HTTP API Protocol

Version 1.12

Version History

Version	Date	Notes
1.0	2025.07.17	Document created
1.1	2025.07.18	1. Update 2.4.3 Modified FileType to FileType_n field; file type must correspond to each file 2. Update 2.4.3 Modified AudioFile_n to AudioFileUrl_n field; file URL should be full path 3. Update 2.4.8 Added FileType field
1.2	2025.11.26	1. Added OSD functionality to version 2.4.10 2. A more detailed description of the audio upload functionality, including the HTTP headers and data, and code examples.
1.3	2025.12.09	1. Added SIP functionality in section 2.5.8
1.4	2026.01.16	1. Added Download audio file in section 2.4.6 2. Update API Alarm Input in section 2.6.4
1.5	2026.03.19	1. Added Alarm Sensor Status in section 2.6.2
1.6	2026.04.08	1. Added Profile Alarm in section 2.6.3
1.7	2026.05.09	1. Added Idle API in section 2.6.7
1.8	2026.06.25	1. Added Log Query in section 2.7.4 2. Added SIP Answer Type in section 2.5.8 3. Added RTSP Answer Type in section 2.5.9
1.9	2026.07.30	1. Added Get TTS Config in section 2.4.13 2. Added Get TTS File in section 2.4.14 3. Added Upload/Test TTS File in section 2.4.15 4. Added Delete TTS file in section 2.4.16 5. Added Download TTS file in section 2.4.17 6. Added Test Speaker Check in section 2.4.18
1.10	2026.08.14	7. Added Get/Set SIP P2P Parameters in section 2.5.9 8. API Alarm Input in section 2.6.6 Add TTS and Speed Parameters
1.11	2026.09.03	1. Added Edit TTS file in section 2.4.18 2. Added Play audio API in section 2.4.7 3. Added Get/Set Heartbeat Parameters in section 2.5.11
1.12	2026.10.14	1. Added BeginDate,EndDate,EveryYear in section 2.4.10/2.4.11 2. Added NAT in section 2.5.10 3. Added Get/Set Multicast Parameters in section 2.5.12 4. Added Configure MQTT in section 4.4

Directory

Version History 2
1 Message Format 5
1.1 URL format 5
1.2 Example 6
1.3 Return Value 6
2 API interface 8
2.1 Login 8
2.2 Logout 9
2.3 Get device information 9
2.4 Media Parameters 10
2.4.1 Get Audio Encoding Parameters 10
2.4.2 Get/Set Audio Parameters 11
2.4.3 Get Audio File 12
2.4.4 Upload Audio File 12
2.4.5 Delete audio file 14
2.4.6 Download audio file 15
2.4.7 Play audio API 15
2.4.8 Get LED front and back light color configuration 16
2.4.9 Get LED front and back light display mode Configuration 16
2.4.10 Get/Set LED Idle Schedule Configuration 17
2.4.11 Get/Set LED Alarm Schedule Configuration 20
2.4.12 Get/Set OSD Parameters 23
2.4.13 Testing speaker sound 24
2.4.14 Get TTSConfig 25
2.4.15 Get TTS File 25
2.4.16 Upload/Test TTS File 26
2.4.17 Delete TTS file 26
2.4.18 Download TTS file 27
2.4.19 Edit TTS file 27
2.4.20 Check Speaker 27
2.5 Network Parameters 28
2.5.1 Get/Set Network Parameters 28
2.5.2 Get/Set Email Parameters 29
2.5.3 Get/Set SNMP Parameters 30
2.5.4 Get/Set WiFi Parameters 31
2.5.5 Get WiFi network parameters 31
2.5.6 Get WiFi list 32
2.5.7 Get/Set HTTPs Parameters 33
2.5.8 Get/Set SIP Parameters 33
2.5.9 Get/Set SIP P2P Parameters 34
2.5.10 Get/Set Other Parameters 35
2.5.11 Get/Set Heartbeat Parameters 36

2.5.12 Get/Set Multicast Parameters 37
2.6 Event 39
2.6.1 Get/Set Probe Alarm Parameters 39
2.6.2 Get/Set Probe Alarm Sensor Status 39
2.6.3 Get/Set Profile Alarm Parameters 40
2.6.4 Get/Set System Log Parameters 43
2.6.5 Query Alarm Log 43
2.6.6 API Alarm Input 44
2.6.7 API Idle Input 49
2.7 System Parameters 53
2.7.1 Get/Set NTP 53
2.7.2 Restore Factory Settings 53
2.7.3 Device Reboot 54
2.7.4 Log Query 54
2.8 User Management 56
2.8.1 User Configuration 56
2.8.2 Add User 56
2.8.3 Delete User 57
2.8.4 Reset User Password 57
3 Device Search 58
3.1 Device Search 58
3.2 Modify device information 59
4 Alarm Linkage 60
4.1 Configure HTTP integration 60
4.2 Configure Broadcast Link 61
4.3 Alarm Content 61
4.4 Configure MQTT 62

1 Message Format

1.1 URL format

http://<host>[:port][abs_path]

Host: Device IP address.

Port: If the port number is omitted, the default port is used.
HTTP port defaults to 80.
HTTPs port defaults to 443.

abs_path: The URL address.

1.2 Example

POST /ISAPI/DeviceInfo?ID=3088 HTTP/1.1

Accept: text/plain, */*; q=0.01

X-Requested-With: XMLHttpRequest

Referer: http://192.168.0.112/BaseInforPage.html

Accept-Language: zh-CN

Accept-Encoding: gzip, deflate

User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; Trident/7.0; rv:11.0) like Gecko

Host: 192.168.0.112

Content-Length: 0

Connection: Keep-Alive

Cache-Control: no-cache

HTTP/1.1 200 OK

Server: GoAhead-Webs/2.5.0

Connection: keep-alive

Content-Length: 256

Content-Type: application/xml

<?xml version="1.0" encoding="utf-8"?>

<DeviceInfo>

<DeviceName>StrobeSpeaker</DeviceName>

<DeviceID>2507150000</DeviceID>

<DeviceModel></DeviceModel>

<SoftwareVer>8C-1F-64-A3-6B-A9</SoftwareVer>

<FactoryInfo>StrobeSpeaker</FactoryInfo>

</DeviceInfo>

1.3 Return Value

The data returned from the server is as follows, with all data returned in HTTP format.

Syntax: HTTP/1.1<HTTP code><HTTP text>

Example: HTTP/1.1 200 OK

HTTP/1.1 404 File Not Found

The following HTTP codes and their meanings

Status	HTTP Text	Description

Code		
200	OK	The request was successful, but an application error may still occur, which will be returned as an application error code.
204	No Content	The server has completed the request but has no new information to return.
400	Bad Request	The request is invalid or cannot be fulfilled. The request requires user authentication or authorisation, which has been denied.
401	Unauthorised	The request requires authentication, but cannot be completed due to a conflict with the current state.
404	Not Found	The server could not find any content matching the request.
409	Conflict	The request could not be completed due to a conflict with the current state of the resource.
500	Internal Error	The server encountered an unexpected situation and was unable to complete the request.
503	Service	The server is temporarily overloaded and unable to process the request.

HTTP/1.1 400 Bad Request

Server: GoAhead-Webs/2.5.0

Connection: keep-alive

Content-Length: 218

Content-Type: application/xml

<?xml version="1.0" encoding="utf-8"?>

<ResponseStatus>

<RequestURL>/ISAPI/AudioInfo/AudioEncodeOptions?ID=3088</RequestURL>

<StatusCode>401</StatusCode>

<StatusString>Invalid Channel number</StatusString>

</ResponseStatus>

Status Code Range	Status Code	Status Description
2XX	200	Success.
4XX	401	Invalid channel number.
	402	Invalid parameter value.
	403	Invalid parameter name.
	501	The server does not support this feature.
	502	The server supports the feature, but it is unavailable in the current

5XX		mode.
	503	Server error.
	504	Failed to open NTP server.
	505	Failed to close NTP server.

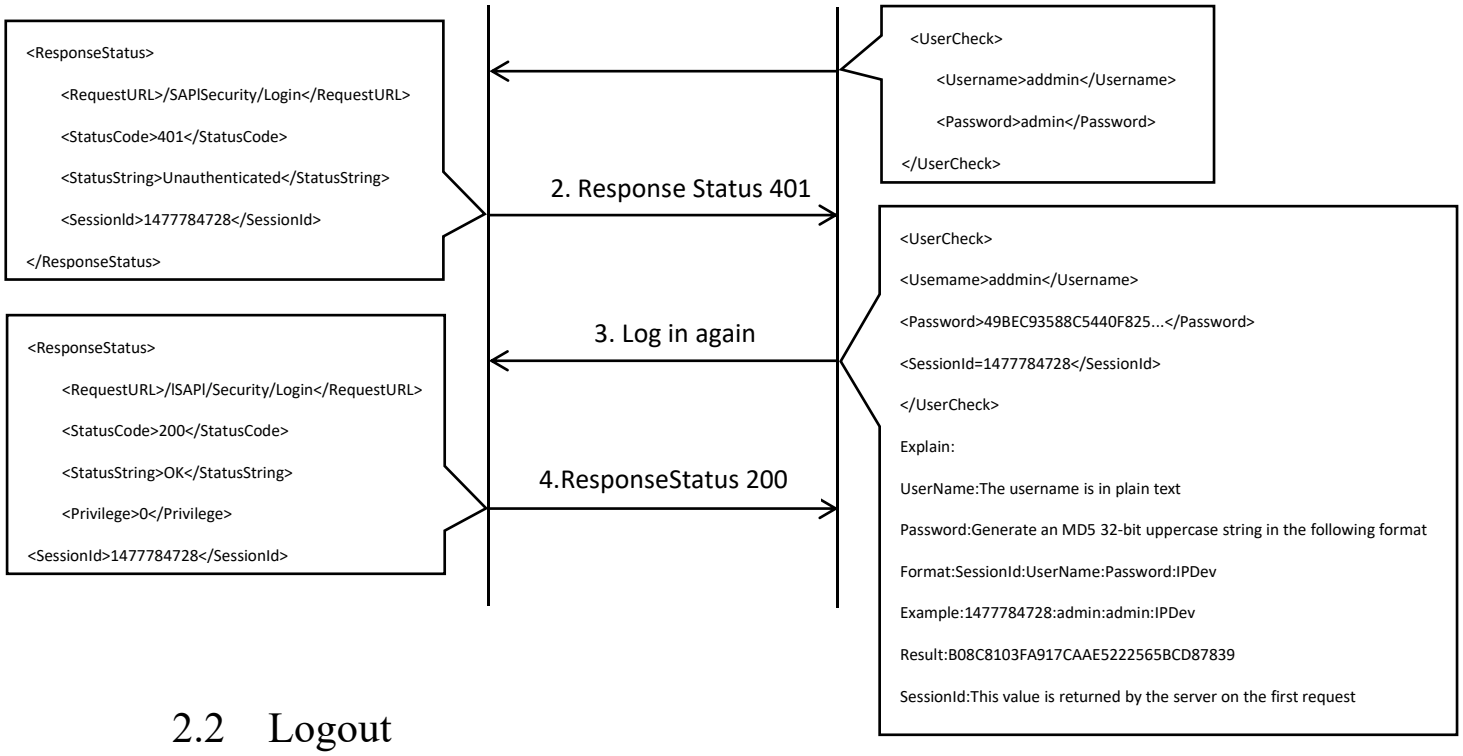
2 API interface

2.1 Login

/ISAPI/Security/ Login			
POST			
Description	Verify the username and password by sending the following XML message via a POST request. The first login requires authentication; subsequent logins require the encrypted content to be included.		
Field	Type	Description	Required
Username	string	Username	Yes
Password	string	Password	Yes
SessionId	string	Session ID	No
If authentication succeeds, return 200 OK			
<UserCheck> <Username><!-- req, xs: string --></Username> <Password><!-- req, xs: string --></Password> <SessionId><!-- required, xs: string --></SessionId> </UserCheck>			

