Reference Manual

MOBOTIX MOVE NVR API V3.0.1

HEVCAdvance™

Table of Contents

The **Network Video Recorder Application Programming Interface** is an HTTP-based API for the XMS Series NVR. All these functions could be used to control or set the configuration of the NVR devices. Except Streaming, these API use the same format in transporting HTTP-based message. As for the NVR RTSP API, please refer to the document **MOVE NVR RTSP API**. Here, we only provide the general command description via HTTP connection.

This document specifies all the API of our web service. All the web service commands need to be issued through http/https protocol. The command syntax is as follows:

```
GET: http://{Server IP}/{Action Path}/<command>?<parameter
arg1>&<parameter arg2>&<….>
POST: http://{Server IP}/{Action Path}/<command>
Param={ “parameter arg1”:value1, “parameter arg2”:value2, ….}
```

The braces "{}" in the syntax indicate a command or a parameter. All responses would be the html, xml or JSON format.

1

HTTP Request

An HTTP-based protocol always includes the request and response messages. The web service of NVR would wait and accept HTTP connection request with a specified port and process the requests in order to do some specified response. The common format of request and response messages would be discussed as follows:

Request Message

To query information/set the configuration of NVR with HTTP GET Method, use the syntax:

```
GET http://<server name>/<Action-URL>?<parameter>=<value>
HTTP/1.1<CRLF> Authorization: Digest <digest-cookie><CRLF>
Host: <server ip-adress><CRLF>
…
<CRLF>
```

To query information/set the configuration of NVR with HTTP POST Method, use the syntax:

```
POST http://<server name>/<Action-URL> HTTP/1.1<CRLF> Authorization:
Digest <digest-cookie><CRLF>
Host: <srerver ip-adress><CRLF>
Content-Type: application/x-www-form-urlencoded<CRLF> Content-
Length: <body length><CRLF>
<CRLF>
Param={“parameter 1”:value1, “parameter 2”:value2, …}
…..
```

Response Message

While NVR receives request, it will do the related action then output result as response message:

```
HTTP/1.1 <HTTP code> <HTTP text><CRLF>
Content-Type: text/plain<CRLF> Content-Length: <body length><CRLF>
<CRLF>
<parameter>=<values><CRLF>
```

Response Status Code

HTTP Code	HTTP text	Description
200	OK	The request has accepted successfully. But the command would fail during the process. Please check each command response.
400	Bad Request	Invalid or unsupported parameters or values.
401	Unauthorized	User authentication needed or authorization refused.
404	Not Found	This API is not supported.

500	Internal Error	Internal Error occurred. The NVR encountered an internal error or the client cannot get the correct status.
503	Service	The NVR is not available for the request due to temporary overload.

API above can ONLY be accessed by authorized user. Please deliver the command under Digest authentication. Most of commands will return current status of NVR and terminate the request afterwards.

2

Log in/out Service API

All the reply JSON format of the http request is as follows:

```
{“code”:”ReplyCode”,“msg”:”ReplyContent”,“name”:”APIName”,“output”:”Otherreplydata”}
```

code: Will Be '0' if success. Other values mean failed.

msg: The message reply by the web server

name: The command name of the API.

output: The information request by the HTTP command.

Log in Service

Purpose: Get the user authentication by username and password. This API will reply a JSON message and a special HttpOnly cookie if success. Other API request (exclude the logout) should contain this special cookie to let the server recognize the authority.

Parameters:

account = Encoded Value (md5(username + ‘:’ + ‘password’))

Parameter Example:

Assume username =”Admin” and password = “1234”.

Then the encoded value is md5(“Admin:1234”) = afdfd53c19b43825da4b297c203e4460. So the parameter will be “account = afdfd53c19b43825da4b297c203e4460”

Reply:

JSON format as follows:

```
{“code”:”ReplyCode”,“name”:“APIName”,“msg”:“ReplyContent”}
```

Cookie:

login-already=’Random UUID Value’

Reply Example: JSON:

```
{“code”:”0”,“name”:“/login”,“msg”:“Loginsuccess.”}
```

Cookie:

login-already=’ 48df92601964701eae2b6d4363ea3a79’

Syntax	http://{server_IP}/auth/login
Method	POST
Parameters	account=%s
Example	http://192.168.6.51/login

Log out Service

Purpose: To cancel the user authentication. Reply: JSON format as follows:

```
{“code”:“ReplyCode”,”name”:“APIName”,“msg”:“ReplyContent”}
```

Reply Example:

{“code”: “0”, ”name”: “/logout”, “msg”: “Logout success”}

Syntax	http://{server_IP}/logout
Method	GET

Parameters	NONE
Example	http://192.168.6.51/logout

3

Playback relative API

Playback Key Get

Purpose: To register the authority to retrieve the playback video. This command will return a secure key number to the user for watching the playback video.

NOTE! This key will auto-expire if the server detects all RTSP streams are disconnected.

Parameters:

```
start_time=%d&group_id=%d&dir=%d&speed=%f
```

start_time: The UTC time in seconds

group_id: The group index of disk dir: 0/-1 (Forward/Backward) speed: 1.0 / 2.0/ 0.5 / 0.25 Reply: JSON format as follows:

{“code”: “Reply Code”, ”name”: “API Name”, “msg”: “Reply Content”, “output”:{“pbkey”: %s, “pb_ch”: [%d, …]}}

pbkey: The security playback key.

pb_ch: Channel indexes in an array that represents the recorded channels at that time.

If no channel is recorded, it will be empty array.

Reply Example:

```
{"code":0,"msg":"Success","name":"/datasearch/gen_pbkey","output":
{"pbkey":"02e5c750bd3acce1c17418e768128a68","pb_ch":[0,1,2,3]}}
```

Syntax	http://{server_IP}/datasearch/gen_pbkey
Method	GET
Parameters	start_time=%d&group_id=%d&dir=%d&speed=%f
Example	http://192.168.6.51/datasearch/gen_pbkey?start_time-e=1497481200&dir=0&speed=1&group_id=0

Playback Configuration Set

Purpose: To setup the local playback time and begin to get data from database.

Parameters: JSON format as follows:

```
{pbkey:%s,start_time:%d,dir:%d,speed:%f,group_id:%d}
```

pbkey: Playback key

start_time: The UTC time in seconds group_id: The group index of disk dir: 0/-1 (Forward/Backward) speed: 1.0 / 2.0/ 0.5 / 0.25

Reply:

```
{"code":"ReplyCode","name":"APIName","msg":"ReplyContent","output":
{"pbtime":%d,"group_id":%d,"dir":%d,"speed":%d,"pb_ch":[%d,…]}}
```

pb_ch: Channel indexes in an array that represents the recorded channels at that time.

If no channel is recorded, it will be empty array.

Reply Example:

```
{"code":0,"msg":"Success","name":"/datasearch/set_pbspeed","output":
{"pbtime":1496876400,"group_id":0,"dir":0,"speed":1,"pb_ch":
[0,1,2,3,4]}}
```

Syntax	http://{server_IP}/datasearch/set_pbspeed
Method	POST
Parameters	{pbkey: %s, start_time: %d, dir: %d, speed: %f, group_id: %d}
Example	http://192.168.6.51/datasearch/set_pbspeed

Playback Bar Get

Purpose: To get the available playback time presented by sections.

Parameters:

```
start_time=%d&end_time=%d&section_num=%d&group_id=%d
```

start_time: Start of time sections (local time in UTC seconds)

end_time: End of time sections (local time in UTC seconds)

section_num: Number of sections in the range from start_time to end_time

group_id: Disk group index

Reply: JSON format as follows:

```
{“code”:“ReplyCode”,”name”:“APIName”,“msg”:“ReplyContent”,“output”:
{“group_id”:%d,“database_start_t”:%d,“database_end_t”:%d,“section_
num”:%d,“pb_bar”:[%d,…,%d]}}
```

group_id: Disk group index

database_start_t: Start of time sections (local time in UTC seconds) database_end_t: End of time sections (local time in UTC seconds) section_num: Number of sections in the range from start_time to end_time

pb_bar: An array of Boolean value to indicate that in the section time the database have recorded data or not. (1: recorded, 0: not recorded)

Reply Example:

```
{”code”:0,”msg”:”Success”,”name”:”/datasearch/get_pbbar”,”output”:
{”group_id”:0,”database_start_t”:1495767531,”database_end_
t”:1497849562,”section_num”:42,”pb_bar”:
```

```
[0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,1,1,1,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0]}}
```

Syntax	http://{server_IP}/datasearch/get_pbbar
Method	GET
Parameter	start_time=%d&end_time=%d&section_num=%d&group_id=%d
Example	http://192.168.6.51/datasearch/get_pbbar?start_time=1495767531&end_time=1497849562&section_num=42&group_id=0

Event List Page Get

Purpose: To get the event-list-pages in the local time scale.

Parameters: JSON format as follows:

```
{“group_id”:%d,“evt_type”:[true/false,true/false,true/false],“start_idx”:%d,“end_idx”:%d,“evt_idx”:%d,“channel”:%d,“start_time”:%d,“end_time”:%d}
```

group_id: Disk group index.

evt_type: A length three true/false values array to indicate the server to filter the specific event or not.

start_idx/end_idx: Specify the return events index. For example, if the number of the searching events is 100000, but you only want to check the first 5 events, then you can set start_idx=0 and end_idx=4.

evt_idx: Filter the events with event index (optional). For example, the event index of “Alarm 2” event is 2. channel: Filter the events with channel index (optional).

start_time/end_time: Filter the events with the local time in UTC seconds (optional). If not specify the value, server will search from the start/end of the database.

Reply: JSON format as follows:

```
{“code”:“ReplyCode”,”name”:“APIName”,“msg”:“ReplyContent”,“output”:{“total”:%d,“start_idx”:%d,“end_idx”:%d,“evt_list”:[{“start_
```

```
time":%d,"channel":%d,"dev_type":%d,"evt_type":%d,"evt_
indx":%d,"ip_addr":%s,"mac":%s,"snap_path":%s},…]}}
```

total: The total number of finding events.

start_idx/end_idx: The current return start/end index of events.

evt_list:

- **start_time:** The triggered time of the event.
- **channel:** The channel index of the event.
- **dev_type:** The device type of the event. (Check out the device type definition at "Camera relative API" section)
- **evt_type:** The event type. (0: motion, 1: video loss, 2: alarm)
- **evt_indx:** The index of the event.
- **ip_addr:** The IP address of the event.
- **mac:** The MAC address of the event.
- **snap_path:** The snapshot path of the event.

Reply Example:

```
{"code":0,"msg":"evtlist_page_get
Success!","name":"/datasearch/evtlist_page_get","output":
{"total":2,"start_idx":0,"end_idx":1,"evt_list":[{ "start_
time":1497852923,"channel":3,"dev_type":0,"evt_type":1,"evt_
indx":0,"ip_addr":"","mac":"","snap_path":""},{"start_
time":1497852943,"channel":4,"dev_type":0,"evt_type":1,"evt_
indx":0,"ip_addr":"","mac":"","snap_path":""}]}}
```

Syntax	http://{server_IP}/datasearch/evtlist_page_get
Method	POST
Parameters	**{"group_id":**%d, "evt_type":[true/false, true/false, true/false], "start_idx":%d, "end_idx":%d, "evt_idx":%d, "channel":%d, "start_time":%d, "end_time":%d }
Example	http://192.168.6.51/datasearch/evtlist_page_get

System Log Page Get

Purpose: To get the system-log-pages in the local time scale.

Parameters: JSON format as follows:

```
{“start_idx”:%d,“end_idx”:%d,“start_time”:%d,“end_time”:%d}
```

start_time/end_time: Filter the events with the local time in UTC seconds.

start_idx/end_idx: Specify the return events index. For example, if the number of the searching events is 100000, but you only want to check the first 5 events, then you can set start_idx=0 and end_idx=4.

Reply: JSON format as follows:

```
{“code”:“ReplyCode”,”name”:“APIName”,“msg”:“ReplyContent”,“output”:
{“total”:%d,“log_list”:
[{“category”:%s,“description”:%s,“host”:%s,“time”:%d,“type”:%s},…]}}
```

total: The number of total system log events

log_list:

- **category:** Category of the event.
- **description:** Description of the event.
- **host:** Host of the event.
- **time:** UTC time of the event.
- **type:** Type of the event.

