
+ExtraFormat	object[]	O	extra stream, details refer to MainFormat array index starts from 0, which means: 0 = extra stream 1 1 = extra stream 2 2 = extra stream 3	
+SnapFormat	object[]	O	snap stream, details refer to MainFormat. only for dahua devices. array index starts from 0, which means: 0-normal snap 1-motion detection snap 2-alarm detection	

Please refer to "4.2.1 Get and Set Configure" for configuration getting and setting. Specific examples are as follows :

Get Config Request Example

<http://10.0.0.8/cgi-bin/configManager.cgi?action=getConfig&name=Encode>

Get Config Response Example

```
table.Encode[0].MainFormat[0].Audio.Bitrate=64
table.Encode[0].MainFormat[0].Audio.Channels[0]=0
table.Encode[0].MainFormat[0].Audio.Compression=G.711A
table.Encode[0].MainFormat[0].Audio.Depth=16
table.Encode[0].MainFormat[0].Audio.Frequency=8000
table.Encode[0].MainFormat[0].Audio.Mode=0
table.Encode[0].MainFormat[0].Audio.Pack=PS
table.Encode[0].MainFormat[0].AudioEnable=true
table.Encode[0].MainFormat[0].Video.resolution=1920x1080
table.Encode[0].MainFormat[0].Video.BitRate=4096
table.Encode[0].MainFormat[0].Video.BitRateControl=CBR
table.Encode[0].MainFormat[0].Video.Compression=H.264
table.Encode[0].MainFormat[0].Video.CustomResolutionName=1080P
table.Encode[0].MainFormat[0].Video.FPS=18
table.Encode[0].MainFormat[0].Video.GOP=36
table.Encode[0].MainFormat[0].Video.Height=1080
table.Encode[0].MainFormat[0].Video.Pack=PS
table.Encode[0].MainFormat[0].Video.Profile=High
table.Encode[0].MainFormat[0].Video.Quality=4
table.Encode[0].MainFormat[0].Video.QualityRange=6
table.Encode[0].MainFormat[0].Video.SVCTLayer=1
table.Encode[0].MainFormat[0].Video.Width=1920
table.Encode[0].MainFormat[0].Video.Priority=0
table.Encode[0].MainFormat[0].VideoEnable=true
table.Encode[0].MainFormat[1].Audio.Bitrate=64
table.Encode[0].MainFormat[1].Audio.Channels[0]=0
table.Encode[0].MainFormat[1].Audio.Compression=G.711A
table.Encode[0].MainFormat[1].Audio.Depth=16
table.Encode[0].MainFormat[1].Audio.Frequency=8000
table.Encode[0].MainFormat[1].Audio.Mode=0
table.Encode[0].MainFormat[1].Audio.Pack=PS
table.Encode[0].MainFormat[1].AudioEnable=true
table.Encode[0].MainFormat[1].Video.resolution=1920x1080
table.Encode[0].MainFormat[1].Video.BitRate=4096
table.Encode[0].MainFormat[1].Video.BitRateControl=CBR
table.Encode[0].MainFormat[1].Video.Compression=H.264
table.Encode[0].MainFormat[1].Video.CustomResolutionName=1080P
table.Encode[0].MainFormat[1].Video.FPS=18
table.Encode[0].MainFormat[1].Video.GOP=36
table.Encode[0].MainFormat[1].Video.Height=1080
table.Encode[0].MainFormat[1].Video.Pack=PS
table.Encode[0].MainFormat[1].Video.Profile=High
```

```
table.Encode[0].MainFormat[1].Video.Quality=4
table.Encode[0].MainFormat[1].Video.QualityRange=6
table.Encode[0].MainFormat[1].Video.SVCTLayer=1
table.Encode[0].MainFormat[1].Video.Width=1920
table.Encode[0].MainFormat[1].Video.Priority=0
table.Encode[0].MainFormat[1].VideoEnable=true
table.Encode[0].MainFormat[2].Audio.Bitrate=64
table.Encode[0].MainFormat[2].Audio.Channels[0]=0
table.Encode[0].MainFormat[2].Audio.Compression=G.711A
table.Encode[0].MainFormat[2].Audio.Depth=16
table.Encode[0].MainFormat[2].Audio.Frequency=8000
table.Encode[0].MainFormat[2].Audio.Mode=0
table.Encode[0].MainFormat[2].Audio.Pack=PS
table.Encode[0].MainFormat[2].AudioEnable=true
table.Encode[0].MainFormat[2].Video.resolution=1920x1080
table.Encode[0].MainFormat[2].Video.BitRate=4096
table.Encode[0].MainFormat[2].Video.BitRateControl=CBR
table.Encode[0].MainFormat[2].Video.Compression=H.264
table.Encode[0].MainFormat[2].Video.CustomResolutionName=1080P
table.Encode[0].MainFormat[2].Video.FPS=18
table.Encode[0].MainFormat[2].Video.GOP=36
table.Encode[0].MainFormat[2].Video.Height=1080
table.Encode[0].MainFormat[2].Video.Pack=PS
table.Encode[0].MainFormat[2].Video.Profile=High
table.Encode[0].MainFormat[2].Video.Quality=4
table.Encode[0].MainFormat[2].Video.QualityRange=6
table.Encode[0].MainFormat[2].Video.SVCTLayer=1
table.Encode[0].MainFormat[2].Video.Width=1920
table.Encode[0].MainFormat[2].Video.Priority=0
table.Encode[0].MainFormat[2].VideoEnable=true
table.Encode[0].MainFormat[3].Audio.Bitrate=64
table.Encode[0].MainFormat[3].Audio.Channels[0]=0
table.Encode[0].MainFormat[3].Audio.Compression=G.711A
table.Encode[0].MainFormat[3].Audio.Depth=16
table.Encode[0].MainFormat[3].Audio.Frequency=8000
table.Encode[0].MainFormat[3].Audio.Mode=0
table.Encode[0].MainFormat[3].Audio.Pack=PS
table.Encode[0].MainFormat[3].AudioEnable=true
table.Encode[0].MainFormat[3].Video.resolution=704x576
table.Encode[0].MainFormat[3].Video.BitRate=2048
table.Encode[0].MainFormat[3].Video.BitRateControl=VBR
table.Encode[0].MainFormat[3].Video.Compression=H.264
table.Encode[0].MainFormat[3].Video.FPS=25
table.Encode[0].MainFormat[3].Video.GOP=50
table.Encode[0].MainFormat[3].Video.Height=576
table.Encode[0].MainFormat[3].Video.Pack=PS
table.Encode[0].MainFormat[3].Video.Profile=Main
```

```

table.Encode[0].MainFormat[3].Video.Quality=4
table.Encode[0].MainFormat[3].Video.QualityRange=6
table.Encode[0].MainFormat[3].Video.SVCTLayer=1
table.Encode[0].MainFormat[3].Video.Width=704
table.Encode[0].MainFormat[3].Video.Priority=0
table.Encode[0].MainFormat[3].VideoEnable=true
table.Encode[0].ExtraFormat[0].Audio.Bitrate=64
...
table.Encode[0].SnapFormat[0].Audio.Bitrate=64
...

```

Set Config Request Example

[http://10.0.0.8/cgi-bin/configManager.cgi?action=setConfig&Encode\[1\].MainFormat\[0\].Video.Compression=MPEG4](http://10.0.0.8/cgi-bin/configManager.cgi?action=setConfig&Encode[1].MainFormat[0].Video.Compression=MPEG4)

Set Config Response Example

OK

4.5.5 [Config] Encode of Region Interested

Config Data Params				
Name	Type	R/O	Description	Example
VideoEncode ROI	object[]	O	ROI(Region of Interest) Array. One element for each channel, and the array index represents the channel number which starts from 0.	
+Regions	int[][4]	O	each point has x and y value, coordinate remap to 0 — 8192.	[,]
+Quality	uint8	O	Image Quality, [1,6] 1: worst quality 6: best quality	4
+Main	bool	O	Enable/Disable	true
+Extra1	bool	O	Enable/Disable	true
+Extra2	bool	O	Enable/Disable	true
+Extra3	bool	O	Enable/Disable	true
+Extra4	bool	O	Enable/Disable	true
+Snapshot	bool	O	Enable/Disable	true
+DynamicTrac k	bool	O	Enable/Disable	true
+DynamicDela yTime	uint16	O	uint is s. default:60	60

Please refer to "4.2.1 Get and Set Configure" for configuration getting and setting. Specific examples are as follows :

Get Config Request Example

<http://10.0.0.8/cgi-bin/configManager.cgi?action=getConfig&name=VideoEncodeROI>

Get Config Response Example

```
table.VideoEncodeROI[0].Regions[0][0]=0  
table.VideoEncodeROI[0].Regions[0][1]=0  
table.VideoEncodeROI[0].Regions[0][2]=0  
table.VideoEncodeROI[0].Regions[0][3]=0  
table.VideoEncodeROI[0].Regions[1][0]=0  
...  
table.VideoEncodeROI[0].Quality=4,  
table.VideoEncodeROI[0].Main=true,  
table.VideoEncodeROI[0].Extra1=true,  
table.VideoEncodeROI[0].Extra2=true,  
table.VideoEncodeROI[0].Extra3=true,  
table.VideoEncodeROI[0].Snapshot=true  
table.VideoEncodeROI[0].DynamicTrack=true,  
table.VideoEncodeROI[0].DynamicDelayTime=60
```

Set Config Request Example

[http://10.0.0.8/cgi-bin/configManager.cgi?action=setConfig&VideoEncodeROI\[0\].DynamicTrack=true](http://10.0.0.8/cgi-bin/configManager.cgi?action=setConfig&VideoEncodeROI[0].DynamicTrack=true)

Set Config Response Example

OK

4.5.6 [Config] Channel Title

Channel title configuration parameters:

Config Data Params

Name	Type	R/O	Description	Example
ChannelTitle	object[]	R	Array One element for each channel, and the array index represents the channel number which starts from 0.	
+Name	char[]	R	The content represented in the form of string. ' ' means line change, and up to two lines are supported.	"CAM1 123"

Please refer to "4.2.1 Get and Set Configure" for configuration getting and setting. Specific examples are as follows :

Get Config Request Example

<http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=ChannelTitle>

Get Config Response Example

```
table.ChannelTitle[0].Name=CAM1|123  
table.ChannelTitle[1].Name=CAM2|456  
...
```

Set Config Request Example

```
http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&ChannelTitle[0].Name=CAM1|123&ChannelTitle[1].Name=CAM2|456
```

Set Config Response Example

```
OK
```

4.5.7 Get Video Input Channels Device Supported

Request URL	http://<server>/cgi-bin/devVideoInput.cgi?action=getCollect		
Method	GET		
Request Params (none)			
Name	Type	R/O	Description
Request Example			
http://192.168.1.108/cgi-bin/devVideoInput.cgi?action=getCollect			

Response Params (key=value format in body)

Name	Type	R/O	Description	Example
result	uint	O	the video input channel numbers	1
Response Example				
result=1				

4.5.8 Get Video Output Channels Device Supported

Request URL	http://<server>/cgi-bin/devVideoOutput.cgi?action=getCollect		
Method	GET		
Request Params (none)			
Name	Type	R/O	Description
Request Example			
http://192.168.1.108/cgi-bin/devVideoOutput.cgi?action=getCollect			

Response Params (key=value format in body)

Name	Type	R/O	Description	Example
result	uint	O	the video output channel numbers	2
Response Example				
result=2				

4.5.9 Get Max Remote Input Channels

Request URL	http://<server>/cgi-bin/magicBox.cgi?action=getProductDefinition		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
Request Example			
name="MaxRemoteInputCh"			

				annels"
Request Example				
http://192.168.1.108/cgi-bin/magicBox.cgi?action=getProductDefinition&name=MaxRemoteInputChannels				

Response Params (key=value format in body)				
Name	Type	R/O	Description	Example
table	object	O		
+MaxRemoteInputChannels	int	R	the max remote input channel num	16
Response Example				
table.MaxRemoteInputChannels=16				

4.5.10 [Config] Video Standard

Config Data Params				
Name	Type	R/O	Description	Example
VideoStandard	char[16]	O	range is {PAL, NTSC}	"PAL"

Please refer to "4.2.1 Get and Set Configuration" for configuration getting and setting. Specific examples are as follows :

Get Config Request Example

http://10.0.0.8/cgi-bin/configManager.cgi?action=getConfig&name=VideoStandard

Get Config Response Example

table.VideoStandard=PAL

Set Config Request Example				
http://10.0.0.8/cgi-bin/configManager.cgi?action=setConfig&VideoStandard=PAL				
Set Config Response Example				
OK				

4.5.11 [Config] Video Widget

Config Data Params				
Name	Type	R/O	Description	Example
VideoWidget	object[]	O	Array. One element for each channel, and the array index represents the channel number which starts from 0.	
+ChannelTitle	object	O	channel title	
++BackColor	uint8[4]	O	Range is [0—255]. BackColor[0]:red value BackColor[1]:green value BackColor[2]:blue value BackColor[3]: alpha value	[0,0,0,128]

++FrontColor	uint8[4]	O	Range is [0—255]. FrontColor[0]:red value FrontColor[1]:green value FrontColor[2]:blue value FrontColor[3]: alpha value	[255,255,255,0]
++Rect	uint16[4]	O	Range is [0—8191]. Rect[0]: top left corner x coordinate (left) Rect[1]: top left corner y coordinate (top) Rect[2]: bottom right x coordinate (right) Rect[3]: bottom right y coordinate (bottom)	[147, 349, 2769, 784]
++EncodeBlend	bool	O	false - widget blend is not enabled.	true
+TimeTitle	object	O	time title	
++BackColor	uint8[4]	O	Range is [0—255]. BackColor[0]:red value BackColor[1]:green value BackColor[2]:blue value BackColor[3]: alpha value	[0,0,0,128]
++FrontColor	uint8[4]	O	Range is [0—255]. FrontColor[0]:red value FrontColor[1]:green value FrontColor[2]:blue value FrontColor[3]: alpha value	[255,255,255,0]
++Rect	uint16[4]	O	Range is [0—8191]. Rect[0]: top left corner x coordinate (left) Rect[1]: top left corner y coordinate (top) Rect[2]: bottom right x coordinate (right) Rect[3]: bottom right y coordinate (bottom)	[147, 349, 2769, 784]
++EncodeBlend	bool	O	false - widget blend is not enabled.	true
++ShowWeek	bool	O	whether to display the day of the week	true
+Covers	object[16]	O	covers	
++BackColor	uint8[4]	O	Range is [0—255]. BackColor[0]:red value BackColor[1]:green value BackColor[2]:blue value BackColor[3]: alpha value	[0,0,0,128]
++FrontColor	uint8[4]	O	Range is [0—255]. FrontColor[0]:red value FrontColor[1]:green value	[255,255,255,0]

			FrontColor[2]:blue value FrontColor[3]: alpha value	
++Rect	uint16[4]	O	Range is [0—8191]. Rect[0]: top left corner x coordinate (left) Rect[1]: top left corner y coordinate (top) Rect[2]: bottom right x coordinate (right) Rect[3]: bottom right y coordinate (bottom)	[147, 349, 2769, 784]
++EncodeBlend	bool	O	false - widget blend is not enabled.	true
+UserDefinedTitle	object[]	O	custom title	
++Text	char[1024]	O	Title content. Character " " means newlines. Limit: Support 2 lines at most.	"11 222"
++TextAlign	uint	O	text alignment Default Right Alignment 0x0000, left aligned 0x0001, align in X coordinate 0x0010, align in Y coordinate 0x0011, centered 0x0002, right aligned 0x0020, aligned at the top 0x0000, align according to bottom 0x0000, aligned according to the top left corner 0x0100, line wrap alignment	0x0001
++BackColor	uint8[4]	O	Range is [0—255]. BackColor[0]:red value BackColor[1]:green value BackColor[2]:blue value BackColor[3]: alpha value	[0,0,0,128]
++FrontColor	uint8[4]	O	Range is [0—255]. FrontColor[0]:red value FrontColor[1]:green value FrontColor[2]:blue value FrontColor[3]: alpha value	[255,255,255,0]
++Rect	uint16[4]	O	Range is [0—8191]. Rect[0]: top left corner x coordinate (left) Rect[1]: top left corner y coordinate (top) Rect[2]: bottom right x coordinate (right) Rect[3]: bottom right y coordinate (bottom)	[147, 349, 2769, 784]

<code>++EncodeBlend</code>	bool	O	false - widget blend is not enabled.	true
----------------------------	------	---	--------------------------------------	------

Please refer to "4.2.1 Get and Set Configure" for configuration getting and setting. Specific examples are as follows :

Get Config Request Example

```
http://10.0.0.8/cgi-bin/configManager.cgi?action=getConfig&name=VideoWidget
```

Get Config Response Example

```
table.VideoWidget[0].ChannelTitle.BackColor[0]=0
table.VideoWidget[0].ChannelTitle.BackColor[1]=0
table.VideoWidget[0].ChannelTitle.BackColor[2]=0
table.VideoWidget[0].ChannelTitle.BackColor[3]=128
table.VideoWidget[0].ChannelTitle.EncodeBlend=true
table.VideoWidget[0].ChannelTitle.FrontColor[0]=255
table.VideoWidget[0].ChannelTitle.FrontColor[1]=255
table.VideoWidget[0].ChannelTitle.FrontColor[2]=255
table.VideoWidget[0].ChannelTitle.FrontColor[3]=0
table.VideoWidget[0].ChannelTitle.Rect[0]=0
table.VideoWidget[0].ChannelTitle.Rect[1]=8191
table.VideoWidget[0].ChannelTitle.Rect[2]=0
table.VideoWidget[0].ChannelTitle.Rect[3]=8191
...
table.VideoWidget[0].UserDefinedTitle[0].BackColor[0]=0
table.VideoWidget[0].UserDefinedTitle[0].BackColor[1]=0
table.VideoWidget[0].UserDefinedTitle[0].BackColor[2]=0
table.VideoWidget[0].UserDefinedTitle[0].BackColor[3]=128
table.VideoWidget[0].UserDefinedTitle[0].EncodeBlend=true
table.VideoWidget[0].UserDefinedTitle[0].FrontColor[0]=255
table.VideoWidget[0].UserDefinedTitle[0].FrontColor[1]=255
table.VideoWidget[0].UserDefinedTitle[0].FrontColor[2]=255
table.VideoWidget[0].UserDefinedTitle[0].FrontColor[3]=0
table.VideoWidget[0].UserDefinedTitle[0].Rect[0]=0
table.VideoWidget[0].UserDefinedTitle[0].Rect[1]=8191
table.VideoWidget[0].UserDefinedTitle[0].Rect[2]=0
table.VideoWidget[0].UserDefinedTitle[0].Rect[3]=8191
...
```

Set Config Request Example

```
http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&VideoWidget[1].Covers[0].BackColor[0]=128&VideoWidget[1].Covers[0].BackColor[1]=128&VideoWidget[1].Covers[0].BackColor[2]=128&VideoWidget[1].Covers[0].BackColor[3]=0
```

Set Config Response Example

```
OK
```

4.5.12 Get Video Input Capability

Request URL	<code>http://<server>/cgi-bin/devVideoInput.cgi?action=getCaps</code>
Method	GET

Request Params (key=value format at URL)

Name	Type	R/O	Description	Example
channel	int	R	video channel index which starts from 1.	1

Request Example

http://192.168.1.108/cgi-bin/devVideoInput.cgi?action=getCaps&channel=1

Response Params (key=value format in body)

Name	Type	R/O	Description	Example
caps	object	O	capability	
+AutoSyncPhase	bool	O	Does it support automatic phase adjustment	false
+Backlight	int	O	0: not support backlight ; 1: support backlight; 2: support regional regulation	0
+ChipID	char[]	O	ID of chips in this channel	0
+CoverCount	int	O	The maximum cover region count.	8
+CoverType	in	O	0: don't support cover 1: support realtime cover 2: support non-realtime cover	1
+CustomManualExposure	bool,	O	true: support use defined manual exposure time	false
+DayNightColor	bool	O	true: support color alternate between day and night.	true
+DownScaling	bool	O	true: support down scaling, binning mode not included.	true
+Exposure	int	O	Exposure grade. 0 — don't support exposure control.	10
+ExternalSyncInput	bool	O	true: support HD signal external synchronization.	true
+FlashAdjust	bool	O	true: support flash adjust	true
+Flip	bool	O	true: support picture flip.	true
+Gain	bool	O	true: support gain control.	true
+GainAuto	bool	O	true: support auto gain.	true
+HorizontalBinning	int	O	Horizontal pixel binning mask, 1 — support 2 pixel binning, 2 — support 3 pixel binning 4 — support 4 pixel binning ... 2^n — support $n+2$ pixel binning	1
+VerticalBinning	int	O	Vertical pixel binning mask, 1 — support 2 pixel binning, 2 — support 3 pixel binning 4 — support 4 pixel binning ... 2^n — support $n+2$ pixel binning	1
+InfraRed	bool	O	true: support Infra compensation	true

+Iris	bool	O	true: support Iris adjust	true
+IrisAuto	bool	O	true: support auto Iris adjust	true
+LadenBitrate	int	O	Unit is Kbps. Maximum value of video stream bitrates, 16bpp, not in binning mode.	
+LimitedAutoExposure	bool	O	true: support auto exposure with time limit.	true
+MaxExposureTime	double	O	upper limit of exposure time, unit: ms	
+MaxExposureTime1	double	O	upper limit of exposure time, unit: ms	
+MaxHeight	int	O	Maximum video height	
+MaxWidth	int	O	Maximum video width	
+MinExposureTime	double	O	lower limit of exposure time, unit: ms	
+MinExposureTime1	double	O	lower limit of exposure time, unit: ms	
+Mirror	bool	O	true: support picture mirror.	true
+NightOptions	bool	O	true: support night options.	true
+ReferenceLevel	bool	O	true: support reference level.	true
+Rotate90	bool	O	true: support clockwise/anticlockwise 90° rotate	true
+SetColor	bool	O	true: support color set.	true
+SignalFormats	char[]	O	<p>It's a string contains supported video input signal formats for this channel. Signal formats are separated by comma.</p> <p>Range is {Inside, BT656, 720p, 1080p, 1080i, 1080sF, 1_3M}</p> <p>Inside — inside input. 1_3M — 1280*960</p>	Inside
+SyncChipChannels	bool	O	True: channels in same chip should be synchronized. Synchronized means video resolution of these channels should be the same.	true
+TitleCount	int	O	Maximum count of blending titles.	
+UpScaling	bool	O	true: support up scaling.	false
+WhiteBalance	int	O	<p>Range is {0, 1, 2, 3}</p> <p>0: don't support white balance.</p> <p>1: support auto white balance</p> <p>2: support auto and pre defined white balance.</p> <p>3: support auto, pre defined and user defined white balance</p>	3

+VideoInIR	object	O	Infrared capability	
++SupportIRPlateMode	bool	O	Does it support plate mode adjustment in infrared mode	false
+Flicker	object	O	Flicker capability	
++Support	uint32	O	0: for not support; 1: for support	0
++MaxAmplitude	uint32	O	Max Amplitude	0
++MaxBiasing	uint32	O	Max Biasing	0
+KillShutLine	object	O		
++Support	bool	O		false
++MaxChannels	uint32	O	channel num	1
++MinKillShutLine	int32	O	the min	0
++MaxKillShutLine	int32	O	the max	0
++SensorNum	uint32	O	total sensor number on this channel	1

Response Example

```

caps.AutoSyncPhase=false
caps.Backlight=2
caps.BrightnessCompensation=true
caps.ChipID=0
caps.CoverCount=4
caps.CoverType=1
caps.CustomManualExposure=true
caps.DayNightColor=true
caps.DayNightColorIO=false
caps.DoubleExposure=0
caps.DownScaling=false
caps.EEModeRange=100
caps.ElectricFocus=false
caps.Exposure=16
caps.ExposureMode=31
caps.ExternalSyncInput=false
caps.FishEye=false
caps.FlashAdjust=false
caps.Flip=true
caps.FormatCount=5
caps.Gain=true
caps.GainAuto=true
caps.Gamma=true
caps.GammaModeRange=100
caps.GlareInhibition=1
caps.HorizontalBinning=0
caps.IRCUT=true
caps.ImageEnhancement.LevelRange[0]=0
caps.ImageEnhancement.LevelRange[1]=100
caps.ImageEnhancement.Support=true

```

```
caps.InfraRed=true
caps.Iris=true
caps.IrisAuto=true
caps.LadenBitrate=972000
caps.LimitedAutoExposure=true
caps.MaxExposureTime=300.0
caps.MaxExposureTime1=0.0
caps.MaxHeight=1080
caps.MaxMultiProfile=3
caps.MaxWidth=1920
caps.MeteringRegionCount=0
caps.MinExposureTime=1.0
caps.MinExposureTime1=0.0
caps.Mirror=true
caps.MultiOptions=false
caps.NightOptions=true
caps.ReferenceLevel=false
caps.Rotate90=true
caps.SetColor=true
caps.SignalFormats=Inside
caps.SignalType[0]=VGA
caps.SnapshotExposure=false
caps.SupportProfile=false
caps.SupportWhiteLevel=true
caps.SupportWriteLevel=false
caps.SyncChipChannels=false
caps.SyncFocus=false
caps.TitleCount=4
caps.TridimDenoise=2
caps.TridimDenoiseDetails=0
caps.UTC=0
caps.UpScaling=false
caps.Version=0
caps.VerticalBinning=0
caps.VideoInDenoise.2D.LevelRange[0]=0
caps.VideoInDenoise.2D.LevelRange[1]=100
caps.VideoInDenoise.2D.Support=true
caps.VideoInDenoise.3D.3DAutoType.ModRange[0]=0
caps.VideoInDenoise.3D.3DAutoType.ModRange[1]=100
caps.VideoInDenoise.3D.Support=true
caps.VideoInDenoise.Support=true
caps.WhiteBalance=3
caps.WideDynamicRange=1
caps.VideoInIR.SupportIRPlateMode=false
caps.Flicker.Support=0,
caps.Flicker.MaxAmplitude= 0,
caps.Flicker.MaxBiasing=0
```

HTTP API V3.35 - Intelbras

```

caps.KillShutLine.Support=false,
caps.KillShutLine.MaxChannels=1,
caps.KillShutLine.MinKillShutLine=0,
caps.KillShutLine.MaxKillShutLine=0

```

4.5.13 Get Coordinates of Current Window

Request URL	http://<server>/cgi-bin/devVideoInput.cgi?action=getCurrentWindow		
Method	GET		
Request Params (key=value format at URL)			
Name	Type	R/O	Description
channel	int	R	video channel index which starts from 1.
Request Example			
http://192.168.1.108/cgi-bin/devVideoInput.cgi?action=getCurrentWindow&channel=1			

Response Params (key=value format in body)			
Name	Type	R/O	Description
rect	int[4]	R	relative coordinates, range is 0-8192.{0,0,0,0} top-left, {8192,0,0,0} top-right, {0,8192,0,0} bottom-left, {8192,8192,0,0} bottom-right
Response Example			
rect[0]=500			
rect[1]=500			
rect[2]=5000			
rect[3]=5000			

4.5.14 Set Coordinates of Current Window

Request URL	http://<server>/cgi-bin/devVideoInput.cgi?action=setCurrentWindow		
Method	GET		
Request Params (key=value format at URL)			
Name	Type	R/O	Description
channel	int	R	video channel index which starts from 1.
+rect	int[]	O	Represents the coordinates of a rectangle, an array of 4 elements, with coordinates normalized to 0-8192 {0,0,0,0} top-left, {8192,0,0,0} top-right, {0,8192,0,0} bottom-left, {8192,8192,0,0} bottom-right
Request Example			
http://192.168.1.108/cgi-bin/devVideoInput.cgi?action=setCurrentWindow&channel=1&rect[0]=0&rect[1]			

```
=0&rect[2]=5000&rect[3]=5000
```

Response Params (OK in body)

Name	Type	R/O	Description	Example
Response Example				
OK				

4.5.15 [Config] Video Out

Config Data Params				
Name	Type	R/O	Description	Example
VideoOut	object[]	O	Array. One element for each channel, and the array index represents the channel number which starts from 0.	
+Margin	uint8[4]	O	Margin 0~100	[0, 0, 0, 0]
+Color	object	O	video out color	
++Brightness	uint8	O	Brightness 0~100	50
++Contrast	uint8	O	Contrast 0~100	50
++Saturation	uint8	O	Saturation 0~100	50
++Hue	uint8	O	Hue 0~100	50
+Mode	object	O	mode	
++Width	int	O	Resolution	800
++Height	int	O	Resolution	600
++BPP	int	O	Bits per pixel.	16
++Format	char[16]	O	The range is {"Auto", "TV", "VGA", "DVI", "HDMI", "SPOT", "SDI", "HDMI/VGA"}.	"VGA"
++RefreshRate	int	O	Refresh rate.	60

Please refer to "4.2.1 Get and Set Configure" for configuration getting and setting. Specific examples are as follows :

Get Config Request Example

```
http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=VideoOut
```

Get Config Response Example

```
table.VideoOut[0].Margin[0]=0
table.VideoOut[0].Margin[1]=0
table.VideoOut[0].Margin[2]=0
table.VideoOut[0].Margin[3]=0
table.VideoOut[0].Color.Brightness=50
table.VideoOut[0].Color.Contrast=50
table.VideoOut[0].Color.Satuation=50
table.VideoOut[0].Color.Hue=50
```

```

table.VideoOut[0].Mode.Width=800
table.VideoOut[0].Mode.Height=600
table.VideoOut[0].Mode.BPP=16
table.VideoOut[0].Mode.Format="Auto"
table.VideoOut[0].Mode.RefreshRate=60
...

```

Set Config Request Example

http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&table.VideoOut[1].Color.Brightness=50

Set Config Response Example

OK

4.5.16 [Config] Smart Encode

Config Data Params

Name	Type	R/O	Description	Example
SmartEncode	object[]	O	Smart Encode config. Array. One element for each channel, and the array index represents the channel number which starts from 0.	
+Enable	bool	O	Open or close the smart encode of the main stream.	true
+Extra	bool[]	O	Open or close the smart encode of the extra stream.	[true]

Please refer to "4.2.1 Get and Set Configure" for configuration getting and setting. Specific examples are as follows :

Get Config Request Example

http://10.0.0.8/cgi-bin/configManager.cgi?action=getConfig&name=SmartEncode

Get Config Response Example

```

table.SmartEncode[0].Enable=false
table.SmartEncode[0].Extra[0]=false
table.SmartEncode[0].Extra[1]=false
...

```

Set Config Request Example

http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&SmartEncode[0].Enable=true

Set Config Response Example

OK

4.5.17 Get Decoder Caps

Request URL	http://<server>/cgi-bin/DevVideoDec.cgi?action=getCaps
Method	GET
Request Params (key=value format at URL)	

Name	Type	R/O	Description	Example
Request Example				
http://192.168.1.108/cgi-bin/DevVideoDec.cgi?action=getCaps				

Response Params (key=value format in body)				
Name	Type	R/O	Description	Example
Video	object	R	The video caps	
+StreamType	char[]	R	The stream types that the device supports, ex : h264, h265 etc.	["h264", "h265"]
+LadenBitrate	int	R	The maximum decoding capability that the device supports (w * h * FPS * 16)/1024, uint: kbps, ex : 7680*4320*30*8/1024 = 7776000	7776000

Response Example
Video.StreamType[0]=h264
Video.StreamType[1]=h265
Video.LadenBitrate=7776000

4.5.18 [Config] PrivacyMasking

Config Data Params				
Name	Type	R/O	Description	Example
PrivacyMasking	object[]	O	PrivacyMasking config, each channel has several masking region	
+Name	char[64]	O	name	"Privacy Mask1"
+Enable	bool	O	Enable/Disable	false
+Postion	double[3]	O	Size is 3 Postion[0] is horizontal angle, range is [-1—1] Postion[1] is vertical angel, range is [-1—1] Postion[2] is zoom, range is [0—1]	[0.0, 0.0, 1.0]
+ShapType	char[16]	O	Shape type,can be: "Rect", "Polygon"	"Rect"
+Rect	Rect	O	It is valid if ShapeType is Rect , top left and bottom right point, each point has x and y value, oordinate remap to 0 — 8192.	Rect
+Polygon	Point[]	O	It is valid if ShapeType is Polygon the first array is point list, minimum item is 3, the second array is point, must be two int, means x and y value, coordinate remap to 0 — 8192.	[[0, 0], [128, 128]]
+Color	Color	O	Mask color, RGBA color, 4 integer range is 0-255	Color

+Mosaic	int	O	Mosaic type, Range is {0, 8, 16, 24, 32} 0: no mosaic 8: mosaic size 8*8 16: mosaic size 16*16 24: mosaic size 24*24 32: mosaic size 32*32	8
+ViewAngle	double	O	View angle, range is [0.0, 360.0], unit: degree	30.0

Please refer to "4.2.1 Get and Set Configure" for configuration getting and setting. Specific examples are as follows :

Get Config Request Example

```
http://10.0.0.8/cgi-bin/configManager.cgi?action=getConfig&name=PrivacyMasking
```

Get Config Response Example

```
table.PrivacyMasking[0][0].Enable=true
table.PrivacyMasking[0][0].Postion[0]=0.0
table.PrivacyMasking[0][0].Postion[1]=0.0
table.PrivacyMasking[0][0].Postion[2]=1.0
table.PrivacyMasking[0][0].ShapeType=Rect
table.PrivacyMasking[0][0].Rect[0]=0
table.PrivacyMasking[0][0].Rect[1]=0
table.PrivacyMasking[0][0].Rect[2]=50
table.PrivacyMasking[0][0].Rect[3]=100
table.PrivacyMasking[0][0].Color[0]=128
table.PrivacyMasking[0][0].Color[1]=128
table.PrivacyMasking[0][0].Color[2]=128
table.PrivacyMasking[0][0].Color[3]=255
table.PrivacyMasking[0][0].Mosaic=8
table.PrivacyMasking[0][0].ViewAngle=30.0
table.PrivacyMasking[0][1].Enable=true
...

```

Set Config Request Example

```
http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&PrivacyMasking[0][0].Enable=true&PrivacyMasking[0][0].Postion[0]=0.0&PrivacyMasking[0][0].Postion[1]=0.0&PrivacyMasking[0][0].Postion[2]=1.0&PrivacyMasking[0][0].ShapeType=Rect&PrivacyMasking[0][0].Rect[0]=0&PrivacyMasking[0][0].Rect[1]=0&PrivacyMasking[0][0].Rect[2]=50&PrivacyMasking[0][0].Rect[3]=100&PrivacyMasking[0][0].Mosaic=8&PrivacyMasking[0][0].ViewAngle=30.0
```

Set Config Response Example

```
OK
```

4.5.19 Get Privacy Masking

Request URL	http://<server>/cgi-bin/PrivacyMasking.cgi?action=getPrivacyMasking			
Method	GET			
Request Params (key=value format at URL)				
Name	Type	R/O	Description	Example

channel	int	R	video channel index which starts from 1	1
offset	int	R	Offset in the result record set, range is [0, Total – 1].	2
limit	int	R	Count of result to get.	5

Request Example

```
http://192.168.1.108/cgi-bin/PrivacyMasking.cgi?action=getPrivacyMasking&channel=1&offset=2&limit=5
```

Response Params (key=value format in body)

Name	Type	R/O	Description	Example
total	int	R	Count of result	5
PrivacyMasking	object[]	R	PrivacyMasking config	
+Index	int	R	Index of PrivacyMasking which starts from 0	0
+Name	char[]	O	PrivacyMasking name	"Privacy Masking1"
+Enable	int	R	1: enable; 0: disable	1
+ShapeType	char[]	O	Shape type, can be: "Rect", "Polygon"	"Rect"
+Rect	int[]	O	It is valid if ShapeType is Rect , top left and bottom right point, each point has x and y value, coordinate remap to 0 — 8192.	[0,0,50,100]
+Polygon	int[][]	O	It is valid if ShapeType is Polygon the first array is point list, minimum item is 3, the second array is point, must be two int, means x and y value, coordinate remap to 0 — 8192.	[[[],[],[]]]
+Color	int[]	O	Mask color, RGBA color, 4 integer range is 0-255	[]
+Mosaic	int	O	Mosaic type, Range is {0, 8, 16, 24, 32} 0: no mosaic 8: mosaic size 8*8 16: mosaic size 16*16 24: mosaic size 24*24 32: mosaic size 32*32	8

Response Example

```
total=5
PrivacyMasking[0].Index=0
PrivacyMasking[0].Name=Privacy Masking1
PrivacyMasking[0].Enable=1
PrivacyMasking[0].ShapeType=Polygon
PrivacyMasking[0].Polygon[0][0]= 0
PrivacyMasking[0].Polygon[0][1]=0
PrivacyMasking[0].Polygon[1][0]=128
```

```

PrivacyMasking[0].Polygon[1][1]=128
PrivacyMasking[0].Polygon[2][0]=256
PrivacyMasking[0].Polygon[2][1]=200
...
PrivacyMasking[0].Color[0]=128,
PrivacyMasking[0].Color[1]=128,
PrivacyMasking[0].Color[2]=128,
PrivacyMasking[0].Color[3]=255,
PrivacyMasking[0].Mosaic=8,
PrivacyMasking[1].Index =1,
PrivacyMasking[1].Enable=1,
...

```

4.5.20 Set Privacy Masking

Request URL	http://<server>/cgi-bin/PrivacyMasking.cgi?action=gotoPrivacyMasking		
Method	GET		
Request Params (key=value format at URL)			
Name	Type	R/O	Description
channel	int	R	video channel index which starts from 1
PrivacyMasking	object	R	Privacy masking config
+Index	int	R	Index of PrivacyMasking which starts from 0
+Name	char[]	O	PrivacyMasking name
+Enable	int	R	1: enable; 0: disable
+ShapeType	char[]	R	Shape type,can be: "Rect", "Polygon"
+Rect	int[]	O	It is valid if ShapeType is Rect , top left and bottom right point, each point has x and y value, coordinate remap to 0 — 8192. mandatory when ShapeType is "Rect"
+Polygon	int[][]	O	It is valid if ShapeType is Polygon the first array is point list, minimum item is 3, the second array is point, must be two int, means x and y value, coordinate remap to 0 — 8192. mandatory when ShapeType is "Polygon"
+Color	int[]	O	Mask color, RGBA color, 4 integer range is 0-255. mandatory when Mosaic is 0
+Mosaic	int	R	Mosaic type, Range is {0, 8, 16, 8}

		24, 32} 0: no mosaic 8: mosaic size 8*8 16: mosaic size 16*16 24: mosaic size 24*24 32: mosaic size 32*32	
--	--	--	--

Request Example

```
http://192.168.1.108/cgi-bin/PrivacyMasking.cgi?action=setPrivacyMasking&channel=2&PrivacyMasking.g.Index=3&PrivacyMasking.Name=Privacy%20Masking1&PrivacyMasking.Enable=1&PrivacyMasking.ShapeType=Rect&PrivacyMasking.Rect[0]=0&PrivacyMasking.Rect[1]=0&PrivacyMasking.Rect[2]=50&PrivacyMasking.Rect[3]=100&PrivacyMasking.Mosaic=8
```

Response Params (OK in body)

Name	Type	R/O	Description	Example
Response Example				
OK				

4.5.21 Goto Privacy Masking

Request URL	http://<server>/cgi-bin/PrivacyMasking.cgi?action=gotoPrivacyMasking			
Method	GET			
Request Params (key=value format at URL)				
Name	Type	R/O	Description	Example
channel	int	R	video channel index which starts from 1	2
index	int	R	Index of PrivacyMasking which starts from 0	3
Request Example				
http://192.168.1.108/cgi-bin/PrivacyMasking.cgi?action=gotoPrivacyMasking&channel=2&index=3				

Response Params (OK in body)

Name	Type	R/O	Description	Example
Response Example				
OK				

4.5.22 Delete Privacy Masking

Request URL	http://<server>/cgi-bin/PrivacyMasking.cgi?action=deletePrivacyMasking			
Method	GET			
Request Params (key=value format at URL)				
Name	Type	R/O	Description	Example
channel	int	R	video channel index which starts from 1	2
index	int	R	Index of PrivacyMasking which	3

		starts from 0	
Request Example			
http://192.168.1.108/cgi-bin/PrivacyMasking.cgi?action=deletePrivacyMasking&channel=2&index=3			

Response Params (OK in body)				
Name	Type	R/O	Description	Example
Response Example				
OK				

4.5.23 Clear Privacy Masking

Request URL	http://<server>/cgi-bin/PrivacyMasking.cgi?action=clearPrivacyMasking			
Method	GET			
Request Params (key=value format at URL)				
Name	Type	R/O	Description	Example
channel	int	R	video channel index which starts from 1	2
Request Example				http://192.168.1.108/cgi-bin/PrivacyMasking.cgi?action=clearPrivacyMasking&channel=2

Response Params (OK in body)	
Name	Type
Response Example	
OK	

4.5.24 Get Privacy Masking Rect

Request URL	http://<server>/cgi-bin/PrivacyMasking.cgi?action=getRealRect			
Method	GET			
Request Params (key=value format at URL)				
Name	Type	R/O	Description	Example
channel	int	R	video channel index which starts from 1	2
index	int	R	Index of PrivacyMasking which starts from 0	
Request Example				http://192.168.1.108/cgi-bin/PrivacyMasking.cgi?action=getRealRect&channel=2&index=1

Response Params (key=value format in body)	
Name	Type
Rect	int[4]
	R

Rect, top left and bottom right point, each point has x and y value, oordinate remap to 0 — 8192.

[0,0,50,100]

4.5.25 [Config] Motion Detection Settings

Config Data Params				
Name	Type	R/O	Description	Example
MotionDetect	object[]	O	Array. One element for each channel, and the array index represents the channel number which starts from 0.	
+Enable	bool	O	Enable/Disable motion detect feature in a channel.	true
+PtzManualEnable	bool	O	Enable/Disable manual control Ptz, only for Hahua devices	true
+OsdTwinkleEnable	bool	O	Osd Twinkle Enable default : false	true
+Row	int	O	motion detect row number	18
+Col	int	O	motion detect colum number	22
+Level	uint8	O	sensitive ,range 1-6	3
+TimeDivide	object[]	O		
++TimeSchedule	TimeSchedule	O	TimeSchedule	
++Level	uint8	O	sensitive in timesection, different with global sensitive.	4
+DetectVersion	char[32]	O	detect version	"V1.0"
+PirMotionEnable	bool	O	Customized Pir Motion Enable	false
+PirMotionLevel	uint8	O	Customized Pir Motion sensitive level	3
+VolumeRatio	uint8	O	Area occupancy ratio	50
+SubRatio	uint8	O	SubRatio	50
+Region	int[]	O	Region	[4194303, 3216384, ...]
+EventHandler	EventHandler	O	Setting of EventHandler is described in SetEventHandler.	
+MotionDetectWindow	object[4]	O	detect windows	
++Threshold	uint8	O	Range is [0—100]. It presents the threshold value when trigger motion detect.	50
++Sensitive	uint8	O	Range is [0—100]. Sensitivity of motion detection. It presents more sensitive if the value is larger.	50

++Region	int[]	O	<p>It is similar with head.Region [LineNum].</p> <p>Currently, a region is divided into 18 lines and 22 blocks per line.</p> <p>A bit describes a block in the line.</p> <p>Bit = 1: motion in this block is monitored.</p> <p>Example:</p> <p>MotionDetect [0].Region [0] = 4194303 (0x3FFFFF): the 22 blocks in channel 0 line 0 is monitored.</p> <p>MotionDetect [0].Region [1] = 0: the 22 blocks in channel 0 line 1 is not monitored.</p> <p>MotionDetect [0].Region [17] = 3: the left two blocks in the last line of channel 0 is monitored.</p>	[4194303, 3145728, ...]
++Id	int	O	It is the Id of a detect window.	33
++Name	char[256]	O	It is the name of a detect window.	"Region1"

Please refer to "4.2.1 Get and Set Configure" for configuration getting and setting. Specific examples are as follows :

Get Config Request Example

```
http://10.0.0.8/cgi-bin/configManager.cgi?action=getConfig&name=MotionDetect
```

Get Config Response Example

```
table.MotionDetect[0].Enable=false
table.MotionDetect[0].EventHandler.AlarmOut=1
table.MotionDetect[0].EventHandler.AlarmOutChannels[0]=0
table.MotionDetect[0].EventHandler.AlarmOutEnable=true
table.MotionDetect[0].EventHandler.AlarmOutLatch=10
table.MotionDetect[0].EventHandler.BeepEnable=false
table.MotionDetect[0].EventHandler.Dejitter=5
table.MotionDetect[0].EventHandler.Delay=0
table.MotionDetect[0].EventHandler.ExAlarmOut=1
table.MotionDetect[0].EventHandler.ExAlarmOutChannels[0]=0
table.MotionDetect[0].EventHandler.ExAlarmOutEnable=false
table.MotionDetect[0].EventHandler.FlashEnable=false
table.MotionDetect[0].EventHandler.FlashLatch=10
table.MotionDetect[0].EventHandler.LogEnable=true
table.MotionDetect[0].EventHandler.MailEnable=false
table.MotionDetect[0].EventHandler.Matrix=1
table.MotionDetect[0].EventHandler.MatrixChannels[0]=0
table.MotionDetect[0].EventHandler.MatrixEnable=false
table.MotionDetect[0].EventHandler.MessageEnable=false
table.MotionDetect[0].EventHandler.PtzLink[0][0]=None
table.MotionDetect[0].EventHandler.PtzLink[0][1]=1
table.MotionDetect[0].EventHandler.PtzLinkEnable=false
```



```
table.MotionDetect[0].EventHandler.TimeSection[6][4]=0 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.TimeSection[6][5]=0 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.TipEnable=false
table.MotionDetect[0].EventHandler.Tour=1
table.MotionDetect[0].EventHandler.TourChannels[0]=0
table.MotionDetect[0].EventHandler.TourEnable=false
table.MotionDetect[0].EventHandler.Voice.AudioFileName=
table.MotionDetect[0].EventHandler.VoiceEnable=false
table.MotionDetect[0].MotionDetectWindow[0].Id=0
table.MotionDetect[0].MotionDetectWindow[0].Name=Region1
table.MotionDetect[0].MotionDetectWindow[0].Region[0]=4194303
table.MotionDetect[0].MotionDetectWindow[0].Region[1]=4194303
table.MotionDetect[0].MotionDetectWindow[0].Region[2]=4194303
table.MotionDetect[0].MotionDetectWindow[0].Region[3]=4194303
table.MotionDetect[0].MotionDetectWindow[0].Region[4]=4194303
table.MotionDetect[0].MotionDetectWindow[0].Region[5]=4194303
table.MotionDetect[0].MotionDetectWindow[0].Region[6]=4194303
table.MotionDetect[0].MotionDetectWindow[0].Region[7]=4194303
table.MotionDetect[0].MotionDetectWindow[0].Region[8]=4194303
table.MotionDetect[0].MotionDetectWindow[0].Region[9]=4194303
table.MotionDetect[0].MotionDetectWindow[0].Region[10]=4194303
table.MotionDetect[0].MotionDetectWindow[0].Region[11]=4194303
table.MotionDetect[0].MotionDetectWindow[0].Region[12]=4194303
table.MotionDetect[0].MotionDetectWindow[0].Region[13]=4194303
table.MotionDetect[0].MotionDetectWindow[0].Region[14]=4194303
table.MotionDetect[0].MotionDetectWindow[0].Region[15]=4194303
table.MotionDetect[0].MotionDetectWindow[0].Region[16]=4194303
table.MotionDetect[0].MotionDetectWindow[0].Region[17]=4194303
table.MotionDetect[0].MotionDetectWindow[0].Sensitive=60
table.MotionDetect[0].MotionDetectWindow[0].Threshold=5
table.MotionDetect[0].MotionDetectWindow[0].Window[0]=0
table.MotionDetect[0].MotionDetectWindow[0].Window[1]=0
table.MotionDetect[0].MotionDetectWindow[0].Window[2]=8191
table.MotionDetect[0].MotionDetectWindow[0].Window[3]=8191
table.MotionDetect[0].MotionDetectWindow[1].Id=1
table.MotionDetect[0].MotionDetectWindow[1].Name=Region2
table.MotionDetect[0].MotionDetectWindow[1].Region[0]=0
table.MotionDetect[0].MotionDetectWindow[1].Region[1]=0
table.MotionDetect[0].MotionDetectWindow[1].Region[2]=0
table.MotionDetect[0].MotionDetectWindow[1].Region[3]=0
table.MotionDetect[0].MotionDetectWindow[1].Region[4]=0
table.MotionDetect[0].MotionDetectWindow[1].Region[5]=0
table.MotionDetect[0].MotionDetectWindow[1].Region[6]=0
table.MotionDetect[0].MotionDetectWindow[1].Region[7]=0
table.MotionDetect[0].MotionDetectWindow[1].Region[8]=0
table.MotionDetect[0].MotionDetectWindow[1].Region[9]=0
table.MotionDetect[0].MotionDetectWindow[1].Region[10]=0
```

```
table.MotionDetect[0].MotionDetectWindow[1].Region[11]=0
table.MotionDetect[0].MotionDetectWindow[1].Region[12]=0
table.MotionDetect[0].MotionDetectWindow[1].Region[13]=0
table.MotionDetect[0].MotionDetectWindow[1].Region[14]=0
table.MotionDetect[0].MotionDetectWindow[1].Region[15]=0
table.MotionDetect[0].MotionDetectWindow[1].Region[16]=0
table.MotionDetect[0].MotionDetectWindow[1].Region[17]=0
table.MotionDetect[0].MotionDetectWindow[1].Sensitive=60
table.MotionDetect[0].MotionDetectWindow[1].Threshold=5
table.MotionDetect[0].MotionDetectWindow[1].Window[0]=0
table.MotionDetect[0].MotionDetectWindow[1].Window[1]=0
table.MotionDetect[0].MotionDetectWindow[1].Window[2]=0
table.MotionDetect[0].MotionDetectWindow[1].Window[3]=0
table.MotionDetect[0].MotionDetectWindow[2].Id=2
table.MotionDetect[0].MotionDetectWindow[2].Name=Region3
table.MotionDetect[0].MotionDetectWindow[2].Region[0]=0
table.MotionDetect[0].MotionDetectWindow[2].Region[1]=0
table.MotionDetect[0].MotionDetectWindow[2].Region[2]=0
table.MotionDetect[0].MotionDetectWindow[2].Region[3]=0
table.MotionDetect[0].MotionDetectWindow[2].Region[4]=0
table.MotionDetect[0].MotionDetectWindow[2].Region[5]=0
table.MotionDetect[0].MotionDetectWindow[2].Region[6]=0
table.MotionDetect[0].MotionDetectWindow[2].Region[7]=0
table.MotionDetect[0].MotionDetectWindow[2].Region[8]=0
table.MotionDetect[0].MotionDetectWindow[2].Region[9]=0
table.MotionDetect[0].MotionDetectWindow[2].Region[10]=0
table.MotionDetect[0].MotionDetectWindow[2].Region[11]=0
table.MotionDetect[0].MotionDetectWindow[2].Region[12]=0
table.MotionDetect[0].MotionDetectWindow[2].Region[13]=0
table.MotionDetect[0].MotionDetectWindow[2].Region[14]=0
table.MotionDetect[0].MotionDetectWindow[2].Region[15]=0
table.MotionDetect[0].MotionDetectWindow[2].Region[16]=0
table.MotionDetect[0].MotionDetectWindow[2].Region[17]=0
table.MotionDetect[0].MotionDetectWindow[2].Sensitive=60
table.MotionDetect[0].MotionDetectWindow[2].Threshold=5
table.MotionDetect[0].MotionDetectWindow[2].Window[0]=0
table.MotionDetect[0].MotionDetectWindow[2].Window[1]=0
table.MotionDetect[0].MotionDetectWindow[2].Window[2]=0
table.MotionDetect[0].MotionDetectWindow[2].Window[3]=0
table.MotionDetect[0].MotionDetectWindow[3].Id=3
table.MotionDetect[0].MotionDetectWindow[3].Name=Region4
table.MotionDetect[0].MotionDetectWindow[3].Region[0]=0
table.MotionDetect[0].MotionDetectWindow[3].Region[1]=0
table.MotionDetect[0].MotionDetectWindow[3].Region[2]=0
table.MotionDetect[0].MotionDetectWindow[3].Region[3]=0
table.MotionDetect[0].MotionDetectWindow[3].Region[4]=0
table.MotionDetect[0].MotionDetectWindow[3].Region[5]=0
```

```

table.MotionDetect[0].MotionDetectWindow[3].Region[6]=0
table.MotionDetect[0].MotionDetectWindow[3].Region[7]=0
table.MotionDetect[0].MotionDetectWindow[3].Region[8]=0
table.MotionDetect[0].MotionDetectWindow[3].Region[9]=0
table.MotionDetect[0].MotionDetectWindow[3].Region[10]=0
table.MotionDetect[0].MotionDetectWindow[3].Region[11]=0
table.MotionDetect[0].MotionDetectWindow[3].Region[12]=0
table.MotionDetect[0].MotionDetectWindow[3].Region[13]=0
table.MotionDetect[0].MotionDetectWindow[3].Region[14]=0
table.MotionDetect[0].MotionDetectWindow[3].Region[15]=0
table.MotionDetect[0].MotionDetectWindow[3].Region[16]=0
table.MotionDetect[0].MotionDetectWindow[3].Region[17]=0
table.MotionDetect[0].MotionDetectWindow[3].Sensitive=60
table.MotionDetect[0].MotionDetectWindow[3].Threshold=5
table.MotionDetect[0].MotionDetectWindow[3].Window[0]=0
table.MotionDetect[0].MotionDetectWindow[3].Window[1]=0
table.MotionDetect[0].MotionDetectWindow[3].Window[2]=0
table.MotionDetect[0].MotionDetectWindow[3].Window[3]=0
table.MotionDetect[0].OsdTwinkleEnable=false
table.MotionDetect[0].PirMotionLevel=3

```

Set Config Request Example

[http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&MotionDetect\[0\].MotionDetectWindow\[0\].Region\[0\]=1&MotionDetect\[0\].MotionDetectWindow\[0\].Region\[1\]=1&MotionDetect\[0\].MotionDetectWindow\[0\].Region\[2\]=1&MotionDetect\[0\].MotionDetectWindow\[0\].Region\[3\]=1&MotionDetect\[0\].MotionDetectWindow\[0\].Region\[4\]=1&MotionDetect\[0\].DetectVersion=V3.0](http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&MotionDetect[0].MotionDetectWindow[0].Region[0]=1&MotionDetect[0].MotionDetectWindow[0].Region[1]=1&MotionDetect[0].MotionDetectWindow[0].Region[2]=1&MotionDetect[0].MotionDetectWindow[0].Region[3]=1&MotionDetect[0].MotionDetectWindow[0].Region[4]=1&MotionDetect[0].DetectVersion=V3.0)

Set Config Response Example

OK

4.5.26 [Config] LAEConfig

Config Data Params

Name	Type	R/O	Description	Example
LAEConfig	object	O		
+Enable	bool	O	turn on/off LAE fuction The following parameters will take effects when it's on	true
+ManulEn	uint32	O	whether use manual mode, 1 for manual mode, 0 for alternative plan	1
+YSIGMA	uint32	O	absolute strength of Y channel Grinding, The smaller the stronger(fuzzier)	1
+Dethr	uint32	O	relative strength of Y channel Grinding, The smaller the stronger(fuzzier)	1
+STS	uint32	O	Y channel STS	1

+GauFilter0	uint32	O	Y channel pre filter0, meet the conditions: GauFilter0 * 2 + GauFilter1 * 2 + GauFilter2 = 256	1
+GauFilter1	uint32	O	Y channel pre filter1	1
+GauFilter2	uint32	O	Y channel pre filter2	1
+UVGauFilter0	uint32	O	UV channel pre filter0, meet the conditions: GauFilter0 * 2 + GauFilter1 * 2 + GauFilter2 = 256	1
+UVGauFilter1	uint32	O	UV channel pre filter1	1
+UVGauFilter2	uint32	O	UV channel pre filter2	1
+UVSigma	uint32	O	absolute strength of UV channel Grinding, The smaller the stronger	1
+EEGauFilter0	uint32	O	EE pre filter0, meet the conditions: GauFilter0 * 2 + GauFilter1 * 2 + GauFilter2 = 256	1
+EEGauFilter1	uint32	O	EE pre filter1, extracts low-frequenc	1
+EEGauFilter2	uint32	O	EE pre filter2	1
+EERatio	uint32	O	strength of enhancement, the bigger the stronger	1
+GEnable	uint32	O	Brightening switch, default false	1
+GainLevel	uint32	O	it is used to choose paramas, as an alternative	1
+DebugPin	uint32	O	debug option	1

Please refer to "4.2.1 Get and Set Configure" for configuration getting and setting. Specific examples are as follows :

Get Config Request Example

<http://10.0.0.8/cgi-bin/configManager.cgi?action=getConfig&name=LAEConfig>

Get Config Response Example

```
table.LAEConfig.Enable=true
table.LAEConfig.ManulEn=1
table.LAEConfig.YSIGMA =1
table.LAEConfig.Dethr =1
table.LAEConfig.STS =1
table.LAEConfig.GauFilter0=1
table.LAEConfig.GauFilter1=1
table.LAEConfig.GauFilter2=1
table.LAEConfig.UVGauFilter0=1
table.LAEConfig.UVGauFilter1=1
table.LAEConfig.UVGauFilter2=1
table.LAEConfig.UVSigma =1
table.LAEConfig.EEGauFilter0=1
table.LAEConfig.EEGauFilter1=1
table.LAEConfig.EEGauFilter2=1
table.LAEConfig.EEratio =1
table.LAEConfig.GEnable =1
```

table.LAEConfig.GainLevel =1
table.LAEConfig.DebugPin =1

Set Config Request Example

http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&LAEConfig.Enable=true&LAEConfig.ManulEn=1

Set Config Response Example

OK

4.5.27 Enable/Disable all privacy masking covers

Request URL	http://<server>/cgi-bin/PrivacyMasking.cgi?action=setPrivacyMaskingEnable		
Method	GET		
Request Params (key=value format at URL)			
Name	Type	R/O	Description
channel	int	R	video channel index which starts from 1
Enable	bool	R	true: Enable, false: Disable
Request Example			
http://192.168.1.108/cgi-bin/PrivacyMasking.cgi?action=setPrivacyMaskingEnable&channel=2&Enable=true			

Response Params (OK in body)
Name
Type
R/O
Description
Example
Response Example
OK

4.5.28 Get enable/disable state of all privacy masking covers

Request URL	http://<server>/cgi-bin/PrivacyMasking.cgi?action=getPrivacyMaskingEnable		
Method	GET		
Request Params (key=value format at URL)			
Name	Type	R/O	Description
channel	int	R	video channel index which starts from 1
Request Example			
http://192.168.1.108/cgi-bin/PrivacyMasking.cgi?action=getPrivacyMaskingEnable&channel=2			

Response Params (key=value format in body)
Name
Type
R/O
Description
Example
Enable
bool
R
true: Enable, false: Disable
false
Response Example
Enable=false

4.6 System

4.6.1 [Config] General

Config Data Params				
Name	Type	R/O	Description	Example
General	object	O	general config	
+LocalNo	uint	O	Local number for remote controller	8
+MachineID	char[]	O	Device ID	"20832748927"
+MachineName	char[]	O	Device name or serial number.	"Dahua001"
+MachineAddress	char[]	O	Address machine places in	"XXXDistrict.YYYRoad"
+MachineGroup	char[]	O	Group machine belongs to	"xxxx"
+LockLoginEnable	bool	O	Enable login failure attempts	true
+CheckDuration	uint32	O	Zeroing cycle, if the number of login attempts within the specified time does not exceed the number of attempts, the number of attempts will be reset to zero Unit seconds 0 for never	30
+LockLoginTimes	int	O	Number of attempts to log in failed	3
+LoginFailLockTime	int	O	Login failure lock time Unit seconds	1800
+MaxOnlineTime	int	O	Maximum user online time Unit seconds	1800
+LocalPolicy	object	O	Local GUI locking policy	
++LockLoginEnable	bool	O	Enable login failure attempts	true
++CheckDuration	uint32	O	Zeroing cycle 0 for never	30
++LockLoginTimes	int	O	Number of attempts to log in failed	3
++LoginFailLockTime	int	O	Login failure lock time Unit seconds	1800
+ActivationTime	char[24]	O	Date when the device started using, date format: "yyyy-MM-dd HH:mm:ss" range "2000-01-01 00:00:00", "2099-01-01 00:00:00"	"2000-01-01 00:00:00"

Please refer to "4.2.1 Get and Set Configure" for configuration getting and setting. Specific examples are as follows :

Get Config Request Example

`http://10.0.0.8/cgi-bin/configManager.cgi?action=getConfig&name=General`

Get Config Response Example

```

table.General.LocalNo=8,
table.General.MachineID=20832748927,
table.General.MachineName=Device001,
table.General.MachineAddress=XXXDistrict.YYYRoad,
table.General.MachineGroup=xxxx,
table.General.LockLoginEnable=true,
table.General.CheckDuration=30,
table.General.LockLoginTimes=3,
table.General.LoginFailLockTime=1800,
table.General.MaxOnlineTime=1800,
table.General.LocalPolicy.LockLoginEnable=true,
table.General.LocalPolicy.CheckDuration=30,
table.General.LocalPolicy.LockLoginTimes=3,
table.General.LocalPolicy.LoginFailLockTime=1800
table.General.ActivationTime=2000-01-01 00:00:00

```

Set Config Request Example

http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&General.MachineName=MyIPC

Set Config Response Example

OK

4.6.2 Get Current Time

Request URL	http://<server>/cgi-bin/global.cgi?action=getCurrentTime		
Method	GET		
Request Params (none)			
Name	Type	R/O	Description
Request Example			
http://192.168.1.108/cgi-bin/global.cgi?action=getCurrentTime			

Response Params (key=value format in body)			
Name	Type	R/O	Description
result	char[20]	O	The time format is "Y-M-D H-m-S". It's not be effected by Locales. TimeFormat in SetLocalesConfig.
Response Example			
result=2011-7-3 21:02:32			

4.6.3 Set Current Time

Request URL	http://<server>/cgi-bin/global.cgi?action=setCurrentTime		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
time	char[20]	R	The time format is "Y-M-D H-m-S". It's

		not be effected by Locales. TimeFormat in SetLocalesConfig.	2
Request Example			
http://192.168.1.108/cgi-bin/global.cgi?action=setCurrentTime&time=2016-01-01%2021:02:32			

Response Params (OK in body)

Response Example

OK

4.6.4 [Config] Locales

Config Data Params				
Name	Type	R/O	Description	Example
Locales	object	O	locales	
+TimeFormat	char[24]	O	<p>Defines time format displayed in video time title.</p> <p>String form is: <i>year-month-day hour:mm:ss</i>.</p> <p>Position of <i>year</i>, <i>month</i> and <i>day</i> can be exchanged.</p> <p>Range of <i>year</i> is {yy, yyyy}</p> <p>yy = year without century, yyyy = year with century.</p> <p>Range of <i>month</i> is {M, MM, MMMM}</p> <p>M = 1 for January, MM = 01 for January, MMMM = Jan for January</p> <p>Range of <i>day</i> is {d, dd}</p> <p>d = 1 for first day, dd = 01 for first day</p> <p>Range of <i>hour</i> is {H, HH, h, hh}</p> <p>H = 1 for 1:00, HH = 01 for 1:00, range is 0-23</p> <p>h = 1 for 1:00, hh = 01 for 1:00, time range is 1-12</p> <p>Example:</p> <p>yyyy-MM-dd HH:mm:ss or MM-dd-yyyy HH:mm:ss or dd-M-yy hh:mm:ss</p>	"yyyy-MM-dd HH:mm:ss"
+DSTEnable	bool	O	Enable/Disable DST (daylight saving time)	false
+DSTStart	object	O	DST start time	
++Year	int	O	Range is [2000-2038]	0
++Month	int	O	Range is [1—12]	1
++Week	int	O	Range is {1, 2, 3, 4, -1, 0}. 0 = Use month day	1

			[1, 2, 3, 4, -1]: use week day. 1 = first week, 2 = second, 3 = third, 4 = fourth, -1 = last.	
++Day	int	O	Range is [0—6] or [1—31] [0—6]: week day, 0 = Sunday, 6 = Saturday [1—31]: month day If Locales.DSTEnd.Week is 0, use month day, otherwise, use week day.	0
++Hour	int	O	Range is [0—23]	0
++Minute	int	O	Range is [0—59]	0
+DSTEnd	object	O	DST end time	
++Year	int	O	Range is [2000-2038]	0
++Month	int	O	Range is [1—12]	1
++Week	int	O	Range is {1, 2, 3, 4, -1, 0}. 0 = Use month day [1, 2, 3, 4, -1]: use week day. 1 = first week, 2 = second, 3 = third, 4 = fourth, -1 = last.	1
++Day	int	O	Range is [0—6] or [1—31] [0—6]: week day, 0 = Sunday, 6 = Saturday [1—31]: month day If Locales.DSTEnd.Week is 0, use month day, otherwise, use week day.	0
++Hour	int	O	Range is [0—23]	0
++Minute	int	O	Range is [0—59]	0

Please refer to "4.2.1 Get and Set Configure" for configuration getting and setting. Specific examples are as follows :

Get Config Request Example

```
http://10.0.0.8/cgi-bin/configManager.cgi?action=getConfig&name=Locales
```

Get Config Response Example

```
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
```

Set Config Request Example

```
http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&Locales.DSTEnable=false
```

Set Config Response Example

```
OK
```

4.6.5 [Config] Holiday Management

Config Data Params

Name	Type	R/O	Description	Example
Holiday	object	O	holiday configuration	
+MonthMask	int[12]	O	Array index presents the index of a month. 0 presents January, 1 presents February, 11 presents December. It is the mask of a month. Every bit present a day. For example, 0x0001 presents the first day of a month is holiday. 0x0002 presents the second day of a month is holiday, 0x0003 presents the first day and second day of a month is holiday.	[3,0,0,0,0,0,0,0, 1610612739,0,0]

Please refer to "4.2.1 Get and Set Configure" for configuration getting and setting. Specific examples are as follows :

Get Config Request Example

```
http://10.0.0.8/cgi-bin/configManager.cgi?action=getConfig&name=Holiday
```

Get Config Response Example

```
table.Holiday.MonthMask[0]=3
table.Holiday.MonthMask[1]=0
table.Holiday.MonthMask[2]=0
table.Holiday.MonthMask[3]=0
table.Holiday.MonthMask[4]=0
table.Holiday.MonthMask[5]=0
table.Holiday.MonthMask[6]=0
table.Holiday.MonthMask[7]=0
table.Holiday.MonthMask[8]=0
table.Holiday.MonthMask[9]=1610612739
table.Holiday.MonthMask[10]=0
table.Holiday.MonthMask[11]=0
```

Set Config Request Example

```
http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&Holiday.MonthMask[0]=3
```

Set Config Response Example

```
OK
```

4.6.6 Get Language Capability

Request URL	http://<server>/cgi-bin/magicBox.cgi?action=getLanguageCaps			
Method	GET			
Request Params (none)				
Name	Type	R/O	Description	Example
Request Example				http://192.168.1.108/cgi-bin/magicBox.cgi?action=getLanguageCaps

Response Params (key=value format in body)				
Name	Type	R/O	Description	Example
Languages	char[]	R	response is a string contains languages with comma separated. Languages include { English, SimpChinese, TradChinese, Italian, Spanish, Japanese, Russian, French, German }	SimpChinese,English,French
Response Example			Languages=SimpChinese,English,French	

4.6.7 [Config] Language

NOTE: After changing language setting, system will automatically reboot!

Config Data Params				
Name	Type	R/O	Description	Example
Language	char[12]	O	The language range is get from interface in GetLanguageCaps	"English"

Please refer to "4.2.1 Get and Set Configure" for configuration getting and setting. Specific examples are as follows :

Get Config Request Example
http://10.0.0.8/cgi-bin/configManager.cgi?action=getConfig&name=Language
Get Config Response Example
http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&Language=SimpChinese

Set Config Request Example
http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&Language=SimpChinese
Set Config Response Example
OK

4.6.8 Get Device Type

Get the device type displayed (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				

4.6.9 Get Hardware Version

Get the device hardware version information.

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 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				

4.6.10 Get Serial Number of Device

Get the serial number of the device.

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	The device serial number	YZC0GZ05100020
Response Example				
sn=YZC0GZ05100020				

4.6.11 Get Machine Name

Get the device machine 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
name
Description
O
The device name
Sample
my machine
Response Example
name=my machine

4.6.12 Get System Information

Get the system information of the device.

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
info
Type
object
R/O
O
Description
System information
+SyncTime
Type
object
R/O
O
Description
For devices without RTC or when the RTC of the device is out of battery, you might also need to synchronize the time.
++Strategy
Type
enumchar[16]
R/O
R
Description
Time synchronization strategy enumchar[16]{ "None" "PerLogin": Synchronizes the time each time you log in }
PerLogin
+2DCode
Type
char[32]
R/O
O
Description
security code 0-9 and capitalized A-Z
123456
+TotalRunTime
Type
uint64
R/O
O
Description
The total running time of the device. Unit: s
12456
+cameraNum
Type
uint8
R/O
O
Description
The number of cameras
1
+cardReader
Type
bool
R/O
O
Description
Whether card swiping is supported: true: supported false: not supported
true
+flashID
Type
uint8[8]
R/O
O
Description
flash ID
[200,209,128,149,64,1

				27,127,200]
+hasRTC	bool	O	Whether the RTC chip is included (for recording the system time). If the filed does not exist, the value is true and the RTC is included by default. RTC included: true No RTC: false	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
```

4.6.13 Get Vendor Information

Request URL	http://<server>/cgi-bin/magicBox.cgi?action=getVendor			
Method	GET			
Request Params (none)				
Name	Type	R/O	Description	Example
Request Example				
http://192.168.1.108/cgi-bin/magicBox.cgi?action=getVendor				

Response Params (key=value format in body)

Name	Type	R/O	Description	Example
vendor	char[32]	O	OEM	TTT

Response Example

```
vendor=TTT
```

4.6.14 Get Software Information

Request URL	http://<server>/cgi-bin/magicBox.cgi?action=getSoftwareVersion			
Method	GET			
Request Params (none)				
Name	Type	R/O	Description	Example

Request Example

http://192.168.1.108/cgi-bin/magicBox.cgi?action=getSoftwareVersion

Response Params (key=value format in body)

Name	Type	R/O	Description	Example
version	char[64]	O	software version	2.212.0000.0.R,build:2013-11-14

Response Example

version=2.212.0000.0.R,build:2013-11-14

4.6.15 Get Version of Onvif

Request URL	http://<server>/cgi-bin/IntervideoManager.cgi?action=getVersion
--------------------	---

Method	GET
---------------	-----

Request Params (key=value format in url)

Name	Type	R/O	Description	Example
Name	char[]	R	Onvif	Onvif

Request Example

http://192.168.1.108/cgi-bin/IntervideoManager.cgi?action=getVersion&Name=Onvif

Response Params (key=value format in body)

Name	Type	R/O	Description	Example
version	char[64]	R	version info	2.4.2

Response Example

version=2.4.2

4.6.16 Get Version of HTTP API

Request URL	http://<server>/cgi-bin/IntervideoManager.cgi?action=getVersion
--------------------	---

Method	GET
---------------	-----

Request Params (key=value format in url)

Name	Type	R/O	Description	Example
Name	char[]	R	CGI	CGI

Request Example

http://192.168.1.108/cgi-bin/IntervideoManager.cgi?action=getVersion&Name=CGI

Response Params (key=value format in body)

Name	Type	R/O	Description	Example
version	char[64]	R	version info	2.0.0

Response Example

version=2.0.0

4.6.17 Get Device Class

Get the class information of the device.

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

Response Params (key=value format in body)			
Name	Type	R/O	Description
class	string	R	The class information of the device.
Response Example			HDVR
class=HDVR			

4.6.18 [Config] Auto Maintain

Config Data Params				
Name	Type	R/O	Description	Example
AutoMaintain	object	O	auto maintain	
+AutoRebootDay	int	O	Range is [-1—7]. Auto reboot day. -1 = never auto reboot 0~6 = Sunday~Saturday 7 = reboot every day	-1
+AutoRebootHour	int	O	Range is [0—23]. Auto reboot hour	0
+AutoRebootMinute	int	O	Range is [0—59]. Auto reboot minute	0
+AutoShutdownDay	int	O	Range is [-1—7]. Auto shutdown day. -1 = never auto shutdown 0~6 = Sunday~Saturday 7 = shutdown every day	-1
+AutoShutdownHour	int	O	Range is [0—23]. Auto shutdown hour	0
+AutoShutdownMinute	int	O	Range is [0—59]. Auto shutdown minute	0
+AutoStartUpDay	int	O	Range is [-1—7]. Auto start day. -1 = never auto start 0~6 = Sunday~Saturday 7 = start every day	-1
+AutoStartUpHour	int	O	Range is [0—23]. Auto start hour	0
+AutoStartUpMinute	int	O	Range is [0—59].	0

			Auto start minute	
--	--	--	-------------------	--

Please refer to "4.2.1 Get and Set Configure" for configuration getting and setting. Specific examples are as follows :

Get Config Request Example

```
http://10.0.0.8/cgi-bin/configManager.cgi?action=getConfig&name=AutoMaintain
```

Get Config Response Example

```
table.AutoMaintain.AutoRebootDay=3
table.AutoMaintain.AutoRebootHour=0
table.AutoMaintain.AutoRebootMinute=0
table.AutoMaintain.AutoShutdownDay=1
table.AutoMaintain.AutoShutdownHour=0
table.AutoMaintain.AutoShutdownMinute=0
table.AutoMaintain.AutoStartUpDay=1
table.AutoMaintain.AutoStartUpHour=2
table.AutoMaintain.AutoStartUpMinute=0
```

Set Config Request Example

```
http://10.0.0.8/cgi-bin/configManager.cgi?action=setConfig&AutoMaintain.AutoRebootDay=7
```

Set Config Response Example

```
OK
```

4.6.19 Reboot

Request URL	http://<server>/cgi-bin/magicBox.cgi?action=reboot			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
delay	char[20]	O	delay time	2016
Request Example				
http://192.168.1.108/cgi-bin/magicBox.cgi?action=reboot				

Response Params (OK in body)

Response Example

```
OK
```

4.6.20 Shutdown

Request URL	http://<server>/cgi-bin/magicBox.cgi?action=shutdown			
Method	GET			
Request Params (none)				
Name	Type	R/O	Description	Example
Request Example				
http://192.168.1.108/cgi-bin/magicBox.cgi?action=shutdown				
Response Params (OK in body)				

Response Example

OK

4.6.21 Factory Reset

Request URL	http://<server>/cgi-bin/magicBox.cgi?action=resetSystemEx		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
type	int	O	<p>0 means all parameters are set to their factory default value;</p> <p>1 means all parameters are set to their factory default value expect the specific parameters;</p> <p>The specific parameters of different device types are different. But it always contains the network settings and user settings. After resetting, the device is reachable on the same ip address as used before the reset.</p> <p>When the Type parameter is not present in the URL, the default value of the Type is 0.</p>
Request Example			http://192.168.1.108/cgi-bin/magicBox.cgi?action=resetSystemEx&type=0

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

OK

4.6.22 Get Tracing Code of Device

Request URL	http://<server>/cgi-bin/magicBox.cgi?action=getTracingCode		
Method	GET		
Request Params (none)			
Name	Type	R/O	Description
Request Example			http://192.168.1.108/cgi-bin/magicBox.cgi?action=getTracingCode

Response Params (key=value format in body)

Name	Type	R/O	Description	Example
tc	char[]	O	Only system default admin can get it. tc is a 64bit binary string.	0001...1101
Response Example				

4.6.23 Add Camera to Specified Channel

Request URL	http://<server>/cgi-bin/LogicDeviceManager.cgi?action=addCameraByGroup		
Method	POST		
Request Params (Json format in body)			
Name	Type	R/O	Description
+group	object[]	R	Add device groups in batches It is an array with the same elements as addCameraByDevice.
++DeviceInfo	object	R	Device details
+++RtspPort	int	O	RTSP port: Required when ProtocolType is not Private, 554 by default.
+++HttpPort	int	O	HTTP port: Required when ProtocolType is not Private, 80 by default.
+++Port	int	O	TCP port: Required when ProtocolType is Private, 37777 by default.
+++UserName	char[32]	R	Username
+++Password	char[32]	R	Password, plain text
+++ProtocolType	char[128]	R	Protocol Type "Private", "Dahua2", "Dahua3", "General", "Onvif", "GB28181", "Ehome"
+++Address	char[16]	R	Device address
++cameras	object[]	R	Video source information
+++uniqueChannel	int	R	Unique channel number, starting from 1
Example			
<pre>{ "group": ["DeviceInfo": { "UserName": "admin", "Password": "pass123456", "ProtocolType": "Private", "Port": 37777, "Address": "172.11.1.161" }, "cameras": [{ "uniqueChannel": 1 }] } }</pre>			

Response Params (JSON format in body)				
Name	Type	R/O	Description	Example
group	object[]	O	device group info	
+deviceID	char[128]	O	device ID	"uuid:4848"
+cameras	object[]	O	camera info	
++uniqueChannel	uint	O	unique channel	1
++failedCode	uint	O	error code. 0: succeed 1: the specific channel does not support configuration	1

Response Example

```
{
  "group": [
    {
      "deviceID": "uuid:4848",
      "cameras": [
        {
          "uniqueChannel": 1,
          "failedCode": 1
        }
      ]
    }
  ]
}
```

4.6.24 Delete Camera by Group

Request URL	http://<server>/cgi-bin/api/LogicDeviceManager/deleteCameraByGroup			
Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
group	object[]	O		
+uniqueChannels	int[]	O	unique channel list When the array size is 1 and the element value is - 1, it indicates that all channels under the "DeviceID" device are deleted	[1...3]
+DeviceID	char[128]	O	Device ID, returned by method addCameraByGroup. When the "uniqueChannels" array size is 1 and the element value is - 1, "DeviceID" is required, otherwise it is not required.	"DH12345678910"
Request Example				
{				
"group": [{}]				

```

        "DeviceID": "DH12345678910",
        "uniqueChannels": [1...3]
    },...{}
}

```

Response Params (JSON format in body)

Name	Type	R/O	Description	Example
info	object[]	O	details	
+failedCode	uint	O	0 indicates success, other indicates failure	0
+deviceID	char[128]	O	Device ID	"DH12345678910"
+uniqueChannel	int	O	unique channel	1

Response Example

```

{
    "info": [
        {
            "failedCode": 0,
            "deviceID": "DH12345678910",
            "uniqueChannel": 1
        },...
    ]
}

```

4.6.25 Acquiring All Available Resources

Customer requests to obtain camera information, such as MAC address and SN, through CGI commands.

Request URL	http://<server>/cgi-bin/LogicDeviceManager.cgi?action=getCameraAll			
Method	GET			
Request Params (key=value format in Url)				
Name	Type	R/O	Description	Example
Request Example				
http://192.168.1.108/cgi-bin/LogicDeviceManager.cgi?action=getCameraAll				

Response Params (key=value format in body)

Name	Type	R/O	Description	Example
camera	object[]	R	camera info	
+ControlID	char[128]	O	Control number. Keyboard users can use the number to locate channels to improve operation efficiency. Control number “0” is not available, which is reserved for special use.	"5001"
+Name	char[128]	O	Name	"XXX Road"
+Enable	bool	O	Enable only when using DeviceID to add/delete devices. Do not enable it through DeviceInfo.	true

			<p>true: Meaning trying to connect the device, but not necessarily successful. You can acquire the connection status through getCameraState.</p> <p>false: Do not connect the device. From user's point of view, it is the same as that the device is not added.</p> <p>This Enable field is only used for the case of adding a device through DeviceID. We keep this field for compatibility and recommend using the Enable in DeviceInfo.</p> <p>When acquiring the field, this Enable and the Enable in DeviceInfo are returned at the same time.</p> <p>When sending the field, you can only select one between this Enable and the Enable in DeviceInfo. We recommend sending the Enable in DeviceInfo.</p>	
+DeviceInfo	DeviceInfo	O	<p>Device details.</p> <p>Use the Enable in the structure to add/delete devices.</p> <p>True: Add devices.</p> <p>False: delete devices from the channel.</p> <p>A device can be added only when "DeviceID" is " " and Info is not NULL.</p>	
++Enable	bool	O	The device is enabled or not.	true
++EncryptStream	int	O	<p>Device stream encryption algorithm (custom field for NVR). (If other encryption algorithms are used, refer to the encryption type definition in stream format code 0x95).</p> <p>0: No encryption by default.</p>	0
++Address	char[16]	O	Device IP address or domain name.	"10.6.5.10"
++Port	int	O	Port number.	37777
++usePreSecret	enumint8	O	Select whether to use NVR preset password to add cameras (the naming remains the same globally,	1

			<p>so the case is not uniform here).</p> <pre>enumint8{ 0: Use UserName/Password to connect devices (default). 1: When using a preset password to connect devices, UserName/Password is not required. 2: When using camera login password to connect devices, UserName/Password is not required. }</pre>	
++UserName	char[32]	O	Username	"admin"
++Password	char[32]	O	Password in plain text.	"admin"
++ProtocolType	enumchar[32]	O	<p>The protocol type of the connected device. For the value range, see the ProtocolType field of the RemoteDevice configuration.</p> <pre>enumchar[32]{ "Private" "Dahua2" "Dahua3" "General" "Onvif" "Ehome" "ICC": The special protocol for the central Intelligence platform. It has not been reviewed, therefore it is not allowed to add the protocol type in this way in the future. "DahuaDSS": The special protocol for the central Intelligence platform. It has not been reviewed, therefore it is not allowed to add the protocol type in this way in the future. }</pre>	"Private"
++VideoInput++ Channels	uint	O	Total number of video input channels, including analog and digital channels	16
++AudioInput++ Channels	uint	O	Total number of audio input channels	4
++DeviceClass	char[64]	O	Device type	"IPC"
++DeviceType	char[64]	O	Device model	"IPC-HF3300"
++HttpPort	int	O	HTTP port number	80
++HttpsPort	int	O	HTTPS port number	443

<code>++RtspPort</code>	int	O	RTSP port number	554
<code>++Name</code>	char[64]	O	Device name	"IPC1"
<code>++MachineAddress</code>	char[256]	O	Device deployment location	"XXX District, YYY Road"
<code>++SerialNo</code>	char[48]	O	Device serial number	"Device12345678"
<code>++VendorAbbr</code>	char[32]	O	Vendor abbreviation (optional) Vendor type: VVV (for example).	"VVV"
<code>++HardID</code>	char[64]	O	Hardware ID (optional)	"454"
<code>++SoftwareVersion</code>	char[64]	O	Software version (optional)	"2.420.000006.0.R.150311"
<code>++ActivationTime</code>	char[24]	O	The date when starting using the device. Date format: "yyyy-MM-dd HH:mm:ss" "Year–Month–Day Hour: Minute: Second" Range {"2000-01-01 00:00:00", /*Minimum date*/ "2099-01-01 00:00:00",/*Maximum date*/}	"2000-01-01 00:00:00"
<code>++NodeType</code>	char[32]	O	Device node type (main or sub node). It is used when adding storage node (DataNodeManager.addDataNode) in N + M cluster mode.	
<code>++Mac</code>	char[]	O	MAC address	"08:00:20:0A:8C:6D"
<code>++OEMVendor</code>	char[]	O	OEM vendor type. During cloud update, for some vendors, the Vendor field is displayed as OEM, and OEMVendor is added to indicate the actual vendor information. OEMVendor is the same as Vendor by default. If OEMVendor is defined in the product definition, refer to the the product definition.	"XXX"
<code>+DeviceID</code>	char[128]	O	Device ID When DeviceID is not " ", ignore DeviceInfo and use the existing DeviceID information in the RemoteDevice configuration.	"dev123"
<code>+Type</code>	enumchar[32]	O	Input channel type. enumchar[32]{ "Local": Local channel (physical channel). "Remote": Remote channel. "Reserved32": Reserve 32	"Remote"

			<p>channels (compatible with the device protocol of the second generation).</p> <p>"Output": Output channel.</p> <p>"Matrix": Analog matrix channel.</p> <p>"Compose": Composite channel.</p> <p>"Cascade": Optical fiber cascading channel.</p> <p>}</p>	
+VideoStream	enumchar[32]	O	<p>Video stream.</p> <p>enumchar[32]{</p> <p>"Main": Main stream</p> <p>"Extra1": Sub stream 1</p> <p>"Extra2": Sub stream 2</p> <p>"Extra3": Sub stream 3</p> <p>"Auto": Automatically selects the appropriate stream.</p> <p>"Preview": Preview raw data stream.</p> <p>"None": No video stream (pure audio stream).</p> <p>}</p>	"Main"
+Channel	uint	O	<p>Channel number of the remote device (optional).</p> <p>DeviceID is unique.</p>	0
+UniqueChannel	int	O	<p>Unique channel number of uniform device number (required).</p> <p>You can obtain video/audio streams by specifying the channel number with 0x11/0xF4 second-generation protocol.</p> <p>Notes: When implementing any addressing function of DeviceID+Channel, LogicDeviceManager.getCameraA II() must be implemented, and the device must support "DeviceID=Unique" unified numbering method for input channels.</p> <p>For setCamera(), uniqueChannel equals -1 means that the device automatically assigns a channel number.</p>	0
+DataNodeID	int	O	Indicates the data node to which the global channel is currently allocated (N+0 cluster).	1

			Counted from 0. If it is -1 or when the field does not exist, it means that it is not allocated currently.	
+DataNodeChannel	int	O	Indicates the local channel of data node to which the global channel is currently allocated (N+0 cluster). Counted from 0. If it is -1 or when the field does not exist, it means that it is not currently allocated.	2
+AutoReset	bool	O	Auto reset. Compatible process: If this parameter is not configured, the default value is false and the channel cannot be reset automatically. when there is no client to access the video stream (streaming client access count changes from non-zero to zero), if AutoReset=true, the device will actively enable the channel to be idle (that is, the video source attribute Enable=false).	false
+VideoStandard	enumchar[32]	O	Current video standard enumchar[32]{ "PAL" "NTSC" "SECAM" }	"PAL"
+Label	enumchar[32]	O	Device label enumchar[32]{ "Indoor": Indoor "Outdoor": Outdoor }	"Indoor"
+ImageInfo	object	O	Device snapshot When adding a device, take a snapshot and save it to local computer.	
++FilePath	char[256]	R	Image path	"/var/local/ch1.jpg"
+PasswordGenType	enumchar[32]	O	Password generation method enumchar[32]{ "UserInput": Entered by users. "Auto": Automatically generated. }	"UserInput"
Response Example				
camera[0].ControlID=5001 camera[0].Name= XXX Road				

```

camera[0].Enable=true
camera[0].DeviceInfo.Enable=true
camera[0].DeviceInfo.EncryptStream=0
camera[0].DeviceInfo.Adress=10.6.5.10
camera[0].DeviceInfo.Port=37777
camera[0].DeviceInfo.usePreSecret=1
camera[0].DeviceInfo.UserName=admin
camera[0].DeviceInfo.Password=pass123456
camera[0].DeviceInfo.ProtocolType=Private
camera[0].DeviceInfo.SoftwareVersion=2.420.0000006.0.R.150311
...
camera[0].DeviceID= dev123
camera[0].ImageInfo. FilePath=/var/local/ch1.jpg"
camera[0].PasswordGenType=UserInput
...
camera[1].ControlID=5002
camera[1].Name= YYY Road

```

4.6.26 Subscribing for Device Online/Offline Status

Subscribe to the online and offline status of the device.

Request URL	http://<server>/cgi-bin/api/LogicDeviceManager/attachCameraState			
Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
channel	int[]	R	Observed unique channel number. "-1" means all unique channels. If one of the array elements is "-1", it will observe all channels.	[1, 2]
heartbeat	int32	O	Send heartbeat interval, range is [1, 60], unit is second. If the URL contains this parameter, and the value is 5, it means every 5 seconds the device should send the heartbeat message to the client, the heartbeat message is a string "Heartbeat". If this parameter is not present, its default value is 60.	10
Request Example				
{ "type": [1, 2], "heartbeat": 10 }				

Response Params (multipart in body)				
Name	Type	R/O	Description	Example

state	object	R	The new status of the camera	
+channel	int32	R	Unique channel number (the channel number with changed status)	1
+connectionState	enumchar[12]	R	Connection status enumchar[12]{ "Connecting" "Connected" "Unconnect": Not connected "Empty": The channel is not configured and there is no information. "Disable": The channel is configured but disabled (the protocol cannot be reflected, and delete it first). "Changed": The information of the device connected to the channel is changed. "Hibernation" }	"Connected"

Response Example

HTTP/1.1 200 OK

Content-Type: multipart/x-mixed-replace; boundary=<boundary>

Connection: close

--<boundary>

Content-Type: application/json

Content-Length: <data length>

{

 "channel" : 1,

 "connectionState" : "Connected"

}

--<boundary>

Content-Type: text/plain

Content-Length: 11

Heartbeat

--<boundary>

...

4.6.27 Get Complete Machine Version

get complete machine version

Request URL	http://<server>/cgi-bin/api/MagicBox/getCompleteMachineVersion
-------------	--

Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
Request Example				
http://192.168.1.108/cgi-bin/api/MagicBox/getCompleteMachineVersion				

Response Params (JSON format in body)				
Name	Type	R/O	Description	Example
CompleteMac hineVersion	char[32]	R	complete machine version	"S2"
Response Example				
{ "CompleteMachineVersion": "S2" }				

4.6.28 Connection test

network connection test of devices and http servers

Request URL	http://<server>/cgi-bin/api/tcpConnect/tcpTest			
Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
Ip	char[40]	R	Http server IP	"10.34.9.21"
Port	int32	R	The port number of the http server	80
Request Example				
{ "Ip": "10.34.9.21", "Port": 80 }				

Response Params (JSON format in body)				
Name	Type	R/O	Description	Example
Connect	bool	R	Whether the connection is successful	true
Response Example				
{ "Connect": true }				

4.6.29 Getting Online Status of the Channel

Request URL	http://<server>/cgi-bin/api/LogicDeviceManager/getCameraState			
Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
uniqueChanne	Int[]	R	Unique channel number	[-1]

ls			If the array is 1, the corresponding value is -1, and it means to get all the channels.	
----	--	--	---	--

Request Example

```
{
    "uniqueChannels": [-1]
}
```

Response Params (JSON format in body)

Name	Type	R/O	Description	Example
states	object[]	R	Online status	
+ channel	uint	R	Channel number	0
++connection State	enumchar[12]	R	Connect status enumchar[12]{ "Connecting": connecting. "Connected": successfully connected "Unconnect": configured but not connect "Empty": The channel is not configured or configured into "" "Disable": The channel is configured but disabled. "UnInit": the front-end devices are not initialized. "Hibernation": hibernate }	"Connected"
++capsState	bool	O	Represents whether the current channel can get capacity from the front-end devices.	true
++errorMessage	char[128]	O	Reason for failed to connect When the connection status is unconnected, it is valid. "LoginConnectFailed", failed to connect to the internet when login. "ErrorHasNotInit", failed to login for the device is not initialized and there are no users. "UserOrPwdNotValid", failed to login for entering the wrong user name or the wrong password. "LoginConnectTimeout", login connection timeout. "LoginFailure", failed to login for unknown reason.	"LoginConnectFailed"

Response Example

```
{
    "states" : [ {
```

```

        "channel" : 0,
        "connectionState" : "Connecting",
        "capsState": true,
        "errorMessage": "LoginConnectFailed"
    }, ..., {}]
}

```

4.6.30 Getting the Recording Status of the Channel

Request URL	http://<server>/cgi-bin/api/recordManager/getStateAll		
Method	POST		
Request Params (JSON format in body)			
Name	Type	R/O	Description
Request Example			
{}			

Response Params (JSON format in body)				
Name	Type	R/O	Description	Example
state	object[]	R	Recording status. The array index represents the corresponding channel.	
+ Main	object	R	Main stream	
++ State	uint	R	Whether the main stream is recording, and if yes, Flag has meaning.	1
+Extra1	object	O	If the recording can be performed in sub streams, then it may exist information such as the recording status of the sub streams, and it is the same with field “Main”.	
+Extra2	object	O	If the recording can be performed in sub streams, then it may exist information such as the recording status of the sub streams, and it is the same with field “Main”.	
+Extra3	object	O	If the recording can be performed in sub streams, then it may exist information such as the recording status of the sub streams, and it is the same with field “Main”.	

Response Example

```
{
    "state" : [ {
        "Main" : {
            "State" : 1
        },
    },
}
```

```

        "Extra1" : {},
        "Extra2" : {},
        "Extra3" : {}
    }, ..., {}]
}

```

4.7 User management

4.7.1 Get Information of a Particular User

Request URL	http://<server>/cgi-bin/userManager.cgi?action=getUserInfo		
Method	GET		
Request Params (key=value format in url)			
Name	Type	R/O	Description
name	char[]	R	user name
Request Example			
http://192.168.1.108/cgi-bin/userManager.cgi?action=getUserInfo?name=admin			

Response Params (key=value format in body)			
Name	Type	R/O	Description
user	object	O	user info
+Name	char[128]	O	user name
+Memo	char[32]	O	remark
+Group	char[32]	O	user group default as follows: admin, user
+Sharable	bool	O	Allow the same user to log in from multiple different IP addresses simultaneously
+Reserved	bool	O	Is it a reserved user? Reserved users cannot be deleted
+PwdValidPeri od	int	O	Remind users to change their account password when it is about to expire.
+AuthorityList	char[][128]	O	authority list

Response Example
user.Name=admin
user.Memo=admin's account
user.Group=admin
user.Reserved=true
user.Sharable=true
user.PwdValidPeriod=30
user.AuthorityList[0]=AuthUserMag
user.AuthorityList[1]=Monitor_01
user.AuthorityList[2]=Replay_01
user.AuthorityList[3]=AuthSysCfg

```

user.AuthorityList[4]=AuthSysInfo
user.AuthorityList[5]=AuthManuCtr

```

4.7.2 Get Information of All Users

Request URL	http://<server>/cgi-bin/userManager.cgi?action=getUserInfoAll			
Method	GET			
Request Params (none)				
Name	Type	R/O	Description	Example
Request Example				http://192.168.1.108/cgi-bin/userManager.cgi?action=getUserInfoAll

Response Params (key=value format in body)				
Name	Type	R/O	Description	Example
user	object[]	O	user infos	
+Id	int	O	user id	1
+Name	char[128]	O	user name	"admin"
+Memo	char[32]	O	remark	"admin's account"
+Group	char[32]	O	user group default as follows: admin, user	"admin"
+Sharable	bool	O	Allow the same user to log in from multiple different IP addresses simultaneously	true
+Reserved	bool	O	Is it a reserved user? Reserved users cannot be deleted	true
+PwdValidPeriod	int	O	Remind users to change their account password when it is about to expire.	30
+AuthorityList	char[][128]	O	authority list	["ShutDown", "Monitor01", "DataFormat",...]

Response Example

```

user[0].Name=admin
user[0].Memo=admin's account
user[0].Group=admin
user[0].Reserved=true
user[0].Sharable=true
user[0].PwdValidPeriod=30
user[0].AuthorityList[0]=AuthUserMag
user[0].AuthorityList[1]=Monitor_01
user[0].AuthorityList[2]=Replay_01
user[0].AuthorityList[3]=AuthSysCfg
user[0].AuthorityList[4]=AuthSysInfo
user[0].AuthorityList[5]=AuthManuCtr
...

```

4.7.3 Get Information of All Active Users

Request URL	http://<server>/cgi-bin/userManager.cgi?action=getActiveUserInfoAll			
Method	GET			
Request Params (none)				
Name	Type	R/O	Description	Example
Request Example				
http://192.168.1.108/cgi-bin/userManager.cgi?action=getActiveUserInfoAll				

Response Params (key=value format in body)				
Name	Type	R/O	Description	Example
user	object[]	O	user info	
+name	char[128]	O	user name	"admin"
+ip	char[32]	O	IP address	"10.43.2.16"
+group	char[32]	O	user group default as follows: admin, user	"admin"
+clienttype	char[32]	O	client type	"web3.0"
+logintime	char[32]	O	login in time	"2011-11-08 09:51:03"
Response Example				
users[0].name=admin users[0].ip=10.43.2.16 users[0].group=admin users[0].clienttype=web3.0 users[0].logintime=2011-11-08 09:51:03 ...				

4.7.4 Get Information of a Particular Group

Request URL	http://<server>/cgi-bin/userManager.cgi?action=getGroupInfo			
Method	GET			
Request Params (key=value format in url)				
Name	Type	R/O	Description	Example
name	char[]	R	name of the group. If the group named groupName does not exist, the device returns Error.	admin
Request Example				
http://192.168.1.108/cgi-bin/userManager.cgi?action=getGroupInfo&name=admin				

Response Params (key=value format in body)				
Name	Type	R/O	Description	Example
group	object	O	group info	
+Name	char[128]	O	The device has one or two default user groups: "admin"or "admin"and "user".	"admin"
+Memo	char[32]	O	remark info	"administrator group"

+AuthorityList	char[][128]	O	The "admin" group has all the authorities of operating the device. The "user" group only has monitoring and replaying authorities.	["ShutDown", "Monitor01", "DataFormat", ...]
----------------	-------------	---	--	---

Response Example

```
group.Name=admin
group.Memo=administrator group
group.AuthorityList[0]=AuthUserMag
group.AuthorityList[1]=Monitor_01
group.AuthorityList[2]=Replay_01
group.AuthorityList[3]=AuthSysCfg
group.AuthorityList[4]=AuthSysInfo
group.AuthorityList[5]=AuthManuCtr
...
```

4.7.5 Get Information of all Groups

Request URL	http://<server>/cgi-bin/userManager.cgi?action=getGroupInfoAll		
Method	GET		
Request Params (none)			
Name	Type	R/O	Description
Request Example			
http://192.168.1.108/cgi-bin/userManager.cgi?action=getGroupInfoAll			

Response Params (key=value format in body)				
Name	Type	R/O	Description	Example
group	object[]	O	group infos	
+Name	char[128]	O	The device has one or two default user groups: "admin"or "admin"and "user".	"admin"
+Memo	char[32]	O	remark info	"administrator group"
+AuthorityList	char[][128]	O	The "admin" group has all the authorities of operating the device. The "user" group only has monitoring and replaying authorities.	["ShutDown", "Monitor01", "DataFormat", ...]

Response Example

```
group[0].Id=1
group[0].Name=admin
group[0].Memo=administrator group
group[0].AuthorityList[0]=AuthUserMag
group[0].AuthorityList[1]=Monitor_01
group[0].AuthorityList[2]=Replay_01
group[0].AuthorityList[3]=AuthSysCfg
...
group[1].Id=1
group[1].Name=user
```

```

group[1].Memo=user group
group[1].AuthorityList[0]=Monitor_01
...

```

4.7.6 Add a New User

Access Control products are not supported.

Request URL	http://<server>/cgi-bin/userManager.cgi?action=addUser			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
user	object	R	user info	
+Name	char[128]	O	user name	"George"
+Password	char[]	O	[password]	"123456"
+Memo	char[32]	O	remark info	"George's account"
+Group	char[32]	O	the range is "admin"and "user". In different group, the user has different authorities.	"user"
+Sharable	bool	O	true means allow multi-point login.	true
+Reserved	bool	O	true means this user can't be deleted.	false
Request Example				
http://192.168.1.108/cgi-bin/userManager.cgi?action=addUser&user.Name=George&user.Password=123456&user.Group=user&user.Sharable=true&user.Reserved=false				

Response Params (OK in body)
Response Example
OK

4.7.7 Delete a User

Access Control products are not supported.

Request URL	http://<server>/cgi-bin/userManager.cgi?action=deleteUser			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
name	char[128]	R	user name	"George"
Request Example				
http://192.168.1.108/cgi-bin/userManager.cgi?action=deleteUser&name=George				

Response Params (OK in body)
Response Example
OK

4.7.8 Modify User Information

Access Control products are not supported.

Request URL	http://<server>/cgi-bin/userManager.cgi?action=modifyUser		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
name	char[128]	R	User name, specified by user name
user	object	R	user info
+Memo	char[32]	O	remark info
+Group	char[32]	O	the range is "admin"and "user". In different group, the user has different authorities.
+Sharable	bool	O	true means allow multi-point login.
+Reserved	bool	O	true means this user can't be deleted.
Request Example			
http://192.168.1.108/cgi-bin/userManager.cgi?action=addUser&user.Name=George&user.Password=123456&user.Group=user&user.Sharable=true&user.Reserved=false			

Response Params (OK in body)
Response Example
OK

4.7.9 Modify User's Password

Access Control products are not supported.

Request URL	http://<server>/cgi-bin/userManager.cgi?action=modifyPassword		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
name	char[128]	R	User name, specified by user name
pwd	char[]	R	new password
pwdOld	char[]	R	Old password should be supplied
Request Example			
http://192.168.1.108/cgi-bin/userManager.cgi?action=modifyPassword&name=George&pwd=abcdef&pwdOld=123456			

Response Params (OK in body)
Response Example
OK

4.7.10 Manager Modify Common User's Password

Access Control products are not supported.

Request URL	http://<server>/cgi-bin/userManager.cgi?action=modifyPasswordByManager			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
userName	char[128]	R	The name of the common user to be modified	"George"
pwd	char[]	R	new password	"abcdef"
managerName	char[]	R	manager name is only admin	"admin"
managerPwd	char[]	R	manager password	"abc123"
accountType	int	R	accountType support 0 is common account, 1 is Onvif account	0

Request Example

http://192.168.1.108/cgi-bin/userManager.cgi?action=modifyPasswordByManager&userName=Geoge&pwd=123456&managerName=admin&managerPwd=abc123&accountType=0

Response Params (OK in body)

Response Example

OK

4.7.11 [Config] User Login Authentication Policy

Config Data Params				
Name	Type	R/O	Description	Example
LoginAuthCtrl	object	O		
+PriSvrPolicy	enumint8	R	login policy setting for Private Protocol. 1 - safe mode, only support digest authentication. (Recommended.); 2 - compatible mode for traditional devices;	1

Please refer to "4.2.1 Get and Set Configure" for configuration getting and setting. Specific examples are as follows :

Get Config Request Example

http://10.0.0.8/cgi-bin/configManager.cgi?action=getConfig&name=LoginAuthCtrl

Get Config Response Example

table.LoginAuthCtrl.PriSvrPolicy=1

Set Config Request Example

http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&LoginAuthCtrl.PriSvrPolicy=2

Set Config Response Example

OK

4.8 Network

4.8.1 Get Network Interfaces

Request URL	http://<server>/cgi-bin/netApp.cgi?action=getInterfaces		
Method	GET		
Request Params (none)			
Name	Type	R/O	Description
Request Example			
http://192.168.1.108/cgi-bin/netApp.cgi?action=getInterfaces			

Response Params (key=value format in body)				
Name	Type	R/O	Description	Example
netInterface	object[]	O	network interface info	
+Name	char[]	O	network interface name. "eth0" - wired network interface "eth2" - wireless network interface "3G" - 3G network interface "bond0" - bond network interface "bond1" - bond network interface Not listed in detail	3g
+Type	char[]	O	Network types that support settings "Normal" — wired network "Wireless" — wireless network "Auto", "TD-SCDMA", "WCDMA", "CDMA1x", "EDGE", "EVDO", "TD-LTE", "FDD-LTE", "NR" — 3G network types.	Auto,WCDMA,TD-SCDMA, TD-LTE,FDD-LTE
+Valid	bool	O	Is network interface valid.	true
+NetCardName	char[8]	O	Real network card name, for example: "eth0","wlan0","ppp5","lte0"	"eth0"
+ConnStatus	enumchar[32]	O	Link connection status of wired network cards, WiFi, and 3G enumchar[32]{ "Inexistence": The network does not exist "Down": close "Disconn": not connected "Connecting" "Connected" }	"Connected"
+Speed	uint8	O	Network card theoretical speed Mbps	100
+PhysicalAddr	char[18]	O	A string of hexadecimal addresses	"00:10:5C:F2:1C:B4"

ess			for the physical address of the network card	
+SupportLongPoE	bool	O	Does the device support long-distance POE speed negotiation	true
Response Example				
netInterface[0].Name=eth0				
netInterface[0].Type=Normal				
netInterface[0].Valid=true				
netInterface[0].NetCardName=eth0				
netInterface[0].ConnStatus=Connected				
netInterface[0].Speed=100				
netInterface[0].PhysicalAddress=00:10:5C:F2:1C:B4				
netInterface[0].SupportLongPoE=true				
...				
netInterface[1].Name=3g				
netInterface[1].Type=Auto,WCDMA,TD-SCDMA,TD-LTE,FDD-LTE				
netInterface[1].Valid=true				
netInterface[1].NetCardName=eth0				
netInterface[1].ConnStatus=Connected				
netInterface[1].Speed=100				
netInterface[1].PhysicalAddress=00:10:5C:F2:1C:B4				
netInterface[1].SupportLongPoE=true				
...				

4.8.2 [Config] Client Access Filter

Config Data Params				Example
Name	Type	R/O	Description	
AccessFilter	object	O	access filter function	
+Enable	bool	O	Enable/Disable access filter function	true
+Type	enumchar[32]	O	Range is {TrustList, BannedList}, TrustList: Trust list is used, banned list is not used. BannedList: Banned list is used, trust list is not used.	"BannedList"
+TrustList	char[]	O	Trusted IP address list , for example: "10.6.10.10-10.6.10.20"	["10.6.10.1", "10.6.10.2"]
+BannedList	char[]	O	Banned IP address list	["10.6.10.1", "10.6.10.2"]

Please refer to "4.2.1 Get and Set Configure" for configuration getting and setting. Specific examples are as follows :

Get Config Request Example

```
http://10.0.0.8/cgi-bin/configManager.cgi?action=getConfig&name=AccessFilter
```

Get Config Response Example

```
table.AccessFilter.Enable=false
```

```

table.AccessFilter.TrustList[0]=10.6.10.23
table.AccessFilter.TrustList[1]=10.6.10.62
table.AccessFilter.BannedList[0]=10.6.10.1
table.AccessFilter.BannedList[1]=10.6.10.2
table.AccessFilter.Type=BannedList

```

Set Config Request Example

`http://10.0.0.8/cgi-bin/configManager.cgi?action=setConfig&AccessFilter.Enable=true&AccessFilter.Type=BannedList&AccessFilter.BannedList[0]=10.6.10.1`

Set Config Response Example

OK

4.8.3 [Config] Network Basic Config

Config Data Params				
Name	Type	R/O	Description	Example
Network	object	O	network basic config specified by the interface name, up to 32 interfaces.	
+Hostname	char[128]	O	Hostname and Domain compose a network address.	"badak"
+Domain	char[128]	O	Domain name.	"dahua"
+DefaultInterface	char[32]	O	Set default network interface when multiple interfaces exist. Range of interfaces is depends on GetInterfaces.	"eth0"
+eth0	object	O	interfcace name eth0 details as follows.	
++PhysicalAddress	char[18]	O	MAC address of interface. HEX string in the form of: xx:xx:xx:xx:xx:xx. Range of x is [0-9, a-f, A-F] Example: 00:10:5c:f2:1c:b4 00:10:5C:F2:1C:B5	"11:2D:A3:4C:5F:66"
++MTU	uint	O	Interface MTU.	1500
++NetMode	enumchar[32]	O	Network transmission mode enumchar[32]{ "adapt" "half10M" "full10M" "half100M" "full100M" "full1000M" "longPoE10M" "longPoE100M"	"adapt"

			}	
++IPAddress	char[40]	O	Interface IP address.	"192.168.0.108"
++SubnetMask	char[40]	O	Network mask string: In the form of x.x.x.x, range of x is [0-255] Example: 255.255.255.0	"255.255.0.0"
++DefaultGateway	char[40]	O	Gateway IP address.	"192.168.0.1"
++DhcpEnable	bool	O	Enable/Disable DHCP.	false
++EnableDhcpReservedIP	bool	O	Enable/Disable when dhcp failed, it will continue sending dhcp request, if it was enabled.	true
++DnsServers	char[2][40]	O	IP address of DNS servers	["221.123.33.228", "221.12.1.228"]
+eth2	object	O	other interface config, refer to eth0	

Please refer to "4.2.1 Get and Set Configure" for configuration getting and setting. Specific examples are as follows :

Get Config Request Example

```
http://10.0.0.8/cgi-bin/configManager.cgi?action=getConfig&name=Network
```

Get Config Response Example

```
table.Network.DefaultInterface=eth0
table.Network.Domain=ttt
table.Network.Hostname=hhh
table.Network.eth0.DefaultGateway=10.7.0.1
table.Network.eth0.DhcpEnable=false
table.Network.eth0.DnsServers[0]=221.123.33.228
table.Network.eth0.DnsServers[1]=221.12.1.228
table.Network.eth0.IPAddress=10.7.2.3
table.Network.eth0.MTU=1500
table.Network.eth0.PhysicalAddress=00:10:5c:f2:1c:b4
table.Network.eth0.SubnetMask=255.255.0.0
```

Set Config Request Example

```
http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&NetWork.Domain=ttt&NetWork.eth0.DhcpEnable=true
```

Set Config Response Example

```
OK
```

4.8.4 [Config] PPPoE

Config Data Params

Name	Type	R/O	Description	Example
PPPoE	object	O	PPPoE	

+Enable	bool	O	Enable/Disable PPPoE.	true
+AuthType	enumchar[] [32]	O	The PPPoE dial-up authentication method, which authentication method to use, is determined by the server. No this field, default to selecting all enumchar[][32]{ "PAP" "CHAP" "EAP" }	["PAP",]
+UserName	char[64]	O	PPPoE user name.	"hz150260"
+Password	char[64]	O	PPPoE user password.	"Dont Know"
+Mode	char[8]	O	dial mode default Custom "Default": Use default network card "Custom": network card specified by eth	"Default"
+Eth	char[32]	O	Network card used for current dialing	"eth0"
+DefaultPPPoEInterface	char[8]	O	The default ppp network card	"ppp0"
+PPPOE1	object	O	Other PPPOE dialing configurations The naming method is:PPPOE1、PPPOE2、PPPOE3	
++Enable	bool	O	Enable/Disable PPPoE.	true
++AuthType	enumchar[] [32]	O	The PPPoE dial-up authentication method, which authentication method to use, is determined by the server. No this field, default to selecting all enumchar[][32]{ "PAP" "CHAP" "EAP" }	["PAP",]
++UserName	char[8]	O	PPPoE user name.	"hz150260"
++Password	char[64]	O	PPPoE user password.	"Don't Know"
++Mode	char[8]	O	dial mode default Custom "Default": Use default network card "Custom": network card specified by eth	"Custom"
++Eth	char[32]	O	Network card used for current dialing	"eth1"

Get Config Request Example

http://10.0.0.8/cgi-bin/configManager.cgi?action=getConfig&name=PPPoE

Get Config Response Example

table.PPPoE.Enable=true,
table.PPPoE.AuthType[0]=PAP,

```

table.PPPoE.UserName=hz150260,
table.PPPoE.Password=Dont Know,
table.PPPoE.Mode=Default,
table.PPPoE.Eth=eth0,
table.PPPoE.DefaultPPPoEInterface=ppp0,
table.PPPoE.PPPOE1.Enable=true,
table.PPPoE.PPPOE1.AuthType[0]=PAP,
table.PPPoE.PPPOE1.UserName=hz150260,
table.PPPoE.PPPOE1.Password=Don't Know,
table.PPPoE.PPPOE1.Mode=Custom,
table.PPPoE.PPPOE1.Eth=eth1

```

Set Config Request Example

http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&PPPoE.UserName=user1&PPPoE.Pa
ssword=123456

Set Config Response Example

OK

4.8.5 [Config] DDNS

Config Data Params

Name	Type	R/O	Description	Example
DDNS	object[10]	O	Dynamic Domain Name System configuration can support multiple DDNS protocols at the same time. The maximum number is 10	
+Enable	bool	O	Multiple DDNS hostname can be configured, but Only one hostname can be enabled, others should not be enabled.	true
+Protocol	enumchar[32]	R	DDNS protocol type. Range is { "PRIVATE DDNS": private ddns version 1 "Private DDNS": private ddns version 2 "CN99 DDNS" "NO-IP DDNS" "Dyndns DDNS" "LUPUS DDNS" "Oray DDNS" "CP Plus DDNS" "Ipplus DDNS" "Q-See DDNS" "SYSDNS DDNS" "Videotrend DDNS" "CEPSA DDNS"	"Private DDNS"

			"G4IP DDNS" "HOSS DDNS" "Intelbras DDNS" "HSY DDNS" "Flir DDNS" "Private P2P DDNS": private ddns for P2P "Mivilante DDNS" "ByDemes DDNS" "DYNDNS OLD DDNS": old dyndns for storate device "WATASHI DDNS" "CNB DDNS" "Peoplefu DDNS" "MINT DDNS" "SPECO DDNS" "SHANY DDNS" "WITHCCTV DDNS": "SONIC DDNS" "KBVISION DDNS" "BOSCH DDNS" }.	
+Address	char[128]	O	DDNS server IP address or name.	"www.ddns.com"
+Port	uint16	O	Range is [1—65535]. Port of DDNS server	5050
+Password	char[64]	O	DDNS user password	"ro238na"
+HostName	char[128]	O	Hostname of this device.	"www.tech.com/IPCamera1"
+KeepAlive	int	O	Range is [1—65535]. Unit is minutes.	10
+DefaultHostName	object	O	Only protocol is in range {"Private DDNS", "DHDDNS", "QUICK DDNS"}, it effects.	
++Enable	bool	O	true: use the DefaultHostName.HostName false: use HostName	true
++HostName	char[128]	O	The default hostname. It cannot be modified.	"9002A9D77133.ttt.com"

Please refer to "4.2.1 Get and Set Configure" for configuration getting and setting. Specific examples are as follows :

Get Config Request Example

```
http://10.0.0.8/cgi-bin/configManager.cgi?action=getConfig&name=DDNS
```

Get Config Response Example

```
table.DDNS[0].Address=www.ttt.com
```

```
table.DDNS[0].Enable=true
```

```
table.DDNS[0].HostName=www.ttt.com
```

```

table.DDNS[0].KeepAlive=10
table.DDNS[0].Password=none
table.DDNS[0].Port=5050
table.DDNS[0].Protocol= Quick DDNS
table.DDNS[0].UserName=user1
table.DDNS[0].DefaultHostName.Enable=false
table.DDNS[0].DefaultHostName.HostName=9002A9D77133.ttt.com
...

```

Set Config Request Example

http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&DDNS[0].Address=www.ttt.com&DDNS[0].Enable=true

Set Config Response Example

OK

4.8.6 [Config] Email

Config Data Params				
Name	Type	R/O	Description	Example
Email	object	O	email configuration	
+Enable	bool	O	Enable/Disable email function	true
+SslEnable	bool	O	Is mandatory encryption enabled? Both SslEnable and TlsEnable can only have one true, usually using port 465	false
+TlsEnable	bool	O	True: enable Tls email.	false
+Address	char[128]	O	SMTP server IP address or name.	"www.xxx.com"
+Port	uint16	O	Range is [1-65535]	25
+UserName	char[128]	O	User name of email account.	"anonymity"
+Password	char[128]	O	User password of email account.	"none"
+Anonymous	bool	O	Enable/Disable anonymous email.	true
+Authentication	bool	O	Enable email authentication	true
+SendAddress	char[254]	O	sender address	"wang_xxx@xxx.com"
+Receivers	char[3][254]	O	Email addresses of 3 receivers.	"1@xxx.com", "2@xxx.com"]
+Title	char[256]	O	Title of email.	"DVR Message"
+OnlyAttachment	bool	O	Only send emails with attachments	false
+AttachEnable	bool	O	Enable/Disable email attachment	true
+SendInterval	uint	O	send interval	10
+HealthReport	object	O	report device status by email	
++Enable	bool	O	email enable	false
++Interval	uint	O	Interval, unit is second ranges is 0~86400	120
+CustomTitle	enumchar[] [32]	O	mail custom title enumchar[] [32]{	["IP", "Index", ..]

		<pre>"MachineName" "IP" "Index" "EventName" }</pre>	
--	--	---	--

Please refer to "4.2.1 Get and Set Configure" for configuration getting and setting. Specific examples are as follows :

Get Config Request Example

```
http://10.0.0.8/cgi-bin/configManager.cgi?action=getConfig&name=Email
```

Get Config Response Example

```
table.Email.Address=www.ttt.com
table.Email.Anonymous=true
table.Email.AttachEnable=true
table.Email.Enable=true
table.Email.HealthReport.Enable=false
table.Email.HealthReport.Interval=61
table.Email.Password=123456
table.Email.Port=26
table.Email.Receivers[0]=x@ttt.com
table.Email.Receivers[1]=y@ttt.com
table.Email.Receivers[2]=z@ttt.com
table.Email.SendAddress=x@ttt.com
table.Email.SslEnable=false
table.Email.TlsEnable=false
table.Email.Authentication=false
table.Email.OnlyAttachment=false
table.Email.SendInterv=10
table.Email.Title=DVRMessage
table.Email.UserName=anonymity
```

Set Config Request Example

```
http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&Email.Address=mail.ttt.com&Email.Anonymous=false
```

Set Config Response Example

```
OK
```

4.8.7 [Config] WLan

Config Data Params				
Name	Type	R/O	Description	Example
WLan	object	O	All wireless network card settings, specified by the wireless interface name	
+eth2	object	O	wileless interface configuration	
++Enable	bool	O	True: Enable WLan on this interface.	true

++SSID	char[32]	O	SSID	"ttt"
++BSSID	char[18]	O	MAC	"00:aa:0a:a0:11:23"
++ConnectEnable	bool	O	Connect Manually	true
++LinkMode	enumchar[32]	O	Range is {Auto, Ad-hoc, and Infrastructure}. Auto – select suitable mode automatically. Ad-hoc – Device with wireless network adapter can connect to each other without Access Point. Infrastructure – Integrate wire and wireless LAN together to share network resource, access point is need in this mode.	"Auto"
++Encryption	enumchar[32]	O	Range is {Off, On, WEP64Bits, WEP128Bits, WPA-PSK-TKIP, WPA-PSK-CCMP} Encryption mode	"Off"
++KeyType	enumchar[32]	O	Range is {Hex, ASCII}	"Hex"
++KeyID	int	O	Range is [0—3] Indicates which key is used. 0: WLan.interface.Keys[0] is used.	0
++Keys	char[4][128]	O	For ASCII key type: 64bits encryption key length is 5, 128bits encryption key length is 13, consists of [0—9, a—z, A—Z] For HEX key type: 64bits encryption key length is 10, 128bits encryption key length is 26, consists of [0—9, a—z, A—Z]	["password1", "password2", "password3", "password4"]
++KeyFlag	bool	O	true: key is configured.	false
+wlan0	object	O	other wileless interface	
+wlan1	object	O	other wileless interface	

Please refer to "4.2.1 Get and Set Configure" for configuration getting and setting. Specific examples are as follows :

Get Config Request Example

```
http://10.0.0.8/cgi-bin/configManager.cgi?action=getConfig&name=WLan
```

Get Config Response Example

```
table.WLan.eth2.Enable=true
table.WLan.eth2.Encryption=off
table.WLan.eth2.KeyFlag=false
table.WLan.eth2.KeyID=0
table.WLan.eth2.KeyType=Hex
table.WLan.eth2.Keys[0]=password1
```

```

table.WLan.eth2.Keys[1]=password2
table.WLan.eth2.Keys[2]=password3
table.WLan.eth2.Keys[3]=password4
table.WLan.eth2.LinkMode=Auto
table.WLan.eth2.SSID=ttt

```

Set Config Request Example

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

Set Config Response Example

OK

4.8.8 Scan Wlan Devices

Request URL	http://<server>/cgi-bin/wlan.cgi?action=scanWlanDevices			
Method	GET			
Request Params (key=value format in url)				
Name	Type	R/O	Description	Example
SSID	char[]	O	SSID	xia_yuguo 13098 Internet
Request Example				
http://192.168.1.108/cgi-bin/wlan.cgi?action=scanWlanDevices&SSID=xia_yuguo 13098 Internet				

Response Params (key=value format in body)				
Name	Type	R/O	Description	Example
found	int	R	Quantity found	1
wlanDevice	object[]	O	wireless devices	
+BSSID	char[18]	O	MAC	"00:aa:0a:a0:11:23"
+ApConnected	enumint	O	connection status range{ 0: not connected 1: connecting 2: connected }	0
+ApMaxBitRate	uint	O	The maximum transmission rate supported by the route, in Bps	54000000
+ApNetworkType	enumint	O	ap network type range { 0: 11b 1: 11a 2: 11a/n 3: 11b/g 4: 11b/g/n }	3
+AuthMode	int	O	Authorization mode	0

+BSSID	char[18]	O	MAC	"00:aa:0a:a0:11:23"
+EncrAlgr	enumint	O	encryption range{ 0: NONE 1: WEP 2: TKIP 3: AES(CCMP) 4: TKIP+AES(mix Mode) 5: UnKnown }	0
+LinkMode	enumint	O	link mode range{ 0: auto 1: adhoc 2: Infrastructure }	0
+LinkQuality	uint8	O	quality, percentage range[0, 100]	30
+RSSIQuality	int8	O	Received signal strength in dBm range [-100, 0]	-68
+SSID	char[32]	O	Specified SSID, if not include any SSID, all Wi-Fi information will be searched and displayed.	"xia_yuguo 13098 Internet"

Response Example

```
found=1
wlanDevice[0].ApConnected=0
wlanDevice[0].ApMaxBitRate=54000000
wlanDevice[0].ApNetWorkType=255
wlanDevice[0].AuthMode=7
wlanDevice[0].BSSID=28:2c:b2:5c:de:36
wlanDevice[0].EncrAlgr=3
wlanDevice[0].LinkMode=0
wlanDevice[0].LinkQuality=31
wlanDevice[0].RSSIQuality=0
wlanDevice[0].SSID=xia_yuguo 13098 Internet
```

4.8.9 [Config] UPnP

Config Data Params				
Name	Type	R/O	Description	Example
UPnP	object	O	UPnP configuration	
+Enable	bool	O	Enable/Disable UPNP feature.	false
+StartDeviceDiscovery	bool	O	Be searched by UPnP	true
+Mode	enumchar[32]	O	mode range{"Manual"}	"Manual"

			"Auto" }	
+MapTable	object[255]	O	UPNP map table	
++Enable	bool	O	Enable/Disable this UPNP map.	true
++ServiceName	char[64]	O	User defined UPnP service name.	"WebService"
++Protocol	enumchar[32]	O	Range is {TCP, UDP}	"TCP"
++InnerPort	uint16	O	Range is [1—65535]. Inner port number	80
++OuterPort	uint16	O	Range is [1—65535]. Outer port number.	8080

Please refer to "4.2.1 Get and Set Configure" for configuration getting and setting. Specific examples are as follows :

Get Config Request Example

```
http://10.0.0.8/cgi-bin/configManager.cgi?action=getConfig&name=UPnP
```

Get Config Response Example

```
table.UPnP.Enable=true
table.UPnP.MapTable[0].Enable=true
table.UPnP.MapTable[0].InnerPort=80
table.UPnP.MapTable[0].OuterPort=8080
table.UPnP.MapTable[0].Protocol=TCP
table.UPnP.MapTable[0].ServiceName=HTTP
```

Set Config Request Example

```
http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&UPnP.Enable=true&UPnP.MapTable[0].Protocol=TCP
```

Set Config Response Example

```
OK
```

4.8.10 Get UPnP Status

Request URL	http://<server>/cgi-bin/netApp.cgi?action=getUPnPStatus			
Method	GET			
Request Params (none)				
Name	Type	R/O	Description	Example
Request Example				
http://192.168.1.108/cgi-bin/netApp.cgi?action=getUPnPStatus				

Response Params (key=value format in body)

Name	Type	R/O	Description	Example
status	object	O	mapping status	
+Working	bool	O	is working	true
+Status	enumchar[32]	O	Enumchar[32]{ Success Unknown}	"Success"

			Error }	
+InnerAddress	char[40]	O	Inner IP	"10.6.2.50"
+OuterAddres s	char[40]	O	Outer IP	"200.1.2.8"
+PortMapStat us	enumchar[] [32]	O	Port mapping status, The order is consistent with the order in MapTable in UpnP configuration. Enumchar[32]{ Success Unknown Failed }	["Success", ...]

Response Example