Note: Following successful authentication, all communication messages must include the SessionId; otherwise, normal communication cannot proceed. Should no messages be exchanged within 30 minutes, the session will time out and disconnect, requiring re-authentication. For example: /ISAPI/DeviceInfo?ID=1477784728. The authentication process is as follows:

WEB Server WEB Client



2.2 Logout

/ISAPI/Security/Logout?ID=SessionID			
POST			
Description	Logout message. POST request with the following XML message content. Normal response returns 200 OK.		
Field	Type	Description	Required
SessionID	string	Session ID	Yes
If authentication succeeds, return 200 OK			
<UserCheck> <SessionID><!-- req, xs: string --></SessionID> </UserCheck>			

2.3 Get device information

/ISAPI/DeviceInfo?ID=SessionID			
POST			
Description	Get device information. Send an empty POST request. Upon successful completion, the following XML message content will be returned:		
Field	Type	Description	Required
DeviceName	string	Device Name	No
DeviceID	int	Device ID	No
DeviceModel	string	Device Model	No
SoftwareVer	string	Software Version	No

MACAddress	string	MAC Address	No
IPAddress	string	IP Address	No
FactoryInfo	string	Manufacturer Information	No

```
<DeviceInfo>
   <DeviceName><!-- req, xs: string --></DeviceName>
   < DeviceID ><!-- req, xs: int --></DeviceID>
   < DeviceModel ><!-- req, xs: string --></DeviceModel>
   <SoftwareVer><!-- req, xs: string -- ></SoftwareVer>
   <MACAddress><!-- req, xs: string -- ></MACAddress>
   <IPAddress><!-- required, xs: string -- ></IPAddress>
  < FactoryInfo ><!-- req, xs: string --></FactoryInfo>
</DeviceInfo>
```

2.4 Media Parameters

2.4.1 Get Audio Encoding Parameters

/ISAPI/AudioInfo/AudioEncodeOptions?ID=SessionID			
POST			
Description	Get multi-channel audio encoding Parameters, such as G711A and G711U. Send an empty POST request; a normal response will contain the following XML message content		
Field	Type	Description	Required
Encoding	string	G711A/G711U//AAC	No
BitrateList	int	Bitrate	No
Items	int	Sample rate	No

```
<GetAudioEncodeOptions>
   <Options>
      <Encoding><!-- req, xs: string --></Encoding>
      <BitrateList>
            <Items><!-- req, xs: int --></Items>
      </BitrateList>
      < SampleRateList >
            <Items><!-- req, xs: int --></Items>
      </SampleRateList>
   </Options>
   <Options>
      <Encoding><!-- req, xs: string --></Encoding>
      <BitrateList>
            <Items><!-- req, xs: int --></Items>
      </BitrateList>
      <SampleRateList>
```

```
            <Items><!-- req, xs: int --></Items>
        </SampleRateList>
    </Options>
<GetAudioEncodeOptions>
```

2.4.2 Get/Set Audio Parameters

/ISAPI/AudioInfo/AudioConfig?ID=SessionID			
POST			
Description	Get audio encoding parameters. POST request with empty message body. Normal response returns the following XML message content		
PUT			
Description	Set audio encoding parameters. Send the following XML message via a PUT request. A normal response returns 200 OK.		
Field	Type	Description	Required
PCMicEnabled	int	0: Disabled 1: Enabled	No
DevMicEnabled	int	0: Disabled 1: Enabled	No
LineInEnabled	int	0: Disabled 1: Enabled	No
Encoding	string	G711A/G711U/G726/AAC	No
Bitrate	int	bitstream	No
Sample Rate	int	Sample Rate	No
VolumeMax	int	Volume Maximum	No
VolumeMin	int	Minimum Volume	No
AudioInVolume	int	Input Volume	No
AudioOutVolume	int	Output Volume	No

```
<AudioConfig>
    <PCMicEnabled><!-- req, xs: int --></PCMicEnabled>
    <DevMicEnabled><!-- req, xs: int --></DevMicEnabled>
    <LineInEnabled><!-- req, xs: int --></LineInEnabled>
    <Encoding><!-- required, xs: string --></Encoding>
    <Bitrate><!-- required, xs: integer --></Bitrate>
    <SampleRate><!-- required, xs: integer --></SampleRate>
    <VolumeMax><!-- req, xs: int --></VolumeMax>
    <VolumeMin><!-- required, xs: integer --></VolumeMin>
    <AudioInVolume><!-- required, xs: integer --></AudioInVolume>
    <AudioOutVolume><!-- required, xs: integer --></AudioOutVolume>
</AudioConfig>
```

2.4.3 Get Audio File

/ISAPI/AudioInfo/GetDelAudioFile?ID=SessionID			
POST			
Description	Get audio file. POST request with empty message body. Normal response returns the following XML message content		
Field	Type	Description	Required
FileCount	int	Number of files	Yes
FileName_n	string	File name	Yes
FileType_n	int	0: System default file 1: Customer-defined file	No
Field Description FileName_n:1-50			
<GetDelAudioFile> <FileCount><!-- req, xs: int --></FileCount> <FileName_1><!-- req, xs: string --></FileName_1> <FileType_1><!-- required, xs: integer --></FileType_1> … <FileName_n><!-- req, xs: int --></FileName_n> <FileType_n><!-- required, xs: integer --></FileType_n> </GetDelAudioFile>			

2.4.4 Upload Audio File

/ISAPI/UploadFile/AudioFile?offset=0&name=base64?ID=SessionID			
POST			
Description	Upload audio file. POST request with empty XML message content. All parameters in URL. Normal response: 200 OK		
Field	Type	Description	Required
offset	int	File offset, starting from zero, incremented by the length of each transmitted packet	Yes
name	string	Base64 filename	Yes
The following HTTP content consists of HTTP headers and data, with the first packet having an offset of 0, the next packet having an offset of 20480, and so on, continuously looping and accumulating until the end: POST /ISAPI/UploadFile/AudioFile?name=S25vdzExMTExLm1wMw==&offset=0&id=1761825777 HTTP/1.1 Host: 192.168.1.88			

Connection: keep-alive

Content-Length: 20480

User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/142.0.0.0 Safari/537.36 Edg/142.0.0.0

Accept: */*

Origin: http://192.168.1.88

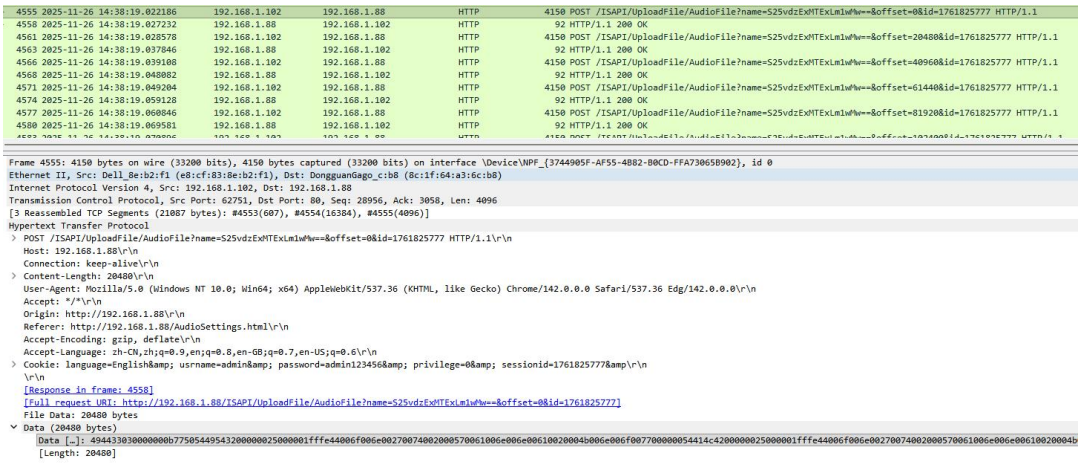
Referer: http://192.168.1.88/AudioSettings.html

Accept-Encoding: gzip, deflate

Accept-Language: zh-CN,zh;q=0.9,en;q=0.8,en-GB;q=0.7,en-US;q=0.6

Cookie: language=English& usrname=admin& password=admin123456& privilege=0& sessionid=1761825777&amp

Data(20480) Byte



Code example:

```
function uploadAudioFile(file, index)
{
	var reader = new FileReader();
	reader.readAsArrayBuffer(file);
	reader.onload = function(ev)
	{
		var dataURL = ev.target.result;
		var base64 = Base64.encode(file.name);
		sendAudioFileData('/ISAPI/UploadFile/AudioFile', base64, new Uint8Array(dataURL), 20480, index);
	}
}

var sendAudioFileData = function(path, name, data, chunkSize, index)
{
	var sendChunk = function(offset)
	{
		var chunk = data.subarray(offset, offset + chunkSize) || '';
```

```
            var opts = {method: 'POST', body: chunk};
            var sessionid = $.GetCookie('sessionid');
            var url = path + '?name=' + name + '&offset=' + offset + '&id=' + sessionid;
            fetch(url, opts).then(function(res)
            {
                if (res.status == "200" && res.statusText == "OK")
                {
                    if (chunk.length > 0)
                    {
                        sendChunk(offset + chunk.length);
                    }
                    else
                    {
                        layui.use('layer', function()
                        {
                            var layer = layui.layer;
                            layer.msg(str_txtUploadSuccess, {icon: 1, time: 2000, offset:
['100px', '30%'], });
                        });
                    }
                }
            });
        };
        sendChunk(0);
};
```

2.4.5 Delete audio file

/ISAPI/AudioInfo/GetDelAudioFile?ID=SessionID			
PUT			
Description	Delete audio file. The PUT request contains the following XML message content. Normal response: 200 OK		
Field	Type	Description	Required
FileName	string	File name	Yes

```
<GetDelAudioFile>
    <FileName><!-- req, xs: string --></FileName>
</GetDelAudioFile>
```

2.4.6 Download audio file

/ISAPI/DownloadFile/AudioFile?ID=SessionID			
POST			
Description	Download audio file. The POST request contains the following XML message content. Normal response: 200 OK		
Field	Type	Description	Required
FileName	string	File name	Yes
Request XML <DownloadFile> <FileName><!-- reg, xs:string --></FileName> </DowloadFile> **Response** HTTP header + Data The message content is as follows: HTTP/1.1 200 OK Content-Type: application/octet-stream Connection: keep-Alive Content-Length: 24654 RIFFF`..WAVEfmt .....@....>.....LIST....INFOISFT....Lavf59.16.100.data.`.....			