Reply Example:

```
{”code”:0,”msg”:”getlogsuccess.”,”name”:”/datasearch/sys_log_page_
get”,”output”:{”log_list”:
[{”category”:”System”,”description”:”PowerLossBoot”,”host”:”Local”,”
time”:1497338936,”type”:”PowerOn”},
{”category”:”System”,”description”:”SystemPowerOn”,”host”:”Local”,”t
ime”:1497338936,”type”:”PowerOn”}],”total”:2}}
```

Syntax	http://{server_IP}/datasearch/sys_log_page_get
Method	POST
Parameters	{ “start_idx”:%d, “end_idx”:%d, “start_time”:%d, “end_time”:%d }
Example	http://192.168.6.51/datasearch/sys_log_page_get

Export Channel List Get

Purpose: To get the available export channels depends on the time range.

Parameters:

```
start_time=%d&end_time=%d&group_id=%d
```

start_time/end_time: The start/end local time in UTC seconds.

group_id: Disk group index.

Reply: JSON format as follows:

```
{“code”:“ReplyCode”,”name”:“APIName”,“msg”:“ReplyContent”,“output”:
{“channel_list”:[%d,…,%d]}}
```

channel_list: An array of available channel indexes.

Reply Example:

```
{”name”:”/datasearch/estimate_export_channel_
list”,”code”:0,”msg”:”Success”,”output”:{”channel_list”:[0,3,4,5]}}
```

Syntax	http://{server_IP}/datasearch/estimate_export_channel_list
Method	GET
Parameters	start_time=%d&end_time=%d&group_id=%d
Example	http://192.168.6.51/datasearch/estime_export_channel_list?start_time=1497855791&end_time=1497855925&group_id=0

Export File

Purpose: To export playback data into the downloadable file.

Parameters: JSON format as follows:

```
{“ch”:%d,“start_time”:%d,“end_time”:%d,“stream_type”:%s,“media_
fmt”:%s,“group_id”:%d}
```

ch: Channel index of data to export.

start_time/end_time: Start/End local time in UTC seconds to export.

stream_type: Stream type of data to export.

media_fmt: Media format of data to export.

group_id: Disk group index.

Reply: JSON format as follows:

```
{“code”:“ReplyCode”,”name”:“APIName”,“msg”:“ReplyContent”}
```

Reply Example:

```
{”code”:0,”msg”:”Success”,”name”:”/datasearch/remote_export_async”}
```

Syntax	http://{server_IP}/datasearch/remote_export_async
Method	POST
Parameters	{“ch”:%d, “start_time”:%d, “end_time”:%d, “stream_type”:%s, “media_fmt”:%s, “group_id”:%d}
Example	http://192.168.6.51/datasearch/remote_export_async

Export File Status Get

Purpose: To check the status of export data.

Parameters: None.

Reply: JSON format as follows:

```
{“code”:“ReplyCode”,”name”:“APIName”,“msg”:“ReplyContent”,“output”:{“status”:%s}}
```

status: “working”/ “idle” (NVR is exporting data./NVR is NOT exporting data).

Reply Example:

```
{”code”:0,”msg”:”Success”,”name”:”/datasearch/remote_export_status”,”output”:{”status”:”working”}}
```

Syntax	http://{server_IP}/datasearch/remote_export_status
Method	GET
Parameters	None.
Example	http://192.168.6.51/datasearch/remote_export_status

Export File List Get

Purpose: To get the downloadable export file list.

Parameters: None.

Reply: JSON format as follows:

```
{“code”:“ReplyCode”,”name”:“APIName”,“msg”:“ReplyContent”,“output”:
{“list”:[{“file_name”:%s,“size_mb”:%s},…]},“export_size”:{“total_
size_mb”:%d,“available_size_mb”:%d}}
```

list:

- **file_name:** File name of each file.
- **size_mb:** File size of each file in MB.

export_size:

- **total_size_mb:** Total file size.
- **available_size_mb:** Available disk size in MB.

Reply Example:

```
{”code”:0,”msg”:”Success”,”name”:”/datasearch/remote_export_
list”,”output”:{”list”:[{”file_name”:”ch0001-170620005913-103-
D.mp4”,”size_mb”:”2.50”},{”file_name”:”ch0001-170620005913-103-
D.srt”,”size_mb“:”0.01”},{”file_name”:”ch0001-170620010225-103-
D.mp4”,”size_mb”:”2.51”},{”file_name”:”ch0001-170620010225-103-
D.srt”,”size_mb”:”0.01”}],”export_size”:{”total_size_
mb”:4833,”available_size_mb”:4347}}}
```

Syntax	http://{server_IP}/datasearch/remote_export_list
Method	GET
Parameters	None.
Example	http://192.168.6.51/datasearch/remote_export_list

Export File Remove Set

Purpose: To remove the downloadable export file list from disk.

Parameters: JSON format as follows:

```
{“name”:%s}
```

name: A file name to remove.

Reply: JSON format as follows:

```
{“code”:“ReplyCode”,”name”:“APIName”,“msg”:“ReplyContent”,“output”:
{“list”:[{“file_name”:%s,“size_mb”:%s},…]},“export_size”:{“total_
size_mb”:%d,“available_size_mb”:%d}}
```

list:

- **file_name:** File name of each file.
- **size_mb:** File size of each file in MB.

export_size:

- **total_size_mb:** Total file size.
- **available_size_mb:** Available disk size in MB.

Reply Example:

```
{”code”:0,”msg”:”Success”,”name”:”/datasearch/remove_remote_export_
list”,”output”:{”list”:[{”file_name“:”ch0001-170620005913-103-
D.mp4”,”size_mb”:”2.50”},{”file_name”:”ch0001-170620005913-103-
D.srt”,“size_mb”:”0.01”}],”export_size”:{”total_size_
mb”:4833,”available_size_mb”:4349}}}
```

Syntax	http://{server_IP}/datasearch/remove_remote_export_list
Method	POST
Parameters	{“name”:%s}
Example	http://192.168.6.51/datasearch/remove_remote_export_list

4

Information relative API

NVR Version Get

Purpose: To get the NVR firmware version.

Parameters: None.

Reply: JSON format as follows:

```
{“code”:“ReplyCode”,”name”:“APIName”,“msg”:“ReplyContent”,output:
{remote_version:%s,ap_version:%s,num_ch:%d,num_alarm:%d,num_
relay:%d,model:%s,proj:%s}}
```

remote_version: Remote version.

ap_version: Firmware version.

num_ch: Max channel number.

num_alarm: Max alarm number.

num_relay: Max relay number.

model: Model name.

proj: Project name.

Reply Example:

```
{"code":0,"msg":"/information/versionSuccess!","name":"version","out
put":{"remote_version":"2016.11.16.b162cbc","ap_
version":"A99999999999","num_ch":32,"num_alarm":4,"num_
relay":2,"model":"H100","proj":"NVR"}}
```

Syntax	http://{server_IP}/information/version
Method	GET
Parameters	NONE
Example	http://192.168.6.51/information/version

Local Time Get

Purpose: To get the local time of the NVR.

Reply: JSON format as follows:

```
{"code":"ReplyCode","name":"APIName","msg":"ReplyContent","output":
{"local_time":%d}}
```

local_time: The local time in UTC seconds.

Reply Example:

```
{"code":0,"msg":"Success","name":"/information/local_time","output":
{"local_time":1497412035}}
```

Syntax	http://{server_IP}/information/local_time
Method	GET
Parameters	NONE
Example	http://192.168.6.51/information/local_time

Network Information Get

Purpose: To get the network information.

Parameters: None.

Reply: JSON format as follows:

```
{“code”:“ReplyCode”,”name”:“APIName”,“msg”:“ReplyContent”,“output”:
{“http_port”:%d,“https_port”:%d,“rtsp_port”:%d,“network_card_info”:
[{“name”:%s,“ip”:%s,“mac”:%s,“dhcp_status”:%d,“net_
mask”:%s,“gateway”:%s,“primary_dns”:%s,“secondary_dns”:%s},…]}}
```

http_port: HTTP port of remote web.

https_port: HTTPS port of remote web

rtsp_port: RTSP port of NVR

network_card_info:

- **name:** Name of the network card
- **ip:** IP address of the network card
- **mac:** MAC address of the network card
- **dhcp_status:** 1: Enable DHCP, 0: Disable DHCP
- **net_mask:** Subnet mask of the network card
- **gateway:** Default gateway of the network card
- **primary_dns:** Primary DNS of the network card
- **secondary_dns:** Secondary DNS of the network card

Reply Example:

```
{”code”:0,”msg”:”/information/network_infoSuccess”,”name”:”network_
info”,”output”:{”http_port”:80,”https_port”:443,”rtsp_
port”:554,”network_card_info”:
[{”name”:”eth0”,”ip”:”192.168.6.146”,”mac”:”00:D0:89:12:40:43”,”dhc
p_status”:true,”net_
mask“:”255.255.255.0”,”gateway”:”192.168.6.254”,”primary_
dns”:”192.168.10.1”,”secondary_dns”:”192.168.10.5”},
{”name”:”eth1”,”ip”:”192.168.50.19”,”mac”:”00:D0:89:12:40:44”,”dhc
p_status”:false,”net_mask”:“255.255.255.0”,”gateway”:””,”primary_
dns”:””,”secondary_dns”:””}]}}
```

Syntax	http://{server_IP}/information/network_info
Method	GET
Parameters	NONE
Example	http://192.168.6.51/information/network_info

User Information Get

Purpose: To get the user information.

Parameters: None.

Reply: JSON format as follows:

```
{“code”:“ReplyCode”,”name”:“APIName”,“msg”:“ReplyContent”,“output”:
{“user”:[{“account”:%s,“password”:%s,“authority”:%d,“covert”:
[true/false,…]},…]}}
```

user:

- **account:** User account
- **password:** User encrypted password
- **authority:** The decimal value conversion from the 5-bits binary value which is mapping to the five authorities (Setup, Live(always 1), PTZ, Device, Playback) is permitted or not.
- **covert:** [true/false, …, true/false], (covert/ not covert) the channels from the account.

> **NOTE!** Please check out the appendix to see how to encrypt/decrypt the password.

Reply Example:

```
{”code”:0,”msg”:”/information/user_
managementgetSuccess!”,”name”:”/information/user_
management”,”output”:{”user”:
[{”account”:”Guest”,”authority”:8,”covert”:
[false,false,false,false,false,false,false,false,false,false,false,f
alse,false,false,false,false,false,false,false,false,false,false,fal
se,false,false,false,false,false,false,false,false,false,false,false
```

```
,false,false,false,false,false,false,false,false,false,false,false,
false,false,false,false,false,false,false,false,false,false,false,f
alse,false,false,false,false,false,false,false,false,false,false,fa
lse,false,false,false,false,false,false,false,false,false,false,fal
se,false,false,false,false,false,false,false,false,false,false,fals
e,false,false,false,false,false,false,false,false,false,false,false
,false,false,false,false,false,false,false,false,false,false,false,
false,false,false,false,false,false,false,false,false,false,false,f
alse,false,false,false,false],”password”:””},
{”account”:”Admin”,”authority”:31,”covert”:
[false,false,false,false,false,false,false,false,false,false,false,
false,false,false,false,false,false,false,false,false,false,false,f
alse,false,false,false,false,false,false,false,false,false,false,fa
lse,false,false,false,false,false,false,false,false,false,false,fal
se,false,false,false,false,false,false,false,false,false,false,fals
e,false,false,false,false,false,false,false,false,false,false,false
,false,false,false,false,false,false,false,false,false,false,false,
false,false,false,false,false,false,false,false,false,false,false,f
alse,false,false,false,false,false,false,false,false,false,false,fa
lse,false,false,false,false,false,false,false,false,false,false,fal
se,false,false,false,false,false,false,false,false,false,false,fals
e,false,false,false,false,false],”password”:”GtBc9p/Z”}]}}
```


Syntax	http://{server_IP}/information/user_management
Method	GET
Parameters	NONE
Example	http://192.168.6.51/information/user_management

Time Configuration Get

Purpose: To get the time configuration.