```
status.InnerAddress=0.0.0.0
status.OuterAddress=0.0.0.0
status.PortMapStatus[0]=Failed
status.PortMapStatus[1]=Failed
status.PortMapStatus[2]=Failed
status.PortMapStatus[3]=Failed
status.Status=Unknown
status.Working=false
```

4.8.11 [Config] NTP

Config Data Params				
Name	Type	R/O	Description	Example
NTP	object	O	NTP	
+Enable	bool	O	Enable/Disable	true
+Address	char[128]	O	main NTP server address or domain name	"clock.isc.org"
+Port	uint16	O	port	37
+ServerList	object[]	O	Alternative NTP Server Address List	
++Enable	bool	O	Enable/Disable	true
++Address	char[40]	O	IP address or domain name	192.168.1.108
++Port	uint16	O	port	123
+UpdatePerio d	uint16	O	Update cycle, in minutes 1~65535, 0 indicates non calibration	30
+Tolerance	int	O	Indicates the tolerance between the set time and the current time, in seconds. If the set time and the current time are within the tolerance range, the current time is not updated. 0 means it is modified every time.	5
+TimeZone	int	O	time zone	13

			{0, 0*3600,"GMT+00:00"}, {1, 1*3600,"GMT+01:00"}, {2, 2*3600,"GMT+02:00"}, {3, 3*3600,"GMT+03:00"}, {4, 3*3600+1800,"GMT+03:30"}, {5, 4*3600,"GMT+04:00"}, {6, 4*3600+1800,"GMT+04:30"}, {7, 5*3600,"GMT+05:00"}, {8, 5*3600+1800,"GMT+05:30"}, {9, 5*3600+1800+900,"GMT+05:45"}, {10, 6*3600,"GMT+06:00"}, {11, 6*3600+1800,"GMT+06:30"}, {12, 7*3600,"GMT+07:00"}, {13, 8*3600,"GMT+08:00"}, {14, 9*3600,"GMT+09:00"}, {15, 9*3600+1800,"GMT+09:30"}, {16, 10*3600,"GMT+10:00"} {17, 11*3600,"GMT+11:00"} {18, 12*3600,"GMT+12:00"} {19, 13*3600,"GMT+13:00"} {20, -1*3600,"GMT-01:00"}, {21, -2*3600,"GMT-02:00"}, {22, -3*3600,"GMT-03:00"}, {23, -3*3600-1800,"GMT-03:30"}, {24, -4*3600,"GMT-04:00"}, {25, -5*3600,"GMT-05:00"}, {26, -6*3600,"GMT-06:00"}, {27, -7*3600,"GMT-07:00"}, {28, -8*3600,"GMT-08:00"}, {29, -9*3600,"GMT-09:00"}, {30, -10*3600,"GMT-10:00"}, {31, -11*3600,"GMT-11:00"}, {32, -12*3600,"GMT-12:00"}, {33, -4*3600-1800,"GMT-4:30"}, {34, 10.5*3600,"GMT+10:30"}, {35, 14*3600, "GMT+14:00"}, {36,-9*3600-1800,"GMT-09:30"}, {37,8*3600+1800,"GMT+08:30"}, {38,8*3600+2700,"GMT+08:45"}, {39,12*3600+2700,"GMT+12:45"} 	
+TimeZoneDes	char[128]	O	time zone description	"Beijing"

Please refer to "4.2.1 Get and Set Configure" for configuration getting and setting. Specific examples are as follows :

Get Config Request Example

<http://10.0.0.8/cgi-bin/configManager.cgi?action=getConfig&name=NTP>

Get Config Response Example

```
table.NTP.Address=clock.isc.org  
table.NTP.Enable=false  
table.NTP.Port=38  
table.NTP.TimeZone=9  
table.NTP.UpdatePeriod=31  
table.NTP.ServerList[0].Enable=true,  
table.NTP.ServerList[0].Address=192.168.1.108,  
table.NTP.ServerList[0].Port=123  
table.NTP.Tolerance=5,  
table.NTP.TimeZoneDesc=GMT
```

Set Config Request Example

```
http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&NTP.Address=time.ttt.com&NTP.Enable=true
```

Set Config Response Example

```
OK
```

4.8.12 [Config] RTSP

Config Data Params				
Name	Type	R/O	Description	Example
RTSP	object	O	RTSP	
+Enable	bool	O	Enable/Disable RTSP.	false
+Port	uint16	O	RTSP port.	554
+RTP	object	O		
++StartPort	uint16	O	RTP start port.	20000
++EndPort	uint16	O	RTP end port.	40000

Please refer to "4.2.1 Get and Set Configure" for configuration getting and setting. Specific examples are as follows :

Get Config Request Example

```
http://10.0.0.8/cgi-bin/configManager.cgi?action=getConfig&name=RTSP
```

Get Config Response Example

```
table.RTSP.Enable=true  
table.RTSP.Port=554  
table.RTSP.RTP.EndPort=40000  
table.RTSP.RTP.StartPort=20000
```

Set Config Request Example

```
http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&RTSP.Enable=true&RTSP.Port=554
```

Set Config Response Example

```
OK
```

4.8.13 [Config] Alarm Server

Config Data Params				
Name	Type	R/O	Description	Example
AlarmServer	object	O	alarm server	
+Enable	bool	O	Enable/Disable Alarm server.	true
+Protocol	enumchar[32]	O	enumchar[32]{ "DAHUA" "BOSCH" "DAHUA_CLOUD" }	"DAHUA"
+Address	char[128]	O	IP address	"www.dahuatech.com"
+Port	uint16	O	port	37777
+EthernetName	char[32]	O	The network card name	"3G"
+UserName	char[64]	O	user name	"anonymity"
+Password	char[64]	O	password	"none"
+ReportWeekDay	int	O	-1 means not to report, 0-6 means to report regularly from Sunday to Saturday, and 7 means to report regularly every day	4
+ReportTime	char[32]	O	Regular report time	"12:00:00"

Please refer to "4.2.1 Get and Set Configure" for configuration getting and setting. Specific examples are as follows :

Get Config Request Example

```
http://10.0.0.8/cgi-bin/configManager.cgi?action=getConfig&name=AlarmServer
```

Get Config Response Example

```
table.AlarmServer.Address=10.7.8.9
table.AlarmServer.Enable=false
table.AlarmServer.Password=
table.AlarmServer.Port=8888
table.AlarmServer.Protocol=ttt
table.AlarmServer.ReportTime=02:00:00
table.AlarmServer.ReportWeekDay=2
table.AlarmServer.UserName=admin
```

Set Config Request Example

```
http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&AlarmServer.Address=as.ttt.com&AlarmServer.Enable=false
```

Set Config Response Example

```
OK
```

4.8.14 [Config] Onvif Service Authorization

Config Data Params

Name	Type	R/O	Description	Example
UserGlobal	object	O	global configuration	
+OnvifLoginCheck	bool	O	If "OnvifLoginCheck" is false, you can get Onvif service directly; if true, you should enter your ID/username and password.	true

Please refer to "4.2.1 Get and Set Configure" for configuration getting and setting. Specific examples are as follows :

Get Config Request Example

```
http://10.0.0.8/cgi-bin/configManager.cgi?action=getConfig&name=UserGlobal
```

Get Config Response Example

```
table.UserGlobal.OnvifLoginCheck=false
```

Set Config Request Example

```
http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&UserGlobal.OnvifLoginCheck=true
```

Set Config Response Example

```
OK
```

4.8.15 [Config] SSSH Config

Config Data Params				
Name	Type	R/O	Description	Example
SSHD	object	O		
+Enable	bool	R	enable/disable default false	false
+Port	uint16	R	sshd listen port , default 22	22

Please refer to "4.2.1 Get and Set Configure" for configuration getting and setting. Specific examples are as follows :

Get Config Request Example

```
http://10.0.0.8/cgi-bin/configManager.cgi?action=getConfig&name=SSHD
```

Get Config Response Example

```
table.SSHD.Enable=true
```

```
table.SSHD.Port=22
```

Set Config Request Example

```
http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&SSHD.Enable=true&SSHD.Port=22
```

Set Config Response Example

```
OK
```

4.8.16 [Config] Cellular Network Traffic Packages

Config Data Params				
Name	Type	R/O	Description	Example
CellularFlux	object	O	Configure cellular network traffic packages.	

+3G	object	O	Connect cellular network. Supports 4 cellular modules. Single module: 3G; multiple modules: 3G, 3G1, 3G2, and 3G3. Notes: Considering the compatibility, the 3G naming is reserved, which can actually represent 4G and 5G network adapters.	
++SIM1	object	O	Configure the traffic package of the first SIM card. If there is a second one, it will be named SIM2, and the third one will be named SIM3.	
+++Enable	bool	O	Enable traffic strategy.	false
+++MonthFlux	object	O	Configure monthly traffic.	
++++FluxUp	uint32	O	The upper limit of traffic usage	299
++++FluxUpUnit	char[32]	O	The upper limit unit of traffic usage. GB and MB are available.	"GB"
++++FluxWarn	uint32	O	The upper limit of traffic reminders	99
++++FluxWarnUnit	char[32]	O	The upper limit unit of traffic reminders. GB and MB are available.	"GB"
++++FluxStartDay	uint32	O	The digit indicates what day the package starts every month.	1

Please refer to "4.2.1 Get and Set Configure" for configuration getting and setting. Specific examples are as follows :

Get Config Request Example

```
http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=CellularFlux
```

Get Config Response Example

```
table.CellularFlux.3G.SIM1.Enable=true
table.CellularFlux.3G.SIM1.MonthFlux.FluxStartDay=1
table.CellularFlux.3G.SIM1.MonthFlux.FluxUp=299
table.CellularFlux.3G.SIM1.MonthFlux.FluxUpUnit=GB
table.CellularFlux.3G.SIM1.MonthFlux.FluxWarn=99
table.CellularFlux.3G.SIM1.MonthFlux.FluxWarnUnit=GB
```

Set Config Request Example

```
http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&CellularFlux.3G.SIM1.Enable=true&CellularFlux.3G.SIM1.MonthFlux.FluxStartDay=1&CellularFlux.3G.SIM1.MonthFlux.FluxUp=299&CellularFlux.3G.SIM1.MonthFlux.FluxUpUnit=GB&CellularFlux.3G.SIM1.MonthFlux.FluxWarn=99&CellularFlux.3G.SIM1.MonthFlux.FluxWarnUnit=GB
```

Set Config Response Example

OK

4.8.17 Obtaining the Traffic Information of Current Month

Acquiring the traffic information of current month

Request URL	http://<server>/cgi-bin/api/DataFlux/getFlux
Method	POST

Request Params (JSON format in body)

Name	Type	R/O	Description	Example
card	char[32]	R	The key value of the first-level node under Wireless. For earlier devices, it represents the SIM card number; For new devices, it represents the module name. The specific values are: enumchar[32]{ 3G 3G1 3G2 3G3 } Single card/module: 3G; Multiple cards/modules: 3G, 3G1, 3G2, and 3G3.	"3G"
SimName	char[32]	O	SIM card name. The first SIM card is SIM1, the second SIM2, and the third SIM3.	"SIM1"
ICCID	char[24]	O	IC card identification code (namely SIM card number). It consists of 20 digits.	

Request Example

```
{
    "card": "3G",
    "SimName": "SIM1",
    "ICCID": "898607E2112251029357"
}
```

Response Params (JSON format in body)

Name	Type	R/O	Description	Example
info	object	R	Traffic information.	
+LastMonthTotalFlux	uint	R	The total traffic of the last month. Unit: 0.1 kb.	8000
+totalFlux	uint	R	Total traffic = uplink traffic + downlink traffic. Unit: 0.1 kb.	8000
+SendFlux	uint	R	Uplink traffic. Unit: 0.1 kb.	3000
+RecvFlux	uint	R	Downlink traffic. Unit: 0.1 kb.	5000
+Time	string	R	Total traffic write time. YYYY-MM-DD hh:mm:ss	"2014-01-01 14:01:01" "
+Record	object[]	R	Traffic records in the last seven days.	
++Date	string	R	Date.	"2014-01-01"
++DaySendFlux	uint	R	Uplink traffic of the day. Unit: 0.1 kb.	3000
++DayRecvFlux	uint	R	Downlink traffic of the day. Unit: 0.1	3000

			kb.	
++DayTotalFlux	uint64	O	The total traffic of the day. Unit: Byte.	8000
++MonthTotalFlux	uint64	O	The total traffic of the month. Unit: Byte.	8000

Response Example

```
{
  "info": {
    "LastMonthTotalFlux" : 8000,
    "totalFlux" : 8000,
    "SendFlux": 3000,
    "RecvFlux": 5000,
    "Time": "2014-01-01 14:01:01",
    "Record" : [
      {"Date": "2014-01-01",
       "DaySendFlux" : 3000,
       "DayRecvFlux" : 3000,
       "DayTotalFlux" : 8000,
       "MonthTotalFlux" : 8000
      }, ..., {}]
  }
}
```

4.8.18 Searching for History Traffic Statistics According to Specified Conditions

Searching for history traffic statistics by specified conditions.

Request URL	http://<server>/cgi-bin/api/DataFlux/queryHistoryFlux			
Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
condition	object	R	Search condition	
+Card	char[32]	R	The key value of the first-level node under Wireless. For earlier devices, it represents the SIM card number; For new devices, it represents the module name. The specific values are: enumchar[32]{ 3G 3G1 3G2 3G3 } Single card/module: 3G; Multiple	"3G"

			cards/modules: 3G, 3G1, 3G2, and 3G3.	
+SimName	char[32]	O	SIM card name. The first SIM card is SIM1, the second SIM2, and the third SIM3.	"SIM1"
+ICCID	char[24]	O	IC card identification code (namely SIM card number). It consists of 20 digits.	"89860116836014532534"
Mode	char[32]	R	Search modes: enumchar[32]{ month: Search by month day: Search by day hour: Search by hour }	"day"
+StartTime	char[20]	O	Start time of statistics	"2021-01-01 00:00:00"
+EndTime	char[20]	O	End time of statistics	"2021-04-01 00:00:00"

Request Example

```
{
  "condition": {
    "Card": "3G",
    "Mode": "month",
    "StartTime": "2022-08-01 00:00:00",
    "EndTime": "2022-09-30 23:59:59",
    "SimName": "SIM1",
    "ICCID": "898607E2112251029357"
  }
}
```

Response Params (JSON format in body)				
Name	Type	R/O	Description	Example
info	object	O	Search results	
+TotalFlux	uint64	R	Total traffic = uplink traffic + downlink traffic. Unit: 0.1 Byte.	1299700
+SendFlux	uint64	R	The total uplink traffic in the statistical time interval. Unit: Byte.	70305
+RecvFlux	uint64	R	The total downlink traffic in the statistical time interval. Unit: Byte.	1229395
+Time	string	R	Data update time. YYYY-MM-DD hh:mm:ss	"2021-03-31 00:00:00"
+Record	object[]	R	Traffic statistic records. maximum 31 entries for variable length array.	
++Date	string	R	Year/month/day. YYYY-MM-DD	"2014-01-01"
++SendFlux	uint64	R	Uplink traffic of the year/month/day.	70305

			Unit: Byte.	
++RecvFlux	uint64	R	Uplink traffic of the year/month/day. Unit: Byte.	1229395
++TotalFlux	uint64	O	The total traffic in the current statistical period. Unit: Byte.	8000
++StartTime	string	O	Start time of this entry. YYYY-MM-DD hh:mm:ss	"2010-05-25 00:00:00"

Response Example

```
{
  "info": {
    "TotalFlux": 1299700,
    "SendFlux": 70305,
    "RecvFlux": 1229395,
    "Time": "2021-03-31 00:00:00",
    "Record": [
      {
        "Date": "2021-03-31",
        "StartTime": "2022-09-01 00:00:00",
        "SendFlux": 70305,
        "RecvFlux": 1229395,
        "TotalFlux": 0
      }, ...
    ]
  }
}
```

4.8.19 [Config]Configuring General Settings of RTMP

The function is the same as RTMP, but the rules for generating Channel ID in URL are different. Existing requirements: The customer requires that the URL of RTMP connection can be directly managed on NVR web page. At the same time, when the related URL cannot be connected successfully, the URL can be automatically obtained from the management platform by using HTTPS interaction for updating, so the following two fields need to be added on the basis of RTMP_NVR configuration: URL: Configure and manage the URL of the main and sub streams on NVR. Key: Enter this filed when using signaling to interact with the management platform. Note: Although the configuration name is RTMP_NVR, both NVR and IPC will use this configuration, which is a general configuration.

Config Data Params

Name	Type	R/O	Description	Example
RTMP_NVR	object	O	Both IPC and NVR are using the general configuration of RTMP.	
+Enable	bool	O	Enable the function.	false
+Address	char[40]	O	RTMP server address	"192.168.1.108"
+Port	uint16	O	RTMP server port	5500
+CustomPath	char[64]	O	Custom path name	"live"
+StreamPath	char[64]	O	Stream path prefix. Different channels are distinguished by suffix numbers.	"liveStream"
+Key	char[64]	O	Used to connect to the cloud server to obtain the key needed for RTMP	"axklala"

			address.	
+Main	object[]	O	It is an array. Each element represents the configuration of a video channel.	
++Enable	bool	O	Enable the function.	false
++ChannelID	uint	O	Custom channel ID. Channel ID in URL.	2
++Url	char[256]	O	URL of RTMP connection	"rtmp://livepush.myqcloud.com/live/camera001"
+Extra	object[]	O	Sub stream 1. It is an array. Each element represents the configuration of a video channel.	
++Enable	bool	O	Enable the function.	false
++ChannelID	uint	O	Custom channel ID. Channel ID in URL.	1
++Url	char[256]	O	URL of RTMP connection	"rtmp://livepush.myqcloud.com/live/camera001"
+Extra2	object[]	O	Sub stream 2. It is an array. Each element represents the configuration of a video channel.	
++Enable	bool	O	Enable the function.	false
++ChannelID	uint	O	Custom channel ID. Channel ID in URL.	1
++Url	char[256]	O	URL of RTMP connection	"rtmp://livepush.myqcloud.com/live/camera001"
+SSLEncryptEnable	bool	O	Enable SSL encryption or not.	false
+AudioSimulation	bool	O	Enable the audio simulation function for the audio-free devices or not.	false

See the following examples to read and modify configurations.

Get Config Request Example

```
http://10.0.0.8/cgi-bin/configManager.cgi?action=getConfig&name=RTMP_NVR
```

Get Config Response Example

```
table.RTMP_NVR.Address=0.0.0.0
table.RTMP_NVR.CustomPath=live
table.RTMP_NVR.Enable=true
table.RTMP_NVR.Extra[0].ChannelID=0
table.RTMP_NVR.Extra[0].Enable=false
table.RTMP_NVR.Extra[0].Url=
table.RTMP_NVR.Extra2[0].ChannelID=0
table.RTMP_NVR.Extra2[0].Enable=false
table.RTMP_NVR.Extra2[0].Url=
table.RTMP_NVR.Key=
```

```

table.RTMP_NVR.Main[0].ChannelID=0
table.RTMP_NVR.Main[0].Enable=true
table.RTMP_NVR.Main[0].Url=
table.RTMP_NVR.Port=1935
table.RTMP_NVR.StreamPath=liveStream

```

Set Config Request Example

http://10.0.0.8/cgi-bin/configManager.cgi?action=setConfig&RTMP_NVR.Enable=true

Set Config Response Example

OK

4.9 Event

4.9.1 [DataType] Event Handler

Config Data Params				
Name	Type	R/O	Description	Example
< <i>EventName</i> >	object/object[]	O	<p>EventHandler configuration.</p> <p>The specific type is determined by the event name. If it is an array, each element in the array represents the configuration of a channel</p> <p><<i>EventName</i>> can be:</p> <ul style="list-style-type: none"> Alarm BlindDetct LossDetect MotionDetect LoginFailureAlarm NetAbort ... 	
+EventHandler	object	O	linkage handler	
++TimeSection	TimeSection[w d][ts]	O	<p>It's an effective time period for eventHanlder everyday.</p> <p>wd (week day) range is [0—6] (Sunday—Saturday)</p> <p>ts (time section) range is [0-23], it's index of time section table.</p> <p>Format: mask hh:mm:ss-hh:mm:ss</p> <p>Mask: {0,1}, hh: [0—24], mm: [00—59], ss: [00—59]</p> <p>Mask 0: this time section is not used.</p>	[[{"1": "12:00:00-18:00:00", "1": "19:00:00-20:00:00"}, {"1": "12:00:00-18:00:00", "1": "19:00:00-20:00:00"}, {"1": "12:00:00-18:00:00", "1": "19:00:00-20:00:00"}, {"1": "12:00:00-18:00:00", "1": "19:00:00-20:00:00"}, {"1": "12:00:00-18:00:00", "1": "19:00:00-20:00:00"}, {"1": "12:00:00-18:00:00", "1": "19:00:00-20:00:00"}, {"1": "12:00:00-18:00:00", "1": "19:00:00-20:00:00"}], [{"1": "12:00:00-18:00:00", "1": "19:00:00-20:00:00"}, {"1": "12:00:00-18:00:00", "1": "19:00:00-20:00:00"}, {"1": "12:00:00-18:00:00", "1": "19:00:00-20:00:00"}, {"1": "12:00:00-18:00:00", "1": "19:00:00-20:00:00"}, {"1": "12:00:00-18:00:00", "1": "19:00:00-20:00:00"}, {"1": "12:00:00-18:00:00", "1": "19:00:00-20:00:00"}, {"1": "12:00:00-18:00:00", "1": "19:00:00-20:00:00"}], [{"1": "12:00:00-18:00:00", "1": "19:00:00-20:00:00"}, {"1": "12:00:00-18:00:00", "1": "19:00:00-20:00:00"}, {"1": "12:00:00-18:00:00", "1": "19:00:00-20:00:00"}, {"1": "12:00:00-18:00:00", "1": "19:00:00-20:00:00"}, {"1": "12:00:00-18:00:00", "1": "19:00:00-20:00:00"}, {"1": "12:00:00-18:00:00", "1": "19:00:00-20:00:00"}, {"1": "12:00:00-18:00:00", "1": "19:00:00-20:00:00"}], [{"1": "12:00:00-18:00:00", "1": "19:00:00-20:00:00"}, {"1": "12:00:00-18:00:00", "1": "19:00:00-20:00:00"}, {"1": "12:00:00-18:00:00", "1": "19:00:00-20:00:00"}, {"1": "12:00:00-18:00:00", "1": "19:00:00-20:00:00"}, {"1": "12:00:00-18:00:00", "1": "19:00:00-20:00:00"}, {"1": "12:00:00-18:00:00", "1": "19:00:00-20:00:00"}, {"1": "12:00:00-18:00:00", "1": "19:00:00-20:00:00"}]

			<p>Mask 1: this time section is used. Example: TimeSection[1][0]=1 12:00:00-18:00:00 Means EventHandler is effective between 12:00:00 and 18:00:00 at Monday.</p>	"1 19:00:00-20:00:00"]]
++RecordEnable	bool	O	Enable/Disable record function.	true
++RecordCloudEnable	bool	O	Enable/Disable record cloud storage	true
++RecordChannels	int[]	O	record channel list	[0, 1, 2]
++RecordLatch	int	O	Range is [10—300]. Unit is seconds, indicates the time to record after input alarm is cleared.	10
++AlarmOutEnable	bool	O	Enable/Disable alarm out function.	true
++AlarmOutChannels	int[]	O	<p>index starts from 0. Range is {0, 1}, 0 — do not output alarm at alarm out channel 1 — output alarm at alarm out channel</p>	[1, 4]
++AlarmOutLatch	int	O	Range is [10—300]. Unit is seconds, indicates the time to output alarm after input alarm is cleared.	10
++ExAlarmOutEnable	bool	O	Enable/Disable extend alarm out ability	true
++ExAlarmOutChannels	int[]	O	extend alarm out channels	[2, 3]
++PtzLink	(char[], int)[]	O	<p>The first dimension represents the channel index. The second dimension represents actions and values. It contains two elements. The first is PTZ action linked with events. Range is {None, Preset, Tour, Pattern} The second is the parameter of PtzLink[ch][0], If PtzLink[ch][0] is Preset: this is preset point.</p>	[["Preset", 3], ["Tour", 4], ["Pattern", 1]]

			Tour: this is tour path number. Pattern: this is pattern number.	
++PtzLinkEnable	bool	O	Enable/Disable PTZ link.	true
++SnapshotEnable	bool	O	Enable/Disable snapshot function.	true
++SnapshotChannels	int[]	O	snapshot channel list	[2, 4]
SnapshotPeriod	int	O	Range is [0—255]. Frames between snapshots. 0 means continuously snapshot for every frame.	3
++SnapshotTimes	int	O	Range is [0—65535] Snapshot times before stop, 0 means don't stop snapshot.	5
++TipEnable	bool	O	Enable/Disable local message box tip.	true
++MailEnable	bool	O	Enable/Disable mail send for alarm.	true
++BeepEnable	bool	O	Enable/Disable beep.	true
++Dejitter	int	O	Range is [0—255]. Alarm signal dejitter seconds. Alarm signal change during this period is ignored.	0
++Delay	uint	O	Range is [0—300]. Delay seconds before setting take effect.	30
++LightingLink	object	O	LightingLink	
+++Enable	bool	O	Enable/Disable LightingLink	true
+++FilckerLightType	enumchar[32]	O	Filcker Light Type range :"WhiteLight","RedBlueLight"	"WhiteLight"
+++LightLinkType	enumchar[32]	O	range:"Filcker","KeepLighting"	"Filcker"
+++FilckerIntevalTime	float	O	Filcker Inteval Time unit: 0.1s	5
+++FilckerTimes	int	O	Filcker Times, default 5	5
+++LightDuration	uint	O	Light Duration unit: s	10
+++WhiteLightTimeSection	char[]	O	time section	
+++LightBright	uint	O	brightness	50
TrigerHttp	object	O	linkage command	
+TrigerHttpCommand	char[]	R	Uri of linkage command	"/cgi-bin/eventManager.cgi?"
+TrigerHttpEnable	bool	R	enable linkage command	true

+TrigerServerName	char[32]	R	linkage server name in [Config] Triger Http User List	"server1"
-------------------	----------	---	--	-----------

Please refer to "4.2.1 Get and Set Configure" for configuration getting and setting. Specific examples are as follows :

Get Config Request Example

```
http://10.0.0.8/cgi-bin/configManager.cgi?action=getConfig&name=MotionDetect[0].EventHandler
```

Get Config Response Example

```
table.MotionDetect[0].EventHandler.AlarmOut=1
table.MotionDetect[0].EventHandler.AlarmOutChannels[0]=0
table.MotionDetect[0].EventHandler.AlarmOutEnable=true
table.MotionDetect[0].EventHandler.AlarmOutLatch=10
table.MotionDetect[0].EventHandler.BeepEnable=false
table.MotionDetect[0].EventHandler.Dejitter=30
table.MotionDetect[0].EventHandler.Delay=0
table.MotionDetect[0].EventHandler.ExAlarmOut=1
table.MotionDetect[0].EventHandler.ExAlarmOutChannels[0]=0
table.MotionDetect[0].EventHandler.ExAlarmOutEnable=true
table.MotionDetect[0].EventHandler.LightingLink.Enable=false
table.MotionDetect[0].EventHandler.LightingLink.FilckerIntevalTime=2
table.MotionDetect[0].EventHandler.LightingLink.FilckerLightType=WhiteLight
table.MotionDetect[0].EventHandler.LightingLink.FilckerTimes=5
table.MotionDetect[0].EventHandler.LightingLink.LightDuration=10
table.MotionDetect[0].EventHandler.LightingLink.LightLinkType=Filcker
table.MotionDetect[0].EventHandler.LightingLink.WhiteLightTimeSection[0][0]=1 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.LightingLink.WhiteLightTimeSection[0][1]=0 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.LightingLink.WhiteLightTimeSection[0][2]=0 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.LightingLink.WhiteLightTimeSection[0][3]=0 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.LightingLink.WhiteLightTimeSection[0][4]=0 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.LightingLink.WhiteLightTimeSection[0][5]=0 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.LightingLink.WhiteLightTimeSection[1][0]=1 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.LightingLink.WhiteLightTimeSection[1][1]=0 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.LightingLink.WhiteLightTimeSection[1][2]=0 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.LightingLink.WhiteLightTimeSection[1][3]=0 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.LightingLink.WhiteLightTimeSection[1][4]=0 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.LightingLink.WhiteLightTimeSection[1][5]=0 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.LightingLink.WhiteLightTimeSection[2][0]=1 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.LightingLink.WhiteLightTimeSection[2][1]=0 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.LightingLink.WhiteLightTimeSection[2][2]=0 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.LightingLink.WhiteLightTimeSection[2][3]=0 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.LightingLink.WhiteLightTimeSection[2][4]=0 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.LightingLink.WhiteLightTimeSection[2][5]=0 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.LightingLink.WhiteLightTimeSection[3][0]=1 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.LightingLink.WhiteLightTimeSection[3][1]=0 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.LightingLink.WhiteLightTimeSection[3][2]=0 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.LightingLink.WhiteLightTimeSection[3][3]=0 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.LightingLink.WhiteLightTimeSection[3][4]=0 00:00:00-23:59:59
```

```
table.MotionDetect[0].EventHandler.LightingLink.WhiteLightTimeSection[3][5]=0 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.LightingLink.WhiteLightTimeSection[4][0]=1 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.LightingLink.WhiteLightTimeSection[4][1]=0 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.LightingLink.WhiteLightTimeSection[4][2]=0 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.LightingLink.WhiteLightTimeSection[4][3]=0 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.LightingLink.WhiteLightTimeSection[4][4]=0 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.LightingLink.WhiteLightTimeSection[4][5]=0 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.LightingLink.WhiteLightTimeSection[5][0]=1 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.LightingLink.WhiteLightTimeSection[5][1]=0 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.LightingLink.WhiteLightTimeSection[5][2]=0 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.LightingLink.WhiteLightTimeSection[5][3]=0 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.LightingLink.WhiteLightTimeSection[5][4]=0 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.LightingLink.WhiteLightTimeSection[5][5]=0 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.LightingLink.WhiteLightTimeSection[6][0]=1 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.LightingLink.WhiteLightTimeSection[6][1]=0 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.LightingLink.WhiteLightTimeSection[6][2]=0 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.LightingLink.WhiteLightTimeSection[6][3]=0 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.LightingLink.WhiteLightTimeSection[6][4]=0 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.LightingLink.WhiteLightTimeSection[6][5]=0 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.MailEnable=false
table.MotionDetect[0].EventHandler.PtzLink[0][0]=None
table.MotionDetect[0].EventHandler.PtzLink[0][1]=0
table.MotionDetect[0].EventHandler.PtzLink[1][0]=None
table.MotionDetect[0].EventHandler.PtzLink[1][1]=0
table.MotionDetect[0].EventHandler.PtzLink[2][0]=None
table.MotionDetect[0].EventHandler.PtzLink[2][1]=0
table.MotionDetect[0].EventHandler.PtzLink[3][0]=None
table.MotionDetect[0].EventHandler.PtzLink[3][1]=0
table.MotionDetect[0].EventHandler.PtzLink[4][0]=None
table.MotionDetect[0].EventHandler.PtzLink[4][1]=0
table.MotionDetect[0].EventHandler.PtzLinkEnable=false
table.MotionDetect[0].EventHandler.Record=1
table.MotionDetect[0].EventHandler.RecordChannels[0]=0
table.MotionDetect[0].EventHandler.RecordEnable=true
table.MotionDetect[0].EventHandler.RecordLatch=10
table.MotionDetect[0].EventHandler.Snapshot=1
table.MotionDetect[0].EventHandler.SnapshotChannels[0]=0
table.MotionDetect[0].EventHandler.SnapshotEnable=true
table.MotionDetect[0].EventHandler.SnapshotPeriod=0
table.MotionDetect[0].EventHandler.SnapshotTimes=0
table.MotionDetect[0].EventHandler.TimeSection[0][0]=1 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.TimeSection[0][1]=0 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.TimeSection[0][2]=0 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.TimeSection[0][3]=0 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.TimeSection[0][4]=0 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.TimeSection[0][5]=0 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.TimeSection[1][0]=1 00:00:00-23:59:59
```

```

table.MotionDetect[0].EventHandler.TimeSection[1][1]=0 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.TimeSection[1][2]=0 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.TimeSection[1][3]=0 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.TimeSection[1][4]=0 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.TimeSection[1][5]=0 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.TimeSection[2][0]=1 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.TimeSection[2][1]=0 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.TimeSection[2][2]=0 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.TimeSection[2][3]=0 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.TimeSection[2][4]=0 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.TimeSection[2][5]=0 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.TimeSection[3][0]=1 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.TimeSection[3][1]=0 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.TimeSection[3][2]=0 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.TimeSection[3][3]=0 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.TimeSection[3][4]=0 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.TimeSection[3][5]=0 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.TimeSection[4][0]=1 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.TimeSection[4][1]=0 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.TimeSection[4][2]=0 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.TimeSection[4][3]=0 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.TimeSection[4][4]=0 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.TimeSection[4][5]=0 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.TimeSection[5][0]=1 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.TimeSection[5][1]=0 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.TimeSection[5][2]=0 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.TimeSection[5][3]=0 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.TimeSection[5][4]=0 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.TimeSection[5][5]=0 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.TimeSection[6][0]=1 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.TimeSection[6][1]=0 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.TimeSection[6][2]=0 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.TimeSection[6][3]=0 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.TimeSection[6][4]=0 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.TimeSection[6][5]=0 00:00:00-23:59:59
table.MotionDetect[0].EventHandler.TipEnable=false
table.MotionDetect[0].EventHandler.Tour=0
table.MotionDetect[0].EventHandler.TourChannels[0]=0
table.MotionDetect[0].EventHandler.TourEnable=false

```

Set Config Request Example

[http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&Alarm\[0\].EventHandler.AlarmOutChannels\[0\]=1&Alarm\[0\].EventHandler.AlarmOutEnable=true](http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&Alarm[0].EventHandler.AlarmOutChannels[0]=1&Alarm[0].EventHandler.AlarmOutEnable=true)

Set Config Response Example

OK

4.9.2 [Config] Alarm Event

Config Data Params				
Name	Type	R/O	Description	Example
Alarm	object[]	O	alarm configuration, array alarm channel, starts from 0	
+Enable	bool	R	Enable/Disable alarm from a input channel	false
+EventHandler	EventHandler	O	Setting of EventHandler is described in SetEventHandler	
+Name	char[]	O	Name of alarm input channel.	Door1
+SensorType	char[]	O	Range is [NC, NO]. NC: normal close NO: normal open	NC

Please refer to "4.2.1 Get and Set Configure" for configuration getting and setting. Specific examples are as follows :

Get Config Request Example

`http://10.0.0.8/cgi-bin/configManager.cgi?action=getConfig&name=Alarm`

Get Config Response Example

`table.Alarm[0].Enable=false
table.Alarm[0].EventHandler....(output of EventHandler is described in GetEventHandler)
table.Alarm[0].Name=Door1
table.Alarm[0].SensorType=NC
table.Alarm[1]....
...`

Set Config Request Example

`http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&Alarm[0].Enable=true`

Set Config Response Example

OK

4.9.3 [Config] Alarm Out

Config Data Params				
Name	Type	R/O	Description	Example
AlarmOut	object[]	O	alarm out configuration, array alarm out channel, starts from 0	
+Name	char[]	O	Alarm out port name.	Beep
+Mode	int	O	Range is {0, 1, 2} 0: automatically alarm 1: force alarm 2: close alarm	0

Please refer to “4.2.1 Get and Set Configure” for configuration getting and setting. Specific examples are as follows :

Get Config Request Example

http://10.0.0.8/cgi-bin/configManager.cgi?action=getConfig&name=AlarmOut

Get Config Response Example

table.AlarmOut[0].Mode=0
table.AlarmOut[0].Name=Beep
...

Set Config Request Example

http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&AlarmOut[0].Mode=0&AlarmOut[0].Name=port1

Set Config Response Example

OK

4.9.4 Get Alarm Input Channels

Request URL	http://<server>/cgi-bin/alarm.cgi?action=getInSlots			
Method	GET			
Request Params (none)				
Name	Type	R/O	Description	Example
Request Example				
http://192.168.1.108/cgi-bin/alarm.cgi?action=getInSlots				

Response Params (key=value format in body)				
Name	Type	R/O	Description	Example
result	int	R	alarm in channel number	2
Response Example				
result=2				

4.9.5 Get Alarm Output Channels

Request URL	http://<server>/cgi-bin/alarm.cgi?action=getOutSlots			
Method	GET			
Request Params (none)				
Name	Type	R/O	Description	Example
Request Example				
http://192.168.1.108/cgi-bin/alarm.cgi?action=getOutSlots				

Response Params (key=value format in body)				
Name	Type	R/O	Description	Example
result	int	R	alarm output channel number.	1
Response Example				
result=1				

4.9.6 Get States of Alarm Input Channels

Request URL	http://<server>/cgi-bin/alarm.cgi?action=getInState			
Method	GET			
Request Params (none)				
Name	Type	R/O	Description	Example
Request Example				
http://192.168.1.108/cgi-bin/alarm.cgi?action=getInState				

Response Params (key=value format in body)				
Name	Type	R/O	Description	Example
result	int	R	A bit in the response result indicates a channel alarm states, result 3 means alarm channel 1 and channel 2 have alarm now.	3
Response Example				
result=3				

4.9.7 Get States of Alarm Output Channels

Request URL	http://<server>/cgi-bin/alarm.cgi?action=getOutState			
Method	GET			
Request Params (none)				
Name	Type	R/O	Description	Example
Request Example				
http://192.168.1.108/cgi-bin/alarm.cgi?action=getOutState				

Response Params (key=value format in body)				
Name	Type	R/O	Description	Example
result	int	R	A bit in the response result indicates a channel, result 1 means alarm is present.	1
Response Example				
result=1				

4.9.8 [Config] Video Blind Event

Config Data Params				
Name	Type	R/O	Description	Example
BlindDetect	object[]	O	video blind configuration, array video channel, starts from 0	
+Enable	bool	R	Enable/Disable blind detect feature.	false

+EventHandle

r

Setting of EventHandler is described

in **SetEventHandler**

+Level	int	O	Range is [1—6]. Sensitivity of blind detection. 1: lowest sensitivity. 6: highest sensitivity.	3
--------	-----	---	---	---

Please refer to “4.2.1 Get and Set Configure” for configuration getting and setting. Specific examples are as follows :

Get Config Request Example

```
http://10.0.0.8/cgi-bin/configManager.cgi?action=getConfig&name=BlindDetect
```

Get Config Response Example

```
table.BlindDetect[0].Enable=false
table.BlindDetect[0].EventHandler....(output of EventHandler is described in GetEventHandler)
table.BlindDetect[0].Level=3
table.BlindDetect[1]....
...
```

Set Config Request Example

```
http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&BlindDetect[0].Enable=true
```

Set Config Response Example

```
OK
```

4.9.9 [Config] Video Loss Event

Config Data Params				
Name	Type	R/O	Description	Example
LossDetect	object[]	O	video loss configuration, array video channel, starts from 0	
+Enable	bool	R	Enable/Disable loss detect feature.	false
+EventHandler	EventHandler	O	Setting of EventHandler is described in SetEventHandler .	

Please refer to “4.2.1 Get and Set Configure” for configuration getting and setting. Specific examples are as follows :

Get Config Request Example

```
http://10.0.0.8/cgi-bin/configManager.cgi?action=getConfig&name=LossDetect
```

Get Config Response Example

```
table.LossDetect[0].Enable=false
table.LossDetect[0].EventHandler...(output of EventHandler is described in GetEventHandler)
table.LossDetect[1]....
...
```

Set Config Request Example

```
http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&LossDetect[0].Enable=true
```

Set Config Response Example

```
OK
```

4.9.10 [Config] Login Failure Event

Config Data Params				
Name	Type	R/O	Description	Example
LoginFailureAlarm	object	O	login failure event config	
+Enable	bool	R	Enable/Disable to notify LoginFailure event. Now this event can be linked with send email and alarm out. The max try login times can be configured in chapter SetGeneralConfig .	false
+EventHandler	EventHandler	O	Setting of EventHandler is described in SetEventHandler .	

Please refer to “4.2.1 Get and Set Configure” for configuration getting and setting. Specific examples are as follows :

Get Config Request Example

```
http://10.0.0.8/cgi-bin/configManager.cgi?action=getConfig&name=LossDetect
```

Get Config Response Example

```
table.LoginFailureAlarm.Enable=false
```

```
table.LoginFailureAlarm.EventHandler...(output of EventHandler is described in GetEventHandler)
```

Set Config Request Example

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

Set Config Response Example

```
OK
```

4.9.11 [Config] Storage Not Exist Event

Config Data Params				
Name	Type	R/O	Description	Example
StorageNotExist	object	O	storage not exist event config	
+Enable	bool	R	Enable/Disable	false
+EventHandler	EventHandler	O	Setting of EventHandler is described in SetEventHandler .	

Please refer to “4.2.1 Get and Set Configure” for configuration getting and setting. Specific examples are as follows :

Get Config Request Example

```
http://10.0.0.8/cgi-bin/configManager.cgi?action=getConfig&name=StorageNotExist
```

Get Config Response Example

```
table.StorageNotExist.Enable=false
```

```
table.StorageNotExist.EventHandler...(output of EventHandler is described in GetEventHandler)
```

Set Config Request Example

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

Set Config Response Example

OK

4.9.12 [Config] Storage Access Failure Event

Config Data Params

Name	Type	R/O	Description	Example
StorageFailure	object	O	storage access failure event config	
+Enable	bool	R	Enable/Disable to notify access failure event	false
+EventHandler	EventHandler	O	Setting of EventHandler is described in SetEventHandler .	

Please refer to “4.2.1 Get and Set Configure” for configuration getting and setting. Specific examples are as follows :

Get Config Request Example

`http://10.0.0.8/cgi-bin/configManager.cgi?action=getConfig&name=StorageFailure`

Get Config Response Example

`table.StorageFailure.Enable=false`

`table.StorageFailure.EventHandler... (output of EventHandler is described in GetEventHandler)`

Set Config Request Example

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

Set Config Response Example

OK

4.9.13 [Config] Storage Low Space Event

Config Data Params

Name	Type	R/O	Description	Example
StorageLowSpace	object	O	storage low space event config	
+Enable	bool	R	Enable/Disable to notify low space event	false
+EventHandler	EventHandler	O	Setting of EventHandler is described in SetEventHandler .	

Please refer to “4.2.1 Get and Set Configure” for configuration getting and setting. Specific examples are as follows :

Get Config Request Example

`http://10.0.0.8/cgi-bin/configManager.cgi?action=getConfig&name=StorageLowSpace`

Get Config Response Example

`table.StorageLowSpace.Enable=false`

`table.StorageLowSpace.EventHandler... (output of EventHandler is described in GetEventHandler)`

Set Config Request Example

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

Set Config Response Example

OK

4.9.14 [Config] Net Abort Event

Config Data Params

Name	Type	R/O	Description	Example
NetAbort	object	O	net abort event config	
+Enable	bool	R	Enable/Disable to notify net abort event	false
+EventHandler	EventHandler	O	Setting of EventHandler is described in SetEventHandler .	

Please refer to “4.2.1 Get and Set Configure” for configuration getting and setting. Specific examples are as follows :

Get Config Request Example

<http://10.0.0.8/cgi-bin/configManager.cgi?action=getConfig&name=NetAbort>

Get Config Response Example

table.NetAbort.Enable=false

table.NetAbort.EventHandler...(output of EventHandler is described in GetEventHandler)

Set Config Request Example

<http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&NetAbort.Enable=true>

Set Config Response Example

OK

4.9.15 [Config] IP Conflict Event

Config Data Params

Name	Type	R/O	Description	Example
IPConflict	object	O	IP conflict event config	
+Enable	bool	R	Enable/Disable to notify IP conflict event	false
+EventHandler	EventHandler	O	Setting of EventHandler is described in SetEventHandler .	
+FilterEth	char[8][16]	O	net cards to be filtered	["eth0"]
+SendArpWhenConflict	bool	O	send ARP when IP conflict default false	false

Please refer to “4.2.1 Get and Set Configure” for configuration getting and setting. Specific examples are as follows :

Get Config Request Example

<http://10.0.0.8/cgi-bin/configManager.cgi?action=getConfig&name=IPConflict>

Get Config Response Example

table.IPConflict.Enable=false

table.IPConflict.EventHandler...(output of EventHandler is described in GetEventHandler)

table.IPConflict.FilterEth[0]=eth0

table.IPConflict.SendArpWhenConflict=false

Set Config Request Example

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

Set Config Response Example

OK

4.9.16 Get Channels Event Happened

Get channels indexes that event of code eventCode happens.

Not all events support this command. Do not recommend to use it, use Attach command instead.

Request URL	http://<server>/cgi-bin/eventManager.cgi?action=getEventIndexes			
Method	GET			
Request Params (key=value format in url)				
Name	Type	R/O	Description	Example
code	char[]	R	eventCode includes: VideoMotion: motion detection event VideoLoss: video loss detection event VideoBlind: video blind detection event. AlarmLocal: alarm detection event. StorageNotExist: storage not exist event. StorageFailure: storage failure event. StorageLowSpace: storage low space event. AlarmOutput: alarm output event.	AlarmLocal
Request Example				<code>http://192.168.1.108/cgi-bin/eventManager.cgi?action=getEventIndexes&code=AlarmLocal</code>

Response Params (key=value format in body)

Name	Type	R/O	Description	Example
channels	int[]	O	The list of channels where the specified event codes have occurred. The element value is the channel number. (This response means event happened on channel 0, channel 2 and channel 3 while video channel index starts from 0)	[0, 2, 3]

Response Example

`channels[0]=0
channels[1]=2
channels[2]=3`

4.9.17 Subscribe to Event Message

Subscribe to event messages. For details on the event name and parameters, see the corresponding [Event] sections.

Request URL	http://<server>/cgi-bin/eventManager.cgi?action=attach		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
codes	string	R	<p>Event code, can be a list of event codes or "All" to cover all event codes.</p> <p>Here are some frequently used event codes:</p> <ul style="list-style-type: none"> VideoMotion: motion detection event SmartMotionHuman: human smart motion detection SmartMotionVehicle: Vehicle smart motion detection VideoLoss: video loss detection event VideoBlind: video blind detection event. AlarmLocal: alarm detection event. StorageNotExist: storage not exist event. StorageFailure: storage failure event. StorageLowSpace: storage low space event. AlarmOutput: alarm output event. AudioMutation: intensity change AudioAnomaly: input abnormal CrossLineDetection: tripwire event CrossRegionDetection: intrusion event LeftDetection: abandoned object detection TakenAwayDetection: missing object detection VideoAbnormalDetection: scene change event FaceDetection: face detect event VideoUnFocus: defocus detect event WanderDetection: loitering detection event RioterDetection: People Gathering event ParkingDetection: parking detection event MoveDetection: fast moving event HeatImagingTemper: temperature alarm event CrowdDetection: crowd density overrun event FireWarning: fire warning event FireWarningInfo: fire warning specific data info ObjectPlacementDetection: placement object detection
Example	[AlarmLocal, VideoMotion, VideoLoss, VideoBlind, FaceDetection]		

			ObjectRemovalDetection: removal object detection AccessControl: Access control event	
keepalive	int	O	<p>Client keep-alive. If this parameter exists, the client will send keep-alive data to the device at an interval in seconds; the value range is [1–60].</p> <p>The keep-alive data can be "keep alive" in the form of string.</p> <p>Note: It is recommended to use the Heartbeat parameter instead of the keepalive parameter.</p>	20
heartbeat	int	O	<p>Server keep-alive; integer; Unit: s; The value range is [1–60].</p> <p>For example, if the parameter exists in the URL and the value is 5, the device will send "heartbeat" to the client as a keep-alive message.</p> <p>Note: The keep-alive message must be sent before the keepalive parameter expires.</p>	5

Request Example

http://192.168.1.108/cgi-bin/eventManager.cgi?action=attach&codes=[AlarmLocal%2CVideoMotion%2CVideoLoss%2CVideoBlind%2CFaceDetection]

Response Params (multipart , key=value format in body , Heartbeat in body)				
Name	Type	R/O	Description	Example
Code	string	R	Event code	VideoBlind
action	string	R	Event action Values: Start/Stop/pulse	Start
index	int	R	Number; meaningless sometimes.	0
data	object	O	Event data; json message body. For detailed message bodies, see the corresponding section.	""

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: text/plain
Content-Length: 39

```
Code=VideoMotion;action=Start;index=0
```

```
--myboundary
```

```
Content-Type: text/plain
```

```
Content-Length: 38
```

```
Code=FaceDetection;action=Start;index=0;data={
```

```
    "Faces": [ { "BoundingBox": [2992,136,6960,8192],  
              "Sex": "Man",  
              "Age": 40,  
              "Feature": [ "WearGlasses", "Smile"],  
              "Eye": 2,  
              "Mouth": 1,  
              "Mask": 1,  
              "Beard": 2  
            }, {...}, ... ]  
}
```

```
--myboundary
```

```
Content-Type: text/plain
```

```
Content-Length: 9
```

```
Heartbeat
```

```
--myboundary
```

```
.....
```

4.9.18 Get Capability of Event Management

Request URL	http://<server>/cgi-bin/eventManager.cgi?action=getCaps		
Method	GET		
Request Params (none)			
Name	Type	R/O	Description
Request Example			
http://192.168.1.108/cgi-bin/eventManager.cgi?action=getCaps			

Response Params (key=value format in body)			
Name	Type	R/O	Description
caps	object	O	capabilities
+TimeSectionEnable	bool	O	default false
+RecordEnable	bool	O	default false
+RecordCloudEnable	bool	O	default false
+AlarmOutEnable	bool	O	default false
+PtzLinkEnable	bool	O	default false
+SnapshotEnable	bool	O	enable snapshot

+SnapshotCloudEnable	bool	O	default false;	false
+SnapshotTimes	uint32[2]	O	The first element represents the minimum value The second element represents the maximum value If this option is not available, it indicates no restrictions	[3, 10]
+MailEnable	bool	O	support Email	false
+BeepEnable	bool	O	support beep	false
+DejitterEnable	bool	O	support dejitter	false
+TipEnable	bool	O		false
+MonitorTourEnable	bool	O		false
+MMSEnable	bool	O	default true	false
+SMSEnable	bool	O	default false	false
+SupportAlarmBell	bool	O	default false	false
+SupportAccessControl	bool	O	default false	false
+SipCallEnable	bool	O	default false	false
+SupportAlarmServer	bool	O	DVR supports this feature by default; IPC does not	false
+SupportPtzLinkDelay	bool	O	default false	false
+SupportPSTNAlarmServer	bool	O	default false	false
+SupportICR	bool	O	default false	false
+BeepTime	bool	O	default false	false
+DejitterRange	object	O	Dejitter range	
++Min	int	O	unit is second	0
++Max	int	O	unit is second	60
+AlarmOutLatch	int[2]	O	Alarm output delay [0,0] for not support If this option is not available, use [1, 300] by default	[1, 300]
+RecordLatch	int[2]	O		[0, 300]
+VoiceEnable	bool	O	default false	true
+VoiceLinkTimeRange	int32[2]	O	The first element represents the minimum value The second element represents the maximum value	[10, 30]
+VoicePlayTimesRange	uint16[2]	O	The first element represents the minimum value The second element represents the maximum value	[1, 10]

+VoiceLinkFileOpt ional	bool	O	default true	true
+LogEnable	bool	O	default false	true
+SupportLightCon trol	bool	O	default false	false
+LinkLightBrightR ange	int32[2]	O	Range of lighting linkage brightness adjustment The first element represents the minimum value The second element represents the maximum value If this option is not available, it indicates that brightness adjustment is not supported	[0,100]
+SupportVideoMa trix	bool	O	default false	false
+LinkDetailCamer aEnable	bool	O	Does it support linkage with detail camera	false
+SupportHTTPUpl oad	object	O	Does it support reporting business data to the server through the HTTP protocol	
++PictureHttpUplo adEnable	bool	O	Does it support reporting picture data to the server through the HTTP protocol	false
++EventHttpUploa dEnable	bool	O	Does it support reporting event data to the server through the HTTP protocol	false
+SMSEnable	bool	O	whether support SMS	false

Response Example

```

caps.AlarmOutEnable=true
caps.BeepEnable=true
caps.DejitterEnable=true
caps.MMSEnable=true
caps.MailEnable=true
caps.MonitorTourEnable=true
caps.PtzLinkEnable=true
caps.RecordEnable=true
caps.SnapshotEnable=true
caps.TimeSectionEnable=true
caps.TipEnable=true
caps.RecordCloudEnable=true,
caps.SnapshotCloudEnable=false,
caps.SnapshotTimes[0]=3,
caps.SnapshotTimes[1]=10,
caps.SupportAlarmBell=false,
caps.SupportAccessControl=false,
caps.SipCallEnable=false,

```

```

caps.SupportAlarmServer=false,
caps.SupportPtzLinkDelay=false,
caps.SupportPSTNALarmServer=false,
caps.SupportICR=false,
caps.BeepTime=false,
caps.DejitterRange.Min=0,
caps.DejitterRange.Max=60
caps.AlarmOutLatch[0]=1,
caps.AlarmOutLatch[1]=300,
caps.RecordLatch[0]=0,
caps.RecordLatch[1]=300,
caps.VoiceEnable=true,
caps.VoiceLinkTimeRange[0]=10,
caps.VoiceLinkTimeRange[1]=30,
caps.VoicePlayTimesRange[0]=1,
caps.VoicePlayTimesRange[1]=10,
caps.VoiceLinkFileOptional=true,
caps.LogEnable=true,
caps.SupportLightControl=false,
caps.LinkLightBrightRange[0]=0,
caps.LinkLightBrightRange[1]=100,
caps.SupportVideoMatrix=false,
caps.LinkDetailCameraEnable=false,
caps.SupportDisableLinkage[0]=0
caps.SupportDisableLinkage[1]=0
caps.SupportDisableLinkage[2]=0
caps.SupportDisableLinkage[3]=0
caps.SupportHTTPUpload.PictureHttpUploadEnable=false
caps.SupportHTTPUpload.EventHttpUploadEnable=false

```

4.9.19 [Config] Net Alarm Event

Net alarm event configuration parameters:

Config Data Params				
Name	Type	R/O	Description	Example
NetAlarm	object[]	R	Network alarm; one-dimensional array. The array index represents the corresponding channel which starts from 0.	
+Enable	bool	O	Enable alarm input (bypass switch)	true
+DefenceAreaType	string	O	Type of the protection zone enumchar[32]{ "Intime": Real-time protection zone "Delay": Time-delay protection zone "Fullday": 24-hour protection zone Get the capability set to judge the	"Intime"

			protection zone type. }	
+DisableDelay	integer	O	Delay disarming time Unit: s Get the maximum delay time from the capability set. Effective when EnableAlways is selected or when the EnableControl is set to Normal. Others: Effective only when the protection zone type is Delay.	30
+EnableDelay	integer	O	Delay arming time Unit: s Get the maximum delay time from the capability set. Effective when the protection zone type is Delay.	30
+Name	string	O	Alarm Channel Name The sensor is solid on or off. enumchar[32]{ "NC" "NO" }	"Door" "NC"
+SensorType	string	O	Alarm linkage For linkage configurations, see SetEventHandler.	
+EventHandler	object	O		

Please refer to "4.2.1 Get and Set Configure" for configuration getting and setting. Specific examples are as follows :

Get Config Request Example

```
http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=NetAlarm
```

Get Config Response Example

```
table.NetAlarm[0].Enable=false
table.NetAlarm[0].Name=channel1
table.NetAlarm[0].SensorType=NO
table.NetAlarm[0].EventHandler= (output of EventHandler is described in GetEventHandler)
```

Set Config Request Example

```
http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&NetAlarm[0].Enable=true
```

Set Config Response Example

```
OK
```

4.9.20 Set Net Alarm State

Set the network alarm status.

Request URL	http://<server>/cgi-bin/netAlarm.cgi?action=setState
-------------	--

Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Sample
channel	int	R	Channel number; access control number; starting from 1	1
alarm	bool	R	Alarm status; bool; "true" means the alarm is enabled	true
name	string	O	Alarm name; string; less than 32 characters	101
trigger	string	O	Alarm triggering source; string; less than 32 characters. For example, Infrared, WaterSensor Manual and more	Remote
desc	string	O	Alarm description; string; less than 128 characters	Zone8

Request Example

http://192.168.1.108/cgi-bin/netAlarm.cgi?action=setState&channel=1&alarm=true&name=somke&trigger=SmokingSensor&desc=Zone8

Response Params (OK in body)

Response Example

OK

4.9.21 Get Supported Events

Get the event list which the device supports.

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

Response Params (key=value format in body)

Name	Type	R/O	Description	Sample
events	string[]	R	Supported event list; one-dimensional array	[VideoMotion, AlarmLocal, FaceDetection, VideoMotion,...]

Response Example

events[0]=VideoMotion
events[1]=AlarmLocal
events[2]=FaceDetection
events[3]=VideoMotion
...

4.9.22 [Config] Triger Http User List

Cooperate with TrigerHttp in event linkage to obtain username and password. This configuration requires a privacy processing process.

Config Data Params				
Name	Type	R/O	Description	Example
TrigerHttpUserList	object[]	R	user info of TrigerHttp	
+TrigerHttpName	char[]	R	user admin	"admin"
+TrigerHttpPassword	char[]	R	password	"admin"
+TrigerServerName	char[32]	R	linkage server name	"Server1"
+TrigerServerIP	char[32]	R	linkage server ip address	"0.0.0.0"
+TrigerServerPort	uint16	R	linkage server port	80
+TrigerHttpsEnable	bool	R	https enable	false

Please refer to "4.2.1 Get and Set Configure" for configuration getting and setting. Specific examples are as follows :

Get Config Request Example

```
http://10.0.0.8/cgi-bin/configManager.cgi?action=getConfig&name=TrigerHttpUserList
```

Get Config Response Example

```
table.TrigerHttpUserList[0].TrigerHttpName=admin  
table.TrigerHttpUserList[0].TrigerHttpPassword=admin  
table.TrigerHttpUserList[0].TrigerServerName=Server1  
table.TrigerHttpUserList[0].TrigerServerIP=0.0.0.0  
table.TrigerHttpUserList[0].TrigerServerPort=80  
table.TrigerHttpUserList[0].TrigerHttpsEnable=false  
table.TrigerHttpUserList[1].TrigerHttpName=admin  
...
```

Set Config Request Example

```
http://10.0.0.8/cgi-bin/configManager.cgi?action=setConfig&TrigerHttpUserList[0].TrigerHttpName=admin&TrigerHttpUserList[0].TrigerHttpPassword=admin
```

Set Config Response Example

```
OK
```

4.10 Record

4.10.1 Get Capability of Recording

Request URL	http://<server>/cgi-bin/recordManager.cgi?action=getCaps			
Method	GET			
Request Params (none)				
Name	Type	R/O	Description	Example
Request Example				
http://192.168.1.108/cgi-bin/recordManager.cgi?action=getCaps				

Response Params (key=value format in body)				
Name	Type	R/O	Description	Example
caps	object	R	Capabilities	
+PacketLengthRange	int[2]	O	Video packaging time range in minutes The first element of the array represents the minimum value, and the second element represents the maximum value. default [1, 60]	[1, 120]
+PacketSizeRange	int[2]	O	Range packaged by file length Unit: kb	[1024, 4096]
+SupportExtraRecordMode	bool	O	if support extra stream record mode default false	true
+SupportHoliday	bool	O	Does the recording plan support holiday function default false	true
+SupportPacketType	char[][16]	O	Supported video packaging methods: Time and Size	["Time", "Size"]
+SupportResumeTransmit	bool	O	Does support broken-point continuingly-transferring default false	true

Response Example

```

caps.MaxPreRecordTime=30
caps.PacketLengthRange[0]=1
caps.PacketLengthRange[1]=60
caps.PacketSizeRange[0]=131072
caps.PacketSizeRange[1]=2097152
caps.SupportExtraRecordMode=true
caps.SupportHoliday=true
caps.SupportPacketType[0]=Time
caps.SupportPacketType[1]=Size
caps.SupportResumeTransmit=false

```

4.10.2 [Config] Record Config

Config Data Params				
Name	Type	R/O	Description	Example
Record	object[]	O	record configuration array, video channel, starts from 0	
+TimeSection	char[8][24][29]	O	Array, the first dimension index range is [0 — 7], [0-6]: (Sunday - Saturday), 7: Holiday the second dimension index range is [0 — 23], time section table index.	[["15535 00:00:00-24:00:00", ...] 15535 [0:00:00-24:00:00], ...]

			Format: mask hh:mm:ss-hh:mm:ss Mask: [0—4294967295], hh: [0—24], mm: [0—59], ss: [0—59] Mask indicates record type by bits: Bit0: regular record Bit1: motion detection record Bit2: alarm record Bit3: card record	
+PreRecord	uint	O	Range is [0—300]. Prerecord seconds, 0 means no prerecord. channel number starts from 0	5

Please refer to “4.2.1 Get and Set Configure” for configuration getting and setting. Specific examples are as follows :

Get Config Request Example

```
http://10.0.0.8/cgi-bin/configManager.cgi?action=getConfig&name=Record
```

Get Config Response Example

```
table.Record[0].PreRecord=6
table.Record[0].TimeSection[1][0]=65535 00:00:00-24:00:00
table.Record[0].TimeSection[1][1]=0 02:00:00-24:00:00
table.Record[0].TimeSection[1][2]=0 03:00:00-24:00:00
table.Record[0].TimeSection[1][3]=0 04:00:00-24:00:00
table.Record[0].TimeSection[1][4]=0 05:00:00-24:00:00
table.Record[0].TimeSection[1][5]=0 06:00:00-24:00:00
```

Set Config Request Example

```
http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&Record[0].TimeSection[0][0]=6
00:00:00-23:59:59
```

Set Config Response Example

OK

4.10.3 [Config] Record Mode

Config Data Params				
Name	Type	R/O	Description	Example
RecordMode	object[]	O	array, the index represents video channel, which starts from 0.	
+Mode	uint8	O	Range is {0, 1, 2 }. 0: automatically record 1: manually record 2: stop record.	0

Please refer to “4.2.1 Get and Set Configure” for configuration getting and setting. Specific examples are as follows :

Get Config Request Example

```
http://10.0.0.8/cgi-bin/configManager.cgi?action=getConfig&name=RecordMode
```

Get Config Response Example

```
table.RecordMode[0].Mode=0
```

Set Config Request Example

```
http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&RecordMode[0].Mode=0
```

Set Config Response Example

```
OK
```

4.10.4 [Config] Media Global

Config Data Params				
Name	Type	R/O	Description	Example
MediaGlobal	object	O	Media global configuration	
+PacketType	uint8	O	package type 0- by time 1-by file length	0
+PacketLength	uint8	O	record package time length in minutes 1~255	60
+PacketSize	uint	O	record package size,unit:KB	1024
+LogRecord	bool	O	record log	false
+LogEncode	bool	O	Record encoding exception logs	false
+SnapFormatAs	char[16]	O	The range is {"MainFormat", "ExtraFormat", "Extra2Format", "Extra3Format"}	"MainFormat"

Please refer to “4.2.1 Get and Set Configure” for configuration getting and setting. Specific examples are as follows :

Get Config Request Example

```
http://10.0.0.8/cgi-bin/configManager.cgi?action=getConfig&name=MediaGlobal
```

Get Config Response Example

```
table.MediaGlobal.OverWrite=true,  
table.MediaGlobal.PacketType=0,  
table.MediaGlobal.PacketLength=60,  
table.MediaGlobal.PacketSize=1024,  
table.MediaGlobal.LogRecord=false,  
table.MediaGlobal.LogEncode=false,  
table.MediaGlobal.SnapFormatAs>MainFormat
```

Set Config Request Example

```
http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&MediaGlobal.SnapFormatAs>MainFormat
```

Set Config Response Example

```
OK
```

4.10.5 Find Media Files

Step 1 Create a media files finder.

Request URL	http://<server>/cgi-bin/mediaFileFind.cgi?action=factory.create		
Method	GET		
Request Params (none)			
Request Example			
http://192.168.1.108/cgi-bin/mediaFileFind.cgi?action=factory.create			

Response Params (key=value format in body)			
Name	Type	R/O	Description
result	string	R	objectid; use it to search for media files
Response Example			
result=08137			

Step 2 Start to find media files satisfied the conditions with the finder.

Request URL	http://<server>/cgi-bin/mediaFileFind.cgi?action=findFile		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
object	string	R	Search object of the media files; Create the object in the previous command, and then search for the object.
condition	object	R	user id
+Channel	int	R	Channel number; starting from 1
+StartTime	string	R	Start time
+EndTime	string	R	End time
+Dirs	string[]	O	The directory where the recordings are saved; array; If this parameter does not exist, the system will search for all directories.
+Types	string[]	O	The type of file that is being searched for; array; the index starts from 0; optional values: {"dav", "jpg", "mp4"} If this parameter does not exist, the system will search for all file types.
+Flags	string[]	O	Search parameter; array; the index starts from 0; optional values: {"Timing", "Manual", "Marker", "Event", "Mosaic", "Cutout"} If this parameter does not exist, the system will search for all files.
+Events	string[]	O	Events related to the recordings; array; the index starts from 0; optional

			values: {"AlarmLocal", "VideoMotion", "VideoLoss", "VideoBlind", "Traffic*", ...} If this parameter does not exist, the system will search for all files.	
+VideoStream	string	O	Stream type; optional values: {"Main", "Extra1", "Extra2", "Extra3"}. If this parameter does not exist, the system will search for all stream types.	Main

Request Example

http://192.168.1.108/cgi-bin/mediaFileFind.cgi?action=findFile&object=08137&condition.Channel=1&condition.Dirs[0]=/mnt/dvr/sda0&condition.Types[0]=dav&condition.Events[0]=AlarmLocal&condition.Events[1]=VideoMotion&condition.StartTime=2014-1-1%2012:00:00&condition.EndTime=2015-1-10%2012:00:00&condition.VideoStream>Main

Response Params (OK in body)

Response Example

OK

Step 3 Get the media file information found by the finder.

Request URL	http://<server>/cgi-bin/mediaFileFind.cgi?action=findNextFile			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Sample
object	string	R	The search object of the media files; Create the object in the previous command, and then search for the object.	08137
count	int	R	Get the number of files that have been found, and the maximum number is 100.	100

Request Example

http://192.168.1.108/cgi-bin/mediaFileFind.cgi?action=findNextFile&object=08137&count=100

Response Params (key=value format in body)

Name	Type	R/O	Description	Sample
found	int	O	The returned number of files that have been found. 0 means no file has been found.	1
items	object[]	O	List of file information	
+Channel	int	O	Channel number; integer; the video channel number starts from 0.	1
+StartTime	string	O	Start time of the recording	2011-1-1 12:00:00
+EndTime	string	O	End time of the recording	2011-1-1 13:00:00
+Type	string	O	File type	dav
+Events	string[]	O	Type of events that are related to the recordings	["AlarmLocal"]

+VideoStream	string	O	Stream type	Main
+FilePath	string	O	File path	/mnt/dvr/sda0/2010/8/1 1/dav/15:40:50.jpg
+Length	int	O	The file length within a specified period	792
+Duration	int	O	File length	3600

Response Example

```
found=1
items[0].Channel=1
items[0].StartTime=2011-1-1 12:00:00
items[0].EndTime=2011-1-1 13:00:00
items[0].Type=dav
items[0].Events[0]=AlarmLocal
items[0].VideoStream>Main
items[0].FilePath=/mnt/dvr/sda0/2010/8/11/dav/15:40:50.jpg
items[0].Length=790
items[0].Duration=3600
```

Step 4 Close the finder.

Request URL	http://<server>/cgi-bin/mediaFileFind.cgi?action=close			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Sample

Request Example

```
http://192.168.1.108/cgi-bin/mediaFileFind.cgi?action=close&object=08137
```

Response Params (OK in body)

Response Example

```
OK
```

Step 5 Destroy the finder.

Request URL	http://<server>/cgi-bin/mediaFileFind.cgi?action=destroy			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Sample

Request Example

```
http://192.168.1.108/cgi-bin/mediaFileFind.cgi?action=destroy&object=08137
```

Response Params (OK in body)

Response Example

OK

4.10.6 Find media files with FaceDetection info

Step 1 Create a media files finder.

This API is the same as the API in "4.10.5 Find Media Files".

Step 2 Start to find media files satisfied the common conditions and FaceDetection condition with the finder.

Start to find media files satisfied the common conditions and FaceDetection conditions with the finder.

Request URL	http://<server>/cgi-bin/mediaFileFind.cgi?action=findFile		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
object	string	R	Search object of the media files; Create the object in the previous command, and then search for the object.
condition	object	R	user id
+Channel	int	R	Channel number; starting from 1
+StartTime	string	R	Start time
+EndTime	string	R	End time
+Dirs	string[]	O	The directory where the recordings are saved; array; If this parameter does not exist, the system will search for all directories.
+Types	string[]	O	Search file types, should be "jpg"
+Flags	string[]	O	Search flags, can be : "Timing", "Manual", "Marker", "Event", "Mosaic", "Cutout" should include "Event"
+Events	string[]	O	Search event list, must be one string : "FaceDetection"
+DB	object	R	DB filter object
++FaceDetectionRecordFilter	object	R	DB filter for FaceDetection
+++ImageType	string	O	Picture type, if omit, means search all type. "GlobalSence": means big picture of the full sence, "Small" : means small picture of the people face.
+++Sex	string	O	Sex, it can be "Man", "Woman", if omit, search all
+++Age	int[]	O	Age range, ex: [25, 40]

+++Glasses	int	O	Glasses Status, 0: all, 1: not wear, 2: wear	1
+++Mask	int	O	Mask Status, 0: all, 1: not wearing mask, 2: wearing mask	1
+++Beard	int	O	Beard Status, 0: all, 1: no beard, 2: has beard	0

Request Example

http://192.168.1.108/cgi-bin/mediaFileFind.cgi?action=findFile&object=08137&condition.Channel=1&condition.StartTime=2014-1-1%2012:00:00&condition.EndTime=2015-1-10%2012:00:00&condition.Types[0]=jpg&condition.Flags[0]=Event&condition.Events[0]=FaceDetection&condition.DB.FaceDetectionRecordFilter.ImageType=GlobalSence&condition.DB.FaceDetectionRecordFilter.Sex=Man&condition.DB.FaceDetectionRecordFilter.Age[0]=25&condition.DB.FaceDetectionRecordFilter.Age[1]=40&condition.DB.FaceDetectionRecordFilter.Glasses=1

Response Params (OK in body)

Response Example

OK

Step 3 Get the media file information found by the finder

Request URL	http://<server>/cgi-bin/mediaFileFind.cgi?action=findNextFile			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Sample
object	string	R	The search object of the media files; Create the object in the previous command, and then search for the object.	08137
count	int	R	Get the number of files that have been found, and the maximum number is 100.	100

Request Example

http://192.168.1.108/cgi-bin/mediaFileFind.cgi?action=findNextFile&object=08137&count=100

Response Params (key=value format in body)

Name	Type	R/O	Description	Sample
found	int	O	The returned number of files that have been found. 0 means no file has been found.	1
items	object[]	O	List of file information	
+Channel	int	O	Channel number; integer; the video channel number starts from 0.	1
+StartTime	string	O	Start time of the recording	2011-1-1 12:00:00
+EndTime	string	O	End time of the recording	2011-1-1 13:00:00
+Type	string	O	File type	dav
+Events	string[]	O	Type of events that are related to the recordings	["FaceDetection"]
+FilePath	string	O	File path	/mnt/dvr/sda0/2010/8/1

				1/dav/15:40:50.jpg
+CutLength	int	O	File length that cut between start time and end time	
+Length	int	O	The file length within a specified period	792
+SummaryNew	object[]	R	DB record object	
++Key	string	R	DB record name, should be "FaceDetectionRecord"	"FaceDetectionRecord"
++Value	object	R	DB record value	
+++ImageType	string	O	Picture type. It can be: "GlobalSence" , "Small"	GlobalSence
+++TimeStamp	object	O	Picture timestamp	
++++UTC	int	O	UTC seconds	134652732
++++UTCMS	int	O	UTC miliseconds	134
+++Sex	string	O	Sex, it can be "Man", "Woman", "Unknown"	"Man"
+++Age	int	O	Age	30
+++Glasses	int	O	Glasses Status, 0: unknown, 1: not wear, 2: wear	1
+++Mask	int	O	Mask Status, 0: unknown, 1: not wearing mask, 2: wearing mask	2
+++Beard	int	O	Beard Status, 0: unknown, 1: no beard, 2: has beard	1

Response Example

```

found=100
items[0].Channel=1
items[0].StartTime=2011-1-1 12:00:00
items[0].EndTime=2011-1-1 13:00:00
items[0].Type=jpg
items[0].Events[0]=FaceDetection
items[0].FilePath=/mnt/dvr/sda0/2010/8/11/dav/15:40:50.jpg
items[0].CutLength=79000
items[0].SummaryNew[0].Key=FaceDetectionRecord
items[0].SummaryNew[0].Value.ImageType=GlobalSence
items[0].SummaryNew[0].Value.TimeStamp.UTC=134652732
items[0].SummaryNew[0].Value.TimeStamp.UTCMS=134
items[0].SummaryNew[0].Value.Sex=Man
items[0].SummaryNew[0].Value.Age=30
items[0].SummaryNew[0].Value.Glasses=1
items[0].SummaryNew[0].Value.Mask=2
items[0].SummaryNew[0].Value.Beard=1
...

```

Step 4 Close the finder.

This API is the same as the API in "4.10.5 Find Media Files".

Step 5 Destroy the finder.

This API is the same as the API in "4.10.5 Find Media Files".

4.10.7 Find media files with FaceRecognition info

Step 1 Create a media files finder.

This API is the same as the API in "4.10.5 Find Media Files".

Step 2 Start to find media files satisfied the common conditions and FaceRecognition conditions with the finder.

Request URL	http://<server>/cgi-bin/mediaFileFind.cgi?action=findFile			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Sample
object	char[]	R	Search object of the media files; Create the object in the previous command, and then search for the object.	08137
condition	object	R	user id	101
+Channel	int	R	Channel number; starting from 1	1
+StartTime	char[]	R	Start time	"2010-05-25 12:05:00"
+EndTime	char[]	R	End time	"2010-05-25 12:10:00"
+Dirs	char[][]	O	The directory where the recordings are saved; array; If this parameter does not exist, the system will search for all directories. Each dir path max string length is 259.	["/mnt/dvr/sda0", "/mnt/dvr/sda1"]
+Types	char[][]	O	Search file types, should be "jpg"	["jpg"]
+Flags	char[][]	O	Search flags, can be : "Timing", "Manual", "Marker", "Event", "Mosaic", "Cutout" should include "Event"	["Timing", "Event"]
+Events	char[][]	O	Search event list, must be one string : "FaceRecognition"	["FaceRecognition"]
+DB	object	R	DB filter object	
++FaceRecognitionRecordFilter	object	R	DB filter for FaceRecognition	
+++MachineAddress	char[]	O	Machine address	
+++StartTime	char[]	R	Start time to search, ex: "2010-05-25 12:05:00"	"2010-05-25 12:05:00"
+++EndTime	char[]	R	End time to search, ex: "2010-06-25 12:05:00"	"2010-06-25 12:05:00"
+++Person	object	O	Search condition of history person and the similar face group person candidates	
++++Name	char[]	O	Person's name, max string length is 15	

++++Sex	char[]	O	Sex, can be "Male", "Female", if omit, search all	
++++Birthday	char[]	O	Birthday, max string length is 11, ex: 1990-5-1	
++++Country	char[]	O	Country, ISO 3166, string length should be 2	
++++Province	char[]	O	Province, max string length is 63	
++++City	char[]	O	City, max string length is 63	
++++Certificate Type	char[]	O	Certificate Type. It can be: "IC", "Passport", "Unknown"	"Passport"
++++ID	char[]	O	Person ID of CertificateType, max string length is 31	
++++GroupID	char[]	O	The identity of the Face Group that this Person in. max string length is 63	
++++Age	int[2]	O	Age range, ex: [25, 40]	30
++++Glasses	int	O	Glasses Status, 0: all, 1: not wear, 2: wear	
++++Mask	int	O	Mask Status, 0: all, 1: not wearing mask, 2: wearing mask	1
++++Beard	int	O	Beard Status, 0: all, 1: no beard, 2: has beard	2
+++GroupID	char[][]	O	GroupID list	
+++SimilaryRange	int[2]	O	Similary Range, ex: [40, 100]	100

Request Example

http://192.168.1.108/cgi-bin/mediaFileFind.cgi?action=findFile&object=08137&condition.Channel=1&condition.StartTime=2014-1-1%2012:00:00&condition.EndTime=2015-1-10%2012:00:00&condition.Types[0]=jpg&condition.Flags[0]=Event&condition.Events[0]=FaceRecognition&condition.DB.FaceRecognitionRecordFilter.RegType=RecSuccess&condition.DB.FaceRecognitionRecordFilter.StartTime=2014-1-1%2012:00:00&condition.DB.FaceRecognitionRecordFilter.EndTime=2015-1-10%2012:00:00&condition.DB.FaceRecognitionRecordFilter.Person.Sex=Male&condition.DB.FaceRecognitionRecordFilter.Person.Country=CN&condition.DB.FaceRecognitionRecordFilter.Person.Age[0]=25&condition.DB.FaceRecognitionRecordFilter.Person.Age[1]=40&condition.DB.FaceRecognitionRecordFilter.Person.Glasses=1&condition.DB.FaceRecognitionRecordFilter.GroupID[0]=10001&condition.DB.FaceRecognitionRecordFilter.GroupID[1]=10003&condition.DB.FaceRecognitionRecordFilter.GroupID[2]=10005&condition.DB.FaceRecognitionRecordFilter.SimilaryRange[0]=40&condition.DB.FaceRecognitionRecordFilter.SimilaryRange[1]=100

Response Params (OK in body)

Response Example

OK

Step 3 Get the media file information found by the finder

Request URL	http://<server>/cgi-bin/mediaFileFind.cgi?action=findNextFile			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Sample

object	string	R	The search object of the media files; Create the object in the previous command, and then search for the object.	08137
count	int	R	Get the number of files that have been found, and the maximum number is 100.	100

Request Example

<http://192.168.1.108/cgi-bin/mediaFileFind.cgi?action=findNextFile&object=08137&count=100>

Response Params (key=value format in body)				
Name	Type	R/O	Description	Sample
found	int	O	The returned number of files that have been found. 0 means no file has been found.	1
items	object[]	O	List of file information	
+Channel	int	O	Channel number; integer; the video channel number starts from 0.	1
+StartTime	string	O	Start time of the recording	2011-1-1 12:00:00
+EndTime	string	O	End time of the recording	2011-1-1 13:00:00
+Type	string	O	File type	dav
+Events	string[]	O	Type of events that are related to the recordings	["FaceDetection"]
+FilePath	string	O	File path	/mnt/dvr/sda0/2010/8/11/dav/15:40:50.jpg
+CutLength	int	O	File length that cut between start time and end time	
+Length	int	O	The file length within a specified period	792
+SummaryNew	object[]	R	DB record object	
++Key	string	R	DB record name, should be "FaceRecognitionRecord"	
++Value	object	R	DB record value	
+++RecResult	int	O	Recognition result, 0 means recognition failed, no candidates. 1 means recognition success, has candidates.	
+++MachineAddress	string	O	Machine address, string max length is 259	
+++IsGlobalScene	bool	O	Is the global scene picture or not	
+++ImageInfo	object	O	Big picture info	
++++Length	int	O	The length of the picture	
++++FilePath	string	O	The file path of the picture, max string length is 259	
+++Object	object	O	The target face info	
++++Sex	string	O	Sex, it can be "Man", "Woman", "Unknown"	
++++Age	int	O	Age	

++++Glasses	int	O	Glasses Status, 0: unknown, 1: not wear, 2: wear	
++++Eye	int	O	Eye status, 0: not detected, 1: close eye, 2: open eye	
++++Mouth	int	O	Mouth status, 0: not detected, 1: close mouth, 2: open mouth	
++++Mask	int	O	Mask status, 0: not detected, 1: not wearing mask, 2: wearing mask	
++++Beard	int	O	Beard status, 0: not detected, 1: no beard, 2: has beard	
+++Candidates	object[]	O	The info of candidates face from face group.	
++++Similarity	int	O	Similarity	
++++Person	object	O	person info	
+++++Name	string	O	Person Name, max string length is 63	
+++++Sex	string	O	Sex, it can be "Male", "Female", "Unknown"	
+++++Birthday	string	O	The person's birthday, ex: "1980-01-01"	
+++++Country	string	O	Country name, length must be 2, value should be according to ISO3166	
+++++Province	string	O	Province name, max string length is 63	
+++++City	string	O	City name, max string length is 63	
+++++Certificate Type	string	O	Certificate Type. It can be: "IC", "Passport", "Unknown"	
+++++ID	string	O	Person ID of CertificateType, max string length is 31	
+++++FeatureState	int	O	Feature State, 0:Unknown, 1:Failed, 2:OK	

Response Example

```

found=100
items[0].Channel=1
items[0].StartTime=2011-1-1 12:00:00
items[0].EndTime=2011-1-1 13:00:00
items[0].Type=jpg
items[0].Events[0]=FaceRecognition
items[0].FilePath =/mnt/dvr/sda0/2010/8/11/dav/15:40:50.jpg
items[0].Length =790
items[0].SummaryNew[0].Key=FaceRecognitionRecord
items[0].SummaryNew[0].Value.RecResult=1
items[0].SummaryNew[0].Value.MachineAddress=AAA
items[0].SummaryNew[0].Value.IsGlobalScene=true
items[0].SummaryNew[0].Value.ImageInfo.Length=123
items[0].SummaryNew[0].Value.ImageInfo.FilePath=/tmp/1.jpg
items[0].SummaryNew[0].Value.Object.Sex=Man
items[0].SummaryNew[0].Value.Object.Age=40
items[0].SummaryNew[0].Value.Object.Glasses=1

```

```

items[0].SummaryNew[0].Value.Object.Eye=2
items[0].SummaryNew[0].Value.Object.Mouth=1
items[0].SummaryNew[0].Value.Candidates[0].Similarity=50
items[0].SummaryNew[0].Value.Candidates[0].Person.Name=ZhangSan
items[0].SummaryNew[0].Value.Candidates[0].Person.Birthday=1980-01-01
items[0].SummaryNew[0].Value.Candidates[0].Person.Sex=Male
items[0].SummaryNew[0].Value.Candidates[0].Person.Country=CN
items[0].SummaryNew[0].Value.Candidates[0].Person.Province=XXX
items[0].SummaryNew[0].Value.Candidates[0].Person.City=YYY
items[0].SummaryNew[0].Value.Candidates[0].Person.CertificateType=IC
items[0].SummaryNew[0].Value.Candidates[0].Person.ID=1234567890
items[0].SummaryNew[0].Value.Candidates[0].Person.FeatureState=0
...

```

Step 4 Close the finder.

This API is the same as the API in "4.10.5 Find Media Files".

Step 5 Destroy the finder.

This API is the same as the API in "4.10.5 Find Media Files".

4.10.8 Find media files with HumanTrait info

Step 1 Create a media files finder.

This API is the same as the API in "4.10.5 Find Media Files".

Step 2 Start to find media files satisfied the common conditions and HumanTrait conditions with the finder

Request URL	http://<server>/cgi-bin/mediaFileFind.cgi?action=findFile			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Sample
object	char[]	R	Search object of the media files; Create the object in the previous command, and then search for the object.	08137
condition	object	R	user id	101
+Channel	int	R	Channel number; starting from 1	1
+StartTime	char[]	R	Start time	"2010-05-25 12:05:00"
+EndTime	char[]	R	End time	"2010-05-25 12:10:00"
+Dirs	char[][]	O	The directory where the recordings are saved; array; If this parameter does not exist, the system will search for all directories. Each dir path max string length is 259.	["/mnt/dvr/sda0", "/mnt/dvr/sda1"]
+Types	char[][]	O	Search file types, should be "jpg"	["jpg"]
+Flags	char[][]	O	Search flags, can be : "Timing", "Manual", "Marker", "Event", "Mosaic",	["Timing", "Event"]

			"Cutout" should include "Event"	
+Events	char[]	O	Search event list, must be one string : " HumanTrait"	["HumanTrait"]
+DB	object	R	DB filter object	
++HumanTrait RecordFilter	object	R	DB filter for HumanTrait	
+++HumanAttributes	object	R	Human attributes	
++++CoatColor	array<string>	O	Coat color, refer to CoatColor in 16.1.3.5 [Event] HumanTrait for available values, max array size is 4	
++++CoatType	array<int>	O	Coat type, refer to CoatType in 16.1.3.5 [Event] HumanTrait for available values, max array size is 2	
++++Trousers Color	array<string>	O	Trousers color, refer to TrousersColor in 16.1.3.5 [Event] HumanTrait for available values, max array size is 4	
++++TrousersT ype	array<int>	O	Trousers type, refer to TrousersType in 16.1.3.5 [Event] HumanTrait for available values, max array size is 2	
++++HasHat	int	O	Has hat or not, 0: all, 1: not has hat, 2: has hat	
++++HasBag	int	O	Has bag or not, 0: all, 1: not has bag, 2: has bag	
++++Sex	string	O	Sex, can be "Man", "Woman", if omit, search all	
++++Age	array<int>	O	Age range, ex: [25, 40]	
++++HairStyle	int	O	Hair style, 0: all, 1: long hair, 2: short hair, 3: ponytail, 4: updo, 5: hiddened	

Request Example

http://192.168.1.108/cgi-bin/mediaFileFind.cgi?action=findFile&object=08137&condition.Channel=1&co
ndition.StartTime=2014-1-1%2012:00:00&condition.EndTime=2015-1-10%2012:00:00&condition.Types[
0]=jpg&condition.Flags[0]=Event&condition.Events[0]=HumanTrait&condition.DB.HumanTraitRecordFilt
er.HumanAttributes.CoatColor[0]=White&condition.DB.HumanTraitRecordFilter.HumanAttributes.CoatC
olor[1]=Yellow&condition.DB.HumanTraitRecordFilter.HumanAttributes.CoatType=1&condition.DB.Hum
anTraitRecordFilter.HumanAttributes.HasHat=2&condition.DB.HumanTraitRecordFilter.HumanAttribut
.Sex=Man&condition.DB.HumanTraitRecordFilter.HumanAttributes.Age[0]=30&condition.DB.HumanTrai
tRecordFilter.HumanAttributes.Age[1]=50&condition.DB.HumanTraitRecordFilter.HumanAttributes.Hair
Style=1

Response Params (OK in body)

Response Example

OK

Step 3 Get the media file information found by the finder

Request URL	http://<server>/cgi-bin/mediaFileFind.cgi?action=findNextFile			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Sample
object	string	R	The search object of the media files; Create the object in the previous command, and then search for the object.	08137
count	int	R	Get the number of files that have been found, and the maximum number is 100.	100
Request Example				
http://192.168.1.108/cgi-bin/mediaFileFind.cgi?action=findNextFile&object=08137&count=100				

Response Params (key=value format in body)				
Name	Type	R/O	Description	Sample
found	int	O	The returned number of files that have been found. 0 means no file has been found.	1
items	object[]	O	List of file information	
+Channel	int	O	Channel number; integer; the video channel number starts from 0.	1
+StartTime	string	O	Start time of the recording	2011-1-1 12:00:00
+EndTime	string	O	End time of the recording	2011-1-1 13:00:00
+Type	string	O	File type	dav
+Events	string[]	O	Type of events that are related to the recordings	["HumanTrait"]
+FilePath	string	O	File path	/mnt/dvr/sda0/2010/8/1 1/dav/15:40:50.jpg
+CutLength	int	O	File length that cut between start time and end time	
+Length	int	O	The file length within a specified period	792
+SummaryNew	object	R	Face info	
++Key	string	R	DB record name, should be "HumanTraitRecord"	"HumanTraitRecord"
++Value	object	R	DB record value	
+++HumanAttributes	object	O	Human attributes	
++++CoatColor	string	O	Coat color, refer to CoatColor in 16.1.3.5 [Event] HumanTrait for available values,	
++++CoatType	int	O	Coat type, refer to CoatType in 16.1.3.5 [Event] HumanTrait for available values,	
++++TrousersColor	string	O	Trousers color, refer to TrousersColor in 16.1.3.5 [Event] HumanTrait for available values,	
++++TrousersTy	int	O	Trousers type, refer to TrousersType in	

pe			<u>16.1.3.5 [Event] HumanTrait</u> for available values,	
++++HasHat	int	O	Has hat or not, 0: unknown, 1: not has hat, 2: has hat	1
++++HasBag	int	O	Has bag or not, 0: unknown, 1: not has bag, 2: has bag	2
++++Sex	string	O	Sex, can be "Man", "Woman", "Unknown"	"Woman"
++++Age	int	O	Age	34
++++HairStyle	int	O	Hair style, 0: unknown, 1: long hair, 2: short hair, 3: ponytail, 4: updo, 5: hiddened	3
++++HasUmbrella	int	O	Has umbrella or not, 0: unknown, 1: not has umbrella, 2: has umbrella	1
++++Bag	int	O	Bag type, 0: unknown, 1: handbag, 2: shoulder bag, 3: knapsack, 4: draw-bar box	2
++++Cap	int	O	Cap style, 0: unknown, 1: normal cap, 2: helmet	1
+++FaceAttributes	object	O	Face attributes	
++++Sex	string	O	Sex, can be "Man", "Woman", "Unknown"	"Woman"
++++Age	int	O	Age	
++++Mask	int	O	Mask status, 0: not detected, 1: not wearing mask, 2: wearing mask	
++++Beard	int	O	Beard status, 0: not detected, 1: no beard, 2: has beard	1
++++Glass	Int	O	Glasses Status, 0: unknown, 1: not wearing, 2: normal Glasses, 3: sun glasses, 4: black frame glasses	2
++++Emotion	string	O	Emotion info. It can be: "Unknown", "Smile", "Anger", "Sadness", "Disgust", "Fear", "Surprise", "Neutral", "Laugh", "Happy", "Confused", "Scream", "Lookaside"	"Unknown"
+++FacePath	string	O	Face picture path, max string length is 259	/mnt/2010/8/11/dav/15:40:50.jpg
+++FaceScenePath	string	O	Face scene picture path, max string length is 259	/mnt/2010/8/11/dav/15:40:51.jpg
+++HumanPath	string	O	Human picture path, max string length is 259	
+++HumanScenePath	string	O	Human scene picture path, max string length is 259	
Response Example				
found=100				
items[0].Channel=1				

```
items[0].StartTime=2011-1-1 12:00:00
items[0].EndTime=2011-1-1 13:00:00
items[0].Type=jpg
items[0].Events[0]=HumanTrait
items[0].FilePath=/mnt/dvr/sda0/2010/8/11/dav/15:40:50.jpg
items[0].Length=790
items[0].SummaryNew.Key=HumanTraitRecord
items[0].SummaryNew.Value.HumanAttributes.CoatColor=White
items[0].SummaryNew.Value.HumanAttributes.CoatType=1
items[0].SummaryNew.Value.HumanAttributes.TrousersColor=Black
items[0].SummaryNew.Value.HumanAttributes.TrousersType=1
items[0].SummaryNew.Value.HumanAttributes.HasHat=2
items[0].SummaryNew.Value.HumanAttributes.HasBag=1
items[0].SummaryNew.Value.HumanAttributes.Sex=Man
items[0].SummaryNew.Value.HumanAttributes.Age=30
items[0].SummaryNew.Value.HumanAttributes.HairStyle=2
items[0].SummaryNew.Value.HumanAttributes.HasUmbrella=1
items[0].SummaryNew.Value.HumanAttributes.Bag=0
items[0].SummaryNew.Value.HumanAttributes.Cap=2
items[0].SummaryNew.Value.FaceAttributes.Sex=Man
items[0].SummaryNew.Value.FaceAttributes.Age=35
items[0].SummaryNew.Value.FaceAttributes.Mask=0
items[0].SummaryNew.Value.FaceAttributes.Beard=1
items[0].SummaryNew.Value.FaceAttributes.Glass=2
items[0].SummaryNew.Value.FaceAttributes.Emotion=Smile
items[0].SummaryNew.Value.FacePath=/mnt/2010/8/11/dav/15:40:50.jpg
items[0].SummaryNew.Value.FaceScenePath=/mnt/2010/8/11/dav/15:40:51.jpg
...

```

Step 4 Close the finder.

This API is the same as the API in "4.10.5 Find Media Files".

Step 5 Destroy the finder.

This API is the same as the API in "4.10.5 Find Media Files".

4.10.9 Find media files with TrafficCar info

Step 1 Create a media files finder.

This API is the same as the API in "4.10.5 Find Media Files".

Step 2 Start to find media files satisfied the common conditions and TrafficCar conditions with the finder

Start to find media files satisfied the common conditions and TrafficCar conditions with the finder.

Note: some request params can use compare condition, it's value is an array, first item is a string of compare condition type, and the following items are compare values.

Compare condition type can be:

"==": means equal, followed with one param, if param type is string, then param value can contain some "*" to match any string.

"||": means equal one of the params, followed with one or more params.

"<>": means inside range, followed by two integer param,

><": means outside range, followed by two integer param,

Request URL	http://<server>/cgi-bin/mediaFileFind.cgi?action=findFile		
Method	GET		

Request Params (key=value format in URL)

Name	Type	R/O	Description	Sample
object	string	R	Search object of the media files; Create the object in the previous command, and then search for the object.	08137
condition	object	R	user id	101
+Channel	int	R	Channel number; starting from 1	1
+StartTime	string	R	Start time	"2010-05-25 12:05:00"
+EndTime	string	R	End time	"2010-05-25 12:10:00"
+Dirs	string[]	O	The directory where the recordings are saved; array; If this parameter does not exist, the system will search for all directories.	["/mnt/dvr/sda0", "/mnt/dvr/sda1"]
+Types	string[]	O	Search file types, should be "jpg"	["jpg"]
+Flags	string[]	O	Search flags, can be : "Timing", "Manual", "Marker", "Event", "Mosaic", "Cutout"	["Timing", "Event"]
+Events	string[]	O	Search event list Ignored, use Event under DB param.	
+DB	object	R	DB object	
++TrafficCar	object	R	DB filter for TrafficCar	
+++PlateNumber	<compare condition>	O	PlateNumber condition, use compare condition format, ex: ["==", "*888"]	["==", "*888"]
+++Event	<compare condition>	O	Event condition, ex: [" ", "TrafficGate", "Alarm*"]	[" ", "TrafficGate", "Alarm*"]
+++Speed	<compare condition>	O	Speed condition, ex: ["<>", 40, 80]	["<>", 40, 80]
+++PlateType	<compare condition>	O	Plate type condition, use compare condition format, ex: ["==", "Armed"],	["==", "Armed"]
+++PlateColor	<compare condition>	O	Plate color condition, use compare condition format, ex: ["==", "Blue"],	["==", "Blue"]
+++VehicleColor	<compare condition>	O	Vehicle color condition, use compare condition format, ex: ["==", "White"],	["==", "White"]

Request Example

```
http://192.168.1.108/cgi-bin/mediaFileFind.cgi?action=findFile&object=08137&condition.Channel=1&condition.StartTime=2014-1-1%2012:00:00&condition.EndTime=2015-1-10%2012:00:00&condition.Types[0]=jpg&condition.Flags[0]=Event&condition.DB.TrafficCar.PlateNumber[0]=%3d%3d&condition.DB.TrafficCar.PlateNumber[1]=%2a888&condition.DB.TrafficCar.Speed[0]=%3c%3e&condition.DB.TrafficCar.Sp
```

```
eed[1]=40&condition.DB.TrafficCar.Speed[2]=80&condition.DB.TrafficCar.VehicleColor[0]=%3d%3d&co  
ndition.DB.TrafficCar.VehicleColor[1]=White
```

Response Params (OK in body)

Response Example

OK

Step 3 Get the media file information found by the finder

Request URL	http://<server>/cgi-bin/mediaFileFind.cgi?action=findNextFile			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Sample
object	string	R	The search object of the media files; Create the object in the previous command, and then search for the object.	08137
count	int	R	Get the number of files that have been found, and the maximum number is 100.	100

Request Example

http://192.168.1.108/cgi-bin/mediaFileFind.cgi?action=findNextFile&object=08137&count=100

Response Params (key=value format in body)				
Name	Type	R/O	Description	Sample
found	int	O	The returned number of files that have been found. 0 means no file has been found.	1
items	object[]	O	List of file information	
+Channel	int	O	Channel number; integer; the video channel number starts from 0.	1
+StartTime	string	O	Start time of the recording	2011-1-1 12:00:00
+EndTime	string	O	End time of the recording	2011-1-1 13:00:00
+Type	string	O	File type	dav
+Events	string[]	O	Type of events that are related to the recordings	["TrafficJunction"]
+FilePath	string	O	File path	/mnt/dvr/sda0/2010/8/1 1/dav/15:40:50.jpg
+CutLength	int	O	File length that cut between start time and end time	
+Length	int	O	The file length within a specified period	792
+Summary	object	R	DB record object	
++TrafficCar	object	R	TrafficCar record info	
+++PlateNumber	string	R	Car plate number	A08888
+++PlateType	string	O	Plate type,	Unknown
+++PlateColor	string	O	Plate color, ex: "Yellow", "Blue", ... etc	Blue

+++VehicleColor	string	O	Vehicle color, ex: "Yellow", "Blue", ... etc	White
+++Country	string	O	Country info. max string length is 19	China
+++Speed	int	O	Vehicle speed, unit is km/hour	70
+++Event	string	O	The event info, ex: "TrafficJunction"	TrafficJunction

Response Example

```

found=100
items[0].Channel=1
items[0].StartTime=2011-1-1 12:00:00
items[0].EndTime=2011-1-1 13:00:00
items[0].Type=jpg
items[0].Events[0]=TrafficJunction
items[0].FilePath=/mnt/dvr/sda0/2010/8/11/dav/15:40:50.jpg
items[0].Length=790
items[0].Summary.TrafficCar.PlateNumber=A08888
items[0].Summary.TrafficCar.PlateType=Unknown
items[0].Summary.TrafficCar.PlateColor=Blue
items[0].Summary.TrafficCar.VehicleColor=White
items[0].Summary.TrafficCar.Country=China
items[0].Summary.TrafficCar.Speed=70
items[0].Summary.TrafficCar.Event=TrafficJunction
...

```

Step 4 Close the finder.

This API is the same as the API in "4.10.5 Find Media Files".

Step 5 Destroy the finder.

This API is the same as the API in "4.10.5 Find Media Files".

4.10.10 Find media files with IVS info

Step 1 Create a media files finder.

This API is the same as the API in "4.10.5 Find Media Files".

Step 2 Start to find media files satisfied the common conditions and IVS conditions with the finder

Request URL	http://<server>/cgi-bin/mediaFileFind.cgi?action=findFile			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Sample
object	string	R	Search object of the media files; Create the object in the previous command, and then search for the object.	08137
condition	object	R	user id	101
+Channel	int	R	Channel number; starting from 1	1

+StartTime	string	R	Start time	"2010-05-25 12:05:00"
+EndTime	string	R	End time	"2010-05-25 12:10:00"
+Dirs	string[]	O	The directory where the recordings are saved; array; If this parameter does not exist, the system will search for all directories.	["/mnt/dvr/sda0", "/mnt/dvr/sda1"]
+Types	string[]	O	Search file types	[".jpg"]
+Flags	string[]	O	Search flags, can be : "Timing", "Manual", "Marker", "Event", "Mosaic", "Cutout"	["Timing", "Event"]
+Events	string[]	O	Search event list	
+DB	object	R	DB object	
++IVS	object	R	DB filter for IVS	
			IVS rule condition. It can be: "CrossLineDetection" , "CrossRegionDetection" , "LeftDetection" , "WanderDetection"	CrossLineDetection
+++Rule	string	O	"MoveDetection" , "RioterDetection" , "CrossFenceDetection" , "TakenAwayDetection" , "PasteDetection" , "Preservation" , "StayDetection" , "TailDetection"	
+++Action	string	O	IVS action. It can be: "Appear" , "Disappear" , "Inside" , "Cross"	"Disappear"
+++ObjectType	array<string>	O	IVS object type, item in array can be: "Unknown" , "Human" , "Vehicle" , "NonMotor"	["Human", "Vehicle", "NonMotor"]

Request Example

http://192.168.1.108/cgi-bin/mediaFileFind.cgi?action=findFile&object=08137&condition.Channel=1&condition.StartTime=2014-1-1%2012:00:00&condition.EndTime=2015-1-10%2012:00:00&condition.Types[0]=jpg&condition.DB.IVS.Rule=CrossLineDetection&condition.DB.IVS.Action=Cross&condition.DB.IVS.ObjectType[0]=Human&condition.DB.IVS.ObjectType[1]=NonMotor

Response Params (OK in body)

Response Example

OK

Step 3 Get the media file information found by the finder

Request URL	http://<server>/cgi-bin/mediaFileFind.cgi?action=findNextFile			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Sample
object	string	R	The search object of the media files; Create the object in the previous command, and then search for the object.	08137
count	int	R	Get the number of files that have been	100

		found, and the maximum number is 100.
Request Example		
http://192.168.1.108/cgi-bin/mediaFileFind.cgi?action=findNextFile&object=08137&count=100		

Response Params (key=value format in body)				
Name	Type	R/O	Description	Sample
found	int	O	The returned number of files that have been found. 0 means no file has been found.	1
items	object[]	O	List of file information	
+Channel	int	O	Channel number; integer; the video channel number starts from 0.	1
+StartTime	string	O	Start time of the recording	2011-1-1 12:00:00
+EndTime	string	O	End time of the recording	2011-1-1 13:00:00
+Type	string	O	File type	dav
+Events	string[]	O	Type of events that are related to the recordings	["CrossLineDetection"]
+FilePath	string	O	File path	/mnt/dvr/sda0/2010/8/11/dav/15:40:50.jpg
+CutLength	int	O	File length that cut between start time and end time	
+Length	int	O	The file length within a specified period	792
+Summary	object	R	DB record object	
++IVS	object	R	IVS record info	
+++Rule	string	R	IVS rule, see above for valid value	CrossLineDetection
+++Action	string	O	IVS action, see above for valid value	Cross
+++ObjectType	string	O	IVS object type, see above for valid value	Human

Response Example

```

found=100
items[0].Channel=1
items[0].StartTime=2011-1-1 12:00:00
items[0].EndTime=2011-1-1 13:00:00
items[0].Type=jpg
items[0].Events[0]=CrossLineDetection
items[0].FilePath=/mnt/dvr/sda0/2010/8/11/dav/15:40:50.jpg
items[0].Length=790
items[0].Summary.IVS.Rule=CrossLineDetection
items[0].Summary.IVS.Action=Cross
items[0].Summary.IVS.ObjectType=Human
...

```

Step 4 Close the finder.

This API is the same as the API in "4.10.5 Find Media Files".

Step 5 Destroy the finder.

This API is the same as the API in "4.10.5 Find Media Files".

4.10.11 Find media files with NonMotor info

Step 1 Create a media files finder.

This API is the same as the API in "4.10.5 Find Media Files".

Step 2 Start to find media files satisfied the common conditions and NonMotor conditions with the finder

Request URL	http://<server>/cgi-bin/mediaFileFind.cgi?action=findFile			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Sample
object	string	R	Search object of the media files; Create the object in the previous command, and then search for the object.	08137
condition	object	R	user id	101
+Channel	int	R	Channel number; starting from 1	1
+StartTime	string	R	Start time	"2010-05-25 12:05:00"
+EndTime	string	R	End time	"2010-05-25 12:10:00"
+Dirs	string[]	O	The directory where the recordings are saved; array; If this parameter does not exist, the system will search for all directories.	["/mnt/dvr/sda0", "/mnt/dvr/sda1"]
+Types	string[]	O	Search file types	["jpg"]
+Flags	string[]	O	Search flags, can be : "Timing", "Manual", "Marker", "Event", "Mosaic", "Cutout"	["Timing", "Event"]
+Events	string[]	O	Search event list	
+DB	object	R	DB object	
++NonMotorRecordFilter	object	R	DB filter for NonMotor	
+++NumOfCycling	int	O	Number of people that cycling, value can be 1 ~ 3	2
+++Color	string	O	NonMotor color, can be "White" "Orange" "Pink" "Black" "Red" "Yellow" "Gray" "Blue" "Green" "Purple" "Brown" "Sliver" "Darkviolet" "Maroon" "Dimgray" "Whitesmoke" "Darkorange" "Mistyrose" "Tomato" "Olive" "Gold" "Darkolivegreen" "Chartreuse" "Greenyellow" "Forestgreen" "Seagreen" "Deepskyblue" "Cyan" "Other"	"Brown"
+++Category	string	O	NonMotor type, can be : "Non-Motor" "Bicycle" "Tricycle" "Motorcycle"	"Non-Motor"

			"DualTriWheelMotorcycle" "LightMotorcycle" "EmbassyMotorcycle" "MarginalMotorcycle" "AreaoutMotorcycle" "ForeignMotorcycle" "TrialMotorcycle" "CoachMotorcycle"	
+++Helmet	int	O	Helmet status, 0 : unknown, 1 : without helmet, 2 : with helmet	1

Request Example

```
GET http://192.168.1.108/cgi-bin/mediaFileFind.cgi?action=findFile&object=08137&condition.Channel=1&condition.StartTime=2014-1-1%2012:00:00&condition.EndTime=2015-1-10%2012:00:00&condition.Types[0]=jpg&condition.DB.NonMotorRecordFilter.NumOfCycling=2&condition.DB.NonMotorRecordFilter.Color=White&condition.DB.NonMotorRecordFilter.Category=Bicycle&condition.DB.NonMotorRecordFilter.Helmet=1
```

Response Params (OK in body)

Response Example

OK

Step 3 Get the media file information found by the finder

Request URL	http://<server>/cgi-bin/mediaFileFind.cgi?action=findNextFile			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Sample
object	string	R	The search object of the media files; Create the object in the previous command, and then search for the object.	08137
count	int	R	Get the number of files that have been found, and the maximum number is 100.	100

Request Example

```
http://192.168.1.108/cgi-bin/mediaFileFind.cgi?action=findNextFile&object=08137&count=100
```

Response Params (key=value format in body)

Name	Type	R/O	Description	Sample
found	int	O	The returned number of files that have been found. 0 means no file has been found.	1
items	object[]	O	List of file information	
+Channel	int	O	Channel number; integer; the video channel number starts from 0.	1
+StartTime	string	O	Start time of the recording	2011-1-1 12:00:00
+EndTime	string	O	End time of the recording	2011-1-1 13:00:00
+Type	string	O	File type	dav
+Events	string[]	O	Type of events that are related to the	["NonMotorDetect"]

			recordings	
+FilePath	string	O	File path	/mnt/dvr/sda0/2010/8/11/dav/15:40:50.jpg
+CutLength	int	O	File length that cut between start time and end time	
+Length	int	O	The file length within a specified period	792
+SummaryNew	object	R	DB record object	
++Key	string	R	DB record name, should be "NonMotorRecordFilter"	
++Value	object	R	DB record value	
+++NumOfCycling	int	O	Number of people that cycling, value can be 1 ~ 3	2
+++Color	string	O	NonMotor color, can be "White" "Orange" "Pink" "Black" "Red" "Yellow" "Gray" "Blue" "Green" "Purple" "Brown" "Sliver" "Darkviolet" "Maroon" "Dimgray" "Whitesmoke" "Darkorange" "Mistyrose" "Tomato" "Olive" "Gold" "Darkolivegreen" "Chartreuse" "Greencyellow" "Forestgreen" "Seagreen" "Deepskyblue" "Cyan" "Other"	White
+++Category	string	O	NonMotor type, can be : "Non-Motor" "Bicycle" "Tricycle" "Motorcycle" "DualTriWheelMotorcycle" "LightMotorcycle" "EmbassyMotorcycle" "MarginalMotorcycle" "AreaoutMotorcycle" "ForeignMotorcycle" "TrialMotorcycle" "CoachMotorcycle"	Bicycle
+++Helmet	int	O	Helmet status, 0 : unknown, 1 : without helmet, 2 : with helmet	1

Response Example

```

found=100
items[0].Channel=1
items[0].StartTime=2011-1-1 12:00:00
items[0].EndTime=2011-1-1 13:00:00
items[0].Type=jpg
items[0].Events[0]=NonMotorDetect
items[0].FilePath=/mnt/dvr/sda0/2010/8/11/dav/15:40:50.jpg
items[0].Length=790
items[0].SummaryNew.Key=NonMotorRecordFilter
items[0].SummaryNew.Value.NumOfCycling=2
items[0].SummaryNew.Value.Color=White
items[0].SummaryNew.Value.Category=Bicycle
items[0].SummaryNew.Value.Helmet=1

```

...

Step 4 Close the finder.

This API is the same as the API in "4.10.5 Find Media Files".

Step 5 Destroy the finder.

This API is the same as the API in "4.10.5 Find Media Files".

4.10.12 Searching for Media Files According to WorkClothesDetection

Step 1 Create a media files finder.

This API is the same as the API in "4.10.5 Find Media Files".

Step 2 Start to find media files satisfied the common conditions and NonMotor conditions with the finder

Request URL	http://<server>/cgi-bin/mediaFileFind.cgi?action=findFile		
Method	GET		
Request Params (key=value format at URL)			
Name	Type	R/O	Description
object	int32	R	Search object of the media files
condition	object	R	Search condition
+Channel	int32	R	Video channel number; integer; the video channel number starts from 1.
+StartTime	string	R	Start time of the search.
+EndTime	string	R	End time of the search.
+Types	array<string>	O	Type of search file. It should be jpg. ["jpg"]
+Flags	array<string>	O	Search condition, including Timing, Manual, Marker, Event, Mosaic, and Cutout. It should be Event. ["Event"]
+Events	array<string>	R	Search event list. It must be WorkClothesDetection. ["WorkClothesDetection"]
+Dirs	array<string>	O	Search directory list. If it does not exist, searches for all directories. ["/mnt/dvr/sda0", "/mnt/dvr/sda1"]
+DB	object	R	The filtering condition object of database.
++WorkClothesDetection	object	R	The filtering condition object of database for WorkClothesDetection.
+++Helmet	object	O	Safety helmet attribute. If this field does not exist, it means to search for attributes of all helmets.
++++HasHelmet	enumint8[6]	R	Wearing safety helmet. enumint8[]: 0: Unknown [1,2]

			1: No 2: Yes }	
++++HelmetColor	array<string>	R	Safety helmet color. For value range, see the following ColorEnum.	["Red", "Orange"]
++++IsCompliant	int32	O	Safety helmet alarm mode. 1: Mismatch attributes alarm; 2: Match attributes alarm.	1
+++Clothes	object	O	Work uniform attribute. If this field does not exist, it means to search for attributes of all work uniforms. It is optional.	
++++HasClothes	enumint8[]	R	Wearing work uniform. enumint8[]: 0: Unknown 1: No 2: Yes }	[1, 2]
++++ClothesColor	array<string>	R	Work uniform color. For value range, see the following ColorEnum.	["Red", "Orange"]
++++HasLegalClothes	int32	O	Work uniform alarm mode. 1: Mismatch attributes alarm; 2: Match attributes alarm.	1
++++TriggerMode	string	O	Detection mode. Attribute detection: TypeDetect; Registered database: CompareDetect.	"TypeDetect"
+++Mask	object	O	Face mask attribute. If this field does not exist, it means to search for attributes of all face masks.	
++++HasMask	int32	O	Wearing face mask. 1: No. 2: Yes.	1
+++Type	string	R	Wear type: Safety Helmet: Helmet. Work uniform: Clothes. Normal hat: NormalHat. Face mask: mask.	"Helmet"
+++NormalHat	object	O	Normal hat attribute. If this field does not exist, it means to search for attributes of all normal hats.	
++++HasNormalHat	enumint8[256]	O	enumint8[]: 0: Unknown 1: Not wearing normal hat 2: Wearing normal hat }	[1,2]
+++Glasses	object	O	Glasses attribute. If this field does not exist, it means to search for attributes of all glasses.	

++++HasGlasses	int32	O	Wearing glasses alarm mode. 0: Triggers an alarm when the target wears glasses. 1: Triggers an alarm when the target does not wear glasses.	0
----------------	-------	---	---	---

Request Example

GET

http://192.168.1.108/cgi-bin/mediaFileFind.cgi?action=findFile&object=08137&condition.Channel=1&condition.StartTime=2014-1-1%2012:00:00&condition.EndTime=2015-1-10%2012:00:00&condition.Types[0]=jpg&condition.Flags[0]=Event&condition.Events[0]=WorkClothesDetection&condition.DB.WorkClothesDetection.Helmet.HasHelmet[0]=1&condition.DB.WorkClothesDetection.Helmet.HelmetColor[0]=Red&condition.DB.WorkClothesDetection.Type=Helmet

Response Params (OK in body)

Name	Type	R/O	Description	Example
Response Example				
OK				

Step 3 Get the media file information found by the finder

Request URL	http://<server>/cgi-bin/mediaFileFind.cgi?action=findNextFile			
Method	GET			
Request Params (key=value format at URL)				
Name	Type	R/O	Description	Example
object	int32	R	Search object of the media files	08137
count	int32	R	The number of acquired results. It should be less than 100.	15
Request Example				
GET http://192.168.1.108/cgi-bin/mediaFileFind.cgi?action=findNextFile&object=08137&count=15				

Response Params (key=value format in body)

Name	Type	R/O	Description	Example
found	int32	R	The number of searched results.	100
items	object[]	R	Searched result array.	
+Channel	int32	O	Video channel number; integer; the video channel number starts from 0.	0
+StartTime	string	R	Start time of recording.	"2010-05-25 12:05:00"
+EndTime	string	R	End time of recording.	"2010-06-25 12:05:00"
+Type	string	O	File type. It should be jpg.	jpg
+Events	array<string>	O	Events related to the recordings.	["WorkClothesDetection"]
+FilePath	string	O	File path. It cannot exceed 259 characters.	"/mnt/2010/8/11/dav/15:40:50.jpg"
+CutLength	int32	O	The file length within a specified period.	79000
+Length	int32	O	The whole file length.	5536
+SummaryNew	object[]	R	Database record object.	

++Key	string	R	Database record name. It should be WorkClothesDetection.	"WorkClothesDetection"
++Value	object	O	Database record value.	
+++Helmet	object	O	Safety helmet attribute. Required.	
++++HasHelmet	enumint8	R	Wearing safety helmet enumint8{ 0: Unknown 1: No 2: Yes }	2
++++HelmetColor	string	R	Safety helmet color. For value range, see the following ColorEnum.	Red
+++Clothes	object	O	Safety helmet attribute.	
++++HasClothes	enumint8	R	Wearing work uniform. enumint8{ 0: Unknown 1: No 2: Yes }	2
++++ClothesColor	string	R	Work uniform color. For value range, see the following ColorEnum.	"Red"
+++HumanPath	string	R	Person thumbnail path.	"/picid/1.jpg"
+++HumanImageLength	uint32	R	Image size. Unit: Byte	123
+++HumanScenelImage	string	R	Panoramic image path corresponding to the person thumbnail.	"/picid/1.jpg"
+++HumanScenelImageLength	uint32	R	Image size. Unit: Byte.	123
+++HumanBoundingBox	Rect	R	Person enclosure box. Rectangular information.	[24, 16, 8152, 8144]
+++Glasses	object	O	Glasses attribute.	
++++GlassesType	enumint	R	Alarm type for glasses detection. enumint{ 0: No glasses 1. Sunglasses 2: Black-rimmed glasses 3: Half-frame glasses 4: Rimless glasses 5: Ordinary glasses 6: Industrial goggles }	1
++++HasGlasses	int32	R	Glasses detection results. 0: Compliant	0

		1: Non-compliance 2: Unknown.	
--	--	----------------------------------	--

Response Example

```

found=100
items[0].Channel=1
items[0].StartTime=2011-1-1 12:00:00
items[0].EndTime=2011-1-1 13:00:00
items[0].Type=jpg
items[0].Events[0]=WorkClothesDetection
items[0].FilePath=/mnt/dvr/sda0/2010/8/11/dav/15:40:50.jpg
items[0].CutLength=79000
items[0].Length=5536
items[0].SummaryNew[0].Key=WorkClothesDetection
items[0].SummaryNew[0].Value.Helmet.HasHelmet=2
items[0].SummaryNew[0].Value.Helmet.HelmetColor=Red
items[0].SummaryNew[0].Value.HumanPath=/picid/1.jpg
items[0].SummaryNew[0].Value.HumanImageLength=123
items[0].SummaryNew[0].Value.HumanScenelImage=/picid/1.jpg
items[0].SummaryNew[0].Value.HumanScenelImageLength=123
items[0].SummaryNew[0].Value.HumanBoundingBox[0]=24
items[0].SummaryNew[0].Value.HumanBoundingBox[1]=16
items[0].SummaryNew[0].Value.HumanBoundingBox[2]=8152
items[0].SummaryNew[0].Value.HumanBoundingBox[3]=8144

```

Step 4 Close the finder.

This API is the same as the API in "4.10.5 Find Media Files".

Step 5 Destroy the finder.

This API is the same as the API in "4.10.5 Find Media Files".

Appendix:

Name	ColorEnum	
Type	char[16]	
Parameter	Description	Remarks
"Other"	Unrecognized (unknown color)	(254,254,254) The value of 254 for R or G is dedicated to special colors, so be careful to use it separately.
"Unknown"	Unknown color (no detection, or it is not in the known color range).	(254,254,252)
"Black"	Black	0,0,0
"Silver"	Silver	192,192,192
"Gray"	Gray	128,128,128
"White"	White	255,255,255
"Maroon"	Chestnut	128,0,0
"Red"	Red	255,0,0
"Purple"	Purple	128,0,128
"Fuchsia"	Magenta (not supported)	255,0,255

"Green"	Green (the RGB value used by the algorithm for Green is 0, 255, 0).	0,128,0
"Lime"	Green (the algorithm names RGB value (0, 255, 0) as Green, and the standard name is Lime, so Lime can be treated as green in actual use).	0,255,0
"Olive"	Olive	128,128,0
"Yellow"	Yellow	255,255,0
"Navy"	Navy (not supported)	0,0,128
"Blue"	Blue	0,0,255
"Teal"	Navy (not supported)	0,128,128
"Aqua"	Aqua (not supported)	0,255,255
"Orange"	Orange (not supported)	255,165,0
"Pink"	Pink	255,192,203
"Brown"	Brown	165,42,42
"Darkviolet"	Dark violet	148,0,211
"Dimgray"	Dark gray	105,105,105
"Whitesmoke"	White smoke	245,245,245
"Darkorange"	Dark orange	255,140,0
"Mistyrose"	Light rosy	255,228,225
"Tomato"	Tomato red	255,99,71
"Gold"	Gold	255,215,0
"Darkolivegreen"	Dark olive green	85,107,47
"Chartreuse"	Yellow green	127,255,0
"Greenyellow"	Green yellow	173,255,47
"Forestgreen"	Forest green	34,139,34
"Seagreen"	Ocean green	46,139,87
"Deepskyblue"	Deep sky blue	0,191,255
"Cyan"	Cyan	0,255,255
"Lightgreen"	Light Green	144,238,144
"Orangered"	Orange red (not supported)	255,69,0

4.10.13 Download Media File with the File Name

Download a file by filename. The <**Filename**> in URL is got by chapter file finding.

Request URL	http://<server>/cgi-bin/RPC_Loadfile/< Filename >
Method	GET
Request Params (none)	
Request Example	
http://192.168.1.108/cgi-bin/RPC_Loadfile/mnt/sd/2015-01-08/001/dav/19/19.57.12-19.58.25[M][0@0][0].dav	

Response Params (binary data in body)
<binary data>: Binary data pack
Response Example

```

HTTP/1.1 200 OK
Server: Device/1.0
Content-Type: Application/octet-stream
Content-Length: xxxx

<file data>

```

4.10.14 Download Media File between Times

Download the media data between start time and end time.

Request URL	http://<server>/cgi-bin/loadfile.cgi?action=startLoad			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
channel	int	R	Channel number; starting from 1	1
startTime	string	R	Start time of the recording in the format of: yyyy-mm-dd hh:mm:ss	"2010-05-25 12:05:00"
endTime	string	R	End time of the recording in the format of: yyyy-mm-dd hh:mm:ss	"2010-05-25 12:10:00"
subtype	int	O	Stream type. If the parameter does not exist, the default value is 0. 0: Main stream 1: Sub stream 1 2: Sub stream 2	0
Types	string	O	The type of files that are being searched for	"dav"
Request Example				
http://192.168.1.108/cgi-bin/loadfile.cgi?action=startLoad&channel=1&startTime=2012-10-8%2013:00:01&endTime=2012-10-8%2014:00:01&subtype=0&Types=dav				

Response Params (binary data in body)

<binary data>: Binary data pack

Response Example

```

HTTP/1.1 200 OK
Server: Device/1.0
Content-Type: Application/octet-stream
Content-Length: xxxx

<file data>

```

4.10.15 Encrypted Download Media File with the File Name

Request URL	http://<server>/cgi-bin/RecordStreamInterleaved.cgi?action=attachStream			
Method	GET			
Request Params (key=value format in URL)				

Name	Type	R/O	Description	Example
path	char[]	R	name of media files which would be downloaded.	/mnt/sd/2019-07-01/001/dav/12/12.36.16-12.36.26[F][0@0][0].dav
password	char[]	O	password for encrypting media file data. if not set password, then use the preset password	"xxxxxxxx"
Request Example				
http://172.29.2.241/cgi-bin/RecordStreamInterleaved.cgi?action=attachStream&path=/mnt/sd/2019-07-01/001/dav/12/12.36.16-12.36.26[F][0@0][0].dav				

Response Params (binary in body)				
Name	Type	R/O	Description	Example
Response Example				
HTTP Code: 200 OK				
Content-Type: Application/octet-stream				
Content-Length: <fileLength>				
Body:				
<data>				
<data>				

4.10.16 Query Total Number of Alarms

Request URL	http://<server>/cgi-bin/api/mediaFileFind/getCount			
Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
object	int32	R	The finder object id	08137
Request Example				
{ "object": 08137 }				

Response Params (JSON format in body)				
Name	Type	R/O	Description	Example
count	uint32	R	total number	100
Response Example				
{ "count": 100 }				

4.11 Log

4.11.1 Find Logs

1. Whether or not found logs satisfied the conditions

Request URL	http://<server>/cgi-bin/log.cgi?action=startFind		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
condition	object	O	query condition
+StartTime	char[20]	O	the start time of log. Format is: yyyy-mm-dd hh:mm:ss.
+EndTime	char[20]	O	the end time of log. Format is: yyyy-mm-dd hh:mm:ss.
+Type	char[]	O	log type. The range is { "System", "Config", "Event", "Storage", "Account", "Data", "File", "CourseRecord" }.
+StartTimeRealUTC	char[32]	O	Indicates the UTC time of log start, in the format: yyyy-mm-ddThh:mm:ssZ. And (StartTimeRealUTC, EndTimeRealUTC) are mutually exclusive; If both exist, use RealUTC time first.
+EndTimeRealUTC	char[32]	O	Indicates the UTC time of log end, in the format: yyyy-mm-ddThh:mm:ssZ. And (StartTimeRealUTC, EndTimeRealUTC) are mutually exclusive; If both exist, use RealUTC time first.
Request Example			
http://192.168.1.108/cgi-bin/log.cgi?action=startFind&condition.StartTime=2011-1-1%2012:00:00&condition.EndTime=2011-1-10%2012:00:00&condition.StartTimeRealUTC=2011-01-01T04:00:00Z&condition.EndTimeRealUTC=2011-01-10T04:00:00Z			

Response Params (key=value format in body)				
Name	Type	R/O	Description	Example
token	uint	O	Retrieved query token If the token is greater than 0, it indicates that the log was found, otherwise it indicates that the log was not found.	1
count	uint	O	The logcount for the search condition.	100
Response Example				
token=1 count=100				

2. Get the particular number of logs

Request URL	http://<server>/cgi-bin/log.cgi?action=doFind			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
token	uint	R	query token	46878
count	uint	O	the count of logs for this query.	2
Request Example				
http://192.168.1.108/cgi-bin/log.cgi?action=doFind&token=1&count=100				

Response Params (key=value format in body)				
Name	Type	R/O	Description	Example
found	uint	O	Count of found log, found is 0 if no log is found.	2
items	object[]	O	items	
+RecNo	uint	O	Log number.	789
+Time	char[32]	O	Time of this log.	2011-05-20 11:59:21
+TimeRealUTC	char[32]	O	the UTC time of log, in the format: yyyy-mm-ddThh:mm:ssZ	2011-05-20T03:59:21Z
+Type	char[]	O	Log type.	"Account.Login"
+User	char[]	O	User name.	System
+Detail	object	O	Log details.	
++Compression	char[]	O	compression	H.264->MJPG
++Data	char[]	O	data	Encode
Response Example				
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].TimeRealUTC=2011-05-20T03:59:21Z items[1].Type=SaveConfig items[1].User=System				

3. Stop query logs

Request URL	http://<server>/cgi-bin/log.cgi?action=stopFind			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
token	uint	R	query token	1
Request Example				

http://192.168.1.108/cgi-bin/log.cgi?action=stopFind&token=1

Response Params (OK in body)

Response Example

OK

4.11.2 Clear All the Logs

Request URL	http://<server>/cgi-bin/log.cgi?action=clear			
Method	GET			
Request Params (none)				
Name	Type	R/O	Description	Example
Request Example				http://192.168.1.108/cgi-bin/log.cgi?action=clear

Response Params (OK in body)

Response Example

OK

4.11.3 Backup Logs

Request URL	http://<server>/cgi-bin/Log.backup?action>All			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
condition	object	O		
+StartTime	char[]	O	the start time of log. 24 hour Format, as: yyyy-mm-dd hh:mm:ss.	2014-8-25 00:02:32
+EndTime	char[]	O	the end time of log. 24 hour Format, as: yyyy-mm-dd hh:mm:ss.	2014-8-25 01:02:32
Request Example				http://192.168.1.108/cgi-bin/Log.backup?action>All&condition.StartTime=2014-8-25%2000:02:32&condition.EndTime=2020-8-25%2001:02:32

Response Params (binary in body)

Name	Type	R/O	Description	Example
Response Example				
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

...

4.11.4 Seek Find Logs

Request URL	http://<server>/cgi-bin/Log.cgi?action=doSeekFind			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
token	uint	R	query token	1
offset	uint	R	offset	2
count	uint	O	the count of logs for this query.	100
Request Example				
http://192.168.1.108/cgi-bin/Log.cgi?action=doFind&token=1&count=100				

Response Params (key=value format in body)				
Name	Type	R/O	Description	Example
found	uint	O	Count of found log, found is 0 if no log is found.	2
items	object[]	O	items	
+RecNo	uint	O	Log number.	789
+Time	char[32]	O	Time of this log.	2011-05-20 11:59:21
+TimeRealUTC	char[32]	O	the UTC time of log, in the format: yyyy-mm-ddThh:mm:ssZ	2011-05-20T03:59:21Z
+Type	char[]	O	Log type.	"Account.Login"
+User	char[]	O	User name.	System
+Detail	object	O	Log details.	
++Compression	char[]	O	compression	H.264->MJPG
++Data	char[]	O	data	Encode

Response Example