2.4.7 Play audio API

ISAPI/Event/UrlPlayAudio?			
POST			
Description	Play audio directly via the API; no session ID is required, using the slot method.Normal response: 200 OK		
Field	Type	Description	Required
PlayEnable	int	0: stop 1: play	Yes
FileType	int	0: Default audio 1: Custom audio	Yes
FileIndex	int	Slot number in this list, starting from 1	Yes
Example: POST/GET ISAPI/Event/UrlPlayAudio Authorization: Basic YWRtaW46YWRtaW4xMjM0NTY= Host: 192.168.1.88			

http://192.168.1.88/ISAPI/Event/UrlPlayAudio?PlayEnable=1&FileType=0&FileIndex=1

2.4.8 Get LED front and back light color configuration

/ISAPI/LED/ColorConfig?ID=SessionID			
POST			
Description	Get LED front and back light color configuration. POST request with the following XML message content		
Field	Type	Description	Required
FrontBackLight	int	0:FrontLight 1: back Light	No
ColorCount	int	Number of lamp colors	No
ColorType_n	int	1:Redcolor 2:Green 3: Blue color 4:White 5:Yellow 6:Violet 7:Cyan	No
Field Description colorType_n:1-n			
Request XML <ColorConfig> <FrontBackLight><!-- req, xs: int --></FrontBackLight> </colorConfiguration> **Response XML** <ColorConfig> <FrontBackLight><!-- req, xs: int --></FrontBackLight> <colorCount><!-- req, xs: int --></colorCount> <colorType_1><!-- req, xs: int --></colorType_1> ... <ColorType_n><!-- required, xs: integer --></ColorType_n> </ColorConfig>			

2.4.9 Get LED front and back light display mode Configuration

/ISAPI/LED/DisplayModeConfig?ID=SessionID

POST			
Description	Get LED front and back light display mode configuration. The POST request contains the following XML message content		
Field	Type	Description	Required
FrontModeCount	int	Number of display modes	No
FrontModeType_n	int	1: Steady 2:One-Second Blink (illuminates 10 times) 3: Two-Second Blink (illuminates 10 times) 4:Five-Second Blink (illuminates 5 times) 5:Breathe 6:Strobe	No
BackModeCount	int	Number of display modes	No
BackModeType_n	int	1: Steady 2:Chase	No
Field Description ModeType_n:1-n			
<DisplayModeConfig> <FrontModeCount><!-- req, xs: int --></FrontModeCount> <FrontModeType_1><!-- req, xs: int --></FrontModeType_1> ... <FrontModeType_n><!-- required, xs: integer --></FrontModeType_n> <BackModeCount><!-- req, xs: int --></BackModeCount> <BackModeType_1><!-- req, xs: int --></BackModeType_1> ... <BackModeType_n><!-- required, xs: integer --></BackModeType_n> </DisplayModeConfig>			

2.4.10 Get/Set LED Idle Schedule Configuration

/ISAPI/LED/IdleScheduleConfig?ID=SessionID			
POST			
Description	Get LED idle schedule configuration. POST request with empty message body. Normal response returns the following XML message content		
PUT			
Description	Set LED idle schedule configuration. PUT request with the following XML message content. Normal response: 200 OK		
Field	Type	Description	Required
ScheduleCount	int	Number of Schedules	No

ScheduleID	int	ScheduleID (range 0-4)	Yes
ScheduleName	string	Schedule Name	No
ScheduleEnable	int	0: Disabled 1: Enabled	No
PlayEnable	int	0: Disabled 1: Enabled	No
PlayFile	string	File name	No
Volume	int	Volume (0-100)	No
CycleMode	int	Cycle Mode: 0: Once only 1: Multiple times 2: Infinite	No
PlayTimes	int	Configure playback count when selecting multiple times	No
Interval	int	Playback interval time, in seconds	No
FrontLightEnable	int	0: Disabled 1: Enabled	No
FrontLightColor1	int	1:RedColor 2: GreenColor 3:BlueColor 4:WhiteColor 5:Yellow 6:Violet 7:Cyan	No
FrontLighColor2	int	0:None 1:RedColor 2: GreenColor 3: BlueColor 4:WhiteColor 5:Yellow 6:Violet 7:Cyan	No
FrontLightDisplayMode	int	Display Mode 1: Steady 2: One-Second Blink (illuminates 10 times) 3: Two-Second Blink (illuminates 10 times) 4: Five-Second Blink (illuminates 5 times) 5: Breathe 6:Strobe	No
FrontLightBrightness	int	Only the following colors can be set to brightness levels 0-255 (low	No

		to high): 1:RedColorr 2: GreenColor 3: BlueColor 4:WhiteColor	
FrontLightFrequency	int	Only Strobe mode allows setting the flashing frequency in milliseconds: default 100	No
BackLightEnable	int	0: Disabled 1: Enabled	No
BackLightColor	int	1:RedColor 2: GreenColor 3:BlueColor 4:WhiteColor 5:Yellow 6:Violet 7:Cyan	No
BacklightDisplayMode	int	Display Mode 1: Steady 2:Chase	No
BacklightBrightness	int	Only the following colors can be set to brightness levels 0-255 (low to high): 1:RedColor 2: GreenColor 3: BlueColor 4:WhiteColor	No
BacklightFrequency	int	Only Strobe mode can set the flashing frequency, in milliseconds: default 100	No
WeekDays	string	Select the time period for accepting alarms. For example, 0123456 indicates Sunday to Saturday.	No
BeginDate	string	Start Date(YY-MM-DD)	No
EndDate	string	End Date(YY-MM-DD)	No
EveryYear	int	EveryYear	No
BeginTime	string	Start time (HH:MM:SS)	No
EndTime	string	End time (HH:MM:SS)	No
Field Description ScheduleCount_n:1-5			
<IdleScheduleConfig> <ScheduleCount><!-- req, xs: int --></ScheduleCount> <ScheduleCount_1>			

```
        <ScheduleID><!-- req, xs: int --></ScheduleID>
        <ScheduleName><!-- required, xs: string --></ScheduleName>
        <ScheduleEnable><!-- req, xs: int --></ScheduleEnable>
        <PlayEnable><!-- required, xs: integer --></PlayEnable>
        <PlayFile><!-- req, xs: string --></PlayFile>
        <Volume><!-- required, xs: integer --></Volume>
        <CycleMode><!-- req, xs: int --></CycleMode>
        <PlayTimes><!-- req, xs: int --></PlayTimes>
        <Interval><!-- required, xs: integer --></Interval>
        <FrontLightEnable><!-- required, xs: integer --></FrontLightEnable>
       <FrontLightColor1><!-- req, xs: int --></FrontLightColor1>
        <FrontLightColor2><!-- req, xs: int --></FrontLightColor2>
        <FrontLightDisplayMode><!-- req, xs: int --></FrontLightDisplayMode>
        <FrontLightBrightness><!-- required, xs: integer --></FrontLightBrightness>
        <FrontLightFrequency><!-- req, xs: int --></FrontLightFrequency>
        <BackLightEnable><!-- required, xs: integer --></BackLightEnable>
        <BackLightcolor><!-- req, xs: int --></BackLightcolor>
        <BackLightDisplayMode><!-- required, xs: integer --></BackLightDisplayMode>
        <BackLightBrightness><!-- required, xs: integer --></BackLightBrightness>
        <BackLightFrequency><!-- required, xs: integer --></BackLightFrequency>
        <WeekDays><!-- required, xs: string --></WeekDays>
        <BeginDate><!-- req, xs: string --></BeginDate>
        <EndDate><!-- required, xs: string --></EndDate>
          <Every Year><!-- required, xs: int --></Every Year>
        <BeginTime><!-- required, xs: string --></BeginTime>
        <EndTime><!-- required, xs: string --></EndTime>
    </ScheduleCount_1>
</IdleScheduleConfig>
```

2.4.11 Get/Set LED Alarm Schedule Configuration

/ISAPI/LED/AlarmScheduleConfig?ID=SessionID			
POST			
Description	Get LED alarm schedule configuration. POST request with empty message body. Normal response returns the following XML message content		
PUT			
Description	Set LED alarm schedule configuration. PUT request with the following XML message content. Normal response: 200 OK		
Field	Type	Description	Required
ScheduleCount	int	Number of schedules	No
ScheduleID	int	ScheduleID (range 0-4)	Yes

ScheduleName	string	Schedule Name	No
ScheduleEnable	int	0: Disabled 1: Enabled	No
AlarmOutEnable	int	0: Disabled 1: Enabled	No
AlarmOutDuration	int	Alarm output duration per trigger, default 60 seconds	No
PlayEnable	int	0: Disabled 1: Enabled	No
PlayFile	string	File name	No
Volume	int	Volume	No
FrontLightEnable	int	0: Disabled 1: Enabled	No
FrontLightColor1	int	1:RedColor 2:GreenColor 3:BlueColor 4:WhiteColor 5:Yellow 6:Violet 7:Cyan	No
FrontLightColor2	int	0:None 1:RedColor 2: GreenColor 3: BlueColor 4:WhiteColor 5:Yellow 6:Violet 7:Cyan	No
FrontLightDisplayMode	int	Display Mode 1: Steady 2: One-Second Blink (illuminates 10 times) 3: Two-Second Blink (illuminates 10 times) 4: Five-Second Blink (illuminates 5 times) 5: Breathe 6:Strobe	No
FrontLightBrightness	int	Only the following colors can be set to brightness levels 0-255 (low to high): 1:Redcolor 2: Green	No

		3: Blue 4:Whitecolor	
FrontLightFrequency	int	Only Strobe mode allows setting the flashing frequency, in milliseconds: default 100	No
BackLightEnable	int	0: Disabled 1: Enabled	No
BackLightColor	int	1:RedColor 2:GreenColor 3:BlueColor 4:WhiteColor 5:Yellow 6:Violet 7:Cyan	No
BacklightDisplayMode	int	Display Mode 1:Steady 2:Chase	No
BacklightBrightness	int	Only the following colors can be set to brightness levels 0-255 (low to high): 1:RedColor 2: GreenColorr 3: BlueColor 4:WhiteColor	No
BacklightFrequency	int	Only Strobe mode can set the flashing frequency, in milliseconds: default 100	No
HttpMsgEnabled	int	0: Disabled 1: Enabled	No
EmailEnabled	int	0: Disabled 1: Enabled	No
BroadcastEnabled	int	0: Disabled 1: Enabled	No
WeekDays	string	Select the time period for accepting alerts. For example, 0123456 indicates Sunday to Saturday.	No
BeginDate	string	Start Date(YY-MM-DD)	No
EndDate	string	End Date(YY-MM-DD)	No
EveryYear	int	EveryYear	No
BeginTime	string	Start time (HH:MM:SS)	No
EndTime	string	End time (HH:MM:SS)	No
Field Description ScheduleCount_n:1-5			

```
<AlarmScheduleConfig>
  <ScheduleCount><!-- req, xs: int --></ScheduleCount>
  <ScheduleCount_1>
   <ScheduleID><!-- req, xs: int --></ScheduleID>
    <ScheduleName><!-- req, xs: string --></ScheduleName>
    <ScheduleEnable><!-- req, xs: int --></ScheduleEnable>
    <AlarmOutEnable><!-- req, xs: int --></AlarmOutEnable>
    <AlarmOutInterval><!-- required, xs: integer --></AlarmOutInterval>
    <PlayEnable><!-- req, xs: int --></PlayEnable>
    <PlayFile><!-- required, xs: string --></PlayFile>
    <Volume><!-- required, xs: integer --></Volume>
    <FrontLightEnable><!-- required, xs: integer --></FrontLightEnable>
    <FrontLightcolor1><!-- req, xs: int --></FrontLightcolor1>
    <FrontLightcolor2><!-- req, xs: int --></FrontLightcolor2>
    <FrontLightDisplayMode><!-- required, xs: integer --></FrontLightDisplayMode>
    <FrontLightBrightness><!-- required, xs: integer --></FrontLightBrightness>
    <FrontLightFrequency><!-- req, xs: int --></FrontLightFrequency>
    <BackLightEnable><!-- required, xs: integer --></BackLightEnable>
    <BackLightcolor><!-- req, xs: int --></BackLightcolor>
   <BackLightDisplayMode><!-- req, xs: int --></BackLightDisplayMode>
    <BackLightBrightness><!-- required, xs: integer --></BackLightBrightness>
    <BackLightFrequency><!-- required, xs: integer --></BackLightFrequency>
    <HTTPMsgEnabled><!-- required, xs: integer --></HTTPMsgEnabled>
    <TCPIPMsgEnabled><!-- required, xs: integer --></TCPIPMsgEnabled>
    <MQTTMsgEnabled><!-- required, xs: integer --></MQTTMsgEnabled>
    <EmailEnabled><!-- required, xs: integer --></EmailEnabled>
    <BroadcastEnabled><!-- required, xs: integer --></BroadcastEnabled>
    <WeekDays><!-- required, xs: string --></WeekDays>
    <BeginDate><!-- req, xs: string --></BeginDate>
    <EndDate><!-- required, xs: string --></EndDate>
   <Every Year><!-- required, xs: int --></Every Year>
    <BeginTime><!-- req, xs: string --></BeginTime>
    <EndTime><!-- required, xs: string --></EndTime>
  </ScheduleCount_1>
</AlarmScheduleConfig>
```