Reply: JSON format as follows:


```
{“code”:“ReplyCode”,”name”:“APIName”,“msg”:“ReplyContent”,"output":
{"date_format":%s,"dst":{”enabled”:true/false,”end_day”:%d,”end_
hour”:%d,”end_minute”:%d,”end_month”:%d,“end_second”:%d,“end_
week”:%d,“end_weekday”:%d,“offset”:%d,“rule”:%d,“start_
```

```
day":%d,"start_hour":%d,"start_minute":%d,"start_month":%d,"start_
second":%d,"start_week":%d,"start_weekday":%d},"iana_id":"","ntp_
server":%s,"timezone_index":%d,"timezone_list":[%s,…],"timezone_
offset":%d,"use_24hr":true/false}}
```

date_format: YYYY/MM/DD, YYYY/DD/MM, MM/DD/YYYY, DD/MM/YYYY

ntp_server: NTP server address

timezone_index: Index at timezone_list

timezone_list: Display time zone list in an array

iana_id: The name of time zone in the IANA (Internet Assigned Numbers Authority) time zone database

timezone_offset: The time difference between the UTC time in seconds. (example: UTC+1.00 => -3600)

use_24hr: true/false. (24-hour clock/ 12-hour clock)

dst:

- **enabled:** Enable DST or not.
- **offset:** DST time offset in seconds.
- **rule:**
  - **0:** DST format in relative day of month. (ex: September's 5th week's Sunday 2:45 ~ April's 1st week's Sunday 3:45)
  - **1:** DST format in absolute date. (ex: September 21 00:00 ~ March 21 00:00)
- **start_day/end_day:** If rule is 1, this value represent the counted days from the beginning of the year(not consider leap year), for example 264 means 9/21. If rule is 0, this value is 0.
- **start_hour/end_hour:** The DST start/end hours.
- **start_minute/end_minute:** The DST start/end minutes.
- **start_month/end_month:** If rule is 0, this value represent the DST start/end month. If rule is 1, this value is 0.
- **start_week/end_week:** If rule is 0, this value represent the DST start/end week. If rule is 1, this value is 0.
- **start_weekday/end_weekday:** If rule is 0, this value represent the DST start/end weekday. If rule is 1, this value is 0.

Reply Example:

```
{"code":0,"msg":"gettime_configsuccess.","name":"/information/time_
config","output":{"date_format":"YYYY/MM/DD","dst":
{"enabled":true,"end_day":0,"end_hour":2,"end_minute":0,"end_
```

```
month”:9,”end_second”:0,”end_week”:4,”end_
weekday”:0,”offset”:3600,”rule”:0,”start_day”:0,”start_
hour”:1,”start_minute”:0,”start_month”:2,”start_second”:0,”start_
week”:4,”start_weekday”:0},”iana_id”:””,”ntp_
server”:”time.nist.gov”,”timezone_index”:-1,”timezone_list”:
[],”timezone_offset”:0,”use_24hr”:false}}
```


Syntax	http://{server_IP}/information/time_config
Method	GET
Parameters	NONE
Example	http://192.168.6.51/information/time_config

5

Setup relative API

Default Connection Setting Get

Purpose: To get the default connection setting.

Parameters: None.

Reply: JSON format as follows:

```
{“code”:“ReplyCode”,”name”:“APIName”,“msg”:“ReplyContent”,“outpu
t”:{“auto_add”:%d,“auto_refresh”:%d,“auto_refresh_
time”:%d,“default_info”:
[{“account”:%s,“password”:%s,“type”:%d},…],“support_cam”:
[%d,…,%d]}}
```

auto_add: Enable/Disable(1/0) auto add device function.

auto_refresh: Enable/Disable(1/0) auto refresh device list function. (Refresh local device list only.)

auto_refresh_time: Time interval to refresh device list in seconds.

default_info:

- **account:** Default user account for the device type.
- **password:** Default user encrypted password for the device type. (To decrypt the password, check out the appendix.)
- **type:** Device type index. (Check out the definition of device type in “Camera relative API” section.)

  support_cam: An array of integer index to specify the NVR support device type. (Check out the definition of device type in “Camera relative API” section.)

Reply Example:

```
{”code”:0,”msg”:”device_default_
accountgetsuccess”,”name”:”/setup/default_connection_info”,”output”:
{”auto_add”:false,”auto_refresh”:false,”auto_refresh_
time”:120,”default_info”:
[{”account”:”Admin”,”password”:”GtBcsw==”,”type”:0},
{”account”:”Admin”,”password”:”GtBcs92K”,”type”:2},
{”account”:”Admin”,”password”:”GtBcsw==”,”type”:3}],”support_cam”:
[0,1,3,12]}}
```

Syntax	http://{server_IP}/setup/default_connection_info
Method	GET
Parameters	NONE
Example	http://192.168.6.51/setup/default_connection_info

Default Connection Setting Set

Purpose: To set the default connection setting.

Parameters: JSON format as follows:

```
{”device_default_list”:[{”type”:%d,”account”:%s,”password”:%s},…]}
```

type: Device type index.

account: New default user account for the device type.

password: New default encrypted password for the device type. (To encrypt the password, check out the appendix.)

Reply: JSON format as follows:

```
{“code”:“ReplyCode”,”name”:“APIName”,“msg”:“ReplyContent”,“output”:
{“auto_add”:%d,“auto_refresh”:%d,“auto_refresh_time”:%d,“default_
info”:[{“account”:%s,“password”:%s,“type”:%d},…],“support_cam”:
[%d,…,%d]}}
```

auto_add: Enable/Disable(1/0) auto add device function.

auto_refresh: Enable/Disable(1/0) auto refresh device list function. (Refresh local device list only.)

auto_refresh_time: Time interval to refresh device list in seconds.

default_info:

- **account:** Default user account for the device type.
- **password:** Default user encrypted password for the device type. (To decrypt the password, check out the appendix.)
- **type:** Device type index. (Check out the definition of device type in “Camera relative API” section.)

support_cam: An array of integer index to specify the NVR support device type. (Check out the definition of device type in “Camera relative API” section.)

Reply Example:

```
{”code”:0,”msg”:”Success.”,”name”:”/setup/default_connection_
info”,”output”:{”auto_add”:false,”auto_refresh”:false,”auto_
refresh_time”:120,”default_info”:
[{”account”:”Admin”,”password”:”1234”,”type”:0},
{”account”:”AdminX”,”password”:”123456”,”type”:2},
{”account”:”AdminY”,”password”:”1234”,”type”:3}],”support_cam”:
[0,1,3,12]}}
```

Syntax	http://{server_IP}/setup/default_connection_info
Method	POST
Parameters	{"device_default_list":[{"type":%d,"account":%s,"password":%s}, …]}
Example	http://192.168.6.51/setup/default_connection_info

Load Default Setting

Purpose: To load the factory default to NVR.

Parameters: None.

Reply: JSON format as follows:

```
{“code”:“ReplyCode”,”name”:“APIName”,“msg”:“ReplyContent”}
```

Reply Example:

```
{“code”:0,”name”:“/setup/load_default”,“msg”:”/setup/load_
defaultSuccess!”}
```

Syntax	http://{server_IP}/setup/load_default
Method	GET/POST
Parameters	NONE
Example	http://192.168.6.51/setup/load_default

System Monitor Configuration Get

Purpose: To get the system monitor configuration.

Parameters: None.

Reply: JSON format as follows:

```
{“code”:“ReplyCode”,”name”:“APIName”,“msg”:“ReplyContent”,“output”:
{“system_monitor_setting”:{“cpu_fan”:{“alarm_out”:[%d,…],“send_
email”:%d,“threshold”:%d},“cpu_temperature”:{“alarm_out”:
[%d,…],“send_email”:%d,“threshold”:%d},“disk_failed”:{“alarm_out”:
[%d,…],“send_email”:%d,“threshold”:%d},“disk_full”:{“alarm_out”:
[%d,…],“send_email”:%d,“threshold”:%d},“disk_slow”:{“alarm_out”:
[%d,…],“send_email”:%d,“threshold”:%d}},“system_monitor_list”:[“cpu_
fan”,“cpu_temperature”,“disk_failed”,“disk_full”,“disk_slow”]}}
```

system_monitor_setting:

- **cpu_temperature:**
  - **threshold:** This integer value represent the CPU temperature threshold in Celsius.
  - **send_email:** Send the email or not (1 or 0) if the CPU temperature over the threshold.
  - **alarm_out:** Trigger the alarm out or not (1 or 0) If the CPU temperature over the threshold. The position in the array means the index of alarm. For example, alarm_out [0] means “Alarm 1”.
- **cpu_fan:**
  - **threshold:** This integer value represent the CPU fan speed threshold in RPM.
  - **send_email:** Send the email or not (1 or 0) if the CPU fan speed over the threshold.
  - **alarm_out:** Trigger the alarm out or not (1 or 0) if the CPU fan speed over the threshold. The position in the array means the index of alarm. For example, alarm_out [0] means “Alarm 1”.
- **disk_failed:**
  - **threshold:** This value us unused.
  - **send_email:** Send the email or not (1 or 0) if the hard disk is faied.
  - **alarm_out:** Trigger the alarm out or not (1 or 0) if the hard disk is faied. The position in the array means the index of alarm. For example, alarm_out[0] means “Alarm 1”.
- **disk_full:**
  - **threshold:** This value us unused.
  - **send_email:** Send the email or not (1 or 0) if the hard disk is full.
  - **alarm_out:** Trigger the alarm out or not (1 or 0) if the hard disk is full. The position in the array means the index of alarm. For example, alarm_out[0] means “Alarm 1”.
- **disk_slow:**
  - **threshold:** This value us unused.
  - **send_email:** Send the email or not (1 or 0) if the hard disk is slow.
  - **alarm_out:** Trigger the alarm out or not (1 or 0) if the hard disk is slow. The position in the array means the index of alarm. For example, alarm_out[0] means “Alarm 1”.

Reply Example:


```
{”code”:0,”msg”:”getsys_evt_configsuccess.”,”name”:”/setup/system_
monitor_config”,”output”:{”system_monitor_setting”:{”cpu_fan”:
{”alarm_out”:[false,false],”send_email”:true,”threshold”:200},”cpu_
temperature”:{”alarm_out”:[true,false],”send_
email”:true,”threshold”:80},”disk_failed”:{”alarm_out”:
[false,false],”send_email”:true,”threshold”:0},”disk_full”:{”alarm_
out”:[false,false],”send_email”:true,”threshold”:0},”disk_slow”:
```

{”alarm_out”:[false,false],“send_email”:true,”threshold”:0}},”system_monitor_setting_list”:[”cpu_fan”,”cpu_temperature”,”disk_failed”,”disk_full”,”disk_slow”]}}

Syntax	http://{server_IP}/setup/system_monitor_config
Method	GET
Parameters	NONE
Example	http://192.168.6.51/setup/system_monitor_config

System Monitor Configuration Set

Purpose: To set the system monitor configuration.

Parameters: JSON format as follows:

{”system_monitor_setting”:{”cpu_fan”:{”alarm_out”:[%d,…],”send_email”:%d,”threshold”:%d},”cpu_temperature”:{”alarm_out”:[%d,…],”send_email”:%d,”threshold”:%d},”disk_failed”:{”alarm_out”:[%d,…],”send_email”:%d,”threshold”:%d},”disk_full”:{”alarm_out”:[%d,…],”send_email”:%d,”threshold”:%d},”disk_slow”:{”alarm_out”:[%d,…],”send_email”:%d,”threshold”:%d},”system_monitor_setting_list”:[”cpu_fan”,”cpu_temperature”,”disk_failed”,”disk_full”,”disk_slow”]}

Reply: JSON format as follows:

{“code”:“ReplyCode”,”name”:“APIName”,“msg”:“ReplyContent”,“output”:{“system_monitor_setting”:{“cpu_fan”:{“alarm_out”:[%d,…],“send_email”:%d,“threshold”:%d},“cpu_temperature”:{“alarm_out”:[%d,…],“send_email”:%d,“threshold”:%d},“disk_failed”:{“alarm_out”:[%d,…],“send_email”:%d,“threshold”:%d},“disk_full”:{“alarm_out”:[%d,…],“send_email”:%d,“threshold”:%d},“disk_slow”:{“alarm_out”:[%d,…],“send_email”:%d,“threshold”:%d}},“system_monitor_list”:[“cpu_fan”,“cpu_temperature”,“disk_failed”,“disk_full”,“disk_slow”]}}

system_monitor_setting:

- **cpu_temperature:**
  - **threshold:** This integer value represent the CPU temperature threshold in Celsius.
  - **send_email:** Send the email or not (1 or 0) if the CPU temperature over the threshold.
  - **alarm_out:** Trigger the alarm out or not (1 or 0) If the CPU temperature over the threshold. The position in the array means the index of alarm. For example, alarm_out [0] means “Alarm 1”.
- **cpu_fan:**
  - **threshold:** This integer value represent the CPU fan speed threshold in RPM.
  - **send_email:** Send the email or not (1 or 0) if the CPU fan speed over the threshold.
  - **alarm_out:** Trigger the alarm out or not (1 or 0) if the CPU fan speed over the threshold. The position in the array means the index of alarm. For example, alarm_out [0] means “Alarm 1”.
- **disk_failed:**
  - **threshold:** This value us unused.
  - **send_email:** Send the email or not (1 or 0) if the hard disk is faied.
  - **alarm_out:** Trigger the alarm out or not (1 or 0) if the hard disk is faied. The position in the array means the index of alarm. For example, alarm_out[0] means “Alarm 1”.
- **disk_full:**
  - **threshold:** This value us unused.
  - **send_email:** Send the email or not (1 or 0) if the hard disk is full.
  - **alarm_out:** Trigger the alarm out or not (1 or 0) if the hard disk is full. The position in the array means the index of alarm. For example, alarm_out[0] means “Alarm 1”.
- **disk_slow:**
  - **threshold:** This value us unused.
  - **send_email:** Send the email or not (1 or 0) if the hard disk is slow.
  - **alarm_out:** Trigger the alarm out or not (1 or 0) if the hard disk is slow. The position in the array means the index of alarm. For example, alarm_out[0] means “Alarm 1”.

