Synology
Surveillance Station Web API

ver: 1.3

SYNOLOGY INCORPORATED

Table of Contents

1. Overview

Surveillance Station provides a programmable interface allowing the 3rd party integrator/installer to develop application that is highly integrated with Surveillance Station. This interface is called "Surveillance Station Web API" or "SS Web API", refer to Figure 1-1 for the entire structure:

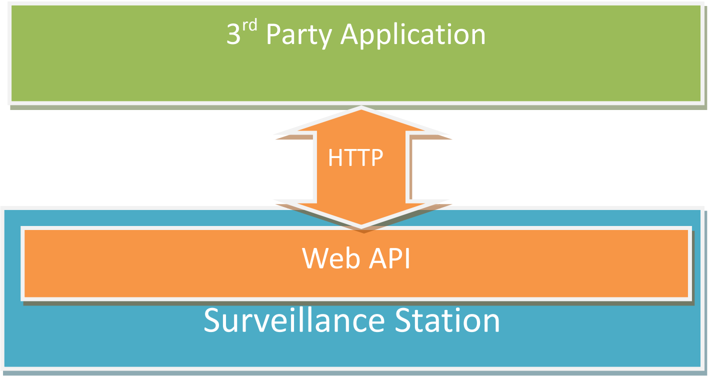

Figure 1-1 Surveillance Station Web API architecture

SS Web API is based on HTTP protocol, including functions like camera configuration, PTZ control, live view streaming, video playback, and … etc.

2. SS Web API Specification

2.1 Concept

SS Web API provides a set of API interface allowing the 3rd party application to interact with Surveillance Station via HTTP Request/Response call.

2.1.1 API Definition

Every API will define the following items:

- **Name**
  It will go after “SYNO.XXX.YYY”. **XXX** will represent the application name, **YYY** will represent the feature category. For example, **SYNO.SurveillanceStation.Camera**, **SurveillanceStation** means the API is provided by the Surveillance Station, **Camera** means this API will retrieve the camera configurations.

- **URL path**
  SS Web API does not preserve a constant URL for every API. Instead, 3rd party applications need to use the command **SYNO.API.Info** to retrieve every API’s URL path. **SYNO.API.Info** is the only API with constant URL, and it does not need authorization, its path is /webapi/query.cgi

- **Method**
  Every method is unique in its own API, and it has to be defined when you use the API. For example, **SYNO.SurveillanceStation.Camera API**, call the method **GetInfo** will retrieve some camera configurations.

- **Version**
  SS Web API will require you designate an API version to ensure the result is within the expectation. SYNO.API.Info will return every API’s supported versions. “Not supported version” will be returned if API is not supported. We may drop the support on the old API version, make sure you keep this in mind.

2.1.2 Operation flow

Ensure your application has followed the protocols below to interact with Surveillance Station via SS Web API:

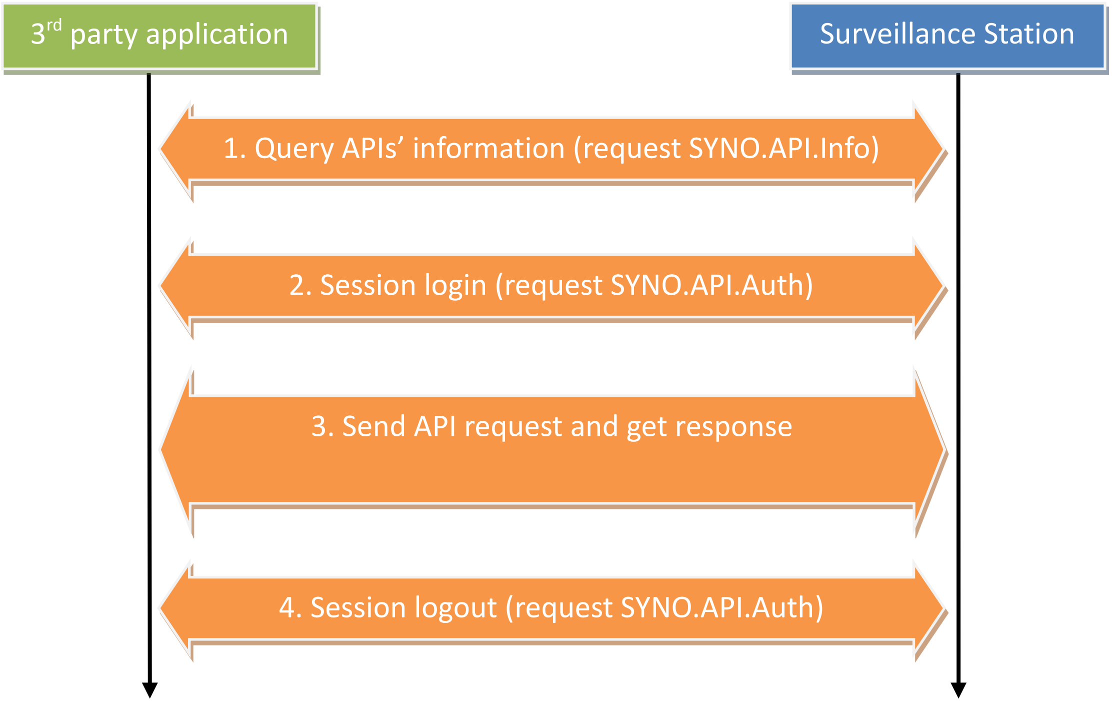
3rd party application
Surveillance Station
1. Query APIs' information (request SYNO.API.Info)
2. Session login (request SYNO.API.Auth)
3. Send API request and get response
4. Session logout (request SYNO.API.Auth)

Figure 2-1 Operation flow of Web API

Step 1. Query APIs' information

SS Web API will require you to log in a session before sending any API request. However, the URL of SYNO.API.Auth remains unknown until you have queried it using **Query** in SYNO.API.Info. All the others' API URL, versions will also be retrieved in this step.

Step 2. Session login

After retrieving the URL for SYNO.API.Auth from step 1, your application can call **Login** method to complete the login process. A HTTP Session Cookie will be returned for authentication after you have successfully log in.

Step 3. Send API request and get response

Your application may now start calling any API requests to interact with Surveillance Station, like retrieving camera configurations, live view streaming, or search recorded clips.

Step 4. Session logout

When the communication is over, your application should call **Logout** method in SYNO.API.Auth to end the session.

2.2 Request & Response Structure

SS Web API is based on HTTP protocol, and Request/Response as the communication structure.

2.2.1 Request

Use HTTP Get/Post to send the request with API’s URL.

Usage:

```
GET /webapi/<URL_PATH>?
api=<API>&method=<METHOD>&version=<VERSION>[&_sid=<SESSION_ID>][&<PARAM_LIST>]
```

Tag	Description
<URL_PATH>	API’s URL path
<API>	Name of the API
<METHOD >	Name of the API method
<VERSION>	The version of API
<SESSION_ID>	Optional, the designate session ID. See 2.3.2
<PARAM_LIST>	Optional, the parameters of the API Method

An Example to retrieve the camera list:

```
GET /webapi/SurveillanceStation/camera.cgi?
api=SYNO.SurveillanceStation.Camera&method=List&version=1&offset=10&limit=3&_sid=Jn5dZ9aS95wh2
```

<URL_PATH>: SurveillanceStation/camera.cgi

<API>: SYNO.SurveillanceStation.Camera

<METHOD >: List

<VERSION>: 1

<PARAM_LIST>: offset=10&limit=3

<SESSION_ID> : Jn5dZ9aS95wh2

To get more detail of session ID, please refer to SYNO.API.Auth.

2.2.2 Response

After receiving the request, API will return the response to the 3rd party application in JSON format.

JSON Containers:

Attribute	Value	Description
success	<boolean>	Whether this request is successful or not.

data	<Result Object>	Optional. It will return data info if this request is successful, for more details please refer to chapters for each API method.
error	<Error Object>	Optional. It will return the error info if this request fails.

<Error Object> Definitions:

Attribute	Value	Description
code	<integer>	The error code defined in 2.2.3.

Example 1: Unable to retrieve the camera list when the API version is wrong

```
{
        "success": false,
        "error": {
                "code": 104
        }
}
```

Example 2: Retrieve the camera list successfully

```
{
        "success": true,
        "data": {
                “total”: 38,
                “offset”: 10,
                “cameras”: {…}
        }
}
```

2.2.3 Common API Error Code

The table shown below describes the general error codes which might be returned by all APIs. For customized error codes of each API, please refer to the corresponding API Method sections to get more details.

