REST API Reference

FortiRecorder 7.2.11

FORTINET DOCUMENT LIBRARY
https://docs.fortinet.com

FORTINET VIDEO LIBRARY
https://video.fortinet.com

FORTINET BLOG
https://blog.fortinet.com

CUSTOMER SERVICE & SUPPORT
https://support.fortinet.com

FORTINET TRAINING & CERTIFICATION PROGRAM
https://www.fortinet.com/training-certification

FORTINET TRAINING INSTITUTE
https://training.fortinet.com

FORTIGUARD LABS
https://www.fortiguard.com

END USER LICENSE AGREEMENT
https://www.fortinet.com/doc/legal/EULA.pdf

FEEDBACK
Email: techdoc@fortinet.com

June 09, 2026
FortiRecorder 7.2.11 REST API Reference
00-7211-0000000-20260609

TABLE OF CONTENTS

Change log **5**
Introduction **6**
Enable the REST API **7**
HTTP request methods and response codes **8**
Permissions 10
Filtering and paging 11
Expansion 12
Authentication **13**
Authentication for video clip service 15
CameraCamera **16**
Example: Apply the "HighResContinuous" profile to a camera 21
Example: Move a camera to home position 21
Example: Move a camera to position "PTZ_preset_1" 21
CameraCameraPtzPresets **22**
Example: Create "PTZ_preset_1" with the camera's current position 23
Example: Update "PTZ_preset_1" with the camera's current position 23
Example: Delete "PTZ_preset_1" 23
CameraEvent **24**
Type 25
Subtype 26
Motion 26
Tamper 26
Face detection 26
Object detection 26
Camera event 27
Recording 27
System event 27
State 28
Example: Get events after 1 AM on January 1, 2025 28
Example: Get events for cameras "cam1" and "cam2" during a specified time period 28
CameraProfile **29**
Recording type 32
Example: Change to high resolution for recordings 32
CameraSnapshot **33**
Example 33
CameraStatus **34**
Action 35
Flag 36
CameraVideoProfile **37**
Example: Modify the video frame rate 39

ScheduleObject ... **40**
Example: Change schedule time range to all day ... 42
SysGlobal ... **43**
Example: Get global settings for FortiRecorder ... 45
SysInterface ... **46**
Example: Get a list of network interfaces ... 48
Example: Allow only RTSP, FortiCentral, and HTTPS connections to port2 ... 48
SysStatusCommand ... **49**
SysStatusSysinfo ... **50**
SysStatusUsage ... **52**
Face_recognitionUser ... **54**
Example: Create a person for face recognition on FortiRecorder ... 55
EmployeeFaceRecord ... **56**
AiLog ... **57**
Example: Find times when unknown people were seen by camera MC51 ... 57
VideoClip ... **58**
Example: Download an MP4 video ... 59
Example: Download a JPG snapshot image ... 60
Streaming video ... **61**
URL schema ... 61
Authentication ... 61
RTSP commands ... 61
PLAY ... 62
Example: VLC video player ... 62
Example: Generic video player ... 64

Change log

The following is a list of documentation changes. For a list of software changes, see the FortiRecorder Release Notes.

Date	Change Description
2026-05-06	Initial release of the FortiRecorder 7.2.11 REST API Reference.

Introduction

This document provides information about the:

- REST API endpoints and methods
- how to use RTSP to play video streams with software such as VLC

for FortiRecorder 7.2.11.

For information on other features, see the FortiRecorder Administration Guide.

Enable the REST API

To use the REST API on FortiRecorder, it must be enabled and permissions granted for accounts that will use API access.

1. On FortiRecorder, either:
   - Use SSH to connect to the CLI.
   - Use a web browser to connect to the GUI, and then go to *Dashboard* > *Console*
2. Enter the following commands:
   ```
   config system global
      set rest-api enable
   end
   ```
3. Enable HTTP and/or HTTPS on port1 or other network interfaces which you want to allow to receive REST API requests.

   If you will use live video streams, also enable RTSP.
   ```
   config system interface
      edit port1
         set allowaccess https ssh rtsp
   end
   ```
4. Enable the REST API access mode for accounts that are allowed to use status and/or configuration URLs.
   ```
   config system admin
      edit YOUR_ACCOUNT
         set access gui cli rest
   end
   ```

   

   Currently the REST API requires accounts with local password authentication. Remote authentication is not supported.

   Accounts may not be allowed to use all of the URLs in the REST API, depending on permissions. Adjust them if needed. See administrator profiles in the FortiRecorder Administration Guide.
5. If you will use the REST API URLs for live video streams, or to download previously recorded video clips or snapshot images, enable those services.
   ```
   config service video-stream
      set status enable
      set camera-groups YOUR_CAMERA_GROUP
      set username YOUR_STREAM_USERNAME
      set password YOUR_SERVICE_PASSWORD
      set protocols https rtsp
   end
   config service video-clip
   set status enable
      set camera-groups YOUR_CAMERA_GROUP
      set password YOUR_SERVICE_PASSWORD
   end
   ```

HTTP request methods and response codes

The FortiRecorder REST API uses standard HTTP/HTTPS request methods and response status codes, including errors.

URLs are case sensitive. Valid methods vary by URL and permissions.

HTTP request methods

Request Method	Description
`GET`	Retrieve all resources or a specific resource
`POST`	Either: • Create a new resource • Perform an action
`PUT`	Update an existing resource
`DELETE`	Delete a resource

HTTP response status codes

Response Status Code	Description
`200 OK`	Success
`400 Bad Request`	Bad request For example, a wrong API URL may have this JSON in the response body: `{ "errorType": 3, "errorMsg": "Failed to retrieve object " }`
`403 Forbidden`	Either: • No permissions for the resource • Authentication token was missing from the request For example, a failed login attempt may have this JSON in the response body: `{ "errorType": 7, "errorMsg": "Failed: Access denied:Add object (AdminLogin) ", "reqAction": 2, "totalRemoteCount": 0, "collection": "[]"`

Response Status Code	Description
	}
404 Not Found	Resource does not exist
405 Method Not Allowed	Resource does not allow the HTTP request method that was used
500 Server Error	Internal error on FortiRecorder

JSON formatting is used for all HTTP request and response bodies unless otherwise mentioned. JSON attributes vary by URL and by the data on your FortiRecorder unless otherwise mentioned.

If the HTTP method does not agree with the JSON attribute `reqAction` in the request body, then **`reqAction` overrides the HTTP method**.

- `"reqAction": 1` is like HTTP `GET`
- `"reqAction": 2` is like HTTP `POST`
- `"reqAction": 3` is like HTTP `DELETE`
- `"reqAction": 5` is like HTTP `PUT`
- `"reqAction": 14` has no HTTP method equivalent; it moves the item

Permissions

Valid HTTP methods vary by URL and your permissions. This document mentions exceptions, if any. For example:

- `/api/v1/SysStatusSysinfo/` only supports `GET` and therefore only requires `read` permissions
- `/api/v1/SysGlobal/` supports both `GET` and `PUT` if the account has `read-write` permissions

Filtering and paging

If `"collection": []` is in the response JSON, then it is an enumeration of items. Collections can be large, and you might want to filter out some results, or limit the number of results in each response or get a specific range (also called "paging"). To do this, each URL in the REST API may support URL parameters, JSON attributes, or both.

For example, a `GET` request to the URL:

`https://HOST_OR_IP/api/v1/CameraStatus`

returns a list of **all** cameras, but a request with this URL parameter (shown in bold typeface):

`https://HOST_OR_IP/api/v1/CameraStatus`**`?showInactiveCamera=0`**

returns a filtered list of only **active** cameras, and:

`https://HOST_OR_IP/api/v1/CameraStatus`**`?showInactiveCamera=1`**

returns a list of only **inactive** cameras.

Other URLs let you use JSON attributes in the request to specify which item number to start with, and how many to include in the response:

```
{
    startIndex: 10,
    pageSize: 20
}
```

Responses indicate either that you have downloaded all items in the `collection`:

```
{
    "nextPage": false
}
```

or more items still exist, and how many of them were in the response:

```
{
    "nextPage": true,
    "totalRemoteCount": 100,
    "subCount": 20,
    "remoteSorting": true
}
```

To get more of the items in the `collection`, make another request with a different `startIndex`, depending on which page of data you want to get.

Expansion

Sometimes items in a `collection` have fewer details compared to a JSON response for each individual item. You can use the identifiers in the collection to get details about each item.

For example, a `GET` request to the URL:

`https://HOST_OR_IP/api/v1/CameraVideoProfile/`

responds with a collection of video profiles, where each `mkey` attribute's value is the name of a profile. To view details about a specific profile, put the `mkey` value on the end of the next `GET` request URL:

`https://HOST_OR_IP/api/v1/CameraVideoProfile/YOUR_VIDEO_PROFILE_NAME`

Use URL encoding if the name of an item on FortiRecorder contains non-ASCII or reserved characters such as spaces, `&`, `#`, `?`,`門` and ü.

Some REST API clients such as Postman or Insomnia encode URLs automatically. Examples with `curl` in this document, however, show the URL as you would manually type or paste it, *without* automatic encoding. For example, instead of a space in a URL, you might see:

`%20`

Authentication

Before you use other REST API URLs, you must make an HTTP `POST` request to log in at the authentication URL.

HTTP Methods	• `POST`
URL	`https://HOST_OR_IP/api/v1/AdminLogin/`

Request JSON	`{`
	`  "name": "YOUR_ACCOUNT",`
	`  "password": "YOUR_PASSWORD"`
	`}`

Use an account name that is allowed to use REST API access, and has permissions for the URLs that you will use. If the account does not have the required permissions, then it is indicated by the HTTP response code and body.

If the login succeeds, then the response contains a header with:

`Set-Cookie: APSCOOKIE_...`

Save and include this cookie value as a header in later REST API requests. This associates the requests with the existing authentication session.

Idle authentication sessions time out. See the idle timeout in the FortiRecorder Administration Guide. To continue using the REST API, authenticate again.

For example, to log in, you could open the command prompt on your computer and enter a `curl` command:

```
curl -H "Content-Type: application/json" -X POST -d "{\"name\":\"YOUR_
ACCOUNT\",\"password\":\"YOUR_PASSWORD\"}" https://HOST_OR_IP/api/v1/AdminLogin -c cookie.txt
```

and then subsequent commands would include (`-b`) and possibly also update (`-c`) the session cookie file:

```
curl -X GET -b cookie.txt -c cookie.txt https://HOST_OR_IP/api/v1/SysStatusSysinfo
```

Do not save passwords in unencrypted files, including the CLI history log.
Unencrypted credentials at rest are a security risk. If an attacker or virus compromises your computer, it could allow unauthorized persons to access your FortiRecorder system. Use a key management system, such as Vault in Postman, to securely store credentials.

On Microsoft Windows with Command Prompt, inside the JSON data, you must put a backslash before each double straight quote (\"). For example (highlighted in bold):
`-d "{\"name\":\"admin\"}"`

Alternatively, you can input a JSON stream from another command.