2.4.12 Get/Set OSD Parameters

/ISAPI/OSD/OSDInfo?ID=SessionID	
POST	
Description	Get OSD Parameters
PUT	

Description	Set OSD Parameters		
Field	Type	Description	Required
OSDType	int	0:Date and Time 1:Custom Information	Yes
Enabled	int	0: Disabled 1: Enabled	No
DateFormat	int	0:YYYY-MM-DD 1:MM-DD-YYYY 2:DD-MM-YYYY	No
OSDText	string	OSDText	No
X	int	X Position(0-1000)	No
Y	int	Y Position(0-1000)	No
FontSize	int	FontSize	No
FontColor	string	FontColor(#000000 - #FFFFFF)	No

```
<OSDInfo>
<OSDParam_0>
<OSDType><!-- req, xs: int --></OSDType>
<Position>
    <Enabled><!-- req, xs: int --></Enabled>
    <OSDText><!-- req, xs: string --></OSDText>
    <X><!-- req, xs: int --></X>
    <Y><!-- req, xs: int --></Y>
  </Position>
  <DateFormat><!-- req, xs: int --></DateFormat>
  <FontSize><!-- req, xs: int --></FontSize>
  <FontColor><!-- req, xs: string --></FontColor>
</OSDParam_0>
</OSDInfo>
```

2.4.13 Testing speaker sound

/ISAPI/TestItem/TestPlayAudio?ID=SessionID			
POST			
Description	Play audio. Send the following XML message via POST request. Normal response: 200 OK		
Field	Type	Description	Required
Enabled	int	0: Disabled 1: Enabled	No
FileType	int	0: System default files 1: Customer-defined files	No
FileName	string	When empty, plays default file	No

```
<TestPlayAudio>
    <Enabled><!-- req, xs: int --></Enabled>
    <FileType><!-- req, xs: int --></FileType>
    <FileName><!-- required, xs: string --></FileName>
</TestPlayAudio>
```

2.4.14 Get TTSConfig

/ISAPI/Audio/GetTTSConfig=SessionID			
POST			
Description	Get TTSConfig. POST request with empty message body. Normal response returns the following XML message content		
Field	Type	Description	Required
SpeedMax	float	Speed Max value	No
SpeedMin	float	Speed Scale Min value	No
Speed	float	Length Scale value	No

```
<TTSConfig>
    <SpeedMax><!-- req, xs: float--></SpeedMax>
    <SpeedMin><!-- req, xs: float--></SpeedMin>
    <Speed><!-- req, xs: float--></Speed>
</TTSConfig>
```

2.4.15 Get TTS File

/ISAPI/Audio/GetDelTTSFile?ID=SessionID			
POST			
Description	Get TTS file. POST request with empty message body. Normal response returns the following XML message content		
Field	Type	Description	Required
FileCount	int	File count	Yes
FileName_n	string	File name	No
FileNameId_n	int	id	No
Text_n	string	Text	No

Field Description

FileName_n:1-10(tts1.wav)

```
<TTSFile>
    <FileCount><!-- req, xs: int --></FileCount>
```

```
    <FileName_n><!-- req, xs: string --></FileName_1>
    <FileNameId_n><!-- req, xs: int --></FileNameId_1>
    <Text_n><!-- req, xs: integer --></Text_1>
</TTSFile>
```

2.4.16 Upload/Test TTS File

/ISAPI/TestItem/TTSFile?=SessionID			
POST			
Description	Get TTS file. POST request with empty message body. Normal response returns the following XML message content		
Field	Type	Description	Required
Text	int	TTS Text	Yes
Speed	int	Speed (0.5-2)	No
Test	int	0: Test TTS file 1: Upload TTS file	Yes
TextMaxLen	int	Text max length	No
Enabled	int	0: Disabled 1: Enabled	No

```
<TTSSpeak>
    <Text><!-- req, xs: integer --></Text>
    <Speed><!-- req, xs: int --></Speed>
    <Test><!-- req, xs: int --></Test>
    <TextMaxLen><!-- req, xs: int --></TextMaxLen>
    <Enabled><!-- req, xs: int --></Enabled>
</TTSSpeak>
```

2.4.17 Delete TTS file

/ISAPI/Audio/GetDelTTSFile?ID=SessionID			
PUT			
Description	Delete TTSfile. The PUT request contains the following XML message content. Normal response: 200 OK		
Field	Type	Description	Required
FileNameId	int	File id	Yes

```
<TTSFile>
    <FileNameId><!-- req, xs: string--><FileNameId>
</TTSFile>
```

2.4.18 Download TTS file

/ISAPI/DownloadFile/TTSFile?ID=SessionID			
POST			
Description	Download TTS file. The POST request contains the following XML message content. Normal response: 200 OK		
Field	Type	Description	Required
FileNameId	int	FileName Id	Yes
Request XML <DownloadTTSFile> <FileNameId><!-- reg, xs:int --></FileNameId> </DownloadTTSFile>			

2.4.19 Edit TTS file

/ISAPI/Audio/SetTTSFile?ID=SessionID			
POST			
Description	Edit TTS file. The POST request contains the following XML message content. Normal response: 200 OK		
Field	Type	Description	Required
FileNameId	int	FileName Id	Yes
Text	int	TTS Text	Yes
Speed	int	Speed (0.5-2)	No
Request XML <SetTTSFile> <FileNameId><!-- reg, xs:int --></FileNameId> <Text><!-- reg, xs:int --></Text> <Speed><!-- reg, xs:int --></Speed> </SetTTSFile>			

2.4.20 Check Speaker

ISAPI/TestItem/CheckSpeaker?ID=SessionID	
POST	
Description	TestSpeaker. Send the following XML message via POST request.

	Normal response: 200 OK		
Field	Type	Description	Required
Enabled	int	0: Disabled 1: Enabled	Yes
Request XML <CheckSpeaker> <Enabled><!-- reg, xs:int --></Enabled> </CheckSpeaker>			

2.5 Network Parameters

2.5.1 Get/Set Network Parameters

/ISAPI/Network /NetworkConfig?ID=SessionID			
POST			
Description	Get network parameters. POST request with empty message. Normal response returns the following XML message content		
PUT			
Description	Configure network parameters and send the following XML message via a PUT request. A normal response returns 200 OK.		
Field	Type	Description	Required
MaxLink	int	Maximum link count (1-10), fixed at 10, cannot be set	No
DHCP	int	0: Disabled 1: Enabled	No
RtspAuthentication	int	0: No authentication 1: Authenticate	No
IpAddr	string	IP address	No
SubnetMask	string	Subnet mask	No
Gateway	string	Gateway	No
DNS1	string	Preferred DNS	No
DNS2	string	Alternate DNS	No
IPv6 Prefix Length	int	IPv6 prefix length	No
MACAddress	string	MAC Address	Yes
Name	string	Network card name	No
<NetworkConfig> <NetLink> <MaxLink><!-- req, xs: int --></MaxLink> <DHCP><!-- required, xs: integer --></DHCP> <RtspAuthentication><!-- required, xs: integer --></RtspAuthentication>			

```
        <Name ><!-- req, xs: string --></Name>
        <Ipv4>
          <IpAddr><!-- req, xs: string --></IpAddr>
            <SubnetMask><!-- req, xs: string --></SubnetMask>
            <Gateway><!-- required, xs: string --></Gateway>
            <DNS1><!-- required, xs: string --></DNS1>
            <DNS2><!-- required, xs: string --></DNS2>
          </Ipv4>
          <Ipv6>
            <IpAddr><!-- req, xs: string --></IpAddr>
            <Gateway><!-- req, xs: string --></Gateway>
            <DNS1><!-- req, xs: string --></DNS1>
            <DNS2><!-- req, xs: string --></DNS2>
            <Ipv6PrefixLength><!-- req, xs: int --></Ipv6PrefixLength>
        </Ipv6>
        <MACAddress><!-- req, xs: string --></MACAddress>
    </NetLink>
</NetworkConfig>
```

2.5.2 Get/Set Email Parameters

/ISAPI/Network/Email?ID=SessionID			
POST			
Description	Get Email parameter. POST request with empty message returns the following XML message content normally		
PUT			
Description	Set email parameters. Send a PUT request with the following XML message content. A normal response returns 200 OK.		
Field	Type	Description	Required
Enabled	int	0: Disabled 1: Enabled	No
SensorAlarmTitle	string	Probe Alarm Title	No
SmtpServer	string	SMTP Server	No
SmtpUser	string	Username	No
SmtpPassword	string	Password	No
Sender	string	Sender	No
Receiver_1	string	Recipient (supports four groups)	No
SmtpPort	int	SMTP port	No

```
<EmailStatus>
    <Enabled><!-- req, xs: int --></Enabled>
    <SensorAlarmTitle><!-- req, xs: string --></SensorAlarmTitle>
    <SmtpServer><!-- req, xs: string --></SmtpServer>
```

```
    <SmtpUser><!-- required, xs: string --></SmtpUser>
    <SmtpPassword><!-- req, xs: string --></SmtpPassword>
    <Sender><!-- req, xs: string --></Sender>
    <Receiver_1><!-- required, xs: string --></Receiver_1>
    <Receiver_2><!-- required, xs: string --></Receiver_2>
    <Receiver_3><!-- req, xs: string --></Receiver_3>
    <Receiver_4><!-- req, xs: string --></Receiver_4>
    <SmtpPort><!-- req, xs: int --></SmtpPort>
</EmailStatus>
```

2.5.3 Get/Set SNMP Parameters

/ISAPI/Network/SNMP?ID=SessionID

POST

Description	Get SNMP parameters. POST request with empty message body. Normal response returns the following XML message content

PUT

Description	Set SNMP parameters. Send the following XML message via a PUT request. A normal response returns 200 OK.		
Field	Type	Description	Required
Enabled->Ver1	int	0:Disabled 1:Enabled	No
Enabled->Ver2	int	0:Disabled 1:Enabled	No
ReadPublic	string	Read Public	No
WritePublic	string	Write Public	No
TrapAddress	string	Trap Address	No
TrapPort	int	Trap Port	No
SnmpPort	int	SNMP Port	No

```
<SNMPStatus>
        <Enabled>
            <Ver1><!-- req, xs: int --></Ver1>
            <Ver2><!-- req, xs: int --></Ver2>
        </Enabled>
        <ReadPublic><!-- req, xs: string --></ReadPublic>
        <WritePublic><!-- req, xs: string --></WritePublic>
        <TrapAddress><!-- req, xs: string --></TrapAddress>
        <SnmpPort><!-- req, xs: int --></SnmpPort>
        <TrapPort><!-- req, xs: int --></TrapPort>
</SNMPStatus>
```

2.5.4 Get/Set WiFi Parameters

/ISAPI/System/WifiConfig?ID=SessionID			
POST			
Description	Get Wi-Fi parameters. POST request with empty message body. Normal response returns the following XML message content		
PUT			
Description	Set Wi-Fi parameters. Send the following XML message via a PUT request. A normal response returns 200 OK.		
Field	Type	Description	Required
Enabled	int	0: Disabled 1: Enabled	No
Status	int	0: Disconnected 1: Connected	No
SSID name	string	Wifi name	No
Passwd	string	Wifi password	No
<WifiConfig> <Enabled><!-- req, xs: int --></Enabled> <Status><!-- required, xs: integer --></Status> <SSIDname><!-- required, xs: string --></SSIDname> <Passwd><!-- required, xs: string --></Passwd> </WifiConfig>			

2.5.5 Get WiFi network parameters

/ISAPI/System/WifiInfo?ID=SessionID			
POST			
Description	Get Wi-Fi network parameters. POST request with empty message body. Normal response returns the following XML message content		
Field	Type	Description	Required
IpAddr	string	IP Address	No
SubMask	string	Subnet mask	No
Broadcast	string	Broadcast	No