Error Code	Description
100	Unknown error
101	Invalid parameters
102	API does not exist
103	Method does not exist
104	This API version is not supported
105	Insufficient user privilege
106	Connection time out
107	Multiple login detected

2.3 API List

The following table is the overview of all APIs defined in this section:

API Name	Description	Section
SYNO.API.Info	Discover all API information	2.3.1
SYNO.API.Auth	Perform session login and logout	2.3.2
SYNO.SurveillanceStation.Info	Retrieve SS-related general information	2.3.3
SYNO.SurveillanceStation.Camera	Retrieve camera-related information	2.3.4
SYNO.SurveillanceStation.PTZ	Perform camera PTZ actions	2.3.5
SYNO.SurveillanceStation.ExternalRecording	Control external recording of cameras	2.3.6
SYNO.SurveillanceStation.Event	Query event information	2.3.7
SYNO.SurveillanceStation.Device	Get information of Visual Station and CMS	2.3.8
SYNO.SurveillanceStation.Emap	Get information of defined E-Maps.	2.3.9
SYNO.SurveillanceStation.Streaming	Get video stream of live view and recorded events	2.3.10

Each API has its following definition:

I. Basic Information

   i. Name: The API name to replace <API> in Web API request syntax.
   ii. Availability: The version of DSM or Surveillance Station which start to support this API.
   iii. Version: The current version of this API.

II. Methods

   i. Request: The request parameters of this method.
   ii. Response: The returned data object containing response keys.

Name of a method is in upper camel case, while name of a request parameter or a response key is in lower camel case for discrimination.

III. Error Code

Corresponding error codes to each API.

2.3.1 SYNO.API.Info

This is the starter API which has its fixed URL path /webapi/query.cgi. The 3rd party application must retrieve other APIs' information by calling “Query” method.

API Name	Version	Availability
SYNO.API.Info	1	DSM 3.1-1594

Method Name	Section	Availability
Query	2.3.1.1	1 and onward

2.3.1.1 Query method

Discover available APIs and corresponding information.

Request

Parameter	Value	Description	Availability
query	ALL, <string>	• ALL: Get information of all available APIs. • <string>: The list of <API Query Unit> to be queried concatenated by “,”.	1 and onward

<API Query Unit> definition:

Notation	Value	Description	Availability
<API Query Unit>	<API Query Name>, <API Query Prefix>	Unit of API query target. It could be full name of an API, or prefix of APIs.	1 and onward
<API Query Name>	<string>	Full name of API to be queried. Must be in “SYNO.XXX.YYY” format. Ex: SYNO.API.Auth.	1 and onward
<API Query Prefix>	<string>	Prefix of APIs to be queried. Must be in “SYNO.XX.” format and ended with “.”. Ex: “SYNO.SurveillanceStation.” will return the information of APIs with this prefix such as “SYNO.SurveillanceStation.Camera”, “SYNO.SurveillanceStation.PTZ”, etc.	1 and onward

Example:

Get information of SYNO.SurveillanceStation.Camera

```
GET /webapi/query.cgi?
api=SYNO.API.Info&method=Query&version=1&query=SYNO.SurveillanceStation.Camera
```

Get information of APIs with the prefix “SYNO.SurveillanceStation.”.

```
GET /webapi/query.cgi?
api=SYNO.API.Info&method=Query&version=1&query=SYNO.SurveillanceStation.
```

Get information of SYNO.API.Auth and the APIs with the prefix “SYNO.SurveillanceStation.”.

```
GET /webapi/query.cgi?
api=SYNO.API.Info&method=Query&version=1&query=SYNO.API.Auth,SYNO.SurveillanceStation.
```

Response

Object contains <API Description Objects> list of the requested API(s).

<API Description Object> definition:

Key	Value	Description	Availability
key	<string>	The string of API name	1 and onward
path	<string>	The string of URL path	1 and onward
minVersion	<integer>	The minimum supported API version	1 and onward
maxVersion	<integer>	The maximum supported API version	1 and onward

Example: Return information of SYNO.API.Auth and the APIs with the prefix “SYNO.SurveillanceStation.”.

```
{
        "SYNO.API.Auth": {
                "path": "auth.cgi",
                "minVersion": 1,
                "maxVersion": 1
        },
        "SYNO.SurveillanceStation.Info": {
                "path": "SurveillanceStation/info.cgi",
                "minVersion": 1,
                "maxVersion": 1
        },
        "SYNO.SurveillanceStation.Camera": {
                "path": "SurveillanceStation/camera.cgi",
                "minVersion": 1,
                "maxVersion": 2
        },
        "SYNO.SurveillanceStation.PTZ": {
                "path": "SurveillanceStation/ptz.cgi",
                "minVersion": 1,
                "maxVersion": 1
        },
        "SYNO.SurveillanceStation.ExternalRecording": {
                "path": "SurveillanceStation/extrecord.cgi",
                "minVersion": 1,
                "maxVersion": 1
        },
        "SYNO.SurveillanceStation.Event": {
                "path": "SurveillanceStation/event.cgi",
                "minVersion": 1,
                "maxVersion": 1
        },
        "SYNO.SurveillanceStation.Device": {
                "path": "SurveillanceStation/device.cgi",
        "minVersion": 1,
        "maxVersion": 1
    },
    "SYNO.SurveillanceStation.Streaming": {
        "path": "SurveillanceStation/streaming.cgi",
        "minVersion": 1,
        "maxVersion": 1
    },
    "SYNO.SurveillanceStation.Emap": {
        "path": "SurveillanceStation/emap.cgi",
        "minVersion": 1,
        "maxVersion": 1
    }
}
```

2.3.2 SYNO.API.Auth

API used to perform session login and logout.

API Name	Version	Availability
SYNO.API.Auth	1	DSM 4.0-2198
	2	DSM 4.0-2251

Method Name	Section	Availability
Login	2.3.2.1	1 and onward
Logout	2.3.2.2	1 and onward

2.3.2.1 Login method

Create new login session. Different accounts have different privilege settings; accounts in DSM admin group will have the highest privilege to all camera, camera groups, and Slave DiskStations.

Request

Parameter	Value	Description	Availability
account	<string>	Login account name	1 and onward
passwd	<string>	Login account password	1 and onward
session	<string>	*Optional.* Application session name. User can assign "SurveillanceStation" to this parameter to login SurveilllanceStation. If not specified, default session is DSM, and SurveillanceStation is also available.	1 and onward
format	cookie, sid	*Optional.* If format is "cookie", session ID is included in both response header and response json data. If format is "sid", se ssion ID is not included in response header, but response json data only. User can append this session ID manually to get access to any other Web API without interrupting other logins. If not specified, default login format is "cookie."	2 and onward

Example1:

```
GET /webapi/<URL_PATH>?
api=SYNO.API.Auth&method=Login&version=1&account=admin&passwd=123456&session=SurveillanceSta
tion
```

Example2:

```
GET /webapi/<URL_PATH>?
```

```
api=SYNO.API.Auth&method=Login&version=2&account=admin&passwd=123456&session=SurveillanceSta
tion&format=sid
```

Response

Name	Value	Description	Availability
sid	<string>	session ID	2 and onward

Example:

```
{
	"sid":"Jn5dZ9aS95wh2"
}
```

If login format is “cookie”, server will deliver header information including Cookie ID.

```
HTTP/1.1 200 OK
Date: Thu, 31 May 2012 09:40:43 GMT
Server: Apache/2.2.22 (Unix)
P3P: CP="IDC DSP COR ADM DEVi TAIi PSA PSD IVAi IVDi CONi HIS OUR IND CNT"
Set-Cookie: id=Jn5dZ9aS95wh2;path=/
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 37
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/plain; charset="UTF-8"
```

If the login format is “sid”, user should append the returned sid in any webapi request.

Example:

```
GET /webapi/<URL_PATH>?
api=SYNO.SurveillanceStation.Camera&method=List&version=1&_sid=Jn5dZ9aS95wh2
```

By sid format login and appending _sid to other webapi, the sessions would not affect each others and different privilege settings can be applied simultaneously.

2.3.2.2 Logout method

Destroy current login session.

Request

Name	Value	Description	Availability
session	<string>	*Optional.* Application session name. User can assign “SurveillanceStation” to this	2 and onward

		parameter to logout SurveillanceStation. If not specified, default session is DSM.	

Example1:

```
GET /webapi/<URL_PATH>?
api=SYNO.API.Auth&method=Logout&version=1&session=SurveillanceStation
```

Example2:

```
GET /webapi/<URL_PATH>?
api=SYNO.API.Auth&method=Logout&version=2&session=SurveillanceStation&_sid=Jn5dZ9aS95wh2
```

Response

This method has no specific response data. It returns an empty success response if it completes without error.