URLs with parameters may also require double quotes around them.

If you do not, then the command line may interpret each JSON attribute as CLI commands or arguments, resulting in various error messages depending on the sequential order of arguments and attributes.

Reserved characters and escape sequences vary by operating system and command line environment; Linux and Mac terminals often do not require this, and Microsoft PowerShell uses different escape sequences.

If you have either:

- private certificate authority (CA) servers
- FortiRecorder with a factory or self-signed certificate

then authentication of all HTTPS requests might fail with certificate errors.

Trust the FortiRecorder certificate or its signing CA. Do not use insecure (`-k`) connections as a workaround.

For example, you could use certutil on the Windows command line:

```
.\certutil.exe -addstore -f "Root" "C:\Users\YOUR_
USERNAME\Downloads\FORTIRECORDER_CA_CHAIN.pem"
```

and then use that root CA trust store with `curl --ca-native` commands.

Secure HTTPS requests *should* fail if the FortiRecorder X.509 certificate is:

- fake (name is wrong, or is not supported by the SNI extension)
- not valid at the current time/date
- not allowed for this usage
- not signed, directly or indirectly, by a CA that your computer trusts, such as for factory or self-signed certificates

See also how to use certificates in the FortiRecorder Administration Guide.

Authentication for video clip service

For the video clip and snapshot image download service, you must log in differently.

HTTP Methods	• POST
URL	`https://HOST_OR_IP/api/v1/ServiceLogin/`

Request JSON	`{`
	`"reqAction": 1,`
	`"name": service-clip",`
	`"password": "YOUR_SERVICE_PASSWORD"`
	`}`

Live video streams via RTSP also uses the same username and password, but uses HTTP digest authentication instead of JSON.

See also the video clip service in the FortiRecorder Administration Guide.

CameraCamera

View and change camera profiles and other settings, and make pan, tilt, and zoom (PTZ) movements.

Permissions required include *Camera configuration*.

HTTP Methods	• GET • PUT • POST • DELETE
URL	For HTTP GET, use either: • https://HOST_OR_IP/api/v1/CameraCamera/ • https://HOST_OR_IP/api/v1/CameraCamera/CAMERA_NAME/ For HTTPPOST, PUT, and DELETE, use: • https://HOST_OR_IP/api/v1/CameraCamera/CAMERA_NAME/ where variables are: • CAMERA_NAME — Unique identifier of a specific resource, such as the camera fd51_front_door

Request JSON	For HTTP GET or DELETE, leave the request body empty.
	For HTTP POST, configure all required settings. For a list of available settings, use HTTP GET.
	For HTTP PUT, configure only settings that you want to change. Omit others.
	For example, you can use PUT to change the profile (see also CameraProfile on page 29):
	`{ "profile": "HighResContinuous" }`
	or use POST to run PTZ commands to move the camera:
	`{ "reqAction": 7, "camMove": TYPE, "magnitude": SPEED, "presetName": PRESET }`
	where attributes often self-explanatory, except:
	• TYPE — Type of movement direction, zoom direction, or focus):

PTZ support varies by camera model.

- `0` — Move home. See also `11`.
- `1` — Move right.
- `2` — Move left.
- `3` — Move up.
- `4` — Move down.
- `5` — Move diagonally up and right.
- `6` — Move diagonally up and left.
- `7` — Move diagonally down and right.
- `8` — Move diagonally down and left.
- `9` — Zoom in.
- `10` — Zoom out.
- `11` — Set the current position as home.
- `12` — Focus in.
- `13` — Focus out.
- `14` — Focus semi-automatically.
- `15` — Focus automatically.
- `17` — Move to a position that is defined in the attribute `presetName`.

- `SPEED` — Speed of movement. Valid values are from 1 to 10.
- `PRESET` — Name of a PTZ preset. See also CameraCameraPtzPresets on page 22.
- `management_mode` — Whether or not the settings are managed via FortiCamera Cloud:
  - `true` — Managed by FortiCamera Cloud. FortiRecorder keeps a local cache of settings to push to cameras that are not cloud native, or for cameras that are cloud native but are in a hybrid deployment for another reason such as local storage.
  - `false` — Managed by FortiRecorder.

  Note: Before you change this setting, see the effects of cloud management on cameras and steps to disable it.

Response JSON

For HTTP `PUT` and `POST`, responses vary by which settings you changed.

For HTTP `GET`, responses vary by URL (`collection` vs. individual resource) and camera model. For example, if you get a FortiCam CD51 camera , it supports many settings:

```
                        {
                            "objectID": "CameraCamera:cd51",
                            "reqAction": 1,
                            "nodePermission": 3,
                            "mkey": "cd51",
                            "location": "",
                            "tags": "",
                            "floor": "",
                            "floorarea": "",
                            "position_x": "",
                            "position_y": "",
                            "face_recognition": false,
                            "face_recognition_schedule": "",
                            "face_recognition_processing_order": 0,
                            "camera_type": 1,
                            "vendor_name": "Fortinet",
                            "model_name": "FCM-CD51",
                            "status": false,
                            "wired_status": 1,
                            "wired_addr": "0.0.0.0",
                            "index": 6,
                            "light_detection": 1,
                            "wlan_status": 1,
                            "wlan_addr": "0.0.0.0",
                            "face_recognition_video_processing_decimation_factor": 0,
                            "area": [],
                            "fw_version": "",
                            "management_mode": 0,
                            "cloud_region": "",
                            "management": 0,
                            "username": "",
                            "password": "",
                            "comm_type": 1,
                            "comm_http_port": 0,
                            "comm_https_port": 443,
                            "push_config": true,
                            "login_password": "CAMERA_ADMIN_PASSWORD",
                            "operator_password": "CAMERA_VIEWER_PASSWORD",
                            "address_mode": 0,
                            "addr": "172.168.1.10",
```

```
                "port": 443,
                "wired_mode": 0,
                "wired_netmask": "0.0.0.0",
                "wired_gateway": "0.0.0.0",
                "wired_primary_dns": "0.0.0.0",
                "wired_secondary_dns": "0.0.0.0",
                "mac": "d4:76:a0:09:fa:2d",
                "lan_mac": "d4:76:a0:09:fa:2d",
                "wlan_mac": "d4:76:a0:09:fa:2e",
                "serial_number": "CDC51ATF20000004",
                "transport_type": 0,
                "transport_udp_port": 554,
                "transport_http_port": 0,
                "transport_https_port": 0,
                "transport_tcp_port": 0,
                "motion": [],
                "profile": "HighResContinuous",
                "view_angle": 1,
                "video_brightness": 50,
                "video_contrast": 50,
                "video_saturation": 50,
                "video_sharpness": 50,
                "power_frequency": 1,
                "video_aspect": 1,
                "recording_media_profile": "",
                "viewing_media_profile": "",
                "video_display_mode": 0,
                "mount": 0,
                "video_orientation": 0,
                "zoom": 0,
                "exposure_mode": 2,
                "exposure_wdr_digital": 0,
                "exposure_wdr_shutter": 0,
                "exposure_gain_max": 1,
                "exposure_time_max": 4,
                "image_dnr": 1,
                "image_dnr_level": 1,
                "image_dnr_smart": 0,
                "image_dis": false,
                "image_digital_zoom": false,
                "overlay_mode": 7,
                "overlay_date_format": 1,
                "alarm_pre": 10,
```

```
                                calibration_tilt": 0,
                                    "view_calibration_roll": 0,
                                    "wifi_status": false,
                                    "wifi_ssid": "",
                                    "wifi_wmm": false,
                                    "wifi_security": 0,
                                    "wifi_wep_auth": 0,
                                    "wifi_wep_keylength": 1,
                                    "wifi_wep_keyindex": 1,
                                    "wifi_wep_key": "",
                                    "wifi_wpa_encrypt": 1,
                                    "wifi_wpa_passphrase": "******",
                                    "wifi_wpa_protocol": 1,
                                    "wifi_wpa_username": "",
                                    "wifi_wpa_password": "******",
                                    "wlan_addr_mode": 0,
                                    "wlan_mask": "0.0.0.0",
                                    "wlan_gateway": "0.0.0.0",
                                    "wlan_primary_dns": "0.0.0.0",
                                    "wlan_secondary_dns": "0.0.0.0",
                                    "ready": false,
                                    "unready_state": 0,
                                    "unready_state_detail": "",
                                    "onvif_recording_stream_uris": "",
                                    "onvif_viewing_stream_uris": "",
                                    "onvif_snapshot_uri": "",
                                    "onvif_snapshot_port": 0,
                                    "onvif_recording_width": 0,
                                    "onvif_recording_height": 0,
                                    "onvif_viewing_width": 0,
                                    "onvif_viewing_height": 0,
                                    "move_home": true,
                                    "move_home_delay": 5,
                                    "move_home_pan": 0,
                                    "move_home_tilt": 0,
                                    "move_home_zoom": 0,
                                    "ptz_presets": [],
                                    "edge_storage_status": false,
                                    "edge_storage_network_failure": false,
                                    "websocket_tls": true,
                                    "area_type": 0,
                                    "area_x1": 0,
                                    "area_y1": 0,
                                    "area_x2": 0,
                                    "area_y2": 0,
                                    "area_x3": 0,
                                    "area_y3": 0,
                                    "area_x4": 0,
```

Example: Apply the "HighResContinuous" profile to a camera

```
curl -X PUT -H "Content-Type: application/json" -d "{\"profile\":\"HighResContinuous\"}" -b
cookie.txt https://HOST_OR_IP/api/v1/CameraCamera/CAMERA_NAME
```

Example: Move a camera to home position

```
curl -X POST -H "Content-Type: application/json" -d "
{\"reqAction\":7,\"camMove\":0,\"magnitude\":4}" -b cookie.txt https://HOST_OR_
IP/api/v1/CameraCamera/CAMERA_NAME
```

Example: Move a camera to position "PTZ_preset_1"

After Example: Create "PTZ_preset_1" with the camera's current position on page 23, you can enter:

```
curl -X POST -H "Content-Type: application/json" -d "
{\"reqAction\":7,\"camMove\":16,\"presetName\":\"PTZ_preset_1\"}" -b cookie.txt https://HOST_OR_
IP/api/v1/CameraCamera/CAMERA_NAME
```

CameraCameraPtzPresets

View or change camera pan, tilt, and zoom (PTZ) presets.

If you want the camera to return to the position that was defined in a PTZ preset, use it in a PTZ command. See Example: Move a camera to position "PTZ_preset_1" on page 21.

Permissions required include *Camera configuration*.