Reply Example:


```
{”code”:0,”msg”:”Success.”,”name”:”/setup/system_monitor_
config”,”output”:{”system_monitor_setting”:{”cpu_fan”:{”alarm_out”:
[false,false],”send_email”:true,”threshold”:200},”cpu_temperature”:
{”alarm_out“:[true,false],”send_email”:false,”threshold”:80},”disk_
failed”:{”alarm_out”:[false,false],”send_
email”:true,”threshold”:0},”disk_full”:{”alarm_out”:
```

[false,false],”send_email”:true,”threshold”:0},”disk_slow”:{”alarm_out”:[false,false],”send_email”:true,”threshold”:0}},”system_monitor_setting_list”:[”cpu_fan”,”cpu_temperature”,”disk_failed”,”disk_full”,”disk_slow”]}}

Syntax	http://{server_IP}/setup/system_monitor_config
Method	POST
Parameters	NONE
Example	http://192.168.6.51/setup/system_monitor_config

Event Management Configuration Get

Purpose: To get the event management configuration.

Parameters: None.

Reply: JSON format as follows:

{”code”:”ReplyCode”,“msg”:”ReplyContent”,“name”:”APIName”,“output”:{”camera”:{”ch0”:{”ch”:%d,“event”:{”motion”:{”alarm_out”:[%d,…],”event_trigger”:%d,”send_email”:%d,”snapshot_size”:%s,”take_snapshot”:%d},”vloss”:{”alarm_out”:[%d,…],”event_trigger”:%d,”send_email”:%d,”snapshot_size”:%s,”take_snapshot”:%d}},”ip”:%s},…},”camera_list”:[%s,…],”default”:{”alarm_out”:[%d,…],”event_trigger”:%d,“send_email”:%d,“snapshot_size”:%s,“take_snapshot”:%d},“nvr”:{”nvr0”:{”event”:{”alarm_in_0”:{”alarm_out”:[%d,…],”channel_map”:[%d,…],”event_trigger”:%d,”indx”:%d,”send_email”:%d,”snapshot_size”:%s,”take_snapshot”:%d},…},”nvr_list”:[%s,…],”snapshot_size_list”:[%s,…]}}

camera_list: An array of string represents the channel names that NVR is current connected.

camera: Each element inside this object represent different channel event setting. The number of element is decided by the current NVR device connection, and the name of the element is defined at “camera_list”.

- **ch:** Channel index of the camera.
- **ip:** IP address of the camera.

- **event:** This object contains two elements: “motion” & “vloss”. “motion” represent the motion event and “vloss” represent the video loss event. All elements have the same property structure as follows:
  - **event_trigger:** Enable event trigger or not.(true/false)
  - **alarm_out:** An array of true/false values to represent which alarm out can trigger when the event occurs.
  - **send_email:** Enable the NVR to send email or not when the event occurs. (true/false)
  - **take_snapshot:** Enable the NVR to take snapshot or not when the event occurs. (true/false)
  - **snapshot_size:** Snapshot size.

default: This object define the default camera event setting:

- **event_trigger:** Enable event trigger or not.(true/false)
- **alarm_out:** An array of true/false values to represent which alarm out can trigger when the event occurs.
- **send_email:** Enable the NVR to send email or not when the event occurs. (true/false)
- **take_snapshot:** Enable the NVR to take snapshot or not when the event occurs. (true/false)
- **snapshot_size:** Snapshot size.

nvr_list: An array of string represents the property names of “nvr”.

nvr: Element inside this object represent the NVR event setting. The name of the element is defined at “nvr_list”.

- **indx:** The index of the NVR.
- **event:** Each elements inside this object represent the Alarm-In event of the NVR. All elements have the same property structure as follows:
  - **indx:** The index of Alarm-In event.
  - **event_trigger:** Enable event trigger or not.(true/false)
  - **alarm_out:** An array of true/false values to represent which alarm out can trigger when the event occurs.
  - **channel_map:** An array of true/false values to represent which channel will show ‘alarm in’ icon when the event occurs.
  - **send_email:** Enable the NVR to send email or not when the event occurs. (true/false)
  - **take_snapshot:** Enable the NVR to take snapshot or not when the event occurs. (true/false)
  - **snapshot_size:** Snapshot size.

snap_shot_size_list: An array of string to represent all possible snapshot size.

Reply Example:


```
{"code":0,"msg":"getevt_actionsuccess.","name":"/setup/event_action_
config","output":{"camera":{"ch0":{"ch":0,"event":{"motion":{"alarm_
out":[true,true],"event_trigger":true,"send_email":false,"snapshot_
size":"CIF","take_snapshot":true},"vloss":{"alarm_out":
[false,true],"event_trigger":true,"send_email":false,"snapshot_
size":"CIF","take_snapshot":true}},"ip":"192.168.6.142"}},"camera_
list":["ch0"],"default":{"alarm_out":[false,false],"event_
trigger":false,"send_email":false,"snapshot_size":"CIF","take_
snapshot":false},"nvr":{"nvr0":{"event":{"alarm_in_0":{"alarm_out":
[false,false],"channel_map":
[false,false,false,false,false,false,false,false,false,false,false,f
alse,false,false,false,false,false,false,false,false,false,false,fal
se,false,false,false,false,false,false,false,false,false],"event_
trigger":false,"indx":0,"send_email":false,"snapshot_
size":"CIF","take_snapshot":false},"alarm_in_1":{"alarm_out":
[false,false],"channel_map":
[false,false,false,false,false,false,false,false,false,false,false,f
alse,false,false,false,false,false,false,false,false,false,false,fal
se,false,false,false,false,false,false,false,false,false],"event_
trigger":false,"indx":1,"send_email":false,"snapshot_
size":"CIF","take_snapshot":false},"alarm_in_2":{"alarm_out":
[false,false],"channel_map":
[false,false,false,false,false,false,false,false,false,false,false,f
alse,false,false,false,false,false,false,false,false,false,false,fal
se,false,false,false,false,false,false,false,false,false],"event_
trigger":false,"indx":2,"send_email":false,"snapshot_
size":"CIF","take_snapshot“:false},"alarm_in_3":{"alarm_out":
[false,false],"channel_map":
[false,false,false,false,false,false,false,false,false,false,false,f
alse,false,false,false,false,false,false,false,false,false,false,fal
se,false,false,false,false,false,false,false,false,false],"event_
trigger":false,"indx":3,"send_email":false,"snapshot_
size":"CIF","take_snapshot":false}},"indx":0}},"nvr_list":
["nvr0"],"snapshot_size_list":["CIF","D1","OriginalSize"]}}
```


Syntax	http://{server_IP}/setup/event_action_config
Method	GET

Parameters	NONE
Example	http://192.168.6.51/setup/event_action_config

Event Management Configuration Set

Purpose: To set the event management configuration.

Parameters: JSON format as follows:

```
{"camera":{"ch0":{"ch":%d,"event":{"motion":{"alarm_out":
[%d,…],"event_trigger":%d,"send_email":%d,"snapshot_size":%s,"take_
snapshot":%d},"vloss":{"alarm_out":[%d,…],"event_trigger":%d,"send_
email":%d,"snapshot_size":%s,"take_
snapshot":%d}},"ip":%s},…},"camera_list":[%s,…],"default":{"alarm_
out":[%d,…],"event_trigger":%d,"send_email":%d,"snapshot_
size":%s,"take_snapshot":%d},"nvr":{"nvr0":{"event":{"alarm_in_0":
{"alarm_out":[%d,…],"channel_map":[%d,…],"event_
trigger":%d,"indx":%d,"send_email":%d,"snapshot_size":%s,"take_
snapshot":%d},…},"nvr_list":[%s,…],"snapshot_size_list":[%s,…]}
```

Reply: JSON format as follows:

```
{"code":"ReplyCode","msg":"ReplyContent","name":"APIName","output":
{"camera":{"ch0":{"ch":%d,"event":{"motion":{"alarm_out":
[%d,…],"event_trigger":%d,"send_email":%d,"snapshot_size":%s,"take_
snapshot":%d},"vloss":{"alarm_out":[%d,…],"event_trigger":%d,"send_
email":%d,"snapshot_size":%s,"take_
snapshot":%d}},"ip":%s},…},"camera_list":[%s,…],"default":{"alarm_
out":[%d,…],"event_trigger":%d,"send_email":%d,"snapshot_
size":%s,"take_snapshot":%d},"nvr":{"nvr0":{"event":{"alarm_in_0":
{"alarm_out":[%d,…],"channel_map":[%d,…],"event_
trigger":%d,"indx":%d,"send_email":%d,"snapshot_size":%s,"take_
snapshot":%d},…},"nvr_list":[%s,…],"snapshot_size_list":[%s,…]}}
```

camera_list: An array of string represents the channel names that NVR is current connected.

camera: Each element inside this object represents a different channel event setting. The number of element is decided by the current NVR device connection, and the name of the element is defined at “camera_list”.

- **ch:** Channel index of the camera.
- **ip:** IP address of the camera.
- **event:** This object contains two elements: “motion” & “vloss”. “motion” represent the motion event and “vloss” represent the video loss event. All elements have the same property structure as follows:
  - **event_trigger:** Enable event trigger or not.(true/false)
  - **alarm_out:** An array of true/false values to represent which alarm out can trigger when the event occurs.
  - **send_email:** Enable the NVR to send email or not when the event occurs. (true/false)
  - **take_snapshot:** Enable the NVR to take snapshot or not when the event occurs. (true/-false)
  - **snapshot_size:** Snapshot size.

default: This object define the default camera event setting:

- **event_trigger:** Enable event trigger or not.(true/false)
- **alarm_out:** An array of true/false values to represent which alarm out can trigger when the event occurs.
- **send_email:** Enable the NVR to send email or not when the event occurs. (true/false)
- **take_snapshot:** Enable the NVR to take snapshot or not when the event occurs. (true/false)
- **snapshot_size:** Snapshot size.

nvr_list: An array of string represents the property names of “nvr”.

nvr: Element inside this object represent the NVR event setting. The name of the element is defined at “nvr_list”.

- **indx:** The index of the NVR.
- **event:** Each elements inside this object represent the Alarm-In event of the NVR. All elements have the same property structure as follows:
  - **indx:** The index of Alarm-In event.
  - **event_trigger:** Enable event trigger or not.(true/false)
  - **alarm_out:** An array of true/false values to represent which alarm out can trigger when the event occurs.
  - **channel_map:** An array of true/false values to represent which channel will show ‘alarm in’ icon when the event occurs.
  - **send_email:** Enable the NVR to send email or not when the event occurs. (true/false)
  - **take_snapshot:** Enable the NVR to take snapshot or not when the event occurs. (true/-false)
  - **snapshot_size:** Snapshot size.

snap_shot_size_list: An array of string to represent all possible snapshot size.