```
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].TimeRealUTC=2011-05-20T03:59:21Z
items[1].Type=SaveConfig
items[1].User=System
```

4.11.5 Export Encryped Log

The binary data should be saved as .zip format.

Request URL	http://<server>/cgi-bin/Log.exportEncrypedLog?action=All		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
condition	object	O	
+StartTime	char[]	O	the start time when log info built. 24 hour Format, as: yyyy-mm-dd hh:mm:ss
+EndTime	char[]	O	the end time when log info built. 24 hour Format, as: yyyy-mm-dd hh:mm:ss
+Password	char[]	R	the input parameter for encrypted log in zip format.
Request Example			
http://192.168.1.108/cgi-bin/Log.exportEncrypedLog?action=All&condition.StartTime=2014-8-25%2000:02:32&condition.EndTime=2020-8-25%2001:02:32&condition.Password=12345			

Response Params (binary in body)				
Name	Type	R/O	Description	Example
Response Example				
HTTP/1.1 200 OK				
Transfer-Encoding: chunked				
CONNECTION: keep_alive				
Content-type: application/binarytet-stream; charset=utf-8				
The binary data of encrypted log in zip format				

4.11.6 [Config] Serial Port Log Redirection

Config Data Params				
Name	Type	R/O	Description	Example
DebugInfoRedir	object	O	The configuration of serial port log redirection.	
+SerialPortLogLevel	uint8	O	logs which need to be redirected. (default is 4) 0: Fatal error; 1: Normal ; 2: Warning;3: Info, 4: The invoke path and data flow tracking in debug level 1 ; 5: The key logic tracking in debug level 2; 6: Other detail information in debug level 3; 7: Debug info for legacy compatible)	4
+SerialPortLogType	uint8	O	log type which need to be redirected. (0: main chip logs; 1: main chip + sub-chip	0

		logs)
--	--	-------

Please refer to "4.2.1 Get and Set Configure" for more info about the get/set operation

Get Config Request Example

```
http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&name=DebugInfoRedir
```

Get Config Response Example

```
table.DebugInfoRedir.SerialPortLogLevel=3
```

```
table.DebugInfoRedir.SerialPortLogType=1
```

Set Config Request Example

```
http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&DebugInfoRedir.SerialPortLogLevel=6&DebugInfoRedir.SerialPortLogType=1
```

Set Config Response Example

```
OK
```

4.12 Upgrader

4.12.1 Strat to Upgrade

Use this message to upload the firmware, and when the device receiving all the data successfully, it will start to upgrade the device, and then use the getState method to get the state.

Request URL	http://<server>/cgi-bin/upgrader.cgi?action=uploadFirmware		
Method	POST		
Request Params (form-data in body)			
Name	Type	R/O	Description
Request Example			
POST /cgi-bin/upgrader.cgi?action=uploadFirmware HTTP/1.1 Host: 192.168.1.108 Connection: keep-alive Content-Type: multipart/form-data; boundary=-----8655433224198 Content-Length: xxxxxxxx -----8655433224198 Content-Disposition:form-data;name="upgrade"; filename= "xxxxxx.bin" Content-Type: application/octet-stream Firmware data.... -----8655433224198--			

Response Params (OK in body)	
Name	Type
Response Example	
OK	

4.12.2 Get Upgrade State

Request URL	http://<server>/cgi-bin/upgrader.cgi?action=getState			
Method	GET			
Request Params (none)				
Name	Type	R/O	Description	Example
Request Example				http://192.168.1.108/cgi-bin/upgrader.cgi?action=getState

Response Params (key=value format in body)				
Name	Type	R/O	Description	Example
state	object	O	state info	
+State	char[]	O	the state of the upgrade, it can be Preparing, Downloading, DownloadFailed, Upgrading, Invalid, Failed, Succeeded, Cancelled, NotEnoughMemory.	Preparing
+Progress	uint	O	the progress of the upgrade. 0~100	20
Response Example				state.State=Upgrading state.Progress=45

4.12.3 Set upgrader url

Request URL	http://<server>/cgi-bin/upgrader.cgi?action=updateFirmwareByUrl			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
Url	char[]	R	Upgrade package address, also support ftp address	https://aaa/bbb/ccc/license.bin
checkType	int	O	check method enum int8{ 0: MD5 }	0
checkSum	char[]	O	Upgrade package checksum	f38ad920
Request Example				http://<server>/cgi-bin/upgrader.cgi?action=updateFirmwareByUrl=https://aaa/bbb/ccc/license.bin&checkType=0&checkSum=f38ad920

Response Params (OK in body)				
Name	Type	R/O	Description	Example
Response Example				OK

4.12.4 Cancel Upgrade

Request URL	http://<server>/cgi-bin/upgrader.cgi?action=cancel			
Method	GET			
Request Params (none)				
Name	Type	R/O	Description	Example
Request Example				
http://<server>/cgi-bin/upgrader.cgi?action=cancel				

Response Params (OK in body)				
Name	Type	R/O	Description	Example
Response Example				
OK				

4.12.5 Checking Cloud Update Version

Check cloud update version

Request URL	http://<server>/cgi-bin/api/CloudUpgrader/check			
Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
way	int32	R	Checking methods. enumint8{ 0: Updating by updating server through direct connection. 1: Checking through proxy server. 2: Detecting by acquiring cached test results. }	0
proxy	object	O	Proxy server address, valid when way==1 (optional).	
+IP	char[40]	R	Network address	"10.1.2.3"
+Port	int32	O	Port	8080
Request Example				
<pre>{ "way" : 0, "proxy" : { "IP" : "10.1.2.3", "Port" : 8080 } }</pre>				

Response Params (JSON format in body)				
Name	Type	R/O	Description	Example
info	object	O	Check results	

+State	enumchar[12]	R	<p>Update status.</p> <p>enumchar[12]{</p> <p>"None": No update is detected</p> <p>"Regular": Regular update (It requires user confirmation, and can only be updated to a later version).</p> <p>"Emergency": Mandatory update (The device automatically performs detection and update, and it can also degrade to an earlier version).</p> <p>"Automatic": Automatic update (The device automatically updates when a new version is available. It is currently for custom use and needs to be enabled)</p> <p>}</p>	"None"
+PackageType	enumchar[32]	O	<p>Update package type of new version.</p> <p>When State is not None, return enumchar[32]{</p> <p>"all": All package.</p> <p>"ptz": PTZ main control package.</p> <p>"web"</p> <p>"logo"</p> <p>"custom"</p> <p>"gui"</p> <p>"pd"</p> <p>"data"</p> <p>"ptz_power": PTZ power.</p> <p>"ptz_light": PTZ light.</p> <p>"ptz_heater": PTZ heater.</p> <p>}</p>	"all"
+OldVersion	char[64]	O	Old version, which needs to be returned when State is not None.	"0000"
+NewVersion	char[64]	O	New version, which needs to be returned when State is not None.	"0004"
+Attention	char[2048]	O	Updated content of the new update package.	"What is new"
+PackageUrl	char[256]	O	Download address of update package (required for agent upgrade).	"https://example.com/1.zip"
+Packageld	char[64]	O	Update package ID	"1d2ee7"
+CheckSum	char[64]	O	SHA-256 checksum of the update package	"F3D288AB"
+BuildTime	char[24]	O	Build time of update package	"08-10-2018 01:01:02"
Response Example				
{	<pre>"info" : {</pre>			

```

        "State" : "None",
        "PackageType" : "all",
        "OldVersion" : "0000",
        "NewVersion" : "0004",
        "Attention" : "What is new",
        "PackageUrl" : "https://example.com/1.zip",
        "Packageld" : "1d2ee7",
        "CheckSum" : "F3D288AB"
        "BuildTime" : "08-10-2018 01:01:02"
    }
}

```

4.12.6 Performing Online Update

Perform online update

Request URL	http://<server>/cgi-bin/api/CloudUpgrader/execute			
Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
NewVersion	char[64]	O	If the cloud version is newer than the new version obtained in the last check, the update fails.	"2.42.00.001"
way	enumint8	R	Update method enumint8{ 0: Updating by updating server through direct connection. 1: Updating by proxy server }.	0
proxy	object	O	Proxy server address, which is valid when way=1 (optional).	
+IP	char[40]	O	Network address	"10.1.2.3"
+Port	uint16	O	Port	8080
info	object	O	Update package information, which is needed when way==1 (optional).	
+PackageUrl	char[256]	O	Download address of the update package (required for agent upgrade).	"https://example.com/1.zip"
+Packageld	char[64]	O	Update package ID	"1d2ee7"
+Checksum	char[64]	O	SHA-256 checksum of the	"F3D288AB"

		update package		
Request Example				
{ "NewVersion" : "2.42.00.001", "way" : 0, "proxy" : { "IP" : "10.1.2.3", "Port" : 8080 }, "info" : { "PackageUrl" : "https://example.com/1.zip", "PackageId" : "1d2ee7", "CheckSum" : "F3D288AB" } }				
Response Params (JSON format in body)				
Name	Type	R/O	Description	Example
Response Example				
{}				

4.12.7 Canceling Online Update

Cancel the online update during the download process. If you have already started writing Flash, you cannot cancel the update.

Request URL	http://<server>/cgi-bin/api/CloudUpgrader/cancel			
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
Response Example				
{}				

4.13 Http Uploading

4.13.1 [Config] Active Image and Event Uploading

Configure the device to upload images and events to the specified address. Before configuration, the client should monitor in the specified address. Once the configuration is complete, the device would

upload images and event data to this address using the command described in chapter " 4.13.2 Active Image and Event Uploading ".

The parameters for uploading image and event are as follows:

Config Data Params				
Name	Type	R/O	Description	Example
PictureHttpUpload	object	R	The parameters for uploading image and event	
+Enable	bool	R	Enable or not	true
+Type	char[16]	O	Authentication "basic": HTTP Basic Authentication "digest": HTTP Digest Authentication	"digest"
+UploadServerList	object[]	R	The list of servers that receives uploaded information	
++Address	char[128]	R	IP address or domain name of the server	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	Code list for uploaded event	["CrossLineDetection", "FaceDetection"]
++HttpsEnable	bool	O	https enable	true
++AuthEnable	bool	O	authentication enable	true

Please refer to "4.2.1 Get and Set Configure" for configuration getting and setting. Specific examples are as follows :

Get Config Request Example

```
http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=PictureHttpUpload
```

Get Config Response Example

```
table.PictureHttpUpload.Enable=true
table.PictureHttpUpload.Type=digest
table.PictureHttpUpload.UploadServerList[0].Address=192.168.1.208
table.PictureHttpUpload.UploadServerList[0].Port=80
table.PictureHttpUpload.UploadServerList[0].UserName=abc
table.PictureHttpUpload.UploadServerList[0].Password=123
table.PictureHttpUpload.UploadServerList[0].Uploadpath=/example/handlepic.php
table.PictureHttpUpload.UploadServerList[0].EventType[0]=CrossLineDetection
table.PictureHttpUpload.UploadServerList[0].EventType[1]=FaceDetection
```

Set Config Request Example

```
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]=CrossLineDetection&PictureHttpUpload.UploadServerList[0].EventType[1]=FaceDetection
```

reHttpUpload.UploadServerList[0].EventType[1]=FaceDetection

Set Config Response Example

OK

4.13.2 Active Image and Event Uploading

The device uploads images and event data to the specified address based on the client configuration. For details on the parameters for each event, please refer to the corresponding [Event] chapters. The IP address, port and URL of the uploaded target server is specified by the PictureHttpUpload. Each set of images and events are sent within one separate HTTP request, in which multiple of images and events are contained. It is uploaded through multipart.

Request URL	http://<Address>:<Port>/<Uploadpath_of_PictureHttpUpload>		
Method	POST		
Request Params (multipart ; JSON in body; binary data in body)			
Name	Type	R/O	Description
Channel	int	R	video channel, which start from 0
Time	char[32]	R	Snap picture time "2022-03-30 15:40:01"
Events	object[]	R	Event Information list
+Code	char[32]	R	Event Code "FaceRecognition"
+Action	char[16]	R	Event action, with the values of "Start", "Stop" "Pulse". "Pulse"
+Index	int	R	Event channel number, starting from 0. 0
+Data	object	R	For specific parameters of each event, please refer to the corresponding chapters for reference.

Request Example

POST http://192.168.1.208/example/handlepic.php HTTP/1.1

User-Agent: Device/1.0

Content-Type: multipart/x-mixed-replace; boundary=<boundary>

Connection: closed

--<boundary>

Content-Type: application/json

Content-Length: <data length>

{

```
    "Channel" : 0,  
    "Time" : "2022-03-30 15:40:01"  
    "Events" : [ {  
        "Code" : "FaceRecognition",  
        "Action" : "Pulse",  
        "Index" : 0  
        "Data" : {  
            "UTC" : 123456789,  
            .....  
        }  
    }
```

```

    }, ..., {}]
}

--<boundary>
Content-Type: image/jpeg
Content-Length: <data length>

<jpeg data>
--<boundary>
...
--<boundary>--

```

Response Params (OK in body)

Response Example

OK

4.13.3 [Config] Active Event Uploading

Configure the device to upload event (without image) to the specified address. Before configuration, the client should monitor in the specified IP address. Once the configuration is complete, the device would upload event data to this address using the command described in chapter " 4.13.4 Active Event Uploading ".

The parameters for uploading event are as follows:

Config Data Params				
Name	Type	R/O	Description	Example
EventHttpUpload	object	R	Parameters for uploading event	
+Enable	bool	R	Whether to enable the function	true
+Type	char[32]	O	Authentication Type "basic":HTTP Basic Authentication "digest" :HTTP Digest Authentication	"digest"
+UploadServerList	object[]	R	The server list that receives uploaded information	
++Address	char[128]	R	IP address or domain name of the server	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/handleevt.php"
++EventType	char[128][40]	O	event types to be uploaded(Only applicable to cameras)	["CrossLineDetection", "face Detection"]
++HttpsEnable	bool	O	https enable	true
++AuthEnable	bool	O	authentication enable	true

Please refer to "4.2.1 Get and Set Configure" for configuration getting and setting. Specific examples are as follows :

Get Config Request Example

```
http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=EventHttpUpload
```

Get Config Response Example

```
table.EventHttpUpload.Enable=true
table.EventHttpUpload.Type=digest
table.EventHttpUpload.UploadServerList[0].Address=192.168.1.208
table.EventHttpUpload.UploadServerList[0].Port=80
table.EventHttpUpload.UploadServerList[0].UserName=abc
table.EventHttpUpload.UploadServerList[0].Password=123
table.EventHttpUpload.UploadServerList[0].Uploadpath=/example/handleevt.php
table.EventHttpUpload.UploadServerList[0].EventType[0]=CrossLineDetection
table.EventHttpUpload.UploadServerList[0].EventType[1]=faceDetection
```

Set Config Request Example

```
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]=CrossLineDetection&EventHttpUpload.UploadServerList[0].EventType[1]=faceDetection
```

Set Config Response Example

```
OK
```

4.13.4 Active Event Uploading

The device upload event data (without image) to the specified address based on the configuration of the client. For the specific parameter of each event, please refer to the corresponding [Event] chapters. The IP address, port and URL of the upload target server is specified by EventHttpUpload. Each event is sent within one separate HTTP request.

Request URL	http://<Address>:<Port>/<Uploadpath_of_EventHttpUpload>			
Method	POST			
Request Params (JSON in body)				
Name	Type	R/O	Description	Example
Code	char[32]	R	Event code	"FaceRecognition"
Action	char[16]	R	"Event action, with the values of "Start", "Stop""Pulse".	"Pulse"
Index	int	R	Event channel number, starting from 0	0
Data	object	R	For the specific parameters for each event, please refer to the corresponding chapters.	

Request Example

Request Example

User-Agent: Device/1.0

Content-Type: application/json

Content-Length: <data length>

```
{
    "Code" : "FaceRecognition",
    "Action" : "Pulse",
    "Index" : 0
    Data
        "UTC" : 123456789,
        ...
    }
}
```

Response Params (OK in body)

Response Example

OK

4.13.5 [Config] Active Report Data Uploading

Configure the device to upload report data to the specified address. Before configuration, the client should monitor in the specified IP address. Once the configuration is complete, the device would upload report data to this address using the command described in chapters 4.13.6 ~ 4.13.11.

The parameters for uploading image are as follows:

Config Data Params				
Name	Type	R/O	Description	Example
ReportHttpUpload	object	R	parameters for uploading event	
+Enable	bool	R	Whether to enable the function	true
+Type	char[16]	O	Authentication Type "basic" :HTTP basic authentication "digest" :HTTP digest authentication	"digest"
+Period	int	O	Upload period (unit: hour)	1
+UploadServerList	object[]	R	The server list that receives uploaded information	
++Address	char[128]	R	IP address or domain name of the server	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/handleevt.php"
++ReportType	char [][40]	O	For the uploaded report data list, specific report data name, please refer to the corresponding command reporting chapter description.	["NumberStat","ObjectDetect"]
++HttpsEnable	bool	O	https enable	true
++AuthEnable	bool	O	authentication enable	true

Please refer to "4.2.1 Get and Set Configure" for configuration getting and setting. Specific examples are as follows :

Get Config Request Example

```
http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=ReportHttpUpload
```

Get Config Response Example

```
table.ReportHttpUpload.Enable=true
table.ReportHttpUpload.Type=digest
table.ReportHttpUpload.Period=1
table.ReportHttpUpload.UploadServerList[0].Address=192.168.1.208
table.ReportHttpUpload.UploadServerList[0].Port=80
table.ReportHttpUpload.UploadServerList[0].UserName=abc
table.ReportHttpUpload.UploadServerList[0].Password=123
table.ReportHttpUpload.UploadServerList[0].Uploadpath=/example/handlerpt.php
table.ReportHttpUpload.UploadServerList[0].ReportType[0]=NumberStat
table.ReportHttpUpload.UploadServerList[0].ReportType[1]=ObjectDetect
```

Set Config Request Example

```
http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&ReportHttpUpload.Enable=true&ReportHttpUpload.Type=digest&ReportHttpUpload.Period=1&ReportHttpUpload.UploadServerList[0].Address=192.168.1.208&ReportHttpUpload.UploadServerList[0].Port=80&ReportHttpUpload.UploadServerList[0].UserName=abc&ReportHttpUpload.UploadServerList[0].Password=123&ReportHttpUpload.UploadServerList[0].Uploadpath=/example/handlerpt.php&ReportHttpUpload.UploadServerList[0].ReportType[0]=NumberStat&ReportHttpUpload.UploadServerList[0].ReportType[1]=ObjectDetect
```

Set Config Response Example

```
OK
```

4.13.6 People Counting Report Data Uploading

If the client subscribes to the Numberstat (Number Statistics) report data through the ReportHttpUpload configuration, the device will use this command to upload the data to the specified address. The IP address, port and URL of the target server is specified by ReportHttpUpload. Each people counting report data is sent within one separate HTTP request.

Request URL	http://<Address>:<Port>/<Uploadpath_of_ReportHttpUpload>			
Method	POST			
Request Params (JSON in body)				
Name	Type	R/O	Description	Example
NumberStat	Object[]	R	NumberStat: each dimension is a channel.	
+SN	char[32]	R	Device serial number	"1C03E08YAZ00020"
+PeopleCount	Object[]	O	Humber statistics, the number of arrays is that of the cycles. If the period is an hour, then the number of array is one. If the period is three hours, then the number of array is three.	
++StartTime	char[20]	O	Start time	"2012-03-14 00:00:00"

EndTime:	char[20]	O	End time	"2012-03-14 23:59:59"
++EnteredSubtotal	uint	O	Subtotal of entered flow	56
++ExitedSubtotal	uint	O	Subtotal of exit flow	56
++AreaID	uint16 port	O	Area ID.	1
++Channel	int	O	Channel No.	0
+ManNumDetection	Object[]	O	Statistics of the number of the people in the region. The number of arrays equals to that of the period. If the time of period is an hour, then the number of array is one. If the time of the period is three hours, then the number of array is three.	
++StartTime	char[20]	O	Start time	"2012-03-14 00:00:00"
EndTime:	char[20]	O	End time	"2012-03-14 23:59:59"
++InsideSubtotal	uint	O	Used with ManNumDetection, QueueDetection, the subtotal of number of people in the region	56
++AreaID	uint16 port	O	Area ID.	1
++AverageStayTime	uint	O	Average regional retention time	10
++Channel	int	O	Channel No.	0
+QueueDetection	Object[]	O	Queuing Management Data. The number of arrays equals to that of the period. If the time of period is an hour, then the number of array is one. If the time of the period is three hours, then the number of array is three.	
++StartTime	char[20]	O	Start time	"2012-03-14 00:00:00"
++EndTime	char[20]	O	Statistics end time	"2012-03-14 23:59:59"
++InsideSubtotal	uint	O	Subtotal people in the area when ManNumDetection and QueueDetection are used	56
++AreaID	uint16	O	Area ID	1
++AverageStayTime	uint	O	Average stranding time in area	10
++Channel	int	O	Channel No.	0
Request Example				
POST http://192.168.1.208:80/example/handlerpt.php HTTP/1.1				
User-Agent: Device/1.0				

Content-Type: application/json

Content-Length: <data length>

```
{  
    "NumberStat" : [ {  
        "SN" : "6M053FDYAQ00003"  
        "ManNumDetection" : [ {  
            "Channel" : 0,  
            "EndTime" : "2021-03-11 17:59:59",  
            "EnteredSubtotal" : 0,  
            "ExitedSubtotal" : 0,  
            "StartTime" : "2021-03-11 17:00:00"  
        }, {  
            "Channel" : 0,  
            "EndTime" : "2021-03-11 6:59:59 PM",  
            "EnteredSubtotal" : 0,  
            "ExitedSubtotal" : 0,  
            "StartTime" : "2021-03-11 6:00:00 PM"  
        }, ..., { } ],  
        "PeopleCount" : [ {  
            "Channel" : 0,  
            "EndTime" : "2021-03-11 5:59:59 PM",  
            "EnteredSubtotal" : 0,  
            "ExitedSubtotal" : 0,  
            "StartTime" : "2021-03-11 5:00:00 PM"  
        }, {  
            "Channel" : 0,  
            "EndTime" : "2021-03-11 6:59:59 PM",  
            "EnteredSubtotal" : 0,  
            "ExitedSubtotal" : 0,  
            "StartTime" : "2021-03-11 6:00:00 PM"  
        }, ..., { } ],  
        "QueueDetection" : [ {  
            "Channel" : 0,  
            "EndTime" : "2021-03-11 5:59:59 PM",  
            "EnteredSubtotal" : 0,  
            "ExitedSubtotal" : 0,  
            "StartTime" : "2021-03-11 5:00:00 PM"  
        }, {  
            "Channel" : 0,  
            "EndTime" : "2021-03-11 6:59:59 PM",  
            "EnteredSubtotal" : 0,  
            "ExitedSubtotal" : 0,  
            "StartTime" : "2021-03-11 6:00:00 PM"  
        }, ..., { } ]  
    }  
}
```

Response Params (OK in body)

Response Example

OK

4.13.7 Video Structuring Report Data Upload

When the client subscribes ObjectDetect (video Structuring) dashboard, the device will upload video structuring dashboard to the specified address. The target service address and port, URL are designated by ReportHttpUpload. Each video structuring report data is sent within one separate HTTP request.

Request URL	http://<Address>:<Port>/<Uploadpath_of_ReportHttpUpload>		
Method	POST		
Request Params (JSON in body)			
Name	Type	R/O	Description
ObjectDetect	Object[] []	R	Video structured data, the first dimension is channel, the second dimension is cycle count, the cycle is an hour, the array is 1, the cycle is 3 hours and the array is 3.
SN	char[32]	R	Device SN
+UTC	uint	O	The end time of statistical Cycle (local UTC time)
+UTCMS	uint	O	Milliseconds
+Period	int	O	Statistical cycle (unit: minute)
+PeriodBySeconds	int	O	Statistical cycle (unit: second)
+MotoVehicles	int	O	Traffic volume of motorized vehicle such as motorbike and tricycles
+Vehicles	int	O	Total count of motorized and non-motorized vehicle
+VehicleTypeFlow	object	O	Statistical data on the traffic volume based on vehicle type
++PasserbyVehicles	int	O	Pedestrian traffic flow

Request Example

POST http://192.168.1.208:80/example/handlerpt.php HTTP/1.1

User-Agent: Device/1.0

Content-Type: application/json

Content-Length: <data length>

```
{
  "ObjectDetect": [ [ {
    "MotoVehicles": 8,
    "Period": 60,
    "PeriodBySeconds": 0,
    "SN": "6J0CB81YAG10101",
    "UTC": 1615492800,
```

```

    "UTCMS" : 0,
    "VehicleTypeFlow" : {
        "PasserbyVehicles" : 6
    },
    "Vehicles" : 17
}, ..., { }], ..., []]
}

```

Response Params (OK in body)

Response Example

OK

4.13.8 People Flow Heat Map Report Data Upload

When the client subscribes HeatMap (customer flow heat map report data, the device will upload video structuring data to the specified address. The target service address and port, URL are designated by ReportHttpUpload. Each people flow heat map report data is sent within one separate HTTP request.

Request URL	http://<Address>:<Port>/<Uploadpath_of_ReportHttpUpload>		
Method	POST		
Request Params (JSON in body)			
Name	Type	R/O	Description
HeatMap	Object[]	R	Heatmap is presented in binary format. Each dimension is a channel.
+Channel	int	O	Channel
+SN	char[32]	R	Device SN
+StartTime	char[20]	O	Statistics start Time
+EndTime	char[20]	O	Statistics end time
+width	uint	O	Image width
+height	uint	O	Image height
+EncodeData	string	O	base64 encoding of Heat map data

Request Example

POST http://192.168.1.208:80/example/handlerpt.php HTTP/1.1

User-Agent: Device/1.0

Content-Type: application/json

Content-Length: <data length>

```

{
    "HeatMap" : [ {
        "Channel" : 0,
        "SN" : "66:66:66:56:78:9a",
        "StartTime" : "2021-03-12 14:00:00",
        "EndTime" : "2021-03-12 15:00:00",
        "height" : 64,
        "width" : 64
        "EncodeData" : "abcd=",
    }, ..., {} ]
}

```

}

Response Params (OK in body)

Response Example

OK

4.13.9 ANPR Report Data Upload

When the client subscribes RoadFlowStat (road monitoring) report data, the device will upload video structuring data to the specified address. The target service address and port, URL are designated by ReportHttpUpload. Each ANPR report data is sent within one separate HTTP request.

Request URL	http://<Address>:<Port>/<Uploadpath_of_ReportHttpUpload>		
Method	POST		
Request Params (JSON in body)			
Name	Type	R/O	Description
RoadFlowStat	Object[] []	R	Road monitoring data, the first dimension is channel, the second dimension is cycle count. The cycle is an hour and the array is 1. The cycle is 3 hours and the array is 3.
SN	char[32]	O	Device SN
+UTC	uint	O	End time of statistical cycle
+UTCMS	uint	O	Milliseconds
+Lane	int	O	User defined lane number
+PresetID	int	O	PTZ preset, which must greater than 0.
+Period	int	O	Statistical cycle (unit: minute)
+PeriodBySeconds	int	O	Statistical cycle (unit: second)
+MotoVehicles	int	O	Traffic flow of non-motorized vehicles such as motorbike and tricycles.
+Vehicles	int	O	Total count of motorized and non-motorized vehicle
+VehicleTypeFlow	object	O	Statistical data on the traffic volume based on vehicle type
++PasserbyVehicles	int	O	Pedestrian traffic flow

Request Example

POST http://192.168.1.208:80/example/handlerpt.php HTTP/1.1

User-Agent: Device/1.0

Content-Type: application/json

Content-Length: <data length>

{

```
"RoadFlowStat": [ [ {  
    "SN": "6J0CB81YAG10101",  
    "UTC": 1615485600,  
    "UTCMS": 0,
```

```

    "Lane" : 1,
    "PresetID" : 0,
    "Period" : 60,
    "PeriodBySeconds" : 0,
    "MotoVehicles" : 0,
    "Vehicles" : 26506
    "VehicleTypeFlow" : [
        "PasserbyVehicles" : 0
    ],
    }, ..., { }], ..., []
}

```

Response Params (OK in body)

Response Example

OK

4.13.10 Crowd Distribution Report Data upload

When the client subscribes CrowdDistriMap (Crowd Distribution) report data, the device will upload video structuring data to the specified address. The target service address and port, URL are designated by ReportHttpUpload. Each crowd distribution report data is sent within one separate HTTP request.

Request URL	http://<Address>:<Port>/<Uploadpath_of_ReportHttpUpload>			
Method	POST			
Request Params (JSON in body)				
Name	Type	R/O	Description	Example
CrowdDistriMap	Object[]	R	Data statistics on crowd distribution. Each dimension is the data of a channel.	
SN	char[32]	O	Device SN	"1C03E08YAZ00020"
+CrowdDistriMap	object	O	Report Data on Crowd Distribution Map	
++DataList	object[64]	O	Return Crowd Distribution Map data list. The array size is 64 at most.	1
+++UTC	uint32	O	UTC time (local UTC time) when the data is recorded	6555478
+++AreaName	char[32]	O	Statistics Area name	"CMD-5"
+++PeopleNum	int	O	The number of people in statistics area	60

Request Example

POST http://192.168.1.208:80/example/handlerpt.php HTTP/1.1

User-Agent: Device/1.0

Content-Type: application/json

Content-Length: <data length>

```
{
    "CrowdDistriMap" : [ {
        "CrowdDistriMap" : {

```

```

    "DataList" : [ {
        "AreaName" : "CDM-1",
        "PeopleNum" : 0,
        "UTC" : 1617375600
    }, {
        "AreaName" : "CDM-1",
        "PeopleNum" : 0,
        "UTC" : 1617379200
    } ]
},
"SN" : "7B0606BYAQ00010"
}, ... , {}]
}

```

Response Params (OK in body)

Response Example

OK

4.13.11 Vehicle Density Report Data Upload

When the client subscribes VehiclesDistri vehicle density) report data, the device will upload video structuring data to the specified address. The target service address and port, URL are designated by ReportHttpUpload. Each vehicle density report data is sent within one separate HTTP request.

Request URL	http://<Address>:<Port>/<Uploadpath_of_ReportHttpUpload>			
Method	POST			
Request Params (JSON in body)				
Name	Type	R/O	Description	Example
VehiclesDistri	Object[]	R	Data statistics on vehicle density distribution. Each dimension is the data of a channel.	
+SN	char[32]	O	Device SN	"1C03E08YAZ00020"
+CongestionDetection	object	O	The data report of vehicle density congestion rules.	0
++DataNum	uint32	O	Number of returned entries on vehicle congestion rules.	60
++DataList	object[]	O	The data list of vehicle congestion rules. The array size equal to DataNum and the element is 64 at most.	
+++UTC	uint32	O	UTC time (local UTC time) when the data is recorded	3665789
+++RuleName	char[32]	O	Rule name	"VD-1"
+++VehiclesNum	int	O	The number of vehicles	32
+VehicleLimitDetection	object	O	The report data of traffic flow limit detection rules	0
++DataNum	uint32	O	Returned data entries on the traffic	1

			flow limit rules	
++DataList	object[]	O	The data list of traffic flow limit detection rules. 64 elements at most.	
+++UTC	uint32	O	UTC time when the data is recorded	3665789
+++RuleName	char[32]	O	Rule name	
+++VehiclesNum	int	O	The number of vehicles	40

Request Example

```
POST http://192.168.1.208:80/example/handlerpt.php HTTP/1.1
User-Agent: Device/1.0
Content-Type: application/json
Content-Length: <data length>

{
  "VehiclesDistri" : [ {
    "CongestionDetection" : {
      "DataList" : [ {
        "RuleName" : "VD-1",
        "UTC" : 1617375600,
        "VehiclesNum" : 0
      }, {
        "RuleName" : "VD-1",
        "UTC" : 1617379200,
        "VehiclesNum" : 0
      } ],
      "DataNum" : 2
    },
    "SN" : "7B0606BYAQ00010",
    "VehicleLimitDetection" : {
      "DataNum" : 0
    }
  }, ... , {} ]
}
```

Response Params (OK in body)
Response Example
OK

4.13.12 FaceAnalysis Report Data Upload

When the client subscribes FaceAnalysis report data, the device will upload video structuring data to the specified address. The target service address and port, URL are designated by ReportHttpUpload. Each vehicle density report data is sent within one separate HTTP request.

Request URL	http://<Address>:<Port>/<Uploadpath_of_ReportHttpUpload>		
Method	POST		
Request Params (JSON in body)			
Name	Type	R/O	Description

FaceAnalysis	Object[]	R	Data statistics on FaceAnalysis. Each dimension is the data of a channel.	
+SN	char[32]	R	Device SN	"8J0751AYAQ00004"
+Channel	uint	O	Channel	0
+Info	Object[]	O	Data statistics on FaceAnalysis. The number of arrays is the number of cycles. The cycle is one hour, the number of arrays is 1. The cycle is 3 hours, and the number of arrays is 3.	
++StartTime	systemtime	O	StartTime	"2022-12-21 11:00:00"
++EndTime	systemtime	O	EndTime	"2022-12-21 11:59:59"
++RepeatCount	uint32	O	Repeat Count	0
++FaceDetectCount	uint32	O	Statistics of face detection times after duplicate removal	102
++FaceAnalysisCount	uint32	O	Statistics of face analysis times after duplicate removal	92

Request Example

POST http://192.168.1.208:80/example/handlerpt.php HTTP/1.1

User-Agent: Device/1.0

Content-Type: application/json

Content-Length: <data length>

```
{
  "FaceAnalysis": [
    {
      "Channel": 0,
      "SN": "8J0751AYAQ00004",
      "Info": [
        {
          "EndTime": "2022-12-21 11:59:59",
          "FaceAnalysisCount": 92,
          "FaceDetectCount": 109,
          "RepeatCount": 0,
          "StartTime": "2022-12-21 11:00:00"
        }, ...
      ]
    }
  ]
}
```

Response Params (OK in body)

Response Example

OK

5.1 Image

5.1.1 [Config] Brightness, Contrast and Saturation

Config Data Params				
Name	Type	R/O	Description	Example
VideoColor	object[] []	O	video colour configuration A two-dimensional array with multiple color configurations corresponding to each video input channel The first dimension corresponds to the channel number, and the second dimension corresponds to the lighting scene. For compatibility, the front element is fixed to represent day, night, and normal.	
+Name	char[16]	O	name	"Day"
+Brightness	uint8	O	Brightness, range is [0—100]	50
+Contrast	uint8	O	Contrast, range is [0—100]	50
+Saturation	uint8	O	Saturation, range is [0—100]	50
+Hue	uint8	O	Hue 0~100	50
+Gamma	uint8	O	Gamma 0~100	50
+ChromaSuppress	uint8	O	Suppress range: 0~100	50
+Style	char[16]	O	colour style range : [Gentle,Standard,Flamboyant]	"Standard"
+TimeSection	char[20]	O	time section	"1 00:00:00-24:00:00"

Please refer to "4.2.1 Get and Set Configure" for configuration getting and setting. Specific examples are as follows :

Get Config Request Example

```
http://10.0.0.8/cgi-bin/configManager.cgi?action=getConfig&name=VideoColor
```

Get Config Response Example

```
table.VideoColor[0][0].Name=Day,  

table.VideoColor[0][0].Brightness=50,  

table.VideoColor[0][0].Contrast=50,  

table.VideoColor[0][0].Saturation=50,  

table.VideoColor[0][0].Hue=50,  

table.VideoColor[0][0].Gamma=50,
```

```
table.VideoColor[0][0].ChromaSuppress=50,
table.VideoColor[0][0].Style=Standard,
table.VideoColor[0][0].TimeSection=1 00:00:00-24:00:00
```

Set Config Request Example

[http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&VideoColor\[0\]\[0\].Brightness=50](http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&VideoColor[0][0].Brightness=50)

Set Config Response Example

OK

5.1.2 [Config] Sharpness

Config Data Params				
Name	Type	R/O	Description	Example
VideoInSharpness	object[][]	O	<p>Sharpness</p> <p>A two-dimensional array with multiple color configurations corresponding to each video input channel</p> <p>The first dimension corresponds to the channel number, and the second dimension corresponds to the lighting scene. For compatibility, the front element is fixed to represent day, night, and normal.</p>	
+Sharpness	int	O	Range is 0—100	50
+Level	int	O	Range is 0—100	20

Please refer to "4.2.1 Get and Set Configure" for configuration getting and setting. Specific examples are as follows :

Get Config Request Example

<http://10.0.0.8/cgi-bin/configManager.cgi?action=getConfig&name=VideoInSharpness>

Get Config Response Example

```
table.VideoInSharpness [0][0].Level=4
table.VideoInSharpness [0][0].Sharpness=8
```

Set Config Request Example

[http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&VideoInSharpness\[0\]\[0\].Level=10&VideoInSharpness\[0\]\[0\].Mode=1&VideoInSharpness\[0\]\[0\].Sharpness=0](http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&VideoInSharpness[0][0].Level=10&VideoInSharpness[0][0].Mode=1&VideoInSharpness[0][0].Sharpness=0)

Set Config Response Example

OK

5.1.3 [Config] Flip, Mirror and Rotate90

Config Data Params				
Name	Type	R/O	Description	Example

VideoImageControl	object[]	O	array index starts from 0, which means video channel (equals to video channel index -1, and so 0 means channel 1).	
+Mirror	bool	O	true: enable video mirror function false: disable video mirror function	false
+Flip	bool	O	true: enable video flip function false: disable video flip function	false
+Rotate90	int	O	Range is {0,1,2} Video rotation: 0: No rotate 1: clockwise rotate 90° 2: anticlockwise rotate 90°	0

Please refer to "4.2.1 Get and Set Configure" for configuration getting and setting. Specific examples are as follows :

Get Config Request Example

```
http://10.0.0.8/cgi-bin/configManager.cgi?action=getConfig&name=VideoImageControl
```

Get Config Response Example

```
table.VideoImageControl[0].Flip=true
```

```
table.VideoImageControl[0].Mirror=false
```

```
table.VideoImageControl[0].Rotate90=0
```

Set Config Request Example

```
http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&VideoImageControl[0].Flip=true
```

Set Config Response Example

```
OK
```

5.2 Exposure

5.2.1 [Config] Exposure Config

Config Data Params				
Name	Type	R/O	Description	Example
VideoInExposure	object[][]	O	A two-dimensional array with multiple color configurations corresponding to each video input channel The first dimension corresponds to the channel number, and the second dimension corresponds to the lighting scene. For compatibility, the front element is fixed to represent day, night, and normal.	
+Mode	uint8	O	0: Auto by default 1: Low noise	0

			2: Anti-smear 4: Manual (range) 5: Aperture priority 6: Manual (fixed) 7: Gain priority 8: Shutter priority 9: Flash light matching mode	
+AntiFlicker	enumint8	O	Range is {0,1,2} AntiFlicker mode: 0: Outdoor 1: 50 Hz AntiFlicker 2: 60 Hz AntiFlicker	0
+Gain	uint8	O	Range is [0—100] If GainAuto is true, it's upper limit of auto gain, else it's the fixed gain adjust value.	1
+GainMin	uint8	O	Range is 0—100; the value must be smaller than GainMax.	10
+GainMax	uint8	O	Range is 0—100; the value must be greater than GainMin.	50
+Value1	double	O	Range is [0-1000], unit is millisecond If ExposureSpeed is 0(AutoExposure enable), it's lower limit of AutoExposure time, otherwise it's time of manualExposure	0.1
+Value2	double	O	Range is [0-1000], unit is millisecond Upper limit of AutoExposure time, should be bigger than ExposureValue1	80
+Iris	uint8	O	Manual Iris setting. Range is 0—100.	10
+IrisAuto	bool	O	Enable Iris automatically. true: IrisAuto false: No IrisAuto	false

Please refer to "4.2.1 Get and Set Configure" for configuration getting and setting. Specific examples are as follows :

Get Config Request Example

```
http://10.0.0.8/cgi-bin/configManager.cgi?action=getConfig&name=VideoInExposure
```

Get Config Response Example

```
table.VideoInExposure[0][0].AntiFlicker=0
table.VideoInExposure[0][0].Gain=50
table.VideoInExposure[0][0].GainMax=50
table.VideoInExposure[0][0].GainMin=0
table.VideoInExposure[0][0].Iris=50
table.VideoInExposure[0][0].IrisAuto=false
table.VideoInExposure[0][0].Mode=0
```

```
table.VideoInExposure[0][0].Value1=40
table.VideoInExposure[0][0].Value2=40
```

Set Config Request Example

```
http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&VideoInExposure[0][0].Iris=50
```

Set Config Response Example

```
OK
```

5.3 Backlight

5.3.1 [Config] Backlight Config

Config Data Params				
Name	Type	R/O	Description	Example
VideoInBacklight	object[]()	O	<p>Light environment configuration.</p> <p>Adjustments in various scenarios, including backlight, strong light, etcA</p> <p>two-dimensional array with multiple color configurations corresponding to each video input channel</p> <p>The first dimension corresponds to the channel number, and the second dimension corresponds to the lighting scene. For compatibility, the front element is fixed to represent day, night, and normal.</p>	
+Mode	char[32]	O	<p>Off: Switched off</p> <p>Backlight: Backlight compensation</p> <p>GlareInhibition: HLC</p> <p>WideDynamic: WDR</p> <p>SSA: Scene adaptation</p>	"Off"
+WideDynamicRange	uint8	O	Range is 1–100.	0
+GlareInhibition	uint8	O	Range is 1–100.	50

Please refer to "4.2.1 Get and Set Configure" for configuration getting and setting. Specific examples are as follows :

Get Config Request Example

```
http://10.0.0.8/cgi-bin/configManager.cgi?action=getConfig&name=VideoInBacklight
```

Get Config Response Example

```
table.VideoInBacklight[0][0].GlareInhibition=50
table.VideoInBacklight[0][0].Mode=Off
table.VideoInBacklight[0][0].WideDynamicRange=50
```

Set Config Request Example

```
http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&VideoInBacklight[0][0].GlareInhibition=50
```

Set Config Response Example

OK

5.4 White Balance

5.4.1 [Config] White Balance Config

Config Data Params				
Name	Type	R/O	Description	Example
VideoInWhiteBalance	object[] []	O	two-dimensional array with multiple color configurations corresponding to each video input channel The first dimension corresponds to the channel number, and the second dimension corresponds to the lighting scene. For compatibility, the front element is fixed to represent day, night, and normal.	
+Mode	char[16]	O	Auto; Indoor; Outdoor; ATW; Manual; Sodium; Natural; StreetLamp; ManualDatum	"Auto"
+GainRed	int	O	Range is [0—100] Gain for red value, Value is effective when WhiteBalance is "Custom."	80
+GainBlue	int	O	Range is [0—100] Gain for blue value, Value is effective when WhiteBalance is "Custom."	80

Please refer to "4.2.1 Get and Set Configure" for configuration getting and setting. Specific examples are as follows :

Get Config Request Example

```
http://10.0.0.8/cgi-bin/configManager.cgi?action=getConfig&name=VideoInWhiteBalance
```

Get Config Response Example

VideoInWhiteBalance[0][0].GainBlue=50

VideoInWhiteBalance[0][0].GainRed=50

VideoInWhiteBalance[0][0].Mode=Auto

Set Config Request Example

```
http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&VideoInWhiteBalance[0][0].GainBlue=50
```

Set Config Response Example

OK

5.5 Day-Night

5.5.1 [Config] Day-Night Config

Config Data Params				
Name	Type	R/O	Description	Example
VideoInDayNight	object[][]	O	two-dimensional array with multiple color configurations corresponding to each video input channel The first dimension corresponds to the channel number, and the second dimension corresponds to the lighting scene. For compatibility, the front element is fixed to represent day, night, and normal.	
+Type	char[16]	O	The range is {"Electron", "Mechanism", "NightICR", "Auto" }, the way of ICR switching.	"Mechanism"
+Mode	char[16]	O	The range is {"Color", "Brightness", "BlackWhite", "Photoresistor", "Gain", "Timing"}. "Color": Always "color" "Brightness": Day/Night Auto "BlackWhite": Always black-and-white "Photoresistor": Switch according to photoresistor, "Gain": Switch according to gain "Timing": Switch according to time	"Brightness"
+Sensitivity	int	O	Range is [1-3]. Sensitivity of switching mode	1
+Delay	int	O	Range is [2-10]. Delay seconds when switching mode.	10

Please refer to "4.2.1 Get and Set Configure" for configuration getting and setting. Specific examples are as follows :

Get Config Request Example

```
http://10.0.0.8/cgi-bin/configManager.cgi?action=getConfig&name=VideoInDayNight
```

Get Config Response Example

```
VideoInDayNight[0][0].Delay=10
```

```
VideoInDayNight[0][0].Mode=Brightness
```

```
VideoInDayNight[0][0].Sensitivity=2
```

```
VideoInDayNight[0][0].Type=Mechanism
```

Set Config Request Example

```
http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&VideoInDayNight[0][0].Mode=BlackWhite
```

Set Config Response Example

OK

5.6 Zoom and Focus

To get the capability set of video input, refer to 4.5.12. For instance, you can use the following URL:

<http://192.168.1.108/cgi-bin/devVideoInput.cgi?action=getCaps&channel=1>

If ElectricFocus or SyncFocus of the return value is true, use 5.6.1 ~ 5.6.4; Otherwise, use 5.6.5 ~ 5.6.6 .

5.6.1 Adjust Focus

Adjust magnification and focus.

Request URL	http://<server>/cgi-bin/devVideoInput.cgi?action=adjustFocus		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
channel	int	O	The video channel number which starts from 1, and the default value is 1.
focus	double	O	Relative stepping position of the vari-focal motor; range: [0~1]. -1 means resetting.
zoom	double	O	Relative stepping position of the zoom motor; range: [0~1]. -1 means resetting.
Request Example			
http://192.168.1.108/cgi-bin/devVideoInput.cgi?action=adjustFocus&focus=0.5&zoom=-0.5			

Response Params (OK in body)

Response Example

OK

5.6.2 Adjust Focus Continuously

Continuously adjust magnification and focus. Firstly, send a "non-zero" value to start zooming and adjusting the focal length to drive the motor to move, and then send "0" to stop the motor from moving. If only one of the operations is required, set the value of the other operation as -1.

Request URL	http://<server>/cgi-bin/devVideoInput.cgi?action=adjustFocusContinuously		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
channel	int	O	The video channel number which starts from 1, and the default value is 1.
focus	double	O	Relative focusing rate; range: [-1,1]. -1: No operation

			0: Stop Positive number: Moving forward Negative number: Moving backward	
zoom	double	O	Relative zooming rate; range: [-1,1]. -1: No operation 0: Stop	-1

Request Example

```
http://192.168.1.108/cgi-bin/devVideoInput.cgi?action=adjustFocusContinuously&channel=1&focus=0.0
2&zoom=-1
```

Response Params (OK in body)

Response Example

OK

5.6.3 Auto Focus

Auto focus.

Request URL	http://<server>/cgi-bin/devVideoInput.cgi?action=autoFocus			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
channel	int	O	The video channel number which starts from 1, and the default value is 1.	1

Request Example

```
http://192.168.1.108/cgi-bin/devVideoInput.cgi?action=autoFocus&channel=1
```

Response Params (OK in body)

Response Example

OK

5.6.4 Get Focus Status

Get the focusing status.

Request URL	http://<server>/cgi-bin/devVideoInput.cgi?action=getFocusStatus			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
channel	int	O	The video channel number which starts from 1, and the default value is 1.	1

Request Example

```
http://192.168.1.108/cgi-bin/devVideoInput.cgi?action=getFocusStatus&channel=1
```

Response Params (key=value format in body)

Name	Type	R/O	Description	Example
status	object	R	Return the status information	

+Focus	double	R	Relative stepping position of the focusing motor; range: [0,1]	0.8
+Zoom	double	R	Relative stepping position of the zooming motor; range: [0,1]	0.3
+Status	char[16]	R	Focusing status "Normal": Normal "Autofocus": Auto focusing	"Normal"

Response Example

status.Focus=0.8

status.Zoom=0.3

status.Status=Normal

5.6.5 [Config] Zoom Config

Zooming configuration parameter:

Config Data Params				
Name	Type	R/O	Description	Example
VideolnZoom	object[][]	R	Zooming configuration parameter; two-dimensional array. The first dimension represents the video channel which starts from 1, and the second dimension represents the lighting scene. The beginning three elements constantly represent day, night and general scene.	
+Name	char[16]	O	Scene name	"Day"
+Speed	int	O	Zooming speed; range: [0–100]	8
+DigitalZoom	bool	O	Whether to enable digital zoom	true

Please refer to "4.2.1 Get and Set Configure" for configuration getting and setting. Specific examples are as follows :

Get Config Request Example

<http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=VideolnZoom>

Get Config Response Example

table.VideolnZoom[0][0].Name=Day

table.VideolnZoom[0][0].Speed=8

table.VideolnZoom[0][0].DigitalZoom=true

...

Set Config Request Example

[http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&VideoInZoom\[0\]\[0\].DigitalZoom=false&VideoInZoom\[0\]\[0\].Speed=8](http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&VideoInZoom[0][0].DigitalZoom=false&VideoInZoom[0][0].Speed=8)

Set Config Response Example

OK

5.6.6 [Config] Focus Config

Vari-focal configuration parameter:

Config Data Params				
Name	Type	R/O	Description	Example
VideoInFocus	object[][]	R	<p>Vari-focal configuration parameter; two-dimensional array.</p> <p>The first dimension represents the video channel which starts from 1, and the second dimension represents the lighting scene. The beginning three elements constantly represent day, night and general scene.</p>	
+Name	char[16]	O	Scene name	"Day"
+Mode	int	O	<p>Focus mode</p> <p>2: Auto focus</p> <p>3: Semi Auto (Customizable, uses auto focus first, and then locks the focusing module. Auto focus is not available at the time, and you have to manually adjust the focal length.)</p> <p>4: Manual focus</p>	2
+FocusLimit	int	O	<p>Recommended limit value of near-field focusing; unit: mm.</p> <p>The value range depends on the device capability.</p>	2000
+FocusFarLimit	int	O	<p>Recommended limit value of far-field focusing; unit: mm.</p> <p>The value range depends on the device capability.</p>	5000
+AutoFocusTrace	int	O	<p>Vari-focal tracking</p> <p>0: Close</p> <p>1: Open</p>	1
+IRCorrection	int	O	<p>IR light focusing and correction</p> <p>0: No correction</p> <p>1: Manual correction</p> <p>2: Auto correction</p>	1
+Sensitivity	int	O	<p>Focusing sensitivity</p> <p>0: High</p> <p>1: Default</p> <p>2: Low</p>	1
+FocusLimitSelect Mode	char[16]	O	<p>Focuings limit mode (The distance limit for near-field focusing)</p> <p>"Manual"</p> <p>"Auto"</p>	"Auto"

Please refer to "4.2.1 Get and Set Configure" for configuration getting and setting. Specific examples are as follows :

Get Config Request Example

```
http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=VideoInFocus
```

Get Config Response Example

```
table.VideoInFocus[0][0].Name=Day
table.VideoInFocus[0][0].Mode=0
table.VideoInFocus[0][0].Sensitivity=1
table.VideoInFocus[0][0].FocusLimitSelectMode=Manual,
table.VideoInFocus[0][0].FocusLimit=2000
table.VideoInFocus[0][0].FocusFarLimit=5000
table.VideoInFocus[0][0].AutoFocusTrace=0
table.VideoInFocus[0][0].IRCorrection=0
...
...
```

Set Config Request Example

```
http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&VideoInFocus[0][0].FocusLimit=2000
&VideoInFocus[0][0].FocusFarLimit=5000&VideoInFocus[0][0].Sensitivity=1
```

Set Config Response Example

```
OK
```

5.7 Lighting

5.7.1 [Config] Lighting Config

Config Data Params				
Name	Type	R/O	Description	Example
Lighting	object[]()	O	two-dimensional array. The first dimension represents the video channel which starts from 1, and the second dimension represents the lighting scene. The beginning three elements constantly represent day, night and general scene.	
+Mode	char[32]	O	Light mode. The range is {"Manual", "Auto", "Off", "ZoomPrio"}. The following are special for composite lamps. The range is { "Timing" , "SmartLight" , "ExclusiveManual" }	"ZoomPrio"
+Correction	int	O	Light compensation. The range is [0—100], effective in ZoomPrio mode.	2
+NearLight	object[]	O	near light.	
++Light	int	O	Range is [0—100].	0

			The luminance of near light.	
++Angle	int	O	Laser lamp angle normalized value 0~100	50
+MiddleLight	object[]	O	middle light.	
++Light	int	O	Range is [0—100]. The luminance of middle light.	0
++Angle	int	O	Laser lamp angle normalized value 0~100	50
+FarLight	object[]	O	far light.	
++Light	int	O	Range is [0—100]. The luminance of far light.	0
++Angle	int	O	Laser lamp angle normalized value 0~100	50

Please refer to "4.2.1 Get and Set Configure" for configuration getting and setting. Specific examples are as follows :

Get Config Request Example

```
http://10.0.0.8/cgi-bin/configManager.cgi?action=getConfig&name=Lighting
```

Get Config Response Example

```
table.Lighting[0][0].Correction=50
table.Lighting[0][0].FarLight[0].Angle=0
table.Lighting[0][0].FarLight[0].Light=0
table.Lighting[0][0].Mode=ZoomPrio
table.Lighting[0][0].NearLight[0].Angle=0
table.Lighting[0][0].NearLight[0].Light=0
table.Lighting[0][0].MiddleLight[0].Angle=50
table.Lighting[0][0].MiddleLight[0].Light=50
table.Lighting[0][1].Correction=50
...
HTTP API V3.35 - Intelbras
```

Set Config Request Example

Turn on light:

```
http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&Lighting[0][0].FarLight[0].Light=10&Lighting[0][0].NearLight[0].Light=90&Lighting[0][0].Mode=Manual
```

Shift the light to ZoomPrio mode:

```
http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&Lighting[0][0].Correction=50&Lighting[0][0].Mode=ZoomPrio
```

Set Config Response Example

OK

5.8 Video in Options

5.8.1 Change binocular camera's splice mode

Request URL	http://<server>/cgi-bin/api/MultiVideo/changeSpliceMode			
Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
State	int32	R	Splicing state 1: Switch to splice mode (Binocular camera's monocular mode) 0: Switch to non-splicing mode (Binocular camera's original mode)	0
Request Example				
{ "State": 0 }				

Response Params (JSON format in body)				
Name	Type	R/O	Description	Example
Response Example				
{ }				

5.8.2 [Config] Video in Options Config

It's not recommended to use the CGI command from " video in options " ; It's now recommended to use the commands in 5.1 – 5.7.

Config Data Params				
Name	Type	R/O	Description	Example
VideoInOptions	object[]	O	Video Input Options A one-dimensional array, with each video input channel corresponding to a configuration, and channel numbers starting from 0.	
+Backlight	int	O	Range is [0—n] n depends on capability in GetVideoInputCaps 0: backlight closed. 1: backlight grade 1 ... n - backlight grade n	0
+DayNightColor	int	O	Range is {0,1,2} 0: always multicolor	0

			1: autoswitch along with brightness, 2: always monochrome,	
+WhiteBalance	char[32]	O	White balance Mode. Range is {Disable, Auto, Custom, Sunny, Cloudy, Home, Office, Night} Some IPC supports common modes: "Disable", "Auto", "Sunny", "Night", "Outdoor", "Custom" Sometimes the device support other advanced modes: "CustomColorTemperature", "Indoor", "ATW", "Manual", "AutoOutdoor", "ManualDatum" and so on.	"Disable"
+Mirror	bool	O	true: enable video mirror function false: disable video mirror function	false
+Flip	bool	O	true: enable video flip function false: disable video flip function	false
+IrisAuto	bool	O	true: IrisAuto false: No IrisAuto	false
+GainRed	uint8	O	Range is [0—100] Gain for red value, Value is effective when WhiteBalance is "Custom."	50
+GainBlue	uint8	O	Range is [0—100] Gain for blue value, Value is effective when WhiteBalance is "Custom."	50
+GainGreen	uint8	O	Range is [0—100] Gain for green value, Value is effective when WhiteBalance is "Custom."	50
+ExposureValue1	double	O	Range is [0.1—80], unit is millisecond If ExposureSpeed is 0(AutoExposure enable), it's lower limit of AutoExposure time, otherwise it's time of manualExposure	0.1
+ExposureValue2	double	O	Range is [0.1—80], unit is millisecond Upper limit of AutoExposure time, should be bigger than ExposureValue1	80
+Gain	uint8	O	Range is [0—100] If GainAuto is true, it's upper limit of auto gain, else it's the fixed gain adjust value.	50
+GainAuto	bool	O	true: GainAuto false: No GainAuto	true
+SignalFormat	enumchar[16]	O	Range is {Inside, BT656, 720p, 1080p, 1080i, 1080sF} Input Signal Mode	"BT656"
+Rotate90	uint8	O	Range is {0,1,2} Video rotation: 0: No rotate	0

			1: clockwise rotate 90° 2: anticlockwise rotate 90°	
+ExternalSync Phase	float	O	Range is [0°—360°] External Synchronous Signal Phase	12.6
+ExternalSync	uint8	O	Range is {0,1} External Synchronous 0: Internal Synchronization 1: External Synchronous	0
+AntiFlicker	int	O	Range is {0,1,2} AntiFlicker mode: 0: Outdoor 1: 50 Hz AntiFlicker 2: 60 Hz AntiFlicker	0
+ExposureSpec ed	int	O	Range is [0 — n+1] n depends on capability in GetVideoInputCaps 0: AutoExposure 1-n-1: manual Exposure grade n: AutoExposure with time limit. n+1: manualExposure with user-defined time (n is supported maximum exposure grade)	0
+ExposureMode	enumint	O	refer to VideoInExposure	0
+SmartIRExposure	bool	O	true: enable, false: disable	true
+FlashControl	object	O	flash control	
++Mode	int	O	Range is {0,1,2} 0: forbid flash 1: always flash 2: auto flash	0
++Value	int	O	Range is [0—15] Flashlight time-unit: 0: 0us, 1: 64us, 2: 128us, 3: 192us ... 15 - 960us	0
++PreValue	int	O	Range is [0—100] It is threshold of brightness value: if brightness is less than this value, flash light will begin to work.	50
++Pole	enumint	O	Range is {0,1, 2, 3} Trigger mode: 0: low level	0

			1: high level 2: rising-edge 3: falling-edge	
+GlareInhibitio n	int	O	Range is [0—100] GlareInhibition: 0: Close GlareInhibition.	1
+NightOptions	object	O	Special configuration options at night, automatically switching to night configuration parameters when the light is dim at night. See VideoInOptions, only parameter "ExternalSyncPhase" is valid.	
+NormalOptio ns	object	O	refer to VideoInOptions	

Please refer to "4.2.1 Get and Set Configure" for configuration getting and setting. Specific examples are as follows :

Get Config Request Example

<http://10.0.0.8/cgi-bin/configManager.cgi?action=getConfig&name=VideoInOptions>

Get Config Response Example

```
head.Backlight=0
table.VideoInOptions[0].DayNightColor=false
table.VideoInOptions[0].ExposureSpeed=0
table.VideoInOptions[0].ExposureValue1=0.100000
table.VideoInOptions[0].ExposureValue2=80.000000
table.VideoInOptions[0].ExternalSync=0
table.VideoInOptions[0].ExternalSyncPhase=0
table.VideoInOptions[0].FlashControl.Mode=0
table.VideoInOptions[0].FlashControl.Pole=0
table.VideoInOptions[0].FlashControl.Value=0
table.VideoInOptions[0].FlashControl.PreValue=0
table.VideoInOptions[0].Flip=false
table.VideoInOptions[0].Gain=50
table.VideoInOptions[0].GainAuto=true
table.VideoInOptions[0].IrisAuto=false
table.VideoInOptions[0].Mirror=false
table.VideoInOptions[0].NightOptions.AntiFlicker=0
table.VideoInOptions[0].NightOptions.Backlight=0
table.VideoInOptions[0].NightOptions.BacklightRegion[0]=3096
table.VideoInOptions[0].NightOptions.BacklightRegion[1]=3096
table.VideoInOptions[0].NightOptions.BacklightRegion[2]=5096
table.VideoInOptions[0].NightOptions.BacklightRegion[3]=5096
table.VideoInOptions[0].NightOptions.BrightnessThreshold=50
table.VideoInOptions[0].NightOptions.DayNightColor=2
table.VideoInOptions[0].NightOptions.ExposureMode=0
table.VideoInOptions[0].NightOptions.ExposureSpeed=0
table.VideoInOptions[0].NightOptions.ExposureValue1=0
```

```
table.VideoInOptions[0].NightOptions.ExposureValue2=40
table.VideoInOptions[0].NightOptions.ExternalSyncPhase=125
table.VideoInOptions[0].NightOptions.Flip=false
table.VideoInOptions[0].NightOptions.Gain=50
table.VideoInOptions[0].NightOptions.GainAuto=true
table.VideoInOptions[0].NightOptions.GainBlue=50
table.VideoInOptions[0].NightOptions.GainGreen=50
table.VideoInOptions[0].NightOptions.GainMax=50
table.VideoInOptions[0].NightOptions.GainMin=0
table.VideoInOptions[0].NightOptions.GainRed=50
table.VideoInOptions[0].NightOptions.GlareInhibition=0
table.VideoInOptions[0].NightOptions.IrisAuto=true
table.VideoInOptions[0].NightOptions.Mirror=false
table.VideoInOptions[0].NightOptions.Profile=3
table.VideoInOptions[0].NightOptions.ReferenceLevel=50
table.VideoInOptions[0].NightOptions.Rotate90=0
table.VideoInOptions[0].NightOptions.SunriseHour=0
table.VideoInOptions[0].NightOptions.SunriseMinute=0
table.VideoInOptions[0].NightOptions.SunriseSecond=0
table.VideoInOptions[0].NightOptions.SunsetHour=23
table.VideoInOptions[0].NightOptions.SunsetMinute=59
table.VideoInOptions[0].NightOptions.SunsetSecond=59
table.VideoInOptions[0].NightOptions.SwitchMode=4
table.VideoInOptions[0].NightOptions.WhiteBalance=Auto
table.VideoInOptions[0].NightOptions.WideDynamicRange=0
table.VideoInOptions[0].NightOptions.WideDynamicRangeMode=0
table.VideoInOptions[0].NormalOptions.AntiFlicker=0
table.VideoInOptions[0].NormalOptions.Backlight=0
table.VideoInOptions[0].NormalOptions.BacklightRegion[0]=3096
table.VideoInOptions[0].NormalOptions.BacklightRegion[1]=3096
table.VideoInOptions[0].NormalOptions.BacklightRegion[2]=5096
table.VideoInOptions[0].NormalOptions.BacklightRegion[3]=5096
table.VideoInOptions[0].NormalOptions.BrightnessThreshold=50
table.VideoInOptions[0].NormalOptions.DayNightColor=1
table.VideoInOptions[0].NormalOptions.ExposureMode=0
table.VideoInOptions[0].NormalOptions.ExposureSpeed=0
table.VideoInOptions[0].NormalOptions.ExposureValue1=0
table.VideoInOptions[0].NormalOptions.ExposureValue2=40
table.VideoInOptions[0].NormalOptions.ExternalSyncPhase=125
table.VideoInOptions[0].NormalOptions.Flip=false
table.VideoInOptions[0].NormalOptions.Gain=50
table.VideoInOptions[0].NormalOptions.GainAuto=true
table.VideoInOptions[0].NormalOptions.GainBlue=50
table.VideoInOptions[0].NormalOptions.GainGreen=50
table.VideoInOptions[0].NormalOptions.GainMax=50
table.VideoInOptions[0].NormalOptions.GainMin=0
table.VideoInOptions[0].NormalOptions.GainRed=50
```

```
table.VideoInOptions[0].NormalOptions.GlareInhibition=0  
table.VideoInOptions[0].NormalOptions.IrisAuto=true  
table.VideoInOptions[0].NormalOptions.Mirror=false  
table.VideoInOptions[0].NormalOptions.Profile=0  
table.VideoInOptions[0].NormalOptions.ReferenceLevel=50  
table.VideoInOptions[0].NormalOptions.Rotate90=0  
table.VideoInOptions[0].NormalOptions.SunriseHour=0  
table.VideoInOptions[0].NormalOptions.SunriseMinute=0  
table.VideoInOptions[0].NormalOptions.SunriseSecond=0  
table.VideoInOptions[0].NormalOptions.SunsetHour=23  
table.VideoInOptions[0].NormalOptions.SunsetMinute=59  
table.VideoInOptions[0].NormalOptions.SunsetSecond=59  
table.VideoInOptions[0].NormalOptions.SwitchMode=0  
table.VideoInOptions[0].ReferenceLevel=50  
table.VideoInOptions[0].ReferenceLevelEnable=false  
table.VideoInOptions[0].Rotate90=0  
table.VideoInOptions[0].SignalFormat=BT656  
table.VideoInOptions[0].WhiteBalance=Disable  
...
```

HTTP API V3.5 - Lebras

Set Config Request Example

Set Auto Exposure:

`http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&VideoInOptions[0].ExposureMode=0&VideoInOptions[0].ExposureSpeed=0`

Set Low Noise:

`http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&VideoInOptions[0].ExposureMode=1&VideoInOptions[0].ExposureSpeed=0&VideoInOptions[0].GainMin=0&VideoInOptions[0].GainMax=60`

Set Low Motion Blur:

`http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&VideoInOptions[0].ExposureMode=2&VideoInOptions[0].ExposureSpeed=0&VideoInOptions[0].GainMin=0&VideoInOptions[0].GainMax=50&VideoInOptions[0].ExposureValue1=0&VideoInOptions[0].ExposureValue2=20`

Set Manual:

`http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&VideoInOptions[0].ExposureMode=4&VideoInOptions[0].ExposureSpeed=32&VideoInOptions[0].GainMin=0&VideoInOptions[0].GainMax=50&VideoInOptions[0].ExposureValue1=40&VideoInOptions[0].ExposureValue2=40`

Set SmartIRExposure:

`http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&VideoInOptions[0].SmartIRExposure=true`

Set Video Rotate:

Flip:

`http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&VideoInOptions[0].Flip=true`

Mirror:

`http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&VideoInOptions[0].Mirror=true`

Or turn 90°:

`http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&VideoInOptions[0].Rotate90=1`

Set White Balance:

`http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&VideoInOptions[0].WhiteBalance=Night`

Or

`http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&VideoInOptions[0].WhiteBalance=Custom&VideoInOptions[0].GainRed=50&VideoInOptions[0].GainBlue=50&VideoInOptions[0].GainGreen=50`

(Sometimes you should set mode first before set GainRed or GainBlue:

`http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&VideoInOptions[0].WhiteBalance=Custom)`

Set Config Response Example

OK

HTTP API V3.35 - Intelbras

6.1 Storage Devices

6.1.1 Get Hard Disk Information

Get the hard disk information.

Request URL	http://<server>/cgi-bin/storageDevice.cgi?action=factory.getPortInfo		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
Request Example			
http://192.168.1.108/cgi-bin/storageDevice.cgi?action=factory.getPortInfo			

Response Params (key=value format in body)			
Name	Type	R/O	Description
into	object	R	Return the hard disk information
+Total	uint	O	The number of ports of the bus, including the expansion bus.
+Plug	uint	O	The number of the mounted IDE (damaged IDEs are excluded)
+Mask	uint64	O	Indicates whether there is an IDE (including damaged IDE) mounted on the bus channel. Each bit represents the main IDE or the sub IDE of the bus channel. If the bit is set to "1", there is a mounted IDE. If the bit is set to "0", there is no mounted IDE. For example, bit 0 represents the main IDE of IDE0, bit1 represents the sub IDE of IDE0, bit2 represents the main IDE of IDE1, bit3 represents the sub IDE of IDE1, and more.
+Bad	uint	O	Indicates whether there is a damaged IDE mounted on each channel. Each bit represents the main or sub IDE of each channel. If the bit is set to "1", there is a damaged IDE. If the bit is set to "0", there is no damaged IDE. See Mask for the corresponding relationship.
+IDE	uint	O	The number of mounted IDE
+Esata	uint	O	The total port number of the eSATA
Response Example			

```

info.Total=2
info.Plug=1
info.Mask=1
info.Bad=0
info.IDE=1
info.Esata=4

```

6.1.2 Get the Name of All Storage Devices

Get the name of all storage devices.

Request URL	http://<server>/cgi-bin/storageDevice.cgi?action=factory.getCollect		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
Request Example			
http://192.168.1.108/cgi-bin/storageDevice.cgi?action=factory.getCollect			

Response Params (key=value format in body)			
Name	Type	R/O	Description
list	char[][64]	R	Return the name of the storage device
Response Example			
list[0]=/dev/sda0 list[1]=/dev/sda1 list[2]=/dev/sg1			

6.1.3 Get Storage Device Information

Get all the storage device information .

Request URL	http://<server>/cgi-bin/storageDevice.cgi?action=getDeviceAllInfo		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
Request Example			
http://192.168.1.108/cgi-bin/storageDevice.cgi?action=getDeviceAllInfo			

Response Params (key=value format in body)			
Name	Type	R/O	Description
list	object	R	Return object
+info	object[]	R	Object array of the storage device information
++Detail	object[]	O	Partition information
+++IsError	bool	O	Whether the partition is abnormal
+++Pointer	uint	O	Partition operation handle
+++TotalBytes	double	O	Total partition space; unit: Byte
+++Type	char[16]	O	Partition type

			"ReadWrite": Read and write partition "ReadOnly": Read only partition	
+++Path	char[128]	O	Partition name	"/mnt/dvr/sda0"
+++UsedBytes	double	O	Occupied partition space; unit: Byte	
++Name	char[32]	R	Device name	"/dev/sda"
++State	char[4]	R	Device status "Error": Failed to get the device status "Initializing": Getting the device status "Success": Successfully get the device status	"Success"

Response Example

```
list.info[0].Detail[0].IsError=false
list.info[0].Detail[0].Pointer=27023434
list.info[0].Detail[0].TotalBytes=0
list.info[0].Detail[0].Type=ReadWrite
list.info[0].Detail[0].Path=/mnt/dvr/sda0
list.info[0].Detail[0].UsedBytes=0
list.info[0].Name=/dev/sda
list.info[0].State=Success
```

6.1.4 Get Storage Capability

Request URL	http://<server>/cgi-bin/storage.cgi?action=getCaps			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
Request Example				
http://192.168.1.108/cgi-bin/storage.cgi?action=getCaps				

Response Params (key=value format in body)				
Name	Type	R/O	Description	Example
caps	object	R	Storage Capabilities	
+RedundantDisk	object	O	redundant	2
++Support	bool	O	does support	false
+SupportRemoteLimit	bool	O	does support remote storage limit	false
imit			default false	
Response Example				
caps.RedundantDisk.Support=false				
caps.SupportRemoteLimit=true				

6.1.5 Format Camera SD-Card

Request URL	http://<server>/cgi-bin/storageDevice.cgi?action=setDevice			
Method	GET			
Request Params (key=value format in URL)				

Name	Type	R/O	Description	Example
type	char[]	R	operation FormatPartition: format partition	FormatPartition
path	char[]	O	The value is got from cgi API "Get storage device information" (/cgi-bin/storageDevice.cgi?action=getDeviceAllInfo).	/dev/sda&
formattype	object[]	O	disk part information	
+fs	char[]	R	file system type	fat32

Request Example
http://192.168.1.108/cgi-bin/storageDevice.cgi?action=setDevice&type=FormatPartition&path=/dev/sda&formattype[0].fs=fat32

Response Params (OK in body)
Response Example
OK

6.1.6 [Config] Hard Disk Recording Type

Config Data Params				
Name	Type	R/O	Description	Example
SupportDiskRecordType	object	O	Supported disk recording type info.	
+DiskRecordType	char[32]	O	Disk recording mode "SMR", "CMR"	"SMR"

Please refer to "4.2.1 Get and Set Configure" for configuration getting and setting. Specific examples are as follows :

Get Config Request Example
http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=SupportDiskRecordType
Get Config Response Example
table.SupportDiskType.DiskRecordType =SMR

Set Config Request Example
http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&SupportDiskRecordType.DiskRecordType=SMR
Set Config Response Example
OK

6.1.7 Getting Disk Information

Request URL	http://<server>/cgi-bin/api/StorageDeviceManager/getDeviceInfos
Method	POST
Request Params (JSON format in body)	

Name	Type	R/O	Description	Example
volume	char[32]	R	<p>Volume type enumchar[32]{ "PhysicalVolume": physical volume (includes physical disk in RAID and VG, and includes USB flash drive). "IndividualPhysicalVolume": Individual physical volume (does not participate in RAID, virtual, and clone groups). "RaidVolume": Raid volume "IndividualRaidVolume": individual RAID volume (does not participate in virtual and other groups) "VolumeGroup": VG virtual volume group "iSCSI": iSCSI volume "GlobalSpareVolume": global hot standby volume "NAS": NAS volume (Includes FTP, SAMBA, NFS) } Keep consistent with the definition of Device Information Volume type value is "", get the information of all volumes, including IndividualPhysical, IndividualRaid、VolumeGroup </p>	"PhysicalVolume"

Request Example

```
{
  "volume": "PhysicalVolume"
}
```

Response Params (JSON format in body)				
Name	Type	R/O	Description	Example
device	object[256]	O	Disk Information	[, ...,]
+Name	char[32]	O	Device name	"/dev/sda"
+Media	char[]	O	Media type "CDROM": disk "DISK": hard drive "FLASH": SD card "FROCK": frock "UNKNOWN": unknown	"CDROM"
+BUS	char[8]	O	Bus type, supporting "ATA", "SATA", "USB", "SDIO", "SCSI", "SAS" and more.	"ATA"
+PhysicNo	uint	O	Physical number of the similar device storage port, starting from 1.	1

+LogicNo	uint	O	Logic number of the similar device storage port	0
+Capacity	double	O	Total capacity, Unit: Byte	314537952
+Volume	enumchar[32]	O	Volume type enumchar[32]{ "PhysicalVolume": independent physical volume/global hot standby "RaidVolume": Raid volume (Includes sub-disk of the soft Raid and local hot standby) "VolumeGroup": VG virtual volume group (once VG is composed of single disk or Raid, then the result will be VG) "iSCSI": iSCSI volume }	"RaidVolume"
+State	enumchar[32]	O	Physical hard disk status information enumchar[32]{ "Error": general faults. Some partition still can be used. "Fatal": serious faults, all of the partitions cannot be used (new DVR fault type) "Offline": offline status of physical disks (not used) "Running": the operating status of the physical disk "RaidSub": the disk now is a single disk and it was a Raid sub disk before. It is likely to automatically add Raid after the device is restarted. /*(the four status are used in single disk)*/ "Active": RAID active "Sync": RAID sync "Spare": RAID hot standby (local) "Faulty": RAID invalid "Rebuilding": RAID rebuild "Removed": RAID removed (not used) "WriteError": RAID write error "WantReplacement": RAID need to be replaces "Replacement": RAID is replacement /*(the 9 status are used in RAID sub disk)*/ "GlobalSpare": global hot standby /*(the status is used in global hot standby disk)*/ "SnapshotParent": snapshot parent "SnapshotChild": snapshot child }	"Running"

			<pre>"VolumeCloneParent": volume clone parent "VolumeCloneChild": volume clone child /*(the 4 status are used in snapshot/volume clone disk)*/ }</pre>	
+Parent	char[32]	O	Name of the storage physical device or name of the parent storage group of the virtual device. It might be VG or RAID.	"/dev/md0"
+Type	enumchar[32]	O	Disk feature (read and write, read-only) enumchar[32]{ "ReadWrite" "ReadOnly" }	"ReadWrite"
+Module	char[32]	O	Device module	"HCT7210SLA360"
+SerialNo	char[32]	O	Device serial number	"STH607MGAS"
+Temperature	float	O	Single disk temperature (unit: °C)	38.5
+Firmware	char[32]	O	Firmware version	"ST6OA31E"
+Partitions	object[]	O	Partition information Each partition has a WorkDirectory. When the disk has fault or is not formatted, the field does not exist.	
++Name	char[32]	O	Partition name	"/dev/sda0"
++Start	double	O	Start to offset. Unit: Byte	0
++Total	double	O	Total capacity. Unit: Byte	2000000000
++Remain	double	O	The remained capacity. Unit: Byte	1000000000
++MountOn	char[256]	O	Mount point. The App use this directory name to access files.	"/mnt/dvr/idea0"
++FileSystem	char[16]	O	File system, such as "DHFS", "EXT3", "EXT4", "XFS", "VFAT", "UNKNOWN" and more. Empty means it is not formatted.	"VFAT"
++IsSupportFs	bool	O	Whether the device supports the current file system.	"ture"
++Status	enumchar[32]	O	Partition status information enumchar[32]{ "LvAvailable": LV available status "LvNotAvailable": LV unavailable status	"LvAvailable"
++Group	char[32]	O	Partition working group	"ReadWrite2"
++IsError	bool	O	Whether the file system of the working directory fails	false
+Raid	object	O	RAID information, and it is only valid to RAID volume.	

++Members	char[][16]	O	RAID member, including sub disk and local hot standby disk.	["/dev/sda", "/dev/sdb", "/dev/sdc"]
++MemberInfos	object[]	O	RAID member information one-dimensional array, and the index is corresponding to Members.	
+++ID	uint16	O	Disk number, and it is used to describe the slot of the disk on the disk cabinet.	1
+++Spare	bool	O	Whether it is local hot standby. True: local hot standby False: RAID sub disk	true
++Level	uint8	O	RAID Level	5
++RaidDevices	uint16	O	Number of RAID devices	3
++TotalDevices	uint16	O	Total number of RAID devices	4
++State	char[][16]	O	RAID status, and it is a combination of various status. RAID sub disk status is achieved by State. Supports status "Active" "Inactive" "Clean" "Failed" "Degraded" "Recovering" "Resyncing" "Reshaping" "Checking" "NotStarted" "Readonly" "DataCorrupt" "CreatVG" RAID disk is being created to VG	["Active", "Degraded"]
++ActiveDevices	uint16	O	Number of active devices	3
++WorkingDevice s	uint16	O	Number of working devices	3
++FailedDevices	uint16	O	Number of failed devices	3
++SpareDevices	uint16	O	Number of hot standby devices	3
++RecoverPercent	double	O	Recover percent. Its is valid when "Recovering" or "Resyncing" is in RAID status. Value: [0, 100]	12.3
++RecoverMBps	double	O	Recover speed. Its is valid when "Recovering" or "Resyncing" is in RAID status. Unit: Mbps	29.531
++RecoverTimeRemain	double	O	Remaining time. When in RAID status, "Recovering" or "Resyncing" is valid. Unit: Minute (s)	30.0
++AliasName	char[24]	O	RAID alias, UTF-8 code.	"Image storage RAID"
++Sync	enumint8	O	Sync mode. I/O resources distributing strategy enumint8{ 0: Self-adaptive (The default value is) 1: I / O is prioritized to RAID synchronization. 2: I / O is prioritized to hard drive to write data. 3: Balance	0

			}	
+ISCSI	object	O	ISCSI information. It is only valid for ISCSI disk.	
++Name	char[16]	O	iSCSI name. It is the same with the Name in NAS configuration.	"iSCSIA"
+Tank	object	O	Information of the expansion drawer where the disk in (Note: it refers to Dahua expansion drawer, not the disk cabinet) You can get the information of the through storage.getTankInfo	
++Level	int	O	Control module is level 0.	1
++Slot	uint16	O	Board number on the related cabinet. (starting from 0). Each cabinet can be inserted on many expansion boards and each expansion drawer (such as Raid card) can cascade many cabinets.	0
++TankNo	uint16	O	Expansion port number of the expansion drawer at the same level (starting from 0). There may have many ports on the expansion board (such as SAS)	1
+Slot	int	O	Slot of the disk.	0
+PowerMode	enumchar[32]	O	Hard drive power status enumchar[32]{ "None": unknown status "Active": active status "StandBy": standby status "Idle": idle status }	"StandBy"
+PreDiskCheck	enumchar[32]	O	Hard drive pre-check status (EVS optional field, and uses with disk pre-check function) enumchar[32]{ "Good": read speed of the hard drive can be up to 120, and there are few mistakes in smart information. Nothing else is wrong. "Warn": there are several mistake records in cmd information and there are mistake records in smart information. "Error": there are several mistake records in cmd information and there are mistake records in smart information. There are records of bad	"Good"

			<p>sectors.</p> <p>"Willfail": hard drive speed is low, under 64 M. There are several mistake records in cmd information and there are mistake records in smart information. There are records of bad sectors.</p> <p>"Fail": the hard drive failed to return.</p> <p>"None": unknown status</p> <p>"Becheck": checking status</p> <p>"Checkfail": failed to check</p> <p>}</p>	
+OpState	enumint	O	<pre>enumint{ 0: Normal working status 1: Standby status 2: Waiting to be formatted 3: Formatting... 4: Waiting to be fragmentated 5: Fragmenting... 6: Waiting to create RAID 7: Creating RAID... 8: Waiting to delete RAID 9: Deleting RAID... 10: Waiting to fix file system 11: Fixing... 12: Waiting to pre-check 13: Pre-checking... 14: Configuring hot standby disk... 15: Create storage pool 16: Delete storage pool }</pre>	0
+ManuFactory	enumchar[32]	O	Hard drive device manufacturer <pre>enumchar[32]{ "Unknown" "WD": WD "SG" : Seagate }</pre>	"WD"
+PosLedState	enumint8	O	Status of the hard dive GPS indicator <pre>enumint8{ 0: The hard dive GPS indicator is off 1: The hard dive GPS indicator is on }</pre>	0
+CmrSize	uint32	O	It refers to CMR hard drive, and the unit is sector (513 bytes) Otherwise, the field is 0.	0
+MRTType	enumint	O	Disk recording method <pre>enumint{</pre>	0

			0: CMR 1: PMR 2: SMR }	
+MediaType	char[32]	O	Disk media type	"SSD"
+DRTTypeMixState	enumchar[32]	O	Hard drive recording method enumchar[32]{ "Single": single record method, "Mix": mixed record method. }	"Mix"

Example

```
{
  "device": {
    "Name" : "/dev/sda",
    "Media" : "CDROM",
    "BUS" : "ATA",
    "PhysicNo" : 1,
    "LogicNo" : 0,
    "Capacity" : 314537952,
    "Volume": "RaidVolume",
    "State": "Running",
    "Parent" : "/dev/md0",
    "Type" : "ReadWrite",
    "Module" : "HCT7210SLA360",
    "SerialNo" : "STH607MGAS",
    "Temperature" : 38.5,
    "Firmware" : "ST6OA31E",
    "Partitions" : [ {
      "Name" : "/dev/sda0",
      "Start" : 0,
      "Total" : 2000000000,
      "Remain" : 1000000000,
      "MountOn" : "/mnt/dvr/idea0",
      "FileSystem" : "VFAT",
      "IsSupportFs" : "ture",
      "Status" : "LvAvailable",
      "Group" : "ReadWrite2",
      "IsError" : false
    }, ..., {} ],
    "Raid" : {
      "Members" : ["/dev/sda", "/dev/sdb", "/dev/sdc"],
      "MemberInfos" : [
        {
          "ID" : 1,
          "Spare" : true,
        }, ..., {}
      ],
      "Level" : 5,
      "RaidDevices" : 3,
    }
  }
}
```

```

        "TotalDevices" : 4,
        "State" : ["Active", "Degraded"],
        "ActiveDevices" : 3,
        "WorkingDevices" : 3,
        "FailedDevices" : 3,
        "SpareDevices" : 3,
        "RecoverPercent" : 12.3,
        "RecoverMBps" : 29.531,
        "RecoverTimeRemain" : 30.0,
        "AliasName" : "image storage RAID"
        "Sync" : 0
    },//end of Raid
    "ISCSI" : {
        "Name" : "iSCSIA"
    }, // end of ISCSI
    "Tank" : {
        "Level" : 1,
        "Slot" : 0,
        "TankNo" : 1
    }, // end of Tank
    "Slot" : 0,
    "PowerMode" : "StandBy",
    "PreDiskCheck" : "Good",
    "OpState" : 0
    "ManuFactory": "WD",
    "PosLedState": 0,
    "CmrSize": 0,
    "MRTType": 0,
    "MediaType": "SSD",
    "DRTTypeMixState": "Mix"
}
}

```

HTTP API V3.35 - Intelbras

6.2 NAS

6.2.1 [Config] NAS Information

Config Data Params				
Name	Type	R/O	Description	Example
NAS	object[]	O		
+Name	char[128]	O	NAS name	"NFS1"
+Enable	bool	O	Enable/Disable the NAS.	true
+Protocol	enumchar[32]	O	The range is { "FTP", "SFTP" "NFS"	"NFS"

			"SMB" "ISCSI" "Cloud" }	
+CloudProtocol	enumchar[32]	O	The range is { "Baidu" "Dropbox" "Kuaipan" "SkyDrive" "EFS" }	"EFS"
+Address	char[128]	O	The IP address or host name.	"www.tech.com"
+Port	uint16	O	NAS port.	21
+UserName	char[32]	O	NAS username.	"anonymity"
+Password	char[32]	O	NAS password.	"none"
+Directory	char[64]	O	Directory name.	"share"

Please refer to "4.2.1 Get and Set Configure" for configuration getting and setting. Specific examples are as follows :

Get Config Request Example

```
http://10.0.0.8/cgi-bin/configManager.cgi?action=getConfig&name=NAS
```

Get Config Response Example

```
table.NAS[0].Name=FTP1
table.NAS[0].Enable = true
table.NAS[0].Protocol =FTP"
table.NAS[0].Address =www.ttt.com
table.NAS[0].Port =21
table.NAS[0].UserName =anonymity
table.NAS[0].Password =none
table.NAS[0].Directory =share
```

Set Config Request Example

```
http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&NAS[0].Name=nas01&NAS[0].Enable=true
```

Set Config Response Example

```
OK
```

6.3 Storage Point

6.3.1 [Config] Record Storage Point

Config Data Params				
Name	Type	R/O	Description	Example
RecordStoragePoint	object[]	O	array index starts from 0, which means video channel>equals to video channel	

			index -1, and so 0 means channel 1).	
+TimingRecord	object	O	Timed recording storage points, each type of recording or capture may have multiple storage points	
++Local	int/bool	O	Local directory number, count from 1. For some reason, The front-end devices use Boolean type. True for SD card, false for no SD card.	1
++Redundant	char[64]	O	Redundant directory name.	"Redundant"
++Remote	char[64]	O	Remote directory name.	"FTP-FTP1"
++AutoSync	bool	O	When remote directory recovers, auto synchronize local directory to remote directory or not.	false
++AutoSyncRange	int	O	From the remote directory recovering time, how long the data needs to be synchronized. The unit is hour. If it is 0, all the data needs to be synchronized.	0
++LocalForEmergency	bool	O	When the remote directory is unusable, save the data the local directory or not.	false
++CompressBefore	int	O	The days' data which will be compressed.	15
+ManualRecord	object	O	refer to "TimingRecord"	
+VideoDetectRecord	object	O	refer to "TimingRecord"	
+AlarmRecord	object	O	refer to "TimingRecord"	
+EventRecord	object	O	refer to "TimingRecord"	
+TimingSnapshot	object	O	refer to "TimingRecord"	
+ManualSnapshot	object	O	refer to "TimingRecord"	
+VideoDetectSnapshot	object	O	refer to "TimingRecord"	
+AlarmSnapshot	object	O	refer to "TimingRecord"	
+EventSnapshot	object	O	refer to "TimingRecord"	

Please refer to "4.2.1 Get and Set Configure" for configuration getting and setting. Specific examples are as follows :

Get Config Request Example

```
http://10.0.0.8/cgi-bin/configManager.cgi?action=getConfig&name=RecordStoragePoint
```

Get Config Response Example

```
table.RecordStoragePoint [0].TimingRecord.Local=1
table.RecordStoragePoint [0].TimingRecord. Redundant =Redundant
table.RecordStoragePoint [0].TimingRecord. Remote =FTP
table.RecordStoragePoint [0].TimingRecord. AutoSync= false
```

```

table.RecordStoragePoint[0].TimingRecord. AutoSyncRange =0
table.RecordStoragePoint[0].TimingRecord. LocalForEmergency =false
table.RecordStoragePoint[0].TimingRecord. CompressBefore =15

```

Set Config Request Example

```
http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&RecordStoragePoint[0].TimingRecord
.Local=local
```

Set Config Response Example

```
OK
```

6.3.2 [Config] Storage Group

Config Data Params				
Name	Type	R/O	Description	Example
StorageGroup	object[]	O	storage group	
+Name	char[32]	O	Storage group name.	"ReadWrite"
+Memo	char[256]	O	Storage group memo.	"For Reading Writing Files"
+FileHoldTime	int	O	How many days the file will hold.	0
+OverWrite	bool	O	Over write or not when there is not enough storage.	true
+Channels	object[]	O		
++MaxPictures	uint32	O	The max pictures beyond which the old pictures will be over written. If it is 0, the old pictures will be not overwritten.	10000
++Path	char[16]	O	The channel path.	"00"
+RecordPathRule	char[64]	O	record path naming rule	">%y-%M-%d/%c/dav/%e/%h/%h.%m.%s-%h.%m.%s.%t"
+PicturePathRule	char[256]	O	picture path naming rule	">%y-%M-%d/%c/jpg/%h/%m/%s.jpg"

Please refer to "4.2.1 Get and Set Configure" for configuration getting and setting. Specific examples are as follows :

Get Config Request Example

```
http://10.0.0.8/cgi-bin/configManager.cgi?action=getConfig&name=StorageGroup
```

Get Config Response Example

```

table.StorageGroup[0].Channels[0].MaxPictures=0
table.StorageGroup[0].FileHoldTime=0
table.StorageGroup[0].Memo=For Reading & Writing Files
table.StorageGroup[0].Name=ReadWrite
table.StorageGroup[0].OverWrite=true
table.StorageGroup[0].PicturePathRule=%y-%M-%d/%c/jpg/%h/%m/%s[%E][%O@%S][%R].jpg
table.StorageGroup[0].RecordPathRule=%y-%M-%d/%c/dav/%h/%h.%m.%s-%h.%m.%s[%E][%O@%S]

```

```
][%R].dav
table.StorageGroup[1].Channels[0].MaxPictures=0
table.StorageGroup[1].FileHoldTime=0
table.StorageGroup[1].Memo=For FTP Files
table.StorageGroup[1].Name=Remote
table.StorageGroup[1].OverWrite=true
table.StorageGroup[1].PicturePathRule=%y-%M-%d/%c/jpg/%h/%m/%s[%E][%O@%S][%R].jpg
table.StorageGroup[1].RecordPathRule=%y-%M-%d/%c/dav/%h/%h.%m.%s-%h.%m.%s[%E][%O@%S]
][%R].da
```

Set Config Request Example

[http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&StorageGroup\[0\].Name=main](http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&StorageGroup[0].Name=main)

Set Config Response Example

OK

6.4 SDEncrypt

6.4.1 Encrypt SD Card

Request URL	<a href="http://<server>/cgi-bin/SDEncrypt.cgi?action=encrypt">http://<server>/cgi-bin/SDEncrypt.cgi?action=encrypt			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
deviceName	char[]	R	The device name is got from cgi API Get storage device information (cgi-bin/storageDevice.cgi?action=getDeviceAllInfo).	/dev/mmc0
password	char[]	O	password	123456
Request Example				
http://192.168.1.108/cgi-bin/SDEncrypt.cgi?action=encrypt&deviceName=/dev/mmc0&password=123456				

Response Params (OK in body)

Response Example

OK

6.4.2 Decrypt SD Card

Request URL	<a href="http://<server>/cgi-bin/SDEncrypt.cgi?action=decrypt">http://<server>/cgi-bin/SDEncrypt.cgi?action=decrypt			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
deviceName	char[]	R	The device name is got from cgi API Get storage device information	/dev/mmc0

			(cgi-bin/storageDevice.cgi?action=getDeviceAllInfo).	
password	char[]	O	password	123456
Request Example				
http://192.168.1.108/cgi-bin/SDEncrypt.cgi?action=decrypt&deviceName=/dev/mmc0&password=123456				

Response Params (OK in body)

Response Example

OK

6.4.3 Clear SD Card Password

Request URL	http://<server>/cgi-bin/SDEncrypt.cgi?action=clearPassword			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
deviceName	char[]	R	The device name is got from cgi API Get storage device information (cgi-bin/storageDevice.cgi?action=getDeviceAllInfo).	/dev/mmc0
password	char[]	O	password	123456
Request Example				
http://192.168.1.108/cgi-bin/SDEncrypt.cgi?action=clearPassword&deviceName=/dev/mmc0&password=123456				

Response Params (OK in body)

Response Example

OK

6.4.4 Modify SD Card Password

Request URL	http://<server>/cgi-bin/SDEncrypt.cgi?action=modifyPassword			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
deviceName	char[]	R	The device name is got from cgi API Get storage device information (cgi-bin/storageDevice.cgi?action=getDeviceAllInfo).	/dev/mmc0
password	char[]	O	new password	123456
oldPassword	char[]	O	old password	654321
Request Example				
http://192.168.1.108/cgi-bin/SDEncrypt.cgi?action=clearPassword&deviceName=/dev/mmc0&password=123456&oldPassword=654321				

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

OK

6.4.5 Get SD Card Operate Error Policy

Request URL	http://<server>/cgi-bin/SDEncrypt.cgi?action=getOperateErrorPolicy			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
deviceName	char[]	R	The device name is got from cgi API Get storage device information (cgi-bin/storageDevice.cgi?action=getDeviceAllInfo).	/dev/mmc0
operate	char[]	R	operation, range{ decrypt, modifyPassword, clearPassword}	decrypt
Request Example				
http://192.168.1.108/cgi-bin/SDEncrypt.cgi?action=getOperateErrorPolicy&deviceName=/dev/mmc0&operate=decrypt				

Response Params (key=value format in body)

Name	Type	R/O	Description	Example
policy	object	R	policy info	
+leftTimes	uint	O	remain operate times, max is 5	5
+lockSeconds	uint	O	lock operate time, unit is seconds, max is 30	30
Response Example				
policy.leftTimes=5				
policy.lockSeconds=30				

6.4.6 [Config] Storage Health Alarm Settings

Config Data Params				
Name	Type	R/O	Description	Example
StorageHealthAlarm	object	O	SD card health alarm	
+Enable	bool	R	enable/disable	true
+LowerLimit	int	O	SD card remaining life, alarm below this value: 0~99	10
+EventHandler	EventHandler	O	Setting of EventHandler is described in SetEventHandler .	

Please refer to “4.2.1 Get and Set Configure” for configuration getting and setting. Specific examples are as follows :

Get Config Request Example

http://10.0.0.8/cgi-bin/configManager.cgi?action=getConfig&name=StorageHealthAlarm

Get Config Response Example

```
table.StorageHealthAlarm.Enable=true  
table.StorageHealthAlarm.LowerLimit=10  
table.StorageHealthAlarm.EventHandler= ... (output of EventHandler is described in GetEventHandler)
```

Set Config Request Example

<http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&StorageHealthAlarm.Enable=true>

Set Config Response Example

OK

HTTP API V3.35 - Intelbras

7.1 GUI

7.1.1 [Config] GUISet

Config Data Params				
Name	Type	R/O	Description	Example
GUISet	object[]	O	array index starts from 0, which means video channel>equals to video channel index -1, and so 0 means channel 1).	
+WindowAlpha	uint8	O	Diaphaneity of the window background.	128
+TimeTitleEnable	bool	O	Show the time title or not.	true
+TimeTitlePos	uint[4]	O	The position of the time title.	
+MenuShowOption	enumint	O	0: Show the directory. 1: Hide the directory. 2: Timing-hide the directory.	0
+MenuAutoHideTime	uint8	O	How many seconds to hide the directory.	3
+ScreenSaveTime	uint8	O	Screen saver time, in minutes 0~120, 0 for disable screen saver	10
+AutoLogout	uint8	O	How many minutes to auto logout. The range is [0-120]. 0 expresses not logout.	10
+ChannelTitleShowEnable	bool	O	Show the channel title or not.	true
+ChannelTitlePos	uint[4]	O	The position of the channel title.	
+AutoGuideEnable	bool	O	Auto guide or not when startup.	true

Please refer to “4.2.1 Get and Set Configure” for configuration getting and setting. Specific examples are as follows :

Get Config Request Example

```
http://10.0.0.8/cgi-bin/configManager.cgi?action=getConfig&name=GUISet
```

Get Config Response Example

```
table.GUISet[0].WindowAlpha =128
table.GUISet[0].TimeTitleEnable =true
table.GUISet[0].TimeTitlePos[0]=0
table.GUISet[0].TimeTitlePos[1]=0
```

```

table.GUISet[0].TimeTitlePos[2]=8191
table.GUISet[0].TimeTitlePos[3]=8191
table.GUISet[0].MenuShowOption=0
table.GUISet[0].MenuAutoHideTime=10
table.GUISet[0].AutoLogout=10
table.GUISet[0].ChannelTitleShowEnable=true
table.GUISet[0].ChannelTitlePos[0]=0
table.GUISet[0].ChannelTitlePos[1]=0
table.GUISet[0].ChannelTitlePos[2]=8191
table.GUISet[0].ChannelTitlePos[3]=8191
table.GUISet[0].AutoGuideEnable=true
...

```

Set Config Request Example

`http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&GUISet[0].WindowAlpha=192&GUISe
t[0].TimeTitleEnable=false&GUISet[0].MenuShowOption=1`

Set Config Response Example

OK

7.2 Split Screen

7.2.1 Split Screen Mode

- Get split screen mode

Request URL	<code>http://<server>/cgi-bin/split.cgi?action=getMode</code>		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
channel	uint	R	the display screen No. Start from 1 and <= 2.
Request Example			
<code>http://192.168.1.108/cgi-bin/storage.cgi?action=getCaps</code>			

Response Params (key=value format in body)			
Name	Type	R/O	Description
mode	char[]	R	mode, range{split1,split2,split4,split6,sp lit8,split9,split12,split16,split20,split25,s plit36,split64,split144,pip1,pip3, "Free", "CompositeSplit1" / "FitDisplayUnit1", "CompositeSplit1" / "FitDisplayUnit4"};
group	uint	R	the No. of a group which contains certain number channels. For example, if 16 video channels display in split4 Mode which contains 4 video channels on Screen, then there are 4 groups and

		each group contains 4 video channels.	
Response Example			
mode=split1 group=4			

- Set split screen mode

Request URL	http://<server>/cgi-bin/split.cgi?action=setMode		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
channel	uint	R	the display screen No. Start from 1 and <= 2.
mode	char[]	R	mode, range{split1,split2,split4,split6,split8,split9,split12,split16,split20,split25,split36,split64,split144,pip1,pip3, "Free", "CompositeSplit1" / "FitDisplayUnit1", "CompositeSplit1" / "FitDisplayUnit4"};
group	uint	R	the No. of a group which contains certain number channels. For example, if 16 video channels display in split4 Mode which contains 4 video channels on Screen, then there are 4 groups and each group contains 4 video channels.

Request Example

http://192.168.1.108/cgi-bin/split.cgi?action=setMode&channel=1&mode=split4&group=1

Response Params (OK in body)
Response Example
OK

7.3 Moniter Tour

7.3.1 [Config] Moniter Tour

Config Data Params				
Name	Type	R/O	Description	Example
MonitorTour	object[]	O	array index starts from 0, which means video channel>equals to video channel index -1, and so 0 means channel 1).	
+Enable	bool	O	MonitorTour or not.	true
+Interval	uint	O	MonitorTour interval.	5
+Mask	object	O		
++Split1	char[]	O	Channel array for split1	"0, 1, 3"
++Split8	char[]	O	Channel array for split8	"0, 1, 3"
+Collections	char[] [32]	O	Split collections	["Favortite1", "Split4.Favortite2"]

Please refer to "4.2.1 Get and Set Configure" for configuration getting and setting. Specific examples are as follows :

Get Config Request Example

```
http://10.0.0.8/cgi-bin/configManager.cgi?action=getConfig&name=MonitorTour
```

Get Config Response Example

```
table.MonitorTour[0].Enable=true
table.MonitorTour[0].Interval=128
table.MonitorTour[0].Mask.Split1=0,1,5
table.MonitorTour[0].Mask.Split8=0,1,5
table.MonitorTour[0].Collections=Favortite1, Favortite2...
```

Set Config Request Example

```
http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&MonitorTour[0].Enable=true
```

Set Config Response Example

```
OK
```

7.3.2 Enable Tour

Request URL	http://<server>/cgi-bin/split.cgi?action=enableTour			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
channel	uint	R	the display screen No. Start from 1 and <= 2.	1
enable	bool	R	true or false	true
Request Example				
http://192.168.1.108/cgi-bin/split.cgi?action=enableTour&channel=1&enable=true				

Response Params (OK in body)
Response Example
OK

7.3.3 [Config] Monitor Collection

Config Data Params				
Name	Type	R/O	Description	Example
MonitorCollection	object	O	monitor collection	
+< collectionname >	object	O	Favorite objects, describing them with scenes Each collection corresponds to a configuration, and the name of the collection object can be specified by the user collectionname is a variable field name	

++Mode	char[]	O	The range is the same as SetSplitMode.	Split1
++Windows	object[]	O	windows	
+++Enable	bool	O	Enable the window or not.	true
+++Device	char	O	The device Id.	device1
+++VideoChannel	uint	O	The video channel.	5
+++VideoStream	char[]	O	The range is {"Main", "Extra1", "Extra2", "Extra3", "Auto"}.	Main
+++AudioChannel	uint	O	The audio channel.	5
+++AudioStream	char[]	O	The range is {"Main", "Extra1", "Extra2", "Extra3", "Auto"}.	Main

Please refer to "4.2.1 Get and Set Configure" for configuration getting and setting. Specific examples are as follows :

Get Config Request Example

```
http://10.0.0.8/cgi-bin/configManager.cgi?action=getConfig&name=MonitorCollection
```

Get Config Response Example

```
table.MonitorCollection.collectionname.Mode=Split1
table.MonitorCollection.collectionname.Windows[0].Enable=true
table.MonitorCollection.collectionname.Windows[0].Device=device1
table.MonitorCollection.collectionname.Windows[0].VideoChannel=5
table.MonitorCollection.collectionname.Windows[0].VideoStream=Main
table.MonitorCollection.collectionname.Windows[0].AudioChannel=5
table.MonitorCollection.collectionname.Windows[0].AudioStream=Main
...
...
```

Set Config Request Example

```
http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&MonitorCollection.Favorite1.Mode=split4&MonitorCollection.Favorite1.Windows[1].Enable=true&MonitorCollection.Favorite1.Windows[1].VideoChannel=2
```

Set Config Response Example

```
OK
```

8.1 PTZ

8.1.1 [Config] PTZ Config

Config Data Params				
Name	Type	R/O	Description	Example
Ptz	object[]	O		
+ProtocolName	char[32]	O	PTZ protocol name depends on PTZ capability. Refer to GetProtocolList to get the protocol list.	"DH_SD"
+Address	int	O	Range is [0—255]. Device address, if there are more than one device connected to this port, distinguish them by this address.	8
+Attribute	multiTypeArray	O	Serial Port Properties The first value represents the baud rate, Range is {1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200}. The second value represents the data bit, Range is {4, 5, 6, 7, 8}. The third value represents the parity bit, Range is {Even, Mark, None, Odd, Space}. The fourth value represents the stop bit, Range is {1, 1.5, 2}.	[115200, 8, "Even", "1"]
+ControlPriority	enumchar[32]	O	PTZ control priority, the range is "RS485", "Net". default : "Net"	"RS485"
+ControlDelayTime	uint	O	PTZ control delay time, unit: second	10
+Homing	int[2]	O	The first value represents the preset point corresponding to automatic homing, with a value range of {-1,0-255}, -1: Indicates that automatic homing is not supported [0-255]: Preset point The second value represents the automatic homing time, which is in seconds when there is no operation	[0, 30]

		for a period of time. The value range is [0-65535]	
--	--	--	--

Please refer to "4.2.1 Get and Set Configure" for configuration getting and setting. Specific examples are as follows :

Get Config Request Example

```
http://10.0.0.8/cgi-bin/configManager.cgi?action=getConfig&name=Ptz
```

Get Config Response Example

```
table.Ptz[0].Address=8
table.Ptz[0].Attribute[0]=115200
table.Ptz[0].Attribute[1]=8
table.Ptz[0].Attribute[2]=Even
table.Ptz[0].Attribute[3]=1
table.Ptz[0].Homing[0]=0
table.Ptz[0].Homing[1]=30
table.Ptz[0].NumberInMatrixs=0
table.Ptz[0].ProtocolName=NONE
```

Set Config Request Example

```
http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&Ptz[0].Address=192.168.0.1&Ptz[0].Attribute[0]=9600
```

Set Config Response Example

```
OK
```

8.1.2 Get PTZ Protocol List

Request URL	http://<server>/cgi-bin/ptz.cgi?action=getProtocolList		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
channel	int	O	video channel index which starts from 1.
Request Example			
http://192.168.1.108/cgi-bin/ptz.cgi?action=getProtocolList&channel=1			

Response Params (key=value format in body)				
Name	Type	R/O	Description	Example
into	object	R	the protocol list that PTZ can support	
+RS	char[]	O	RS485	["Pelco", "DH-SD1"]

Response Example
info.RS[0]=Pelco info.RS[1]=DH-SD1 info.Coaxial[0]=HD-CVI info.Coaxial[1]=HD-CVI2.0

8.1.3 Get PTZ Capability of Current Protocol

Request URL	http://<server>/cgi-bin/ptz.cgi?action=getCurrentProtocolCaps			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
channel	int	O	video channel index which starts from 1.	1
Request Example				
http://192.168.1.108/cgi-bin/ptz.cgi?action=getProtocolList&channel=1				

Response Params (key=value format in body)				
Name	Type	R/O	Description	Example
caps	object	R	capabilities	
+AlarmLen	uint	O	Alarm length in protocol.	4
+AuxMax	int	O	Maximum number for auxiliary functions.	255
+AuxMin	int	O	Minimum number for auxiliary functions.	1
+CamAddrMax	int	O	Maximum channel address.	255
+CamAddrMin	int	O	Minimum channel address.	1
+Flip	bool	O	True or false, support picture flip or not.	false
+Focus	bool	O	True or false, support focus or not.	true
+Iris	bool	O	True or false, support Iris adjusts or not.	true
+Menu	bool	O	True or false, support internal menu of the PTZ or not.	true
+MonAddrMax	uint	O	Maximum monitor address.	255
+MonAddrMin	uint	O	Minimum monitor address.	1
+Name	char[]	O	Name of the operation protocol.	"DH-SD"
+Pan	bool	O	True or false, support pan or not.	
+PanSpeedMax	uint	O	Maximum pan speed.	255
+PanSpeedMin	uint	O	Minimum pan speed.	0
+PatternMax	uint	O	Maximum pattern path number.	8
+PatternMin	uint	O	Minimum pattern path number.	1
+PresetMax	uint	O	Maximum preset point number.	255
+PresetMin	uint	O	Minimum preset point number.	1
+Tile	bool	O	True or false, support tilt or not.	true
+Zoom	bool	O	True or false, support zoom or not.	true
+TileSpeedMin	uint	O	Maximum tile speed.	2
+TileSpeedMax	uint	O	Minimum tile speed.	25
+TourMin	uint	O	Maximum tour path number.	1
+TourMax	uint	O	Minimum tour path number.	255
+Type	uint	O	Type of PTZ protocol.	0
+ PtzMotionRange	object	O	range	
++HorizontalAngle	uint[2]	O	Horizontal angle range, [0] for minimum angle, [1] for maximum angle it only when Pan was true	[0, 360]
++VerticalAngle	int[2]	O	Vertical angle range, [0] for minimum angle, [1] for maximum angle	[-20, 90]

			it only when Tile was true	
+ZoomMax	uint	O	Maximum Zoom. it only when Zoom was true	20
+ZoomMin	uint	O	Minimum Zoom it only when Zoom was true	1

Response Example

```

caps.AlarmLen=0
caps.AuxMax=8
caps.AuxMin=1
caps.CamAddrMax=255
caps.CamAddrMin=1
caps.Flip=false
caps.Focus=false
caps.Interval=200
caps.Iris=false
caps.Menu=false
caps.MonAddrMax=255
caps.MonAddrMin=0
caps.Name=DH-SD1
caps.Pan=false
caps.PanSpeedMax=255
caps.PanSpeedMin=1
caps.PatternMax=5
caps.PatternMin=1
caps.PresetMax=80
caps.PresetMin=1
caps.Tile=false
caps.TileSpeedMax=255
caps.TileSpeedMin=1
caps.TourMax=7
caps.TourMin=0
caps.Type=1
caps.Zoom=false
caps.PtzMotionRange.HorizontalAngle[0]=0
caps.PtzMotionRange.HorizontalAngle[1]=360
caps.PtzMotionRange.VerticalAngle[0]=-20
caps.PtzMotionRange.VerticalAngle[1]=90
caps.ZoomMax=30
caps.ZoomMin=1

```

HTTP API V3.35 - Intelbras

8.1.4 Get PTZ Status

Request URL	http://<server>/cgi-bin/ptz.cgi?action=getStatus			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
channel	int	O	video channel index which starts from 1.	1

Request Example

http://192.168.1.108/cgi-bin/ptz.cgi?action=getProtocolList&channel=1

Response Params (key=value format in body)

Name	Type	R/O	Description	Example
status	object	R	ptz status	
+UTC	uint	O	UTC	6538920
MoveStatus	char[8]	O	range{"Idle", "Moving", "Unknown"}	"Idle"
ZoomStatus	char[8]	O	Zoom status range{ "Idle", "Zooming", "Unknown" }	"Idle"
PresetID	int	O	preset id	10
Position	double[3]	O	Comma separated numeric string The first element represents the horizontal angle, Normalize values to -1~1 The first element represents the vertical angle, Normalize values to -1~1 The first element represents the magnificationNormalize values to 0~1	[0.0, 0.0, 1.0]

Response Example

status.UTC=6538920
status.MoveStatus=Idle
status.ZoomStatus=Idle
status.PresetID=10
status.Position[0]=0.0
status.Position[1]=0.0
status.Position[2]=1.0

8.1.5 PTZ Control

- PTZ Basic Movement

Start moving the PTZ.

Request URL	http://<server>/cgi-bin/ptz.cgi?action=start			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
channel	int	R	The PTZ channel index; starting from 1	1
code	char[16]	R	See the following table for the operation codes for PTZ movement.	"Up"
arg1	int	O	Operation parameter 1: See the following table for the meaning of the operation code.	0
arg2	int	O	Operation parameter 2: See the following table for the meaning of the operation code.	1
arg3	int	O	Operation parameter 3: See the following table for the meaning of the operation code.	0

Request Example

<http://192.168.1.108/cgi-bin/ptz.cgi?action=start&channel=1&code=Up&arg1=0&arg2=1&arg3=0>

Response Params (OK in body)

Response Example

OK

Stop moving the PTZ.

Request URL http://<server>/cgi-bin/ptz.cgi?action=stop

Method GET

Request Params (key=value format in URL)

Name	Type	R/O	Description	Example
channel	int	R	The PTZ channel index; starting from 1	1
code	char[16]	R	See the following table for the operation codes for PTZ movement.	"Up"
arg1	int	O	Operation Parameter 1, reserved.	0
arg2	int	O	Operation Parameter 2, reserved.	0
arg3	int	O	Operation Parameter 3, reserved.	0

Request Example

<http://192.168.1.108/cgi-bin/ptz.cgi?action=stop&code=Up&channel=1&arg1=0&arg2=0&arg3=0>

Response Params (OK in body)

Response Example

OK

Appendix: Operation codes for PTZ movement and the parameters.

Code	Code description	arg1	arg2	arg3
Up	Move up	0	Vertical motion speed; range: [1–8]	0
Down	Move down	0	Vertical motion speed; range: [1–8]	0
Left	Move left	0	Horizontal motion speed; range: [1–8]	0
Right	Move right	0	Horizontal motion speed; range: [1–8]	0
LeftUp	Move in an upper-left direction	Vertical motion speed; range: [1–8]	Horizontal motion speed; range: [1–8]	0
RightUp	Move in an upper-right direction	Vertical motion speed; range: [1–8]	Horizontal motion speed; range: [1–8]	0
LeftDown	Move in an lower-left direction	Vertical motion speed; range: [1–8]	Horizontal motion speed; range: [1–8]	0
RightDown	Move in an lower-right direction	Vertical motion speed; range: [1–8]	Horizontal motion speed; range: [1–8]	0
ZoomWide	Zoom in	0	0	0
ZoomTele	Zoom out	0	0	0
FocusNear	Focus (near-field)	0	0	0
FocusFar	Focus (far-field)	0	0	0

Code	Code description	arg1	arg2	arg3
IrisLarge	Increase the aperture	0	0	0
IrisSmall	Decrease the aperture	0	0	0

- PTZ Continuously Moving

Start continuously moving the PTZ

Request URL	http://<server>/cgi-bin/ptz.cgi?action=start&code=Continuously			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
channel	int	R	The PTZ channel index; starting from 1	1
code	char[16]	R	PTZ continuously moving operation code, must be "Continuously".	"Continuously"
arg1	int	O	See the table below for the motion direction and step length.	5
arg2	int	O	See the table below for the motion direction and step length.	5
arg3	int	O	Zooming speed; range: [-100–100]	5
arg4	int	O	Overtime period; unit: s (maximum 3600 s). If the PTZ does not receive the stop command before the preset overtime period, it will stop moving automatically.	60

Request Example

http://192.168.1.108/cgi-bin/ptz.cgi?action=start&code=Continuously&channel=1&arg1=5&arg2=5&arg3=5&arg4=60

Response Params (OK in body)

Response Example

OK

Appendix: Direction of the continuous movement of the PTZ and the step length parameters.

Move description	arg1	arg2
Continuously move left	< -4	0
Continuously move right	> 4	0
Continuously move up	0	> 4
Continuously move down	0	< -4
Continuously move in an upper-left direction	< -4	> 4
Continuously move in an upper-right direction	> 4	> 4
Continuously move in a lower-left direction	< -4	< -4
Continuously move in a lower-right direction	> 4	< -4

Stop continuously moving the PTZ

Request URL	http://<server>/cgi-bin/ptz.cgi?action=stop&code=Continuously			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
channel	int	R	The PTZ channel index; starting from 1	1
code	char[16]	R	PTZ continuously moving operation code, must be "Continuously".	"Continuously"
arg1	int	O	Operation Parameter 1, reserved.	0
arg2	int	O	Operation Parameter 2, reserved.	0
arg3	int	O	Operation Parameter 3, reserved.	0
arg4	int	O	Operation Parameter 4, reserved.	0
Request Example				
http://192.168.1.108/cgi-bin/ptz.cgi?action=stop&code=Continuously&channel=1&arg1=0&arg2=0&arg3=0&arg4=0				

Response Params (OK in body)
Response Example
OK

- 3D Positioning

The PTZ moves to the specified position [startX, startY], [endX, endY] on the screen through 3D positioning .

Request URL	http://<server>/cgi-bin/ptzBase.cgi?action=moveDirectly			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
channel	int	R	The PTZ channel index; starting from 1	1
startPoint	int[2]	R	The start point of the target matrix [startX, startY]; relative coordinate; the value of X and Y is normalized to a number between 0 and 8192.	[7253,2275]
endpoint	int[2]	R	The end point of the target matrix [startX, startY]; relative coordinate; the value of X and Y is normalized to a number between 0 and 8192.	[7893,3034]
Request Example				
http://192.168.1.108/cgi-bin/ptzBase.cgi?action=moveDirectly&channel=1&startPoint[0]=7253&startPoint[1]=2275&endPoint[0]=7893&endPoint[1]=3034				

Response Params (OK in body)
Response Example
OK

- Relative PTZ Movement

Relative PTZ movement

Request URL	http://<server>/cgi-bin/ptz.cgi?action=moveRelatively			
Method	GET			

Request Params (key=value format in URL)

Name	Type	R/O	Description	Example
channel	int	R	The PTZ channel index; starting from 1	1
arg1	double	O	Relative horizontal motion; normalized to [-1, 1]	0.1
arg2	double	O	Relative vertical motion; normalized to [-1, 1]	0.1
arg3	double	O	Relative zoom; normalized to [-1, 1]	0.5

Request Example

```
http://192.168.1.108/cgi-bin/ptz.cgi?action=moveRelatively&channel=1&arg1=0.1&arg2=0.1&arg3=0.5
```

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

```
OK
```

- Accurate PTZ Positioning

Accurate PTZ positioning; the actual parameter range depends on the device. For capability details, execute the command mentioned in "8.1.3 Get PTZ Capability of Current Protocol".

Request URL	http://<server>/cgi-bin/ptz.cgi?action=moveAbsolutely			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
channel	int	R	The PTZ channel index; starting from 1	1
arg1	double	O	Relative horizontal position; normalized to [-1, 1]; arg1 < 0: Angle = 180.0 × arg1 + 360.0, and the actual range is [180.0, 360.0]; arg1 ≥ 0: Angle = 180.0 × arg1, and the actual range is [0, 180.0]	-0.8
arg2	double	O	Absolute vertical position; normalized to [-1, 1]; Angle = -180.0 × arg2, the actual range is [-180.0, 180.0]	0.3
arg3	double	O	Absolute zoom; normalized to [-1, 1]	0.5

Request Example

```
http://192.168.1.108/cgi-bin/ptz.cgi?action=moveAbsolutely&channel=1&arg1=-0.8&arg2=0.3&arg3=0.5
```

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

```
OK
```

8.1.6 Preset

- Getting Preset Information

Get the preset information

Request URL	http://<server>/cgi-bin/ptz.cgi?action=getPresets			
Method	GET			
Request Params (key=value format in URL)				

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/ptz.cgi?action=getPresets&channel=1
```

Response Params (key=value format in body)				
Name	Type	R/O	Description	Example
presets	object[]	R	List of preset information	
+Index	int	R	The preset number; starting from 1	1
+Name	char[256]	O	The preset name	"preset1"
+Type	int	O	Preset type 0: Normal preset 1: Preset configured with smart rules 2: Special preset	0
+PresetFunction	char[16]	O	Function List of the Special Presets "VideoBlack": Day/Night Mode (B/W) "VideoColor": Day/Night Mode (Color) "VideoBrightness": Day/Night Mode (Auto)	"VideoBlack"
+Position	int[3]	O	The coordinate of the preset and zoom; three integers The first parameter is the horizontal coordinate; range: [0,3599], referring to 0° to 359.9° (the number of degree is expanded by 10 times). The second parameter is the vertical coordinate; range: [-1800,1800], referring to -180° to 180° (the number of degree is expanded by 10 times). The third parameter is the expansion parameter; range: [0,128], referring to the zoom range.	[900, -900, 5]

Response Example

```
presets[0].Index=1
presets[0].Name=Preset 1
presets[0].Type=0
presets[0].PresetFunction="VideoBlack"
presets[0].Position=[900, -900, 5]
...
```

- Moving to the Preset

Move to the preset.

Request URL	http://<server>/cgi-bin/ptz.cgi?action=start&code=GotoPreset		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description

channel	int	R	The PTZ channel index; starting from 1	1
arg1	int	O	Ignore	0
arg2	int	R	The preset number; starting from 1	1
arg3	int	O	Ignore	0

Request Example

```
http://192.168.1.108/cgi-bin/ptz.cgi?action=start&code=GotoPreset&channel=1&arg1=0&arg2=1&arg3=0
```

Response Params (OK in body)

Response Example

OK

- Configuring Preset

Request URL http://<server>/cgi-bin/ptz.cgi?action=start&code=SetPreset

Method GET

Request Params (key=value format in URL)

Name	Type	R/O	Description	Example
channel	int	R	The PTZ channel index; starting from 1	1
arg1	int	O	Ignore	0
arg2	int	R	The preset number; starting from 1	1
arg3	int	O	Ignore	0

Request Example

```
http://192.168.1.108/cgi-bin/ptz.cgi?action=start&code=SetPreset&channel=1&arg1=0&arg2=2&arg3=0
```

Response Params (OK in body)

Response Example

OK

- Configuring Preset Name

Configure a name for the preset.

Request URL http://<server>/cgi-bin/ptz.cgi?action=SetPreset

Method GET

Request Params (key=value format in URL)

Name	Type	R/O	Description	Example
channel	int	R	The PTZ channel index; starting from 1	1
arg1	int	R	The preset number; starting from 1	2
arg2	char[256]	R	The preset name	"preset2"

Request Example

```
http://192.168.1.108/cgi-bin/ptz.cgi?action=SetPreset&channel=1&arg1=2&arg2=preset2
```

Response Params (OK in body)

Response Example

OK

- Deleting Preset

Delete a preset.