HTTP Methods	• `GET` • `PUT` • `POST` • `DELETE` To create a new PTZ preset, use PTZ commands (see CameraCamera on page 16) to position the camera, and then use `POST`. To update the PTZ preset, use PTZ commands to change the position, and then use `PUT`.
URL	For HTTP `GET`, use: • `https://HOST_OR_IP/api/v1/CameraCamera/CAMERA_NAME/CameraCameraPtzPresets/` For HTTP `POST`, `PUT`, and `DELETE`, use: • `https://HOST_OR_IP/api/v1/CameraCamera/CAMERA_NAME/CameraCameraPtzPresets/PRESET_NAME` where variables are: • `CAMERA_NAME` — Unique identifier of a specific resource, such as the camera `fd51_front_door` • `PRESET_NAME` — Unique identifier of a specific resource, such as the PTZ preset `PTZ_preset_1`
Request JSON	Leave empty.

Response JSON	Responses vary by HTTP method, URL (`collection` vs. individual resource), and camera model. For example, if you create a PTZ preset, you may receive a response like this:
	`{`
	`"objectID": "CameraCameraPtzPresets:PTZ_preset2{P:door_cam}",`
	`"reqAction": 2,`
	`"nodePermission": 3,`
	`"parent_mkey": "door_cam",`
	`"mkey": "PTZ_preset1",`
	`"pan": 0,`
	`"tilt": 0,`

```
                "zoom": 0
            }
```

Example: Create "PTZ_preset_1" with the camera's current position

```
curl -X POST -H "Content-Type:application/json" -d "{}" -b cookie.txt https://HOST_OR_
IP/api/v1/CameraCamera/CAMERA_NAME/CameraCameraPtzPresets/PTZ_preset_1
```

After you move the camera, you can return to the preset position by following Example: Move a camera to position "PTZ_preset_1" on page 21.

Example: Update "PTZ_preset_1" with the camera's current position

```
curl -X PUT -H "Content-Type:application/json" -d "{}" -b cookie.txt https://HOST_OR_
IP/api/v1/CameraCamera/CAMERA_NAME/CameraCameraPtzPresets/PTZ_preset_1
```

Example: Delete "PTZ_preset_1"

```
curl -X DELETE -b cookie.txt https://HOST_OR_IP/api/v1/CameraCamera/CAMERA_
NAME/CameraCameraPtzPresets/PTZ_preset_1
```

CameraEvent

Timeline of motion detection video recordings and annotations about related system events that may interrupt recordings, such as reboots. See also the FortiRecorder Log Reference.

Permissions required include *Camera notifications*.

HTTP Methods	• `GET`
URL	Either: • `https://HOST_OR_IP/api/v1/CameraEvent` • `https://HOST_OR_IP/api/v1/CameraEvent?device_name=CAMERA_NAMES&start_time=TIMESTAMP&end_time=TIMESTAMP&evt_filter=FILTER&startIndex=INDEX&pageSize=PAGE_SIZE` where variables (zero or more) are: • `CAMERA_NAMES` — Optional. Comma-separated unique identifiers of specific devices, such as the cameras named `fd51,cd55`. See also CameraCamera on page 16. • `TIMESTAMP` — Optional. Timestamp in the format `YYYY-MM-DD-HH-MM-SS` (year, month, day, hour, minute, second). • `INDEX` — Optional. See also Filtering and paging on page 11. • `PAGE_SIZE` — Optional. See also Filtering and paging on page 11. • `FILTER` — Optional. Event type. Each event has both `atype` and `subtype` JSON attribute, indicating whether the event is a motion detection, annotation, etc These can be used for `FILTER`. See Type on page 25 and Subtype on page 26.
Request JSON	Leave empty.

Response JSON	Responses vary by URL (`collection` vs. individual resource). For example, if you get an individual event log message, you may receive a response like this:
	`{`
	`"objectID": "CameraEventCollection:",`
	`"reqAction": 1,`
	`"totalRemoteCount": 2,`
	`"subCount": 2,`
	`"remoteSorting": true,`
	`"nextPage": false,`
	`"nodePermission": 3,`
	`"nodeAccessDetails": 1,`
	`"collection": [`

```
            {
                "mkey": "FK400D3016000013\/System\/001735840431-
1048576:1\/1",
                "start_time": "2025-01-02 12:53:51",
                "start_timestamp": 1735840431,
                "type": 1048576,
                "subtype": 1,
                "forward_from_camera": ""
            },
            {
                "mkey": "FK400D3016000013\/System\/001735833948-
1048576:32\/1",
                "start_time": "2025-01-02 11:05:48",
                "start_timestamp": 1735833948,
                "type": 1048576,
                "subtype": 32,
                "forward_from_camera": ""
            }
        ]
}
```


Type

Only one bit is active, but when used in `FILTER` in the URL, multiple bits can be set.

- `0` — Evt_None
- `1 << 0` — Evt_Detect_Generic
- `1 << 1` — Evt_Detect_Motion (see Motion on page 26)
- `1 << 2` — Evt_Detect_Audio
- `1 << 3` — Evt_Detect_DI (digital input)
- `1 << 4` — Evt_Detect_PIR (passive infrared)
- `1 << 5` — Evt_Detect_Tamper (see Tamper on page 26)
- `1 << 6` — Evt_Detect_Face_Detection (see Face detection on page 26)
- `1 << 7` — Evt_Detect_Physical_Access
- `1 << 8` — Evt_Detect_Object_Detection
- `1 << 16` — Evt_Camera (see Camera event on page 27)
- `1 << 17` — Evt_Recording (see Recording on page 27)
- `1 << 18` — Evt_Schedule (see Camera event on page 27)
- `1 << 19` — Evt_Annotate
- `1 << 20` — Evt_System (see System event on page 27)
- `1 << 21` — Evt_Notification

Subtype

Subtype is relative to the event type. Only one bit is active.

Motion

- `0` — SubEvt_Motion_None
- `1 << 0` — SubEvt_Motion_Motion
- `1 << 1` — SubEvt_Motion_MotionAlarm
- `1 << 2` — SubEvt_Motion_ObjectInside
- `1 << 3` — SubEvt_Motion_Crossed

Tamper

- `0` — SubEvt_Tamper_None
- `1 << 0` — SubEvt_Tamper_Realtime
- `1 << 1` — SubEvt_Tamper_Tamper
- `1 << 2` — SubEvt_Tamper_SceneChanged

Face detection

- `0` — SubEvt_Face_None
- `1 << 0` — SubEvt_Face_Blocked
- `1 << 1` — SubEvt_Face_VIP
- `1 << 2` — SubEvt_Face_Expired
- `1 << 3` — SubEvt_Face_Unknown
- `1 << 4` — SubEvt_Face_Generic
- `1 << 5` — SubEvt_Face_Masked
- `1 << 6` — SubEvt_Face_Unmasked

Object detection

- `0` — SubEvt_Object_None
- `1 << 0` — SubEvt_Object_Person
- `1 << 1` — SubEvt_Object_Motion
- `1 << 2` — SubEvt_Object_Weapon
- `1 << 3` — SubEvt_Object_Vehicle
- `1 << 4` — SubEvt_Object_Animal

- `1 << 5` — SubEvt_Object_Item
- `1 << 6` — SubEvt_Object_Sports

Camera event

- `0` — SubEvt_Camera_None
- `1 << 0` — SubEvt_Camera_Reset
- `1 << 1` — SubEvt_Camera_Reboot
- `1 << 2` — SubEvt_Camera_Power Up
- `1 << 3` — SubEvt_Camera_Restart
- `1 << 4` — SubEvt_Camera_Disable
- `1 << 5` — SubEvt_Camera_Enable
- `1 << 6` — SubEvt_Camera_SD_Format
- `1 << 7` — SubEvt_Camera_Upgrade
- `1 << 8` — SubEvt_Camera_Suspend
- `1 << 9` — SubEvt_Camera_Resume
- `1 << 10` — SubEvt_Camera_Interruption

Recording

- `0` — SubEvt_Rec_None
- `1 << 0` — SubEvt_Rec_Continuous
- `1 << 1` — SubEvt_Rec_Detection
- `1 << 2` — SubEvt_Rec_Manual
- `1 << 3` — SubEvt_Rec_Temp

System event

- `0` — SubEvt_System_None
- `1 << 0` — SubEvt_System_Startup
- `1 << 1` — SubEvt_System_Halt
- `1 << 2` — SubEvt_System_Reboot
- `1 << 3` — SubEvt_System_Reload
- `1 << 4` — SubEvt_System_Disk
- `1 << 5` — SubEvt_System_Upgrade
- `1 << 6` — SubEvt_System_Downgrade
- `1 << 7` — SubEvt_System_Loadgui
- `1 << 8` — SubEvt_System_Update

State

Multiple bits can be active at the same time.

- `0` — State_None
- `1 << 0` — State_Active (camera is recording)
- `1 << 1` — State_Inactive (camera is not recording)
- `1 << 1` — State_Edge (stored on the camera's SD card)
- `1 << 1` — State_NonEdge (stored by FortiRecorder)
- `1 << 1` — State_Locked (recording file is locked)
- `1 << 1` — State_Unlocked

Example: Get events after 1 AM on January 1, 2025

```
curl -X GET -b cookie.txt "https://HOST_OR_IP/api/v1/CameraEvent?start_time=2025-01-01-01-00-00"
```

On Microsoft Windows with the `curl` command, if the URL has parameters, then you must put double quotes (") around the URL. Otherwise the command line may try to interpret URL parameters after each ampersand (&), question mark (?), or asterisk/star (*) as commands or arguments. Errors at the end of the JSON indicate this problem but do not accurately show the cause, such as:

```
'reqAction' is not recognized as an internal or external command, operable
     program or batch file.
```

Reserved characters and escape sequence vary by operating system and command line environment.

Example: Get events for cameras "cam1" and "cam2" during a specified time period

```
curl -X GET -b cookie.txt "https://HOST_OR_IP/api/v1/ Timeline?device_name=cam1,cam2&start_
time=2025-01-01-01-00-00&end_time=2025-01-02-14-00-00"
```

CameraProfile

Camera profiles.

Permissions required include *Camera configuration*.

HTTP Methods	• GET • POST • PUT • DELETE
URL	For HTTP GET, use either: • https://HOST_OR_IP/api/v1/CameraVideoProfile/ • https://HOST_OR_IP/api/v1/CameraVideoProfile/PROFILE_NAME/ For HTTPPOST, PUT, and DELETE, use: • https://HOST_OR_IP/api/v1/CameraVideoProfile/PROFILE_NAME/ where variables are: • PROFILE_NAME — Unique identifier of a specific resource, such as the video profile high-resolution
Request JSON	For HTTP GET or DELETE, leave the request body empty. For HTTP POST, configure all required settings. For a list of available settings, use HTTP GET. For HTTP PUT, configure only settings that you want to change. Omit others.