2.3.2.3 API Error Code

Code	Description
100	Unknown error.
101	The account parameter is not specified.
400	Invalid password.
401	Guest or disabled account.
402	Permission denied.
403	One time password not specified.
404	One time password authenticate failed.

2.3.3 SYNO.SurveillanceStation.Info

This API provides a method to acquire SS-related information such as package version, package UI path, and the total number of camera and installed licenses.

API Name	Version	Availability
SYNO.SurveillanceStation.Info	1	Surveillance Station 6.0-2337

Method Name	Section	Availability
GetInfo	2.3.3.1	1 and onward

2.3.3.1 GetInfo method

Get SS-related general information. If the user is logged in, the complete information is provided. Otherwise only version and path information is sent.

Request

No parameter is required.

Example:

```
GET /webapi/<URL_PATH>?
api=SYNO.SurveillanceStation.Info&method=GetInfo&version=1
```

Response

Name	Value	Description	Availability
version	<Version Object>	Version object to represent package version of Surveillance Station. For all users.	1 and onward
path	<string>	UI path to Surveillance Station. For all users.	1 and onward
customizedPortHttp	<integer>	*Optional.* Customized port of Surveillance Station (HTTP). For Surveillance-login users only.	1 and onward
customizedPortHttps	<integer>	*Optional.* Customized port of Surveillance Station (HTTPS). For Surveillance-login users only.	1 and onward
cameraNumber	<integer>	The total number of installed cameras. For Surveillance-login users only.	1 and onward
licenseNumber	<integer>	The total number of installed licenses. For Surveillance-login users only.	1 and onward
maxCameraSupport	<integer>	Maximum number of camera support for this DS. For Surveillance-login users only.	1 and onward

<Version Object> definition:

Name	Value	Description	Availability
major	<integer>	Major version of Surveillance Station.	1 and onward
minor	<integer>	Minor version of Surveillance Station.	1 and onward
build	<integer>	Build number of Surveillance Station.	1 and onward

Example:

```
{
        "version": {
                "major": 6,
                "minor": 0,
                "build": 2250
        },
        "path": "/webman/3rdparty/SurveillanceStation",
        "customizedPortHttp": 9900,
        "customizedPortHttps": 9901,
        "cameraNumber": 20,
        "licenseNumber": 30,
        "maxCameraSupport": 40
}
```

2.3.3.2 API Error Code

Code	Description
400	Execution failed.

2.3.4 SYNO.SurveillanceStation.Camera

This API provides a set of methods to acquire camera-related information and to enable/disable cameras.

API Name	Version	Availability
SYNO.SurveillanceStation.Camera	1	Surveillance Station 6.0-2337
	2	Surveillance Station 6.1

Method Name	Section	Availability
List	2.3.4.1	1 and onward
GetInfo	2.3.4.2	1 and onward
GetCapability	2.3.4.3	1 and onward
ListGroup	2.3.4.4	1 and onward
GetSnapshot	2.3.4.5	1 and onward
GetCapabilityByCamId		4 and onward

2.3.4.1 List method

Get the list of all cameras.

Request

Parameter	Value	Description	Availability
offset	<integer>	*Optional*. The offset to be shifted in the total result. If not specified, the offset will be 0.	1 and onward
limit	<integer>	*Optional*. Number of cameras to be returned. If not specified, return cameras to the end of camera list.	1 and onward
additional	<string>	*Optional*. The list of <Additional Information> to be queried concatenated by “,”. If not specified, there will be no additional information.	1 and onward

<Additional Information> definition:

Parameter	Value	Description	Availability
device	<string>	To get <Device Setting Object>.	1 and onward
video	<string>	To get <Video Setting Object>.	1 and onward
record	<string>	To get <Record Setting Object>.	1 and onward
schedule	<string>	To get <Schedule Setting Object>.	1 and onward
advanced	<string>	To get <Advanced Setting Object>.	1 and onward

Example: List 2 cameras starting from offset 10 with all additional information.

```
GET /webapi/<URL_PATH>?
api=SYNO.SurveillanceStation.Camera&method=List&version=1
```

```
&offset=10&limit=2&additional=device,video,record,schedule,advanced
```

Response

Key	Value	Description	Availability
total	<integer>	The number of total installed cameras.	1 and onward
offset	<integer>	The shifted offset in the total result.	1 and onward
cameras	Array of <Camera Object>	The list of queried cameras.	1 and onward

<Camera Object> definition:

Key	Value	Description	Availability
id	<CAMERA_ID>	Unique camera ID.	1 and onward
name	<string>	Camera name.	1 and onward
host	<string>	Host name or IP address of camera.	1 and onward
status	0, 1, 2, 3, 4, 5, 6, 7	Current camera status. • 0: Normal • 1: Setting • 2: Disconnected • 3: Unauthorized • 4: Resolution error • 5: Disabled • 6: Unknown • 7: Deleted – video archives remained	1 and onward
recStatus	0, 1, 2, 3, 4, 5, 6	Current recording status. • 0: Not recording now • 1: Continuous Recording • 2: Motion Detection Recording • 3: Alarm Recording • 4: Motion Detection and Alarm Recording • 5: Manual Recording • 6: External Recording	2 and onward
enabled	<boolean>	Camera is enabled or not.	1 and onward
additional	The object which may contain some of the following objects: <Device Setting Object>, <Video Setting Object>, <Record Setting Object>, <Schedule Setting Object>, <Advanced Setting Object>	*Optional*. The additional information corresponding to the user request.	1 and onward

<Device Setting Object> definition:

Key	Value	Description	Availability
httpPort	1 … 65535	HTTP port of camera.	1 and onward
vendor	<string>	Camera vendor.	1 and onward
model	<string>	Camera model.	1 and onward
channel	1 …	*Optional*. The channel number of camera or video server. It show only when the camera support multi-	1 and onward

		channel.	
videoPath	<string>	*Optional*. The user-defined video source path. It shows only when the video source path is user-defined.	1 and onward

<Video Setting Object> definition:

Key	Value	Description	Availability
videoCodec	MJPEG, MPEG4, H264, MXPEG	Video codec.	1 and onward
streamingType	HTTP, RTSP	Streaming protocol.	1 and onward
audioCodec	DISABLED, PCM , G711, G726, AAC, AMR	*Optional*. Audio codec. It shows when the camera within the videoCodec supports audio functionality.	1 and onward
recQuality	1 … 5	Image quality of recording stream.	1 and onward
recResolution	<string>	Video resolution of recording stream.	1 and onward
recFps	1 … 30	Frames per second of recording stream.	1 and onward
liveQuality	1 … 5	Image quality of live view stream.	1 and onward
liveResolution	<string>	Video resolution of live view stream.	1 and onward
liveFps	1 … 30	Frames per second of live view stream.	1 and onward

<Record Setting Object> definition:

Key	Value	Description	Availability
recTime	5, 10, 20, 30, 40, 50, 60	Maximum recording time of an event (minutes).	1 and onward
preRecTime	5, 10, 15, 20 , 25, 30	Event pre-recording time (seconds).	1 and onward
postRecTime	5, 10, 15, 20 , 25, 30, 60, 90, 120	Event post-recording time (seconds).	1 and onward
eventFolder	<string>	The event archive folder name.	1 and onward
eventPrefix	<string>	The event file name prefix.	1 and onward
rotationDay	0 … 1825	Rotation by day. 0 stands for unlimited.	1 and onward
rotationSize	<integer>	Rotation by size. 0 stands for unlimited.	1 and onward

<Schedule Setting Object> definition:

Key	Value	Description	Availability
sun	<schedule string>	The recording schedule of Sunday.	1 and onward
mon	<schedule string>	The recording schedule of Monday.	1 and onward
tue	<schedule string>	The recording schedule of Tuesday.	1 and onward
wed	<schedule string>	The recording schedule of Wednesday.	1 and onward
thu	<schedule string>	The recording schedule of Thursday.	1 and onward
fri	<schedule string>	The recording schedule of Friday.	1 and onward
sat	<schedule string>	The recording schedule of Saturday.	1 and onward

<Advanced Setting Object> definition:

Key	Value	Description	Availability
liveSource	0, 1	Live View source of the camera. • 0: From camera	1 and onward

		• 1: From Surveillance Station	
mdSource	0, 1	Motion detection source of the camera. • 0: From camera • 1: From Surveillance Station	1 and onward
rotationLimitReachedAction	0, 1	Action to take when space or time limit is reached. • 0: Remove old archives • 1: Stop recording	1 and onward
hardwareInstallation	0, 1	*Optional*. The installation type of fisheye camera. • 0: Ceiling • 1: Wall Mount	1 and onward