Request URL	http://<server>/cgi-bin/ptz.cgi?action=start&code=ClearPreset			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
channel	int	R	The PTZ channel index; starting from 1	1
arg1	int	O	Ignore	0
arg2	int	R	The preset number; starting from 1	2
arg3	int	O	Ignore	0
Request Example				
http://192.168.1.108/cgi-bin/ptz.cgi?action=start&code=ClearPreset&channel=1&arg1=0&arg2=2&arg3=0				

Response Params (OK in body)
Response Example
OK

8.1.7 Tour

- Start the tour

Request URL	http://<server>/cgi-bin/ptz.cgi?action=start&code=StartTour			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
channel	int	R	PTZ channel index which starts from 1. Range is [1 — n].	1
arg1	int	R	the number of tour route	1
arg2	int	R	Ignore	0
arg3	int	R	Ignore	0
Request Example				
http://192.168.1.108/cgi-bin/ptz.cgi?action=start&channel=1&code=StartTour&arg1=1&arg2=0&arg3=0				

Response Params (OK in body)
Response Example
OK

- Stop the tour

Request URL	http://<server>/cgi-bin/ptz.cgi?action=start&code=StopTour			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
channel	int	R	PTZ channel index which starts from 1. Range is [1 — n].	1
arg1	int	R	the number of tour route	1
arg2	int	R	Ignore	0
arg3	int	R	Ignore	0
Request Example				
http://192.168.1.108/cgi-bin/ptz.cgi?action=start&channel=1&code=StopTour&arg1=1&arg2=0&arg3=0				

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

OK

- Add tour group

Request URL	http://<server>/cgi-bin/ptz.cgi?action=setTour
--------------------	--

Method	GET
---------------	-----

Request Params (key=value format in URL)

Name	Type	R/O	Description	Example
channel	int	R	PTZ channel index which starts from 1. Range is [1 — n].	1
arg1	int	R	the number of tour route	1
arg2	int	R	tour name	1

Request Example

http://192.168.1.108/cgi-bin/ptz.cgi?action=setTour&channel=1&arg1=1&arg2=1

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

OK

- Delete tour group

Request URL	http://<server>/cgi-bin/ptz.cgi?action=start&code=ClearTour
--------------------	---

Method	GET
---------------	-----

Request Params (key=value format in URL)

Name	Type	R/O	Description	Example
channel	int	R	PTZ channel index which starts from 1. Range is [1 — n].	1
arg1	int	R	the number of tour group	1
arg2	int	R	Ignore	0
arg3	int	R	Ignore	0

Request Example

http://192.168.1.108/cgi-bin/ptz.cgi?action=start&channel=1&code=ClearTour&arg1=1&arg2=0&arg3=0

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

OK

- Add tour preset

Request URL	http://<server>/cgi-bin/ptz.cgi?action=start&code=AddTour
--------------------	---

Method	GET
---------------	-----

Request Params (key=value format in URL)

Name	Type	R/O	Description	Example
channel	int	R	PTZ channel index which starts from 1. Range is [1 — n].	1
arg1	int	R	the number of tour route	1
arg2	int	R	the number of preset	2
arg3	int	R	Ignore	0

Request Example

http://192.168.1.108/cgi-bin/ptz.cgi?action=start&channel=1&code=AddTour&arg1=1&arg2=2&arg3=0

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

OK

- Delete tour preset

Request URL	http://<server>/cgi-bin/ptz.cgi?action=start&code=DelTour
--------------------	---

Method	GET
---------------	-----

Request Params (key=value format in URL)

Name	Type	R/O	Description	Example
channel	int	R	PTZ channel index which starts from 1. Range is [1 — n].	1
arg1	int	R	the number of tour route	1
arg2	int	R	the number of preset	2
arg3	int	R	Ignore	0

Request Example

http://192.168.1.108/cgi-bin/ptz.cgi?action=start&channel=1&code=DelTour&arg1=1&arg2=2&arg3=0

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

OK

8.1.8 Scan

- Set left boundary

Request URL	http://<server>/cgi-bin/ptz.cgi?action=start&code=SetLeftLimit
--------------------	--

Method	GET
---------------	-----

Request Params (key=value format in URL)

Name	Type	R/O	Description	Example
channel	int	R	PTZ channel index which starts from 1. Range is [1 — n].	1
arg1	int	R	the scan number	1
arg2	int	R	Ignore	0
arg3	int	R	Ignore	0

Request Example

http://192.168.1.108/cgi-bin/ptz.cgi?action=start&channel=1&code=SetLeftLimit&arg1=1&arg2=0&arg3=0
--

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

OK

- Set right boundary

Request URL	http://<server>/cgi-bin/ptz.cgi?action=start&code=SetRightLimit			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
channel	int	R	PTZ channel index which starts from 1. Range is [1 — n].	1
arg1	int	R	the scan number	1
arg2	int	R	Ignore	0
arg3	int	R	Ignore	0
Request Example				
http://192.168.1.108/cgi-bin/ptz.cgi?action=start&channel=1&code=SetRightLimit&arg1=1&arg2=0&arg3=0				

Response Params (OK in body)
Response Example
OK

- Start scan

Request URL	http://<server>/cgi-bin/ptz.cgi?action=start&code=AutoScanOn			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
channel	int	R	PTZ channel index which starts from 1. Range is [1 — n].	1
arg1	int	R	the scan number	1
arg2	int	R	Ignore	0
arg3	int	R	Ignore	0
Request Example				
http://192.168.1.108/cgi-bin/ptz.cgi?action=start&channel=1&code=AutoScanOn&arg1=1&arg2=0&arg3=0				

Response Params (OK in body)
Response Example
OK

- Stop scan

Request URL	http://<server>/cgi-bin/ptz.cgi?action=start&code=AutoScanOff			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
channel	int	R	PTZ channel index which starts from 1. Range is [1 — n].	1
arg1	int	R	the scan number	1
arg2	int	R	Ignore	0

arg3	int	R	Ignore	0
Request Example				
http://192.168.1.108/cgi-bin/ptz.cgi?action=start&channel=1&code=AutoScanOff&arg1=1&arg2=0&arg3=0				

Response Params (OK in body)

Response Example

OK

8.1.9 Pattern

- Start pattern record

Request URL	http://<server>/cgi-bin/ptz.cgi?action=start&code=SetPatternBegin			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
channel	int	R	PTZ channel index which starts from 1. Range is [1 — n].	1
arg1	int	R	pattern number	1
arg2	int	R	Ignore	0
arg3	int	R	Ignore	0
Request Example				
http://192.168.1.108/cgi-bin/ptz.cgi?action=start&channel=1&code=SetPatternBegin&arg1=1&arg2=0&arg3=0				

Response Params (OK in body)

Response Example

OK

- Stop pattern record

Request URL	http://<server>/cgi-bin/ptz.cgi?action=start&code=SetPatternEnd			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
channel	int	R	PTZ channel index which starts from 1. Range is [1 — n].	1
arg1	int	R	pattern number	1
arg2	int	R	Ignore	0
arg3	int	R	Ignore	0
Request Example				
http://192.168.1.108/cgi-bin/ptz.cgi?action=start&channel=1&code=SetPatternEnd&arg1=1&arg2=0&arg3=0				

Response Params (OK in body)

Response Example

OK

- Start pattern

Request URL	http://<server>/cgi-bin/ptz.cgi?action=start&code=StartPattern			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
channel	int	R	PTZ channel index which starts from 1. Range is [1 — n].	1
arg1	int	R	pattern number	1
arg2	int	R	Ignore	0
arg3	int	R	Ignore	0
Request Example				
http://192.168.1.108/cgi-bin/ptz.cgi?action=start&channel=1&code=StartPattern&arg1=1&arg2=0&arg3=0				

Response Params (OK in body)
Response Example
OK

- Stop pattern

Request URL	http://<server>/cgi-bin/ptz.cgi?action=start&code=StopPattern			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
channel	int	R	PTZ channel index which starts from 1. Range is [1 — n].	1
arg1	int	R	pattern number	1
arg2	int	R	Ignore	0
arg3	int	R	Ignore	0
Request Example				
http://192.168.1.108/cgi-bin/ptz.cgi?action=start&channel=1&code=StopPattern&arg1=1&arg2=0&arg3=0				

Response Params (OK in body)
Response Example
OK

8.1.10 Pan

- Start pan

Request URL	http://<server>/cgi-bin/ptz.cgi?action=start&code=AutoPanOn			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
channel	int	R	PTZ channel index which starts from 1. Range is [1 — n].	1

arg1	int	R	Ignore	0
arg2	int	R	Ignore	0
arg3	int	R	Ignore	0

Request Example

```
http://192.168.1.108/cgi-bin/ptz.cgi?action=start&channel=1&code=AutoPanOn&arg1=0&arg2=0&arg3=0
```

Response Params (OK in body)

Response Example

```
OK
```

- Stop pan

Request URL	http://<server>/cgi-bin/ptz.cgi?action=start&code=AutoPanOff			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
channel	int	R	PTZ channel index which starts from 1. Range is [1 — n].	1
arg1	int	R	Ignore	0
arg2	int	R	Ignore	0
arg3	int	R	Ignore	0

Request Example

```
http://192.168.1.108/cgi-bin/ptz.cgi?action=start&channel=1&code=AutoPanOff&arg1=0&arg2=0&arg3=0
```

Response Params (OK in body)

Response Example

```
OK
```

8.1.11 [Config] PTZ Auto Movement

Config Data Params				
Name	Type	R/O	Description	Example
PtzAutoMovement	object[][]	O	A two-dimensional array, the first dimension represents the channel of the ptz, and the second dimension represents the task configuration of the channel.	
+Enable	bool	O	Enable/Disable PtzAutoMovement	true
+TimeSection	TimeSchedule	O	The time range during which the PTZ timed action takes effect, for example: TimeSchedule[week][section]=1 10:00:00-11:00:00	[["0 00:00-23:59:59","0 00:00-23:59:59","0 00:00-23:59:59","0 00:00-23:59:59","0 00:00-23:59:59"]]

			week: represents the day of the week, with values ranging from 0 to 6. 0 corresponds to Sunday, 1 corresponds to Monday, and so on. section: represents the time period, with a maximum of 6	00:00:00-23:59:59"]]
+Function	enumchar[32]	O	enumchar[32]{ Scan Preset Pattern Tour None }	"Scan"
+ScanId	int	O	Scan Id, start from 1	1
+PresetId	int	O	Preset Id, start from 1	1
+PatternId	int	O	Pattern Id, start from 1	1
+TourId	int	O	Tour Id, start from 1	1
+AutoHoming	object	O	auto homing	
++Enable	bool	O	enable/disable	true
++Time	uint	O	Recover time, unit is second.	300
+SnapshotEnable	bool	O	Enable/Disable Snap, when "Fuction" is "Preset".	false
+SnapshotDelayTime	int	O	Delay time of snap, when "Fuction" is "Preset".	30

Please refer to "4.2.1 Get and Set Configure" for configuration getting and setting. Specific examples are as follows :

Get Config Request Example

```
http://10.0.0.8/cgi-bin/configManager.cgi?action=getConfig&name=PtzAutoMovement
```

Get Config Response Example

```
table.PtzAutoMovement[0][0].AutoHoming.Time=30
table.PtzAutoMovement[0][0].Enable=false
table.PtzAutoMovement[0][0].Function=None
table.PtzAutoMovement[0][0].PatternId=0
table.PtzAutoMovement[0][0].PresetId=0
table.PtzAutoMovement[0][0].ScanId=0
table.PtzAutoMovement[0][0].SnapshotDelayTime=30
table.PtzAutoMovement[0][0].SnapshotEnable=false
table.PtzAutoMovement[0][0].TimeSection[0][0]=0 00:00:00-23:59:59
table.PtzAutoMovement[0][0].TimeSection[0][1]=0 00:00:00-23:59:59
table.PtzAutoMovement[0][0].TimeSection[0][2]=0 00:00:00-23:59:59
table.PtzAutoMovement[0][0].TimeSection[0][3]=0 00:00:00-23:59:59
table.PtzAutoMovement[0][0].TimeSection[0][4]=0 00:00:00-23:59:59
table.PtzAutoMovement[0][0].TimeSection[0][5]=0 00:00:00-23:59:59
table.PtzAutoMovement[0][0].TimeSection[1][0]=0 00:00:00-23:59:59
table.PtzAutoMovement[0][0].TimeSection[1][1]=0 00:00:00-23:59:59
```

```
table.PtzAutoMovement[0][0].TimeSection[1][2]=0 00:00:00-23:59:59  
table.PtzAutoMovement[0][0].TimeSection[1][3]=0 00:00:00-23:59:59  
table.PtzAutoMovement[0][0].TimeSection[1][4]=0 00:00:00-23:59:59  
table.PtzAutoMovement[0][0].TimeSection[1][5]=0 00:00:00-23:59:59  
table.PtzAutoMovement[0][0].TimeSection[2][0]=0 00:00:00-23:59:59  
table.PtzAutoMovement[0][0].TimeSection[2][1]=0 00:00:00-23:59:59  
table.PtzAutoMovement[0][0].TimeSection[2][2]=0 00:00:00-23:59:59  
table.PtzAutoMovement[0][0].TimeSection[2][3]=0 00:00:00-23:59:59  
table.PtzAutoMovement[0][0].TimeSection[2][4]=0 00:00:00-23:59:59  
table.PtzAutoMovement[0][0].TimeSection[2][5]=0 00:00:00-23:59:59  
table.PtzAutoMovement[0][0].TimeSection[3][0]=0 00:00:00-23:59:59  
table.PtzAutoMovement[0][0].TimeSection[3][1]=0 00:00:00-23:59:59  
table.PtzAutoMovement[0][0].TimeSection[3][2]=0 00:00:00-23:59:59  
table.PtzAutoMovement[0][0].TimeSection[3][3]=0 00:00:00-23:59:59  
table.PtzAutoMovement[0][0].TimeSection[3][4]=0 00:00:00-23:59:59  
table.PtzAutoMovement[0][0].TimeSection[3][5]=0 00:00:00-23:59:59  
table.PtzAutoMovement[0][0].TimeSection[4][0]=0 00:00:00-23:59:59  
table.PtzAutoMovement[0][0].TimeSection[4][1]=0 00:00:00-23:59:59  
table.PtzAutoMovement[0][0].TimeSection[4][2]=0 00:00:00-23:59:59  
table.PtzAutoMovement[0][0].TimeSection[4][3]=0 00:00:00-23:59:59  
table.PtzAutoMovement[0][0].TimeSection[4][4]=0 00:00:00-23:59:59  
table.PtzAutoMovement[0][0].TimeSection[4][5]=0 00:00:00-23:59:59  
table.PtzAutoMovement[0][0].TimeSection[5][0]=0 00:00:00-23:59:59  
table.PtzAutoMovement[0][0].TimeSection[5][1]=0 00:00:00-23:59:59  
table.PtzAutoMovement[0][0].TimeSection[5][2]=0 00:00:00-23:59:59  
table.PtzAutoMovement[0][0].TimeSection[5][3]=0 00:00:00-23:59:59  
table.PtzAutoMovement[0][0].TimeSection[5][4]=0 00:00:00-23:59:59  
table.PtzAutoMovement[0][0].TimeSection[5][5]=0 00:00:00-23:59:59  
table.PtzAutoMovement[0][0].TimeSection[6][0]=0 00:00:00-23:59:59  
table.PtzAutoMovement[0][0].TimeSection[6][1]=0 00:00:00-23:59:59  
table.PtzAutoMovement[0][0].TimeSection[6][2]=0 00:00:00-23:59:59  
table.PtzAutoMovement[0][0].TimeSection[6][3]=0 00:00:00-23:59:59  
table.PtzAutoMovement[0][0].TimeSection[6][4]=0 00:00:00-23:59:59  
table.PtzAutoMovement[0][0].TimeSection[6][5]=0 00:00:00-23:59:59  
table.PtzAutoMovement[0][0].TourId=0  
  
...
```

Set Config Request Example

[http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&PtzAutoMovement\[0\]\[0\].Function=PreSet&PtzAutoMovement\[0\]\[0\].PresetId=1](http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&PtzAutoMovement[0][0].Function=PreSet&PtzAutoMovement[0][0].PresetId=1)

Set Config Response Example

OK

8.1.12 PTZ Restart

Request URL	<a href="http://<server>/cgi-bin/ptz.cgi?action=start&code=Restart">http://<server>/cgi-bin/ptz.cgi?action=start&code=Restart
--------------------	---

Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
channel	int	R	PTZ channel index which starts from 1. Range is [1 — n].	1
arg1	int	R	Ignore	0
arg2	int	R	Ignore	0
arg3	int	R	Ignore	0
Request Example				
http://192.168.1.108/cgi-bin/ptz.cgi?action=start&channel=1&code=Restart&arg1=0&arg2=0&arg3=0				

Response Params (OK in body)
Response Example
OK

8.1.13 PTZ Reset

Request URL	http://<server>/cgi-bin/ptz.cgi?action=start&code=Reset			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
channel	int	R	PTZ channel index which starts from 1. Range is [1 — n].	1
arg1	int	R	Ignore	0
arg2	int	R	Ignore	0
arg3	int	R	Ignore	0
Request Example				
http://192.168.1.108/cgi-bin/ptz.cgi?action=start&channel=1&code=Reset&arg1=0&arg2=0&arg3=0				
Response Params (OK in body)				
Response Example				
OK				

8.1.14 OSD Menu

- Enter the menu

Request URL	http://<server>/cgi-bin/ptz.cgi?action=start&code=Menu			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
channel	int	R	PTZ channel index which starts from 1. Range is [1 — n].	1
arg1	int	R	Ignore	0
arg2	int	R	Ignore	0

arg3	int	R	Ignore	0
------	-----	---	--------	---

Request Example

http://192.168.1.108/cgi-bin/ptz.cgi?action=start&channel=1&code=Menu&arg1=0&arg2=0&arg3=0

Response Params (OK in body)

Response Example

OK

- Exit the menu

Request URL	http://<server>/cgi-bin/ptz.cgi?action=start&code=Exit			
Method	GET			

Request Params (key=value format in URL)

Name	Type	R/O	Description	Example
channel	int	R	PTZ channel index which starts from 1. Range is [1 — n].	1
arg1	int	R	Ignore	0
arg2	int	R	Ignore	0
arg3	int	R	Ignore	0

Request Example

http://192.168.1.108/cgi-bin/ptz.cgi?action=start&channel=1&code=Exit&arg1=0&arg2=0&arg3=0

Response Params (OK in body)				
Response Example				
OK				

- Confirm

Request URL	http://<server>/cgi-bin/ptz.cgi?action=start&code=Enter			
Method	GET			

Request Params (key=value format in URL)

Name	Type	R/O	Description	Example
channel	int	R	PTZ channel index which starts from 1. Range is [1 — n].	1
arg1	int	R	Ignore	0
arg2	int	R	Ignore	0
arg3	int	R	Ignore	0

Request Example

http://192.168.1.108/cgi-bin/ptz.cgi?action=start&channel=1&code=Enter&arg1=0&arg2=0&arg3=0

Response Params (OK in body)				
Response Example				
OK				

- Start the basic operation of menu

Request URL	http://<server>/cgi-bin/ptz.cgi?action=start			
--------------------	--	--	--	--

Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
code	char[]	R	range: { MenuUp MenuDown MenuLeft MenuRight }	MenuUp
channel	int	R	PTZ channel index which starts from 1. Range is [1 — n].	1
arg1	int	R	Ignore	0
arg2	int	R	Ignore	0
arg3	int	R	Ignore	0
Request Example				
http://192.168.1.108/cgi-bin/ptz.cgi?action=start&channel=1&code=MenuUp&arg1=0&arg2=0&arg3=0				

Response Params (OK in body)
Response Example
OK

8.1.15 [Config] Electronic PTZ

Config Data Params				
Name	Type	R/O	Description	Example
EptzLink	object[]	O	A one-dimensional array with element subscripts representing channel numbers.	
+Enable	bool	O	Whether to enable the electronic PTZ function in this channel. Channel indicates the channel number	true
+DisplayMode	char[32]	O	Display mode, used to select the number of channels to track; Options are: "original" - normal mode; "Oneplusone" - 1 + 1 mode; "Oneplusthree" - 1 + 3 mode; "Oneplusfive" - 1 + 5 mode	"Original"
+TrackEnable	bool	O	Whether to start linkage tracking; True - on, False close	true
+TrackTime	char[32]	O	Tracking duration options, including full-time tracking and manual configuration of tracking duration. "Fulltimetrack": full-time tracking, indicating that the tracking duration lasts	"Manual"

			until the target disappears; "Manual": customize the tracking duration. If this mode is selected, it will be tracked according to the user configured tracking duration.	
+ManualTrackTime	uint16[2]	O	If tracktime is in "manual" mode, it will be tracked according to the tracking duration range configured by the user in this field ([minimum tracking time, maximum tracking time]), which can be configured for 5-300 seconds. The maximum duration of tracking is the disappearance time of the target. For example, if the tracking duration is set to 50 seconds and the target disappears in 30 seconds, the tracking will stop). 'l' represents the index of the array, and 'i' can take 0 or 1	[5,300]
+TrackRect	uint[8][8][4]	O	The coordinate information of the tracking target frame represents one of the points. The first dimension represents the display mode, the second dimension represents the number of frames, and the third dimension is the bounding box of the rectangular frame, which is normalized to the 8192 coordinate system.	

Please refer to "4.2.1 Get and Set Configure" for configuration getting and setting. Specific examples are as follows :

Get Config Request Example

```
http://10.0.0.8/cgi-bin/configManager.cgi?action=getConfig&name=EptzLink
```

Get Config Response Example

```
table.EptzLink[0].Enable=true
table.EptzLink[0].DisplayMode=Original
```

Set Config Request Example

```
http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&EptzLink[0].Enable=true
```

Set Config Response Example

```
OK
```

8.1.16 Get View Range Status

Request URL	http://<server>/cgi-bin/api/ptz/getViewRangeStatus
Method	POST
Request Params (JSON format in body)	

Name	Type	R/O	Description	Example
Channel	int32	R	Video channel number, starting from 0	0

Request Example

```
{
    "Channel": 0
}
```

Response Params (JSON format in body)				
Name	Type	R/O	Description	Example
status	object	O	view range status	
+Distance	double	O	Visual distance in meters	20.0
+AngelH	double	O	Horizontal viewing angle Unit: degrees	30.5
+AzimuthH	double	O	Horizontal azimuth angle (The angle between the central axis of the included angle projected by the visual field on the horizontal plane and the reference axis on the horizontal plane) Expressed in radians, normalized to -1~1. Turn counterclockwise for positive direction	0.5
+AngelV	double	O	Vertical viewing angle Unit: degrees	30.5
+AzimuthV	double	O	Vertical azimuth angle (The angle between the central axis of the included angle projected by the visual field on the vertical plane and the angle of the reference axis on the vertical plane) Expressed in radians, normalized to -1~1.	0.5
+InclinationH	double	O	Horizontal inclination angle (The included angle between the equipment horizontal datum plane and the horizontal plane) Unit: degrees Normalized to -1~1.	0.5

Response Example

```
{
    "status": {
        "Distance": 20.0,
        "AngelH": 30.5,
        "AzimuthH": 0.5,
        "AngelV": 30.5,
```

```

        "AzimuthV" : 0.5,
        "InclinationH" : 0.5
    }
}

```

8.2 Wiper

8.2.1 Move Continuously

Make the wiper continuously move until you call the stopMove function to stop it.

Request URL	http://<server>/cgi-bin/rainBrush.cgi?action=moveContinuously		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
channel	int	O	Video channel number, starting from 1; the default value is 1
interval	int	R	Interval for the motion of the wiper; unit: second
Request Example			
http://192.168.1.108/cgi-bin/rainBrush.cgi?action=moveContinuously&channel=1&interval=5			

Response Params (OK in body)
Response Example
OK

8.2.2 Stop Move

Stop the motion of the wiper.

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

Response Params (OK in body)
Response Example
OK

8.2.3 Move Once

Control the wiper to move once.

Request URL	http://<server>/cgi-bin/rainBrush.cgi?action=moveOnce
--------------------	---

Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example

channel	int	O	video channel index which starts from 1, default 1 if not specified.	1
---------	-----	---	--	---

Request Example
http://192.168.1.108/cgi-bin/rainBrush.cgi?action=moveOnce&channel=1

Response Params (OK in body)
Response Example
OK

8.3 Illuminator

The following commands are applicable to non-intelligent illuminators.

8.3.1 [Config] Visible-light Illuminator

Config Data Params				
Name	Type	R/O	Description	Example
SignLight	object[]	R	A one-dimensional array with element subscripts representing channel numbers	1
+onCycle	uint32	R	Range[0-100]	30

Please refer to "4.2.1 Get and Set Configure" for configuration getting and setting. Specific examples are as follows :

Get Config Request Example
http://10.0.0.8/cgi-bin/configManager.cgi?action=getConfig&name=SignLight
Get Config Response Example
table.SignLight[0].onCycle=30
table.SignLight[1].onCycle=35
...

Set Config Request Example
http://10.0.0.8/cgi-bin/configManager.cgi?action=setConfig&SignLight[0].onCycle=30
Set Config Response Example
OK

8.3.2 [Config]Configuring Lighting

Illuminator configuration is used by IPC/SD production line. The illumination effect depends on the type of installed illuminator. Usage constraint: The illuminator type of the device is unique, only IR light, white light or other types of lights. If there are many different types of lights, only the Mode field of this configuration is used, and other fields are not used. In June, 2017, schedule control requirements were added to the white light. The second dimensional lighting configuration was expanded from one element to four elements. The 0 element of the second dimension indicates the configuration that takes effect

immediately. If the 0 element is modified, it needs to respond immediately. When the SupportByTime of LightingControl capability is true, it means that schedule setting is supported. Elements 1, 2 and 3 in the second dimension represent day, night, and normal respectively. Compatibility rule: If Lighting [0] is modified, it will take effect immediately (compatible with the usage without schedule function). If Lighting [0] is not modified, it means that the lights are controlled by schedule, and the configuration of [1]/[2]/[3] take effect. Subscripts 0 and 123 are not synchronized with each other. Doorbell products of intelligent building production line use Mode field to control the switch of illuminators, and only "SmartLight" and "Off" modes are used.

Config Data Params				
Name	Type	R/O	Description	Example
Lighting	object[][]	O	2D array. The first dimension corresponds to video input channels, and each video channel has several configurations.	
+Mode	char[32]	O	<p>Light mode.</p> <p>"Manual": Manually control the brightness and angle (the following settings are valid only when this field is available).</p> <p>"Auto": No need to set the brightness and angle.</p> <p>"Off": Turn off the light.</p> <p>"ZoomPrio": Zoom Priority. If LightTypeComplex of LightingControl takes effect, the corresponding LightingZoomPrio configuration in this mode takes effect. Otherwise, the LightingZoomPrio configuration will not take effect.</p> <p>The following modes are dedicated to composite lights.</p> <p>"Timing": schedule mode, which corresponds to the configuration of LightingSchedule.</p> <p>"SmartLight": Smart light mode (used by PTZ cameras). If this mode is selected, the newly added SmartLighting configuration is used. Doorbells of intelligent production line also use this mode, but do not use SmartLighting configuration.</p> <p>"ExclusiveManual": Multiple lights are supported, but only one light is used in manual mode, and only ManualLighting configuration is used.</p> <p>The LightingLink configuration in EventHandler takes effect in any illuminator modes.</p> <p>"ForceOn": Turn on the light forcibly (devvideoinput.forceinraredlight cannot be used because the light status needs to be kept).</p>	"ZoomPrio"

+Correction	int	O	Light compensation, only valid in zoom priority mode. Two ranges: 0–4; 0–100. There is no capability differences. Recommended range: 0–100.	2
+Sensitive	int	O	Light sensitivity, only valid in zoom priority mode. Range: 0–5. 3 by default.	3
+Times	int	O	Turn-on time, only valid in auto mode. Unit: Second (required by turnstiles with face recognition function)	30
+NearLight	object[]	O	Near light group. The number of arrays is controlled by the LightingControl capability set.	
++Light	int	O	Percentage of light brightness (1–100). 0: Turn off.	0
++Angle	int	O	Normalized angle of laser light. Range: 0–100	50
+MiddleLight	object[]	O	Medium light group. The number of arrays is controlled by the LightingControl capability set.	
++Enable	bool	O	Switch. Light field is used.	true
++Light	int	O	Percentage of light brightness (1–100). 0: Turn off.	0
++Angle	int	O	Normalized angle of laser light Range: 0–100	50
+FarLight	object[]	O	Far light. The number of arrays is controlled by the LightingControl capability set.	
++Light	int	O	Percentage of light brightness (1–100). 0: Turn off.	0
++Angle	int	O	Normalized angle of laser light Range: 0–100	50

See the following examples to read and modify configurations.

Get Config Request Example

```
http://10.0.0.8/cgi-bin/configManager.cgi?action=getConfig&name=Lighting
```

Get Config Response Example

```
table.Lighting[0][0].Correction=50
table.Lighting[0][0].FarLight[0].Angle=50
table.Lighting[0][0].FarLight[0].Light=50
table.Lighting[0][0].Mode=ZoomPrio
table.Lighting[0][0].NearLight[0].Angle=50
table.Lighting[0][0].NearLight[0].Light=50
table.Lighting[0][0].Sensitive=3
```

Set Config Request Example

```
http://10.0.0.8/cgi-bin/configManager.cgi?action=setConfig&Lighting[0][0].Mode=Manual
```

Set Config Response Example

```
OK
```

8.3.3 [Config]Configuring Lighting V2

Config Data Params				
Name	Type	R/O	Description	Example
Lighting_V2	object[][][]	O	<p>3D array. The first dimension corresponds to video input channels, and each video channel has several configurations. The second dimension array represents the corresponding configuration of day and night. From element 0, it represents day, night, normal, front light, general backlight, strong backlight, low illuminance, and custom. The third dimension indicates the type of light (IR light, white light, and laser light).</p>	
+LightType	enumchar[32]	O	<p>Light type. When the device controls the light, it is uniformly controlled by the light type.</p> <pre>enumchar[32]{ /* The capability is determined by the LightTypeComplex field in LightingControl */ "InfraredLight": IR light. "WhiteLight": White light. "LaserLight": Laser light. "AIMixLight": Intelligent mixing light, which can switch between IR and white lights according to intelligent ID. "PilotLight": Indicator. }</pre>	"InfraredLight"
+Mode	enumchar[32]	O	<p>Light mode.</p> <pre>enumchar[32]{ "Manual": Manually control the brightness and angle (the following settings are valid only when this field is available). "ZoomPrio": Zoom priority. "Auto": No need to set the brightness and angle. "Off": Turn off the light.</pre>	"ZoomPrio"

			<p>"DuskToDawn": Photosensitive mode, which directly judges whether to turn on the light according to the value transmitted by the photosensitive device. The configuration is saved only for application, and does not synchronize to the system settings.</p> <p>"ForceOn": Turn on the light forcibly.</p> <p>}</p>	
+Correction	int	O	<p>Light compensation, only valid in zoom priority mode.</p> <p>Range: 0–100.</p>	2
+PercentOfMaxBrightness	uint8	O	<p>The percentage of the upper limit brightness of the current white light relative to the maximum brightness of the general white light.</p> <p>Range: 0–100. The device calculates the actual limit value according to the percentage of the maximum brightness value of the general white light.</p>	100
+Sensitive	int	O	<p>Light sensitivity, only valid in zoom priority mode (the protocol description has no limit on mode, and it can be configured according to capability).</p> <p>Range: 0–5.</p> <p>The default value is 3.</p> <p>0–1: Low</p> <p>2–3: Medium</p> <p>4–5: High</p>	3
+LightSwitchDelay	int	O	<p>Delay time of illuminator switch.</p> <p>Range: 2–120 seconds.</p> <p>4 by default.</p>	4
+NearLight	object[]	O	<p>Near light group.</p> <p>The number of arrays is controlled by the LightingControl capability set.</p>	
++Light	int	O	<p>Percentage of light brightness (1–100).</p> <p>0: Turn off.</p>	0
++Angle	int	O	<p>Normalized angle of laser light</p> <p>Range: 0–100.</p>	50
++ColorTemperature	int	O	<p>Color temperature of light.</p> <p>0: Unknown. It is used for compatibility with previous fields.</p> <p>1: Cold light.</p> <p>2 : Warm light.</p>	0
+MiddleLight	object[]	O	<p>Medium light group.</p> <p>The number of arrays is controlled by the LightingControl capability set.</p>	

++Light	int	O	Percentage of light brightness (1–100). 0: Turn off.	0
++Angle	int	O	Normalized angle of laser light Range: 0–100.	50
++ColorTemperature	int	O	Color temperature of light. 0: Unknown. It is used for compatibility with previous fields. 1: Cold light. 2: Warm light.	0
+FarLight	object[]	O	Far light. The number of arrays is controlled by the LightingControl capability set.	
++Light	int	O	Percentage of light brightness (1–100). 0: Turn off.	0
++Angle	int	O	Normalized angle of laser light Range: 0–100.	50
++ColorTemperature	int	O	Color temperature of light. 0: Unknown. It is used for compatibility with previous fields. 1: Cold light. 2: Warm light.	0
+AlMixLightSwitchDelay	int	O	It takes effect under the intelligent illumination scheme, indicating the delay time when switching between IR and white lights to prevent the frequent switching from affecting the service life of light and user's experience. Range: 0–300 seconds. 30 seconds by default.	30
+State	char[32]	O	Light status: Off; flicker; on.	"On"
+Color	char[32]	O	Light color: Blue Red Green	Blue
+TimeSection	char[7][24][24]	O	Configure the solid-on time of the light. The first dimension indicates Sunday to Saturday, corresponding to the index 0–6. The second dimension represents period list of the day, with a maximum of 24. The third dimension represents the interval of each time period, and the format is "Enable Hour: Minute: Second-Hour: Minute: Second", and a space is used between enable and time information. 1 indicates that the period is valid, and 0 indicates invalid. The first Hour: Minute: Second is the start time, and	[["1 00:00:00-24:00:00",...],...[]]

			the second one is the end time. The end time can be 24:00:00.	
+TimeSection Enable	bool	O	Enable the period for solid on of the light. TimeSection will not take effect until the light is turned on.	true
+GlobalLight	object[]	O	Global illumination. It uses the internal light of the device to perform illumination. The number of lights is controlled by the LightingControl capability set.	
++Light	int	O	Percentage of light brightness (1–100). 0: Turn off.	50
+CenterLight	object[]	O	Central illumination. It uses the external light of the device to perform illumination. The number of lights is controlled by the LightingControl capability set.	
++Light	int	O	Percentage of light brightness (1–100). 0: Turn off.	50
+ManualLighting	object	O	Manually configure the light.	
++Mode	uint32	O	0: Flashes. 1: Solid on.	0
++LightOnDuration	uint32	O	Duration when the light is turned on the manually. Unit: Second. 0 means that the light will not turn off automatically and needs to be turned off manually.	180
++LightOffDuration	uint32	O	Duration when the light is turned off the manually. Unit: Second. 0 means that the light will immediately restore to the status before manual operation.	0

See the following examples to read and modify configurations.

Get Config Request Example

```
http://10.0.0.8/cgi-bin/configManager.cgi?action=getConfig&name=Lighting_V2
```

Get Config Response Example

```
table.Lighting_V2[0][0][0].Correction=50
table.Lighting_V2[0][0][0].LightType=InfraredLight
table.Lighting_V2[0][0][0].MiddleLight[0].Angle=50
table.Lighting_V2[0][0][0].MiddleLight[0].Light=50
table.Lighting_V2[0][0][0].Mode=Auto
table.Lighting_V2[0][0][0].PercentOfMaxBrightness=100
table.Lighting_V2[0][0][0].Sensitive=3
```

Set Config Request Example

```
http://10.0.0.8/cgi-bin/configManager.cgi?action=setConfig&Lighting_V2[0][0][0].Correction=50
```

Set Config Response Example

OK

8.3.4 [Config]Configuring Light Schemes

Config Data Params				
Name	Type	R/O	Description	Example
LightingScheme	object[][]	O	<p>Light scheme.</p> <p>2D array. The first dimension corresponds to video input channels, and each video channel has several configurations. The second dimension array represents day, night, and general configurations. From element 0, it represents day, night, normal, front light, general backlight, strong backlight, low illuminance, and custom, which correspond to the second dimension of Lighting_V2.</p>	
+LightingMode	enumchar[16]	O	<p>Light scheme.</p> <pre>Enumchar[16]{ "MixMode": Mixed light scheme. "WhiteMode": White light scheme. "NormalMode": Soft and dual lights are not supported. "InfraredMode": IR scheme. "AIMode": AI scheme. "Off": Night vision is disabled. }</pre>	"MixMode"
+SchemeSchedule	object	O	The lighting scheme is switched by period, and only when the corresponding camera attribute configuration (day, night, normal, etc.) takes effect, the lighting scheme schedule based on this configuration takes effect.	
++Enable	bool	O	It takes effect or not (after it takes effect according to the schedule, the original LightingMode will no longer take effect). True: takes effect. False: do not take effect.	true
++TimeSectionByWeek	char[7][6][36]	O	<p>Effective time period. For example, "05:40:00-18:20:00 WhiteMode" means that the white light scheme takes effect from 5: 40 to 18: 20. WhiteMode is the name of lighting scheme, which is a variable string (corresponding to the optional field of LightingMode). 2D array. The first dimension represents 7 days of a week (array subscript 0 corresponds to</p>	["05:40:00-18:20:00 WhiteMode", ...]]

		Sunday, followed by Monday to Saturday). The second dimension represents the periods of a day, with a maximum of 6 periods. All time periods in a day cannot overlap with each other, and the total must be 24 hours.	
--	--	---	--

See the following examples to read and modify configurations.

Get Config Request Example

```
http://10.0.0.8/cgi-bin/configManager.cgi?action=getConfig&name=LightingScheme
```

Get Config Response Example

```
table.LightningScheme[0][0].LightingMode=AIMode
```

Set Config Request Example

```
http://10.0.0.8/cgi-bin/configManager.cgi?action=setConfig&LightingScheme[0][0].LightingMode=InfraredMode
```

Set Config Response Example

```
OK
```

8.4 Flashlight

8.4.1 [Config] Flashlight Config

Configuration parameters of the flashing light (Not recommended).

Config Data Params

Name	Type	R/O	Description	Example
FlashLight	object	R	Configuration parameters of the flashing light	
+Enable	bool	R	Whether to enable the flashing light	true
+Brightness	int	R	Brightness; range [0, 100]	50
+TimeSection	char[7][24][32]	R	Two-dimensional array in strings. The first dimension of the array is the day of the week; range: [0-6] (Sunday–Saturday). The second dimension is the period index. One day is divided into multiple periods, and the range is [0–23]. Each period is set as a string in the format of mask hh: mm: ss-hh: mm: ss Mask: Value range: [0, 1] hh: Hour; range: [0–24] mm: Minute; range: [0–59] ss: Second; range: [0–59] Mask value:	[["1 00:00:00-23:59:59", "0 00:00:00-23:59:59", "0 00:00:00-23:59:59", "0 00:00:00-23:59:59", ...]]

		0: Disable the parameter in the period 1: Enable the parameter in the period For example: TimeSection[1][0]=1 12:00:00—18:00:00 Meaning the flashing light works from 12:00:00 through 18:00:00 on Monday.	
--	--	--	--

For getting and setting the configurations, see 4.2.1 Get and Set Configure”.

Get Config Request Example

```
http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=FlashLight
```

Get Config Response Example

```
table.FlashLight.Enable=true
table.FlashLight.Brightness=50
table.FlashLight.TimeSection[0][0]=1 00:00:00-23:59:59
table.FlashLight.TimeSection[0][1]=0 00:00:00-23:59:59
...
table.FlashLight.TimeSection[6][5]=0 00:00:00-23:59:59
...
```

Set Config Request Example

```
http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&FlashLight.Enable=true&FlashLight.Brightness=50&FlashLight.TimeSection[1][0]=1%2012:00:00-18:00:00
```

Set Config Response Example

```
OK
```

8.5 Coaxial Control IO

8.5.1 Control White Light or Speaker

Send commands for controlling the white light and speaker

Request URL	http://<server>/cgi-bin/coaxialControlIO.cgi?action=control		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
channel	int	O	It is the video channel number for the white light, and the audio output channel for the speaker. The default value is 0.
info	object[]	R	Operation details
+Type	int	R	Operation type 1: White light 2: Speaker
+IO	int	R	Switch 1: On 2: Off
+TriggerMode	int	R	Trigger Mode 1: Linked trigger

Request Example

```
http://192.168.1.108/cgi-bin/coaxialControlIO.cgi?action=control&channel=1&info[0].Type=1&info[0].IO=1&info[0].TriggerMode=2
```

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

```
OK
```

8.5.2 Get White Light and Speaker Status

Get the status of the white light and the speaker.

Request URL	http://<server>/cgi-bin/coaxialControlIO.cgi?action=getstatus		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
channel	int	O	The video channel number which starts from 1, and the default value is 1.
Request Example			
http://192.168.1.108/cgi-bin/coaxialControlIO.cgi?action=getstatus&channel=1			

Response Params (key=value format in body)			
Name	Type	R/O	Description
status	object	R	Return status information
+whitelight	char[4]	R	White light status, "on" or "off"
+speaker	char[4]	R	Speaker status, "on" or "off"
Response Example			
status.whitelight=on status.speaker=on			

8.6 Pir Alarm

8.6.1 [Config] Pir Parameter

- Get pir parameter

Request URL	http://<server>/cgi-bin/pirAlarm.cgi?action=getPirParam		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
channel	int	O	video channel index, starts from 1, default 1
Request Example			
http://192.168.1.108/cgi-bin/pirAlarm.cgi?action=getPirParam&channel=1			

Response Params (key=value format in body)			
Name	Type	R/O	Description
configEx	object[]	R	PIR

+Enable	bool	O	Enable/Disable motion detect feature in a channel.	true
+DetectWindow	object[<i>WinNum</i>]	O	<i>WinNum</i> Index of detect window, there are 4 detect windows at present. Each window is divided into 18 lines and 22 blocks/line.	
++Level	int	O	Range is [1—6]. Sensitivity of pir 1: lowest sensitivity. 6: highest sensitivity.	1
++Id	int	O	It is the Id of a detect window.	1
++Name	string	O	It is the name of a detect window.	xxx
++Sensitive	int	O	Range is [0—100]. It presents more sensitive if the value is larger.	2
++Threshold	int	O	Range is [0—100]. It presents the threshold value when trigger motion detect.	2
++Region	int[<i>LineNum</i>]	O	<i>LineNum</i> Index of region, region is divided into lines and each line has several blocks, a line is described by a 32 bit integer, a bit for a block. 0=Line 1 1=Line 2 Currently, region is divided into 18 lines and 22 blocks/line. A bit describes a block in the line. Bit = 1: motion in this block is monitored. Example: MotionDetect[0].Region[0] = 4194303 (0x3FFFFF):: motion in channel 0 line 0's 22 blocks is monitored. MotionDetect[0].Region[1] = 0: motion in line 1's 22 blocks is not monitored. MotionDetect[0].Region[17] = 3: in the last line of channel 0, motion in the left two blocks is monitored.	[4194303, 0, 3]
+TimeSection	char[wd][ts][]	O	wd (week day) range is [0—6] (Sunday - Saturday) ts (time section) range is [0 — 23], timesection table index. Format: mask hh:mm:ss-hh:mm:ss	["1 00:00:00-24:00:00", "1 00:00:00-24:00:00", "1 00:00:00-24:00:00", "

			Mask: [0—65535], hh: [0—24], mm: [0—59], ss: [0—59] Mask indicates record type by bits: Bit0: regular record Bit1: motion detection record Bit2: alarm record Bit3: card record	1 00:00:00-24:00:00"," 1 00:00:00-24:00:00"," 1 00:00:00-24:00:00"]]
+PirLink	object	O	pir linkage	
++RecordChannels	int[]	O	Range is {0, 1} 0 – do not record on video channel <i>ch</i> 1 – record on video channel <i>ch</i>	[0,1,0]
++RecordEnable	bool	O	Enable/Disable record function.	true
++RecordLatch	int	O	Range is [10—300]. Unit is seconds, indicates the time to record after input alarm is cleared.	10
++AlarmOutChannels	int[]	O	Range is {0, 1}, <i>ch</i> is alarm out channel index. 0 — do not output alarm at alarm out channel <i>ch</i> 1 — output alarm at alarm out channel <i>ch</i>	[0,0,1]
++AlarmOutEnable	bool	O	Enable/Disable alarm out function.	true
++AlarmOutLatch	Int	O	Range is [10—300]. Unit is seconds, indicates the time to output alarm after input alarm is cleared.	15
++SnapshotChannels	int[]	O	Range is {0, 1} 0 — do not snapshot on video channel <i>ch</i> 1 — snapshot on video channel <i>ch</i>	[0,0,1]
++SnapshotEnable	bool	O	Enable/Disable snapshot function.	true
++Dejitter	int	O	Range is [0—255]. Alarm signal dejitter seconds. Alarm signal change during this period is ignored.	10
++MailEnable	bool	O	Enable/Disable mail send for alarm.	true
++AlarmBellEnable	bool	O	Enable/Disable mail send for alarm.	true
++AlarmBellLatch	int	O	Range is [10, 300] Unit is seconds	10
++LogEnable	bool	O	Enable/Disable log for alarm.	true
Response Example				
configEx[0].Enable=true configEx[0].PirLink.LightingLink.Enable=true configEx[0].PirLink.LightingLink.LightLinkType=Filcker configEx[0].PirLink.LightingLink.FilckerIntervalTime=5 configEx[0].PirLink.LightingLink.LightDuration=10 configEx[0].PirLink.LightingLink.WhiteLightTimeSection=TimeSection				

```

configEx[0].PirLink.TimeSection[0][0]=1 00:00:00-24:00:00
configEx[0].PirLink.TimeSection[0][1]=0 02:00:00-24:00:00
configEx[0].PirLink.TimeSection[0][2]=0 03:00:00-24:00:00
configEx[0].PirLink.TimeSection[0][3]=0 04:00:00-24:00:00
configEx[0].PirLink.TimeSection[0][4]=0 05:00:00-24:00:00
configEx[0].PirLink.TimeSection[0][5]=0 06:00:00-24:00:00
configEx[0].RecordEnable=true
configEx[0].RecordChannels=[0, 1, 2]
configEx[0].RecordLatch=10
configEx[0].AlarmOutEnable=true
configEx[0].AlarmOutChannels=[1, 4]
configEx[0].AlarmOutLatch=10
configEx[0].SnapshotEnable=true
configEx[0].SnapshotChannels=[2, 4]
configEx[0].MailEnable=true
configEx[0].AlarmBellEnable=true
configEx[0].AlarmBellLatch=10
configEx[0].Dejitter=0
configEx[0].LogEnable=true
configEx[0].DetectWindow[0].Level=3
configEx[0].DetectWindow[0].Id=0
configEx[0].DetectWindow[0].Name=Region0
configEx[0].DetectWindow[0].Sensitive=58
configEx[0].DetectWindow[0].Threshold=4
configEx[0].DetectWindow[0].Region[0]=3932160
configEx[0].DetectWindow[0].Region[1]=3932160
...
...
configEx[0].DetectWindow[1]...

```

- set pir parameter

Request URL	http://<server>/cgi-bin/pirAlarm.cgi?action=setPirParam		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
channel	int	O	video channel index, starts from 1, default 1
configEx	object[]	O	PIR config
+Enable	bool	O	Enable/Disable motion detect feature in a channel.
+...			other params refer getPirParam
Request Example			
http://192.168.1.108/cgi-bin/pirAlarm.cgi?action=setPirParam&channel=1&configEx[1].Enable=true&configEx[1].PirLink.LightingLink.Enable=true&...			

Response Params (OK in body)

Response Example

OK

8.7 SCADA

8.7.1 Searching for SCADA Attributes

Request URL	http://<server>/cgi-bin/api/SCADA/getAttribute			
Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
condition	object	R		
+DeviceId	char[32]	R	Device ID	"01001010111"
+ID	char[]	No	An array of corresponding monitoring point ID. If there is no node, return all IDs under DeviceId.	["01001010111", "01001010112",...]
Request Example				
{ "condition": { "DeviceId": "01001010111", "ID": ["01001010111", "01001010112",...] } }	<i>HTTP API V3.5 - Intelbras</i>			

Response Params (JSON format in body)				
Name	Type	R/O	Description	Example
info	object[]	R		
+SignalName	char[128]	R	Point name	"Environment temperature"
+Unit	char[20]	R	Unit	"°C"
+Type	char[20]	R	Type of signal point YC: remote metering; analog input. YX: remote signaling; digital input. YT: remote regulation; analog output. YK: remote control; digital output.	"YC"
+ID	char[32]	R	Signal point ID	"01001010111"
+StartDelay	uint	O	Start delay of alarms, unit: second	120
+StopDelay	uint	O	End delay of alarms, unit: second	20
+Period	uint	O	Storing and reporting period, unit: second	180
+Threshold	float	O	Alarm threshold	40.0
+AlarmWaveValue	float	O	Alarm hysteresis. Only when the hysteresis range is exceeded, the alarm can be restored. Alarm recovery and	2

			alarm delay do not take effect at the same time.	
+AbsoluteVal	float	O	Absolute threshold. Stores and reports continuous data points (AI, AO).	0.2
+RelativeVal	float	O	Percentage threshold. Stores and reports continuous data points (AI, AO). It is used when the absolute threshold is disabled or is "0".	0
+Status	uint	R	Data status of signal point 0: normal 1: Level-1 alarm 2: Level-2 alarm 3: Level-3 alarm 4: Level-4 alarm 5: Operation event 6: Invalid data	0
+DisplayOptions	uint	O	Displays data: BIT0: Displays the switch. 0: No; 1: Yes. BIT1: Displays the switch when scrolling. 0: No; 1: Yes.	0x00000000
+Valid	bool	R	Invalid signal point true: Yes false: No	true
+Delay	uint32	O	Alarm delay time. When an alarm is triggered, the device response such as reporting and linkage is delayed. Unit: Second.	120
+Describe	char[]	O	Point description. For example, 0&Normal; 1&Alarm. Supports up to 120 characters.	"Environment temperature"
+IECCode	char[]	O	Power 104 protocol point number. Supports up to 16 characters.	"12345"
+HJCode	char[]	O	Environment 212 protocol point number. Supports up to 16 characters.	"a01001"

Response Example

```
{
  "info": [
    {
      "SignalName": "Environment temperature",
      "Unit": "°C",
      "Type": "YC",
      "ID": "01001010111",
      "StartDelay": 120,
      "StopDelay": 20,
      "Period": 180,
      "Threshold": 40.0,
      "AlarmWaveVal": 2,
    }
  ]
}
```

```

        "AbsoluteVal": 0.2,
        "RelativeVal": 0,
        "Status": 0,
        "DisplayOptions": 0x00000000,
        "Valid": true,
        "Delay": 120,
        "Describe": "Environment temperature",
        "IECCode": "12345",
        "HJCode": "a01001"
    },...{}
}

```

8.7.2 Configuring SCADA Attributes

You can use this method to disable or enable the point of SCADA devices.

Request URL	http://<server>/cgi-bin/api/SCADA/setAttribute		
Method	POST		
Request Params (JSON format in body)			
Name	Type	R/O	Description
DeviceId	char[32]	R	Detector
points	object[]	R	
+ID	char[32]	R	
+SignalName	char[128]	O	Point name
+StartDelay	uint	O	Start delay of alarms, unit: second
+StopDelay	uint	O	End delay of alarms, unit: second
+Period	uint	O	Storing and reporting period, unit: second
+Threshold	float	O	Alarm threshold
+AlarmWaveValue	float	O	Alarm hysteresis. Only when the hysteresis range is exceeded, the alarm can be restored. Alarm recovery and alarm delay do not take effect at the same .
+AbsoluteVal	float	O	Absolute threshold. Stores and reports continuous data points (AI, AO).
+RelativeVal	float	O	Percentage threshold. Stores and reports continuous data points (AI, AO). It is used when the absolute threshold is disabled or is "0".
+Status	enumint	O	Data status of signal point enumint{ 0: normal 1: Level-1 alarm 2: Level-2 alarm 3: Level-3 alarm 4: Level-4 alarm}

			5: Operation event 6: Invalid data }	
+DisplayOptions	uint	O	Displays data: BIT0: Displays the switch or not. 0: No; 1: Yes. BIT1: Displays the switch when scrolling. 0: No; 1: Yes.	0x00000000
+Valid	bool	R	Valid signal point. true: Valid. false: Invalid.	true

Request Example

```
{
  "DeviceId": "01001010111",
  "points": [
    {
      "ID": "01001010110",
      "SignalName": "Environment temperature",
      "StartDelay": 120,
      "StopDelay": 20,
      "Period": 180,
      "Threshold": 40.0,
      "AlarmWaveVal": 2,
      "AbsoluteVal": 0.2,
      "RelativeVal": 0,
      "Status": 0,
      "DisplayOptions": 0x00000000,
      "Valid": true
    }, ...
  ]
}
```

HTTP API V3.35 - Intelbras

Response Params (JSON format in body)

Name	Type	R/O	Description	Example
state	object	O		
+Success	char[]	O	The list with successful IDs	["01001010111"]
+Fail	char[]	O	The list with failed IDs	["01001010122"]

Response Example

```
{
  "state": {
    "Success": [ "01001010111"],
    "Fail": [ "01001010122"]
  }
}
```

8.7.3 Obtaining Real-time Data of Monitoring Points

Request URL	http://<server>/cgi-bin/api/SCADA/get
Method	POST
Request Params (JSON format in body)	

Name	Type	R/O	Description	Example
condition	object	R		
+DeviceId	char[16]	R	Detector ID	"01001010111"
+ID	char[]	O	An array of corresponding monitoring point ID If there is no node, return all IDs under the DeviceId.	["01001010111", "01001010112",...]
+IsHandle	bool	O	The returned data is processed (such as filtering invalid data). "false": No, "true": Yes. Default: false.	false

Request Example

```
{
  "condition": {
    "DeviceId": "01001010111",
    "ID": ["01001010111", "01001010112",...],
    "IsHandle": false
  }
}
```

Response Params (JSON format in body)				
Name	Type	R/O	Description	Example
info	object[]	R		
+Type	enumchar[32]	O	Point type enumchar[32]{ YC: Remote metering; analog input. YX: Remote signaling; digital input. YT: Remote regulation; analog output. YK: Remote control; digital output. }	"YC"
+ID	char[32]	R	Point	"01001010111"
+PointName	char[32]	O	Matches the point table name.	"aaa"
+MeasuredVal	float	R	The actual measured value. You can change the type. YX: Integer YC: Floating-point number	213.1
+SetupVal	float	O	The configured value. You can change the type. YK: Integer YT: Floating-point number	123.0
+Status	enumint	O	Data status enumint{ 0: Normal 1: Level-1 alarm 2: Level-2 alarm 3: Level-3 alarm 4: Level-4 alarm	0

			5: Operation Event 6: Invalid data }	
+RecordTime	char[20]	O	Collection time	"2015-1-3 10:10:45"

Response Example

```
{
  "info": [
    {
      "Type": "YC",
      "ID": "01001010111",
      "PointName": "aaa",
      "MeasuredVal": 213.1,
      "SetupVal": 123.0,
      "Status": 0,
      "RecordTime": "2015-1-3 10:10:45"
    }, ...
  ]
}
```

8.7.4 Configuring Monitoring Points

Request URL	http://<server>/cgi-bin/api/SCADA/set			
Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
DeviceId	char[32]	R	Detector	"01001010111"
points	object[]	R		
+Type	char[32]	O	Point type YT YK	"YT"
+ID	char[64]	R	Point number	"01001010111"
+SetupVal	float	R	The configured value YT: Floating-point number YK: Integer	12.3
Request Example				
{	"DeviceId": "01001010111", "points": [{ "Type": "YT", "ID": "01001010111", "SetupVal": 12.3 }]			
}				

Response Params (JSON format in body)				
Name	Type	R/O	Description	Example
state	object	O		
+Success	char[][]	O	The list with successfully controlled and adjusted IDs	["01001010111"]
+Fail	char[][]	O	The list with unsuccessfully controlled	["01001010122"]

		and adjusted IDs	
Response Example			

```
{
  "state": {
    "Success": ["01001010111"],
    "Fail": ["01001010122"]
  }
}
```

8.7.5 Starting Searching for Historical Data

Start searching for historical data based on device ID and time, and get search token.

Request URL	http://<server>/cgi-bin/api/SCADA/startFind		
Method	POST		
Request Params (JSON format in body)			
Name	Type	R/O	Description
condition	object	R	Search conditions
+StartTime	char[20]	R	Start time
+EndTime	char[20]	O	End time If you don't enter content in this field, it means up to now.
+DeviceId	char[32]	R	Device ID
+ID	char[32]	O	ID of corresponding monitoring point
Request Example			
{			
"condition": {			
"StartTime": "2015-1-2 10:10:45",			
"EndTime": "2015-1-3 10:10:45",			
"DeviceId": "01001010111",			
"ID": "01001010111"			
}			
}			

Response Params (JSON format in body)			
Name	Type	R/O	Description
token	int	R	Obtained search token
Response Example			
{			
"token": 46878,			
"totalCount": 3333			
}			

8.7.6 Obtaining Historical Data

Obtain historical data based on search token.

Request URL	http://<server>/cgi-bin/api/SCADA/doFind
--------------------	--

Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
token	int	R	Search token	46878
beginNumber	int	R	Start number of the search. The search starts from the "beginNumber" records, and returns predefined number of records. 0<=beginNumber<=totalCount-1	0
count	int	R	Number of traffic statistics for each search	24
Request Example				
{				
"token": 46878,				
"beginNumber": 0,				
"count": 24				
}				

Response Params (JSON format in body)				
Name	Type	R/O	Description	Example
found	int	R	Number of searched entries	12
info	object[]	R	Traffic statistics It is an array and each element represents a traffic record that satisfies the condition.	
+Type	char[32]	R	Point type enumchar[32]{ YC: remote metering; analog input YX: Remote signaling; digital input: YT: Remote regulation; analog output YK: Remote control; digital output }	"YC"
+ID	char[32]	R	Monitoring point ID	"01001010111"
+MeasuredVal	float	R	Value You can change this filed type, which is relevant to Type filed. When Type is YC, this value is floating point number; When Type is YX, this value is integer.	213.1
+SetupVal	float	R	Configured value You can change this filed type, which is relevant to Type filed. When Type is YC, this value is floating point number; When Type is YX, this value is integer.	123.0
+Status	int	R	Point status	0
+RecordTime	char[20]	R	Record time	"2015-1-3 10:10:45"
Response Example				

```
{
  "found": 12,
  "info": [
    {
      "Type": "YC",
      "ID": "01001010111",
      "MeasuredVal": 213.1,
      "SetupVal": 123.0,
      "Status": 0,
      "RecordTime": "2015-1-3 10:10:45"
    },...{}
  ]
}
```

8.7.7 Stopping Searching for Historical Data

Request URL	http://<server>/cgi-bin/api/SCADA/stopFind		
Method	POST		
Request Params (JSON format in body)			
Name	Type	R/O	Description
token	int	R	Search token
Request Example			
{	HTTP API V3.35 - Intelbras		
"token": 46878			
}			

Response Params (JSON format in body)			
Name	Type	R/O	Description
Response Example			
{}			

8.7.8 Obtaining IDs of External Devices Connected to the Host

Request URL	http://<server>/cgi-bin/api/SCADA/getDeviceList		
Method	POST		
Request Params (JSON format in body)			
Name	Type	R/O	Description
Request Example			
{}			

Response Params (JSON format in body)			
Name	Type	R/O	Description
devices	object[]	R	Device ID array
+DeviceId	char[32]	R	Device ID
+DevName	char[32]	O	Device name
+DevCode	char[8]	O	Device model
+Slot	int	O	Virtual slot number In general, 0 means analog channel and 1 means digital channel. If it is greater than 1, it means
			0

			RS-485, RS-232, or network channel. The actual channel number = virtual slot number – 2.	
+Level	int	O	The detector address configured by the environment surveillance server. A slot lot can be connected to multiple detectors, and different detectors are distinguished by address.	1

Response Example

```
{
  "devices": [
    {
      "DeviceId": "0101012345",
      "DevName": "UPS1",
      "DevCode": "1801",
      "Slot": 0,
      "Level": 1
    }, ...
  ]
}
```

8.8 Gyro

8.8.1 get gyroscope info

Request URL	http://<server>/cgi-bin/api/Gyro/getData			
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
detail	object	R		
+AccelX	double	R	Gravitational acceleration in X direction, unit m/s ²	15,3
+AccelY	double	R	Gravitational acceleration in Y direction, unit m/s ²	20.7
+AccelZ	double	R	Gravitational acceleration in Z direction, unit m/s ²	10.5
+Pitch	double	O	Pitch angle, in degrees range[-90, 90]	1.3
+Roll	double	O	roll angle, in degrees range[-180, 180]	23.6
+Yaw	double	O	yaw angle, in degrees	120.2

			range[-180, 180]	
+AngularSpeedX	double	R	X-direction angular velocity (rad/s)	2.5
+AngularSpeedY	double	R	Y-direction angular velocity (rad/s)	3.3
+AngularSpeedZ	double	R	Z-direction angular velocity (rad/s)	3.6

Response Example

```
{
  "detail": {
    "AccelX": 15.3,
    "AccelY": 20.7,
    "AccelZ": 10.5,
    "Pitch": 1.3,
    "Roll": 23.6,
    "Yaw": 120.2,
    "AngularSpeedX": 2.5,
    "AngularSpeedY": 3.3,
    "AngularSpeedZ": 3.6
  }
}
```

8.9 Other

8.9.1 [Config]Day/Night Settings of PTZ Module

Config Data Params				
Name	Type	R/O	Description	Example
VideoInDayNight	object[][]	O	Configuring Day/Night Settings of PTZ Module object[][] 2D array. The first dimension corresponds to the channel number and the second dimension corresponds to the lighting scene. For compatibility, the first three dimensions relatively represent day, night, and general scenes. The number of objects in the second array is not fixed, so how many lighting scenes are determined by PD capability VideoInSceneLink.	
+Name	char[16]	O	Scene name. For details on lighting scene name, see PD protocol VideoInSceneLink.	"Day"
+Type	char[16]	O	ICR switching mode: Electron, Mechanism, NightICR, and Auto.	"Mechanism"
+Mode	char[16]	O	Color/B&W mode:	"Brightness"

			<p>"Color": Alway color. "Brightness": Auto switches the mode according to brightness. "BlackWhite": Always blackwhite. "Photoresistor": Photoresistor. "Gain": Switches the mode according to the gain. "Timing": Auto switches the mode according to the time (the specific switching time is determined by the TimeSchedule field).</p>	
+Sensitivity	int	O	<p>Color/B&W Sensitivity. Range: 1–3. In the case of supporting sensitivity expansion, the sensitivity range can be expanded according to the capacity. See PD VideoIndayNight. SupportSensitivityrange for the range of values.</p>	1
+Delay	int	O	<p>Delay time of switching day/night mode. Range: 3–30 seconds.</p>	10
+PreValue	int	O	<p>Brightness preset value. The IR filter switches the brightness. Below a brightness, use a filter, and use another filter when it higher than one brightness. When IRCUTMode is Auto, the value range is 0–100.</p>	50
+IRPlateMode	char[16]	O	<p>IR license plate mode. "BlackWhite": : Black&white mode. "Color": Color mode.</p>	"BlackWhite"
+TimeSchedule	TimeSchedule	O	<p>Switch schedule. It is required in timing mode. The type is TimeSection[7][6], which is set in 7 days a week, and 6 periods can be set in one day, each of which is in the following format: "1 00:00:00-23:59:59". 1: Enable. 00:00:00: Start time (hours, minutes and seconds). 23:59:59: End time (hours, minutes and seconds).</p>	[["1 00:00:00-23:59:59", ...], ..., [...]]
+Snapshot	object	O	<p>Configure snapshot of double shutters. Note: When testing the production line clarity of the three-lens stereo analysis devices, you can switch</p>	

			color/B&W mode of the left and right shutters by the Mode field under Snapshot when double shutters are used.	
++Type	char[16]	O	Configure color/B&W mode for snapshot of double shutters. ICR switching mode: Electron, Mechanism, NightICR, and Auto.	"Mechanism"
++Mode	char[16]	O	Configure color/B&W mode for snapshot of double shutters. "Color": Alway color. "Brightness": Auto switches the mode according to brightness. "BlackWhite": Always blackwhite. "Photoresistor": Photoresistor. "Gain": Switches the mode according to the gain. "Timing": Auto switches the mode according to the time (the specific switching time is determined by the TimeSchedule field).	"Brightness"
++Sensitivity	int32	O	Configure color/B&W sensitivity for snapshot of double shutters. Range: 1–3. In the case of supporting sensitivity expansion, the sensitivity range can be expanded according to the capacity. See PD VideoInDayNight. SupportSensitivityrange for the range of values.	1
++Delay	int32	O	Configure the delay time of day/night mode for snapshot of double shutters. Range: 3–30 seconds.	10

See the following examples to read and modify configurations.

Get Config Request Example

```
http://10.0.0.8/cgi-bin/configManager.cgi?action=getConfig&name=VideoInDayNight
```

Get Config Response Example

```
table.VideoInDayNight[0][0].Delay=6
table.VideoInDayNight[0][0].Mode=Brightness
table.VideoInDayNight[0][0].Name=Day
table.VideoInDayNight[0][0].Sensitivity=2
table.VideoInDayNight[0][0].Type=Mechanism
```

Set Config Request Example

```
http://10.0.0.8/cgi-bin/configManager.cgi?action=setConfig&VideoInDayNight[0][0].Mode=BlackWhite
```

Set Config Response Example

OK

HTTP API V3.35 - Intelbras

9.1 Video Analyse Event

9.1.1 [Event] LeftDetection

Event Code	LeftDetection			
Event action	Start/Stop			
Event index	0			
Event Data				
Name	Type	R/O	Description	Example
+Object	object	R	The object that left.	
++BoundingBox	uint16[4]	R	The detected object bounding box, 4 interge, refer to x's value of left—top point, y's value of left—top point, x's value of right—bottom point, y's value of right—bottom point. Coordinate remap to 0 — 8192.	[2992,1136,4960,5192]

Event Response Example (multipart , key=value format in body , binary data in body) (response to 4.4.3 snap subscription)

--<boundary>
Content-Type: text/plain
Content-Length: <length>

Events[0].EventBaseInfo.Code=LeftDetection
Events[0].EventBaseInfo.Action=Start
Events[0].EventBaseInfo.Index=0
Events[0].Object.BoundingBox[0]= 4392
Events[0].Object.BoundingBox[1]=4136
Events[0].Object.BoundingBox[2]=6960
Events[0].Object.BoundingBox[3]=6512
.....

--<boundary>
Content-Type: image/jpeg
Content-Length: <image size>

<Jpeg image data>
--<boundary>

9.1.2 [Event] TakenAwayDetection

When detect some object was taken away, send this event.

Event Code	TakenAwayDetection			
Event action	Start/Stop			
Event index	0			
Event Data				
Name	Type	R/O	Description	Example
+Object	object	R	The object that was taken away.	
++BoundingBox	uint16[4]	R	The detected object bounding box, 4 interge, refer to x's value of left—top point, y's value of left—top point, x's value of right—bottom point, y's value of right—bottom point. Coordinate remap to 0 — 8192.	[2992,1136,4960,5192]

Event Response Example (multipart , key=value format in body , binary data in body) (response to 4.4.3 snap subscription)

--<boundary>

Content-Type: text/plain

Content-Length: <length>

Events[0].EventBaseInfo.Code=TakenAwayDetection

Events[0].EventBaseInfo.Action=Start

Events[0].EventBaseInfo.Index=0

Events[0].Object.BoundingBox[0]= 4392

Events[0].Object.BoundingBox[1]=4136

Events[0].Object.BoundingBox[2]=6960

Events[0].Object.BoundingBox[3]=6512

.....

--<boundary>

Content-Type: image/jpeg

Content-Length: <image size>

<Jpeg image data>

--<boundary>

9.1.3 [Event] WanderDetection

When detect some object was wandering, send this event

Event Code	WanderDetection
Event action	Start/Stop
Event index	0
Event Data	

Name	Type	R/O	Description	Example
+Objects	object[]	R	The objects that was wandering.	
++BoundingBox	uint16[4]	R	The detected object bounding box, 4 interge, refer to x's value of left—top point, y's value of left—top point, x's value of right—bottom point, y's value of right — bottom point. Coordinate remap to 0 — 8192.	[2992,1136,4960,5192]
+Tracks	int[][20][2]	O	The object wandering tracks, array of polyline, one polyline for one object, polyline is array of points, point is array of two int, x's value and y's value. Coordinate remap to 0 — 8192.	
+DetectRegion	int[20][2]	R	The detection region, the first array is point list, max item is 20, the second array is point, must be two int, means x and y value, coordinate remap to 0 — 8192.	

Event Response Example (multipart , key=value format in body , binary data in body) (response to 4.4.3 snap subscription)

--<boundary>

Content-Type: text/plain

Content-Length: <length>

Events[0].EventBaseInfo.Code=WanderDetection

Events[0].EventBaseInfo.Action=Start

Events[0].EventBaseInfo.Index=0

Events[0].Objects[0].BoundingBox[0]= 4392

Events[0].Objects[0].BoundingBox[1]=4136

Events[0].Objects[0].BoundingBox[2]=6960

Events[0].Objects[0].BoundingBox[3]=6512

Events[0].Tracks[0][0][0]=23

Events[0].Tracks[0][0][1]=23

Events[0].Tracks[0][1][0]=500

Events[0].Tracks[0][1][1]=401

Events[0].Tracks[0][2][0]=1003

Events[0].Tracks[0][2][1]=192

...

Events[0].DetectRegion[0][0]=192

Events[0].DetectRegion[0][1]=192

Events[0].DetectRegion[1][0]=562

Events[0].DetectRegion[1][1]=552

Events[0].DetectRegion[2][0]=600

Events[0].DetectRegion[2][1]=733

Events[0].DetectRegion[3][0]=200

Events[0].DetectRegion[3][1]=270

```

...
--<boundary>
Content-Type: image/jpeg
Content-Length: <image size>

<Jpeg image data>
--<boundary>

```

9.1.4 [Event] StayDetection

When detect some object was stay, send this event

Event Code	StayDetection		
Event action	Start/Stop		
Event index	0		
Event Data			
Name	Type	R/O	Description
+Object	object	R	The object that was stay.
++BoundingBox	uint16[4]	R	The detected object bounding box, 4 interge, refer to x's value of left—top point, y's value of left—top point, x's value of right—bottom point, y's value of right — bottom point. Coordinate remap to 0 — 8192.
+Objects	object[]	R	If detect several object, store in this array.
++BoundingBox	uint16[4]	R	The detected object bounding box, 4 interge, refer to x's value of left—top point, y's value of left—top point, x's value of right—bottom point, y's value of right — bottom point. Coordinate remap to 0 — 8192.
+DetectRegion	int[20][2]	R	The detection region, the first array is point list, max item is 20, the second array is point, must be two int, means x and y value, coordinate remap to 0 — 8192.
+AreaID	int	O	The area id, begin from 1, if omit, means single area.
+PresetID	int	O	The preset id, if omit, means preset is unknown.

Event Response Example (multipart , key=value format in body , binary data in body) (response to 4.4.3 snap subscription)

```

--<boundary>
Content-Type: text/plain
Content-Length: <length>

```

```

Events[0].EventBaseInfo.Code=StayDetection
Events[0].EventBaseInfo.Action=Start
Events[0].EventBaseInfo.Index=0
Events[0].Object.BoundingBox[0]= 4392
Events[0].Object.BoundingBox[1]=4136
Events[0].Object.BoundingBox[2]=6960
Events[0].Object.BoundingBox[3]=6512
Events[0].Objects[0].BoundingBox[0]= 4392
Events[0].Objects[0].BoundingBox[1]=4136
Events[0].Objects[0].BoundingBox[2]=6960
Events[0].Objects[0].BoundingBox[3]=6512
Events[0].DetectRegion[0][0]=192
Events[0].DetectRegion[0][1]=192
Events[0].DetectRegion[1][0]=562
Events[0].DetectRegion[1][1]=552
Events[0].DetectRegion[2][0]=600
Events[0].DetectRegion[2][1]=733
Events[0].DetectRegion[3][0]=200
Events[0].DetectRegion[3][1]=270
...
Events[0].AreaID=2
Events[0].PresetID=1
--<boundary>
Content-Type: image/jpeg
Content-Length: <image size>

<Jpeg image data>
--<boundary>

```

9.1.5 [Event] HumanTrait

When detect a human trait, send this event.

Event Code	HumanTrait			
Event action	Pulse			
Event index	The channel index relate to this event, start from 0.			
Event Data				
Name	Type	R/O	Description	Example
+HumanAttribute s	object	O	The human attributes.	

++BoundingBox	Array<int>	O	The detected human bounding box, 4 interge, refer to x's value of left-top point, y's value of left-top point, x's value of right-bottom point, y's value of right-bottom point. Coordinate remap to 0 — 8192.	
++Sex	string	O	Sex, can be "Man", "Woman", "Unknown".	"Woman"
++Age	int	O	Age.	
++Angle	int	O	Angle, 0: unknown, 1: front, 2: side, 3: back.	
++CoatColor	char[]	O	Coat color, can be: "White", "Orange", "Pink", "Black", "Red", "Yellow", "Gray", "Blue", "Green", "Purple", "Brown", "Sliver", "Darkviolet", "Maroon", "Dimgray", "Whitesmoke", "Darkorange", "Mistyrose", "Tomato", "Olive", "Gold", "Darkolivegreen", "Chartreuse", "Greenyellow", "Forestgreen", "Seagreen", "Chartreuse", "Deepskyblue", "Cyan", "Other".	"White"
++CoatType	int	O	Coat type, 0: unknown, 1: long sleeve, 2: short sleeve.	2
++TrousersColor	char[]	O	Trousers color, value can be that of CoatColor.	"White"
++TrousersType	int	O	Trousers type, 0: unknown, 1: long pants, 2: short pants, 3: skirt.	2
++HasHat	int	O	Has hat or not, 0: unknown, 1: not has hat, 2: has hat.	1
++HasBag	int	O	Has bag or not, 0: unknown, 1: not has bag, 2: has bag.	1
++HasUmbrella	int	O	Has umbrella or not, 0: unknown, 1: not has umbrella, 2: has umbrella.	1
++Bag	int	O	Bag type, 0: unknown, 1: handbag, 2: shoulder bag, 3: knapsack, 4: draw-bar box.	1
++UpperPattern	int	O	Upper clothes pattern, 0: unknown, 1: pure color, 2: stripe, 3: pattern, 4: gap, 5: grid.	2
++HairStyle	int	O	Hair style, 0: unknown, 1: long hair, 2: short hair, 3: ponytail, 4: updo, 5: hiddened.	3

++Cap	int	O	Cap style, 0: unknown, 1: normal cap, 2: helmet.	0
+FaceAttributes	object	O	If the human's face can be detected, find it's attributes.	
++BoundingBox	int[4]	O	The detected face bounding box, 4 interge, refer to x's value of left-top point, y's value of left-top point, x's value of right-bottom point, y's value of right-bottom point. Coordinate remap to 0 — 8192.	
++Sex	string	O	Sex, it can be "Man", "Woman", "Unknown".	"Woman"
++Age	int	O	Age.	23
++Feature	char[]	O	Face feature, can be some of the following : "WearGlasses", "SunGlasses", "NoGlasses", "Smile", "Anger", "Sadness", "Disgust", "Fear", "Surprise", "Neutral", "Laugh", "Happy", "Confused", "Scream".	["Smile"]
++Eye	int	O	Eye status, 0: not detected, 1: close eye, 2: open eye.	2
++Mouth	int	O	Mouth status, 0: not detected, 1: close mouth, 2: open mouth.	1
++Mask	int	O	Mask status, 0: not detected, 1: not wearing mask, 2: wearing mask.	2
++Beard	int	O	Beard status, 0: not detected, 1: no beard, 2: has beard.	1
++Glass	Int	O	Glasses status, 0: unknown, 1: not wearing, 2: normal Glasses, 3: sun glasses, 4: black frame glasses.	1

[Example]

```
--<boundary>
Content-Type: text/plain
Content-Length: <length>
```

```
Events[0].EventBaseInfo.Code=HumanTrait
Events[0].EventBaseInfo.Action=Pulse
Events[0].EventBaseInfo.Index=0
Events[0].HumanAttributes.BoundingBox[0]=1341
Events[0].HumanAttributes.BoundingBox[1]=2451
Events[0].HumanAttributes.BoundingBox[2]=4513
Events[0].HumanAttributes.BoundingBox[3]=4135
```

```

Events[0].HumanAttributes.Sex=Man
Events[0].HumanAttributes.Age=30
Events[0].HumanAttributes.CoatColor=White
Events[0].HumanAttributes.CoatType=1
Events[0].HumanAttributes.TrousersColor=Black
Events[0].HumanAttributes.TrousersType=1
Events[0].HumanAttributes.HasHat=1
Events[0].HumanAttributes.HasBag=2
Events[0].FaceAttributes.BoundingBox[0]=1341
Events[0].FaceAttributes.BoundingBox[1]=2451
Events[0].FaceAttributes.BoundingBox[2]=4513
Events[0].FaceAttributes.BoundingBox[3]=4135
Events[0].FaceAttributes.Sex=Man
Events[0].FaceAttributes.Age=30
Events[0].FaceAttributes.Feature[0]=Smile
Events[0].FaceAttributes.Eye=2
Events[0].FaceAttributes.Mouth=1
Events[0].FaceAttributes.Glass=1
--<boundary>
Content-Type: image/jpeg
Content-Length: <image size>

<Jpeg image data>
--<boundary>

```

9.1.6 [Event] CrossLineDetection

When detect some object cross the line, send this event.

Event Code	CrossLineDetection		
Event action	Start/Stop		
Event index	0		
Event Data			
Name	Type	R/O	Description
+Object	object	R	The object that cross the line.
++BoundingBox	uint16[4]	R	The detected object bounding box, 4 interge, refer to x's value of left—top point, y's value of left—top point, x's value of right—bottom point, y's value of right-bottom point. Coordinate remap to 0 — 8192.
+Objects	object[]	O	If detect several object, store in this array.
++BoundingBox	uint16[4]	R	The detected object bounding
			[2992,1136,4960,5192]

			box, 4 interge, refer to x's value of left—top point, y's value of left—top point, x's value of right—bottom point, y's value of right-bottom point. Coordinate remap to 0 — 8192.	
+DetectLine	int[20][2]	R	The detection line, the first array is point list, max item is 20, the second array is point, must be two int, means x and y value, coordinate remap to 0 — 8192.	
+Direction	string	O	The crossline direction, can be : "LeftToRight", "RightToLeft", "Any".	"LeftToRight"

Event Response Example (multipart , key=value format in body , binary data in body) (response to 4.4.3 snap subscription)

--<boundary>

Content-Type: text/plain

Content-Length: <length>

```

Events[0].EventBaseInfo.Code=CrossLineDetection
Events[0].EventBaseInfo.Action=Start
Events[0].EventBaseInfo.Index=0
Events[0].Object.BoundingBox[0]= 4392
Events[0].Object.BoundingBox[1]=4136
Events[0].Object.BoundingBox[2]=6960
Events[0].Object.BoundingBox[3]=6512
Events[0].Objects[0].BoundingBox[0]= 4392
Events[0].Objects[0].BoundingBox[1]=4136
Events[0].Objects[0].BoundingBox[2]=6960
Events[0].Objects[0].BoundingBox[3]=6512
Events[0].DetectLine[0][0]=192
Events[0].DetectLine[0][1]=192
Events[0].DetectLine[1][0]=562
Events[0].DetectLine[1][1]=552
Events[0].DetectLine[2][0]=600
Events[0].DetectLine[2][1]=733
Events[0].DetectLine[3][0]=200
Events[0].DetectLine[3][1]=270
...

```

Events[0].Direction=LeftToRight

...

--<boundary>

Content-Type: image/jpeg

Content-Length: <image size>

<Jpeg image data>

--<boundary>

9.1.7 [Event] CrossRegionDetection

When detect some object cross the region, send this event

Event Code	CrossLineDetection		
Event action	Start/Stop		
Event index	0		
Event Data			
Name	Type	R/O	Description
+Object	object	R	The object that cross the region.
++BoundingBox	uint16[4]	R	The detected object bounding box, 4 interge, refer to x's value of left—top point, y's value of left-top point, x's value of right—bottom point, y's value of right-bottom point. Coordinate remap to 0 — 8192.
+Objects	object[]	O	If detect several object, store in this array.
++BoundingBox	uint16[4]	R	The detected object bounding box, 4 interge, refer to x's value of left—top point, y's value of left—top point, x's value of right—bottom point, y's value of right-bottom point. Coordinate remap to 0 — 8192.
+DetectRegion	int[20][2]	R	The detection region, the first array is point list, max item is 20, the second array is point, must be two int, means x and y value, coordinate remap to 0 — 8192.
+Action	char[]	R	The cross action, can be : "Appear", "Disappear", "Cross", "Inside"
+Direction	char[]	O	The cross direction, valid on if the "Action" is "Cross", can be : "Enter", "Leave", "Both".

Event Response Example (multipart , key=value format in body , binary data in body) (response to 4.4.3 snap subscription)

--<boundary>

Content-Type: text/plain
Content-Length: <length>

Events[0].EventBaseInfo.Code=CrossLineDetection
Events[0].EventBaseInfo.Action=Start
Events[0].EventBaseInfo.Index=0
Events[0].Object.BoundingBox[0]= 4392
Events[0].Object.BoundingBox[1]=4136
Events[0].Object.BoundingBox[2]=6960
Events[0].Object.BoundingBox[3]=6512
Events[0].Objects[0].BoundingBox[0]= 4392
Events[0].Objects[0].BoundingBox[1]=4136
Events[0].Objects[0].BoundingBox[2]=6960
Events[0].Objects[0].BoundingBox[3]=6512
Events[0].DetectRegion[0][0]=192
Events[0].DetectRegion[0][1]=192
Events[0].DetectRegion[1][0]=562
Events[0].DetectRegion[1][1]=552
Events[0].DetectRegion[2][0]=600
Events[0].DetectRegion[2][1]=733
Events[0].DetectRegion[3][0]=200
Events[0].DetectRegion[3][1]=270
...
Events[0].Action=Disappear
Events[0].Direction=LeftToRight
...
--<boundary>
Content-Type: image/jpeg
Content-Length: <image size>

<Jpeg image data>
--<boundary>

9.1.8 [Event] QueueStayDetection

When detect the queue stay time too long, send this event

Event Code	QueueStayDetection			
Event action	Start/Stop			
Event index	0			
Event Data				
Name	Type	R/O	Description	Example
+Object	object	O	The object that was stay.	
++BoundingBox	uint16[4]	O	The detected object bounding box, 4 interge, refer to x's value of left—top point, y's value of left—top	[2992,1136,4960,5192]

			point, x's value of right—bottom point, y's value of right—bottom point. Coordinate remap to 0 — 8192.	
+Objects	object[]	O	If detect several object, store in this array.	
++BoundingBox	uint16[4]	O	The detected object bounding box, 4 interge, refer to x's value of left—top point, y's value of left—top point, x's value of right—bottom point, y's value of right—bottom point. Coordinate remap to 0 — 8192.	[2992,1136,4960,5192]
+DetectRegion	int[20][2]	O	The detection region, the first array is point list, max item is 20, the second array is point, must be two int, means x and y value, coordinate remap to 0 — 8192.	
+AreaID	int	O	The area id, begin from 1, if omit, means single area.	2
+PresetID	int	O	The preset id, Valid ID starts from 1; 0 means meaningless, not involved. if omit, means preset is unknown.	1

Event Response Example (multipart , key=value format in body , binary data in body) (response to 4.4.3 snap subscription)

--<boundary>

Content-Type: text/plain

Content-Length: <length>

```

Events[0].EventBaseInfo.Code=QueueStayDetection
Events[0].EventBaseInfo.Action=Start
Events[0].EventBaseInfo.Index=0
Events[0].Object.BoundingBox[0]= 4392
Events[0].Object.BoundingBox[1]=4136
Events[0].Object.BoundingBox[2]=6960
Events[0].Object.BoundingBox[3]=6512
Events[0].Objects[0].BoundingBox[0]= 4392
Events[0].Objects[0].BoundingBox[1]=4136
Events[0].Objects[0].BoundingBox[2]=6960
Events[0].Objects[0].BoundingBox[3]=6512
Events[0].DetectRegion[0][0]=192
Events[0].DetectRegion[0][1]=192
Events[0].DetectRegion[1][0]=562
Events[0].DetectRegion[1][1]=552
Events[0].DetectRegion[2][0]=600
Events[0].DetectRegion[2][1]=733

```

```

Events[0].DetectRegion[3][0]=200
Events[0].DetectRegion[3][1]=270
...
Events[0].AreaID=2
Events[0].PresetID=1
--<boundary>
Content-Type: image/jpeg
Content-Length: <image size>

<Jpeg image data>
--<boundary>

```

9.1.9 [Event] QueueNumDetection

When detect the queue people number exceed limit, send this event

Event Code	QueueNumDetection		
Event action	Pulse		
Event index	0		
Event Data			
Name	Type	R/O	Description
+ManList	object[]	O	The people info list.
++BoundingBox	uint16[4]	O	The detected people bounding box, 4 interge, refer to x's value of left-top point, y's value of left-top point, x's value of right-bottom point, y's value of right-bottom point. Coordinate remap to 0 — 8192.
++Stature	int	R	The people's stature, unit is cm.
+AreaID	int	O	The area id, begin from 1, if omit, means single area.
+PresetID	int	O	The preset id, Valid ID starts from 1; 0 means meaningless, not involved. if omit, means preset is unknown.

Event Response Example (multipart , key=value format in body , binary data in body) (response to 4.4.3 snap subscription)

```

--<boundary>
Content-Type: text/plain
Content-Length: <length>
```

```

Events[0].EventBaseInfo.Code=QueueNumDetection
Events[0].EventBaseInfo.Action=Pulse
Events[0].EventBaseInfo.Index=0
Events[0].ManList[0].BoundingBox[0]= 4392
```

```

Events[0].ManList[0].BoundingBox[1]=4136
Events[0].ManList[0].BoundingBox[2]=6960
Events[0].ManList[0].BoundingBox[3]=6512
Events[0].ManList[0].Stature =176
Events[0].ManList[1].BoundingBox[0]= 275
...
Events[0].AreaID=2
Events[0].PresetID=1
--<boundary>
Content-Type: image/jpeg
Content-Length: <image size>

<Jpeg image data>
--<boundary>

```

9.2 FaceRecognitionServer

9.2.1 Create Face Group

Request URL	http://<server>/cgi-bin/faceRecognitionServer.cgi?action=createGroup			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
groupName	string	R	The face group name, max string length is 127.	Test1
groupDetail	string	O	The description detail of the face group, max string length is 255.	ForTest1
Request Example				
http://192.168.1.108/cgi-bin/faceRecognitionServer.cgi?action=createGroup&groupName=Test1&group Detail=ForTest1				

Response Params (key=value format in body)	
Name	
Type	
R/O	
Description	
Example	
groupId	
string	
R	
The identity of the created face group, max string length is 63.	
10000	
Response Example	
groupId=10000	

9.2.2 Modify Face Group

Request URL	http://<server>/cgi-bin/faceRecognitionServer.cgi?action=modifyGroup			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example

groupID	string	R	The identity of the face group, max string length is 63.	10000
groupName	string	R	The name of the the face group, max string length is 127.	Test1
groupDetail	string	O	Description detail of the face group, max string length is 255.	ForTest1

Request Example

http://192.168.1.108/cgi-bin/faceRecognitionServer.cgi?action=modifyGroup&groupID=10000&groupName=Test1&groupDetail=ForTest1

Response Params (OK in body)

Response Example

OK

9.2.3 Delete Face Group

Request URL	http://<server>/cgi-bin/faceRecognitionServer.cgi?action=deleteGroup			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
groupId	string	R	The identity of the face group, max string length is 63.	10000

Request Example

http://192.168.1.108/cgi-bin/faceRecognitionServer.cgi?action=deleteGroup&groupId=10000

Response Params (OK in body)

Response Example

OK

9.2.4 Deploy Face Group

There are two ways to deploy the group. One is based on the group (putDisposition), and the another one is based on the channel (setGroup).

- Put disposition to group

Deploy the face group to some video channels. If the video channel has been deployed already, it will change the similary.

Request URL	http://<server>/cgi-bin/faceRecognitionServer.cgi?action=putDisposition			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
groupId	string	R	The identity of the face group, max string length is 63.	10000
list	object[]	R	List of disposition info.	

+channel	int	R	Video channel index which starts from 1.	1
+similary	int	R	The threshold of the face similary, 0 — 100.	

Request Example

http://192.168.1.108/cgi-bin/faceRecognitionServer.cgi?action=putDisposition&groupID=10000&list[0].channel=1&list[0].similary=80&list[1].channel=2&list[1].similary=70

Response Params (key=value format in body)

Name	Type	R/O	Description	Example
report	bool[]	R	Result of putting disposition for each request channel.	[true, false]

Response Example

report[0]=true
report[1]=false

- Delete some disposition from group

Remove the deployment of face group from some video channels.

Request URL	http://<server>/cgi-bin/faceRecognitionServer.cgi?action=deleteDisposition			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
groupId	string	R	The identity of the face group, max string length is 63.	10000
channel	int[]	R	Video channel index which starts from 1.	[1,2]

Request Example

http://192.168.1.108/cgi-bin/faceRecognitionServer.cgi?action=deleteDisposition&groupId=10000&channel[0]=1&channel[1]=2

Response Params (key=value format in body)

Name	Type	R/O	Description	Example
report	bool[]	R	Result of deleting disposition for each request channel.	[true, false]

Response Example

report[0]=true
report[1]=false

- set disposition group to channel

Deploy some face groups to one video channel. If the video channel has been deployed already, it will change the similary.

Note: This method will do an overwrite operation.

Request URL	http://<server>/cgi-bin/faceRecognitionServer.cgi?action=setGroup			
Method	GET			

Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
channel	int[]	R	Video channel index which starts from 1.	[1,2]
list	object[]	O	List of disposition info, if not exist, remove all disposition from channel.	
+groupId	int	R	The identity of the face group, max string length is 63.	10002
+similary	int	R	The threshold of the face similary, 0 — 100.	80

Request Example

```
http://192.168.1.108/cgi-bin/faceRecognitionServer.cgi?action=setGroup&channel=1&list[0].groupId=10000&list[0].similary=80&list[1].groupId=10002&list[1].similary=75
```

Response Params (OK in body)
Response Example
OK

- get disposition group from channel

Get the Deployment about the video channel.

Note: If the video channel does not deploy any group, then the response will be success with empty http body.

Request URL	http://<server>/cgi-bin/faceRecognitionServer.cgi?action=getGroup			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
channel	int[]	R	Video channel index which starts from 1.	[1,2]

Request Example

```
http://192.168.1.108/cgi-bin/faceRecognitionServer.cgi?action=getGroup&channel=1
```

Response Params (key=value format in body)				
Name	Type	R/O	Description	Example
groupId	int[]	R	The identity of the face group, max string length is 63.	[10001, 10002, 10003,...]
similary	int[]	R	The threshold of the face similary, 0 — 100.	[80,75,82,...]

Response Example

```
groupId[0]=10001
groupId[1]=10003
groupId[2]=10006
....
similary[0]=80
similary[1]=75
similary[2]=85
```

....

9.2.5 Find Face Group

Find the face group. If the groupID is not present in the URL, it will return all the groups.

Request URL	http://<server>/cgi-bin/faceRecognitionServer.cgi?action=findGroup			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
groupId	char[]	O	The identity of the face group, max string length is 63.	10000
Request Example				
http://192.168.1.108/cgi-bin/faceRecognitionServer.cgi?action=findGroup				

Response Params (key=value format in body)

Name	Type	R/O	Description	Example
GroupList	object[]	R	The face group information list.	
+groupId	char[]	R	The identity of the face group, max string length is 63.	10000
+groupName	char[]	R	Name of the face group, max string length is 127.	test1
+groupDetail	char[]	O	Description detail of the face group, max string length is 255.	fortest1
+groupSize	int	R	The number of face in this face group.	30
+channels	int[]	O	Video channel index which starts from 0.	
+similarity	int[]	O	The threshold of the face similiarity.	
+groupType	char[]	O	The type of face group	BlackListDB
+TimeSection	char[][]	O	The time section of face group	

Response Example

```
GroupList[0].groupId=00001
GroupList[0].groupName=Test1
GroupList[0].groupDetail=ForTest1
GroupList[0].groupSize=30
GroupList[0].channels[0]=1
GroupList[0].channels[1]=2
...
GroupList[0].similarity[0]=80
GroupList[0].similarity[1]=75
...
GroupList[0].groupType=BlackListDB
GroupList[0].TimeSection[0][0]=1 00:00:00-23:59:59
GroupList[0].TimeSection[0][1]=0 00:00:00-23:59:59
GroupList[0].TimeSection[0][2]=0 00:00:00-23:59:59
GroupList[0].TimeSection[0][3]=0 00:00:00-23:59:59
GroupList[0].TimeSection[0][4]=0 00:00:00-23:59:59
```

```
GroupList[0].TimeSection[0][5]=0 00:00:00-23:59:59  
GroupList[0].TimeSection[1][0]=1 00:00:00-23:59:59  
GroupList[0].TimeSection[1][1]=0 00:00:00-23:59:59  
GroupList[0].TimeSection[1][2]=0 00:00:00-23:59:59  
GroupList[0].TimeSection[1][3]=0 00:00:00-23:59:59  
GroupList[0].TimeSection[1][4]=0 00:00:00-23:59:59  
GroupList[0].TimeSection[1][5]=0 00:00:00-23:59:59  
GroupList[0].TimeSection[2][0]=1 00:00:00-23:59:59  
GroupList[0].TimeSection[2][1]=0 00:00:00-23:59:59  
GroupList[0].TimeSection[2][2]=0 00:00:00-23:59:59  
GroupList[0].TimeSection[2][3]=0 00:00:00-23:59:59  
GroupList[0].TimeSection[2][4]=0 00:00:00-23:59:59  
GroupList[0].TimeSection[2][5]=0 00:00:00-23:59:59  
GroupList[0].TimeSection[3][0]=1 00:00:00-23:59:59  
GroupList[0].TimeSection[3][1]=0 00:00:00-23:59:59  
GroupList[0].TimeSection[3][2]=0 00:00:00-23:59:59  
GroupList[0].TimeSection[3][3]=0 00:00:00-23:59:59  
GroupList[0].TimeSection[3][4]=0 00:00:00-23:59:59  
GroupList[0].TimeSection[3][5]=0 00:00:00-23:59:59  
GroupList[0].TimeSection[4][0]=1 00:00:00-23:59:59  
GroupList[0].TimeSection[4][1]=0 00:00:00-23:59:59  
GroupList[0].TimeSection[4][2]=0 00:00:00-23:59:59  
GroupList[0].TimeSection[4][3]=0 00:00:00-23:59:59  
GroupList[0].TimeSection[4][4]=0 00:00:00-23:59:59  
GroupList[0].TimeSection[4][5]=0 00:00:00-23:59:59  
GroupList[0].TimeSection[5][0]=1 00:00:00-23:59:59  
GroupList[0].TimeSection[5][1]=0 00:00:00-23:59:59  
GroupList[0].TimeSection[5][2]=0 00:00:00-23:59:59  
GroupList[0].TimeSection[5][3]=0 00:00:00-23:59:59  
GroupList[0].TimeSection[5][4]=0 00:00:00-23:59:59  
GroupList[0].TimeSection[5][5]=0 00:00:00-23:59:59  
GroupList[0].TimeSection[6][0]=1 00:00:00-23:59:59  
GroupList[0].TimeSection[6][1]=0 00:00:00-23:59:59  
GroupList[0].TimeSection[6][2]=0 00:00:00-23:59:59  
GroupList[0].TimeSection[6][3]=0 00:00:00-23:59:59  
GroupList[0].TimeSection[6][4]=0 00:00:00-23:59:59  
GroupList[0].TimeSection[6][5]=0 00:00:00-23:59:59
```

```
GroupList[1].groupID=00003  
GroupList[1].groupName=Test3  
GroupList[1].groupDetail=ForTest3  
GroupList[1].groupSize=50  
GroupList[1].channels[0]=1  
GroupList[1].channels[1]=2  
...  
GroupList[1].similarity[0]=70  
GroupList[1].similarity[1]=85
```

...

9.2.6 Re-Abstract Feature By Group

- Start ReAbstract

Abstract features for the groups.

About the process of the re-extract, the device will use an event named "**FaceFeatureAbstract**" to report the process.

Request URL	http://<server>/cgi-bin/faceRecognitionServer.cgi?action=groupReAbstract		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
groupId	char[][64]	R	The identity of the face group, max string length is 63.
Request Example			
http://192.168.1.108/cgi-bin/faceRecognitionServer.cgi?action=groupReAbstract&groupId[0]=10000&groupId[1]=10001			

Response Params (key=value format in body)			
Name	Type	R/O	Description
token	int	R	The identity of this operation.
Response Example			
token=12345			

- Stop ReAbstract

Request URL	http://<server>/cgi-bin/faceRecognitionServer.cgi?action=stopGroupReAbstract		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
token	int	R	The identity of this operation.
Request Example			
http://192.168.1.108/cgi-bin/faceRecognitionServer.cgi?action=stopGroupReAbstract&token=12345			

Response Params (OK in body)
Response Example
OK

9.2.7 Add Person

Request URL	http://<server>/cgi-bin/faceRecognitionServer.cgi?action=addPerson		
Method	POST		
Request Params (key=value format in URL , binary data in body)			
Name	Type	R/O	Description

groupId	char[]	R	The identity of the face group that this person to add. Max string length is 63.	10000
name	char[]	R	The person name, max string length is 63.	ZhangSan
birthday	char[]	O	The person's birthday, ex: "1980-01-01".	"1980-01-01"
sex	char[]	O	Sex, it can be "Male", "Female", "Unknown".	Male
country	char[]	O	The country name, length must be 2, and value should be according to ISO3166.	CN
province	char[]	O	The province name, max string length is 63.	XXX
city	char[]	O	The city name, max string length is 63.	YYY
certificateType	char[]	O	The certificate type. It can be: "IC", "Passport", "Unknown".	IC
id	char[]	O	The ID of certificate type, max string length is 31.	3333333333333

Request Example

```
POST http://<server>/cgi-bin/faceRecognitionServer.cgi?action=addPerson&groupId=10000&name=ZhangSan&birthday=1980-01-05&sex=Male&country=CN&province=XXX&city=YYY HTTP/1.1
Content-Type: image/jpeg
Content-Length: <image size>

<JPEG image data>
```

Response Params (key=value format in body)

Name	Type	R/O	Description	Example
uid	char[32]	R	The id for this Person, max string length is 31.	"0005"

Response Example

```
uid=0005
```

9.2.8 Modify Person

Modify a person's info.

Note: If you do not want to change the image about the person, the request should not contain the image data.

Note: You should provide at least one optional param to update.

Request URL	http://<server>/cgi-bin/faceRecognitionServer.cgi?action=modifyPerson			
Method	POST			
Request Params (key=value format in URL , binary data in body)				
Name	Type	R/O	Description	Example
uid	char[]	R	The identity of the Person, max string length is 31.	0005

groupId	char[]	R	The identity of the Face Group that this Person in. max string length is 63.	10000
name	char[]	O	The person's name, max string length is 63.	ZhangSan
birthday	char[]	O	The person's birthday, ex: "1980-01-01".	"1980-01-01"
sex	char[]	O	Sex, it can be "Male", "Female", "Unknown".	Male
country	char[]	O	The country name, length must be 2, and value should be according to ISO3166.	CN
province	char[]	O	The province name, max string length is 63.	XXX
city	char[]	O	The city name, max string length is 63.	YYY
certificateType	char[]	O	The certificate type. It can be: "IC", "Passport", "Unknown".	IC
id	char[]	O	The ID of certificate type, max string length is 31.	3333333333333

Request Example

POST http://<server>/cgi-bin/faceRecognitionServer.cgi?action=modifyPerson&uid=0005&groupId=10000&name=ZhangSan&birthday=1980-01-05&sex=Male&country=CN&province=XXX&city=YYY HTTP/1.1
Content-Type: image/jpeg
Content-Length: <image size>
<JPEG image data>

Response Params (OK in body)

Response Example

OK

9.2.9 Delete Person

Request URL	http://<server>/cgi-bin/faceRecognitionServer.cgi?action=deletePerson			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
uid	char[]	R	The identity of the person, max string length is 31.	001
groupId	char[]	R	The identity of the face group that this Person in. max string length is 63.	10000

Request Example

http://192.168.1.108/cgi-bin/faceRecognitionServer.cgi?action=deletePerson&uid=001&groupId=10000

Response Params (OK in body)

Response Example

OK

9.2.10 Find Person

- Start to find

Note: the returned token will be expired after 60 seconds without any doFind call.

Request URL	http://<server>/cgi-bin/faceRecognitionServer.cgi?action=startFind			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
condition	object	R	Search scope condition.	
+GroupId	char[]	R	The list of identity of the face group, max string length is 63.	["10000","10001"]
person	object	O	Person condition.	
+Name	char[]	O	Person Name, max string length is 63.	xxx
+Sex	char[]	O	Sex, it can be "Male", "Female", "Unknown".	"Female"
+Country	char[]	O	Country name, length must be 2, and value should be according to ISO3166.	CN
+Province	char[]	O	Province name, max string length is 63.	
+City	char[]	O	City name, max string length is 63.	
+CertificateType	char[]	O	Certificate Type. It can be: "IC", "Passport", "Unknown".	Passport
+ID	char[]	O	Person ID of CertificateType, max string length is 31.	
+FeatureState	int	O	Feature State, 0:Unknown, 1:Failed, 2:OK.	1

Request Example

http://<server>/cgi-bin/faceRecognitionServer.cgi?action=startFind&condition.GroupID[0]=10000&condition.GroupID[1]=10003&person.Sex=Male&person.Country=CN&person.FeatureState=1

Response Params (key=value format in body)				
Name	Type	R/O	Description	Example
token	uint	R	Token for this search, use this token to get result and stop search.	123456789
totalCount	int	R	Result num, return -1 means still searching.	24

Response Example

token=123456789
totalCount=24

- Get find result

Note: the returned token will be expired after 60 seconds without any doFind call.

Request URL	http://<server>/cgi-bin/faceRecognitionServer.cgi?action=doFind		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
token	uint	R	Token for this search, use this token to get result and stop search.
index	uint	R	The index in search result, should between 0 and totalCount -1.
Request Example			
http://<server>/cgi-bin/faceRecognitionServer.cgi?action=doFind&token=123456789&index=0			

Response Params (multipart, key=value format in body, binary in body)			
Name	Type	R/O	Description
person	object	R	Person condition.
+UID	string	R	The identity of the person, max string length is 31.
+GroupID	string	R	The identity of the face group that this Person in. max string length is 63.
+Name	string	R	The person name, max string length is 63.
+Sex	string	O	Sex, it can be "Male", "Female", "Unknown".
+Birthday	string	O	The person's birthday, ex: "1980-01-01".
+Country	string	O	Country name, length must be 2, and value should be according to ISO3166.
+Province	string	O	Province name, max string length is 63.
+City	string	O	City name, max string length is 63.
+CertificateType	string	O	Certificate Type, can be: "IC", "Passport", "Unknown".
+ID	string	O	Person ID of CertificateType, max string length is 31.
+FeatureState	int	O	Feature State, 0:Unknown, 1:Failed, 2:OK.
Response Example			
HTTP/1.1 200 OK			
Server: Device/1.0			
Content-Type: multipart/x-mixed-replace; boundary=<boundary>			
Content-Length: <length>			

```
--<boundary>
Content-Type: text/plain
Content-Length: <length>

person.UID=0005
person.GroupID=10000
person.Name=ZhangSan
person.Birthday=1980-01-01
person.Sex=Male
person.Country=CN
person.Province=XXX
person.City=YYY
person.CertificateType=IC
person.ID=1234567890
person.FeatureState=0
--<boundary>
Content-Type: image/jpeg
Content-Length: <image size>

< jpeg image data ... >
--<boundary>--
```

- Stop finding

Request URL	http://<server>/cgi-bin/faceRecognitionServer.cgi?action=stopFind			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
token	uint	R	The token for this search, use this token to get result and stop search.	123456789
Request Example				
http://<server>/cgi-bin/faceRecognitionServer.cgi?action=stopFind&token=123456789				

Response Params (OK in body)

Response Example

OK

9.2.11 Re-Abstract Features By Person

- Start ReAbstract

About the process of the re-extract, the device will use an event named "**FaceFeatureAbstract**" to report the process.

Request URL	http://<server>/cgi-bin/faceRecognitionServer.cgi?action=reAbstract
Method	GET

Request Params (key=value format in URL)

Name	Type	R/O	Description	Example
UID	char[][31]	O	The list of identity of person, max string length is 31.	["10000", "10001"]

Request Example

http://192.168.1.108/cgi-bin/faceRecognitionServer.cgi?action=reAbstract&uid[0]=001&uid[1]=002

Response Params (OK in body)

Response Example

OK

- Stop ReAbstract

Request URL http://<server>/cgi-bin/faceRecognitionServer.cgi?action=stopReAbstract

Method GET

Request Params (none)

Name	Type	R/O	Description	Example
------	------	-----	-------------	---------

Request Example

http://192.168.1.108/cgi-bin/faceRecognitionServer.cgi?action=stopReAbstract

Response Params (OK in body)

Response Example

OK

9.2.12 [Config] Face Recognition AlarmOut Setting

Config Data Params

Name	Type	R/O	Description	Example
FaceRecognitionAlarm	object[]	R	Each face group has one config object in this array.	
+GroupID	char[]	R	The face group ID, max string length is 63.	0017
+GroupName	char[]	R	The face group name, max string length is 127.	wsd
+AlarmOutEnable	Bool	R	Enable AlarmOut or not.	true
+AlarmChannel	object[]	R	Each AlarmOut channel has one config object in this array.	
++AlarmRuleMask	Int	R	Alarm rule mask. <ul style="list-style-type: none"> Bit 0 : recognition success Bit 1 : recognition failed 	0
++AlarmOutLatch	Int	R	Alarm out delay, unit is second, value between 1 and 300.	5

Please refer to "4.2.1 Get and Set Configure" for configuration getting and setting. Specific examples are as follows :

Get Config Request Example

http://10.0.0.8/cgi-bin/configManager.cgi?action=getConfig&name=FaceRecognitionAlarm

Get Config Response Example

```

table.FaceRecognitionAlarm[0].GroupID=0017
table.FaceRecognitionAlarm[0].GroupName=wsd
table.FaceRecognitionAlarm[0].AlarmOutEnable=true
table.FaceRecognitionAlarm[0].AlarmChannel[0].AlarmRuleMask=0
table.FaceRecognitionAlarm[0].AlarmChannel[0].AlarmOutLatch=5
table.FaceRecognitionAlarm[0].AlarmChannel[1].AlarmRuleMask=0
table.FaceRecognitionAlarm[0].AlarmChannel[1].AlarmOutLatch=8
...
table.FaceRecognitionAlarm[1].GroupID=0018
table.FaceRecognitionAlarm[1].GroupName=cst
table.FaceRecognitionAlarm[1].AlarmOutEnable=true
table.FaceRecognitionAlarm[1].AlarmChannel[0].AlarmRuleMask=0
table.FaceRecognitionAlarm[1].AlarmChannel[0].AlarmOutLatch=10
table.FaceRecognitionAlarm[1].AlarmChannel[1].AlarmRuleMask=0
table.FaceRecognitionAlarm[1].AlarmChannel[1].AlarmOutLatch=15
...

```

Set Config Request Example

http://10.0.0.8/cgi-bin/configManager.cgi?action=setConfig&FaceRecognitionAlarm[0].GroupID=0017&FaceRecognitionAlarm[0].GroupName=wsd&FaceRecognitionAlarm[0].AlarmOutEnable=true&FaceRecognitionAlarm[0].AlarmChannel[0].AlarmRuleMask=0&FaceRecognitionAlarm[0].AlarmChannel[0].AlarmOutLatch=5

Set Config Response Example

OK

9.2.13 Find Person by Picture

- Start to find

Start to find person in face groups by picture. The search may last for some time, so the response may push at regular intervals until 100% Progress

Note: If you want to find person in face groups by person info, see above "Find Person"API.

Note: The returned token will be expired after 60 seconds without any doFind call.

Request URL	http://<server>/cgi-bin/faceRecognitionServer.cgi?action=startFindByPic			
Method	POST			
Request Params (multipart; key=value format in URL, binary in body)				
Name	Type	R/O	Description	Example
GroupID	char[]()	R	Face groups to find, max string length is 63.	["10000","10001"]
Similarity	int	R	Similarity percent, 1 — 100.	80
MaxCandidate	int	O	Max Candidate result number.	500
Request Example				
POST http://<server>/cgi-bin/faceRecognitionServer.cgi?action=startFindByPic&GroupID[0]=00001&GroupID[1]=00003&Similarity=80&MaxCandidate=500				

Content-Type: image/jpeg
Content-Length: <image size>

<JPEG data>

Response Params (multipart; json format in body)

Name	Type	R/O	Description	Example
token	uint	R	The token of this search, use this token to get result and stop search.	123456789
progress	uint	R	Search Progress, 100 means finished.	20
totalCount	int	R	Result num, return -1 means still searching.	-1

Response Example

HTTP/1.1 200 OK

Server: Device/1.0

Content-Type: multipart/x-mixed-replace; boundary=<boundary>

Connection: closed

--<boundary>

Content-Type: application/json

Content-Length: <length>

{ "token": 123456789,

 "progress": 20

 "totalCount": -1

}

--<boundary>

Content-Type: application/json

Content-Length: <length>

{ "token": 123456789,

 "progress": 60

 "totalCount": -1

}

--<boundary>

Content-Type: application/json

Content-Length: <length>

{ "token": 123456789,

 "progress": 100

 "totalCount": 350

}

--<boundary>--

- Get the find result

Get the find result, reply using multipart format, first part is json string to describe all candidate person, then the following parts are the person's pictures, refer by UID and GroupID in part header Content-Info.

Note: the returned token will be expired after 60 seconds without any doFind call.

Request URL	http://<server>/cgi-bin/faceRecognitionServer.cgi?action=doFindByPic		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
token	uint	R	The token of this search, use this token to get result and stop search.
index	uint	R	The index in search result, should between 0 and totalCount -1.
count	uint	R	Number of result to get, start from index.
Request Example			
http://<server>/cgi-bin/faceRecognitionServer.cgi?action=doFindByPic&token=123456789&index=0&count=10			

Response Params (multipart; json format in body, binary in body)				
Name	Type	R/O	Description	Example
Found	int	R	Number of result person that return.	
Candidates	object[]	R	Candidates Person.	
+Person	object	R	Person Info.	
++UID	char[]	R	System id for this Person, max string length is 31.	1234
++GroupID	char[]	R	The identity of the Face Group that this Person in. max string length is 63.	100001
++Name	char[]	R	Person Name, max string length is 63.	xxx
++Birthday	char[]	O	Birthday ex: "1980-01-01".	"1980-01-01"
++Sex	char[]	O	Sex, it can be "Male", "Female", "Unknown".	"Female"
++Country	char[]	O	Country name, length must be 2, value should be according to ISO3166.	CN
++Province	char[]	O	Province name, max string length is 63.	
++City	char[]	O	City name, max string length is 63.	
++CertificateType	char[]	O	Certificate Type. It can be "IC", "Passport", or "Unknown".	Passport
++ID	char[]	O	Person ID of CertificateType, max string length is 31.	1234567890

++FeatureState	int	O	Feature State, 0:Unknown, 1:Failed, 2:OK.	0
++HomeAddress	char[]	O	Home Address, ex: "binanRoad NO1199"	binanRoad NO1199
+Similarity	int	R	Similarity.	85

Response Example

HTTP/1.1 200 OK

Server: Device/1.0

Content-Type: multipart/x-mixed-replace; boundary=<boundary>

Connection: closed

--<boundary>

Content-Type: application/json

Content-Length: <length>

```
{
  "Found" : 10,
  "Candidates" : [
    {
      "person" : {
        "UID" : "0001",
        "GroupID" : "001",
        "Name" : "ZhangSan",
        "Birthday" : "1980-01-05",
        "Sex" : "Male",
        ...
      },
      "Similarity" : 85
    },
    {
      "person" : {
        "UID" : "0002",
        "GroupID" : "002",
        "Name" : "LiSi",
        "Birthday" : "1980-01-06",
        "Sex" : "Male",
        ...
      },
      "Similarity" : 80
    },
    { ... }, ...
  ]
}
```

--<boundary>

Content-Info: UID=0001&GroupID=001

Content-Type: image/jpeg

Content-Length: <length>

```

< jpeg image data ... >
--<boundary>
Content-Info: UID=0002&GroupID=002
Content-Type: image/jpeg
Content-Length: <length>

```

```

< jpeg image data ... >
--<boundary>
...

```

- Stop finding

Request URL	http://<server>/cgi-bin/faceRecognitionServer.cgi?action=stopFindByPic			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
token	uint	R	Token for this search, use this token to get result and stop search.	123456789
Request Example				
http://<server>/cgi-bin/faceRecognitionServer.cgi?action=stopFindByPic&token=123456789				

Response Params (OK in body)
Response Example
OK

9.2.14 Find History Person by Picture

- Start to find

Start to find person in capture history by picture. The search may last for some time, so the response may push at regular intervals until 100% Progress.

Note: If you want to find person in capture history by person info, please refer to "mediaFileFind"API.

Note: the returned token will be expired after 60 seconds without any doFind call.

Request URL	http://<server>/cgi-bin/faceRecognitionServer.cgi?action=startFindHistoryByPic			
Method	POST			
Request Params (multipart; key=value format in URL, binary in body)				
Name	Type	R/O	Description	Example
Channel	int	R	Video channel index which starts from 0.	0
StartTime	string	R	Start time to search, ex: 2018-01-13T00:00:00Z.	2018-01-13T00:00:00Z
EndTime	string	R	End time to search, ex: 2018-01-14T00:00:00Z.	2018-01-14T00:00:00Z
Similarity	int	R	Similarity percent, 1 — 100.	80
MaxCandidate	int	O	Max Candidate result number.	200
Request Example				

```

POST http://<server>/cgi-bin/faceRecognitionServer.cgi?action=startFindHistoryByPic&Channel=0&StartTime=2018-01-13T00:00:00Z&EndTime=2018-01-14T00:00:00Z&Type=All&Similarity=80&MaxCandidate=500
Content-Type: image/jpeg
Content-Length: <image size>

<JPEG data>

```

Response Params (multipart; json format in body)

Name	Type	R/O	Description	Example
token	uint	R	Token for this search, use this token to get result and stop search.	123456789
progress	uint	R	Search Progress, 100 means finished.	20
totalCount	int	R	Result num, return -1 means still searching.	-1

Response Example

HTTP/1.1 200 OK

Server: Device/1.0

Content-Type: multipart/x-mixed-replace; boundary=<boundary>

Connection: closed

--<boundary>

Content-Type: application/json

Content-Length: <length>

{ "token": 123456789,

 "progress": 20

 "totalCount": -1

}

--<boundary>

Content-Type: application/json

Content-Length: <length>

{ "token": 123456789,

 "progress": 60

 "totalCount": -1

}

--<boundary>

Content-Type: application/json

Content-Length: <length>

{ "token": 123456789,

 "progress": 100

 "totalCount": 350

}

--<boundary>--

- Get find result

Get the find result, reply by multipart, first part is json string to describe all candidate person, then the following part is the person's picture, refer by UID in part header Content-Info.

Note: the returned token will be expired after 60 seconds without any doFind call.

Request URL	http://<server>/cgi-bin/faceRecognitionServer.cgi?action=doFindByPic		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
token	uint	R	Token for this search, use this token to get result and stop search.
index	uint	R	The index in search result, should between 0 and totalCount -1.
count	uint	R	Number of result person to get, start from Index.
Request Example			
http://<server>/cgi-bin/faceRecognitionServer.cgi?action=doFindHistoryByPic&token=123456789&index=0&count=12			

Response Params (multipart; json format in body, binary in body)			
Name	Type	R/O	Description
Found	int	R	Number of result person that return.
Candidates	object[]	R	Candidates Person.
+Person	object	R	Person Info.
++UID	string	R	System id for this Person, max string length is 63.
++Sex	string	O	Sex. It can be "Male", "Female" or "Unknown".
++Age	int	O	Age.
++Glasses	int	O	Glasses Status, 0: all, 1: not wear, 2: wear.
+Similarity	int	R	Similarity.
+Time	string	O	Appear time of the Person, format is "2013-09-02 00:00:00".

Response Example

HTTP/1.1 200 OK

Server: Device/1.0

Content-Type: multipart/x-mixed-replace; boundary=<boundary>

Connection: closed

--<boundary>

Content-Type: application/json

```

Content-Length: <length>

{
  "Found" : 12,
  "Candidates" : [
    {
      "person" : {
        "UID" : "0001",
        "Sex" : "Male",
        "Age" : 30,
        "Glasses": 1
      },
      "Similarity" : 85
      "Time": "2013-09-02 00:00:00"
    },
    {
      "person" : {
        "UID" : "0002",
        "Sex" : "Male",
        "Age" : 50,
        "Glasses": 2
      },
      "Similarity" : 80
      "Time": "2013-09-02 00:00:00"
    },
    { ... }, ...
  ]
}

--<boundary>
Content-Info: UID=0001
Content-Type: image/jpeg
Content-Length: <length>

< jpeg image data ... >
--<boundary>
Content-Info: UID=0002
Content-Type: image/jpeg
Content-Length: <length>

< jpeg image data ... >
--<boundary>
...

```

- Stop finding

Request URL	http://<server>/cgi-bin/faceRecognitionServer.cgi?action=stopFindHistoryByPic
Method	GET

Request Params (key=value format in URL)

Name	Type	R/O	Description	Example
token	uint	R	Token for this search, use this token to get result and stop searching.	123456789

Request Example

http://<server>/cgi-bin/faceRecognitionServer.cgi?action=stopFindHistoryByPic&token=123456789

Response Params (OK in body)

Response Example

OK

9.2.15 [Event] FaceDetection

When the video channel disposition with some face group, and the video channel detect a face, send this event.

Event Code	FaceDetection			
Event action	Start/Stop			
Event index	0			
Event Data				
Name	Type	R/O	Description	Example
+Faces	object[]	R	The detected faces info.	
++BoundingBox	int[4]	R	The detected face bounding box, 4 interge, refer to x's value of left—top point, y's value of left—top point, x's value of right—bottom point, y's value of right—bottom point. Coordinate remap to 0 — 8192.	[2992,136,6960,8192]
++Sex	char[]	O	Sex, it can be "Man", "Woman".	"Man"
++Age	int	O	Age.	40
++Feature	char[][]	O	Face feature, can be some of the following: "WearGlasses", "SunGlasses", "NoGlasses", "Smile", "Anger", "Sadness", "Disgust", "Fear", "Surprise", "Neutral", "Laugh", "Happy", "Confused", "Scream".	["WearGlasses", "Smile"]
++Eye	int	O	Eye status, 0: not detected, 1: close eye, 2: open eye.	2
++Mouth	int	O	Mouth status, 0: not detected, 1: close mouth, 2: open mouth.	1
++Mask	int	O	Mask status, 0: not detected, 1: not wearing mask, 2: wearing mask.	1
++Beard	int	O	Beard status, 0: not detected, 1: no	2

		beard, 2: has beard.	
--	--	----------------------	--

Event Response Example (multipart , JSON format in body) (response to 4.9.17 event subscription)

--<boundary>

Content-Type: text/plain

Content-Length: <length>

```
Code=FaceDetection;action=Start;index=0;data={  
    "Faces": [ { "BoundingBox": [2992,136,6960,8192],  
        "Sex": "Man",  
        "Age": 40,  
        "Feature": [ "WearGlasses", "Smile"],  
        "Eye": 2,  
        "Mouth": 1,  
        "Mask": 1,  
        "Beard": 2  
    }, {...}, ... ]  
}
```

--<boundary>

9.2.16 [Event] FaceRecognition

When the video channel disposition with some face group, and the video channel detect a face, after recognize in the face groups, send this event.

Event Code	FaceRecognition			
Event action	Pulse			
Event index	0			
Event Data				
Name	Type	R/O	Description	Example
+UID	String	R	The identity of the Person, max string length is 31.	
+Candidates	object[]	O	The Candidates person list.	
++Person	object	O	The information of candidate person.	
+++UID	string	R	The identity of the Person, max string length is 31.	
+++GroupID	string	R	The identity of the Face Group that this Person in. max string length is 63.	
+++Name	string	O	Person Name, max string length is 63.	ZhangSan
+++Birthday	string	O	Birthday ex: "1980-01-01".	"2000-01-01"
+++Sex	string	O	Sex, it can be "Male", "Female", "Unknown".	"Man"
+++Country	string	O	Country name. The length must be	

			2, and value should be according to ISO3166.	
+++Province	string	O	Province name, max string length is 63.	
+++City	string	O	City name, max string length is 63.	
+++CertificateType	string	O	Certificate Type. It can be: "IC", "Passport", "Unknown".	
+++ID	string	O	Person ID of CertificateType, max string length is 31.	
++Similarity	int	O	Similarity of the Candidates person and the detected person, value between 1 — 100.	89
+Face	object	O	The attribute information of face.	
++Sex	string	O	Sex, it can be "Man", "Woman".	"Man"
++Age	int	O	Age.	23
++Feature	string[]	O	Face feature, can be some of the following : "WearGlasses", "SunGlasses", "NoGlasses", "Smile", "Anger", "Sadness", "Disgust", "Fear", "Surprise", "Neutral", "Laugh", "Happy", "Confused", "Scream".	["WearGlasses", "Anger"]
++Eye	int	O	Eye status, 0: not detected, 1: close eye, 2: open eye.	1
++Mouth	int	O	Mouth status, 0: not detected, 1: close mouth, 2: open mouth.	1
++Mask	int	O	Mask status, 0: not detected, 1: not wearing mask, 2: wearing mask.	1
++Beard	int	O	Beard status, 0: not detected, 1: no beard, 2: has beard.	1

Event Response Example (multipart , key=value format in body , binary data in body) (response to 4.4.3 snap subscription)

```
--<boundary>
Content-Type: text/plain
Content-Length: <length>
```

```
Events[0].EventBaseInfo.Code=FaceRecognition
Events[0].EventBaseInfo.Action=Pulse
Events[0].EventBaseInfo.Index=0
Events[0].UID=00105
Events[0].Candidates[0].Person.UID=0012
Events[0].Candidates[0].Person.GroupID=10000
Events[0].Candidates[0].Person.Name=ZhangSan
Events[0].Candidates[0].Person.Birthday=1980-01-02
Events[0].Candidates[0].Person.Sex=Male
...
```

```

Events[0].Candidates[0].Similarity=80
Events[0].Candidates[1].Person.UID=0014
Events[0].Candidates[1].Person.GroupID=10000
Events[0].Candidates[1].Person.Name=Lisi
Events[0].Candidates[1].Person.Birthday=1980-01-05
Events[0].Candidates[1].Person.Sex=Male
...
Events[0].Candidates[1].Similarity=75
...
Events[0].Face.Sex=Man
Events[0].Face.Age=20
Events[0].Face.Feature[0]=SunGlasses
Events[0].Face.Feature[1]=Smile
Events[0].Face.Eye=2
Events[0].Face.Mouth=1
Events[0].Face.Mask=1
Events[0].Face.Beard=2
--<boundary>
Content-Type: image/jpeg
Content-Length: <image size>

<Jpeg image data>
--<boundary>

```

9.2.17 [Event] FaceFeatureAbstract

When Re-Abstract Feature By Group or By Person, the abstract progress detail will send in this event.

Event Code	FaceFeatureAbstract			
Event action	Start/Stop			
Event index	0			
Event Data				
Name	Type	R/O	Description	Example
+Infos	object[100]	R	Abstrace detail Info, max size is 100.	
++State	String	R	Abstract state, it can be : ● "Success" : Abstract success; ● "False" : Failed to abstract; ● "Process" : In Process;	Progress
++Process	int	O	The abstract progress.	30
++UID	string	O	The identity of the person, max string length is 31.	20005
++GroupID	string	O	The identity of the face group, max string length is 63.	10000

Event Response Example (multipart , JSON format in body) (response to 4.9.17 event subscription)

```
--<boundary>
Content-Type: text/plain
Content-Length: <length>

Code=FaceFeatureAbstract;action=Start;index=0;data={

    "Infos": [ { "State": "Progress",
        "Progress": 30,
        "UID": "20005",
        "GroupID": "10000"
    }, {...}, ... ]
}

--<boundary>
```

9.2.18 [Config] Face Recognition Event Handler Setting

Config Data Params				
Name	Type	R/O	Description	Example
FaceRecognitionEventHandler	object[]	R	Each face group has one config object in this array.	
+GroupId	char[]	R	The face group ID, max string length is 63.	0017
+GroupName	char[]	R	The face group name, max string length is 127.	wsd
+EventEnableMask	int	R	Report event mask, 0 means not to report event. ● Bit 0 : recognition success ● Bit 1 : recognition failed	3
+RecordEnableMask	int	R	Record media file mask., 0 means not to record. ● Bit 0 : recognition success ● Bit 1 : recognition failed	0
+RecordLatch	int	R	Record latch time, unit is second.	10
+SnapEnableMask	int	R	Snap picture mask., 0 means not to snap picture ● Bit 0 : recognition success ● Bit 1 : recognition failed	3
+MailEnableMask	int	R	Send mail mask., 0 means not to send mail. ● Bit 0 : recognition success ● Bit 1 : recognition failed	0

Please refer to "4.2.1 Get and Set Configure" for configuration getting and setting. Specific examples are as follows :

Get Config Request Example

```
http://10.0.0.8/cgi-bin/configManager.cgi?action=getConfig&name=FaceRecognitionEventHandler
```

Get Config Response Example

```
table.FaceRecognitionEventHandler[0].GroupID=0017  
table.FaceRecognitionEventHandler[0].GroupName=wsd  
table.FaceRecognitionEventHandler[0].EventEnableMask=3  
table.FaceRecognitionEventHandler[0].RecordEnableMask=0  
table.FaceRecognitionEventHandler[0].RecordLatch=10  
table.FaceRecognitionEventHandler[0].SnapEnableMask=3  
table.FaceRecognitionEventHandler[0].MailEnableMask=0  
  
...  
table.FaceRecognitionEventHandler[1].GroupID=0018  
table.FaceRecognitionEventHandler[1].GroupName=cst  
table.FaceRecognitionEventHandler[1].EventEnableMask=3  
table.FaceRecognitionEventHandler[1].RecordEnableMask=0  
table.FaceRecognitionEventHandler[1].RecordLatch=10  
table.FaceRecognitionEventHandler[1].SnapEnableMask=3  
table.FaceRecognitionEventHandler[1].MailEnableMask=0  
  
...
```

Set Config Request Example

http://10.0.0.8/cgi-bin/configManager.cgi?action=setConfig&FaceRecognitionEventHandler[0].GroupID=0017&FaceRecognitionEventHandler[0].GroupName=wsd&FaceRecognitionEventHandler[0].EventEnableMask=3&FaceRecognitionEventHandler[0].RecordEnableMask=0&FaceRecognitionEventHandler[0].RecordLatch=10&FaceRecognitionEventHandler[0].SnapEnableMask=3&FaceRecognitionEventHandler[0].MailEnableMask=0

Set Config Response Example

OK

9.2.19 [Config] Face-ID Recognition Threshold

Request URL	http://<server>/cgi-bin/configManager.cgi?action=setConfig			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
CitizenPictureCompareRule	object	R	Citizen picture compare	
+Threshold	uint8	R	Face-ID comparison threshold [1, 100]	60
Request Example				
http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&CitizenPictureCompareRule.Threshold=60				

Response Params (OK in body)				
Name	Type	R/O	Description	Example
Response Example				
OK				

9.2.20 Export Face Database

The exported data is binary.