Response JSON	Responses vary by HTTP method and URL (collection vs. individual resource). For example, if you get the list of camera profiles, you may receive a response like this:
	<pre>{
	"objectID": "CameraProfileCollection:",
	"reqAction": 1,
	"totalRemoteCount": 12,
	"subCount": 12,
	"remoteSorting": true,
	"nextPage": false,
	"nodePermission": 3,
	"nodeAccessDetails": 1,
	"collection": [
	{
	"mkey": "CloudCameraProfile-0001",</pre>

```
                "management_mode": 1,
                "continuous_retention_disposition": 0,
                "continuous_retention_period": 1,
                "continuous_retention_period_units": 2,
                "detection_retention_disposition": 0,
                "detection_retention_period": 1,
                "detection_retention_period_units": 2,
                "compression": false,
                "compression_period": 1,
                "compression_period_units": 2,
                "viewing_stream":
"Always:CloudCameraVideoViewingProfile",
                "recording_stream":
"Always:CloudCameraVideoRecordingProfile-0001",
                "recording_type": "Always:3",
                "isReferenced": 1},
                ...
            }]
        }
}
```

where attributes are:

- `continuous_retention_disposition` — Storage strategy for recordings when `recording_type` includes continuous recording.
  - `0` — Keep until overwritten. Retain video until all available disk space, both local and remote, is almost full. Then the oldest video will be overwritten.
  - `1` — Delete. Remove video when it exceeds the maximum age in `continuous_retention_period`, or if the disk is full. Recordings are stored on the hard disk as multiple video files. The oldest part of the recording is deleted or moved first. If remote storage is configured, video is first stored on the local disk, then transferred to remote storage when the local disk needs space for newer video. When the remote disk is full too, the video will finally be deleted from it.
  - `2` — Move video to external storage when it exceeds the maximum age, or if the disk is full. This option has no effect if remote storage is disabled.
- `continuous_retention_period` — If `continuous_retention_disposition` is `1` or `2`, remove continuous recording files after this amount of time. Also configure `continuous_retention_units`.
- `continuous_retention_period_units` — Unit of time for `continuous_retention_period`. Valid values are:
  - `0` — Days.
  - `1` — Weeks.
  - `2` — Months.
  - `3` — Years.

- `detection_retention_disposition` — Storage strategy for recordings triggered when `recording_type` includes any non-continuous trigger, such as motion detection. Options are the same as `continuous_retention_disposition`, except that valid values also include:
  - `3` — Use continuous recordings if available. Marks the time ranges of motion detection inside of continuous recordings on the timeline instead of storing them as separate copies of the video files (sometimes also called "zero penalty" because the NVR system resources are not spent on copies). This reduces CPU load, but does not allow continuous recordings to be deleted, and FortiRecorder only keeps the section of video that was marked by motion detection for an amount of time that varies by, for example, whether or not the live video stream is currently being used.
- `detection_retention_period` — If `detection_retention_disposition` is `1` or `2`, remove detection-triggered recording files after this amount of time. Also configure `detection_retention_units`.
- `detection_retention_period_units` — Unit of time for `detection_retention_period`. Valid values are:
  - `0` — Days.
  - `1` — Weeks.
  - `2` — Months.
  - `3` — Years.
- `compression` — Whether or not to reduce the disk space usage of continuous recordings by removing P-frames. This can reduce video frame rate. It can also increase CPU and disk space usage. Valid values are:
  - `0` — Do not compress.
  - `1` — Compress.
- `compression_period` — If `compression` is `1`, remove P-frames after this amount of time. Also configure `compression_period_units`.
- `compression_period_units` — Unit of time for `compression_period`. Valid values are:
  - `0` — Days.
  - `1` — Weeks.
  - `2` — Months.
  - `3` — Years.
- `viewing_stream` — Video profile used by **live view video streams**, at each scheduled time. Separate each part of the tuple with a colon and separate each tuple with a comma, such as `"Away:high-resolution,BusinessHours:med-resolution"`. See CameraVideoProfile on page 37 and ScheduleObject on page 40.
- `recording_stream` — Video profile used by **recording video streams**, at each scheduled time. Formatting is the same as `viewing_stream`.
- `recording_type` — When to use the specified recording triggers, where to store the recordings, and which triggers cause the camera to begin

	recording, expressed as a bit mask, such as `"Always:17"`. See Recording type on page 32 and ScheduleObject on page 40.

Recording type

Multiple bits can be active at the same time.

- `1 << 0` — Store on FortiRecorder.
- `1 << 1` — Store on the camera's SD card ("edge recording").
- `1 << 4` — Continuous recording.
- `1 << 5` — Motion detection recording.
- `1 << 6` — Digital input.
- `1 << 7` — Audio detection.
- `1 << 8` — Passive infrared (PIR) detection.
- `1 << 9` — Tamper detection.

Example: Change to high resolution for recordings

Updates the recording stream (but not the live view stream) to use high resolution.

```
curl -X PUT -b cookie.txt -H "Content-Type: application/json" -d "{\"recording_stream\":\"high-resolution\"}" https://HOST_OR_IP/api/v1/CameraProfile/PROFILE_NAME/CameraProfileVideoSchedule/SCHEDULE_NAME/
```

On Microsoft Windows with Command Prompt, inside the JSON data, you must put a backslash before each double straight quote (\"). For example (highlighted in bold): `-d "{\"recording_stream\":\"high-resolution\"}"`

Alternatively, you can input a JSON stream from another command.

If you do not, the command line may interpret each JSON attribute as CLI commands or arguments, resulting in various error messages depending on the sequential order of arguments and attributes.

Reserved characters and escape sequences vary by operating system and command line environment; Linux and Mac terminals often do not require this, and Microsoft PowerShell uses different escape sequences.

CameraSnapshot

Snapshot image in JPG file format from a camera's live stream, at the current timestamp. Alternatively, see VideoClip on page 58.

Permissions required include both *Camera live view* and *Video playback*.

HTTP Methods	• `GET`
URL	`https://HOST_OR_IP/api/v1/CameraSnapshot?camera=CAMERA_NAME&reqAction=21` where variables are: • `CAMERA_NAME` — Unique identifier of the camera, such as `fd51` • `reqAction` — Action. Takes precedence over the method in the HTTP header. Required value is `21` (process and download the file). If you do not include it, FortiRecorder returns a `200 OK` status, but the body is a JSON object instead of an image. If you include it but have the wrong value, FortiRecorder may return either a `200 OK` or `403 Forbidden` error status.

Example

```
curl -X GET -b cookie.txt -o snapshot.jpg
"https://172.20.141.94/api/v1/CameraSnapshot?camera=MC51&reqAction=21"
```

On Microsoft Windows with Command Prompt and the `curl` command, if the URL has parameters, then you must put double straight quotes (") around the URL. If you do not, the command line may try to interpret URL parameters after each ampersand (&), question mark (?), or asterisk/star (*) as commands or arguments. Errors at the end of the JSON indicate this problem but do not accurately show the cause, such as:

```
'reqAction' is not recognized as an internal or external command, operable
      program or batch file.
```

Reserved characters and escape sequence vary by operating system and command line environment. Linux and Mac terminals often do not require this, and Microsoft PowerShell uses different escape sequences.

CameraStatus

Camera statuses.

Permissions required include both *Camera status* and *Camera configuration*.

HTTP Methods	• `GET`

URL	Either:
	• `https://HOST_OR_IP/api/v1/CameraStatus`
	• `https://HOST_OR_IP/api/v1/CameraStatus?showInactiveCamera=0`
	where variables are:
	• `showInactiveCamera` — Whether or not to include cameras whose `status` is 2:
	  • `0` — Do not include. Default value if this parameter is not included in the URL.
	  • `1` — Include.

Response JSON	{ "objectID": "CameraStatusCollection:", "reqAction": 1, "totalRemoteCount": 5, "subCount": 5, "remoteSorting": true, "nextPage": false, "nodePermission": 3, "nodeAccessDetails": 1, "collection": [{ "mkey": "FD50", "status": 1, "action_scheduled": 3, "action_pending": 2, "action_current": 1, "action_problem": 0, "last_query": 1762267129, "flag": 1, "detail_code": 0, "detail_info": "", "support_status": 0, "eos_version": "" }, ...]

```
}
```

where attributes are:

- `mkey` — Unique name of the device.
- `status` — Whether the device is:
  - `1` — Enabled.
  - `2` — Disabled.
- `action_scheduled` — Scheduled actions, represented as a bitmask. See Action on page 35.
- `action_pending` — Action started but not yet active, represented as a bitmask. See Action on page 35.
- `action_current` — Current action, represented as a bitmask. See Action on page 35.
- `action_problem` — Errors during action, represented as a bitmask. See Action on page 35.
- `last_query` — Timestamp of the last status query in UTC in seconds since January 1, 1970, UTC (also called the Unix epoch). Convert to the Unix epoch to a UTC relative timestamp, and then convert UTC to your local time zone.
- `flag` — Connectivity and activity status, represented as a bitmask. See Flag on page 36.
- `detail_code` — Error code, if any.
- `detail_info` — Error code details, if any.
- `support_status` — Whether the device has reached the end of support version:
  - `0` — Device is supported by this FortiRecorder version.
  - `1` — Device is not supported by this FortiRecorderversion.
- `eos_version` — FortiRecorder version number when this device will reach end-of-support, such as `7.4.0`.

Action

Multiple bits can be active at the same time.

- `0` — Idle.
- `1 << 0` — Continuous recording. See also Recording type on page 32.
- `1 << 1` — Motion detection recording.
- `1 << 2` — Digital input.
- `1 << 3` — Audio detection.
- `1 << 4` — Passive infrared (PIR) detection.
- `1 << 5` — Tamper detection.

- `1 << 8` — Continuous recording on the camera's SD card ("edge recording"). See also Recording type on page 32.
- `1 << 9` — Motion recording on the camera's SD card ("edge recording").
- `1 << 10` — Digital input recording on the camera's SD card ("edge recording").
- `1 << 11` — Audio recording on the camera's SD card ("edge recording").
- `1 << 12` — Passive infrared (PIR) recording on the camera's SD card ("edge recording").
- `1 << 13` — Tamper detection recording on the camera's SD card ("edge recording").

Flag

- `0` — End-of-support reached. FortiRecorder will not try to connect to the camera.
- `1` — Active. Camera has recently communicated with FortiRecorder.
- `2` — Inactive. Camera has not recently communicated with FortiRecorder.
- `3` — Camera is not configured.
- `4` — Camera is unreachable.
- `5` — Camera is not configured and has a default address.
- `6` — Camera has an invalid address.
- `7` — Camera has a default address.
- `8` — Camera is being configured.
- `9` — Camera has a configuration error.
- `10` — Camera is upgrading its software.
- `11` — Camera is rebooting.
- `12` — Camera is not configured and has an invalid address.
- `13` — Duplicate IP address detected. Camera has the same IP address as another device.
- `14` — Camera is managed by another FortiRecorder.

CameraVideoProfile

Video profiles.

Permissions required include *Camera configuration*.