Example:

```
{
    "total": 20,
    "offset": 10,
    "cameras": [
        {
            "id": 50,
            "name": "My AXIS",
            "host": "192.168.1.100",
            "enabled": true,
            "status": 0,
            "recStatus": 1,
            "additional": {
                "device": {
                    "httpPort": 80
                    "vendor": "AXIS",
                    "model": "M1054"
                },
                "video": {
                    "videoCodec": "H.264",
                    "streamingType": "RTSP",
                    "audioCodec": "G711",
                    "recQuality": 5,
                    "recResolution": "640x480",
                    "recFps": 10,
                    "liveQuality": 5,
                    "liveResolution": "640x480",
                    "liveFps": 10
                },
                "record": {
                    "recTime": 60,
                    "preRecTime": 5,
                    "postRecTime": 5,
                    "eventFolder": "My AXIS",
                    "eventPrefix": "My AXIS",
                    "rotationDay": 0,
                    "rotationSize": 10
                },
                "schedule": {
                    "sun": "000000000000000000000000000000000000000000000000",
                    "mon": "222222222222222222222222222222222222222222222222",
                    "tue": "111111111111111111111111111111111111111111111111",
                    "wed": "333333333333333333333333333333333333333333333333",
                    "thu": "111111111111111111111111111111111111111111111111",
                    "fri": "222222222222222222222222222222222222222222222222",
                    "sat": "000000000000000000000000000000000000000000000000"
                },
                "advanced": {
                    "liveSource": 1,
                    "mdSource": 0,
                    "rotationLimitReachedAction": 1
                }
            }
        },
        {
            "id": 51,
            "name": "My Vivotek Fisheye",
            "host": "192.168.1.101",
            "enabled": false,
            "status": 5,
            "recStatus": 0,
            "additional": {
                "device": {
                    "httpPort": 80,
                    "vendor": "Vivotek",
                    "model": "FE8171V"
                },
                "video": {
                    "videoCodec": "MJPEG",
                    "streamingType": "HTTP",
                    "recQuality": 5,
                    "recResolution": "1280x1280",
                    "recFps": 5,
                    "liveQuality": 5,
                    "liveResolution": "1280x1280",
                    "liveFps": 5
                },
                "record": {
                    "recTime": 60,
                    "preRecTime": 5,
                    "postRecTime": 5,
                    "eventFolder": "My Vivotek Fisheye",
                    "eventPrefix": "My Vivotek Fisheye",
                    "rotationDay": 0,
                    "rotationSize": 10
                },
                "schedule": {
                    "sun": "111111111111111111111111111111111111111111111111",
                    "mon": "222222222222222222222222222222222222222222222222",
                    "tue": "111111111111111111111111111111111111111111111111",
                    "wed": "333333333333333333333333333333333333333333333333",
                    "thu": "111111111111111111111111111111111111111111111111",
                    "fri": "111111111111111111111111111111111111111111111111",
                    "sat": "111111111111111111111111111111111111111111111111"
                },
                "advanced": {
                    "liveSource": 1,
                    "mdSource": 0,
                    "rotationLimitReachedAction": 1,
                    "hardwareInstallation ": 0
                }
            }
        }
    ]
}
```

2.3.4.2 GetInfo method

Get specific camera settings.

Request

Parameter	Value	Description	Availability
cameraIds	<string>	The list of <CAMERA_ID> to be queried concatenated by “,”.	1 and onward
additional	<string>	*Optional*. The list of <Additional Information> to be queried, concatenated by “,”. If not specified, there will be no additional information.	1 and onward

Example: Get camera information with “device” information.

```
GET /webapi/<URL_PATH>?
api=SYNO.SurveillanceStation.Camera&method=GetInfo&version=1&cameraIds=50&additional=device
```

Response

Key	Value	Description	Availability
cameras	Array of <Camera Object>	The list of all queried cameras.	1 and onward

Example: Get camera information response.

```
{
    "cameras": [
        {
            "id": 50,
            "name": "My AXIS",
            "host": "192.168.1.100",
            "enabled": true,
            "status": 0,
            "recStatus": 1,
            "additional": {
                    "device": {
                            "httpPort": 80,
                            "vendor": "AXIS",
                            "model": "M1054"
                    }
            }
        }
    ]
}
```

2.3.4.3 GetCapability method

Get capability of a specific camera model.

Request

Parameter	Value	Description	Availability
vendor	<string>	Name of the camera vendor.	1 and onward
model	<string>	Name of the camera model.	1 and onward

Example: Get capability of AXIS P5532 camera.

```
GET /webapi/<URL_PATH>?
api=SYNO.SurveillanceStation.Camera&method=GetCapability&version=1&vendor=AXIS&model=P5532
```

Response

Key	Value	Description	Availability
ptzPan	<boolean>	Capability to perform pan action.	1 and onward
ptzTilt	<boolean>	Capability to perform tilt action.	1 and onward
ptzZoom	<boolean>	Capability to perform zoom action.	1 and onward
ptzHome	<boolean>	Capability to move to home position.	1 and onward
ptzPresetNumber	<integer>	The maximum number of preset supported by the model. 0 stands for preset incapability.	1 and onward

Example:

```
{
        "ptzPan": true,
        "ptzTilt": true,
        "ptzZoom": false,
        "ptzHome": true,
        "ptzPresetNumber": 32
}
```

2.3.4.4 ListGroup method

Get all camera group information.

Request

Parameter	Value	Description	Availability
offset	<integer>	*Optional*. The offset to be shifted in the total result. If not specified, the offset will be 0.	1 and onward
limit	<integer>	*Optional*. Number of camera groups to be returned. If not specified, return camera groups to the end of camera group list.	1 and onward

Example:

```
GET /webapi/<URL_PATH>?
api=SYNO.SurveillanceStation.Camera&method=ListGroup&version=1&offset=10&limit=3
```

Response

Key	Value	Description	Availability
total	<integer>	The number of total camera groups.	1 and onward
offset	<integer>	The shifted offset in the total result.	1 and onward
cameraGroups	Array of <Camera Group Object>	The camera group list.	1 and onward

<Camera Group Object> definition:

Key	Value	Description	Availability
id	<CAMERA_GROUP_ID>	Unique camera group ID.	1 and onward
name	<string>	Camera group name.	1 and onward
cameraIds	Array of <CAMERA_ID>	The list of camera ID within this group.	1 and onward
description	<string>	The description of the camera group.	1 and onward

Example:

```
{
        "total": 38,
        "offset": 10,
        "cameraGroups": [
                {
                        "id": 1,
                        "name": "Group 1F",
                        "cameraIds": [10,11,12],
                        "description": "The camera group of 1F cameras"
                },
                {
                        "id": 2,
                        "name": "Group 2F",
                "cameraIds": [13,14,15],
                "description": "The camera group of 2F cameras"
            },
            {
                "id": 3,
                "name": "Group 3F",
                "cameraIds": [16,17],
                "description": "The camera group of 3F cameras"
            }
        ]
}
```

2.3.4.5 GetSnapshot method

Get the up-to-date snapshot of the selected camera in JPEG format.

Request

Parameter	Value	Description	Availability
cameraId	<CAMERA_ID>	Unique camera ID.	1 and onward

Example:

```
GET /webapi/<URL_PATH>?
api=SYNO.SurveillanceStation.Camera&method=GetSnapshot&version=1&cameraId=10
```

Response

The binary JPEG image data.

Example: HTTP response with image data

```
HTTP/1.0 200 OK
Content-Type: image/jpeg

<Binary JPEG image data>
```

2.3.4.6 API Error Code

Code	Description
400	Execution failed.
401	Parameter invalid.
402	Camera disabled.
403	The method is for Manager only.

2.3.5 SYNO.SurveillanceStation.PTZ

This API provides a set of methods to execute PTZ action, and to acquire PTZ related information such as patrol list or patrol schedule of a camera.

API Name	Version	Availability
SYNO.SurveillanceStation.PTZ	1	Surveillance Station 6.0-2337
	2	Surveillance Station 6.1

Method Name	Section	Availability
Move	2.3.5.1	1 and onward
Zoom	2.3.5.2	1 and onward
ListPreset	2.3.5.3	1 and onward
GoPreset	2.3.5.4	1 and onward
ListPatrol	2.3.5.5	1 and onward
RunPatrol	2.3.5.6	2 and onward
GetPatrolSchedule	2.3.5.7	1 and onward

2.3.5.1 Move method

Control the PTZ camera to move its lens.