Reply Example:


```
{”code”:0,”msg”:”Success.”,”name”:”/setup/event_action_
config”,”output”:{”camera”:{”ch0”:{”ch”:0,”event”:{”motion”:
{”alarm_out”:[true,true],”event_trigger”:true,”send_
email”:true,”snapshot_size”:”CIF”,”take_snapshot”:false},”vloss”:
{”alarm_out”:[false,true],”event_trigger”:true,”send_
email”:false,”snapshot_size“:”CIF”,”take_
snapshot”:true}},”ip”:”192.168.6.142”}},”camera_list”:
[”ch0”],”default”:{”alarm_out”:[false,false],”event_
trigger”:false,”send_email”:false,”snapshot_size”:”CIF”,”take_
snapshot”:false},”nvr”:{”nvr0”:{“event”:{”alarm_in_0”:{”alarm_out”:
[false,false],”channel_map”:
[false,false,false,false,false,false,false,false,false,false,false,
false,false,false,false,false,false,false,false,false,false,false,f
alse,false,false,false,false,false,false,false,false,false],”event_
trigger”:true,”indx”:0,”send_email”:false,”snapshot_
size”:”CIF”,”take_snapshot”:false},”alarm_in_1”:{”alarm_out”:
[false,false],”channel_map”:
[false,false,false,false,false,false,false,false,false,false,false,
false,false,false,false,false,false,false,false,false,false,false,f
alse,false,false,false,false,false,false,false,false,false],”event_
trigger”:false,”indx”:1,”send_email”:false,”snapshot_
size”:”CIF”,”take_snapshot”:false},”alarm_in_2”:{”alarm_out”:
[false,false],”channel_map”:
[false,false,false,false,false,false,false,false,false,false,false,
false,false,false,false,false,false,false,false,false,false,false,f
alse,false,false,false,false,false,false,false,false,false],”event_
trigger”:false,”indx”:2,”send_email”:false,”snapshot_
size”:”CIF”,”take_snapshot”:false},”alarm_in_3”:{”alarm_out”:
[false,false],”channel_map”:
[false,false,false,false,false,false,false,false,false,false,false,
false,false,false,false,false,false,false,false,false,false,false,f
alse,false,false,false,false,false,false,false,false,false],”event_
trigger”:false,”indx”:3,”send_email”:false,”snapshot_
size”:”CIF”,”take_snapshot”:false}},”indx”:0}},”nvr_list”:
[”nvr0”],”snapshot_size_list”:[”CIF”,”D1”,”OriginalSize”]}}
```

Syntax	http://{server_IP}/setup/event_action_config
Method	POST
Parameters	{"camera":{"ch0":{"ch":%d,"event":{"motion":{"alarm_out":[%d, …],"event_trig-ger":%d,"send_email":%d,"snapshot_size":%s,"take_snapshot":%d},"vloss":{"alarm_out":[%d, …],"event_trigger":%d,"send_e mail":%d,"snapshot_size":%s,"take_snap-shot":%d}},"ip":%s}, …},"camera_list":[%s, …],"d efault":{"alarm_out":[%d, …],"event_trigger":%d, "send_email":%d, "snapshot_size":%s, "take_snap-shot":%d},"nvr":{"nvr0":{"event":{"alarm_in_0":{"alarm_out":[%d, …],"channel_map":[%d, …],"eve nt_trigger":%d,"indx":%d,"send_email":%d,"snapshot_size":%s,"take_snapshot":%d}, …},"nvr_list":[%s, …],"snapshot_size_list":[%s, …]}
Example	http://192.168.6.51/setup/event_action_config

Email Setting Get

Purpose: To get the email setting.

Parameters: None.

Reply: JSON format as follows:

```
{“code”:“ReplyCode”,”name”:“APIName”,“msg”:“ReplyContent”,“output”:
{"account":%s,"address":[%s,%s,%s],"current_
provider":%s,"password":%s,"provider_list":[%s,%s,%s],"provider_
opt":{"custom":{"enabled_ssl":%d,"port":%d,"server_site":%s,"show_
name":%s},"gmail":{"enabled_ssl":%d,"port":%d,"server_
site":%s,"show_name":%s},"yahoo_mail":{"enabled_
ssl":%d,"port":%d,"server_site":%s,"show_name":%s}}}}
```

account: Account of the SMTP server.

password: Password of the SMTP server.

address: Email address of the recipient.

provider_list: An array of strings to represent all supported SMTP provider.

current_provider: Current selected SMTP provider.

provider_opt: Elements inside this object represent the different SMTP provider setting. All of them have the same property structure as follows:

- **enabled_ssl:** Enable SSL or not. (true/false).
- **port:** Port of the SMTP server.
- **server_site:** Server site of the SMTP server.
- **show_name:** GUI display name of this SMTP provider.

Reply Example:


```
{"code":0,"msg":"getemail_configsuccess.","name":"/setup/email_
config","output":{"account":"","address":["","",""],"current_
provider":"yahoo_mail","password":"","provider_list":
["gmail","yahoo_mail","custom"],"provider_opt":{"custom":{"enabled_
ssl":true,"port":465,"server_site":"smtp.mail.yahoo.com","show_
name":"Custom"},"gmail":{"enabled_ssl":true,"port":465,"server_
site":"smtp.gmail.com","show_name":"Gmail"},"yahoo_mail":{"enabled_
ssl":true,"port":465,"server_site":"smtp.mail.yahoo.com","show_
name":"YahooMail"}}}}
```


Syntax	http://{server_IP}/setup/email_config
Method	GET
Parameters	NONE
Example	http://192.168.6.51/setup/email_config

Email Setting Set

Purpose: To set the email setting.

Parameters: JSON format as follows:


```
{"account":%s,"address":[%s,%s,%s],"current_
provider":%s,"password":%s,"provider_list":[%s,%s,%s],"provider_
opt":{"custom":{"enabled_ssl":%d,"port":%d,"server_site":%s,"show_
name":%s},"gmail":{"enabledssl":%d,"port":%d,"server_
site":%s,"show_name":%s},"yahoo_mail":{"enabled_
ssl":%d,"port":%d,"server_site":%s,"show_name":%s}}}
```

Reply: JSON format as follows:

```
{“code”:“ReplyCode”,”name”:“APIName”,“msg”:“ReplyContent”,“output”:
{"account":%s,"address":[%s,%s,%s],"current_
provider":%s,"password":%s,"provider_list":[%s,%s,%s],"provider_
opt":{"custom":{"enabled_ssl":%d,"port":%d,"server_site":%s,"show_
name":%s},"gmail":{"enabled_ssl":%d,"port":%d,"server_
site":%s,"show_name":%s},"yahoo_mail":{"enabled_
ssl":%d,"port":%d,"server_site":%s,"show_name":%s}}}}
```

account: Account of the SMTP server. Password: Password of the SMTP server. Address: Email address of the recopoent.

provider_list: An array of strings to represent all supported SMTP provider.

current_provider: Current selected SMTP provider.

provider_opt: Elements inside this object represent the different SMTP provider setting. All of them have the same property structure as follows:

- **enabled_ssl:** Enable SSL or not. (true/false)
- **port:** Port of the SMTP server.
- **server_site:** Server site of the SMTP server.
- **show_name:** GUI display name of this SMTP provider.

Reply Example:

```
{”code”:0,”msg”:”Success.”,”name”:”/setup/email_config”,”output”:
{”account”:””,”address”:[””,””,””],”current_provider”:”yahoo_
mail”,”password”:”GtBcs92K”,”provider_list”:[”gmail”,”yahoo_
mail”,”custom”],”provider_opt”:{”custom”:{”enabled_
ssl”:true,”port”:465,”server_site”:”smtp.gmail.com”,”show_
name”:”Custom”},”gmail”:{”enabled_ssl”:true,”port”:465,”server_
site”:”smtp.gmail.com”,”show_name”:”Gmail”},”yahoo_mail”:{”enabled_
ssl”:true,”port”:465,”server_site”:”smtp.mail.yahoo.com”,”show_
name”:”YahooMail”}}}}
```

Syntax	http://{server_IP}/setup/email_config
Method	POST

Parameters	{ "account":%s,"address":[%s, %s,%s],"current_provider":%s,"password":%s,"provider_list":[%s,%s,%s],"provider_opt":{"c ustom":{"enabled_ssl":%d,"port":%d,"server_site":%s,"show_name":%s},"gmail":{"ena bled_ssl":%d,"port":%d,"server_site":%s,"show_name":%s},"yahoo_mail":{"enabled_ssl ":%d,"port":%d,"server_site":%s,"show_name":%s}}}
Example	http://192.168.6.51/setup/email_config

DDNS Setting Get

Purpose: To get the DDNS setting.

Parameters: None.

Reply: JSON format as follows:

```
{“code”:“ReplyCode”,”name”:“APIName”,“msg”:“ReplyContent”,“output”:
{"enabled":%d,"hostname":%s,"password":%s,"port":%d,"refresh_
status":%s,"server_list":[%s,…],"server_name":%s,"username":%s}}
```

enabled: Enable DDNS or not. (true/false)

hostname: Host name of the DDNS.

username: User name of the DDNS.

password: Encrypted password of the DDNS. (To decrypt the password, check out the appendix.)

port: Port of the DDNS.

refresh_status: DDNS setup success or not. (“Success.”/“Failed.”).

server_list: An array of string to represent all supported DDNS server.

server_name: Current selected server name.

Reply Example:

```
{”code”:0,”msg”:”getddns_configsuccess.”,”name”:”/setup/ddns_
config”,”output”:
{”enabled”:false,”hostname”:””,”password”:””,”port”:80,“refresh_
status”:”Success.”,”server_list”:
[”DynDNS”,”ChangeIP”,”NoIP”],”server_name”:”DynDNS”,”username”:””}}
```

Syntax	http://{server_IP}/setup/ddns_config
Method	GET
Parameters	NONE
Example	http://192.168.6.51/setup/ddns_config

DDNS Setting Set

Purpose: To set the DDNS setting.

Parameters: JSON format as follows:

```
{”enabled”:%d,”hostname”:%s,”password”:%s,”port”:%d,”refresh_
status”:%s,”server_list”:[%s,…],”server_name”:%s,”username”:%s}
```

Reply: JSON format as follows:

```
{“code”:“ReplyCode”,”name”:“APIName”,“msg”:“ReplyContent”,“output”:
{"enabled":%d,"hostname":%s,"password":%s,"port":%d,"refresh_
status":%s,"server_list":[%s,…],"server_name":%s,"username":%s}}
```

enabled: Enable DDNS or not. (true/false)

hostname: Host name of the DDNS.

username: User name of the DDNS.

password: Encrypted password of the DDNS. (To decrypt the password, check out the appendix.)

port: Port of the DDNS.

refresh_status: DDNS setup success or not. (“Success.”/“Failed.”).

server_list: An array of string to represent all supported DDNS server.

server_name: Current selected server name.

Reply Example:

{”code”:0,”msg”:”Success.”,”name”:”/setup/ddns_config”,”output”:{”en-
abled”:true,”hostname”:””,”passw ord”:”GtBc9p/Z”,”port”:80,”refresh_status”:”Failed.”,”server_list”:
[”DynDNS”,”ChangeIP”,”NoIP”],”server_

name”:”DynDNS”,”username”:””}}

Syntax	http://{server_IP}/setup/ddns_config
Method	POST
Parameters	{"enabled":%d,"hostname":%s,"password":%s,"port":%d,"refresh_status":%s,"server_list":[%s,…],"server_name":%s,"username":%s}
Example	http://192.168.6.51/setup/ddns_config

Account Add

Purpose: To add a new account.

Parameters: JSON format as follows:

```
{”account”:”%s,”password”:%s}
```

account: New user account.

password: New user encrypted password.

NOTE! Please check out the appendix to see how to encrypt/decrypt the password.