HTTP Methods	• GET • POST • PUT • DELETE
URL	For HTTP GET, use either: • https://HOST_OR_IP/api/v1/CameraVideoProfile/ • https://HOST_OR_IP/api/v1/CameraVideoProfile/PROFILE_NAME/ For HTTPPOST, PUT, and DELETE, use: • https://HOST_OR_IP/api/v1/CameraVideoProfile/PROFILE_NAME/ where variables are: • PROFILE_NAME — Unique identifier of a specific resource, such as the video profile high-resolution
Request JSON	For HTTP GET or DELETE, leave the request body empty. For HTTP POST, configure all required settings. For a list of available settings, use HTTP GET. For HTTP PUT, configure only settings that you want to change. Omit others.
Response JSON	For HTTP PUT and POST, responses vary by which settings you changed. For HTTP GET, responses vary by URL (collection vs. individual resource): For example, if you get a video profile, your response may look like this: { "objectID": "CameraVideoProfile:high-resolution", "reqAction": 1, "nodePermission": 3, "mkey": "high-resolution", "management_mode": 0, "video_resolution": 9, "video_codec": 0, "video_fps": 15, "video_gop": 3, "video_bitrate_mode": 0, "video_bitrate": 2048, "video_max_bitrate": 2048,

```
    "video_quality": 1,
    "audio": true
}
```

where attributes are :

- `audio` — Whether or not to include an audio track (if the camera has a microphone or audio input):
  - `true` — Enabled.
  - `false` — Disabled.
- `video_codec` — . Valid values are:
  - `0` — Default.
  - `3` — H.264 AVC.
  - `4` — H.265 HEVC.
- `video_fps` — Smoothness of motion in frames per second. Conventional video is 24 frames per second.
- `video_gop` — Number of seconds between each reference frame in the group of pictures. Longer intervals save bandwidth, but slightly delay the start of live video streams. Valid values are:
  - `0` — Automatic.
  - `1` — 0.25 seconds.
  - `2` — 0.5 seconds.
  - `3` — 1 second.
  - `4` — 2 seconds.
  - `5` — 3 seconds.
  - `6` — 4 seconds.
- `video_resolution` — Amount of detail in number of pixels. Lower resolutions have less detail but are faster to transmit. Higher resolutions produce a clearer image but require more bandwidth. A higher resolution is preferable if the camera is recording a large space, such as a parking lot, where small details like faces and license plates could be important. Valid values are:
  - `0` — Low.
  - `1` — Medium.
  - `2` — High.
  - `3` — Extra high.
  - `4` — 0.5 megapixels.
  - `5` — 1 megapixel.
  - `6` — 2 megapixels.
  - `7` — 3 megapixels.
  - `8` — 4 megapixels.
  - `9` — 5 megapixels.
  - `10` — 6 megapixels.
  - `11` — 9 megapixels.

    - `12` — 12 megapixels.
- `video_bitrate_mode` — . Valid values are:
    - `0` — Variable. Automatically adjust the stream to the minimum bit rate required by the current video frames while maintaining video quality. Bitrate is less when there is less motion. This setting increases image noise.
    - `1` — Fixed. Manually specify a constant bit rate in `video_bitrate`. This guarantees calculated video retention time and bandwidth but might reduce image quality if, for example, if there is a sudden burst of motion, like rain or flashing lights.
    - `2` — Constrained. Automatically adjusts the stream to reduce the bitrate usage while recording less motion, and increasing bitrate while recording more motion. Unlike `0`, however, you can set the maximum bitrate that the camera uses in `video_max_bitrate`.
- `video_bitrate` — If `video_bitrate_mode` is `1`, indicate the bitrate of the video.
- `video_max_bitrate` — If `video_bitrate_mode` is `2`, indicate the maximum bitrate of the video.
- `video_quality` — If `video_bitrate_mode` is `0` or `2`, indicate the target quality of the video. Valid values are:
    - `0` — Low.
    - `1` — Standard.
    - `2` — Good.
    - `3` — High.
    - `4` — Extra high.

Example: Modify the video frame rate

```
curl -X PUT -H "Content-Type:application/json" -d "{\"video_fps\":15}" -b cookie.txt https://HOST_
OR_IP/api/v1/CameraVideoProfile/high-resolution
```

On Microsoft Windows with Command Prompt, inside the JSON data, you must put a backslash before each double straight quote (\"). For example (highlighted in bold): `-d "{\"video_fps\":15}"`

Alternatively, you can input a JSON stream from another command.

If you do not, the command line may interpret each JSON attribute as CLI commands or arguments, resulting in various error messages depending on the sequential order of arguments and attributes.

Reserved characters and escape sequences vary by operating system and command line environment; Linux and Mac terminals often do not require this, and Microsoft PowerShell uses different escape sequences.

ScheduleObject

View or change schedules.

Permissions required include *Camera configuration*.

HTTP Methods	• GET • POST • PUT • DELETE
URL	For HTTP GET, use either: • https://HOST_OR_IP/api/v1/ScheduleObject/ • https://HOST_OR_IP/api/v1/ScheduleObject/SCHEDULE_NAME For HTTP POST, PUT, or DELETE, use: • https://HOST_OR_IP/api/v1/ScheduleObject/SCHEDULE_NAME where variables are: • SCHEDULE_NAME — Unique identifier of a specific resource, such as the schedule Away. The schedules must be of type Assistant.
Request JSON	For HTTP GET or DELETE, leave the request body empty. For HTTP PUT or POST, configure only settings that you want to change. Omit others. For a list of available settings, use HTTP GET.

Response JSON	For HTTP PUT, responses vary by which settings you changed.
	For HTTP GET, responses vary by URL (collection vs. individual resource). For example, if you get an individual schedule, you may receive a response like this:
	
	`{`
	`"objectID": "ScheduleObject:BusinessHours",`
	`"reqAction": 1,`
	`"nodePermission": 3,`
	`"mkey": "BusinessHours",`
	`"description": "Typical business hours.",`
	`"management_mode": 0,`
	`"type": 0,`
	`"assist_active": false,`
	`"all_day": false,`
	`"date_start": "",`
	`"start_time_type": 0,`
	`"time_start": "09-00",`
	`"date_end": "",`
	`"end_time_type": 0,`
	`"time_end": "17-00",`
	`"days": 62,`

```
    "zenith": 0,
    "sunrise_offset": 0,
    "sunset_offset": 0
}
```

where attributes are often self-explanatory, except:

- `type` — Frequency of the schedule.
  - `0` — Recurring on multiple days of the week.
  - `1` — One date or time range only.
  - `2` — Used by the default schedule named `Away`.
- `days` — Which days of the week to apply a recurring schedule to. Total of the following numbers, depending on which days you choose:
  - `1` — Sunday.
  - `2` — Monday.
  - `4` — Tuesday.
  - `8` — Wednesday.
  - `16` — Thursday.
  - `32` — Friday.
  - `64` — Saturday.
  - `62` — Monday through Friday.
  - `127` — All days.
- `all_day` — .Whether or not the schedule applies to the entire day:
  - `true` — All day.
  - `false` — Part of the day, defined by `time_start` and `time_end`, or times defined by the sunrise and sunset on each date.
- `start_time_type` — Either:
  - `0` — Time specified in `time_start`.
  - `1` — Sunrise on that date.
  - `2` — Sunset on that date.
- `end_time_type` — Either:
  - `0` — Time specified in `time_end`.
  - `1` — Sunrise on that date.
  - `2` — Sunset on that date
- `date_start` — Which date to start a one-time schedule. Format is `"YYYY-MM-DD"` (year, month, day). This setting applies if `type` is `1`.
- `date_end` — Which date to stop a one-time schedule. Format is `"YYYY-MM-DD"`. This setting applies if `type` or is `1`.
- `time_start` — Which time to begin the schedule. Format is `"HH-MM"` (hour, minute) according to a 24-hour clock. This setting applies if `start_time_type` is `0`.
- `time_end` — Which time to stop the schedule. Format is `"HH-MM"`. This setting applies if `end_time_type` is `0`.
- `sunrise_offset` — .Minutes of offset from the sunrise time . This

setting applies if `start_time_type` or `end_time_type` is `1`.

- `sunset_offset` — Minutes of offset from the sunset time. This setting applies if `start_time_type` or `end_time_type` is `2`.

Example: Change schedule time range to all day

```
curl -X PUT -H "Content-Type: application/json" -d "{\"all_day\":true}" -b cookie.txt
https://HOST_OR_IP/api/v1/ScheduleObject/SCHEDULE_NAME
```

After you configure the schedule, you can use it in another part of the configuration. See CameraProfile on page 29.

On Microsoft Windows with Command Prompt, inside the JSON data, you must put a backslash before each double straight quote (\"). For example (highlighted in bold): `-d "{\"all_day\":true}"`

Alternatively, you can input a JSON stream from another command.

If you do not, the command line may interpret each JSON attribute as CLI commands or arguments, resulting in various error messages depending on the sequential order of arguments and attributes.

Reserved characters and escape sequences vary by operating system and command line environment; Linux and Mac terminals often do not require this, and Microsoft PowerShell uses different escape sequences.

SysGlobal

System-wide settings such as the authentication session idle timeout. For descriptions, see the FortiRecorder Administration Guide.

Permissions required include *System access*.

HTTP Methods	• GET • PUT
URL	`https://HOST_OR_IP/api/v1/SysGlobal/`

Request JSON	For GET requests, leave the request body empty.
	For PUT requests, configure only settings that you want to change. Omit others. For example:
	
	`{`
	`  "admin_lockout_threshold": 5`
	`}`
	
	For a list of available settings, use HTTP GET.

Response JSON	For PUT requests, responses vary by which settings you changed.
	For GET requests, responses are like the following. (The list is not exhaustive.)
	
	`{`
	`  "objectID": "SysGlobal:",`
	`  "reqAction": 1,`
	`  "nodePermission": 3,`
	`  "default_certificate": "Factory",`
	`  "hostname": "FK400D3016000001",`
	`  "local_domain_name": "",`
	`  "admin_timeout": 120,`
	`  "pki_mode": true,`
	`  "pki_certificate_req": 0,`
	`  "lcd_protection": false,`
	`  "lcd_pin": "******",`
	`  "disk_monitor": false,`
	`  "pre_login_banner": 0,`
	`  "post_login_banner": 0,`
	`  "ldap_sess_cache_enable": true,`
	`  "ldap_worker_threads": 15,`
	`  "set_strong_crypto": 1,`

```
        "ssl_versions": 28,
        "port_http": 80,
        "port_https": 443,
        "port_ssh": 22,
        "port_telnet": 23,
        "admin_scp": true,
        "iscsi_initiator_name": "iqn.2011-
    08.com.fortirecorder:5a79d1157d",
        "ldap_server_sys_status": true,
        "dh_params": 1024,
        "tftp": true,
        "rest_api": true,
        "fabric_api_token": "6^7^5#I9ba83mD$my+VUMv3qQVX$j75!",
        "cloud_acct_id": "",
        "admin_lockout_threshold": 3,
        "admin_lockout_duration": 3,
        "admin_maintainer": true,
        "hsts_max_age": 365,
        "csrf_token": true,
        "public_address": "nvr.example.com",
        "public_https_port": 443,
        "public_http_port": 80,
        "public_rtsp_port": 554,
        "public_ftp_port": 21,
        "public_frc_central_port": 8550,
        "public_notify_tcp_port": 3010,
        "public_notify_http_port": 3011,
        "HLS_over_HTTP": 0,
        "port_frc_central": 8550,
        "frc_central_secure": false,
        "port_rtsp": 554,
        "automatic_camera_add": true,
        "cors_address": "",
        "chromecast_appid": "AD3B8D83",
        "chromecast_http": false,
        "chromecast_delay": 0,
        "show_topology": false,
        "clip_extraction_threads": 1,
        "face_recognition": true,
        "strong_crypto_notification": false,
        "override_cloud_status": false,
        "override_cloud_tunnel_address": "0.0.0.0",
        "override_cloud_signal_address": "0.0.0.0",
        "override_cloud_tunnel_port": 10000,
        "override_cloud_signal_port": 61614,
        "cloud_region": "",
        "fortirecorder_id": "9223653511831489586",
        "tags": ""
```

where attributes often self-explanatory, except:

- `set_strong_crypto` — Whether or not strong encryption is required (this deprecates ciphers and older protocol versions with known weaknesses and vulnerabilities):
  - `0` — Disabled.
  - `1` — Enabled.
- `pre-login-banner` — Whether or not to show the legal disclaimer before a user or administrator logs into the GUI:
  - `0` — Disabled.
  - `1` — Enabled.