Request

Parameter	Value	Description	Availability
cameraId	<CAMERA_ID>	Unique camera ID.	1 and onward
direction	up, down, left, right, home	Direction to move. • up: Move lens up. • down: Move lens down. • left: Move lens left. • right: Move lens right. • home: Move lens to home position.	1 and onward

Example: Move lens of a camera to left.

```
GET /webapi/<URL_PATH>?
api=SYNO.SurveillanceStation.PTZ&method=Move&version=1&cameraId=10&direction=left
```

Response

This method has no specific response data. It returns an empty success response if it completes without error.

2.3.5.2 Zoom method

Control the PTZ camera to zoom in or zoom out.

Request

Parameter	Value	Description	Availability
cameraId	<CAMERA_ID>	Unique camera ID.	1 and onward
control	in, out	Zoom control. in: Make camera to zoom in. out: Make camera to zoom out.	1 and onward

Example: Control a camera to do zoom in.

```
GET /webapi/<URL_PATH>?
api=SYNO.SurveillanceStation.PTZ&method=Zoom&version=1&cameraId=10&control=in
```

Response

This method has no specific response data. It returns an empty success response if it completes without error.

2.3.5.3 ListPreset method

List all presets of the PTZ camera.

Request

Parameter	Value	Description	Availability
cameraId	<CAMERA_ID>	Unique camera ID.	1 and onward
offset	<integer>	*Optional*. The offset to be shifted in the total result. If not specified, the offset will be 0.	1 and onward
limit	<integer>	*Optional*. Number of presets to be returned. If not specified, return presets to the end of preset list.	1 and onward

Example:

```
GET /webapi/<URL_PATH>?
api=SYNO.SurveillanceStation.PTZ&method=ListPreset&version=1&offset=10&limit=3&cameraId=10
```

Response

Key	Value	Description	Availability
total	<integer>	The number of total presets.	1 and onward
offset	<integer>	The shifted offset in the total result.	1 and onward
presets	Array of <Preset Object>	The preset list of the target camera.	1 and onward

<Preset Object> definition:

Key	Value	Description	Availability
id	<PRESET_ID>	Unique preset ID.	1 and onward
name	<string>	Preset name.	1 and onward

Example:

```
{
        “total”: 38,
        “offset”: 10,
        “presets”: [
                {
                        "id": 100,
                        "name": “My Preset 1”
                },
                {
                        "id": 101,
                        "name": “My Preset 2”
                },
                {
                        "id": 102,
                        "name": “My Preset 3”
                },
        ]
}
```

2.3.5.4 GoPreset method

Move the camera lens to a pre-defined preset position.

Request

Parameter	Value	Description	Availability
cameraId	<CAMERA_ID>	Unique camera ID.	1 and onward
presetId	<PRESET_ID>	Unique preset ID.	1 and onward

Example:

```
GET /webapi/<URL_PATH>?
api=SYNO.SurveillanceStation.PTZ&method=GoPreset&version=1&cameraId=10&presetId=10
```

Response

This method has no specific response data. It returns an empty success response if it completes without error.

2.3.5.5 ListPatrol method

Enumerate the patrol list of a PTZ camera.

Request

Parameter	Value	Description	Availability
cameraId	<CAMERA_ID>	Unique camera ID.	1 and onward
offset	<integer>	*Optional*. The offset to be shifted in the total result. If not specified, the offset will be 0.	1 and onward
limit	<integer>	*Optional*. Number of patrols to be returned. If not specified, list patrols to the end of patrol list.	1 and onward

Example:

```
GET /webapi/<URL_PATH>?
api=SYNO.SurveillanceStation.PTZ&method=ListPatrol&version=1&offset=10&limit=2&cameraId=10
```

Response

Key	Value	Description	Availability
total	<integer>	The number of total patrols.	1 and onward
offset	<integer>	The shifted offset in the total result.	1 and onward
patrols	Array of <Patrol Object>	The patrol list of the target camera.	1 and onward

<Patrol Object> definition:

Key	Value	Description	Availability
id	<PATROL_ID>	Unique patrol ID.	1 and onward
name	<string>	Patrol name.	1 and onward
stayTime	5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60	The stay time of one preset position (seconds).	1 and onward
sequence	Array of <PRESET_ID>	The preset execution sequence list.	1 and onward

Example:

```
{
        "total": 38,
        "offset": 10,
        “patrols”: [
                {
                        "id": 10,
                        "name": “My Patrol 1”,
                        "stayTime": 5,
                        "sequence": [100, 101, 102]
                },
                {
                    "id": 11,
                    "name": “My Patrol 2”,
                    "stayTime": 60,
                    "sequence": [103, 104]
                }
        ]
}
```

2.3.5.6 RunPatrol method

Force the camera to execute the specific patrol.

Request

Parameter	Value	Description	Availability
cameraId	<CAMERA_ID>	Unique camera ID.	2 and onward
patrolId	<PATROL_ID>	Unique patrol ID.	2 and onward

Example:

```
GET /webapi/<URL_PATH>?
api=SYNO.SurveillanceStation.PTZ&method=RunPatrol&version=2&cameraId=10&patrolId=11
```

Response

This method has no specific response data. It returns an empty success response if it completes without error.

2.3.5.7 GetPatrolSchedule method

Get the patrol schedule table of the PTZ camera.

Request

Parameter	Value	Description	Availability
cameraId	<CAMERA_ID>	Unique camera ID.	1 and onward

Example:

```
GET /webapi/<URL_PATH>?
api=SYNO.SurveillanceStation.PTZ&method=GetPatrolSchedule&version=1&cameraId=10
```

Response

Key	Value	Description	Availability
interval	5, 10, 15, 20, 25, 30, 40, 50, 60, 90, 120	The interval of each patrol (minutes).	1 and onward

cycle	0 … 10	• 0: Repeat infinitely. • >0: Repeat times of each patrol.	1 and onward
schedule	<Patrol Schedule Object>	The patrol schedule of the camera.	1 and onward

<Patrol Schedule Object> definition:

Key	Value	Description	Availability
sun	Array of <Patrol Schedule Unit Object>	48 units of schedule info of Sunday. Each unit to stand for half-hour.	1 and onward
mon	Array of <Patrol Schedule Unit Object>	48 units of schedule info of Monday.	1 and onward
tue	Array of <Patrol Schedule Unit Object>	48 units of schedule info of Tuesday.	1 and onward
wed	Array of <Patrol Schedule Unit Object>	48 units of schedule info of Wednesday.	1 and onward
thu	Array of <Patrol Schedule Unit Object>	48 units of schedule info of Thursday.	1 and onward
fri	Array of <Patrol Schedule Unit Object>	48 units of schedule info of Friday.	1 and onward
sat	Array of <Patrol Schedule Unit Object>	48 units of schedule info of Saturday.	1 and onward

<Patrol Schedule Unit Object> definition:

Key	Value	Description	Availability
type	0, 1, 2, 3, 4	Scheduled recording type. • 0: No scheduled plan. • 1: Continuous Recording • 2: Motion Detection Recording • 3: Alarm Recording • 4: Motion Detection and Alarm Recording	1 and onward
patrolId	0, <PATROL_ID>	• 0: No scheduled plan. • <PATROL_ID>: Unique patrol ID.	1 and onward

Example:

```
{
        "interval": 2,
        "cycle": 2,
        "schedule": {
                “sun”: [
                        {"type": 2, patrolId: 10}, {"type": 2, patrolId: 10}, {"type": 2, patrolId: 10},
                        {"type": 2, patrolId: 10}, {"type": 2, patrolId: 10}, {"type": 2, patrolId: 10},
                        {"type": 2, patrolId: 10}, {"type": 2, patrolId: 10}, {"type": 2, patrolId: 10},
                        {"type": 2, patrolId: 10}, {"type": 2, patrolId: 11}, {"type": 2, patrolId: 11},
                        {"type": 2, patrolId: 11}, {"type": 2, patrolId: 11}, {"type": 2, patrolId: 11},
                        {"type": 2, patrolId: 11}, {"type": 2, patrolId: 11}, {"type": 2, patrolId: 11},
                        {"type": 2, patrolId: 11}, {"type": 2, patrolId: 11}, {"type": 2, patrolId: 11},
                        {"type": 2, patrolId: 10}, {"type": 2, patrolId: 10}, {"type": 2, patrolId: 10},
                        {"type": 2, patrolId: 10}, {"type": 2, patrolId: 10}, {"type": 2, patrolId: 10},
                        {"type": 2, patrolId: 10}, {"type": 2, patrolId: 10}, {"type": 2, patrolId: 10},
                        {"type": 2, patrolId: 10}, {"type": 2, patrolId: 10}, {"type": 2, patrolId: 10},
            {"type": 1, patrolId: 10}, {"type": 1, patrolId: 10}, {"type": 1, patrolId: 10},
            {"type": 1, patrolId: 10}, {"type": 1, patrolId: 10}, {"type": 1, patrolId: 10},
            {"type": 1, patrolId: 10}, {"type": 1, patrolId: 10}, {"type": 1, patrolId: 10},
            {"type": 1, patrolId: 10}, {"type": 1, patrolId: 10}, {"type": 1, patrolId: 10},
            {"type": 0, patrolId: 0}, {"type": 0, patrolId: 0}, {"type": 0, patrolId: 0}
        ],
        “mon”: [...],
        “tue”: [...],
        “wed”: [...],
        “thu”:  [...],
        “fri”: [...],
        “sat”: [...]
    }
}
```

2.3.5.8 API Error Code

Code	Description
400	Execution failed.
401	Parameter invalid.
402	Camera disabled.

