

```
except OSError:
    break
if v and vt == winreg.REG_SZ and os.path.isdir(vc_dir):
    try:
        version = int(float(v))
    except (ValueError, TypeError):
        continue
    if version >= 14 and version > best_version:
        best_version, best_dir = version, vc_dir
return best_version, best_dir

def _find_vc2017():
    """Returns "15, path" based on the result of invoking 'vcvarsall.bat'
    if no install is found, returns "None, None"
    """
    try:
        path = subprocess.check_output(
            os.path.join(root, "Microsoft Visual Studio", "VC", "bin", "vswhere.exe"),
            stderr=subprocess.STDOUT,
            cwd=os.path.join(root, "Microsoft Visual Studio", "Common7", "Tools", "x64",
                             "Installation"),
            encoding='mbcs', errors='strict').strip()
    except (subprocess.CalledProcessError, UnicodeDecodeError):
        return None, None

    path = os.path.join(path, "VC", "Auxiliary", "Build")
    if os.path.isdir(path):
        return 15, path

    return None, None

PLAT_SPEC_TO_RUNTIME = {
    'x86' : 'x86',
    'x86_amd64' : 'x64',
    'x86_arm' : 'arm',
    'x86_arm64' : 'arm64'
}

def _find_vcvarsall(plat_spec):
    # bpo-38597: Removed vcruntime return value
    _, best_dir = _find_vc2017()

    if not best_dir:
        best_version, best_dir = _find_vc2015()
```

# Reference Manual

## MOBOTIX MOVE NVR API V3.0.1

© 2021 MOBOTIX AG



Beyond Human Vision

MOBOTIX MOVE

# Table of Contents

---

<b>Table of Contents</b> .....	<b>2</b>
<b>HTTP Request</b> .....	<b>5</b>
Request Message .....	5
Response Message .....	6
Response Status Code .....	6
<b>Log in/out Service API</b> .....	<b>9</b>
Log in Service .....	9
Log out Service .....	10
<b>Playback relative API</b> .....	<b>11</b>
Playback Key Get .....	11
Playback Configuration Set .....	12
Playback Bar Get .....	13
Event List Page Get .....	14
System Log Page Get .....	15
Export Channel List Get .....	16
Export File .....	17
Export File Status Get .....	18
Export File List Get .....	18
Export File Remove Set .....	19
<b>Information relative API</b> .....	<b>21</b>
NVR Version Get .....	21
Local Time Get .....	22
Network Information Get .....	22
User Information Get .....	24
Time Configuration Get .....	25

---

<b>Setup relative API</b> .....	<b>29</b>
Default Connection Setting Get .....	29
Default Connection Setting Set .....	30
Load Default Setting .....	31
System Monitor Configuration Get .....	32
System Monitor Configuration Set .....	34
Event Management Configuration Get .....	36
Event Management Configuration Set .....	39
Email Setting Get .....	42
Email Setting Set .....	43
DDNS Setting Get .....	45
DDNS Setting Set .....	45
Account Add .....	46
Account Delete .....	48
Account Edit .....	50
Account Authority Change .....	52
<b>Record relative API</b> .....	<b>55</b>
Template Configuration Get .....	55
Template Configuration Set .....	57
Record Configuration Get .....	60
Record Configuration Set .....	61
Record Group Information Get .....	63
<b>Camera relative API</b> .....	<b>65</b>
Camera List Get .....	65
Camera Search .....	66
Camera Connection Setup .....	68
Camera Disconnect All .....	69
<b>Appendix</b> .....	<b>71</b>
Encrypt/Decrypt Password .....	71

---

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.

