

CompleteView REST Cameras Interface

The CompleteView [REST](#) Cameras interface is currently supported via both HTTP and HTTPS as enabled by the embedded Web Server and Secure Web Server respectively. It uses the existing CompleteView HTTP authentication scheme, which means that credentials are conveyed via a Basic authentication header. It returns textual resource information in multiple formats, currently only JSON and HTML. Similarly, media is returned in multiple formats, currently MJPEG, JPEG, MPEG4 and H.264 for video, and G.711 uLaw for audio. This interface currently takes in all parameters via the request URI string and not in the HTTP request body in a format such as JSON.

The REST interface processes requests and transmits media on CompleteView Server's configurable Web Server and / or Secure Web Server TCP port(s). REST is a programmatic communications interface, not a user interface. However, the design goals of this interface include adherence to the REST "spirit," allowing it to be easily used from a browser for experimentation and testing purposes.

Current practice on the web is to use the HTTP GET method for just about everything. However, REST is stricter about method semantics. For example, it makes a distinction between requesting a *representation* of a resource -a GET- and requesting an *update* to a resource -a PUT. Some HTTP frameworks only support GET, and most browsers send a GET request when the user enters a URI into the location bar. The workaround in our REST interface is to allow the requester to override the HTTP method that his or her framework/browser generates with an optional, method query-string parameter. For example, these two requests have the same semantics from the perspective of our REST interface:

```
PUT /cameras/9/position/2
```

```
GET /cameras/9/position/2?method=put
```

This is the only parameter available for all requests, although it is probably only useful for those that change resource state, such as for PTZ control.

The following sections describe the syntax and semantics of the CompleteView REST interface, or API. It is sometimes referred to as the REST *Cameras* interface to distinguish it from any other that may be added in the future, such as a REST administrative interface.

Support Information

For any questions or to report any bugs discovered in the CompleteView REST interface, please email apisupport@salientsys.com.

Authenticating Requests

The REST interface requires requests to include CompleteView user credentials transmitted in an HTTP Basic authentication header as described [here](#).

Examples

Request

The following GET request includes the authorization header with a username of “admin” and an empty (“”) password.

```
GET /cameras/1/clip?start=2015-06-23T12:12:12Z HTTP/1.1\r\n
Host: 192.168.103.58:4502\r\n
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:38.0) Gecko/20100101
Firefox/38.0\r\n
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8\r\n
Accept-Language: en-US,en;q=0.5\r\n
Accept-Encoding: gzip, deflate\r\n
Connection: keep-alive\r\n
Authorization: Basic YWRtaW46\r\n
\r\n
```

GET /cameras

Gets server information and a list of configured cameras and their associated attributes such as user permissions (based on the user that has been authenticated with the REST interface) and camera capabilities.

Resource Information

Response formats	JSON, JSON-P, HTML
Requires authentication?	Yes

Parameters

Name	Required	Description
accept	No	Default: text/html Supported values: application/json, application/javascript, text/html Used to declare the type of response encoding you would like.
callback	No	If <i>callback</i> is specified, <i>accept</i> must be either application/javascript or text/html. Used in order to complete a JSON-P request by wrapping the response in a javascript function call.

Enumerated Values

Property path	Values
<i>license</i> - server object	cvlite, standard, enterprise, trial, cloud, unknown
<i>state</i> - each camera object	failed, online, disabled
<i>ipStreamType</i> - each camera object	mjpeg, h263, mpeg4, h264, unknown

Notes

- Dimensions are for the current, live video stream. If the camera is not online at the time of the request, dimensions are not included.
- Recorded video may have dimensions that differ from what is currently being streamed from the camera.

Examples

Request

```
GET /cameras?accept=application/json
```

Response

```
{
  "servers": [
    {
      "address": "StephenDev-PC",
      "name": "StephenDev-PC1",
      "serialNumber": "0d763f7ce0d8db8055189beaaab5cc11",
      "license": "enterprise",
      "dptzPlayback": true,
      "product": {
        "name": "CompleteView",
        "version": "4.5.0.96"
      },
      "utcOffset": "-05:00",
      "cameras": [
        {
          "id": 1,
          "name": "AXIS Q1602 at 192.168.103.203",
          "state": "online",
          "audioEnabled": false,
          "lightAvailable": false,
          "disableStreamProcessing": false,
          "ipStreamType": "h264",
          "permissions": {
            "live": true,
            "recorded": true,
            "exportable": true,
            "allow_snapshot": true,
            "allow_light_control": true,
            "audio": true,
            "quickTrackCamera": false
          },
          "dimensions": {
            "width": 640,
            "height": 480
          }
        }
      ]
    }
  ]
}
```

```
    },  
    "ptz": {  
      "pan": true,  
      "tilt": true,  
      "zoom": true,  
      "reset": true,  
      "presets": 10,  
      "digital": true,  
      "presetNames": [  
        "",  
        "",  
        "",  
        "",  
        "",  
        "",  
        "",  
        "",  
        "",  
        ""  
      ]  
    }  
  },  
  {  
    "id": 2,  
    "name": "Axis Q6044 - 720P",  
    "state": "online",  
    "audioEnabled": false,  
    "lightAvailable": false,  
    "disableStreamProcessing": false,  
    "ipStreamType": "h264",  
    "permissions": {  
      "live": true,  
      "recorded": true,  
      "exportable": true,  
      "allow_snapshot": true,  
      "allow_light_control": true,  
      "audio": true,  
      "quickTrackCamera": false  
    },  
    "dimensions": {  
      "width": 1280,  
      "height": 720  
    },  
    "ptz": {  
      "pan": true,  
      "tilt": true,  
      "zoom": true,  
      "iris": true,  
      "focus": true,  
      "autoIris": true,  
    }  
  }  
}
```

```

        "autoFocus": true,
        "presets": 10,
        "digital": false,
        "presetNames": [
            "Main Entrance",
            "Loading Dock",
            "Parking Lot",
            "",
            "",
            "",
            "",
            "",
            "",
            ""
        ]
    }
}
]
}
]
}
}