2.3.6 SYNO.SurveillanceStation.ExternalRecording

This API provides methods to start or stop external recording of a camera.

API Name	Version	Availability
SYNO.SurveillanceStation.ExternalRecording	1	Surveillance Station 6.0-2337
	2	Surveillance Station 6.1

Method Name	Section	Availability
Record	2.3.6.1	1 and onward

2.3.6.1 Record method

Start or stop external recording of a camera.

Request

Parameter	Value	Description	Availability
cameraId	<CAMERA_ID>	Unique camera ID.	1 and onward
action	start, stop	Start or stop external recording.	1 and onward

Example:

Start external recording of target camera.

```
GET /webapi/<URL_PATH>?
api=SYNO.SurveillanceStation.ExternalRecording&method=Record&version=1&cameraId=10&action=start
```

Stop external recording of target camera.

```
GET /webapi/<URL_PATH>?
api=SYNO.SurveillanceStation.ExternalRecording&method=Record&version=1&cameraId=10&action=stop
```

Response

Key	Value	Description	Availability
success	<boolean>	Whether starting/stopping recording is successful or not.	2 and onward

2.3.6.2 API Error Code

Code	Description
400	Execution failed.
401	Parameter invalid.
402	Camera disabled.

2.3.7 SYNO.SurveillanceStation.Event

This API provides method to query event information.

API Name	Version	Availability
SYNO.SurveillanceStation.Event	1	Surveillance Station 6.0-2337

Method Name	Section	Availability
Query	2.3.7.1	1 and onward

2.3.7.1 Query method

Query event list by specific filter conditions.

Request

Parameter	Value	Description	Availability
offset	<integer>	*Optional*. The offset to be shifted in the total result. If not specified, the offset will be 0.	1 and onward
limit	<integer>	*Optional*. Number of events to be returned. If not specified, return events to the end of event list.	1 and onward
mode	<string>	*Optional*. The list of recording mode to be queried concatenated by ",". • 0: Continuous Recording • 1: Motion Detection Recording • 2: Alarm Recording • 3: Manual Recording • 4: External Recording If not specified, this parameter will be ignored and get all related events.	1 and onward
locked	0, 1	*Optional*. The lock status of the events to be queried. • 0: No • 1: Yes If not specified, this parameter will be ignored and get all related events.	1 and onward
cameraIds	<string>	*Optional*. The list of <Camera ID> to be queried concatenated by ",". If not specified, this parameter will be ignored and get all related events.	1 and onward
fromTime	<timestamp>	*Optional*. Query start time.	1 and onward

		If not specified, this parameter will be ignored and get all related events.	
toTime	<timestamp>	*Optional*. Query stop time. If not specified, this parameter will be ignored and get all related events.	1 and onward

Example:

```
GET /webapi/<URL_PATH>?
api=SYNO.SurveillanceStation.Event&method=Query&version=1
&offset=10&limit=2&sourceType=1&sourceId=10&mode=1,2,3&locked=0&cameraIds=10,11,12&fromTime
=1330639810&toTime=1330643410
```

Response

Key	Value	Description	Availability
total	<integer>	The number of total events.	1 and onward
offset	<integer>	The shifted offset in the total result.	1 and onward
events	Array of <Event Object>	List of the queried events.	1 and onward

<Event Object> definition:

Key	Value	Description	Availability
id	<EVENT_ID>	Unique event ID.	1 and onward
cameraId	<CAMERA_ID>	Unique camera ID.	1 and onward
videoCodec	MJPEG, MPEG4, H264, MXPEG	Video codec.	1 and onward
audioCodec	<string>	*Optional*. Audio codec. It shows when the event contains audio.	1 and onward
eventSize	<float>	File size of the event (MB).	1 and onward
frameCount	<integer>	Total number of frames in the event.	1 and onward
startTime	<timestamp>	Event start time.	1 and onward
stopTime	<timestamp>	Event stop time.	1 and onward
status	0, 1, 2	The status of the event. • 0: Recorded • 1: Recording • 2: Locked	1 and onward
mode	0, 1, 2, 3, 4	The recording mode of the event. • 0: Continuous Recording • 1: Motion Detection Recording • 2: Alarm Recording • 3: Manual Recording • 4: External Recording	1 and onward

Example:

```
{
    "total": 1000,
    "offset": 10,
    "events": [
        {
            "id": 500,
            "cameraId": 1,
            "videoCodec": "MPEG4",
            "audioCodec": "MPEG4-GENERIC",
            "eventSize": 23.174,
            "frameCount": 937,
            "startTime": 1330640410,
            "stopTime": 1330641010,
            "status": 0,
            "mode": 0
        },
        {
            "id": 501,
            "cameraId": 2,
            "videoCodec": "MJPEG",
            "eventSize": 56.264,
            "frameCount": 1596,
            "startTime": 1330641610,
            "stopTime": 1330642210,
            "status": 1,
            "mode": 1
        }
    ]
}
```

2.3.8 SYNO.SurveillanceStation.Device

This API provides methods to get device information such as Visual Station and Slave DS.

API Name	Version	Availability
SYNO.SurveillanceStation.Device	1	Surveillance Station 6.0-2337

Method Name	Section	Availability
ListVS	2.3.8.1	1 and onward
ListCMS	2.3.8.2	1 and onward
GetServiceSetting	2.3.8.3	1 and onward

2.3.8.1 ListVS method

List all installed Visual Stations.

Request

Parameter	Value	Description	Availability
offset	<integer>	*Optional*. The offset to be shifted in the total result. If not specified, the offset will be 0.	1 and onward
limit	<integer>	*Optional*. Number of Visual Stations to be returned. If not specified, list Visual Stations to the end of Visual Station list.	1 and onward

Example:

```
GET /webapi/<URL_PATH>?
api=SYNO.SurveillanceStation.Device&method=ListVS&version=1&offset=10&limit=1
```

Response

Key	Value	Description	Availability
total	<integer>	The total number of installed Visual Stations.	1 and onward
offset	<integer>	The shifted offset in the total result.	1 and onward
visualStations	Array of <Visual Station Object>	The list of all Visual Stations.	1 and onward

<Visual Station Object> definition:

Key	Value	Description	Availability
id	<VS_ID>	Unique Visual Station ID.	1 and onward
name	<string>	Visual Station name.	1 and onward
dhcp	<boolean>	If DHCP is set to get IP.	1 and onward
ip	<string>	IP address.	1 and onward

mac	<string>	MAC address.	1 and onward
platform	VS60, VS80, VS240HD	Platform model.	1 and onward
version	<string>	Firmware version in the below format: [major].[minor]-[build_number] Example: 1.3-0308	1 and onward
enable	<boolean>	Enable status.	1 and onward
lock	<boolean>	Lock status.	1 and onward
status	0, 1, 2, 3, 4, 5	The current status of the VS. • 0: Normal • 1: Configuring • 2: Disconnected • 3: Reset • 4: Occupied • 5: Disabled	1 and onward

Example:

```
{
        "total": 38,
        "offset": 10,
        "visualStations": [
                {
                        "id": 922,
                        "name": "VS80",
                        "dhcp": true,
                        "ip": "192.168.38.91",
                        "macAddr": "00:11:32:AA:BB:CC",
                        "platform": "VS80",
                        "version": "1.3-0303",
                        "enable": true,
                        "lock": false,
                        "status": 0
                }
        ]
}
```

2.3.8.2 ListCMS method

List all installed Slave DSs.

Request

Parameter	Value	Description	Availability
offset	<integer>	*Optional*. The offset to be shifted in the total result. If not specified, the offset will be 0.	1 and onward
limit	<integer>	*Optional*.	1 and onward

		Number of Slave DSs to be returned. If not specified, list Slave DSs to the end of Slave DS list.	