Example: Get global settings for FortiRecorder

```
curl -X GET -H "Content-Type:application/json" -b cookie.txt https://HOST_OR_IP/api/v1/SysGlobal/
```

SysInterface

Network interfaces such as *port1*. For descriptions, see the FortiRecorder Administration Guide.

Permissions required include *System access*.

HTTP Methods	• GET • POST • PUT • DELETE Physical network interfaces cannot be created nor deleted.
URL	For HTTP GET, use either: • https://HOST_OR_IP/api/v1/SysInterface/ • https://HOST_OR_IP/api/v1/SysInterface/INTERFACE_NAME For HTTPPOST, PUT, and DELETE, use: • https://HOST_OR_IP/api/v1/SysInterface/INTERFACE_NAME where variables are: • INTERFACE_NAME — Optional. Unique identifier of a specific resource, such as the network interface port1

Request JSON	For HTTP GET or DELETE, leave the request body empty.
	For HTTP POST, configure all required settings. For a list of available settings, use HTTP GET.
	For HTTP PUT, configure only settings that you want to change. Omit others. For example:
	{
	"link_status": true
	}

Response JSON	For HTTP PUT, responses vary by which settings you changed. For HTTP GET, responses vary by HTTP method and URL (collection vs. individual resource). For example, if you get port2: { "objectID": "SysInterface:port2", "reqAction": 1, "nodePermission": 3, "mkey": "port2",

```
    "type": 0,
    "aggregate_master": 0,
    "bridge_member": true,
    "ip": "172.168.1.103/24",
    "ip6": "::/0",
    "status": true,
    "interface": "",
    "aggregate_member": "",
    "incoming_mode": 2,
    "outgoing_mode": 0,
    "local": true,
    "allowaccess": 23,
    "discover": true,
    "webaccess": 1,
    "mode": 0,
    "connection": false,
    "defaultgw": false,
    "mtu": 1500,
    "speed": 0,
    "mac_addr": "00:10:f3:37:6c:e2",
    "vlanid": 1,
    "aggregate_monitor": 0,
    "aggregate_arp_ip": "",
    "aggregate_algorithm": 0,
    "aggregate_mode": 4,
    "rx_queue": 0,
    "tx_queue": 0,
    "link_status": true
}
```


where attributes often self-explanatory, except:

- `type` — Type of network interface:
  - `0` — Physical.
  - `1` — VLAN.
  - `2` — Aggregate.
  - `3` — Redundant.
- `allowaccess` — Protocols that are allowed to access FortiRecorder on this network interface. Total of the following numbers (depending on which protocols you allow):
  - `1` — HTTPS.
  - `2` — PING.
  - `4` — SSH.
  - `8` — SNMP.
  - `16` — HTTP.
  - `32` — Telnet.
  - `128` — FortiCentral.

    - `256` — RTSP.
- `discover` — Whether or not to automatically discover cameras connected to this port:
    - `true` — Enabled.
    - `false` — Disabled.
- `link_status` — Administrative status:
    - `true` — Interface is up.
    - `false` — Interface is down.

Example: Get a list of network interfaces

```
curl -X GET -b cookie.txt https://HOST_OR_IP/api/v1/SysInterface/
```

Example: Allow only RTSP, FortiCentral, and HTTPS connections to port2

```
curl -X PUT -H "Content-Type:application/json" -d "{\"allowaccess\":385}" -b cookie.txt
https://HOST_OR_IP/api/v1/SysInterface/port2
```

SysStatusCommand

Reboot, shutdown, or reload the FortiRecorder configuration.

Permissions required include *System access*.

HTTP Methods	• `POST`
URL	`https://HOST_OR_IP/api/v1/SysStatusCommand/`

Request JSON	`{`
	`  "action": ACTION_ID`
	`}`
	
	where valid values are:
	• `1` — Restart.
	• `2` — Halt.
	• `3` — Reload the configuration from disk.

SysStatusSysinfo

FortiRecorder system status information.See also SysStatusUsage on page 52. For descriptions, see the FortiRecorder Administration Guide.

Permissions required include *System status*.

HTTP Methods	• GET
URL	`https://HOST_OR_IP/api/v1/SysStatusSysinfo/`
Response JSON	{ "objectID": "SysStatusSysinfo:", "reqAction": 1, "nodePermission": 3, "serial_number": "FK100GS123000034", "up_time": "1 18 41 22", "system_time": 1758227265, "firmware_version": "v7.2.5(GA), build284, 2025.07.18", "current_admin": "rest", "admin_num": 2, "log_disk_capacity": 18583, "log_disk_used": 8, "log_disk_status": 0, "remote_video_disk_status": 1, "log_disk_info": "Capacity 18 GB, Used 8 MB (0.05%), Free 18 GB", "remote_video_disk_info": "n\/a", "actual_retention_remote": 0, "estimated_retention_remote": 0, "actual_retention_local": 3049097, "estimated_retention_local": 0, "retention_status": 2, "data_capacity_local": 0, "data_capacity_remote": 0, "data_capacity_status": 0, "data_rate_local": 0, "data_rate_remote": 0, "nas_status": 0, "local_storage_status": 0, "remote_storage_status": 0, "log_storage_status": 0, "ai_model_status": 1, "free_space_low_local": 0, "free_space_low_remote": 0, "systime_str": "Thu Sep 18 16:27:45 EDT 2025"

```
}
```

where attributes often self-explanatory, except:

- `actual_retention_local` — Amount of video stored in seconds.
- `estimated_retention_local` — Video retention estimate, based on bandwidth usage for the previous 5 minutes.
- `log_storage_status` — Log disk status:
  - `0` — OK.
  - `1` — Not mounted.
  - `2` — Not accessible.
  - `3` — Not writable (for example, read-only).
- `log_disk_capacity` — Log disk capacity in megabytes (MB)
- `system_time` — Time on FortiRecorder in UTC, since the Unix epoch.
- `up_time` — Time since startup in days, hours, minutes, and seconds, each separated by a space. For example, `2 1 5 0` indicates an uptime of 2 days 1 hour 5 minutes.

SysStatusUsage

FortiRecorder system resource usage information such as CPU usage in percent, disk space usage in percent , and network bandwidth usage. See also SysStatusSysinfo on page 50.

Historical usage data such as `cpu_history` can be used to generate charts and graphs with third-party tools.

Permissions required include *System status*.

HTTP Methods	• GET
URL	`https://HOST_OR_IP/api/v1/SysStatusUsage/`

Response JSON	{
	"objectID": "SysStatusUsage:",
	"reqAction": 1,
	"nodePermission": 3,
	"cpu": 8,
	"memory": 43,
	"log_disk": 2,
	"mail_disk": 90,
	"system_load": 15,
	"active_sessions": 26,
	"network_usage": 11200,
	"cpu_history": "{
	"values": [8, 10, 9, 10, 10, 10, 10, 10, 9, 9, 8, 9, 10, 9, 10, 9, 8, 8, 8, 9, 9, 9, 10, 9, 8, 8, 10, 9, 8, 8, 8],
	"x_labels": ["15:17:40", "15:18:00", "15:18:20", "15:18:40", "15:19:00", "15:19:20", "15:19:40", "15:20:00", "15:20:20", "15:20:40", "15:21:00", "15:21:20", "15:21:40", "15:22:00", "15:22:20", "15:22:40", "15:23:00", "15:23:20", "15:23:40", "15:24:00", "15:24:20", "15:24:40", "15:25:00", "15:25:20", "15:25:40", "15:26:00", "15:26:20", "15:26:40", "15:27:00", "15:27:20", "15:27:40"],
	"y_legend": "%",
	"y_step": 10,
	"y_max": 100
	}",
	"memory_history": "{
	"values": [43, 49, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43],
	"x_labels": ["15:17:40", "15:18:00", "15:18:20", "15:18:40", "15:19:00", "15:19:20", "15:19:40", "15:20:00", "15:20:20",

```
"15:20:40", "15:21:00", "15:21:20", "15:21:40", "15:22:00",
"15:22:20", "15:22:40", "15:23:00", "15:23:20", "15:23:40",
"15:24:00", "15:24:20", "15:24:40", "15:25:00", "15:25:20",
"15:25:40", "15:26:00", "15:26:20", "15:26:40", "15:27:00",
"15:27:20", "15:27:40"],
        "y_legend": "%",
        "y_step": 10,
        "y_max": 100
    }",
    "session_history": "{
        "values": [26, 26, 25, 26, 26, 24, 25, 26, 24, 25, 26, 25,
27, 26, 25, 26, 26, 25, 26, 26, 25, 25, 26, 25, 26, 25, 25, 25, 27,
25, 26],
        "x_labels": ["15:17:40", "15:18:00", "15:18:20", "15:18:40",
"15:19:00", "15:19:20", "15:19:40", "15:20:00", "15:20:20",
"15:20:40", "15:21:00", "15:21:20", "15:21:40", "15:22:00",
"15:22:20", "15:22:40", "15:23:00", "15:23:20", "15:23:40",
"15:24:00", "15:24:20", "15:24:40", "15:25:00", "15:25:20",
"15:25:40", "15:26:00", "15:26:20", "15:26:40", "15:27:00",
"15:27:20", "15:27:40"],
        "y_step": 10,
        "y_max": 30
    }",
    "network_history": "{
        "values": [11200, 14569, 14791, 14838, 14962, 14768, 14652,
14770, 14800, 14656, 14446, 14506, 14777, 14648, 14232, 13236,
13327, 13315, 13213, 13138, 13092, 13076, 12802, 12162, 11159,
11409, 11324, 11193, 11296, 11525, 11200],
        "x_labels": ["15:17:40", "15:18:00", "15:18:20", "15:18:40",
"15:19:00", "15:19:20", "15:19:40", "15:20:00", "15:20:20",
"15:20:40", "15:21:00", "15:21:20", "15:21:40", "15:22:00",
"15:22:20", "15:22:40", "15:23:00", "15:23:20", "15:23:40",
"15:24:00", "15:24:20", "15:24:40", "15:25:00", "15:25:20",
"15:25:40", "15:26:00", "15:26:20", "15:26:40", "15:27:00",
"15:27:20", "15:27:40"],
        "y_legend": "Kbps",
        "y_step": 10000,
        "y_max": 20000
    }",
    "remote_video_disk": 0
}
```


where attributes often self-explanatory, except:

- `cpu_history` etc. — Array of values, units (`x_labels` etc.) and legend on the X- and Y-axis, and the maximum scale of the Y-axis.
- `network_usage` — Bandwidth usage in kilobits per second (Kbps).