Gateway	string	Gateway	No
Mac Address	string	Mac Address	No

```
<WifiInfo>
    <IpAddr><!-- req, xs: string --></IpAddr>
    <SubMask><!-- req, xs: string --></SubMask>
    <Broadcast><!-- req, xs: string --></Broadcast>
    <Gateway><!-- required, xs: string --></Gateway>
    <MacAddr><!-- required, xs: string --></MacAddr>
</WifiInfo>
```

2.5.6 Get WiFi list

/ISAPI/System/WifiScanResult?ID=SessionID			
POST			
Description	Get other Wi-Fi parameters. POST request with empty message body. Normal response returns the following XML message content		
Field	Type	Description	Required
Count	int	Number of Wi-Fi networks	No
Signal Level	int	Signal strength, with the following values representing weak to strong signals, where 0 is weakest and 4 is strongest: 0: (~, -100] dB 1: (-100, -88] dB 2: (-88, -77] dB 3: (-77, -55] dB 4: (-55, ~) dB	No
SSID name	string	Wifi name	No

```
<ScanResult>
    <Count><!-- req, xs: int --></Count>
    <Wifi_0>
        <SignalLevel><!-- required, xs: string --></SignalLevel>
        <SSIDname><!-- required, xs: string --></SSIDname>
    </Wifi_0>
        ...
    <Wifi_n>
        <SignalLevel><!-- required, xs: string --></SignalLevel>
        <SSIDname><!-- req, xs: string --></SSIDname>
    </Wifi_n>
```

</ScanResult>

2.5.7 Get/Set HTTPs Parameters

/ISAPI/Network/HTTPs?ID=SessionID			
POST			
Description	Get HTTPs parameters. POST request with empty message body. Normal response returns the following XML message content		
PUT			
Description	Set HTTPs parameters. Send the following XML message via PUT request. Normal response: 200 OK		
Field	Type	Description	Required
Enabled	int	0: Disabled 1: Enabled	No
HTTPsPort	int	HTTPs port (443)	No
<HTTPsStatus> <Enabled><!-- req, xs: int --></Enabled> <HTTPsPort><!-- required, xs: integer --></HTTPsPort> </HTTPsStatus>			

2.5.8 Get/Set SIP Parameters

/ISAPI/Network/SIP?ID=SessionID			
POST			
Description	Get server SIP parameters. POST request with empty message body. Normal response returns the following XML message content		
PUT			
Description	Set server SIP parameters. Send the following XML message via PUT request. A normal response returns 200 OK.		
Field	Type	Description	Required
Enabled	int	0: Disabled 1:Enabled	No
UserName	string	Username (maximum 64 characters)	Yes
PassWord	string	Password (maximum 64 characters)	Yes
AuthId	string	Authentication ID	Yes

ServerIp	string	Server Ip	Yes
ServerPort	int	Server Port	Yes
Protocol	int	0:UDP 1:TCP 2:TLS	Yes
Encryption	int	0:None 1:SRTP	Yes
AnswerType	int	0:Answer Immediately 1:Answer Profile	Yes
AnswerProfileID	int	0:Profile1 1:Profile2 2:Profile3 3:Profile4 4:Profile5	Yes

```
<SIPConfig>
   <Enabled><!-- req, xs: int --></Enabled>
   <UserName><!-- req, xs: string --></UserName>
   <PassWord><!-- req, xs: string --></PassWord>
   <AuthId><!-- req, xs: string --></AuthId>
   <ServerIp><!-- req, xs: string --></ServerIp>
   <ServerPort><!-- req, xs: int --></ServerPort>
   <Protocol><!-- req, xs: int --></Protocol>
   <Encryption><!-- req, xs: int --></Encryption>
   <AnswerType><!-- req, xs: int --></AnswerType>
   <AnswerProfileID><!-- req, xs: int --></AnswerProfileID>
</SIPConfig>
```

2.5.9 Get/Set SIP P2P Parameters

/ISAPI/Network/SIPP2P?ID=SessionID			
POST			
Description	Get server SIPP2P parameters. POST request with empty message body. Normal response returns the following XML message content		
PUT			
Description	Set server SIPP2P parameters. Send the following XML message via PUT request. A normal response returns 200 OK.		
Field	Type	Description	Required

Enabled	int	0: Disabled 1:Enabled	No
Address	string	Address	Yes

```
<SIPP2PConfig>
  <Enabled><!-- req, xs: int --></Enabled>
  <Address><!-- req, xs: string --></Address>
</SIPP2PConfig>
```

2.5.10 Get/Set Other Parameters

/ISAPI/Network/ExtServerParam?ID=SessionID			
POST			
Description	Get other parameters		
PUT			
Description	Set other parameters		
Field	Type	Description	Required
Option_n		Option_1: HTTP Service Option_2: ONVIF Option_3: RTSP Option_6: NAT	
ServerType	int	1: HTTP Service 2: ONVIF 3: RTSP	Yes
Enabled	int	0: Disabled 1: Enabled	No
Port	int	Port	No
AnswerType	int	0:Answer Immediately 1:Answer Profile	No
AnswerProfileID	int	0:Profile1 1:Profile2 2:Profile3 3:Profile4 4:Profile5	No

```
<ExtServer>
  <Option_1>
    <ServerType><!-- req, xs: int --></ServerType>
    <Enabled><!-- req, xs: int --></Enabled>
    <Port><!-- required, xs: integer --></Port>
  </Option_1>
 <Option_3>
 <ServerType>3</ServerType>
```

```
        <Enabled><!-- req, xs: int --></Enabled>
        <Port><!-- req, xs: int --></Port>
        <RtspAuthentication><!-- req, xs: int --></RtspAuthentication>
        <AnswerType><!-- req, xs: int --></AnswerType>
        <AutoAnswer><!-- req, xs: int --></AutoAnswer>
        <AnswerProfileID><!-- req, xs: int --></AnswerProfileID>
        <Profile><!-- req, xs: int --></Profile>
  </Option_3>
  <Option_6>
     <NatEnable><!-- req, xs: int --></NatEnable>
     <WanIp><!-- req, xs: string --></WanIp>
     <WanHttpPort><!-- req, xs: int --></WanHttpPort>
     <WanRtspPort><!-- req, xs: int --></WanRtspPort>
   </Option_6>
</ExtServer>
```

2.5.11 Get/Set Heartbeat Parameters

/ISAPI/Network/Heartbeat?ID=SessionID			
POST			
Description	Get Heartbeat parameters. POST request with empty message body. Normal response returns the following XML message content		
PUT			
Description	Set Heartbeat parameters. Send the following XML message via PUT request. A normal response returns 200 OK.		
Field	Type	Description	Required
Enabled	int	0: Disabled 1:Enabled	No
Server	string	Server Ip	Yes
Protocol	int	0:HTTP	Yes
Method	int	0:POST 1:PUT	Yes
ContentType	int	0: application/json	No
AuthMethod	int	0: None 1:Basic auth	No
UserName	string	Username (maximum 64 characters)	No
PassWord	string	Password (maximum 64 characters)	No

Interval	string	30-300s	Yes
Content	string	{ "mac_address":"%SYSTEM@MAC%", "ip_address":"%SYSTEM@IP%" }	No

```
<HeartbeatConfig >
        <Enabled><!-- req, xs: int --></Enabled>
        <Protocol><!-- req, xs: int --></Protocol>
        <Server><!-- req, xs: int --></Server>
        <Method><!-- req, xs: int --></Method>
        <ContentType><!-- req, xs: int --></ContentType>
        <AuthMethod><!-- req, xs: int --></AuthMethod>
        <UserName><!-- req, xs: string --></UserName>
        <Password><!-- req, xs: string --></Password>
        <Interval><!-- req, xs: string --></Interval>
        <Content><!-- req, xs: string --></Content>
</HeartbeatConfig>
```

2.5.12 Get/Set Multicast Parameters

ISAPI/Audio/MulticastController?ID=SessionID			
POST			
Description	Get Multicast parameters. POST request with empty message body. Normal response returns the following XML message content		
PUT			
Description	Set Multicast parameters. Send the following XML message via PUT request. A normal response returns 200 OK.		
Field	Type	Description	Required
Enabled	int	0: Disabled 1:Enabled	No
Master	string	0: Disabled 1:Enabled	Yes
SIP	int	0: Disabled 1:Enabled	Yes
ONVIF	int	0: Disabled 1:Enabled	Yes
AudioCodec	int	PCMA,PCMU	No

AnswerType	int	0:Answer Immediately 1:Answer Profile	Yes
AnswerProfileID	int	0:Profile1 1:Profile2 2:Profile3 3:Profile4 4:Profile5	Yes
Playing	int	0: yes 1:no	No
ActiveSource	string	Active Source	No
Sources	string	Source_n	No

```
<AudioMulticastController>
   <Enabled><!-- req, xs: int --></Enabled>
   <Master><!-- req, xs: string --></Master>
   <SIP><!-- req, xs: int --></SIP>
   <ONVIF><!-- req, xs: int --></ONVIF>
   <AudioCodec><!-- req, xs: string --></AudioCodec>
   <AnswerType><!-- req, xs: string --></AnswerType>
   <AutoAnswer><!-- req, xs: string --></AutoAnswer>
   <AnswerProfileID>0</AnswerProfileID>
   <Profile><!-- req, xs: string --></Profile>
   <Playing><!-- req, xs: string --></Playing>
   <ActiveSource><!-- req, xs: string --></ActiveSource>
   <Sources>
      <Source_1>
      <id><!-- req, xs: string --></id>
      <label><!-- req, xs: string --></label>
      <ipAddress><!-- req, xs: string --></ipAddress>
      <port><!-- req, xs: int --></port>
      <prio><!-- req, xs: string --></prio>
      <type><!-- req, xs: string --></type>
      </Source_1>
      <Source_n>
      ...
      </Source_n>
   </Sources>
</AudioMulticastController>
```

2.6 Event

2.6.1 Get/Set Probe Alarm Parameters

/ISAPI/Event/SensorAlarmParam?ID=SessionID			
POST			
Description	Get probe alarm parameters		
PUT			
Description	Set probe alarm parameters		
Field	Type	Description	Required
Enabled	int	0: Disabled 1: Enabled	Yes
AlarmInType	int	0: High level 1: Low Level	Yes
AlarmOutType	int	0: High level 1: Low level	Yes
<AlarmConfig> <Enabled><!-- req, xs: int --></Enabled> <AlarmInType><!-- req, xs: int --></AlarmInType> <AlarmOutType><!-- req, xs: int --></AlarmOutType> </AlarmConfig>			

2.6.2 Get/Set Probe Alarm Sensor Status

/ISAPI/Event/SensorAlarmStatus?ID=SessionID			
POST			
Description	Get probe alarm Status		
PUT			
Description	Set probe alarm Status		
Field	Type	Description	Required
AlarmInTriggered	int	0: Disabled 1: Enabled	Yes
AlarmOutChannels	int	0: Disabled 1: Enabled	Yes
Triggered	int	0: Disabled 1: Enabled	Yes
<SensorAlarmStatus> <AlarmInTriggered><!-- req, xs: int --></AlarmInTriggered> <AlarmOutChannels><!-- req, xs: int --></AlarmOutChannels>			

```
    <AlarmOut_0>
        <Triggered><!-- req, xs: int --></Triggered>
    </AlarmOut_0>
</SensorAlarmStatus>
```