```

Request

```

GET
/cameras?accept=application/javascript&callback=jQuery16203128881240285475_1
322766825181&_=1322768532287

```

Response

```

jQuery16203128881240285475_1322766825181({
  "servers": [
    {
      "address": "StephenDev-PC",
      "name": "StephenDev-PC1",
      "serialNumber": "0d763f7ce0d8db8055189beaaab5cc11",
      "license": "enterprise",
      "dptzPlayback": true,
      "product": {
        "name": "CompleteView",
        "version": "4.5.0.96"
      },
      "utcOffset": "-05:00",
      "cameras": [
        {
          "id": 1,
          "name": "AXIS Q1602 at 192.168.103.203",
          "state": "online",
          "audioEnabled": false,
          "lightAvailable": false,
          "disableStreamProcessing": false,

```

```
    "ipStreamType": "h264",
    "permissions": {
      "live": true,
      "recorded": true,
      "exportable": true,
      "allow_snapshot": true,
      "allow_light_control": true,
      "audio": true,
      "quickTrackCamera": false
    },
    "dimensions": {
      "width": 640,
      "height": 480
    },
    "ptz": {
      "pan": true,
      "tilt": true,
      "zoom": true,
      "reset": true,
      "presets": 10,
      "digital": true,
      "presetNames": [
        "",
        "",
        "",
        "",
        "",
        "",
        "",
        "",
        "",
        ""
      ]
    }
  },
  {
    "id": 2,
    "name": "Axis Q6044 - 720P",
    "state": "online",
    "audioEnabled": false,
    "lightAvailable": false,
    "disableStreamProcessing": false,
    "ipStreamType": "h264",
    "permissions": {
      "live": true,
      "recorded": true,
      "exportable": true,
      "allow_snapshot": true,
      "allow_light_control": true,
      "audio": true,
```


Parameters

Name	Required	Description
camera-id	Yes	Integral camera ID, from 1 to the number of cameras on this server. This parameter should be a part of the request URI path and not the list of query parameters.
start	Yes	UTC start date / time string as defined in RFC 3339 of the clip to receive.
stop	No	Default: <i>start</i> date plus 60 seconds UTC stop date / time string as defined in RFC 3339 of the clip to receive.
accept	No	Default: video/x-motion-jpeg Supported values: video/h264, video/mpeg4, video/x-motion-jpeg, audio/basic Used to declare the media type you would like to receive.
quality	No	Default: 100 Supported values: 0 - 100 Lower values corresponds to higher video compression and vice versa.
bitrate	No	Default: -1 (no change to recorded bitrate) Supported values: 0 to infinity bits per second Used to specify the desired bitrate the encoder should attempt to transmit video at. This is a suggested value as certain levels of compression cannot be reached.
keyrate	No	Default: 8 Supported values: 1 to 90 Keyrate or GOP size decides how often an I frame should be sent.
autoplay	No	Default: 1 (enable auto play) Supported values: 0 or 1 Determines whether or not the media should begin playing as soon as the request is received. If set to 0, the first frame or GOP will be sent so that an initial image can be rendered. Video or audio playback can be started using the same VCR commands used for regular recorded media.
transcode	No	Default: 1 (enable transcoding) Supported values: 0 or 1 Determines whether the clip should be transcoded or the native recorded format should be streamed.

Notes

- The TCP socket connection is closed when the stream has reached its end point.
- The *accept*, *quality*, *bitrate* and *keyrate* parameters only have an effect if transcoding is enabled.
- *stop* must be later in time than *start*.
- The *quality*, *bitrate* and *keyrate* parameters are only relevant for video streaming.
- The initial HTTP response headers will include some useful information regarding the media stream. For video streams, they include video height and width, if the video stream has audio associated with it, the media type (which if transcoding is turned off, will be the native format), the Connection ID (used for Digital PTZ), the average time per frame which is used to calculate the frame rate and a cookie that should be recorded and sent with VCR control requests as well as Digital PTZ requests. For audio streams, they include the number of audio channels, the bits per sample and the sample rate.
- Not every multipart section will include media, recipients should always check the Content-Type and / or Content-Length.