Face_recognitionUser

Information about people used to identify them when their face is detected. This is used by face recognition AI on FortiRecorder.

Permissions required include *Camera analytics*.

HTTP Methods	• GET • POST • PUT • DELETE
URL	For HTTP GET, use either: • https://HOST_OR_IP/api/v1/Face_recognitionUser/ • https://HOST_OR_IP/api/v1/Face_recognitionUser/PERSON_NAME/ For HTTPPOST, PUT, and DELETE, use: • https://HOST_OR_IP/api/v1/Face_recognitionUser/PERSON_NAME/ where variables are: • USER_NAME — Unique identifier of a specific person defined for face recognition.

Response JSON	For HTTP PUT and POST, responses vary by which settings you changed.
	For HTTP GET, responses vary by URL (collection vs. individual resource): For example, if you get a person profile, your response may look like this:
	{
	"objectID": "Face_recognitionUser:pt001",
	"reqAction": 1,
	"nodePermission": 3,
	"mkey": "pt001",
	"department": "default-department",
	"role": "default-role",
	"display_name": "",
	"image_content": "face-recognition/employees/pt001/default_image.jpg",
	"group1_summary": {
	"images": [
	{
	"image_path": "face-recognition/employees/pt001/group1/image/68_1584718836.jpg",
	"is_local": true,

```
                "mkey": "68_1584718836"
            },
            {
                "image_path": "face-
recognition/employees/pt001/group1/image/73_1584718823.jpg",
                "is_local": true,
                "mkey": "73_1584718823"
            },
            {
                "image_path": "face-
recognition/employees/pt001/group1/image/74_1584718818.jpg",
                "is_local": true,
                "mkey": "74_1584718818"
            }
        ]
    },
    "department_display": "default-department",
    "role_display": "default-role",
    "default_images": "[]",
    "percent": {}
}
```


where attributes are:

- `is_local` — Whether or not you have already uploaded each thubnail image of the person:
  - `true` — A thumbnail image exists on FortiRecorder.
  - `false` — No thumbnail image exists yet on FortiRecorder. You can either upload it via the GUI, or use the REST API. See EmployeeFaceRecord on page 56.

Example: Create a person for face recognition on FortiRecorder

```
curl -X POST -b cookie.txt "https://HOST_OR_IP/api/v1/Face_recognitionUser/PERSON_NAME"
```

After you create the entry, you must upload thumbnails of the person's face in order for FortiRecorder to be able to identify them. See EmployeeFaceRecord on page 56.

EmployeeFaceRecord

Images of people used to identify them when their face is detected. This is used by face recognition AI on FortiRecorder.

Permissions required include *Camera analytics*.

HTTP Methods	• `POST` • `DELETE`
URL	`https://HOST_OR_IP/api/v1/EmployeeFaceRecord/`

Request JSON	For HTTP `POST`:
	
	`{`
	`"employeeUsername": "PERSON_NAME",`
	`"fileIndex": "FILE_NAME",`
	`"option": "raw",`
	`"content": "...=="`
	`}`
	
	where attributes are:
	• `employeeUsername` — Unique identifier of a specific person defined for face recognition.
	• `fileIndex` — Unique file name that will be used to store the thumbnail image on FortiRecorder. To get a list of existing file names, see Face_recognitionUser on page 54.
	• `option` — Either `raw` or `mctnn` (if the image has been processed by face recognition).
	• `content` — HTTP `POST` only. Base64-encoded string of the thumbnail image.
	For HTTP `DELETE`, omit `option` and `content`.

AiLog

Logs messages by face recognition AI on FortiRecorder.

Permissions required include *Camera analytics*.

HTTP Methods	• GET
URL	`https://HOST_OR_IP/api/v1/AiLog?type=activity&subtype=activity&range=TIMESTAMP&startIndex=INDEX&pageSize=ITEMS&start_time=TIMESTAMP&end_time=TIMESTAMP&person={user_id}&camera={camera_id}` where variables are: • `TIMESTAMP` — Optional (except for `range=TIMESTAMP`, which is a fixed value). Timestamp in the format `YYYY-MM-DD-HH-MM-SS` (year, month, day, hour, minute, second). • `INDEX` — Optional. See also Filtering and paging on page 11. • `PAGE_SIZE` — Optional. See also Filtering and paging on page 11. • `person` — Either the name of a person profile, `KNOWN` (all known persons), or `UNKNOWN` (all unknown persons). • `camera` — Name of a camera. See also CameraCamera on page 16. Other parts of the URL are required, but their values cannot be changed.

Example: Find times when unknown people were seen by camera MC51

```
curl -X GET -b cookie.txt "https://HOST_OR_
IP/api/v1/AiLog?Type=activity&subtype=activity&person=UNKNOWN&camera=MC51"
```

On Microsoft Windows with Command Prompt and the `curl` command, if the URL has parameters, then you must put double straight quotes (") around the URL. If you do not, the command line may try to interpret URL parameters after each ampersand (&), question mark (?), or asterisk/star (*) as commands or arguments. Errors at the end of the JSON indicate this problem but do not accurately show the cause, such as:

```
'subtype' is not recognized as an internal or external command, operable
     program or batch file.
```

Reserved characters and escape sequence vary by operating system and command line environment. Linux and Mac terminals often do not require this, and Microsoft PowerShell uses different escape sequences.

VideoClip

Download previously recorded video clips and snapshot images.

Permissions required include *Video playback* and *Camera live view*.

Timestamps are in UTC in seconds since January 1, 1970, UTC (also called the Unix epoch). Convert the local time zone on FortiRecorder to UTC, and then convert to the Unix epoch relative timestamp.

For example, January 1, 2025 at 1 PM EST may convert to 6 PM GMT, which is equivalent to an epoch timestamp of 1735754400.

Most request JSON attributes are optional, not required. If a request does not include them, then the video is downloaded as-is.

Time required to receive a response increases if you ask FortiRecorder to convert the video. Do not include optional attributes unless you want FortiRecorder to convert the video to a different size or quality, omit audio, etc. Conversion is best effort. It may not be possible in all cases to achieve a video clip with those exact characteristics.

HTTP Methods	• `POST`
URL	`https://HOST_OR_IP/api/v1/VideoClip`

Request JSON	`{`
	`"reqAction":21,`
	`"camera":"CAMERA_NAME",`
	`"audio":1,`
	`"begin":1584734700,`
	`"end":1584734900,`
	`"height":1440,`
	`"width":2560,`
	`"fps":30,`
	`"codec":"h265",`
	`"gop":60,`
	`"qvalue":50,`
	`"bitrate":1000,`
	`"osdFormat":"24h",`
	`"osdPosition":"Top-Left"`
	`}`
	
	where attributes are:
	• `audio` — Optional. Whether or not to include audio. Valid values are `0` (no audio) or `1` (include audio, if any).
	• `begin` — Timestamp for the start of the video clip in UTC. If `begin` is the same as `end`, then instead of a movie, it downloads an image.

- `bitrate` — Optional. Quality, in bits per second.

  

  `bitrate` is ignored if `qvalue` is used.

- `camera` — Camera name, as defined on FortiRecorder.
- `codec` — Optional. Encoding type. Valid values are either `h264` or `h265`.
- `end` — Timestamp for the end of the video clip in UTC. If `begin` is the same as `end`, then instead of a movie, it downloads an image.
- `fps` — Optional. Smoothness of motion in frames per second. A common starting point is 20 or 30.
- `gop` — Optional. Number of frames for each reference frame in the group of pictures. Default value may vary by camera model.
- `height` — Optional. Height in pixels.
- `osdFormat` — Optional. Format of the on-screen display of the timestamp in the video, with the clock as either `"12h"` or `"24h"`.
- `osdPosition` — Optional. Position of the on-screen display of the timestamp in the video. Valid values are `"Top-Left"`, `"Top-Right"`, `"Bottom-Left"`, or `"Bottom-Right"`.
- `qvalue` — Optional.Quality. Valid values are from 10 to 50. A reasonable starting point is 32. The greater the `qvalue`, the less is the bitrate and video quality.

  

  `bitrate` is ignored if `qvalue` is used.

- `reqAction` — Action. Takes precedence over the method in the HTTP header. Required value is `21` (process and download the file).
- `width` — Optional. Width in pixels.

Example: Download an MP4 video

```
curl -X POST -H "Content-Type:application/json" -d "{\"reqAction\":21,\"camera\":\"CAMERA_
NAME\",\"begin\":1762206615,\"end\":1762206675}" -b cookie.txt https://HOST_OR_
IP/api/v1/VideoClip/ -o VIDEO_NAME.mp4
```

Example: Download a JPG snapshot image

```
curl -X POST -H "Content-Type:application/json" -d "{\"reqAction\" :21,\"camera\":\"CAMERA_
NAME\",\"begin\":1735754400,\"end\":1735754400}" -b cookie.txt https://HOST_OR_
IP/api/v1/VideoClip/ -o IMAGE_NAME.jpg
```

Streaming video

In addition to using the REST API endpoint to download video clips and snapshots, you can also use a video player that supports RTSP (RFC 2326), such as VLC, to connect to FortiRecorder and play live or previously recorded video.

URL schema

Use a URL like:

```
rtsp://HOST_OR_IP:8554/cam=CAMERA_NAME
```

where:

- `CAMERA_NAME` — Unique name of the camera in the FortiRecorder configuration.
- `TIMESTAMP` — Start time of the video in UTC in seconds since January 1, 1970, UTC (also called the Unix epoch). Convert the local time zone on FortiRecorder to UTC, and then convert to the Unix epoch relative timestamp. If you use the URL without this parameter, then FortiRecorder starts the video at 60 seconds before the current time.

Authentication

For authentication, see Authentication for video clip service on page 15.

RTSP commands

FortiRecorder will respond to the following RTSP requests:

- `OPTIONS` — Gets the list of supported commands.
- `DESCRIBE` — Gets the SDP describing the stream.
- `SETUP` — Initializes RTP senders or receivers.
- `PLAY` — Starts playback, seeks, resumes after pause.
- `PAUSE` — Pauses the stream of RTP packets. Resume playback with the `PLAY` command (no range).
- `TEARDOWN` - Stops the session.