## 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: <server 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.





## 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") = afd53c19b43825da4b297c203e4460.** So the parameter will be "account = afd53c19b43825da4b297c203e4460"

#### 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	<a href="http://192.168.6.51/login">http://192.168.6.51/login</a>

---

## 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	<a href="http://192.168.6.51/logout">http://192.168.6.51/logout</a>

---

---

## 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	<a href="http://192.168.6.51/datasearch/set_pbspeed">http://192.168.6.51/datasearch/set_pbspeed</a>
---------	---

---

## 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”:  
[0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,1,1,1,1,1,1,1,0,0,0,0,0,0,0,0,  
0,0,0,0,0,0,0,0,0]}}
```

---

Syntax	<a href="http://{server_IP}/datasearch/get_pbbar">http://{server_IP}/datasearch/get_pbbar</a>
--------	---

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

---

Parame start\_time=%d&end\_time=%d&section\_num=%d&group\_id=%d  
ter

Exmpl http://192.168.6.51/datasearch/get\_pbbar?start\_time=1495767531&end\_time-  
e e=1497849562&sec  
tion\_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_idx”:%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\_idx:** 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_
idx”:0,“ip_addr”:“”,“mac”:“”,“snap_path”:“”},{“start_
time”:1497852943,“channel”:4,“dev_type”:0,“evt_type”:1,“evt_
idx”:0,“ip_addr”:“”,“mac”:“”,“snap_path”:“”}]}}
```

Syntax [http://{server\\_IP}/datasearch/evtlist\\_page\\_get](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](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": "getlogs success.",
  "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",
          "time": 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"}
```

## Playback relative API

### Export File Status Get

---

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	<a href="http://192.168.6.51/datasearch/remote_export_async">http://192.168.6.51/datasearch/remote_export_async</a>

---

## 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	<a href="http://192.168.6.51/datasearch/remote_export_status">http://192.168.6.51/datasearch/remote_export_status</a>

---

## 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	<a href="http://192.168.6.51/datasearch/remote_export_list">http://192.168.6.51/datasearch/remote_export_list</a>

# 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	<a href="http://{server_IP}/datasearch/remove_remote_export_list">http://{server_IP}/datasearch/remove_remote_export_list</a>
Method	POST
Parameters	{“name”:%s}
Example	<a href="http://192.168.6.51/datasearch/remove_remote_export_list">http://192.168.6.51/datasearch/remove_remote_export_list</a>

---

---

## 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", "output": { "remote_version": "2016.11.16.b162cbc", "ap_version": "A9999999999999", "num_ch": 32, "num_alarm": 4, "num_relay": 2, "model": "H100", "proj": "NVR" } }
```

Syntax	<a href="http://{server_IP}/information/version">http://{server_IP}/information/version</a>
Method	GET
Parameters	NONE
Example	<a href="http://192.168.6.51/information/version">http://192.168.6.51/information/version</a>

---

## 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	<a href="http://{server_IP}/information/local_time">http://{server_IP}/information/local_time</a>
Method	GET
Parameters	NONE
Example	<a href="http://192.168.6.51/information/local_time">http://192.168.6.51/information/local_time</a>

---

## 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”,“dch  
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”,“dch  
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







**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 5<sup>th</sup> week's Sunday 2:45 ~ April's 1<sup>st</sup> 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_configs success.", "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	<a href="http://192.168.6.51/information/time_config">http://192.168.6.51/information/time_config</a>

---



---

## 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”,“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": "device_default_accountgetsucess", "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	<code>http://{server_IP}/setup/default_connection_info</code>
Method	GET
Parameters	NONE
Example	<a href="http://192.168.6.51/setup/default_connection_info">http://192.168.6.51/setup/default_connection_info</a>

---

## 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.

## Setup relative API

### System Monitor Configuration Get

---

**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	<a href="http://{server_IP}/setup/load_default">http://{server_IP}/setup/load_default</a>
Method	GET/POST
Parameters	NONE
Example	<a href="http://192.168.6.51/setup/load_default">http://192.168.6.51/setup/load_default</a>

---

# 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_configs 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": 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	<a href="http://{server_IP}/setup/system_monitor_config">http://{server_IP}/setup/system_monitor_config</a>
Method	GET
Parameters	NONE
Example	<a href="http://192.168.6.51/setup/system_monitor_config">http://192.168.6.51/setup/system_monitor_config</a>

---

## 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,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,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,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	<a href="http://192.168.6.51/setup/event_action_config">http://192.168.6.51/setup/event_action_config</a>