Reply: JSON format as follows:

```
{“code”:“ReplyCode”,”name”:“APIName”,“msg”:“ReplyContent”,“output”:
{“user”:[{“account”:%s,“password”:%s,“authority”:%d,“covert”:
[true/false,…]},…]}}
```

user:

- **account:** User account
- **password:** User encrypted password
- **authority:** The decimal value conversion from the 5-bits binary value which is mapping to the five authorities (Setup, Live(always 1), PTZ, Device, Playback) is permitted or not.
- **covert:** [true/false, …, true/false], (covert/ not covert) the channels from the account

Reply Example:

```
{”code”:0,”msg”:”/setup/account_
addSuccess!”,”name”:”/setup/account_add”,”output”:{”user”:
[{”account”:”Guest”,”authority”:8,”covert”:
```

[false,false,false,false,false,false,false,false,false,false,false,f
alse,false,false,false,false,false,false,false,false,false,false,false,fal
se,false,false,false,false,false,false,false,false,false,false,false
,false,false,false,false,false,false,false,false,false,false,false,f
alse,false,false,false,false,false,false,false,false,false,false,false,fal
se,false,false,false,false,false,false,false,false,false,false,false
,false,false,false,false,false,false,false,false,false,false,false,f
alse,false,false,false,false,false,false,false,false,false,false,false,fal
se,false,false,false,false,false,false,false,false,false,false,false
,false,false,false,false,false,false,false,false,false,false,false,f
alse,false,false,false,false,false,false,false,false,false,false,false,fal
se,false,false,false],"password":""},
{"account":"Admin","authority":31,"covert":
[false,false,false,false,false,false,false,false,false,false,false,f
alse,false,false,false,false,false,false,false,false,false,false,false,fal
se,false,false,false,false,false,false,false,false,false,false,false
,false,false,false,false,false,false,false,false,false,false,false,f
alse,false,false,false,false,false,false,false,false,false,false,false,fal
se,false,false,false,false,false,false,false,false,false,false,false
,false,false,false,false,false,false,false,false,false,false,false,f
alse,false,false,false,false,false,false,false,false,false,false,false,fal
se,false,false,false,false,false,false,false,false,false,false,false
,false,false,false,false,false,false,false,false,false,false,false,f
alse,false,false,false,false,false,false,false,false,false,false,false,fal
se,false,false,false],"password":"GtBc9p/Z"},
{"account":"Tony","authority":8,"covert":
[false,false,false,false,false,false,false,false,false,false,false,f
alse,false,false,false,false,false,false,false,false,false,false,false,fal
se,false,false,false,false,false,false,false,false,false,false,false
,false,false,false,false,false,false,false,false,false,false,false,f
alse,false,false,false,false,false,false,false,false,false,false,false,fal
se,false,false,false,false,false,false,false,false,false,false,false
,false,false,false,false,false,false,false,false,false,false,false,f
alse,false,false,false,false,false,false,false,false,false,false,false,fal
se,false,false,false,false,false,false,false,false,false,false,false
,false,false,false,false,false,false,false,false,false,false,false,f
alse,false,false,false,false,false,false,false,false,false,false,false,fal
se,false,false,false],"password":"GtBc9p/Z"}]}}

Syntax	http://{server_IP}/setup/account_add
Method	POST
Parameters	{"account":"%s,"password":%s}
Example	http://192.168.6.51/setup/account_add

Account Delete

Purpose: To delete an account.

Parameters: JSON format as follows:

```
{”account”:”%s,”password”:%s}
```

account: User account to delete.

password: User encrypted password to delete.

NOTE! Please check out the appendix to see how to encrypt/decrypt the password.

Reply: JSON format as follows:

```
{“code”:“ReplyCode”,”name”:“APIName”,“msg”:“ReplyContent”,“output”:
{“user”:[{“account”:%s,“password”:%s,“authority”:%d,“covert”:
[true/false,…]},…]}}
```

user:

- **account:** User account
- **password:** User encrypted password
- **authority:** The decimal value conversion from the 5-bits binary value which is mapping to the five authorities (Setup, Live(always 1), PTZ, Device, Playback) is permitted or not.

covert: [true/false, …, true/false], (covert/ not covert) the channels from the account

Reply Example:

```
{”code”:0,”msg”:”/setup/account_
deleteSuccess!”,”name”:”/setup/account_delete”,”output”:{”user”:
[{”account”:”Guest”,”authority”:8,”covert”:
```

```
[false,false,false,false,false,false,false,false,false,false,false,f
alse,false,false,false,false,false,false,false,false,false,false,fal
se,false,false,false,false,false,false,false,false,false,false,false
,false,false,false,false,false,false,false,false,false,false,false,f
alse,false,false,false,false,false,false,false,false,false,false,fal
se,false,false,false,false,false,false,false,false,false,false,false
,false,false,false,false,false,false,false,false,false,false,false,f
alse,false,false,false,false,false,false,false,false,false,false,fal
se,false,false,false,false,false,false,false,false,false,false,false
,false,false,false,false,false,false,false,false,false,false,false,f
alse,false,false,false,false,false,false,false,false,false,false,fal
se,false,false,false],”password”:””},
{”account”:”Admin”,”authority”:31,”covert”:
[false,false,false,false,false,false,false,false,false,false,false,f
alse,false,false,false,false,false,false,false,false,fals,false,fals
e,false,false,false,false,false,false,false,false,false,false,false,
false,false,false,false,false,false,false,false,false,false,false,fa
lse,false,false,false,false,false,false,false,false,false,false,fals
e,false,false,false,false,false,false,false,false,false,false,false,
false,false,false,false,false,false,false,false,false,false,false,fa
lse,false,false,false,false,false,false,false,false,false,false,fals
e,false,false,false,false,false,false,false,false,false,false,false,
false,false,false,false,false,false,false,false,false,false,false,fa
lse,false,false,false,false,false,false,false,false,false,false,fals
e,false,false,false],”password”:”GtBc9p/Z”}]}}
```

Syntax	http://{server_IP}/setup/account_delete
Method	POST
Parameters	{"account":%s,"password":%s}
Example	http://192.168.6.51/setup/account_delete

Account Edit

Purpose: To edit the name or password of the account.

Parameters: JSON format as follows:

```
{”account”:%s,”password”:%s,”new_account”:%s,”new_password”:%s}
```

account: Original user account.

assword: Original user encrypted password.

new_account: New user account.

new_password: New user encrypted password.

NOTE! Please check out the appendix to see how to encrypt/decrypt the password.

Reply: JSON format as follows:

```
{“code”:“ReplyCode”,”name”:“APIName”,“msg”:“ReplyContent”,“output”:
“output”:{“user”:
[{“account”:%s,“password”:%s,“authority”:%d,“covert”:
[true/false,…]},…]}}
```

user:

- **account:** User account
- **password:** User encrypted password
- **authority:** The decimal value conversion from the 5-bits binary value which is mapping to the five authorities (Setup, Live(always 1), PTZ, Device, Playback) is permitted or not.

covert: [true/false, …, true/false], (covert/ not covert) the channels from the account.

Reply Example:

```
{”code”:0,”msg”:”/setup/account_
editSuccess!”,”name”:”/setup/account_edit”,”output”:{”user”:
[{”account”:”Guest”,”authority”:8,”covert”:
[false,false,false,false,false,false,false,false,false,false,false,
false,false,false,false,false,false,false,false,false,false,false,f
alse,false,false,false,false,false,false,false,false,false,false,fa
lse,false,false,false,false,false,false,false,false,false,false,fal
se,false,false,false,false,false,false,false,false,false,false,fals
e,false,false,false,false,false,false,false,false,false,false,false
,false,false,false,false,false,false,false,false,false,false,false,
false,false,false,false,false,false,false,false,false,false,false,f
alse,false,false,false,false,false,false,false,false,false,false,fa
lse,false,false,false,false,false,false,false,false,false,false,fal
se,false,false,false,false,false,false,false,false,false,false,fals
e,false,false,false,false,false],”password”:””},
{”account”:”Admin”,”authority”:31,”covert”:
```

[false,false,false,false,false,false,false,false,false,false,false,f
alse,false,false,false,false,false,false,false,false,false,false,fal
se,false,false,false,false,false,false,false,false,false,false,false
,false,false,false,false,false,false,false,false,false,false,false,f
alse,false,false,false,false,false,false,false,false,false,false,fal
se,false,false,false,false,false,false,false,false,false,false,false
,false,false,false,false,false,false,false,false,false,false,false,f
alse,false,false,false,false,false,false,false,false,false,false,fal
se,false,false,false,false,false,false,false,false,false,false,false
,false,false,false,false,false,false,false,false,false,false,false,f
alse,false,false,false,false,false,false,false,false,false,false,fal
se,false,false,false],"password":"GtBc9p/Z"},
{"account":"TonyR","authority":15,"covert":
[false,false,false,false,false,false,false,false,false,false,false,f
alse,false,false,false,false,false,false,false,false,false,false,fal
se,false,false,false,false,false,false,false,false,false,false,false
,false,false,false,false,false,false,false,false,false,false,false,f
alse,false,false,false,false,false,false,false,false,false,false,fal
se,false,false,false,false,false,false,false,false,false,false,false
,false,false,false,false,false,false,false,false,false,false,false,f
alse,false,false,false,false,false,false,false,false,false,false,fal
se,false,false,false,false,false,false,false,false,false,false,false
,false,false,false,false,false,false,false,false,false,false,false,f
alse,false,false,false,false,false,false,false,false,false,false,fal
se,false,false,false],"password":"GtBc9p/Z"}]}}

Syntax	http://{server_IP}/setup/account_edit
Method	POST
Parameters	{"account":%s,"password":%s,"new_account":%s,"new_password":%s}
Example	http://192.168.6.51/setup/account_edit

Account Authority Change

Purpose: To change the authority of the account.

Parameters: JSON format as follows:

```
{“user”:[{“account”:%s,“password”:%s,“authority”:%d,“covert”:
[true/false,…]},…]}
```

Reply: JSON format as follows:

```
{“code”:“ReplyCode”,”name”:“APIName”,“msg”:“ReplyContent”,“output”:
{“user”:[{“account”:%s,“password”:%s,“authority”:%d,“covert”:
[true/false,…]},…]}}
```

user:

- **account:** User account
- **password:** User encrypted password
- **authority:** The decimal value conversion from the 5-bits binary value which is mapping to the five authorities (Setup, Live(always 1), PTZ, Device, Playback) is permitted or not.

covert: [true/false, …, true/false], (covert/ not covert) the channels from the account.

NOTE! Please check out the appendix to see how to encrypt/decrypt the password.

Reply Example:

```
{”code”:0,”msg”:”/setup/account_auth_
changeSuccess!”,”name”:”/setup/account_auth_change”,”output”:
{”user”:[{”account”:”Guest”,”authority”:8,”covert”:
[false,false,false,false,false,false,false,false,false,false,false,
false,false,false,false,false,false,false,false,false,false,false,f
alse,false,false,false,false,false,false,false,false,false,false,fa
lse,false,false,false,false,false,false,false,false,false,false,fal
se,false,false,false,false,false,false,false,false,false,false,fals
e,false,false,false,false,false,false,false,false,false,false,false
,false,false,false,false,false,false,false,false,false,false,false,
false,false,false,false,false,false,false,false,false,false,false,f
alse,false,false,false,false,false,false,false,false,false,false,fa
lse,false,false,false,false,false,false,false,false,false,false,fal
se,false,false,false,false,false,false,false,false,false,false,fals
e,false,false,false,false,false],”password”:””},
{”account”:”Admin”,”authority”:31,”covert”:
[false,false,false,false,false,false,false,false,false,false,false,
false,false,false,false,false,false,false,false,false,false,false,f
```

```
alse,false,false,false,false,false,false,false,false,false,false,fal
se,false,false,false,false,false,false,false,false,false,false,false
,false,false,false,false,false,false,false,false,false,false,false,f
alse,false,false,false,false,false,false,false,false,false,false,fal
se,false,false,false,false,false,false,false,false,false,false,false
,false,false,false,false,false,false,false,false,false,false,false,f
alse,false,false,false,false,false,false,false,false,false,false,fal
se,false,false,false,false,false,false,false,false,false,false,false
,false,false,false,false,false,false,false,false,false,false,false,f
alse,false,false,false],”password”:”GtBc9p/Z”},
{”account”:”TonyR”,”authority”:27,“covert”:
[false,false,false,false,false,false,false,false,false,false,false,f
alse,false,false,false,false,false,false,false,false,false,false,fal
se,false,false,false,false,false,false,false,false,false,false,false
,false,false,false,false,false,false,false,false,false,false,false,f
alse,false,false,false,false,false,false,false,false,false,false,fal
se,false,false,false,false,false,false,false,false,false,false,false
,false,false,false,false,false,false,false,false,false,false,false,f
alse,false,false,false,false,false,false,false,false,false,false,fal
se,false,false,false,false,false,false,false,false,false,false,false
,false,false,false,false,false,false,false,false,false,false,false,f
alse,false,false,false,false,false,false,false,false,false,false,fal
se,false,false,false],”password”:”GtBc9p/Z”}]}}
```


Syntax	http://{server_IP}/setup/account_auth_change
Method	POST
Parameters	{“user”: [{“account”: %s, “password”: %s, “authority”: %d,“covert”:[true/false, …]}, …] }
Example	http://192.168.6.51/setup/account_auth_change

6

Record relative API

Template Configuration Get

Purpose: To get the template configuration.