Request URL	http://<server>/cgi-bin/api/FaceLiblmExport/export		
Method	POST		
Request Params (JSON format in body)			
Name	Type	R/O	Description
groupId	char[64]	R	Person group ID
Password	char[64]	O	Unzip password
Request Example			
{	<pre>"groupId": "10", "password": "abcd"</pre>		
}			

Response Params (binary in body)	
Name	
Type	
R/O	
Description	
Example	
Response Example	
HTTP/1.1 200 OK	
NTP Server Device/1.0	
Content-Type: application/octet-stream	
Content-Length: <length>	
< binary data>	

9.2.21 Importing Face Database

The imported data is binary.

Request URL	http://<server>/cgi-bin/api/FaceLiblmExport/import		
Method	POST		
Request Params (multipart in body)			
Name	Type	R/O	Description
password	char[64]	O	Unzip password
Request Example			
POST http://<server>/cgi-bin/api/FaceLiblmExport/import	HTTP/1.1		
User-Agent: client/1.0			
Content-Type: multipart/x-mixed-replace; boundary=<boundary>			
--<boundary>			
Content-Type: application/json			
Content-Length: 20			
{	<pre>"password": "abcd"</pre>		
}			

```
--<boundary>
Content-Type: application/octet-stream
Content-Length: 800

< binary data>
--<boundary>
```

Response Params (JSON format in body)

Name	Type	R/O	Description	Example
Response Example				
{}				

9.2.22 Remote Delete Person

Request URL	http://<server>/cgi-bin/api/RemoteFaceRecognition/deletePerson			
Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
channel	int	R	logical channel	0
person	object	O	person info	
+UID	char[32]	O	The identity of the person.	"120837"
+GroupId	char[64]	O	The identity of the face group that this Person is in. Specify both GroupID and UID to delete.	"00001"
Request Example				
{				
"channel": 0,				
"person": {				
"UID": "120837",				
"GroupId": "00001"				
}				
}				

Response Params (JSON format in body)

Name	Type	R/O	Description	Example
Response Example				
{}				

9.2.23 Add Face Lib Download Task

Request URL	http://<server>/cgi-bin/api/NetFileTransfer/addFaceLibDownloadTask			
Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example

FaceLibVersion	char[32]	R	face lib version	"xxxx"
PacketTotal	uint	R	packet total num	10
PacketIndex	uint	R	packet index, starts from 1	3
TaskInfo	object[100]	R	task info	
+URLList	char[4][256]	R	resource list format as: "sftp://username:password @ip:port/xxxx/1.jpg" "xx"]	["sftp://admin:admin@10.12 .4.84:554/media/1.jpg", "xx"]
+ID	char[32]	R	driver ID	"123456789"
+FaceUUID	char[32]	R		"xxxx"
GroupId	char[64]	O	group ID	"1"

Request Example

```
{
    "FaceLibVersion": "xxxx",
    "PacketTotal": 10,
    "PacketIndex": 3,
    "TaskInfo": [
        "URLList": ["sftp://admin:admin@10.12.4.84:554/media/1.jpg", "xx"],
        "ID": "123456789",
        "FaceUUID": "xxxx"
    ],...{}],
    "GroupId": "1"
}
```

Response Params (JSON format in body)

Name	Type	R/O	Description	Example
Response Example				
{}				

9.3 People Counting

9.3.1 Get Summary

Get summary information of people statistics in video.

Request URL	http://<server>/cgi-bin/videoStatServer.cgi?action=getSummary			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
channel	int	O	Video channel index which starts from 1, default is 1.	1
Request Example				
http://192.168.1.108/cgi-bin/videoStatServer.cgi?action=getSummary&channel=1				

Response Params (key=value format in body)

Name	Type	R/O	Description	Example

summary	object	R	Summary information of people statistics in video .	
+Channel	int	R	Video channel index which starts from 0.	1
+RuleName	string	R	<p>Rule type, it can be:</p> <ul style="list-style-type: none"> • "ManNumDetection": count the people num in region, and the detail stat is in "InsideSubtotal" param. • "NumberStat": count the people entering and leaving the region, and the detail stat is in "EnteredSubtotal" and "ExitedSubtotal"param. 	"NumberStat"
+EnteredSubtotal	object	O	People enter stat.	
++Total	int	R	Total enter num.	14
++Today	int	R	Today enter num.	0
++Hour	int	R	This hour enter num.	0
++TotalInTimeSection	int	O	Today enter num after call clearSectionStat.	0
+ExitedSubtotal	object	O	People leave stat.	
++Total	int	R	Total leave num.	32
++Today	int	R	Today leave num.	0
++Hour	int	R	This hour leave num.	0
++TotalInTimeSection	int	O	Today leave num after call clearSectionStat.	0
+PassedSubtotal	object	O	People pass stat.	
++Total	int	R	Total pass num.	32
++Today	int	R	Today pass num.	0
++Hour	int	R	This hour pass num.	0
++TotalInTimeSection	int	O	Today pass num after call clearSectionStat	0
+EnteredDupSubtotal	object	O	People duplicate enter stat.	
++Total	int	R	Total duplicate enter num.	14
++Today	int	R	Today duplicate enter num.	0
++Hour	int	R	This hour duplicate enter num.	0
++TotalInTimeSection	int	O	Today duplicate enter num after call clearSectionStat	0
+ExitedDupSubtotal	object	O	People duplicate leave stat.	
++Total	int	R	Total duplicate leave num.	32
++Today	int	R	Today duplicate leave num.	0
++Hour	int	R	This hour duplicate leave num.	0
++TotalInTimeSection	int	O	Today duplicate leave num after call clearSectionStat	0
+InsideSubtotal	object	O	People inside region stat.	

I				
++Total	int	R	Today inside num.	65
++ManStayStat	object[]	O	The entering and leaving stat of people that leaved.	
+++EnterTime	string	O	People enter time.	2012-01-04 00:00:00
+++ExitTime	string	O	People leave time.	2012-01-04 00:00:45

Response Example

```

summary.Channel=0
summary.RuleName=NumberStat
summary.EnteredSubtotal.Today=0
summary.EnteredSubtotal.Total=14
summary.EnteredSubtotal.TotalInTimeSection=0
summary.ExitedSubtotal.Today=0
summary.ExitedSubtotal.Total=32
summary.ExitedSubtotal.TotalInTimeSection=0
summary.PassedSubtotal.Hour=0
summary.PassedSubtotal.Today=0
summary.PassedSubtotal.Total=142
summary.PassedSubtotal.TotalInTimeSection=0
summary.EnteredDupSubtotal.Hour=0
summary.EnteredDupSubtotal.Today=0
summary.EnteredDupSubtotal.Total=0
summary.EnteredDupSubtotal.TotalInTimeSection=0
summary.EnteredSubtotal.TotalInTimeSection=17
summary.ExitedDupSubtotal.Hour=0
summary.ExitedDupSubtotal.Today=0
summary.ExitedDupSubtotal.Total=0
summary.ExitedDupSubtotal.TotalInTimeSection=0
summary.InsideSubtotal.Total=65
summary.InsideSubtotal.ManStayStat[0].EnterTime=2012-01-04 00:00:00
summary.InsideSubtotal.ManStayStat[0].ExitTime=2012-01-04 00:00:45
summary.InsideSubtotal.ManStayStat[1].EnterTime=2012-01-04 00:00:00
summary.InsideSubtotal.ManStayStat[1].ExitTime=2012-01-04 00:00:45

```

9.3.2 Query the Count of People

- Start to find

Start to find video stat info, in response, there is a token for further info finding process, and there is a totalCount shows how many data count(s).

Request URL	http://<server>/cgi-bin/videoStatServer.cgi?action=startFind		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
channel	int	O	Video channel index which starts from 1,
			1

			default is 1.	
condition	object	R	Find condition.	
+StartTime	string	R	Find time range start	2012-01-04 00:00:00
+EndTime	string	R	Find time range end	2012-01-04 00:02:00
+Granularity	string	R	The information granularity returned by the query requirements. The range is { Hour, Day, Week, Month, Season, Year } (Note: Most devices only support Hour, Day, Week.)	Hour
+RuleType	string	O	Rule type, it can be: NumberStat , ManNumDetection . If omit, default is NumberStat	NumberStat
+MinStayTime	int	O	Valid when ruleType is ManNumDetection , report people stay over this minimal time.	20
+PlanID	int	O	The plan id, only valid for dome camera.	2
+PtzPresetId	int	O	The ptz preset index which starts from 1, only valid for dome camera. Note: new device should use PtzPresetId instead of PlanID.	1
+AreaID	Array<int>	O	The area id which starts from 1, max array size is 20.	[1,2]
+OtherRule	string	O	The Another search rule, can be : AverageStayTime AverageStayTime	AverageStayTime

Request Example

http://192.168.1.108/cgi-bin/videoStatServer.cgi?action=startFind&channel=1&condition.StartTime=2011-01-01%2012:00:00&condition.EndTime=2011-01-10%2012:00:00&condition.Granularity=Hour&condition.RuleType=NumberStat&condition.MinStayTime=20&condition.AreaID[0]=2&condition.AreaID[1]=3

Response Params (key=value format in body)

Name	Type	R/O	Description	Example
token	int	R	Token for this search, use this token to get result and stop search.	12345
totalCount	int	R	Number of find result.	56

Response Example

token=12345
totalCount=56

- Get the find result

Get the find result of Video Stat info with channel, token, begin Number and count.

Request URL	http://<server>/cgi-bin/videoStatServer.cgi?action=doFind			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
channel	int	O	Video channel index which starts from 1, default is 1. NOTE: must be the same as startFind.	1

token	int	R	Token for this search, use this token to get result and stop searching.	12345
beginNumber	int	R	The start count. It must be between 0 and totalCount -1.	0
count	int	R	The count of info for this query.	20

Request Example

`http://192.168.1.108/cgi-bin/videoStatServer.cgi?action=doFind&channel=1&token=12345&beginNumber=0&count=20`

Response Params (key=value format in body)				
Name	Type	R/O	Description	Example
found	int	R	Number of result that return.	20
info	array<object>	R	Result information of video Stat.	
+Channel	int	R	Video channel index which starts from 0.	0
+PlanID	int	O	The plan id, only valid for dome camera.	2
+PtzPresetId	int	O	The ptz preset index which starts from 1, only valid for dome camera.	1
+AreaID	int	O	The area id which starts from 1.	3
+AverageStayTime	int	O	The average stay time, only valid when startFind with OtherRule param's value is "AverageStayTime".	20
+RuleName	string	R	Rule name, it can be: <ul style="list-style-type: none"> • "ManNumDetection": count people num in region, detail stat is in "InsideSubtotal"param • "NumberStat": count people enter and leave region, detail stat is in "EnteredSubtotal"and "ExitedSubtotal"param. 	NumberStat
+StartTime	string	O	Find time range start	2012-01-04 00:00:00
+EndTime	string	O	Find time range end	2012-01-04 00:02:00
+EnteredSubtotal	int	O	Total enter num.	14
+ExitedSubtotal	int	O	Total leave num.	5
+InsideSubtotal	int	O	Total inside num.	65
+PassedSubtotal	int	O	Total pass num.	14
+EnteredDupSubtotal	int	O	Total duplicate entered num.	14
+ExitedDupSubtotal	int	O	Total duplicate leave num.	5

Response Example

```
found=20
info[0].Channel=0
info[0].AreaID=2
info[0].RuleName=NumberStat
```

```

info[0].StartTime=2012-03-14 00:00:00
info[0].EndTime=2012-04-14 00:00:00
info[0].EnteredSubtotal=14
info[0].ExitedSubtotal=5
info[0].InsideSubtotal=65
info[0].PassedSubtotal=0
info[0].EnteredDupSubtotal=0
info[0].ExitedDupSubtotal=0
info[1].Channel=0
info[1].AreaID=3
info[1].RuleName=NumberStat
info[1].StartTime=2012-03-14 00:00:00
info[1].EndTime=2012-04-14 00:00:00
info[1].EnteredSubtotal=14
info[1].ExitedSubtotal=5
info[1].InsideSubtotal=65
info[1].PassedSubtotal=0
info[1].EnteredDupSubtotal=0
info[1].ExitedDupSubtotal=0
...

```

- Stop the searching session

Stop query video stat by channel and token.

Request URL	http://<server>/cgi-bin/videoStatServer.cgi?action=stopFind			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
channel	int	O	Video channel index which starts from 1, default is 1. NOTE: must be the same as startFind.	1
token	int	R	Token for this search, use this token to get result and stop searching.	12345
Request Example				GET http://192.168.1.108/cgi-bin/videoStatServer.cgi?action=stopFind&channel=1&token=12345

Response Params (OK in body)
Response Example
OK

9.3.3 Clear the People Count Information

Clear the people count information.

Request URL	http://<server>/cgi-bin/videoStatServer.cgi?action=clearSectionStat			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example

channel	int	O	Video channel index which starts from 1, default is 1.	1
---------	-----	---	--	---

Request Example

GET http://192.168.1.108/cgi-bin/videoStatServer.cgi?action=clearSectionStat&channel=1

Response Params (OK in body)

Response Example

OK

9.3.4 Subscribe the People Count Information

Subscribe the people count information.

Request URL	http://<server>/cgi-bin/videoStatServer.cgi?action=attach			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
channel	int	O	Video channel index which starts from 1, default is 1.	1
heartbeat	int	O	Send heartbeat interval, range is [1, 60], unit is second. If the URL contains this parameter, and the value is 5, it means every 5 seconds the device should send the heartbeat message to the client, the heartbeat message is an string "Heartbeat". If this parameter is not present, its default value is 60.	5

Request Example

http://192.168.1.108/cgi-bin/videoStatServer.cgi?action=attach&channel=1&heartbeat=5

Response Params (key=value format in body)

Name	Type	R/O	Description	Example
summary	object	R	Summary information of video Stat.	
+Channel	int	R	Video channel index which starts from 0.	0
+RuleName	string	R	Rule type, it can be: <ul style="list-style-type: none"> • "ManNumDetection": Count people num in region, detail stat is in "InsideSubtotal" param. • "NumberStat": Count people entering and leaving region, detail stat is in "EnteredSubtotal" and "ExitedSubtotal" param. 	NumberStat
+EnteredSubtotal	object	O	People enter stat.	
++Total	int	R	Total enter num.	14
++Today	int	R	Today enter num.	2
++Hour	int	R	This hour enter num.	1
++TotalInTime	int	O	Today enter num after call	0

Section			clearSectionStat.		
+ExitedSubtotal	object	O	People leave stat.		
++Total	int	R	Total leave num.	32	
++Today	int	R	Today leave num.	3	
++Hour	int	R	This hour leave num.	2	
++TotalInTimeSection	int	O	Today leave num after call clearSectionStat.	0	
+PassedSubtotal	object	O	People pass stat.		
++Total	int	R	Total pass num.	32	
++Today	int	R	Today pass num.	0	
++Hour	int	R	This hour pass num.	0	
++TotalInTimeSection	int	O	Today pass num after call clearSectionStat	0	
+EnteredDupSubtotal	object	O	People duplicate enter stat.		
++Total	int	R	Total duplicate enter num.	14	
++Today	int	R	Today duplicate enter num.	0	
++Hour	int	R	This hour duplicate enter num.	0	
++TotalInTimeSection	int	O	Today duplicate enter num after call clearSectionStat	0	
+ExitedDupSubtotal	object	O	People duplicate leave stat.		
++Total	int	R	Total duplicate leave num.	32	
++Today	int	R	Today duplicate leave num.	0	
++Hour	int	R	This hour duplicate leave num.	0	
++TotalInTimeSection	int	O	Today duplicate leave num after call clearSectionStat	0	
+InsideSubtotal	object	O	People inside region stat.		
++Total	int	R	Today inside num.	65	
++ManStayStat	array<object>	O	The entering and leaving stat of people that leaved.		
+++EnterTime	string	O	People enter time	2012-01-04 00:00:00	
+++ExitTime	string	O	People leave time	2012-01-04 00:00:45	

Response Example

HTTP/1.1 200 OK

Server: Device/1.0

Content-Type: multipart/x-mixed-replace; boundary=<boundary>

Connection: closed

--<boundary>

Content-Type: text/plain

Content-Length: <length>

summary.Channel=0

```

summary.RuleName=NumberStat
summary.EnteredSubtotal.Today=2
summary.EnteredSubtotal.Total=14
summary.EnteredSubtotal.TotalInTimeSection=1
summary.ExitedSubtotal.Today=3
summary.ExitedSubtotal.Total=32
summary.ExitedSubtotal.TotalInTimeSection=2
summary.PassedSubtotal.Hour=0
summary.PassedSubtotal.Today=0
summary.PassedSubtotal.Total=142
summary.PassedSubtotal.TotalInTimeSection=0
summary.EnteredDupSubtotal.Hour=0
summary.EnteredDupSubtotal.Today=0
summary.EnteredDupSubtotal.Total=0
summary.EnteredDupSubtotal.TotalInTimeSection=0
summary.EnteredSubtotal.TotalInTimeSection=17
summary.ExitedDupSubtotal.Hour=0
summary.ExitedDupSubtotal.Today=0
summary.ExitedDupSubtotal.Total=0
summary.ExitedDupSubtotal.TotalInTimeSection=0
summary.InsideSubtotal.Total=65
summary.InsideSubtotal.ManStayStat[0].EnterTime=2012-01-04 00:00:00
summary.InsideSubtotal.ManStayStat[0].ExitTime=2012-01-04 00:00:45
summary.InsideSubtotal.ManStayStat[1].EnterTime=2012-01-04 00:00:00
summary.InsideSubtotal.ManStayStat[1].ExitTime=2012-01-04 00:00:45
--<boundary>
Content-Type: text/plain
Content-Length: 11

```

Heartbeat
--<boundary>
Content-Type: text/plain
Content-Length: <length>

```

summary.Channel=0
summary.RuleName=NumberStat
summary.EnteredSubtotal.Today=2
summary.EnteredSubtotal.Total=14
...

```

9.3.5 Clear statistics in time section

Request URL	http://<server>/cgi-bin/videoStatServer.cgi?action=clearSectionStat		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
channel	int	O	Video channel index which starts from 1,
			1

			default is 1.	
AreaID	int	O	The area index which starts from 1, valid for multiple area device, if omit means clear all area statistics.	2
PtzPresetId	int	O	The ptz preset index which starts from 1.	1

Request Example

```
http://192.168.1.108/cgi-bin/videoStatServer.cgi?action=clearSectionStat&AreaID=2
```

Response Params (OK in body)

Response Example

```
OK
```

9.3.6 [Config] Video Widget Number Status

People Counting Overlay OSD Configuration Parameters

Config Data Params				
Name	Type	R/O	Description	Example
VideoWidgetNumberStat	object[]	R	Array, one element for each channel, and the array index is the channel number, starting from 0.	
+EncodeBlend	bool	R	Whether to overlay it to the main stream video encoding	true
+ShowEnterNum	bool	R	Whether to display the number of people entering	true
+ShowExitNum	bool	R	Whether to display the number of people leaving	true
+TextAlign	int	R	Text alignment mode: 0: Left alignment 2: Right alignment	0

Please refer to "4.2.1 Obtaining and Setting Configuration" to obtain and set the configuration. The specific examples are as follows:

Get Config Request Example

```
http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=VideoWidgetNumberStat
```

Get Config Response Example

```
table.VideoWidgetNumberStat[0].EncodeBlend=true
table.VideoWidgetNumberStat[0].ShowEnterNum=true
table.VideoWidgetNumberStat[0].ShowExitNum=true
table.VideoWidgetNumberStat[0].TextAlign=0
...
```

Set Config Request Example

```
http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&VideoWidgetNumberStat[0].EncodeBlend=true&VideoWidgetNumberStat[0].ShowEnterNum=true&VideoWidgetNumberStat[0].ShowExitNum=true&VideoWidgetNumberStat[0].TextAlign=0
```

Set Config Response Example

OK

9.3.7 [Event] NumberStat

When people number triggers the rule, send this event.

Event Code	NumberStat			
Event action	Start/Stop			
Event index	0			
Event Data				
Name	Type	R/O	Description	Example
+Number	int	R	Total number in detect region.	
+EnteredNumber	int	R	Total number that enter the detect region.	180
+ExitedNumber	int	R	Total number that leave the detect region.	60
+PassedNumber	int	O	Total pass number in detect region.	60
+EnteredDupNumber	int	O	Total duplicate number that enter the detect region.	180
+ExitedDupNumber	int	O	Total duplicate number that leave the detect region.	60
+Type	char[]	R	Number overrun type, it can be: "EnterOver", "ExitOver", "InsideOver", "PassOver".	EnterOver
+AreaID	int	O	The area id, begin from 1, if omit, means single area.	2
+PresetID	int	O	The preset id, if omit, means preset is unknown.	1

Event Response Example (multipart , JSON format in body) (response to 4.9.17 event subscription)

--<boundary>
Content-Type: text/plain
Content-Length: <length>

Code=NumberStat;action=Start;index=0;data={
 "Number": 120,
 "EnteredNumber": 180,
 "ExitedNumber": 60,
 "PassedNumber": 60,
 "EnteredDupNumber": 180,
 "ExitedDupNumber": 60,
 "Type": "EnterOver",
 "Area": 2
}

9.3.8 [Event] ManNumDetection

When people number triggers the rule, send this event.

Event Code	ManNumDetection			
Event action	Start/Stop			
Event index	0			
Event Data				
Name	Type	R/O	Description	Example
+ManList	object[]	R	The people info list.	
++BoundingBox	int[4]	R	The detected people bounding box, 4 interge, refer to x's value of left-top point, y's value of left-top point, x's value of right-bottom point, y's value of right-bottom point. Coordinate remap to 0 — 8192.	[2992,1136,4960,5192]
++Stature	int	R	The people's stature, unit is cm.	170
+AreaID	int	O	The area id, begin from 1, if omit, means single area.	2
+PresetID	int	O	The preset id, if omit, means preset is unknown.	1

Event Response Example (multipart , JSON format in body) (response to 4.9.17 event subscription)

--<boundary>

Content-Type: text/plain

Content-Length: <length>

```
Code=ManNumDetection;action=pulse;index=0;data={  
    "ManList": [ { "BoundingBox": [2992,1136,4960,5192], "Stature": 170 },  
                { "BoundingBox": [4392,4136,6960,6512], "Stature": 175 },  
                {...}, ... ],  
    "AreaID" : 2  
}
```

--<boundary>

9.3.9 [Event] CrowdDetection

When crowd density overrun, send this event.

Event Code	CrowdDetection			
Event action	Start/Stop			
Event index	0			
Event Data				
Name	Type	R/O	Description	Example
+CrowdList	object[]	O	The global crowd density overrun list.	

++Center	int[]	R	The center point, must be two int, means x and y value, coordinate remap to 0 — 8192.	[5734,2377]
++Radius	int	R	The radius length.	10
+RegionList	object[]	O	The people num overrun region list.	
++RegionID	int	R	The region index.	0
++PeopleNum	int	R	The people count in region.	100

Event Response Example (multipart , JSON format in body) (response to 4.9.17 event subscription)

--<boundary>

Content-Type: text/plain

Content-Length: <length>

```
Code=CrowdDetection;action=start;index=0;data={  
    "CrowdList": [ { "Center" : [5734,2377], "Radius" : 10}, ...{} ],  
    "RegionList" : [ { "RegionID" : 0, "PeopleNum" : 100 },...{} ]  
}  
--<boundary>
```

9.3.10 [Event] LeaveDetection

Event Code	LeaveDetection			
Event Action	Pulse			
Event Index				
Event Data				
Name	Type	R/O	Description	Example
+Name	char[128]	O	event name	"LeaveDetection"
+DetectRegion	uint16[20][2]	O	detect region	
+Count	int	O	alarm count	100
+TriggerMode	enumchar[16]	O	trigger mode range { "NoPerson" "Leave" "Static" "OutPersonLimit" "NoMoving" "OnTable" }	"NoPerson"

Event Response Example (multipart , JSON format in body) (response to 4.9.17 event subscription)

--<boundary>

Content-Type: text/plain

Content-Length: <length>

```
Code=LeaveDetection;action=Pulse;index=0;data={  
    "Name": "LeaveDetection",
```

```

    "DetectRegion": [],
    "Count": 100,
    "TriggerMode": "NoPerson",
    ...
}

--<boundary>

```

9.4 Heat Map

9.4.1 Get Heat Map Information

Get heat map statistics by time

Request URL	http://<server>/cgi-bin/heatMap.cgi?action=getPicByTime		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
channel	int	R	Video channel number, starting from 1,
StartTime	char[32]	R	Heat Map Start time, 24 h: yyyy-MM-dd HH:mm:ss.
EndTime	char[32]	R	Heat Map End time, 24 h: yyyy-MM-dd HH:mm:ss.
PtzPresetId	int	O	Preset, starting from 1
Request Example			
http://192.168.1.108/cgi-bin/heatMap.cgi?action=getPicByTime&channel=1&StartTime=2015-08-20%2000:00:00&EndTime=2015-08-21%2023:59:59&PtzPresetId=1			

Response Params (binary data in body)
Response Example
HTTP/1.1 200 OK
Server: Device/1.0
Content-Type: application/octet-stream
Content-Length: <length>
<heatmap data>

Appendix: Format of the heat map

0	1	2	3	4	...	15	16	17	18	...
Width	Height	Reserved					Data: every byte symbolize a pixel			

9.4.2 Get People Heat Map Information

Request URL	http://<server>/cgi-bin/videoStatServer.cgi?action=getHeatMap		
Method	GET		
Request Params (key=value format in URL)			

Name	Type	R/O	Description	Example
channel	int	O	Video channel index which starts from 1, default is 1.	1
PlanID	int	O	The plan id, only valid for dome camera.	2
StartTime	string	R	The start time, ex: "2010-05-12 20:00:00"	"2010-05-12 20:00:00"
EndTime	string	R	The end time, ex: "2010-05-12 22:00:00"	"2010-05-12 22:00:00"
HeatMapType	string	R	The heat map type, it can be: "AverageStayTime": average stay time heat map "HumanStatistics": human statistics heat map "HumanTrack" : human track heat map	AverageStayTime

Request Example

http://192.168.1.108/cgi-bin/videoStatServer.cgi?action=getHeatMap&channel=1&PlanID=2&StartTime=2010-05-12%2020:00:00&EndTime=2010-05-12%2022:00:00&HeatMapType=AverageStayTime

Response Params (binary data in body)

Name	Type	R/O	Description	Example
Response Example				
HTTP/1.1 200 OK				
Server: Device/1.0				
Content-Type: application/octet-stream				
Content-Length: <length>				
<heat map binary data>				

Appendix A: The heat map binary data format when HeatMapType is AverageStayTime.

octet	0	1	2	3	4~7	8~11	12~15
value	version	channel	present	reserved	total data length	total lines	lines in this packet
octet	16~31						
value	reserved						
octet	32~35				36~39	40~43	44~44+4* (N-1)
value	line number				data num in this line	data 1	data N
octet	44+4* (N-1) +1~44+4* (N-1) +4			
value	next line number				data num in next line	data 1	data N

Note: If "lines in this packet" is 0, then all data has been sent, and connection will be closed.

Appendix B: The heat map binary data format when HeatMapType is HumanStatistics.

octet	0	1	2	3	4~7	8~11	12~15
value	vers	cha	pres	reserv	total data length	total lines	lines in this packet

	ion	nnel	et	ed			
octet	16~31						
value	reserved						
octet	32~35			36~39		40~43	44~44+4* (N-1)
value	line number			data num in this line		data 1	data N
octet	44+4* (N-1) +1~ 44+4* (N-1) +4		
value	next line number			data num in next line		data 1	data N

Note: If "lines in this packet" is 0, then all data has been sent, and connection will be closed.

Appendix C: The heat map binary data format when HeatMapType is HumanTrack.

octet	0	1	2	3	4~7	8~11	12~15
value	version	channel	present	reserved	total data length	total tracks	tracks in this packet
octet	16~31						
value	reserved						
octet	32~35			36~39		40~43	44~44+4* (N-1)
value	Object ID			track point num in this object		track point 1	track point N
octet	44+4* (N-1) +1~ 44+4* (N-1) +4		
value	next object ID			track point num in next object		track point 1	track point N

- Note:** If "tracks in this packet" is 0, then all data has been sent, and connection will be closed.

9.4.3 Subscribe People Realtime Trace Information

Request URL	http://<server>/cgi-bin/videoStatServer.cgi?action=attachRealTraceProc				
Method	GET				
Request Params (key=value format in URL)					
Name	Type	R/O	Description	Example	
channel	int	O	Video channel index which starts from 1, default is 1.	1	
heartbeat	int	O	Send heartbeat interval, range is [1, 60], unit is second. If the URL contains this parameter, and the value is 5, it means every 5 seconds the device should send the heartbeat message to the client, the heartbeat message are "Heartbeat". If this parameter is not present, its default value is 60.	5	
Request Example					

http://192.168.1.108/cgi-bin/videoStatServer.cgi?action=attachRealTraceProc&channel=1&heartbeat=5

Response Params (multipart; binary data in body)

Name	Type	R/O	Description	Example
Response Example				
			HTTP/1.1 200 OK	
			Server: Device/1.0	
			Content-Type: multipart/x-mixed-replace; boundary=<boundary>	
			Connection: closed	
			--<boundary>	
			Content-Type: application/octet-stream	
			Content-Length: <length>	
			<realtime trace binary data>	
			--<boundary>	
			Content-Type: text/plain	
			Content-Length: 11	
			Heartbeat	
			--<boundary>	
			Content-Type: application/octet-stream	
			Content-Length: <length>	
			<realtime trace binary data>	
			--<boundary>	
			...	

- **Appendix A:** The realtime trace binary data format.

octet	0	1	2	3	4~7	8~11	12~15
value	version	chan nel	pres et	reser ved	total data length	reserved	object num in this packet
octet					16~31		
value					reserved		
octet		32~35		36~37	38~39	40~43	44~47
value		object ID 1		object coord x,	object coord y	current time in utc	reserved
octet		48~51		52~53	54~55	56~57	58~61
value		object ID 2		object coord x	object coord y	current time in utc	reserved

9.4.4 Get People History Trace Information

Request URL	http://<server>/cgi-bin/videoStatServer.cgi?action=getHistoryTrace			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
channel	int	O	Video channel index which starts from 1, default is 1.	1
PlanID	int	O	The plan id, only valid for dome camera.	5
StartTime	string	R	The start time, ex: "2010-05-12 20:00:00"	"2010-05-12 20:00:00"
EndTime	string	R	The end time, ex: "2010-05-12 22:00:00"	"2010-05-12 22:00:00"
Request Example				
http://192.168.1.108/cgi-bin/videoStatServer.cgi?action=getHistoryTrace&channel=1&PlanID=2&StartTime=2010-05-12%2020:00:00&EndTime=2010-05-12%2022:00:00				

Response Params (multipart; binary data in body)				
Name	Type	R/O	Description	Example
Response Example				
HTTP/1.1 200 OK				
Server: Device/1.0				
Content-Type: multipart/x-mixed-replace; boundary=<boundary>				
Connection: closed				
--<boundary>				
Content-Type: application/octet-stream				
Content-Length: <length>				
<history trace binary data>				
--<boundary>				
Content-Type: application/octet-stream				
Content-Length: <length>				
<history trace binary data>				
--<boundary>				
...				

- **Appendix A:** The history trace binary data format.

octet	0	1	2	3	4~7	8~11	12~15
value	version	channel	present	reserved	total data length	total object num	object num in this packet
octet	16~31						
value	reserved						
octet	32~35			36~37	38~39	40~43	44~47

value	object ID 1	object coord x	object coord y	current time in utc	reserved
octet	48~51	52~53	54~55	56~57	58~61
value	object ID 2	object coord x	object coord y	current time in utc	reserved

Note: If "object num in this packet" is 0, then all data has been sent, and connection will be closed.

9.4.5 Subscribe Heat Map Raw Data

Request URL	http://<server>/cgi-bin/HeatMapManager.cgi?action=attachRaw						
Method	GET						
Request Params (key=value format in URL)							
Name	Type	R/O	Description	Example			
channel	int	O	Video channel index which starts from 1, default is 1.	1			
heartbeat	int	O	Send heartbeat interval, range is [1, 60], unit is second. If the URL contains this parameter, and the value is 5, it means every 5 seconds the device should send the heartbeat message to the client, the heartbeat message are "Heartbeat". If this parameter is not present, its default value is 60.	5			
Request Example							
http://192.168.1.108/cgi-bin/HeatMapManager.cgi?action=attachRaw&channel=1&heartbeat=5							

Response Params (multipart; binary data in body)					
Name	Type	R/O	Description	Example	
Response Example					
HTTP/1.1 200 OK					
Server: Device/1.0					
Content-Type: multipart/x-mixed-replace; boundary=<boundary>					
Connection: closed					
--<boundary>					
Content-Type: application/octet-stream					
Content-Length: <length>					
<heap map raw data>					
--<boundary>					
Content-Type: text/plain					
Content-Length: 11					
Heartbeat					
-<boundary>					
Content-Type: application/octet-stream					

Content-Length: <length>

```
<heap map raw data>
--<boundary>
...
```

- **Appendix A:** The heap map raw data format.

octet	0	1	2	3	4 ~ 23
value	width		height		StartTime, ex: "2012-01-04 00:00:00"
octet	24~43			44 ~ 47	
value	EndTime, ex: "2012-01-04 01:00:00"			reserved	
octet	48	49	...		
value	Data: every byte symbolize a pixel				

9.5 Crowd Distribute Map

9.5.1 Get Channel Caps

Request URL	http://<server>/cgi-bin/crowdDistrMap.cgi?action=getCaps		
Method	GET		
Request Params (none)			
Name	Type	R/O	Description
Request Example			
http://192.168.1.108/cgi-bin/crowdDistrMap.cgi?action=getCaps			

Response Params (key=value format in body)	
Name	Type
CrowdCapsList	object[]
+channel	uint
+Support	bool
Response Example	
CrowdCapsList[0].channel=1 CrowdCapsList[0].Support=true CrowdCapsList[1].channel=2 CrowdCapsList[1].Support=false ...	

9.5.2 Subscribe to Realtime Crowd Stat

Subscribe the crowd distribute map information, return info at regular time.

Request URL	http://<server>/cgi-bin/crowdDistrMap.cgi?action=attach
--------------------	---

Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
channel	int	O	video channel index to subscribe, starts from 1, default 1	1

Request Example

<http://192.168.1.108/cgi-bin/crowdDistrMap.cgi?action=attach&channel=1&heartbeat=5>

Response Params (multipart; binary data in body)				
Name	Type	R/O	Description	Example
CrowdStatData	object[]	R	Array, each element is a detection area, and each detection area can have multiple people counting areas	
+Channel	uint	R	channel	1
+GloabalPeopleNum	uint	R	global people count number	10
+RegionNum	uint	R	region number	1
+RegionPeopleList	object[]	R	list	
++RegionID	uint	R	region id	0
++RegionName	char[128]	R	region name	"xxx"
++Region	uint16[20][2]	R	region	[10,10,0,100]
++RegionPeopleNum	uint	R	people number	100
+CrowdEventNum	uint	O	crowd event number default 0	2
+CrowdList	object[]	O	crowd list	
++Center	Point	R	Central coordinate	[12,50]
++Radius	uint	R	radius	10
+RegionEventNum	uint	O	number, default 0	2
+RegionList	object[]	O	alarm region list	
++Region	uint16[20][2]	R	region	[10,10,0,100]
++RegionID	uint	R	region id	0
++PeopleNum	uint	R	Statistics of the number of people in the region	100

Response Example

```

HTTP/1.1 200 OK
Content-Type: multipart/x-mixed-replace; boundary=<boundary>
Connection: close

--<boundary>
Content-Type: text/plain
Content-Length: <data length>

CrowdStatData[0].Channel=1
CrowdStatData[0].GloabalPeopleNum =10
CrowdStatData[0].RegionNum =1
CrowdStatData[0].RegionPeopleList[0].RegionID=0
CrowdStatData[0].RegionPeopleList[0].Region[0][0]=10
CrowdStatData[0].RegionPeopleList[0].Region[0][1]=10
CrowdStatData[0].RegionPeopleList[0].Region[1][0]=10
CrowdStatData[0].RegionPeopleList[0].Region[1][1]=100
...
CrowdStatData[0].RegionPeopleList[0].RegionPeopleNum=100
CrowdStatData[0].CrowdEventNum =2
CrowdStatData[0].CrowdList[0].Center=2
CrowdStatData[0].CrowdList[0].Radius=2
CrowdStatData[0].RegionEventNum =2
CrowdStatData[0].RegionList[0].Region[0][0]=10
CrowdStatData[0].RegionList[0].Region[0][1]=10
CrowdStatData[0].RegionList[0].Region[1][0]=10
CrowdStatData[0].RegionList[0].Region[1][0]=100
...
CrowdStatData[0].RegionList[0].RegionID=0
CrowdStatData[0].RegionList[0].PeopleNum=100
--<boundary>
Content-Type: text/plain
Content-Length: 11

```

Heartbeat
--<boundary>
Content-Type: text/plain
Content-Length: <data length>

CrowdStatData[0].Channel=1

...

9.5.3 Get Current Crowd Stat

Get crowd distribuite map information, return info only once.

Request URL	http://<server>/cgi-bin/crowdDistrMap.cgi?action=getSummary
Method	GET

Request Params (none)				
Name	Type	R/O	Description	Example
channel	int	O	video channel index to subscribe, starts from 1, default 1	1

Request Example

http://192.168.1.108/cgi-bin/crowdDistrMap.cgi?action=getSummary&channel=1

Response Params (key=value format in body)				
Name	Type	R/O	Description	Example
CrowdStatData	object[]	R	Array, each element is a detection area, and each detection area can have multiple people counting areas	
+Channel	uint	R	channel	1
+GloabalPeopleNum	uint	R	global people count number	10
+RegionNum	uint	R	region number	1
+RegionPeopleList	object[]	R	list	
++RegionID	uint	R	region id	0
++RegionName	char[128]	R	region name	"xxx"
++Region	uint16[20][2]	R	region	[10,10,0,100]
++RegionPeopleNum	uint	R	people number	100
+CrowdEventNum	uint	O	crowd event number default 0	2
+CrowdList	object[]	O	crowd list	
++Center	Point	R	Central coordinate	[12,50]
++Radius	uint	R	radius	10
+RegionEventNum	uint	O	number, default 0	2
+RegionList	object[]	O	alarm region list	
++Region	uint16[20][2]	R	region	[10,10,0,100]
++RegionID	uint	R	region id	0
++PeopleNum	uint	R	Statistics of the number of people in the region	100

Response Example

```

CrowdStatData[0].Channel=1
CrowdStatData[0].GloabalPeopleNum =10
CrowdStatData[0].RegionNum =1
CrowdStatData[0].RegionPeopleList[0].RegionID=0
CrowdStatData[0].RegionPeopleList[0].Region[0][0]=10
CrowdStatData[0].RegionPeopleList[0].Region[0][1]=10
CrowdStatData[0].RegionPeopleList[0].Region[1][0]=10
CrowdStatData[0].RegionPeopleList[0].Region[1][1]=100

```

```

...
CrowdStatData[0].RegionPeopleList[0].RegionPeopleNum=100
CrowdStatData[0].CrowdEventNum =2
CrowdStatData[0].CrowdList[0].Center=2
CrowdStatData[0].CrowdList[0].Radius=2
CrowdStatData[0].RegionEventNum =2
CrowdStatData[0].RegionList[0].Region[0][0]=10
CrowdStatData[0].RegionList[0].Region[0][1]=10
CrowdStatData[0].RegionList[0].Region[1][0]=10
CrowdStatData[0].RegionList[0].Region[1][0]=100
...
CrowdStatData[0].RegionList[0].RegionID=0
CrowdStatData[0].RegionList[0].PeopleNum=100

```

9.6 Video Analyse

9.6.1 Get Video Analyse Capability

Request URL	http://<server>/cgi-bin/devVideoAnalyse.cgi?action=getcaps			
Method	GET			
Request Params (none)				
Name	Type	R/O	Description	Example
channel	int	R	video channel index which starts from 1.	1
Request Example				
http://192.168.1.108/cgi-bin/devVideoAnalyse.cgi?action=getcaps&channel=1				

Response Params (key=value format in body)				
Name	Type	R/O	Description	Example
caps	object	R	capabilities	
+SupportedScene	char[32][16]	O	scene list	["Normal", "ATM", "NumberStat"]
+SupportedScenes	object	O	Supported Scenes	
++FaceDetection	object	O	face detection	
+++SupportedCalibrateParams	object	O	Calibrate Params	
++++Groud	object	O	groud	
+++++HorizontalStaffs	uint16[2]	O	Range of values for the number of horizontal staffs the first is the minimum and the second is the maximum	[1, 2]
+++++VerticalStaffs	uint16[2]	O	Range of values for the number of vertical staffs	[3, 3]

			the first is the minimum and the second is the maximum	
++NumberStat	object	O	number stat	
+++CameraType	uint8	O	range {0,1,2}	1
+++SupportedRules	object	O	rules	
++StereoBehavior	object	O	Stereo Behavior	
+++OnlyFindLastEvtList	char[32][32]	O	event type list in which last ones are supported to be inquired.	["ParkingStatusChangeDetection"]
+MaxModules	uint8	O	max modules	5
+MaxRules	uint8	O	The maximum number of rules that can be saved for all preset points in each channel. default 32	32
+SupportedRules	char[128][128]	O	Union of rule lists supported by class	["CrossLineDetection", "PasteDetection", "FlowStat"]
+MaxCelibateAreas	int	O	Maximum number of calibration regions	10
+MaxStaffs	int	O	Maximum number of staffs	10
+MaxPointOfLine	int	O	Maximum number of lines	2
+MaxPointOfRegion	int	O	Maximum number of regions	4
+MaxExcludeRegions	int	O	Maximum number of exclude regions	10
+MaxInternalOptions	int	O	Maximum number of internal options	512
+ComplexSizeFilter	bool	O	Does support complex size filter	true
+CalibrateBoxes	uint8[2]	O		[2, 3]
+SpecifiedObjectFilter	bool	O	does support specified object filter	true

Response Example

```

caps.CalibrateBoxes[0]=2
caps.CalibrateBoxes[1]=3
caps.ComplexSizeFilter=false
caps.MaxCelibateAreas=10
caps.MaxExcludeRegions=0
caps.MaxInternalOptions=512
caps.MaxModules=1
caps.MaxPointOfLine=20
caps.MaxPointOfRegion=20
caps.MaxRules=10
caps.MaxStaffs=4

```

```

caps.SpecifiedObjectFilter=true
caps.SupportedRules[0]=CrossLineDetection
caps.SupportedRules[1]=CrossRegionDetection
caps.SupportedRules[2]=LeftDetection
caps.SupportedRules[3]=TakenAwayDetection
caps.SupportedScene[0]=Normal
caps.SupportedScene[1]=FaceDetection
caps.SupportedScene[2]=VideoDiagnosis
caps.SupportedScenes.FaceDetection.SupportedCalibrateParams.Groud.HorizontalStaffs[0]=0
caps.SupportedScenes.FaceDetection.SupportedCalibrateParams.Groud.HorizontalStaffs[1]=0
caps.SupportedScenes.FaceDetection.SupportedCalibrateParams.Groud.VerticalStaffs[0]=0
caps.SupportedScenes.FaceDetection.SupportedCalibrateParams.Groud.VerticalStaffs[1]=0
caps.SupportedScenes.StereoNumber.SupportedRules.ManNumDetection.SupportLocalDataStore=false
caps.SupportedScenes.NumberStat.CameraType=1
caps.SupportedScenes.NumberStat.SupportedRules.NumberStat.MaxRules=8
caps.SupportedScenes.StereoBehavior.OnlyFindLastEvtList[0]=ManStandDetection

```

9.6.2 [Config] Video Analyse Global

Config Data Params				
Name	Type	R/O	Description	Example
VideoAnalyse Global	object[]	O	array index starts from 0, which means video channel (equals to video channel index -1, and so 0 means channel 1).	
+Scene	object	O	scene	
++Type	char[]	O	Scene class, the range is { "", "Normal", "Indoor", "ATM", "Traffic", "FaceRecognition", "FaceDetection", "NumberStat", "HeatMap", "VideoDiagnosis", "VehicleAnalyse", "TrafficPatrol", "CourseRecord", "Vehicle" , "ObjectDetect", "VehicleCompare"}	"Normal"
++PtzPresetId	uint8	O	Range is 0—255, 0 means that the scene is unassociated with PTZ.	1
++Depth	enumchar[8]	O	Picture distance feature, the range is { "Normal", "Far", "Middle", "Near" }	"Normal"
++Detail	object	O	Detail config of a scene. For example, when Scene.Type is "Normal", it's detail includes CameraAngle, CameraDistance, CameraHeight, etc.	
+TimePeriod	object	O	period	
++Day	char[2][16]	O	The start and end time of Day, it's format is hh:mm:ss	["8:00:00", "20:00:00"]
++Night	char[2][16]	O	The start and end time of Night, it's	["20:00:00", "7:00:00"]

Please refer to "4.2.1 Obtaining and Setting Configuration" to obtain and set the configuration. The specific examples are as follows:

Get Config Request Example

```
http://10.0.0.8/cgi-bin/configManager.cgi?action=getConfig&name=VideoAnalyseGlobal
```

Get Config Response Example

```
table.VideoAnalyseGlobal[0].Scene.Type=Normal
table.VideoAnalyseGlobal[0].Scene.PtzPresetId=1
table.VideoAnalyseGlobal[0].Scene.Depth=Far
table.VideoAnalyseGlobal[0].Scene.Detail.CameraAngle=30
table.VideoAnalyseGlobal[0].Scene.Detail.CameraDistance=10.000000
table.VideoAnalyseGlobal[0].Scene.Detail.CameraHeight=6.200000
table.VideoAnalyseGlobal[0].TimePeriod.Day[0]=8:00:00
table.VideoAnalyseGlobal[0].TimePeriod.Day[1]=20:00:00
table.VideoAnalyseGlobal[0].TimePeriod.Night[0]=20:00:00
table.VideoAnalyseGlobal[0].TimePeriod.Night[1]=8:00:00
...
...
```

Set Config Request Example

```
http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&VideoAnalyseGlobal[0].Scene.Type=
Normal&VideoAnalyseGlobal[0].Scene.PtzPresetId=1
```

Set Config Response Example

```
OK
```

9.6.3 [Config] Video Analyse Rule

Config Data Params				
Name	Type	R/O	Description	Example
VideoAnalyseRule	object[<i>ChannelNo</i>][<i>RuleNo</i>]	O	<p>ChannelNo: integer, array index starts from 0, which means video channel (equals to video channel index -1, and so 0 means channel 1).</p> <p>RuleNo =rule index.</p> <p>ParamName start with head.Config is only effective with {"CrossLineDetection", "CrossRegionDetection", "LeftDetection", "TakenAwayDetection"}.</p>	
+Name	char[128]	O	Rule name, it must be unique.	"GuardLine1"
+Enable	bool	O	Enable/Disable this rule.	true
+Type	char[32]	R	The range is {"CrossLineDetection", "CrossRegionDetection", "LeftDetection", "TakenAwayDetection", "VideoAbnormalDetection", "FaceDetection", "AudioMutation", "AudioAnomaly", "VideoUnFocus",}	"CrossLineDetection" "

			"WanderDetection", "RioterDetection", "ParkingDetection", "MoveDetection", "NumberStat", "HeatMap", "FaceAttribute", "VehicleDetect", "NonMotorDetect", "HumanTrait", "ObjectPlacement", "ObjectRemoval" }.	
+EventHandle	EventHandler	O	Setting of EventHandler is described in SetEventHandler .	
+Config	object	O	configuration Depending on the specific rule type, The following parameters are only valid when Type is the following value: {"CrossLineDetection", "CrossRegionDetection", "LeftDetection", "TakenAwayDetection"}.	
+DetectLine	uint16[20][2]	O	The line point The coordinates of the point is unified within the range of 0–8192.	
+Direction	enumchar[16]	O	enumchar[16]{ "LeftToRight" "RightToLeft" "Both" }	"LeftToRight"
+SizeFilter	object	O	size filter	
++MaxSize	int[2]	O	Maximum width and Maximum height. The width and the height of the object must not be beyond maximum value. Adapt to: {"CrossLineDetection", "CrossRegionDetection", "LeftDetection", "TakenAwayDetection", "FaceDetection", "WanderDetection", "RioterDetection", "ParkingDetection", "MoveDetection"}.	
++MinSize	int[2]	O	Minimum width and Minimum height. The width and the height of the object must not be less than minimum value.	
+DetectRegion	uint16[20][2]	O	The coordinates of the point is unified within the range of 0–8192. Adapt to : {"CrossRegionDetection", "LeftDetection", "TakenAwayDetection", "WanderDetection", "RioterDetection", "ParkingDetection", "MoveDetection"}.	
+MinDuration	uint	O	Range is 1—600, adapt to {"LeftDetection", "TakenAwayDetection", "WanderDetection"}. Range is 10-300, adapt to {"RioterDetection"}.	0

			Range is 6-300, adapt to {"ParkingDetection"}.	
+Sensitivity	uint8	O	Range is 1—10, adapt to {"RioterDetection", "MoveDetection"}.	6
+EnterThresh old	uint16	O	Range is 0—100000000, adapt to {"NumberStat"}.	0
+ExitThreshol d	uint16	O	Range is 0—100000000, adapt to {"NumberStat"}.	0
+InsideThresh old	uint16	O	Range is 0—100000000, adapt to {"NumberStat"}.	0
+DirectionStat s	object	O		方向统计配置
++Enable	bool	O	Enable/Disable object detect Direction. Rule range is {"VehicleDetect", "NonMotorDetect", "HumanTrait"}.	false
++DetectLine	uint16[20][2]	O	The line point The coordinates of the point is unified within the range of 0—8192.	
++Direction	enumchar[16]	O	enumchar[16]{ "Both" "LeftToRight" "RightToLeft" }	"Both"
+FilterUnAlive Enable	bool	O	Enable the non-living filtering function. It is false by default.	false
+EyesDistThreshold	uint32	O	Set filtering threshold of pupillary distance. If the actual pupillary distance is less than the threshold, it will be filtered. Refer to GB/T 35678-2017: Value range [50—no upper limit] required for face recognition application image technology. The default value 0 indicates no filtering.	50
+HelmetEnable	bool	O	Enable safety helmet detection. With the function enabled, the face recognition result carries information related to safety helmet detection. It is false by default.	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: Automatical mode 1: Thermal image detection mode 2: Calibration mode	0
+TempStrateg y	object	O	strategy	

++TempValueMax	double	O	Maximum value of normal face temperature range	45.1
++TempValueMin	double	O	Minimum value of normal face temperature range	35.5
++TempType	uint8	O	Strategy type for calculating face temperature: 0: Use the highest temperature 1: Use the average temperature	0
++SlideNum	uint8	O	The number of sliding cache frames. 0 means no sliding, and the maximum cache is 32 frames.	0
++HighTempStrategy	object	O	valid whenTempType=0	
+++TempDete ctRegion	uint8	O	0: Forehead part 1: Whole face	0
++AverageTe mpStrategy	object	O	valid whenTempType=1	
+++StrategyTy pe	uint8	O	Parameter for average temperature strategy (valid when tempType is 1). Average temperature strategy type: 0: Center point 1: high temperature point	0
+++PointNum	uint8	O	Parameter for average temperature strategy (valid when tempType is 1). The number of points used to find the average value, and it is the square of the integer, such as 4, 9, and 16.	9
+++EnableFilt er	bool	O	Parameter for average temperature strategy (valid when tempType is 1). Whether to enable the removal of the highest temperature and the lowest temperature to take the average value: False: Disable True: Enable	false
+FilterMaskUn AliveEnable	bool	O	Select whether to enable liveness detection for person wearing mask. This value is related to FilterUnAliveEnable. When FilterUnAliveEnable is true (liveness detection is enabled), this field is valid.	false

Please refer to "4.2.1 Obtaining and Setting Configuration" to obtain and set the configuration. The specific examples are as follows:

Get Config Request Example

```
http://10.0.0.8/cgi-bin/configManager.cgi?action=getConfig&name=VideoAnalyseRule
```

Get Config Response Example

table.VideoAnalyseRule[0][0].Name= line1
table.VideoAnalyseRule[0][0].Type=CrossLineDetection

table.VideoAnalyseRule[0][0].VideoAnalyseRule[0][0].Enable =true table.VideoAnalyseRule[0][0].VideoAnalyseRule[0][0].EventHandler= (output of EventHandler is described in GetEventHandler) ...

Set Config Request Example

http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&VideoAnalyseRule[0][0].Name=myAnalyseRule1&VideoAnalyseRule[0][0].Type=CrossLineDetection

Set Config Response Example

OK

9.6.4 Get Last Event Info

Get Last Event Info

Request URL	http://<server>/cgi-bin/devVideoAnalyse.cgi?action=getLastEventInfo		
Method	GET		
Request Params (none)			
Name	Type	R/O	Description
channel	int	R	Starts from 0, which means video channel (equals to video channel index -1, and so 0 means channel 1).
ClassName	char[]	R	means Class name.
EventName	char[]	R	means Event name.
Request Example			
http://192.168.1.108/cgi-bin/devVideoAnalyse.cgi?action=getLastEventInfo&channel=1&ClassName=StereoBehavior&EventName=ManStandDetection			

Response Params (key=value format in body)

Name	Type	R/O	Description	Example
State	int	R	enumint{ 0: off 1: on }	1
EventInfo	object	R	event info	
Code	char[32]	R	event code	"BusDoor"

Response Example

State=1 EventInfo.Code=ManStandDetection EventInfo.Action=Start

9.6.5 [Config] GlobalDeviceParam

Config Data Params				
Name	Type	R/O	Description	Example
GlobalDeviceParam	object	O	Global Device Param	
+LocateHeight	float	O	Device locate Height	10.5
+Profile	uint32	O	range:{ 0-outdoor 1-indoor } default 0	0

Please refer to "4.2.1 Obtaining and Setting Configuration" to obtain and set the configuration. The specific examples are as follows:

Get Config Request Example

```
http://10.0.0.8/cgi-bin/configManager.cgi?action=getConfig&name=GlobalDeviceParam
```

Get Config Response Example

```
table.GlobalDeviceParam.LocateHeight=10.5
```

```
table.GlobalDeviceParam.Profile=0
```

Set Config Request Example

```
http://10.0.0.8/cgi-bin/configManager.cgi?action=setConfig&GlobalDeviceParam.LocateHeight=10.5
```

Set Config Response Example

```
OK
```

9.6.6 Get Template Rule

Gets all intelligent rule configuration templates and default values under the specified category

Request URL	http://<server>/cgi-bin/VideoInAnalyse.cgi?action=getTemplateRule			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
Class	Char[32]	R	the same meaning with Type in config VideoAnalyseGlobal "Normal" "Traffic" "TrafficPatrol" "FaceDetection"(shared by face detect and face recognition) in added, Web can use "ALL" for all rules.	"Normal"
Channel	integer	R	video channel, which starts from 1.	1
Example				
<code>http://<server>/cgi-bin/VideoInAnalyse.cgi?action=getTemplateRule&Channel=1&Class=Normal</code>				

Response Params (key=value format in body)

Name	Type	R/O	Description	Example
Rule	object	R	rule info	
+Normal	object	R	business class	
++CrossRegionDetection	object	R	cross region templet	
++CrossLineDetection	object	R	default config of crossline format as the same as each object in the second dimension of VideoAnalyseRule	
+++Config	object	R	detail of rule config	
++++SizeFilter	object	R		
+++++MinSize	int[2]	R		[0, 0]
+++++MaxSize	int[2]	R		[8191, 8191]
+++++DetectLine	object[]	R		
+++++Direction	char[]	R	direction	"Both"
+++PtzPresetId	integer	R	Preset number	0
+++ObjectTypees	char[][]	R	type of detect object	["Unknown"]
+++Type	char[64]	R	rule type	"CrossLineDetection"
+++Class	char[16]	R	business plan class	"Normal"
+++Enable	bool	R	enable the rule	false
+++Id	integer	R	rule id	0

Example

```

Rule.Normal.CrossLineDetection.Id=0
Rule.Normal.CrossLineDetection.Enable=false
Rule.Normal.CrossLineDetection.Class=Normal
Rule.Normal.CrossLineDetection.Type=CrossLineDetection
Rule.Normal.CrossLineDetection.ObjectTypes[0]=Unknown
Rule.Normal.CrossLineDetection.PtzPresetId=0
Rule.Normal.Config.Direction=Both
Rule.Normal.Config.SizeFilter.MaxValue[0]=[8191, 8191]
Rule.Normal.Config.SizeFilter.MinValue[0]=[0, 0]
...

```

9.6.7 [Config] IntelliSchemeTourEnableSetting

Config Data Params

Name	Type	R/O	Description	Example
IntelliSchemeTour	object	R	IntelliSchemeTour Setting	
+Enable	bool	R	IntelliSchemeTour Enable Setting	True

Please refer to "4.2.1 Obtaining and Setting Configuration" to obtain and set the configuration. The specific examples are as follows:

Get Config Request Example

```
http://10.0.0.8/cgi-bin/configManager.cgi?action=getConfig&name=IntelliSchemeTour
```

Get Config Response Example

```
table.IntelliSchemeTour.Enable=true
```

Set Config Request Example

```
http://172.27.2.72/cgi-bin/configManager.cgi?action=setConfig&IntelliSchemeTour.Enable=true
```

Set Config Response Example

```
OK
```

9.6.8 [Config] Intelligent Tour Plan

Config Data Params				
Name	Type	R/O	Description	Example
IntelliSchemeTour	object	O	Intelligent package tour plan. No channel sensitive.	
+Enable	bool	O	Whether the package tour plan is enabled.	true
+TourPriMode	enumchar[32]	O	Tour priority mode enumchar[32]{ "Switch": Scene priority, tour in strict accordance with the defined tour time. "Detect": Detection priority, with the complete set of data as the condition for scene switching upon completion of scene detection (For illegal parking detection cameras. Multiple images may be needed in one scene. In this case, the camera can rotate to the next scene after all images are captured). "Tour": Tour priority, similar to detection priority, with a set of data as the condition for scene switching upon completion of scene detection (both close shot and long shot available. Apply close shot first and then long shot according to the tour path). } For details, refer to the capability set IntelliScheme.getCaps.	"Switch"

+IdleWaitingTime	uint32	O	Idle waiting time, that is, the idle delay time until the tour plan continues to take effect after a user stops operating the device, in second(s).	10
+SceneSwitchingTime	uint32	O	Scene switching time. When the tour plan is enabled and the priority mode is either detection priority or tour priority, it will automatically rotate to the next scene after an object cannot be detected for a certain period of time.	20
+TourPlan	object[7][16]	O	Package tour plan 2D array. The first dimension indicates the day of a week, totally 7 weekdays, and the second dimension indicates the tour groups supported that day.	
++TimeSection	TimeSection	O	Period of time. 00:00:00 indicates the start time and 23:59:59 indicates the end time.	"1 00:00:00-23:59:59"
++TourPath	object[]	O	Tour path 1D array, to adjust the order of packages. Up to 10 packages are supported. It is null by default and added by a user. Field values are specified by a user.	null
+++SchemeID	uint32	O	Intelligent package ID, value range: 1-300.	1
+++Duration	uint32	O	Retention time, value range: 30-36,000 s or 0. It is 0 by default if this field is unavailable. Indicates that the switching has not been made during the period of time.	600
+++Speed	uint32	O	The speed of the camera rotating to the preset when a package includes a preset channel of a PTZ camera, value range: 1-10 speed levels.	7

Please refer to "4.2.1 Obtaining and Setting Configuration" to obtain and set the configuration. The specific examples are as follows:

Get Config Request Example

<http://10.0.0.8/cgi-bin/configManager.cgi?action=getConfig&name=IntelliSchemeTour>

Get Config Response Example


```

table.IntelliSchemeTour.TourPlan[5][11].TimeSection=1 00:00:00-00:00:00
table.IntelliSchemeTour.TourPlan[5][12].TimeSection=1 00:00:00-00:00:00
table.IntelliSchemeTour.TourPlan[5][13].TimeSection=1 00:00:00-00:00:00
table.IntelliSchemeTour.TourPlan[5][14].TimeSection=1 00:00:00-00:00:00
table.IntelliSchemeTour.TourPlan[5][15].TimeSection=1 00:00:00-00:00:00
table.IntelliSchemeTour.TourPlan[6][0].TimeSection=1 00:00:00-00:00:00
table.IntelliSchemeTour.TourPlan[6][1].TimeSection=1 00:00:00-00:00:00
table.IntelliSchemeTour.TourPlan[6][2].TimeSection=1 00:00:00-00:00:00
table.IntelliSchemeTour.TourPlan[6][3].TimeSection=1 00:00:00-00:00:00
table.IntelliSchemeTour.TourPlan[6][4].TimeSection=1 00:00:00-00:00:00
table.IntelliSchemeTour.TourPlan[6][5].TimeSection=1 00:00:00-00:00:00
table.IntelliSchemeTour.TourPlan[6][6].TimeSection=1 00:00:00-00:00:00
table.IntelliSchemeTour.TourPlan[6][7].TimeSection=1 00:00:00-00:00:00
table.IntelliSchemeTour.TourPlan[6][8].TimeSection=1 00:00:00-00:00:00
table.IntelliSchemeTour.TourPlan[6][9].TimeSection=1 00:00:00-00:00:00
table.IntelliSchemeTour.TourPlan[6][10].TimeSection=1 00:00:00-00:00:00
table.IntelliSchemeTour.TourPlan[6][11].TimeSection=1 00:00:00-00:00:00
table.IntelliSchemeTour.TourPlan[6][12].TimeSection=1 00:00:00-00:00:00
table.IntelliSchemeTour.TourPlan[6][13].TimeSection=1 00:00:00-00:00:00
table.IntelliSchemeTour.TourPlan[6][14].TimeSection=1 00:00:00-00:00:00
table.IntelliSchemeTour.TourPlan[6][15].TimeSection=1 00:00:00-00:00:00
table.IntelliSchemeTour.TourPriMode=Switch

```

Set Config Request Example

<http://172.27.2.72/cgi-bin/configManager.cgi?action=setConfig&IntelliSchemeTour.TourPriMode=Switch>

Set Config Response Example

OK

9.6.9 Export Intelligent Diagnosis, Allowlist, and Blocklist Information

Export information, such as intelligent diagnosis, allowlist, and blocklist information. The export data format is csv

Request URL	<a href="http://<server>/cgi-bin/api/lmExport/exportData">http://<server>/cgi-bin/api/lmExport/exportData			
Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
type	enumint	O	Exported data type. enumint{ 0: License plate allowlist 1: License plate blocklist 2: License plate blocklist and allowlist 3: Voice talk contacts 4: Custom password 5: Card recordset 6: Face information for access	0

			control 7: Fingerprints for access control 8: IPC information of voice talk 9: Calling person information 10: Public password recordset 11: Operation and maintenance information }	
--	--	--	--	--

Request Example

```
{
  "type": 1
}
```

Response Params (binary in body)

Name	Type	R/O	Description	Example
------	------	-----	-------------	---------

Response Example

HTTP/1.1 200 OK
Server: Device/1.0
Content-Type: application/octet-stream
Content-Length: <length>
< binary data>

9.6.10 Import Intelligent O&M, Allowlist, and Blocklist Information

Import intelligent O&M, allowlist, and blocklist information.

Request URL	http://<server>/cgi-bin/api/lmExport/importData			
Method	POST			
Request Params (multipart in body)				
Name	Type	R/O	Description	Example
type	uint32	R	Imported data type enumint{ 0 : License plate allowlist 1 : License plate blocklist 2 : License plate blocklist and allowlist 3: Voice talk contacts 4: Custom password 5: Card recordset 6: Face information for access control 7: Fingerprint for access control 8: IPC information of voice talk 9: Calling person information 10: Public password recordset 11: Operation and maintenance	0

			information }	
isOverWrite	bool	R	Whether to overwrite the original data. If isOverWrite=true, clear all license plate lists, and then import the data. If isOverWrite=false, it will be imported directly, even if the license plate to be imported is duplicated with the existing license plate of the device.	true

Request Example

POST http://<server>/cgi-bin/api/lmExport/importData HTTP/1.1

User-Agent: client/1.0

Content-Type: multipart/x-mixed-replace; boundary=<boundary>

--<boundary>

Content-Type: application/json

Content-Length: 12

```
{
  "type": 0,
  "isOverWrite": true
}
```

--<boundary>

Content-Type: application/octet-stream

Content-Length: 800

< binary data >

--<boundary>

Response Params (JSON format in body)

Name	Type	R/O	Description	Example
Response Example				
{}				

9.6.11 Get Intelligent Capability

Request URL	http://<server>/cgi-bin/intelli.cgi?action=getCaps			
Method	GET			
Request Params (none)				
Name	Type	R/O	Description	Example
capsName	string	O	Name of capability, it can be: AnalyseMode: get analyse mode capacity MultiChannelMode: get multi channel seperate intelligent capacity and	AnalyseMode

		combined intelligent capacity MultiVideoAnalyse: get all intelligent capacity from the multichannel device in each channel if omit, means get all capacity	
--	--	--	--

Request Example

```
http://192.168.1.108/cgi-bin/devVideoAnalyse.cgi?action=getLastEventInfo&channel=1&ClassName=Stereobehavior&EventName=ManStandDetection
```

Response Params (key=value format in body)

Name	Type	R/O	Description	Example
caps	object	R	The video service capabilities.	
+AnalyseMode	char[]	O	Analyse mode, can be : RealStream, RecordFileStream, PicFileStream	RecordFileStream
+MultiChannelMode	object[]	O	Multi channel description	
++IndepMode	object[][]	O	Multi channel open at one time separately	
+++Channel	int	R	Video channel number, start from 0	
+++Type	char[]	R	Business class	
++CompMode	object[][]	O	Multi channel combined mode	
+++Channel	int	R	Video channel number, start from 0	0
+++Type	char[]	R	Business class	
+MultiVideoAnalyse	object[]	O	List of channel capabilities	
++caps	object	R	Video analyse capabilities	
+Algorithm	object[]	O	Algorithm version information	
++Class	char[]	R	Intelligent analyse class, ex : ObjectDetect	
++AlgorithmVersion	char[]	R	Algorithm version	
++AlgorithmVendor	char[]	R	Algorithm Vendor name	
+TotalCapacity	object[]	O	Intelligent analyse total capacity	
++Class	char[]	R	Intelligent analyse class, ex : ObjectDetect	
++Type	char[][]	R	Intelligent analyse rules, ex : ["FaceDetection"]	
++Number	int	R	The maximum number of video channels that can be analysed at same time	

Response Example

```
caps.AnalyseMode=RealStream
caps.MultiChannelMode.IndepMode[0][0].Channel=0
caps.MultiChannelMode.IndepMode[0][0].Type=Normal
caps.MultiChannelMode.IndepMode[0][1].Channel=1
caps.MultiChannelMode.IndepMode[0][1].Type=ObjectDetect
...
caps.MultiChannelMode.CompMode[0][0].Channel=0
```

```

caps.MultiChannelMode.CompMode[0][0].Type=Normal
caps.MultiChannelMode.CompMode[0][1].Channel=1
caps.MultiChannelMode.CompMode[0][1].Type=ObjectDetect
...
caps.Algorithm[0].Class=Normal
caps.Algorithm[0].AlgorithmVersion=V2.8
caps.Algorithm[0].AlgorithmVendor=XXX
caps.Algorithm[1].Class=ObjectDetect
caps.Algorithm[1].AlgorithmVersion=V2.8
caps.Algorithm[1].AlgorithmVendor=YYY
...
caps.TotalCapacity[0].Class=Normal
caps.TotalCapacity[0].Type[0]=FaceDetection
caps.TotalCapacity[0].Number=3
caps.TotalCapacity[1].Class=ObjectDetect
caps.TotalCapacity[1].Type[0]=FaceDetection
caps.TotalCapacity[1].Number=3
...

```

9.6.12 Subscribe Resource Usage Info

Request URL	http://<server>/cgi-bin/intelli.cgi?action=attachResource		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
heartbeat	int	O	Send heartbeat interval, range is [1, 60], unit is second. If the URL contains this parameter, and the value is 5, it means every 5 seconds the device should send the heartbeat message to the client, the heartbeat message are "Heartbeat". If this parameter is not present, its default value is 60.
Request Example			
http://192.168.1.108/cgi-bin/intelli.cgi?action=attachResource&heartbeat=5			

Response Params (multipart; key=value format in body)			
Name	Type	R/O	Description
RemainCapacity	object[]	R	Remain capacity of intelligent analyse
+Class	string	R	Intelligent analyse class, ex : ObjectDetect
+Number	int	O	Remain number of video channels that can be analysed
Response Example			
HTTP/1.1 200 OK			
Server: Device/1.0			
Content-Type: multipart/x-mixed-replace; boundary=<boundary>			

Connection: closed

--<boundary>

Content-Type: text/plain

Content-Length: <length>

RemainCapacity[0].Class=Normal

RemainCapacity[0].Number=1

RemainCapacity[1].Class=ObjectDetect

RemainCapacity[1].Number=2

--<boundary>

Content-Type: text/plain

Content-Length: 11

Heartbeat

--<boundary>

Content-Type: text/plain

Content-Length: <length>

RemainCapacity[0].Class=Normal

RemainCapacity[0].Number=1

RemainCapacity[1].Class=ObjectDetect

RemainCapacity[1].Number=1

...

9.6.13 Export Encrypted Files

Request URL	http://<server>/cgi-bin/api/SecuritylmExport/exportData		
Method	POST		
Request Params (JSON format in body)			
Name	Type	R/O	Description
key	char[1024]	R	Password string encrypted with public key and encoded by Base64
type	enumint	R	Exported types enumint{ 0: License plate allowlist 1: License plate blocklist 2: License plate allowlist/blocklist 3: Voice talk contacts 4: Custom password 5: Card recordset 6: Face information for access control 7: Fingerprint for access control 8: Network camera information of voice talk}

		9: Calling person information 10: Public password recordset 11: Operation and maintenance information 12: Face data 13: Face images }	
--	--	--	--

Request Example

```
{
  "key": "1BBzdSS***",
  "type": 0
}
```

Response Params (binary in body)

Name	Type	R/O	Description	Example
Response Example				
HTTP/1.1 200 OK				
Server:	Device/1.0			
Content-Type:	application/octet-stream			
Content-Length:	<length>			
< binary data>				
//Before the device returns the actual data, use AES symmetric encryption (the key is generated according to the plaintext conversion after key decryption).				
//AES-256bit-CBC. Filling mode: PKCS7.				

9.6.14 Platform intelligent control

Request URL	http://<server>/cgi-bin/api/intelli/setPollingConfig		
Method	POST		
Request Params (JSON format in body)			
Name	Type	R/O	Description
Config	object[]	O	Length max 256
+Enable	bool	O	Whether it is intelligently turned on
+Channel	uint32	O	Channel number
+GlobalType	char[64]	O	Global configuration, only support single channel and single intelligence,
+RuleType	char[16][64]	O	Rule configuration, which represents all rules enabled in the GlobalType or GlobalTypeList scenario
+GlobalTypeLi st	char[6][64]	O	The global configuration list, the extension supports the opening of multiple intelligence of a single channel, choose one of GlobalType and GlobalTypeList, GlobalTypeList will be used first
Request Example			

```
{
  "Config": [
    {
      "Enable": true,
      "Channel": 12,
      "GlobalType": "Normal",
      "RuleType": ["CrossLineDetection"],
      "GlobalTypeList": ["Normal", "GasStation"]
    }, ...
  ]
}
```

Response Params (JSON format in body)

Name	Type	R/O	Description	Example
Response Example				
{}				

9.6.15 Get Scene List

Get the list of intelligent schemes that have been opened for a channel.

Request URL	http://<server>/cgi-bin/devVideoAnalyse.cgi?action=getSceneList		
Method	GET		
Request Params (JSON format in body)			
Name	Type	R/O	Description
Request Example			
http://<server>/cgi-bin/devVideoAnalyse.cgi?action=getSceneList			

Response Params (key=value format in body)

Name	Type	R/O	Description	Sample
VideoAnalyseGlobal	object[]	R	List of intelligent schemes opened for each video channel.	
+Scenes	char[32][16]	R	If none of the intelligent schemes are enabled, return VideoAnalyseGlobal[0].Scenes[0]=""	["Normal", "HeatMap"]
Response Example				
VideoAnalyseGlobal[0].Scenes[0]=Normal VideoAnalyseGlobal[0].Scenes[1]=FaceDetection				

9.6.16 Open Intelligent Schemes

Open several intelligent schemes for a channel

Request URL	http://<server>/cgi-bin/devVideoAnalyse.cgi?action=enableScene
Method	GET

Request Params (key=value format in URL)

Name	Type	R/O	Description	Example
typeList	char[32][16]	R	For the list of intelligent schemes to open, refer to getSceneList	["Normal", "HeatMap"]
channel	int32	R	Video channel, counted from 0	0

Request Example

http://192.168.1.108/cgi-bin/devVideoAnalyse.cgi?action=enableScene&typeList=[Normal,HeatMap]&channel=0

Response Params (key=value format in body)

Name	Type	R/O	Description	Sample
response	char[256]	R	Result "OK" is returned for success, and "Error" is returned for failure	"OK"

Response Example

OK

9.6.17 Close Intelligent Schemes

Close several intelligent schemes for a channel

Request URL	http://<server>/cgi-bin/devVideoAnalyse.cgi?action=disableScene			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
typeList	char[32][16]	R	For the list of intelligent schemes to close, refer to getSceneList	["Normal", "HeatMap"]
channel	int32	O	Video channel, counted from 0	0

Request Example

http://192.168.1.108/cgi-bin/devVideoAnalyse.cgi?action=disableScene&typeList=[Normal,HeatMap]&channel=0

Response Params (OK in body)

Name	Type	R/O	Description	Sample
response	char[256]	R	Result "OK" is returned for success, and "Error" is returned for failure	"OK"

Response Example

OK

9.6.18 Get Algorithm Version Of Channel

Get algorithm version according to channel and algorithm type.

Request URL	http://<server>/cgi-bin/api/intelli/getChannelAlgVersion			
Method	POST			

Request Params (JSON format in body)

Name	Type	R/O	Description	Example
Channel	int32	R	-1 means to obtain all channels. Other values mean corresponding channels	-1
AlgType	char[32]	O	Refer to VAClassEnum. "Null" or without this field, get all algorithm version information	"Normal"

Request Example

```
{
    "Channel": -1,
    "AlgType": "Normal"
}
```

Response Params (JSON format in body)

Name	Type	R/O	Description	Example
Info	object[] [256]	R	Version information	
+ClassType	char[32]	R	Refer to VAClassEnum	"Normal"
+AlgTypeVersion	char[32] [256]	R	Algorithm version information.	["R497490.0_V3.004.000000.1.R"]

Response Example

```
{
    "Info": [
        {
            "ClassType": "Normal",
            "AlgTypeVersion": ["R497490.0_V3.004.000000.1.R"]
        }
    ]
}
```

9.7 WorkSuitCompareServer

9.7.1 Add Compliance Library

Request URL	http://<server>/cgi-bin/api/WorkSuitCompareServer/createGroup			
Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
Group	object	R	Compliance library group info	
+GroupName	char[128]	R	Compliance library group name, up to 127 characters.	"takeout"
+CutoutPolicy	uint32	O	Preferred solution 0: Full body 1: Upper body	1
+Similarity	uint8	O	Similarity threshold, range [1, 100], default 67	90

+GroupType	enumchar[16]	O	Compliance library group type enumchar[16]{ "BlockListDB": Registry (control list) to save external imported data }	"BlockListDB"
+GroupDetail	char[256]	O	group remark information	"community owner"
+Type	enumchar[16]	O	type, default: "WorkSuit" enumchar[16]{ "WorkSuit" "ClothesCommon" } In the future, there may be traffic police comparison, takeout comparison and worker comparison, and "clothescommon" may be used for comparison through clothing characteristics	"ClothesCommon"

Request Example

```
{
    "Group": {
        "GroupName": "takeout",
        "CutoutPolicy": 1,
        "Similarity": 90,
        "GroupType": "BlockListDB",
        "GroupDetail": "community owner",
        "Type": "ClothesCommon"
    }
}
```

HTTP API V3.35 - Intelbras

Response Params (JSON format in body)

Name	Type	R/O	Description	Example
GroupID	char[64]	R	Compliance library group ID	"10000"

Response Example

```
{
    "GroupID": "10000"
}
```

9.7.2 Delete Compliance Library

Request URL	http://<server>/cgi-bin/api/WorkSuitCompareServer/deleteGroup			
Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
GroupID	char[64]	R	Compliance library group ID	"000001"
Request Example				

```
{
    "GroupID": "000001"
}
```

Response Params (JSON format in body)

Name	Type	R/O	Description	Example
Response Example				
{}				

9.7.3 Find Compliance Library

Request URL	http://<server>/cgi-bin/api/WorkSuitCompareServer/findGroup			
Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
GroupID	char[64]	R	Compliance library group ID, "" for all Compliance library group IDs	"000001"
Request Example				
{				
"GroupID": "000001"				
}				

Response Params (JSON format in body)

Name	Type	R/O	Description	Example
GroupList	object[64]	O	List of found compliance library group information	
+GroupID	char[64]	O	Compliance library group ID	"000001"
+GroupName	char[128]	O	Compliance library group name, up to 127 characters.	"taken"
+CutoutPolicy	uint32	O	Preferred solution 0: Full body 1: Upper body	0
+GroupType	enumchar[16]	O	Compliance library group type enumchar[16]{ "BlockListDB": Registry (control list) to save external imported data }	"BlockListDB"
+GroupDetail	char[256]	O	group remark info	"community owner"
+GroupSize	int	O	size	30
+Channels	int16[1024]	O	The list of video channel numbers to which the current group is bound (see setgroup). If not associated with any video channel, it should be [-1]	[0]
+Similarity	uint8[1024]	O	It corresponds to the similarity threshold when Binding video. The	[90]

			array length is equal to channels. It is meaningless when no video channel is bound. Each value range [1, 100]	
+FeatureState	uint[4]	O	The number of work suit in various states in the group and the work suit that have not completed modeling (feature extraction) cannot be identified by the algorithm Array subscript correspondence [0] - the number of work suit ready for modeling does not guarantee the success of modeling [1] - the number of work suit failed in modeling, the picture does not meet the algorithm requirements, and the picture needs to be replaced [2] - the number of work suit successfully modeled, and the data can be used for work suit identification by algorithm [3] - the number of models that have been successfully modeled but become unavailable due to algorithm upgrade, it will be available after remodeled.	[10,20,30,40]

Response Example

```
{
  "GroupList": [
    {
      "GroupID": "000001",
      "GroupName": "taken",
      "CutoutPolicy": 0,
      "GroupType": "BlockListDB",
      "GroupDetail": "community owner",
      "GroupSize": 30,
      "Channels": [0],
      "Similarity": [90],
      "FeatureState": [10,20,30,40]
    },...{}]
}
```

9.7.4 Get Compliance Library Arming Information of Channels

Request URL	http://<server>/cgi-bin/api/WorkSuitCompareServer/getGroup			
Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example

Channel	int	R	Video channel number	0
Request Example				
{				
	"Channel": 0			
}				

Response Params (JSON format in body)				
Name	Type	R/O	Description	Example
GroupList	object[128]	O	List of compliance library group information	
+GroupId	char[64]	O	Compliance group ID, up to 63 characters.	"001"
+Similarity	uint8	O	Similarity threshold; value range: 1–100.	80
+CutoutPolicy	uint32	O	Preferred solution 0: Full body 1: Upper body	1

Response Example

```
{
  "GroupList": [
    {
      "GroupId": "001",
      "Similarity": 80,
      "CutoutPolicy": 1
    }, ...
  ]
}
```

9.7.5 Modify Compliance Group Information

Request URL	http://<server>/cgi-bin/api/WorkSuitCompareServer/modifyGroup			
Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
Group	object	R	Compliance library group info	
+GroupId	char[64]	R	Compliance group ID, up to 63 characters	"0001"
+GroupName	char[128]	O	Compliance library group name, up to 127 characters.	"park"
+Similarity	uint8	O	Similarity threshold, range [1, 100], default 67	90
+GroupType	enumchar[16]	O	Compliance library group type enumchar[16]{ "BlockListDB": Registry (control list) to save external imported data }	"BlockListDB"
+GroupDetail	char[256]	O	group remark info	"community owner"
Request Example				
{				
	"Group": {			
	"GroupId": "0001",			

```

        "GroupName": "park",
        "Similarity": 90,
        "GroupType": "BlockListDB",
        "GroupDetail": "community owner"
    }
}

```

Response Params (JSON format in body)

Name	Type	R/O	Description	Example
Response Example				
{}				

9.7.6 Deploy Compliance Library

Request URL	http://<server>/cgi-bin/api/WorkSuitCompareServer/setGroup			
Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
Channel	int	R	video channel number	0
GroupList	object[128]	O	Arming information list: If the list does not exist, the arming of all compliance libraries for the channels are deleted.	
+GroupID	char[64]	O	Compliance group ID, up to 63 characters	"001"
+Similarity	uint8	O	Similarity threshold for each compliance group. Action after the comparison similarity is higher than the threshold Absence of this field means no modification Similarity range [0, 100]	80

Request Example

```
{
    "Channel": 0,
    "GroupList": [
        {
            "GroupID": "001",
            "Similarity": 80
        }, ...
    ]
}
```

Response Params (JSON format in body)

Name	Type	R/O	Description	Example
Response Example				
{}				

9.7.7 Find Workwear Information in Compliance Library

Request URL	http://<server>/cgi-bin/api/WorkSuitCompareServer/startFind		
Method	POST		
Request Params (JSON format in body)			
Name	Type	R/O	Description
Condition	object	O	Search condition.
+GroupId	char[128][64]	O	Array list of compliance library IDs. Each compliance library ID is up to 63 characters.
WorkSuit	object	O	Search condition of workwear information
+FeatureState	uint8	O	Feature value status: 0: Unknown; 1: Extraction failed; 2: Extracted successfully; 3: Modeled but algorithm upgrade causes data unavailability and remodeling is required.
Request Example			
<pre>{ "Condition": { "GroupId": ["00001","00002"] }, "WorkSuit": { "FeatureState": 1 } }</pre>			

Response Params (JSON format in body)			
Name	Type	R/O	Description
Token	uint	O	Token for search task, which is used to get search results and stop the search.
TotalCount	int	O	Total number of results returned.
Response Example			
<pre>{ "Token": 2342343, "TotalCount": 3333 }</pre>			

9.7.8 Get Find Workwear Information Result

Acquire the work uniform information from the search result.

Note: If doFind operation is not performed for 60 seconds, the returned token will be invalid.

Request URL	http://<server>/cgi-bin/api/WorkSuitCompareServer/doFind
--------------------	--

Method	POST		
Request Params (JSON format in body)			
Name	Type	R/O	Description
Condition	object	R	Search for the input parameter of work uniform.
+Token	uint	R	Search token.
+BeginNumber	uint	R	Start number of the search. The search starts from the "beginNumber" records, and returns predefined number (count) of records. $0 \leq \text{beginNumber} \leq \text{totalCount}-1$
+Count	int	R	Number of entries obtained each time. It should not exceed 50.
+NeedData	uint	O	Image format mask returned by search results. It is added for IPC. bit0 means to return the HTTP link of image. For example, 1 only returns HTTP link. If the field does not exist, the server decides how to return it.
Request Example			
{ "Condition": { "Token": 46878, "BeginNumber": 0, "Count": 20, "NeedData": 1 } }			

Response Params (JSON format in body)			
Name	Type	R/O	Description
Results	object	R	Output parameter of the search.
+Found	int	R	The actual number of entries returned by this search.
+Candidates	object[]	R	Information list of work uniform to be selected. Maximum 50 information lists can be returned at one time.
++WorkSuit	object	R	Searched work uniform information list.
+++UID	string	O	Work uniform identifier. The

			maximum length is 31 characters.	
+++GroupID	string	O	Compliance database ID. The maximum length is 63 characters.	10000
+++GroupName	string	O	Name of work uniform group.	Group1
+++FeatureState	uint	O	Feature status. 0: Unknown. 1: Extraction failed. 2: Extraction succeeded. 3: Algorithm upgrade makes data unavailable and it needs to be remodeled. 4: Calculating features.	0
+++FeatureErrCode	uint	O	<p>Recorded reasons for modeling failure.</p> <p>Only valid when FailedCnt is greater than 0.</p> <p>0: Unknown (other errors)</p> <p>1: Modeling failed.</p> <p>2: System errors (for example, errors caused by invalid license and unstated modeling analyzer).</p> <p>3: Database operation failed.</p> <p>4: image decoding failed.</p> <p>5: Multiple targets.</p> <p>6: No target.</p>	0
+++ImagePath	string	O	Work uniform thumbnail path.	/mnt/2010/8/1/dav/15:40:50.jpg
+++SourceUID	string	O	Unique identifier of panoramic image. It is generated by the platform.	1
+++SourceFileName	string	O	Panoramic image name.	aaa
+++Image	object	O	Image information	[.]
++++ Width	uint	O	Image width	100
++++ Height	uint	O	Image height	50
++++ Offset	uint	O	Offset in the binary data block	0
++++ Length	uint	O	Image size	10000

Response Example

```
{
  "Results": {
    "Found": 12,
    "Candidates": [
      {
        "WorkSuit": {
          "UID": "111",
          "GroupID": "0001",
        }
      }
    ]
  }
}
```

```

        "GroupName": "group1",
        "FeatureState": 1,
        "FeatureErrCode": 2,
        "ImagePath": "/mnt/2010/8/1/dav/15:40:50.jpg",
        "SourceUID": "1",
        "SourceFileName": "aaaaa",
        "Image": {
            "Width": 1920,
            "Height": 1080,
            "Offset": 0,
            "Length": 1156
        },...{}]
    }
}
}

```

9.7.9 Stop Find Workwear Information

Request URL	http://<server>/cgi-bin/api/WorkSuitCompareServer/stopFind			
Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
Token	uint	R	Token for search task, which is used to get search results and stop the search.	46878
Request Example				
{ "Token": 46878 }				

Response Params (JSON format in body)

Name	Type	R/O	Description	Example
Response Example				
{}				

9.7.10 Delete Workwear Information

Request URL	http://<server>/cgi-bin/api/WorkSuitCompareServer/deleteByUID			
Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
UID	char[64][32]]	O	Workwear identifier list, up to 31 characters.	["123", "234"]
Request Example				
{ "UID": ["123", "234"]				

}

Response Params (JSON format in body)

Name	Type	R/O	Description	Example
ErrCode	uint[64]	O	Error code list, indicating error messages when deleting information. 0: Success. 1: The workwear does not exist. 2: Database operation failed.	[0,0]

Response Example

```
{
    "ErrCode": [0,0]
}
```

9.7.11 Re-extracting Features by Workwear

Request URL	http://<server>/cgi-bin/api/WorkSuitCompareServer/reAbstract			
Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
WorkSuit	object[64]	O	Multiple work suit information, [] means to reconstruct all work suit with missing or mismatched feature vectors	
+UID	char[32]	O	Workwear identifier list; each UID is up to 31 characters.	"120837"
Token	uint	O	The token value of the work clothes modeling. The token can be obtained through getappendtoken	1
Request Example				
{	<pre> "WorkSuit": [{ "UID": "120837" }, ...], "Token": 1 }</pre>			

Response Params (JSON format in body)

Name	Type	R/O	Description	Example
Response Example				
{}				

9.7.12 Stop Re-extracting Workwear Features

Request URL	http://<server>/cgi-bin/api/WorkSuitCompareServer/stopReAbstract			
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
Response Example				{}
{}				

9.8 Smart Motion Detection

9.8.1 [Config] SmartMotionDetect

Config Data Params				
Name	Type	R/O	Description	Example
SmartMotionDetect	object[]	O	PrivacyMasking config, each channel has one config object	
+Enable	bool	O	Enable/Disable	false
+Sensitivity	char[16]	O	Detection sensitivity, can be: "Low", "Middle", "High"	"Middle"
+SmartTrack	bool	O	Enable/Disable smart track default false	true
+ObjectTypes	object	O	Detection object type	
--Human	bool	O	Whether detect motion of human	true
--Vehicle	bool	O	Whether detect motion of vehicle	true
--Animal	bool	O	Whether detect motion of animal	true
+Mode	char[16]	O	range { "General", "Agile" }	"General"

Please refer to "4.2.1 Get and Set Configure" for configuration getting and setting. Specific examples are as follows :

Get Config Request Example

```
http://10.0.0.8/cgi-bin/configManager.cgi?action=getConfig&name=SmartMotionDetect
```

Get Config Response Example

```
table.SmartMotionDetect[0].Enable=true
table.SmartMotionDetect[0].Sensitivity=Middle
table.SmartMotionDetect[0].ObjectTypes.Human=true
table.SmartMotionDetect[0].ObjectTypes.Vehicle=true
...

```

Set Config Request Example

```
http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&SmartMotionDetect[0].Enable=true&SmartMotionDetect[0].Sensitivity=Middle&SmartMotionDetect[0].ObjectTypes.Human=true&SmartMotionDetect[0].ObjectTypes.Vehicle=true
```

Set Config Response Example

```
OK
```

9.8.2 Start SMD Data Search

Request URL	http://<server>/cgi-bin/api/SmdDataFinder/startFind			
Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
Condition	object	R	Search conditions	
+Channel	int32	O	Channel number. -1 means searching for all channels	0
+Channels	int[]	O	Channel number array (starting from 0). There should be at least one field, Channel or Channels. Generally, If both Channel and Channels fields exist, Channels should prevail.	
+SmdType	char[8][32]	O	Video type. Array "smdTypeHuman": Human. "smdTypeVehicle": Vehicle. "smdTypeHumanAndVehicle": Human and vehicle.	["smdTypeHuman", "smdTypeVehicle", "smdTypeHumanAndVehicle"]
+StartTime	char[32]	O	Start time	"2017-08-01 00:00:00"
+EndTime	char[32]	O	End time	"2017-08-02 00:00:00"
+Order	char[16]	O	Search for ordering method. "ascOrder": Ascending order (default value). "descOrder": Descending order.	"ascOrder"

Request Example

```
{
  "Condition": {
    "Channel": 0,
    "Channels": [0,1,2,5],
    "SmdType": ["smdTypeHuman", "smdTypeVehicle", "smdTypeHumanAndVehicle" ],
    "StartTime": "2017-08-01 00:00:00",
    "EndTime": "2017-08-02 00:00:00",
    "Order": "ascOrder"
  }
}
```

Response Params (JSON format in body)

Name	Type	R/O	Description	Example
Token	int	O	Return searched ID	1
Count	int	O	Total number of searched entries	100

Response Example

```
{
  "Token": 1,
```

```

    "Count": 100
}

```

9.8.3 Get SMD Data Search Result

Request URL	http://<server>/cgi-bin/api/SmdDataFinder/doFind			
Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
Token	int	R	Search ID	1
Offset	int32	R	Location offset	0
Count	int	R	Number of searched items	100
Request Example				
{	{ "Token": 1, "Offset": 0, "Count": 100 }			

Response Params (JSON format in body)				
Name	Type	R/O	Description	Example
SmdInfo	object[]	R	It is an array. Each element represents the historical data of a record data at a certain time.	
+Channel	int32	O	Channel number, starting from 0	1
+EndTime	char[32]	O	End time. "201	"2017-08-02 00:00:00"
+Type	char[32]	O	Video type. "smdTypeHuman": Human. "smdTypeVehicle": Vehicle. "smdTypeHumanAndVehicle": Human and vehicle.	"smdTypeHuman"
+StartTime	char[32]	O	Start time	2017-01-01 10:00:
Response Example				
{	{ "SmdInfo": [{ "Channel": 1, "StartTime": "2017-08-01 00:00:00", "EndTime": "2017-08-02 00:00:00", "Type": "smdTypeHuman" }, ... {}] }			

9.8.4 End SMD Data Search

Request URL	http://<server>/cgi-bin/api/SmdDataFinder/stopFind	
Method	POST	

Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
Token	int	O	Search ID	1
Request Example				
{ "Token": 1 }				

Response Params (JSON format in body)				
Name	Type	R/O	Description	Example
Response Example				{}

9.9 Intelligent analysis tasks

9.9.1 Add task

Request URL	http://<server>/cgi-bin/api/analyseTaskManager/add			
Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
Task	object	R	Task object	
+TaskUserData	char[512]	R	Task data, add it intact to the response	"asdvtgrffg"
+SourceType	char[32]	R	Data source type Use devVideoAnalyse component for local streams (LocalStream), and it is not included in this component group. "LocalStream": Local stream Currently, the local stream is used only by NVR and SMD. Use devVideoAnalyse for other cases. "RemoteStream": Remote real stream "LocalVideoFile": Local video files "LocalPictureFile": Local picture files "RemoteVideoFile": Remote video files "RemotePictureFile": Remote picture files "PushVideoFile": Actively pushed video files "PushPictureFile": Actively pushed picture files	"StreamSource"

			<p>"OffLineVideoFile": Offline video files (from third parties)</p> <p>PushPictureFileByRule: Remote picture files. Rules and picture information are not included when adding tasks. Through pushing the picture interface, each picture is attached with different rule information (Used in energy-related scenes currently).</p> <p>"IOTData": IoT device data</p> <p>For active pushed files such as PushVideoFile, the command does not specify the file information, but pushes data through command such as analysePushVideoFile.</p>	
+IsStart	bool	O	<p>Starts immediately after being created or not HTTP://192.168.1.100:8080/api/v1/video/(Valid when the file type is LocalVideoFile, LocalPictureFile, RemoteVideoFile, and RemotePictureFile. Add, attach and then start in case of result loss) If the value is true, the command will start immediately after being created.</p>	true
+LocalStream	object	O	<p>Local real-time video sources Valid when SouceType is LocalStream</p>	
++Channel	int	O	Video channel	0
++Subtype	uint8	O	<p>Stream type: 0: Main stream 1: Sub stream 1 2: Sub stream 2</p>	0
+RemoteStrea m	object	O	<p>Remote real-time stream source Valid when SouceType is RemoteStream</p>	
++Protocol	char[32]	O	<p>Video stream protocol Supported protocols by now: "Private" "RTSP" "Onvif" "GB28181" "HIKVISION" "BSCP"</p>	"RTSP"
++Path	char[256]	O	Video stream URL	"rtsp://10.35.114.145:554/dss/monitor/param?camer

				aid=AGfQv8TwA1ALIPO2 9LA9L2&substream=1&tra ckID=801"
++IP	char[64]	O	IP Address	"10.35.114.145"
++Port	uint16	O	Port	554
++UserName	char[64]	O	Username	"admin"
++Password	char[64]	O	Password	"admin123"
++Channel	int	O	Video channel	0
++Subtype	uint8	O	Stream type: 0: Main stream 1: Sub stream 1 2: Sub stream 2	0
++ChannelId	char[256]	O	Platform channel No, used to bind the device channels.	"1000254\$1\$0\$110"
+LocalVideoFil e	object	O	Local video file, and only 1 video file is supported Valid when SouceType is LocalVideoFile	
++FileID	char[128]	O	File ID	"file-1234"
++Path	char[260]	O	File name	"/xxxx/a.dav"
+LocalPicture File	object	O	Local picture file, and multiple picture files are supported. Valid when SouceType is LocalPictureFile	
++Files	object[]	O		
+++FileID	char[128]	O	File ID	"file-1234"
+++Path	char[260]	O	File name	"/xxxx/a.jpg"
+RemoteVide oFile	object	O	Remote video file, and only 1 video file is supported. Valid when SouceType is RemoteVideoFile	
++Protocol	char[16]	O	Protocol for accessing remote files "RTSP", "RabbitMq" "Private" "Onvif" "GB28181" "HIKVISION" "BSCP" (FTP and RTSP are expected to be covered)	"RTSP"
++VideoAnaly sisProcess	uint32	O	Video analysis progress. When SourceType is RemoteVideoFile, the valid value range is 0–100. 100 means the analysis is compete.	66
++Path	char[260]	O	File path	"rtsp://192.168.1.1:554/xx xx/a.dav"

++IP	char[64]	O	Remote IP address	"192.168.1.1"
++Port	int	O	Remote port	554
++Channel	int	O	Video channel	1
++Subtype	uint8	O	Stream type: 0: Main stream 1: Sub stream 1 2: Sub stream 2	1
++Username	char[64]	O	Username	"admin"
++Password	char[64]	O	Password	"123456"
++FileID	char[128]	O	File ID	"file-1234"
++StartTime	char[64]	O	Start time	"2010-05-25 00:00:00"
++EndTime	char[64]	O	End time	"2010-05-25 23:59:59"
+OffLineVideoFile	object	O	Remote video file, and only 1 video file is supported (the file imported from a third party). Valid when SouceType is OffLineVideoFile	
++Protocol	char[16]	R	Protocol for accessing remote files RTSP	"RTSP"
++FileSize	uint64	O	File size (in bytes)	1000
++VideoAnalysisProcess	uint32	O	Video analysis progress. When SourceType is OffLineVideoFile, the valid value range is 0–100. 100 means the analysis is compete.	66
++Path	char[260]	R	File path	"rtsp://192.168.1.1:554/xx xx/a.dav"
++IP	char[64]	R	Remote IP address	"192.168.1.1"
++Port	int	R	Remote port	4000
++Username	char[64]	R	Username	"admin"
++Password	char[64]	R	Password	"123456"
++FileID	char[128]	R	File ID	"file-1234"
++isReportPTS	bool	O	Reports the relative timestamp or not	true
+RemotePictureFile	object	O	Remote picture file, and multiple picture files are supported. The server must be identical. Valid when SouceType is RemotePictureFile	
++Protocol	char[16]	O	Protocol for accessing remote files HTTP, RabbitMq, ActiveMq (FTP and RTSP are expected to be covered)	"HTTP"
++IP	char[64]	O	Remote IP address	"192.168.1.1"
++Port	int	O	Remote port	80
++Username	char[64]	O	Username	"admin"
++Password	char[64]	O	Password	"123456"

++Files	object[]	O	File list	
+++FileID	char[128]	O	File ID	"file-1234"
+++Path	char[260]	O	File name	"http://192.168.1.1:80/xxx x/a.jpg"
+Global	object	O	Global settings	
++Lanes	object[8]	O	Lane information; each lane includes 2 border lines. Maximum length: 8 arrays For highway all-in-one device, a channel supports up to 8 pairs of lane lines.	
+++Enable	bool	O	Enable lane The rules take effect only when the lane is enabled. It tackles the detection errors when a lane is under construction in a specific period. When the filed is empty, the lane is enabled by default.	true
+++Number	int	O	Lane No.	0
+++LeftLine	uint16[20][2]	O	Left lane line. The direction of the lane line refers to the lane direction, and the line at the left side of the lane is left lane line. The coordinates of the point is unified within the range of 0–8192.	
+++LeftLineExt	uint16[64][2]	O	LeftLineExt is the extension field of LeftLine. When the point number of the left lane line is less than 20, use LeftLine only. When the point number of the left lane line is more than 20, use LeftLine for the former 20 points, and LeftLineExt for the rest. LeftLine and LeftLineExt comprise the left lane line.	
+++LeftLineType	enumchar[16]	O	Attributes of the left lane line enumchar[16]{ "WhiteSolid": White solid line "WhiteDotted": White dotted line "Yellow": Yellow line "UpSolidDownDotted": Solid at the top and dotted at the bottom "UpDottedDownSolid": Dotted at the top and solid at the bottom}	"WhiteSolid"
+++RightLine	uint16[20][2]	O	Right lane line. The direction of the lane line refers to the lane direction, and the line at the right side of the	

			lane is right lane line. The coordinates of the point is unified within the range of 0–8192.	
+++RightLineExt	uint16[64][2]	O	RightLineExt is the extension filed of RightLine.	++++RightLineExt
+++RightLineType	char[16]	O	Attributes of the right lane line Refer to LeftLineType	"WhiteSolid"
++CalibrateArea	object[]	O	Calibration Area An array. Each scene includes multiple calibration areas. If the file does not exist, the whole scene is the calibration area. CalibrateArea, CalibrateArea1 CalibrateAreaN refers to scene, scene 1..... scene N. For highway all-in-one device, a channel supports 1 calibration area.	
+++Area	uint[20][2]	O	Calibrates a polygon area, and uses all. For highway all-in-one device: 1 area.	
+++Staffs	object[]	O	Ruler line A calibration area requires multiple ruler lines. Use 4 elements for depth of field calibration (3 vertical elements and 1 horizontal element) Use 2 elements for lane calibration (an element for each side) No element is needed for license plate calibration (illegal parking), and use VideoAnalyseCalibrate to configure separately. No element is needed for dual-PTZ calibration, and draw an area. LaneNew only covers 1 vertical straight line segment, and the coordinates of the vertical straight line must be entered. For highway all-in-one device, 1 Staffs for a calibration area.	
++++Type	enumchar[32]	O	Ruler type enumchar[32]{ "Horizontal": Horizontal line segment "Vertical": Vertical line segment "Any": Any line segment, unused	"Horizontal"

			"Cross": Cross line segment, unused } For highway all-in-one device (1 staffs): 1 type	
++++Start	int[2]	O	Coordinates of the start point The coordinates is unified within the range of 0–8192	[0, 0]
++++End	int[2]	O	Coordinates of the end point The coordinates is unified within the range of 0–8192	[100, 100]
++++Length	double	O	Actual length, in meters Note: The current system layer length, in meters	1.0
++GlobalDetectionEnable	bool	O	Enable detection identifiers in the area	false
++ObjectArea	object[20]	O	Arrays in the target detection area	
+++Area	uint16[20][2]	O	Area, refer to the configurations of DetectRegion.	
+++Type	enumchar[16]	O	Target area type enumchar[16]{ "SignalLight": Signal light "LEDScreen": LED screen }	"SignalLight"
++CameraHeight	float	O	Installation height of the camera in meters	6.2
+Module	object	O	Module configuration	
++ExcludeRegion	uint16[] [20] [2]	O	Zones that should be excluded from the detection area, usually zones that might interfere with the algorithms. A detection area might have 0 or more excluded zones. Array of areas. For the definition of area, see DetectRegion. The maximum number depends on the capability set. For highway all-in-one device, up to 10 excluded areas are supported.	[, ...,]
++SizeFilter	SizeFilter	O	Physical size filter by default. If both the default object type and a specific filter are selected, choose the specific filter. Highway all-in-one device supports a maximum box and a minimum box.	SizeFilter
++ObjectFilter	object[]	O	The specific filter for each object. It is a number.	

+++ObjectType e	char[16]	O	Object type. For available values, see ObjectTypeEnum.	"Vechicle"
+++SizeFilter	SizeFilter	O	Size filter	
+++Enable	bool	O	Enable the size filter of the specific object or not	true
++DetectRegi on	uint16[20][2]	O	Detection area. Coordinates of the peaks of the polygon.	
++SpecialDete ctRegion	object[10]	O	Specifies the general service detection area and exclusion area. Up to 10 service types are supported.	
+++Type	char[16]	O	Which service scheme category does the detection area parameter suits. For available values, see VAClassEnum.	"Normal"
+++DetectReg ion	uint16[10][20][2]	O	Detection area The coordinates of the peaks is unified within the range of 0–8192.	
+++ExcludeRe gion	uint16[10][20][2]	O	Zones that should be excluded from the detection area, usually zones that might interfere with the algorithms. A detection area might have 0 or more excluded zones.	[, ...,]
+Rules	object[]	R	Analysis rule	
++Class	char[16]	R	Analysis category The values are identical with the service scheme categories of VideoAnalyseGlobal.	"ObjectDetect"
++Type	char[32]	R	Analysis rule type The values are identical with the rule types of ideoAnalyseRule.	"ObjectDetect"
++Name	char[128]	R	Rule name. Rule name of devices without presets must be different. For devices with presets, the rule name in the same preset must be different. The name of rules in different presets can overlap.	"GuardLine1"
++ObjectType s	char[16][16]	O	List of detected object types Note: Not all rules need the object type. It is used primarily for the earliest behavior algorithms. We recommend you add the parameter to rule config. Here it is not changed due to compatibility. It is identical with VideoAnalyseRule(Base). When only the object type in the	["Human", "Vehicle"]

			<p>array is detected, the array cannot be empty. See VideoAnalysModule.ObjectType When ObjectTypes is empty, or there is unknown filed in the array, no matter whether there is other element, it means the classifier is not enabled.</p> <p>If Unknown does not exist in the array, only specific object types will be detected.</p>	
++Config	object	R	For specific rules, see "VideoAnalyseRule(XXXX)"	
++Experience Config	object[128]	O	Experience database configuration array	
+++ID	char[64]	O	Experience database ID	"1"
+++Similarity	uint8	O	<p>Similarity threshold: The range is [1–100], and it is 67 by default.</p>	67
+++IsPositive	bool	O	<p>True or not: true: True false: False</p>	true
+MQConfig	char[4096]	O	MQ configurations. See PaaS protocol configuration center > Operator configuration. When the remote access type is RabbitMq or ActiveMq, try to acquire MQ from this field. ActiveMq is used for the exchange of sensitive information.	""
+IsRepeat	int	O	Whether repetition is allowed. 0 means R, and it is the default value. 1 means no.	0

Request Example

```
{
    "Task": {
        "TaskUserData": "asdvtgrffg",
        "SourceType": "StreamSource",
        "IsStart": true,
        "LocalStream": {
            "Channel": 0,
            "Subtype": 0
        },
        "RemoteStream": {
            "Protocol": "RTSP",
            "Path": "rtsp://10.35.114.145:554/dss/monitor/param?cameraid=AGfQv8TwA1ALIPO29LA9L2&substream=1&trackID=801",
        }
    }
}
```

```
"IP": "10.35.114.145",
"Port": 554,
"UserName": "admin",
"Password": "admin123",
"Channel": 0,
"Subtype": 0,
"ChannelId": "1000254$1$0$110"
},
"LocalVideoFile": {
    "FileID": "file-1234",
    "Path": "/xxxx/a.dav"
},
"LocalPictureFile": {
    "Files": [
        {
            "FileID": "file-1234",
            "Path": "/xxxx/a.jpg"
        },...{}]
},
"RemoteVideoFile": {
    "Protocol": "RTSP",
    "VideoAnalysisProcess": 66,
    "Path": "rtsp://192.168.1.1:554/xxxx/a.dav",
    "IP": "192.168.1.1",
    "Port": 554,
    "Channel": 1,
    "Subtype": 1,
    "Username": "admin",
    "Password": "123456",
    "FileID": "file-1234",
    "StartTime": "2010-05-25 00:00:00",
    "EndTime": "2010-05-25 23:59:59"
},
"OffLineVideoFile": {
    "Protocol": "RTSP",
    "FileSize": 1000,
    "VideoAnalysisProcess": 66,
    "Path": "rtsp://192.168.1.1:554/xxxx/a.dav",
    "IP": "192.168.1.1",
    "Port": 4000,
    "Username": "admin",
    "Password": "123456",
    "FileID": "file-1234",
    "isReportPTS": true
},
"RemotePictureFile": {
    "Protocol": "HTTP",
    "IP": "192.168.1.1",
```

```
"Port": 80,
"Username": "admin",
"Password": "123456",
"Files": [
    {
        "FileID": "file-1234",
        "Path": "http://192.168.1.1:80/xxxx/a.jpg"
    },...{}]
},
"Global": {
    "Lanes": [
        {
            "Enable": true,
            "Number": 0,
            "LeftLine": ,
            "LeftLineExt": ,
            "LeftLineType": "WhiteSolid",
            "RightLine": ,
            "RightLineExt": ,
            "RightLineType": "WhiteSolid"
        },...{}],
    "CalibrateArea": [
        {
            "Area": ,
            "Staffs": [
                {
                    "Type": "Horizontal",
                    "Start": [0, 0],
                    "End": [100, 100],
                    "Length": 1.0
                },...{}]
        },...{}],
    "GlobalDetectionEnable": false,
    "ObjectArea": [
        {
            "Area": ,
            "Type": "SignalLight"
        },...{}],
    "CameraHeight": 6.2
},
"Module": {
    "ExcludeRegion": [, ..., ],
    "SizeFilter": SizeFilter,
    "ObjectFilter": [
        {
            "ObjectType": "Vechicle",
            "SizeFilter": ,
            "Enable": true
        },...{}],
    "DetectRegion": ,
    "SpecialDetectRegion": [
        {
            "Type": "Normal",
            "DetectRegion": ,
            "ObjectFilter": [
                {
                    "ObjectType": "Vechicle"
                }
            ]
        }
    ]
}
```

```

        "ExcludeRegion": [, ...],
    },...{}]
},
"Rules": [
    "Class": "ObjectDetect",
    "Type": "ObjectDetect",
    "Name": "GuardLine1",
    "ObjectTypes": ["Human", "Vehicle"],
    "Config": {
    },
    "ExperienceConfig": [
        "ID": "1",
        "Similarity": 67,
        "IsPositive": true
    },...{}]
},...{}],
"MQConfig": "",
"IsRepeat": 0
}
}

```

Response Params (JSON format in body)

Name	Type	R/O	Description	Example
TaskID	uint32	O	Task ID, > 0. (If it is failed, the result is false)	1234
VirtualChannel	uint32	O	The corresponding virtual channel number of the task. The value range is within the values of ChannelManager.getVirtualChannels.	1234
Path	char[256]	O	RTSP address of the AI stream	"rtsp://192.168.1.1:554/xx"

Response Example

```
{
    "TaskID": 1234,
    "VirtualChannel": 1234,
    "Path": "rtsp://192.168.1.1:554/xx"
}
```

9.9.2 Attach task result

Subscribe to all results (do not specify TaskID), or subscribe to specific TaskID. If you want to subscribe to the results of newly created tasks, to avoid event loss, we recommend you not set immediate start when creating the tasks. You can subscribe to the results first, and then run the start command.

Request URL	http://<server>/cgi-bin/api/analyseTaskManager/attachResult
Method	POST
Request Params (JSON format in body)	

Name	Type	R/O	Description	Example
TaskID	uint[64]	O	Subscribed TaskID When it is not entered, or the array is empty, the results of all tasks are subscribed to.	[1, 2, 3]
Filters	object	O	Filter conditions for subscription	
+Events	char[64][32]	O	Filtered event	["FaceDetection", ...]
+ImageDataFlag	uint32	O	0 means containing pictures. It is the default value. 1 means not containing pictures.	0
+ImageDataType	char[16][16]	O	Optional. It is used to combine the images of the event. One-dimensional array. If "ImageData Type" does not exist, it means reporting all the pictures of the event. If there is an element in the array, it reports pictures based on the element requirements. The value of each element: 1. "ObjectImage": Reporting target cutout 2. "SceneImage": Reporting scene image	["ObjectImage", "SceneImage"]
Heartbeat	int	O	Server keep-alive; integer; Unit: s; The value range is [1–60]. For example, if the parameter exists in the URL and the value is 5, the device will send "heartbeat" to the client as a keep-alive message. Note: The keep-alive message must be sent before the keepalive parameter expires.	5

Request Example

```
{
  "TaskID": [ 1, 2, 3 ],
  "Filters": {
    "Events": [ "FaceDetection", ... ],
    "ImageDataFlag": 0,
    "ImageData Type": ["ObjectImage", "SceneImage"],
    "Heartbeat": 5
  }
}
```

Response Params (JSON format in body)

Name	Name	Name	Name	Name
SID	uint32	R	Subscribed SID	1234U

Infos	object[]	O		
+TaskID	uint	O	Task ID	1
+TaskUserData	char[512]	O	Task data	"asdvtgrffg"
+UserData	char[64]	O	Video source data, including the video source information,.and it corresponds to the UserData field in addPollingTask.	"asdvtgrffg"
+TaskCustomData	TaskCustomData	O	Custom data	
+UserDefineData	char[512]	O	Custom data by the user, and it corresponds to the UserDefineData field in analyseTaskManager.analysePushPictureFileByRule.	
+FileID	char[128]	O	File ID. Valid when analyzing the file, and offered when a file analysis task is complete.	"file-1234"
+FileState	int	O	File analysis status: 0: Analyzing 1: Complete 2: Failed	0
+FileAnalyseMsg	char[256]	O	Analyzing additional information, usually the failure reason.	"Decode Failed, Download Failed"
+Events	object[]	O	Event information, such as TrafficJunction, HumanTrait and more. For the format, see EventInfo in EventManager. Offered when the analysis results come out. Simultaneously reports up to 32 events.	
+Flag	uint32	O	Tag status: 1: Unprocessed 2: Valid 3: Invalid	2
+Image	object	O	Linked snapshots	
++Offset	uint	O	Offset of images in binary data	0
++Length	uint	O	Image length (in bytes)	60000
Response Example				
HTTP/1.1 200 OK				
Content-Type: multipart/x-mixed-replace; boundary=<boundary>				
Connection: close				
--<boundary>				
Content-Type: application/json				
Content-Length: <data length>				

```
{
  "SID": "1234U",
  "Infos": [
    {
      "TaskID": 1,
      "TaskUserData": "asdvtgrffg",
      "UserData": "asdvtgrffg",
      "TaskCustomData": null,
      "FileID": "file-1234",
      "FileState": 0,
      "FileAnalyseMsg": "Decode Failed, Download Failed",
      "Events": [],
      "Flag": 2,
      "Image": {
        "Offset": 0,
        "Length": 60000
      }
    }
  ]
}

--<boundary>
Content-Type: text/plain
Content-Length: 11
Heartbeat
--<boundary>
...

```

9.9.3 Remove task

Request URL	http://<server>/cgi-bin/api/analyseTaskManager/remove		
Method	POST		
Request Params (JSON format in body)			
Name	Type	R/O	Description
TaskID	uint32	R	Task ID
Request Example			
{	{ "TaskID": 1 }		

Response Params (JSON format in body)
Name
Type
R/O
Description
Response Example
{}

9.9.4 Push Picture File

Valid when data source type is PushPictureFile. support pushing multiple files at the same time, each file should be completed, not support chunk

Request URL	http://<server>/cgi-bin/api/analyseTaskManager/analysePushPictureFile		
Method	POST		
Request Params (JSON format in body)			
Name	Type	R/O	Description
TaskID	uint	R	Task ID
Files	object[]	R	File list URL path and binary data are supported, but only one way can be selected at a time
+FileID	char[128]	R	File ID
+Url	char[512]	O	Remote file URL , with the necessary information for access, including username, password
+Offset	uint	O	Offset in binary data, bytes
+Length	uint	O	File size, bytes
+XRayCustomInfo	XRayCustomInfo	O	Customer information, X-ray machine customization
+ModelPath	char[512]	O	Algorithm model path
Request Example			
<pre>{ "TaskID": 1, "Files": [{ "FileID": "file-1234", "Url": "ftp://username:password@hostname:port/filepath", "Offset": 0, "Length": 256000, "XRayCustomInfo": , "ModelPath": "/aaa/bbb/ccc" }, ...] }</pre> <p>--<boundary> Content-Type: image/jpeg Content-Length:<image size> <JPEG image data> --<boundary></p>			

Response Params (JSON format in body)			
Name	Type	R/O	Description
Response Example			
{}			

9.10 SceneModeManager

9.10.1 Get Scene Mode Capabilities

Get Scene Mode Capabilities

Request URL	http://<server>/cgi-bin/api/SceneModeManager/getCaps		
Method	POST		
Request Params (JSON format in body)			
Name	Type	R/O	Description
DeviceID	char[128]	O	Device ID. For cascading scenarios.
Channel	int32	O	Video channel number. If -1, query all channels. If this field is not available, query the device scenario mode.
Request Example			
{ "DeviceID": "12345678", "Channel": 0 }	HTTP API V35 Examples		

Response Params (JSON format in body)			
Name	Type		
caps	object[1024]		
+SupportedMode	char[32][32]		
Response Example			
{ "caps": [{"SupportedMode": ["AcuPick"], "Channel": 0 },...{}] }	HTTP API V35 Examples		

9.10.2 Get The Current Scene Mode

Get the current scene mode

Request URL	http://<server>/cgi-bin/api/SceneModeManager/getMode		
Method	POST		
Request Params (JSON format in body)			

Name	Type	R/O	Description	Example
Channel	int32	O	Video channel number. If - 1, query all channels. If this field is not available, query the device scenario mode.	0
DeviceID	char[128]	O	Device ID. For cascading scenarios.	"12345678"

Request Example

```
{
  "Channel": 0,
  "DeviceID": "12345678"
}
```

Response Params (JSON format in body)				
Name	Type	R/O	Description	Example
infos	object[]	O		
+Channel	int32	O	Video channel number. If this field is not available, query the device scenario mode.	0
+Mode	char[32]	R	Scene Mode.	"None"

Response Example

```
{
  "infos": [
    {
      "Channel": 0,
      "Mode": "None"
    }, ...
  ]
}
```

9.10.3 Get The Default Scene Mode

Get the default scene mode

Request URL	http://<server>/cgi-bin/api/SceneModeManager/getDefault			
Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
Channel	int32	O	Video channel number. If - 1, query all channels. If this field is not available, query the device scenario mode.	0
DeviceID	char[128]	O	Device ID. For cascading scenarios.	"12345678"

Request Example

```
{
  "Channel": 0,
  "DeviceID": "12345678"
}
```

Response Params (JSON format in body)

Name	Type	R/O	Description	Example
infos	object[]	O	The content is the same as SceneModeManager.getMode	

Response Example

```
{
  "infos": [
    {}...{}
  ]
}
```

9.10.4 Set The Scene Mode

Set the scene mode. To check the mode settings and whether the device needs to be restarted, please use the DevVariableCapsManager.checkStart.

Request URL <http://<server>/cgi-bin/api/SceneModeManager/setMode>

Method POST

Request Params (JSON format in body)

Name	Type	R/O	Description	Example
infos	object[1024]	O	The content is the same as return of "SceneModeManager.getMode".	
DeviceID	char[128]	O	Device ID. For cascading scenarios.	"12345678"

Request Example

```
{
  "infos": [
    {}...{},
    "DeviceID": "12345678"
}
```

Response Params (JSON format in body)

Name	Type	R/O	Description	Example
------	------	-----	-------------	---------

Response Example

```
{}
```

9.11 ExperienceRegistry

9.11.1 Acquiring Experience Database Capabilities

Acquire the capabilities of the experience database.

Request URL	<a href="http://<server>/cgi-bin/api/ExperienceRegistry/getCaps">http://<server>/cgi-bin/api/ExperienceRegistry/getCaps			
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
GroupTypeList	object[16]	O	Information on the supported function databases.	o
+GroupType	enumchar[32]	O	<p>The type of the function databases.</p> <pre>enumchar[32]{ "FilterEvent": The sample database of filtered events (enter negative samples for the sub-attribute when it is false alarm database). If filtered events for positive samples are supported later, sub-attribute can be extended to positive samples. }</pre>	"FilterEvent"
+MaxGroupNum	uint32	O	The maximum number of groups supported by the current database.	10
+PicStorageNum	uint32	O	The maximum number of images that can be stored in the current database (the sum of images in all groups).	1,000
+ImportSmallNumber	uint32	O	The maximum number of small images that can be imported at a time.	50
+ImportBigNumber	uint32	O	The maximum number of panoramic images that can be imported at a time.	40
+SinglePicMaxLength	int32	O	The largest size of a single panoramic image	4096

			that can be imported. Unit: KB. -1 means there is no size limit.	
+SupportDetectWay	int8	O	Supported import methods. Each binary digit represents different functions. 1 (0001B): Supports importing large images (cutouts are needed); 2 (0010B): Supports importing small images (cutouts are not needed); 3 (0011B): A combination of 0001B and 0010B. It supports importing large and small images.	2
+SupportedScenes	object[64]	O	Supported business categories. If the field does not exist, it means databases cannot be created by category.	
++ClassType	char[32]	O	Solution types.	"Normal"
++ClassAlias	char[32]	O	Class alias. If the field does not exist or if it is "", it means there is no alias.	""
++Subclasses	bool	O	Whether there are function flag bits. The client determines whether to request sub-functions according to this flag bit. This field is used if there is no SupportedRules.	false
++FuncList	enumchar[16][32]	O	List of sub-functions. enumchar[16][32]{ "Positives": Positive samples. "Negative": Negative samples. }	["Positives", "Negative"]

++SupportedRules	object[16]	O	The types of rules that can be created. If the field does not exist, it means that databases cannot be created by rule.	
+++RuleType	char[32]	O	Rule types.	"CrossLineDetection"
+++EventType	char[32]	O	The types of events corresponding to the rules.	"CrossLineDetection"
+++Subclasses	bool	O	Whether there are function flag bits. The client determines whether to request sub-functions according to this flag bit. This field is used if there is no SupportedObjects.	false
+++FuncList	enumchar[16][32]	O	List of sub-functions. enumchar[16][32]{ "Positives": Positive samples. "Negative": Negative samples. }	["Positives", "Negative"]
+++SupportedObjects	object[16]	O	List of detection target types. If the field does not exist, it means that it is not supported.	
++++ObjectType	enumchar[32]	O	Types of detection targets.	"Human"
++++Subclasses	bool	O	Whether there are function flag bits. The client determines whether to request sub-functions according to this flag bit.	false
++++FuncList	enumchar[16][32]	O	List of sub-functions.	["Positives", "Negative"]

			enumchar[16][32]{ "Positives": Positive samples. "Negative": Negative samples. }	
+SupportedChannel	bool	O	Whether registered database can be created by channel. true: supported; false: not supported. It is not supported by default.	true
+SupportDisposition	bool	O	<p>Whether channel arming is supported. true: supported; false: not supported. It is not supported by default.</p> <p>Note: If the registered database is created with a channel ID, the channel ID that is used when the channel is armed must be the same.</p>	true

Response Example

```
{
  "GroupTypeList": [
    {
      "GroupType": "FilterEvent",
      "ImportSmallNumber": 50,
      "ImportBigNumber": 50,
      "SinglePicMaxLength": 50,
      "MaxGroupNum": 10,
      "PicStorageNum": 1000,
      "SupportDetectWay": 2,
      "SupportedChannel": true,
      "SupportDisposition": true,
      "SupportedScenes": [
        {
          "Scene": "Scene1"
        }
      ]
    }
  ]
}
```

```

    "ClassAlias": "",

    "ClassType": "Normal",

    "FuncList": [
        "Negative"
    ],
    "Subclasses": true,
    "SupportedRules": [
        {
            "RuleType": "CrossLineDetection",
            "EventType": "CrossLineDetection",
            "FuncList": [
                "Negative"
            ],
            "Subclasses": true,
            "SupportedObjects": [
                {
                    "ObjectType": "Human",
                    "FuncList": [
                        "Negative"
                    ],
                    "Subclasses": true
                }
            ]
        }
    ]
}

```

9.11.2 Creating Experience Database

Create the experience database.

Request URL	http://<server>/cgi-bin/api/ExperienceRegistry/createGroup
Method	POST

Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
Group	object	R	Information on the experience database.	
+GroupName	char[128]	R	Name of the experience database.	Express
+GroupDetail	char[256]	O	Remarks on the database.	"Community Homeowners"
+SampleAttribute	uint8	O	Sample attributes. 1: Positive samples; 2: Negative samples.	1
+GroupType	enumchar[32]	O	The type of the function database. enumchar[32]{ "FilterEvent": Sample database of filtered events (including positive sample database and negative sample database). }	"FilterEvent"
+Channel	int	O	The channel of the group starting from 0. If this field does not exist or if it is -1, it means that it is not dependent on the channel.	0
+ClassType	char[32]	O	Solution types.	"Normal"
+RuleType	char[32]	O	Rule types	"CrossLineDetection"
+ObjectType	char[32]	O	Types of supported detection targets.	"Human"
+Disposition	object[64]	O	Information on the channel arming.	
+Enable	bool	O	Whether the channel is armed. true: armed; false: not armed.	false
+DeployedChann	int16	O	The ID of the video	0

el			channel to which the current group is bound. It starts from o.	
+Similarity	uint8	o	The similarity threshold of the feature comparison corresponding to the bound video channel.	90

Request Example

```
{
  "Group": {
    "GroupName": "z1",
    "GroupDetail": "11",
    "SampleAttribute": 1,
    "GroupType": "FilterEvent",
    "Channel": o,
    "ClassType": "Normal",
    "RuleType": "RuleType",
    "ObjectType": "ObjectType",
    "Disposition": [
      {
        "DeployedChannel": o,
        "Similarity": 90,
        "Enable": false
      }
    ]
  }
}
```

Response Params (JSON format in body)				
Name	Type	R/O	Description	Example
GroupId	char[64]	R	Experience database ID.	"11"
Response Example				
<pre>{ "GroupId": "34" }</pre>				

9.11.3 Searching for Group Information on Experience Database

Search for the group information on the experience database.