- Multipart sections that do include media will include additional headers such as Pre-roll (video), Sync-Point (video), various timestamps, and whether or not the stream is at the end.
- There is another type of multipart section that MAY be interlaced with media sections to notify a recipient when the server has to switch AVI files. During this process, it is possible that the media type or presence of audio has changed. This should be handled by client code.

Examples

Video - Request

```
GET /cameras/1/clip?start=2015-06-23T12:12:12Z
```

Response

The below response includes the possible multipart sections that MAY come across the wire in this particular request scenario. Generally, there are one or more media sections, zero or more transition sections, and one or more end of stream notifications (depending on whether or not the position of the stream has been changed and the stream ends more than once).

```
HTTP/1.1 200 OK\r\n
Date: Tue, 23 Jun 2015 20:44:39 GMT\r\n
Connection: close\r\n
Cache-Control: public, no-store\r\n
Expires: Tue, 23 Jun 2016 20:44:39 GMT\r\n
Content-Type: multipart/x-mixed-replace;boundary=myboundary\r\n
Set-Cookie: stream=38740441; version=1\r\n
Media-Type: video/x-motion-jpeg\r\n
Has-Audio: false\r\n
Connection-ID: 100\r\n
Width: 1400\r\n
Height: 1050\r\n
Avg-Time-Per-Frame: 333333\r\n
\r\n
--myboundary\r\n
Content-Type: image/jpeg\r\n
Sync-Point: yes\r\n
Pre-roll: no\r\n
Media-Start: invalid\r\n
Media-End: invalid\r\n
Stream-Start: 131184\r\n
Stream-End: 464517\r\n
Content-Length: 99609\r\n
X-Date: 2015-06-23T12:12:12.013Z\r\n
End-of-Stream: no\r\n
\r\n
{video-frame}
\r\n
--myboundary\r\n
Media-Type: video/x-motion-jpeg\r\n
Has-Audio: false\r\n
Content-Length: 0\r\n
```

```
\r\n
--myboundary\r\n
Content-Type: image/jpeg\r\n
Sync-Point: no\r\n
Pre-roll: no\r\n
Media-Start: invalid\r\n
Media-End: invalid\r\n
Stream-Start: invalid\r\n
Stream-End: invalid\r\n
Content-Length: 0\r\n
X-Date: 0\r\n
End-of-Stream: yes\r\n
\r\n
```

Audio - Request

Note that the cookie that was sent with the video stream was sent with the audio request.

```
GET /cameras/1/clip?accept=audio/basic&start=2015-06-25T20:11:55.199Z
HTTP/1.0\r\n
Host: 192.168.103.58:4502\r\n
User-Agent: CompleteView Web Client 4.5.1.10\r\n
Authorization: Basic YWRtaW46\r\n
Cookie: stream=38751481\r\n
\r\n
```

Response

There are no transition multipart sections in the audio stream as there are with the video stream.

```
HTTP/1.1 200 OK\r\n
Date: Thu, 25 Jun 2015 20:12:42 GMT\r\n
Connection: close\r\n
Cache-Control: public, no-store\r\n
Expires: Thu, 25 Jun 2016 20:12:42 GMT\r\n
Content-Type: multipart/x-mixed-replace;boundary=myboundary\r\n
Audio-Num-Channels: 1\r\n
Audio-Sample-Rate: 8000\r\n
Audio-Bits-Per-Sample: 8\r\n
Set-Cookie: stream=38751481; version=1\r\n
\r\n
--myboundary\r\n
Content-Type: audio/basic\r\n
Media-Start: invalid\r\n
Media-End: invalid\r\n
Stream-Start: 0\r\n
Stream-End: 320000\r\n
Content-Length: 256\r\n
End-of-Stream: no\r\n
\r\n
{audio-samples}
\r\n
```

```
--myboundary\r\n
Content-Type: audio/basic\r\n
Media-Start: invalid\r\n
Media-End: invalid\r\n
Stream-Start: 320000\r\n
Stream-End: 640000\r\n
Content-Length: 256\r\n
End-of-Stream: no\r\n
\r\n
{audio-samples}
\r\n
```

GET /cameras/:camera-id/clips

Gets a list of video clip recordings available for a specified time period.

Resource Information

Response formats	JSON, JSON-P, HTML
Requires authentication?	Yes
Permissions required	Playback

Parameters

Name	Required	Description
camera-id	Yes	Integral camera ID, from 1 to the number of cameras on this server. This parameter should be a part of the request URI path and not the list of query parameters.
start	Yes	UTC start date / time string as defined in RFC 3339 of the time period you would like to search.
stop	Yes	UTC stop date / time string as defined in RFC 3339 of the time period you would like to search.
accept	No	Default: text/html Supported values: application/json, application/javascript, text/html Used to declare the type of response encoding you would like.
callback	No	If <i>callback</i> is specified, <i>accept</i> must be either application/javascript or text/html. Used in order to complete a JSON-P request by wrapping the response in a javascript function call.
categories	No	Default: scheduled,motion,alarm Supported values: Must be a comma delimited list of scheduled, motion or alarm Used to specify which clip categories you would like to search.
limit	No	Default: 500 Supported values: 0 to 500 The number of clip results you would like to limit your search to.
skip	No	Default: 0 Supported values: 0 to 500 The number of clip results you would like to skip before results are returned.