Example:

```
GET /webapi/<URL_PATH>?
api=SYNO.SurveillanceStation.Device&method=ListCMS&version=1&offset=10&limit=2
```

Response

Key	Value	Description	Availability
total	<integer>	The total number of installed Slave DSs.	1 and onward
offset	<integer>	The shifted offset in the total result.	1 and onward
slaveDSs	Array of <Slave DS Object>	The Slave DS list.	1 and onward

<Slave DS Object> definition:

Key	Value	Description	Availability
id	<SLAVE_DS_ID>	Unique Slave DS ID.	1 and onward
name	<string>	Slave DS name.	1 and onward
ip	<string>	IP address or host name.	1 and onward
port	1 … 65535	HTTP port.	1 and onward
enable	<boolean>	Enable status.	1 and onward
status	0, 1, 2 3, 4	The current status of the VS. • 0: Normal • 1: Disabled • 2: Unauthorized • 3: Disconnected • 4: Error	1 and onward

Example:

```
{
	"total": 38,
	"offset": 10,
	"slaveDSs": [
		{
			"id": 1,
			"name": "My DS 1",
			"ip": "192.168.1.101",
			"port": 5000,
			"enable": true,
			"status": 0
		},
		{
			"id": 2,
			"name": "My DS 2",
			"ip": "192.168.1.102",
			"port": 5000,
                    "enable": false,
                    "status": 1
                }
        ]
}
```

2.3.8.3 GetServiceSetting method

Get the setting of Centralized Management and Visual Station service.

Request

No parameter is required.

Example:

```
GET /webapi/<URL_PATH>?
api=SYNO.SurveillanceStation.Device&method=GetServiceSetting&version=1
```

Response

Key	Value	Description	Availability
cmsEnabled	<boolean>	If Centralized Management is enabled.	1 and onward
cmsType	0, 1	*Optional*. Centralized Management type. It shows when cmsEnabled is true. • 0: Master • 1: Slave	1 and onward
vsEnabled	<boolean>	If Visual Station service is enabled.	1 and onward

Example:

```
{
        "cmsEnabled": true,
        "cmsType": 0,
        "vsEnabled": false
}
```

2.3.8.4 API Error Code

Code	Description
400	Execution failed.
410	Service is not enabled.

2.3.9 SYNO.SurveillanceStation.Emap

This API provides methods to get information about user-defined E-Maps.

Name	Version	Availability
SYNO.SurveillanceStation.Emap	1	Surveillance Station 6.1

Method Name	Section	Availability
List	2.3.9.1	1 and onward
GetInfo	2.3.9.2	1 and onward
GetImage	2.3.9.3	1 and onward

2.3.9.1 List method

Get the list of all E-Maps.

Request

Parameter	Value	Description	Availability
offset	<integer>	*Optional*. The offset to be shifted in the total result. If not specified, the offset will be 0.	1 and onward
limit	<integer>	*Optional*. Number of cameras to be returned. If not specified, return E-Maps to the end of E-Map list.	1 and onward

Example: List 2 E-Maps starting from offset 10.

```
GET /webapi/<URL_PATH>?
api=SYNO.SurveillanceStation.Emap&method=List&version=1&offset=10&limit=2
```

Response

Key	Value	Description	Availability
total	<integer>	The number of total E-maps.	1 and onward
offset	<integer>	The shifted offset in the total result.	1 and onward
emaps	Array of <E-Map Object>	The list of queried E-Maps.	1 and onward

<E-Map Object> definition:

Key	Value	Description	Availability
id	<EMAP_ID>	Unique E-Map ID.	1 and onward
name	<string>	E-Map name.	1 and onward
imageFormat	BMP, JPG, JPEG, GIF, PNG	Image format of the E-Map image.	1 and onward
imageWidth	<integer>	Image width of the E-Map image.	1 and onward
imageHeight	<integer>	Image height of the E-Map image.	1 and onward
items	Array of <E-Map Item Object>	The list of items marked on the E-Map.	1 and onward

<E-Map Item Object> definition:

Key	Value	Description	Availability
type	0, 1	Type of the E-Map item. • 0: Camera • 1: E-Map	1 and onward
dsId	0, <SLAVE_DS_ID>	Unique DS ID which this E-Map item belongs to. 0: On local host. <SLAVE_DS_ID>: ID of a Slave DS.	1 and onward
itemId	<integer>	Unique ID of this item. When this item is a camera, it means <CAMERA_ID>. When this item is an E-Map, it means <EMAP_ID>.	1 and onward
itemName	<string>	Name of this item. When this item is a camera, it describes camera name. When this item is an E-Map, it describes E-Map name.	1 and onward
xCoordinate	<integer>	X-coordinate of the E-Map item assuming that top-left of the E-Map is the origin.	1 and onward
yCoordinate	<integer>	Y-coordinate of the E-Map item assuming that top-left of the E-Map is the origin.	1 and onward
direction	0, 1, 2, 3, 4, 5, 6, 7,	Direction of the camera. If the type of this item is E-Map, direction will be always 0. • 0: Westward • 1: Northwestward • 2: Northward • 3: Northeastward • 4: Eastward • 5: Southeastward • 6: Southward • 7: Southwestward	1 and onward

Example:

```
{
        "total": 20,
        "offset": 10,
        "emaps": [
                {
                        "id": 30,
                        "name": "Lobby",
                        "imageFormat": “PNG”,
                        "imageWidth": 640,
                        "imageHeight": 480,
                        "items": [
                                {
                                        "type": 0,
                                        "dsId": 0,
                                        "itemId": 50,
                                        "itemName": "M1054",
                                        "xCoordinate": 10,
                                        "yCoordinate": 50,
                                        "direction": 5
                                },
                                {
                                        "type": 0,
                                        "dsId": 0,
                                        "itemId": 51,
                                        "itemName": "M1011",
                                        "xCoordinate": 10,
                                        "yCoordinate": 400,
                                        "direction": 3
                                },
                                {
                                        "type": 1,
                                        "dsId": 0,
                                        "itemId": 31,
                                        "itemName": "2F",
                                        "xCoordinate": 400,
                                        "yCoordinate": 400,
                                        "direction": 0
                                }
                        ]
                },
                {
                        "id": 31,
                        "name": "2F",
                        "imageFormat": “JPG”,
                        "imageWidth": 480,
                        "imageHeight": 480,
                        "items": [
                                {
                                        "type": 0,
                                        "dsId": 0,
                                        "itemId": 52,
                                        "itemName": "M1054",
                                        "xCoordinate": 20,
                                        "yCoordinate": 240,
                                        "direction": 4
                                },
                                {
                                        "type": 0,
                                        "dsId": 1,
                                        "itemId": 5,
                                        "itemName": "P1346",
                                        "xCoordinate": 400,
                                        "yCoordinate": 25,
                                        "direction": 6
                                }
                        ]
                }
        ]
}
```

2.3.9.2 GetInfo method

Get specific E-Map settings.

Request

Parameter	Value	Description	Availability
emapIds	<string>	The list of <EMAP_ID> to be queried concatenated by “,”.	1 and onward

Example: Get E-Map information with ID 31.

```
GET /webapi/<URL_PATH>?
api=SYNO.SurveillanceStation.Emap&method=GetInfo&version=1&emapIds=31
```

Response

Key	Value	Description	Availability
emaps	Array of <E-Map Object>	The list of all queried E-Maps.	1 and onward

Example:

```
{
        "emaps": [
                {
                        "id": 31,
                        "name": "2F",
                        "imageFormat": “JPG”,
                        "imageWidth": 480,
                        "imageHeight": 480,
                        "items": [
                                {
                                        "type": 0,
                                        "dsId": 0,
                                        "itemId": 52,
                                        "itemName": "M1054",
                                        "xCoordinate": 20,
                                        "yCoordinate": 240,
                                        "direction": 4
                                },
                                {
                                        "type": 0,
                                        "dsId": 1,
                                        "itemId": 5,
                                        "itemName": "P1346",
                                        "xCoordinate": 400,
                                        "yCoordinate": 25,
                                        "direction": 6
                                }
                        ]
                }
        ]
}
```

2.3.9.3 GetImage method

Get the image of the selected E-Map in its original format.

Request

Parameter	Value	Description	Availability
emapId	<EMAP_ID>	Unique E-Map ID.	1 and onward

Example:

```
GET /webapi/<URL_PATH>?
api=SYNO.SurveillanceStation.Emap&method=GetImage&version=1&emapId=10
```

Response

The binary image data. Its MIME type depends on E-map's original image format.