Request URL	http://<server>/cgi-bin/api/ExperienceRegistry/findGroup			
Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
+GroupId	char[64][64]	No	Group ID of the experience database. If the field is empty or not filled in, it means searching information on all groups.	["oooooo1"]
+GroupType	enumchar[32]	No	The type of the function database.	"FilterEvent"
Request Example				
<pre>{ "GroupId": [], "GroupType": "FilterEvent" }</pre>				

Response Params (JSON format in body)				
Name	Type	R/O	Description	Example
GroupList	object[64]	O	Information list of the experience database.	
+GroupName	char[128]	O	Name of the experience database.	Express
+GroupDetail	char[256]	O	Remarks on the database.	"Community Homeowners"
+SampleAttribute	uint8	O	Sample attributes. 1: Positive samples; 2: Negative samples.	1
+GroupType	enumchar[32]	O	The type of the function database. enumchar[32]{	"FilterEvent"

			"FilterEvent": Sample database of filtered events (including positive sample database and negative sample database). }	
+Channel	int	O	The channel of the group. It starts from 0. If this field does not exist or if it is -1, it means that it is not dependent on the channel.	0
+ClassType	char[32]	O	Solution types.	"Normal"
+RuleType	char[32]	O	Rule types.	"CrossLineDetection"
+ObjectType	char[32]	O	Types of supported detection targets.	"Human"
+Disposition	object[64]	O	Information on the channel arming. Its size depends on the capabilities of the video channel.	
+Enable	bool	O	Whether the channel is armed. true: armed; false: not armed.	false
+DeployedChannel	int16	O	The ID of the video channel to which the current group is bound. It starts from 0.	0
+Similarity	uint8	O	The similarity threshold of the feature comparison corresponding to the bound video channel.	90
Response Example				
{ "GroupList": [{ "GroupName": "z1", "GroupType": "CrossLineDetection", "GroupID": 1, "GroupOrder": 1, "GroupStatus": "Normal", "GroupRule": "CrossLineDetection", "GroupObject": "Human", "GroupChannel": 0, "GroupDisposition": { "GroupArmed": false }, "GroupDeployedChannel": 0, "GroupSimilarity": 90 }] }				

```

    "GroupDetail": "11",
    "SampleAttribute": 1,
    "GroupType": "FilterEvent",
    "Channel": 0,
    "ClassType": "Normal",
    "RuleType": "RuleType",
    "ObjectType": "ObjectType",
    "Disposition": [
        {
            "DeployedChannel": 0,
            "Similarity": 90,
            "Enable": false
        }
    ]
}

```

9.11.4 Deleting Information on Experience Database Groups

Deleting the information on the experience database groups.

Request URL	http://<server>/cgi-bin/api/ExperienceRegistry/deleteGroup			
Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
GroupID	char[64]	R	Experience database group ID.	"1"
Request Example				
{ "GroupID": "1" }				

Response Params (JSON format in body)				
Name	Type	R/O	Description	Example
Response Example				

{}

9.11.5 Modifying Information on Experience Database Groups

Modify the information on the experience database groups.

Request URL	http://<server>/cgi-bin/api/ExperienceRegistry/modifyGroup			
Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
Group	object	R	Information on the experience database.	
+GroupId	char[64]	R	Experience database ID.	"0001"
+GroupName	char[128]	O	Name of the experience database.	Express
+GroupDetail	char[256]	O	Remarks on the database.	"Community Homeowners"
+Disposition	object[64]	O	Information on the channel arming. If this field does not exist or is empty, it means not modifying the arming information.	
+Enable	bool	O	Whether the channel is armed. true: armed; false: not armed.	false
+DeployedChannel	int16	O	The ID of the video channel to which the current group is bound. It starts from 0.	0
+Similarity	uint8	O	The similarity threshold of the feature comparison corresponding to the bound video channel.	90
Request Example				
{				
"Group": {				

```

    "GroupID": "1",
    "GroupName": "zl",
    "GroupDetail": "11",
    "Disposition": [
        {
            "DeployedChannel": 0,
            "Similarity": 90,
            "Enable": false
        }
    ]
}

```

Response Params (JSON format in body)				
Name	Type	R/O	Description	Example
Response Example				
{}				

9.11.6 Searching for Remaining Storage Size of the Experience Database

Request URL	http://<server>/cgi-bin/api/ExperienceRegistry/getGroupSpaceInfo			
Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
GroupType	enumchar[32]	O	The type of the function database.	"FilterEvent"
Request Example				
{				
	"GroupType": "FilterEvent"			
}				

Response Params (JSON format in body)				
Name	Type	R/O	Description	Example

TotalSize	uint32	O	The total storage size (MB) of the experience database.	11
FreeSize	uint32	O	The remaining storage size (MB) of the experience database.	11
TotalCount	uint32	O	The total number of items that can be imported to the experience database.	11
FreeCount	uint32	O	The remaining number of items that can be imported to the experience database.	11

Response Example

```
{
    "TotalSize": 11,
    "FreeSize": 11,
    "TotalCount": 11,
    "FreeCount": 11
}
```

9.11.7 Acquiring Additional Tokens

Acquire additional tokens to use them with multiAppend.

Request URL	http://<server>/cgi-bin/api/ExperienceRegistry/getDetectToken			
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
Token	uint	O	Detection tokens.	1

Response Example
{ "Token": 17 }

9.11.8 Adding Multiple Experience Database Samples Asynchronously

Add multiple experience database samples asynchronously to use them with getDetectToken.

Request URL	http://<server>/cgi-bin/api/ExperienceRegistry/multiAppend			
Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
PicInfo	object[128]	O	Information on multiple images.	
+GroupId	char[64]	R	Experience database ID.	"1"
+BigID	char[64]	O	Unique ID of the large image.	"1"
+BigName	char[64]	O	Name of the large image.	"aaaaa"
+SmallName	char[64]	O	Name of the small image.	"aaaaa"
+ObjectRect	Rect[32]	O	The coordinate arrays of the target rectangular. The value ranges from 0 to 8191.	[0,0,100,100]
+Image	object	O	Image information.	
++Width	uint16	O	Image width.	100
++Height	uint16	O	Image height.	50
++Offset	uint32	O	Offset in the binary data blocks.	0
++Length	uint32	O	Image size.	100000
++FilePath	char[256]	O	File path. 1-Supports HTTP URL: "http://www.dahua.com/ 1.jpg" 2-Supports: FTP URL:	"/mnt/2010/8/1/dav/15:4 0:50.jpg"

			<p>"ftp://ftp.dahua.com/1.jpg"</p> <p>3-Supports local path of the server:</p> <p>a) "C:/pic/1.jpg"</p> <p>b)</p> <p>"/mnt/2010/8/1/dav/15:40:50.jpg"</p> <p>It conflicts with Offset and Length.</p>	
Info	object	O	Status flag.	
+Token	uint16	O	The token of the image import.	++Token
+State	bool	O	Whether the image import request is finished. true: finished. false: not finished.	++State
+DetectWay	int32	O	1: Import large images; 2: import small images.	++DetectWay
Request Example				
Content-Type: application/json Content-Length: <data length> { "PicInfo": [{ "GroupID": "1", "BigID": "1", "BigName": "aaaaa", "Image": { "Width": 100, "Height": 50, "Offset": 0, "Length": 100000, "FilePath": "/mnt/2010/8/1/dav/15:40:50.jpg" }, "SmallName": "aaaaa", "ObjectRect": [[

```

        0,
        0,
        100,
        100

    ]
}

],
"Info": {
    "Token": 1,
    "State": false,
    "DetectWay": 2
}
}
--<boundary>
Content-Type: image/jpeg
Content-Length: <image size>
<Jpeg image data>
--<boundary>

```

Response Params (JSON format in body)

Name	Type	R/O	Description	Example
ErrCode	uint8[128]	O	Error code. { 1: Successful. 2: The database ID does not exist. 3: Incorrect parameters. 4: Failed to perform operation on the file. 5: Failed to download the image information. 6: Failed to download the feature information. 7: Failed to perform operation on the database. 8: Unknown error. 9: The quantity exceeds	[1, 2]

			<p>the limit.</p> <p>10: Insufficient storage space.</p> <p>}</p>	
Response Example				
{	"ErrCode": [1,	2]

9.11.9 Searching for Image Information on the Experience Database

Search for the image information on the experience database.

Request URL	http://<server>/cgi-bin/api/ExperienceRegistry/startFind			
Method	POST			
Request Params (SON format in body)				
Name	Type	R/O	Description	Example
Condition	object	R	Search conditions.	
+GroupId	char[128][64]	R	List of database group IDs.	["00001","00002"]
+FeatureState	uint8[5]	O	<p>The status of feature values.</p> <p>0: Unknown.</p> <p>1: Failed to model. The images might not meet the requirements and need to be replaced.</p> <p>2: There are feature values that can be used. If the feature values of one image has been calculated, it is enough.</p> <p>3: Feature values are being calculated. The actual calculation process only takes tens of milliseconds. It is mainly</p>	[1,2]

			<p>used for images that are placed after the batch queue when the feature values are reconstructed in batches. When there are no feature values, the program will switch to this status.</p> <p>4: Modeling is complete, but the data cannot be used due to algorithm update. Remodeling is needed.</p>	
--	--	--	---	--

Request Example

```
{
  "Condition": {
    "GroupID": ["00001"]
  },
  "FeatureState" : [1,4]
}
```

Response Params (JSON format in body)				
Name	Type	R/O	Description	Example
Token	uint	O	Acquired search token.	2342343
TotalCount	uint64	O	Total number of results that meet the search conditions.	3333
Response Example				
{				
	"Token": 2342343,			
	"TotalCount": 3333			
}				

9.11.10 Reading Search Results of Experience Database Images

Read the search results of the experience database images.

Request URL	http://<server>/cgi-bin/api/ExperienceRegistry/doFind
Method	POST

Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
Condition	object	R	Input parameters of the search.	
+Token	uint	R	Search token.	46878
+BeginNumber	uint	R	Starting sequence number of the search. It means that the search starts from the beginNumber records, and returns count records. o<beginNumber<=totalCount-1	0
+Count	int	R	Number of entries acquired each time.	20

Request Example

```
{
  "Token": 46878,
  "BeginNumber": 0,
  "Count": 20
}
```

Response Params (JSON format in body)				
Name	Type	R/O	Description	Example
Results	object	R	Output parameters of the search.	
+Found	int	R	Number of feature entries found this time.	12
+PicInfoList	object[1024]	O	List of searched feature information.	
++GroupID	char[64]	O	Database ID.	"12"
++SmallID	char[64]	O	Small image ID.	"1"
++SmallUrl	char[64]	O	URL of small image.	"/mnt/small.jpg"
++SmallImportTi	uint32	O	The UTC time when the	1634604948

me			small image was imported (unit: second).	
++FeatureState	uint8	O	<p>The status of feature values.</p> <p>0: Unknown.</p> <p>1: Failed to model. The images might not meet the requirements and need to be replaced.</p> <p>2: There are feature values that can be used. If the feature values of one image has been calculated, it is enough.</p> <p>3: Feature values are being calculated. The actual calculation process only takes tens of milliseconds. It is mainly used for images that are placed after the batch queue when the feature values are reconstructed in batches. When there are no feature values, the program will switch to this status.</p> <p>4: Modeling is complete, but the data cannot be used due to algorithm update. Remodeling is needed.</p>	1

Response Example

```
{
  "Results": {
    "Found": 12,
    "PicInfoList": [
      {
        "GroupID": "12",
        "SmallIID": "1",
        "SmallUrl": "/mnt/small.jpg",
        "SmallImportTime": 1634604948,
      }
    ]
  }
}
```

```

        "FeatureState": 2
    }
]
}
}

```

9.11.11 Stopping Searching for Image Information on Experience Database

Stop searching for the image information on the experience database.

Request URL	http://<server>/cgi-bin/api/ExperienceRegistry/stopFind			
Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
Token	uint	R	Search token.	46878
Request Example				
<pre>{ "Token": 46878 }</pre>				

Response Params (JSON format in body)				
Name	Type	R/O	Description	Example
Response Example				
{}				

9.11.12 Deleting Image Information on the Experience Database

Delete the image information.

Request URL	http://<server>/cgi-bin/api/ExperienceRegistry/deleteByFeatureID			
Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
GroupId	char[64]	O	Database ID.	"5"

SmallID	char[100][64]	O	Unique ID of the small image.	["123", "234"]
Request Example				
{				
"GroupID": "1",				
"SmallID": [
"123",				
"234"				
]				
}				
Response Params (JSON format in body)				
Name	Type	R/O	Description	Example
+ErrCode	uint[100]	O	Error code which indicates the deleting result. 1: Successful. 2: The feature ID does not exist. 3: Failed to perform operation on the database. 4: Failed to update the arming database. 5: Failed to perform operation on the storage service. 6: The image does not exist.	[1,2]
Response Example				
{				
"ErrCode": [
1,				
2				
]				
}				

9.11.13 Subscribing Feature Extraction Results

Subscribe to feature extraction progress and use it with multiAppend, reAbstract and groupReAbstract.

Request URL	http://<server>/cgi-bin/api/ExperienceRegistry/attachFeatureState			
Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
Tokens	uint[32]	O	Search token. If this field does not exist, it means subscribe all tokens.	[23, 25]
Heartbeat	int	O	<p>The keep-alive time of the server keep-alive. It is an integer (unit: second). The value range is [1,60]. For example, if URL comes with this parameter and the value is 5, it means that the device sends a keep-alive message to the client every 5 seconds, and the keep-alive message is "Heartbeat".</p> <p>Note: The keep-alive message must be sent before the keep-alive time expires.</p>	5
Request Example				
<pre>HTTP/1.1 200 OK Content-Type: multipart/x-mixed-replace; boundary=<boundary> Connection: close --<boundary> Content-Type: application/json Content-Length: <data length> { "SID": 1234U, "Infos": [{ "Feature": "Face Detection", "Status": "PENDING" }, { "Feature": "Object Detection", "Status": "PENDING" }] }</pre>				

```

"TaskID": 1,
"TaskUserData": "asdvtgrffg",
"UserData": "asdvtgrffg",
"TaskCustomData": null,
"FileID": "file-1234",
"FileState": 0,
"FileAnalyseMsg": "Decode Failed, Download Failed",
"Events": [{}],
"Flag": 2,
"Image": {
    "Offset": 0,
    "Length": 6000
}
}
]
}
--<boundary>
Content-Type: text/plain
Content-Length: 11
Heartbeat
--<boundary>

```

Response Params (JSON format in body)				
Name	Type	R/O	Description	Example
SID	uint32	R	Subscription ID.	1234
info	object	R	The pushed data content.	
+Progress	uint8	O	Modelling progress [0,100]	34
+Token	uint32	O	Subscription token.	1
+SucceedCnt	uint	O	The number of items that are successfully modeled.	100
+FailedCnt	uint	O	The number of items that failed to be modeled.	20
+FailedPicDetails	object[100]	O	The details of the images	

			failed to be modelled in the current sending progress.	
++BigName	char[64]	No	The name of the image to be imported.	"aaaaa"
++SmallName	char[64]	O	Name of the small image.	"aa_1.jpg"
++ErrCode	uint8	O	Error code. { 1: Modeling failed. 2: System errors (for example, errors caused by invalid license and unstated modeling analyzer). 3: Failed to perform operation on the database. 4: The number of image exceeds the limit. 5: The remaining storage space is insufficient. 6: Failed to decode the image. 7: Failed to verify the image resolution. 8: No targets were detected. 9: Failed to write the image files. 10: Failed to download the images through HTTP. 11: Requests other than HTTP are not supported. }	1

Response Example

HTTP/1.1 200 OK
 Content-Type: multipart/x-mixed-replace; boundary=<boundary>
 Connection: close
 --<boundary>

```

Content-Type: application/json
Content-Length: <data length>
{
    "SID": 123,
    "info": {
        "Progress": 34,
        "Token": 1,
        "SucceedCnt": 100,
        "FailedCnt": 20,
        "FailedPicDetails": [
            {
                "SmallName": "aaaaa",
                "BigName": "aaaaa",
                "ErrCode": 1
            }
        ]
    }
}
--<boundary>
Content-Type: text/plain
Content-Length: 11
Heartbeat
--<boundary>

```

HTTP API V3.35 - Intelbras

9.11.14 Remodeling Experience Database

The feature vectors of the images are stored in the database after the calculation of the algorithm database. The feature vectors need to be reconstructed after the algorithm is updated. Non-blocking interface used with attachFeatureState.

Request URL	http://<server>/cgi-bin/api/ExperienceRegistry/reAbstract			
Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
GroupID	char[64]	O	Database ID.	"ooooo1"
PicInfoList	object[100]	O	Information on multiple images.	

+SmallID	char[64]	O	Reconstruct by the unique ID of the small image.	"120837"
Request Example				
{				
"GroupID": "00001",				
"PicInfoList": [
{				
"SmallID": "111"				
}				
]				
}				

Response Params (JSON format in body)				
Name	Type	R/O	Description	Example
Token	uint32	O	The token of the image modeling.	1
Response Example				
{				
"Token": 1				
}				

9.11.15 Modeling All Items in the Experience Databases

The functions are the same as that of reAbstract interface. This is to remodel all items by database.

Request URL	http://<server>/cgi-bin/api/ExperienceRegistry/groupReAbstract			
Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
GroupID	char[64][64]	O	List of group IDs. There must be at least one group ID.	["00001"]
Request Example				
{				
"GroupID": [

```

        "oooo01"
    ]
}

```

Response Params (JSON format in body)				
Name	Type	R/O	Description	Example
Token	uint32	O	The token of the image modeling.	1
Response Example				
{				
	"Token": 1			
}				

9.11.16 Unbinding Experience Databases

Remove all rule binding information of the database/

Request URL	http://<server>/cgi-bin/api/ExperienceRegistry/unbind			
Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
GroupID	char[64][64]	O	List of group IDs. There must be at least one group ID.	["oooo01"]
Request Example				
{				
	"GroupID": [
	"oooo01"			
]			
}				

Response Params (JSON format in body)				
Name	Type	R/O	Description	Example
Response Example				
{}				

9.11.17 Searching for Records of Suspected False Alarm Events

Search for the records of suspected false alarm events.

Request URL	http://<server>/cgi-bin/api/ExperienceRegistry/startFindFilterEvent			
Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
Condition	object	O	Search conditions.	
+StartTime	char[20]	O	Start time.	"2012-01-04 00:00:00"
+EndTime	char[20]	O	End time.	"2012-01-04 23:59:59"
Request Example				
{ "Condition": { "StartTime": "2012-01-04 00:00:00", "EndTime": "2012-01-04 23:59:59" } }				

Response Params (JSON format in body)				
Name	Type	R/O	Description	Example
Token	uint32	R	Acquired search token.	123
TotalCount	uint32	R	Total number of results that meet the search conditions.	23
Response Example				
{ "Token": 123, "TotalCount": 23 }				

9.11.18 Reading Search Results for Suspected False Alarm Events

Read the search results for the suspected false alarm events.

Request URL	http://<server>/cgi-bin/api/ExperienceRegistry/doFindFilterEvent
--------------------	--

Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
Condition	object	R	Input parameters of the search.	
+Token	uint32	R	Search token.	123
+BeginNumber	uint32	R	Starting sequence number of the search. It means that the search starts from the beginNumber records, and returns count records. 0<=BeginNumber<= TotalCount-1	0
+Count	uint32	R	Number of entries acquired each time.	20
Request Example				
<pre>{ "Condition": { "Token": 123, "BeginNumber": 0, "Count": 20 } }</pre>				

Response Params (JSON format in body)				
Name	Type	R/O	Description	Example
Results	object	O	Output parameters of the search.	
+Found	uint32	O	Total number of entries found this time.	12
+Info	object[1000]	O	The suspected false alarm events that were found.	
++EventType	char[32]	O	Event type.	"CrossLineDetection"

++ClassType	char[32]	O	Category of the AI event.	"Normal"
++UTC	uint32	O	The UTC time when the event occurred with zone deviation (unit: second).	152463285
++RuleType	char[32]	O	Rule types.	"CrossLineDetection"
Response Example				
{ "Results": { "Found": 12, "Info": [{ "EventType": "CrossLineDetection", "ClassType": "Normal", "UTC": 152463285, "RuleType": "CrossLineDetection" }] } }				

9.11.19 Stopping Searching for Records of Suspected False Alarm Events

Stop searching for the records of the suspected false alarm events.

Request URL	http://<server>/cgi-bin/api/ExperienceRegistry/stopFindFilterEvent			
Method	POST			
Request Params (JSON format in bod)				
Name	Type	R/O	Description	Example
Token	uint32	R	Search token.	123
Request Example				
{ "Token": 123 }				

Response Params (JSON format in body)				
Name	Type	R/O	Description	Example
Response Example				
{}				

HTTP API V3.35 - Intelbras

HTTP API V3.35 - Intelbras

10.1 Intelligent Traffic Event

10.1.1 [Event] TrafficJunction

Event Code	TrafficJunction		
Event action	Start/Stop		
Event index	0		
Event Data			
Name	Type	R/O	Description
+GroupId	int	O	The id of event group.
+CountInGroup	int	O	Event count in the event group.
+IndexInGroup	int	O	The index of this event in the event group, start from 1.
+Lane	int	O	Lane number, start from 0.
+TriggerType	int	O	Trigger type. It can be: 0—car detector, 1—radar, 2—video.
+Speed	int	O	Vehicle speed, unit is km/hour.
+Vehicle	object	O	The information of vehicle object.
++BoundingBox	int[4]	R	The detected car bounding box, 4 interge, refer to x's value of left-top point, y's value of left-top point, x's value of right-bottom point, y's value of right-bottom point. Coordinate remap to 0 — 8192.
++Text	char[]	O	Vehicle logo.
++SubText	char[]	O	Vehicle sub logo.
++SubBrand	int	O	Vehicle sub brand index.
++BrandYear	int	O	Vehicle brand year index.
+TrafficCar	object	O	Traffic Car info.
++RecNo	int	R	The record id.
++PlateNumber	char[]	R	Car plate number.
++PlateType	char[]	O	Plate type.
++PlateColor	char[]	O	Plate color, ex: "Yellow", "Blue", ... etc.
++VehicleColor	char[]	O	Vehicle color, ex: "Yellow", "Blue", ... etc.
++BoundingBox	int[4]	R	The detected plate bounding box, 4 interge, refer to x's value of left-top point, y's value of left-top point, x's value of right-bottom point, y's value

			of right-bottom point. Coordinate remap to 0 — 8192.	
++Country	char[]	O	Country info. Max string length is 19.	
++Speed	int	O	Vehicle speed, unit is km/hour.	
++Event	char[]	O	The event info, ex: "TrafficJunction".	"TrafficJunction"
+CommInfo	object	O	Traffic event common info.	
++Seat	object[]	O	Vehicle front seat info.	
+++Type	char[]	O	Front seat type. It can be: "Main""Slave".	"Main"
+++Status	char[][]	O	Some driver status. It can be some of the following: "Smoking", "Calling".	
+++SunShade	string	O	Sunshade status. It can be: "Unknow", "WithSunShade", "WithoutSunShade".	"WithSunShade"
+++ShadePos	int[4]	O	The detected sun shade bounding box, 4 interge, refer to x's value of left-top point, y's value of left-top point, x's value of right-bottom point, y's value of right-bottom point. Coordinate remap to 0 — 8192.	
+++SafeBelt	char[]	O	SafeBelt status. It can be: "Unknow", "WithSafeBelt", "WithoutSafeBelt".	WithoutSafeBelt

Event Response Example (multipart , key=value format in body , binary data in body) (response to 4.4.3 snap subscription)

--<boundary>

Content-Type: text/plain

Content-Length: <length>

```

Events[0].EventBaseInfo.Code=TrafficJunction
Events[0].EventBaseInfo.Action=Pulse
Events[0].EventBaseInfo.Index=0
Events[0].GroupID=123
Events[0].CountInGroup=3
Events[0].IndexInGroup=1
Events[0].Lane=0
Events[0].Vehicle.BoundingBox[0]=1341
Events[0].Vehicle.BoundingBox[1]=2451
Events[0].Vehicle.BoundingBox[2]=4513
Events[0].Vehicle.BoundingBox[3]=4135
Events[0].Vehicle.Text=Audi
Events[0].Vehicle.SubText=A6L
Events[0].Vehicle.SubBrand=5

```

```

Events[0].Vehicle.BrandYear=2
Events[0].TrafficCar.RecNo=123
Events[0].TrafficCar.PlateNumber=AC00003
Events[0].TrafficCar.PlateColor=Yellow
Events[0].TrafficCar.VehicleColor=Blue
Events[0].TrafficCar.BoundingBox[0]=1341
Events[0].TrafficCar.BoundingBox[1]=2451
Events[0].TrafficCar.BoundingBox[2]=4513
Events[0].TrafficCar.BoundingBox[3]=4135
Events[0].TrafficCar.Country=China
Events[0].CommInfo.Seat[0].Type=Main
Events[0].CommInfo.Seat[0].Status[0]=Smoking
Events[0].CommInfo.Seat[0].SunShade=WithSunShade
Events[0].CommInfo.Seat[0].ShadePos[0]=2021
Events[0].CommInfo.Seat[0].ShadePos[1]=3041
Events[0].CommInfo.Seat[0].ShadePos[2]=2151
Events[0].CommInfo.Seat[0].ShadePos[3]=3661
Events[0].CommInfo.Seat[0].SafeBelt=WithoutSafeBelt
...
--<boundary>
Content-Type: image/jpeg
Content-Length: <image size>

<Jpeg image data>
--<boundary>

```

10.1.2 [Event] TrafficRetrograde

When detects vehicle retrograde, send this event.

Event params is the same as **TrafficJunction**, except for event Code is **TrafficRetrograde**.

10.1.3 [Event] TrafficJam

When detects traffic jam, send this event.

Event Code	TrafficJam			
Event action	Start/Stop			
Event index	0			
Event Data				
Name	Type	R/O	Description	Example
+GroupId	int	O	The id of event group.	123
+CountInGroup	int	O	Event count in the event group.	1
+IndexInGroup	int	O	The index of this event in the event group, start from 1.	1
+Lane	int	O	Lane number, start from 0.	0
+StartJaming	int	O	Start jam time, UTC seconds.	123456789
+AlarmInterval	int	O	Alarm interval, unit is second.	180

+JamLenght	int	O	Jam length, percentage of the lane.	70
+JamRealLength	int	O	Jam real length, unit is metre.	120

Event Response Example (multipart , key=value format in body , binary data in body) (response to 4.4.3 snap subscription)

--<boundary>

Content-Type: text/plain

Content-Length: <length>

Events[0].EventBaseInfo.Code=TrafficJam

Events[0].EventBaseInfo.Action=Pulse

Events[0].EventBaseInfo.Index=0

Events[0].GroupID=123

Events[0].CountInGroup=3

Events[0].IndexInGroup=1

Events[0].Lane=0

Events[0].StartJaming=123456789

Events[0].AlarmInterval=180

Events[0].JamLenght=70

Events[0].JamRealLength=120

...

--<boundary>

Content-Type: image/jpeg

Content-Length: <image size>

<Jpeg image data>

--<boundary>

10.1.4 [Event] TrafficUnderSpeed

When detects vehicle under speed, send this event.

Event params is the same as **TrafficJunction**, except for event Code is **TrafficUnderSpeed**, and add following params:

Event Code	TrafficUnderSpeed			
Event action	Start/Stop			
Event index	0			
Event Data				
Name	Type	R/O	Description	Example
+SpeedLimit	int[2]	O	Speed limit, 2 integer, min speed and max speed.	[60,120]
+UnderSpeeding Percentage	int	O	Percentage of under speed.	20
+...			The other event parameters are the same as TrafficJunction	

Event Response Example (multipart , key=value format in body , binary data in body) (response to 4.4.3 snap subscription)

```
4.4.3 snap subscription )
```

```
--<boundary>
```

```
Content-Type: text/plain
```

```
Content-Length: <length>
```

```
Events[0].EventBaseInfo.Code=TrafficUnderSpeed
```

```
Events[0].EventBaseInfo.Action=Pulse
```

```
Events[0].EventBaseInfo.Index=0
```

```
Events[0].GroupID=123
```

```
Events[0].CountInGroup=3
```

```
Events[0].IndexInGroup=1
```

```
Events[0].Lane=0
```

```
Events[0].Vehicle.BoundingBox[0]=1341
```

```
Events[0].Vehicle.BoundingBox[1]=2451
```

```
Events[0].Vehicle.BoundingBox[2]=4513
```

```
Events[0].Vehicle.BoundingBox[3]=4135
```

```
Events[0].Vehicle.Text=Audi
```

```
Events[0].Vehicle.SubText=A6L
```

```
Events[0].Vehicle.SubBrand=5
```

```
Events[0].Vehicle.BrandYear=2
```

```
Events[0].TrafficCar.RecNo=123
```

```
Events[0].SpeedLimit[0]=60
```

```
Events[0].SpeedLimit[1]=120
```

```
Events[0].UnderSpeedingPercentage=20
```

```
...
```

```
--<boundary>
```

```
Content-Type: image/jpeg
```

```
Content-Length: <image size>
```

```
<Jpeg image data>
```

```
--<boundary>
```

10.1.5 [Event] TrafficOverSpeed

When detects vehicle over speed, send this event.

Event params is the same as **TrafficJunction**, except for event Code is **TrafficOverSpeed**, and add following params:

Event Code	TrafficOverSpeed			
Event action	Start/Stop			
Event index	0			
Event Data				
Name	Type	R/O	Description	Example
+SpeedLimit	int[2]	O	Speed limit, 2 integer, min speed and max speed.	[60,120]
+UnderSpeeding Percentage	int	O	Percentage of over speed.	20

+...			The other event parameters are the same as TrafficJunction	
------	--	--	--	--

Event Response Example (multipart , key=value format in body , binary data in body) (response to 4.4.3 snap subscription)

--<boundary>

Content-Type: text/plain

Content-Length: <length>

Events[0].EventBaseInfo.Code=TrafficOverSpeed

Events[0].EventBaseInfo.Action=Pulse

Events[0].EventBaseInfo.Index=0

Events[0].GroupID=123

Events[0].CountInGroup=3

Events[0].IndexInGroup=1

Events[0].Lane=0

Events[0].Vehicle.BoundingBox[0]=1341

Events[0].Vehicle.BoundingBox[1]=2451

Events[0].Vehicle.BoundingBox[2]=4513

Events[0].Vehicle.BoundingBox[3]=4135

Events[0].Vehicle.Text=Audi

Events[0].Vehicle.SubText=A6L

Events[0].Vehicle.SubBrand=5

Events[0].Vehicle.BrandYear=2

Events[0].TrafficCar.RecNo=123

Events[0].SpeedLimit[0]=60

Events[0].SpeedLimit[1]=120

Events[0].UnderSpeedingPercentage=20

...

--<boundary>

Content-Type: image/jpeg

Content-Length: <image size>

<Jpeg image data>

--<boundary>

10.1.6 [Event] TrafficPedestrain

Event Code	TrafficPedestrain			
Event action	Pulse			
Event index	0			
Event Data				
Name	Type	R/O	Description	Example
+GroupId	int	O	The id of event group.	123
+CountInGroup	int	O	Event count in the event group.	3
+IndexInGroup	int	O	The index of this event in the event	1

			group, start from 1.	
+Lane	int	O	Lane number, start from 0.	0
+Vehicle	object	O	Actually this is the human that detected, not vehicle.	
++Category	char[]	R	Must be " Passerby ", means this actually is a human.	Passerby
++BoundingBox	int[4]	R	The detected car bounding box, 4 interge, refer to x's value of left-top point, y's value of left-top point, x's value of right-bottom point, y's value of right-bottom point. Coordinate remap to 0 — 8192.	[]

Event Response Example (multipart , key=value format in body , binary data in body) (response to 4.4.3 snap subscription)

--<boundary>

Content-Type: text/plain

Content-Length: <length>

Events[0].EventBaseInfo.Code=TrafficPedestrain

Events[0].EventBaseInfo.Action=Pulse

Events[0].EventBaseInfo.Index=0

Events[0].GroupID=123

Events[0].CountInGroup=3

Events[0].IndexInGroup=1

Events[0].Lane=0

Events[0].Vehicle.Category=Passerby

Events[0].Vehicle.BoundingBox[0]=1341

Events[0].Vehicle.BoundingBox[1]=2451

Events[0].Vehicle.BoundingBox[2]=4513

Events[0].Vehicle.BoundingBox[3]=4135

...

--<boundary>

Content-Type: image/jpeg

Content-Length: <image size>

<Jpeg image data>

--<boundary>

10.1.7 [Event] TrafficParking

When detects vehicle illegal parking, send this event.

Event params is the same as **TrafficJunction**, except for event Code is **TrafficParking**, and remove param "**speed**", add following params:

Event Code	TrafficParking
Event action	Start/Stop

Event index	0			
Event Data				
Name	Type	R/O	Description	Example
+StartParking	int	O	The start time of parking, UTC seconds.	6538920
+AlarmInterval	int	O	The alarm interval.	180
+ParkingAllowedTime	int	O	The allowed time of parking.	1600
+...			The other event parameters are the same as TrafficJunction	

Event Response Example (multipart , key=value format in body , binary data in body) (response to 4.4.3 snap subscription)

--<boundary>

Content-Type: text/plain

Content-Length: <length>

Events[0].EventBaseInfo.Code=TrafficParking

Events[0].EventBaseInfo.Action=Pulse

Events[0].EventBaseInfo.Index=0

Events[0].GroupID=123

Events[0].CountInGroup=3

Events[0].IndexInGroup=1

Events[0].Lane=0

Events[0].Vehicle.BoundingBox[0]=1341

Events[0].Vehicle.BoundingBox[1]=2451

Events[0].Vehicle.BoundingBox[2]=4513

Events[0].Vehicle.BoundingBox[3]=4135

Events[0].Vehicle.Text=Audi

Events[0].Vehicle.SubText=A6L

Events[0].Vehicle.SubBrand=5

Events[0].Vehicle.BrandYear=2

Events[0].TrafficCar.RecNo=123

Events[0].StartParking=6538920

Events[0].AlarmInterval=180

Events[0].ParkingAllowedTime=1600

...

--<boundary>

Content-Type: image/jpeg

Content-Length: <image size>

<Jpeg image data>

--<boundary>

10.2 Traffic Flow

10.2.1 [Event] TrafficFlowStat

When traffic flow trigger the rule, send this event

Event Code	TrafficFlowStat		
Event action	Start/Stop		
Event index	0		
Event Data			
Name	Type	R/O	Description
+FlowStates	object[]	R	Traffic flow info, each object in list is traffic flow info about one lane.
++Lane	int	R	Lane number, start from 0.
++Flow	int	R	Traffic flow number.
++Period	int	R	Traffic stat time, unit is minute.
++PeriodByMili	int	O	Traffic stat time, unit is millisecond, value should between 0 and 59999.
++DrivingDirection	char[]	O	Driving direction, should be an array of three strings: <ul style="list-style-type: none">• 1st string: direction, can be: "Approach", "Leave"• 2nd string: Approach position name• 3rd string: Leave position name

Event Response Example (multipart , JSON format in body) (response to 4.9.17 event subscription)

--<boundary>
Content-Type: text/plain
Content-Length: xxxx

Code=TrafficFlowStat;action=Pulse;index=0;data={
 "FlowStates": [{ "Lane": 0,
 "Flow": 50,
 "Period": 5,
 "DrivingDirection": ["Approach", "XXXCity", "YYCity"]
 }, {...}, ...]
}

10.2.2 Find Traffic Flow History

Request URL	http://<server>/cgi-bin/recordFinder.cgi?action=find		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description

name	char[]	R	The name of record table. It should be "TrafficFlow".	TrafficFlow
count	int	O	Max result to return, default is 1024.	100
StartTime	char[]	O	The start of the record's CreateTime.	123456700
EndTime	char[]	O	The End of the record's CreateTime.	123456800
condition	object	O	Search condition.	
+Channel	int	O	Video channel index which starts from 0.	0
+Lane	int	O	Lane index, starts from 0.	0

Request Example

http://192.168.1.108/cgi-bin/recordFinder.cgi?action=find&name=TrafficFlow&condition.Channel=0&condition.Lane=0&StartTime=123456700&EndTime=123456800&count=100

Response Params (key=value format in body)

Name	Type	R/O	Description	Example
totalCount	int	O	Total record num that find.	1000
found	int	O	Record num that returned.	100
records	object[]	R	The records that returned.	
+RecNo	int	R	Record id.	12345
+CreateTime	int	R	The create time of record.	123456789
+StatisticsTime	int	R	Traffic flow statistics time, UTC seconds.	123456789
+Period	int	R	Statistics period, unit is second.	300
+Channel	int	R	Video channel index which starts from 0.	0
+Lane	int	R	Lane index, starts from 0.	0
+Vehicles	int	R	Total vehicle num.	220
+AverageSpeed	float	O	Average speed of the vehicle, -1 means no vehicle, 0 means congestion.	21.8

Response Example

```

totalCount=1000
found=100
records[0].RecNo=12345
records[0].CreateTime=123456789
records[0].StaticsticsTime=123456789
records[0].Period=300
records[0].Channel=0
records[0].Lane=0
records[0].Vehicles=250
records[0].AverageSpeed=25.4
...
records[1].RecNo=13579
records[1].CreateTime=123456799
records[1].StaticcticsTime=123456799
records[1].Period=300
records[1].Channel=0
records[1].Lane=0
records[1].Vehicles=220
records[1].AverageSpeed=21.8
...

```

10.2.3 Start Traffic Statistics Search

Request URL	http://<server>/cgi-bin/api/trafficFlowStat/startFind		
Method	POST		
Request Params (JSON format in body)			
Name	Type	R/O	Description
condition	object	R	Traffic flow statistical conditions
+StartTime	systemtime	R	Start time, accurate to hour
+EndTime	systemtime	R	End time, accurate to hour
+Channel	int	O	Channel number
+Lane	int	O	Lane number
+Lanes	int[]	O	Lane number array, supports data search by multiple lanes
+ClassType	int	O	As video metadata and road monitoring use the same database table, they need to be distinguished according to their types (0 represents video metadata and 1 represents road monitoring).
+Granularity	enumchar[8]	O	<p>The granularity of statistical information that needs to be returned in the searching process. This needs to be supported by getCaps.</p> <pre>Enumchar[8]{ "Minute": By 5 minutes (hourly report) "Hour": By hour (daily report) "Day": By day (monthly report) "Month": By month (yearly report) }</pre> <p>The default value is Hour.</p>
+PresetID	int[]	O	Preset
+Direction	enumchar[16]	O	<p>Search by statistical direction</p> <pre>enumchar[16]{ "Both" "LeftToRight" "RightToLeft" }</pre>
+Channels	int[256]	O	<p>Channel number array. Used in structured traffic statistics. In multi-channel query, the channel number is distributed to query the total data of all channels</p> <p>There must be at least one channel and channels field. Generally, they</p>

		should not both exist. If both exist, channels shall prevail.	
--	--	---	--

Request Example

```
{
  "condition": {
    "StartTime": "2010-05-25 00:00:00",
    "EndTime": "2010-05-25 23:59:59",
    "Channel": 0,
    "Lane": 0,
    "Lanes": [0],
    "ClassType": 1,
    "Granularity": "Hour",
    "PresetID": [1],
    "Direction": "Both",
    "Channels": [0,1,2,5]
  }
}
```

Response Params (JSON format in body)

Name	Type	R/O	Description	Example
token	uint	O	token	2342343
totalCount	uint	O	Total number of results that meet the query criteria	333

Response Example

```
{
  "token": 2342343,
  "totalCount": 333
}
```

10.2.4 Get Traffic Statistics

Request URL	http://<server>/cgi-bin/api/trafficFlowStat/doFind		
Method	POST		

Request Params (JSON format in body)

Name	Type	R/O	Description	Example
token	uint	R	Search token.	46878
beginNumber	uint	R	Search begin number. That is to start from the beginNumber records, fetch the count records, and return. $0 \leq \text{beginNumber} \leq \text{totalCount} - 1$	0
count	uint	R	Number of traffic statistic entries per search	24

Request Example

```
{
  "token": 46878,
```

```
{
  "beginNumber": 0,
  "count": 24
}
```

Response Params (JSON format in body)

Name	Type	R/O	Description	Example
found	int	R	Number of entries that are searched	12
info	Info[]	R	Traffic statistics (For details, see traffic-flow statistics status information structure in standard.)	[, ...]

Response Example

```
{
  "found": 12,
  "info": [
    {
      "PresetID" : 1
      "DrivingDirection" : ["Approach", "XXXCity", "YYCity"],
      "Lane" : 1,
      "MachineAddress" : "XXX District, YYY Road",
      "MachineName" : "Device001",
      "UTC" : 1465389203,
      "UTCMS" : 123,
      ....
    },...{}]
}
```

10.2.5 End Traffic Statistics Search

Request URL http://<server>/cgi-bin/api/trafficFlowStat/stopFind

Method POST

Request Params (JSON format in body)

Name	Type	R/O	Description	Example
token	uint	R	Search token.	46878

Request Example

```
{
  "token": 46878
}
```

Response Params (JSON format in body)

Name	Type	R/O	Description	Example

Response Example

```
{}
```

10.3 Traffic Record

10.3.1 Insert Traffic BlockList/AllowList Record

Request URL	http://<server>/cgi-bin/recordUpdater.cgi?action=insert		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
name	char[]	R	The record table name, "TrafficBlackList" for BlockList and "TrafficRedList" for AllowList.
PlateNumber	char[]	R	The number of car plate, max string length is 31. It must be unique.
MasterOfCar	char[]	O	The car owner, max string length is 15
PlateColor	char[]	O	Plate color, max string length is 31, ex: "Yellow", "Blue", ... etc.
PlateType	char[]	O	Plate type, max string length is 31.
VehicleType	char[]	O	Vehicle type, max string length is 31.
VehicleColor	char[]	O	Vehicle color, max string length is 31. ex: "Yellow", "Blue", ... etc.
BeginTime	char[]	O	Begin time, ex: "2010-05-25 00:00:00".
CancelTime	char[]	O	Cancel time, ex: "2010-06-25 00:00:00".
AuthorityList	object	O	Authority list, only valid for "TrafficRedList" table.
+OpenGate	bool	O	Authority to open the gate.

Request Example

http://192.168.1.108/cgi-bin/recordUpdater.cgi?action=insert&name=TrafficBlackList&PlateNumber=AC00001&MasterOfCar=ZhangSan&PlateColor=Yellow&VehicleColor=Blue&BeginTime=2011-01-01%2012:00:00&CancelTime=2011-01-10%2012:00:00

Response Params (key=value format in body)				
Name	Type	R/O	Description	Example
RecNo	int	R	The new record's id, return -1 if the device handles asynchronously.	12345
Response Example				

RecNo=12345

10.3.2 Update Traffic BlockList/AllowList Record

Request URL	http://<server>/cgi-bin/recordUpdater.cgi?action=update		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
name	char[]	R	The record table name, TrafficBlackList

			"TrafficBlackList" for BlockList and "TrafficRedList" for AllowList.	
recno	int	R	record id	12345
PlateNumber	char[]	R	The number of car plate, max string length is 31. It must be unique.	AC00001
MasterOfCar	char[]	O	The car owner, max string length is 15	
PlateColor	char[]	O	Plate color, max string length is 31, ex: "Yellow", "Blue", ... etc.	Yellow
PlateType	char[]	O	Plate type, max string length is 31.	
VehicleType	char[]	O	Vehicle type, max string length is 31.	
VehicleColor	char[]	O	Vehicle color, max string length is 31. ex: "Yellow", "Blue", ... etc.	Yellow
BeginTime	char[]	O	Begin time, ex: "2010-05-25 00:00:00".	"2010-05-25 00:00:00"
CancelTime	char[]	O	Cancel time, ex: "2010-06-25 00:00:00".	"2010-06-25 00:00:00"
AuthorityList	object	O	Authority list, only valid for "TrafficRedList" table.	
+OpenGate	bool	O	Authority to open the gate.	true

Request Example

http://192.168.1.108/cgi-bin/recordUpdater.cgi?action=update&name=TrafficBlackList&recno=12345&PlateNumber=AC00001&MasterOfCar=ZhangSan&PlateColor=Yellow&VehicleColor=Blue&BeginTime=2011-01-01%2012:00:00&CancelTime=2011-01-10%2012:00:00

Response Params (OK in body)

Response Example

OK

10.3.3 Remove Traffic BlockList/AllowList Record

Request URL	http://<server>/cgi-bin/recordUpdater.cgi?action=remove			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	
name	char[]	R	The record table name, "TrafficBlackList" for BlockList and "TrafficRedList" for AllowList.	
recno	int	R	The record id.	12345

Request Example

http://192.168.1.108/cgi-bin/recordUpdater.cgi?action=remove&name=TrafficBlackList&recno=12345

Response Params (OK in body)

Response Example

OK

10.3.4 Find Traffic BlockList/AllowList Record

Request URL	http://<server>/cgi-bin/recordFinder.cgi?action=find			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
name	char[]	R	The record table name, "TrafficBlackList"for BlockList and "TrafficRedList"for AllowList.	TrafficBlackList
count	int	O	Max result to return, default is 1024.	100
StartTime	string	O	The start of the record's CreateTime.	123456700
EndTime	string	O	The end of the record's CreateTime.	123456800
condition	object	O	Search condition.	
+PlateNumber	char[]	O	Car plate number, max string length is 47.	AC00001
+PlateNumber Vague	char[]	O	Car plate number substring, match any car plate number that contain this substring, max string length is 47.	
+PlateNumber VagueGroup	char[][]	O	Car plate number substring array, max string length is 47.	
+QueryCount	int	O	Query count, default is 1000.	500
+QueryResultB egin	int	O	Begin number in the result set, default is 0.	0
Request Example				http://192.168.1.108/cgi-bin/recordFinder.cgi?action=find&name=TrafficBlackList&condition.PlateNumber=AC00001&StartTime=123456700&EndTime=123456800&count=100

Response Params (key=value format in body)				
Name	Type	R/O	Description	Example
totalCount	int	O	Total record num that find.	1000
found	int	O	Record num that returned.	100
records	object[]	R	The records that returned.	
+RecNo	int	R	Record id.	12345
+CreateTime	int	R	The create time of record.	123456789
+PlateNumber	char[]	R	Car plate number.	AC00001
+MasterOfCar	char[]	O	Car owner.	ZhangSan
+...	-	-	...<see above insert command for other params of the record>	
Response Example				
totalCount=1000 found=100 records[0].RecNo=12345 records[0].CreateTime=123456789 records[0].PlateNumber=AC00001 records[0].MasterOfCar=ZhangSan ... records[1].RecNo=13579				

```

records[1].CreateTime=123456799
records[1].PlateNumber=AC00001
records[1].MasterOfCar=LiSi
...

```

10.3.5 RemoveEx Traffic BlockList/AllowList Record

Request URL	http://<server>/cgi-bin/recordUpdater.cgi?action=removeEx		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
name	char[]	R	The record table name, "TrafficBlackList" for BlockList and "TrafficRedList" for AllowList.
recno	int	O	The record id.
PlateNumber	string	O	The number of car plate, max string length is 31. It must be unique.
Request Example			
http://192.168.1.108/cgi-bin/recordUpdater.cgi?action=removeEx&name=TrafficBlackList&PlateNumber=AC00001			

Response Params (OK in body)
Response Example
OK

10.3.6 Import Traffic BlockList/AllowList

- Import the traffic blocklist and allowlist

Upload a blocklist or allowlist file to the device, and then import the data into the device. This operation might take a long time.

Request URL	http://<server>/cgi-bin/trafficRecord.cgi?action=uploadFile&Type=<Type>		
Method	POST		
Request Params (key=value format in URL ; binary data in body)			
Name	Type	R/O	Description
Type	char[16]	R	File type, value: "TrafficBlackList" Traffic blocklist "TrafficRedList" Traffic allowlist
format	char[16]	R	File format, fixed to "CSV"
code	char[16]	R	Encoding format, the value is "utf-8" or "GB2312"
Request Example			
POST http://192.168.1.108/cgi-bin/trafficRecord.cgi?action=uploadFile&Type=TrafficBlackList&format=CSV&code=utf-8 HTTP/1.1 User-Agent: Client/1.0 Content-Type: multipart/form-data; boundary=<boundary> Content-Length:XXXX			

```
--<boundary>
Content-Disposition: form-data; name="blackfile"; filename="TrafficBlackList.CSV"
Content-Type: application/vnd.ms-excel

<File data....>
--<boundary>--
```

Response Params (OK in body)

Response Example

OK

10.3.7 Export Traffic BlockList/AllowList

- Asynchronously export traffic blocklist and allowlist

Notify the device to export the traffic blocklist and allowlist to the file. This operation is non-blocking.
To obtain the export result, please run the "getFileExportState" command.

Request URL	http://<server>/cgi-bin/recordUpdater.cgi?action=exportAsyncFile&name=<Name>		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
name	char[16]	R	File type, value: "TrafficBlackList" Traffic blocklist "TrafficRedList" Traffic allowlist
filename	char[256]	O	Export Filename
format	char[16]	R	File format, fixed to "CSV"
code	char[16]	R	Encoding format, the value is "utf-8" or "GB2312"
Request Example			
http://<server>/cgi-bin/recordUpdater.cgi?action=exportAsyncFile&name=TrafficBlackList&filename=RecordFile01&format=CSV&code=utf-8			

Response Params (OK in body)

Response Example

OK

- Get the status of the exported traffic blocklist and allowlist

Get the result of the exported traffic blocklist and allowlist

Request URL	http://<server>/cgi-bin/recordUpdater.cgi?action=getFileExportState&name=<Name>		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
name	char[16]	R	File type, value: "TrafficBlackList" Traffic blocklist "TrafficRedList" Traffic allowlist
Request Example			

`http://<server>/cgi-bin/recordUpdater.cgi?action=getFileExportState&name=TrafficBlackList`

Response Params (key=value format in body)

Name	Type	R/O	Description	Example
state	int	R	Exporting status: 0: Success 1: Failure 2: running 3: the file is invalid 4: the file is too large 5: the file has duplicate data	1

Response Example

state=0

- Download the exported traffic blocklist and allowlist file

Download the exported traffic blocklist and allowlist file

Request URL	<code>http://<server>/cgi-bin/recordUpdater.cgi?action=downloadFile&Type=<Type></code>		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
Type	char[16]	R	File type, value: "TrafficBlackList": Traffic blocklist "TrafficRedList": Traffic allowlist
filename	char[256]	O	The exported file name must be the same as the file name used in the export command.

Request Example

`http://<server>/cgi-bin/recordUpdater.cgi?action=downloadFile&Type=TrafficBlackList&filename=RecordFile01`

Response Params (binary in body)

Response Example

HTTP/1.1 200 OK

Server: Device/1.0

Content-Type: application/octet-stream

Content-Length: xxx

<File Data...>

10.3.8 Export Traffic Flow

- Export traffic flow records

Notify the device to export the traffic flow records to the file. This operation is non-blocking. To obtain the export result, please run the "getFileExportState" command.

Request URL	<code>http://<server>/cgi-bin/recordUpdater.cgi?action=exportAsyncFile&name=TrafficFlow</code>
Method	GET

Request Params (key=value format in URL)

Name	Type	R/O	Description	Example
name	char[16]	R	File type, fixed value: "TrafficFlow"	"TrafficFlow"
filename	char[256]	O	Name of the exported file	"RecordFile01"
format	char[16]	R	Format of the exported file, fixed to "CSV"	CSV
code	char[16]	R	Encoding format of the exported file, the value is "utf-8" or "GB2312"	"utf-8"

Request Example

http://<server>/cgi-bin/recordUpdater.cgi?action=exportAsyncFile&name=TrafficFlow&filename=RecordFile01&format=CSV&code=utf-8

Response Params (OK in body)

Response Example

OK

- Get the exported traffic flow record results

Get the exported traffic flow record results

Request URL	http://<server>/cgi-bin/recordUpdater.cgi?action=getFileExportState&name=TrafficFlow		
Method	GET		

Request Params (key=value format in URL)

Name	Type	R/O	Description	Example
name	char[16]	R	File type, fixed value: "TrafficFlow"	"TrafficFlow"

Request Example

http://<server>/cgi-bin/recordUpdater.cgi?action=getFileExportState&name=TrafficFlow

Response Params (key=value format in body)

Name	Type	R/O	Description	Example
state	int	R	Export status, the value is 0: success, 1: failure, 2: running, 3: the file is invalid, 4: the file is too large, 5: the file has duplicate data	1

Response Example

state=0

- Download Export 0 traffic flow records File

Download the exported traffic flow log file

Request URL	http://<server>/cgi-bin/recordUpdater.cgi?action=downloadFile&Type=TrafficFlow		
Method	GET		

Request Params (key=value format in URL)

Name	Type	R/O	Description	Example
Type	char[16]	R	File type, fixed value: "TrafficFlow"	"TrafficFlow"
filename	char[256]	O	The exported file name must be the same as the file name used in the export command.	"RecordFile01"

Request Example

`http://<server>/cgi-bin/recordUpdater.cgi?action=downloadFile&Type=TrafficFlow&filename=RecordFile01`

Response Params (binary in body)

Response Example

HTTP/1.1 200 OK

Server: Device/1.0

Content-Type: application/octet-stream

Content-Length: xxx

<File Data...>

10.3.9 Export Traffic Snap Event Info

- Export traffic snapshot event records

Notify the device to export the traffic snapshot event records to the file. This operation is non-blocking. To obtain the export result, please run the “getFileExportState” command.

Request URL	<code>http://<server>/cgi-bin/recordUpdater.cgi?action=exportAsyncFileByConditon&name=TrafficSnapEventInfo</code>			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
name	char[16]	R	File type, fixed value: "TrafficSnapEventInfo"	"TrafficSnapEventInfo"
filename	char[256]	O	Export Filename	"RecordFile01"
format	char[16]	R	File format, fixed to "CSV"	CSV
code	char[16]	R	Encoding format, the value is "utf-8" or "GB2312"	"utf-8"
condition	object	R	Data Conditions	"utf-8"
+startTime	char[16]	R	Export the Start time, format: yyyy-MM-dd HH:mm:ss	"2010-04-08 16:12:46"
+endTime	char[16]	R	Export the End time, format: yyyy-MM-dd HH:mm:ss	"2010-04-08 18:12:46"
Request Example				
<code>http://<server>/cgi-bin/recordUpdater.cgi?action=exportAsyncFileByConditon&name=TrafficSnapEventInfo&filename=RecordFile01&format=CSV&code=utf-8&condition.startTime=2010-04-08%2016:12:46&condition.endTime=2010-04-08%2018:12:46</code>				

Response Params (OK in body)

Response Example

OK

- Obtain and export traffic snapshot event record results

Obtain and export traffic snapshot event record results

Request URL	<code>http://<server>/cgi-bin/recordUpdater.cgi?action=getFileExportState&name=TrafficSnapEventInfo</code>
--------------------	--

Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description

Example

name	char[16]	R	File type, fixed value: "TrafficSnapEventInfo"
------	----------	---	--

Request Example

http://<server>/cgi-bin/recordUpdater.cgi?action=getFileExportState&name=TrafficSnapEventInfo

Response Params (key=value format in body)			
Name	Type	R/O	Description

state	int	R	Export status, the value is 0: success, 1: failure, 2: running, 3: the file is invalid, 4: the file is too large, 5: the file has duplicate data
-------	-----	---	--

Response Example

state=0

- Download the exported traffic snapshot event record file

Download the exported traffic snapshot event record file

Request URL	http://<server>/cgi-bin/recordUpdater.cgi?action=downloadFile&Type=TrafficSnapEventInfo		
--------------------	---	--	--

Method	GET		
---------------	-----	--	--

Request Params (key=value format in URL)

Name	Type	R/O	Description
Type	char[16]	R	File type, fixed value: "TrafficSnapEventInfo"
filename	char[256]	O	The exported file name must be the same as the file name used in the export command.

Request Example

http://<server>/cgi-bin/recordUpdater.cgi?action=downloadFile&Type=TrafficSnapEventInfo&filename=RecordFile01

Response Params (binary in body)

Response Example

HTTP/1.1 200 OK

Server: Device/1.0

Content-Type: application/octet-stream

Content-Length: xxx

<File Data...>

10.4 Traffic Snap Operation

10.4.1 Open Strobe

Request URL	http://<server>/cgi-bin/trafficSnap.cgi?action=openStrobe
--------------------	---

Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
channel	int	R	video channel index which starts from 1.	1
info	object	R	info	
openType	char[]	R	For now, the value is fixed to "Normal"	"Normal"
plateNumber	char[]	R	the plateNumber of the TrafficCar.	046XRW
Request Example				
http://192.168.1.108/cgi-bin/trafficSnap.cgi?action=openStrobe&channel=1&info.openType=Normal&info.plateNumber=046XRW				

Response Params (OK in body)
Response Example
OK

10.4.2 Close Strobe

Request URL	http://<server>/cgi-bin/api/trafficSnap/closeStrobe			
Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
info	object	O	close strobe params	null
+location	int32	O	lane number, default 0	0
Request Example				
{ "info": { "location": 0 } }				

Response Params (JSON format in body)
Name
Type
R/O
Description
Example
Response Example
{}

10.4.3 Open/Close Unlicensed Vehicle Detection

Request URL	http://<server>/cgi-bin/trafficSnap.cgi?			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
action	char[]	R	this value can be open and close	open
channel	int	R	video channel index which starts from 1.	1
name	char[]	R	name	UnlicensedVehicle

Request Example

http://192.168.1.108/cgi-bin/trafficSnap.cgi?action=open&channel=1&name=UnlicensedVehicle

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

OK

10.4.4 Manual Snap

Take a snapshot manually. For intelligent traffic device, it should use this method to take a snapshot. But, the response is not image data. If you want to get the image data, please follow these steps:

1. Use the method mentioned chapter (4.4.3 Subscribe to snapshot) to subscribe the image data, and the eventcode is "TrafficManualSnap".
2. Use the "manSnap" to take a snapshot manually.
3. In the connection which built in the Step 1, the device will send the image data.

Request URL	http://<server>/cgi-bin/trafficSnap.cgi?action=manSnap
--------------------	--

Method	GET
---------------	-----

Request Params (key=value format in URL)

Name	Type	R/O	Description	Example
channel	int	R	video channel index which starts from 1.	1

Request Example

http://192.168.1.108/cgi-bin/trafficSnap.cgi?action=manSnap&channel=1

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

OK

10.5 Traffic Parking

10.5.1 Get the Specific Parking Space Status

Request URL	http://<server>/cgi-bin/trafficSnap.cgi?action=getParkingSpaceStatus
--------------------	--

Method	GET
---------------	-----

Request Params (key=value format in URL)

Name	Type	R/O	Description	Example
channel	int32	O	video channel index which starts from 1.	1
condition	object	O		
+Lane	int[]	O	The Lane value which starts from 0.	[1,2]
+ResponseLevel	int	O	The Level value , refer to condition	0

Request Example

[http://192.168.1.108/cgi-bin/trafficSnap.cgi?action=getParkingSpaceStatus&condition.Lane\[0\]=0&condition.Lane\[1\]=255](http://192.168.1.108/cgi-bin/trafficSnap.cgi?action=getParkingSpaceStatus&condition.Lane[0]=0&condition.Lane[1]=255)

Response Params (key=value format in body)

Name	Type	R/O	Description	Example
status	object[]	O		
+Lane	int	O	lane number	1
+PictureId	uint	O	picture id	123456
+TrafficCar	TrafficCar	O	car info	
++...			the members refer to TrafficCar	

Response Example

```
status[0].Lane=0
status[0].PictureId=5
status[0].TrafficCar.CountInGroup=1
...
status[1].Lane=1
status[1].PictureId=4
status[1].TrafficCar.CountInGroup=1
...
```

10.5.2 Get All Status of Parking Spaces

Request URL	http://<server>/cgi-bin/trafficParking.cgi?action=getAllParkingSpaceStatus			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
Request Example				
http://192.168.1.108/cgi-bin/trafficParking.cgi?action=getAllParkingSpaceStatus				

Response Params (key=value format in body)

Name	Type	R/O	Description	Example
status	object[]	O		
+Lane	int32	O		0
+CustomParkNo	char[32]	O		"A2701"
+Status	enumchar[8]	O	status Enumchar[8]{ Park NoPark }	"Park"
+SpaceType	enumint	O	0: trolley parking space, 1: Crane parking space	0
+Plate	char[64]	O	car plate number	"ZA888"
+Time	uint64	O	enter time	1678778135
+Enable	bool	O	enable/disable	true
sceneType	enumchar[32]	O	scene type; "parkingspace": parking space detection	"ParkingStatistics"

			type; and "parkingstatistics": parking space statistics type	
statisticsMode	enumchar[32]	O	s Statistical mode, valid when the sceneType is "parkingstatistics". enumchar[32]{ "AreaMode" "SpaceMode" }	"AreaMode"
areaStatus	object[10]	O	area status. It is an array. It is valid when scenetype is "parkingstatistics" and statisticsmode is "areamode"	
+Name	char[32]	O	name	"A01"
+Count	uint32	O	total count	10
+SpaceType	enumint	O	parking space type: 0: trolley parking space, 1: Crane parking space	0
+RemainCnt	int32	O	remain count	5

Response Example

```

status[0].Lane=0
status[0].CustomParkNo = A2701
status[0].Status = Park
status[0].SpaceType = 0
...
status[1].Lane=1
status[1].Status = NoPark
...
sceneType = ParkingStatistics
statisticsMode = AreaMode
areaStatus[0].Name = A01
areaStatus[0].Count = 10
areaStatus[0].SpaceType = 0
areaStatus[0].RemainCnt = 5
...
areaStatus[1].Name = A02
areaStatus[1].Count = 20
...

```

10.5.3 [Config] Parking Space Light State

Use this method. It can get the light state config. For example, it can know that when the space is free, then the light should be green, and the space is full, the light should be red.

Config Data Params				
Name	Type	R/O	Description	Example
ParkingSpaceLightState	object	O	Parking space indicator light	
+SpaceFree	object	O	light color when the parking space was in idle status.	
++Red	enumint	O	red light	0

			enumint{ 0: off 1: on 2: flshing }	
++Yellow	enumint	O	enumint{ 0: off 1: on 2: flshing }	0
++Blue	enumint	O	enumint{ 0: off 1: on 2: flshing }	0
++Green	enumint	O	enumint{ 0: off 1: on 2: flshing }	1
++Purple	enumint	O	enumint{ 0: off 1: on 2: flshing }	0
++White	enumint	O	enumint{ 0: off 1: on 2: flshing }	0
++Pink	enumint	O	enumint{ 0: off 1: on 2: flshing }	0
++Cyan	enumint	O	enumint{ 0: off 1: on 2: flshing }	0
++Color	char[8][32]	O	Color range: "Off" "Red" "Yellow" "Blue" "Green" "Purple"	["Green","Off"]

			"White" "Pink" "Cyan" "Off"	
+SpaceFull	object	O	light color when the parking space was occupied refer to SpaceFree	
+SpaceOverLine	object	O	light color when the parking space was pressing line refer to SpaceFree	
+SpaceOrder	object	O	light color when the parking space was Scheduled refer to SpaceFree	
+NetworkException	object	O	light color when network was aborted	
++NetPortAbort	object[]	O	refer to SpaceFree	
+SpaceSpecial	object	O	light color of special space refer to SpaceFree	
+SpaceCharging	object	O	light color of charging parking space refer to SpaceFree	
+SpaceAbnormity	object	O	light color when the parking space was abnormal. refer to SpaceFree	
+AbnormalAlarm	object	O	Abnormal	
++VideoBlind	object	O	light color when video blind refer to SpaceFree	
+SpaceFreeMode	enumint	O	0: Long on, 1: Flashing, 2: Off Default 0	1
+SpaceFullMode	enumint	O	When the parking space is full, the indicator light mode is: 0: permanently on, 1: flashing, and 2: off Default 0	2

Please refer to "4.2.1 Obtaining and Setting Configuration" to obtain and set the configuration. The specific examples are as follows:

Get Config Request Example

```
http://10.0.0.8/cgi-bin/configManager.cgi?action=getConfig&name=ParkingSpaceLightState
```

Get Config Response Example

```
table.ParkingSpaceLightState.SpaceFree.Blue=0
table.ParkingSpaceLightState.SpaceFree.Green=1
table.ParkingSpaceLightState.SpaceFree.Pink=0
table.ParkingSpaceLightState.SpaceFree.Purple=0
table.ParkingSpaceLightState.SpaceFree.Red=0
table.ParkingSpaceLightState.SpaceFree.White=0
table.ParkingSpaceLightState.SpaceFree.Yellow=0
table.ParkingSpaceLightState.SpaceFull.Blue=0
```

```

table.ParkingSpaceLightState.SpaceFull.Green=0
table.ParkingSpaceLightState.SpaceFull.Pink=0
table.ParkingSpaceLightState.SpaceFull.Purple=0
table.ParkingSpaceLightState.SpaceFull.Red=1
table.ParkingSpaceLightState.SpaceFull.White=0
table.ParkingSpaceLightState.SpaceFull.Yellow=0
table.ParkingSpaceLightState.SpaceOrder.Blue=0
table.ParkingSpaceLightState.SpaceOrder.Green=0
table.ParkingSpaceLightState.SpaceOrder.Pink=0
table.ParkingSpaceLightState.SpaceOrder.Purple=0
table.ParkingSpaceLightState.SpaceOrder.Red=0
table.ParkingSpaceLightState.SpaceOrder.White=0
table.ParkingSpaceLightState.SpaceOrder.Yellow=1
table.ParkingSpaceLightState.SpaceOverLine.Blue=0
table.ParkingSpaceLightState.SpaceOverLine.Green=0
table.ParkingSpaceLightState.SpaceOverLine.Pink=0
table.ParkingSpaceLightState.SpaceOverLine.Purple=0
table.ParkingSpaceLightState.SpaceOverLine.Red=0
table.ParkingSpaceLightState.SpaceOverLine.White=0
table.ParkingSpaceLightState.SpaceOverLine.Yellow=1
table.ParkingSpaceLightState.SpaceSpecial.Blue=0
table.ParkingSpaceLightState.SpaceSpecial.Green=0
table.ParkingSpaceLightState.SpaceSpecial.Pink=0
table.ParkingSpaceLightState.SpaceSpecial.Purple=0
table.ParkingSpaceLightState.SpaceSpecial.Red=0
table.ParkingSpaceLightState.SpaceSpecial.White=0
table.ParkingSpaceLightState.SpaceSpecial.Yellow=1

```

Set Config Request Example

`http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&ParkingSpaceLightState.SpaceFree.Blue=1&ParkingSpaceLightState.SpaceFree.Green=0&ParkingSpaceLightState.SpaceFree.Pink=0&ParkingSpaceLightState.SpaceFree.Purple=0&ParkingSpaceLightState.SpaceFree.Red=0&ParkingSpaceLightState.SpaceFree.White=0&ParkingSpaceLightState.SpaceFree.Yellow=0`

Set Config Response Example

OK

10.5.4 Set Order State

Request URL	<code>http://<server>/cgi-bin/trafficParking.cgi?action=setOrderState</code>			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
state	object[]	R	state info	
+Lane	int	R	for now, fixed to 0	0
+State	char[]	R	the value can be "Ordered"or "Free".	Ordered
Request Example				
<code>http://192.168.1.108/cgi-bin/trafficParking.cgi?action=setOrderState&state[0].Lane=0&state[0].State=Or</code>				

dered

Response Params (OK in body)

Response Example

OK

10.5.5 Set Light State

Request URL	http://<server>/cgi-bin/trafficParking.cgi?action=setLightState			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
state	object[]	R	state info	
+LightNo	int	R	the No. of the Lane.	0
+Color	char[]	R	it can be Red, Yellow, Blue, Green, Purple, White, Pink.	Ordered
+State	int	O	it can be 0, 1, 2.0 means close;1 means open;2 means twinkle;	
+Enable	bool	O	true or false, enable or not.	
Request Example				
http://192.168.1.108/cgi-bin/trafficParking.cgi?action=setLightState&state[0].LightNo=0&state[0].Color=Red&state[0].State=0&state[0].Enable=true				

Response Params (OK in body)

Response Example

OK

10.5.6 [Config] Parking Space Access Filter Setting

Config Data Params				
Name	Type	R/O	Description	Example
ParkingSpaceAccessFilter	object	O	ParkingSpaceAccessFilter config object	
+Enable	bool	O	Enable filter or not	true
+Type	enumchar[32]	O	Filter type, can be: "BannedList", "TrustList".	"BannedList"
+TrustList	char[][40]	O	IP address list that trust	["10.6.10.1", "10.6.10.2"]
+BannedList	char[][40]	O	IP address list that banned.	["10.6.10.3", "10.6.10.4"]

Please refer to "4.2.1 Obtaining and Setting Configuration" to obtain and set the configuration. The specific examples are as follows:

Get Config Request Example

http://10.0.0.8/cgi-bin/configManager.cgi?action=getConfig&name=ParkingSpaceAccessFilter

Get Config Response Example

table.ParkingSpaceAccessFilter.Enable=false
table.ParkingSpaceAccessFilter.Type=TrustList
table.ParkingSpaceAccessFilter.TrustList[0]=172.24.2.14

table.ParkingSpaceAccessFilter.BannedList[0]=172.24.2.15

Set Config Request Example

http://10.0.0.8/cgi-bin/ConfigManager.cgi?action=setConfig&ParkingSpaceAccessFilter.Enable=true&ParkingSpaceAccessFilter.Type=TrustList&ParkingSpaceAccessFilter.TrustList[0]=172.24.2.14&ParkingSpaceAccessFilter.BannedList[0]=172.24.2.15

Set Config Response Example

OK

10.5.7 Set OverLine State

Request URL	http://<server>/cgi-bin/trafficParking.cgi?action=setOverLineState		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
state	object[]	R	state
+Lane	int	R	for now, fixed to 0.
+State	char[]	R	the value can be "OverLine" or "StopOverLine".
Request Example			
http://192.168.1.108/cgi-bin/trafficParking.cgi?action=setOverLineState&state[0].Lane=0&state[0].State=OverLine			

Response Params (OK in body)
Response Example
OK

10.5.8 Set Parking Control Info

Request URL	http://<server>/cgi-bin/api/trafficParking/setParkControlInfo		
Method	POST		
Request Params (JSON format in body)			
Name	Type	R/O	Description
Channel	uint32	O	Channel number, starting from 0
ScreenShowIn	object[16]	O	Screen display control information, up to 16 arrays.
+ScreenNo	uint32	O	Screen number, starting from 0
+Text	char[256]	O	Display content: When the value of Text Type is 1, the Text is displayed in the format of time. %Y: Year %M: Month %D: Day %H: Hour (24-hour system) %h: Hour (12-hour system)

			<p>%m: Minute %S: Second %W: Week %T: Display a.m or p.m %X: Displays the general text When the value of Text Type is 2, the Text is displayed in the form of QR code URL. When the value of Text Type is 3, if a specific file is displayed or played, the text will be displayed in the form of QR code URL, such as xx.jpg and xx.mp4. For the device to automatically play the resource files on the device, configure Text as Local.</p>	
+Type	uint32	O	<p>Content format: 0: General text 1: Local time (notifies the device to automatically update the time displayed on the screen to the local time) 2: QR code 3: Resource file</p>	1
+Color	uint32	O	<p>Text color: 0: Green 1: Red 2: Yellow 3: White</p>	1
+RollMode	uint32	O	<p>Text scrolling mode: 0: NA (Do not scroll when the content does not exceed the screen) 1: Scroll left and right 2: Scroll up and down 3: Crop (Crop the part that exceeds the screen)</p>	1
+RollSpeed	uint32	O	<p>Scrolling speed, divided into 5 levels from 1 to 5 from slower to faster</p>	1
+DisplayEffect	uint32	O	<p>Display effect: When RollMode is 0/1: { 0: Move left 1: Move right 2: Move up 3: Move down } When RollMode is 2:</p>	1

			<pre>{ 0: Up 1: Down } When RollMode is 3: { 0: Left alignment 1: Center 2: Right alignment }</pre>	
BroadcastInfo	object[16]	O	Broadcasting control information, up to 16 arrays.	
+Text	char[256]	O	Audio text	"welcome"
+Type	uint32	O	Text type 0: General text 1: License plate (Letters cannot be combined into vocabularies for broadcasting) 2: Time text (Being broadcast in the format of time) 3: String of digits (Being broadcast in the form of strings, and cannot be combined into numbers)	0

Request Example

```
{
  "Channel": 0,
  "ScreenShowInfo": [
    {
      "ScreenNo": 4,
      "Text": "%H:%m:%S",
      "Type": 1,
      "Color": 1,
      "RollMode": 1,
      "RollSpeed": 1,
      "DisplayEffect": 1
    }, ... {}
  ],
  "BroadcastInfo": [
    {
      "Text": "welcome",
      "Type": 0
    }, ... {}
  ]
}
```

Response Params (JSON format in body)

Name	Type	R/O	Description	Example
Response Example				
{}				

10.5.9 Set Parking Space Lighting Plan

Request URL	http://<server>/cgi-bin/api/trafficParking/setSpaceLightPlan			
Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
Channel	uint32	O	channel number which starts from 0	0
PhysicalLane	uint8	O	Physical parking space number, prioritize using this item. If it is not available, use CustomParkNo instead	0
CustomParkNo	char[32]	O	Custom parking lot number(Parking lot)	"A2701"
LightPlan	object[]	O	Lighting Plan	
+Color	enumchar[16]	O	Parking light color enumchar[16]{ "Red" "Yellow" "Green" "Blue" "Purple" "White" "Pink" }	"Red"
+State	enumint	O	Parking light status enumint{ 0: permanently off 1: permanently on 2: blink 3: Consistent with previous state }	1
+KeepTime	int32	O	Duration, in seconds -1 for permanently on	30
Request Example				
{ "Channel": 0, "PhysicalLane": 0, "CustomParkNo": "A2701", "LightPlan": ["Color": "Red", "State": 1, "KeepTime": 30],...{} }				

Response Params (JSON format in body)

Name	Type	R/O	Description	Example
------	------	-----	-------------	---------

Response Example

```
{}
```

10.6 Vehicles Distribution

10.6.1 Subscribe Vehicles Distribution Data

Request URL	http://<server>/cgi-bin/vehiclesDistribution.cgi?action=attach		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
Channel	int	R	Video channel index which starts from 1
heartbeat	int	O	Send heartbeat interval, range is [1, 60], unit is second. If the URL contains this parameter, and the value is 5, it means every 5 seconds the device should send the heartbeat message to the client, the heartbeat message are "Heartbeat". If this parameter is not present, its default value is 60.

Request Example

```
http://192.168.1.108/cgi-bin/intelli.cgi?action=attachResource&heartbeat=5
```

Response Params (multipart; key=value format in body)			
Name	Type	R/O	Description
Channel	int	R	Video channel index which starts from 1
VehiclesData	object[]	R	Vehicles distribution data info.
+PtzPresetId	int	R	The ptz preset index which starts from 1.
+RuleId	int	R	The rule id.
+RuleType	int	R	The rule type, 197 : vehicle congestion detection, 198 : vehicle limit detection.
+VehiclesNum	int	R	The vehicles number.
+QueueLen	int	R	The vehicle queue length.
+Region	int[32][2]	R	The detection region, the first array is point list, max item is 32, the second array is point, must be two int, means x and y value, coordinate remap to 0 - 8192.

Response Example

HTTP/1.1 200 OK

Server: Device/1.0

Content-Type: multipart/x-mixed-replace; boundary=<boundary>

Connection: closed

--<boundary>

Content-Type: text/plain

```
Content-Length: <length>
```

```
Channel=1
VehiclesData[0].PtzPresetId=1
VehiclesData[0].RuleId=1
VehiclesData[0].RuleType=197
VehiclesData[0].VehiclesNum=25
VehiclesData[0].QueueLen=53
VehiclesData[0].Region[0][0]=1032
VehiclesData[0].Region[0][1]=1035
VehiclesData[0].Region[1][0]=1045
VehiclesData[0].Region[1][1]=5072
VehiclesData[0].Region[2][0]=6163
VehiclesData[0].Region[2][1]=5127
VehiclesData[0].Region[3][0]=6031
VehiclesData[0].Region[3][1]=1063
```

```
--<boundary>
```

```
Content-Type: text/plain
```

```
Content-Length: 11
```

```
Heartbeat
```

```
-<boundary>
```

```
Content-Type: text/plain
```

```
Content-Length: <length>
```

```
Channel=1
```

```
VehiclesData[0].PtzPresetId=2
```

```
...
```

HTTP API V3.35 - Intelbras

10.7 Vehicle Manager

10.7.1 Adding Vehicle Groups

Add vehicle groups, and then add vehicles to the groups.

Request URL	http://<server>/cgi-bin/api/VehicleRegisterDB/createGroup		
Method	POST		
Request Params (JSON format in body)			
Name	Type	R/O	Description
group	object	R	Group information
+GroupName	char[128]	R	Group name
+GroupDetail	char[256]	O	Remarks of the group
+GroupType	enumchar[16]	O	Group type enumchar[16]{

			"AllowListDB": Allowlist "BlockListDB": Blocklist }	
--	--	--	---	--

Request Example

```
{
  "group": {
    "GroupName": "Serious criminal database",
    "GroupDetail": "Key criminals",
    "GroupType": "AllowListDB"
  }
}
```

Response Params (JSON format in body)

Name	Type	R/O	Description	Example
groupId	char[64]	R	Group ID assigned by server	"10000"

Response Example

```
{
  "groupId": "10000"
}
```

10.7.2 Modifying Vehicle Groups

Request URL	http://<server>/cgi-bin/api/VehicleRegisterDB/modifyGroup			
Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
group	object	R	Group information	
+GroupId	char[128]	R	Group ID	"0001"
+GroupName	char[128]	O	Group name	"Serious criminal database"
+GroupDetail	char[256]	O	Remarks of the group	"Key criminals"
+GroupType	enumchar[16]	O	Group type enumchar[16]{ "AllowListDB": Allowlist "BlockListDB": Blocklist }	"AllowListDB "

Request Example

```
{
  "group": {
    "GroupId": "0001",
    "GroupName": "Serious criminal database",
    "GroupDetail": "Key criminals",
    "GroupType": "AllowListDB"
  }
}
```

Response Params (JSON format in body)

Name	Type	R/O	Description	Example
------	------	-----	-------------	---------

Response Example

{}

10.7.3 Deleting Vehicle Groups

Request URL	http://<server>/cgi-bin/api/VehicleRegisterDB/deleteGroup		
Method	POST		
Request Params (JSON format in body)			
Name	Type	R/O	Description
groupID	char[64]	R	Group ID. "" means to delete all groups.
Request Example			
{	"groupID": "000001"		
}			

Response Params (JSON format in body)

Name	Type	R/O	Description	Example
Response Example				
{				

10.7.4 Searching for Vehicle Groups

Request URL	http://<server>/cgi-bin/api/VehicleRegisterDB/findGroup		
Method	POST		
Request Params (JSON format in body)			
Name	Type	R/O	Description
groupID	char[64]	R	Group ID. "" means to search for all group information.
Request Example			
{	"groupID": "000001"		
}			

Response Params (JSON format in body)

Name	Type	R/O	Description	Example
GroupList	object[]	R	List of searched vehicle group information	
+groupID	char[64]	R	Group ID	"000001"
+groupName	char[128]	O	Group name	"Serious criminal database"
+groupDetail	char[256]	O	Remarks of the group	"Vehicles of key criminals"
+groupType	enumchar[16]	O	Group type enumchar[16]{ "AllowListDB": Allowlist "BlockListDB": Blocklist }	"AllowListDB"

+groupSize	int	O	Number of vehicles in the current group	30
+channels	int16[1024]	O	List of video channel numbers to which the current group is bound. If the current group is not bound to any video channel, enter [-1] in this parameter.	[0,]

Response Example

```
{
  "GroupList": [
    {
      "groupID": "000001",
      "groupName": "Serious criminal database",
      "groupDetail": "Vehicles of key criminals",
      "groupType": "AllowListDB",
      "groupSize": 30,
      "channels": [0]
    }, ...
  ]
}
```

10.7.5 Adding Vehicle Records

Add vehicle records to vehicle groups.

Request URL	http://<server>/cgi-bin/api/VehicleRegisterDB/multiAppend			
Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
vehicle	object[]	R	Multiple vehicle information	[,]
+UID	uint64	R	Unique identifier of the vehicle, generated by the server to represent a unique identifier (primary key) in the program. It is different from the ID field.	120837
+GroupID	char[64]	R	Group ID to which the vehicle belongs	"00001"
+GroupName	char[128]	O	Group name to which the vehicle belongs (exists only when the parameter exits)	"Group1"
+PlateNumber	char[64]	O	License plate number	"ZA12345"
+PlateCountry	char[3]	O	The country where the vehicle is located (2 bytes), conforming to ISO3166.	"CN"
+PlateType	uint16	O	License plate type	12
+Type	uint16	O	Vehicle type (such as sedans and trucks)	2
+Brand	uint16	O	Vehicle logo. You need to get the real logo through the mapping table, which is the same as CarLogoIndex of ANPR events.	10

+Serie	uint16	O	Vehicle sub-brand. You need to get the real sub-brand through the mapping table, which is the same as the SubBrand of ANPR event.	1005
+CarSeriesMode YearIndex	uint16	O	Vehicle model year. You need to get the real model year through the mapping table, which is the same as BrandYear of ANPR event.. The model year No. of vehicle head ranges from 1 to 999; the model year No. of vehicle tail ranges 1001 to 1999. 0 means unknown, and 1000 is reserved.	12
+VehicleColor	uint8[4]	O	Vehicle color. The first element represents the red component value. The second element represents the green component value. The third element represents the blue component value. The fourth element represents the transparency component value (meaningless).	[128, 128, 128, 255]
+VehicleColorSta te	uint8	O	Vehicle color status. 0: Unknown; 1: Known.	0
+PlateColor	uint8[4]	O	License plate color. It is the same as vehicle color.	[128, 128, 128, 255]
+PlateColorState	uint8	O	License plate color status. 0: Unknown; 1: Known.	0
+Name	char[64]	O	Name of the vehicle owner	"ZhangSan"
+Sex	char[16]	O	Gender "Male" "Female" "Unknown" (such as no field)	"Male"
+CertificateType	char[16]	O	ID card type IC: IC card Passport Officer: Officer card Unknown	"IC"
+ID	char[32]	O	Person ID, employee ID, and other IDs	"123456789"
+Country	char[3]	O	Nationality of vehicle owner (2 bytes), conforming to ISO3166.	"CN"
+Province	char[64]	O	Province	"XXX"
+City	char[64]	O	City	"YYY"
+HomeAddress	char[128]	O	Home address of the registered person (IVSS requirement)	"ZZZ Road"
+Email	char[32]	O	Email address of the vehicle owner	"user@example.com"

+PhoneNo	char[128]	O	Phone No. of the registered vehicle owner	"13xxxxx5678"
replace	bool	O	Overwrite the same record True: Yes; false: No.	true

Request Example

```
{
  "vehicle": {
    "UID" : 120837,
    "GroupID" : "00001",
    "GroupName" : "Group1",
    "PlateNumber" : "ZA12345",
    "PlateCountry" : "CN",
    "PlateType" : 12,
    "Type" : 2,
    "Brand" : 10,
    "Serie" : 1005,
    "CarSeriesModelYearIndex" : 12,
    "VehicleColor" : [128, 128, 128, 255],
    "VehicleColorState" : 0,
    "PlateColor" : [128, 128, 128, 255],
    "PlateColorState" : 0,
    "Name" : "ZhangSan",
    "Sex" : "Male",
    "CertificateType" : "IC",
    "ID" : "123456789",
    "Country" : "CN",
    "Province" : "XXX",
    "City" : "YYY",
    "HomeAddress" : "ZZZ Road",
    "Email" : "user@example.com",
    "PhoneNo" : "13xxxxx5678",
  },...{}
  "replace": true
}
```

Response Params (JSON format in body)				
Name	Type	R/O	Description	Example
ErrCode	uint[]	O	Error code. Defined the hexadecimal error code before you use the error code. (Available for new devices). 0x0: Operation successful. 0x11340200: Database operation failed. 0x11340202: The license plate do not exist.	[0, 0]

		0x11340205: Exceeded the maximum number of license plate database.	
Response Example			
{ "ErrCode": [0, 0] }			

10.7.6 Modifying Vehicle Information

Request URL	http://<server>/cgi-bin/api/VehicleRegisterDB/modifyVehicle		
Method	POST		
Request Params (JSON format in body)			
Name	Type	R/O	Description
vehicle	object	R	Vehicle information
+UID	uint64	R	Unique identifier of the vehicle, generated by the server to represent a unique identifier (primary key) in the program. It is different from the ID field.
+GroupID	char[64]	R	Group ID to which the vehicle belongs
+GroupName	char[128]	O	Group name to which the vehicle belongs (exists only when the parameter exists)
+PlateNumber	char[64]	O	License plate number
+PlateCountry	char[3]	O	The country where the vehicle is located (2 bytes), conforming to ISO3166.
+PlateType	uint16	O	License plate type
+Type	uint16	O	Vehicle type (such as sedans and trucks)
+Brand	uint16	O	Vehicle logo. You need to get the real logo through the mapping table, which is the same as CarLogoIndex of ANPR events.
+Serie	uint16	O	Vehicle sub-brand. You need to get the real sub-brand through the mapping table, which is the same as the SubBrand of ANPR event.
+CarSeriesModelYearIndex	uint16	O	Vehicle model year. You need to get the real model year through the mapping table, which is the same as BrandYear of ANPR event.. The model year No. of vehicle head ranges from 1 to 999; the model year No. of vehicle tail ranges 1001 to 1999. 0 means unknown, and 1000 is reserved.
+VehicleColor	uint8[4]	O	Vehicle color. The first element represents the red
			[128, 128, 128, 255]

			component value. The second element represents the green component value. The third element represents the blue component value. The fourth element represents the transparency component value (meaningless).	
+VehicleColorState	uint8	O	Vehicle color status. 0: Unknown; 1: Known.	0
+PlateColor	uint8[4]	O	License plate color. It is the same as vehicle color.	[128, 128, 128, 255]
+PlateColorStatus	uint8	O	License plate color status. 0: Unknown; 1: Known.	0
+Name	char[64]	O	Name of the vehicle owner	"ZhangSan"
+Sex	char[16]	O	Gender "Male" "Female" "Unknown" (such as no field)	"Male"
+CertificateType	char[16]	O	ID card type IC: IC card Passport Officer: Officer card Unknown	"IC"
+ID	char[32]	O	Person ID, employee ID, and other IDs.	"13xxxxx6789"
+Country	char[3]	O	Nationality of vehicle owner (2 bytes), conforming to ISO3166.	"CN"
+Province	char[64]	O	Province	"XXX"
+City	char[64]	O	City	"YYY"
+HomeAddress	char[128]	O	Home address of the registered person (IVSS requirement)	"ZZZ Road"
+Email	char[32]	O	Email address of the vehicle owner	"user@example.com"
+PhoneNo	char[128]	O	Phone No. of the registered vehicle owner.	"13xxxxx5678"

Request Example

```
{
  "vehicle": {
    "UID" : 120837,
    "GroupID" : "00001",
    "GroupName" : "Group1",
    "PlateNumber" : "ZA12345",
    "PlateCountry" : "CN",
    "PlateType" : 12,
    "Type" : 2,
    "Brand" : 10,
    "Serie" : 1005,
```

```

    "CarSeriesModelYearIndex" : 12,
    "VehicleColor" : [128, 128, 128, 255],
    "VehicleColorState" : 0,
    "PlateColor" : [128, 128, 128, 255],
    "PlateColorState" : 0,
    "Name" : "ZhangSan",
    "Sex" : "Male",
    "CertificateType" : "IC",
    "ID" : "123456789",
    "Country" : "CN",
    "Province" : "XXX",
    "City" : "YYY",
    "HomeAddress" : "ZZZ Road",
    "Email" : "user@example.com",
    "PhoneNo" : "13xxxxx5678",
  }
}

```

Response Params (JSON format in body)

Name	Type	R/O	Description	Example
Response Example				
{}				

10.7.7 Deleting Vehicle Information

Request URL	http://<server>/cgi-bin/api/VehicleRegisterDB/deleteVehicle			
Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
vehicle	object	R		
+UID	uint64	R	Unique vehicle identifier. It can also be deleted by groupID and plateNumber if UID is not provided.	124
+groupID	char[64]	R	Vehicle group ID	"3"
+plateNumber	char[64]	O	Licence plate	"ZA12345"
Request Example				
{				
"vehicle": {				
"UID": 124,				
"groupID": "3",				
"plateNumber": "ZA12345"				
}				
}				

Response Params (JSON format in body)

Name	Type	R/O	Description	Example
Response Example				

{}

10.7.8 Searching for Vehicles from Registered Database

Request URL	http://<server>/cgi-bin/api/VehicleRegisterDB/startFind			
Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
vehicle	object	R	Vehicle information. GroupID is required. Other vehicle descriptions are optional, and you can search for vehicles with these descriptions from specific databases.	
+UID	uint64	R	Unique identifier of the vehicle, generated by the server to represent a unique identifier (primary key) in the program. It is different from the ID field.	120837
+GroupID	char[64]	R	Group ID to which the vehicle belongs	"00001"
+GroupName	char[128]	O	Group name to which the vehicle belongs (exists only when the parameter exists)	"Group1"
+PlateNumber	char[64]	O	License plate number	"ZA12345"
+PlateCountry	char[3]	O	The country where the vehicle is located (2 bytes), conforming to ISO3166.	"CN"
+PlateType	uint16	O	License plate type	12
+Type	uint16	O	Vehicle type (such as sedans and trucks)	2
+Brand	uint16	O	Vehicle logo. You need to get the real logo through the mapping table, which is the same as CarLogoIndex of ANPR events.	10
+Serie	uint16	O	Vehicle sub-brand. You need to get the real sub-brand through the mapping table, which is the same as the SubBrand of ANPR event.	1005
+CarSeriesModelYearIndex	uint16	O	Vehicle model year. You need to get the real model year through the mapping table, which is the same as BrandYear of ANPR event.. The model year No. of vehicle head ranges from 1 to 999; the model year No. of vehicle tail ranges 1001 to 1999. 0 means unknown, and 1000 is reserved.	12
+VehicleColor	uint8[4]	O	Vehicle color. The first element represents the red component value.	[128, 128, 128, 255]

			The second element represents the green component value. The third element represents the blue component value. The fourth element represents the transparency component value (meaningless).	
+VehicleColorState	uint8	O	Vehicle color status. 0: Unknown; 1: Known.	0
+PlateColor	uint8[4]	O	License plate color. It is the same as vehicle color.	[128, 128, 128, 255]
+PlateColorStatus	uint8	O	License plate color status. 0: Unknown; 1: Known.	0
+Name	char[64]	O	Name of the vehicle owner	"ZhangSan"
+Sex	char[16]	O	Gender "Male" "Female" "Unknown" (such as no field)	"Male"
+CertificateType	char[16]	O	ID card type IC: IC card Passport Officer: Officer card Unknown	"IC"
+ID	char[32]	O	Person ID, employee ID, and other IDs.	"123456789"
+Country	char[3]	O	Nationality of vehicle owner (2 bytes), conforming to ISO3166.	"CN"
+Province	char[64]	O	Province	"XXX"
+City	char[64]	O	City	"YYY"
+HomeAddress	char[128]	O	Home address of the registered person (IVSS requirement)	"ZZZ Road"
+Email	char[32]	O	Email address of the vehicle owner	"user@example.com"
+PhoneNo	char[128]	O	Phone No. of the registered vehicle owner.	"13xxxxx5678"

Request Example

```
{
  "vehicle": {
    "UID" : 120837,
    "GroupID" : "00001",
    "GroupName" : "Group1",
    "PlateNumber" : "ZA12345",
    "PlateCountry" : "CN",
    "PlateType" : 12,
    "Type" : 2,
    "Brand" : 10,
    "Serie" : 1005,
    "CarSeriesModelYearIndex" : 12,
    "VehicleColor" : [128, 128, 128, 255],
  }
}
```

```

        "VehicleColorState" : 0,
        "PlateColor" : [128, 128, 128, 255],
        "PlateColorState" : 0,
        "Name" : "ZhangSan",
        "Sex" : "Male",
        "CertificateType" : "IC",
        "ID" : "123456789",
        "Country" : "CN",
        "Province" : "XXX",
        "City" : "YYY",
        "HomeAddress" : "ZZZ Road",
        "Email" : "user@example.com",
        "PhoneNo" : "13xxxxx5678",
    }
}

```

Response Params (JSON format in body)

Name	Type	R/O	Description	Example
token	uint	R	Obtained search token	23443U
totalCount	int	R	Total number of qualified results	3333

Response Example

```
{
    "token": 23443U,
    "totalCount": 3333
}
```

10.7.9 Obtaining Vehicle Search Results

Request URL	http://<server>/cgi-bin/api/VehicleRegisterDB/doFind		
Method	POST		

Request Params (JSON format in body)

Name	Type	R/O	Description	Example
condition	object	R	Input parameter of face search	
+token	uint	R	Search token	23443U
+beginNumber	uint32	R	Start number of the search. The search starts from the “beginNumber” records, and returns predefined number (count) of records. 0<=beginNumber<= totalCount-1	0
+count	int	R	Number of entries obtained each time	20

Request Example

```
{
    "condition": {
        "token": 23443U,
        "beginNumber": 0,
        "count": 20
    }
}
```

}

Response Params (JSON format in body)

Name	Type	R/O	Description	Example
results	object	R	Output parameter of the search	
+found	int	R	Number of searched entries. One vehicle can be registered to multiple databases.	12
+candidates	object[]	O	Information list of vehicles to be selected	
++Vehicle	object	R	Vehicle information	Vehicle
++DifferentAttr ibutres	char[16][16]	R	A collection of attributes that do not match the database. The license plate must be the same, but other elements might be different for false-registered vehicle. Country: License plate registered place Brand: Vehicle logo Type: Vehicle type VehicleColor: Vehicle color PlateColor: Plate color	["VehicleColor", "Type"]

Response Example

```
{
  "results": {
    "found": 12,
    "candidates": [
      {
        "Vehicle": {Vehicle
        },
        "DifferentAttributres": ["VehicleColor", "Type"]
      },...
    ]
  }
}
```

10.7.10 Stopping Searching for Vehicles

Request URL http://<server>/cgi-bin/api/VehicleRegisterDB/stopFind

Method POST

Request Params (JSON format in body)

Name	Type	R/O	Description	Example
token	uint	R	Search token	23443U

Request Example

```
{
  "token": 23443U
}
```

Response Params (JSON format in body)

Name	Type	R/O	Description	Example
------	------	-----	-------------	---------

Response Example

{}

10.7.11 Add PlateNumberLib Download Task

Request URL	http://<server>/cgi-bin/api/NetFileTransfer/addPlateNumberLibDownloadTask		
Method	POST		
Request Params (JSON format in body)			
Name	Type	R/O	Description
PlateNumberLibVersion	char[32]	O	PlateNumber Lib Version
PacketTotal	uint	R	total package count
PacketIndex	uint	R	packet index, starts from 1
TaskInfo	object[100]	R	task info
+URLList	char[4][256]	R	resource url list ["sftp://admin:admin@10.12.4.84:554/media/1.txt", "xx"]

Request Example

```
{
    "PlateNumberLibVersion": ,
    "PacketTotal": 10,
    "PacketIndex": 3,
    "TaskInfo": [
        "URLList": ["sftp://admin:admin@10.12.4.84:554/media/1.txt", "xx"]
    ], ...
}
```

Response Params (JSON format in body)

Name	Type	R/O	Description	Example
Response Example				
{}				

11.1 Thermography Manager

11.1.1 Get Capability of Thermography

Request URL	http://<server>/cgi-bin/ThermographyManager.cgi?action=getCaps		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
channel	int	R	video channel index which starts from 1.
Request Example			
http://192.168.1.108/cgi-bin/ThermographyManager.cgi?action=getCaps&channel=1			

Response Params (key=value format in body)			
Name	Type	R/O	Description
+PresetModes	char[][32]	O	the preset mode. Range is { "Indoor", "Outdoor", "Default" }
+Brightness	object	O	brightness
++Max	int	O	the maximum
++Min	int	O	the mminimum
++Step	int	O	step
+Sharpness	object	O	Sharpness
++Max	int	O	the maximum
++Min	int	O	the mminimum
++Step	int	O	step
+EZoom	object	O	zoom
++Max	int	O	the maximum
++Min	int	O	the mminimum
++Step	int	O	step
+ThermographyGamma	object	O	ThermographyGamma
++Max	int	O	the maximum
++Min	int	O	the mminimum
++Step	int	O	step
+SmartOptimizer	object	O	smart optimizer
++Max	int	O	the maximum
++Min	int	O	the mminimum
++Step	int	O	step
+Agc	object	O	agc

++Max	int	O	the maximum	255
++Min	int	O	the mminimum	0
++Step	int	O	step	5
+AgcMaxGain	object	O	agc max gain	
++Max	int	O	the maximum	255
++Min	int	O	the mminimum	0
++Step	int	O	step	5
+AgcPlateau	object	O	agc plateau	
++Max	int	O	the maximum	100
++Min	int	O	the mminimum	0
++Step	int	O	step	5
+PresetColorization	enumchar[] [16]	O	preset colorization mode enumchar[] [16]{ "White Hot" "Black Hot" "Fusion" "Rainbow" "Globow" "Ironbow1" "Ironbow2" "Sepia" "Color1" "Color2" "Icefir" "Rain" "Red Hot" "Green Hot" "Spring" "Summer" "Winter" "Autumn" "Alarm" }	["Ironbow2", "IceFire"]
+PresetROIModes	char[] [16]	O	Preset ROI mode. Range is {"Full Screen", "Sky", "Ground", "Horizon", "Center 75%", "Center 50%", "Center 25%", "Custom"}	["Full Screen", "Sky", "Ground", "Horizon", "Center 75%", "Center 50%", "Center 25%", "Custom"]
+FFCPeriod	object	O	FFC period	
++MaxN	int	O	Maximum value in N-mode video mode	1000
++MaxP	int	O	Maximum value in P-mode video mode	1200
++Min	int	O	the mminimum	5
++Step	int	O	step	1
+FliterMode	char[] [16]	O	range is {"RawData", "YData"}	["RawData", "YData"]

+CamType	enumint8	O	enumint8{ 0: tau 1: lepton 2: Self-Developed }	0
+BaseBrightness	object	O	Base brightness	
++Max	int	O	the maximum	100
++Min	int	O	the mminimum	0
++Step	int	O	step	5
+StretchIntensity	object	O	Stretch intensity	
++Max	int	O	the maximum	100
++Min	int	O	the mminimum	0
++Step	int	O	step	5
+LCEValue	object	O	LCEValue	
++Max	int	O	the maximum	128
++Min	int	O	the mminimum	0
++Step	int	O	step	5
+LCESupport	bool	O	LCE	false
+HistGramValue	object	O	histogram	
++Max	uint	O	the maximum	32
++Min	uint	O	the mminimum	0
++Step	uint	O	step	1
+WideDynamicSupport	bool	O	does support WideDynamic	false
+SupportGainModes	char[][32]	O	range is: { GainModeAuto, GainModeLowTemp, GainModeHighTemp }	["GainModeAuto", "GainModeLowTemp", "GainModeHighTemp"]
+HeatMapTypes	char[16][32]	O	range is: "Gray", "Matrix"}	["Gray", "Matrix"]

Response Example

```

caps.PresetModes = Indoor
caps.Brightness.Max = 100
caps.Brightness.Min = 0
caps.Brightness.Step = 1
caps.Sharpness.Max= 100
caps.Sharpness.Min = 0
caps.Sharpness.Step = 5
caps.EZoom.Max= 24
caps.EZoom.Min = 0
caps.EZoom.Step = 1
caps.ThermographyGamma.Max= 8
caps.ThermographyGamma.Min = -8
caps.ThermographyGamma.Step = 1

```

```

caps.SmartOptimizer.Max= 100
caps.SmartOptimizer.Min = 0
caps.SmartOptimizer.Step = 5
caps.Agc.Max= 255
caps.Agc.Min = 0
caps.Agc.Step = 5
caps.AgcMaxGain.Max= 255
caps.AgcMaxGain.Min = 0
caps.AgcMaxGain.Step = 5
caps.AgcPlateau.Max= 100
caps.AgcPlateau.Min = 0
caps.AgcPlateau.Step = 5
caps.PresetColorization[0]= Ironbow2
caps.PresetROIModes[0]= Full Screen

```

11.1.2 [Config] Thermography Options

Config Data Params				
Name	Type	R/O	Description	Example
Thermography Options	object[][]	O	thermography options config It is a two-dimensional array, with the first dimension representing the channel	
+EZoom	int	O	Range is [0—24]. Range and step are got from interface in "11.1.1 Get Capability of Thermography".	12
+ThermographyGamma	int	O	Gamma -8~8 Range is got from interface in "11.1.1 Get Capability of Thermography".	0
+Colorization	char[16]	O	"White Hot", "Black Hot", "Ironbow2", "IceFire" Range is got from interface in "11.1.1 Get Capability of Thermography".	"White Hot"
+SmartOptimizer	int	O	range is 0 ~100	10
+OptimizedRegion	object	O		
++Type	char[16]	O	Range is {"Full Screen", "Sky", "Ground", "Horizontal", "Center 75%",	"Full Screen"

			"Center 50%", "Center 25%", "Custom" }	
++Enable	bool	O	true: enable false: not enable	true
++Regions	Rect[]	O	region array Range is [0—8191].	[Rect,]
+Agc	uint8	O	Range is [0—255]. Range and step are got from interface in "11.1.1 Get Capability of Thermography".	10
+AgcMaxGain	uint8	O	Range is [0—255]. Range and step are got from interface in "11.1.1 Get Capability of Thermography".	10
+AgcPlateau	uint8	O	Range and step are got from interface in "11.1.1 Get Capability of Thermography".	10
+HighTempGa inMode	object	O		
++Agc	uint8	O		10
++AgcMaxGai n	uint8	O		10
++AgcPlateau	uint8	O	Agc Plateau	10
+Mode	char[32]	O	mode, range is { "HighTemperature", "LowTemperature", "Auto"}	"HighTemperature"
+Auto	object	O		
++LowToHigh	uint32	O		13
++LHROI	uint32	O	percentage range is[0—100]	15
++LowToHigh Delay	uint32	O		10
++HighToLow	uint32	O		12
++HLROI	uint32	O	percentage range is[0—100]	95
++HighToLow Delay	uint32	O		10
+ContrastRect	Rect	O	region, Range is [0—8191].	
+BaseBrightne ss	uint32	O	Base brightness	0
+StretchIntens ity	uint32	O	Stretch intensity	0

Please refer to "4.2.1 Obtaining and Setting Configuration" to obtain and set the configuration. The specific examples are as follows:

Get Config Request Example

<http://10.0.0.8/cgi-bin/configManager.cgi?action=getConfig&name=ThermographyOptions>

Get Config Response Example

```
table.ThermographyOptions[0][0].EZoom=0
table.ThermographyOptions[0][0].Colorization=White Hot
table.ThermographyOptions[0][0].SmartOptimizer=10
table.ThermographyOptions[0][0].OptimizedRegion.Type=Custom
table.ThermographyOptions[0][0].OptimizedRegion.Enable= true
table.ThermographyOptions[0][0].OptimizedRegion.Regions[i][0u]=0
table.ThermographyOptions[0][0].OptimizedRegion.Regions[i][1u]=0
table.ThermographyOptions[0][0].OptimizedRegion.Regions[i][2u]=0
table.ThermographyOptions[0][0].OptimizedRegion.Regions[i][3u]=0
table.ThermographyOptions[0][0].Agc=10
table.ThermographyOptions[0][0].AgcMaxGain=10
table.ThermographyOptions[0][0].AgcPlateau=10
table.ThermographyOptions[0][0].Mode="HighTemperature"
table.ThermographyOptions[0][0].Auto.LowToHigh=13
table.ThermographyOptions[0][0].Auto.LHROI=15
table.ThermographyOptions[0][0].Auto.HighToLow=12
table.ThermographyOptions[0][0].Auto.HLROI=95
```

Set Config Request Example

http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&ThermographyOptions[0][0].Optimize
dRegion.Type=Gound

Set Config Response Example

OK

11.1.3 Get ExternSystem Information

Request URL	http://<server>/cgi-bin/ThermographyManager.cgi?action=getExternSystemInfo			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
channel	int	R	video channel index which starts from 1.	1
Request Example				
http://192.168.1.108/cgi-bin/ThermographyManager.cgi?action=getCaps&channel=1				

Response Params (key=value format in body)

Name	Type	R/O	Description	Example
sysInfo	object	R		
+SerialNumber	char[]	O	Serial Number	"11111"
+SoftwareVersion	char[]	O	Software Version	"11111"
+FirmwareVersion	char[]	O	Firmware Version	"11111"
+LibVersion	char[]	O	Lib Version	"11111"
Response Example				
sysInfo.SerialNumber = 11111111123				
sysInfo.SoftwareVersion = 2222222222222				

```

sysInfo.FirmwareVersion= 3333333333333
sysInfo.LibVersion = 44444444444

```

11.1.4 Get Information of Preset Mode

Request URL	http://<server>/cgi-bin/ThermographyManager.cgi?action=getPresetParam			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
channel	int	R	video channel index which starts from 1.	1
mode	char[]	R	mode, range is got from interface in "11.1.1 Get Capability of Thermography".	Default
Request Example				
http://192.168.1.108/cgi-bin/ThermographyManager.cgi?action=getPresetParam&channel=1&mode=Default				

Response Params (key=value format in body)				
Name	Type	R/O	Description	Example
presetInfo	object	O		
+Brightness	int	O	Brightness	50
+Sharpness	int	O	Sharpness	50
+EZoom	int	O	EZoom	12
+ThermographyGamma	int	O	Gamma	0
+Colorization	char[16]	O	Colorization	"White Hot"
+SmartOptimizer	int	O	SmartOptimizer	10
+OptimizedRegion	object	O		
++Type	char[16]	O	OptimizedRegion type	"Full Screen"
++Enable	bool	O	enable/disable	true
++Regions	Rect[]	O	regions	[[0,0,0],]
+Agc	uint32	O	agc	10
+AgcMaxGain	uint32	O	AgcMaxGain	10
+AgcPlateau	uint32	O	AgcPlateau	10
Response Example				
presetInfo.Brightness = 50 presetInfo.Sharpness= 50 presetInfo.EZoom= 12 presetInfo.ThermographyGamma= 0 presetInfo.Colorization= White Hot presetInfo.SmartOptimizer= 10 presetInfo.OptimizedRegion.Type= Full Screen presetInfo.OptimizedRegion.Enable=true presetInfo.OptimizedRegion.Regions[0][0]=0				

```

presetInfo.OptimizedRegion.Regions[0][1]=0
presetInfo.OptimizedRegion.Regions[0][2]=0
presetInfo.OptimizedRegion.Regions[0][3]=0
presetInfo.Agc= 10
presetInfo.AgcMaxGain=10
presetInfo.AgcPlateau = 10

```

11.1.5 Get Optimized Region Information

Request URL	http://<server>/cgi-bin/ThermographyManager.cgi?action=getOptimizedRegion			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
channel	int	R	video channel index which starts from 1.	1
Request Example				
http://192.168.1.108/cgi-bin/ThermographyManager.cgi?action=getOptimizedRegion&channel=1				

Response Params (key=value format in body)				
Name	Type	R/O	Description	Example
optimizedRegion	object	O		
+Type	char[]	O	OptimizedRegion type	Full Screen
+Enable	bool	O	enable/disable	true
+Regions	rect[]	O	the region is a rectangle	[]
Response Example				
optimizedRegion.Type= Full Screen optimizedRegion.Enable= true optimizedRegion.Regions[0][0]=0 optimizedRegion.Regions[0][1]=0 optimizedRegion.Regions[0][2]=0 optimizedRegion.Regions[0][3]=0				

11.1.6 Enable Shutter

Request URL	http://<server>/cgi-bin/ThermographyManager.cgi?action=enableShutter			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
channel	int	R	video channel index which starts from 1.	1
enable	bool	R	true or false, enable or not.	true
Request Example				
http://192.168.1.108/cgi-bin/ThermographyManager.cgi?action=enableShutter&channel=1&enable=true				

Response Params (OK in body)				
Name	Type	R/O	Description	Example

Response Example

OK

11.1.7 Fix Focus

Request URL	http://<server>/cgi-bin/ThermographyManager.cgi?action=fixFocus			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
linkVideoChannel	int[]	R	video channel index which starts from 1.	[1,2]
speed	float	O	range is 0.0-1.0.	0.5
Request Example				
http://192.168.1.108/cgi-bin/ThermographyManager.cgi?action=fixFocus&linkVideoChannel[0]=1&linkVideoChannel[1]=2				

Response Params (OK in body)

Name	Type	R/O	Description	Example
Response Example				
OK				

11.1.8 Do Flat Field Correction

Request URL	http://<server>/cgi-bin/ThermographyManager.cgi?action=doFFC			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
channel	int	R	video channel index which starts from 1.	1
Request Example				
http://192.168.1.108/cgi-bin/ThermographyManager.cgi?action=doFFC&channel=1				

Response Params (OK in body)

Name	Type	R/O	Description	Example
Response Example				
OK				

11.2 Radiometry

11.2.1 Get Capability of Radiometry

Request URL	http://<server>/cgi-bin/RadiometryManager.cgi?action=getCaps
--------------------	--

Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
channel	int	R	video channel index which starts from 1.	1
Request Example				
http://192.168.1.108/cgi-bin/RadiometryManager.cgi?action=getCaps&channel=1				

Response Params (key=value format in body)				
Name	Type	R/O	Description	Example
caps	object	R	caapabilities	
+TotalNum	object	O	total	
++MaxNum	uint	O	max num	8
++Spot	object	O	spot	
+++MaxSpots	uint	O	max spot number	8
++Line	object	O	line	
+++MaxLines	int	O	max line number	1
++Area	object	O	area	
+++MaxAreas	int	O	max area number	8
+TemperPresets	object	O		
++MaxPresets	int	O	max preset number	256
+MeterInfo	object	O	meter	
++Type	char[][16]	O	range:["Spot","Line","Area"]	["Spot","Line","Area"]
++ObjectEmissivity	object	O	Emissivity	
+++Max	int	O	the maximum	100
+++Min	int	O	the mminimum	0
+++Default	int	O	default	0
+++Step	int	O	step	1
++ObjectDistanceMeter	object	O	distance	
+++Max	int	O	the maximum	100
+++Min	int	O	the mminimum	0
+++Default	int	O	default	0
+++Step	int	O	step	1
++ReflectedTemperature	object	O	reflect temperature	
+++Max	int	O	the maximum	100
+++Min	int	O	the mminimum	0
+++Default	int	O	default	0
+++Step	int	O	step	1
++RelativeHumidity	object	O	Relative humidity	
+++Max	int	O	the maximum	100
+++Min	int	O	the mminimum	0
+++Default	int	O	default	0

+++Step	int	O	step	1
++AtmosphericTemperature	object	O	Atmospheric temperature	
+++Max	int	O	the maximum	100
+++Min	int	O	the minimum	0
+++Default	int	O	default	0
+++Step	int	O	step	1
+Statistics	object	O		
++MinPeriod	int	O	Min period	60
+Isotherm	object	O	Isotherm parameters	
++MaxTemp	int	O	the maximum	327.0
++MinTemp	int	O	the minimum	-20.0

Response Example

```

caps.TotalNum.MaxNum=8
caps.TotalNum.Spot.MaxSpots=8
caps.TotalNum.Line.MaxLines=1
caps.TotalNum.Area.MaxAreas=8
caps.TemperPresets.MaxPresets=256
caps.MeterInfo.Type[0u]=Spot
caps.MeterInfo.Type[1u]=Area
caps.MeterInfo.ObjectEmissivity.Max=100
caps.MeterInfo.ObjectEmissivity.Min=0
caps.MeterInfo.ObjectEmissivity.Default=0
caps.MeterInfo.ObjectEmissivity.Step=1
caps.MeterInfo.ObjectDistanceMeter.Max=100
caps.MeterInfo.ObjectDistanceMeter.Min=0
caps.MeterInfo.ObjectDistanceMeter.Default=0
caps.MeterInfo.ObjectDistanceMeter.Step=1
caps.MeterInfo.ReflectedTemperature.Max=100
caps.MeterInfo.ReflectedTemperature.Min=0
caps.MeterInfo.ReflectedTemperature.Default=0
caps.MeterInfo.ReflectedTemperature.Step=1
caps.MeterInfo.RelativeHumidity.Max=100
caps.MeterInfo.RelativeHumidity.Min=0
caps.MeterInfo.RelativeHumidity.Default=0
caps.MeterInfo.RelativeHumidity.Step=1
caps.MeterInfo.AtmosphericTemperature.Max=100
caps.MeterInfo.AtmosphericTemperature.Min=0
caps.MeterInfo.AtmosphericTemperature.Default=0
caps.MeterInfo.AtmosphericTemperature.Step=1
caps.Statistics.MinPeriod=60
caps.Isotherm.MaxTemp=327.0
caps.Isotherm.MinTemp=-20.0

```

11.2.2 [Config] Heat Image Thermometry

Config Data Params

Name	Type	R/O	Description	Example
HeatImagingThermometry	object	O	HeatImaging Thermometry	
+RelativeHumidity	uint32	O	The Relative Humidity range and step are got from interface in getCaps .	50
+AtmosphericTemperature	float	O	The Atmospheric Temperature range and step are got from interface in getCaps .	20
+ObjectEmissivity	float	O	The Object Emissivity range and step are got from interface in getCaps .	1.0
+ObjectDistance	uint32	O	The Object Distance range and step are got from interface in getCaps . Unit is meter.	100
+ReflectedTemperature	float	O	The Reflected Temperature range and step are got from interface in getCaps	20.1
+TemperatureUnit	char[16]	O	Range is {Centigrade, Fahrenheit}.	"Centigrade"
+Isotherm	object	O	Isotherm	
++Enable	bool	O	true or false	true
++MinLimitTemp	int32	O	MinValue range is got form interface in getCaps . MinValue must be smaller than MaxValue.	100
++MediumTemp	int32	O	medium value	120
++MaxLimitTemp	int32	O	MaxValue range is got form interface in getCaps . MaxValue must be bigger than MinValue	140
++SaturationTemp	int32	O	Saturation temperature value	160
++ColorBarDisplay	bool	O	true or false	true
++Rect	Rect	O	range: 0-8191	
+HotSpotFollow	bool	O	true or false	true
+TemperEnable	bool	O	true or false	true
+HotSpotColorMode	char[16]	O	range is {"Auto" "Manual" }	"Manual"
+HighCTMakerColor	Color	O		
+LowCTMakerColor	Color	O		
+Altitude	float	O	unit: m	150.0
+Visibility	float	O	unit: km	5.0
+RainFallCapacity	float	O	unit:mm	0.0
+ThermometryMode	int	O	Temperature measurement mode: 0. Body surface mode	1

			1. Body temperature mode	
+HotSpotOsdShow	bool	O	Is the hotspot temperature OSD displayed	false
+ColdSpotFollow	bool	O	true or false	false

Please refer to "4.2.1 Obtaining and Setting Configuration" to obtain and set the configuration. The specific examples are as follows:

Get Config Request Example

```
http://10.0.0.8/cgi-bin/configManager.cgi?action=getConfig&name=HeatImagingThermometry
```

Get Config Response Example

```
table.HeatImagingThermometry.RelativeHumidity=50
table.HeatImagingThermometry.AtmosphericTemperature=20.0
table.HeatImagingThermometry.HotSpotOsdShow=false
```

Set Config Request Example

```
http://10.0.0.8/cgi-bin/configManager.cgi?action=setConfig&HeatImagingThermometry.RelativeHumidity=50
```

Set Config Response Example

OK

11.2.3 [Config] Thermometry Rule

Config Data Params				
Name	Type	R/O	Description	Example
ThermometryRule	object[][]	O	Temperature measurement rule configuration It is a two-dimensional array, with the first dimension being the channel number and the second dimension being the temperature measurement point rule, supporting multi-channel and multiple types of temperature measurement	
+Enable	bool	O	Enable/Disable	true
+PresetId	int	O	Range [0—PresetMax] PresetMax is got from interface in GetCurrentProtocolCaps .	0
+RuleId	int	O	Range [0—MaxNum] MaxNum is got from interface in getCaps .	0
+Name	char[64]	O	Radiometry rule name.	"xxxx"
+Type	char[16]	O	Range is {Spot, Line, Area}.	"Spot"
+MeterRegion	object	O	position	
++Coordinates	Point[]	O	List of temperature measurement point coordinates The point format is [x, y], where x and y are coordinate values of type int	[Point,]

			Using a relative coordinate system, with a value range of 0~8191	
+T	int	O	Temperature Sample period. Unit is Second.	3
+AlarmSetting	object[]	O	alarm setting	
++Id	int	O	Range [0—65535],unique alarm id	0
++Enable	bool	O	Enable/Disable	true
++Result	char[16]	O	Depend on the vaule of Type Spot : {Vaule} Line: { Max, Min, Aver} Area: {Max, Min, Aver, Std, Mid, ISO}	"Max"
++AlarmCondition	char[16]	O	Range is {Below, Match , Above }	"Below"
++Threshold	float	O	Alarm threshold	20.0
++Hysteresis	float	O	Alarm hysteresis	0.1
++Duration	int	O	The duration time of alarm. Unit is second	30
++PreThreshoId	float	O	PreAlarm threshold	10.0
++PreDuration	uint32	O	The duration time of preAlarm. Unit is second	30
++SlopeResult	enumchar[16]	O	enumchar[16]{ "Aver", "Max", "Min" }	"Aver"
+LocalParameters	object	O	Local parameters	
++Enable	bool	O	Enable/Disable	true
++ObjectEmissivity	float	O	Range [0 — 1] Accuracy is 0.01	0.95
++ObjectDistance	uint	O	Object distance The range is got from interface in getCaps .	10
++ReflectedTemp	float	O	Object Reflected Temperature The range is got from interface in getCaps .	0

Please refer to "4.2.1 Obtaining and Setting Configuration" to obtain and set the configuration. The specific examples are as follows:

Get Config Request Example

```
http://10.0.0.8/cgi-bin/configManager.cgi?action=getConfig&name=ThermometryRule
```

Get Config Response Example

```
table.ThermometryRule[0][0].Enable = true
table.ThermometryRule[0][0].PresetId =0
table.ThermometryRule[0][0].RuleId=0
table.ThermometryRule[0][0].Name=SpotName
```

```

table.ThermometryRule[0][0].Type=Spot
table.ThermometryRule[0][0].MeterRegion.Coordinates[PointNo][0]= 0
table.ThermometryRule[0][0].MeterRegion.Coordinates[PointNo][1]= 0
...
table.ThermometryRule[0][0].T=3
table.ThermometryRule[0][0].AlarmSetting[0].Id=0
table.ThermometryRule[0][0].AlarmSetting[0].Enable=true
table.ThermometryRule[0][0].AlarmSetting[0].Result =Max
table.ThermometryRule[0][0].AlarmSetting[0].AlarmSetting[0]Condition=Below
table.ThermometryRule[0][0].AlarmSetting[0].Threshold=20.0
table.ThermometryRule[0][0].AlarmSetting[0].PreThreshold=10.0
table.ThermometryRule[0][0].AlarmSetting[0].PreDuration=30
table.ThermometryRule[0][0].AlarmSetting[0].Hysteresis=0.1
table.ThermometryRule[0][0].AlarmSetting[0].Duration=30
table.ThermometryRule[0][0].LocalParameters.Enable=true
table.ThermometryRule[0][0].LocalParameters.ObjectEmissivity=0.95
table.ThermometryRule[0][0].LocalParameters.ObjectDistance=0.95
table.ThermometryRule[0][0].LocalParameters.ReflectedTemp=0

```

Set Config Request Example

`http://10.0.0.8/cgi-bin/configManager.cgi?action=setConfig&ThermometryRule[0][0].Name=name1`

Set Config Response Example

OK

11.2.4 [Config] Heat Image Temper Event

Config Data Params

Name	Type	R/O	Description	Example
HeatImagingTemper	object[]	O	Heat Imaging Temper config A one-dimensional array, where the subscript of the array represents the channel number	
+Enable	bool	R	Enable/Disable Heat Imaging Temper feature.	false
+EventHandler	EventHandler	O	Setting of EventHandler is described in SetEventHandler .	

Please refer to "4.2.1 Obtaining and Setting Configuration" to obtain and set the configuration. The specific examples are as follows:

Get Config Request Example

`http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=HeatImagingTemper`

Get Config Response Example

table.HeatImagingTemper[0].Enable=false

table.HeatImagingTemper[0].EventHandler....(output of EventHandler is described in **SetEventHandler**)

table.HeatImagingTemper[1]....

...

Set Config Request Example

```
http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&HeatImagingTemper[0].Enable=true
```

Set Config Response Example

```
OK
```

11.2.5 Get Temperature of Particular Point

Request URL	http://<server>/cgi-bin/RadiometryManager.cgi?action=getRandomPointTemper		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
channel	int	R	video channel index which starts from 1
coordinate	int[2]	R	Coordinate of temperature measurement point
Request Example			
<pre>http://192.168.1.108/cgi-bin/RadiometryManager.cgi?action=getRandomPointTemper&channel=1&coordinate[0]=1024&coordinate[1]=1024</pre>			

Response Params (key=value format in body)			
Name	Type	R/O	Description
TempInfo	object	O	temperature
+Type	char[]	O	type, range is: {"Spot", "Area", "Line"}, it is fixed to spot
+TemperAver	float	O	temperature
Response Example			
<pre>TempInfo.Type=Spot TempInfo.TemperAver=27.5</pre>			

11.2.6 Get Temperature of Particular Condition

Request URL	http://<server>/cgi-bin/RadiometryManager.cgi?action=getTemper		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
condition	object	O	condition
+PresetId	int	O	Range [0- PresetMax] PresetMax is got from interface in GetCurrentProtocolCaps .
+RuleId	int	O	Range [0- MaxNum] MaxNum is got from interface in getCaps
+Type	enumchar[16]	O	type. enumchar[16]{ "Spot"

			"Area" "Line" }	
+Name	char[64]	O	Name is got from interface in GetThermometryRuleConfig	"xxxx"
+Channel	int	O	Video channel index. Start from 1	0
Request Example				
http://192.168.1.108/cgi-bin/RadiometryManager.cgi?action=getTemper&condition.PresetId=0&condition.RuleId=0&condition.Type=Spot&condition.Name=Spot1&condition.channel=1				

Response Params (key=value format in body)				
Name	Type	R/O	Description	Example
TempInfo	object	O	temperature info	
+Type	char[]	O	type. enumchar[16]{ "Spot" "Area" "Line" }	Spot
+TemperAver	float	O	temperature	27.5

Response Example

```
TempInfo.Type=Spot
TempInfo.TemperAver=27.5
```

11.2.7 Find Temperature Information

- Start to query temperature information

Request URL	http://<server>/cgi-bin/RadiometryManager.cgi?action=startFind			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
condition	object	O	condition	
+StartTime	char[]	O	The start time to find.	2010-04-01%200:00:00
+EndTime	char[]	O	The end time to find.	2010-04-08%200:00:00
+Type	char[]	O	The type of data. Range is {Spot, Line, Area}	"Spot"
+Period	int	O	Range is {5, 10, 15, 30}, minute	5
+channel	int	O	Video channel index. Start from 1	1
Request Example				
http://192.168.1.108/cgi-bin/RadiometryManager.cgi?action=startFind&condition.StartTime=2010-04-01%200:00:00&condition.EndTime=2010-04-08%200:00:00&condition.Type=Spot&condition.channel=1&condition.Period=5				

Response Params (key=value format in body)				
Name	Type	R/O	Description	Example

token	uint	R	query token	46878
totalCount	uint	R	total count	333
Response Example				
token=46878				
totalCount=333				

2. Get the data of temperature

Request URL	http://<server>/cgi-bin/RadiometryManager.cgi?action=doFind			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
token	uint	R	query token, get from interface of the first step above.	46878
beginNumber	uint	O	the begin index in this query	16
count	uint	O	the number you want to query.	16
Request Example				
http://192.168.1.108/cgi-bin/RadiometryManager.cgi?action=doFind&token=46878&beginNumber=16&count=16				

Response Params (key=value format in body)				
Name	Type	R/O	Description	Example
found	int	R	return info	12
+info	object[]	R		
++Time	char[20]	O	time	"2010-04-08 16:12:46"
++PresetId	int	O	preset id	0
++RuleId	int	O	rule id	0
++Type	char[16]	O	"Spot" "Area" "Line"	"Spot"
++Name	char[64]	O	name	"xxxx"
++Coordinate	Point[]	O	points, up to 8,using the 8192 relative coordinate system.	[Point,]
++Channel	int	O	Video channel index. Start from 1	1
++TemperatureUnit	char[16]	O	"Centigrade" "Fahrenheit"	"Centigrade"
++QueryTempInfo	object	O		
+++TemperAve	float	O	average temperature	50.1
+++TemperMax	float	O	maximum temperature	50.2
+++TemperMin	float	O	minimum temperature	50.0
+++MaxPoint	Point	O	Highest temperature point, using the 8192 relative coordinate system.	[0,0]
+++MinPoint	Point	O	lowest temperature point, using the 8192 relative coordinate system.	[0,0]
Response Example				
found=12				

```

info[i].Time=2010-04-08 16:12:46
info[i].PresetId=0
info[i].RuleId=0
info[i].Type=Spot
info[i].Name=xxxx
info[i].Coordinate[0]=1024
info[i].Coordinate[1]=2048
info[i].Channel=0
info[i].TemperatureUnit=Centigrade
info[i].QueryTemperInfo.TemperAve=50.1
info[i].QueryTemperInfo.TemperMax=50.2
info[i].QueryTemperInfo.TemperMin=50.0

```

3. Stop finding temperature information

Request URL	http://<server>/cgi-bin/RadiometryManager.cgi?action=stopFind		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
token	uint	R	query token, get from interface of the first step.
Request Example			
http://192.168.1.108/cgi-bin/RadiometryManager.cgi?action=stopFind&token=46878			

Response Params (OK in body)			
Name	Type	R/O	Description
Response Example			
OK			

11.2.8 Subscribe Temperature Information

Subscribe temperature information

Request URL	http://<server>/cgi-bin/RadiometryManager.cgi?action=attachTemper		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
channel	int	R	Video channel number, starting from 1
heartbeat	int	O	Heartbeat interval. The unit is second, and the default value is 5. When the device sends temperature 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".