Notes

- *stop* must be later in time than *start*.

Examples

Request

```
GET
/cameras/1/clips?accept=application/json&start=2015-06-23T21:44:12Z&stop=2015-06-23T21:46:12Z
```

Response

```
{
  "clips": [
    {
      "begin": "2015-06-23T21:44:51Z",
      "end": "2015-06-23T21:45:00Z",
      "xBegin": "2015-06-23T21:44:51.957Z",
      "xEnd": "2015-06-23T21:45:00.127Z",
      "size": 80,
      "type": "motion"
    },
    {
      "begin": "2015-06-23T21:45:00Z",
      "end": "2015-06-23T21:45:19Z",
      "xBegin": "2015-06-23T21:45:00.130Z",
      "xEnd": "2015-06-23T21:45:19.153Z",
      "size": 128,
      "type": "scheduled"
    },
    {
      "begin": "2015-06-23T21:45:19Z",
      "end": "2015-06-23T21:45:27Z",
      "xBegin": "2015-06-23T21:45:19.157Z",
      "xEnd": "2015-06-23T21:45:27.107Z",
      "size": 73,
      "type": "motion",
      "description": "StartMotionTimedReason_Click"
    }
  ]
}
```

GET /cameras/:camera-id/videofiles

Gets a list of video files available for a specified time period. This differs from the [clips](#) call in that every underlying AVI file is returned instead of being coalesced together.

Resource Information

Response formats	JSON, JSON-P, HTML
Requires authentication?	Yes
Permissions required	Playback

Parameters

Name	Required	Description
camera-id	Yes	Integral camera ID, from 1 to the number of cameras on this server. This parameter should be a part of the request URI path and not the list of query parameters.
start	Yes	UTC start date / time string as defined in RFC 3339 of the time period you would like to search.
stop	Yes	UTC stop date / time string as defined in RFC 3339 of the time period you would like to search.
accept	No	Default: text/html Supported values: application/json, application/javascript, text/html Used to declare the type of response encoding you would like.
callback	No	If <i>callback</i> is specified, <i>accept</i> must be either application/javascript or text/html. Used in order to complete a JSON-P request by wrapping the response in a javascript function call.
categories	No	Default: scheduled,motion,alarm Supported values: Must be a comma delimited list of scheduled, motion or alarm Used to specify which video file categories you would like to search.
limit	No	Default: 500 Supported values: 0 to 500 The number of video file results you would like to limit your search to.
skip	No	Default: 0 Supported values: 0 to 500 The number of video file results you would like to skip before results are returned.

Notes

- *stop* must be later in time than *start*.

Examples

Request

```
GET
```

/cameras/1/videofiles?accept=application/json&start=2017-01-30T00:00:00Z&stop=2017-01-30T23:59:59Z

Response

```
{
  "video_files": [
    {
      "id": 1,
      "fileStart": "2017-01-30T16:50:28.843Z",
      "fileEnd": "2017-01-30T16:53:29.847Z",
      "size": 29295,
      "type": "scheduled"
    },
    {
      "id": 5,
      "fileStart": "2017-01-30T16:53:29.860Z",
      "fileEnd": "2017-01-30T16:56:30.847Z",
      "size": 26399,
      "type": "scheduled"
    },
    {
      "id": 9,
      "fileStart": "2017-01-30T16:56:30.857Z",
      "fileEnd": "2017-01-30T16:59:31.850Z",
      "size": 24694,
      "type": "scheduled"
    }
  ]
}
```

GET /cameras/:camera-id/videofilesdownload

Retrieves an individual AVI recording file. This is intended to be called with the fileStart and fileEnd time received from the [video files](#) call.

Resource Information

Response formats	AVI
Requires authentication?	Yes
Permissions required	Playback

Parameters

Name	Required	Description
camera-id	Yes	Integral camera ID, from 1 to the number of cameras on this server. This parameter should be a part of the request URI path and not the list of query parameters.
start	Yes	UTC start date / time string as defined in RFC 3339 of the video file you would like to retrieve.
stop	Yes	UTC stop date / time string as defined in RFC 3339 of the video file you would like to retrieve.

Notes

- *stop* must be later in time than *start*.

Examples

Request

```
GET
/cameras/1/videofilesdownload?accept=application/json&start=2017-01-30T16:50:28.843Z&stop=2017-01-30T16:53:29.847Z
```

Response

```
HTTP/1.1 200 OK\r\n
Content-Type: application/octet-stream\r\n
Last-Modified: Mon, 30 Jan 2017 16:50:28 GMT\r\n
Content-Length: 29998722\r\n
Connection: close\r\n
Cache-Control: max-age=86400\r\n
Expires: Tue, 31 Jan 18:11:36 GMT\r\n
\r\n
{file-contents}
```

GET /cameras/:camera-id/presetnames

Gets a list of preset names for a particular camera.