2.6.3 Get/Set Profile Alarm Parameters

/ISAPI/Event/AlarmProfileConfig?ID=SessionID			
POST			
Description	Get Profile Alarm parameters. POST request with empty message body. Normal response returns the following XML message content		
PUT			
Description	Set Profile Alarm parameters. Send the following XML message via a PUT request. A normal response returns 200 OK.		
Field	Type	Description	Required
AlarmProfileCount	int	Profile Count	No
AlarmProfileID	int	AlarmProfileID（rage0-4）	Yes
AlarmProfileName	string	AlarmProfileName	No
AlarmEnable	int	0:Disabled 1:Enabled	No
AlarmOutEnable	int	0:Disabled 1:Enabled	No
AlarmOutDuration	int	Alarm output duration per trigger, default 60 seconds	No
PlayEnable	int	0:Disabled 1:Enabled	No
PlayFile	string	PlayFile	No
Volume	int	Volume	No
FrontLightEnable	int	0:Disabled 1:Enabled	No
FrontLightColor1	int	1:RedColor 2:GreenColor 3:BlueColor 4:WhiteColor 5:Yellow 6:Violet 7:Cyan	No
FrontLightColor2	int	0:None 1:RedColor	No

		2:GreenColor 3:BlueColor 4:WhiteColor 5:Yellow 6:Violet 7:Cyan	
FrontLightDisplayMode	int	DisplayMode 1:Steady 2:OneSecBlink (illuminates 10 times) 3:TwoSecBlink (illuminates 10 times) 4:FiveSecBlink (illuminates 10 times) 5:Breathe 6:Strobe	No
FrontLightBrightness	int	Only the following colors can be set to brightness levels 0-255 (low to high): 1:RedColor 2:GreenColor 3:BlueColor 4:WhiteColor	No
FrontLightFrequency	int	Only Strobe mode can set the flashing frequency, in milliseconds: default 100	No
BackLightEnable	int	0:Disabled 1:Enabled	No
BackLightColor	int	1:RedColor 2:GreenColor 3:BlueColor 4:WhiteColor 5:Yellow 6:Violet 7:Cyan	No
BackLightDisplayMode	int	DisplayMode 1:Steady 2:Chase	No
BackLightBrightness	int	Only the following colors can be set to brightness levels 0-255 (low to high): 1:RedColor 2:GreenColor 3:BlueColor	No

		4:WhiteColor	
BackLightFrequency	int	Only Strobe mode can set the flashing frequency, in milliseconds: default 100	No
HttpMsgEnabled	int	0:Disabled 1:Enabled	No
EmailEnabled	int	0:Disabled 1:Enabled	No
BroadcastEnabled	int	0:Disabled 1:Enabled	No

Field description

AlarmProfileCount_n:1-5

```
<AlarmProfileConfig>
  <AlarmProfileCount><!-- req, xs: int --></AlarmProfileCount>
  <AlarmProfileCount_1>
    <AlarmProfileID><!-- req, xs: int --></AlarmProfileID>
    <AlarmProfileName><!-- req, xs: string --></AlarmProfileName>
    <AlarmEnable><!-- req, xs: int --></AlarmEnable>
    <AlarmOutEnable><!-- req, xs: int --></AlarmOutEnable>
    <AlarmOutInterval><!-- req, xs: int --></AlarmOutInterval>
    <PlayEnable><!-- req, xs: int --></PlayEnable>
    <PlayFile><!-- req, xs: string --></PlayFile>
    <Volume><!-- req, xs: int --></Volume>
    <FrontLightEnable><!-- req, xs: int --></FrontLightEnable>
    <FrontLightColor1><!-- req, xs: int --></FrontLightColor1>
    <FrontLightColor2><!-- req, xs: int --></FrontLightColor2>
    <FrontLightDisplayMode><!-- req, xs: int --></FrontLightDisplayMode>
    <FrontLightBrightness><!-- req, xs: int --></FrontLightBrightness>
    <FrontLightFrequency><!-- req, xs: int --></FrontLightFrequency>
    <BackLightEnable><!-- req, xs: int --></BackLightEnable>
    <BackLightColor><!-- req, xs: int --></BackLightColor>
    <BackLightDisplayMode><!-- req, xs: int --></BackLightDisplayMode>
    <BackLightBrightness><!-- req, xs: int --></BackLightBrightness>
    <BackLightFrequency><!-- req, xs: int --></BackLightFrequency>
    <HttpMsgEnabled><!-- req, xs: int --></HttpMsgEnabled>
    <EmailEnabled><!-- req, xs: int --></EmailEnabled>
    <BroadcastEnabled><!-- req, xs: int --></BroadcastEnabled>
  </AlarmProfileCount_1>
</AlarmProfileConfig>
```

2.6.4 Get/Set System Log Parameters

/ISAPI/Log/LogConfig?ID=SessionID			
POST			
Description	Query system logs		
Field	Type	Description	Required
LogEnabled	int	0:Disabled 1:Enabled	No
ServerIp	string	Log Serve rIp	Yes
ServerPort	string	Log Server Port	Yes
<LogConfig> <LogEnabled><!-- req, xs: int --></LogEnabled> <ServerIp><!-- req, xs: string --></ServerIp> <ServerPort><!-- req, xs: int --></ServerPort> </LogConfig>			

2.6.5 Query Alarm Log

/ISAPI/Log/QueryLog?ID=SessionID			
POST			
Description	Query alarm log		
Field	Type	Description	Required
StartTime	string	Start time: 1 July 2018 18:00:00	Yes
EndTime	string	End Time: 1 July 2018 18:00:00	Yes
AlarmCount	int	Alarm Count	No
AlarmType	int	Alarm Type: 0: Sensor Alarm 1: API Alarm 2: Broadcast	No
AlarmTime	string	Time (TZ format, e.g.: 2018-07-01T18:00:00Z)	No
Request XML <QueryLog> <StartTime><!-- req, xs: string --></StartTime> <EndTime><!-- req, xs: string --></EndTime> </QueryLog> **Response XML** <QueryLog> <AlarmCount><!-- req, xs: int --></AlarmCount>			

```
    <AlarmCount_1>
        <AlarmType><!-- req, xs: int --></AlarmType>
        <AlarmTime><!-- req, xs: string --></AlarmTime>
    <AlarmCount_1>
    ...
    <AlarmCount_n>
        <AlarmType><!-- req, xs: int --></AlarmType>
        <AlarmTime><!-- req, xs: string --></AlarmTime>
    <AlarmCount_n>
</QueryLog>
```

2.6.6 API Alarm Input

Method 1: Http api + xml

http://<DeviceIp>/ISAPI/Event/ApiAlarm			
POST/GET			
Description	Api Alarm		
Field	Type	Description	Required
AlarmEnable	int	0: Disabled 1: Enabled	No
AlarmOutEnable	int	0: Disabled 1: Enabled	No
AlarmOutDuration	int	Alarm output duration per trigger, default 60 seconds	No
PlayEnable	int	0: Disabled 1: Enabled	No
PlayFile	string	File name	No
TTS	string	Text	No
Speed	int	Speed(0.5-2)	No
Volume	int	Volume(0-100)	No
CycleMode	int	CycleMode： 0:Once only 1:Multiple times 2:Infinite	No
PlayTimes	int	Configure playback count when selecting multiple times	No
Interval	int	Playback interval time, in seconds	No
FrontLightEnable	int	0: Disabled 1: Enabled	No
FrontLightColor1	int	1:RedColor 2:GreenColor	No

		3:BlueColor 4:WhiteColor 5:Yellow 6:Violet 7:Cyan	
FrontLightColor2	int	0:None 1:RedColor 2:GreenColor 3:BlueColor 4:WhiteColor 5:Yellow 6:Violet 7:Cyan	No
FrontLightDisplayMode	int	Display Mode 1: Steady 2: One-Second Blink (illuminates 10 times) 3: Two-Second Blink (illuminates 10 times) 4: Five-Second Blink (illuminates 5 times) 5: Breathe 6:Strobe	No
FrontLightBrightness	int	Only the following colors can be set to brightness levels 0-255 (low to high): 1:RedColor 2:GreenColor 3:BlueColor 4:WhiteColor	No
FrontLightFrequency	int	Only Strobe mode allows setting the flashing frequency, in milliseconds: default 100	No
BackLightEnable	int	0: Disabled 1: Enabled	No
BackLightColor	int	1:RedColor 2:GreenColor 3:BlueColor 4:WhiteColor 5:Yellow 6:Violet 7:Cyan	No
BackLightDisplayMode	int	Display Mode 1:Steady	No

		2:Chase	
BackLightBrightness	int	Only the following colors can be set to brightness levels 0-255 (low to high): 1:RedColor 2:GreenColor 3:BlueColor 4:WhiteColor	No
BackLightFrequency	int	Only Strobe mode can set the flashing frequency, in milliseconds: default 100	
HttpMsgEnabled	int	0: Disabled 1: Enabled	No
EmailEnabled	int	0: Disabled 1: Enabled	No
BroadcastEnabled	int	0: Disabled 1: Enabled	No

```
<ApiAlarm>
   <AlarmEnable><!-- req, xs: int --></AlarmEnable>
   <AlarmOutEnable><!-- req, xs: int --></AlarmOutEnable>
   <AlarmOutDuration><!-- req, xs: int --></AlarmOutDuration>
   <PlayEnable><!-- req, xs: int --></PlayEnable>
   <PlayFile><!-- req, xs: string --></PlayFile>
   <TTS><!-- req, xs: string --></TTS>
   <Speed><!-- req, xs: int --></Speed>
   <Volume><!-- req, xs: int --></Volume>
   <CycleMode><!-- req, xs: int --></CycleMode>
   <PlayTimes><!-- req, xs: int --></PlayTimes>
   <Interval><!-- req, xs: int --></Interval>
   <FrontLightEnable><!-- req, xs: int --></FrontLightEnable>
   <FrontLightColor1><!-- req, xs: int --></FrontLightColor1>
   <FrontLightColor2><!-- req, xs: int --></FrontLightColor2>
   <FrontLightDisplayMode><!-- req, xs: int --></FrontLightDisplayMode>
   <FrontLightBrightness><!-- req, xs: int --></FrontLightBrightness>
   <BackLightEnable><!-- req, xs: int --></BackLightEnable>
   <BackLightColor><!-- req, xs: int --></BackLightColor>
   <BackLightDisplayMode><!-- req, xs: int --></BackLightDisplayMode>
   <BackLightBrightness><!-- req, xs: int --></BackLightBrightness>
   <HttpMsgEnabled><!-- req, xs: int --></HttpMsgEnabled>
   <EmailEnabled><!-- req, xs: int --></EmailEnabled>
   <BroadcastEnabled><!-- req, xs: int --></BroadcastEnabled>
</ApiAlarm>
```

Example:

```
http://192.168.88/ISAPI/Event/ApiAlarm HTTP/1.1
User-Agent: Apifox/1.0.0 (https://apifox.com)
Content-Type: application/xml
Authorization: Basic YWRtaW46YWRtaW4xMjM0NTY=
Accept: */*
Host: 192.168.1.88
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Content-Length: 784

<ApiAlarm>
<Alarmenable>1</Alarmenable>
<AlarmOutEnable>1</AlarmOutEnable>
<AlarmOutDuration>20</AlarmOutDuration>
<PlayEnable>1</PlayEnable>
<PlayFile>You are not Alone.wav</PlayFile>
<Speed><!-- req, xs: int --></Speed>
<Volume><!-- req, xs: int --></Volume>
<Volume>80</Volume>
<CycleMode>1</CycleMode>
<PlayTimes>3</PlayTimes>
<Interval>5</Interval>
<FrontLightEnable>1</FrontLightEnable>
<FrontLightColor1>1</FrontLightColor1>
<FrontLightColor2>2</FrontLightColor2>
<FrontLightDisplayMode>6</FrontLightDisplayMode>
<FrontLightBrightness>20</FrontLightBrightness>
<FrontLightFrequency>100</FrontLightFrequency>
<BackLightEnable>1</BackLightEnable>
<BackLightColor>1</BackLightColor>
<BackLightDisplayMode>1</BackLightDisplayMode>
<BackLightBrightness>20</BackLightBrightness>
<BackLightFrequency>100</BackLightFrequency>
<HttpMsgEnabled>0</HttpMsgEnabled>
<BroadcastEnabled>0</BroadcastEnabled>
<EmailEnabled>0</EmailEnabled>
</ApiAlarm>
```