Parameters: None.

Reply: JSON format as follows:

```
{“code”:“ReplyCode”,”name”:“APIName”,“msg”:“ReplyContent”,“outpu
t”:{"channel_setting":{"ch0":{"edge":%d,"group_id":%d,"template_
name":%s,"chidx":%d,"mac":%s,"ip":%s,"title":%s,"type":%d}},"rec_
type_opts":{"0":"record","1":"no_record","2":"evt_only"},"record_
template":{"Default":{"Fri":[%d,..],"Mon":[%d,..],"Sat":
[%d,..],"Sun":[%d,..],"Thu":[%d,..],"Tue":[%d,..],"Wed":
[%d,..]},…},”group_num_max”:%d,”group_rec_list”:[%d,…]}}
```

rec_type_opts: Define the value of the record type.

record_template: Each element inside this object represents different record template setting. The name of the element is equal to the name of the template. All template elements have the same property structure: seven 24-length array which represent the 24 hours record setting of a day. (“Sun”: Sunday, “Mon”: Monday, …, “Sat”: Saturday). The value inside the array is defined at “rec_type_opts”.

channel_setting: Each element inside this object represents different channel record template setting. All elements have the same property structure as follows:

- **ch_idx:** Channel index.
- **mac:** MAC address of the channel.
- **ip:** IP address of the channel.
- **title:** Title of the channel.
- **type:** Device type of the channel.
- **group_id:** Group index of the channel.
- **template_name:** Record template of the channel.

group_num_max: Max disk group number.

group_rec_list: Current activated disk group index in an array.

Reply Example:


```
{”code”:0,”msg”:”record_
templategetsuccess”,”name”:”/record/template”,”output”:{”channel_
setting”:{”ch0”:{”edge”:0,”group_id”:0,”template_
name”:”Default”,”ch_
idx”:0,”mac”:””,”ip”:”192.168.6.142”,”title”:”NewDevice”,”type”:1}},
”rec_type_opts”:{”0”:”record”,”1”:”no_record”,”2”:”evt_
only”},”record_template”:{”Default”:{”Fri”:
[0,0,0,0,0,0,0,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],”Mon”:
[0,0,0,0,0,0,0,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],”Sat”:
[0,0,0,0,0,0,0,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],”Sun”:
[0,0,0,0,0,0,0,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],”Thu”:
[0,0,0,0,0,0,0,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],”Tue”:
[0,0,0,0,0,0,0,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],”Wed”:
[0,0,0,0,0,0,0,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]},”Template”:
{”Fri”:[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],”Mon”:
[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],”Sat”:
[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],”Sun”:
[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],”Thu”:
[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],”Tue”:
```

```
[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],"Wed":
[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]}},"group_num_
max":8,"group_rec_list":[0]}}
```

Syntax	http://{server_IP}/record/template
Method	GET
Parameters	NONE
Example	http://192.168.6.51/record/template

Template Configuration Set

Purpose: To set the template configuration.

Parameters: JSON format as follows:

If you want to update the template of channels, send:

```
{"channel_setting":{"ch%d":{"template_name":%s}}}
```

ch%d: Specify which channel

template_name: New template of the channel.

If you want to add new template, send:

```
{"add_template":[%s,…]}
```

add_template: Specify the new template names in an array of string.

If you want to delete template, send:

```
{"remove_template":[%s,…]}
```

remove_template: Specify the template names you want to remove in an array of string.

If you want to update the setting of the template, send:

```
{"record_template":{"TemplateName":{"Sun":[%d…],"Mon":[%d…],"Tue":
[%d…],"Wed":[%d…],"Thu":[%d…],"Fri":[%d…],"Sat":[%d…]}}}
```

record_template: The element name of this object is the template name you want to edit. And the properties of the elements are seven length-24 arrays of index which represent the record setting of per hour of a day (“Sun”: Sunday, “Mon”: Monday, …, etc). The definition of the integer in the array can be found from the reply message of “Template Configuration Get”.

Reply: JSON format as follows:

```
{“code”:“ReplyCode”,”name”:“APIName”,“msg”:“ReplyContent”,“output”:
{"channel_setting":{"ch0":{"edge":%d,"group_id":%d,"template_
name":%s,"ch_idx":%d,"mac":%s,"ip":%s,"title":%s,"type":%d}},"rec_
type_opts":{"0":"record","1":"no_record","2":"evt_only"},"record_
template":{"Default":{"Fri":[%d,..],"Mon":[%d,..],"Sat":
[%d,..],"Sun":[%d,..],"Thu":[%d,..],"Tue":[%d,..],"Wed":
[%d,..]},…},”group_num_max”:%d,”group_rec_list”:[%d,…]}}
```

rec_type_opts: Define the value of the record type.

record_template: Each element inside this object represents different record template setting. The name of the element is equal to the name of the template. All template elements have the same property structure: seven 24-length array which represent the 24 hours record setting of a day (“Sun”: Sunday, “Mon”: Monday, …, “Sat”: Saturday). The value inside the array is defined at “rec_type_opts”.

channel_setting: Each element inside this object represents different channel record template setting. All elements have the same property structure as follows:

- **ch_idx:** Channel index.
- **mac:** MAC address of the channel.
- **ip:** IP address of the channel.
- **title:** Title of the channel.
- **type:** Device type of the channel.
- **group_id:** Group index of the channel.
- **template_name:** Record template of the channel.

group_num_max: Max disk group number.

group_rec_list: Current activated disk group index in an array.

Reply Example:

```
{”code”:0,”msg”:”Success.”,”name”:”/record/template”,”output”:
{”channel_setting”:{”ch0”:{”edge”:0,”group_id”:0,”template_
name”:”Template”},”ch1”:{”edge”:0,”group_id”:0,”template_
name”:””},”ch10”:{”edge“:0,”group_id”:0,”template_name”:””},”ch11”:
{”edge”:0,”group_id”:0,”template_name”:””},”ch12”:{”edge“:0,”group_
id”:0,”template_name”:””},”ch13”:{”edge”:0,”group_id”:0,”template_
name”:””},”ch14”:{”edge“:0,”group_id”:0,”template_name”:””},”ch15”:
{”edge”:0,”group_id”:0,”template_name”:””},”ch16”:{”edge“:0,”group_
```

id":0,"template_name":""},"ch17":{"edge":0,"group_id":0,"template_
name":""},"ch18":{"edge“:0,"group_id":0,"template_name":""},"ch19":
{"edge":0,"group_id":0,"template_name":""},"ch2":{"edge":0,"group_
id":0,"template_name":""},"ch20":{"edge":0,"group_id":0,"template_
name":""},"ch21":{"edge":0,"group_id":0,"template_name":""},"ch22":
{"edge":0,"group_id":0,"template_name":""},"ch23":{"edge":0,"group_
id":0,"template_name":""},"ch24":{"edge":0,"group_id":0,"template_
name":""},"ch25":{"edge":0,"group_id":0,"template_name":""},"ch26":
{"edge":0,"group_id":0,"template_name":""},"ch27":{"edge":0,"group_
id":0,"template_name":""},"ch28":{"edge":0,"group_id":0,"template_
name":""},"ch29":{"edge":0,"group_id":0,"template_name":""},"ch3":
{"edge":0,"group_id":0,"template_name":""},"ch30":{"edge":0,"group_
id":0,"template_name":""},"ch31":{"edge":0,"group_id":0,"template_
name":""},"ch4":{"edge":0,"group_id":0,"template_name":""},"ch5":
{"edge":0,"group_id":0,"template_name":""},"ch6":{"edge":0,"group_
id":0,"template_name":""},"ch7":{"edge":0,"group_id":0,"template_
name":""},"ch8":{"edge":0,"group_id":0,"template_name":""},"ch9":
{"edge":0,"group_id":0,"template_name":""}},"rec_type_opts":
{"0":"record","1":"no_record","2":"evt_only"},"record_template":
{"Default":{"Fri":
[0,0,0,0,0,0,0,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],Mon:
[0,0,0,0,0,0,0,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],Sat:
[0,0,0,0,0,0,0,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],Sun:
[0,0,0,0,0,0,0,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],Thu:
[0,0,0,0,0,0,0,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],Tue:
[0,0,0,0,0,0,0,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],Wed:
[0,0,0,0,0,0,0,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]},Template:{Fri:
[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],Mon:
[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],"Sat":
[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],"Sun":
[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],"Thu":
[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],"Tue":
[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],"Wed":
[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]},"Template(2)":
{"Fri":[2,0,0,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],"Mon":
[2,0,0,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],"Sat":
[2,0,0,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],"Sun":
[2,0,0,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],"Thu":
[2,0,0,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],"Tue":

```
[2,0,0,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],"Wed":
[2,0,0,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]}}}}
```

Syntax	http://{server_IP}/record/record_template
Method	POST

Parameters	**If you want to update the template of channels, send:**
	`{"channel_setting":{"ch%d":{"template_name":%s}}}`
	If you want to add new template, send:
	`{"add_template":[%s,…]}`
	If you want to delete template, send:
	`{"remove_template":[%s,…]}`
	If you want to update the setting of the template, send:
	`{"record_template":{"TemplateName":{"Sun":[%d…],"Mon":[%d…],"Tue":[%d…],"Wed":[%d…],"Thu":[%d…],"Fri":[%d…],"Sat":[%d…]}}}`

Example	http://192.168.6.51/record/template

Record Configuration Get

Purpose: To get the record configuration.

Parameters: None.

Reply: JSON format as follows:

```
{“code”:“ReplyCode”,”name”:“APIName”,“msg”:“ReplyContent”,“output”:
{”data_life_time”:{”cur”:%d,”max”:%d,”min”:%d},”edge_archiver_
interval”:%d,”record_circular”:{”enable”:%d,”percentage”:%d,”size_
mb”:%d,”type”:%s,”type_opt”:[”percentage”,”size”]},”record_post_
alarm”:{”cur”:%d,”enable”:%d,”max”:%d,”min”:%d},”record_pre_alarm”:
{”cur”:%s,”enable”:%d,”max”:%d,“min”:%d}}}
```

edge_archiver_interval: Time interval to check edge archive in seconds.

data_life_time: Data life time of circular recording.

- **cur:** Current value. (Unit: day).
- **max:** Max possible value. (Unit: day)
- **min:** Minimum possible value. (Unit: day)

record_circular:

- **enable:** Enable circular recording or not. (true/false)

record_post_alarm: Post alarm setting.

- **enable:** Enable or not. (true/false)
- **cur:** Current value. (Unit: second)
- **max:** Max possible value. (Unit: second)
- **min:** Minimum possible value. (Unit: second)

record_pre_alarm: Pre alarm setting.

- **enable:** Enable or not. (true/false)
- **cur:** Current value. (Unit: second)
- **max:** Max possible value. (Unit: second)
- **min:** Minimum possible value. (Unit: second)

Reply Example:


```
{”code”:0,”msg”:”record_
configgetsuccess”,”name”:”/record/config”,”output”:{”data_life_
time”:{”cur”:60,”max”:60,”min”:0},”edge_archiver_
interval”:15,”record_circular”:{”enable”:true},”record_post_alarm”:
{”cur”:15,”enable”:true,”max”:120,”min”:15},”record_pre_alarm”:
{”cur”:0,”enable”:true,”max”:120,”min”:0}}}
```


Syntax	http://{server_IP}/record/config
Method	GET
Parameters	NONE
Example	http://192.168.6.51/record/config

Record Configuration Set

Purpose: To set the record configuration.

Parameters: JSON format as follows:

```
{”data_life_time”:{”cur”:%d,”max”:%d,”min”:%d},”edge_archiver_
interval”:%d,”record_circular”:{”enable”:%d},”record_post_alarm”:
{”cur”:%d,”enable”:%d,”max”:%d,”min”:%d},”record_pre_alarm”:
{”cur”:%s,”enable”:%d,”max”:%d,”min”:%d}}
```

Reply: JSON format as follows:

```
{“code”:“ReplyCode”,”name”:“APIName”,“msg”:“ReplyContent”,“output”:
{data_life_time”:{”cur”:%d,”max”:%d,”min”:%d},”edge_archiver_
interval”:%d,”record_circular”:{”enable”:%d},”record_post_alarm”:
{”cur”:%d,”enable”:%d,”max”:%d,”min”:%d},”record_pre_alarm”:
{”cur”:%s,”enable”:%d,”max”:%d,”min”:%d}}}
```

edge_archiver_interval: Time interval to check edge archive in seconds.

data_life_time: Data life time of circular recording.