Resource Information

Response formats	JSON, JSON-P, HTML
Requires authentication?	Yes
Permissions required	PTZ

Parameters

Name	Required	Description
camera-id	Yes	Integral camera ID, from 1 to the number of cameras on this server. This parameter should be a part of the request URI path and not the list of query parameters.
accept	No	Default: text/html Supported values: application/json, application/javascript, text/html Used to declare the type of response encoding you would like.
callback	No	If <i>callback</i> is specified, <i>accept</i> must be either application/javascript or text/html. Used in order to complete a JSON-P request by wrapping the response in a javascript function call.

Examples

Request

```
GET /cameras/1/presetnames?accept=application/json
```

Response

```
{
  "presetNames": [
    "Main Entrance",
    "Loading Dock",
    "Parking Lot",
    "",
    "",
    "",
    "",
    "",
    "",
    "",
    ""
  ]
}
```

PUT /cameras/:camera-id/light

Changes the brightness of a light on or attached to a supported camera.

Resource Information

Response formats	HTML
Requires authentication?	Yes
Permissions required	Light

Parameters

Name	Required	Description
camera-id	Yes	Integral camera ID, from 1 to the number of cameras on this server. This parameter should be a part of the request URI path and not the list of query parameters.
lightvalue	Yes	Supported values: 0 to 100 The brightness of the light to set, where 0 is completely off and 100 is as bright as possible.

Notes

- Currently only supported for Axis cameras with built in lights.

Examples

Request

```
PUT /cameras/1/light?lightvalue=100
```

GET /cameras/:camera-id/media

Gets a live multipart stream of video or audio or a still image of live or recorded video. This endpoint is also used to control playback via VCR controls.

Resource Information

Response formats	Multipart content with either MJPEG, H.264 or MPEG4 video frames, or G.711 uLaw audio samples. JPEG
Requires authentication?	Yes
Permissions required (live viewing)	Live, Audio (if requesting audio)
Permissions required (recorded viewing)	Playback, Audio (if requesting audio)

Parameters

Name	Required	Description
camera-id	Yes	Integral camera ID, from 1 to the number of cameras on this server. This parameter should be a part of the request URI path and not the list of query parameters.
accept	No	Default: video/x-motion-jpeg Supported values: video/h264, video/mpeg4, video/x-motion-jpeg, audio/basic, image/jpeg Used to declare the media type you would like to receive. Will either result in a stream of audio or video, or a single JPEG result.
fps	No	Default: 4 Supported values: 1 to 30 The frames per second of the video to stream, however, it will not exceed what is being natively streamed from the camera.

Name	Required	Description
start	No	Default: Not set (indicating live viewing) UTC start date / time string as defined in RFC 3339 of the recorded clip to stream.
stop	No	Default: <i>start</i> date plus 60 seconds UTC stop date / time string as defined in RFC 3339 of the recorded clip to receive.
position	No	Default: Not set UTC date / time string as defined in RFC 3339 of the point within the clip that you would like to seek to. If the <i>start</i> parameter is specified, then this parameter is used to tell playback to initially start at this position, while still allowing the entire clip to be played. If <i>position</i> is specified in the absence of the <i>start</i> parameter, then it will be used to perform a seek operation. The parameter is not allowed to be before the <i>start</i> time or after the <i>stop</i> time. Calls to perform a seek will require that the appropriate Cookie be included in the request.
quality	No	Default: 100 Supported values: 0 - 100 Lower values corresponds to higher video compression and vice versa.
bitrate	No	Default: -1 (no change to native bitrate) Supported values: 0 to infinity bits per second Used to specify the desired bitrate the encoder should attempt to transmit video at. This is a suggested value as certain levels of compression cannot be reached.
keyrate	No	Default: 8 Supported values: 1 to 90 Keyrate or GOP size decides how often an I frame should be sent.
width	No	Default: -1 (send the native resolution) Supported values: 16 to 4096 The pixel width of the image to send. The image that is sent may not be exactly what is requested as the image is locked to its native aspect ratio. Use the Width and Height response headers to get the exact values. If <i>width</i> is specified, <i>height</i> must also be specified.
height	No	Default: -1 (send the native resolution) Supported values: 16 to 4096 The pixel height of the image to send. The image that is sent may not be exactly what is requested as the image is locked to its native aspect ratio. Use the Width and Height response headers to get the exact values. If <i>height</i> is specified, <i>width</i> must also be specified.
action	No	Default: Not set Supported values: play, pause, first, last, forward, back VCR control options for recorded media streams. <i>first</i> skips to the first frame of the clip, <i>last</i> skips to the last frame of the clip, <i>forward</i> steps forward one frame from the current position and <i>back</i> steps backwards one frame from the current position. Calls using this parameter will require that the appropriate Cookie be included in the request.
autoplay	No	Default: 1 (enable auto play) Supported values: 0 or 1 Determines whether or not the media should begin playing as soon as the request is received. If set to 0, the first frame or GOP will be sent so that an initial image can be rendered. Video or audio playback can be started using the VCR commands.

Name	Required	Description
allowupscale	No	Default: 1 (allow up scaling) Supported values: 0 or 1 Specifies whether or not the server should cap images to the native resolution.
speed	No	Default: -1 (not set) Supported values: Floating point values from 0 to infinity Specifies the speed at which recorded media should be played back.