PLAY

If you do not specify a playback time, the default is 60 seconds before the current time.

If you specify a playback time, FortiRecorder will start playing recorded video with that timestamp, or, if no recorded video exists for that time, a nearby time instead.

To seek, issue a similar `PLAY` request with the new playback time specified. To resume after pausing, issue a `PLAY` request without the playback time specified.

The playback time is specified using the time range header with `clock` in absolute units with ISO 8601 timestamps, using UTC.

For example, to request video at 2:30 PM on October 31, 2025, the header would be:

```
Range: clock=20251031T143000-
```

Recorded time of frames

During playback, gaps between videos will be skipped. To communicate the recorded frame time to the RTSP client, each frame's NTP timestamp is embedded as an RTP extension. See section 6.4 of the ONVIF Streaming Specification.

Example: VLC video player

VLC does not currently support absolute times, so use the URL schema which specifies the start time.

VLC will set the playback time using the normal playback time (`npt`) units in the time range header:

```
Range: npt=0-
```

On the first request with a `PLAY` command, this will be the offset from the start time (usually zero). To resume after pausing, VLC issues a `PLAY` command without the `Range:` specified. To seek, VLC issues a `PLAY` command again, but with the `Range:`.

To allow seeking outside of the default range, FortiRecorder tries to re-center the slider by adjusting the returned range:

```
Range: npt=1800-3600
```

For example:

```
OPTIONS rtsp://172.20.131.56:8554/cam=cam1 RTSP/1.0
CSeq: 2
User-Agent: LibVLC/3.0.18 (LIVE555 Streaming Media v2016.11.28)

RTSP/1.0 200 OK
CSeq: 2
Public: OPTIONS, DESCRIBE, SETUP, TEARDOWN, PLAY, PAUSE

DESCRIBE rtsp://172.20.131.56:8554/cam=cam1 RTSP/1.0
CSeq: 3
User-Agent: LibVLC/3.0.18 (LIVE555 Streaming Media v2016.11.28)
```

```
Accept: application/sdp

RTSP/1.0 401 Unauthorized
CSeq: 3
WWW-Authenticate: Digest realm="FortiRecorder", nonce="bda3ff5dd24b4b7175cd7420af971b16"

DESCRIBE rtsp://172.20.131.56:8554/cam=cam1 RTSP/1.0
CSeq: 4
Authorization: Digest username="test", realm="FortiRecorder",
      nonce="bda3ff5dd24b4b7175cd7420af971b16", uri="rtsp://172.20.131.56:8554/cam=cam1",
      response="2d602b41a77a93e0c606f17d79638faf"
User-Agent: LibVLC/3.0.18 (LIVE555 Streaming Media v2016.11.28)
Accept: application/sdp

RTSP/1.0 200 OK
CSeq: 4
Content-Base: rtsp://172.20.131.56:8554/cam=cam1
Content-Length: 249
v=0
o=- 0 0 IN IP4 127.0.0.1
t=0 0
a=tool:FortiRecorder
m=video 0 RTP/AVP 96
a=rtpmap:96 H264/90000
a=fmtp:96 packetization-mode=1; sprop-parameter-sets=Z0IAKOkAoAQNcSAA27oAM3+YA2IEJQ==,aM4xUg==;
      profile-level-id=420028
a=control:streamid=0

SETUP rtsp://172.20.131.56:8554/cam=cam1/streamid=0 RTSP/1.0
CSeq: 5
Authorization: Digest username="test", realm="FortiRecorder",
      nonce="bda3ff5dd24b4b7175cd7420af971b16", uri="rtsp://172.20.131.56:8554/cam=cam1",
      response="e8c10f302cafb5d7133766ad088deb72"
User-Agent: LibVLC/3.0.18 (LIVE555 Streaming Media v2016.11.28)
Transport: RTP/AVP;unicast;client_port=65070-65071

RTSP/1.0 200 OK
CSeq: 5
Transport: RTP/AVP;unicast;client_port=65070-65071;server_port=20028-20029;ssrc=19d26fa8
Session: kQ16P8x1hF0qgrGp

PLAY rtsp://172.20.131.56:8554/cam=cam1 RTSP/1.0
CSeq: 6
Authorization: Digest username="test", realm="FortiRecorder",
      nonce="bda3ff5dd24b4b7175cd7420af971b16", uri="rtsp://172.20.131.56:8554/cam=cam1",
      response="9e62f34bde86046e32e7894476458635"
User-Agent: LibVLC/3.0.18 (LIVE555 Streaming Media v2016.11.28)
Session: kQ16P8x1hF0qgrGp
Range: npt=0.000-

RTSP/1.0 200 OK
Range: npt=3540-3600
CSeq: 6
RTP-Info: url=rtsp://172.20.131.56:8554/cam=cam1/streamid=0;seq=1;rtptime=2616060667
Session: kQ16P8x1hF0qgrGp

PAUSE rtsp://172.20.131.56:8554/cam=cam1 RTSP/1.0
CSeq: 7
```

```
Authorization: Digest username="test", realm="FortiRecorder",
      nonce="bda3ff5dd24b4b7175cd7420af971b16", uri="rtsp://172.20.131.56:8554/cam=cam1",
      response="907fe83beeb1ed15301f9dd8b79f049b"
User-Agent: LibVLC/3.0.18 (LIVE555 Streaming Media v2016.11.28)
Session: kQ16P8x1hF0qgrGp

RTSP/1.0 200 OK
CSeq: 7
Session: kQ16P8x1hF0qgrGp

PLAY rtsp://172.20.131.56:8554/cam=cam1 RTSP/1.0
CSeq: 8
Authorization: Digest username="test", realm="FortiRecorder",
      nonce="bda3ff5dd24b4b7175cd7420af971b16", uri="rtsp://172.20.131.56:8554/cam=cam1",
      response="9e62f34bde86046e32e7894476458635"
User-Agent: LibVLC/3.0.18 (LIVE555 Streaming Media v2016.11.28)
Session: kQ16P8x1hF0qgrGp
Range: npt=2818.440-

RTSP/1.0 200 OK
Range: npt=2812-3600
CSeq: 8
RTP-Info: url=rtsp://172.20.131.56:8554/cam=cam1/streamid=0;seq=534;rtptime=2616574891
Session: kQ16P8x1hF0qgrGp

TEARDOWN rtsp://172.20.131.56:8554/cam=cam1 RTSP/1.0
CSeq: 9
Authorization: Digest username="test", realm="FortiRecorder",
      nonce="bda3ff5dd24b4b7175cd7420af971b16", uri="rtsp://172.20.131.56:8554/cam=cam1",
      response="c3b4afc480000393849b5a552fbd3722"
User-Agent: LibVLC/3.0.18 (LIVE555 Streaming Media v2016.11.28)
Session: kQ16P8x1hF0qgrGp

RTSP/1.0 200 OK
CSeq: 9
Session: kQ16P8x1hF0qgrGp
```

Example: Generic video player

```
OPTIONS rtsp://172.20.131.56:8554/cam=cam1 RTSP/1.0
CSeq: 0

RTSP/1.0 200 OK
CSeq: 0
Public: OPTIONS, DESCRIBE, SETUP, TEARDOWN, PLAY, PAUSE

DESCRIBE rtsp://172.20.131.56:8554/cam=cam1 RTSP/1.0
CSeq: 1

RTSP/1.0 401 Unauthorized
CSeq: 1
WWW-Authenticate: Digest realm="FortiRecorder", nonce="427e06651ff20c8c4edb0d2b7425062c"

DESCRIBE rtsp://172.20.131.56:8554/cam=cam1 RTSP/1.0
```

```
CSeq: 2
Authorization: Digest username="test", realm="FortiRecorder",
      nonce="427e06651ff20c8c4edb0d2b7425062c", uri="rtsp://172.20.131.56:8554/cam=cam1",
      response="a085f7fac48b09f98a59bd6978d914ae"

RTSP/1.0 200 OK
CSeq: 2
Content-Base: rtsp://172.20.131.56:8554/cam=cam1
Content-Length: 249
v=0
o=- 0 0 IN IP4 127.0.0.1
t=0 0
a=tool:FortiRecorder
m=video 0 RTP/AVP 96
a=rtpmap:96 H264/90000
a=fmtp:96 packetization-mode=1; sprop-parameter-sets=Z0IAKOkAoAQNcSAA27oAM3+YA2IEJQ==,aM4xUg==;
      profile-level-id=420028
a=control:streamid=0

SETUP rtsp://172.20.131.56:8554/cam=cam1 RTSP/1.0
CSeq: 3
Authorization: Digest username="test", realm="FortiRecorder",
      nonce="427e06651ff20c8c4edb0d2b7425062c", uri="rtsp://172.20.131.56:8554/cam=cam1",
      response="7bc23478b463d1e3e130eae2f6155ec4"
Transport: RTP/AVP;unicast;client_port=8382-8383

RTSP/1.0 200 OK
CSeq: 3
Transport: RTP/AVP;unicast;client_port=8382-8383;server_port=20942-20943;ssrc=19d26fa8
Session: rQ4NTXL27tda9abD

PLAY rtsp://172.20.131.56:8554/cam=cam1 RTSP/1.0
CSeq: 4
Session: rQ4NTXL27tda9abD
Authorization: Digest username="test", realm="FortiRecorder",
      nonce="427e06651ff20c8c4edb0d2b7425062c", uri="rtsp://172.20.131.56:8554/cam=cam1",
      response="91298a6df73132a544c85adb9082c3d0"
Range: clock=20230712T191144-

RTSP/1.0 200 OK
Range: clock=20230712T191144-
CSeq: 4
RTP-Info: url=rtsp://172.20.131.56:8554/cam=cam1/streamid=0;seq=1;rtptime=2519022532
Session: rQ4NTXL27tda9abD

PLAY rtsp://172.20.131.56:8554/cam=cam1 RTSP/1.0
CSeq: 5
Session: rQ4NTXL27tda9abD
Authorization: Digest username="test", realm="FortiRecorder",
      nonce="427e06651ff20c8c4edb0d2b7425062c", uri="rtsp://172.20.131.56:8554/cam=cam1",
      response="91298a6df73132a544c85adb9082c3d0"
Range: clock=20230712T191154-

RTSP/1.0 200 OK
Range: clock=20230712T191154-
CSeq: 5
RTP-Info: url=rtsp://172.20.131.56:8554/cam=cam1/streamid=0;seq=92;rtptime=2519072025
Session: rQ4NTXL27tda9abD
```

```
TEARDOWN rtsp://172.20.131.56:8554/cam=cam1 RTSP/1.0
CSeq: 6
Session: rQ4NTXL27tda9abD
Authorization: Digest username="test", realm="FortiRecorder",
      nonce="427e06651ff20c8c4edb0d2b7425062c", uri="rtsp://172.20.131.56:8554/cam=cam1",
      response="00dd6110ab53bb2400b4601700eb286b"

RTSP/1.0 200 OK
CSeq: 6
Session: rQ4NTXL27tda9abD
```

www.fortinet.com