Example:

```
HTTP/1.0 200 OK
Content-Type: image/jpeg

<Binary JPEG image data>
```

2.3.9.4 API Error Code

Code	Description
400	Execution failed.
401	Parameter invalid.

2.3.10 SYNO.SurveillanceStation.Streaming

This API provides methods to get Live View or Event video stream.

Name	Version	Availability
SYNO.SurveillanceStation.Streaming	1	Surveillance Station 6.0-2337

Method Name	Section	Availability
LiveStream	2.3.10.1	1 and onward
EventStream	2.3.10.2	1 and onward

2.3.10.1 LiveStream method

Get a HTTP Live View video stream of the camera with given ID.

Request

Parameter	Value	Description	Availability
cameraId	<CAMERA_ID>	Unique camera ID.	1 and onward

Example:

```
GET /webapi/<URL_PATH>?
api=SYNO.SurveillanceStation.Streaming&method=LiveStream&version=1&cameraId=10
```

Response

Server will deliver a server push multipart image stream in response. Some HTTP headers may be inserted between the boundary string and the data chunk, as listed below:

Header	Description
Content-Type	The possible values are listed as follows: ● “image/JPEG” – JPEG image data part ● “image/MP4V-ES” – MPEG4 image data part ● “image/H264” – H264 image data part ● “audio/AC3” – AAC audio data part ● “audio/L16” – PCM audio data part ● “audio/PCMU” – G.711 u-law audio data part ● “audio/PCMA” – G.711 a-law audio data part ● “audio/G726” – G.726 audio data part ● “audio/AMR” – AMR audio data part
Content-Length	The data length of the data chunk
Vdo-ExtraSize:<Size> <Video extra data>	(MPEG4/H.264 only) The video extra data for decode. Note that there are <Size> bytes of <Video extra data> followed by this header
Ado-ExtraSize:<Size> <Audio extra data>	(MPEG4/H.264 only) The audio extra data for decode. Note that there are <Size> bytes of <Audio extra data> followed by this header
Timestamp	(MPEG4/H.264 only) The timestamp of the video frame

Codec	(MPEG4/H.264 only) The video and audio codec of the current stream

Example 1: MJPEG Live View stream

```
HTTP/1.1 200 OK
Date: Fri, 25 May 2012 05:59:39 GMT
Server: Apache/2.2.22 (Unix)
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Transfer-Encoding: chunked
Content-Type: multipart/x-mixed-replace;boundary=myboundary

--myboundary
Content-Type: image/jpeg
Content-Length: 46414

<JPEG image data>
--myboundary
Content-Type: image/jpeg
Content-Length: 59831

<JPEG image data>
--myboundary
Content-Type: image/jpeg
Content-Length: 36914

<JPEG image data>
...
```

Example 2: MPEG4 Live View stream

```
HTTP/1.1 200 OK
Date: Sat, 03 Mar 2012 07:18:45 GMT
Server: Apache/2.2.22 (Unix) mod_ssl/2.2.22 OpenSSL/1.0.0g
Connection: close
Content-Type: multipart/x-mixed-replace;boundary=myboundary

--myboundary
Vdo-ExtraSize:31
<MPEG4 video extra data>
Ado-ExtraSize:7
<MPEG4 audio extra data>
Timestamp: 1330759125734514
Codec:MP4V-ES;L16
Content-Type: image/MP4V-ES
Content-Length: 2349

<MPEG4 video data>
--myboundary
Vdo-ExtraSize:31
<MPEG4 video extra data>
```

```
Ado-ExtraSize:7
<MPEG4 audio extra data>
Timestamp:1330958772513972
Codec:MP4V-ES;L16
Content-Type: audio/L16
Content-Length: 1024

<MPEG4 audio data>
…
```

2.3.10.2 EventStream method

Get HTTP video stream of the specific recording event.

Request

Parameter	Value	Description	Availability
eventId	<EVENT_ID>	Unique event ID. Event ID could be obtained by SYNO.SurveillanceStation.Event API (section 2.3.7).	1 and onward

Note that you can insert the "Range" header within request messages to indicate the particular range of the video file. For example:

Range: bytes=0-9999999

This means to request the data between 0 and 9999999 byte of the event file.

Example: The request header to get the whole event **.**

```
GET /webapi/<URL_PATH>?
api=SYNO.SurveillanceStation.Streaming&method=EventStream&version=1
&eventId=200
User-Agent: My Media Player
Range: bytes=0-
Icy-MetaData: 1
```

Response

Server will deliver a 206 Partial Content response containing the following headers:

Header	Description
Accept-Ranges	Let clients know that server accept byte range request.
Content-Length	The data length of the data chunk actually returned.
Content-Range	The requested range of byte for the data in the body.
Content-Type	The possible values are shown as follows: ● "video/avi" – MJPEG format event ● "video/mp4" – MPEG4/H.264 format event

Example: MPEG4 Event stream**.**

```
HTTP/1.1 206 Partial Content
Date: Mon, 05 Mar 2012 15:50:34 GMT
Server: Apache/2.2.22 (Unix) mod_ssl/2.2.22 OpenSSL/1.0.0g
Accept-Ranges: bytes
Content-Length: 9298055
Content-Range: bytes 19244797-28542851/28542852
Connection: close
Content-Type: video/mp4

<MPEG4 event file data>
```

2.3.10.3 API Error Code

Code	Description
400	Execution failed.
401	Parameter invalid.
402	Camera disabled.
403	Insufficient license.

3. Resources

REST:

- http://en.wikipedia.org/wiki/Representational_state_transfer

SOAP:

- http://en.wikipedia.org/wiki/SOAP

JSON:

- http://en.wikipedia.org/wiki/JSON

HTTP:

- Hypertext Transfer Protocol – HTTP/1.0 http://www.w3.org/Protocols/rfc1945/rfc1945
- Hypertext Transfer Protocol – HTTP/1.1 http://www.w3.org/Protocols/rfc2616/rfc2616.html

Appendix: Valid values

The following valid values are used in this document:

Value	Description
m ... n	Any number between number m and number n.
m …	Any number larger than or equal to number m.
<boolean>	A Boolean data type.
<integer>	Any number between 0 and 4,294,967,295 (2^{32}-1).
<float>	A floating point number.
<string>	Any string encoded by UTF-8.
<timestamp>	Unix time, seconds elapsed since UTC of Thursday, Jan 1st, 1970.
<value 1>, <value 2>, <value 3>, ...	Enumeration, only the given values are valid.
Array of <object>	A list of object in JSON array format.
<schedule string>	A string consists of 48 digits to represent the scheduling of a weekday. Note that each digit stands for the schedule type of half-hour: 0: No scheduled plan 1: Continuous Recording 2: Motion Detection Recording 3: Alarm Recording 4: Motion Detection and Alarm Recording Example: “111111111111222222222222333333333333444444444444”
<XXX Object>	The particular JSON object created to carry particular information. For example: • <Camera Object> defined by “List” method in section 2.3.4 is used to represent the information of a camera. • <Event Object> defined by “Query” method in section 2.3.7 is used to represent the information of an event.
<CAMERA_ID>	A unique ID to identify an installed camera. The camera IDs can be obtained by calling “List” method of SYNO.SurveillanceStation.Camera (section 2.3.4).
<CAMERA_GROUP_ID>	A unique ID to identify an existing camera group. The IDs can be obtained by calling “ListGroup” method of SYNO.SurveillanceStation.Camera (section 2.3.4).
<PRESET_ID>	A unique ID to identify an existing preset position of a camera. The IDs can be obtained by calling “ListPreset” method of SYNO.SurveillanceStation.PTZ (section 2.3.5).
<PATROL_ID>	A unique ID to identify an existing patrol of a camera. The IDs can be obtained by calling “ListPatrol” method of SYNO.SurveillanceStation.PTZ (section 2.3.5).
<VS_ID>	A unique ID to identify an installed Visual Station. The IDs can be obtained by calling “ListVS” method of SYNO.SurveillanceStation.Device (section 2.3.8).
<SLAVE_DS_ID>	A unique ID to identify an installed Slave DS in CMS list. The IDs can be obtained by calling “ListCMS” method of SYNO.SurveillanceStation.Device (section 2.3.8).
<EVENT_ID>	A unique ID to identify a recorded event. The IDs can be obtained by calling “Query” method of SYNO.SurveillanceStation.Event (section 2.3.7).
<EMAP_ID>	A unique ID to identify an existing E-Map. The IDs can be obtained by calling “List” method of SYNO.SurveillanceStation.Emap (section 2.3.9).