Notes

- *stop* must be later in time than *start*.
- The *fps*, *quality*, *bitrate*, *keyrate*, *width* and *height* parameters are only relevant for video streaming.
- The *action* and *speed* parameters are only relevant for recorded video streaming.
- The initial HTTP response headers will include some useful information regarding the media stream. For video streams, they include video height and width, if the video stream has audio associated with it (recorded), the media type (recorded), the Connection ID (used for Digital PTZ), the average time per frame which is used to calculate the frame rate and a cookie that should be recorded and sent with VCR control requests as well as Digital PTZ requests. For audio streams they include the number of audio channels, the bits per sample and the sample rate.
- Not every multipart section will include media when streaming recorded media, recipients should always check the Content-Type and / or Content-Length.
- Multipart sections that do include media will include additional headers such as Pre-roll (video), Sync-Point (video), various timestamps, the camera status (live video) and whether or not the stream is at the end (recorded).
- There is another type of multipart section that MAY be interlaced with media sections to notify a recipient when the server has to switch AVI files when streaming recorded media. During this process, it is possible that the media type or presence of audio has changed. This should be handled by client code.

Examples

Live Video - Request

```
GET /cameras/1/media
```

Response

```
HTTP/1.1 200 OK\r\n
Date: Tue, 23 Jun 2015 20:44:39 GMT\r\n
Connection: close\r\n
Cache-Control: public, no-store\r\n
Expires: Tue, 23 Jun 2016 20:44:39 GMT\r\n
Content-Type: multipart/x-mixed-replace;boundary=myboundary\r\n
Connection-ID: 100\r\n
Width: 1400\r\n
Height: 1050\r\n
```

```
\r\n
--myboundary\r\n
Content-Type: image/jpeg\r\n
Sync-Point: yes\r\n
Pre-roll: no\r\n
Media-Start: invalid\r\n
Media-End: invalid\r\n
Stream-Start: 131184\r\n
Stream-End: 464517\r\n
Content-Length: 99609\r\n
End-of-Stream: no\r\n
Camera-Status: 0\r\n
\r\n
{video-frame}
\r\n
```

Live Video with Parameters - Request

```
GET /cameras/1/media?width=640&height=480&fps=10&quality=75
```

Live Audio - Request

```
GET /cameras/1/media?accept=audio/basic
```

Live JPEG Snapshot - Request

```
GET /cameras/1/media?accept=image/jpeg
```

Recorded Video - Request

```
GET /cameras/1/media?start=2015-06-23T21:44:12Z&stop=2015-06-23T21:46:12Z
```

Response

```
HTTP/1.1 200 OK\r\n
Date: Thu, 25 Jun 2015 20:36:20 GMT\r\n
Connection: keep-alive\r\n
Cache-Control: no-cache, no-store\r\n
Expires: -1\r\n
Content-Type: multipart/x-mixed-replace;boundary=myboundary\r\n
Set-Cookie: stream=38745961; version=1\r\n
Has-Audio: true\r\n
Connection-ID: 191\r\n
Width: 1266\r\n
Height: 950\r\n
\r\n
--myboundary\r\n
Content-Type: image/jpeg\r\n
Sync-Point: yes\r\n
```

```
Pre-roll: no\r\n
Media-Start: invalid\r\n
Media-End: invalid\r\n
Stream-Start: 855254\r\n
Stream-End: 1188587\r\n
Content-Length: 99734\r\n
X-Date: 2015-06-25T20:11:55.199Z\r\n
End-of-Stream: no\r\n
\r\n
{video-frame}
\r\n
```

Recorded Video with Start Position - Request

```
GET
/cameras/1/media?start=2015-06-23T21:44:12Z&stop=2015-06-23T21:46:12Z&positi
on=2015-06-23T21:45:12Z
```

Recorded Audio - Request

Note that the cookie that was sent with the video stream was sent with the audio request.

```
GET
/cameras/1/media?accept=audio/basic&start=2015-06-25T20:11:55.199Z&stop=2015
-06-25T20:21:55.199Z HTTP/1.0\r\n
Host: 192.168.103.58:4502\r\n
User-Agent: CompleteView Web Client 4.5.1.10\r\n
Authorization: Basic YWRtaW46\r\n
Cookie: stream=38745961\r\n
\r\n
```

Response

```
HTTP/1.1 200 OK\r\n
Date: Thu, 25 Jun 2015 20:39:27 GMT\r\n
Connection: keep-alive\r\n
Cache-Control: no-cache, no-store\r\n
Expires: -1\r\n
Content-Type: multipart/x-mixed-replace;boundary=myboundary\r\n
Audio-Num-Channels: 1\r\n
Audio-Sample-Rate: 8000\r\n
Audio-Bits-Per-Sample: 8\r\n
Set-Cookie: stream=38745961; version=1\r\n
\r\n
--myboundary
Content-Type: audio/basic\r\n
Media-Start: invalid\r\n
Media-End: invalid\r\n
Stream-Start: -163\r\n
Stream-End: 308587\r\n
Content-Length: 247\r\n
End-of-Stream: no\r\n
```

```
\r\n  
{audio-samples}  
\r\n
```