Method 2：Http api + url

http://192.168.1.88/ISAPI/Event/UrlApiAlarm?alarmenable=1&alarmoutenable=1&alarmoutduration=20&playenable=1&playfile=You are not Alone.wav&volume=80&cyclemode=1&playtimes=3&interval=5&flightenable=1&flightcolor1=1&flightcolor2=2&flightdisplaymode=6&flightbrightness=20&flightfrequency=100&blightenable=1&blightcolor=1&blightdisplaymode=1&blightbrightness=20&blightfrequency=100&httpmsgenabled=0&emailenabled=0&broadcastenabled=0

Example:
POST/GET /ISAPI/Event/UrlApiAlarm
Authorization: Basic YWRtaW46YWRtaW4xMjM0NTY=
Host: 192.168.1.88

Example1: start play audio three times, interval is 5, and volume is 80
http://192.168.1.88/ISAPI/Event/UrlApiAlarm?alarmenable=1&alarmoutenable=1&alarmoutduration=30&playenable=1&playfile=You are not Alone.wav&volume=80&cyclemode=1&playtimes=3&interval=5

Example2: start play TTS text three times, interval is 5, and volume is 80
http://192.168.1.110/ISAPI/Event/UrlApiAlarm?alarmenable=1&alarmoutenable=1&alarmoutduration=30&playenable=1&tts=Please leave this restricted area right now.&speed=1&volume=80&cyclemode=1&playtimes=3&interval=5

Example3: start front light with color green and red, display mode strobe, brightness 20, frequency 100
http://192.168.1.88/ISAPI/Event/UrlApiAlarm?alarmenable=1&alarmoutenable=1&alarmoutduration=30&flightenable=1&flightcolor1=1&flightcolor2=2&flightdisplaymode=6&flightbrightness=20&flightfrequency=100

Example4: start back light with color green and red, display mode strobe, brightness 20, frequency 100
http://192.168.1.88/ISAPI/Event/UrlApiAlarm?alarmenable=1&alarmoutenable=1&alarmoutduration=30&blightenable=1&blightcolor=1&blightdisplaymode=1&blightbrightness=20&blightfrequency=100

Example5: start audio + front light + back light
http://192.168.1.88/ISAPI/Event/UrlApiAlarm?alarmenable=1&alarmoutenable=1&alarmoutduration=30&playenable=1&playfile=You are not Alone.wav&volume=80&cyclemode=1&playtimes=3&interval=5&flightenable=1&flightcolor1=1&flightcolor2=2&flightdisplaymode=6&flightbrightness=20&flightfrequency=100&blightenable=1&blightcolor=1&blightdisplaymode=1&blightbrightness=100&blightfrequency=100&httpmsgenabled=0&emailenabled=0&broadcastenabled=0

Method 3：Http api + url

/ISAPI/Event/ProfileApiAlarm/Profile1
/ISAPI/Event/ProfileApiAlarm/Profile2

Distribute the relevant alarm configuration parameters via Profile1

2.6.7 API Idle Input

http://<DeviceIp>/ISAPI/IdleScheduleConfig			
POST/GET			
Description	Api Idle		
Field	Type	Description	Required
ScheduleID	int	ScheduleID (range 0-4)	Yes
ScheduleEnable	int	0: Disabled 1: Enabled	No
PlayEnable	int	0: Disabled 1: Enabled	No
PlayFile	string	File name	No
Volume	int	Volume (0-100)	No
CycleMode	int	Cycle Mode: 0: Once only 1: Multiple times 2: Infinite	No
PlayTimes	int	Configure playback count when selecting multiple times	No
Interval	int	Playback interval time, in seconds	No
FrontLightEnable	int	0: Disabled 1: Enabled	No
FrontLightColor1	int	1:RedColor 2: GreenColor 3:BlueColor 4:WhiteColor 5:Yellow 6:Violet 7:Cyan	No
FrontLighColor2	int	0:None 1:RedColor 2: GreenColor 3: BlueColor 4:WhiteColor	No

		5:Yellow 6:Violet 7:Cyan	
FrontLightDisplayMode	int	Display Mode 1: Steady 2: One-Second Blink (illuminates 10 times) 3: Two-Second Blink (illuminates 10 times) 4: Five-Second Blink (illuminates 5 times) 5: Breathe 6:Strobe	No
FrontLightBrightness	int	Only the following colors can be set to brightness levels 0-255 (low to high): 1:RedColorr 2: GreenColor 3: BlueColor 4:WhiteColor	No
FrontLightFrequency	int	Only Strobe mode allows setting the flashing frequency in milliseconds: default 100	No
BackLightEnable	int	0: Disabled 1: Enabled	No
BackLightColor	int	1:RedColor 2: GreenColor 3:BlueColor 4:WhiteColor 5:Yellow 6:Violet 7:Cyan	No
BacklightDisplayMode	int	Display Mode 1: Steady 2:Chase	No
BacklightBrightness	int	Only the following colors can be set to brightness levels 0-255 (low to high): 1:RedColor 2: GreenColor 3: BlueColor 4:WhiteColor	No
BacklightFrequency	int	Only Strobe mode can set the flashing frequency, in milliseconds:	No

		default 100	
WeekDays	string	Select the time period for accepting alarms. For example, 0123456 indicates Sunday to Saturday.	No
BeginTime	string	Start time (HH:MM:SS)	No
EndTime	string	End time (HH:MM:SS)	No

```
<IdleScheduleConfig>
    <ScheduleID><!-- req, xs: int --></ScheduleID>
    <ScheduleEnable><!-- req, xs: int --></ScheduleEnable>
    <PlayEnable><!-- required, xs: integer --></PlayEnable>
    <PlayFile><!-- req, xs: string --></PlayFile>
    <Volume><!-- required, xs: integer --></Volume>
    <CycleMode><!-- req, xs: int --></CycleMode>
    <PlayTimes><!-- req, xs: int --></PlayTimes>
    <Interval><!-- required, xs: integer --></Interval>
    <FrontLightEnable><!-- required, xs: integer --></FrontLightEnable>
   <FrontLightColor1><!-- req, xs: int --></FrontLightColor1>
    <FrontLightColor2><!-- req, xs: int --></FrontLightColor2>
    <FrontLightDisplayMode><!-- req, xs: int --></FrontLightDisplayMode>
    <FrontLightBrightness><!-- required, xs: integer --></FrontLightBrightness>
    <FrontLightFrequency><!-- req, xs: int --></FrontLightFrequency>
    <BackLightEnable><!-- required, xs: integer --></BackLightEnable>
    <BackLightcolor><!-- req, xs: int --></BackLightcolor>
    <BackLightDisplayMode><!-- required, xs: integer --></BackLightDisplayMode>
    <BackLightBrightness><!-- required, xs: integer --></BackLightBrightness>
    <BackLightFrequency><!-- required, xs: integer --></BackLightFrequency>
    <WeekDays><!-- required, xs: string --></WeekDays>
    <BeginTime><!-- required, xs: string --></BeginTime>
    <EndTime><!-- required, xs: string --></EndTime>
</IdleScheduleConfig>
```

Example:

```
http://192.168.1.88/ISAPI/IdleScheduleConfig    HTTP/1.1
User-Agent: Apifox/1.0.0 (https://apifox.com)
Content-Type: application/xml
Authorization: Basic YWRtaW46YWRtaW4xMjM0NTY=
Accept: */*
Host: 192.168.1.88
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Content-Length: 784

<IdleScheduleConfig>
<ScheduleID>1</ScheduleID>
<ScheduleEnable>1</ScheduleEnable>
<PlayEnable>1</PlayEnable>
<PlayFile>You are not Alone.wav</PlayFile>
<Volume>80</Volume>
<CycleMode>1</CycleMode>
<PlayTimes>3</PlayTimes>
<Interval>5</Interval>
<FrontLightEnable>1</FrontLightEnable>
<FrontLightColor1>1</FrontLightColor1>
<FrontLightColor2>1</FrontLightColor2>
<FrontLightDisplayMode>1</FrontLightDisplayMode>
<FrontLightBrightness>20</FrontLightBrightness>
<FrontLightFrequency>100</FrontLightFrequency>
<BackLightEnable>1</BackLightEnable>
<BackLightColor>1</BackLightColor>
<BackLightDisplayMode>1</BackLightDisplayMode>
<BackLightBrightness>20</BackLightBrightness>
<BackLightFrequency>100</BackLightFrequency>
<WeekDays>0123456</WeekDays>
<BeginTime>00:00:00</BeginTime>
<EndTime>23:59:59</EndTime>
</IdleScheduleConfig>
```

2.7 System Parameters

2.7.1 Get/Set NTP

/ISAPI/System/NTP?ID=SessionID			
POST			
Description	Get NTP,POST request with empty message body. Normal response returns the following XML message content		
PUT			
Description	Set NTP, PUT request with the following XML message content. Normal response: 200 OK		
Field	Type	Description	Required
TZ	string	Time Zone e.g.: GMT+08:00 or GMT-08:00	No
IsUpdateTime	int	Update time: 0: No 1: Yes	No
Enabled	int	0: Disabled 1: Enabled	No
DST	int	Daylight Saving Time: 0: No 1: Yes	No
UTCDateTime	string	TZ format, e.g.: 2018-07-01T18:00:00Z	No
NTPServer	string	NTP server	No

```
<NTPStatus>
    <TimeZone>
        <TZ><!-- req, xs: string --></TZ>
    </TimeZone>
    <IsUpdateTime><!-- req, xs: int --></IsUpdateTime>
    <Enabled><!-- required, xs: integer --></Enabled>
    <DST><!-- required, xs: integer --></DST>
    <UTCDateTime><!-- required, xs: string --></UTCDateTime>
    <NTPServer><!-- required, xs: string --></NTPServer>
</NTPStatus>
```

2.7.2 Restore Factory Settings

/ISAPI/System/RestoreDefault?ID=SessionID

PUT	
Description	Restore factory settings. The POST request contains the following XML message content
Field Description Enabled: Factory reset (0: Disabled 1: Enabled)	
<RestoreDefault> <Enabled><!-- req, xs: int --></Enabled> </RestoreDefault>	

2.7.3 Device Reboot

/ISAPI/System/Reboot?ID=SessionID	
PUT	
Description	Device restart. POST request with the following XML message content
Field Description Enabled: Device reboot (0: Disabled 1: Enabled)	
<Reboot> <Enabled><!-- req, xs: int --></Enabled> </Reboot>	

2.7.4 Log Query

/ISAPI/Log/GetLogMainType?ID=SessionID			
POST			
Description	Get Log MainType.		
GetLogMainType	int	0:All Type 1:Operation 2:Alarm 3:Information 4:Abnormal	Yes
<GetLogMainType> <Count>5</Count> <MainPara0> <MainTypeDesc>All Type</MainTypeDesc> <MainType>0</MainType> </MainPara0> <MainPara1> <MainTypeDesc>Operation</MainTypeDesc> <MainType>1</MainType> </MainPara1> <MainPara2>			

<MainTypeDesc>Alarm</MainTypeDesc>
<MainType>2</MainType>
</MainPara2>
<MainPara3>
<MainTypeDesc>Information</MainTypeDesc>
<MainType>3</MainType>
</MainPara3>
<MainPara4>
<MainTypeDesc>Abnormal</MainTypeDesc>
<MainType>4</MainType>
</MainPara4>
</GetLogMainType>

/ISAPI/Log/QueryLog?ID=SessionID

POST

Description	Query log		
Field	Type	Description	Required
StartTime	string	Start time: 1 July 2018 18:00:00	Yes
EndTime	string	End Time: 1 July 2018 18:00:00	Yes
Logmaintype	int	Type: 0:All Type 1:Operation 2:Alarm 3:Information 4:Abnormal	Yes

Request XML

<QueryLog>
<logmaintype><!-- req, xs: string --></logmaintype>
<starttime><!-- req, xs: string --></starttime>
<endtime><!-- req, xs: string --></endtime>
</QueryLog>