Request Example

http://192.168.1.108/cgi-bin/RadiometryManager.cgi?action=attachTemper&channel=2&heartbeat=5

Response Params (multipart ; key=value format in body)

Name	Type	R/O	Description	Example
info	object[]	R	Temperature information array	
+Time	char[32]	R	Record time, format: yyyy-MM-dd HH:mm:ss	"2010-04-08 16:12:46"
+PresetId	int	O	Preset number, starting from 1	1
+RuleId	int	O	Rule number	1
+Type	char[16]	R	Search Type "Spot": Spot "Area": Area "Line": Line	"Spot"
+Name	char[64]	O	Query item name	"xxxx"
+Coordinate	int[8][2]	O	Query the coordinates of temperature monitoring point, up to 8, using the 8192 relative coordinate system.	[[1024, 2048]]
+Channel	int	O	Video channel number, starting from 0	0
+TemperatureUnit	char[16]	R	Temperature unit, value: "Centigrade" : Centigrade "Fahrenheit": Fahrenheit	"Centigrade"
+QueryTemperInfo	object	R	Temperature information	
++TemperAve	float	R	Average temperature	50.1
++TemperMax	float	R	Maximum temperature	50.2
++TemperMin	float	R	Minimum temperature	50.0

Response Example

HTTP/1.1 200 OK

Server: Device/1.0

Content-Type: multipart/x-mixed-replace; boundary=<boundary>

Connection: closed

--<boundary>

Content-Type: text/plain

Content-Length: <data length>

info[0].Time=2010-04-08 16:12:46

info[0].PresetId=1

info[0].RuleId=1

info[0].Type=Spot

info[0].Name=xxxx

info[0].Coordinate[0][0]=1024

info[0].Coordinate[0][1]=2048

info[0].Channel=0

info[0].TemperatureUnit=Centigrade

info[0].QueryTemperInfo.TemperAve=50.1

```

info[0].QueryTemperInfo.TemperMax=50.2
info[0].QueryTemperInfo.TemperMin=50.0
.....
--<boundary>
Content-Type: text/plain
Content-Length: 11

Heartbeat
--<boundary>
.....

```

11.2.9 Subscribe Radiometry Data

Subscribe temperature distribution data

Request URL	http://<server>/cgi-bin/RadiometryManager.cgi?action=attachProc			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
channel	int	R	Video channel number, starting from 1	1
heartbeat	int	O	Heartbeat interval. The unit is second, the value range is [1,60], the default value is 5. When the device sends temperature 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".	5

Request Example

http://192.168.1.108/cgi-bin/RadiometryManager.cgi?action=attachTemper&channel=2&heartbeat=5

Response Params (multipart ; key=value format in body)				
Name	Type	R/O	Description	Example
dataInfo	object	R	Temperature distribution data information	
+Width	int	R	Image width, in pixels	1920
+Height	int	R	Image height, in pixels	1080
+Channel	int	O	Video channel number, starting from 0	0
+Time	char[32]	R	Fetching data time, format: yyyy-MM-dd HH:mm:ss	"2010-04-08 16:12:46"
+Length	int	O	Data length	2073600
+sensorType	char[16]	O	Algorithm type, value: "Tau"	"Tau"
+Unzip	object	O	Decompression parameter, used when sensorType is Tau	
++ParamR	uint	O	Parameter R required for temperature conversion	1
++ParamB	uint	O	Parameter B required for	1

			temperature conversion	
++ParamF	uint	O	Parameter F required for temperature conversion	1
++ParamO	uint	O	Parameter O required for temperature conversion	1

Response Example

HTTP/1.1 200 OK

Server: Device/1.0

Content-Type: multipart/x-mixed-replace; boundary=<boundary>

Connection: closed

--<boundary>

Content-Type: text/plain

Content-Length: <data length>

dataInfo.Width=1920

dataInfo.Height=1080

dataInfo.Channel=0

dataInfo.Time=2010-05-25 00:00:00

dataInfo.Length=2073600

dataInfo.sensorType="Tau"

dataInfo.Unzip.ParamR=1

dataInfo.Unzip.ParamB=1

dataInfo.Unzip.ParamF=1

dataInfo.Unzip.ParamO=1

--<boundary>

Content-Type: application/octet-stream

Content-Length: <data length>

<Binary data>

--<boundary>

Content-Type: text/plain

Content-Length: 11

Heartbeat

--<boundary>

.....

11.2.10 Fetch Radiometry Data

Request URL	http://<server>/cgi-bin/RadiometryManager.cgi?action=toFetch			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
+channel	int	R	video channel index which starts from 1.	1
Request Example				

http://192.168.1.108/cgi-bin/RadiometryManager.cgi?action=toFetch&channel=2

Response Params (key=value format in body)

Name	Type	R/O	Description	Example
status	char[]	R	Range is {Ready, Busy}. "Ready"means service available and "Busy"means service busy.	Ready

Response Example

status=Ready

11.2.11 [Config] FireWarning Config

Config Data Params

Name	Type	R/O	Description	Example
FireWarning	object[][]	O	Fire warning configuration, used in conjunction with FireWarningMode configuration It is a two-dimensional array. The first dimension represents the video input channel.	
+SmdFilterEnable	bool	O	whether smd filter take effect	false
+Enable	bool	O	whether fire detect take effect	true
+PresetId	int	O	The PresetId	0
+Row	int	O	rows of fire detect area	32
+Col	int	O	cols of fire detect area	40
+Mode	char[16]	O	Range is {"Auto", "Normal"}.	"Auto"
+Type	char[16]	O	"Fire", "Smoke"	"Fire"
+TimeDurationEnable	bool	O	whether include fire detect duration,only take effect in SpaceExClude mode	false
+FireDuration	int	O	fire last times	60
+MovingTargetFilterEnable	bool	O	whether moving target filter take effect	false
+SunReflectEnable	bool	O	whether sun reflect filter take effect	false
+EventHandler	EventHandler	O	Setting of EventHandler is described in SetEventHandler.	
+DetectWindow	object[]	O		
++Regions	int64[]	O	detect area mask	[123456789..., 123456789,...]
++Postion	float[]	O	Spatial exclusion zone information, with the first dimension representing the spatial coordinate PTZ Valid when the fire warning mode is'	[0.0, 0.0, 1.0]

			SpaceExCloud '	
++TargetSize	int	O	target size Unit: Pixels	3
++Sensitivity	int	O	Rage {0,100}	5
++Id	int	O	Rage {0,...}	33
++Name	char[32]	O	detect window name	"Region1"

Please refer to "4.2.1 Obtaining and Setting Configuration" to obtain and set the configuration. The specific examples are as follows:

Get Config Request Example

```
http://10.0.0.8/cgi-bin/configManager.cgi?action=getConfig&name=FireWarning
```

Get Config Response Example

```
table.FireWarning[0][0].Enable=true
table.FireWarning[0][0].PresetId=0
table.FireWarning[0][0].Row = 31
table.FireWarning[0][0].Col = 40
table.FireWarning[0][0].Mode=Auto
table.FireWarning[0][0].SmdFilterEnable=false
table.FireWarning[0][0].MovingTargetFilterEnable=false
table.FireWarning[0][0].SunReflectEnable=false
table.FireWarning[0][0].TimeDurationEnable=false
table.FireWarning[0][0].FireDuration= 15
table.FireWarning[0][0].DetectWindow[windowsNum].Regions[0]=123468789
table.FireWarning[0][0].DetectWindow[windowsNum].Regions[1]=123468789
table.FireWarning[0][0].DetectWindow[windowsNum].Regions[2]=123468789
table.FireWarning[0][0].DetectWindow[windowsNum].Regions[3]=123468789
table.FireWarning[0][0].DetectWindow[windowsNum].Postion[0]=0
table.FireWarning[0][0].DetectWindow[windowsNum].Postion[1]=0
table.FireWarning[0][0].DetectWindow[windowsNum].Postion[2]=0
table.FireWarning[0][0].DetectWindow[windowsNum].Postion[3]=0
table.FireWarning[0][0].DetectWindow[windowsNum].Sensitivity = 95
table.FireWarning[0][0].DetectWindow[windowsNum].Id=1
table.FireWarning[0][0].DetectWindow[windowsNum].Name=windName
table.FireWarning[0][0].EventHandler=(output of EventHandler is described in GetEventHandler)
...

```

Set Config Request Example

```
http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&FireWarning[0][0].Eanble=false
```

Set Config Response Example

```
OK
```

11.2.12 [Config] FireWarningMode Config

Config Data Params				
Name	Type	R/O	Description	Example
FireWarningMode	object[]	O	Fire warning mode configuration Is an array representing video	

			channels, starting from 0 and supporting a maximum of 32 channels	
+Mode	char[16]	O	Rage is {"PtzPreset","SpaceExClude"}	"PtzPreset"

Please refer to "4.2.1 Obtaining and Setting Configuration" to obtain and set the configuration. The specific examples are as follows:

Get Config Request Example

```
http://10.0.0.8/cgi-bin/configManager.cgi?action=getConfig&name=FireWarningMode
```

Get Config Response Example

```
table.FireWarningMode[0].Mode="PtzPreset"
```

Set Config Request Example

```
http://10.0.0.8/cgi-bin/configManager.cgi?action=setConfig&FireWarningMode[0].Mode=SpaceExClude
```

Set Config Response Example

```
OK
```

11.2.13 Get Current Hot Cold Spot

Request URL	http://<server>/cgi-bin/TemperCorrection.cgi?action=getCurrentHotColdSpot			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
channel	int	R	video channel index which starts from 1, only thermography channels are valid.	1
Request Example				
http://192.168.1.108/cgi-bin/TemperCorrection.cgi?action=getCurrentHotColdSpot&channel=1				

Response Params (key=value format in body)				
Name	Type	R/O	Description	Example
info	object	O	The current hot code spot info.	
+HotPoint	int[2]	O	The hot spot position, must be two int, means x and y value, coordinate remap to 0 — 8192.	[1150, 2320]
+HotSpotValue	float	O	The hot spot temperature value.	35.5
+ColdPoint	int[2]	O	The cold spot position, must be two int, means x and y value, coordinate remap to 0 — 8192.	[5452, 6192]
+ColdSpotValue	float	O	The cold spot temperature value.	24.3
+TemperatureUnit	int	O	The temperature unit : 0 Centigrade, 1 Fahrenheit	0
Response Example				
info.HotPoint[0]=1150				

```

info.HotPoint[1]=2320
info.HotSpotValue=35.5
info.ColdPoint[0]=5452
info.ColdPoint[1]=6192
info.ColdSpotValue=24.3
info.TemperatureUnit=0

```

11.2.14 [Config] Heat Image Temper PreAlarm Event

Config Data Params				
Name	Type	R/O	Description	Example
PreAlarmEvent	object[]	O	Temperature pre alarm configuration, one-dimensional array. Video channel number, starting from 0	
+Enable	bool	R	Enable/Disable Heat Imaging Temper feature.	false
+EventHandler	EventHandler	O	Setting of EventHandler is described in SetEventHandler .	

Please refer to "4.2.1 Obtaining and Setting Configuration" to obtain and set the configuration. The specific examples are as follows:

Get Config Request Example

```
http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=PreAlarmEvent
```

Get Config Response Example

```

table.PreAlarmEvent[0].Enable=false
table.PreAlarmEvent[0].EventHandler....(output of EventHandler is described in SetEventHandler)
table.PreAlarmEvent[1]....
...

```

Set Config Request Example

```
http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&PreAlarmEvent[0].Enable=false&PreAlarmEvent[0].EventHandler.BeepEnable=false
```

Set Config Response Example

```
OK
```

11.2.15 Get Heat Map Info

Get Heat Map Info.

Request URL	http://<server>/cgi-bin/RadiometryManager.cgi?action=getHeatMapsDirectly			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
channel	int	R	video channel index which starts from 1	2
Request Example				

http://192.168.1.108/cgi-bin/RadiometryManager.cgi?action=getHeatMapsDirectly&channel=2

Response Params (multipart, binary in body)

Name	Type	R/O	Description	Example
dataInfo	object	O	Data info	
+Height	integer	O	height	0
+Width	integer	O	width	0
+Channel	integer	O	Channel number	0
+Time	string	O	Time of getting data	2010-05-25 00:00:00
+Length	integer	O	Data length	0
+sensorType	string	O	Algorithm type	"Tau"
+Unzip	object	O	unzip parameters, it works when sensorType was "Tau".	
++ParamR	integer	O	Parameter R for temperature conversion	1
++ParamB	integer	O	Parameter B for temperature conversion	1
++ParamF	integer	O	Parameter F for temperature conversion	1
++ParamO	integer	O	Parameter O for temperature conversion	1

Response Example

HTTP/1.1 200 OK

Server: Device/1.0

Content-Type: multipart/x-mixed-replace; boundary=<boundary>

Connection: closed

--<boundary>

Content-Type: text/plain

Content-Length: <data length>

dataInfo.Height=0

dataInfo.Width=0

dataInfo.Channel=0

dataInfo.Time=2010-05-25 00:00:00

dataInfo.Length=0

dataInfo.sensorType="Tau"

dataInfo.Unzip.ParamR=1

dataInfo.Unzip.ParamB=1

dataInfo.Unzip.ParamF=1

dataInfo.Unzip.ParamO=1

--<boundary>

Content-Type: application/octet-stream

Content-Length: <data length>

```
<Binary data>
```

```
--<boundary>
```

```
.....
```

11.2.16 Get Temperature of Particular Region

Request URL	http://<server>/cgi-bin/api/RadiometryManager/getRandomRegionTemper		
Method	POST		
Request Params (JSON format in body)			
Name	Type	R/O	Description
Channel	int	R	Video channel index. Start from 0
Polygon	int[8][2]	R	a polygon area. the first array is point list, minimum item is 3, the second array is point, must be two int, means x and y value, coordinate remap to 0 — 8192.
Request Example			
{ "Channel": 1, "Polygon": [[0,0], [192,3],[160,80]] }			

Response Params (JSON format in body)			
Name	Type	R/O	Description
RegionTempIn fo	object	R	Temperature information of the area
+Temperature Unit	char[32]	R	Temperature unit (currently configured temperature unit). There are two types of temperature units supported: "Centigrade" "Fahrenheit"
+TemperAver	int32	R	The parameter value of the average temperature in the temperature measurement area, with an accuracy of 0.01 and an increase of 100 times
+TemperMax	int32	R	The parameter value of the maximum temperature in the temperature measurement area, with an accuracy of 0.01 and an increase of 100 times
+TemperMin	int32	R	The parameter value of the minimum temperature in the temperature measurement area, with an accuracy of 0.01 and an increase of 100 times
+TemperMaxP oint	uint[2]	R	The point with the highest temperature, must be two int, means x and y value,

			coordinate remap to 0 — 8192.	
+TemperMinPoint	uint[2]	R	The point with the lowest temperature, must be two int, means x and y value, coordinate remap to 0 — 8192.	[200,600]
+RuleName	char[32]	O	rule name	"RuleName"

Response Example

```
{
    "RegionTempInfo": {
        "TemperatureUnit": "Centigrade",
        "TemperAver": 3100,
        "TemperMax": 3200,
        "TemperMin": 3000,
        "TemperMaxPoint": [100,300],
        "TemperMinPoint": [200,600],
        "RuleName": "RuleName"
    }
}
```

11.3 TemperCustom

11.3.1 Set Environment Temperature

Request URL	http://<server>/cgi-bin/TemperCustom.cgi?action=setEnvTemp			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
EnvironmentTemp	int	R	environment temperature Magnified 100 times, the unit is 0.01Celsius degree	2800

Request Example

http://192.168.1.108/cgi-bin/TemperCustom.cgi?action=setEnvTemp&EnvironmentTemp=2800

Response Params(OK in body)				
Name	Type	R/O	Description	Example
Response Example				
OK				

12 Access Control APIs

12.1 Access Control

12.1.1 Open Door

Open the door

Request URL	http://<server>/cgi-bin/accessControl.cgi?action=openDoor			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
channel	int	R	the index of door, starts from 1	1
UserID	int	O	remote user ID	101
Type	char[16]	O	the open type, default value is "Remote"	"Remote"
Request Example				
http://192.168.1.108/cgi-bin/accessControl.cgi?action=openDoor&channel=1&UserID=101&Type=Remote				

Response Params (OK in body)
Response Example
OK

12.1.2 Close Door

Close the door

Request URL	http://<server>/cgi-bin/accessControl.cgi?action=closeDoor			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
channel	int	R	the index of door, starts from 1.	1
UserID	int	O	remote user ID.	101
Type	char[16]	O	the open type, default value is "Remote".	"Remote"
Request Example				
http://192.168.1.108/cgi-bin/accessControl.cgi?action=closeDoor&channel=1&UserID=101&Type=Remote				

Response Params (OK in body)
Response Example
OK

12.1.3 Get Door Status

Get status of the door

Request URL	http://<server>/cgi-bin/accessControl.cgi?action=getDoorStatus		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
channel	int	R	the index of door, starts from 1
Request Example			
http://192.168.1.108/cgi-bin/accessControl.cgi?action=getDoorStatus&channel=1			

Response Params (key=value format in body)			
Name	Type	R/O	Description
Info	object	R	door status info
+status	char[16]	R	door status, the range is {Open, Break, Close}
Response Example			
Info.status=Open			

12.1.4 Get Lock Status

Request URL	http://<server>/cgi-bin/accessControl.cgi?action=getLockStatus		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
channel	int	O	the index of lock, starts from 1. Default is 1 if not present. Parameters in Response :
Request Example			
http://192.168.1.108/cgi-bin/accessControl.cgi?action=getLockStatus&channel=0			

Response Params (key=value format in body)			
Name	Type	R/O	Description
Info	object	O	lock info
+onLineStatus	char[]	O	the range is { "OnLine", "OffLine" }
+status	char[]	O	the range is { "Open", "Close", "Abnormal", "FakeLocked", "Unknown" }
Response Example			
Info.onLineStatus=Online			
Info.status=Close			

12.1.5 Capture Fingerprint

Only ID verification terminals such as ASHZ320/520 are supported. Common access control is not supported.

Capture fingerprint data, then report it via 'Fingerprint' event.

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 data	
+ReaderID	char[]	R	Reader machine ID	101
+FingerPrintName	char[]	O	Fingerprint name (For smart building products)	aaaa
+ UserID	char[]	O	User Identity (For smart building products)	10221
heartbeat	int	R	Heartbeat interval (sec)	10
timeout	int	R	Timeout (sec)	20
Request Example				
http://<server>/cgi-bin/accessControl.cgi?action=captureFingerprint&info.ReaderID=101&info.FingerPrintName=aaaa&info.UserID=10221&heartbeat=5&timeout=10				

Response Params (multipart , json format in body , Heartbeat in body)	
Name	Type
Description	
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: text/plain	
Content-Length: 238002	
Code=Fingerprint;action=Pulse;index=0;data={	
"CollectResult": true,	
"FingerPrintID": 60,	
"RecNo": 1234,	
"ReaderID": "1",	
"UserID": "001",	
"CardNo": "001",	
"FingerprintPacket": {	
"Length": 512,	
"Count": 3	
},	

```

        "FingerprintData": "xxx",
        "BinaryData": Binary Data
    }

```

```

--myboundary
Content-Type: text/plain
Content-Length: 9

Heartbeat

--myboundary
.....
```

12.1.6 Capture Face Picture

Only ID verification terminals such as ASHZ320/520 are supported. Common access control is not supported.

Capture face data, then report it via 'CitizenPictureCompare' event.

Request URL	http://<server>/cgi-bin/accessControl.cgi?action=captureCmd			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
+type	int	R	Capture entity types: 0x01 - Face 0x02 - Identification card	1
+UserID	string	O	User ID	10221
heartbeat	int	R	Heartbeat interval (sec)	5
timeout	int	R	Timeout (sec)	10
Request Example				
http://<server>/cgi-bin/accessControl.cgi?action=captureCmd&type=1&UserID=10221&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: text/plain				
Content-Length: 2380				

```

Code=CitizenPictureCompare;action=Pulse;index=0;data={
    "UTC": 1999999999,
    "CompareResult": true,
    "Similarity": 90,
    "Threshold": 80,
    "Citizen": "ZhangSan",
    "Sex": 1,
    "Minzu": 1,
    "Birth": "1980-01-01",
    ...
}

```

--myboundary

Content-Type: text/plain

Content-Length: 9

Heartbeat

--myboundary

.....

12.1.7 Query AccessControl Record

Request URL	http://<server>/cgi-bin/recordFinder.cgi?action=find			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
name	string	R	The record table name. It should be " AccessControlCardRec ".	"AccessControlCardRec"
count	int	O	Max result to return, default is 1024.	100
StartTime	string	O	The start of the record's CreateTime.	123456700
EndTime	string	O	The end of the record's CreateTime.	123456800
condition	object	O	Search condition.	
+CardNo	string	O	Access user card number.	123456
Request Example				
http://192.168.1.108/cgi-bin/recordFinder.cgi?action=find&name=AccessControlCardRec&StartTime=123456700&EndTime=123456800&condition.CardNo=12001&count=100				

Response Params (key=value format in body)				
Name	Type	R/O	Description	Example
totalCount	int	O	Total record num that find.	1000
found	int	O	Record num that returned.	100
records	object[]	R	The records that returned.	

+RecNo	int	R	Record id.	12345
+FaceIndex	uint8	O		0
+CreateTime	int	O	Record create time, UTC time.	123456789
+CardNo	int	R	Access user card number.	12001
+CardName	string	O	Access user card name, max string length is 31.	ZhangSan
+CardType	int	O	The card type: 0: Normal Card, 1: VIP Card, 2: Visitor Card, 3: Patrol Card, 4: Blocklist Card, 5: Stress Card, 0xff: Mother Card	0
+Password	string	O	The Access card's password.	123456
+UserID	string	R	The user's id.	ZhangSan
+Type	string	O	The event type. It can be: "Entry", "Exit".	Exit
+Status	int	O	Open door result. It can be: 0—failed, 1—success. If this param does not exist, that means success.	1
+Method	int	R	Open door method. It can be: <ul style="list-style-type: none">• 0: by password• 1: by access card• 2: by access card and then password• 3: by password and then access card• 6: by fingerprint• 15: by face recognition	1
+Door	int	O	The index of the door. (This param is not supported by video talk device)	5
+ReaderID	string	O	The access user card ID of reader. (This param is not supported by video talk device)	
+ErrorCode	int	O	The error code, valid only when Status is 0.	
+URL	string	O	The picture's URL, max string length is 127. (This param is not supported by access control device)	
+AttendanceStat e	enumint	O		1
+Similarity	uint8	O	Similarity	75
+SnapFaceURL	char[128]	O		"/mnt/109/20141104/1.jpg"
+CitizenPictureU RL	char[128]	O		"/mnt/109/20141104/1.jpg "

+RecordURL	string	O	The record video's URL, max string length is 127. (This param is not supported by access control device)	
+CitizenIDResult	bool	O	citizen ID compare result	true
+CitizenIDName	string	O	citizen name	Zhangsan
+CitizenIDNo	string	O	citizen ID number	34200000000000000000
+CitizenIDSex	int	O	sex:{ 0: unknow 1: male 2: female 9: not stated }	1
+CitizenIDMinzu	int	O	citizen ID Minzu	1
+CitizenIDBirth	string	O	birthday, format as "1980-01-01"	1980-01-01
+CitizenIDAddress	string	O	address	BinanRoad
+CitizenIDAuthority	string	O	signing and issuing organization	gggg
+CitizenIDStart	string	O	Effective date	1996-01-01
+CitizenIDEnd	string	O	Expiration date, and "Endless" mens Long term effectiveness	2006-01-01
+HatType	int	O		
+HatColor	char[16]	O	enum char[16]{ "Unknown" "Transparent" "Other" "White", "Orange", "Pink", "Black", "Red", "Yellow", "Gray", "Blue", "Green", "Purple", "Brown", "Silver", "Darkviolet", "Maroon", "Dimgray", "Whitesmoke", "Darkorange", "Mistyrose", "Tomato", "Olive", }	"Yellow"

			"Gold", "Darkolivegreen", "Chartreuse", "Greenyellow", "Forestgreen", "Seagreen", "Deepskyblue", "Cyan", "ShadowGreen" }	
+IsOverTemperature	bool	O	is over temperature	true
+TemperatureUnit	uint	O	Temperature unit (0 ° C, 1 ° F, 2 Kelvin)	0
+CurrentTemperature	float	O	current temperature	36.8
+RemainingTimeS	uint	O	remain time	200
+Mask	uint8	O	default 0 0: unknow 1 without mask 2 with mask	0
+Score	uint8	O	quality	3
+ButtonCheck	uint8	O		0
+QRCode	char[512]	O	QRCode	"111111111111"
VTONumber	char[32]	否	VTO number	"8001"
RoomNumber	char[32]	否	VTH room number	"9901"
Notes	char[32]	否	comments	
ReservedInt	int32	否	reserved	
ReservedString	char[32]	否	reserved	

Response Example

```

totalCount=1000
found=100
records[0].RecNo=12345
records[0].CreateTime=123456789
records[0].CardNo=12001
records[0].CardName=ZhangSan
records[0].UserID=ZhangSan
records[0].Type=Entry
records[0].Method=1
...
records[1].RecNo=13579
records[1].CreateTime=123456799
records[1].CardNo=12001
records[1].CardName=ZhangSan
records[1].UserID=ZhangSan
records[1].Type=Exit

```

```
records[1].Method=1
```

```
...
```

12.1.8 Query Access Control Alarm Record

Find the AccessControlAlarmRecord record. (This api is supported by access control device.)

Request URL	http://<server>/cgi-bin/recordFinder.cgi?action=find			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
name	char[]	R	fixed to "AccessControlAlarmRecord"	AccessControlAlarmRecord
StartTime	char[]	O	start time formt: 2014-8-25%2000:01:32	2014-8-25%2000:01:32
EndTime	char[]	O	end time formt: 2014-8-25%2000:02:32	2014-8-25%2000:02:32
count	int	O	record count	500
Request Example				
http://192.168.1.108/cgi-bin/recordFinder.cgi?action=find&name=AccessControlAlarmRecord&StartTime=2014-8-25%2000:02:32&EndTime=2014-8-25%2001:02:32&count=500				

Response Params (key=value format in body)				
Name	Type	R/O	Description	Example
totalCount	int	R	total count	200
found	int	R	return count	100
records	object[]	R	record info	
+RecNo	int	O	record number	1234
+CreateTime	int	O	UTC	12345678
+UserID	char[]	O	user ID	1254
+EventCode	char[]	O	Enumchar[32]{ DoorNotClosed BreakIn RepeatEnter Duress AlarmLocal ChassisIntruded MaliciousAccessControl AccessControlBlacklist }	AlarmLocal
+DevAddrs	int	O		1
+IndexNum	int	O	channel number which starts from 0	0
+Time	char[]	O	event occurred time (UTC with time zone and daylight saving time deviation)	2017-05-10 16:00:01

Response Example

```
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
...
...
```

12.1.9 [Event] AccessControl

Event Code	AccessControl		
Event action	Pulse		
Event index	0		
Event Data			
Name	Type	R/O	Description
+RecNo	int	R	The record id
+Name	string	O	The name of the door
+Type	string	O	The event type. It can be: "Entry", "Exit"
+Status	int	O	Open door result, can be: 0—failed, 1—success If this param does not exist, that means success.
+Method	int	R	Open door method, can be: 0: by password 1: by access card 2: by access card and then password 3: by password and then access card 6: by fingerprint 7: by password and access card and fingerprint together 8 : by password and fingerprint together 9 : by access card and fingerprint together 10: reserved 11: by multiple access user 12: by key 13: by duress password 14: by QR code, local

			15: by face recognition, local 16: reserved 17: by ID card 18: by face and ID card 19: by Bluetooth 20: by custom password 21: by UserId and password 22: by face and password 23: by fingerprint and password 24: by fingerprint and face 25: by access card and face 26: by face or password 27: by fingerprint or password 28: by fingerprint or face 29: by access card or face 30: by access card or fingerprint 31: by fingerprint and face and password 32: by access card and face and password 33: by access card and fingerprint and password 34: by access card and fingerprint and face 35: by fingerprint or face or password 36: by access card or face or password 37: by access card or fingerprint or face 38: by access card and fingerprint and face and password 39: by access card or fingerprint or face or password 40: by ID card and face, or access card or face 41: by ID card or QR code or face 42: by DTMF(SIPINFO, RFC2833, INBAND) 43: by QR code, remote 44: by face recognition, remote 45: by ID card (match with fingerprint in ID card) 46: by temporary password 47: by health code	
+CardNo	string	O	Card number if the door is opened by card	123456

+UserID	string	R	The user id	
+ErrorCode	integer	O	error code	
+ObjectProperties	object	O	Dynamic structure info	
		O	is with mask, it can be { 0: unknow 1: without mask 2: with mask }	0
+Mask	integer			
+ManTemperatureInfo	object	O	people's temperature info	
++CurrentTemperature	float	O	people's current temperature	36.7
		O	temperature unit:{ 0: centigrade 1: Fahrenheit 2: Kelvin }	0
++TemperatureUnit	integer			
++IsOverTemperature	bool	O	is over temperature	false

Event Response Example (multipart , key=value format in body , binary data in body) (response to 4.4.3 snap subscription)

--<boundary>
Content-Type: text/plain
Content-Length: <length>

Events[0].EventBaseInfo.Code=AccessControl
Events[0].EventBaseInfo.Action=Pulse
Events[0].EventBaseInfo.Index=0
Events[0].RecNo=123
Events[0].Name=Door1
Events[0].Type=Entry
Events[0].Status=1
Events[0].Method=1
Events[0].CardNo=09DDAABB
Events[0].UserID=101

...

--<boundary>
Content-Type: image/jpeg
Content-Length: <image size>

<Jpeg image data>
--<boundary>

12.1.10 [Event] CitizenPictureCompare

Event Code	CitizenPictureCompare			
Event action	Pulse			
Event index	0			
Event Data				
Name	Type	R/O	Description	Example
+UTC	uint32	O	UTC time	
+CompareResult	bool	O	compare result. If the similarity is greater than or equal to the threshold, the comparison is considered successful	
+Similarity	integer	O	Similarity of two pictures, Unit: percentage range[1, 100]	
+Threshold	integer	O	check threshold range[1, 100]	
+Citizen	String	O	citzen name	
+Sex	integer	O	sex, it can be: { 0 unknow 1 male 2 female 9 not stated }	
+Minzu	integer	O	Minzu (refer to CitizenIDCard)	
+Birth	String	O	birthday, such as "1980-01-01"	
+Address	String	O	address	
+Number	String	O	citizen ID	
+Authority	String	O	signing and issuing organization	
+Start	String	O	Effective date	
+End	String	O	Expiration date, and "Endless" mens Long term effectiveness	
+ImageInfo	array<object>	O	Picture information, the first for the face cutout, the second for the ID card photo	
++Offset	integer	O	the offset in binary data	
++Length	integer	O	length of picture, unit: Byte	
++Width	integer	O	Picture width, pixels	
++Height	integer	O	Picture height, pixels	
+ImageInfoEx	array<object>	O	extension of picture info, (The total number of images uploaded is determined by imageinfo and imageinfoex) 6 at most	
++Type	integer	O	picture type 0 Local face database	

			1 Shooting scene map	
++Offset	integer	O	the offset in binary data	
++Length	integer	O	length of picture, unit: Byte	
++Width	integer	O	Picture width, pixels	
++Height	integer	O	Picture height, pixels	
+CardNo	String	O	IC card number (for building products)	
+CellPhone	String	O	phone number (input number before comparison) (for building products)	
+BuildingNumber	String	O	building number (for building products)	
+BuildingUnitNumber	String	O	building entrance number (for building products)	
+BuildingRoomNumber	String	O	room number (for building products)	
+PersonnelRelationship	String	O	relationship (for building products)	
+Method	integer	O	method (refer to OpenDoorMethod)	
+EventGroupID	integer	O	event group id, used to associate different events are with same action. (Used with AccessControlevents)	
+CallNumber	String	O	phone number to be called.(called after compare succeed) (for building products)	
+EventType	integer	O	event type: 0: compare result 1: Face acquisition 2: visitor registration 3: Face permission distribution 4: inquirement of ID card face group	
+UserID	String	O	user ID	
+FaceIndex	integer	O	face index, range [0,4]	0
+Mask	integer	O	is with mask, it can be { 0: unknow 1: without mask 2: with mask } default 0.	0

Event Response Example (multipart , key=value format in body , binary data in body) (response to 4.4.3 snap subscription)

--<boundary>

Content-Type: text/plain

Content-Length: <length>

```

Events[0].EventBaseInfo.Code= CitizenPictureCompare
Events[0].EventBaseInfo.Action=Pulse
Events[0].EventBaseInfo.Index=0
Events[0].UTC=1999999999
Events[0].CompareResult=true
Events[0].Similarity=80
Events[0].Threshold=75
...
--<boundary>
Content-Type: image/jpeg
Content-Length: <image size>

<Jpeg image data>
--<boundary>

```

12.1.11 [Event] Door Status Event

Event Code	DoorStatus		
Event action	Pulse		
Event index	0		
Event Data			
Name	Type	R/O	Description
+UTC	uint32	O	Standard UTC time (without DST deviation of time zone). Required for access control products, and optional for intercom products.
+Status	enumchar[32]	O	Door status Enumchar[32]{ "Open" "Close" "CloseAlways" "OpenAlways" "Normal" }
			CloseAlways

Event Response Example (multipart , key=value format in body , binary data in body) (response to 4.4.3 snap subscription)

```

--<boundary>
Content-Type: text/plain
Content-Length: <length>

Events[0].EventBaseInfo.Code= DoorStatus
Events[0].EventBaseInfo.Action=Pulse
Events[0].EventBaseInfo.Index=0
Events[0].UTC=1999999999
Events[0].Status=CloseAlways
--<boundary>

```

12.1.12 [Config] Access Control General Setting

Config Data Params				
Name	Type	R/O	Description	Example
AccessControlGeneral	object	O	AccessControlGeneral config object	
+AccessProperty	string	O	Access property, can be : "unidirect", "bidirect" (This param is supported by access control device.)	bidirect
+ABLock	object	O	AB Lock setting (This param is supported by access control device.)	
++Enable	bool	O	Enable AB Lock or not.	
++Doors	int[]	O	AB Lock groups, each group has several doors, one door can be opened only when all other doors in the AB lock group are in closed state.	
+CustomPasswordEnable	bool	O	Whether to enable custom password. (This param is supported by video talk device.)	
+CommonPassword	string	O	The common password. (This param is supported by video talk device.)	
+ButtonExitEnable	bool	O	Whether to enable the open door button. (This param is supported by video talk device.)	
+CheckSensorBeforeLock	bool	O	Whether to check the sensor before lock the door. (This param is supported by video talk device.)	
+CheckSensorTime	int	O	The check sensor time, in seconds. (This param is supported by video talk device.)	
+DuressPassword	string	O	The duress password. (This param is supported by video talk device.)	
+DuressEnable	bool	O	Whether to enable duress password. (This param is supported by video talk device.)	
+UnlockRecordType	Array<string>	O	Unlock Record Type,can be {"Password", "Button", "FingerPrint", "QRCode", "Card", "Remote", "BlueTooth", "Face"}	
+SensorType	integer	O	sensor type, 0 for always open; 1 for always close.	
+CallLiftType	string	O	call lift protocol type	
+CallLiftEnable	bool	O	whether to enable call lift	
+AccessVoice	object	O	Door opening prompt tone	
++CurrentVoiceID	integer	O	the current prompt tone ID, default value 0	
++VoiceList	array<object>	O	voice list, max length is 16	

	>		voice ID can be: 0: Validate succeed 1: door open succeed 2: unlock succeed 3: punch time clocks succeed 4: welcome 5: Welcome to come again 6: thank you 7: custom voice }	
+++VoicelD	integer	O	voice id	
+++VoiceName	string	O	voice info	
+++FileName	string	O	file full path with name	

Please refer to "4.2.1 Get and Set Configure" for configuration getting and setting. Specific examples are as follows :

Get Config Request Example

```
http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=AccessControlGeneral
```

Get Config Response Example

```
table.AccessControlGeneral.AccessProperty=bidirect
table.AccessControlGeneral.ABLock.Enable=true
table.AccessControlGeneral.ABLock.Doors[0][0]=1
table.AccessControlGeneral.ABLock.Doors[0][1]=2
table.AccessControlGeneral.ABLock.Doors[0][2]=3
table.AccessControlGeneral.ABLock.Doors[1][0]=4
table.AccessControlGeneral.ABLock.Doors[1][1]=5
table.AccessControlGeneral.ABLock.Doors[1][2]=6
table.AccessControlGeneral.CustomPasswordEnable=true
table.AccessControlGeneral.CommonPassword=123456
table.AccessControlGeneral.ButtonExitEnable=true
table.AccessControlGeneral.CheckSensorBeforeLock=true
table.AccessControlGeneral.CheckSensorTime=30
table.AccessControlGeneral.DuressPassword=654321
table.AccessControlGeneral.DuressEnable=true
```

Set Config Request Example

```
http://10.0.0.8/cgi-bin/configManager.cgi?action=setConfig&AccessControlGeneral.AccessProperty=bidi
rect&AccessControlGeneral.ABLock.Enable=true&AccessControlGeneral.ABLock.Doors[0][0]=1&Acces
sControlGeneral.ABLock.Doors[0][1]=2&AccessControlGeneral.ABLock.Doors[0][2]=3&AccessControlG
eneral.CustomPasswordEnable=true&AccessControlGeneral.CheckSensorBeforeLock=true
```

Set Config Response Example

OK

12.1.13 [Config] Access Control Setting

Config Data Params

Name	Type	R/O	Description	Example
AccessControl	object[]	R	The AccessControl config array for every access control channel	
+Enable	bool	O	Whether to enable config for this channel.	
+State	string	O	Door state, can be : "Normal", "CloseAlways", "OpenAlways", "NoPersonNC", "NoPersonNO"	
+Method	int	O	<p>Open door method, can be following value, default is 2 :</p> <p>0 : only by password 1 : only by access card 2 : by password or access card 3 : by access card first then password 4 : by password first then access card 5 : different method in differenct time range 6 : only by fingerprint 7 : by password or access card or fingerprint 8 : by password and access card and fingerprint together 9 : by password and fingerprint together 10 : by access card and fingerprint together 11 : by multiple access user 12 : by face (match with picture in ID card) 13 : by face and ID card 14 : by face (match with picture in ID card) or access card or fingerprint 15 : by face and ID card, or access card or fingerprint 16 : by UserID and password 17 : only by face 18 : by face and password together 19 : by fingerprint and password together 20 : by fingerprint and face together 21 : by access card and face together 22 : by face or password 23 : by fingerprint or password 24 : by fingerprint or face 25 : by access card or face 26 : by access card or fingerprint 27 : by fingerprint and face and password together 28 : by access card and face and password together</p>	

			<p>29 : by access card and fingerprint and password together</p> <p>30 : by access card and fingerprint and face together</p> <p>31 : by fingerpint or face or password</p> <p>32 : by access card or face or password</p> <p>33 : by access card or fingerprint or face</p> <p>34 by access card and fingerprint and face and password together</p> <p>35 : by access card or fingerprint or face or password</p> <p>36 : by face and ID card, or access card or face</p> <p>37 : by face (match with picture in ID card) or access card or face</p> <p>38 : by access card and password, or fingerprint and password</p> <p>39 : by ID card (match with picture in ID card) or face</p> <p>40 : by ID card (match with fingerprint in ID card)</p> <p>41 : by ID card (match with fingerprint and picture in ID card)</p> <p>42 : by ID card or access card or fingerprint or face or password</p> <p>43 : by multy user method</p> <p>44 : by ID card or health code</p>	
+OpenAlwaysTi me	int	O	The time range that working, value is index in AccessTimeSechdule config.	
+CloseAlwaysTi me	int	O	The time range that always closed, value is index in AccessTimeSechdule config.	
+HolidayTime	int	O	The time range that working in holidays, value is index in AccessTimeSechdule config.	
+UnlockHoldInte rval	int	O	The Unlock holding interval, unit is milliseconds, value should between 250 to 20000. (This param is also supported by video talk device.)	
+UnlockReloadI nterval	int	O	The Unlock reload interval, unit is milliseconds. (This param is also supported by video talk device.)	
+AccessProtocol	string	O	The access control process, can be : "Local", "Dahua", "Remote", "Private" (This param is also supported by video talk device.)	
+BreakInAlarmE nable	bool	O	Whether to enable the breakin alarm.	

+RepeatEnterAlarm	bool	O	Whether to enable the repeat enter alarm.	
+DoorNotClosedAlarmEnable	bool	O	Whether to enable the alarm when door not closed.	
+DuessAlarmEnable	bool	O	Whether to enable the duress alarm.	
+FirstEnter	object	O	The first enter setting.	
++Enable	bool	O	Whenther to enable the first enter function,	
++Status	string	O	The status, can be : "KeepOpen" "Normal"	
++Time	int	O	The time range that enable first enter function, value is index in AccessTimeSechdule config.	
+CardNoConvert	int	O	The card number convert setting : 0 – not convert, 1 – convert using NOT operation, 2 : use HIDpro convert	
+MaliciousAccessControlEnable	bool	O	Whether to enable malicious access alarm.	
+AutoRemoteCheck	object	O	auto open door remotely	
++Enable	bool		whether to enable the function	
++Time	uint		working period , value is index in AccessTimeSechdule config.	
+Name	string	O	name	
+RepeatEnterTime	integer	O	Repeat entry time, unit second, 0 means do not start	
+CloseTimeout	integer	O	Closing overtime	
+SensorEnable	bool	O	sensor enable	
+TimeSchedule	Array<Array<object>>		Open the door in different periods it works when Method is "different method in differenct time range".	
++TimeSection	string	O	time section, format as "hh:mm:ss-hh:mm:ss"	
++Method	integer	O	Open door method, can be following value, default is 2 : 0 : only by password 1 : only by access card 2 : by password or access card 3 : by access card first then password 4 : by password first then access card 5 : different method in differenct time range 6 : only by fingerprint 7 : by password or access card or fingerprint 8 : by password and access card and	

			fingerprint together 9 : by password and fingerprint together 10 : by access card and fingerprint together 11 : by multiple access user 17 : only by face 35 : by access card or fingerprint or face or password	
+CustomPasswordEnable	bool	O	whether to enable admin password	
+RemoteCheck	bool	O	Is platform validation required	
+LockMode	uint8	O	electric relay mode 2: Ring mode	

Please refer to "4.2.1 Get and Set Configure" for configuration getting and setting. Specific examples are as follows :

Get Config Request Example

```
http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=AccessControl
```

Get Config Response Example

```
table.AccessControl[0].Enable=true
table.AccessControl[0].SN=1C03E08YAZ00020
table.AccessControl[0].Name=Door1
table.AccessControl[0].State=Normal
table.AccessControl[0].LocalControlEnable=true
table.AccessControl[0].RemoteControlEnable=true
table.AccessControl[0].Mode= HandProtected
...
```

Set Config Request Example

```
http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&AccessControl[0].Enable=true&AccessControl[0].SN=1C03E08YAZ00020&AccessControl[0].State=Normal&AccessControl[0].LocalControlEnable=true
```

Set Config Response Example

```
OK
```

12.1.14 [Config] Wiegand Setting

Config Data Params				
Name	Type	R/O	Description	Example
Wiegand	object[]	R	Wiegand config	
+Mode	int	R	Wiegand mode, 0: wiegand input, 1 : wiegand output	1
+PulseWidth	int	R	The pulse width, unit is microseconds	200
+PulseStep	int	R	The pulse step, unit is microseconds	1000
+TransferMode	int	R	The transfer mode, 0 : wiegand 34bit transfer, 4 byte card number, 2 bit	1

			checksum, 1 : wiegand 66bit transfer, 8 byte card number, 2 bit checksum, 2 : wiegand 26bit transfer, 3 byte card number, 2 bit checksum,	
+OutType	int	R	The output type, 0 : output ID, 1 : output card number	1
+InputType	integer	O	the input type, the value is 32bits. It won't accept any input when value is 0. Bit0: card number input Bit1: password input bit2~Bit31:reserved	3
+Doors	integer	O	door index, starts from 0, and 0 means door 1	0

Please refer to "4.2.1 Get and Set Configure" for configuration getting and setting. Specific examples are as follows :

Get Config Request Example

```
http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=Wiegand
```

Get Config Response 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
...
```

Set Config Request Example

```
http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&Wiegand[0].Mode=1&Wiegand[0].PulseWidth=200& Wiegand[0].PulseStep=1000
```

Set Config Response Example

```
OK
```

12.1.15 [Config] Access Time Schedule Setting

The following Apis are not supported by video talk device.

Config Data Params				
Name	Type	R/O	Description	Example
AccessTimeSchedule	object[]	R	AccessTimeSchedule config object array	
+Name	string	O	The schedule name, max string length is 63.	
+Enable	bool	R	Whether to enable this time schedule	

+TimeSechdule	array< array< string > >	O	The time schedule array, first array has max 8 item, refer to 7 day in a week (the first one is Sunday)and last one is holiday, each item is an array has max 6 time section, each time section is a string, format is "enable, hour:minite:second – hour:minite:second", for example : "1 00:00:00-12:00:00"	
---------------	--------------------------	---	---	--

Please refer to "4.2.1 Get and Set Configure" for configuration getting and setting. Specific examples are as follows :

Get Config Request Example

```
http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=AccessTimeSchedule
```

Get Config Response Example

```
table.AccessTimeSchedule[0].Name=TS1
table.AccessTimeSchedule[0].Enable=true
table.AccessTimeSchedule[0].TimeSchedule[0][0]=1 00:00:00-12:00:00
table.AccessTimeSchedule[0].TimeSchedule[0][1]=1 15:00:00-20:00:00
table.AccessTimeSchedule[0].TimeSchedule[1][0]=1 00:00:00-12:00:00
table.AccessTimeSchedule[0].TimeSchedule[1][1]=1 15:00:00-20:00:00
...
...
```

Set Config Request Example

```
http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&AccessTimeSchedule[0].Name=TS1&
AccessTimeSchedule[0].Enable=true
```

Set Config Response Example

```
OK
```

12.1.16 [Config] Special Day Group Setting

The following Apis are not supported by video talk device.

Config Data Params				
Name	Type	R/O	Description	Example
SpecialDayGrou p	array<object >	R	SpecialDayGroup config object array	
+Name	string	O	The special day group name	SpecialDayGroup1
+Enable	bool	R	Whether to enable this special day group	true
+Days	array<object >	O	The special days in group	
++SpecialDayNa me	string	O	The special day name.	NationalDay
++StartTime	string	O	The special day start datetime	2017-10-01 00:00:00
++EndTime	string	O	The special day end datetime	2017-10-07 23:59:59

Please refer to "4.2.1 Get and Set Configure" for configuration getting and setting. Specific examples are as follows :

Get Config Request Example

```
http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=SpecialDayGroup
```

Get Config Response Example

```
table.SpecialDayGroup[0].Name=SpecialDayGroup1  
table.SpecialDayGroup[0].Enable=true  
table.SpecialDayGroup[0].Days[0].SpecialDayName=NationalDay  
table.SpecialDayGroup[0].Days[0].StartTime=2017-10-01 00:00:00  
table.SpecialDayGroup[0].Days[0].StartTime=2017-10-07 23:59:59  
...
```

Set Config Request Example

```
http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&SpecialDayGroup[0].Name=SpecialDayGroup1&SpecialDayGroup[0].Enable=true
```

Set Config Response Example

```
OK
```

12.1.17 [Config] Special Days Schedule Setting

The following Apis are not supported by video talk device.

Config Data Params

Name	Type	R/O	Description	Example
SpecialDaysSchedule	array<object>	R	SpecialDaysSchedule config object array	
+Name	string	O	The special day schedule name	
+Enable	bool	R	Whether to enable this special day schedule	
+GroupNo	int	R	The special day group number, value is the index of SpecialDayGroup setting array.	
+TimeSection	array<string>	R	The time section array, the format of the item is "enable, hour:minite:second – hour:minite:second", for example : "1 00:00:00-12:00:00".	
+Doors	array<int>	R	The doors array.	

Please refer to "4.2.1 Get and Set Configure" for configuration getting and setting. Specific examples are as follows :

Get Config Request Example

```
http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=SpecialDaysSchedule
```

Get Config Response Example

```
table.SpecialDaysSchedule[0].Name=SpecialDayGroup1  
table.SpecialDaysSchedule[0].Enable=true  
table.SpecialDaysSchedule[0].GroupNo=1
```

```

table.SpecialDaysSchedule[0].TimeSection[0]=1 00:00:00-12:00:00
table.SpecialDaysSchedule[0].TimeSection[1]=1 15:00:00-20:00:00
table.SpecialDaysSchedule[0].Doors[0]=2
table.SpecialDaysSchedule[0].Doors[1]=3
...

```

Set Config Request Example

```
http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&SpecialDaysSchedule[0].Name=Spec  
ialDayGroup1
```

Set Config Response Example

```
OK
```

12.1.18 [Config] MeasureTemperature Setting

The following Apis are not supported by video talk device.

Config Data Params				
Name	Type	R/O	Description	Example
MeasureTemperature	object	R	MeasureTemperature config object	
++Enable	bool	R	whether to enable measure temperature function	
+MaskOpt	uint	R	mask mode 0: do not detect mask 1: mask prompt 2: mask intercept	
+OnlyTemperatu reMode	bool	O	whether to enable the single function, temperature monitoring	
+TemperatureDis play	bool	R	whether to show the temperature	
+TemperatureUn it	uint	O	temperature unit:{ 0: centigrade 1: Fahrenheit }	
+Type	integer	R	Temperature monitoring method, it can be: 0: Single chip microcomputer infrared temperature monitoring 1: the thermal imaging principle of temperature monitoring 2: Temperature monitoring module of Guide 3: single point temperature monitoring at wrist	
+GuideModuleP aram	object	O	param of temperature monitoring module of Guide	

++Threshold	float	O	the temperature threshold, unit is °C	
++CalibrationModel	uint32	O	Calibration Model, it can be: 0: indoor mode 1: wall mounting mode 2: Gate mode 3: exclusive floor mode	
++Correct	Float	O	temperature correct value, unit is °C	
++DebugModelEnable	Bool	O	whether to enable debug mode of temperature monitoring, show temperature on the top of face	
++HeatDisplayEnable	bool	O	whether to show heat	
++MaxDistance	uint32	O	the max distance for temperature monitoring, unit: cm	
++ProjectDebugModel	bool	O	whether to enable project debug mode	
++RectEnable	Bool	O	whether to show the rectangle of temperature monitoring area on the video.	
++TempRandReplaceThreshold	Float	O	the threshold under which the temperature value will be replaced with valid one. When threshold value is 0, the function won't work.	
++ValidTemperatureLowerLimit	float	O	the lower limit of valid temperature value	
+InfraredTemperatureParam	object	O	param of infrared temperature monitoring	
++Correct	float	O	temperature correct value, unit is °C	
++DebugModelEnable	bool	O	whether to enable debug mode of temperature monitoring, show temperature on the top of face	
++MaxDistance	Uint32	O	the max distance for temperature monitoring, unit: cm	
++OverTemperatureMaxDistance	Uint32	O	Maximum distance for reporting high temperature events, In this distance, the high temperature was measured and reported directly. Beyond this distance, the high temperature was measured, it will indicate closing to the point, and then measured again (CM)	
++RectEnable	bool	O	whether to show the rectangle of temperature monitoring area on the video.	
++RetentionTime	Uint32	O	temperature retention Time, unit: ms	
++SensorType	string	O	sensor type in temperature monitoring module	

			"90641", "90640"	
++Threshold	Uint32	O	temperature threshold, unit : °C	
++ValidTemperatureLowerLimit	Float	O	the lower limit of valid temperature value	
+ThermalImagingParam	object	O	param of thermalImaging	
++RetentionTime	Uint32	O	temperature retention Time, unit: ms	
++Threshold	Uint32	O	temperature threshold, unit : °C	
+WristTemperatureParam	Object	O	param of wrist temperature monitoring	
++Correct	Float	O	temperature correct value, unit is °C	
++InvalidTemperatureDistance	Uint32	O	invalid temperature monitoring distance	
++TemperatureTimeout	Uint32	O	timeout of temperature monitoring, unit: s	
++Threshold	Float	O	temperature threshold, unit : °C	
++ValidTemperatureDistance	Uint32	O	valid temperature monitoring distance,	
++ValidTemperatureLowerLimit	float	O	the lower limit of valid temperature value	

Please refer to "4.2.1 Get and Set Configure" for configuration getting and setting. Specific examples are as follows :

Get Config Request Example

```
http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=MeasureTemperature
```

Get Config Response Example

```
table.MeasureTemperature.Enable=true
table.MeasureTemperature.GuideModuleParam.CalibrationModel=1
table.MeasureTemperature.GuideModuleParam.Correct=0
table.MeasureTemperature.GuideModuleParam.DebugModelEnable=false
table.MeasureTemperature.GuideModuleParam.HeatDisplayEnable=false
table.MeasureTemperature.GuideModuleParam.MaxDistance=0
table.MeasureTemperature.GuideModuleParam.ProjectDebugModel=false
table.MeasureTemperature.GuideModuleParam.RectEnable=true
table.MeasureTemperature.GuideModuleParam.TempRandReplaceThreshold=0
table.MeasureTemperature.GuideModuleParam.Threshold=37.300000
table.MeasureTemperature.GuideModuleParam.ValidTemperatureLowerLimit=30
table.MeasureTemperature.InfraredTemperatureParam.Correct=0
table.MeasureTemperature.InfraredTemperatureParam.DebugModelEnable=false
table.MeasureTemperature.InfraredTemperatureParam.MaxDistance=100
table.MeasureTemperature.InfraredTemperatureParam.OverTemperatureMaxDistance=100
table.MeasureTemperature.InfraredTemperatureParam.RectEnable=true
table.MeasureTemperature.InfraredTemperatureParam.RetentionTime=500
table.MeasureTemperature.InfraredTemperatureParam.SensorType=
table.MeasureTemperature.InfraredTemperatureParam.Threshold=37.300000
```

```

table.MeasureTemperature.InfraredTemperatureParam.ValidTemperatureLowerLimit=30
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=3
table.MeasureTemperature.WristTemperatureParam.Correct=0
table.MeasureTemperature.WristTemperatureParam.InvalidTemperatureDistance=24
table.MeasureTemperature.WristTemperatureParam.TemperatureTimeout=10
table.MeasureTemperature.WristTemperatureParam.Threshold=37.300000
table.MeasureTemperature.WristTemperatureParam.ValidTemperatureDistance=5
table.MeasureTemperature.WristTemperatureParam.ValidTemperatureLowerLimit=35
...

```

Set Config Request Example

<http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&MeasureTemperature.Enable=true&MeasureTemperature.GuideModuleParam.CalibrationModel=1>

Set Config Response Example

OK

12.1.19 [Config] CitizenPictureCompare Setting

Config Data Params				
Name	Type	R/O	Description	Example
CitizenPictureCompare	object	R	CitizenPictureCompare config object	
+Threshold	uint8	R	threshold of citizen picture compare, range [1, 100]	90
+UnlockEnable	bool	R	whether enable unlock if false, it won't unlock, when citizen picture compare succeed.	false
+FuncEnable	bool	O	whether to enable citizen picture comparison	false
+CitizenIDCheck	bool	O	is citizen ID check supported	false
+SysMode	string	O	device work mode, it can be: { "FaceCollect" , "VisitorVerify", "CitizenCompare", "CitizenIDCheck", "Other" }	CitizenCompare

Please refer to "4.2.1 Get and Set Configure" for configuration getting and setting. Specific examples are as follows :

Get Config Request Example

<http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=CitizenPictureCompare>

Get Config Response Example

```

table.CitizenPictureCompare.SysMode=CitizenCompare
table.CitizenPictureCompare.Threshold=50
table.CitizenPictureCompare.UnlockEnable=false
table.CitizenPictureCompare.FuncEnable=false
table.CitizenPictureCompare.CitizenIDCheck=false
...

```

Set Config Request Example

`http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&CitizenPictureCompare.SysMode=CitizenCompare&CitizenPictureCompare.Threshold=50`

Set Config Response Example

OK

12.2 Access Control Manager

12.2.1 Get Access Control Capability

Request URL	<code>http://<server>/cgi-bin/accessControlManager.cgi?action=getCaps</code>		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
Request Example			
<code>http://192.168.1.108/cgi-bin/accessControlManager.cgi?action=getCaps</code>			

Response Params (key=value format in body)			
Name	Type	R/O	Description
caps	object	R	The access control capabilities.
+AccessControlChannels	int	R	The access control channel number.
+AccessControlAlarmRecord	bool	O	Support log access control alarm record or not. (This param is not supported by video talk device)
+CustomPasswordEncryption	int	O	The custom password crypt type, 0 : plain text, 1 : MD5
+SupportFingerPrint	int	O	Support fingerprint type, 0 : unknown, 1 : not support, 2 : support
+OnlySingleDoorAuth	int	O	Support single door auth type, 0 : not support, 1 : support, (This param is not used by video talk device)
+AsynAuth	int	O	Support async auth type, 0 : not support, 1 : support, (This param is not supported by video talk device)

+SpecialDaysSchedule	object	O	Special days schedule capabilities, (This param is not supported by video talk device)	
++Support	bool	O	Support special days schedule or not. (This param is not supported by video talk device)	true
++MaxSpecialDaysSchedules	int	O	Max special days schedule number. (This param is not supported by video talk device)	6
++MaxTimePeriodsPerDay	int	O	Max time periods per day. (This param is not supported by video talk device)	6
++MaxSpecialDayGroups	int	O	Max special day groups. (This param is not supported by video talk device)	6
++MaxDaysInSpecialDayGroup	int	O	Max special days in special day group. (This param is not supported by video talk device)	16
+SupportCallLift	enumint	O	Does it support elevator calling. enumint{ 0: Not supported 1: Support } Default 0	0
+SupportUnlocking	bool	O	Does the device support unlocking function	true
+HasCardAuth	bool	O	Does the device support card swiping permission	false

Response Example

```

caps.AccessControlChannels=3
caps.AccessControlAlarmRecord=true
caps.CustomPasswordEncryption=1
caps.SupportFingerPrint=1
caps.OnlySingleDoorAuth=1
caps.AsynAuth=1
caps.SpecialDaysSchedule.Support=true
caps.SpecialDaysSchedule.MaxSpecialDaysSchedules=5
caps.SpecialDaysSchedule.MaxTimePeriodsPerDay=5
caps.SpecialDaysSchedule.MaxSpecialDayGroups=5
caps.SpecialDaysSchedule.MaxDaysInSpecialDayGroup=5
caps.SupportCallLift=0
caps.SupportUnlocking=true
caps.HasCardAuth=true

```

12.2.2 Add SubController

Request URL	http://<server>/cgi-bin/accessControlManager.cgi?action=addDevice
-------------	---

Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
devAddr	int	R	SubController ID, range is [1...255]
name	char[]	R	SubController Name
Request Example			
http://192.168.1.108/cgi-bin/accessControlManager.cgi?action=addDevice&devAddr=1&name=Sub1			

Response Params (key=value format in body)	
Name	
Type	
R/O	
Description	
FaiedCode	
int	
R	
0-Success; 1-Already exists; 2-Reach quantity limit; 3-Other reason	
Example	
0	
Response Example	
FaiedCode=0	

12.2.3 Modify SubController

Request URL	http://<server>/cgi-bin/accessControlManager.cgi?action=modifyDevice		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
devAddr	int	R	SubController ID, range is [1...255]
name	char[]	R	SubController Name
Request Example			
http://192.168.1.108/cgi-bin/accessControlManager.cgi?action=modifyDevice&devAddr=1&name=Sub1			

Response Params (OK in body)
Response Example
OK

12.2.4 Remove SubController

Request URL	http://<server>/cgi-bin/accessControlManager.cgi?action=removeDevice		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
devAddrs	int[]	R	SubController ID, range is [1...255]
Request Example			
http://192.168.1.108/cgi-bin/accessControlManager.cgi?action=removeDevice&devAddrs[0]=4&devAddrs[1]=5			

Response Params (key=value format in body)
Name
Type
R/O
Description
devInfo
object[]
R
result
+DevAddr
int
R
the removed SubController ID
+FaiedCode
int
R
0-Success; 1-Fail; 2-Other
Example
4
0

Response Example

```
devInfo[0].DevAddr=4  
devInfo[0].FaiedCode=0  
devInfo[1].DevAddr=5  
devInfo[1].FaiedCode=1
```

12.2.5 Get SubController Info

Request URL	http://<server>/cgi-bin/accessControlManager.cgi?action=getSubControllerInfo		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
devAddrs	int[]	R	SubController ID, range is [1...255]
Request Example			
http://192.168.1.108/cgi-bin/accessControlManager.cgi?action=getSubControllerInfo&devAddrs[0]=4&devAddrs[1]=5			

Response Params (key=value format in body)

Name	Type	R/O	Description	Example
devInfo	object[]	O		
+DevAddr	int	O	SubController ID, range is [1...255]	1
+Name	char[64]	O	name	"xxxxxx"
+AccessProperty	enumchar[32]	O	range is [unidirect, bidirect]	"bidirect"
+DeviceType	char[64]	O	type	"ACS-1202B-S"
+VerSion	char[64]	O	version	"V1.00428"
+DoorToReadID	object[]	O	the door	
++Door	uint	O	door index	0
++ReaderID	char[][32]	O	ReaderID used by the door	["1","2"]

Response Example

```
devInfo[0].DevAddr=4  
devInfo[0].Name=Sub4  
devInfo[0].AccessProperty=unidirect  
devInfo[0].DeviceType=ACS-1202B-S  
devInfo[0].VerSion=V1.00428  
devInfo[0].DoorToReadID[0].Door=0  
devInfo[0].DoorToReadID[0].ReaderID[0]=1  
devInfo[0].DoorToReadID[0].ReaderID[1]=2  
devInfo[0].DoorToReadID[1].Door=1  
devInfo[0].DoorToReadID[1].ReaderID[0]=3  
devInfo[0].DoorToReadID[1].ReaderID[1]=4  
devInfo[1].DevAddr=5  
devInfo[1].Name=Sub5  
devInfo[1].AccessProperty=bidirect  
devInfo[1].DeviceType=ACS-1202B-S
```

```

devInfo[1].VerSion=V1.00428
devInfo[1].DoorToReadID[0].Door=0
devInfo[1].DoorToReadID[0].ReaderID[0]=11
devInfo[1].DoorToReadID[0].ReaderID[1]=12

```

12.2.6 Get SubController States

Request URL	http://<server>/cgi-bin/accessControlManager.cgi?action=getSubControllerStates		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
devAddrs	int[]	R	SubController ID, range is [1...255]
Request Example			
http://192.168.1.108/cgi-bin/accessControlManager.cgi?action=getSubControllerStates&devAddrs[0]=4&devAddrs[1]=5			

Response Params (key=value format in body)			
Name	Type	R/O	Description
devStates	object[]	R	status
+DevAddr	int	R	SubController ID, range is [1...255]
+State	int	R	0-Offline; 1-Online; 2-Other
Response Example			
devStates[0].DevAddr=4 devStates[0].State=0 devStates[1].DevAddr=5 devStates[1].State=1			

12.2.7 Set RepeatEnter Route

Request URL	http://<server>/cgi-bin/accessControlManager.cgi?action=setRepeatEnterRoute		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
index	uint32	O	route index
routeInfo	object	O	route info
+PointInfo	object[]	O	is an array, stands for points in route
++ReaderID	char[] [3 2]	O	reader id list
+TimeSections	uint	O	subscript of AccessTimeSchedule
+Flag	int	O	0-Disable this route; 1-Enable this route
+Name	char[64]	O	route name
Request Example			
http://192.168.1.108/cgi-bin/accessControlManager.cgi?action=setRepeatEnterRoute&index=0&routeInfo.PointInfo[0].ReaderID[0]=11&routeInfo.PointInfo[0].ReaderID[1]=22&routeInfo.PointInfo[1].ReaderID[0]=33&routeInfo.PointInfo[1].ReaderID[1]=44&routeInfo.TimeSections=1&routeInfo.Flag=1&routeInfo.Name=123			

Response Params (OK in body)

Response Example

OK

12.2.8 Get RepeatEnter Route

Request URL	http://<server>/cgi-bin/accessControlManager.cgi?action=getRepeatEnterRoute		
--------------------	---	--	--

Method	GET		
---------------	-----	--	--

Request Params (key=value format in URL)

Name	Type	R/O	Description	Example
index	uint32	R	route index	0

Request Example

http://192.168.1.108/cgi-bin/accessControlManager.cgi?action=addDevice&devAddr=1&name=Sub1
--

Response Params (key=value format in body)

Name	Type	R/O	Description	Example
routelInfo	object	O	route info	
+PointInfo	object[]	O	is an array, stands for points in route	
++ReaderID	char[] [32]	O	reader id list	["1","2"]
+TimeSections	uint	O	subscript of AccessTimeSchedule	1
+Flag	enumint	O	0-Disable this route; 1-Enable this route	0
+Name	char[64]	O	route name	123

Response Example

routelInfo.PointInfo[0].ReaderID[0]=11
routelInfo.PointInfo[0].ReaderID[1]=22
routelInfo.PointInfo[1].ReaderID[0]=33
routelInfo.PointInfo[1].ReaderID[1]=44
routelInfo.TimeSections=1
routelInfo.Flag=1
routelInfo.Name=123

12.2.9 Set ABLock Route

Request URL	http://<server>/cgi-bin/accessControlManager.cgi?action=setABLockRoute		
--------------------	--	--	--

Method	GET		
---------------	-----	--	--

Request Params (key=value format in URL)

Name	Type	R/O	Description	Example
index	uint32	O	route index	0
routelInfo	object	O	route info	
+PointInfo	object[]	O	is an array, stands for points in route	
++Doors	int[]	O	door list	[0,1]
+TimeSections	int32	O	subscript of AccessTimeSchedule	1
+Flag	enumint	O	0-Disable this route; 1-Enable this route	1
+Name	char[64]	O	route name	"ABLockRoute"

Request Example

```
http://192.168.1.108/cgi-bin/accessControlManager.cgi?action=setABLockRoute&index=0&routelInfo.Po
intInfo[0].Doors[0]=11&routelInfo.PointInfo[0].Doors[1]=22&routelInfo.PointInfo[1].Doors[0]=33&routelInfo.
PointInfo[1].Doors[1]=44&routelInfo.TimeSections=1&routelInfo.Flag=1&routelInfo.Name=123
```

Response Params (OK in body)

Response Example

OK

12.2.10 Get ABLock Route

Request URL	http://<server>/cgi-bin/accessControlManager.cgi?action=getABLockRoute		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
index	uint32	R	route index
Request Example			
http://192.168.1.108/cgi-bin/accessControlManager.cgi?action=getABLockRoute&index=0			

Response Params (key=value format in body)

Name	Type	R/O	Description	Example
routelInfo	object	O	route info	
+PointInfo	object[]	O	is an array, stands for points in route	
++Doors	int[]	O	door list	[0,1]
+TimeSections	int32	O	subscript of AccessTimeSchedule	1
+Flag	enumint	O	0-Disable this route; 1-Enable this route	1
+Name	char[64]	O	route name	"ABLockRoute"

Response Example

```
routelInfo.PointInfo[0].Doors[0]=11
routelInfo.PointInfo[0].Doors [1]=22
routelInfo.PointInfo[1].Doors [0]=33
routelInfo.PointInfo[1].Doors [1]=44
routelInfo.TimeSections=1
routelInfo.Flag=1
routelInfo.Name=123
```

12.2.11 Get Log Status

Request URL	http://<server>/cgi-bin/accessControlManager.cgi?action=getLogStatus		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
devAddr	int	R	SubController ID, range is [1...255]
Request Example			
http://192.168.1.108/cgi-bin/accessControlManager.cgi?action=getLogStatus&devAddr=1			

Response Params (OK in body)

Response Example

OK

12.2.12 Sync Offline Log

Request URL	http://<server>/cgi-bin/accessControlManager.cgi?action=syncOfflineLog			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
devAddr	int	R	SubController ID, range is [1...255]	1
Request Example				
http://192.168.1.108/cgi-bin/accessControlManager.cgi?action=syncOfflineLog&devAddr=1				

Response Params (OK in body)
Response Example
OK

12.2.13 Sync SubController Time

Request URL	http://<server>/cgi-bin/accessControlManager.cgi?action=syncSubControllerTime			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
devAddrs	int	R	SubController ID, range is [1...255]	[4, 5]
Request Example				
http://192.168.1.108/cgi-bin/accessControlManager.cgi?action=syncSubControllerTime&devAddrs[0]=4&devAddrs[1]=5				

Response Params (key=value format in body)
Response Example
errorDevAddrs[0]=4

12.3 Access User Account (V1)

12.3.1 Add Access User Face

Request URL	http://<server>/cgi-bin/FaceInfoManager.cgi?action=add		
Method	POST		
Request Params (JSON format in body)			
Name	Type	R/O	Description
UserID	char[]	R	The user's id.
Info	object	R	The user's info.
+UserName	char[]	O	The user's name.
+RoomNo	char[][]	O	The user's room num list, max array size is 32, max string size is 12. (This param is not supported by access control.)
+FaceData	char[][]	O	The user's face feature, encode with base64, max array size is 20, and max string size is 2k. Note: There must be at least one between FaceData and PhotoData.
+PhotoData	char[][]	O	The user's face photo, encode with base64, max array size is 5, and max string size is 200k. Note: There must be at least one between FaceData and PhotoData.
Request Example			
POST http://<server>/cgi-bin/FaceInfoManager.cgi?action=add Content-Type: application/json Content-Length: <length>			
{ "UserID": "102", "Info": { "UserName": "ZhangSan", "RoomNo": ["301", "303", ...], "FaceData": ["xxxx", "xxxx", ...], "PhotoData": ["yyyy", "yyyy", ...] } }			

Response Params (JSON format in body)	
Name	Type
Response Example	
{	}

12.3.2 Modify Access User Face

Request URL	http://<server>/cgi-bin/FaceInfoManager.cgi?action=update		
Method	POST		
Request Params (JSON format in body)			
Name	Type	R/O	Description
UserID	char[]	R	The user's id. Example: 10000
Info	object	R	The user's info.
+UserName	char[]	O	The user's name. Example: ZhangSan
+RoomNo	char[][]	O	The user's room num list, max array size is 32, max string size is 12. (This param is not supported by access control.) Example: ["301", "303", ...]
+FaceData	char[][]	O	The user's face feature, encode with base64, max array size is 20, and max string size is 2k. Note: There must be at least one between FaceData and PhotoData. Example: ["xxxx", "xxxx", ...]
+PhotoData	char[][]	O	The user's face photo, encode with base64, max array size is 5, max string size is 200k. Note: There must be at least one between FaceData and PhotoData. Example: ["yyyy", "yyyy", ...]

Request Example

POST http://<server>/cgi-bin/FaceInfoManager.cgi?action=update

Content-Type: application/json

Content-Length: <length>

```
{
  "UserID": "102",
  "Info": {
    "UserName": "ZhangSan",
    "RoomNo": [ "301", "303", ... ],
    "FaceData": [ "xxxx", "xxxx", ... ],
    "PhotoData": [ "yyyy", "yyyy", ... ]
  }
}
```

Response Params (JSON format in body)

Name	Type	R/O	Description	Example
Response Example				
{}				

12.3.3 Delete Access User Face

Request URL	http://<server>/cgi-bin/FaceInfoManager.cgi?action=remove			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
UserID	char[]	R	The user's id.	10000
Request Example				
http://192.168.1.10/cgi-bin/FaceInfoManager.cgi?action=remove&UserID=102				

Response Params (OK in body)

Response Example

OK

12.3.4 Find Access User Face

- Start to find access user face

Request URL	http://<server>/cgi-bin/FaceInfoManager.cgi?action=startFind			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
Condition	object	O	Search condition.	
+UserID	char[]	O	The user's id.	1001
Request Example				
http://192.168.1.108/cgi-bin/FaceInfoManager.cgi?action=startFind&Condition.UserID=1001				

Response Params (JSON format in body)

Name	Type	R/O	Description	Example
Token	int	O	The token of this search, use this token to get result and stop search.	1234
Total	int	O	Result num, return 0 if not found.	20

Response Example

```
{  
    "Token": 1234,  
    "Total": 20  
}
```

- Get the find result

Request URL	http://<server>/cgi-bin/FaceInfoManager.cgi?action=doFind			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
Token	int	R	The token of this search, use this token to get result and stop search.	1234

Offset	int	R	Offset in the result record set, range is [0, Total – 1].	1001
Count	int	R	Count of result to get.	100
Request Example				
http://192.168.1.108/cgi-bin/FaceInfoManager.cgi?action=doFind&Token=1234&Offset=0&Count=20				

Response Params (JSON format in body)				
Name	Type	R/O	Description	Example
Info	object[]	R	The face information of user.	
+UserID	char[]	R	The user's id.	"1001"
+MD5	char[][]	O	The user's face photo's MD5 hash string, max array size is 5, max string size is 33.	["A12847B24",...]
+EigenMD5	char[][]	O	An array of maximum 5 items for photo's MD5 digest characters. The array element length is 32 bytes.	["A12847B24",...]

Response Example
{ "Info": [{ "UserID": "102", "MD5": ["xxxx", "xxxx", ...] }, ...{}] }

- Stop the find session

Request URL	http://<server>/cgi-bin/FaceInfoManager.cgi?action=stopFind			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
Token	int	R	Token for this search, use this token to get result and stop search.	1234

Request Example
http://192.168.1.108/cgi-bin/FaceInfoManager.cgi?action=stopFind&Token=1234

Response Params (OK in body)
Response Example
OK

12.3.5 Add Access User Card and Fingerprint
<ul style="list-style-type: none"> Add Access user card only
Request URL
http://<server>/cgi-bin/recordUpdater.cgi?action=insert&name=AccessControlCard

Method

GET

Request Params (key=value format in URL)

Name	Type	R/O	Description	Example
name	char[]	R	Access user card and fingerprint record name, should be " AccessControlCard ".	"AccessControlCard"
CardName	char[]	R	Access user card name, max string length is 32.	ZhangSan
CardNo	char[]	R	Access user card number.	12345
UserID	char[]	R	The user's id.	10000
CardStatus	int	R	Card status, 0 means normal, and abnormal status can be sum of following: <ul style="list-style-type: none"> • 1 << 0: report the loss of the card • 1 << 1: the card is withdrawn • 1 << 2: the card is frozen • 1 << 3: the card is arrearage • 1 << 4: the card exceeds the time limit • 1 << 5: the card is arrearage but still can open the door, and there will be voice prompts. 	0
CardType	int	O	The card type: 0 : Normal Card, 1: VIP Card, 2: Visitor Card, 3: Patrol Card, 4: Blocklist Card, 5: Stress Card, 0xff: Mother Card	0
Password	char[]	O	The Access card's password. (This param is not supported by video talk device)	123456
Doors	int[]	O	The index of the doors that this card can open. (This param is not supported by video talk device)	
TimeSections	int[]	O	The index of the time sections of each door that this card can open. (This param is not supported by video talk device)	
VTOPosition	char[]	O	VTO position number. (This param is not supported by access control device)	01018001
ValidDateStart	char[]	O	The start of valid date, format is "yyyyMMdd hhmmss".	20151022%2009381 1
ValidDateEnd	char[]	O	The end of valid date, format is "yyyyMMdd hhmmss".	20151222%2009381 1
IsValid	bool	O	Is the card still valid. (This param is not supported by video talk device)	

Request Example

```
http://192.168.1.108/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&VTOPosition=01018001&ValidDateStart=20151022%20093811&ValidDateEnd=20151222%20093811
```

Response Params (key=value format in body)

Name	Type	R/O	Description	Example
recono	int	R	The record id.	12345
Response Example				
recono=12345				

- Add access user card with fingerprint

Add access user card with fingerprint. If add only access user card, please use :

<http://<server>/cgi-bin/recordUpdater.cgi?action=insert&name=AccessControlCard>

Request URL	http://<server>/cgi-bin/recordUpdater.cgi?action=insertEx			
Method	POST			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
name	char[]	R	Access user card and fingerprint record name, should be "AccessControlCard".	"AccessControlCard"
CardName	char[]	O	Access user card name, max string length is 32.	ZhangSan
CardNo	char[]	O	Access user card number.	12345
UserID	char[]	R	The user's id.	10000
CardStatus	int	O	Card status, 0 means normal, and abnormal status can be sum of following: <ul style="list-style-type: none"> • 1 << 0: report the loss of the card • 1 << 1: the card is withdrawn • 1 << 2: the card is frozen • 1 << 3: the card is arrearage • 1 << 4: the card exceeds the time limit • 1 << 5: the card is arrearage but still can open the door, and there will be voice prompts. 	0
CardType	int	O	The card type: 0: Normal Card, 1: VIP Card, 2: Visitor Card, 3: Patrol Card, 4: Blocklist Card, 5: Stress Card, 0xff: Mother Card	0
Password	char[]	O	The Access card's password.	123456
Doors	int[]	O	The index of the doors that this card can open.	

TimeSections	int[]	O	The index of the time sections of each door that this card can open.	
VTOPosition	char[]	O	VTO position number. (This param is not supported by access control device)	01018001
ValidDateStart	char[]	O	The start of valid date, format is "yyyyMMdd hhmmss".	20151022%20093811
ValidDateEnd	char[]	O	The end of valid date, format is "yyyyMMdd hhmmss".	20151222%20093811
IsValid	bool	O	Is the card still valid.	true
FingerprintPacket	object	O	The fingerprint packet info.	
+Length	int	O	The length of each fingerprint packet.	500
+Count	int	O	The count of fingerprint packet.	3

Request Example

http://192.168.1.108/cgi-bin/recordUpdater.cgi?action=insertEx&name=AccessControlCard&CardName=ZhangSan&CardNo=12345&UserID=102&CardStatus=0&Password=123456&Doors[0]=1&Doors[1]=3&Doors[2]=5&VTOPosition=01018001&ValidDateStart=20151022%20093811&ValidDateEnd=20151222%20093811&FingerprintPacket.Length=500&FingerprintPacket.Count=3

Content-Type: application/octet-stream

Content-Length: <length>

<fingerprint packet binary data>

Response Params (key=value format in body)

Name	Type	R/O	Description	Example
recno	int	R	The record id.	12345

Response Example

recno=12345

12.3.6 Modify Access User Card and Fingerprint

- Update Access user card only

Note: You should provide at lease one optional param to update.

Request URL	http://<server>/cgi-bin/recordUpdater.cgi?action=update			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
name	char[]	R	Access user card and fingerprint record name, should be "AccessControlCard".	"AccessControlCard"
recno	int	R	The record id.	1234
CardName	char[]	O	Access user card name, max string length is 31.	ZhangSan
CardNo	char[]	O	Access user card number.	12345

			Primary key, modification is not allowed in principle	
UserID	char[]	O	The user's id. Primary key, modification is not allowed in principle	10000
CardStatus	int	O	Card status, 0 means normal, and abnormal status can be sum of following: <ul style="list-style-type: none"> • 1 << 0: report the loss of the card • 1 << 1: the card is withdrawn • 1 << 2: the card is frozen • 1 << 3: the card is arrearage • 1 << 4: the card exceeds the time limit • 1 << 5: the card is arrearage but still can open the door, and there will be voice prompts. 	0
CardType	int	O	The card type: 0: Normal Card, 1: VIP Card, 2: Visitor Card, 3: Patrol Card, 4: Blocklist Card, 5: Stress Card, 0xff: Mother Card	0
Password	char[]	O	The Access card's password.	123456
Doors	int[]	O	The index of the doors that this card can open.	
TimeSections	int[]	O	The index of the time sections of each door that this card can open.	
VTOPosition	char[]	O	VTO position number. (This param is not supported by access control device)	01018001
ValidDateStart	char[]	O	The start of valid date, format is "yyyyMMdd hhmmss".	20151022%20093811
ValidDateEnd	char[]	O	The end of valid date, format is "yyyyMMdd hhmmss".	20151222%20093811
IsValid	bool	O	Is the card still valid.	

Request Example

```
http://192.168.1.108/cgi-bin/recordUpdater.cgi?action=update&name=AccessControlCard&recno=12345&CardName=ZhangSan&CardNo=12345&UserID=102&CardStatus=0&CardType=0&Password=123456&Doors[0]=1&Doors[1]=3&Doors[2]=5&ValidDateStart=20151022%20093811&ValidDateEnd=20151222%20093811
```

Response Params (OK in body)

Response Example

OK

- Update access user card and fingerprint

Note: You should provide at least one optional param to update.

Request URL	http://<server>/cgi-bin/recordUpdater.cgi?action=updateEx			
Method	POST			
Request Params (key=value format in URL, binary in body)				
Name	Type	R/O	Description	Example
name	char[]	R	Access user card and fingerprint record name, should be "AccessControlCard" .	"AccessControlCard"
recno	int	R	The record id.	1234
CardName	char[]	O	Access user card name, max string length is 32.	ZhangSan
CardNo	char[]	O	Access user card number.	12345
UserID	char[]	O	The user's id.	10000
CardStatus	int	O	<p>Card status, 0 means normal, and abnormal status can be sum of following:</p> <ul style="list-style-type: none"> • 1 << 0: report the loss of the card • 1 << 1: the card is withdrawn • 1 << 2: the card is frozen • 1 << 3: the card is arrearage • 1 << 4: the card exceeds the time limit • 1 << 5: the card is arrearage but still can open the door, and there will be voice prompts. 	0
CardType	int	O	<p>The card type:</p> <p>0: Normal Card, 1: VIP Card, 2: Visitor Card, 3: Patrol Card, 4: Blocklist Card, 5: Stress Card, 0xff: Mother Card</p>	0
Password	char[]	O	The Access card's password.	123456
Doors	int[]	O	The index of the doors that this card can open.	
TimeSections	int[]	O	The index of the time sections of each door that this card can open.	
VTOPosition	char[]	O	VTO position number. (This param is not supported by access control device)	01018001
ValidDateStart	char[]	O	The start of valid date, format is "yyyyMMdd hhmmss" .	20151022%20093811
ValidDateEnd	char[]	O	The end of valid date, format is "yyyyMMdd hhmmss" .	20151222%20093811
IsValid	bool	O	Is the card still valid.	true
FingerprintPacket	object	O	The fingerprint packet info.	
+Length	int	O	The length of each fingerprint packet.	500

+Count	int	O	The count of fingerprint packet.	3
Request Example				
http://192.168.1.108/cgi-bin/recordUpdater.cgi?action=updateEx&name=AccessControlCard&recno=12345&CardName=ZhangSan&CardNo=12345&UserID=102&CardStatus=0&CardType=0&Password=123456&Doors[0]=1&Doors[1]=3&Doors[2]=5&ValidDateStart=20151022%20093811&ValidDateEnd=20151222%20093811&FingerprintPacket.Length=500&FingerprintPacket.Count=3				
Content-Type: application/octet-stream				
Content-Length: <length>				
<fingerprint packet binary data>				

Response Params (OK in body)

Response Example

OK

12.3.7 Delete Access User Card and Fingerprint

- Delete Access user card and fingerprint record by recno

The video talk device can not delete fingerprint data.

Request URL	http://<server>/cgi-bin/recordUpdater.cgi?action=remove			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
name	char[]	R	Access user card and fingerprint record name, should be "AccessControlCard".	"AccessControlCard"
recno	int	R	The record id.	12345
Request Example				
http://192.168.1.108/cgi-bin/recordUpdater.cgi?action=remove&name=AccessControlCard&recno=12345				

Response Params (OK in body)

Response Example

OK

- Delete all Access user card and fingerprint records

The video talk device can not delete fingerprint data.

Request URL	http://<server>/cgi-bin/recordUpdater.cgi?action=clear			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
name	char[]	R	Access user card and fingerprint record name, should be "AccessControlCard".	"AccessControlCard"
Request Example				

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

Response Params (OK in body)

Response Example

OK

12.3.8 Find Access User Card and Fingerprint

- Find Access user card by condition

Request URL	http://<server>/cgi-bin/recordFinder.cgi?action=find			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
name	char[]	R	Record Name, should be "AccessControlCard".	"AccessControlCard"
count	int	O	Max result to return, default is 1024.	100
StartTime	char[]	O	The start of the record's CreateTime.	123456700
EndTime	char[]	O	The End of the record's CreateTime.	123456800
condition	object	O	Search condition.	
+CardNo	char[]	O	Access user card number.	1234
+UserID	char[]	O	The user's id.	12345
+IsValid	bool	O	The access card valid or not.	true
Request Example				
http://192.168.1.108/cgi-bin/recordFinder.cgi?action=find&name=AccessControlCard&condition.UserID=103&StartTime=123456700&EndTime=123456800&count=100				

Response Params (key=value format in body)

Name	Type	R/O	Description	Example
totalCount	int	O	The total record num that find.	1000
found	int	O	The record num that returned.	100
records	array<object>	R	The records that returned.	
+RecNo	int	R	The record id.	
+CreateTime	int	O	The create time of record.	
+CardName	char[]	R	The access user card name, max string length is 32.	ZhangSan
+CardNo	char[]	R	The access user card number.	12345
+UserID	char[]	R	The user's id.	10000
+CardStatus	int	R	Card status, 0 means normal, and abnormal status can be sum of following: • 1 << 0: report the loss of the	0

			<p>card</p> <ul style="list-style-type: none"> • 1 << 1: the card is withdrawn • 1 << 2: the card is frozen • 1 << 3: the card is arrearage • 1 << 4: the card exceeds the time limit • 1 << 5: the card is arrearage but still can open the door, and there will be voice prompts. 	
+CardType	int	O	<p>The card type:</p> <p>0: Normal Card, 1: VIP Card, 2: Visitor Card, 3: Patrol Card, 4: Blocklist Card, 5: Stress Card, 0xff: Mother Card</p>	0
+Password	char[]	O	<p>The Access card's password. (This param is not supported by video talk device)</p>	123456
+Doors	int[]	O	<p>The index of the doors that this card can open. (This param is not supported by video talk device)</p>	
+TimeSections	int[]	O	<p>The index of the time sections of each door that this card can open. (This param is not supported by video talk device)</p>	
+VTOPosition	char[]	O	<p>VTO position number. (This param is not supported by access control device)</p>	01018001
+ValidDateStart	char[]	O	<p>The start of valid date, format is "yyyyMMdd hhmmss".</p>	20151022%20093811
+ValidDateEnd	char[]	O	<p>The end of valid date, format is "yyyyMMdd hhmmss".</p>	20151222%20093811
+IsValid	bool	O	<p>Is the card still valid. (This param is not supported by video talk device)</p>	true
+Handicap	bool	O	<p>is the cad of extend time</p>	true
+CitizenIDNo	char[]	O	<p>citizen card id</p>	34200000000000000000
+RepeatEnterRoute	uint[12]	O		[1,2]
+RepeatEnterRouteTimeout	int	O	<p>timeout of repeater enter route, unit: s</p>	300
+UserType	int	O	<p>user type, it can be { 1: normal user 2: user in Blocklist }</p>	0
+FirstEnter	bool	O	<p>is with first enter authority</p>	true
+DynamicCheck	char[]	O	<p>dynamic check code</p>	235843758

Code				
+UseTime	int	O	count of use	10
+SpecialDaysSchedule	uint[]	O	Holiday plan identification. The value is the subscript number configured by SpecialDaysSchedule	[1,2]
+Section	char[64]	O	Section name	"XXXX"
+SectionID	uint32	O	Section ID	"123"
+Authority	uint8	O	1: Admin; 2: General user	2
+UserName	char[32]	O	(deprecated)	"9901"
+PersonId	char[32]	O	(deprecated)	"101"

Response Example

```

totalCount=1000
found=100
records[0].RecNo=12345
records[0].CreateTime=123456789
records[0].CardName=ZhangSan
records[0].CardNo=300
records[0].UserID=103
records[0].CardStatus=0
records[0].CardType=0
records[0].Doors[0]=1
records[0].Doors[1]=3
records[0].Doors[2]=5
records[0].VTOPosition=01018001
records[0].ValidStart=20151022 093811
records[0].ValidEnd=20151222 093811
records[0].IsValid=true
records[0].Handicap=false
records[0].CitizenIDNo=
records[0].RepeatEnterRouteTimeout=4294967295
records[0].UserType =0
records[0].FirstEnter=false
records[0].DynamicCheckCode=
records[0].UseTime=0
...
records[1].RecNo=13579
records[1].CreateTime=123456799
records[1].StaticsticsTime=123456799
records[1].CardName=ZhangSan
records[1].CardNo=302
records[1].UserID=103
records[1].CardStatus=0
records[1].CardType=0

```

```

records[1].Doors[0]=2
records[1].Doors[1]=4
records[1].Doors[2]=6
records[1].VTOPosition=01018002
records[1].ValidStart=20151022 093811
records[1].ValidEnd=20151222 093811
records[1].IsValid=true
records[1].Handicap=false
records[1].CitizenIDNo=
records[1].RepeatEnterRouteTimeout=4294967295
records[1].UserType=0
records[1].FirstEnter=false
records[1].DynamicCheckCode=
records[1].UseTime=0
...

```

- Find Access user card by recno

Request URL	http://<server>/cgi-bin/recordUpdater.cgi?action=get		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
name	char[]	R	Record Name, should be "AccessControlCard".
recno	int	R	The record id.
Request Example			
http://192.168.1.108/cgi-bin/recordUpdater.cgi?action=get&name=AccessControlCard&recno=12345			

Response Params (key=value format in body)			
Name	Type	R/O	Description
record	object	R	The record that returned.
+RecNo	int	R	The record id.
+CreateTime	int	R	The create time of record.
+CardName	char[]	R	The access user card name, max string length is 32.
+CardNo	char[]	R	The access user card number.
+UserID	char[]	R	The user's id.
+CardStatus	int	R	Card status, 0 means normal, and abnormal status can be sum of following: <ul style="list-style-type: none"> • 1 << 0: report the loss of the card • 1 << 1: the card is withdrawn • 1 << 2: the card is frozen • 1 << 3: the card is arrearage • 1 << 4: the card exceeds the time limit

			<ul style="list-style-type: none"> • 1 << 5: the card is arrearage but still can open the door, and there will be voice prompts. 	
+CardType	int	O	The card type: 0: Normal Card, 1: VIP Card, 2: Visitor Card, 3: Patrol Card, 4: Blocklist Card, 5: Stress Card, 0xff: Mother Card	0
+Password	char[]	O	The Access card's password. (This param is not supported by video talk device)	123456
+Doors	int[]	O	The index of the doors that this card can open. (This param is not supported by video talk device)	
+TimeSections	int[]	O	The index of the time sections of each door that this card can open. (This param is not supported by video talk device)	
+VTOPosition	char[]	O	VTO position number. (This param is not supported by access control device)	01018001
+ValidDateStart	char[]	O	The start of valid date, format is "yyyyMMdd hhmmss".	20151022%20093811
+ValidDateEnd	char[]	O	The end of valid date, format is "yyyyMMdd hhmmss".	20151222%20093811
+IsValid	bool	O	Is the card still valid. (This param is not supported by video talk device)	true
+Handicap	bool	O	is the cad of extend time	true
+CitizenIDNo	char[]	O	citizen card id	3420000000000000000
+RepeatEnterRouteTimeout	int	O	timeout of repeater enter route, unit: s	300
+UserType	int	O	user type, it can be { 1: normal user 2: user in Blocklist }	0
+FirstEnter	bool	O	is with first enter authority	true
+DynamicCheckCode	char[]	O	dynamic check code	235843758
+UseTime	int	O	count of use	10

Response Example

```

record.RecNo=4
record.CreateTime=123456789
record.CardName=ZhangSan
record.CardNo=300
record.UserID=103

```

```

record.CardStatus=0
record.CardType=0
record.Doors[0]=1
record.Doors[1]=3
record.Doors[2]=5
record.VTOPosition=01018001
record.ValidStart=20151022 093811
record.ValidEnd=20151222 093811
record.IsValid=true
records.Handicap=false
records.CitizenIDNo=
records.RepeatEnterRouteTimeout=4294967295
records.UserType=0
records.FirstEnter=false
records.DynamicCheckCode=
records.UseTime=0
...

```

- Find Access user card and fingerprint by recno

This api is not supported by video talk device

Request URL	http://<server>/cgi-bin/recordUpdater.cgi?action=getEx		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
name	char[]	R	Record Name, should be "AccessControlCard".
recno	int	R	The record id.
Request Example			
http://192.168.1.108/cgi-bin/recordUpdater.cgi?action=getEx&name=AccessControlCard&recno=12345			

Response Params (key=value format in body)				
Name	Type	R/O	Description	Example
record	object	R	The record that returned.	
+RecNo	int	R	The record id.	
+CreateTime	int	R	The create time of record.	
+CardName	char[]	R	The access user card name, max string length is 32.	ZhangSan
+CardNo	char[]	R	The access user card number.	12345
+UserID	char[]	R	The user's id.	10000
+CardStatus	int	R	Card status, 0 means normal, and abnormal status can be sum of following: • 1 << 0: report the loss of the card • 1 << 1: the card is withdrawn • 1 << 2: the card is frozen	0

			<ul style="list-style-type: none"> • 1 << 3: the card is arrearage • 1 << 4: the card exceeds the time limit • 1 << 5: the card is arrearage but still can open the door, and there will be voice prompts. 	
+CardType	int	O	The card type: 0: Normal Card, 1: VIP Card, 2: Visitor Card, 3: Patrol Card, 4: Blocklist Card, 5: Stress Card, 0xff: Mother Card	0
+Password	char[]	O	The Access card's password. (This param is not supported by video talk device)	123456
+Doors	int[]	O	The index of the doors that this card can open. (This param is not supported by video talk device)	
+TimeSections	int[]	O	The index of the time sections of each door that this card can open. (This param is not supported by video talk device)	
+VTOPosition	char[]	O	VTO position number. (This param is not supported by access control device)	01018001
+ValidDateStart	char[]	O	The start of valid date, format is "yyyyMMdd hhmmss".	20151022%20093811
+ValidDateEnd	char[]	O	The end of valid date, format is "yyyyMMdd hhmmss".	20151222%20093811
+IsValid	bool	O	Is the card still valid. (This param is not supported by video talk device)	true
+Handicap	bool	O	is the cad of extend time	true
+CitizenIDNo	char[]	O	citizen card id	34200000000000000000
+RepeatEnterRouteTimeout	int	O	timeout of repeater enter route, unit: s	300
+UserType	int	O	user type, it can be { 1: normal user 2: user in Blocklist }	0
+FirstEnter	bool	O	is with first enter authority	true
+DynamicCheckCode	char[]	O	dynamic check code	235843758
+UseTime	int	O	count of use	10
+FingerprintPacket	object	O	The fingerprint packet info.	500
++Length	int	O	The length of each fingerprint	3

			packet.	
++Count	int	O	The count of fingerprint packet.	
Response Example				
HTTP/1.1 200 OK				
Server: Device/1.0				
Content-Type: multipart/x-mixed-replace; boundary=<boundary>				
Content-Length: <length>				
--<boundary>				
Content-Type: text/plain				
Content-Length: <length>				
record.RecNo=4				
record.CreateTime=123456789				
record.CardName=ZhangSan				
record.CardNo=300				
record.UserID=103				
record.CardStatus=0				
record.CardType=0				
record.Doors[0]=1				
record.Doors[1]=3				
record.Doors[2]=5				
record.VTOPosition=01018001				
record.ValidStart=20151022 093811				
record.ValidEnd=20151222 093811				
record.IsValid=true				
records.Handicap=false				
records.CitizenIDNo=				
records.RepeatEnterRouteTimeout=4294967295				
records.UserType=0				
records.FirstEnter=false				
records.DynamicCheckCode=				
records.UseTime=0				
...				
record.FingerprintPacket.Length=500				
record.FingerprintPacket.Count=3				
--<boundary>				
Content-Type: application/octet-stream				
Content-Length: <length>				
<fingerprint packet binary data>				
--<boundary>--				

12.3.9 Get the Total Number of Records of Access User Card and Fingerprint

Request URL	http://<server>/cgi-bin/recordFinder.cgi?action=getQuerySize		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
name	char[]	R	Access user card and fingerprint record name, should be "AccessControlCard".
Request Example			
http://192.168.1.108/cgi-bin/recordFinder.cgi?action=getQuerySize&name=AccessControlCard			

Response Params (key=value format in body)				
Name	Type	R/O	Description	Example
count	int	R	The total number of records.	150
Response Example				count=150

12.4 Access User Account (V2)

12.4.1 Add Access User

Insert or update multiple new users. Update when the current user exists. This only applies to the AccessUser entity here.

Request URL	http://<server>/cgi-bin/AccessUser.cgi?action=insertMulti		
Method	POST		
Request Params (JSON format in body)			
Name	Type	R/O	Description
UserList	object[]	R	User list, with up to 10 entries
+UserID	string	R	User ID
+UserName	string	O	User name
+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: Extend time user
+UseTime	uint16	O	Limit of passing times for guest users
+IsFirstEnter	bool	O	First user authority or not. false: No; true: Yes
+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: General user

+CitizenIDNo	string	O	ID card number	
+Password	string	O	The password when unlocking by card + password. The password when unlocking by User ID + 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.	[1,2,3,4]
+SpecialDaysSchedule	uint32[64]	O	Holiday plan identification. The value is the subscript number configured by SpecialDaysSchedule (defined in the configuration).	[1,2]
+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.	

Request Example

```
{
  "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 Params (OK in body)

Response Example

OK

12.4.2 Modify Access User

Update information of multiple users.

Request URL	http://<server>/cgi-bin/AccessUser.cgi?action=updateMulti		
Method	POST		
Request Params (JSON format in body)			
Name	Type	R/O	Description
UserList	object[]	R	User list, with up to 10 entries
+UserID	string	R	User ID
+UserName	string	O	User name
+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: Extend time user
+UseTime	uint16	O	Limit of passing times for guest users
+IsFirstEnter	bool	O	First user authority or not. false: No; true: Yes
+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: General user
+CitizenIDNo	string	O	ID card number
+Password	string	O	The password when unlocking by card + password. The password when unlocking by User ID + 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.
Request Example			

```
{
  "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 Params (OK in body)

Response Example

OK

12.4.3 Delete All Access User

Delete all user information.

Request URL	http://<server>/cgi-bin/AccessUser.cgi?action=removeAll		
Method	GET		
Request Params (none)			
Name	Type	R/O	Description
Request Example			
http://192.168.1.108/cgi-bin/AccessUser.cgi?action=removeAll			

Response Params (OK in body)

Response Example

OK

12.4.4 Delete Multiple Access Users

Delete information of multiple users.

Request URL	http://<server>/cgi-bin/AccessUser.cgi?action=removeMulti		
Method	GET		
Request Params (key=value format at URL)			

Name	Type	R/O	Description	Example
UserIDList	char[][]	R	User ID list, with up to 10 entries	["103", "102"]
Request Example				
http://192.168.1.108/cgi-bin/AccessUser.cgi?action=removeMulti&UserIDList[0]=103&UserIDList[1]=102				

Response Params (OK in body)

Response Example

OK

12.4.5 Find Multiple Access Users

Search for information of multiple users.

Request URL	http://<server>/cgi-bin/AccessUser.cgi?action=list		
Method	GET		
Request Params (key=value format at URL)			
Name	Type	R/O	Description
UserIDList	char[][]	R	User ID list, with up to 10 entries
Request Example			
http://192.168.1.108/cgi-bin/AccessUser.cgi?action=list&UserIDList[0]=103&UserIDList[1]=102			

Response Params (key=value format in body)

Name	Type	R/O	Description	Example
Users	object[]	R	The records that returned	
+ UserID	string	R	User ID	1
+UserName	string	O	User name	
+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: Extend time user	
+UseTime	uint16	O	Limit of passing times for guest users	
+IsFirstEnter	bool	O	First user authority or not. false: No; true: Yes	
+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: General 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 Example

```

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
...

```

12.4.6 Start Find Access User Related Information

Start searching for user related information.

Request URL	http://<server>/cgi-bin/AccessUser.cgi?action=startFind		
Method	GET		
Request Params (key=value format at URL)			
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	User name	
+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: Extend time user	
+UseTime	uint16	O	Limit of passing times for guest users	
+IsFirstEnter	bool	O	First user authority or not. false: No; true: Yes	
+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: General 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.	

Request Example

<http://192.168.1.108/cgi-bin/AccessUser.cgi?action=startFind&Condition.UserID=1>

Response Params (JSON format in body)				
Name	Type	R/O	Description	Example
Token	uint32	R	Search token.	1234
Total	uint32	R	Total number of entries found this time	20
Caps	uint32	R	Search capability: Maximum	20

			number of records that can be returned each time.
--	--	--	---

Response Example

```
{
  "Token": 1234,
  "Total": 20,
  "Caps ": 20
}
```

12.4.7 Get Find Result of Access User Related Information

Get user related information.

Request URL	http://<server>/cgi-bin/AccessUser.cgi?action=doFind		
Method	GET		
Request Params (key=value format at URL)			
Name	Type	R/O	Description
Token	int	R	Search token
Offset	int	R	Offset
Count	int	R	Number of entries obtained this time
Request Example			
http://192.168.1.108/cgi-bin/AccessUser.cgi?action=doFind&Token=1234&Offset=0&Count=20			

Response Params (JSON format in body)			
Name	Type	R/O	Description
info	object[]	R	user info
+UserID	string	R	User ID
+UserName	string	O	User name
+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: Extend time user
+UseTime	uint16	O	Limit of passing times for guest users
+IsFirstEnter	bool	O	First user authority or not. false: No; true: Yes
+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: General 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 Example

```
{
  "Info": [ {
    "UserID": "102",
    "UserName": "",
    "UserType": 0,
    "UseTime": 1,
    ...
  }, { ... }, ... ]
}
```

12.4.8 Stop Find Access User Related Information

Get user related information.

Request URL	http://<server>/cgi-bin/AccessUser.cgi?action=stopFind		
Method	GET		
Request Params (key=value format at URL)			
Name	Type	R/O	Description
Token	int	R	Search token.
Request Example			
http://192.168.1.108/cgi-bin/AccessUser.cgi?action=stopFind&Token=1234			

Response Params (OK in body)

Response Example

```
OK
```

12.4.9 Add Multiple Access Cards

Send information of multiple card numbers.

Request URL	http://<server>/cgi-bin/AccessCard.cgi?action=insertMulti		
Method	POST		
Request Params (JSON format in body)			
Name	Type	R/O	Description
CardList	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
Request Example			
{ CardList["UserID" : "100013" "CardNo" : "" "CardType" : 0 "CardName" : "201-zhangSan" "CardStatus" : 0 } ,...,{}] }			
Response Params (OK in body)			
Response Example			
OK			

12.4.10 Modify Multiple Access Cards

Update information of multiple card numbers.

Request URL	http://<server>/cgi-bin/AccessCard.cgi?action=updateMulti			
Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
CardList	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	
Request Example				
<pre>{ CardList[{ "UserID" : "100013" "CardNo" : "" "CardType" : 0 "CardName" : "201-ZhangSan" "CardStatus" : 0 } ,...,{}] }</pre>				

Response Params (OK in body)

Response Example

OK

12.4.11 Delete All Access Cards

Remove all card number information.

Request URL	http://<server>/cgi-bin/AccessCard.cgi?action=removeAll		
Method	GET		
Request Params (none)			

Name	Type	R/O	Description	Example
Request Example				
http://192.168.1.108/cgi-bin/AccessCard.cgi?action=removeAll				
Response Params (OK in body)				
Response Example				
OK				

12.4.12 Delete Multiple Access Cards

Remove information of multiple card numbers.

Request URL	http://<server>/cgi-bin/AccessCard.cgi?action=removeMulti		
Method	GET		
Request Params (key=value format at URL)			
Name	Type	R/O	Description
CardNoList	char[10][]	R	User card number list, with up to 10 entries
Request Example			
http://192.168.1.108/cgi-bin/AccessCard.cgi?action=removeMulti&CardNoList[0]=12345678&CardNoList[1]=12345687			

Response Params (OK in body)
Response Example
OK

12.4.13 Find Multiple Access Cards

Search for information of multiple card numbers.

Request URL	http://<server>/cgi-bin/AccessCard.cgi?action=list		
Method	GET		
Request Params (key=value format at URL)			
Name	Type	R/O	Description
CardNoList	char[10][]	R	User card number list, with up to 10 entries
Request Example			
http://192.168.1.108/cgi-bin/AccessCard.cgi?action=list&CardNoList[0]=12345678&CardNoList[1]=12345687			

Response Params (key=value format in body)
Name
Cards
+CardNo
+UserID
+CardType
Type
R/O
Description
Example

			Enumint{ 0: Ordinary card 1: VIP card 2: Guest card 3: Patrol card 4: Blocklist card 5: Duress card}	
+CardName	string	O	Card name	xxx
+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	0

Response Example

```
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
...
```

12.4.14 Start Find Access User Card Related Information

Start searching for card number related information.

Request URL	http://<server>/cgi-bin/AccessCard.cgi?action=startFind			
Method	GET			
Request Params (key=value format at URL)				
Name	Type	R/O	Description	Example
Condition	object	R	Search conditions. Users can perform conditional search according to the user information field.	
+UserID	string	O	User ID	1
+CardNo	string	O	User card number	1234
+CardType	uint16	O	Card type Enumint{ 0: Ordinary card}	0

			1: VIP card 2: Guest card 3: Patrol card 4: Blocklist card 5: Duress card	
+CardName	string	O	Card name	aaa
+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	0

Request Example

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

Response Params (JSON format in body)				
Name	Type	R/O	Description	Example
Token	uint32	R	Search token.	1234
Total	uint32	R	Total number of entries found this time	20
Caps	uint32	R	Search capability: Maximum number of records that can be returned each time.	20

Response Example

```
{
  "Token": 1234,
  "Total": 20,
  "Caps": 20
}
```

12.4.15 Get Find Result of Access User Card Related Information

Get card number related information.

Request URL	http://<server>/cgi-bin/AccessCard.cgi?action=startFind			
Method	GET			
Request Params (key=value format at URL)				
Name	Type	R/O	Description	Example
Token	int	R	Search token.	1234
Offset	int	R	Offset	0
Count	int	R	Number of entries obtained this time	20

Request Example

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

Response Params (JSON format in body)

Name	Type	R/O	Description	Example
info	object[]	R	info	
+UserID	string	O	User ID	1
+CardNo	string	O	User card number	1234
+CardType	uint16	O	Card type Enumint{ 0: Ordinary card 1: VIP card 2: Guest card 3: Patrol card 4: Blocklist card 5: Duress card	0
+CardName	string	O	Card name	aaa
+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	0

Response Example

```
{
  "Info": [ {
    "UserID": "102",
    ...
  }, { ... }, ... ]
}
```

12.4.16 Stop Find Access User Card Related Information

Stop searching for card number related information.

Request URL	http://<server>/cgi-bin/AccessCard.cgi?action=stopFind		
Method	GET		
Request Params (key=value format at URL)			
Name	Type	R/O	Description
Token	int	R	Search token.
Request Example			
http://192.168.1.108/cgi-bin/AccessCard.cgi?action=stopFind&Token=1234			

Response Params (OK in body)
Response Example
OK

12.4.17 Add Multiple Access User Fingerprint

Insert multiple fingerprint information.

Request URL	http://<server>/cgi-bin/AccessFingerprint.cgi?action=insertMulti		
Method	POST		
Request Params (JSON format in body)			
Name	Type	R/O	Description
vecPackets	char[][]	R	Fingerprint feature data
AccessFingerprints	object[]	R	Fingerprint list
+UserID	char[]	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	char[][]	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.
Request Example			
{			
"vecPackets": ["xxxx", "xxxx", ...],			
"AccessFingerprints": [
{			
"UserID": "102",			
"FingerprintPacket" :			
{			
"Length" : 810,			
"Count" : 3,			
"DuressIndex" : 2			
}			
}			
]			
}			
Response Params (OK in body)			
Response Example			
OK			

12.4.18 Modify Access User Fingerprint

Update multiple fingerprint information.

Request URL	http://<server>/cgi-bin/AccessFingerprint.cgi?action=updateMulti		
Method	POST		
Request Params (JSON format in body)			
Name	Type	R/O	Description
vecPackets	char[][]	R	Binary fingerprint data (Binary Data). ["xxxx", "xxxx", ...]
AccessFingerprints	object[]	R	Fingerprint list
+UserID	char[]	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	char[][]	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.
Request Example			
{	<pre>"vecPackets": ["xxxx", "xxxx", ...], "AccessFingerprints": ["UserID": "102", "FingerprintPacket" : { "Length" : 810, "Count" : 3, "DuressIndex" : 2 }] }</pre>		

Response Params (OK in body)

Response Example

OK

12.4.19 Delete All Access User Fingerprint

Delete all fingerprint information.

Request URL	http://<server>/cgi-bin/AccessFingerprint.cgi?action=removeAll			
Method	GET			
Request Params (none)				
Name	Type	R/O	Description	Example
Request Example				http://192.168.1.108/cgi-bin/AccessFingerprint.cgi?action=removeAll

Response Params (OK in body)

Response Example

OK

12.4.20 Delete Multiple Access User Fingerprint

Delete fingerprint information.

Request URL	http://<server>/cgi-bin/AccessFingerprint.cgi?action=removeMulti			
Method	GET			
Request Params (key=value format at URL)				
Name	Type	R/O	Description	Example
UserIDList	char[][]	R	User card number list	["101"]
Request Example				http://192.168.1.108/cgi-bin/AccessFingerprint.cgi?action=removeMulti&UserIDList[0]=101

Response Params (OK in body)

Response Example

OK

12.4.21 Find Access User Fingerprint

Search for user fingerprint information.

Request URL	http://<server>/cgi-bin/AccessFingerprint.cgi?action=get			
Method	GET			
Request Params (key=value format at URL)				
Name	Type	R/O	Description	Example
UserID	string	R	user ID	1
Request Example				http://192.168.1.108/cgi-bin/AccessFingerprint.cgi?action=get&UserID=1

Response Params (key=value format in body)

Name	Type	R/O	Description	Example
FingerprintData	BinaryData	R	Binary fingerprint data (Binary)	xxx

			Data).	
FingerprintPacket	object	R	Description of fingerprint data	
+Length	uint32	R	Length of a single fingerprint package	810
+Count	uint32	R	Number of fingerprint packages	3
+DataURL	char[][]	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.	1

Response Example

```
FingerprintPacket.Length=810
FingerprintPacket.Count=3
FingerprintPacket.DuressIndex=1
FingerprintData=xxx
```

12.4.22 Add Multiple Access User Face

Insert multiple face information.

Request URL	http://<server>/cgi-bin/AccessFace.cgi?action=insertMulti			
Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
FaceList	object[]	R	Face list, with up to 10 entries	
+UserID	char[]	R	User ID	"102"
+FaceData	char[][]	O	Base64 of red light face templates, 8192 Bytes/per face, maximum 20.	["xxxx", "xxxx", ...]
+PhotoData	char[][]	O	Base64 of white light face images, 200kx5.	["yyyy", "yyyy", ...]
+PhotoURL	char[][]	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.	["yyyy", "yyyy", ...]
Request Example				
{				
"FaceList": [

```
{
    "UserID": "102",
    "FaceData": [ "xxxx", "xxxx", ... ],
    "PhotoData": [ "yyyy", "yyyy", ... ],
    "PhotoURL": [ "yyyy", "yyyy", ... ],
},
.....
]
}
```

Response Params (OK in body)

Response Example

OK

12.4.23 Update Multiple Access User Face

Update multiple face information.

Request URL	http://<server>/cgi-bin/AccessFace.cgi?action=updateMulti		
Method	POST		
Request Params (JSON format in body)			
Name	Type	R/O	Description
FaceList	object[]	R	Face list, with up to 10 entries
+UserID	char[]	R	User ID
+FaceData	char[][]	O	Base64 of red light face templates, 8192 Bytes/per face, maximum 20.
+PhotoData	char[][]	O	Base64 of white light face images, 200kx5.
+PhotoURL	char[][]	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.

Request Example

```
{
    "FaceList":[{
        "UserID": "102",
        "FaceData": [ "xxxx", "xxxx", ... ],
        "PhotoData": [ "yyyy", "yyyy", ... ],
        "PhotoURL": [ "yyyy", "yyyy", ... ],
    },
    .....
]
}
```

Response Params (OK in body)

Response Example

OK

12.4.24 Delete All Access User Face

Delete all face information.

Request URL	http://<server>/cgi-bin/AccessFace.cgi?action=removeAll		
Method	GET		
Request Params (none)			
Name	Type	R/O	Description
Request Example			
http://192.168.1.108/cgi-bin/AccessFace.cgi?action=removeAll			

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

OK

12.4.25 Delete Multiple Access User Face

Delete multiple face information.

Request URL	http://<server>/cgi-bin/AccessFace.cgi?action=removeMulti		
Method	GET		
Request Params (key=value format at URL)			
Name	Type	R/O	Description
UserIDList	char[]	R	User card number list, with up to 10 entries
Request Example			
http://192.168.1.108/cgi-bin/AccessFace.cgi?action=removeMulti&UserIDList[0]=101&UserIDList[1]=102			

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

OK

12.4.26 Find Access User Face

Search for access user face information.

Request URL	http://<server>/cgi-bin/AccessFace.cgi?action=list		
Method	GET		
Request Params (key=value format at URL)			
Name	Type	R/O	Description
UserIDList	char[]	R	User ID list, with up to 10 entries

Request Example

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

Response Params (key=value format in body)

Name	Type	R/O	Description	Example
FaceDataList	object[]	R	The records that returned.	
+UserID	char[]	R	User ID	"101"
+FaceData	char[][]	O	Base64 of red light face templates, 8192 Bytes/per face, maximum 20.	["xxxx", "xxxx", ...]
+PhotoData	char[][]	O	Base64 of white light face images, 200kx5.	["yyyy", "yyyy", ...]
+PhotoURL	char[][]	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.	["yyyy", "yyyy", ...]

Response Example

FaceDataList[0].UserID=1
 FaceDataList[0].PhotoData[0]="xxxx",
 FaceDataList[0].PhotoData[1]="xxxx", ,
 FaceDataList[0].FaceData[0]="xxxx"
 FaceDataList[0].FaceData[1]="xxxx",
 ...
 FaceDataList[1].UserID=2
 FaceDataList[1].PhotoData[0]="xxxx",
 FaceDataList[1].PhotoData[1]="xxxx", ,
 FaceDataList[1].FaceData[0]="xxxx"
 FaceDataList[1].FaceData[1]="xxxx",
 ...

12.4.27 Start Find Access User Face Related Information

Start searching for face related information.

Request URL	http://<server>/cgi-bin/AccessFace.cgi?action=startFind			
Method	GET			
Request Params (key=value format at URL)				
Name	Type	R/O	Description	Example
Condition	object	R	Search conditions. Users can perform conditional search according to the user information field.	
+UserID	char[]	R	User ID	1
Request Example				
http://192.168.1.108/cgi-bin/AccessFace.cgi?action=startFind&Condition.UserID=1				

Response Params (JSON format in body)

Name	Type	R/O	Description	Example
Token	uint32	R	Search token.	1234
Total	uint32	R	Total number of entries found this time	20
Caps	uint32	R	Search capability: Maximum number of records that can be returned each time.	20

Response Example

```
{
  "Token": 1234,
  "Total": 20,
  "Caps": 20
}
```

12.4.28 Get Find Result of Access User Face Related Information

Get face related information.

Request URL	http://<server>/cgi-bin/AccessFace.cgi?action=doFind		
Method	GET		
Request Params (key=value format at URL)			
Name	Type	R/O	Description
Token	int	R	Search token.
Offset	int	R	Offset
Count	int	R	Number of entries obtained this time

Request Example

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

Response Params (JSON format in body)

Name	Type	R/O	Description	Example
info	object[]	R	The Person information that returned.	
+UserID	char[]	R	User ID	1
+FaceData	char[][]	O	Base64 of red light face templates, 8192 Bytes/per face, maximum 20.	["xxxx", "xxxx", ...]
+PhotoData	char[][]	O	Base64 of white light face images, 200kx5.	["yyyy", "yyyy", ...]
+PhotoURL	char[][]	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	["yyyy", "yyyy", ...]

		available now.	
Response Example			
{ "Info": [{ "UserID": "102", ... }, { ... }, ...] }			

12.4.29 Stop Find Access User Face Related Information

Stop searching for face related information.

Request URL	http://<server>/cgi-bin/AccessFace.cgi?action=stopFind		
Method	GET		
Request Params (key=value format at URL)			
Name	Type	R/O	Description
Token	int	R	Search token.
Request Example			
http://192.168.1.108/cgi-bin/AccessFace.cgi?action=stopFind&Token=1234			

Response Params (OK in body)
Response Example
OK

12.4.30 Access Control Protocol Capability Query

Second-Generation Access Control Protocol Capability Query

Request URL	http://<server>/cgi-bin/api/AC/getCaps		
Method	POST		
Request Params (JSON format in body)			
Name	Type	R/O	Description
WantMethods	bool	O	Whether to request to return the Methods list, please obtain according to actual needs.
WantCaps	bool	O	Whether to request to return to the capability item list, please obtain according to actual needs.
Request Example			
{ "WantMethods": True }			

Response Params			
Name	Type	R/O	Description

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
++Encryption Method	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
++HasBTCardAuth	Bool	O	Whether supporting Bluetooth card authentication False- no; true- yes	false
++HasFaceAuth	bool	O	Whether ** 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: Yes	0
++MaxInsertRate	uint16	O	The general maximum number of data can be sent at a time. Default conservative set value similar to 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

			False: Not supported; true: Supported.	
+++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	Supports fast import function Enumint { 0: Not supported 1: Supported } Not supported if it does not exist.	0
++SupportFastCheck	enumint	O	Supports 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
++IncrementalImport	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	2

			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 back-end fingerprint comparison.	
++SupportHelmet	enumint	O	Supports safety helmet function. (Takend 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/"
++FacelimagePath	char[512]	O	** The local storage directory of base pictures.	"/mnt/data/Facelimage/"
++HasIrisAuth	bool	O	Whether ** identification authentication is supported.	False
++IrisImagePath	char[512]	O	Added ** opening authentication type, and added ** base storage path.	"/mnt/data/Image/"
+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 ** 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
++MaxFingerprintsSize	uint16	O	Maximum bytes number of single fingerprint data.	810
++MaxFingerprints	uint32	O	Fingerprint storage limit.	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	** Storage Limit.	600
++RecognitionType	uint8	O	** Recognition Type.	1
++RecognitionAlgorithmVendor	uint16	O	** Identification algorithm provider. 0: Unknown; 1: Dahua; 2: SenseTime; 3: Yitu; 4: Hanvon; 5: Huoyan.	0
++RecognitionVersion	uint32	O	** Identification 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 photo, KB.	20
++MaxPhotoSize	uint16	O	Maximum size of white light photo, KB.	20
++MaxGetPhotoRate	uint16	O	The maximum amount of acquisitions per time by the white light 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 the characteristic value is supported.	True-
++MultiFaceD	object	O	-	

etect				
+++Support	bool	O	Whether multi-detection identification is supported.	True
+++MaxNums	uint32	O	The maximum number of detections supported at a time.	3
+AccessIris	object	O	AccessIris capability set.	
++MaxInsertRate	uint16	O	Maximum number of inserts per time.	10
++MinIrisPhotoSize	uint16	O	Minimum size of the image, unit KB.	1
++MaxIrisPhotoSize	uint16	O	Maximum size of the image, unit KB.	50
++MaxIrisGroups	uint16	O	Maximum number of groups supported by each user**.	1
++RecognitionAlgorithmVendor	uint16	O	** Identification algorithm provide identification, 0: Unknown; 1: Dahua.	0
++RecognitionVersion	uint32	O	** The algorithm (model) version number, if the version number has multiple digits, every 8 bit represents a version from high to low according to Major/Minor, for example, 1.5.2 represents as 0x00010502.	
++MaxIrisesCount	uint32	O	** Storage Limit.	10000
+AccessBTCard	object	O	Capability set of the Bluetooth card	
++MaxInsertRate	uint32	O	Maximum number of each insertion	5
++MaxCards	uint32	O	Maximum number of storage cards	5000

Response Example

```
{
  "Caps": {
    "AC": {
      ...
    }
  }
}
```

12.4.31 Inserting Bluetooth Card Information

Request URL	http://<server>/cgi-bin/api/AccessBTCard/insertMulti			
Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example

BTCardList	object[]	R		
+AKID	uint64	R	The sequence number of the Bluetooth card	123456
+UserID	char[32]	R	User ID	"123"
+BTCardNo	char[32]	R	Bluetooth card number	"1234567"

Request Example

```
{
    "BTCardList": [, ....]
}
```

Response Params (JSON format in body)

Name	Type	R/O	Description	Example
Response Example				

12.4.32 Searching for Information on Bluetooth Card

Request URL	http://<server>/cgi-bin/api/AccessBTCard/list			
Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
CardAKIDList	uint64[5]	R	List of Bluetooth card sequence number	[1,2,3,4,5]
Request Example				

Request Example

```
{
    "CardAKIDList": [1,2,3,4,5]
}
```

Response Params (JSON format in body)				
Name	Type	R/O	Description	Example
BTCards	object[]	O	The list of searched Bluetooth card information.	
+AKID	uint64	R	Sequence number of the Bluetooth card.	12345
+UserID	char[32]	R	User ID	"123"
+BTCardNo	char[32]	R	Bluetooth card number.	"1234567"
Response Example				

12.4.33 Deleting All Bluetooth Card Information

Request URL	http://<server>/cgi-bin/api/AccessBTCard/removeAll			
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
Response Example				
{}				

12.4.34 Deleting Bluetooth Card Information

Request URL	http://<server>/cgi-bin/api/AccessBTCard/removeMulti			
Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
CardAKIDList	uint64[5]	R	List of Bluetooth card sequence number	[1,2,3,4,5]
Request Example				
<pre>{ "CardAKIDList": [1,2,3,4,5] }</pre>				

Response Params (JSON format in body)				
Name	Type	R/O	Description	Example
Response Example				
{}				

12.4.35 Searching for Information on Access Control Bluetooth Card

Search for information on access control Bluetooth card.

Request URL	http://<server>/cgi-bin/api/AccessBTCard/startFind			
Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
Condition	object	O	Search conditions	
+UserID	char[32]	O	User ID	"123"
+AKID	uint64	O	Sequence number of the Bluetooth card	123456
+BTCardNo	char[32]	O	Bluetooth card number	"1234567"
Request Example				
<pre>{ "Condition": { } }</pre>				

Response Params (JSON format in body)				
Name	Type	R/O	Description	Example

Token	uint32	R	Search token	
Total	uint32	R	Total number of entries returned	10
Caps	uint32	R	Search capability. Maximum number of entries provided by doFind each time.	

Response Example

```
{
  "Token": ,
  "Total": 10,
  "Caps": 
}
```

12.4.36 Acquiring the Search Results of Bluetooth Card

Request URL	http://<server>/cgi-bin/api/AccessBTCard/doFind			
Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
Token	uint32	R	Search token	123
Offset	uint32	R	The offset position of the search	12
Count	uint32	R	The number of entries required for this search, which cannot exceed the Caps of the startFind.	6

Request Example