Recorded JPEG Snapshot - Request

```
GET /cameras/1/media?accept=image/jpeg&start=2015-06-23T21:44:12Z
```

Recorded Clip Seek - Request

The appropriate cookie will need to be included in this request.

```
GET /cameras/1/media?position=2015-06-23T21:44:12Z
```

Recorded Clip Set Speed - Request

The appropriate cookie will need to be included in this request.

```
GET /cameras/1/media?speed=2.0
```

Recorded Clip Pause - Request

The appropriate cookie will need to be included in this request.

```
GET /cameras/1/media?action=pause
```

GET /cameras/:camera-id/thumbnails

Gets a multipart stream of JPEG frames from a recorded time period with frames being grabbed from specified intervals.

Resource Information

Response formats	Multipart content with either JPEG frames
Requires authentication?	Yes
Permissions required	Playback

Parameters

Name	Required	Description
camera-id	Yes	Integral camera ID, from 1 to the number of cameras on this server. This parameter should be a part of the request URI path and not the list of query parameters.
start	Yes	UTC start date / time string as defined in RFC 3339 of the recorded clip to get thumbnails for.
stop	No	Default: start date plus 60 seconds UTC stop date / time string as defined in RFC 3339 of the recorded clip to get thumbnails for.

Name	Required	Description
interval	No	Default: 10 seconds Supported values: 1 - 864000000000 The interval between thumbnails in 100 nanosecond units.
quality	No	Default: 100 Supported values: 0 - 100 Lower values corresponds to higher image compression and vice versa.
width	No	Default: -1 (send the native resolution) Supported values: 16 to 4096 The pixel width of the image to send. The image that is sent may not be exactly what is requested as the image is locked to its native aspect ratio. If <i>width</i> is specified, <i>height</i> must also be specified.
height	No	Default: -1 (send the native resolution) Supported values: 16 to 4096 The pixel height of the image to send. The image that is sent may not be exactly what is requested as the image is locked to its native aspect ratio. If <i>height</i> is specified, <i>width</i> must also be specified.

Notes

- *stop* must be later in time than *start*.
- Multipart sections that do include media will include additional headers such as Pre-roll (video), Sync-Point (video), various timestamps, the camera status (live video) and whether or not the stream is at the end (recorded).

Examples

Thumbnails - Request

```
GET /cameras/1/thumbnails?start=2015-06-23T12:10:00Z
```

Response

```
HTTP/1.1 200 OK\r\n
Date: Tue, 23 Jun 2015 20:44:39 GMT\r\n
Connection: close\r\n
Cache-Control: public, no-store\r\n
Expires: Tue, 23 Jun 2016 20:44:39 GMT\r\n
Content-Type: multipart/x-mixed-replace;boundary=myboundary\r\n
\r\n
--myboundary\r\n
Content-Type: image/jpeg\r\n
Sync-Point: yes\r\n
Pre-roll: no\r\n
Media-Start: invalid\r\n
Media-End: invalid\r\n
Stream-Start: 131184\r\n
Stream-End: 464517\r\n
Content-Length: 99609\r\n
X-Date: 2015-06-25T20:11:55.199Z\r\n
```

```
End-of-Stream: no\r\n
Camera-Status: 0\r\n
\r\n
{video-frame}
\r\n
```

Thumbnails with Parameters - Request

```
GET
/cameras/1/thumbnails?start=2015-06-23T12:10:00Z&width=640&height=480&interval=600000000&quality=75
```

PUT /cameras/:camera-id/position

Controls either digital or physical PTZ on either a live or a recorded video stream.

Resource Information

Response formats	HTML
Requires authentication?	Yes
Permissions required	PTZ

Parameters

Name	Required	Description
camera-id	Yes	Integral camera ID, from 1 to the number of cameras on this server. This parameter should be a part of the request URI path and not the list of query parameters.
x	No	Default: Not set Supported values: -1000 to 1000 Specifies the speed at which to move along the x axis, panning the camera to the left or right (See signedness semantics below). 0 specifies a stop of all motion.
y	No	Default: Not set Supported values: -1000 to 1000 Specifies the speed at which to move along the y axis, tilting the camera up or down (See signedness semantics below). 0 specifies a stop of all motion.
z	No	Default: Not set Supported values: -1000 to 1000 Specifies the speed at which to move along the z axis, zooming the camera in or out (See signedness semantics below). 0 specifies a stop of all motion.
f	No	Default: Not set Supported values: -1000 to 1000 Specifies the speed at which to focus the camera, focusing the camera near or far (See signedness semantics below). 0 specifies a stop of all motion.

Name	Required	Description
i	No	Default: Not set Supported values: -1000 to 1000 Specifies the speed at which to adjust the camera's iris, closing or opening the iris (See signedness semantics below). 0 specifies a stop of all motion.
dptz_id	No	Default: Not set Supported values: unsigned integers Specifies the connection ID of the video stream that you would like to perform Digital PTZ on. This ID is returned from video media requests in the Connection-ID HTTP header. This parameter is required if the camera is configured for Digital PTZ or if you are trying to manipulate a recorded stream.