- **cur:** Current value. (Unit: day).
- **max:** Max possible value. (Unit: day)
- **min:** Minimum possible value. (Unit: day)

record_circular:

- **enable:** Enable circular recording or not. (true/false)

record_post_alarm: Post alarm setting.

- **enable:** Enable or not. (true/false)
- **cur:** Current value. (Unit: second)
- **max:** Max possible value. (Unit: second)
- **min:** Minimum possible value. (Unit: second)

record_pre_alarm: Pre alarm setting.

- **enable:** Enable or not. (true/false)
- **cur:** Current value. (Unit: second)
- **max:** Max possible value. (Unit: second)
- **min:** Minimum possible value. (Unit: second)

Reply Example:

```
{”code”:0,”msg”:”Success.”,”name”:”/record/config”,”output”:{”data_
life_time”:{”cur”:60,”max”:60,”min”:0},”edge_archiver_
interval”:15,”record_circular”:{”enable”:false},”record_post_alarm”:
```

{”cur”:15,”enable”:true,”max”:120,”min”:15},”record_pre_alarm”:{”cur”:0,”enable”:true,”max”:120,”min”:0}}}

Syntax	http://{server_IP}/record/config
Method	POST
Parameters	{"data_life_time":{"cur":%d,"max":%d,"min":%d},"edge_archiver_interval":%d,"record_c ircular":{"enable":%d},"record_post_alarm":{"cur":%d,"enable":%d,"max":%d,"min":%d},"record_pre_alarm":{"cur":%s,"enable":%d,"max":%d,"min":%d}}
Example	http://192.168.6.51/record/config

Record Group Information Get

Purpose: To get the record group configuration.

Parameters: None.

Reply: JSON format as follows:

{“code”:“ReplyCode”,”name”:“APIName”,“msg”:“ReplyContent”,“output”:{”record_info”:[{”path”:[{”mount_path”:%s,”export_path”:%s,”snap_path”:%s,”total_size_mb”:%d,”available_size_mb”:%d}],”group_id”:%d,”sys_time_offset”:%d,”sys_start_time”:%d,”HDD_num_max”:%d,”is_circular”:%d}],”group_support”:%d}}

group_support: NVR support group or not. (true/false)

record_info: Each element inside this array represents a disk group relative setting.

- **group_id:** Index of group.
- **sys_time_offset:** System time offset in second.
- **sys_start_time:** System start-time in second.
- **HDD_num_max:** Max available HDD number.
- **is_circular:** Enable circular or not. (1:Enable, 0:DIsable)

- **path:**
  - **mount_path:** Disk mounting path.
  - **export_path:** Disk export math.
  - **snap_path:** Snapshot saved path.
  - **total_size_mb:** Total size in MB.
  - **available_size_mb:** Remained size in MB.

Reply Example:


```
{”code”:0,”msg”:”record_infoSuccess”,”name”:”/record/record_
info”,”output”:{”record_info”:[{”path”:[{”mount_path”:”/ext_
000”,”export_path”:”/ext_export_000”,”snap_path”:”/ext_snap_
000”,”total_size_mb”:917196,”available_size_mb”:774171}],”group_
id”:0,”sys_time_offset”:0,”sys_start_time”:1497852889,”HDD_num_
max”:1,”is_circular”:0}],”group_support”:true}}
```


Syntax	http://{server_IP}/record/record_info
Method	GET
Parameters	NONE
Example	http://192.168.6.51/record/record_info

7

Camera relative API

Camera List Get

Purpose: To get the information from connected camera including the channel title, port, name and model.

Parameters: None.

Reply: JSON format as follows:

```
{“code”:“ReplyCode”,”name”:“APIName”,“msg”:“ReplyContent”,“outpu
t”:{“camera_list”:[{“account”:%s,“ch”:%d,“http_
port”:%d,“ip”:%s,“mac:%s,“model”:%s,“password”:%s,“protocol”:%s,“
rtsp_port”:%d,“title”:%s,“type”:%d},…]}}
```

camera_list:

- **account:** IP camera user account
- **password:** IP camera user password (encrypted password, checkout the appendix to decrypt.)

- **ch:** channel number
- **title:** channel title
- **ip:** IP address of the IP camera
- **mac:** MAC address of the IP camera
- **protocol:** RTSP protocol of the IP camera
- **rtsp_port:** RTSP port of the IP camera
- **http_port :** HTTP port of the IP camera
- **type:** device type

Reply Example:


```
{”code”:0,”msg”:”/information/camera_list_
getSuccess!”,”name”:”/information/camera_list”,”output”:{”camera_
list”:[{”account”:”Admin”,”ch”:0,”http_
port”:80,”ip”:”192.168.6.142”,”mac”:””,”model”:””,”password”:”T5sB5o
vTjF6sEPCj”,”protocol”:”RTP+RTSP”,”rtsp_
port”:554,”title”:”NewDevice”,”type”:1}]}}
```


Syntax	http://{server_IP}/information/camera_list
Method	GET
Parameters	NONE
Example	http://192.168.6.51/information/camera_list

Camera Search

Purpose: To search available camera.

Parameters:

```
model=%d
```

model: The device type of the camera.

Reply:

```
{“code”:“ReplyCode”,”name”:“APIName”,“msg”:“ReplyContent”,“output”:
{”total”:%d,”dev_list”:
[{”model”:%s,”proj”:%s,”title”:%s,”ip”:%s,”netmask”:%s,”gateway”:%s
,”dns”:%s,”mac”:%s,“http_port”:%d,“type”:%d},…]}}
```

total: Total number of found camera list.

dev_list:

- **model:** The model name of the camera.
- **proj:** The project name of the camera.
- **title:** The title of the camera.
- **ip:** The IP address of the camera.
- **netmask:** The netmask of the camera.
- **gateway:** The gateway of the camera.
- **dns:** The DNS address of the camera.
- **mac:** The MAC address of the camera.
- **http_port:** The HTTP port of the ca,era.
- **type:** The device type.

Reply Example:

```
{”code”:0,”msg”:”device_
searchSuccess!”,”name”:”/camera/devicesearch”,”output”:
{”total”:4,”dev_list”:[{”model”:”R2SI-G”,”proj”:”R2SI-
G“,”title”:”Howard_test_
S3L”,”ip”:”192.168.0.179”,”netmask”:”255.255.255.0”,”gateway”:”192.
168.0.254”,“dns”:”0.0.0.0”,”mac”:”00:D0:89:14:33:38”,”http_
port”:80,”type”:1},{”model”:”R3V6-L”,”proj”:”R3V6-
L”,”title”:”Howard_test_
S3Lm”,”ip”:”192.168.6.105”,”netmask”:”255.255.255.0”,”gateway”:”192
.168.6.254”,”dns”:”192.168.10.1”,”mac”:”00:D0:89:13:7C:68”,”http_
port”:80,”type”:1},{”model”:”831R2SH-T4”,”proj”:”831R2SH-
T4”,”title”:”IRPTZ”,”ip”:”192.168.6.126”,”netmask”:”255.255.255.0”,
”gateway”:”192.168.6.254”,“dns”:”192.168.10.1”,”mac”:”00:D0:89:52:9
1:28”,”http_port”:80,”type”:1},{”model”:”R3SD-L”,”proj”:”R3SD-
L”,”title”:”MegaPixelCamera”,”ip”:”192.168.6.142”,”netmask”:”255.25
```

5.255.0”,”gateway”:”192.168.6.254”,”dns”:”192.168.10.1”,”mac”:”00:D0:89:15:44:50”,”http_port”:80,”type”:1}]}}

Syntax	http://{server_IP}/camera/devicesearch
Method	GET
Parameters	NONE
Example	http://192.168.6.51/camera/devicesearch

Camera Connection Setup

Purpose: To setup the camera connection.

Parameters: JSON format as follows:

```
{”list”:
[{”model”:%s,”proj”:%s,”title”:%s,”ip”:%s,”mac”:%s,”dev”:%s,”http_port”:%d,”type”:%d,”rtsp_port”:%d,”protocol”:%d,”account”:”Admin”,”password”:”GtBcsw==”,”cmd”:%d,“ch”:%d}]}
```

Reply: JSON format as follows:

```
{“code”:“ReplyCode”,”name”:“APIName”,“msg”:“ReplyContent”,“output”:{”list”:[{”account”:%s,”cmd”:%d,”dns”:%s,”gateway”:%s,”http_port”:%s,”ip”:%s,”mac”:%s,”model”:%s,”netmask”:%s,”password”:%s,”proj”:%s,”protocol”:%s,”rtsp_port”:%d,”title”:%s,”type”:%s}]}}
```

list:

- **cmd:** 0: Connect device. 1: Disconnect device. 2: Edit device setting.
- **model:** The model name of the camera.
- **proj:** The project name of the camera.
- **title:** The title of the camera.
- **ip:** The IP address of the camera.
- **netmask:** The netmask of the camera.
- **gateway:** The gateway of the camera.
- **dns:** The DNS address of the camera.

- **mac:** The MAC address of the camera.
- **http_port:** The HTTP port of the camera.
- **type:** The device type of the device.

Reply Example:


```
{”code”:0,”msg”:””,”name”:”/camera/camera_connection”,”output”:
{”list”:
[{”account”:”Admin”,”cmd”:0,”dns”:”192.168.10.1”,”gateway”:”192.168
.6.254”,”http_
port”:80,”ip”:”192.168.6.105”,”mac”:”00:D0:89:13:7C:68”,”model”:”R3
V6-L”,”netmask”:”255.255.255.0”,”password”:”1234”,”proj”:”R3V6-
L”,”protocol”:”RTP+RTSP”,”rtsp_port”:554,”title”:”Howard_test_
S3Lm”,”type”:1}]}}
```


Syntax	http://{server_IP}/camera/camera_connection
Method	POST
Parameters	{"list":[{"model":%s,"proj":%s,"title":%s,"ip":%s,"mac":%s,"dev":%s,"http_port":%d ,"type":%d,"rtsp_port":%d,"- protocol":%d,"account":"Admin","password":"GtBcsw==","cmd":%d,“ch”:%d}]}
Example	http://192.168.6.51/camera/camera_connection

Camera Disconnect All

Purpose: To disconnect all cameras.

Parameters: None.

Reply: JSON format as follows:

```
{“code”:“ReplyCode”,”name”:“APIName”,“msg”:“ReplyContent”}
```

Reply Example:


```
{”code”:0,”name”:”/camera/clear_all”,”msg”:”/camera/clear_
allSuccess.”}
```

Syntax	http://{server_IP}/camera/clear_all
Method	GET
Parameters	NONE
Example	http://192.168.6.146/camera/clear_all

8

Appendix

Encrypt/Decrypt Password

Purpose: To decrypt the password from the server, and encrypt the password before send to server.

JavaScript Code:

```
var aes = require('aes-js');    /*https://github.com/ricmoo/aes-js*/

var aes_key = [0, 1, 2, 3, 4, 5, 6, 7, 8, 22, 33, 55, 120, 12, 76, 87];

function decrypt_remote_text(text){

   var decrypt_text = "";

   var b = new Buffer(text, "base64");

   var text_bytes = aes.util.convertStringToBytes(b);
```

```
   var aes_ctr = new aes.ModeOfOperation.ctr(aes_key, new
aes.Counter(0));

   var decrypted_bytes = aes_ctr.decrypt(text_bytes);

   decrypt_text = aes.util.convertBytesToString(decrypted_bytes);

   return decrypt_text;

}


function encrypt_remote_text(text){

var encrypt_text = “”;

   var text_bytes = aes.util.convertStringToBytes(text);

   var aes_ctr = new aes.ModeOfOperation.ctr(aes_key, new
aes.Counter(0));

   var encrypted_bytes = aes_ctr.encrypt(text_bytes);

   var b = new Buffer(encrypted_bytes);

   encrypt_text = b.toString(’base64’);

   return encrypt_text;

}
```

MOBOTIX

BeyondHumanVision

EN_08/21

MOBOTIX AG • Kaiserstrasse • D-67722 Langmeil • Tel.: +49 6302 9816-103 • sales@mobotix.com • www.mobotix.com