## 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".

## Setup relative API

### Event Management Configuration Set

---

- **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, f
alse, 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, f
alse, 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, f
alse, 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, f
alse, 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" ] ] }
```

## Setup relative API

### Email Setting Get

---

Syntax	<code>http://{server_IP}/setup/event_action_config</code>
Method	POST
Parameters	<code>{"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, ...]}</code>
Example	<a href="http://192.168.6.51/setup/event_action_config">http://192.168.6.51/setup/event_action_config</a>

---

## 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_configs success.",
  "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	<a href="http://192.168.6.51/setup/email_config">http://192.168.6.51/setup/email_config</a>

## 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": {
      "enabled_ssl": %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_

```

## Setup relative API

### Email Setting Set

---

```
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 recopent.

**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](http://{server_IP}/setup/email_config)

Method POST

Parameters { "account":%s,"address":[%s, %s,%s],"current\_pro-  
vider":%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](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_configs success.",
  "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:

## Setup relative API

### Account Add

---

```
{ "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": "", "password": "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	<a href="http://192.168.6.51/setup/ddns_config">http://192.168.6.51/setup/ddns_config</a>

---

## Account Add

**Purpose:** To add a new account.

**Parameters:** JSON format as follows:

```
{ "account": "%s", "password": %s }
```

**account:** New user account.

---

















## Setup relative API

### Account Authority Change

---

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

---

---

## 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”,“output”:{“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_templategetsucess",
  "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": [0,0,0,0,0,0,0,0,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0],
        "Mon": [0,0,0,0,0,0,0,0,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0],
        "Sat": [0,0,0,0,0,0,0,0,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0],
        "Sun": [0,0,0,0,0,0,0,0,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0],
        "Thu": [0,0,0,0,0,0,0,0,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0],
        "Tue": [0,0,0,0,0,0,0,0,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0],
        "Wed": [0,0,0,0,0,0,0,0,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0],
        "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_
```

## Record relative API

### Template Configuration Set

---

```
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":
[0,0,0,0,0,0,0,0,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],Mon:
[0,0,0,0,0,0,0,0,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],Sat:
[0,0,0,0,0,0,0,0,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],Sun:
[0,0,0,0,0,0,0,0,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],Thu:
[0,0,0,0,0,0,0,0,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],Tue:
[0,0,0,0,0,0,0,0,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],Wed:
[0,0,0,0,0,0,0,0,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]},Template:{Fri:
[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],Mon:
[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],"Sat":
[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],"Sun":
[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],"Thu":
[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],"Tue":
[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],"Wed":
[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]},"Template(2)":
{"Fri":[2,0,0,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],"Mon":
[2,0,0,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],"Sat":
[2,0,0,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],"Sun":
[2,0,0,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],"Thu":
[2,0,0,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],"Tue":
[2,0,0,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],"Wed":
[2,0,0,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]}}}}}

```

Syntax [http://{server\\_IP}/record/record\\_template](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	<a href="http://{server_IP}/record/config">http://{server_IP}/record/config</a>
Method	GET
Parameters	NONE
Example	<a href="http://192.168.6.51/record/config">http://192.168.6.51/record/config</a>

## 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\_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}}

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_
```

## Record relative API

### Record Group Information Get

---

```
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	<a href="http://{server_IP}/record/record_info">http://{server_IP}/record/record_info</a>
Method	GET
Parameters	NONE
Example	<a href="http://192.168.6.51/record/record_info">http://192.168.6.51/record/record_info</a>

---



---

## 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”,“output”:{“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

## Camera relative API

### Camera Search

---

- **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	<a href="http://192.168.6.51/information/camera_list">http://192.168.6.51/information/camera_list</a>

---

## 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 camera.
- **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:D0:89:52:91:28",
        "http_port": 80,
        "type": 1
      },
      {
        "model": "R3SD-L",
        "proj": "R3SD-L",
        "title": "MegaPixelCamera",
        "ip": "192.168.6.142",
        "netmask": "255.255.255.0",
        "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	<a href="http://192.168.6.51/camera/devicesearch">http://192.168.6.51/camera/devicesearch</a>

# 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", "pro  
j": "%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": "R3V  
6-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`



---

## 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\_09/21

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

MOBOTIX is a trademark of MOBOTIX AG registered in the European Union, the U.S.A., and in other countries. Subject to change without notice. MOBOTIX do not assume any liability for technical or editorial errors or omissions contained herein. All rights reserved. © MOBOTIX AG 2021