```
{
  "Token": 123,
  "Offset": 12,
  "Count": 6
}
```

Response Params (JSON format in body)				
Name	Type	R/O	Description	Example
Info	AccessBT Card[1000]	R	Bluetooth card information	[,]
Response Example				
{				
	"Info": [,]			
}				

12.4.37 Ending the Bluetooth Card Search

End the Bluetooth card search.

Request URL	http://<server>/cgi-bin/api/AccessBTCard/stopFind			
Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example

Token	uint32	R	Search token	2342
Request Example				
{				

Response Params (JSON format in body)				
Name	Type	R/O	Description	Example
Response Example				
{				

12.5 Admin Password

12.5.1 Add Access Control Admin Password

Request URL	http://<server>/cgi-bin/recordUpdater.cgi?action=insert			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
name	char[]	R	Access control admin password record name, should be "AccessControlCustomPassword".	AccessControlCus tomPassword
UserID	char[]	R	The user's id.	101
OpenDoorPassw ord	char[]	R	Ths user's open door password.	123456
AlarmPassword	char[]	O	Ths user's alarm password. (This param is not supported by video talk device)	
Doors	int[]	O	The index of the doors that admin password can open. (This param is not supported by video talk device)	
TimeSections	int[]	O	The index of the time sections of each door that this card can open. (This param is not supported by video talk device)	
VTOPosition	char[]	O	VTO position number. (This param is not supported by access control device)	
ValidDateStart	char[]	O	The start of valid date, format is "yyyyMMdd hhmmss".	
ValidDateEnd	char[]	O	The end of valid date, format is "yyyyMMdd hhmmss".	
ValidCounts	int	O	The password's valid counts.	
OriginSmartGate	char[]	O	The origin smart gateway address.	

Way		(Video talk device customize.)	
Request Example			
http://192.168.1.108/cgi-bin/recordUpdater.cgi?action=insert&name=AccessControlCustomPassword&UserID=102&OpenDoorPassword=123456&Doors[0]=1&Doors[1]=3&Doors[2]=5&VTOPosition=01018001&ValidDateStart=20151022%20093811&ValidDateEnd=20151222%20093811&ValidCounts=30			

Response Params (key=value format in body)				
Name	Type	R/O	Description	Example
recno	int	R	The record id.	1234
Response Example				
recno=12345				

12.5.2 Modify Access Control Admin Password

Note: You should provide at least one optional param to update.

Request URL	http://<server>/cgi-bin/recordUpdater.cgi?action=update			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
name	string	R	Access control admin password record name, should be "AccessControlCustomPassword".	AccessControlCustomPassword
recno	int	R	The record id.	1234
UserID	string	R	The user's id.	101
OpenDoorPassword	string	R	Ths user's open door password.	123456
AlarmPassword	string	O	Ths user's alarm password. (This param is not supported by video talk device)	
Doors	int[]	O	The index of the doors that admin password can open.	
TimeSections	int[]	O	The index of the time sections of each door that this card can open. (This param is not supported by video talk device)	
VTOPosition	string	O	VTO position number. (This param is not supported by access control device)	
ValidDateStart	string	O	The start of valid date, format is "yyyyMMdd hhmmss".	
ValidDateEnd	string	O	The end of valid date, format is "yyyyMMdd hhmmss".	
ValidCounts	int	O	The password's valid counts.	
OriginSmartGateWay	string	O	The origin smart gateway address. (Video talk device customize.)	

Request Example

```
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%20093811&ValidDateEnd=20151222%20093811
```

Response Params (OK in body)

Name	Type	R/O	Description	Example
Response Example				
OK				

12.5.3 Delete Access Control Admin Password

- Delete access control admin password record by recno

Request URL	http://<server>/cgi-bin/recordUpdater.cgi?action=remove			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
name	string	R	Access control admin password record name, should be "AccessControlCustomPassword".	AccessControlCustomPassword
recno	int	R	The record id.	12345
Request Example				
http://192.168.1.108/cgi-bin/recordUpdater.cgi?action=remove&name=AccessControlCustomPassword&recno=12345				

Response Params (OK in body)

Name	Type	R/O	Description	Example
Response Example				
OK				

- Delete all the access control admin password records

Request URL	http://<server>/cgi-bin/recordUpdater.cgi?action=clear			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
name	string	R	Access control admin password record name, should be "AccessControlCustomPassword".	AccessControlCustomPassword
Request Example				
http://192.168.1.108/cgi-bin/recordUpdater.cgi?action=clear&name=AccessControlCustomPassword				

Response Params (OK in body)

Name	Type	R/O	Description	Example
Response Example				

OK

12.5.4 Find Access Control Admin Password

- Find Access control admin password by condition

Request URL	http://<server>/cgi-bin/recordFinder.cgi?action=find			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
name	string	R	Record Name, should be "AccessControlCustomPassword".	AccessControlCustomPassword
count	int	O	Max result to return, default is 1024.	1024
StartTime	string	O	The start of the record's CreateTime.	123456700
EndTime	string	O	The End of the record's CreateTime.	123456800
condition	object	O	Search condition.	
+UserID	string	O	The user's id.	101
Request Example				
http://192.168.1.108/cgi-bin/recordFinder.cgi?action=find&name=AccessControlCustomPassword&condition.UserID=103&StartTime=123456700&EndTime=123456800&count=100				

Response Params (key=value format in body)				
Name	Type	R/O	Description	Example
totalCount	int	O	The total record num that find.	2
found	int	O	The record num that returned.	2
records	array<object>	R	The records that returned.	
+RecNo	int	R	The record id.	1234
+CreateTime	int	O	The create time of record.	
+UserID	string	R	The user's id.	101
+OpenDoorPassWord	string	R	Ths user's open door password.	123456
+AlarmPassword	string	O	Ths user's alarm password. (This param is not supported by video talk device)	
+Doors	Int[]	O	The index of the doors that admin password can open. (This param is not supported by video talk device)	
+TimeSections	Int[]	O	The index of the time sections of each door that this card can open. (This param is not supported by video talk device)	
+VTOPosition	string	O	VTO position number. (This param is not supported by video talk device)	

+ValidDateStart	string	O	The start of valid date, format is "yyyyMMdd hhmmss".	20151022 093811
+ValidDateEnd	string	O	The end of valid date, format is "yyyyMMdd hhmmss".	20151023 093811
+ValidCounts	int	O	The password's valid counts.	
+OriginSmartGateWay	string	O	The origin smart gateway address. (Video talk device customize.)	

Response Example

```

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[0].VTOPosition=01018001
records[0].ValidStart=20151022 093811
records[0].ValidEnd=20151222 093811
...
records[1].RecNo=13579
records[1].CreateTime=123456799
records[1].UserID=103
records[1].OpenDoorPassword=123456
records[1].Doors[0]=2
records[1].Doors[1]=4
records[1].Doors[2]=6
records[1].VTOPosition=01018002
records[1].ValidStart=20151022 093811
records[1].ValidEnd=20151222 093811
...

```

- Find Access control admin password by recno

Request URL	http://<server>/cgi-bin/recordUpdater.cgi?action=get			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
name	string	R	Record Name, should be " AccessControlCustomPassword ".	AccessControlCustomPassword
recno	int	R	The record id.	1024
Request Example				
http://192.168.1.108/cgi-bin/recordUpdater.cgi?action=get&name=AccessControlCustomPassword&recno=3				
Response Params (key=value format in body)				

Name	Type	R/O	Description	Example
record	object	R	The record that returned.	
+RecNo	int	R	The record id.	1234
+CreateTime	int	O	The create time of record.	
+UserID	string	R	The user's id.	101
+OpenDoorPassword	string	R	Ths user's open door password.	123456
+AlarmPassword	string	O	Ths user's alarm password. (This param is not supported by video talk device)	
+Doors	Int[]	O	The index of the doors that admin password can open. (This param is not supported by video talk device)	
+TimeSections	Int[]	O	The index of the time sections of each door that this card can open. (This param is not supported by video talk device)	
+VTOPosition	string	O	VTO position number. (This param is not supported by video talk device)	
+ValidDateStart	string	O	The start of valid date, format is "yyyyMMdd hhmmss".	20151022 093811
+ValidDateEnd	string	O	The end of valid date, format is "yyyyMMdd hhmmss".	20151023 093811
+ValidCounts	int	O	The password's valid counts.	
+OriginSmartGateway	string	O	The origin smart gateway address. (Video talk device customize.)	

Response Example

```

record.RecNo=3
record.CreateTime=123456789
record.UserID=103
record.OpenDoorPassword=123456
record.Doors[0]=1
record.Doors[1]=3
record.Doors[2]=5
record.VTOPosition=01018001
record.ValidStart=20151022 093811
record.ValidEnd=20151222 093811
...

```

12.5.5 Get the Total Number of Records of Access Control Admin Password

Request URL	http://<server>/cgi-bin/recordFinder.cgi?action=getQuerySize
Method	GET

Request Params (key=value format in URL)

Name	Type	R/O	Description	Example
name	string	R	Access control admin password record name, should be "AccessControlCustomPassword"	AccessControlCustomPassword

Request Example

http://192.168.1.108/cgi-bin/recordFinder.cgi?action=getQuerySize&name=AccessControlCustomPassword

Response Params (key=value format in body)

Name	Type	R/O	Description	Example
count	string	R	The total number of records.	150

Response Example

count=150

13

Intelligent Building APIs

13.1 Video Talk

The following Apis are supported by VTT / VTA products.

13.1.1 Subscribe Video Talk Status

Subscribe the video talk status. When client disconnect, it will unsubscribe.

Request URL	http://<server>/cgi-bin/VideoTalkPeer.cgi?action=attachState		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
Request Example			
http://192.168.1.108/cgi-bin/VideoTalkPeer.cgi?action=attachState			

Response Params (multipart ; key=value format in body)			
Name	Type	R/O	Description
SID	integer	No	Server subscription ID
state	object	No	Called state
+State	string	No	Call state Enumchar[32]{ "Ringing": passive call "Inviting": active call "Answer" "Refuse" "Cancel" "Hangup" "Busying" }
+Talkback	object	No	Transmission method, audio and video encoding format, listening port and other session description information
++Pack	string	No	Packing Mode It only supports "RTP" currently.
++Protocol	string	No	Transport protocol type Enumchar[32]{ TCP UDP }
++Type	string	No	Talk type Enumchar[32]{

			"Talk": Talk "Broadcast": Broadcast }	
++Audio	object	No	Audio Description	
+++Format	object[]	No	Supported audio decoding formats	
++++Compression	string	No	Audio compression format Enumchar[32]{ PCM ADPCM "G.711A" "G.711Mu" "G.726" "G.729" "MPEG2" AMR AAC }	PCM
++++Frequency	integer	No	Audio Sampling Frequency	44000
++++Depth	integer	No	Sampling depth	16
++Video	object	No	Video description	
+++VideoPort	integer	No	Receive video stream port When receiving multiple video channels, use different ports respectively.	7000
+++Format	object[]	No	Supported video decoding formats	
++++Compression	string	No	Video compression format Enumchar[32]{ "MPEG4" "MPEG2" "MPEG1" "MJPG" "H.263" "H.264" }	H.264
++++Frequency	integer	No	Video sampling frequency	9000
++MediaAddr	string	No	Streaming address It can be an intercom or broadcast address	224.10.10.10

Response Example

HTTP/1.1 200 OK

Server: Device/1.0

Content-Type: multipart/x-mixed-replace; boundary=<boundary>

Connection: closed

--<boundary>

Content-Type: text/plain

Content-Length: <data length>

```

SID=315
state.State=Answer
state.Talkback.Pack=RTP
state.Talkback.Protocol=UDP
state.Talkback.Type=Talk
state.Talkback.Audio.AudioPort=6000
state.Talkback.Audio.Format[0].Compression=PCM
state.Talkback.Audio.Format[0].Frequency=44000
state.Talkback.Audio.Format[0].Depth=16
state.Talkback.Audio.Format[1].Compression=G.711A
state.Talkback.Audio.Format[1].Frequency=44000
state.Talkback.Audio.Format[1].Depth=16
state.Talkback.Video.VideoPort=7000
state.Talkback.Video.Format[0].Compression=H.264
state.Talkback.Video.Format[0].Frequency=90000
state.Talkback.Video.Format[1].Compression=MJPEG
...
state.Talkback. MediaAddr=224.10.10.10
--<boundary>
...

```

13.1.2 Unsubscribe Video Talk Status

Unsubscribe the video talk status.

Request URL	http://<server>/cgi-bin/VideoTalkPeer.cgi?action=detachState			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
SID	int	R	the subscribe id, which is the response of attachState	13606
Request Example				
http://192.168.1.108/cgi-bin/VideoTalkPeer.cgi?action=detachState&SID=101				

Response Params (OK in body)
Response Example
OK

13.1.3 Invite Server on Video Talk

Start the video talk conversation.

Request URL	http://<server>/cgi-bin/VideoTalkPeer.cgi?action=invite			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
Talkback	integer	R	Channel number, starting from 1	1

+Pack	string	O	Packing Mode It only supports "RTP" currently.	RTP
+Protocol	string	O	Transport protocol type Enumchar[32]{ TCP "UDP" }	UDP
+Type	string	O	Talk type Enumchar[32]{ "Talk": Talk "Broadcast": Broadcast }	Talk
+Audio	object	O	Audio Description	
++Format	object[]	O	Supported audio decoding formats	
+++Compression	string	O	Audio compression format Enumchar[32]{ PCM "ADPCM" "G.711A" "G.711Mu" "G.726" "G.729" "MPEG2" "AMR" "AAC" }	PCM
+++Frequency	integer	O	Audio Sampling Frequency	44000
+++Depth	integer	O	Sampling depth	16
+Video	object	O	Video description	
++VideoPort	integer	O	Receive video stream port When receiving multiple video channels, use different ports respectively.	7000
++Format	object[]	O	Supported video decoding formats	
+++Compression	string	O	Video compression format Enumchar[32]{ "MPEG4" "MPEG2" "MPEG1" "MJPG" "H.263" "H.264" }	H.264
+++Frequency	integer	O	Video sampling frequency	9000
+MediaAddr	string	O	Streaming address It can be an intercom or broadcast address	224.10.10.10
Request Example				

http://192.168.1.108/cgi-bin/VideoTalkPeer.cgi?action=invite&Talkback.Protocol=UDP&Talkback.Type=Talk&Talkback.MediaAddr=224.10.10.10

Response Params (OK in body)

Response Example

OK

13.1.4 Cancel the Video Talk

Cancel video talk conversation.

Request URL	http://<server>/cgi-bin/VideoTalkPeer.cgi?action=cancel		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
Request Example			
http://192.168.1.108/cgi-bin/VideoTalkPeer.cgi?action=cancel			

Response Params (OK in body)

Response Example

OK

13.1.5 Answer the Invitation

Answer the call.

Request URL	http://<server>/cgi-bin/VideoTalkPeer.cgi?action=answer		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
Talkback	integer	R	Channel number, starting from 1
+Pack	string	O	Packing Mode It only supports "RTP" currently.
+Protocol	string	O	Transport protocol type Enumchar[32]{ TCP "UDP" }
+Type	string	O	Talk type Enumchar[32]{ "Talk": Talk "Broadcast": Broadcast }
+Audio	object	O	Audio Description
++Format	object[]	O	Supported audio decoding formats
+++Compression	string	O	Audio compression format Enumchar[32]{ PCM

			"ADPCM" "G.711A" "G.711Mu" "G.726" "G.729" "MPEG2" "AMR" "AAC" }	
+++Frequency	integer	O	Audio Sampling Frequency	44000
+++Depth	integer	O	Sampling depth	16
+Video	object	O	Video description	
++VideoPort	integer	O	Receive video stream port When receiving multiple video channels, use different ports respectively.	7000
++Format	object[]	O	Supported video decoding formats	
+++Compression	string	O	Video compression format Enumchar[32]{ "MPEG4" "MPEG2" "MPEG1" "MJPG" "H.263" "H.264" }	H.264
+++Frequency	integer	O	Video sampling frequency	9000
+MediaAddr	string	O	Streaming address It can be an intercom or broadcast address	224.10.10.10

Example

http://192.168.1.108/cgi-bin/VideoTalkPeer.cgi?action=answer&Talkback.Protocol=UDP&Talkback.Type=Talk&Talkback.MediaAddr=224.10.10.10

Response Params (OK in body)

Response Example

OK

13.1.6 Refuse to Answer the Video Talk Invitation

Refuse answer the call.

Request URL	http://<server>/cgi-bin/VideoTalkPeer.cgi?action=refuse		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
Request Example			

<http://192.168.1.108/cgi-bin/VideoTalkPeer.cgi?action=refuse>

Response Params (OK in body)

Response Example

OK

13.1.7 Hang Up

Close it when the conversation is over.

Request URL	http://<server>/cgi-bin/VideoTalkPeer.cgi?action=hangup		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
Request Example			
http://192.168.1.108/cgi-bin/VideoTalkPeer.cgi?action=hangup			

Response Params (OK in body)

Response Example

OK

13.2 Intelligent Building Record

13.2.1 Query Video Talk Log Record

This API is supported by VTO products.

Request URL	http://<server>/cgi-bin/recordFinder.cgi?action=find		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
name	string	R	It is fixed to "VideoTalkLog"
condition	object	O	condition
+CallType	string	O	call type, range is: {"Incoming", "Outgoing"}.
+EndState	string	O	end state of the call Enumchar[32]{ "Missed" "Received" }
+count	int	O	the number of records to get
Request Example			
http://192.168.1.108/cgi-bin/recordFinder.cgi?action=find&name=VideoTalkLog&condition.CallType=Incoming&condion.EndState=Missed&count=500			

Response Params (key=value format in body)

Name	Type	R/O	Description	Example
totalCount	int	O	the record count which match	2

			condition	
found	int	O	the record count to return	2
records	object[]	O	records	
+RecNo	int	O	record id	1234
+CreateTime	int	O	create time	
+CallType	string	O	all type. The range is { "Incoming" "Outgoing" }	Incoming
+EndState	string	O	the range is {"Missed" , "Received"}	Missed
+PeerNumber	string	O	peer number	501
+PeerType	string	O	peer type Enumchar[32]{ "VTO" "VTH" "VTS" }	VTO
+OfflineCall	int	O	Whether to call out when disconnected from the platform 0: Online call out, 1: Offline call out, Other: Undefined	0
+LocalNumber	string	O	local number	8001
+TalkTime	int	O	Talk time, unit: S	550
+MessageTime	int	O	Message time, unit: S	100
+PicturePath	string	O	picture path	/picutre
+RecordPath	string	O	record path	/record
+CallIdEx	string	O	extern id	123456
+ReadFlag	int	O	Has it been read	0
+Name	string	O	peer name	东门 VTT

Response Example

```
totalCount=1000
found=500
records[0].RecNo=789
records[0].CreateTime=123456789
records[0].CallType=Incoming
records[0].EndState=Received
records[0].PeerNumber=501
...
```

13.2.2 Insert Announcement Record

This api is supported by VTO products.

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	Announcement Content	stringData
ExpirTime	string	O	the time when the Announcement expire, format: 2012-01-01%2012:00:00	2012-01-01%2012:00:00
IssueTime	string	O	Announcement issue time, format: 2012-01-01%2012:00:00	2012-01-01%2012:00:00
Title	string	O	title of the announcement	Anounce1
User	string	O	the number the Announcement issued to	101
State	int	O	the state of the Announcement. 0 init, 1 send , 2 overdue	1
ReadFlag	int	O	the read flag. 0 not read, 1 read.	0
BackgroundPicture	uint32	O	0: background1, 1: background2, 2: background3	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	the record index of the new record	232

Response Example

recno=232

13.2.3 Query Alarm Record

Find the AlarmRecord record. (This api is supported by video talk device.)

Request URL	http://<server>/cgi-bin/recordFinder.cgi?action=find			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
name	string	R	It is fixed to "AlarmRecord"	AlarmRecord
StartTime	string	O	The start time ,format : 2014-8-25%2000:01:32	2014-8-25%2000:01:32
EndTime	string	O	The end time, format: 2014-8-25%2000:02:32	2014-8-25%2000:02:32
count	int	O	the number of records to get, The record count, default 1024	500

Request Example

http://192.168.1.108/cgi-bin/recordFinder.cgi?action=find&name=AlarmRecord&StartTime=2014-8-25%

2000:02:32&EndTime=2014-8-25%2001:02:32&count=500

Response Params (key=value format in body)

Name	Type	R/O	Description	Example
totalCount	int	R	the record count which match condition	200
found	int	R	the record count to return	100
records	object[]	R	records	
+RecNo	int	O	record id	1234
+CreateTime	int	O	create time UTC	12345678
+Channel	int	O	channel	12
+SenseMethod	string	O	the range is: { "DoorMagnetism", "PassiveInfrared", "GasSensor", "SmokingSensor", "WaterSensor", "ActiveInfrared", "CallButton", "UrgencyButton", "Steal", "Perimeter", "PreventRemove", "DoorBell" }	DoorMagnetism
+RoomNumber	string	O	room number	101
+ReadFlag	int	O	Enumint{ 0: read 1: unread }	1
+Comment	string	O	remark	friend

Response Example

```
totalCount=1000
found=500
records[0].RecNo=789
records[0].CreateTime=123456789
records[0].Channel=0
records[0].SenseMethod=DoorMagnetism
records[0].RoomNumber=501
records[0].ReadFlag=0
records[0].Comment=Friend
...
```

13.3 SIP

The user needs to connect video intercom through CGI protocol, but do not know how to use this protocol to configure the device. This document provides the corresponding CGI protocol according to the configuration data format involved in the specific service.

13.3.1 [Config] SIP Configuration

Config Data Params

Name	Type	R/O	Description	Example
------	------	-----	-------------	---------

SIP	object	R	Configuration name	
+AccoutName	string	O	Account name	
+SIPServer	string	R	SIP server IP address or domain name	
+SIPServerPort	int	R	SIP server port number	
+OutboundProxy	string	O	Proxy server IP address or domain name	
+OutboundProxyID	string	O	Proxy server ID	
+OutboundProxyPort	Int	O	Proxy server port number	
+UserID	String	R	User account ID Generally a phone number	
+UserType	int	O	User type	
+AuthID	String	R	Authentication ID	8001
+AuthPassword	String	R	Authentication password	pass123456
+STUNServer	String	O	STUN (Simple Traversal of UDP over NATs) server IP address or domain name	
+RegisterRealm	String	R	Registration domain	
+RegExpiration	int	O	Registration interval, in seconds	
+LocalSIPPort	Int	R	Local SIP port 0-65535	
+LocalRTPPort	int	R	Local RTP port 0-65535	
+UnregisterOnReboot	bool	O	Restart to delete registration information. true: Delete false: Not delete	true
+DefaultCallNumber	string	O	Default call number	
+MediaDetail	Object	O	Media configuration	
++VideoStream	string	O	Video stream enumchar[32]{ "Main": Main stream "Extra1": Sub stream 1 "Extra2": Sub stream 2 "Extra3": Sub stream 3 }	Main
++AudioStream	string	O	Audio stream enumchar[32]{ "Main": Main stream "Extra1": Sub stream 1 "Extra2": Sub stream 2 "Extra3": Sub stream 3 }	Main
+RouteEnable	bool	O	Enable SIP cross-router or not.	

+Route	string	O	Router address, which can be IP address or domain name.	
+SIPServerLogin UserName	string	O	Username to log in to VTNC For intelligent building only	
+SIPServerLogin PWD	string	O	Password to log in to VTNC For intelligent building only	
+IsMainVTO	string	O	Whether the door station is a standby server For intelligent building only	
+SIPServerRedu ndancy	string	O	Standby server IP address For intelligent building only	
+SIPServerRedu ndancyUserNam e	string	O	Standby server login username For intelligent building only	
+SIPServerRedu ndancyPassWor d	string	O	Standby server login password For intelligent building only	
+AnalogNumber Start	string	O	The start number of the analog indoor monitor supported in the analog system For intelligent building only	
+AnalogNumber End	string	O	The end number of the analog indoor monitor supported in the analog system For intelligent building only	
+UserEnable	bool	R	Enable registration For intelligent building only true: Register to the SIP server. false: Not register to the SIP server.	true

Please refer to "4.2.1 Get and Set Configure" for configuration getting and setting. Specific examples are as follows :

Get Config Request Example

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

Get Config Response Example

```
table.SIP.AuthID=8001
table.SIP.AuthPassword=pass123456
table.SIP.IsMainVTO=0
table.SIP.LocalRTPPort=15000
table.SIP.LocalSIPPort=5060
table.SIP.OutboundProxy=192.168.1.111
table.SIP.OutboundProxyID=8000
table.SIP.OutboundProxyPort=5060
table.SIP.RegisterRealm=VDP
table.SIP.Route[0]=sip:10.30.1.2:5060;lr
table.SIP.RouteEnable=true
```

```

table.SIP.SIPServer=192.168.1.111
table.SIP.SIPServerID=8000
table.SIP.SIPServerLoginPWD=pass123456
table.SIP.SIPServerLoginUserName=admin
table.SIP.SIPServerPort=5060
table.SIP.SIPServerRedundancyPassWord=pass123456
table.SIP.SIPServerRedundancyUserName=admin
table.SIP.UserEnable=true
table.SIP.UserID=8001
table.SIP.UserType=2

```

Set Config Request Example

`http://10.0.0.8/cgi-bin/configManager.cgi?action=setConfig&SIP.SIPServer=192.168.1.108&SIP.SIPServerPort=5060&SIP.RegisterRealm=VDP&SIP.RouteEnable=true`

Set Config Response Example

OK

13.3.2 [Config] Registrar Configuration

Config Data Params				
Name	Type	R/O	Description	Example
Registrar	object[]	R	One-dimensional array. Each subscript represents a registration server.	
+RegistrarName	char[]	O	Registration server name, which must be unique. (Note: For VTH, use the IP address and port in the configuration to connect VTO.)	
+Enable	bool	O	Register to the registration server or not.	true
+ServerType	enumchar[32]	R	Server type, which is used only for SIP video talk, instead of VT video talk. Enumchar[32]{ "VTO" "H500" "VTNC" "ZYCOO" "ThirdParty" "3CXSystem" "Asterisk" }	VTO
+GeneralServerInfo	object	O		
++Address	char[40]	O	IP address	
++Port	uint	O	Registration port number	

++Password	char[64]	O	Password registered to the register server, which will be used when registration authentication is required.	
------------	----------	---	--	--

Please refer to "4.2.1 Get and Set Configure" for configuration getting and setting. Specific examples are as follows :

Get Config Request Example

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

Get Config Response Example

```
table.Registar[0].ServerType=VTO
table.Registar[0].Enable =true
table.Registar[0].RegistarName =xxx
table.Registar[0].GeneralServerInfo.Address=10.0.0.15
table.Registar[0].GeneralServerInfo.Port=61255
table.Registar[0].GeneralServerInfo.Password=123456
...
...
```

Set Config Request Example

```
http://10.0.0.8/cgi-bin/configManager.cgi?action=setConfig&Registar.ServerType=VTO
```

Set Config Response Example

```
OK
```

13.4 Room Number Database Management

13.4.1 Adding Room Number

Request URL	http://<server>/cgi-bin/recordUpdater.cgi?action=insert			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
name	char[]	R	it is fixed to VideoTalkContact	VideoTalkContact
CreateTime	uint	O	Start time UTC seconds, read only	
FirstName	char[]	O	First name	
FamilyName	char[]	O	Last name	
VTShortNumber	char[]	R	Short number for video intercom	123
VTMiddleNumber	char[]	O	Middle number for video intercom	
VTLongNumber	char[]	O	Long number for video intercom or serial number of analog indoor monitor	330103001101010151
VTNetAddress	char[]	O	Network address for video intercom	
MacAdress	char[]	O	MAC address	
VTOPosition	char[]	O	Door number linked with indoor monitor	

VTSlaveBindMode	enumint	O	Mode when accessing to the analog indoor monitor for video talk Enumint{ 0: SubAddress 1: SubID+SubPort }	
VTSlaveld	uint32	O	Allocator address when accessing to the analog indoor monitor for video talk (Change string to uint32 for consistency)	
VTSlavePort	uint32	O	Allocator port when accessing to the analog indoor monitor for video talk (Change string to uint32 for consistency)	
VTSlaveAddress	char[]	O	Address of the analog indoor monitor for video talk	
NickName	char[]	O	Nickname	
Notes	char[]	O	Notes	
Type	enumchar[32]	O	User type Enumchar[32]{ "VTH": Indoor monitor "VTO": Door station }	VTH
RegisterType	enumchar[32]	R	Registration method Enumchar[32]{ "public" "local" }	public
VTHPassword	char[]	R	Registration password	pass123456
VTOBuilding	char[]	O	Building number	
VTOUnit	char[]	O	Unit number	
GroupNumber	char[]	O	Group	
Channel	uint32	O	Channel number, based on which the mobile phone subscribes to the call notification message.	
Floors	char[256] [4]	O	Floor number (lift control requirements), with up to 256 characters	
LiftControlByVTH	bool	O	Lift control is triggered by the indoor monitor opening the door.	false

Request Example

<http://<server>/cgi-bin/recordUpdater.cgi?action=insert&name=VideoTalkContact&VTShortNumber=101&RegisterType=public&VTHPassword=pass123456>

Response Params (key=value format in body)

Name	Type	R/O	Description	Example
recno	int	R	Record ID of successfully creation	12345

Response Example

recno=12345

13.4.2 Getting Records by Video Talk short number

Request URL	http://<server>/cgi-bin/recordFinder.cgi?action=find			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
name	char[]	R	It is fixed to VideoTalkContact	VideoTalkContact
cout	int	O	Maximum record number of search result	10
condition	object	R	search condition	
+VTShortNum ber	char[]	R	video talk short number	101
Request Example				
http://<server>/cgi-bin/recordFinder.cgi?action=find&name=VideoTalkContact&count=10&condition.VTS hortNumber=101				

Response Params (key=value format in body)				
Name	Type	R/O	Description	Example
totalCount	int	O	total record number of search result	10
found	int	O	Number of records returned	1
records	object[]	R	Records returned	
+RecNo	int	R	Record ID	12345
+CreateTime	uint	O	Start time UTC seconds, read only	
+FirstName	char[]	O	First name	dafei
+FamilyName	char[]	O	Last name	liu
+VTShortNum ber	char[]	R	Short number for video intercom	123
+VTMiddleNu mber	char[]	O	Middle number for video intercom	
+VTLongNumb er	char[]	O	Long number for video intercom or serial number of VTH analog indoor monitor	33010300110101015 1
+VTNetAddres s	char[]	O	Network address for video intercom	
+MacAdress	char[]	O	MAC address	
+VTOPosition	char[]	O	Door number linked with indoor monitor	
+VTSlaveBind Mode	enumint	O	Mode when accessing to the analog indoor monitor for video talk Enumint{ 0: SubAddress 1: SubID+SubPort }	
+VTSlaveld	uint32	O	Allocator address when accessing to the analog indoor monitor for video talk (Change string to uint32 for consistency)	
+VTSlavePort	uint32	O	Allocator port when accessing to the analog indoor monitor for video talk	

			(Change string to uint32 for consistency)	
+VTSIaveAddress	char[]	O	Address of the analog indoor monitor for video talk	
+NickName	char[]	O	Nickname	
+Notes	char[]	O	Notes	
+Type	enumchar[32]	O	User type Enumchar[32]{ "VTH": Indoor monitor "VTO": Door station }	
+RegisterType	enumchar[32]	R	Registration method Enumchar[32]{ "public" "local" }	
+VTHPassword	char[]	R	Registration password	123456
+VTOBuilding	char[]	O	Building number	
+VTOUnit	char[]	O	Unit number	
+GroupNumber	char[]	O	Group	
+LiftControlByVTH	bool	O	Lift control is triggered by the indoor monitor opening the door.	false

Response Example

totalCount=10
 found=1
 records[0].RecNo=12345
 records[0].CreateTime=123456789
 records[0].FirstName=San
 records[0].FamilyName=Zhang
 records[0].VTShortNumber=0101
 records[0].VTMiddleNumber=11010101
 records[0].VTLONGNumber=330103001101010151
 records[0].VTNetAddress=127.0.0.1
 records[0].MacAdress=0A:3E:FF:2A:50:41
 records[0].VTOPosition=01018001
 records[0].VTSIaveBindMode=0
 records[0].VTSIaveId=1258421
 records[0].VTSIavePort=1258421
 records[0].VTSIaveAddress=04:b3:01:f7
 records[0].NickName=Nick
 records[0].Notes=Friend
 records[0].Type=VTH
 records[0].RegisterType=public
 records[0].VTHPassword=123456
 records[0].VTOBuilding=01
 records[0].VTOUnit=01

```

records[0].GroupNumber=301
records[0].Channel=1
records[0].Floors[0]=1
records[0].LiftControlByVTH=true

```

13.4.3 Getting Records by recno

Request URL	http://<server>/cgi-bin/recordUpdater.cgi?action=get		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
name	char[]	R	It is fixed to VideoTalkContact
recno	int	R	record id
Request Example			
http://<server>/cgi-bin/recordUpdater.cgi?action=get&name=VideoTalkContact&recno=12345			

Response Params (key=value format in body)			
Name	Type	R/O	Description
record	object	R	record
+RecNo	int	R	Record ID
+CreateTime	uint	O	Start time UTC seconds, read only
+FirstName	char[]	O	First name
+FamilyName	char[]	O	Last name
+VTShortNumber	char[]	R	Short number for video intercom
+VTMiddleNumber	char[]	O	Middle number for video intercom
+VTLONGNumber	char[]	O	Long number for video intercom or serial number of VTH analog indoor monitor
+VTNetAddress	char[]	O	Network address for video intercom
+MacAdress	char[]	O	MAC address
+VTOPosition	char[]	O	Door number linked with indoor monitor
+VTSlaveBindMode	enumint	O	Mode when accessing to the analog indoor monitor for video talk Enumint{ 0: SubAddress 1: SubID+SubPort }
+VTSlaveld	uint32	O	Allocator address when accessing to the analog indoor monitor for video talk (Change string to uint32 for consistency)
+VTSlavePort	uint32	O	Allocator port when accessing to the analog indoor monitor for video talk (Change string to uint32 for consistency)

+VTSIveAddress	char[]	O	Address of the analog indoor monitor for video talk	
+NickName	char[]	O	Nickname	
+Notes	char[]	O	Notes	
+Type	enumchar[32]	O	User type Enumchar[32]{ "VTH": Indoor monitor "VTO": Door station }	
+RegisterType	enumchar[32]	R	Registration method Enumchar[32]{ "public" "local" }	
+VTHPassword	char[]	R	Registration password	
+VTOBuilding	char[]	O	Building number	
+VTOUnit	char[]	O	Unit number	
+GroupNumber	char[]	O	Group	
+LiftControlByVTH	bool	O	Lift control is triggered by the indoor monitor opening the door.	false

Response Example

record.RecNo=12345
 record.CreateTime=123456789
 record.FirstName=三
 record.FamilyName=张
 record.VTShortNumber=0101
 record.VTMiddleNumber=11010101
 record.VTLongNumber=330103001101010151
 record.VTNetAddress=127.0.0.1
 record.MacAdress=0A:3E:FF:2A:50:41
 record.VTOPosition=01018001
 record.VTSIveBindMode=0
 record.VTSIveId=1258421
 record.VTSIvePort=1258421
 record.VTSIveAddress=04:b3:01:f7
 record.NickName=Nick
 record.Notes=Friend
 record.Type=VTH
 record.RegisterType=public
 record.VTHPassword=123456
 record.VTOBuilding=01
 record.VTOUnit=01
 record.GroupNumber=301
 record.Channel=1
 record.Floors[0]=1

record.LiftControlByVTH=true

13.4.4 Updating Room Number Records

Request URL	http://<server>/cgi-bin/recordUpdater.cgi?action=update		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
name	char[]	R	It is fixed to VideoTalkContact
recno	int	R	Record ID
CreateTime	uint	O	Start time UTC seconds, read only
FirstName	char[]	O	First name
FamilyName	char[]	O	Last name
VTShortNumber	char[]	R	Short number for video intercom
VTMiddleNumber	char[]	O	Middle number for video intercom
VTLongNumber	char[]	O	Long number for video intercom or serial number of VTH analog indoor monitor
VTNetAddress	char[]	O	Network address for video intercom
MacAdress	char[]	O	MAC address
VTOPosition	char[]	O	Door number linked with indoor monitor
VTSlaveBindMode	enumint	O	Mode when accessing to the analog indoor monitor for video talk Enumint{ 0: SubAddress 1: SubID+SubPort }
VTSlaveId	uint32	O	Allocator address when accessing to the analog indoor monitor for video talk (Change string to uint32 for consistency)
VTSlavePort	uint32	O	Allocator port when accessing to the analog indoor monitor for video talk (Change string to uint32 for consistency)
VTSlaveAddress	string	O	Address of the analog indoor monitor for video talk
NickName	char[]	O	Nickname
Notes	char[]	O	Notes
Type	enumchar[32]	O	User type Enumchar[32]{ "VTH": Indoor monitor "VTO": Door station }
RegisterType	enumchar[32]	R	Registration method Enumchar[32]{ "public"
			public

			"local" }	
VTHPassword	char[]	R	Registration password	pass123456
VTOBuilding	char[]	O	Building number	
VTOUnit	char[]	O	Unit number	
GroupNumber	char[]	O	Group	
Channel	uint32	O	Channel number, based on which the mobile phone subscribes to the call notification message.	
Floors	char[256] [4]	O	Floor number (lift control requirements), with up to 256 characters	
LiftControlByV TH	bool	O	Lift control is triggered by the indoor monitor opening the door.	false

Request Example

http://<server>/cgi-bin/recordUpdater.cgi?action=update&name=VideoTalkContact&recno=12345&First Name=Green&FamilyName=Jim&VTHPassword=654321

Response Params (OK in body)

Response Example

OK

13.4.5 Deleting Records by recno

Request URL	http://<server>/cgi-bin/recordUpdater.cgi?action=remove			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
name	char[]	R	It is fixed to VideoTalkContact	VideoTalkContact
recno	int	R	record id	12345

Request Example

http://<server>/cgi-bin/recordUpdater.cgi?action=remove&name=VideoTalkContact&recno=12345

Response Params (OK in body)

Response Example

OK

13.4.6 Clearing All Room Numbers

Request URL	http://<server>/cgi-bin/recordUpdater.cgi?action=clear			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
name	char[]	R	Database name, which must be VideoTalkContact.	VideoTalkContact

Request Example

http://<server>/cgi-bin/recordUpdater.cgi?action=clear&name=VideoTalkContact

Response Params (OK in body)

| **Response Example** |
| OK |

13.4.7 Getting Total Quantity of Room Number

Request URL	http://<server>/cgi-bin/recordFinder.cgi?action=getQuerySize		
--------------------	--	--	--

| **Method** | GET | | |
| **Request Params** (key=value format in URL) | | | |

Name	Type	R/O	Description	Example
name	char[]	R	Database name, which must be VideoTalkContact.	VideoTalkContact

| **Request Example** |
| http://<server>/cgi-bin/recordFinder.cgi?action=getQuerySize&name=VideoTalkContact |

Response Params (key=value format in body)

| | | | | | | |-------------|-------------|------------|--------------------|----------------| | Name | Type | R/O | Description | Example | |-------------|-------------|------------|--------------------|----------------| |
| | | | | | | |-------|-----|---|-------------------------|-----| | count | int | R | Total number of records | 200 | |-------|-----|---|-------------------------|-----| |
| **Response Example** |
| count=200 |

13.5 ElevatorFloorCounter

13.5.1 Set Elevator Floor Info

Request URL	http://<server>/cgi-bin/ElevatorFloorCounter.cgi?action=setElevatorFloorInfo			
--------------------	--	--	--	--

| **Method** | GET | | | |
| **Request Params** (key=value format in URL) | | | | |

Name	Type	R/O	Description	Example
Channel	int	R	ChannelNo, start from 1, default 1	1
CheckBaseFloor	char[]	R	Calibrate base floor, unit floor, It's not necessarily numbers	5

| **Request Example** |
| http://192.168.1.108/cgi-bin/ElevatorFloorCounter.cgi?action=setElevatorFloorInfo&CheckBaseFloor=5&Channel=1 |

Response Params (OK in body)

| **Response Example** |
| OK |

13.5.2 Get Elevator WorkInfo

Request URL	http://<server>/cgi-bin/ElevatorFloorCounter.cgi?action=getElevatorWorkInfo			
--------------------	---	--	--	--

| **Method** | GET | | | |

Request Params (key=value format in URL)

Name	Type	R/O	Description	Example
Channel	int	R	ChannelNo, start from 1, default 1	1

Request Example

```
http://192.168.1.108/cgi-bin/ ElevatorFloorCounter.cgi?action=getElevatorWorkInfo&Channel=1
```

Response Params (key=value format in body)

Name	Type	R/O	Description	Example
table	object	R	Current elevator operation information	
+ElevatorFloor	int	O	floor	20
+ElevatorWorkState	int	O	state	1
+ElevatorWorkSpeed	int	O	speed	20
+IsCheckFloor	bool	O	whether to calibrate	true

Response Example

```
table.ElevatorFloor=20
table.ElevatorWorkState=1
table.ElevatorWorkSpeed=20
table.IsCheckFloor=true
```

13.5.3 Get Capability

Request URL	http://<server>/cgi-bin/ElevatorFloorCounter.cgi?action=getCaps			
--------------------	---	--	--	--

Method	GET			
---------------	-----	--	--	--

Request Params (key=value format in URL)

Name	Type	R/O	Description	Example
Channel	int	R	ChannelNo, start from 1, default 1	1

Request Example

```
http://192.168.1.108/cgi-bin/ ElevatorFloorCounter.cgi?action=getCaps&Channel=1
```

Response Params (key=value format in body)

Name	Type	R/O	Description	Example
caps	object	R	elevator floor capacities	
+IsElevatorFloorCounter	bool	R	Whether the floor configuration function is supported	true

Response Example

```
caps.IsElevatorFloorCounter=true
```

14.1 File Finder

14.1.1 Create a File Finder

Example:

File 1:

```
items[0].Channel =1
items[0].StartTime =2011-1-1 12:00:00
items[0].EndTime =2011-1-1 13:00:00
items[0].Events[0]=AlarmLocal
items[0].VideoStream=Main
items[0].Length =790
items[0].Duration = 3600
```

File 2:

```
items[0].Channel =1
items[0].StartTime =2011-1-1 13:00:00
items[0].EndTime =2011-1-1 14:00:00
items[0].Events[0]=AlarmLocal
items[0].VideoStream=Main
items[0].Length =790
items[0].Duration = 3600
```

file1 and file2 will be combined to file3

File 3:

```
items[0].Channel =1
items[0].StartTime =2011-1-1 12:00:00
items[0].EndTime =2011-1-1 14:00:00
items[0].Events[0]=AlarmLocal
items[0].VideoStream=Main
items[0].Length =1580
items[0].Duration = 7200
```

HTTP API V3.35 - Intelbras

Request URL	http://<server>/cgi-bin/FileFindHelper.cgi?action=startFind			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
condition	object	R	query condition	
+channel	uint32	R	video channel index which starts from 1.	1
+startTime	char[20]	R	start time when recording. format:	"2010-05-25 00:00:00"

			yyyy-mm-dd hh:mm:ss	
+endTime	char[20]	R	end time when recording. format: yyyy-mm-dd hh:mm:ss	"2010-05-25 23:59:59"
+streamType	char[32]	R	which video stream type you want to find. The range of stream is {"Main", "Extra1", "Extra2", "Extra3"}.	"Main"
+flags	char[64][32]	O	which flags of the file you want to find. It is an array. The index starts from 0. The range of flag is {"Timing", "Marked", "Event", "Restrict"}. If omitted, find files with all the flags.	["Timing", "Event", "Marked", ...]
+events	char[64][32]	O	by which event the record file is triggered. It is an array. The index starts from 0. The range of event is {"AlarmLocal", "VideoMotion", "VideoLoss"}. This condition can be omitted. If omitted, find files of all the events.	["AlarmLocal", "VideoMotion"]
combineMode	object	O	combine info	
+granularity	uint32	O	by which granularity to combine files	16
+types	char[][]	O	which types of the file you want to combined. It is an array. The index starts from 0. The range of combine type is {"AlarmLocal", "VideoMotion", "Timing", "VideoLoss"}. This condition can be omitted. If omitted, file will not be combined.	["AlarmLocal", "VideoMotion"]

Request Example

http://172.23.1.66/cgi-bin/fileFindHelper.cgi?action=startFind&condition.channel=1&condition.startTime=2014-1-1%2012:00:00&condition.endTime=2015-1-10%2012:00:00&condition.streamType>Main&condition.flags[0]=Event&condition.events[0]=AlarmLocal&condition.events[1]=VideoMotion&combineMode.granularity=16&combineMode.types[0]=AlarmLocal&combineMode.types[0]=VideoMotion

Response Params (key=value format in body)

Name	Type	R/O	Description	Example
result	int	R	The find Id	08137

Response Example

result=08137

14.1.2 Create a Motion File Finder

Request URL	http://<server>/cgi-bin/FileFindHelper.cgi?action=startMotionFind
-------------	---

Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
condition	object	R	query condition	
+channel	uint32	R	video channel index which starts from 1.	1
+startTime	char[20]	R	start time when recording. format: yyyy-mm-dd hh:mm:ss	"2010-05-25 00:00:00"
+endTime	char[20]	R	end time when recording. format: yyyy-mm-dd hh:mm:ss	"2010-05-25 23:59:59"
+streamType	char[32]	R	which video stream type you want to find. The range of stream is {"Main", "Extra1", "Extra2", "Extra3"}.	"Main"
motionRegion	object	R		
+senseLevel	uint32	R	the motion sensitive level, range is 0–6, 0 represent all level	16
+rects	int[4]	O	relative coordinates, rect0 and rect2 range is 0—21, rect1 and rect3 range is 0—17. {0,0,0,0} top—left, {21,0,0,0} top—right, {0,17,0,0} bottom—left, {21,17,0,0} bottom—right	["AlarmLocal", "VideoMotion"]

Request Example

http://172.23.1.66/cgi-bin/fileFindHelper.cgi?action=startMotionFind&condition.channel=1&condition.startTime=2014-1-1%2012:00:00&condition.endTime=2015-1-10%2012:00:00&condition.streamType>Main&condition.flags[0]=Event&condition.events[0]=AlarmLocal&condition.events[1]=VideoMotio&motionRegion.senseLevel=1&motionRegion.rects[1][0]=0&motionRegion.rects[1][1]=0&motionRegion.rects[1][2]=21&motionRegion.rects[1][3]=17

Response Params (key=value format in body)				
Name	Type	R/O	Description	Example
result	int	R	The find Id	08137
Response Example				
result=08137				

14.1.3 Get the File Information Found by the Finder

Request URL	http://<server>/cgi-bin/FileFindHelper.cgi?action=findNext			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
findId	int	R	The find Id is created by API Create a file finder or API Create a motion file finder. Must create a finder before finding files.	08137
count	int	R	count to show	100

Request Example

http://192.168.1.108/cgi-bin/FileFindHelper.cgi?action=findNext&findId=08137&count=100
--

Response Params (key=value format in body)

Name	Type	R/O	Description	Example
found	int	R	found items	1
items	object[]	R	results	
+channel	int	O	video channel index which starts from 1.	1
+startTime	char[20]	O	start time when recording. format: yyyy-mm-dd hh:mm:ss	"2010-05-25 00:00:00"
+endTime	char[20]	O	end time when recording. format: yyyy-mm-dd hh:mm:ss	"2010-05-25 23:59:59"
+fileType	char[]	O	file type, range: {"dav", "jpg" }	"dav"
+events	char[][]	O	event type list	["AlarmLocal"]
+streamType	char[32]	O	The range of stream is {"Main", "Extra1", "Extra2", "Extra3"}.	"Main"
+length	int	O	record length,unit:Byte	790
+duration	int	O	record duration, unit: s	3600

Response Example

found=1
items[0]. channel =1
items[0]. startTime =2011-1-1 12:00:00
items[0]. endTime =2011-1-1 13:00:00
items[0]. fileType =dav
items[0]. events[0]=AlarmLocal
items[0]. streamType=Main
items[0]. length =790
items[0]. duration = 3600

14.1.4 Stop the Finder

Request URL	http://<server>/cgi-bin/FileFindHelper.cgi?action=stopFind			
Method	GET			
Request Params (none)				
Name	Type	R/O	Description	Example
findId	int	R	The find Id is created by API Create a file finder or API Create a motion file finder. Must create a finder before finding files.	08137

Request Example

http://192.168.1.108/cgi-bin/FileFindHelper.cgi?action=stopFind&findId =08137

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

OK

14.1.5 Get Bound Files

Request URL	http://<server>/cgi-bin/FileFindHelper.cgi?action=getBoundFile			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
condition	object	R	query condition	
+channel	uint32	R	video channel index which starts from 1.	1
+startTime	char[20]	R	start time when recording. format: yyyy-mm-dd hh:mm:ss	"2010-05-25 00:00:00"
+endTime	char[20]	R	end time when recording. format: yyyy-mm-dd hh:mm:ss	"2010-05-25 23:59:59"
+streamType	char[32]	R	which video stream type you want to find. The range of stream is {"Main", "Extra1", "Extra2", "Extra3"}.	"Main"
+flags	char[64][32]	O	which flags of the file you want to find. It is an array. The index starts from 0. The range of flag is {"Timing", "Marked", "Event", "Restrict"}. If omitted, find files with all the flags.	["Timing", "Event", "Marked", ...]
+events	char[64][]	O	event type list	["AlarmLocal"]
Request Example				
http://<server>/cgi-bin/FileFindHelper.cgi?action=getBoundFile&condition.channel=1&condition.startTime=2014-1-1%2012:00:00&condition.endTime=2015-1-10%2012:00:00&condition.streamType>Main&condition.flags[0]=Timing				

Response Params (key=value format in body)				
Name	Type	R/O	Description	Example
found	int	R	found items	2
items	object[]	R	results	
+channel	int	O	video channel index which starts from 1.	1
+startTime	char[20]	O	start time when recording. format: yyyy-mm-dd hh:mm:ss	"2010-05-25 00:00:00"
+endTime	char[20]	O	end time when recording. format: yyyy-mm-dd hh:mm:ss	"2010-05-25 23:59:59"
+flags	char[64][32]	O	which flags of the file you want to find. It is an array. The index starts from 0. The range of flag is {"Timing", "Marked", "Event", "Restrict"}. If omitted, find files with all the flags.	["Timing", "Event", "Marked", ...]

+events	char[64][]	O	event type list	["AlarmLocal"]
+streamType	char[32]	O	video stream type. The range of stream is {"Main", "Extra1", "Extra2", "Extra3"}.	"Main"
+length	int	O	record length,unit:Byte	790
+duration	int	O	record duration, unit: s	3600

Response Example

```
found=2
items[0].channel =1
items[0].startTime =2011-1-1 12:00:00
items[0].endTime =2011-1-1 13:00:00
items[0].flags [0]= Timing
items[0].streamType=Main
items[0].length =790
items[0].duration = 3600
items[1].channel =1
items[1].startTime =2011-1-1 13:00:00
items[1].endTime =2011-1-1 14:00:00
items[1].events[0]= Timing
items[1].streamType=Main
items[1].length =790
items[1].duration = 3600
```

14.2 BandLimit

14.2.1 Get Bandwidth Limit State

Request URL	http://<server>/cgi-bin/BandLimit.cgi?action=getLimitState			
Method	GET			
Request Params (key=value format in url)				
Name	Type	R/O	Description	Example
Request Example				
http://192.168.1.108/cgi-bin/bandLimit.cgi?action=getLimitState				

Response Params (key=value format in body)				
Name	Type	R/O	Description	Example
limit	bool	R	true/false	true

Response Example

```
limit=true
```

14.3 Record Files Protection

14.3.1 Add Protection

Request URL	http://<server>/cgi-bin/FileManager.cgi?action=addConditionList			
Method	GET			
Request Params (key=value format in url)				
Name	Type	R/O	Description	Example
condition	object	R	query condition	
+Channel	uint32[]	R	Video channel index which starts from 1.	1
+startTime	char[20]	R	The time format is "Y-M-D H-m-S", example 2011-7-3%2021:02:32	"2010-05-25 00:00:00"
+endTime	char[20]	R	The time format is "Y-M-D H-m-S"	"2010-05-25 23:59:59"
+Types	char[][]	R	An array. The range is { "RecordProtect", "RecordRestrict" }.	["RecordProtect", "RecordRestrict"]
Request Example				http://192.168.1.108/cgi-bin/FileManager.cgi?action=addConditionList&condition.Types[0]=RecordRestrict&condition.Types[1]=RecordProtect&condition.StartTime=2014-7-3%2021:02:32&condition.EndTime=2014-7-3%2023:02:32&condition.Channel[0]=1&condition.Channel[1]=3

Response Params (OK in body)
Response Example
OK

14.3.2 Cancel Protection

Request URL	http://<server>/cgi-bin/FileManager.cgi?action=cancelConditionList			
Method	GET			
Request Params (key=value format in url)				
Name	Type	R/O	Description	Example
condition	object	R	query condition	
+Channel	uint32[]	R	Video channel index which starts from 1.	1
+startTime	char[20]	R	The time format is "Y-M-D H-m-S", example 2011-7-3%2021:02:32	"2010-05-25 00:00:00"
+endTime	char[20]	R	The time format is "Y-M-D H-m-S"	"2010-05-25 23:59:59"
+Types	char[][]	R	An array. The range is { "RecordProtect", "RecordRestrict" }.	["RecordProtect", "RecordRestrict"]
Request Example				http://192.168.1.108/cgi-bin/FileManager.cgi?action=cancelConditionList&condition.Types[0]=RecordRestrict&condition.Types[1]=RecordProtect&condition.StartTime=2014-7-3%2021:02:32&condition.EndTime=2014-7-3%2023:02:32&condition.Channel[0]=1&condition.Channel[1]=3

Response Params (OK in body)
Response Example

OK

14.3.3 Remove Protection

Request URL	http://<server>/cgi-bin/FileManager.cgi?action=removeConditionList			
Method	GET			
Request Params (key=value format in url)				
Name	Type	R/O	Description	Example
condition	object	R	query condition	
+Channel	uint32[]	R	Video channel index which starts from 1.	1
+startTime	char[20]	R	The time format is "Y-M-D H-m-S", example 2011-7-3%2021:02:32	"2010-05-25 00:00:00"
+endTime	char[20]	R	The time format is "Y-M-D H-m-S"	"2010-05-25 23:59:59"
+Types	char[][]	R	An array. The range is { "RecordProtect", "RecordRestrict" }.	["RecordProtect", "RecordRestrict"]
Request Example				
http://192.168.1.108/cgi-bin/FileManager.cgi?action=removeConditionList&condition.Types[0]=RecordRestrict&condition.Types[1]=RecordProtect&condition.StartTime=2014-7-3%2021:02:32&condition.EndTime=2014-7-3%2023:02:32&condition.Channel[0]=0&condition.Channel[1]=3				

Response Params (OK in body)
Response Example
OK

14.3.4 DownloadFile

download file

Request URL	http://<server>/cgi-bin/FileManager.cgi?action=downloadFile			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
fileName	string	R	filename or path	download.jpg
Request Example				
http://192.168.1.108/cgi-bin/FileManager.cgi?action=downloadFile&fileName=download.jpg				

Response Params (multipart binary data)				
Name	Type	R/O	Description	Example
Response Example				
HTTP/1.1 200 OK Content-type: text/plain; charset=utf-8 Connection: close Set-Cookie: secure; HttpOnly CONTENT-LENGTH: <length>				

<Binary Data>

14.3.5 UploadFile

Upload radar photos

Request URL	http://<server>/cgi-bin/FileManager.cgi?action=uploadFile&fileName=< FileName >&Path=< Path >		
Method	POST		
Request Params (key=value format in url)			
Name	Type	R/O	Description
fileName	string	R	File name, only the types bmp and jpg are available for radars.
Path	string	R	The file path to store
Request Example			
<pre>POST http://192.168.1.108/cgi-bin/FileManager.cgi?action=uploadFile&fileName=xxxxxx.bmp&fileLength=124879 HTTP/1.1 Host: 192.168.1.108 Connection: keep-alive Content-Type: multipart/form-data;boundary=-----8655433224198 Content-Length: xxxxxxxx -----8655433224198 Content-Disposition:form-data;name="upload"; filename="xxxxxx.bmp" Content-Type: image/jpeg or application/x-MS-bmp photo data....</pre>			
-----8655433224198--			

Response Params (OK in body)				
Parameter	Type	Required	Description	Example
Response Example				
OK				

14.3.6 List all elements in the specified directory

Request URL	http://<server>/cgi-bin/api/FileManager/list		
Method	POST		
Request Params (JSON format in body)			
Name	Type	R/O	Description
path	char[260]	R	To get the path, there will be internal security verification. For files uploaded externally, such as obtaining the SFTP directory, you can obtain the NAS configuration and find the corresponding

		directory for query.	
Request Example			
{			
	"path": "/mnt/dvr/sda0/2010/8/11/dav"		
}			

Response Params (JSON format in body)				
Name	Type	R/O	Description	Example
elementInfo	object[]	R	File element information	
+type	enumchar[16]	O	File element type enumchar[16]{ "File" "Directory" }	"File"
+file	object	O	File info (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	Creation time	"2010-4-15 9:58:32"
++ModifyTime	char[20]	O	Modification time	"2010-4-15 9:58:32"
++Size	double	O	File size, the decimal part is meaningless. Unit: byte	1873.0
++path	char[260]	O	Absolute path	"/mnt/dvr/sda0/2010/8/11/dav.jpg"
++Desc	char[128]	O	File Custom Description	"xxxxxx"
+directory	object	O	directory information ,it works when Type="Directory"	
++CreateTime	char[20]	O	Creation time, formatted as "Y-M-D H-m-S"	"2010-4-15 9:58:32"
++path	char[260]	O	Absolute path	"/mnt/dvr/sda0/2010/8/11/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": "/mnt/dvr/sda0/2010/8/11/dav.jpg",
"Desc": "xxxxxx"
},
"directory": {
"CreateTime": "2010-4-15 9:58:32",
"path": "/mnt/dvr/sda0/2010/8/11/dav"
}

```

},...{}]
}

```

14.3.7 DownloadFile By Condition

ITC custom use. Search for downloaded files according to condition, no other response, only the found file Packet that meets the condition.

Request URL	http://<server>/cgi-bin/api/FileManager/findDownload			
Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
Condition	object	R		
+Channel	int	R	Channel number, starting from 0, -1 indicates querying all channels	-1
+StartTime	char[20]	R	StartTime	"2010-05-25 12:05:00"
+EndTime	char[20]	R	EndTime	"2010-05-25 12:10:00"
+Types	char[][8]	O	List files of type It is an array that can query multiple file types at once. Without the Types field, it means querying any type. "Jpg" refers to images in general	["jpg"]
+DB	object	O	Database query criteria	
++TrafficCar	object	O	Traffic vehicle query, when adding new protocols in the future, multiple query criteria are combined with AND	
+++PlateNumber	char[2][16]	O	Query condition: The last four digits of the license plate are 8888 To maintain format neutrality and support * as a wildcard, SQL class databases will replace * with % during implementation 'Unknown' indicates that the query is unknown	["==","*8888"]
+++Event	char[][16]	O	Event Type List It is an array that can query multiple event types at once. Without	["TrafficGate"]

		an Event field, it means querying the TrafficGate type.	
Request Example			
{			
"Condition": {			
"Channel": -1,			
"StartTime": "2010-05-25 12:05:00",			
"EndTime": "2010-05-25 12:10:00",			
"Types": ["jpg"],			
"DB": {			
"TrafficCar": {			
"PlateNumber": ["==", "*8888"],			
"Event": ["TrafficGate"]			
}			
}			
}			
}			

Response Params (multipart, binary in body)

Response Example

HTTP/1.1 200 OK
Server: Device/1.0
Connection: keep-alive
Content-Type: multipart/ form-data; boundary=<boundary>

--<boundary>
Content-Type: application/octet-stream
Content-Disposition: form-data; name="name of file1"

<bytes of file1>
--<boundary>
Content-Type: application/octet-stream
Content-Disposition: form-data; name="name of file2"

<bytes of file2>
--<boundary>
.....

14.4 Daylight

14.4.1 Get Daylight

Request URL	http://<server>/cgi-bin/global.cgi?action=getDST			
Method	GET			
Request Params (none)				
Name	Type	R/O	Description	Example
Request Example				
http://192.168.1.108/cgi-bin/global.cgi?action=getDST				

Response Params (key=value format in body)				
Name	Type	R/O	Description	Example
result	int	R	1/0, yes or not in daylight saving time	1
Response Example				
result = 1				

15.1 Discover Devices

15.1.1 Discover Devices on Internet

Request URL	<a href="http://<server>/cgi-bin/deviceDiscovery.cgi?action=attach">http://<server>/cgi-bin/deviceDiscovery.cgi?action=attach			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
DeviceClass	char[64]	R	range: {VTO, VTH, VTT, VTS, VTNC, SHG}	VTO
Request Example				
http://192.168.1.108/cgi-bin/deviceDiscovery.cgi?action=attach&DeviceClass=VTO				

Response Params (key=value format in body)				
Name	Type	R/O	Description	Example
deviceInfo	object[]	O	device info	
+MachineName	char[64]	O	name	"IPC1"
+DeviceClass	char[64]	O	class	"IPC"
+DeviceType	char[64]	O	type	"IPC-HF3300"
+SerialNo	char[64]	O	serial number	"123456789"
+Version	char[64]	O	software version	"1.1.1.1"
+Mac	char[18]	O	MAC	"00:10:5C:F2:1C:B4"
+VideoInputChannels	uint	O	video input channels	32
+RemoteVideoInputChannels	uint	O	remote video input channels	32
+VideoOutputChannels	uint	O	video output channels	32
+AlarmInputChannels	uint	O	alarm input channels	16
+AlarmOutputChannels	uint	O	alarm output channels	16
+Port	uint16	O	port	37777
+Vendor	char[64]	O	vendor info	"Private"
+IPv4Address	object	O	Ipv4 address info	
++IPAddress	char[40]	O	IP	"192.168.1.112"
++SubnetMask	char[40]	O	mask	"255.255.255.0"
++DefaultGateway	char[40]	O	default gateway	"192.168.1.1"
++DhcpEnable	bool	O	enable/disable DHCP	true
+IPv6Address	object	O	Ipv6 address info	
++DhcpEnable	bool	O	enable/disable DHCP	true
++IPAddress	char[40]	O	IP	"2001::123:123/64"

++DefaultGateway	char[40]	O	default gateway	"2001::123:1"
+HttpPort	uint	O	http service port	80
Response Example				
deviceInfo[index].AlarmInputChannels=8				
deviceInfo[index].AlarmOutputChannels=0				
deviceInfo[index].DeviceClass=VTO				
deviceInfo[index].DeviceType=VTO2000A				
deviceInfo[index].HttpPort=80				
deviceInfo[index].IPv4Address.DefaultGateway=172.12.0.1				
deviceInfo[index].IPv4Address.DhcpEnable=false				
deviceInfo[index].IPv4Address.IPAddress=172.12.7.102				
deviceInfo[index].IPv4Address.SubnetMask=255.255.0.0				
deviceInfo[index].IPv6Address.DefaultGateway=2008::1				
deviceInfo[index].IPv6Address.DhcpEnable=false				
deviceInfo[index].IPv6Address.IPAddress=2008::6/112				
deviceInfo[index].Mac=00:01:5b:01:44:77				
deviceInfo[index].MachineName=YZZ4DZ008D00031				
deviceInfo[index].Port=37777				
deviceInfo[index].RemoteVideoInputChannels=0				
deviceInfo[index].SerialNo=YZZ4DZ008D00031				
deviceInfo[index].Vendor=Multi				
deviceInfo[index].Version=1.200.0.0				
deviceInfo[index].VideoInputChannels=1				
deviceInfo[index].VideoOutputChannels=16				

15.2 Open Platform

15.2.1 Application Start and Stop

- Start Application

Request URL	http://<server>/cgi-bin/installManager.cgi?action=start			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
appname	char[128]	R	the application name	"FaceDemo"
appid	uint	O	the application id	1234
Request Example				
http://192.168.1.108/cgi-bin/installManager.cgi?action=start&appname=FaceDemo&appid=1234				

Response Params (OK in body)

Response Example

OK

- Stop Application

Request URL	http://<server>/cgi-bin/installManager.cgi?action= stop			
--------------------	---	--	--	--

Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
appname	char[128]	R	the application name	"FaceDemo"
appid	uint	O	the application id	1234
Request Example				
http://192.168.1.108/cgi-bin/installManager.cgi?action=stop&appname=FaceDemo&appid=1234				

Response Params (OK in body)
Response Example
OK

15.2.2 Install Application

- Install application with app data

Request URL	http://<server>/cgi-bin/dhop.cgi?action=uploadApp			
Method	POST			
Request Params (multipart, binary in body)				
Name	Type	R/O	Description	Example
Request Example				
POST /cgi-bin/dhop.cgi?action=uploadApp HTTP/1.1 Host: 172.29.2.176 Content-Length: 413124 Content-Type:multipart/form-data;boundary=<boundary> --<boundary> Content-Disposition: form-data; name="dev_upgrade"; filename="demo.bin" Content-Type: application/octet-stream <app data> --<boundary>--				

Response Params (OK in body)
Response Example
OK

- Install Application with app download url

Request URL	http://<server>/cgi-bin/dhop.cgi?action=installAppByUrl			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
Url	char[]	R	The app download url	https://aaa/bbb/ccc/app.bin
Request Example				
http://192.168.1.108/cgi-bin/dhop.cgi?action=installAppByUrl&Url=https://aaa/bbb/ccc/app.bin				

Response Params (OK in body)

Response Example

OK

15.2.3 Update Application and License

- Update Application with app download url

Request URL	http://<server>/cgi-bin/dhop.cgi?action=updateAppByUrl		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
Url	char[]	R	The app name
appName	char[]	R	The app download url
Request Example			
http://192.168.1.108/cgi-bin/dhop.cgi?action=updateAppByUrl&appName=xxx&Url=https://aaa/bbb/ccc/app.bin			

Response Params (OK in body)
Response Example

OK

- Update firmware with firmware download url

Request URL	http://<server>/cgi-bin/dhop.cgi?action=updateFirmwareByUrl		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
Url	char[]	R	The firmware download url
Request Example			
http://192.168.1.108/cgi-bin/dhop.cgi?action=updateFirmwareByUrl&Url=https://aaa/bbb/ccc/firmware.bin			

Response Params (OK in body)
Response Example

OK

- Update application license with license download url

Request URL	http://<server>/cgi-bin/dhop.cgi?action=updateLicenseByUrl		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
appName	char[]	R	The app name
Url	char[]	R	The license download url

			re.bin
Request Example			
http://192.168.1.108/cgi-bin/dhop.cgi?action=updateLicenseByUrl&appName=xxx&Url=https://aaa/bbb/cc/license.bin			

Response Params (OK in body)
Response Example
OK

- Update application license with license data

Request URL	http://<server>/cgi-bin/dhop.cgi?action=uploadLicense		
Method	POST		
Request Params (multipart, binary in body)			
Name	Type	R/O	Description
appName	char[]	R	The app name
Request Example			
POST /cgi-bin/dhop.cgi?action=uploadLicense&appName=xxx HTTP/1.1 Host: 172.29.2.176 Content-Type: application/octet-stream Content-Length: <length> <app license data>			

Response Params (OK in body)
Response Example
OK

15.2.4 Uninstall Application

Request URL	http://<server>/cgi-bin/dhop.cgi?action=uninstall		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
appName	char[]	R	The app name
Request Example			
http://192.168.1.108/cgi-bin/dhop.cgi?action=uninstall&appName=xxx			

Response Params (OK in body)
Response Example
OK

15.2.5 Download Application Log

Request URL	http://<server>/cgi-bin/dhop.cgi?action=downloadLog		
Method	GET		

Request Params (multipart, binary in body)

Name	Type	R/O	Description	Example
appName	string	R	The app name	xxx

Request Example

```
http://192.168.1.108/cgi-bin/dhop.cgi?action=downloadLog&appName=xxx
```

Response Params (log file content)**Response Example**

HTTP/1.1 200 OK

Server: Device/1.0

Content-Type: Application/octet-stream

Content-Length: <length>

<log file data>

15.3 GPS

15.3.1 Get Capablity

Get the capability of position system.

Request URL	http://<server>/cgi-bin/GpsControl.cgi?action=getCaps		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
Request Example			
http://192.168.1.108/cgi-bin/GpsControl.cgi?action=getCaps			

Response Params (key=value format in body)

Name	Type	R/O	Description	Example
caps	object	O	caps	
+SUPPORTMODE	char[] [32]	O	range: { "GPS" "BEIDOU" "GLONASS" "MIXGB": GPS and BEIDOU "MIXGG": GPS and GLONASS }	["GPS", "BEIDOU"]

Response Example

```
caps.SUPPORTMODE[0]=GPS
caps.SUPPORTMODE[1]=BEIDOU
```

15.3.2 [Config] GPS config

Config Data Params

Name	Type	R/O	Description	Example
GPS	object[]	O		
+Enable	bool	O	true/false	false
+SyncTime	bool	O	true/false	false
+CurMode	char[32]	O	"GPS" "BEIDOU"	"GPS"

Please refer to "4.2.1 Get and Set Configure" for configuration getting and setting. Specific examples are as follows :

Get Config Request Example

```
http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=GPS
```

Get Config Response Example

```
Table.GPS[0].Enable=true
Table.GPS[0].CurMode="GPS"
Table.GPS[0].SyncTime=true
Table.GPS[1].Enable=false
Table.GPS[1].CurMode="BEIDOU"
Table.GPS[1].SyncTime=false
```

Set Config Request Example

```
http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&GPS[0].Enable=true&GPS[0].CurMod
e=GPS&GPS[0].SyncTime=true
```

Set Config Response Example

OK or ERROR

15.3.3 Get GPS Status

Request URL	http://<server>/cgi-bin/positionManager.cgi?action=getStatus			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
Request Example				
http://192.168.1.108/cgi-bin/positionManager.cgi?action=getStatus				

Response Params (key=value format in body)

Name	Type	R/O	Description	Example
status	object	R	GPS	
+Time	char[20]	O	time, ex: 2012-01-04 23:30:30.	2012-01-04 23:30:30
+Longitude	double	O	Longitude, in degrees, positive East, negative West	120.175556
+Latitude	double	O	Latitude, in degrees, positive north, negative South	30.186389
+Altitude	double	O	Altitude, unit: m	45.0
+Speed	double	O	speed, unit: km/h	30.00
+Bearing	double	O	Direction angle, unit: degree	45.3
+WorkStatus	int	O	GPS working status, 0=unlocated, 1=un-differenced	2

			positioning, 2= differential positioning, 3= invalid PPS, 6= estimating	
+SatelliteCount	int	O	Satellite number	11
Response Example				
status.Time=2011-07-03 21:02:32				
status.Longitude=120.175556				
status.Latitude=30.186389				
status.Speed=30.00				
status.Altitude=45.0				
status.Bearing=45.3				
status.SatelliteCount=11				
status.WorkStatus=1				

15.4 Lens Function

15.4.1 Get Lens Capability

Request URL	http://<server>/cgi-bin/LensFunc.cgi?action=getCaps			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
Channel	uint	R	Video channel index which starts from 1.	1
Request Example				
http://192.168.1.108/cgi-bin/LensFunc.cgi?action=getCaps&Channel=1				

Response Params (key=value format in body)				
Name	Type	R/O	Description	Example
Caps	object	R	The capacity.	
+DepthField	uint8	R	Whether support depth field adjust, 0 : not support, 1 : support	1
+AngleAdjust	object	O	The angel adjust capacity.	
++Support	uint8	R	Whether support angel adjust, 0 : not support, 1 : only support horizontal adjust, 2 : only support vertical adjust, 3 : support both horizontal and vertical adjust.	3
++MaxVelocity	uint8	R	The max velocity, range from 1 to 8	8
Response Example				
Caps.DepthField=1				
Caps.AngleAdjust.Support=3				

Caps.AngleAdjust.MaxVelocity=8

15.4.2 Adjust Angle Continuously

Request URL	http://<server>/cgi-bin/LensFunc.cgi?action=adjustAngleContinuously			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
Channel	uint	R	Video channel index which starts from 1.	1
Direction	uint8	R	The moving direction, 0 : negative X axis direction, 1 : positive X axis direction, 2 : negative Y axis direction, 3 : positive Y axis direction	2
Velocity	uint8	R	The adjust velocity, range from 1 to MaxVelocity	2
Request Example				
http://192.168.1.108/cgi-bin/LensFunc.cgi?action=adjustAngleContinuously&Channel=1&Direction=2&Velocity=2				

Response Params (OK in body)
Response Example
OK

15.4.3 Stop Adjusting Angle

Request URL	http://<server>/cgi-bin/LensFunc.cgi?action=stopAdjustingAngle			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
Channel	uint	R	Video channel index which starts from 1.	1
Request Example				
http://192.168.1.108/cgi-bin/LensFunc.cgi?action=stopAdjustingAngle&Channel=1				

Response Params (OK in body)
Response Example
OK

15.4.4 Adjust Depth Field

Request URL	http://<server>/cgi-bin/LensFunc.cgi?action=adjustDepthField			
Method	GET			

Request Params (key=value format in URL)

Name	Type	R/O	Description	Example
Channel	uint	R	Video channel index which starts from 1.	1
DepthField	float	R	The sensor depth position, range from 0 to 1, -1 means reset.	0.3

Request Example

```
http://192.168.1.108/cgi-bin/LensFunc.cgi?action=adjustDepthField&Channel=1&DepthField=0.3
```

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

```
OK
```

15.4.5 Adjust Depth Field Continuously

Request URL	http://<server>/cgi-bin/LensFunc.cgi?action=adjustDepthFieldContinuously			
Method	GET			

Request Params (key=value format in URL)

Name	Type	R/O	Description	Example
Channel	uint	R	Video channel index which starts from 1.	1
DepthFieldSpeed	float	R	The sensor depth position adjust velocity, range from -1 to 1, positive and negative value means difference direction, 0 means stop adjust.	0.03

Request Example

```
http://192.168.1.108/cgi-bin/LensFunc.cgi?action=adjustDepthFieldContinuously&Channel=1&DepthFieldSpeed=0.03
```

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

```
OK
```

15.4.6 Get Depth Field Status

Request URL	http://<server>/cgi-bin/LensFunc.cgi?action=getDepthFieldStatus			
Method	GET			

Request Params (key=value format in URL)

Name	Type	R/O	Description	Example
Channel	uint	R	Video channel index which starts from 1.	1

Request Example

```
http://192.168.1.108/cgi-bin/LensFunc.cgi?action=getDepthFieldStatus&Channel=1
```

Response Params (key=value format in body)

Name	Type	R/O	Description	Example
Status	object	R	The depth position status.	
+Status	string	R	The depth position status, "Normal" : normal status, "AutoDepthField" : adjusting depth position.	Normal
+DepthField	float	R	The sensor board relative position, range from 0 to 1	0.3
+DepthFieldSteps	uint	R	The depth position total step.	1000
+ResetResult	string	O	The depth position reset result, only valid when reset. "Success" : reset success, "Failed" : reset failed	Success

Response Example

Status.Status=Normal

Status.DepthField=0.3

Status.DepthFieldSteps=1000

Status.ResetResult=Success

15.4.7 Auto Adjust Depth Field

Request URL	http://<server>/cgi-bin/LensFunc.cgi?action=autoAdjustDepthField			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
Channel	uint	R	Video channel index which starts from 1.	1
Request Example				
http://192.168.1.108/cgi-bin/LensFunc.cgi?action= autoAdjustDepthField&Channel=1				

Response Params (OK in body)

Response Example

OK

15.4.8 Scene Correction

Request URL	http://<server>/cgi-bin/api/LensFunc/correctScene			
Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
Channel	int	R	Channel number, starting from 1.	1
Type	enumint	R	Correction type. enumint{ 0: Automatic correction	1

			1: Manual correction } O	
Direction	enumint	O	Rotation direction. This parameter is only valid when the correction method is manual correction. enumint{ 0: Clockwise 1: Counter clockwise }	1
Step	int	O	Speed (range: 0–8). 0 means stop rotation. When the rotation mode is continuous rotation, it means the rotation speed. When the rotation mode is point-based rotation, it means the speed of each rotation. This parameter is only valid when the correction mode is manual correction.	5
Mode	enumint	O	Rotation mode. This parameter is only valid when the correction mode is manual correction. enumint{ 0: Continuous rotation 1: Point-based rotation }	1

Request Example

```
{
  "Channel": 1,
  "Type": 1,
  "Direction": 1,
  "Step": 5,
  "Mode": 1
}
```

Response Params (JSON format in body)

Name	Type	R/O	Description	Example
Response Example				
{}				

15.4.9 Reset Angle

Reset camera lens angle

Request URL	http://<server>/cgi-bin/LensManager.cgi?action=resetAngle		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
Channel	uint	R	Video channel index which starts from 1.
Request Example			

http://192.168.1.108/cgi-bin/LensManager.cgi?action=resetAngle&Channel=1

Response Params (OK in body)

Response Example

OK

15.5 FishEye

15.5.1 Get FishEye Capability

The method described in the Section “4.5.12 Get video input capability” should be used first. In the response, it will contain these message "caps.FishEye=false", if the value of the FishEye is true, Then you can use the method described below to get the detail capability.

Request URL	http://<server>/cgi-bin/devVideoInput.cgi?action=getCapsEx		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
channel	uint	R	Video channel index which starts from 1.
name	char[]	R	it is fixed to VideoInFishEye
Request Example			
http://192.168.1.108/cgi-bin/devVideoInput.cgi?action=getCapsEx&channel=1&name=VideoInFishEye			

Response Params (key=value format in body)			
Name	Type	R/O	Description
caps	object	R	capabilities
+Type	char[16]	O	it can be Chip, Plugin, and ChipAndPlugin. Chip means only support calibrate by device. Plugin means only support calibrate by plugin. ChipAndPlugin means support both.
+MountMode	char[][32]	O	It can be: "WallMode", "CeilMode", "Floor Mode", "180CeilMode"
+CalibrateMode	char[][32]	O	It can be:"Original", "Config", "Panorama", "DoublePanorama", "OriginalPlusThreeEPtzRegion", "Single", "FourEPtzRegion", "TwoEPtzRegion", "Normal"
+EPtzCmd	char[][32]	O	It can be ZoomIn, ZoomOut, Up, Down, Left, Right, RotateClock,

		RotateAntiClock, Stop, TapView, and ShowRegion.
Response Example		
caps.Type=Chip caps.MountMode[0]=WallMode caps.MountMode[1]=CeilMode caps.CalibrateMode[0]=Original caps.CalibrateMode[1]=Config caps.CalibrateMode[2]=Panorama caps.EPtzCmd[0]=Up caps.EPtzCmd[1]=Down		

15.5.2 [Config] FishEye Setting

Fisheye camera configuration, currently used to save user set window coordinate information and fisheye resolution information.

Config Data Params				
Name	Type	R/O	Description	Example
FishEye	object[]	O	Each channel corresponds to a configuration.	
+Position	object	O	Window position information	
++FourEPtzRegion	object[]	O	Window coordinate information set by the user in 4-screen mode	
+++WinID	uint8	O	Window ID [1,4]	1
+++X	uint16	O	The abscissa of the focus of the Etz (electronic pan tilt) in the image before correction. The value range is the abscissa range of the fish eye circle	1
+++Y	uint16	O	The ordinate of the focus of the Etz (electronic pan tilt) in the image before correction. The value range is the ordinate range of the fish eye circle	1
+++HorizontalAngle	int	O	Horizontal angle of EPtz 0~360	90
+++VerticalAngle	int	O	Vertical angle of EPtz 0~90	90
++OriginalPlusThreeEPtzRegion	object[3]	O	The window coordinate information set by the user in 1+3 mode, in the same format as FourEPtzRegion mode	

++TwoEPtzRegion	object[2]	O	Same format as FourEPtzRegion mode	
++Single	object[1]	O	Same format as FourEPtzRegion mode	
+PlaceHolder	uint8	O	Lens installation method 1(CeilMode), 2(WallMode), 3(FloorMode).	1
+CalibrateMode	char[16]	O	Fish eye correction mode Original, Config, Panorama, DoublePanorama, OriginalPlusThreeEPtzRegion, Single, FourEPtzRegion, TwoEPtzRegion, and Normal.	"Original"
+FocusPostion	object	O	Fish eye focal point position (X-ray machine requirement)	
++CircularOffset	int32	O	The deflection angle of a circle.	0
++PanoramaOffset	int32	O	Panoramic offset	0
++Valid	int32	O	Fish eye focal position. 0 indicates invalid, 1 indicates valid	0

Examples of reading and modifying configurations are as follows:

Get Config Request Example

<http://10.0.0.8/cgi-bin/configManager.cgi?action=getConfig&name=FishEye>

Get Config Response Example

```
table.FishEye[0].CalibrateMode=DoublePanorama
table.FishEye[0].PlaceHolder=1
table.FishEye[0].Position.FourEPtzRegion[0].HorizontalAngle=120
table.FishEye[0].Position.FourEPtzRegion[0].VerticalAngle=120
table.FishEye[0].Position.FourEPtzRegion[0].WinID=1
table.FishEye[0].Position.FourEPtzRegion[0].X=2648
table.FishEye[0].Position.FourEPtzRegion[0].Y=2648
table.FishEye[0].Position.FourEPtzRegion[1].HorizontalAngle=120
table.FishEye[0].Position.FourEPtzRegion[1].VerticalAngle=120
table.FishEye[0].Position.FourEPtzRegion[1].WinID=2
table.FishEye[0].Position.FourEPtzRegion[1].X=5544
table.FishEye[0].Position.FourEPtzRegion[1].Y=2648
table.FishEye[0].Position.FourEPtzRegion[2].HorizontalAngle=120
table.FishEye[0].Position.FourEPtzRegion[2].VerticalAngle=120
table.FishEye[0].Position.FourEPtzRegion[2].WinID=3
table.FishEye[0].Position.FourEPtzRegion[2].X=2648
table.FishEye[0].Position.FourEPtzRegion[2].Y=5544
```

```

table.FishEye[0].Position.FourEPtzRegion[3].HorizontalAngle=120
table.FishEye[0].Position.FourEPtzRegion[3].VerticalAngle=120
table.FishEye[0].Position.FourEPtzRegion[3].WinID=4
table.FishEye[0].Position.FourEPtzRegion[3].X=5544
table.FishEye[0].Position.FourEPtzRegion[3].Y=5544
table.FishEye[0].Position.OriginalPlusThreeEPtzRegion[0].HorizontalAngle=60
table.FishEye[0].Position.OriginalPlusThreeEPtzRegion[0].VerticalAngle=60
table.FishEye[0].Position.OriginalPlusThreeEPtzRegion[0].WinID=2
table.FishEye[0].Position.OriginalPlusThreeEPtzRegion[0].X=5544
table.FishEye[0].Position.OriginalPlusThreeEPtzRegion[0].Y=2648
table.FishEye[0].Position.OriginalPlusThreeEPtzRegion[1].HorizontalAngle=60
table.FishEye[0].Position.OriginalPlusThreeEPtzRegion[1].VerticalAngle=60
table.FishEye[0].Position.OriginalPlusThreeEPtzRegion[1].WinID=3
table.FishEye[0].Position.OriginalPlusThreeEPtzRegion[1].X=2648
table.FishEye[0].Position.OriginalPlusThreeEPtzRegion[1].Y=5544
table.FishEye[0].Position.OriginalPlusThreeEPtzRegion[2].HorizontalAngle=60
table.FishEye[0].Position.OriginalPlusThreeEPtzRegion[2].VerticalAngle=60
table.FishEye[0].Position.OriginalPlusThreeEPtzRegion[2].WinID=4
table.FishEye[0].Position.OriginalPlusThreeEPtzRegion[2].X=5544
table.FishEye[0].Position.OriginalPlusThreeEPtzRegion[2].Y=5544
table.FishEye[0].Position.Single[0].HorizontalAngle=60
table.FishEye[0].Position.Single[0].VerticalAngle=60
table.FishEye[0].Position.Single[0].WinID=1
table.FishEye[0].Position.Single[0].X=4096
table.FishEye[0].Position.Single[0].Y=4096

```

Set Config Request Example

`http://10.0.0.8/cgi-bin/configManager.cgi?action=setConfig&FishEye[0].PlaceHolder=3&FishEye[0].CalibrateMode=FourEPtzRegion`

Set Config Response Example

OK

15.6 Radar Adaptor

15.6.1 Get Radar Capability

Note: This interface is deprecated. Please use the interface getCapsEx.

Radar equipment detection capability request

Request URL	<code>http://<server>/cgi-bin/radarAdaptor.cgi?action=getCaps</code>			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
channel	uint	R	channel index which starts from 1.	1
Request Example				

<http://192.168.1.108/cgi-bin/radarAdaptor.cgi?action=getCaps&channel=1>

Response Params (key=value format in body)

Name	Type	R/O	Description	Example
caps	object	R	capabilities	
+DetectionRange	int	O	Zoom in 100 times, while the true value is 50.00	5000
+DetectionAngle	int	O	Zoom in 100 times	8003
+DetectionHuman	int	O	Zoom in 100 times	4000

Response Example

```
caps.DetectionRange = 5000
caps.DetectionAngle = 8003
caps.DetectionHuman = 4000
```

15.6.2 Get Radar Capability (Enhanced)

Radar equipment caps request

Request URL	<a href="http://<server>/cgi-bin/radarAdaptor.cgi?action=getCapsEx">http://<server>/cgi-bin/radarAdaptor.cgi?action=getCapsEx			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
channel	uint	R	channel index which starts from 1.	1
Request Example				
http://192.168.1.108/cgi-bin/radarAdaptor.cgi?action=getCapsEx&Channel=1				

Response Params (key=value format in body)

Name	Type	R/O	Description	Example
caps	object	R	capabilities	
+DetectionRange	int	O	Radar detection range Zoom in 100 times unit: meter	5000
+DetectionAngle	int	O	Radar detection angle Zoom in 100 times unit: degree	8003
+DetectionHuman	int	O	Radar human detection range Zoom in 100 times unit: meter	4000
+Capacity	object	O		
++Support	bool	O	Power setting enable	true
++List	int[24]	O	Power allows setting list, no more than 24 choices	[150, 300]
+RadarScene	object	O		
++Support	bool	O	Scene setting enable	true

+RadarChannel	object	O		
++Support	bool	O	Radar channel setting enable	true
++List	uint[24]	O	Channel allows setting list, no more than 24 choices	[1, 2]
+MovedDetect	object	O		
++Support	bool	O	Device motion detection setting enabled	true

Response Example

```

caps.DetectionRange = 5000
caps.DetectionAngle = 8003
caps.DetectionHuman = 4000
caps.Capacity.Support = true
caps.Capacity.List[0] = 150
...
caps.Capacity.List[23] = 300
caps.RadarScene.Support = true
caps.RadarChannel.Support = true
caps.RadarChannel.List[0] = 1
...
caps.RadarChannel.List[23] = 2
caps.MovedDetect.Support = true

```

15.6.3 Get Status

Radar Equipment Status Request

Request URL	http://<server>/cgi-bin/radarAdaptor.cgi?action=getStatus			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
channel	uint	R	channel index which starts from 1.	1

Request Example

```
http://192.168.1.108/cgi-bin/radarAdaptor.cgi?action=getStatus&channel=1
```

Response Params (key=value format in body)				
Name	Type	R/O	Description	Example
status	object	R		
+State	char[]	R	Radar state, the range is { "Normal" , "Abnormal" }	Normal

Response Example

```
status.State = Normal
```

15.6.4 Calculate Real Size

Radar real map size calculation

Request URL	http://<server>/cgi-bin/radarAdaptor.cgi?action=calculateRealSize			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
channel	uint	R	channel index which starts from 1.	1
distance	double	R	Segment distance, Unit: m	10.01
startPoint	int[2]	R	Line segment start coordinates	[10,20]
endPoint	int[2]	R	Line segment end coordinates	[30,40]
Request Example				
http://192.168.0.108/cgi-bin/radarAdaptor.cgi?action=calculateRealSize&channel=1&distance=10.01&startPoint[0]=10&startPoint[1]=20&endPoint[0]=30&endPoint[1]=40				

Response Params (key=value format in body)				
Name	Type	R/O	Description	Example
realSize	object	R		
+realMap	int[2]	R	Length and width of actual map, unit:meter	[300,400]
Response Example				
status.State = Normal				

15.6.5 Subscribe Alarm Point Info

Radar track point request; A detach occurs when the connection is disconnected.

Request URL	http://<server>/cgi-bin/radarAdaptor.cgi?action=attachAlarmPointInfo			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
Channel	uint	R	Video channel index which starts from 1	1
heartbeat	int	O	Send heartbeat interval, range is [1, 60], unit is second. If the URL contains this parameter, and the value is 5, it means every 5 seconds the device should send the heartbeat message to the client, the heartbeat message are "Heartbeat". If this parameter is not present, its default value is 60.	5
Request Example				
http://192.168.0.108/cgi-bin/radarAdaptor.cgi?action=attachAlarmPointInfo&Channel=1&heartbeat=5				

Response Params (multipart format in body)

Name	Type	R/O	Description	Example
Channel	int	O	Video channel number, radar equipment does not pay attention to the channel The channel number starts from 0 and defaults to 0	0
info	object[]	O	Up to 64 points	
+PointType	int	O	Mask for point type Bit0: Invalid Bit1: The current point is a vanishing trajectory point Bit2: The current point is being monitored through linkage Bit3: The current point is the alarm point in the alarm area Note: There may be multiple states of points, for example, when the value is 0xC, it indicates that the current point is both the point being monitored and the alarm point in the alarm area	0x08
+RegionNumber	int	O		1
+ObjectType	int	O	Mask of the type of object the point refers to 0x00 Unrecognized target 0x01 Target is a person 0x02 Target is transportation 0x03 Target is Tree 0x04 Target is a building 0x05 Target is screen 0x06 Target is an animal 0x07 Target is a large ship 0x08 Target is China Shipbuilding Corporation 0x09 Target is a small boat 0x10 Target is a stationary target type	0x00
+TrackID	int	O	range [0,63]	10
+Distance	int	O	The polar coordinate value of the current point pixel - distance, multiplied by 100 times the result, in meters	1500
+Angle	int	O	The polar coordinate value of the current point - angle, multiplied	26

			by 100 times the result, in degrees	
+Longitude	int32	O	Longitude, transmitted in integer form, 1000000 times, valid to 6 decimal places, for example, 120125400 represents 120.1254; Positive numbers represent the east longitude, negative numbers represent the west longitude	120125400
+Latitude	int32	O	Latitude, transmitted in integer form, 1000000 times, valid to 6 decimal places, and filled with 0 if less than 6 decimal places, for example, 120125400 represents 120.1254; Positive numbers represent north latitude, negative numbers represent south latitude	120125400
+Speed	int	O	The result of increasing the current point speed by 100 times, in meters per second	14
+TrackerIP	uint32	O	inet_addr("192.168.1.109")	0
+CoordinatesPosX	int	O	Horizontal coordinate, normalized to 8192	1024
+CoordinatesPosY	int	O	Vertical coordinate, normalized to 8192	1024
Rule	object[]	O	rule	
+Id	int	O	rule id	
+PointNumber	int	O	count	0
RealUTC	int32	O		4646496

Response Example

HTTP/1.1 200 OK

Server: Device/1.0

Content-Type: multipart/x-mixed-replace; boundary=<boundary>

Connection: closed

--<boundary>

Content-Type: application/octet-stream

Content-Length: <length>

Channel=1

info[0].PointType=0x80

info[0].RegionNumber=1

info[0].ObjectType=1

info[0].TrackID=1

```

info[0].Distance=1001 //Zoom in 100 times, while the true value is 10.01
info[0].Angle=2010 //Zoom in 100 times
info[0].Speed=501 //Zoom in 100 times
.....
info[3].PointType=0x80
.....

--<boundary>
Content-Type: text/plain
Content-Length: 11

Heartbeat
-<boundary>
Content-Type: application/octet-stream
Content-Length: <length>

Channel=1
info[0].PointType=0x80
.....
--<boundary>
...

```

15.6.6 Manual Locate

Radar manual positioning request

Request URL	http://<server>/cgi-bin/radarAdaptor.cgi?action=manualLocate			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
channel	uint	R	channel index which starts from 1.	1
point	int[2]	R	Pixel coordinates in the code stream	[10,10]
Request Example				
http://192.168.1.108/cgi-bin/radarAdaptor.cgi?action=manualLocate&channel=1&point[0]=10&point[1]=10				

Response Params (OK in body)
Response Example
OK

15.6.7 Start Radar Calibration

Radar equipment calibration

Request URL	http://<server>/cgi-bin/radarAdaptor.cgi?action=startRadarCalibration
--------------------	---

Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
channel	uint	R	channel index which starts from 1.	1
mode	uint	R	Calibration mode, 0:auto, 1:manual,2:init	1
direction	uint	R	Calibration direction, effective in manual mode , 0: invalid, 1: top, 2: bottom, 3: left, 4: right, 5: left_top, 6: right_top, 7: left_bottom, 8: right_bottom	1
step	uint	R	step size	5
Request Example				
http://192.168.1.108/cgi-bin/radarAdaptor.cgi?action=startRadarCalibration&channel=1&mode=1&direction=1&step=5				

Response Params (OK in body)
Response Example
OK

15.6.8 Add Radar Link SD

Radar link SD added

Request URL	http://<server>/cgi-bin/radarAdaptor.cgi?action=addRadarLinkSD			
Method	GET			
Request Params (key=value format in URL)				
Name	Type	R/O	Description	Example
Devices	array<object>	R	Add or update the added device login information	
+PassWord	char[32]	R	password, plaintext	"123456"
+UserName	char[128]	R	user name	"admin"
+Port	integer	R	port	5000
+SDLinkIP	char[32]	R	device ip adress	"192.168.1.108"
Channel	integer	R	video channel, which starts from 1.	1
Request Example				
http://192.168.1.108/cgi-bin/radarAdaptor.cgi?action=addRadarLinkSD&Devices[0].SDLinkIP=10.11.17.98&Devices[0].Port=5000&Devices[0].Usename=admin&Devices[0].PassWord=123456&Channel=1				

Response Params (key=value format in body)
Name
Type
R/O
Description
Example
SD
object
R
dome camera
+Info
object
R
information
++ErrorCode
integer
R
errorcode 0: normal 1: overtime
0

		2: Wrong user name or password 3: The device does not support linkage	
Response Example			
SD.Info.ErrorCode=0			

15.6.9 Del Radar Link SD

Remove radar link SD

Request URL	http://<server>/cgi-bin/radarAdaptor.cgi?action=delRadarLinkSD		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
Devices	array<object>	R	Delete the SD added in radar equipment Empty array will delete all linkage equipment by default
+SDLLinkIP	char[32]	R	device to be deleted
Channel	integer	R	video channel, which starts from 1.
Request Example			
http://192.168.1.108/cgi-bin/radarAdaptor.cgi?action=delRadarLinkSD&Devices[0].SDLLinkIP=10.11.17.9&Channel=1			

Response Params (key=value format in body)			
Name	Type	R/O	Description
Response Example			
OK			

15.6.10 Get Link SD State

Request URL	http://<server>/cgi-bin/radarAdaptor.cgi?action=getLinkSDState		
Method	GET		
Request Params (key=value format in URL)			
Name	Type	R/O	Description
channel	uint	R	channel index which starts from 1.
ip	char[24][]	O	IP address of the linkage device
Request Example			
http://192.168.0.108/cgi-bin/radarAdaptor.cgi?action=getLinkSDState&channel=1&ip[0]=192.168.1.115&ip[1]=192.168.1.116			

Response Params (key=value format in body)			
Name	Type	R/O	Description
state	object[]	R	device state
+SDLLinkIP	char[32]	O	device ip
+State	uint8	O	0: out of time 1: offline

+Channel	int	O	2: disconnected virtual channel,starts from 1	1
----------	-----	---	--	---

Response Example

```
state[0].SDLinkIP=192.168.1.115
state[0].State=0
state[0].Channel=1
state[1].SDLinkIP=192.168.1.116
state[1].State=0
state[1].Channel=2
```

15.6.11 [Config] MapPara

Config Data Params				
Name	Type	R/O	Description	Example
MapPara	object[]	R	One-dimensional array, with one configuration for each video channel	
+RadarCoordinate	object	R	Configuration of radar position on the map	null
++RadarDirectionAngle	double	R	Configuration of radar direction (°)	70.5
++RadarPixelPoint	int[]	R	Pixel coordinates of radar on the map	[10, 20]
+MapSize	object	R	Configuration of map size	null
++PixelLine	int[][]	R	Pixel coordinates of line segment in the image	[[10,10], [20,20]]
++Distance	double	R	The actual length represented by the line segment in the image (m)	10.1

Please refer to "4.2.1 Get and Set Configure" for configuration getting and setting. Specific examples are as follows :

Get Config Request Example

```
http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=MapPara
```

Get Config Response Example

```
table.MapPara.RadarCoordinate.RadarDirectionAngle=70.5
table.MapPara.RadarCoordinate.RadarPixelPoint[0]=10
table.MapPara.RadarCoordinate.RadarPixelPoint[1]=20
table.MapPara.MapSize.PixelLine[0][0]=10
table.MapPara.MapSize.PixelLine[0][1]=10
table.MapPara.MapSize.PixelLine[1][0]=20
table.MapPara.MapSize.PixelLine[1][1]=20
table.MapPara.MapSize.Distance=10.1
```

Set Config Request Example

```
http://10.0.0.8/cgi-bin/configManager.cgi?action=setConfig&MapPara.MapSize.Distance=10.1&MapPara.RadarCoordinate.RadarDirectionAngle=70.5
```

Set Config Response Example

OK

15.6.12 [Config] RadarAnalyseRule

RadarAnalyseRule parameters:

Config Data Params				
Name	Type	R/O	Description	Example
RadarAnalyseRule	object[] [24]	R	Radar alarm linkage configuration object[] [24] Two-dimensional array. The first dimension indicates video channel, and the second dimension indicates multiple linkage configurations.	
+EventHandle	object	R	Alarm linkage configuration	null
++LogEnable	bool	R	Whether to record alarm logs. The type is fixed to EventStart/EventStop/EventPulse and the original event type is recorded in the Detail.Code field in the log.	false
++TimeSection	TimeSection[] []	R	Alarm period. TimeSection is used for the filtering of notify action. No data is sent if it is not within the period. All fields except this one are used for listener response.	[["1 00:00:00-23:59:59", "0 00:00:00-23:59:59", "0 00:00:00-23:59:59", "0 00:00:00-23:59:59", "0 00:00:00-23:59:59", "0 00:00:00-23:59:59",]]
++SnapshotChannels	int[]	R	List of snapshot channel numbers. One-dimensional array. Each member indicates that the corresponding channel needs to take snapshot, and the channel number starts from 0.	[0]
++SnapshotEnable	bool	R	Enable snapshotg	false
++RecordLatc h	int	R	Record delay time (second) Range [10, 300]	10
++RecordChann els	int[]	R	List of record channel numbers. One-dimensional array. Each member indicates that the corresponding channel needs to record, and the channel number starts from 0.	[0]
++RecordEna ble	bool	R	Record enabling, with RecordChannels. Record is enabled if it is true; start recording when the event	false

			action is start and stop recording when it is stop. Record is not enabled if it is false.	
++VoiceEnable	bool	R	Voice prompt	false
++BeepEnable	bool	R	Buzzer	false
++MessageEnable	bool	R	Upload to the alarm center server.	false
++MailEnable	bool	R	Send email.	false
++TipEnable	bool	R	Local message box prompt	false
++AlarmOutLatch	int	R	Output delay time (second) after the alarm output stops. Range [10, 300]	10
++AlarmOutEnable	bool	R	Enable alarm output	false
++AlarmOutChannels	int[]	R	List of alarm output channel numbers. One-dimensional array. Each member indicates that the corresponding channel needs to output alarm, and the channel number starts from 0.	[0]
+Enable	bool	R	Enable alarm enabling	false
+AlarmOutNumber	int	R	Alarm configuration number, which is unique.	1
+SDLinkIP	char[32]	O	Controlled device IP (which is optional and reserved. Ignore this field due to business adjustment).	"192.168.1.108"

Please refer to "4.2.1 Get and Set Configure" for configuration getting and setting. Specific examples are as follows :

Get Config Request Example

```
http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=RadarAnalyseRule
```

Get Config Response Example

table.RadarAnalyseRule.Enable=true
table.RadarAnalyseRule.SDLinkIP=192.168.1.108
table.RadarAnalyseRule.EventHandler=....(output of EventHandler is described in GetEventHandler)

Set Config Request Example

```
http://10.0.0.8/cgi-bin/configManager.cgi?action=setConfig&RadarAnalyseRule.Enable=true&RadarAnalyseRule.SDLinkIP=192.168.1.108
```

Set Config Response Example

OK

15.6.13 [Config] RadarCalibration

Config Data Params				
Parameter	Type	R/O	Description	Example
RadarCalibration	object[]	R	Configuration of radar and PTZ cameras calibration. One-dimensional array, with one configuration for each video channel.	
+SlopeAngle	double	R	Slope compensation angle.	10.0
+InstallHeight	double	R	Radar installation height (m).	10.5
+CalibrationParas	object[]	R	Calibration parameter.	null
++TiltRecoupAngle	double	R	Vertical compensation angle of linked PTZ camera. The range is -90° to 90°; 0° in the horizontal direction; 0 to 90° below the horizontal direction; -90° to 0° above the horizontal direction.	5.0
++LinkSDHeight	double	R	Installation height of linked PTZ camera (m).	10.5
++CalibrationPos	object[]	R	Array of calibration points.	null
+++Ptz	float[3]	R	PTZ value (normalized value) of calibration points. The first element is the horizontal angle, which is normalized to -1 to 1. The second element is the vertical angle, which is normalized to -1 to 1. The third element is the zoom times, which is normalized to 0 to 1.	[0.95, 0.43, 0.12]
+++AlarmPixelPoint	int[2]	R	Pixel coordinates of calibration points.	[10, 20]
++SDLinkIP	char[32]	R	Controlled device IP.	"192.168.1.108"

Please refer to "4.2.1 Get and Set Configure" for configuration getting and setting. Specific examples are as follows :

Get Config Request Example

```
http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=RadarCalibration
```

Get Config Response Example

```
table.RadarCalibration[0].SlopeAngle=10.0
table.RadarCalibration[0].InstallHeight=10.5
table.RadarCalibration[0].CalibrationParas[0].TiltRecoupAngle=5.0
...

```

Set Config Request Example

```
http://10.0.0.8/cgi-bin/configManager.cgi?action=setConfig&RadarCalibration[0].SlopeAngle=10.0&RadarCalibration[0].CalibrationParas[0].TiltRecoupAngle=5.0
```

Set Config Response Example

```
OK
```

15.6.14 [Config] RadarGuardLine

Config Data Params				
Parameter	Type	R/O	Description	Example
RadarGuardLine	object[][][10]	R	Two-dimensional array. The first dimension indicates video channel, and the second dimension indicates multiple rule lines (up to 10).	
+AlarmOutNumber	integer	R	Alarm output configuration. Configure the AlarmOutNumber field according to RadarAnalyseRule.	1
+TargetFilter	integer	R	Target filtering bit0: Reserved bit1: Human. bit2: Vehicle. bit3: Animal (Enable by setting it to 1)	0x01
+Polygon	integer	R	Rule point, pixel coordinates [width, height]. Range of number of points [3, 24]	[[0, 0], [128, 128], ..., []]
+Type	char[32]	R	Rule type, alarm or shield type. High Alarm Alarm Shield	"Alarm"
+Enable	bool	R	Whether to enable the rules	true
+RegionNumber	integer	R	Rule line number Range [1, 10]	1
+Name	char[128]	R	Rule name, which cannot be repeated.	"GuardLine1"

Please refer to "4.2.1 Get and Set Configure" for configuration getting and setting. Specific examples are as follows :

Get Config Request Example

```
http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=RadarGuardLine
```

Get Config Response Example

```
table.RadarGuardLine[0][0].AlarmOutNumber =1
table.RadarGuardLine[0][0].Enable=true
table.RadarGuardLine[0][0].Type=Alarm
table.RadarGuardLine[0][0].RegionNumber=1
...
```

Set Config Request Example

```
http://10.0.0.8/cgi-bin/configManager.cgi?action=setConfig&RadarGuardLine[0][0].AlarmOutNumber
=1&RadarGuardLine[0][0].Enable=true
```

Set Config Response Example

```
OK
```

15.6.15 [Config] RadarLink

Config Data Params				
Name	Type	R/O	Description	Example
RadarLink	object[]	R	One-dimensional array, with one configuration for each video channel	
+RadarLink	bool	R	Enable radar-PTZ linkage	true

Please refer to "4.2.1 Get and Set Configure" for configuration getting and setting. Specific examples are as follows :

Get Config Request Example

```
http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=RadarLink
```

Get Config Response Example

```
table.RadarLink[0].RadarLink=true
```

```
table.RadarLink[1].RadarLink=true
```

```
...
```

Set Config Request Example

```
http://10.0.0.8/cgi-bin/configManager.cgi?action=setConfig&RadarLink[0].RadarLink=true
```

Set Config Response Example

```
OK
```

15.6.16 [Config] RadarLinkDevice

Config Data Params				
Name	Type	R/O	Description	Example
RadarLinkDevice	object[][]	R	Remote device linkage configuration. object[][24] Two-dimensional array. The first dimension indicates video channel, and the second dimension indicates multiple remote linkage devices.	
+DeviceName	char[128]	R	Device name	"XXX northdoor"
+DeviceType	char[32]	R	Device type	"200w sd"
+Vendor	char[32]	R	Device source	"VVV"
+SDLinkIP	char[32]	R	Controlled device IP	"192.168.1.108"

Please refer to "4.2.1 Get and Set Configure" for configuration getting and setting. Specific examples are as follows :

Get Config Request Example

```
http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=RadarLinkDevice
```

Get Config Response Example

```
table.RadarLinkDevice[0][0].DeviceName=NorthDoor
```

```
table.RadarLinkDevice[0][0].DeviceType=xxx
```

```

table.RadarLinkDevice[0][0].Vendor=VVV
table.RadarLinkDevice[0][0].SDLinkIP=192.168.1.108
...

```

Set Config Request Example

```
http://10.0.0.8/cgi-bin/configManager.cgi?action=setConfig&RadarLinkDevice[0][0].DeviceName=NorthDoor&RadarLinkDevice[0][0].DeviceType=xxx
```

Set Config Response Example

```
OK
```

15.6.17 [Config] RadarPara

RadarPara

Config Data Params

Parameter	Type	R/O	Description	Example
RadarPara	object[]	R	Radar function configuration. One-dimensional array, with one configuration for each video channel.	
+Sensitivity	object	O	Recognition sensitivity information.	null
++Level	integer	O	Optional. Skip if this field does not exist. The range is -5 to 5; 0 is the default value; the smaller the value, the higher the human recognition rate; the larger the value, the higher the vehicle recognition rate.	0
+TargetRatio	object	R	Tracking screen ratio for radar-PTZ linkage.	null
++Ratio	integer	R	Reciprocal of screen ratio. Required. Calculate the reciprocal. For example, when the screen ratio is 0.5, Ratio = 2; when the screen ratio is 0.25, Ratio = 4.	1
+RadarChannel	object	R	Radar channel configuration.	null
++Route	integer	R	Channel selection. 1: Channel 1 2: Channel 2	1
+Capacity	object	R	Configuration of radar transmitting power.	null
++Power	integer	R	Radar transmitting power is measured by detection distance (m).	300
+Structured	object	R	Structured information configuration	null
++TrackDisplayTime	integer	O	Range [1, 30] (s).	5
++TrackType	integer	O	0: Disable 1: Trajectory box 2: Trajectory	0
++Enable	bool	R	Enable structured information display	false
+Scene	object	R	Scene configuration	null

++Type	integer	R	Scene selection. 1: Default. 2: Shrub. 3: Spacious. 4: Custom.	1
--------	---------	---	--	---

Please refer to "4.2.1 Get and Set Configure" for configuration getting and setting. Specific examples are as follows :

Get Config Request Example

```
http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=RadarPara
```

Get Config Response Example

```
table.RadarPara[0].Sensitivity.Level=1  
table.RadarPara[0].TargetRatio.Ratio=1  
table.RadarPara[0].Scene.Type=1  
...
```

Set Config Request Example

```
http://10.0.0.8/cgi-bin/configManager.cgi?action=setConfig&RadarPara[0].Sensitivity.Level=1&RadarPara[0].Scene.Type=1
```

Set Config Response Example

OK

15.6.18 [Config] RadarTrackGlobal

Config Data Params				
Name	Type	R/O	Description	Example
RadarTrackGlobal	object[]	R	One-dimensional array, with one configuration for each video channel	
+TrackSwitchMode	char[32]	R	Tracking switching mode Tour/time priority/distance priority Rotation/TimePriority/DistancePriority	"Rotation"
+TrackSwitchTime	int	R	Tracking switching time (s) Range [1, 15]	5
+TrajectoryTime	int	R	Trajectory time (s) Range [3, 30]	5
+SectorDisable	bool	R	Enable protection area hiding	false

Please refer to "4.2.1 Get and Set Configure" for configuration getting and setting. Specific examples are as follows :

Get Config Request Example

```
http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=RadarTrackGlobal
```

Get Config Response Example

```
table.RadarTrackGlobal[0].TrackSwitchMode=Rotation  
table.RadarTrackGlobal[0].TrackSwitchTime=5
```

```

table.RadarTrackGlobal[0].TrajectoryTime=5
table.RadarTrackGlobal[0].SectorDisable=false
...

```

Set Config Request Example

```
http://10.0.0.8/cgi-bin/configManager.cgi?action=setConfig&RadarTrackGlobal[0].SectorDisable=false&RadarTrackGlobal[0].TrackSwitchMode=Rotation&RadarTrackGlobal[0].TrackSwitchTime=5
```

Set Config Response Example

```
OK
```

15.6.19 [Config] RemoteSDLink

Config Data Params				
Name	Type	R/O	Description	Example
RemoteSDLink	object[]/[24]	R	Remote device linkage enabling configuration. Two-dimensional array. The first dimension indicates video channel, and the second dimension indicates multiple remote linkage devices.	
+RadarLink	bool	R	Enable linkage.	true
+SDLinkIP	char[32]	R	Controlled device IP.	"192.168.1.108"

Please refer to "4.2.1 Get and Set Configure" for configuration getting and setting. Specific examples are as follows :

Get Config Request Example

```
http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=RemoteSDLink
```

Get Config Response Example

```

table.RemoteSDLink[0][0].RadarLink=true
table.RemoteSDLink[0][0].SDLinkIP=192.168.1.108
table.RemoteSDLink[0][1].RadarLink=true
table.RemoteSDLink[0][1].SDLinkIP=192.168.1.107
...

```

Set Config Request Example

```
http://10.0.0.8/cgi-bin/configManager.cgi?action=setConfig&RemoteSDLink[0][0].RadarLink=true&RemoteSDLink[0][0].SDLinkIP=192.168.1.108
```

Set Config Response Example

```
OK
```

15.7 Water Radar

15.7.1 Acquire Radar Capability

Request URL	http://<server>/cgi-bin/api/WaterRadar/getCaps
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
Support	bool	R	Whether the water conservancy radar function is supported. false: Not Supported true: Supported	true
Response Example				
{ "Support": true }				

15.7.2 Get Radar Detection Target Data

Request URL	http://<server>/cgi-bin/api/WaterRadar/getObjectInfo			
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
WaterLevel	float	R	Water level, CM	1000.0
Response Example				
{ "WaterLevel": 1000.0 }				

15.8 Water Quality Detection

15.8.1 Get Capability

Acquiring the interface for water quality detection.

Request URL	http://<server>/cgi-bin/api/WaterDataStatServer/getCaps			
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
Support	enumint	O	Supports water quality detection currently. enumint{ 0: No 1: Yes }	0
SupportLocalDataStore	enumint	O	Supports local storage. enumint{ 0: No 1: Yes }	0

Response Example

```
{
  "Support": 0,
  "SupportLocalDataStore": 0
}
```

15.8.2 Get Real-time Detection Data

The water quality detection data is refreshed every minute, and you can acquire the data in real time.

Request URL	http://<server>/cgi-bin/api/WaterDataStatServer/getWaterData			
Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
Type	enumchar[32][32]	O	enumchar[32][32]{ "Quality": Water quality category "PH": pH value "NTU": Turbidity value "NH3-N": Ammonia nitrogen value }	["PH", " NTU ", ...]

Request Example

```
{
  "Type": ["PH", " NTU ", ...]
}
```

Response Params (JSON format in body)

Name	Type	R/O	Description	Example
Quality	enumint	O	The smaller the category value, the better the water quality. enumint { 1: I Category 2: II Category 3: III Category 4: IV Category 5: V Category }	1

			}	
UploadInfo	object	O	Reported data of water quality detection. Reports all water quality detection data when an alarm is triggered.	
+PH	float	O	pH (1–14)	5.2
+NTU	float	O	Turbidity (0–500 NTU)	200.0
+NH3-N	float	O	Ammonia nitrogen (0–50 mg/L)	20.0
+TN	float	O	Ammonia nitrogen (0–50 mg/L)	25.0
+SD	float	O	Transparency (0–30 m)	20.0
+COD	float	O	Chemical oxygen demand (0–100 mg/L)	50.0
+NN	float	O	Nitrite-nitrogen (0–500 mg/L)	20.0
+DO	float	O	Dissolved oxygen (0–10 mg/L)	5.0
+Chl-a	float	O	Chlorophyll A (0–300 ug/L)	200.0
+TP	float	O	Total phosphorus (0–5 mg/L)	2.5
+CODMn	float	O	Permanganate index (0–100 mg/L)	20.0
+SS	float	O	Floating objects (0–1000 mg/L)	20.0
+BOD5	float	O	Biochemical oxygen demand for 5 days (0–50 mg/L)	25.0
+NO3-N	float	O	Nitrate (0–500 mg/L)	20.0
+TSI	float	O	Eutrophic index (no range)	20.0
+BlackSmellyLevel	enumchar[32]	O	Black and odorous level enumchar[32]{ "Normal" "Light" "Heavy" }	"Light"
FlunkType	enumchar[32][32]	O	Parameter value exceeds its threshold enumchar[32][32]{ "PH": pH value "NTU": Turbidity value "NH3-N": Ammonia nitrogen value }	["PH", "NTU", ...]

Response Example

```
{
  "Quality": 1,
  "UploadInfo": {
    "PH": 5.2,
    "NTU": 200.0,
    "NH3-N": 20.0,
    "TN": 25.0,
    "SD": 20.0,
    "COD": 50.0,
    "NN": 20.0,
    "DO": 5.0,
    "Chl-a": 200.0,
    "TP": 2.5,
    "CODMn": 20.0,
```

```

        "SS": 20.0,
        "BOD5": 25.0,
        "NO3-N": 20.0,
        "TSI": 20.0,
        "BlackSmellyLevel": "Light"
    },
    "FlunkType": ["PH", "NTU", ...]
}

```

15.8.3 Start Find Water Quality Report

Searching for water quality report.

Request URL	http://<server>/cgi-bin/api/WaterDataStatServer/startFind		
Method	POST		
Request Params (JSON format in body)			
Name	Type	R/O	Description
condition	Object	O	Search conditions of water quality detection
+StartTime	char[20]	O	Start time (temporarily accurate to the hour)
+Type	enumchar[32][32]	O	enumchar[32][32]{ "Quality": Water quality category "PH": pH value "NTU": Turbidity value "NH3-N": Ammonia nitrogen value }
+PresetID	int[]	O	Preset ((one-dimensional array))
+EndTime	char[20]	O	End time (temporarily accurate to the hour)
Request Example			
{	"condition": { "StartTime": "2010-05-25 00:00:00", "Type": ["PH", "NTU"], "PresetID": [1], "EndTime": "2010-05-25 00:00:00" } }		

Response Params (JSON format in body)				
Name	Type	R/O	Description	Example
token	uint32	O	Obtained search token	2342343
totalCount	uint32	O	Total number of qualified results	233
Response Example				
{				
"token": 2342343,				

```

        "totalCount": 233
    }

```

15.8.4 Get Find Water Quality Report Search Results

Searching for water quality report.

Request URL	http://<server>/cgi-bin/api/WaterDataStatServer/doFind			
Method	POST			
Request Params (JSON format in body)				
Name	Type	R/O	Description	Example
token	uint32	O	Search token	46878
beginNumber	uint32	O	Starting sequence number of the search. It means that the search starts from the beginNumber records, and returns count records. $0 \leq \text{beginNumber} \leq \text{totalCount}-1$.	0
count	int32	O	Number of traffic statistics for each search	24
Request Example				<i>HTTP API V3.35 - InterStage</i>
{				
"token": 46878,				
"beginNumber": 0,				
"count": 24				
}				

Response Params (JSON format in body)				
Name	Type	R/O	Description	Example
found	uint32	O	Number of searched entries	12
info	object[]	O	Traffic statistics. It is an array and each element represents a traffic record that satisfies the condition.	
+StartTime	char[20]	O	Start time	2018-03-03 00:00:00",
+Quality	enumint	O	enumint { 1: I Category 2: II Category 3: III Category 4: IV Category 5: V Category }	1
+PH	float	O	pH (1–14)	5.5
+NTU	float	O	Turbidity (0–500 NTU)	250.0
+NH3-N	float	O	Ammonia nitrogen (0–50 mg/L)	25.0
+TN	float	O	Total nitrogen (0–50 mg/L)	25.0
+SD	float	O	Transparency (0–30 m)	20.0
+COD	float	O	Chemical oxygen demand (0–100 mg/ L)	20.0
+NN	float	O	Nitrite-nitrogen (0–500 mg/ L)	250.0
+Chl-a	float	O	Chlorophyll A (0–300 ug/ L)	20.0

+TP	float	O	Total phosphorus (0–5 mg/L)	2.5
+CODMn	float	O	Permanganate index (0–100 mg/L)	20.0
+SS	float	O	Floating objects (0–1000 mg/L)	500.0
+BOD5	float	O	Biochemical oxygen demand for 5 days (0–50 mg/L)	25.0
+NO3-N	float	O	Nitrate (0–500 mg/L)	250.0
+TSI	float	O	Eutrophic index (no range)	25.0
+BlackSmellyLevel	enumchar[32]	O	Black and odorous level enumchar[32]{ "Normal" "Light" "Heavy" }	"Light"
+DO	float	O	Dissolved oxygen (0–10 mg/L)	20.0

Response Example

```
{
  "found": 12,
  "info": [
    {
      "StartTime": "2012-03-14 00:00:00",
      "Quality": 1,
      "PH": 5.5,
      "NTU": 250.0,
      "NH3-N": 25.0,
      "TN": 25.0,
      "SD": 20.0,
      "COD": 20.0,
      "NN": 250.0,
      "Chl-a": 20.0,
      "TP": 2.5,
      "CODMn": 20.0,
      "SS": 500.0,
      "BOD5": 25.0,
      "NO3-N": 250.0,
      "TSI": 25.0,
      "BlackSmellyLevel": "Light",
      "DO": 20.0
    }, ...
  ]
}
```

15.8.5 Stop Find Water Quality Report

Stopping the search of water quality detection database.

Request URL	http://<server>/cgi-bin/api/WaterDataStatServer/stopFind		
Method	POST		
Request Params (JSON format in body)			
Name	Type	R/O	Description
token	uint32	O	Search token
			Example
			46284

Request Example

```
{
    "token": 46284
}
```

Response Params (JSON format in body)

Name	Type	R/O	Description	Example
Response Example				
{}				

15.9 Advertisement

15.9.1 Modifying the Display Duration of Image Advertisements

Modify the display duration of image advertisements.

Request URL http://<server>/cgi-bin/api/VideoOutput/changeSustain

Method POST

Request Params (JSON format in body)

Name	Type	R/O	Description	Example
Sustain	int	R	Display duration of each image	20

Request Example

```
{
    "Sustain": 20
}
```

Response Params (JSON format in body)

Name	Type	R/O	Description	Example
Response Example				
{}				

15.9.2 Releasing Advertisements

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	Number of the configuration information of the video output port	0
Number	int	O	Number of the current advertisement plan. Call the parameter to configure different advertisement plans.	0
TimeSection	TimeSchedule	O	Advertisement playing period. It is a two-dimensional array in which the first seven elements correspond to seven days of a week, and the eighth element corresponds to holidays, with a	[["1 00:00:00-07:00:00", "2 09:00:00-17:30:00", "3

			maximum of six periods per day. The eighth element can be left blank or entered as null, indicating that the holiday period is not supported.	17:30:00-23:59:59"], ..., [],]
Enable	bool	O	Enable the play	true
Name	char[128]	O	Advertisement name	"Happy National Day"
StartTime	char[20]	O	Start time of the play	"2016-08-10 10:00:00"
EndTime	char[20]	O	End time of the play Effective only for loop play	"2016-08-10 12:00:00"
Mode	char[16]	O	Video play mode Once: Only play each video once Repeat: Loop play the videos until the Endtime Alone: Cut-in play (solely)	"Once"
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 files Remove: Delete files Clear: Clear the files	"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 publishing mode 3: Home page mode The files are played only under the corresponding 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	Address of the file extension resource: The original URL field is too short, and not enough for the supporting ICC	"http://10.35.81.187:8927/1952fdd7-48e8-11eb-b015-08eded28a

			<p>platform (currently it has reached 160 bytes), which needs to be increased.</p> <p>Compatibility logic of platform: When the Support field of the device capability interface</p> <p><code>VideoOutput.getAdvertisementCaps</code> returns true, the real download address is filled in the URLEx field, and the URL field is filled with an empty string.</p> <p>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.</p>	344/20210315/1/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 seven elements correspond to seven days of a week, and the eighth element corresponds to holidays, with a maximum of six 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"], ..., [],]
+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 No.	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 National Day",
  "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://10.35.81.187:8927/1952fd7-48e8-11eb-b015-08ed28a344/20210315/1/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				
{}				

15.9.3 Getting the List of Advertisement Resources

Get the list of advertisement resources

Request URL	http://<server>/cgi-bin/api/VideoOutput/list										
Method	POST										
Request Params (JSON format in body)											
<table border="1"> <thead> <tr> <th>Name</th><th>Type</th><th>R/O</th><th>Description</th></tr> </thead> <tbody> <tr> <td>path</td><td>char[260]</td><td>R</td><td>Path to get the files</td></tr> </tbody> </table>				Name	Type	R/O	Description	path	char[260]	R	Path to get the files
Name	Type	R/O	Description								
path	char[260]	R	Path to get the files								
Request Example											
<pre>{ "path": "/PublishFilePath/2010/8/11/dav" }</pre>											

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 (Effective when Type="File")	
++FileType	char[64]	O	Type of the file node "Wireshark/tcpdump", see Wireshark capture file type	"Wireshark/tcpdump"
++CreateTime	char[20]	O	Time of creation	"2010-4-15 9:58:32"

++ModifyTime	char[20]	O	Time of modification	"2010-4-15 9:58:32"
++Size	double	O	File size, in bytes. The figures after the decimal point do not make sense.	1873.0
++path	char[260]	O	Relative path	"/PublishFilePath/2010/8/11/dav.jpg"
++Desc	char[128]	O	Custom description of the file	"xxxxxxx"
+directory	object	O	Directory information, effective when the type is Directory.	
++CreateTime	char[20]	O	Time of creation, in the form of 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"
      }
    }, ...
  ]
}
```

15.9.4 Searching for the Released Advertisements

Used with deliveryFile for intercom in buildings

Request URL	http://<server>/cgi-bin/api/VideoOutput/queryDeliveredFile		
Method	POST		
Request Params (JSON format in body)			
Name	Type	R/O	Description
Request Example			
{}			

Response Params (JSON format in body)			
Name	Type	R/O	Description
Info	Object[]	R	Information of the transferred file
+Enable	bool	O	Enable the play
Example			
Info	[{"name": "file", "path": "/PublishFilePath/2010/8/11/dav.jpg", "size": 1873.0, "modifyTime": "2010-4-15 9:58:32", "createTime": "2010-4-15 9:58:32", "desc": "xxxxxxx"}]		
+Enable	true		

+Number	int	R	Number of the current advertisement plan. Call the parameter to configure different advertisement plans.	0
+TimeSection	TimeSchedule	R	Advertisement playing period. It is a two-dimensional array in which the first seven elements correspond to seven days of a week, and the eighth element corresponds to holidays, with a maximum of six 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 National Day"
+StartTime	char[20]	O	Start time of the play	"2016-08-10 10:00:00"
+EndTime	char[20]	O	End time of the play	"2016-08-10 12:00:00"
+Mode	char[16]	O	Video play mode Once: Only play each video once Repeat: Loop play the videos until the Endtime Alone: Cut-in play (solely)	"Once"
+PlayTimes	int	O	Playing times When in "Once" mode.	0
+FileList	object[]	O	List of the video files	Supports up to 20 videos
++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	Path that the file is saved to the device	"/PublishFilePath/1.dav"
++Sustain	int	O	Display duration of each image, effective only when FileType is Image. Unit: Second	5
++URL	char[512]	O	File resource address, and the maximum length is 512 bytes. It has the same length requirement with VideoOutput.deliveryFile (release at the same time for intercom in buildings).	"ftp://192.168.1.108/1.dav"
++TimeSection	TimeSchedule	O	Advertisement playing period. It is a two-dimensional array in which the first seven elements correspond to seven days of a week, and the eighth element corresponds to holidays, with a	[["1 00:00:00-07:00:00", "1 09:00:00-17:30:00", "1 17:30:00-23:59:59"], ..., [],]

			maximum of six periods per day. The eighth element can be left blank or entered as null, indicating that the holiday period is not supported.	
++Status	int	O	File status, 0: normal; 1: not support file; 2: file format error; 3: file damaged; 4: file too large; 5: file too small; 6: file already removed; 7: file in deleting; 8: file saved; 9: file in saving; 10: file in editing	1
++Size	int	O	Files size, in byte.	102400
++ID	int	O	File No.	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 National Day ",
      "StartTime": "2016-08-10 10:00:00",
      "EndTime": "2016-08-10 12:00:00",
      "Mode": "Once",
      "FileList": [{"Up to 20
        "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
      }]
    }
  ]
}
```

```
 },...{}]  
 },{},...{}])
```

15.9.5 Deleting Advertisement Resources

Delete the advertisement 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][12 8]	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

```
{}
```