Response XML

<QueryLog>
<Count><!-- req, xs: string --></Count>
<LogPara0>
<LogMainType><!-- req, xs: string --></LogMainType>
<LogSubType><!-- req, xs: string --></LogSubType>
<LogUserName><!-- req, xs: string --></LogUserName>
<LogMainTypeDescribe><!-- req, xs: string --></LogMainTypeDescribe>
<LogSubTypeDescribe><!-- req, xs: string --></LogSubTypeDescribe>

```
            <LogTime><!-- req, xs: string --></LogTime>
        </LogPara0>
        ...
        <LogPara n>
        ...
        </LogPara n>
<QueryLog>
```

2.8 User Management

2.8.1 User Configuration

/ISAPI/UserManagement/UsrConfig?ID=SessionID			
POST			
Description	Get user information. POST request with empty message body. Normal response returns the following XML message content		
PUT			
Description	Set user information		
Field	Type	Description	Required
UserName	string	Username (maximum 32 characters)	Yes
Password	string	Password (maximum 32 characters)	Yes
Privilege	int	0: Administrator 1:Maintenancepersonnel 2: Ordinary User	Yes

```
<UserManagement >
    <UserName ><!-- req, xs: string--></ UserName >
    <Password ><!-- req, xs: string--></Password >
    <Privilege ><!-- req, xs: int --></Privilege >
</UserManagement >
```

2.8.2 Add User

/ISAPI/UserManagement/AddUsr?ID=SessionID			
PUT			
Description	Add User		
Field	Type	Description	Required
UserName	string	Username (maximum	Yes

		32 characters)	
Password	string	Password (maximum 32 characters)	Yes
Privilege	int	0: Administrator 1:Maintenance personnel 2: Ordinary User	Yes
<AddUsr> <UserName ><!-- req, xs: string--></ UserName > <PassWord ><!-- required, xs: string--></PassWord > <Privilege ><!-- required, xs: integer --></Privilege > </AddUsr>			

2.8.3 Delete User

/ISAPI/UserManagement/DelUsr?ID=SessionID			
PUT			
Description	Delete User		
Field	Type	Description	Required
UserName	string	Username (maximum 32 characters)	Yes
<DelUsr> <UserName ><!-- req, xs: string--></ UserName > </DelUsr>			

2.8.4 Reset User Password

/ISAPI/UserManagement/ResetPassword			
PUT			
Description	Resetting the user password requires a device restart within 30 seconds after sending the command for the changes to take effect.		
Field	Type	Description	Required
UserName	string	Username (maximum 32 characters)	Yes
PassWord	string	Password (maximum 32 characters)	Yes
<ResetPassword> <UserName><!-- req, xs: string--></UserName> <PassWord><!-- req, xs: string--></PassWord> </ResetPassword>			

Note: Maximum of 30 users.

3 Device Search

3.1 Device Search

Protocol Type: UDP Broadcast
Protocol Port: 3704
Broadcast IP Address: 255.255.255.255

UDP			
Description	Device Discovery		
Field	Type	Description	Required
DeviceName	string	Device Name	Yes
DeviceModel	string	Device Model	Yes
SerialNumber	string	Serial Number	Yes
SoftwareVersion	string	Software Version	Yes
IP Address	string	IP Address	Yes
MacAddress	string	MAC address	Yes
SubnetMask	string	Subnet Mask	Yes
Gateway	string	Gateway	Yes
DNSIPAddress 1	string	Preferred DNS	Yes
DNSIPAddress 2	string	Alternate DNS	Yes
HTTPPort	int	HTTP port	Yes
HTTPsPort	int	HTTPS port	Yes
ONVIFPort	int	ONVIF port	Yes
RTSPPort	int	RTSP port	Yes
DHCPEnabled	int	0: Disabled 1: Enabled	Yes
Request XML <DiscoveryReq/>			
Response XML <DiscoveryRes> <DeviceName><!-- req, xs: string --></DeviceName> <DeviceModel><!-- req, xs: string --></DeviceModel> <SerialNumber><!-- req, xs: string --></SerialNumber> <SoftwareVersion><!-- req, xs: string --></SoftwareVersion> <IPAddr><!-- required, xs: string --></IPAddr> <MacAddr><!-- required, xs: string --></MacAddr> <SubnetMask><!-- required, xs: string --></SubnetMask> <Gateway><!-- required, xs: string --></Gateway>			

```
    <DNSIpAddr1><!-- required, xs: string --></DNSIpAddr1>
    <DNSIpAddr2><!-- req, xs: string --></DNSIpAddr2>
    <HTTPPort><!-- req, xs: int --></HTTPPort>
    <HTTPsPort><!-- required, xs: integer --></HTTPsPort>
    <ONVIFPort><!-- required, xs: integer --></ONVIFPort>
    <RTSPPort><!-- required, xs: integer --></RTSPPort>
    <DHCPEnabled><!-- required, xs: integer --></DHCPEnabled>
</DiscoveryRes>
```

3.2 Modify device information

Protocol Type: UDP Broadcast
Protocol port: 3704
Broadcast IP Address: 255.255.255.255

UDP

Description	Modify device information		
Field	Type	Description	Required
DeviceUsername	string	Username	DevicePassword
DevicePassword	string	Password	Yes
IPAddress	string	IP address	Yes
MacAddress	string	MAC Address	Yes
NewIP	string	New IP	Yes
SubnetMask	string	Subnet Mask	Yes
Gateway	string	Gateway	Yes
DHCPEnabled	int	0: Disabled 1: Enabled	Yes
DNSIPAddress1	string	Preferred DNS	No
DNSIPAddress2	string	Alternate DNS	No

Request XML

```
<ModifyDeviceReq>
    <DeviceUsername><!-- req, xs: string --></DeviceUsername>
    <DevicePassword><!-- required, xs: string --></DevicePassword>
    <IPAddr><!-- required, xs: string --></IPAddr>
    <MacAddr><!-- required, xs: string --></MacAddr>
    <NewIP><!-- required, xs: string --></NewIP>
    <SubnetMask><!-- required, xs: string --></SubnetMask>
    <Gateway><!-- required, xs: string --></Gateway>
    <DHCPEnabled><!-- required, xs: integer --></DHCPEnabled>
    <DNSIPAddr1><!-- required, xs: string --></DNSIPAddr1>
    <DNSIPAddr2><!-- required, xs: string --></DNSIPAddr2>
</ModifyDeviceReq>
```

UDP

Description	Modify device information response		
Field	Type	Description	Required
Result	string	Result (Success/Fail)	Yes
MAC	string	Device MAC address	Yes
IPAddr	string	Device IP address	Yes

Response XML

```
<ModifyDeviceRes>
    <Result><!-- req, xs: string --></Result>
    <MAC><!-- req, xs: string --></MAC>
    <IPAddr><!-- req, xs: string --></IPAddr>
</ModifyDeviceRes>
```

4 Alarm Linkage

4.1 Configure HTTP integration

/ISAPI/AlarmLink/HttpLinkConfig?ID=SessionID			
POST			
Description	HTTP message transmission; POST request with empty message body; normal response returns the following XML message content		
PUT			
Description	HTTP message configuration: PUT request with the following XML message content, returns 200 OK as expected.		
Field	Type	Description	Required
Enabled	int	0: Disable 1: Enabled	No
Server	string	Server	No
AuthMethod	int	0:None 1:Private Key Value Pairs 2:Bearer Authentication 3: Basic Authentication	No
AuthKey	string	AuthKey	No
AuthValue	string	AuthValue	No

```
<HttpLinkConfig>
    <Enabled><!-- req, xs: int --></Enabled>
    <Server><!-- required, xs: string --></Server>
    <AuthMethod><!-- required, xs: string --></AuthMethod>
    <AuthKey><!-- req, xs: string --></AuthKey>
    <AuthValue><!-- req, xs: string --></AuthValue>
</HttpLinkConfig>
```

4.2 Configure Broadcast Link

/ISAPI/AlarmLink/BroadcastLinkConfig?ID=SessionID			
POST			
Description	Get BroadcastConfig message transmission. POST request with empty message returns the following XML message content as normal		
PUT			
Description	Configure BroadcastConfig message sending. PUT request with the following XML message content. Normal response: 200 OK		
Field	Type	Description	Required
BroadcastType	int	0: Disabled 1: Full Network 2: IP List 3: IP Segment	No
IpCount	int	IpCount	
IpAddr_n-1	string	Each IP separated by a line break	No
BeginIp	string	Start IP	
EndIp	string	End IP	
IsAcceptBroadcast	int	Accept Broadcast Alarm Messages from Other Devices 0: Disabled 1: Enabled	No

```
<BroadcastLinkConfig>
  <BroadcastType><!-- req, xs: int --></BroadcastType>
  <IpCount><!-- req, xs: int --></IpCount>
 <IpAddr_0><! -- req, xs: string --></IpAddr_0>
  ...
  <IpAddr_n-1><! -- req, xs: string --></IpAddr_n-1>
  <BeginIp><! -- req, xs: string --></BeginIp>
  <EndIp><! -- req, xs: string --></EndIp>
  <IsAcceptBroadcast><!-- req, xs: int --></IsAcceptBroadcast>
</BroadcastLinkConfig>
```

4.3 Alarm Content

/ISAPI/AlarmLink/AlarmLinkMsg?ID=SessionID
POST

Description	Alarm message transmission		
Field	Type	Description	Required
DeviceID	int	Device ID	No
SoftwareVer	string	Software Version	No
MAC Address	string	MAC Address	No
IPAddress	string	IP Address	No
AlarmType	int	0: GPIO 1:HTTP	No
AlarmTime	string	Time (TZ format, e.g.: 2018-07-01T18:00:00Z)	No

```
<AlarmLinkMsg>
   <DeviceID><! -- req, xs: int --></DeviceID>
   <SoftwareVer><! -- required, xs: string --></SoftwareVer>
   <MACAddress><! -- req, xs: string --></MACAddress>
   <IPAddress><! -- required, xs: string --></IPAddress>
   <AlarmType><! -- required, xs: integer --></AlarmType>
   <AlarmTime><! -- required, xs: string --></AlarmTime>
</AlarmLinkMsg>
```

4.4 Configure MQTT

/ISAPI/AlarmLink/MqttLinkConfig?ID=SessionID			
POST			
Description	Alarm message transmission		
Field	Type	Description	Required
Enabled	int	0: Disable 1: Enabled	No
Port	string	Port	No
ClientId	string	Client Id	No
User	string	Username (maximum 32 characters)	No
Password	int	Password (maximum 32 characters)	No
Topic	string	Topic：device/mac/alarm	No
QoS	string	0: Send at most once 1: Send at least once 3:Exactly once	No
PublishOnEnable	string	0: Disable 1: Enabled	No
PublishOffEnable	int	0: Disable 1: Enabled	No
SetString	string	Set String	No

ResetString	string	Reset String	No
TriggerString	string	Trigger String	No
ReleaseString	string	Release String	No

```
<MqttLinkConfig >
   <Enabled><! -- req, xs: int --></Enabled>
   <Server><! -- required, xs: string --></Server>
   <Port><! -- req, xs: int --></Port>
   <ClientId><! -- req, xs: int --></ClientId>
   <User><! -- required, xs: string --></User>
   <Password><! -- required, xs: string --></Password>
   <Topic><! -- required, xs: string --></Topic>
   <QoS><! -- required, xs: string --></QoS>
   <PublishOnEnable><! -- req, xs: int --></PublishOnEnable>
   <PublishOffEnable><! -- req, xs: int --></PublishOffEnable>
   <SetString><! -- required, xs: string --></SetString>
   <ResetString><! -- required, xs: string --></ResetString>
   <TriggerString><! -- required, xs: string --></TriggerString>
   <ReleaseString><! -- required, xs: string --></ReleaseString>
</MqttLinkConfig>
```