Signedness semantics

	+	-
Pan	right	left
Tilt	down	up
Zoom	in	out
Focus	near	far
Iris	close	open

Notes

- If no parameters are specified, this will result in an error, as there is nothing useful to do.
- The Cookie stream-id received from a recorded video media request is required if Digital PTZ is being controlled on a recorded video stream.

Examples

Pan Left and Tilt Up - Request

```
PUT /cameras/1/position?x=-500&y=-500
```

Zoom In - Request

```
PUT /cameras/1/position?z=700
```

Zoom In (Digital PTZ) - Request

```
PUT /cameras/1/position?z=700&dptz_id=67
```

Stop all Movement - Request

```
PUT /cameras/1/position?x=0&y=0&z=0&f=0&i=0
```

PUT /cameras/:camera-id/position/:ptz-id

Controls either digital or physical PTZ presets on either a live or a recorded video stream.

Resource Information

Response formats	HTML
Requires authentication?	Yes
Permissions required	PTZ

Parameters

Name	Required	Description
camera-id	Yes	Integral camera ID, from 1 to the number of cameras on this server. This parameter should be a part of the request URI path and not the list of query parameters.
ptz-id	Yes	Integral preset ID, from 1 to the number of presets the camera supports. For recorded video streams or Digital PTZ cameras, there are only 10 implicit preset positions that are hardcoded on the server side. This parameter should be a part of the request URI path and not the list of query parameters.
dptz_id	No	Default: Not set Supported values: unsigned integers Specifies the connection ID of the video stream that you would like to perform Digital PTZ on. This ID is returned from video media requests in the Connection-ID HTTP header. This parameter is required if the camera is configured for Digital PTZ or if you are trying to manipulate a recorded stream.
current	No	Including this parameter indicates that you would like to set this preset to the camera's current position. There is no value associated with this parameter.
preset_name	No	Supported values: Both empty and non empty strings Including this parameter indicates that you would like to set the preset's name to the supplied value.

Notes

- The Cookie stream-id received from a recorded video media request is required if Digital PTZ is being controlled on a recorded video stream.

Examples

Go to Preset - Request

```
PUT /cameras/1/position/7
```

Go to Preset (Digital PTZ) - Request

```
PUT /cameras/1/position/7?dptz_id=67
```

Set Preset - Request

```
PUT /cameras/1/position/11?current
```

Set Preset and Preset Name - Request

```
PUT /cameras/1/position/11?current&preset_name=Main%20Entrance
```

CompleteView REST Users Interface

The CompleteView [REST](#) Users interface is very similar to the Cameras interface, in that it follows a similar syntax and requires the same authentication. The users interface was built to give insight into the Active Directory users, Active Directory groups and CompleteView groups that the CompleteView server has access to.

GET /users/groups

Gets the CompleteView groups configured on the server.

Resource Information

Response formats	JSON
Requires authentication?	Yes
Permissions required	Admin user

Parameters

Name	Required	Description
user	Yes	Name of a CompleteView user to see the groups that user is a member of.

Examples

Request

```
GET /users/groups?user=admin
```

Response

```
{
  "users": [
    {
      "user": "Administrators"
    },
    {
      "user": "Test Group"
    }
  ]
}
```

```
}
```

GET /users/activedirusers

Gets the Active Directory users that CompleteView has access to.

Resource Information

Response formats	JSON
Requires authentication?	Yes
Permissions required	Admin user

Examples

Request

```
GET /users/activedirusers
```

Response

```
{
  "users": [
    {
      "user": "Guest"
    },
    {
      "user": "Administrator"
    },
    {
      "user": "VM"
    },
    {
      "user": "GuestAcct"
    },
    {
      "user": "demo"
    },
    {
      "user": "UPS"
    },
    {
      "user": "temp"
    }
  ]
}
```

GET /users/activedirgroups

Gets the Active Directory users that CompleteView has access to.

Resource Information

Response formats	JSON
Requires authentication?	Yes
Permissions required	Admin user

Parameters

Name	Required	Description
user	No	Name of an Active Directory user to see the groups that user is a member of.

Notes

- If nested group support is enabled in the server configuration, this query could potentially take several minutes to complete.

Examples

All Groups - Request

```
GET /users/activedirgroups
```

Response

```
{
  "users": [
    {
      "user": "Domain Computers"
    },
    {
      "user": "Domain Controllers"
    },
    {
      "user": "Guests"
    },
    {
      "user": "Domain Users"
    },
    {
      "user": "Domain Admins"
    },
    {
      "user": "Users"
    }
  ]
}
```

```
}  
]  
}
```

User Specific Groups - Request

```
GET /users/activedirgroups?user=test
```

Response

```
{  
  "users": [  
    {  
      "user": "Domain Users"  
    },  
    {  
      "user": "Domain Admins"  
    },  
    {  
      "user": "Users"  
    }  
  ]  
}
```

From:
<https://wiki.salientsys.com/> - **Salient's Wiki**

Permanent link:
https://wiki.salientsys.com/engineering:development:apis:cv_cameras_rest

Last update: **2017/01/30 12:15**

